

PAC55XX Family User Guide

Power Application Controller[™]

Multi-Mode Power Manager[™]
Configurable Analog Front End[™]
Application Specific Power Drivers[™]
ARM[®] Cortex[™]-M4 Controller Core



TABLE OF CONTENTS

| | | |
|-------|---------------------------------------|----|
| 1 | OVERVIEW | 49 |
| 2 | STYLE AND FORMATTING CONVENTIONS..... | 50 |
| 2.1 | Number Representation..... | 50 |
| 2.2 | Formatting Styles..... | 50 |
| 3 | MEMORY AND REGISTER MAP..... | 51 |
| 3.1 | Memory Map..... | 51 |
| 3.2 | FLASH Memory Map | 52 |
| 3.3 | INFO-1 Register Map..... | 56 |
| 3.4 | INFO-2 Register Map..... | 57 |
| 3.5 | INFO-3 Register Map..... | 59 |
| 3.6 | Peripheral Register Maps..... | 59 |
| 4 | SCC..... | 77 |
| 4.1 | Overview..... | 77 |
| 4.2 | Features | 77 |
| 4.3 | System Block Diagram..... | 78 |
| 4.4 | Clock Sources..... | 79 |
| 4.4.1 | ROSCCLK | 79 |
| 4.4.2 | CLKREF | 79 |
| 4.4.3 | XTALCLK..... | 79 |
| 4.4.4 | EXTCLK..... | 79 |
| 4.5 | System Clocks | 80 |
| 4.5.1 | FRCLK..... | 80 |
| 4.5.2 | PLLCLK..... | 80 |
| 4.5.3 | SCLK..... | 80 |
| 4.5.4 | PCLK..... | 80 |
| 4.5.5 | ACLK..... | 81 |
| 4.5.6 | HCLK..... | 81 |
| 4.6 | PLL Configuration | 83 |
| 4.7 | Peripheral Clock Selection..... | 84 |

| | | |
|---------|--|-----|
| 4.8 | Low-Power Clock Gating..... | 85 |
| 4.9 | Clock Failure Detection..... | 86 |
| 4.10 | USART Mode..... | 86 |
| 4.11 | GPIO Drive Strength..... | 86 |
| 4.12 | GPIO Pull-up an Pull-down Configuration..... | 87 |
| 4.13 | Digital Peripheral MUX..... | 87 |
| 4.14 | Interrupt Vector Table..... | 87 |
| 4.15 | Register Summary..... | 90 |
| 4.16 | Register Detail..... | 91 |
| 4.16.1 | CCSCTL..... | 91 |
| 4.16.2 | CCSPLLCTL..... | 94 |
| 4.16.3 | CCSROSCTRIM..... | 94 |
| 4.16.4 | PAMUXSEL..... | 95 |
| 4.16.5 | PBMUXSEL..... | 97 |
| 4.16.6 | PCMUXSEL..... | 99 |
| 4.16.7 | PDMUXSEL..... | 101 |
| 4.16.8 | PEMUXSEL..... | 103 |
| 4.16.9 | PFMUXSEL..... | 105 |
| 4.16.10 | PGMUXSEL..... | 107 |
| 4.16.11 | PAPUEN..... | 109 |
| 4.16.12 | PBPUEN..... | 110 |
| 4.16.13 | PCPUEN..... | 111 |
| 4.16.14 | PDPHEN..... | 112 |
| 4.16.15 | PEPUEN..... | 113 |
| 4.16.16 | PFPUEN..... | 114 |
| 4.16.17 | PGPUEN..... | 115 |
| 4.16.18 | PAPDEN..... | 116 |
| 4.16.19 | PBPDEN..... | 117 |
| 4.16.20 | PCPDEN..... | 118 |
| 4.16.21 | PDPDEN..... | 119 |
| 4.16.22 | PEPDEN..... | 120 |
| 4.16.23 | PFPDEN..... | 121 |

| | | |
|---------|--|-----|
| 4.16.24 | PGPDEN | 122 |
| 4.16.25 | PADS | 123 |
| 4.16.26 | PBDS | 125 |
| 4.16.27 | PCDS | 127 |
| 4.16.28 | PDDS | 129 |
| 4.16.29 | PEDS | 131 |
| 4.16.30 | PFDS | 133 |
| 4.16.31 | PGDS..... | 135 |
| 5 | MEMCTL..... | 137 |
| 5.1 | Overview..... | 137 |
| 5.2 | Features | 137 |
| 5.3 | System Block Diagram..... | 138 |
| 5.4 | Functional Description | 139 |
| 5.4.1 | Clocking..... | 139 |
| 5.4.2 | Low-Power Operation | 141 |
| 5.4.3 | Invalid Memory Accesses | 141 |
| 5.4.4 | SRAM | 141 |
| 5.4.5 | FLASH Read Cache | 142 |
| 5.4.6 | Erasing a FLASH Page..... | 142 |
| 5.4.7 | Erase all FLASH Pages | 143 |
| 5.4.8 | Erase INFO-3..... | 143 |
| 5.4.9 | Writing FLASH | 143 |
| 5.5 | Code Protection | 144 |
| 5.5.1 | Code Protection Variables | 144 |
| 5.5.2 | Code Security Levels | 146 |
| 5.5.3 | Level 0 – UNLOCKED | 146 |
| 5.5.4 | Level 1 – RW PROTECTION | 146 |
| 5.5.5 | Level 2 – SWD DISABLED | 147 |
| 5.5.6 | Level 3 – PERMANENT SWD DISABLED | 148 |
| 5.6 | Register Summary | 149 |
| 5.7 | Register Detail | 150 |
| 5.7.1 | MEMCTL | 150 |

| | | |
|-------|------------------------------------|-----|
| 5.7.2 | MEMSTATUS | 151 |
| 5.7.3 | FLASHLOCK | 152 |
| 5.7.4 | FLASHPAGE | 152 |
| 5.7.5 | SWDUNLOCK | 152 |
| 5.7.6 | FLASHERASE | 152 |
| 6 | DPM..... | 153 |
| 6.1 | Overview..... | 153 |
| 6.2 | Digital Peripheral Functions | 154 |
| 6.3 | Peripheral MUX Settings..... | 157 |
| 7 | WWDT | 159 |
| 7.1 | Overview..... | 159 |
| 7.2 | Features | 159 |
| 7.3 | System Block Diagram..... | 159 |
| 7.4 | Functional Description | 159 |
| 7.5 | Register Summary | 161 |
| 7.6 | Register Detail | 162 |
| 7.6.1 | WWDTCTL | 162 |
| 7.6.2 | WWDTLOAD | 162 |
| 7.6.3 | WWDTCTR..... | 163 |
| 7.6.4 | WWDTINTF | 163 |
| 7.6.5 | WWDTCLEAR | 163 |
| 7.6.6 | WWDTLOCK | 163 |
| 8 | RTC | 165 |
| 8.1 | Overview..... | 165 |
| 8.2 | Features | 165 |
| 8.3 | System Block Diagram..... | 165 |
| 8.4 | Functional Description | 166 |
| 8.4.1 | Overview..... | 166 |
| 8.5 | Register Summary | 167 |
| 8.6 | Register Detail | 168 |
| 8.6.1 | RTCCTL | 168 |
| 8.6.2 | RTCTIME..... | 169 |

| | | |
|--------|---------------------------------------|-----|
| 8.6.3 | RTCDATE..... | 169 |
| 8.6.4 | RTCTIMESET..... | 170 |
| 8.6.5 | RTCDATESET..... | 170 |
| 8.6.6 | RTCALARMSET..... | 171 |
| 9 | GPIOA..... | 172 |
| 9.1 | Overview..... | 172 |
| 9.2 | GPIOA Block Diagram..... | 172 |
| 9.3 | Features..... | 173 |
| 9.4 | Functional Description..... | 174 |
| 9.4.1 | IO Mode..... | 174 |
| 9.4.2 | Digital Output..... | 174 |
| 9.4.3 | Digital Input..... | 174 |
| 9.4.4 | GPIOA Interrupts..... | 174 |
| 9.4.5 | GPIOA De-bouncing..... | 175 |
| 9.4.6 | GPIOA Drive Strength..... | 175 |
| 9.4.7 | GPIOA Schmitt Trigger..... | 176 |
| 9.4.8 | GPIOA Weak Pull-up and Pull-down..... | 176 |
| 9.5 | Peripheral IO Mapping..... | 177 |
| 9.6 | Register Summary..... | 178 |
| 9.7 | Register Detail..... | 179 |
| 9.7.1 | GPIOAMODE..... | 179 |
| 9.7.2 | GPIOAOUTMASK..... | 180 |
| 9.7.3 | GPIOAOUT..... | 181 |
| 9.7.4 | GPIOAIN..... | 182 |
| 9.7.5 | GPIOAINTEN..... | 183 |
| 9.7.6 | GPIOAINTFLAGRAW..... | 184 |
| 9.7.7 | GPIOAINTFLAGMASKED..... | 185 |
| 9.7.8 | GPIOAINTCLEAR..... | 186 |
| 9.7.9 | GPIOAINTTYPE..... | 187 |
| 9.7.10 | GPIOAINTCFG..... | 188 |
| 9.7.11 | GPIOAINTEDGEBOTH..... | 190 |
| 9.7.12 | GPIOADEBOUNCE..... | 191 |

| | | |
|---------|--|-----|
| 9.7.13 | GPIOADSET | 192 |
| 9.7.14 | GPIOADOCLEAR | 193 |
| 10 | GPIOB | 194 |
| 10.1 | Overview | 194 |
| 10.2 | GPIOB Block Diagram | 194 |
| 10.3 | Features | 195 |
| 10.4 | Functional Description | 196 |
| 10.4.1 | IO Mode | 196 |
| 10.4.2 | Digital Output | 196 |
| 10.4.3 | Digital Input | 196 |
| 10.4.4 | GPIOB Interrupts | 196 |
| 10.4.5 | GPIOB De-bouncing | 197 |
| 10.4.6 | GPIOB Drive Strength | 197 |
| 10.4.7 | GPIOB Schmitt Trigger | 198 |
| 10.4.8 | GPIOB Weak Pull-up and Pull-down | 198 |
| 10.5 | Peripheral IO Mapping | 199 |
| 10.6 | Register Summary | 200 |
| 10.7 | Register Detail | 201 |
| 10.7.1 | GPIOBMODE | 201 |
| 10.7.2 | GPIOBOUTMASK | 202 |
| 10.7.3 | GPIOBOUT | 203 |
| 10.7.4 | GPIOBIN | 204 |
| 10.7.5 | GPIOBINTEN | 205 |
| 10.7.6 | GPIOBINTFLAGRAW | 206 |
| 10.7.7 | GPIOBINTFLAGMASKED | 207 |
| 10.7.8 | GPIOBINTCLEAR | 208 |
| 10.7.9 | GPIOBINTTYPE | 209 |
| 10.7.10 | GPIOBINTCFG | 210 |
| 10.7.11 | GPIOBINTEDGEBOTH | 212 |
| 10.7.12 | GPIOBDEBOUNCE | 213 |
| 10.7.13 | GPIOBDOSET | 214 |
| 10.7.14 | GPIOBDOCLEAR | 215 |

| | | |
|---------|---------------------------------------|-----|
| 11 | GPIOC..... | 216 |
| 11.1 | Overview..... | 216 |
| 11.2 | GPIOC Block Diagram | 216 |
| 11.3 | Features | 217 |
| 11.4 | Functional Description | 218 |
| 11.4.1 | IO Mode..... | 218 |
| 11.4.2 | Digital Output..... | 218 |
| 11.4.3 | Digital Input..... | 218 |
| 11.4.4 | GPIOC Interrupts | 218 |
| 11.4.5 | GPIOC De-bouncing..... | 219 |
| 11.4.6 | GPIOC Drive Strength | 219 |
| 11.4.7 | GPIOC Schmitt Trigger..... | 220 |
| 11.4.8 | GPIOC Weak Pull-up and Pull-down..... | 220 |
| 11.5 | Peripheral IO Mapping..... | 221 |
| 11.6 | Register Summary | 222 |
| 11.7 | Register Detail | 223 |
| 11.7.1 | GPIOCMODE | 223 |
| 11.7.2 | GPIOCOUTMASK | 224 |
| 11.7.3 | GPIOCOUT | 225 |
| 11.7.4 | GPIOCIN | 226 |
| 11.7.5 | GPIOCINTEN | 227 |
| 11.7.6 | GPIOCINTFLAGRAW..... | 228 |
| 11.7.7 | GPIOCINTFLAGMASKED | 229 |
| 11.7.8 | GPIOCINTCLEAR | 230 |
| 11.7.9 | GPIOCINTTYPE | 231 |
| 11.7.10 | GPIOCINTCFG | 232 |
| 11.7.11 | GPIOCINTEDGEBOTH..... | 234 |
| 11.7.12 | GPIOCDEBOUNCE | 235 |
| 11.7.13 | GPIOCDOSET | 236 |
| 11.7.14 | GPIOCDOCLEAR | 237 |
| 12 | GPIOD..... | 238 |
| 12.1 | Overview..... | 238 |

| | | |
|---------|---------------------------------------|-----|
| 12.2 | GPIOD Block Diagram | 238 |
| 12.3 | Features | 239 |
| 12.4 | Functional Description | 240 |
| 12.4.1 | IO Mode..... | 240 |
| 12.4.2 | Digital Output..... | 240 |
| 12.4.3 | Digital Input..... | 240 |
| 12.4.4 | Analog Input | 240 |
| 12.4.5 | GPIOD Interrupts | 241 |
| 12.4.6 | GPIOD De-bouncing..... | 241 |
| 12.4.7 | GPIOD Drive Strength | 242 |
| 12.4.8 | GPIOD Schmitt Trigger | 242 |
| 12.4.9 | GPIOD Weak Pull-up and Pull-down..... | 242 |
| 12.5 | Peripheral IO Mapping | 244 |
| 12.6 | Register Summary | 245 |
| 12.7 | Register Detail | 246 |
| 12.7.1 | GPIODMODE | 246 |
| 12.7.2 | GPIODOUTMASK | 247 |
| 12.7.3 | GPIODOUT | 248 |
| 12.7.4 | GPIODIN | 249 |
| 12.7.5 | GPIODINTEN | 250 |
| 12.7.6 | GPIODINTFLAGRAW..... | 251 |
| 12.7.7 | GPIODINTFLAGMASKED | 252 |
| 12.7.8 | GPIODINTCLEAR | 253 |
| 12.7.9 | GPIODINTTYPE | 254 |
| 12.7.10 | GPIODINTCFG | 255 |
| 12.7.11 | GPIODINTEDGEBOTH..... | 257 |
| 12.7.12 | GPIODDEBOUNCE | 258 |
| 12.7.13 | GPIODDOSET | 259 |
| 12.7.14 | GPIODDOCLEAR | 260 |
| 13 | GPIOE..... | 261 |
| 13.1 | Overview..... | 261 |
| 13.2 | GPIOE Block Diagram | 261 |

| | | |
|---------|---------------------------------------|-----|
| 13.3 | Features | 262 |
| 13.4 | Functional Description | 263 |
| 13.4.1 | IO Mode..... | 263 |
| 13.4.2 | Digital Output..... | 263 |
| 13.4.3 | Digital Input..... | 263 |
| 13.4.4 | GPIOE Interrupts | 263 |
| 13.4.5 | GPIOE De-bouncing..... | 264 |
| 13.4.6 | GPIOE Drive Strength..... | 264 |
| 13.4.7 | GPIOE Schmitt Trigger | 265 |
| 13.4.8 | GPIOE Weak Pull-up and Pull-down..... | 265 |
| 13.5 | Peripheral IO Mapping..... | 266 |
| 13.6 | Register Summary | 267 |
| 13.7 | Register Detail | 268 |
| 13.7.1 | GPIOEMODE..... | 268 |
| 13.7.2 | GPIOEOUTMASK..... | 269 |
| 13.7.3 | GPIOEOUT..... | 270 |
| 13.7.4 | GPIOEIN..... | 271 |
| 13.7.5 | GPIOEINTEN..... | 272 |
| 13.7.6 | GPIOEINTFLAGRAW | 273 |
| 13.7.7 | GPIOEINTFLAGMASKED | 274 |
| 13.7.8 | GPIOEINTCLEAR..... | 275 |
| 13.7.9 | GPIOEINTTYPE | 276 |
| 13.7.10 | GPIOEINTCFG | 277 |
| 13.7.11 | GPIOEINTEDGEBOTH..... | 279 |
| 13.7.12 | GPIOEDEBOUNCE..... | 280 |
| 13.7.13 | GPIOEDOSET | 281 |
| 13.7.14 | GPIOEDOCLEAR..... | 282 |
| 14 | GPIOF..... | 283 |
| 14.1 | Overview..... | 283 |
| 14.2 | GPIOF Block Diagram | 283 |
| 14.3 | Features | 284 |
| 14.4 | Functional Description | 285 |

| | | |
|---------|--|-----|
| 14.4.1 | IO Mode..... | 285 |
| 14.4.2 | Digital Output..... | 285 |
| 14.4.3 | Digital Input..... | 285 |
| 14.4.4 | Analog Input | 285 |
| 14.4.5 | GPIOF Interrupts | 286 |
| 14.4.6 | GPIOF De-bouncing | 286 |
| 14.4.7 | GPIOF Drive Strength..... | 287 |
| 14.4.8 | GPIOF Schmitt Trigger | 287 |
| 14.4.9 | GPIOF Weak Pull-up and Pull-down | 287 |
| 14.5 | Peripheral IO Mapping | 289 |
| 14.6 | Register Summary | 290 |
| 14.7 | Register Detail | 291 |
| 14.7.1 | GPIOFMODE..... | 291 |
| 14.7.2 | GPIOFOUTMASK..... | 292 |
| 14.7.3 | GPIOFOUT | 293 |
| 14.7.4 | GPIOFIN..... | 294 |
| 14.7.5 | GPIOFINTEN..... | 295 |
| 14.7.6 | GPIOFINTFLAGRAW | 296 |
| 14.7.7 | GPIOFINTFLAGMASKED..... | 297 |
| 14.7.8 | GPIOFINTCLEAR..... | 298 |
| 14.7.9 | GPIOFINTTYPE | 299 |
| 14.7.10 | GPIOFINTCFG..... | 300 |
| 14.7.11 | GPIOFINTEDGEBOTH | 302 |
| 14.7.12 | GPIOFDEBOUNCE..... | 303 |
| 14.7.13 | GPIOFDOSET..... | 304 |
| 14.7.14 | GPIOFDOCLEAR..... | 305 |
| 15 | GPIOG | 306 |
| 15.1 | Overview..... | 306 |
| 15.2 | GPIOG Block Diagram..... | 306 |
| 15.3 | Features | 307 |
| 15.4 | Functional Description | 308 |
| 15.4.1 | IO Mode..... | 308 |

| | | |
|---------|--|-----|
| 15.4.2 | Digital Output | 308 |
| 15.4.3 | Digital Input..... | 308 |
| 15.4.4 | Analog Input | 308 |
| 15.4.5 | GPIOG Interrupts..... | 309 |
| 15.4.6 | GPIOG De-bouncing..... | 309 |
| 15.4.7 | GPIOG Drive Strength | 310 |
| 15.4.8 | GPIOG Schmitt Trigger..... | 310 |
| 15.4.9 | GPIOG Weak Pull-up and Pull-down | 310 |
| 15.5 | Peripheral IO Mapping | 312 |
| 15.6 | Register Summary | 313 |
| 15.7 | Register Detail | 314 |
| 15.7.1 | GPIOGMODE | 314 |
| 15.7.2 | GPIOGOUTMASK | 315 |
| 15.7.3 | GPIOGOUT | 316 |
| 15.7.4 | GPIOGIN | 317 |
| 15.7.5 | GPIOGINTEN | 318 |
| 15.7.6 | GPIOGINTFLAGRAW..... | 319 |
| 15.7.7 | GPIOGINTFLAGMASKED | 320 |
| 15.7.8 | GPIOGINTCLEAR | 321 |
| 15.7.9 | GPIOGINTTYPE | 322 |
| 15.7.10 | GPIOGINTCFG | 323 |
| 15.7.11 | GPIOGINTEDGEBOTH..... | 325 |
| 15.7.12 | GPIOGDEBOUNCE | 326 |
| 15.7.13 | GPIOGDOSET | 327 |
| 15.7.14 | GPIOGDOCLEAR | 328 |
| 16 | GPTIMERA | 329 |
| 16.1 | Overview..... | 329 |
| 16.2 | Features | 329 |
| 16.3 | System Block Diagram..... | 329 |
| 16.4 | Functional Description | 330 |
| 16.5 | Register Summary | 331 |
| 16.6 | Register Detail | 331 |

| | | |
|---------|------------------------------------|-----|
| 16.6.1 | GPTACTL | 331 |
| 16.6.2 | GPTACTR | 331 |
| 17 | GPTIMERB | 332 |
| 17.1 | Overview..... | 332 |
| 17.2 | Features | 332 |
| 17.3 | System Block Diagram..... | 332 |
| 17.4 | Functional Description | 333 |
| 17.5 | Register Summary | 334 |
| 17.6 | Register Detail | 334 |
| 17.6.1 | GPTBCTL..... | 334 |
| 17.6.2 | GPTBCTR | 334 |
| 18 | PWM TIMER A..... | 335 |
| 18.1 | Overview..... | 335 |
| 18.2 | Timer A Block Diagram | 336 |
| 18.3 | Features | 337 |
| 18.4 | Functional Description | 339 |
| 18.4.1 | Timer Clock Structure | 339 |
| 18.4.2 | Timer Counter..... | 339 |
| 18.4.3 | Up Mode..... | 340 |
| 18.4.4 | Up/Down Mode | 340 |
| 18.4.5 | Up/Down Asymmetric mode..... | 341 |
| 18.4.6 | Up Mode with Phase Delay..... | 342 |
| 18.4.7 | Timer Synchronization | 342 |
| 18.4.8 | Base Timer TAPRD Latching | 343 |
| 18.4.9 | CCR Timer Latching | 344 |
| 18.4.10 | CCR Capture Period/Duty Mode | 345 |
| 18.4.11 | Timer Whole Latching | 346 |
| 18.4.12 | Inverting CCR PWM Output | 346 |
| 18.4.13 | Base Timer Interrupts..... | 346 |
| 18.4.14 | CCR Compare Interrupts..... | 346 |
| 18.4.15 | CCR Capture Interrupts..... | 347 |
| 18.4.16 | Timer IRQ signal | 347 |

| | | |
|---------|----------------------------------|-----|
| 18.4.17 | Skipping CCR Interrupts..... | 348 |
| 18.4.18 | Timer ADC Triggers | 349 |
| 18.4.19 | Dead-Time Generators (DTG) | 350 |
| 18.4.20 | Timer A DTG Output | 352 |
| 18.5 | QEP | 353 |
| 18.6 | Peripheral IO Mapping | 355 |
| 18.7 | Timer A Register Summary | 356 |
| 18.8 | Register Detail | 357 |
| 18.8.1 | TACTL | 357 |
| 18.8.2 | TAINT | 358 |
| 18.8.3 | TAPRD | 358 |
| 18.8.4 | TACTR..... | 359 |
| 18.8.5 | TAQEPCTL..... | 360 |
| 18.8.6 | TACCTL0..... | 362 |
| 18.8.7 | TACCTR0..... | 363 |
| 18.8.8 | TACCTL1..... | 364 |
| 18.8.9 | TACCTR1 | 365 |
| 18.8.10 | TACCTL2 | 366 |
| 18.8.11 | TACCTR2 | 367 |
| 18.8.12 | TACCTL3 | 368 |
| 18.8.13 | TACCTR3 | 369 |
| 18.8.14 | TACCTL4 | 370 |
| 18.8.15 | TACCTR4 | 371 |
| 18.8.16 | TACCTL5 | 372 |
| 18.8.17 | TACCTR5 | 373 |
| 18.8.18 | TACCTL6 | 374 |
| 18.8.19 | TACCTR6 | 375 |
| 18.8.20 | TACCTL7 | 376 |
| 18.8.21 | TACCTR7 | 377 |
| 18.8.22 | TADTGCTL0 | 378 |
| 18.8.23 | TADTGCTL1 | 378 |
| 18.8.24 | TADTGCTL2 | 379 |

| | | |
|---------|------------------------------------|-----|
| 18.8.25 | TADTGCTL3 | 379 |
| 19 | PWM TIMER B | 380 |
| 19.1 | Overview | 380 |
| 19.2 | Timer B Block Diagram | 381 |
| 19.3 | Features | 382 |
| 19.4 | Functional Description | 384 |
| 19.4.1 | Timer Clock Structure | 384 |
| 19.4.2 | Timer Counter | 384 |
| 19.4.3 | Up Mode | 385 |
| 19.4.4 | Up/Down Mode | 385 |
| 19.4.5 | Up/Down Asymmetric mode | 386 |
| 19.4.6 | Up Mode with Phase Delay | 387 |
| 19.4.7 | Timer Synchronization | 387 |
| 19.4.8 | Base Timer TAPRD Latching | 388 |
| 19.4.9 | CCR Timer Latching | 389 |
| 19.4.10 | CCR Capture Period/Duty Mode | 390 |
| 19.4.11 | Timer Whole Latching | 391 |
| 19.4.12 | Inverting CCR PWM Output | 391 |
| 19.4.13 | Base Timer Interrupts | 391 |
| 19.4.14 | CCR Compare Interrupts | 391 |
| 19.4.15 | CCR Capture Interrupts | 392 |
| 19.4.16 | Timer IRQ signal | 392 |
| 19.4.17 | Skipping CCR Interrupts | 393 |
| 19.4.18 | Timer ADC Triggers | 394 |
| 19.4.19 | Dead-Time Generators (DTG) | 395 |
| 19.4.20 | Timer B DTG Output | 397 |
| 19.5 | QEP | 398 |
| 19.6 | Peripheral IO Mapping | 400 |
| 19.7 | Timer B Register Summary | 401 |
| 19.8 | Register Detail | 402 |
| 19.8.1 | TBCTL | 402 |
| 19.8.2 | TBINT | 403 |

| | | |
|---------|------------------------------|-----|
| 19.8.3 | TBPRD | 403 |
| 19.8.4 | TBCTR..... | 404 |
| 19.8.5 | TBQEPCTL..... | 405 |
| 19.8.6 | TBCCTL0..... | 407 |
| 19.8.7 | TBCCTR0..... | 408 |
| 19.8.8 | TBCCTL1..... | 409 |
| 19.8.9 | TBCCTR1 | 410 |
| 19.8.10 | TBCCTL2 | 411 |
| 19.8.11 | TBCCTR2 | 412 |
| 19.8.12 | TBCCTL3 | 413 |
| 19.8.13 | TBCCTR3 | 414 |
| 19.8.14 | TBCCTL4 | 415 |
| 19.8.15 | TBCCTR4 | 416 |
| 19.8.16 | TBCCTL5 | 417 |
| 19.8.17 | TBCCTR5 | 418 |
| 19.8.18 | TBCCTL6 | 419 |
| 19.8.19 | TACCTR6 | 420 |
| 19.8.20 | TBCCTL7 | 421 |
| 19.8.21 | TBCCTR7 | 422 |
| 19.8.22 | TBDTGCTL0 | 423 |
| 19.8.23 | TBDTGCTL1 | 423 |
| 19.8.24 | TBDTGCTL2 | 424 |
| 19.8.25 | TBDTGCTL3 | 424 |
| 20 | PWM TIMER C..... | 425 |
| 20.1 | Overview..... | 425 |
| 20.2 | Timer C Block Diagram | 426 |
| 20.3 | Features | 427 |
| 20.4 | Functional Description | 429 |
| 20.4.1 | Timer Clock Structure | 429 |
| 20.4.2 | Timer Counter..... | 429 |
| 20.4.3 | Up Mode | 430 |
| 20.4.4 | Up/Down Mode | 430 |

| | | |
|---------|------------------------------------|-----|
| 20.4.5 | Up/Down Asymmetric mode..... | 431 |
| 20.4.6 | Up Mode with Phase Delay..... | 432 |
| 20.4.7 | Timer Synchronization | 432 |
| 20.4.8 | Base Timer TAPRD Latching | 433 |
| 20.4.9 | CCR Timer Latching | 434 |
| 20.4.10 | CCR Capture Period/Duty Mode | 435 |
| 20.4.11 | Timer Whole Latching | 436 |
| 20.4.12 | Inverting CCR PWM Output | 436 |
| 20.4.13 | Base Timer Interrupts..... | 436 |
| 20.4.14 | CCR Compare Interrupts..... | 436 |
| 20.4.15 | CCR Capture Interrupts..... | 437 |
| 20.4.16 | Timer IRQ signal | 437 |
| 20.4.17 | Skipping CCR Interrupts..... | 438 |
| 20.4.18 | Timer ADC Triggers | 439 |
| 20.4.19 | Dead-Time Generators (DTG)..... | 440 |
| 20.4.20 | Timer C DTG Output | 442 |
| 20.5 | QEP..... | 443 |
| 20.6 | Peripheral IO Mapping..... | 445 |
| 20.7 | Timer C Register Summary..... | 446 |
| 20.8 | Register Detail | 447 |
| 20.8.1 | TCCTL..... | 447 |
| 20.8.2 | TCINT | 448 |
| 20.8.3 | TCPRD | 449 |
| 20.8.4 | TCCTR | 449 |
| 20.8.5 | TCQEPCTL | 450 |
| 20.8.6 | TCCCTL0 | 452 |
| 20.8.7 | TCCCTR0..... | 453 |
| 20.8.8 | TCCCTL1 | 454 |
| 20.8.9 | TCCCTR1..... | 455 |
| 20.8.10 | TCCCTL2..... | 456 |
| 20.8.11 | TCCCTR2 | 457 |
| 20.8.12 | TCCCTL3..... | 458 |

| | | |
|---------|------------------------------------|-----|
| 20.8.13 | TCCCTR3 | 459 |
| 20.8.14 | TCCCTL4..... | 460 |
| 20.8.15 | TCCCTR4 | 461 |
| 20.8.16 | TCCCTL5..... | 462 |
| 20.8.17 | TCCCTR5 | 463 |
| 20.8.18 | TCCCTL6..... | 464 |
| 20.8.19 | TCCCTR6 | 465 |
| 20.8.20 | TCCCTL7..... | 466 |
| 20.8.21 | TCCCTR7 | 467 |
| 20.8.22 | TCDTGCTL0..... | 468 |
| 20.8.23 | TCDTGCTL1 | 468 |
| 20.8.24 | TCDTGCTL2..... | 469 |
| 20.8.25 | TCDTGCTL3..... | 469 |
| 21 | PWM TIMER D..... | 470 |
| 21.1 | Overview..... | 470 |
| 21.2 | Timer D Block Diagram..... | 471 |
| 21.3 | Features | 472 |
| 21.4 | Functional Description | 474 |
| 21.4.1 | Timer Clock Structure | 474 |
| 21.4.2 | Timer Counter..... | 474 |
| 21.4.3 | Up Mode | 475 |
| 21.4.4 | Up/Down Mode | 475 |
| 21.4.5 | Up/Down Asymmetric mode..... | 476 |
| 21.4.6 | Up Mode with Phase Delay..... | 477 |
| 21.4.7 | Timer Synchronization | 477 |
| 21.4.8 | Base Timer TAPRD Latching | 478 |
| 21.4.9 | CCR Timer Latching | 479 |
| 21.4.10 | CCR Capture Period/Duty Mode | 480 |
| 21.4.11 | Timer Whole Latching | 481 |
| 21.4.12 | Inverting CCR PWM Output | 481 |
| 21.4.13 | Base Timer Interrupts..... | 481 |
| 21.4.14 | CCR Compare Interrupts..... | 481 |

| | | |
|---------|------------------------------------|-----|
| 21.4.15 | CCR Capture Interrupts..... | 482 |
| 21.4.16 | Timer IRQ signal | 482 |
| 21.4.17 | Skipping CCR Interrupts..... | 483 |
| 21.4.18 | Timer ADC Triggers | 484 |
| 21.4.19 | Dead-Time Generators (DTG)..... | 485 |
| 21.4.20 | Timer D DTG Output | 487 |
| 21.5 | QEP | 488 |
| 21.6 | Peripheral IO Mapping..... | 490 |
| 21.7 | Timer D Peripheral IO Mapping..... | 491 |
| 21.8 | Register Detail | 493 |
| 21.8.1 | TDCTL..... | 493 |
| 21.8.2 | TDINT | 494 |
| 21.8.3 | TCPRD | 495 |
| 21.8.4 | TDCTR | 495 |
| 21.8.5 | TDQEPCTL | 496 |
| 21.8.6 | TDCCTL0 | 498 |
| 21.8.7 | TDCCTR0..... | 499 |
| 21.8.8 | TDCCTL1 | 500 |
| 21.8.9 | TDCCTR1..... | 501 |
| 21.8.10 | TDCCTL2..... | 502 |
| 21.8.11 | TDCCTR2 | 503 |
| 21.8.12 | TDCCTL3..... | 504 |
| 21.8.13 | TDCCTR3 | 505 |
| 21.8.14 | TDCCTL4..... | 506 |
| 21.8.15 | TDCCTR4 | 507 |
| 21.8.16 | TDCCTL5..... | 508 |
| 21.8.17 | TDCCTR5 | 509 |
| 21.8.18 | TDCCTL6..... | 510 |
| 21.8.19 | TDCCTR6 | 511 |
| 21.8.20 | TDCCTL7..... | 512 |
| 21.8.21 | TDCCTR7 | 513 |
| 21.8.22 | TDDTGCTL0..... | 514 |

| | | |
|---------|--|-----|
| 21.8.23 | TDDTGCTL1 | 514 |
| 21.8.24 | TDDTGCTL2 | 515 |
| 21.8.25 | TDDTGCTL3 | 515 |
| 22 | ADC AND DTSE | 516 |
| 22.1 | Overview | 516 |
| 22.2 | Features | 516 |
| 22.3 | Functional Description | 517 |
| 22.3.1 | ADC | 517 |
| 22.3.2 | Enabling the ADC | 517 |
| 22.3.3 | ADC Power-Down Mode | 517 |
| 22.3.4 | ADC Clock | 518 |
| 22.3.5 | EMUX | 518 |
| 22.3.6 | EMUX Interrupts | 519 |
| 22.3.7 | Manual ADC Conversions | 519 |
| 22.3.8 | Manual ADC Conversions with Repeat | 520 |
| 22.3.9 | ADC Conversion Timing | 520 |
| 22.3.10 | ADC Manual Mode Interrupts | 521 |
| 22.4 | DTSE | 522 |
| 22.4.1 | Overview | 522 |
| 22.4.2 | DTSE Conversion Sequence Example | 523 |
| 22.4.3 | Trigger Mapping Table | 523 |
| 22.4.4 | Sequence Configuration Table | 525 |
| 22.4.5 | Result Table | 527 |
| 22.4.6 | DTSE Conversion State Machine | 527 |
| 22.4.7 | Clearing DTSE Interrupts | 528 |
| 22.5 | Peripheral IO Mapping | 528 |
| 22.6 | Register Summary | 529 |
| 22.7 | Register Detail | 531 |
| 22.7.1 | EMUXCTL | 531 |
| 22.7.2 | EMUXDATA | 531 |
| 22.7.3 | ADCCTL | 532 |
| 22.7.4 | ADCRES | 534 |

| | | |
|---------|-------------------------|-----|
| 22.7.5 | ADCINT | 534 |
| 22.7.6 | DTSETRIGENT0TO3..... | 537 |
| 22.7.7 | DTSETRIGENT4TO7..... | 538 |
| 22.7.8 | DTSETRIGENT8TO11..... | 539 |
| 22.7.9 | DTSETRIGENT12TO15..... | 540 |
| 22.7.10 | DTSETRIGENT16TO19 | 541 |
| 22.7.11 | DTSETRIGENT20TO23 | 542 |
| 22.7.12 | DTSETRIGENT24TO27 | 544 |
| 22.7.13 | DTSETRIGENT28TO31 | 545 |
| 22.7.14 | DTSESEQCFG0 | 547 |
| 22.7.15 | DTSESEQCFG1 | 548 |
| 22.7.16 | DTSESEQCFG2 | 549 |
| 22.7.17 | DTSESEQCFG3 | 550 |
| 22.7.18 | DTSESEQCFG4 | 551 |
| 22.7.19 | DTSESEQCFG5 | 552 |
| 22.7.20 | DTSESEQCFG6 | 553 |
| 22.7.21 | DTSESEQCFG7 | 554 |
| 22.7.22 | DTSESEQCFG8 | 555 |
| 22.7.23 | DTSESEQCFG9 | 556 |
| 22.7.24 | DTSESEQCFG10..... | 557 |
| 22.7.25 | DTSESEQCFG11..... | 558 |
| 22.7.26 | DTSESEQCFG12..... | 559 |
| 22.7.27 | DTSESEQCFG13..... | 560 |
| 22.7.28 | DTSESEQCFG14..... | 561 |
| 22.7.29 | DTSESEQCFG15..... | 562 |
| 22.7.30 | DTSESEQCFG16..... | 563 |
| 22.7.31 | DTSESEQCFG17..... | 564 |
| 22.7.32 | DTSESEQCFG18..... | 565 |
| 22.7.33 | DTSESEQCFG19..... | 566 |
| 22.7.34 | DTSESEQCFG20..... | 567 |
| 22.7.35 | DTSESEQCFG21..... | 568 |
| 22.7.36 | DTSESEQCFG22..... | 569 |

| | | |
|---------|------------------------------|-----|
| 22.7.37 | DTSESEQCFG23..... | 570 |
| 22.7.38 | DTSERES0 | 571 |
| 22.7.39 | DTSERES1 | 571 |
| 22.7.40 | DTSERES2 | 571 |
| 22.7.41 | DTSERES3 | 571 |
| 22.7.42 | DTSERES4 | 572 |
| 22.7.43 | DTSERES5 | 572 |
| 22.7.44 | DTSERES6 | 572 |
| 22.7.45 | DTSERES7 | 572 |
| 22.7.46 | DTSERES8 | 573 |
| 22.7.47 | DTSERES9 | 573 |
| 22.7.48 | DTSERES10 | 573 |
| 22.7.49 | DTSERES11 | 573 |
| 22.7.50 | DTSERES12 | 574 |
| 22.7.51 | DTSERES13 | 574 |
| 22.7.52 | DTSERES14 | 574 |
| 22.7.53 | DTSERES15 | 574 |
| 22.7.54 | DTSERES16 | 575 |
| 22.7.55 | DTSERES17 | 575 |
| 22.7.56 | DTSERES18 | 575 |
| 22.7.57 | DTSERES19 | 575 |
| 22.7.58 | DTSERES20 | 576 |
| 22.7.59 | DTSERES21 | 576 |
| 22.7.60 | DTSERES22 | 576 |
| 22.7.61 | DTSERES23 | 576 |
| 23 | CRC | 577 |
| 23.1 | Overview..... | 577 |
| 23.2 | Block Diagram | 577 |
| 23.3 | Features | 577 |
| 23.4 | Functional Description | 578 |
| 23.4.1 | CRC Polynomial | 578 |
| 23.5 | Register Summary | 579 |

| | | |
|---------|--|-----|
| 23.6 | Register Detail | 580 |
| 23.6.1 | CRCCTL..... | 580 |
| 23.6.2 | CRCDATAIN..... | 580 |
| 23.6.3 | CRCSEED | 580 |
| 23.6.4 | CRCOUT | 581 |
| 24 | USART A..... | 582 |
| 24.1 | Overview..... | 582 |
| 24.2 | Features | 582 |
| 24.3 | System Block Diagram..... | 583 |
| 24.4 | Functional Description | 583 |
| 24.5 | UART Mode Functional Description | 584 |
| 24.5.1 | UART Mode System Block Diagram | 584 |
| 24.5.2 | Mode Configuration..... | 585 |
| 24.5.3 | Baud Rate Configuration..... | 585 |
| 24.5.4 | FIFO Reset | 586 |
| 24.5.5 | UART Configuration..... | 587 |
| 24.5.6 | UART Scratch Pad..... | 587 |
| 24.5.7 | UART Interrupts..... | 588 |
| 24.5.8 | Transmit Operation | 588 |
| 24.5.9 | Transmit Interrupts..... | 588 |
| 24.5.10 | Receive Operation..... | 589 |
| 24.5.11 | Receive Interrupts | 589 |
| 24.6 | UART Register Summary..... | 590 |
| 24.7 | UART Register Detail..... | 591 |
| 24.7.1 | UARTARBR | 591 |
| 24.7.2 | UARATHR | 591 |
| 24.7.3 | UARTADLR | 591 |
| 24.7.4 | UARTAIER | 592 |
| 24.7.5 | UARTAIIR..... | 592 |
| 24.7.6 | UARTAFCR | 593 |
| 24.7.7 | UARTALCR | 594 |
| 24.7.8 | UARTALSR | 595 |

| | | |
|---------|--|-----|
| 24.7.9 | UARTASCR..... | 596 |
| 24.7.10 | UARTAEFR..... | 596 |
| 24.8 | SSP Functional Description | 598 |
| 24.8.1 | SSP Mode System Block Diagram..... | 598 |
| 24.8.2 | Mode Configuration..... | 598 |
| 24.8.3 | SSP Overview..... | 599 |
| 24.8.4 | Clock Configuration..... | 599 |
| 24.8.5 | SSP Clock Constraints..... | 599 |
| 24.8.6 | SSP Configuration | 600 |
| 24.8.7 | SSP Slave Select Configuration..... | 601 |
| 24.8.8 | SSP Interrupts | 601 |
| 24.8.9 | Master Mode Operation | 602 |
| 24.8.10 | Slave Mode Operation..... | 602 |
| 24.8.11 | SSP Status..... | 602 |
| 24.9 | SSP Register Summary | 604 |
| 24.10 | SSP Register Detail..... | 605 |
| 24.10.1 | SSPACON..... | 605 |
| 24.10.2 | SSPASTAT | 606 |
| 24.10.3 | SSPADAT | 607 |
| 24.10.4 | SSPACLK | 608 |
| 24.10.5 | SSPAIMSC..... | 608 |
| 24.10.6 | SSPARIS | 609 |
| 24.10.7 | SSPAMIS | 609 |
| 24.10.8 | SSPAICLR | 610 |
| 24.10.9 | SSPASSCR..... | 611 |
| 25 | USART B..... | 612 |
| 25.1 | Overview..... | 612 |
| 25.2 | Features | 612 |
| 25.3 | System Block Diagram..... | 613 |
| 25.4 | Functional Description | 613 |
| 25.5 | UART Mode Functional Description | 615 |
| 25.5.1 | UART B Mode System Block Diagram..... | 615 |

| | | |
|---------|-------------------------------------|-----|
| 25.5.2 | Mode Configuration..... | 616 |
| 25.5.3 | Baud Rate Configuration..... | 616 |
| 25.5.4 | FIFO Reset | 617 |
| 25.5.5 | UART Configuration..... | 618 |
| 25.5.6 | UART Scratch Pad..... | 618 |
| 25.5.7 | UART Interrupts..... | 619 |
| 25.5.8 | Transmit Operation | 619 |
| 25.5.9 | Transmit Interrupts..... | 619 |
| 25.5.10 | Receive Operation..... | 620 |
| 25.5.11 | Receive Interrupts | 620 |
| 25.6 | UART Register Summary..... | 621 |
| 25.7 | UART Register Detail..... | 622 |
| 25.7.1 | UARTBRBR..... | 622 |
| 25.7.2 | UARBTHR | 622 |
| 25.7.3 | UARTBDLR | 622 |
| 25.7.4 | UARTBIER | 623 |
| 25.7.5 | UARTBIIR..... | 623 |
| 25.7.6 | UARTBFCR | 625 |
| 25.7.7 | UARTBLCR | 626 |
| 25.7.8 | UARTBLSR | 627 |
| 25.7.9 | UARTBSCR..... | 628 |
| 25.7.10 | UARTBEFR..... | 628 |
| 25.8 | SSP Functional Description | 629 |
| 25.8.1 | SSP Mode System Block Diagram..... | 629 |
| 25.8.2 | Mode Configuration..... | 629 |
| 25.8.3 | SSP Overview..... | 630 |
| 25.8.4 | Clock Configuration..... | 630 |
| 25.8.5 | SSP Clock Constraints..... | 630 |
| 25.8.6 | SSP Configuration | 631 |
| 25.8.7 | SSP Slave Select Configuration..... | 632 |
| 25.8.8 | SSP Interrupts | 632 |
| 25.8.9 | Master Mode Operation | 634 |

| | | |
|---------|--|-----|
| 25.8.10 | Slave Mode Operation..... | 634 |
| 25.8.11 | SSP Status..... | 634 |
| 25.9 | SSP Register Summary | 635 |
| 25.10 | SSP Register Detail..... | 636 |
| 25.10.1 | SSPBCON..... | 636 |
| 25.10.2 | SSPBSTAT | 637 |
| 25.10.3 | SSPBDAT | 638 |
| 25.10.4 | SSPBCLK | 639 |
| 25.10.5 | SSPBIMSC..... | 639 |
| 25.10.6 | SSPBRIS | 640 |
| 25.10.7 | SSPBMIS | 640 |
| 25.10.8 | SSPBICLR | 641 |
| 25.10.9 | SSPBSSCR..... | 642 |
| 26 | USART C | 643 |
| 26.1 | Overview..... | 643 |
| 26.2 | Features | 643 |
| 26.3 | System Block Diagram..... | 644 |
| 26.4 | Functional Description | 644 |
| 26.5 | UART Mode Functional Description | 646 |
| 26.5.1 | UART C Mode System Block Diagram..... | 646 |
| 26.5.2 | Mode Configuration..... | 647 |
| 26.5.3 | Baud Rate Configuration..... | 647 |
| 26.5.4 | FIFO Reset | 648 |
| 26.5.5 | UART Configuration..... | 649 |
| 26.5.6 | UART Scratch Pad..... | 649 |
| 26.5.7 | UART Interrupts..... | 650 |
| 26.5.8 | Transmit Operation | 650 |
| 26.5.9 | Transmit Interrupts..... | 650 |
| 26.5.10 | Receive Operation..... | 651 |
| 26.5.11 | Receive Interrupts | 651 |
| 26.6 | UART Register Summary..... | 652 |
| 26.7 | UART Register Detail..... | 653 |

| | | |
|---------|-------------------------------------|-----|
| 26.7.1 | UARTCRBR..... | 653 |
| 26.7.2 | UARTCTHR..... | 653 |
| 26.7.3 | UARTCDLR..... | 653 |
| 26.7.4 | UARTCIER..... | 654 |
| 26.7.5 | UARTCIIR..... | 654 |
| 26.7.6 | UARTCFCR..... | 656 |
| 26.7.7 | UARTCLCR..... | 657 |
| 26.7.8 | UARTCLSR..... | 658 |
| 26.7.9 | UARTCSCR..... | 659 |
| 26.7.10 | UARTCEFR..... | 659 |
| 26.8 | SSP Functional Description..... | 660 |
| 26.8.1 | SSP Mode System Block Diagram..... | 660 |
| 26.8.2 | Mode Configuration..... | 660 |
| 26.8.3 | SSP Overview..... | 661 |
| 26.8.4 | Clock Configuration..... | 661 |
| 26.8.5 | SSP Clock Constraints..... | 661 |
| 26.8.6 | SSP Configuration..... | 662 |
| 26.8.7 | SSP Slave Select Configuration..... | 663 |
| 26.8.8 | SSP Interrupts..... | 663 |
| 26.8.9 | Master Mode Operation..... | 665 |
| 26.8.10 | Slave Mode Operation..... | 665 |
| 26.8.11 | SSP Status..... | 665 |
| 26.9 | SSP Register Summary..... | 666 |
| 26.10 | SSP Register Detail..... | 667 |
| 26.10.1 | SSPCCON..... | 667 |
| 26.10.2 | SSPCSTAT..... | 668 |
| 26.10.3 | SSPCDAT..... | 669 |
| 26.10.4 | SSPCCLK..... | 670 |
| 26.10.5 | SSPCIMSC..... | 670 |
| 26.10.6 | SSPCRIS..... | 671 |
| 26.10.7 | SSPCMIS..... | 671 |
| 26.10.8 | SSPICLR..... | 672 |

| | | |
|---------|--|-----|
| 26.10.9 | SSPCSSCR | 673 |
| 27 | USART D | 674 |
| 27.1 | Overview..... | 674 |
| 27.2 | Features | 674 |
| 27.3 | System Block Diagram..... | 675 |
| 27.4 | Functional Description | 675 |
| 27.5 | UART Mode Functional Description | 677 |
| 27.5.1 | UART D Mode System Block Diagram..... | 677 |
| 27.5.2 | Mode Configuration..... | 678 |
| 27.5.3 | Baud Rate Configuration..... | 678 |
| 27.5.4 | FIFO Reset | 679 |
| 27.5.5 | UART Configuration..... | 680 |
| 27.5.6 | UART Scratch Pad..... | 680 |
| 27.5.7 | UART Interrupts..... | 681 |
| 27.5.8 | Transmit Operation | 681 |
| 27.5.9 | Transmit Interrupts..... | 681 |
| 27.5.10 | Receive Operation..... | 682 |
| 27.5.11 | Receive Interrupts | 682 |
| 27.6 | UART Register Summary..... | 683 |
| 27.7 | UART Register Detail..... | 684 |
| 27.7.1 | UARTDRBR..... | 684 |
| 27.7.2 | UARTDTHR | 684 |
| 27.7.3 | UARTDDLRL | 684 |
| 27.7.4 | UARTDIER | 685 |
| 27.7.5 | UARTDIIR..... | 685 |
| 27.7.6 | UARTDFCR | 686 |
| 27.7.7 | UARTDLCR | 687 |
| 27.7.8 | UARTDLR | 688 |
| 27.7.9 | UARTDSCR..... | 689 |
| 27.7.10 | UARTDEFRL..... | 689 |
| 27.8 | SSP Functional Description | 690 |
| 27.8.1 | SSP Mode System Block Diagram..... | 690 |

| | | |
|---------|-------------------------------------|-----|
| 27.8.2 | Mode Configuration..... | 690 |
| 27.8.3 | SSP Overview..... | 691 |
| 27.8.4 | Clock Configuration..... | 691 |
| 27.8.5 | SSP Clock Constraints..... | 691 |
| 27.8.6 | SSP Configuration | 692 |
| 27.8.7 | SSP Slave Select Configuration..... | 693 |
| 27.8.8 | SSP Interrupts | 693 |
| 27.8.9 | Master Mode Operation | 695 |
| 27.8.10 | Slave Mode Operation..... | 695 |
| 27.8.11 | SSP Status..... | 695 |
| 27.9 | SSP Register Summary | 696 |
| 27.10 | SSP Register Detail..... | 697 |
| 27.10.1 | SSPDCON | 697 |
| 27.10.2 | SSPDSTAT | 698 |
| 27.10.3 | SSPDDAT | 699 |
| 27.10.4 | SSPDCLK | 700 |
| 27.10.5 | SSPDIMSC | 700 |
| 27.10.6 | SSPDRIS | 701 |
| 27.10.7 | SSPDMIS..... | 701 |
| 27.10.8 | SSPDICLR | 702 |
| 27.10.9 | SSPDSSCR | 703 |
| 28 | I2C | 704 |
| 28.1 | Overview..... | 704 |
| 28.2 | Features | 704 |
| 28.3 | System Block Diagram..... | 704 |
| 28.4 | Functional Description | 705 |
| 28.4.1 | Enabling I2C | 705 |
| 28.4.2 | I2C Clocking | 705 |
| 28.4.3 | I2C Master Write..... | 705 |
| 28.4.4 | I2C Master Read..... | 706 |
| 28.4.5 | I2C Slave Write..... | 707 |
| 28.4.6 | I2C Slave Read..... | 707 |

| | | |
|---------|------------------------------|-----|
| 28.4.7 | I2C Slave Addressing | 707 |
| 28.4.8 | I2C Interrupts | 708 |
| 28.4.9 | I2C Status | 709 |
| 28.4.10 | I2C Reset | 710 |
| 28.5 | Register Summary | 711 |
| 28.6 | Register Detail | 712 |
| 28.6.1 | I2CCONSET | 712 |
| 28.6.2 | I2CCONCLR | 713 |
| 28.6.3 | I2CSTAT | 713 |
| 28.6.4 | I2CDAT | 714 |
| 28.6.5 | I2CCLK | 714 |
| 28.6.6 | I2CADR0 | 714 |
| 28.6.7 | I2CADRM0 | 714 |
| 28.6.8 | I2CXADR0 | 715 |
| 28.6.9 | I2CXADRM0 | 715 |
| 28.6.10 | I2CRST | 716 |
| 28.6.11 | I2CADR1 | 716 |
| 28.6.12 | I2CADRM1 | 716 |
| 28.6.13 | I2CADR2 | 717 |
| 28.6.14 | I2CADRM2 | 717 |
| 28.6.15 | I2CADR3 | 717 |
| 28.6.16 | I2CADRM3 | 717 |
| 29 | CAN | 719 |
| 29.1 | Features | 719 |
| 29.2 | System Block Diagram | 720 |
| 29.3 | Functional Description | 721 |
| 29.4 | Register Summary | 722 |
| 29.5 | Register Detail | 723 |
| 29.5.1 | CAN_ISR_SR_CMR_MR | 723 |
| 29.5.2 | CAN_BTR1_BTR0_RMC_IMR | 725 |
| 29.5.3 | CANTXBUF | 726 |
| 29.5.4 | CANRXBUF | 726 |

| | | |
|--------|--------------------------------|-----|
| 29.5.5 | CANACR | 726 |
| 29.5.6 | CANACRM | 726 |
| 29.5.7 | CAN_ALC_TXERR_RXERR_ECC | 727 |
| 30 | ARM CORTEX-M4F REFERENCE | 729 |
| 31 | LEGAL INFORMATION..... | 730 |

LIST OF FIGURES

| | |
|---|-----|
| Figure 3-1 Memory Map | 51 |
| Figure 4-1 CCS Block Diagram | 78 |
| Figure 5-1 Memory Controller System Block Diagram | 138 |
| Figure 6-1 GPIO and Peripheral MUX Block Diagram | 153 |
| Figure 7-1 WWDT System Block Diagram | 159 |
| Figure 8-1 RTC System Block Diagram..... | 165 |
| Figure 9-1 GPIOA Block Diagram | 172 |
| Figure 10-1 GPIOB Block Diagram | 194 |
| Figure 11-1 GPIOC Block Diagram | 216 |
| Figure 12-1 GPIOD Block Diagram | 238 |
| Figure 13-1 GPIOE Block Diagram | 261 |
| Figure 14-1 GPIOF Block Diagram..... | 283 |
| Figure 15-1 GPIOG Block Diagram..... | 306 |
| Figure 16-1 GP Timer A System Block Diagram | 329 |
| Figure 17-1 GP Timer B System Block Diagram | 332 |
| Figure 18-1 Timer A Simplified Block Diagram | 335 |
| Figure 18-2 Timer A Block Diagram | 336 |
| Figure 18-3 Up Mode PWM Waveform | 340 |
| Figure 18-4 Up/Down Mode PWM Waveform | 340 |
| Figure 18-5 Up/Down Asymmetric Mode PWM Waveform | 341 |
| Figure 18-6 Up Mode with Phase Delay | 342 |
| Figure 18-7 Period/Duty Capture Mode PWM Waveform | 345 |
| Figure 18-8 CCR Interrupt Skipping Diagram | 348 |
| Figure 18-9 Timer A DTG Output | 352 |
| Figure 19-1 Timer B Simplified Block Diagram | 380 |
| Figure 19-2 Timer B Block Diagram | 381 |
| Figure 19-3 Up Mode PWM Waveform | 385 |
| Figure 19-4 Up/Down Mode PWM Waveform | 385 |
| Figure 19-5 Up/Down Asymmetric Mode PWM Waveform | 386 |
| Figure 19-6 Up Mode with Phase Delay | 387 |
| Figure 19-7 Period/Duty Capture Mode PWM Waveform | 390 |
| Figure 19-8 CCR Interrupt Skipping Diagram | 393 |
| Figure 19-9 Timer B DTG Output | 397 |
| Figure 20-1 Timer C Simplified Block Diagram..... | 425 |
| Figure 20-2 Timer C Block Diagram | 426 |
| Figure 20-3 Up Mode PWM Waveform | 430 |
| Figure 20-4 Up/Down Mode PWM Waveform | 431 |
| Figure 20-5 Up/Down Asymmetric Mode PWM Waveform | 431 |
| Figure 20-6 Up Mode with Phase Delay | 432 |
| Figure 20-7 Period/Duty Capture Mode PWM Waveform | 435 |
| Figure 20-8 CCR Interrupt Skipping Diagram | 438 |

| | |
|--|-----|
| Figure 20-9 Timer C DTG Output..... | 442 |
| Figure 21-1 Timer D Simplified Block Diagram..... | 470 |
| Figure 21-2 Timer D Block Diagram..... | 471 |
| Figure 21-3 Up Mode PWM Waveform | 475 |
| Figure 21-4 Up/Down Mode PWM Waveform | 476 |
| Figure 21-5 Up/Down Asymmetric Mode PWM Waveform | 476 |
| Figure 21-6 Up Mode with Phase Delay..... | 477 |
| Figure 21-7 Period/Duty Capture Mode PWM Waveform | 480 |
| Figure 21-8 CCR Interrupt Skipping Diagram..... | 483 |
| Figure 21-9 Timer D DTG Output..... | 487 |
| Figure 22-1 ADC and DTSE Simplified Block Diagram..... | 516 |
| Figure 22-2 ADC Conversion Timing Diagram (single conversion)..... | 520 |
| Figure 22-3 DTSE Simplified Block Diagram..... | 522 |
| Figure 22-4 DTSE Conversion Sequence Example..... | 523 |
| Figure 23-1 CRC Block Diagram..... | 577 |
| Figure 24-1 USART A System Block Diagram..... | 583 |
| Figure 24-2 UART A Mode System Block Diagram..... | 584 |
| Figure 24-3 SSP A System Block Diagram | 598 |
| Figure 25-1 USART B System Block Diagram..... | 613 |
| Figure 25-2 UART B Mode System Block Diagram..... | 615 |
| Figure 25-3 SSP B System Block Diagram | 629 |
| Figure 26-1 USART C System Block Diagram | 644 |
| Figure 26-2 UART C Mode System Block Diagram..... | 646 |
| Figure 26-3 SSP C System Block Diagram | 660 |
| Figure 27-1 USART D System Block Diagram | 675 |
| Figure 27-2 UART D Mode System Block Diagram..... | 677 |
| Figure 27-3 SSP D System Block Diagram | 690 |
| Figure 28-1 I2C System Block Diagram | 704 |
| Figure 29-1 CAN System Block Diagram | 720 |

LIST OF TABLES

| | |
|--|----|
| Table 3-1 Embedded FLASH Register Map | 52 |
| Table 3-2 INFO-1 Register Map | 56 |
| Table 3-3 INFO-2 Register Map | 57 |
| Table 3-4 INFO-1, INFO-2 Field Name and Description | 58 |
| Table 3-5 INFO-3 Register Map | 59 |
| Table 3-6 ADC Register Map | 59 |
| Table 3-7 I2C Register Map | 61 |
| Table 3-8 USARTA Register Map | 62 |
| Table 3-9 USARTB Register Map | 63 |
| Table 3-10 USARTC Register Map | 64 |
| Table 3-11 USARTD Register Map | 65 |
| Table 3-12 Timer A Register Map | 66 |
| Table 3-13 Timer B Register Map | 67 |
| Table 3-14 Timer C Register Map | 68 |
| Table 3-15 Timer D Register Map | 69 |
| Table 3-16 CAN Register Map | 70 |
| Table 3-17 GP Timer A Register Map | 70 |
| Table 3-18 GP Timer B Register Map | 70 |
| Table 3-19 Memory Controller Register Map | 70 |
| Table 3-20 System and Clock Control (SCC) Register Map | 71 |
| Table 3-21 Windowed Watchdog Timer (WWDT) Register Map | 72 |
| Table 3-22 Real-Time Clock (RTC) Register Map | 72 |
| Table 3-23 CRC Register Map | 72 |
| Table 3-24 GPIOA Register Map | 73 |
| Table 3-25 GPIOB Register Map | 73 |
| Table 3-26 GPIOC Register Map | 74 |
| Table 3-27 GPIOD Register Map | 74 |
| Table 3-28 GPIO E Register Map | 75 |
| Table 3-29 GPIO F Register Map | 75 |
| Table 3-30 GPIO G Register Map | 76 |
| Table 4-1 CCS Clock Source Selections | 79 |
| Table 4-2 SCLK Clock Input Selections | 80 |
| Table 4-3 PCLK Clock Divider Settings | 80 |
| Table 4-4 ACLK Clock Divider Settings | 81 |
| Table 4-5 HCLK Clock Divider Settings | 81 |
| Table 4-6 PLL Configuration Variables | 83 |
| Table 4-7 PLL Configuration Examples | 83 |
| Table 4-8 Peripheral Clock Input | 84 |
| Table 4-9 Clock Failure Detection Configuration | 86 |
| Table 4-10 PAC55XX Interrupt Vector Table | 87 |
| Table 4-11 SCC Register Summary | 90 |

| | |
|--|-----|
| Table 5-1 Memory Controller MCLK Divider | 139 |
| Table 5-2 Memory Controller FLASH Wait States | 140 |
| Table 5-3 FLASH Read Clock Configuration Examples..... | 140 |
| Table 5-4 INFO-3 Format for Code Protection | 144 |
| Table 5-5 Code Protection Levels | 145 |
| Table 5-6 FLASH Controller Register Summary..... | 149 |
| Table 6-1 Peripheral Functions | 154 |
| Table 6-2 Peripheral MUX Settings | 157 |
| Table 7-1 WWDT Input Clock Divider..... | 160 |
| Table 7-2 WWDT Register Summary | 161 |
| Table 8-1 RTC Register Summary | 167 |
| Table 9-1 GPIOA Mode Configuration..... | 174 |
| Table 9-2 GPIOA Interrupt Configuration | 175 |
| Table 9-3 GPIOA Drive Strength..... | 175 |
| Table 9-4 GPIOA Peripheral IO Mapping | 177 |
| Table 9-5 GPIOA Register Summary | 178 |
| Table 10-1 GPIOB Mode Configuration..... | 196 |
| Table 10-2 GPIOB Interrupt Configuration | 197 |
| Table -3 GPIOB Drive Strength..... | 197 |
| Table 10-4 GPIOB Peripheral IO Mapping | 199 |
| Table 10-5 GPIOB Register Summary | 200 |
| Table 11-1 GPIOC Mode Configuration | 218 |
| Table 11-2 GPIOC Interrupt Configuration | 219 |
| Table 11-3 GPIOC Drive Strength..... | 219 |
| Table 11-4 GPIOC Peripheral IO Mapping | 221 |
| Table 11-5 GPIOC Register Summary | 222 |
| Table 12-1 GPIOD Mode Configuration | 240 |
| Table 12-2 GPIOD Interrupt Configuration | 241 |
| Table 12-3 GPIOD Drive Strength..... | 242 |
| Table 12-4 GPIOD Peripheral IO Mapping | 244 |
| Table 12-5 GPIOD Register Summary | 245 |
| Table 13-1 GPIOE Mode Configuration..... | 263 |
| Table 13-2 GPIOE Interrupt Configuration | 264 |
| Table 13-3 GPIOE Drive Strength..... | 264 |
| Table 13-4 GPIOE Peripheral IO Mapping | 266 |
| Table 13-5 GPIOE Register Summary | 267 |
| Table 14-1 GPIOF Mode Configuration | 285 |
| Table 14-2 GPIOF Interrupt Configuration..... | 286 |
| Table 14-3 GPIOF Drive Strength | 287 |
| Table 14-4 GPIOF Peripheral IO Mapping | 289 |
| Table 14-5 GPIOF Register Summary | 290 |
| Table 15-1 GPIOG Mode Configuration | 308 |
| Table 15-2 GPIOG Interrupt Configuration | 309 |

| | |
|---|-----|
| Table 15-3 GPIOG Drive Strength..... | 310 |
| Table 15-4 GPIOG Peripheral IO Mapping..... | 312 |
| Table 15-5 GPIOG Register Summary..... | 313 |
| Table 16-1 GPTimer A Input Clock Divider..... | 330 |
| Table 16-2 GPTimer A Register Summary..... | 331 |
| Table 17-1 GPTimer B Input Clock Divider..... | 333 |
| Table 17-2 GPTimer B Register Summary..... | 334 |
| Table 18-1 Timer A Peripheral IO Mapping..... | 355 |
| Table 19-1 Timer B Peripheral IO Mapping..... | 400 |
| Table 20-1 Timer C Peripheral IO Mapping..... | 445 |
| Table 21-1 Timer D Register Summary..... | 491 |
| Table 22-1 ADC Input Clock Divider..... | 518 |
| Table 22-2 EMUX Input Clock Divider..... | 519 |
| Table 22-3 DTSE Trigger Mapping Table..... | 524 |
| Table 22-4 DTSE Sequence Configuration Table Entry..... | 526 |
| Table 22-5 ADC Peripheral IO Mapping..... | 528 |
| Table 22-6 ADC and DTSE Register Summary..... | 529 |
| Table 23-1 CRC Polynomial Selections..... | 578 |
| Table 23-2 CRC Register Summary..... | 579 |
| Table 24-1 USART A UART Mode Registers..... | 585 |
| Table 24-2 UART A Parity Modes..... | 587 |
| Table 24-3 UART A Word Length..... | 587 |
| Table 24-4 UART A Interrupt ID..... | 588 |
| Table 24-5 USART A Register Summary (UART Mode)..... | 590 |
| Table 24-6 USART A UART Mode Registers..... | 598 |
| Table 24-7 SSP A Frame Format..... | 600 |
| Table 24-8 SSP A Data Size Select..... | 601 |
| Table 24-9 SSP A Status..... | 602 |
| Table 24-10 USART A Register Summary (SSP Mode)..... | 604 |
| Table 25-1 USART B UART Mode Registers..... | 616 |
| Table 25-2 UART A Parity Modes..... | 618 |
| Table 25-3 UART B Word Length..... | 618 |
| Table 25-4 UART B Interrupt ID..... | 619 |
| Table 25-5 USART B Register Summary (UART Mode)..... | 621 |
| Table 25-6 USART B UART Mode Registers..... | 629 |
| Table 25-7 SSP B Frame Format..... | 631 |
| Table 25-8 SSP B Data Size Select..... | 632 |
| Table 25-9 SSP B Status..... | 634 |
| Table 25-10 USART B Register Summary (SSP Mode)..... | 635 |
| Table 26-1 USART C UART Mode Registers..... | 647 |
| Table 26-2 UART C Parity Modes..... | 649 |
| Table 26-3 UART C Word Length..... | 649 |
| Table 26-4 UART C Interrupt ID..... | 650 |

| | |
|--|-----|
| Table 26-5 USART C Register Summary (UART Mode) | 652 |
| Table 26-6 USART C UART Mode Registers | 660 |
| Table 26-7 SSP C Frame Format | 662 |
| Table 26-8 SSP C Data Size Select | 663 |
| Table 26-9 SSP C Status | 665 |
| Table 26-10 USART C Register Summary (SSP Mode) | 666 |
| Table 27-1 USART D UART Mode Registers | 678 |
| Table 27-2 UART D Parity Modes | 680 |
| Table 27-3 UART D Word Length | 680 |
| Table 27-4 UART D Interrupt ID | 681 |
| Table 27-5 USART D Register Summary (UART Mode) | 683 |
| Table 27-6 USART D UART Mode Registers | 690 |
| Table 27-7 SSP D Frame Format | 692 |
| Table 27-8 SSP C Data Size Select | 693 |
| Table 27-9 SSP D Status | 695 |
| Table 27-10 USART D Register Summary (SSP Mode) | 696 |
| Table 28-1 I2C F_{SAMP} and F_{CLK} configuration examples | 705 |
| Table 28-2 I2C Slave Address Matching | 707 |
| Table 28-3 I2C Status Codes | 709 |
| Table 28-4 I2C Register Summary | 711 |
| Table 29-1 CAN Register Summary | 722 |

LIST OF REGISTERS

| | |
|--|-----|
| Register 4-1 CCSCTL (CCS Configuration, 400D 0400h) | 91 |
| Register 4-2 CCSPLLCTL (CCS PLL Configuration, 400D 0404h)..... | 94 |
| Register 4-3 CCSROSCTRIM (CCS ROSC Trim Configuration, 400D 0408h) | 94 |
| Register 4-4 PAMUXSEL (PA Peripheral MUX Select, 400D 040Ch)..... | 95 |
| Register 4-5 PBMUXSEL (PB Peripheral MUX Select, 400D 0410h) | 97 |
| Register 4-6 PCMUXSEL (PC Peripheral MUX Select, 400D 0414h)..... | 99 |
| Register 4-7 PDMUXSEL (PD Peripheral MUX Select, 400D 0418h)..... | 101 |
| Register 4-8 PEMUXSEL (PE Peripheral MUX Select, 400D 041Ch)..... | 103 |
| Register 4-9 PFMUXSEL (PF Peripheral MUX Select, 400D 0420h)..... | 105 |
| Register 4-10 PGMUXSEL (PG Peripheral MUX Select, 400D 0424h) | 107 |
| Register 4-11 PAPUEN (PA Pull-up Enable, 400D 0428h)..... | 109 |
| Register 4-12 PBPUE (PB Pull-up Enable, 400D 042Ch) | 110 |
| Register 4-13 PCPUEN (PC Pull-up Enable, 400D 0430h) | 111 |
| Register 4-14 PDPUE (PD Pull-up Enable, 400D 0034h) | 112 |
| Register 4-15 PEPUE (PE Pull-up Enable, 400D 0438h)..... | 113 |
| Register 4-16 PPFUE (PF Pull-up Enable, 400D 043Ch) | 114 |
| Register 4-17 PGPUEN (PG Pull-up Enable, 400D 0440h)..... | 115 |
| Register 4-18 PAPDEN (PA Pull-down Enable, 400D 0444h) | 116 |
| Register 4-19 PBPDEN (PB Pull-down Enable, 400D 0448h) | 117 |
| Register 4-20 PCPDEN (PC Pull-down Enable, 400D 044Ch) | 118 |
| Register 4-21 PDPDEN (PD Pull-down Enable, 400D 0450h)..... | 119 |
| Register 4-22 PEPDEN (PE Pull-down Enable, 400D 0454h) | 120 |
| Register 4-23 PFPDEN (PF Pull-down Enable, 400D 0458h)..... | 121 |
| Register 4-24 PGPDEN (PG Pull-down Enable, 400D 045Ch)..... | 122 |
| Register 4-25 PADS (PA Drive Strength/Schmitt Trigger, 400D 0460h) | 123 |
| Register 4-26 PBDS (PB Drive Strength/Schmitt Trigger, 400D 0464h) | 125 |
| Register 4-27 PCDS (PC Drive Strength/Schmitt Trigger, 400D 0468h)..... | 127 |
| Register 4-28 PDDS (PD Drive Strength/Schmitt Trigger, 400D 046Ch) | 129 |
| Register 4-29 PEDS (PE Drive Strength/Schmitt Trigger, 400D 0470h) | 131 |
| Register 4-30 PFDS (PF Drive Strength/Schmitt Trigger, 400D 0474h) | 133 |
| Register 4-31 PGDS (PG Drive Strength/Schmitt Trigger, 400D 0478h) | 135 |
| Register 5-1 MEMCTL (Memory Controller Configuration, 400D 0000h) | 150 |
| Register 5-2 MEMSTATUS (Memory Controller Status, 400D 0004h)..... | 151 |
| Register 5-3 FLASHLOCK (Memory Controller Status, 400D 0008h)..... | 152 |
| Register 5-4 FLASHPAGE (FLASH Page, 400D 000Ch) | 152 |
| Register 5-5 SWDUNLOCK (SWD Unlock, 400D 0010h)..... | 152 |
| Register 5-6 FLASHERASE (FLASH Erase, 400D 0020h)..... | 152 |
| Register 7-1 WWDTCTL (WWDT Control, 400D 0800h) | 162 |
| Register 7-2 WWDTLOAD (WWDT Load Value, 400D 0804h)..... | 162 |
| Register 7-3 WWDTCTR (WWDT Counter Value, 400D 0808h) | 163 |
| Register 7-4 WWDTINTF (WWDT Interrupt Flag, 400D 080Ch)..... | 163 |

| | |
|--|-----|
| Register 7-5 WWDTCLEAR (WWDT Interrupt Clear, 400D 0810h)..... | 163 |
| Register 7-6 WWDTLOCK (WWDT Lock, 400D 0814h)..... | 163 |
| Register 8-1 RTCCTL (RTC Control, 400D 0C00h)..... | 168 |
| Register 8-2 RTCTIME (RTC Time, 400D 0C04h)..... | 169 |
| Register 8-3 RTCDATE (RTC Date, 400D 0C08h)..... | 169 |
| Register 8-4 RTCTIMESET (RTC Time Setting, 400D 0C0Ch)..... | 170 |
| Register 8-5 RTCDATESET (RTC Date Setting, 400D 0C10h)..... | 170 |
| Register 8-6 RTCALARMSET (RTC Alarm Setting, 400D 0C14h)..... | 171 |
| Register 9-1 GPIOAMODE (GPIOA Mode Configuration, 400D 1400h)..... | 179 |
| Register 9-2 GPIOAOUTMASK (GPIOA Output Mask, 400D 1404h)..... | 180 |
| Register 9-3 GPIOAOUT (GPIOA Output Data Value, 400D 1408h)..... | 181 |
| Register 9-4 GPIOAIN (GPIOA Input Data Value, 400D 140Ch)..... | 182 |
| Register 9-5 GPIOAINTEN (GPIOA Interrupt Enable, 400D 1410h)..... | 183 |
| Register 9-6 GPIOAINTFLAGRAW (GPIOA Interrupt Flag Raw, 400D 1414h)..... | 184 |
| Register 9-7 GPIOAINTFLAGMASKED (GPIOA Interrupt Flag Masked, 400D 1418h)..... | 185 |
| Register 9-8 GPIOAINTCLEAR (GPIOA Interrupt Clear, 400D 141Ch)..... | 186 |
| Register 9-9 GPIOAINTTYPE (GPIOA Interrupt Type Configuration, 400D 1420h)..... | 187 |
| Register 9-10 GPIOAINTCFG (GPIOA Interrupt Value Configuration, 400D 1424h)..... | 188 |
| Register 9-11 GPIOAINTEDGEBOTH (GPIOA Interrupt Both Edge Configuration, 400D 1428h) | 190 |
| Register 9-12 GPIOADEBOUNCE (GPIOA De-bounce Enable, 400D 142Ch)..... | 191 |
| Register 9-13 GPIOADDOSET (GPIOA Data Output Set, 400D 1430h)..... | 192 |
| Register 9-14 GPIOADOCLEAR (GPIOA Data Output Clear, 400D 1434h)..... | 193 |
| Register 10-1 GPIOBMODE (GPIOB Mode Configuration, 400D 1800h)..... | 201 |
| Register 10-2 GPIOBOUTMASK (GPIOB Output Mask, 400D 1804h)..... | 202 |
| Register 10-3 GPIOBOUT (GPIOB Output Data Value, 400D 1808h)..... | 203 |
| Register 10-4 GPIOBIN (GPIOB Input Data Value, 400D 180Ch)..... | 204 |
| Register 10-5 GPIOBINTEN (GPIOB Interrupt Enable, 400D 1810h)..... | 205 |
| Register 10-6 GPIOBINTFLAGRAW (GPIOB Interrupt Flag Raw, 400D 1814h)..... | 206 |
| Register 10-7 GPIOBINTFLAGMASKED (GPIOB Interrupt Flag Masked, 400D 1818h)..... | 207 |
| Register 10-8 GPIOBINTCLEAR (GPIOB Interrupt Clear, 400D 181Ch)..... | 208 |
| Register 10-9 GPIOBINTTYPE (GPIOB Interrupt Type Configuration, 400D 1820h)..... | 209 |
| Register 10-10 GPIOBINTCFG (GPIOB Interrupt Value Configuration, 400D 1824h)..... | 210 |
| Register 10-11 GPIOBINTEDGEBOTH (GPIOB Interrupt Both Edge Configuration, 400D 1828h) | 212 |
| Register 10-12 GPIOBDEBOUNCE (GPIOB De-bounce Enable, 400D 182Ch)..... | 213 |
| Register 10-13 GPIOBDDOSET (GPIOB Data Output Set, 400D 1830h)..... | 214 |
| Register 10-14 GPIOBDOCLEAR (GPIOB Data Output Clear, 400D 1834h)..... | 215 |
| Register 11-1 GPIOCMODE (GPIOC Mode Configuration, 400D 1C00h)..... | 223 |
| Register 11-2 GPIOCOUTMASK (GPIOC Output Mask, 400D 1C04h)..... | 224 |
| Register 11-3 GPIOCOUT (GPIOC Output Data Value, 400D 1C08h)..... | 225 |
| Register 11-4 GPIOCIN (GPIOC Input Data Value, 400D 1C0Ch)..... | 226 |
| Register 11-5 GPIOCINTEN (GPIOC Interrupt Enable, 400D 1C10h)..... | 227 |

| | |
|--|-----|
| Register 11-6 GPIOCINTFLAGRAW (GPIOC Interrupt Flag Raw, 400D 1C14h) | 228 |
| Register 11-7 GPIOCINTFLAGMASKED (GPIOC Interrupt Flag Masked, 400D 1C18h) | 229 |
| Register 11-8 GPIOCINTCLEAR (GPIOC Interrupt Clear, 400D 1C1Ch) | 230 |
| Register 11-9 GPIOCINTTYPE (GPIOC Interrupt Type Configuration, 400D 1C20h) | 231 |
| Register 11-10 GPIOCINTCFG (GPIOC Interrupt Configuration, 400D 1C24h) | 232 |
| Register 11-11 GPIOCINTEDGEBOTH (GPIOC Interrupt Both Edge Configuration, 400D 1C28h) | 234 |
| Register 11-12 GPIOCDEBOUNCE (GPIOC De-bounce Enable, 400D 1C2Ch) | 235 |
| Register 11-13 GPIOCDOSET (GPIOC Data Output Set, 400D 1C30h) | 236 |
| Register 11-14 GPIOCDOCLEAR (GPIOC Data Output Clear, 400D 1C34h) | 237 |
| Register 12-1 GPIODMODE (GPIOD Mode Configuration, 400D 2000h) | 246 |
| Register 12-2 GPIODOUTMASK (GPIOD Output Mask, 400D 2004h) | 247 |
| Register 12-3 GPIODOUT (GPIOD Output Data Value, 400D 2008h) | 248 |
| Register 12-4 GPIODIN (GPIOD Input Data Value, 400D 200Ch) | 249 |
| Register 12-5 GPIODINTEN (GPIOD Interrupt Enable, 400D 2010h) | 250 |
| Register 12-6 GPIODINTFLAGRAW (GPIOD Interrupt Flag Raw, 400D 2014h) | 251 |
| Register 12-7 GPIODINTFLAGMASKED (GPIOD Interrupt Flag Masked, 400D 2018h) | 252 |
| Register 12-8 GPIODINTCLEAR (GPIOD Interrupt Clear, 400D 201Ch) | 253 |
| Register 12-9 GPIODINTTYPE (GPIOD Interrupt Type Configuration, 400D 2020h) | 254 |
| Register 12-10 GPIODINTCFG (GPIOD Interrupt Configuration, 400D 2024h) | 255 |
| Register 12-11 GPIODINTEDGEBOTH (GPIOD Interrupt Both Edge Configuration, 400D 2028h) | 257 |
| Register 12-12 GPIODDEBOUNCE (GPIOD De-bounce Enable, 400D 202Ch) | 258 |
| Register 12-13 GPIODDOSET (GPIOD Data Output Set, 400D 2030h) | 259 |
| Register 12-14 GPIODDOCLEAR (GPIOD Data Output Clear, 400D 2034h) | 260 |
| Register 13-1 GPIOEMODE (GPIOE Mode Configuration, 400D 2400h) | 268 |
| Register 13-2 GPIOEOUTMASK (GPIOE Output Mask, 400D 2404h) | 269 |
| Register 13-3 GPIOEOUT (GPIOE Output Data Value, 400E 2408h) | 270 |
| Register 13-4 GPIOEIN (GPIOE Input Data Value, 400D 240Ch) | 271 |
| Register 13-5 GPIOEINTEN (GPIOE Interrupt Enable, 400D 2410h) | 272 |
| Register 13-6 GPIOEINTFLAGRAW (GPIOE Interrupt Flag Raw, 400D 2414h) | 273 |
| Register 13-7 GPIOEINTFLAGMASKED (GPIOE Interrupt Flag Masked, 400D 2418h) | 274 |
| Register 13-8 GPIOEINTCLEAR (GPIOE Interrupt Clear, 400D 241Ch) | 275 |
| Register 13-9 GPIOEINTTYPE (GPIOE Interrupt Type Configuration, 400D 2420h) | 276 |
| Register 13-10 GPIOEINTCFG (GPIOE Interrupt Configuration, 400D 2424h) | 277 |
| Register 13-11 GPIOEINTEDGEBOTH (GPIOE Interrupt Both Edge Configuration, 400D 2428h) | 279 |
| Register 13-12 GPIOEDEBOUNCE (GPIOE De-bounce Enable, 400D 242Ch) | 280 |
| Register 13-13 GPIOEDOSET (GPIOE Data Output Set, 400D 2430h) | 281 |
| Register 13-14 GPIOEDOCLEAR (GPIOE Data Output Clear, 400D 2434h) | 282 |
| Register 14-1 GPIOFMODE (GPIOF Mode Configuration, 400D 2800h) | 291 |
| Register 14-2 GPIOFOUTMASK (GPIOF Output Mask, 400D 2804h) | 292 |
| Register 14-3 GPIOFOUT (GPIOF Output Data Value, 400E 2808h) | 293 |

| | |
|---|-----|
| Register 14-4 GPIOFIN (GPIOF Input Data Value, 400D 280Ch) | 294 |
| Register 14-5 GPIOFINTEN (GPIOF Interrupt Enable, 400D 2810h) | 295 |
| Register 14-6 GPIOFINTFLAGRAW (GPIOF Interrupt Flag Raw, 400D 2814h) | 296 |
| Register 14-7 GPIOFINTFLAGMASKED (GPIOF Interrupt Flag Masked, 400D 2818h) | 297 |
| Register 14-8 GPIOFINTCLEAR (GPIOF Interrupt Clear, 400D 281Ch) | 298 |
| Register 14-9 GPIOFINTTYPE (GPIOF Interrupt Type Configuration, 400D 2820h) | 299 |
| Register 14-10 GPIOFINTCFG (GPIOF Interrupt Configuration, 400D 2824h) | 300 |
| Register 14-11 GPIOFINTEDGEBOTH (GPIOF Interrupt Both Edge Configuration, 400D 2828h) | 302 |
| Register 14-12 GPIOFDEBOUNCE (GPIOF De-bounce Enable, 400D 282Ch) | 303 |
| Register 14-13 GPIOFDOSET (GPIOF Data Output Set, 400D 2830h) | 304 |
| Register 14-14 GPIOFDOCLEAR (GPIOF Data Output Clear, 400D 2834h) | 305 |
| Register 15-1 GPIOGMODE (GPIOG Mode Configuration, 400D 2C00h) | 314 |
| Register 15-2 GPIOGOUTMASK (GPIOG Output Mask, 400D 2C04h) | 315 |
| Register 15-3 GPIOGOUT (GPIOG Output Data Value, 400E 2C08h) | 316 |
| Register 15-4 GPIOGIN (GPIOG Input Data Value, 400D 2C0Ch) | 317 |
| Register 15-5 GPIOGINTEN (GPIOG Interrupt Enable, 400D 2C10h) | 318 |
| Register 15-6 GPIOGINTFLAGRAW (GPIOG Interrupt Flag Raw, 400D 2C14h) | 319 |
| Register 15-7 GPIOGINTFLAGMASKED (GPIOG Interrupt Flag Masked, 400D 2C18h) | 320 |
| Register 15-8 GPIOGINTCLEAR (GPIOG Interrupt Clear, 400D 2C1Ch) | 321 |
| Register 15-9 GPIOGINTTYPE (GPIOG Interrupt Type Configuration, 400D 2C20h) | 322 |
| Register 15-10 GPIOGINTCFG (GPIOG Interrupt Configuration, 400D 2C24h) | 323 |
| Register 15-11 GPIOGINTEDGEBOTH (GPIOG Interrupt Both Edge Configuration, 400D 2C28h) | 325 |
| Register 15-12 GPIOGDEBOUNCE (GPIOG De-bounce Enable, 400D 2C2Ch) | 326 |
| Register 15-13 GPIOGDOSET (GPIOG Data Output Set, 400D 2C30h) | 327 |
| Register 15-14 GPIOGDOCLEAR (GPIOG Data Output Clear, 400D 2C34h) | 328 |
| Register 16-1 GPTACTL (GPTimer A Control, 400B 0000h) | 331 |
| Register 16-2 GPTACTR (GPTimer A Counter, 400B 0004h) | 331 |
| Register 17-1 GPTBCTL (GPTimer B Control, 400C 0000h) | 334 |
| Register 17-2 GPTBCTR (GPTimer B Counter, 400C 0004h) | 334 |
| Register 18-1 TACTL (Timer A Control, 4006 0000h) | 357 |
| Register 18-2 TAIN (Timer A Interrupt Control, 4006 0004h) | 358 |
| Register 18-3 TAPRD (Timer A Period, 4006 0008h) | 358 |
| Register 18-4 TACTR (Timer A Counter, 4006 000Ch) | 359 |
| Register 18-5 TAQEPCTL (Timer A QEP Control, 4006 0010h) | 360 |
| Register 18-6 TACCTL0 (Timer A CCR Control 0, 4006 0100h) | 362 |
| Register 18-7 TACCTR0 (Timer A CCR Counter 0, 4006 0104h) | 363 |
| Register 18-8 TACCTL1 (Timer A CCR Control 1, 4006 0108h) | 364 |
| Register 18-9 TACCTR1 (Timer A CCR Counter 1, 4006 010Ch) | 365 |
| Register 18-10 TACCTL2 (Timer A CCR Control 2, 4006 0110h) | 366 |
| Register 18-11 TACCTR2 (Timer A CCR Counter 2, 4006 0114h) | 367 |
| Register 18-12 TACCTL3 (Timer A CCR Control 3, 4006 0118h) | 368 |

| | |
|--|-----|
| Register 18-13 TACCTR3 (Timer A CCR Counter 3, 4006 011Ch) | 369 |
| Register 18-14 TACCTL4 (Timer A CCR Control 4, 4006 0120h)..... | 370 |
| Register 18-15 TACCTR4 (Timer A CCR Counter 4, 4006 0124h)..... | 371 |
| Register 18-16 TACCTL5 (Timer A CCR Control 5, 4006 0128h)..... | 372 |
| Register 18-17 TACCTR5 (Timer A CCR Counter 5, 4006 012Ch) | 373 |
| Register 18-18 TACCTL6 (Timer A CCR Control 6, 4006 0130h)..... | 374 |
| Register 18-19 TACCTR6 (Timer A CCR Counter 6, 4006 0134h)..... | 375 |
| Register 18-20 TACCTL7 (Timer A CCR Control 7, 4006 0138h)..... | 376 |
| Register 18-21 TACCTR7 (Timer A CCR Counter 7, 4006 013Ch) | 377 |
| Register 18-22 TADTGCTL0 (Timer A Dead-Time Generator Control 0, 4006 0200h) | 378 |
| Register 18-23 TADTGCTL1 (Timer A Dead-Time Generator Control 1, 4006 0204h) | 378 |
| Register 18-24 TADTGCTL2 (Timer A Dead-Time Generator Control 2, 4006 0208h) | 379 |
| Register 18-25 TADTGCTL3 (Timer A Dead-Time Generator Control 3, 4006 020Ch) | 379 |
| Register 19-1 TBCTL (Timer B Control, 4007 0000h)..... | 402 |
| Register 19-2 TBINT (Timer B Interrupt Control, 4007 0004h) | 403 |
| Register 19-3 TBPRD (Timer A Period, 4007 0008h) | 403 |
| Register 19-4 TBCTR (Timer B Counter, 4007 000Ch) | 404 |
| Register 19-5 TBQEPCTL (Timer B QEP Control, 4007 0010h)..... | 405 |
| Register 19-6 TBCCTL0 (Timer B CCR Control 0, 4007 0100h) | 407 |
| Register 19-7 TBCCTR0 (Timer B CCR Counter 0, 4007 0104h)..... | 408 |
| Register 19-8 TBCCTL1 (Timer A CCR Control 1, 4007 0108h) | 409 |
| Register 19-9 TBCCTR1 (Timer B CCR Counter 1, 4007 010Ch) | 410 |
| Register 19-10 TBCCTL2 (Timer B CCR Control 2, 4007 0110h)..... | 411 |
| Register 19-11 TBCCTR2 (Timer B CCR Counter 2, 4007 0114h)..... | 412 |
| Register 19-12 TBCCTL3 (Timer B CCR Control 3, 4007 0118h)..... | 413 |
| Register 19-13 TBCCTR3 (Timer B CCR Counter 3, 4007 011Ch) | 414 |
| Register 19-14 TBCCTL4 (Timer B CCR Control 4, 4007 0120h)..... | 415 |
| Register 19-15 TBCCTR4 (Timer B CCR Counter 4, 4007 0124h)..... | 416 |
| Register 19-16 TBCCTL5 (Timer B CCR Control 5, 4007 0128h)..... | 417 |
| Register 19-17 TBCCTR5 (Timer B CCR Counter 5, 4007 012Ch) | 418 |
| Register 19-18 TBCCTL6 (Timer B CCR Control 6, 4007 0130h)..... | 419 |
| Register 19-19 TBCCTR6 (Timer B CCR Counter 6, 4007 0134h)..... | 420 |
| Register 19-20 TBCCTL7 (Timer B CCR Control 7, 4007 0138h)..... | 421 |
| Register 19-21 TBCCTR7 (Timer B CCR Counter 7, 4007 013Ch) | 422 |
| Register 19-22 TBDTGCTL0 (Timer B Dead-Time Generator Control 0, 4007 0200h) | 423 |
| Register 19-23 TBDTGCTL1 (Timer B Dead-Time Generator Control 1, 4007 0204h) | 423 |
| Register 19-24 TBDTGCTL2 (Timer B Dead-Time Generator Control 2, 4007 0208h) | 424 |
| Register 19-25 TBDTGCTL3 (Timer B Dead-Time Generator Control 3, 4007 020Ch) | 424 |
| Register 20-1 TCCTL (Timer C Control, 4008 0000h) | 447 |
| Register 20-2 TCINT (Timer C Interrupt Control, 4008 0004h) | 448 |
| Register 20-3 TCPRD (Timer C Period, 4008 0008h)..... | 449 |
| Register 20-4 TCCTR (Timer C Counter, 4008 000Ch)..... | 449 |
| Register 20-5 TCQEPCTL (Timer C QEP Control, 4008 0010h) | 450 |

| | |
|---|-----|
| Register 20-6 TCCCTL0 (Timer C CCR Control 0, 4008 0100h) | 452 |
| Register 20-7 TCCCTR0 (Timer C CCR Counter 0, 4008 0104h) | 453 |
| Register 20-8 TCCCTL1 (Timer C CCR Control 1, 4008 0108h) | 454 |
| Register 20-9 TCCCTR1 (Timer C CCR Counter 1, 4008 010Ch)..... | 455 |
| Register 20-10 TCCCTL2 (Timer C CCR Control 2, 4008 0110h) | 456 |
| Register 20-11 TCCCTR2 (Timer C CCR Counter 2, 4008 0114h) | 457 |
| Register 20-12 TCCCTL3 (Timer C CCR Control 3, 4008 0118h) | 458 |
| Register 20-13 TCCCTR3 (Timer C CCR Counter 3, 4008 011Ch)..... | 459 |
| Register 20-14 TCCCTL4 (Timer C CCR Control 4, 4008 0120h) | 460 |
| Register 20-15 TCCCTR4 (Timer C CCR Counter 4, 4008 0124h) | 461 |
| Register 20-16 TCCCTL5 (Timer C CCR Control 5, 4008 0128h) | 462 |
| Register 20-17 TCCCTR5 (Timer C CCR Counter 5, 4008 012Ch)..... | 463 |
| Register 20-18 TCCCTL6 (Timer C CCR Control 6, 4008 0130h) | 464 |
| Register 20-19 TCCCTR6 (Timer C CCR Counter 6, 4008 0134h) | 465 |
| Register 20-20 TCCCTL7 (Timer C CCR Control 7, 4008 0138h) | 466 |
| Register 20-21 TCCCTR7 (Timer C CCR Counter 7, 4008 013Ch)..... | 467 |
| Register 20-22 TCDTGCTL0 (Timer C Dead-Time Generator Control 0, 4008 0200h)..... | 468 |
| Register 20-23 TCDTGCTL1 (Timer C Dead-Time Generator Control 1, 4008 0204h)..... | 468 |
| Register 20-24 TCDTGCTL2 (Timer C Dead-Time Generator Control 2, 4008 0208h)..... | 469 |
| Register 20-25 TCDTGCTL3 (Timer C Dead-Time Generator Control 3, 4008 020Ch)..... | 469 |
| Register 21-1 TDCTL (Timer D Control, 4009 0000h) | 493 |
| Register 21-2 TDINT (Timer D Interrupt Control, 4009 0004h) | 494 |
| Register 21-3 TDPRD (Timer D Period, 4009 0008h)..... | 495 |
| Register 21-4 TDCTR (Timer D Counter, 4009 000Ch)..... | 495 |
| Register 21-5 TDQEPCTL (Timer D QEP Control, 4009 0010h) | 496 |
| Register 21-6 TDCCTL0 (Timer D CCR Control 0, 4009 0100h) | 498 |
| Register 21-7 TDCCTR0 (Timer D CCR Counter 0, 4009 0104h) | 499 |
| Register 21-8 TDCCTL1 (Timer D CCR Control 1, 4009 0108h) | 500 |
| Register 21-9 TDCCTR1 (Timer D CCR Counter 1, 4009 010Ch)..... | 501 |
| Register 21-10 TDCCTL2 (Timer D CCR Control 2, 4009 0110h) | 502 |
| Register 21-11 TDCCTR2 (Timer D CCR Counter 2, 4009 0114h) | 503 |
| Register 21-12 TDCCTL3 (Timer D CCR Control 3, 4009 0118h) | 504 |
| Register 21-13 TDCCTR3 (Timer D CCR Counter 3, 4009 011Ch)..... | 505 |
| Register 21-14 TDCCTL4 (Timer D CCR Control 4, 4009 0120h) | 506 |
| Register 21-15 TDCCTR4 (Timer D CCR Counter 4, 4009 0124h) | 507 |
| Register 21-16 TDCCTL5 (Timer D CCR Control 5, 4009 0128h) | 508 |
| Register 21-17 TDCCTR5 (Timer D CCR Counter 5, 4009 012Ch)..... | 509 |
| Register 21-18 TDCCTL6 (Timer D CCR Control 6, 4009 0130h) | 510 |
| Register 21-19 TDCCTR6 (Timer D CCR Counter 6, 4009 0134h) | 511 |
| Register 21-20 TDCCTL7 (Timer D CCR Control 7, 4009 0138h) | 512 |
| Register 21-21 TDCCTR7 (Timer D CCR Counter 7, 4009 013Ch)..... | 513 |
| Register 21-22 TDDTGCTL0 (Timer D Dead-Time Generator Control 0, 4009 0200h)..... | 514 |
| Register 21-23 TDDTGCTL1 (Timer D Dead-Time Generator Control 1, 4009 0204h)..... | 514 |

| | |
|--|-----|
| Register 21-24 TDDTGCTL2 (Timer D Dead-Time Generator Control 2, 4009 0208h)..... | 515 |
| Register 21-25 TDDTGCTL3 (Timer D Dead-Time Generator Control 3, 4009 020Ch) | 515 |
| Register 22-1 EMUXCTL (EMUX Control, 4000 0000h) | 531 |
| Register 22-2 EMUXDATA (EMUX Data, 4000 0004h) | 531 |
| Register 22-3 ADCCTL (ADC Configuration, 4000 0008h) | 532 |
| Register 22-4 ADCRES (ADC Result, 4000 000Ch)..... | 534 |
| Register 22-5 ADCINT (ADC Interrupt Control, 4000 0010h) | 534 |
| Register 22-6 DTSETRIGENT0TO3 (DTSE Trigger Entry 0 to 3, 4000 0040h) | 537 |
| Register 22-7 DTSETRIGENT4TO7 (DTSE Trigger Entry 4 to 7, 4000 0044h) | 538 |
| Register 22-8 DTSETRIGENT8TO11 (DTSE Trigger Entry 8 to 11, 4000 0048h) | 539 |
| Register 22-9 DTSETRIGENT12TO15 (DTSE Trigger Entry 12 to 15, 4000 004Ch)..... | 540 |
| Register 22-10 DTSETRIGENT16TO19 (DTSE Trigger Entry 16 to 19, 4000 0050h) | 541 |
| Register 22-11 DTSETRIGENT20TO23 (DTSE Trigger Entry 20 to 23, 4000 0054h) | 542 |
| Register 22-12 DTSETRIGENT24TO27 (DTSE Trigger Entry 24 to 27, 4000 0058h) | 544 |
| Register 22-13 DTSETRIGENT28TO31 (DTSE Trigger Entry 28 to 31, 4000 005Ch)..... | 545 |
| Register 22-14 DTSESEQCFG0 (DTSE Sequence Configuration 0, 4000 0080h) | 547 |
| Register 22-15 DTSESEQCFG1 (DTSE Sequence Configuration 1, 4000 0084h) | 548 |
| Register 22-16 DTSESEQCFG2 (DTSE Sequence Configuration 2, 4000 0088h)..... | 549 |
| Register 22-17 DTSESEQCFG3 (DTSE Sequence Configuration 3, 4000 008Ch) | 550 |
| Register 22-18 DTSESEQCFG4 (DTSE Sequence Configuration 4, 4000 0090h) | 551 |
| Register 22-19 DTSESEQCFG5 (DTSE Sequence Configuration 5, 4000 0094h) | 552 |
| Register 22-20 DTSESEQCFG6 (DTSE Sequence Configuration 6, 4000 0098h) | 553 |
| Register 22-21 DTSESEQCFG7 (DTSE Sequence Configuration 7, 4000 009Ch) | 554 |
| Register 22-22 DTSESEQCFG8 (DTSE Sequence Configuration 8, 4000 00A0h)..... | 555 |
| Register 22-23 DTSESEQCFG9 (DTSE Sequence Configuration 9, 4000 00A4h)..... | 556 |
| Register 22-24 DTSESEQCFG10 (DTSE Sequence Configuration 10, 4000 00A8h)..... | 557 |
| Register 22-25 DTSESEQCFG11 (DTSE Sequence Configuration 11, 4000 00ACh) | 558 |
| Register 22-26 DTSESEQCFG12 (DTSE Sequence Configuration 12, 4000 00B0h)..... | 559 |
| Register 22-27 DTSESEQCFG13 (DTSE Sequence Configuration 13, 4000 00B4h)..... | 560 |
| Register 22-28 DTSESEQCFG14 (DTSE Sequence Configuration 14, 4000 00B8h)..... | 561 |
| Register 22-29 DTSESEQCFG15 (DTSE Sequence Configuration 15, 4000 00BCh) | 562 |
| Register 22-30 DTSESEQCFG16 (DTSE Sequence Configuration 16, 4000 00C0h) | 563 |
| Register 22-31 DTSESEQCFG17 (DTSE Sequence Configuration 17, 4000 00C4h) | 564 |
| Register 22-32 DTSESEQCFG18 (DTSE Sequence Configuration 18, 4000 00C8h) | 565 |
| Register 22-33 DTSESEQCFG19 (DTSE Sequence Configuration 19, 4000 00CCh) | 566 |
| Register 22-34 DTSESEQCFG20 (DTSE Sequence Configuration 20, 4000 00D0h) | 567 |
| Register 22-35 DTSESEQCFG21 (DTSE Sequence Configuration 21, 4000 00D4h) | 568 |
| Register 22-36 DTSESEQCFG22 (DTSE Sequence Configuration 22, 4000 00D8h) | 569 |
| Register 22-37 DTSESEQCFG23 (DTSE Sequence Configuration 23, 4000 00DCh)..... | 570 |
| Register 22-38 DTSERES0 (DTSE Conversion Result 0, 4000 0100h)..... | 571 |
| Register 22-39 DTSERES1 (DTSE Conversion Result 1, 4000 0104h)..... | 571 |
| Register 22-40 DTSERES2 (DTSE Conversion Result 2, 4000 0108h)..... | 571 |
| Register 22-41 DTSERES3 (DTSE Conversion Result 3, 4000 010Ch) | 571 |

| | |
|--|-----|
| Register 22-42 DTSERES4 (DTSE Conversion Result 4, 4000 0110h)..... | 572 |
| Register 22-43 DTSERES5 (DTSE Conversion Result 5, 4000 0114h)..... | 572 |
| Register 22-44 DTSERES6 (DTSE Conversion Result 6, 4000 0118h)..... | 572 |
| Register 22-45 DTSERES7 (DTSE Conversion Result 7, 4000 011Ch) | 572 |
| Register 22-46 DTSERES8 (DTSE Conversion Result 8, 4000 0120h)..... | 573 |
| Register 22-47 DTSERES9 (DTSE Conversion Result 9, 4000 0124h)..... | 573 |
| Register 22-48 DTSERES10 (DTSE Conversion Result 10, 4000 0128h)..... | 573 |
| Register 22-49 DTSERES11 (DTSE Conversion Result 11, 4000 012Ch) | 573 |
| Register 22-50 DTSERES12 (DTSE Conversion Result 12, 4000 0130h)..... | 574 |
| Register 22-51 DTSERES13 (DTSE Conversion Result 13, 4000 0134h)..... | 574 |
| Register 22-52 DTSERES14 (DTSE Conversion Result 14, 4000 0138h)..... | 574 |
| Register 22-53 DTSERES15 (DTSE Conversion Result 15, 4000 013Ch) | 574 |
| Register 22-54 DTSERES16 (DTSE Conversion Result 16, 4000 1240h)..... | 575 |
| Register 22-55 DTSERES17 (DTSE Conversion Result 17, 4000 0144h)..... | 575 |
| Register 22-56 DTSERES18 (DTSE Conversion Result 18, 4000 0148h)..... | 575 |
| Register 22-57 DTSERES19 (DTSE Conversion Result 19, 4000 014Ch) | 575 |
| Register 22-58 DTSERES20 (DTSE Conversion Result 20, 4000 0150h)..... | 576 |
| Register 22-59 DTSERES21 (DTSE Conversion Result 21, 4000 0154h)..... | 576 |
| Register 22-60 DTSERES22 (DTSE Conversion Result 22, 4000 0158h)..... | 576 |
| Register 22-61 DTSERES23 (DTSE Conversion Result 23, 4000 015Ch) | 576 |
| Register 23-1 CRCCTL (CRC Control, 400D 1000h) | 580 |
| Register 23-2 CRCDATAIN (CRC Data Input, 400D 1004h) | 580 |
| Register 23-3 CRCSEED (CRC Seed Value, 400D 1008h)..... | 580 |
| Register 23-4 CRCOUT (CRC Data Output, 400D 100Ch)..... | 581 |
| Register 24-1 UARTARBR (UART A Receive Buffer Register, 4002 0000h)..... | 591 |
| Register 24-2 UARTATHR (UART A Transmit Holding Register, 4002 0004h)..... | 591 |
| Register 24-3 UARTADLR (UART A Divisor Latch Register, 4002 0008h)..... | 591 |
| Register 24-4 UARТАIER (UART A Interrupt Enable Register, 4002 000Ch)..... | 592 |
| Register 24-5 UARТАIIR (UART A Interrupt Identification Register, 4002 0010h)..... | 592 |
| Register 24-6 UARТАFCR (UART A FIFO Control Register, 4002 0014h)..... | 593 |
| Register 24-7 UARТАLCR (UART A Line Control Register, 4002 0018h) | 594 |
| Register 24-8 UARТАLSR (UART A Line Status Register, 4002 0020h)..... | 595 |
| Register 24-9 UARТАSCR (UART A Scratch Pad Register, 4002 0028h)..... | 596 |
| Register 24-10 UARТАEFR (UART A Enhanced Feature Register, 4002 002Ch)..... | 596 |
| Register 24-11 SSPACON (SSP A Control Register, 4002 0000h) | 605 |
| Register 24-12 SSPASTAT (SSP A Status Register, 4002 0004h)..... | 606 |
| Register 24-13 SSPADAT (SSP A Data Register, 4002 0008h) | 607 |
| Register 24-14 SSPACLK (SSP A Clock Register, 4002 000Ch) | 608 |
| Register 24-15 SSPAIMSC (SSP A Interrupt Mask Set and Clear Enable Register, 4002 0010h) | 608 |
| Register 24-16 SSPARIS (SSP A Raw Interrupt Status Register, 4002 0014h)..... | 609 |
| Register 24-17 SSPAMIS (SSP A Masked Interrupt Status Register, 4002 0018h)..... | 609 |
| Register 24-18 SSPAICLR (SSP A Interrupt Clear Register, 4002 001Ch) | 610 |

| | |
|---|-----|
| Register 24-19 SSPASSCR (SSP A Slave Select Configuration Register, 4002 0028h) | 611 |
| Register 25-1 UARTBRBR (UART B Receive Buffer Register, 4003 0000h) | 622 |
| Register 25-2 UARTBTHR (UART B Transmit Holding Register, 4003 0004h) | 622 |
| Register 25-3 UARTBDLR (UART B Divisor Latch Register, 4003 0008h) | 622 |
| Register 25-4 UARTBIER (UART B Interrupt Enable Register, 4003 000Ch) | 623 |
| Register 25-5 UARTBIIR (UART B Interrupt Identification Register, 4003 0010h) | 623 |
| Register 25-6 UARTBFCR (UART B FIFO Control Register, 4003 0014h) | 625 |
| Register 25-7 UARTBLCR (UART B Line Control Register, 4003 0018h) | 626 |
| Register 25-8 UARTBLSR (UART B Line Status Register, 4003 0020h) | 627 |
| Register 25-9 UARTBSCR (UART B Scratch Pad Register, 4003 0028h) | 628 |
| Register 25-10 UARTBEFR (UART B Enhanced Feature Register, 4003 002Ch) | 628 |
| Register 25-11 SSPBCON (SSP B Control Register, 4003 0000h) | 636 |
| Register -12 SSPBSTAT (SSP B Status Register, 4003 0004h) | 637 |
| Register 25-13 SSPBDAT (SSP B Data Register, 4003 0008h) | 638 |
| Register 25-14 SSPBCLK (SSP B Clock Register, 4003 000Ch) | 639 |
| Register 25-15 SSPBIMSC (SSP B Interrupt Mask Set and Clear Enable Register, 4003 0010h) | 639 |
| Register 25-16 SSPBRIS (SSP B Raw Interrupt Status Register, 4003 0014h) | 640 |
| Register 25-17 SSPBMIS (SSP B Masked Interrupt Status Register, 4003 0018h) | 640 |
| Register 25-18 SSPBICLR (SSP B Interrupt Clear Register, 4003 001Ch) | 641 |
| Register 25-19 SSPBSSCR (SSP B Slave Select Configuration Register, 4003 0028h) | 642 |
| Register 26-1 UARTCRBR (UART C Receive Buffer Register, 4004 0000h) | 653 |
| Register 26-2 UARTCTHR (UART C Transmit Holding Register, 4004 0004h) | 653 |
| Register 26-3 UARTCDLR (UART C Divisor Latch Register, 4004 0008h) | 653 |
| Register 26-4 UARTCIER (UART C Interrupt Enable Register, 4004 000Ch) | 654 |
| Register 26-5 UARTCIIR (UART C Interrupt Identification Register, 4004 0010h) | 654 |
| Register 26-6 UARTCFCR (UART C FIFO Control Register, 4004 0014h) | 656 |
| Register 26-7 UARTCLCR (UART C Line Control Register, 4004 0018h) | 657 |
| Register 26-8 UARTCLSR (UART C Line Status Register, 4004 0020h) | 658 |
| Register 26-9 UARTCSCR (UART C Scratch Pad Register, 4004 0028h) | 659 |
| Register 26-10 UARTCEFR (UARTC Enhanced Feature Register, 4004 002Ch) | 659 |
| Register 26-11 SSPCCON (SSP C Control Register, 4004 0000h) | 667 |
| Register 26-12 SSPCSTAT (SSP C Status Register, 4004 0004h) | 668 |
| Register 26-13 SSPCDAT (SSP C Data Register, 4004 0008h) | 669 |
| Register 26-14 SSPCCLK (SSP C Clock Register, 4004 000Ch) | 670 |
| Register 26-15 SSPCIMSC (SSP C Interrupt Mask Set and Clear Enable Register, 4004 0010h) | 670 |
| Register 26-16 SSPCRIS (SSP C Raw Interrupt Status Register, 4004 0014h) | 671 |
| Register 26-17 SSPCMIS (SSP C Masked Interrupt Status Register, 4004 0018h) | 671 |
| Register 26-18 SSPCICLR (SSP C Interrupt Clear Register, 4004 001Ch) | 672 |
| Register 26-19 SSPCSSCR (SSP C Slave Select Configuration Register, 4004 0028h) | 673 |
| Register 27-1 UARTDRBR (UART D Receive Buffer Register, 4005 0000h) | 684 |
| Register 27-2 UARTDTHR (UART D Transmit Holding Register, 4005 0004h) | 684 |

| | |
|---|-----|
| Register 27-3 UARTDDL (UART D Divisor Latch Register, 4005 0008h)..... | 684 |
| Register 27-4 UARTDIER (UART D Interrupt Enable Register, 4005 000Ch) | 685 |
| Register 27-5 UARTDIIR (UART D Interrupt Identification Register, 4005 0010h)..... | 685 |
| Register 27-6 UARTDFCR (UART D FIFO Control Register, 4005 0014h) | 686 |
| Register 27-7 UARTDLCR (UART D Line Control Register, 4005 0018h)..... | 687 |
| Register 27-8 UARTDLR (UART D Line Status Register, 4005 0020h)..... | 688 |
| Register 27-9 UARTDSCR (UART D Scratch Pad Register, 4005 0028h) | 689 |
| Register 27-10 UARTDEFER (UART D Enhanced Feature Register, 4005 002Ch) | 689 |
| Register 27-11 SSPDCON (SSP D Control Register, 4005 0000h) | 697 |
| Register 27-12 SSPDSTAT (SSP D Status Register, 4005 0004h) | 698 |
| Register 27-13 SSPDDAT (SSP D Data Register, 4005 0008h)..... | 699 |
| Register 27-14 SSPDCLK (SSP D Clock Register, 4005 000Ch)..... | 700 |
| Register 27-15 SSPDIMSC (SSP D Interrupt Mask Set and Clear Enable Register, 4005 0010h) | 700 |
| Register 27-16 SSPDRIS (SSP D Raw Interrupt Status Register, 4005 0014h) | 701 |
| Register 27-17 SSPDMIS (SSP D Masked Interrupt Status Register, 4005 0018h) | 701 |
| Register 27-18 SSPDICLR (SSP D Interrupt Clear Register, 4005 001Ch) | 702 |
| Register 27-19 SSPDSSCR (SSP D Slave Select Configuration Register, 4005 0028h)..... | 703 |
| Register 28-1 I2CCONSET (I2C Control Set, 4001 0000h) | 712 |
| Register 28-2 I2CCONCLR (I2C Control Clear, 4001 0004h)..... | 713 |
| Register 28-3 I2CSTAT (I2C Status, 4001 0008h) | 713 |
| Register 28-4 I2CDAT (I2C Data, 4001 000Ch) | 714 |
| Register 28-5 I2CCLK (I2C Clock, 4001 0010h)..... | 714 |
| Register 28-6 I2CADR0 (I2C Slave Address 0, 4001 0014h) | 714 |
| Register 28-7 I2CADRM0 (I2C Slave Address Mask 0, 4001 0018h)..... | 714 |
| Register 28-8 I2CXADR0 (I2C Extended Slave Address 0, 4001 001Ch)..... | 715 |
| Register 28-9 I2CXADRM0 (I2C Extended Slave Address Mask 0, 4001 0020h)..... | 715 |
| Register 28-10 I2CRST (I2C Software Reset, 4001 0024h) | 716 |
| Register 28-11 I2CADR1 (I2C Slave Address 1, 4001 0028h) | 716 |
| Register 28-12 I2CADRM1 (I2C Slave Address Mask 1, 4001 002Ch) | 716 |
| Register 28-13 I2CADR2 (I2C Slave Address 2, 4001 0030h) | 717 |
| Register 28-14 I2CADRM2 (I2C Slave Address Mask 2, 4001 0034h)..... | 717 |
| Register 28-13 I2CADR3 (I2C Slave Address 3, 4001 0038h) | 717 |
| Register 28-14 I2CADRM3 (I2C Slave Address Mask 3, 4001 003Ch) | 717 |
| Register 29-1 CAN_ISR_SR_CMR_MR (CAN Mode/Command/Status/ISR Register, 400A 0000h) | 723 |
| Register 29-2 CAN_BTR1_BTR0_RMC_IMR (CAN Bus Timing/RX Message Counter/Interrupt Mask Register, 400A 0004h)..... | 725 |
| Register 29-3 CANTXBUF (CAN Transmit Buffer Register, 400A 0008h) | 726 |
| Register 29-4 CANRXBUF (CAN Receive Buffer Register, 400A 000Ch) | 726 |
| Register 29-5 CANACR (CAN Acceptance Code Register, 400A 0010h)..... | 726 |
| Register 29-6 CANACRM (CAN Acceptance Code Mask Register, 400A 0014h) | 726 |

Register 29-7 CAN_ALC_TXERR_RXERR_ECC (CAN Error Code Capture Register, 400A
0018h)727

1 OVERVIEW

This document is the PAC55XX Family User Guide. It details the operation of all of the peripherals in the PAC55XX family of controllers.

Each feature is described in detail with block diagrams and all programming registers needed for its operation. Not every peripheral is available in each PAC device, so the user should consult the device data sheet to determine which peripherals and IOs are available in your selected devices.

2 STYLE AND FORMATTING CONVENTIONS

This chapter describes the formatting and styles used throughout this document.

2.1 Number Representation

Numbers other than decimal will have a postfix indicator. All numbers use little endian formatting, with the most significant bit/digit to the left. Digits for binary and hexadecimal representation are grouped with a single space every four digits to improve readability. Binary numbers use “b” as a postfix and hexadecimal numbers use “h” as a postfix.

For example, 1011b binary = Bh hexadecimal = 11 decimal.

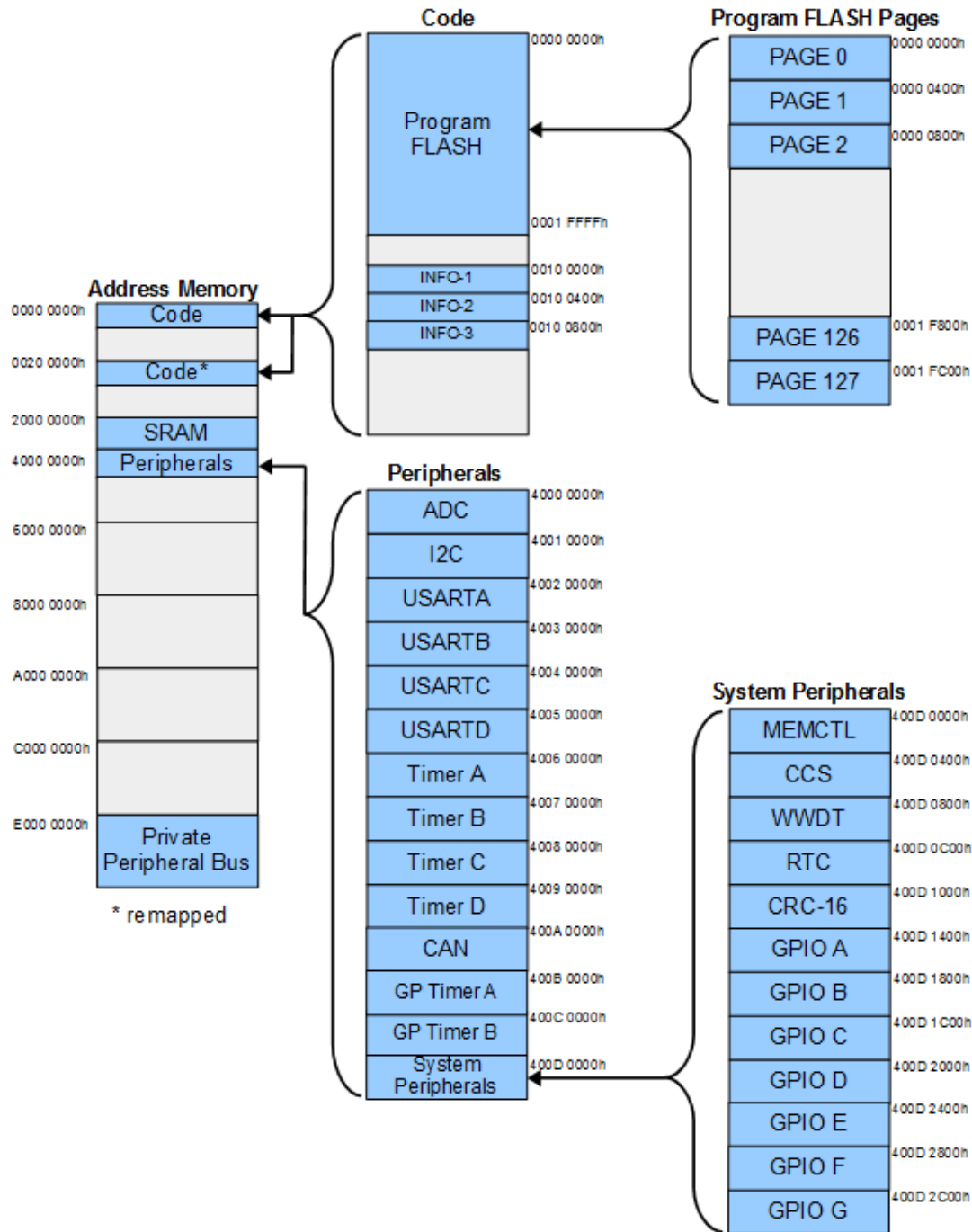
2.2 Formatting Styles

| TYPE | EXAMPLE | DESCRIPTION |
|--------------------------------------|---------------------------|--|
| Register Name | RTCCTL | Register names use a capital letter and boldface type. |
| Register Bit(s) | RTCCTL.RTCCLKDIV | Register bits are always represented with the register name separated with a period. |
| Function selected by register bit(s) | [RTCCTL.RTCCLKDIV] | Within text blocks, functions selected with a register bit setting are set in brackets. For example [RTCCTL.RTCCLKDIV] means divider settings /2 to /65536. |
| Pin Function | PA5 | Pin functions use capital letters |
| Internal signals | <i>PWMA3</i> | Internal signals use <i>italicized</i> font. |
| Formulas | CLK = FCLK / DIV | Formulas use monospaced text. |
| Links | Link | Hyperlinks are <u>underlined and blue</u> . |
| CPU Mnemonic | MRS | CPU Mnemonic uses monospaced text. |
| Operands | <i>{Rd, }, Rn, Rm</i> | Operands use <i>monospaced italic</i> text. |
| Code examples | b loopA | Code examples use monospaced text. |

3 MEMORY AND REGISTER MAP

3.1 Memory Map

Figure 3-1 Memory Map



3.2 FLASH Memory Map

Table 3-1 Embedded FLASH Register Map

| ADDRESS | NAME | DESCRIPTION |
|------------|-----------|----------------|
| 0000 0000h | EFLASHP0 | EFLASH page 0 |
| 0000 0400h | EFLASHP1 | EFLASH page 1 |
| 0000 0800h | EFLASHP2 | EFLASH page 2 |
| 0000 0C00h | EFLASHP3 | EFLASH page 3 |
| 0000 1000h | EFLASHP4 | EFLASH page 4 |
| 0000 1400h | EFLASHP5 | EFLASH page 5 |
| 0000 1800h | EFLASHP6 | EFLASH page 6 |
| 0000 1C00h | EFLASHP7 | EFLASH page 7 |
| 0000 2000h | EFLASHP8 | EFLASH page 8 |
| 0000 2400h | EFLASHP9 | EFLASH page 9 |
| 0000 2800h | EFLASHP10 | EFLASH page 10 |
| 0000 2C00h | EFLASHP11 | EFLASH page 11 |
| 0000 3000h | EFLASHP12 | EFLASH page 12 |
| 0000 3400h | EFLASHP13 | EFLASH page 13 |
| 0000 3800h | EFLASHP14 | EFLASH page 14 |
| 0000 3C00h | EFLASHP15 | EFLASH page 15 |
| 0000 4000h | EFLASHP16 | EFLASH page 16 |
| 0000 4400h | EFLASHP17 | EFLASH page 17 |
| 0000 4800h | EFLASHP18 | EFLASH page 18 |
| 0000 4C00h | EFLASHP19 | EFLASH page 19 |
| 0000 5000h | EFLASHP20 | EFLASH page 20 |
| 0000 5400h | EFLASHP21 | EFLASH page 21 |
| 0000 5800h | EFLASHP22 | EFLASH page 22 |
| 0000 5C00h | EFLASHP23 | EFLASH page 23 |
| 0000 6000h | EFLASHP24 | EFLASH page 24 |
| 0000 6400h | EFLASHP25 | EFLASH page 25 |
| 0000 6800h | EFLASHP26 | EFLASH page 26 |
| 0000 6C00h | EFLASHP27 | EFLASH page 27 |
| 0000 7000h | EFLASHP28 | EFLASH page 28 |
| 0000 7400h | EFLASHP29 | EFLASH page 29 |
| 0000 7800h | EFLASHP30 | EFLASH page 30 |
| 0000 7C00h | EFLASHP31 | EFLASH page 31 |

| | | |
|-------------------|-----------|----------------|
| 0000 8000h | EFLASHP32 | EFLASH page 32 |
| 0000 8400h | EFLASHP33 | EFLASH page 33 |
| 0000 8800h | EFLASHP34 | EFLASH page 34 |
| 0000 8C00h | EFLASHP35 | EFLASH page 35 |
| 0000 9000h | EFLASHP36 | EFLASH page 36 |
| 0000 9400h | EFLASHP37 | EFLASH page 37 |
| 0000 9800h | EFLASHP38 | EFLASH page 38 |
| 0000 9C00h | EFLASHP39 | EFLASH page 39 |
| 0000 A000h | EFLASHP40 | EFLASH page 40 |
| 0000 A400h | EFLASHP41 | EFLASH page 41 |
| 0000 A800h | EFLASHP42 | EFLASH page 42 |
| 0000 AC00h | EFLASHP43 | EFLASH page 43 |
| 0000 B000h | EFLASHP44 | EFLASH page 44 |
| 0000 B400h | EFLASHP45 | EFLASH page 45 |
| 0000 B800h | EFLASHP46 | EFLASH page 46 |
| 0000 BC00h | EFLASHP47 | EFLASH page 47 |
| 0000 C000h | EFLASHP48 | EFLASH page 48 |
| 0000 C400h | EFLASHP49 | EFLASH page 49 |
| 0000 C800h | EFLASHP50 | EFLASH page 50 |
| 0000 CC00h | EFLASHP51 | EFLASH page 51 |
| 0000 D000h | EFLASHP52 | EFLASH page 52 |
| 0000 D400h | EFLASHP53 | EFLASH page 53 |
| 0000 D800h | EFLASHP54 | EFLASH page 54 |
| 0000 DC00h | EFLASHP55 | EFLASH page 55 |
| 0000 E000h | EFLASHP56 | EFLASH page 56 |
| 0000 E400h | EFLASHP57 | EFLASH page 57 |
| 0000 E800h | EFLASHP58 | EFLASH page 58 |
| 0000 EC00h | EFLASHP59 | EFLASH page 59 |
| 0000 F000h | EFLASHP60 | EFLASH page 60 |
| 0000 F400h | EFLASHP61 | EFLASH page 61 |
| 0000 F800h | EFLASHP62 | EFLASH page 62 |
| 0000 FC00h | EFLASHP63 | EFLASH page 63 |
| 0001 0000h | EFLASHP64 | EFLASH page 64 |
| 0001 0400h | EFLASHP65 | EFLASH page 65 |
| 0001 0800h | EFLASHP66 | EFLASH page 66 |
| 0001 0C00h | EFLASHP67 | EFLASH page 67 |

| | | |
|-------------------|------------|-----------------|
| 0001 1000h | EFLASHP68 | EFLASH page 68 |
| 0001 1400h | EFLASHP69 | EFLASH page 69 |
| 0001 1800h | EFLASHP70 | EFLASH page 70 |
| 0001 1C00h | EFLASHP71 | EFLASH page 71 |
| 0001 2000h | EFLASHP72 | EFLASH page 72 |
| 0001 2400h | EFLASHP73 | EFLASH page 73 |
| 0001 2800h | EFLASHP74 | EFLASH page 74 |
| 0001 2C00h | EFLASHP75 | EFLASH page 75 |
| 0001 3000h | EFLASHP76 | EFLASH page 76 |
| 0001 3400h | EFLASHP77 | EFLASH page 77 |
| 0001 3800h | EFLASHP78 | EFLASH page 78 |
| 0001 3C00h | EFLASHP79 | EFLASH page 79 |
| 0001 4000h | EFLASHP80 | EFLASH page 80 |
| 0001 4400h | EFLASHP81 | EFLASH page 81 |
| 0001 4800h | EFLASHP82 | EFLASH page 82 |
| 0001 4C00h | EFLASHP83 | EFLASH page 83 |
| 0001 5000h | EFLASHP84 | EFLASH page 84 |
| 0001 5400h | EFLASHP85 | EFLASH page 85 |
| 0001 5800h | EFLASHP86 | EFLASH page 86 |
| 0001 5C00h | EFLASHP87 | EFLASH page 87 |
| 0001 6000h | EFLASHP88 | EFLASH page 88 |
| 0001 6400h | EFLASHP89 | EFLASH page 89 |
| 0001 6800h | EFLASHP90 | EFLASH page 90 |
| 0001 6C00h | EFLASHP91 | EFLASH page 91 |
| 0001 7000h | EFLASHP92 | EFLASH page 92 |
| 0001 7400h | EFLASHP93 | EFLASH page 93 |
| 0001 7800h | EFLASHP94 | EFLASH page 94 |
| 0001 7C00h | EFLASHP95 | EFLASH page 95 |
| 0001 8000h | EFLASHP96 | EFLASH page 96 |
| 0001 8400h | EFLASHP97 | EFLASH page 97 |
| 0001 8800h | EFLASHP98 | EFLASH page 98 |
| 0001 8C00h | EFLASHP99 | EFLASH page 99 |
| 0001 9000h | EFLASHP100 | EFLASH page 100 |
| 0001 9400h | EFLASHP101 | EFLASH page 101 |
| 0001 9800h | EFLASHP102 | EFLASH page 102 |
| 0001 9C00h | EFLASHP103 | EFLASH page 103 |

| | | |
|-------------------|------------|-----------------|
| 0001 A000h | EFLASHP104 | EFLASH page 104 |
| 0001 A400h | EFLASHP105 | EFLASH page 105 |
| 0001 A800h | EFLASHP106 | EFLASH page 106 |
| 0001 AC00h | EFLASHP107 | EFLASH page 107 |
| 0001 B000h | EFLASHP108 | EFLASH page 108 |
| 0001 B400h | EFLASHP109 | EFLASH page 109 |
| 0001 B800h | EFLASHP110 | EFLASH page 110 |
| 0001 BC00h | EFLASHP111 | EFLASH page 111 |
| 0001 C000h | EFLASHP112 | EFLASH page 112 |
| 0001 C400h | EFLASHP113 | EFLASH page 113 |
| 0001 C800h | EFLASHP114 | EFLASH page 114 |
| 0001 CC00h | EFLASHP115 | EFLASH page 115 |
| 0001 D000h | EFLASHP116 | EFLASH page 116 |
| 0001 D400h | EFLASHP117 | EFLASH page 117 |
| 0001 D800h | EFLASHP118 | EFLASH page 118 |
| 0001 DC00h | EFLASHP129 | EFLASH page 119 |
| 0001 E000h | EFLASHP120 | EFLASH page 120 |
| 0001 E400h | EFLASHP121 | EFLASH page 121 |
| 0001 E800h | EFLASHP122 | EFLASH page 122 |
| 0001 EC00h | EFLASHP123 | EFLASH page 123 |
| 0001 F000h | EFLASHP124 | EFLASH page 124 |
| 0001 F400h | EFLASHP125 | EFLASH page 125 |
| 0001 F800h | EFLASHP126 | EFLASH page 126 |
| 0001 FC00h | EFLASHP127 | EFLASH page 127 |

3.3 INFO-1 Register Map

Table 3-2 INFO-1 Register Map

| ADDRESS | OFFSET | | | |
|-------------------------|-----------|---|---|---|
| | 3 | 2 | 1 | 0 |
| 0010 0000h | UNIQUEID | | | |
| 0010 0004h | | | | |
| 0010 0008h | | | | |
| 0010 000Ch | RESERVED | | | |
| 0010 0010h | FLASHTRIM | | | |
| 0010 0014h | | | | |
| 0010 0018h | | | | |
| 0010 001Ch – 0010 03FFh | RESERVED | | | |

3.4 INFO-2 Register Map

Table 3-3 INFO-2 Register Map

| ADDRESS | OFFSET | | | |
|-------------------------|----------|---|-----------|-------|
| | 3 | 2 | 1 | 0 |
| 0010 0400h | Reserved | | CHECKSUM | |
| 0010 0404h | PACIDR | | | |
| 0010 0408h | DIEREV | | | |
| 0010 040Ch | PKGDC | | | |
| 0010 0410h | FTTEMP | | TEMPS | |
| 0010 0414h | ADCGAIN | | ADCOFFSET | |
| 0010 0418h | Reserved | | VREF | |
| 0010 041Ch | ROSC | | | |
| 0010 0420h | Reserved | | | SECEN |
| 0010 0424h | MAXADDR | | | |
| 0010 0428h | ROSCTRIM | | | |
| 0010 042Ch | CLKREF | | | |
| 0010 0430h-0010 043B | Reserved | | | |
| 0010 003C | SWDFUSE | | | |
| 0010 0440h – 0010 07FFh | Reserved | | | |

Table 3-4 INFO-1, INFO-2 Field Name and Description

| NAME | DESCRIPTION |
|-----------|--|
| UNIQUEID | 96-bit unique device ID. |
| FLASHTRIM | 96-bit FLASH trim value. |
| CHECKSUM | 16-bit checksum of INFO-2 from address 0010 0002h to 0010 03FFh. |
| | |
| PACIDR | Device part number and revision. |
| DIEREV | Die revision. |
| PKGDC | Package topside date code. |
| TEMPS | Internal temperature sensor reading at FTTEMP temperature in ADC counts. |
| FTTEMP | Test temperature for internal temperature sensor in °C. |
| ADCOFFSET | ADC offset in mV. |
| ADCGAIN | ADC gain in mV. |
| VREF | Voltage reference in mV. |
| ROSC | ROSC frequency measured in Hz (nominal: 16MHz). |
| SECEN | Code Security Enabled |
| MAXADDR | Maximum FLASH address |
| ROSCTRIM | 7-bit ROSC trim value for 16MHz ROSC Frequency (bits 6:0). |
| CLKREF | CLKREF frequency measured in Hz (nominal: 4MHz). |
| SWDFUSE | SWD Fuse Key Value. |

3.5 INFO-3 Register Map

Table 3-5 INFO-3 Register Map

| ADDRESS | CONTENTS |
|-------------------------|---------------------------|
| 0010 0800h – 0010 0BFFh | User Defined ¹ |

3.6 Peripheral Register Maps

Table 3-6 ADC Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|-----------------------------|--------|------------|
| EMUXCTL | 4000 0000h | EMUX Control | RW | 0000 0000h |
| EMUXDATA | 4000 0004h | EMUX Data | RW | 0000 0000h |
| ADCCTL | 4000 0008h | ADC Control | RW | 0000 0000h |
| ADCRES | 4000 000Ch | ADC Result | RW | 0000 0000h |
| ADCINT | 4000 0010h | ADC Interrupt Control | RW | 0000 0000h |
| DTSETRIGENT0TO3 | 4000 0040h | DTSE Trigger Entry 0 to 3 | RW | 0000 0000h |
| DTSETRIGENT4TO7 | 4000 0044h | DTSE Trigger Entry 4 to 7 | RW | 0000 0000h |
| DTSETRIGENT8TO11 | 4000 0048h | DTSE Trigger Entry 8 to 11 | RW | 0000 0000h |
| DTSETRIGENT12TO15 | 4000 004Ch | DTSE Trigger Entry 12 to 15 | RW | 0000 0000h |
| DTSETRIGENT16TO19 | 4000 0050h | DTSE Trigger Entry 16 to 19 | RW | 0000 0000h |
| DTSETRIGENT20TO23 | 4000 0054h | DTSE Trigger Entry 20 to 23 | RW | 0000 0000h |
| DTSETRIGENT24TO27 | 4000 0058h | DTSE Trigger Entry 24 to 27 | RW | 0000 0000h |
| DTSETRIGENT28TO31 | 4000 005Ch | DTSE Trigger Entry 28 to 31 | RW | 0000 0000h |
| DTSESEQCFG0 | 4000 0080h | DTSE Sequence Config 0 | RW | 0000 0000h |
| DTSESEQCFG1 | 4000 0084h | DTSE Sequence Config 1 | RW | 0000 0000h |
| DTSESEQCFG2 | 4000 0088h | DTSE Sequence Config 2 | RW | 0000 0000h |
| DTSESEQCFG3 | 4000 008Ch | DTSE Sequence Config 3 | RW | 0000 0000h |
| DTSESEQCFG4 | 4000 0090h | DTSE Sequence Config 4 | RW | 0000 0000h |
| DTSESEQCFG5 | 4000 0094h | DTSE Sequence Config 5 | RW | 0000 0000h |
| DTSESEQCFG6 | 4000 0098h | DTSE Sequence Config 6 | RW | 0000 0000h |
| DTSESEQCFG7 | 4000 009Ch | DTSE Sequence Config 7 | RW | 0000 0000h |
| DTSESEQCFG8 | 4000 00A0h | DTSE Sequence Config 8 | RW | 0000 0000h |
| DTSESEQCFG9 | 4000 00A4h | DTSE Sequence Config 9 | RW | 0000 0000h |
| DTSESEQCFG10 | 4000 00A8h | DTSE Sequence Config 10 | RW | 0000 0000h |

¹ If code security is enabled, then INFO-3 is reserved for code security functions. It may be erased by the memory controller during operation. See section below on Code Protection for more details.

| | | | | |
|---------------------|------------|-------------------------|----|------------|
| DTSESEQCFG11 | 4000 00ACh | DTSE Sequence Config 11 | RW | 0000 0000h |
| DTSESEQCFG12 | 4000 00B0h | DTSE Sequence Config 12 | RW | 0000 0000h |
| DTSESEQCFG13 | 4000 00B4h | DTSE Sequence Config 13 | RW | 0000 0000h |
| DTSESEQCFG14 | 4000 00B8h | DTSE Sequence Config 14 | RW | 0000 0000h |
| DTSESEQCFG15 | 4000 00BCh | DTSE Sequence Config 15 | RW | 0000 0000h |
| DTSESEQCFG16 | 4000 00C0h | DTSE Sequence Config 16 | RW | 0000 0000h |
| DTSESEQCFG17 | 4000 00C4h | DTSE Sequence Config 17 | RW | 0000 0000h |
| DTSESEQCFG18 | 4000 00C8h | DTSE Sequence Config 18 | RW | 0000 0000h |
| DTSESEQCFG19 | 4000 00CCh | DTSE Sequence Config 19 | RW | 0000 0000h |
| DTSESEQCFG20 | 4000 00D0h | DTSE Sequence Config 20 | RW | 0000 0000h |
| DTSESEQCFG21 | 4000 00D4h | DTSE Sequence Config 21 | RW | 0000 0000h |
| DTSESEQCFG22 | 4000 00D8h | DTSE Sequence Config 22 | RW | 0000 0000h |
| DTSESEQCFG23 | 4000 00DCh | DTSE Sequence Config 23 | RW | 0000 0000h |
| DTSERES0 | 4000 0100h | DTSE Result 0 | RO | 0000 0000h |
| DTSERES1 | 4000 0104h | DTSE Result 1 | RO | 0000 0000h |
| DTSERES2 | 4000 0108h | DTSE Result 2 | RO | 0000 0000h |
| DTSERES3 | 4000 010Ch | DTSE Result 3 | RO | 0000 0000h |
| DTSERES4 | 4000 0110h | DTSE Result 4 | RO | 0000 0000h |
| DTSERES5 | 4000 0114h | DTSE Result 5 | RO | 0000 0000h |
| DTSERES6 | 4000 0118h | DTSE Result 6 | RO | 0000 0000h |
| DTSERES7 | 4000 011Ch | DTSE Result 7 | RO | 0000 0000h |
| DTSERES8 | 4000 0120h | DTSE Result 8 | RO | 0000 0000h |
| DTSERES9 | 4000 0124h | DTSE Result 9 | RO | 0000 0000h |
| DTSERES10 | 4000 0128h | DTSE Result 10 | RO | 0000 0000h |
| DTSERES11 | 4000 012Ch | DTSE Result 11 | RO | 0000 0000h |
| DTSERES12 | 4000 0130h | DTSE Result 12 | RO | 0000 0000h |
| DTSERES13 | 4000 0134h | DTSE Result 13 | RO | 0000 0000h |
| DTSERES14 | 4000 0138h | DTSE Result 14 | RO | 0000 0000h |
| DTSERES15 | 4000 013Ch | DTSE Result 15 | RO | 0000 0000h |
| DTSERES16 | 4000 0140h | DTSE Result 16 | RO | 0000 0000h |
| DTSERES17 | 4000 0144h | DTSE Result 17 | RO | 0000 0000h |
| DTSERES18 | 4000 0148h | DTSE Result 18 | RO | 0000 0000h |
| DTSERES19 | 4000 014Ch | DTSE Result 19 | RO | 0000 0000h |
| DTSERES20 | 4000 0150h | DTSE Result 20 | RO | 0000 0000h |
| DTSERES21 | 4000 0154h | DTSE Result 21 | RO | 0000 0000h |
| DTSERES22 | 4000 0158h | DTSE Result 22 | RO | 0000 0000h |

| | | | | |
|------------------|------------|----------------|----|------------|
| DTSERES23 | 4000 015Ch | DTSE Result 23 | RO | 0000 0000h |
|------------------|------------|----------------|----|------------|

Table 3-7 I2C Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|--|--------|------------|
| I2CCONSET | 4001 0000h | I2C Control Set Register | RW | 0000 0000h |
| I2CCONCLR | 4001 0004h | I2C Control Clear Register | WO | -- |
| I2CSTAT | 4001 0008h | I2C Status Register | RO | 0000 00F8h |
| I2CDAT | 4001 000Ch | I2C Data Register | RW | 0000 0000h |
| I2CCLK | 4001 0010h | I2C Clock Control Register | RW | 0000 0000h |
| I2CADR0 | 4001 0014h | I2C Slave Address Register 0 | RW | 0000 0000h |
| I2CADRM0 | 4001 0018h | I2C Slave Address Mask Register 0 | RW | 0000 00FEh |
| I2CXADR0 | 4001 001C | I2C Extended Slave Address Register 0 | RW | 0000 0000h |
| I2CXADM0 | 4001 0020h | I2C Extended Slave Address Mask Register 0 | RW | 0000 07FEh |
| I2CRST | 4001 0024h | I2C Software Reset Register | RW | 0000 0000h |
| I2CADR1 | 4001 0028h | I2C Slave Address Register 1 | RW | 0000 0000h |
| I2CADRM1 | 4001 002Ch | I2C Slave Address Mask Register 1 | RW | 0000 00FEh |
| I2CADR2 | 4001 0030h | I2C Slave Address Register 2 | RW | 0000 0000h |
| I2CADRM2 | 4001 0034h | I2C Slave Address Mask Register 2 | RW | 0000 00FEh |
| I2CADR3 | 4001 0038h | I2C Slave Address Register 3 | RW | 0000 0000h |
| I2CADRM3 | 4001 003Ch | I2C Slave Address Mask Register 3 | RW | 0000 00FEh |

Table 3-8 USARTA Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------------|------------|---|--------|------------|
| UARTARBR | 4002 0000h | UART A Receive Buffer Register | RO | -- |
| UARTATHR | 4002 0004h | UART A Transmit Holding Register | WO | -- |
| UARTADLR | 4002 0008h | UART A Divisor Latch Register | RW | 0000 0001h |
| UARTAIER | 4002 000Ch | UART A Interrupt Enable Register | RW | 0000 0000h |
| UARTAIIR | 4002 0010h | UART A Interrupt Identification Register | RO | 0000 0001h |
| UARTAFCR | 4002 0014h | UART A FIFO Control Register | RW | 0000 0000h |
| UARTALCR | 4002 0018h | UART A Line control Register | RW | 0000 0000h |
| UARTALSR | 4002 0020h | UART A Line Status Register | RO | 0000 0060h |
| UARTASCR | 4002 0028h | UART A Scratch Pad Register | RW | -- |
| UARTAEFR | 4002 002Ch | UART A Enhanced Features Register | RW | 0000 0000h |
| SSPACON | 4002 0000h | SSP A Control Register | RW | 0000 0000h |
| SSPASTAT | 4002 0004h | SSP A Status Register | RO | 0000 0003h |
| SSPADAT | 4002 0008h | SSP A Data Register | RW | 0000 0000h |
| SSPACLK | 4002 000Ch | SSP A Clock Control Register | RW | 0000 0000h |
| SSPAIMSC | 4002 0010h | SSP A Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPARIS | 4002 0014h | SSP A Raw Interrupt Status Register | RO | 0000 0000h |
| SSPAMIS | 4002 0018h | SSP A Masked Interrupt Status Register | RO | 0000 0000h |
| SSPAICLR | 4002 001Ch | SSP A Interrupt Clear Register | RW | 0000 0000h |
| SSPASSCR | 4002 0028h | SSP A Slave Select Configuration Register | RW | 0000 0000h |

Table 3-9 USARTB Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---|--------|------------|
| UARTBRBR | 4003 0000h | UART B Receive Buffer Register | RO | -- |
| UARTBTHR | 4003 0004h | UART B Transmit Holding Register | WO | -- |
| UARTBDLR | 4003 0008h | UART B Divisor Latch Register | RW | 0000 0001h |
| UARTBIER | 4003 000Ch | UART B Interrupt Enable Register | RW | 0000 0000h |
| UARTBIIR | 4003 0010h | UART B Interrupt Identification Register | RO | 0000 0001h |
| UARTBFCSR | 4003 0014h | UART B FIFO Control Register | RW | 0000 0000h |
| UARTBLCR | 4003 0018h | UART B Line control Register | RW | 0000 0000h |
| UARTBLSR | 4003 0020h | UART B Line Status Register | RO | 0000 0060h |
| UARTBSCR | 4003 0028h | UART B Scratch Pad Register | RW | -- |
| UARTBEFR | 4003 002Ch | UART B Enhanced Features Register | RW | 0000 0000h |
| SSPBCON | 4003 0000h | SSP B Control Register | RW | 0000 0000h |
| SSPBSTAT | 4003 0004h | SSP B Status Register | RO | 0000 0003h |
| SSPBDAT | 4003 0008h | SSP B Data Register | RW | 0000 0000h |
| SSPBCLK | 4003 000Ch | SSP B Clock Control Register | RW | 0000 0000h |
| SSPBIMSC | 4003 0010h | SSP B Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPBRIS | 4003 0014h | SSP B Raw Interrupt Status Register | RO | 0000 0000h |
| SSPBMIS | 4003 0018h | SSP B Masked Interrupt Status Register | RO | 0000 0000h |
| SSPBICLR | 4003 001Ch | SSP B Interrupt Clear Register | RW | 0000 0000h |
| SSPBSSCR | 4003 0028h | SSP B Slave Select Configuration Register | RW | 0000 0000h |

Table 3-10 USARTC Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|---|--------|------------|
| UARTCRBR | 4004 0000h | UART C Receive Buffer Register | RO | -- |
| UARTCTHR | 4004 0004h | UART C Transmit Holding Register | WO | -- |
| UARTCDLR | 4004 0008h | UART C Divisor Latch Register | RW | 0000 0001h |
| UARTCIER | 4004 000Ch | UART C Interrupt Enable Register | RW | 0000 0000h |
| UARTCIIR | 4004 0010h | UART C Interrupt Identification Register | RO | 0000 0001h |
| UARTCFCR | 4004 0014h | UART C FIFO Control Register | RW | 0000 0000h |
| UARTCLCR | 4004 0018h | UART C Line control Register | RW | 0000 0000h |
| UARTCLSR | 4004 0020h | UART C Line Status Register | RO | 0000 0060h |
| UARTCSCR | 4004 0028h | UART C Scratch Pad Register | RW | -- |
| UARTCEFR | 4004 002Ch | UART C Enhanced Features Register | RW | 0000 0000h |
| SSPCCON | 4004 0000h | SSP C Control Register | RW | 0000 0000h |
| SSPCSTAT | 4004 0004h | SSP C Status Register | RO | 0000 0003h |
| SSPCDAT | 4004 0008h | SSP C Data Register | RW | 0000 0000h |
| SSPCCLK | 4004 000Ch | SSP C Clock Control Register | RW | 0000 0000h |
| SSPCIMSC | 4004 0010h | SSP C Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPCRIS | 4004 0014h | SSP C Raw Interrupt Status Register | RO | 0000 0000h |
| SSPCMIS | 4004 0018h | SSP C Masked Interrupt Status Register | RO | 0000 0000h |
| SSPCICLR | 4004 001Ch | SSP C Interrupt Clear Register | RW | 0000 0000h |
| SSPCSSCR | 4004 0028h | SSP C Slave Select Configuration Register | RW | 0000 0000h |

Table 3-11 USARTD Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|---|--------|------------|
| UARTDRBR | 4005 0000h | UART D Receive Buffer Register | RO | -- |
| UARTDTHR | 4005 0004h | UART D Transmit Holding Register | WO | -- |
| UARTDDLRLR | 4005 0008h | UART D Divisor Latch Register | RW | 0000 0001h |
| UARTDIER | 4005 000Ch | UART D Interrupt Enable Register | RW | 0000 0000h |
| UARTDIIR | 4005 0010h | UART D Interrupt Identification Register | RO | 0000 0001h |
| UARTDFCR | 4005 0014h | UART D FIFO Control Register | RW | 0000 0000h |
| UARTDLCR | 4005 0018h | UART D Line control Register | RW | 0000 0000h |
| UARTDLRSR | 4005 0020h | UART D Line Status Register | RO | 0000 0060h |
| UARTDSCR | 4005 0028h | UART D Scratch Pad Register | RW | -- |
| UARTDEFER | 4005 002Ch | UART D Enhanced Features Register | RW | 0000 0000h |
| SSPDCON | 4005 0000h | SSP D Control Register | RW | 0000 0000h |
| SSPDSTAT | 4005 0004h | SSP D Status Register | RO | 0000 0003h |
| SSPDSTAT | 4005 0008h | SSP D Data Register | RW | 0000 0000h |
| SSPDCLK | 4005 000Ch | SSP D Clock Control Register | RW | 0000 0000h |
| SSPDIMSC | 4005 0010h | SSP D Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPDRIS | 4005 0014h | SSP D Raw Interrupt Status Register | RO | 0000 0000h |
| SSPDMIS | 4005 0018h | SSP D Masked Interrupt Status Register | RO | 0000 0000h |
| SSPDICLR | 4005 001Ch | SSP D Interrupt Clear Register | RW | 0000 0000h |
| SSPDSSCR | 4005 0028h | SSP D Slave Select Configuration Register | RW | 0000 0000h |

Table 3-12 Timer A Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---------------------------|--------|------------|
| TACTL | 4006 0000h | Timer A Control | RW | 0000 0000h |
| TAINT | 4006 0004h | Timer A Interrupt Control | RW | 0000 0000h |
| TAPRD | 4006 0008h | Timer A Period | RW | 0000 0000h |
| TACTR | 4006 000Ch | Timer A Counter | RW | 0000 0000h |
| TAQEPCTL | 4006 0010h | Timer A QEP Control | RW | 0000 0000h |
| TACCTL0 | 4006 0100h | Timer A CC Control 0 | RW | 0000 0000h |
| TACCTR0 | 4006 0104h | Timer A CC Counter 0 | RW | 0000 0000h |
| TACCTL1 | 4006 0108h | Timer A CC Control 1 | RW | 0000 0000h |
| TACCTR1 | 4006 010Ch | Timer A CC Counter 1 | RW | 0000 0000h |
| TACCTL2 | 4006 0110h | Timer A CC Control 2 | RW | 0000 0000h |
| TACCTR2 | 4006 0114h | Timer A CC Counter 2 | RW | 0000 0000h |
| TACCTL3 | 4006 0118h | Timer A CC Control 3 | RW | 0000 0000h |
| TACCTR3 | 4006 011Ch | Timer A CC Counter 3 | RW | 0000 0000h |
| TACCTL4 | 4006 0120h | Timer A CC Control 4 | RW | 0000 0000h |
| TACCTR4 | 4006 0124h | Timer A CC Counter 4 | RW | 0000 0000h |
| TACCTL5 | 4006 0128h | Timer A CC Control 5 | RW | 0000 0000h |
| TACCTR5 | 4006 012Ch | Timer A CC Counter 5 | RW | 0000 0000h |
| TACCTL6 | 4006 0130h | Timer A CC Control 6 | RW | 0000 0000h |
| TACCTR6 | 4006 0134h | Timer A CC Counter 6 | RW | 0000 0000h |
| TACCTL7 | 4006 0138h | Timer A CC Control 7 | RW | 0000 0000h |
| TACCTR7 | 4006 013Ch | Timer A CC Counter 7 | RW | 0000 0000h |
| TADTGCTL0 | 4006 0200h | Timer A DTG Control 0 | RW | 0000 0000h |
| TADTGCTL1 | 4006 0204h | Timer A DTG Control 1 | RW | 0000 0000h |
| TADTGCTL2 | 4006 0208h | Timer A DTG Control 2 | RW | 0000 0000h |
| TADTGCTL3 | 4006 020Ch | Timer A DTG Control 3 | RW | 0000 0000h |

Table 3-13 Timer B Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|---------------------------|--------|------------|
| TBCTL | 4007 0000h | Timer B Control | RW | 0000 0000h |
| TBINT | 4007 0004h | Timer B Interrupt Control | RW | 0000 0000h |
| TBPRD | 4007 0008h | Timer B Period | RW | 0000 0000h |
| TBCTR | 4007 000Ch | Timer B Counter | RW | 0000 0000h |
| TBQEPCTL | 4007 0010h | Timer B QEP Control | RW | 0000 0000h |
| TBCCTL0 | 4007 0100h | Timer B CC Control 0 | RW | 0000 0000h |
| TBCCTR0 | 4007 0104h | Timer B CC Counter 0 | RW | 0000 0000h |
| TBCCTL1 | 4007 0108h | Timer B CC Control 1 | RW | 0000 0000h |
| TBCCTR1 | 4007 010Ch | Timer B CC Counter 1 | RW | 0000 0000h |
| TBCCTL2 | 4007 0110h | Timer B CC Control 2 | RW | 0000 0000h |
| TBCCTR2 | 4007 0114h | Timer B CC Counter 2 | RW | 0000 0000h |
| TBCCTL3 | 4007 0118h | Timer B CC Control 3 | RW | 0000 0000h |
| TBCCTR3 | 4007 011Ch | Timer B CC Counter 3 | RW | 0000 0000h |
| TBCCTL4 | 4007 0120h | Timer B CC Control 4 | RW | 0000 0000h |
| TBCCTR4 | 4007 0124h | Timer B CC Counter 4 | RW | 0000 0000h |
| TBCCTL5 | 4007 0128h | Timer B CC Control 5 | RW | 0000 0000h |
| TBCCTR5 | 4007 012Ch | Timer B CC Counter 5 | RW | 0000 0000h |
| TBCCTL6 | 4007 0130h | Timer B CC Control 6 | RW | 0000 0000h |
| TBCCTR6 | 4007 0134h | Timer B CC Counter 6 | RW | 0000 0000h |
| TBCCTL7 | 4007 0138h | Timer B CC Control 7 | RW | 0000 0000h |
| TBCCTR7 | 4007 013Ch | Timer B CC Counter 7 | RW | 0000 0000h |
| TBDTGCTL0 | 4007 0200h | Timer B DTG Control 0 | RW | 0000 0000h |
| TBDTGCTL1 | 4007 0204h | Timer B DTG Control 1 | RW | 0000 0000h |
| TBDTGCTL2 | 4007 0208h | Timer B DTG Control 2 | RW | 0000 0000h |
| TBDTGCTL3 | 4007 020Ch | Timer B DTG Control 3 | RW | 0000 0000h |

Table 3-14 Timer C Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---------------------------|--------|------------|
| TCCTL | 4008 0000h | Timer C Control | RW | 0000 0000h |
| TCINT | 4008 0004h | Timer C Interrupt Control | RW | 0000 0000h |
| TCPRD | 4008 0008h | Timer C Period | RW | 0000 0000h |
| TCCTR | 4008 000Ch | Timer C Counter | RW | 0000 0000h |
| TCQEPCTL | 4008 0010h | Timer C QEP Control | RW | 0000 0000h |
| TCCCTL0 | 4008 0100h | Timer C CC Control 0 | RW | 0000 0000h |
| TCCCTR0 | 4008 0104h | Timer C CC Counter 0 | RW | 0000 0000h |
| TCCCTL1 | 4008 0108h | Timer C CC Control 1 | RW | 0000 0000h |
| TCCCTR1 | 4008 010Ch | Timer C CC Counter 1 | RW | 0000 0000h |
| TCCCTL2 | 4008 0110h | Timer C CC Control 2 | RW | 0000 0000h |
| TCCCTR2 | 4008 0114h | Timer C CC Counter 2 | RW | 0000 0000h |
| TCCCTL3 | 4008 0118h | Timer C CC Control 3 | RW | 0000 0000h |
| TCCCTR3 | 4008 011Ch | Timer C CC Counter 3 | RW | 0000 0000h |
| TCCCTL4 | 4008 0120h | Timer C CC Control 4 | RW | 0000 0000h |
| TCCCTR4 | 4008 0124h | Timer C CC Counter 4 | RW | 0000 0000h |
| TCCCTL5 | 4008 0128h | Timer C CC Control 5 | RW | 0000 0000h |
| TCCCTR5 | 4008 012Ch | Timer C CC Counter 5 | RW | 0000 0000h |
| TCCCTL6 | 4008 0130h | Timer C CC Control 6 | RW | 0000 0000h |
| TCCCTR6 | 4008 0134h | Timer C CC Counter 6 | RW | 0000 0000h |
| TCCCTL7 | 4008 0138h | Timer C CC Control 7 | RW | 0000 0000h |
| TCCCTR7 | 4008 013Ch | Timer C CC Counter 7 | RW | 0000 0000h |
| TCDTGCTL0 | 4008 0200h | Timer C DTG Control 0 | RW | 0000 0000h |
| TCDTGCTL1 | 4008 0204h | Timer C DTG Control 1 | RW | 0000 0000h |
| TCDTGCTL2 | 4008 0208h | Timer C DTG Control 2 | RW | 0000 0000h |
| TCDTGCTL3 | 4008 020Ch | Timer C DTG Control 3 | RW | 0000 0000h |

Table 3-15 Timer D Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---------------------------|--------|------------|
| TDCTL | 4009 0000h | Timer D Control | RW | 0000 0000h |
| TDINT | 4009 0004h | Timer D Interrupt Control | RW | 0000 0000h |
| TDPRD | 4009 0008h | Timer D Period | RW | 0000 0000h |
| TDCTR | 4009 000Ch | Timer D Counter | RW | 0000 0000h |
| TDQEPCTL | 4009 0010h | Timer D QEP Control | RW | 0000 0000h |
| TDCCTL0 | 4009 0100h | Timer D CC Control 0 | RW | 0000 0000h |
| TDCCTR0 | 4009 0104h | Timer D CC Counter 0 | RW | 0000 0000h |
| TDCCTL1 | 4009 0108h | Timer D CC Control 1 | RW | 0000 0000h |
| TDCCTR1 | 4009 010Ch | Timer D CC Counter 1 | RW | 0000 0000h |
| TDCCTL2 | 4009 0110h | Timer D CC Control 2 | RW | 0000 0000h |
| TDCCTR2 | 4009 0114h | Timer D CC Counter 2 | RW | 0000 0000h |
| TDCCTL3 | 4009 0118h | Timer D CC Control 3 | RW | 0000 0000h |
| TDCCTR3 | 4009 011Ch | Timer D CC Counter 3 | RW | 0000 0000h |
| TDCCTL4 | 4009 0120h | Timer D CC Control 4 | RW | 0000 0000h |
| TDCCTR4 | 4009 0124h | Timer D CC Counter 4 | RW | 0000 0000h |
| TDCCTL5 | 4009 0128h | Timer D CC Control 5 | RW | 0000 0000h |
| TDCCTR5 | 4009 012Ch | Timer D CC Counter 5 | RW | 0000 0000h |
| TDCCTL6 | 4009 0130h | Timer D CC Control 6 | RW | 0000 0000h |
| TDCCTR6 | 4009 0134h | Timer D CC Counter 6 | RW | 0000 0000h |
| TDCCTL7 | 4009 0138h | Timer D CC Control 7 | RW | 0000 0000h |
| TDCCTR7 | 4009 013Ch | Timer D CC Counter 7 | RW | 0000 0000h |
| TDDTGCTL0 | 4009 0200h | Timer D DTG Control 0 | RW | 0000 0000h |
| TDDTGCTL1 | 4009 0204h | Timer D DTG Control 1 | RW | 0000 0000h |
| TDDTGCTL2 | 4009 0208h | Timer D DTG Control 2 | RW | 0000 0000h |
| TDDTGCTL3 | 4009 020Ch | Timer D DTG Control 3 | RW | 0000 0000h |

Table 3-16 CAN Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|--------------------------------|------------|---|--------|-----------|
| CAN_ISR_SR_CMRR_MR | 400A 0000h | CAN Mode/Command/Status/ISR Register | RW | 00000004h |
| CAN_BTR1_BTR0_RMC_IMR | 400A 0004h | CAN Bus Timing/RX Message Counter/Interrupt Mask Register | RW | 00000000h |
| CANTXBUF | 400A 0008h | CAN Transmit Buffer Register | RW | 00000000h |
| CANRXBUF | 400A 000Ch | CAN Receive Buffer Register | RO | 00000000h |
| CANACR | 400A 0010h | CAN Acceptance Code Register | RW | 00000000h |
| CANAMR | 400A 0014h | CAN Acceptance Mask Register | RW | 00000000h |
| CAN_ALC_TXERR_RXERR_ECC | 400A 0018h | CAN ALC/TXERR/RXERR/ECC Register | RO | 00000000h |

Table 3-17 GP Timer A Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------------|------------|-------------------|--------|------------|
| GPTACTL | 400B 0000h | GPTimer A Control | RW | 00FF FF00h |
| GPTACTR | 400B 0004h | GPTimer A Counter | RO | 00FF FFFFh |

Table 3-18 GP Timer B Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------------|------------|-------------------|--------|------------|
| GPTBCTL | 400C 0000h | GPTimer B Control | RW | 00FF FF00h |
| GPTBCTR | 400C 0004h | GPTimer B Counter | RO | 00FF FFFFh |

Table 3-19 Memory Controller Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|---------------------------------|--------|------------|
| MEMCTL | 400D 0000h | Memory Controller Configuration | RW | 0032 0000h |
| MEMSTATUS | 400D 0004h | Memory Controller Status | RW | 0000 0000h |
| FLASHLOCK | 400D 0008h | FLASH Lock Access | RW | 0000 0000h |
| FLASHPAGE | 400D 000Ch | FLASH Page | RW | 0000 0000h |
| SWDUNLOCK | 400D 0010h | SWD Unlock | RW | 0000 0000h |
| FLASHERASE | 400D 0020h | FLASH Erase | RW | 0000 0000h |

Table 3-20 System and Clock Control (SCC) Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|-----------------------------------|--------|------------|
| CCSCTL | 400D 0400h | CCS Control | RW | 0000 F004h |
| CCSPLLCTL | 400D 0404h | PLL Control | RW | 0000 0000h |
| CCSROSTRIM | 400D 0408h | ROSC Trim Control | RW | 0000 007Fh |
| PAMUXSEL | 400D 040Ch | PA Peripheral MUX Select | RW | 0011 1110h |
| PBMUXSEL | 400D 0410h | PB Peripheral MUX Select | RW | 0000 0000h |
| PCMUXSEL | 400D 0414h | PC Peripheral MUX Select | RW | 0000 0000h |
| PDMUXSEL | 400D 0418h | PD Peripheral MUX Select | RW | 0000 0000h |
| PEMUXSEL | 400D 041Ch | PE Peripheral MUX Select | RW | 0000 0000h |
| PFMUXSEL | 400D 0420h | PF Peripheral MUX Select | RW | 0000 3333h |
| PGMUXSEL | 400D 0424h | PG Peripheral MUX Select | RW | 0000 0000h |
| PAPUEN | 400D 0428h | PA Weak Pull-up Enable | RW | 0000 0000h |
| PBPUEN | 400D 042Ch | PB Weak Pull-up Enable | RW | 0000 0000h |
| PCPUEN | 400D 0430h | PC Weak Pull-up Enable | RW | 0000 0000h |
| PDPUEN | 400D 0434h | PD Weak Pull-up Enable | RW | 0000 0000h |
| PEPUEN | 400D 0438h | PE Weak Pull-up Enable | RW | 0000 0000h |
| PFFUEN | 400D 043Ch | PF Weak Pull-up Enable | RW | 0000 0000h |
| PGPUEN | 400D 0440h | PG Weak Pull-up Enable | RW | 0000 0000h |
| PAPDEN | 400D 0444h | PA Weak Pull-down Enable | RW | 0000 0000h |
| PBPDEN | 400D 0448h | PB Weak Pull-down Enable | RW | 0000 0000h |
| PCPDEN | 400D 044Ch | PC Weak Pull-down Enable | RW | 0000 0000h |
| PDPDEN | 400D 0450h | PD Weak Pull-down Enable | RW | 0000 0000h |
| PEPDEN | 400D 0454h | PE Weak Pull-down Enable | RW | 0000 0000h |
| PFPDEN | 400D 0458h | PF Weak Pull-down Enable | RW | 0000 0000h |
| PGPDEN | 400D 045Ch | PG Weak Pull-down Enable | RW | 0000 0000h |
| PADS | 400D 0460h | PA Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PBDS | 400D 0464h | PB Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PCDS | 400D 0468h | PC Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PDDS | 400D 046Ch | PD Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PEDS | 400D 0470h | PE Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PFDS | 400D 0474h | PF Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PGDS | 400D 0478h | PG Drive Strength/Schmitt Trigger | RW | 0000 0000h |

Table 3-21 Windowed Watchdog Timer (WWDT) Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|-------------------------|--------|------------|
| WWDTCTL | 400D 0800h | WWDT Control | RW | 0000 0000h |
| WWDTLOAD | 400D 0804h | WWDT Load Counter Value | RW | 0010 FFFFh |
| WWDTCTR | 400D 0808h | WWDT Counter | RO | 0000 FFFFh |
| WWDTINTF | 400D 080Ch | WWDT Interrupt Flag | RO | 0000 0000h |
| WWDTCLEAR | 400D 0810h | WWDT Clear | WO | -- |
| WWDTLOCK | 400D 0814h | WWDT Lock | RW | 0000 0000h |

Table 3-22 Real-Time Clock (RTC) Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|--------------------|------------|-------------------|--------|------------|
| RTCCTL | 400D 0C00h | RTC Control | RW | 0000 0008h |
| RTCTIME | 400D 0C04h | RTC Time | RO | 0000 0000h |
| RTCDATE | 400D 0C08h | RTC Date | RO | 0001 0100h |
| RTCTIMESET | 400D 0C0Ch | RTC Time Setting | RW | 0000 0000h |
| RTCDATESET | 400D 0C10h | RTC Date Setting | RW | 0001 0100h |
| RTCALARMSET | 400D 0C14h | RTC Alarm Setting | RW | 0000 0000h |

Table 3-23 CRC Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|-----------------|--------|------------|
| CRCCTL | 400D 1000h | CRC Control | RW | 0000 0000h |
| CRCDATAIN | 400D 1004h | CRC Data Input | RW | 0000 0000h |
| CRCSEED | 400D 1008h | CRC Seed Value | RW | 0000 0000h |
| CRCDATAOUT | 400D 100Ch | CRC Data Output | RO | 0000 0000h |

Table 3-24 GPIOA Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|------------------------------|--------|------------|
| GPIOAMODE | 400D 1400h | GPIOA Pin Mode Select | RW | 0000 FFFFh |
| GPIOAOUTMASK | 400D 1404h | GPIOA Data Output Write Mask | RW | 0000 0000h |
| GPIOAOUT | 400D 1408h | GPIOA Data Output Value | RW | FFFF FFFFh |
| GPIOAIN | 400D 140Ch | GPIOA Data Input Value | RO | -- |
| GPIOAINTEN | 400D 1410h | GPIOA Interrupt Enable | RW | 0000 0000h |
| GPIOAINTFLAGRAW | 400D 1414h | GPIOA Interrupt Flag Raw | RW | 0000 0000h |
| GPIOAINTFLAGMASKED | 400D 1418h | GPIOA Interrupt Flag Masked | RW | 0000 0000h |
| GPIOAINTCLEAR | 400D 141Ch | GPIOA Interrupt Clear | WO | 0000 0000h |
| GPIOAINTTYPE | 400D 1420h | GPIOA Interrupt Type | RW | 0000 0000h |
| GPIOAINTVALUE | 400D 1424h | GPIOA Interrupt Value | RW | 0000 0000h |
| GPIOAINTEDGEBOTH | 400D 1428h | GPIOA Interrupt Edge Both | RW | 0000 0000h |
| GPIOADEBOUNCE | 400D 142Ch | GPIOA De-bounce Filter | RW | 0000 0000h |
| GPIOADOSET | 400D 1430h | GPIOA Data Output Set | WO | 0000 0000h |
| GPIOADOCLEAR | 400D 1434h | GPIOA Data Output Clear | WO | 0000 0000h |

Table 3-25 GPIOB Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|------------------------------|--------|------------|
| GPIOBMODE | 400D 1800h | GPIOB Pin Mode Select | RW | 0000 FFFFh |
| GPIOBOUTMASK | 400D 1804h | GPIOB Data Output Write Mask | RW | 0000 0000h |
| GPIOBOUT | 400D 1808h | GPIOB Data Output Value | RW | FFFF FFFFh |
| GPIOBIN | 400D 180Ch | GPIOB Data Input Value | RO | -- |
| GPIOBINTEN | 400D 1810h | GPIOB Interrupt Enable | RW | 0000 0000h |
| GPIOBINTFLAGRAW | 400D 1814h | GPIOB Interrupt Flag Raw | RW | 0000 0000h |
| GPIOBINTFLAGMASKED | 400D 1818h | GPIOB Interrupt Flag Masked | RW | 0000 0000h |
| GPIOBINTCLEAR | 400D 181Ch | GPIOB Interrupt Clear | WO | 0000 0000h |
| GPIOBINTTYPE | 400D 1820h | GPIOB Interrupt Type | RW | 0000 0000h |
| GPIOBINTVALUE | 400D 1824h | GPIOB Interrupt Value | RW | 0000 0000h |
| GPIOBINTEDGEBOTH | 400D 1828h | GPIOB Interrupt Edge Both | RW | 0000 0000h |
| GPIOBDEBOUNCE | 400D 182Ch | GPIOB De-bounce Filter | RW | 0000 0000h |
| GPIOBDOSSET | 400D 1830h | GPIOB Data Output Set | WO | 0000 0000h |
| GPIOBDOCLEAR | 400D 1834h | GPIOB Data Output Clear | WO | 0000 0000h |

Table 3-26 GPIOC Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|------------------------------|--------|------------|
| GPIOCMODE | 400D 1400h | GPIOC Pin Mode Select | RW | 0000 FFFFh |
| GPIOCOUTMASK | 400D 1404h | GPIOC Data Output Write Mask | RW | 0000 0000h |
| GPIOCOUT | 400D 1408h | GPIOC Data Output Value | RW | FFFF FFFFh |
| GPIOCIN | 400D 140Ch | GPIOC Data Input Value | RO | -- |
| GPIOCINTEN | 400D 1410h | GPIOC Interrupt Enable | RW | 0000 0000h |
| GPIOCINTFLAGRAW | 400D 1414h | GPIOC Interrupt Flag Raw | RW | 0000 0000h |
| GPIOCINTFLAGMASKED | 400D 1418h | GPIOC Interrupt Flag Masked | RW | 0000 0000h |
| GPIOCINTCLEAR | 400D 141Ch | GPIOC Interrupt Clear | WO | 0000 0000h |
| GPIOCINTTYPE | 400D 1420h | GPIOC Interrupt Type | RW | 0000 0000h |
| GPIOCINTVALUE | 400D 1424h | GPIOC Interrupt Value | RW | 0000 0000h |
| GPIOCINTEDGEBOTH | 400D 1428h | GPIOC Interrupt Edge Both | RW | 0000 0000h |
| GPIOCDEBOUNCE | 400D 142Ch | GPIOC De-bounce Filter | RW | 0000 0000h |
| GPIOCDOSET | 400D 1430h | GPIOC Data Output Set | WO | 0000 0000h |
| GPIOCDOCLEAR | 400D 1434h | GPIOC Data Output Clear | WO | 0000 0000h |

Table 3-27 GPIOD Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|------------------------------|--------|------------|
| GPIODMODE | 400D 2000h | GPIOD Pin Mode Select | RW | 0000 FFFFh |
| GPIODOUTMASK | 400D 2004h | GPIOD Data Output Write Mask | RW | 0000 0000h |
| GPIODOUT | 400D 2008h | GPIOD Data Output Value | RW | FFFF FFFFh |
| GPIODIN | 400D 200Ch | GPIOD Data Input Value | RO | -- |
| GPIODINTEN | 400D 2010h | GPIOD Interrupt Enable | RW | 0000 0000h |
| GPIODINTFLAGRAW | 400D 2014h | GPIOD Interrupt Flag Raw | RW | 0000 0000h |
| GPIODINTFLAGMASKED | 400D 2018h | GPIOD Interrupt Flag Masked | RW | 0000 0000h |
| GPIODINTCLEAR | 400D 201Ch | GPIOD Interrupt Clear | WO | 0000 0000h |
| GPIODINTTYPE | 400D 2020h | GPIOD Interrupt Type | RW | 0000 0000h |
| GPIODINTVALUE | 400D 2024h | GPIOD Interrupt Value | RW | 0000 0000h |
| GPIODINTEDGEBOTH | 400D 2028h | GPIOD Interrupt Edge Both | RW | 0000 0000h |
| GPIODDEBOUNCE | 400D 202Ch | GPIOD De-bounce Filter | RW | 0000 0000h |
| GPIODDOSET | 400D 2030h | GPIOD Data Output Set | WO | 0000 0000h |
| GPIODDOCLEAR | 400D 2034h | GPIOD Data Output Clear | WO | 0000 0000h |

Table 3-28 GPIO E Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOEMODE | 400D 2400h | GPIO E Pin Mode Select | RW | 0000 FFFFh |
| GPIOEOUTMASK | 400D 2404h | GPIO E Data Output Write Mask | RW | 0000 0000h |
| GPIOEOUT | 400D 2408h | GPIO E Data Output Value | RW | FFFF FFFFh |
| GPIOEIN | 400D 240Ch | GPIO E Data Input Value | RO | -- |
| GPIOEINTEN | 400D 2410h | GPIO E Interrupt Enable | RW | 0000 0000h |
| GPIOEINTFLAGRAW | 400D 2414h | GPIO E Interrupt Flag Raw | RW | 0000 0000h |
| GPIOEINTFLAGMASKED | 400D 2418h | GPIO E Interrupt Flag Masked | RW | 0000 0000h |
| GPIOEINTCLEAR | 400D 241Ch | GPIO E Interrupt Clear | WO | 0000 0000h |
| GPIOEINTTYPE | 400D 2420h | GPIO E Interrupt Type | RW | 0000 0000h |
| GPIOEINTVALUE | 400D 2424h | GPIO E Interrupt Value | RW | 0000 0000h |
| GPIOEINTEDGEBOOTH | 400D 2428h | GPIO E Interrupt Edge Both | RW | 0000 0000h |
| GPIOEDEBOUNCE | 400D 242Ch | GPIO E De-bounce Filter | RW | 0000 0000h |
| GPIOEDOSET | 400D 2430h | GPIO E Data Output Set | WO | 0000 0000h |
| GPIOEDOCLEAR | 400D 2434h | GPIO E Data Output Clear | WO | 0000 0000h |

Table 3-29 GPIO F Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOFMODE | 400D 2800h | GPIO F Pin Mode Select | RW | 0000 FFFFh |
| GPIOFOUTMASK | 400D 2804h | GPIO F Data Output Write Mask | RW | 0000 0000h |
| GPIOFOUT | 400D 2808h | GPIO F Data Output Value | RW | FFFF FFFFh |
| GPIOFIN | 400D 280Ch | GPIO F Data Input Value | RO | -- |
| GPIOFINTEN | 400D 2810h | GPIO F Interrupt Enable | RW | 0000 0000h |
| GPIOFINTFLAGRAW | 400D 2814h | GPIO F Interrupt Flag Raw | RW | 0000 0000h |
| GPIOFINTFLAGMASKED | 400D 2818h | GPIO F Interrupt Flag Masked | RW | 0000 0000h |
| GPIOFINTCLEAR | 400D 281Ch | GPIO F Interrupt Clear | WO | 0000 0000h |
| GPIOFINTTYPE | 400D 2820h | GPIO F Interrupt Type | RW | 0000 0000h |
| GPIOFINTVALUE | 400D 2824h | GPIO F Interrupt Value | RW | 0000 0000h |
| GPIOFINTEDGEBOOTH | 400D 2828h | GPIO F Interrupt Edge Both | RW | 0000 0000h |
| GPIOFDEBOUNCE | 400D 282Ch | GPIO F De-bounce Filter | RW | 0000 0000h |
| GPIOFDOSET | 400D 2830h | GPIO F Data Output Set | WO | 0000 0000h |
| GPIOFDOCLEAR | 400D 2834h | GPIO F Data Output Clear | WO | 0000 0000h |

Table 3-30 GPIO G Register Map

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOGMODE | 400D 2C00h | GPIO G Pin Mode Select | RW | 0000 FFFFh |
| GPIOGOUTMASK | 400D 2C04h | GPIO G Data Output Write Mask | RW | 0000 0000h |
| GPIOGOUT | 400D 2C08h | GPIO G Data Output Value | RW | FFFF FFFFh |
| GPIOGIN | 400D 2C0Ch | GPIO G Data Input Value | RO | -- |
| GPIOGINTEN | 400D 2C10h | GPIO G Interrupt Enable | RW | 0000 0000h |
| GPIOGINTFLAGRAW | 400D 2C14h | GPIO G Interrupt Flag Raw | RW | 0000 0000h |
| GPIOGINTFLAGMASKED | 400D 2C18h | GPIO G Interrupt Flag Masked | RW | 0000 0000h |
| GPIOGINTCLEAR | 400D 2C1Ch | GPIO G Interrupt Clear | WO | 0000 0000h |
| GPIOGINTTYPE | 400D 2C20h | GPIO G Interrupt Type | RW | 0000 0000h |
| GPIOGINTVALUE | 400D 2C24h | GPIO G Interrupt Value | RW | 0000 0000h |
| GPIOGINTEDGEBOTH | 400D 2C28h | GPIO G Interrupt Edge Both | RW | 0000 0000h |
| GPIOGDEBOUNCE | 400D 2C2Ch | GPIO G De-bounce Filter | RW | 0000 0000h |
| GPIOGDOSET | 400D 2C30h | GPIO G Data Output Set | WO | 0000 0000h |
| GPIOGDOCLEAR | 400D 2C34h | GPIO G Data Output Clear | WO | 0000 0000h |

4 SCC

4.1 Overview

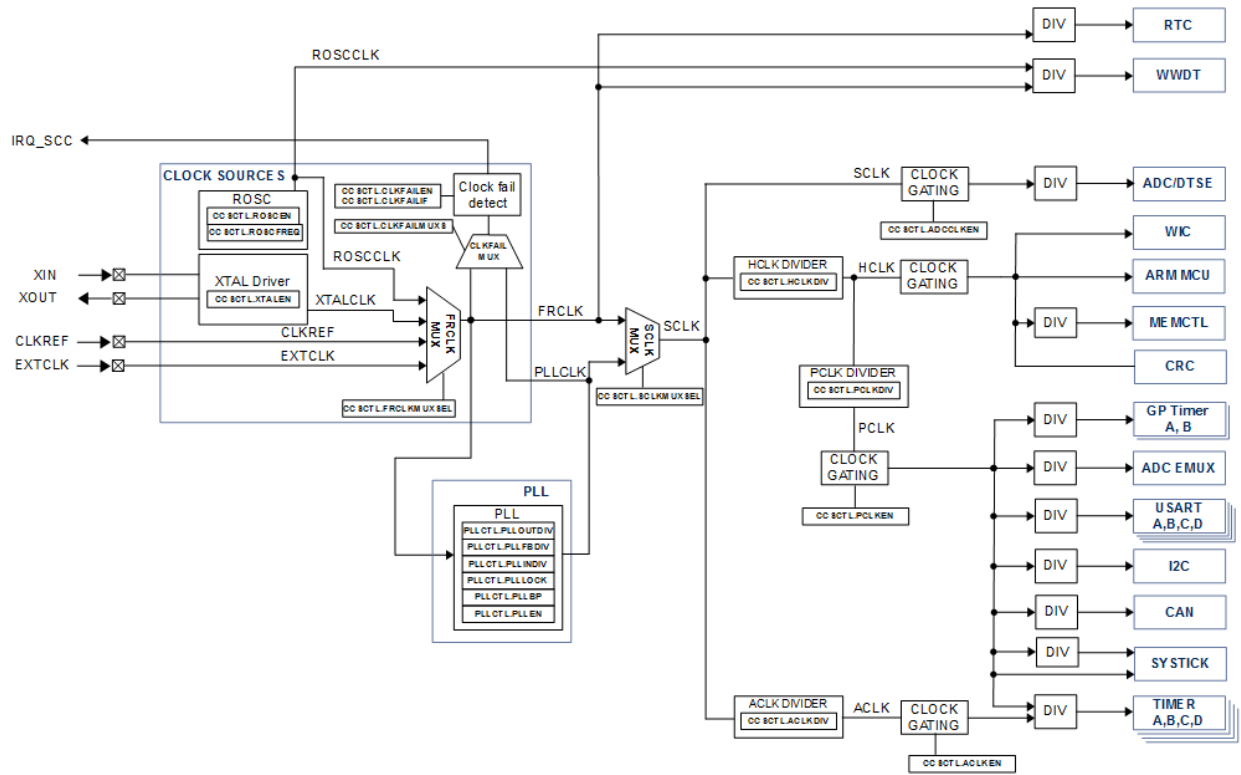
The System and Clock Control (SCC) module controls the clock system for the PAC55XX family of devices. The SCC contains the Clock Control System (CCS) as well as other system control functions.

4.2 Features

- Clock Control System (CCS):
 - 4 clock sources:
 - 4MHz internally generated 1% RC oscillator
 - 16MHz Ring Oscillator
 - Crystal Driver for external 2-10MHz crystals
 - External clock input for up to 20MHz external clock sources
 - 300MHz PLL
 - 5 system clocks for peripherals with programmable clock dividers
 - Clock gating for low-power mode support
 - Clock Failure Detection with configurable interrupt
- 32 User Interrupts
 - 8 levels of priority

4.3 System Block Diagram

Figure 4-1 CCS Block Diagram



4.4 Clock Sources

The PAC55XX can support up to 4 clock sources:

- Ring Oscillator Clock (**ROSCCLK**)
- Crystal Driver Clock (**XTALCLK**)
- Clock Reference (**CLKREF**)
- External Clock (**EXTCLK**)

The PAC55XX may select any of these clock sources using the FRCLK MUX. The output of the FRCLK MUX becomes the **FRCLK** system clock.

Table 4-1 CCS Clock Source Selections

| CCSCTL.FRCLKMUXSEL | FRCLK |
|--------------------|---------|
| 00b (default) | ROSCCLK |
| 01b | CLKREF |
| 10b | XTALCLK |
| 11b | EXTCLK |

4.4.1 ROSCCLK

The **ROSCCLK** is the output of the 16MHz Ring Oscillator. **ROSCCLK** is the default system clock when the PAC55XX comes out of reset. **ROSCCLK** is the default input for **FRCLK** and can optionally be used as a 2nd time-base for the WWDT peripheral.

The ROSC may be enabled or disabled using the **CCSCTL.ROSCEN** bit.

4.4.2 CLKREF

The **CLKREF** is an internally-generated and trimmed 1% accurate 4MHz time-base. The **CLKREF** is always running, even when not used.

4.4.3 XTALCLK

The **XTALCLK** is the output of the Crystal Driver module and can generate a very accurate system time-base. The **XTALCLK** may be selected by the FRCLK MUX for **FRCLK**.

The crystal driver supports a 2-10MHz external crystal. The crystal driver may be enabled or disabled using the **CCSCTL.XTALEN** bit.

4.4.4 EXTCLK

The **EXTCLK** is an external clock input that allows up to 20MHz, 50% duty cycle clock to be used as a system clock.

EXTCLK is available through a device pin using the Digital Peripheral MUX.

4.5 System Clocks

4.5.1 FRCLK

FRCLK (Free-Running Clock) is the output of the FRCLK MUX and is the default input to the SCLK MUX used for generating **SCLK**. **FRCLK** is the input to the PLL.

4.5.2 PLLCLK

PLLCLK (PLL Clock) is the clock output of the PLL. **PLLCLK** is an optional input to the **SCLK** MUX used for generating **SCLK**. The input to the PLL is **FRCLK**.

4.5.3 SCLK

SCLK (System Clock) is the clock output of the SCLK MUX. SCLK is gated when the PAC55XX is put into ARM deep sleep mode.

The input to the SCLK MUX may be selected as shown below.

Table 4-2 SCLK Clock Input Selections

| CCSCTL.SCLKMUXSEL | SCLK |
|-------------------|--------|
| 0b (default) | FRCLK |
| 1b | PLLCLK |

4.5.4 PCLK

PCLK (APB Clock) is the main peripheral system clock which is generated from **HCLK**. There is an optional input clock divider for **PCLK** that may be used to reduce the frequency to less than 150MHz for the digital peripherals.

PCLK may be gated when the PAC55XX is put into ARM deep sleep mode.

The **PCLK** divider may be configured as follows.

Table 4-3 PCLK Clock Divider Settings

| CCSCTL.PCLKDIV | PCLK |
|----------------|---------|
| 000b (default) | SCLK /1 |
| 001b | SCLK /2 |
| 010b | SCLK /3 |
| 011b | SCLK /4 |
| 100b | SCLK /5 |
| 101b | SCLK /6 |
| 110b | SCLK /7 |
| 111b | SCLK /8 |

4.5.5 ACLK

ACLK (Auxiliary Clock) is clock generated from **SCLK**. **ACLK** is used as an optional clock source for the PWM Timers, to provide very fine timer resolution. The maximum speed of **ACLK** is 300MHz and it has an input clock divider.

ACLK may be gated when the PAC55XX is put into ARM deep sleep mode.

The **ACLK** divider may be configured as follows.

Table 4-4 ACLK Clock Divider Settings

| CCSCTL.ACLKDIV | ACLK |
|----------------|---------|
| 000b (default) | SCLK /1 |
| 001b | SCLK /2 |
| 010b | SCLK /3 |
| 011b | SCLK /4 |
| 100b | SCLK /5 |
| 101b | SCLK /6 |
| 110b | SCLK /7 |
| 111b | SCLK /8 |

4.5.6 HCLK

HCLK (AHB Clock) is clock generated from **SCLK**. **HCLK** is used as the clock source for the **PCLK** system clock and the AHB peripherals such as the ARM Cortex-M4F MCU, memory controller and CRC engine. The maximum speed of **HCLK** is 150MHz and it has an input clock divider.

HCLK may be gated when the PAC55XX is put into ARM deep sleep mode.

The **HCLK** divider may be configured as follows.

Table 4-5 HCLK Clock Divider Settings

| CCSCTL.HCLKDIV | HCLK |
|----------------|---------|
| 000b (default) | SCLK /1 |
| 001b | SCLK /2 |
| 010b | SCLK /3 |
| 011b | SCLK /4 |
| 100b | SCLK /5 |
| 101b | SCLK /6 |

| | |
|-------------|---------|
| 110b | SCLK /7 |
| 111b | SCLK /8 |

4.6 PLL Configuration

The clock input to the PLL may be configured to be **CLKREF**, **XTALCLK** or **EXTCLK** by using **CCSCTL.PLLMUXSEL**. The PLL may be enabled or disabled using **CCSCTL.PLEN**.

To configure the PLL output frequency, the user must know the specific input frequency and the user must set the PLL dividers. The variables used for calculating the PLL output frequency are shown below.

Table 4-6 PLL Configuration Variables

| VARIABLE | SOURCE | DESCRIPTION | CONSTRAINTS |
|-----------|------------------------|-----------------------------|---|
| PLLOUT | - | PLL Output Frequency in MHz | 62.5MHz – 300MHz |
| PLLIN | CCSCTL.PLLMUXS | PLL Input Frequency in MHz | 1MHz \leq PLLIN / PLLINDIV \leq 50MHz |
| PLLINDIV | PLLCTL.PLLINDIV | PLL Input Divider | $1 \leq$ PLLINDIV \leq 15 |
| PLLOUTDIV | PLLCTL.PLLODIV | PLL Output Divider | $0 \leq$ PLLOUTDIV \leq 3 |
| PLLFBDIV | PLLCTL.PLLFBDIV | PLL Feedback Divider | $4 \leq$ PLLFBDIV \leq 16383 |
| PLLVCO | | PLL VCO Frequency | 200MHz \leq PLLVCO \leq 400MHz |

The following formulas can be used to calculate the PLL VCO frequency and PLL output frequency:

- $NO = 2^{PLLOUTDIV}$
- $PLLOUT = PLLIN * (PLLFBDIV / PLLINDIV) / (1 / NO)$
- $PLLVCO = PLLIN * (PLLFBDIV / PLLINDIV)$

The following table shows some example configurations that may be used for PLL configuration.

Table 4-7 PLL Configuration Examples

| PLLIN | PLLINDIV | PLLFBDIV | PLLOUTDIV | PLLVCO | PLLOUT |
|-------|----------|----------|-----------|---------|---------|
| 4MHz | 1 | 15 | 1 | 60MHz | 30 MHz |
| 4 MHz | 1 | 25 | 0 | 100 MHz | 50 MHz |
| 4MHz | 1 | 50 | 1 | 200 MHz | 100 MHz |
| 4 MHz | 1 | 50 | 0 | 200 MHz | 200 MHz |
| 4 MHz | 1 | 75 | 0 | 300 MHz | 300 MHz |

After changing any of the PLL dividers, there is a 500 μ s time period before the PLL is locked and stable. The user should take care to make sure that **PLLOUT** is not used as SCLK, until the PLL has locked.

Additionally, the user can read the **PLLCTL.LOCK** bit to see if the PLL is locked.

4.7 Peripheral Clock Selection

Each peripheral in the PAC55XX has at least one clock input that is available.

The following table shows the set of system peripherals, and what system clocks are available for their use.

Table 4-8 Peripheral Clock Input

| PERIPHERAL | FRCLK | SCLK | PCLK | PCLK /3 | ACLK | HCLK | ROSCCLK | REFCLK |
|-----------------------------------|-------|------|------|---------|------|------|---------|--------|
| RTC | X | | | | | | | |
| WWDT | X | | | | | | X | |
| WIC (Wakeup Interrupt Controller) | | | | | | X | | |
| ADC and DTSE | | X | | | | | | |
| GP Timer A, B | | | X | | | | | |
| ADC EMUX | | | X | | | | | |
| USART A, B, C, D | | | X | | | | | |
| I2C | | | X | | | | | |
| CAN | | | X | | | | | |
| CRC | | | | | | X | | |
| Timer A, B, C, D | | | X | | X | | | |
| SysTick Timer | | | X | X | | X | | |
| ARM MCU | | | | | | X | | |
| SRAM | | | | | | X | | |
| Memory Controller | | | | | | X | X | |
| CCS Clock Failure Detect | | | | | | | | X |

4.8 Low-Power Clock Gating

The PAC55XX supports low-power operation. There are three different modes of low-power operation:

- ARM Cortex-M4F Sleep Mode
- ARM Cortex-M4F Deep Sleep Mode
- System Hibernate Mode

When ARM Cortex-M4F Sleep Mode is entered, just the MCU is put to sleep. All other clocks and peripherals remain active. Any interrupt may wake up the system from this mode.

In System Hibernate Mode, the entire MCU is not powered at all. To wake up, the user must set a wakeup timer in the Analog sub-system, or wake the system with a push-button event on the Analog sub-system. For more information on this mode, see the device-specific User Guide.

In ARM Cortex-M4F Deep Sleep Mode, the user may gate several clocks in the PAC55XX to the peripherals to reduce energy consumption as much as possible, without removing power to the device. Only certain peripherals are able to 'wake up' the device from this state.

Before entering this mode, the firmware should take the following steps:

- Set **CCSCTL.SCLKMUXSEL** to 0b to set **FRCLK** as the **SCLK** input clock (not **PLLCLK**)
- Disable the PLL by setting **CCSCTL.PLLEN** = 0b
- Set **CCSCTL.FRCLKMUXSEL** to 01b (**CLKREF**) to generate **FRCLK** from **CLKREF**, which will always be present
- Disable ROOSC by setting **CCSCTL.ROSCEN** = 0b
- Disable any unwanted analog peripherals such as the ADC (**ADCCTL.ENABLE** = 0b)
- Disable PCLK by setting **CCSCTL.PCLKEN** = 0b
- Disable ACLK by setting **CCSCTL.ACLKEN** = 0b
- Disable ADCCLK by setting **CCSCTL.ADCCLKEN** = 0b

When in this mode, the only remaining clocks are **FRCLK**, and possibly **ROSCCLK** if the user did not disable it. In order to wake up from this mode, a peripheral that uses **FRCLK** or **ROSCCLK** must be used, such as the RTC, WWDT or GPIO.

The STCLK (SysTick timer alternative clock) is fixed to HCLK / 8. This clock may be gated automatically during sleep mode by setting the **CCSCTL.STCLKSLPEN** to 1b. When set to a 0b, this clock remains active during deep sleep mode.

Interrupts from any of these peripherals while in ARM Cortex-M4F deep sleep mode will wake up the device.

For details on which instructions to execute to enter ARM Cortex-M4F deep sleep mode, refer to the ARM Cortex-M4F Technical Reference Manual at <http://www.arm.com>.

4.9 Clock Failure Detection

The clock control system (CCS) has a clock failure detection circuit. This circuit may be enabled to detect a clock failure on **FRCLK** or **PLLCLK**. To enable this circuit, set **CCSCTL.CLKFAILEN** to 1b. This circuit has an input clock of **CLKREF**, so it will always be clocked.

This circuit may sample two clocks: **FRCLK** or **PLLCLK**. The user may select the clock failure detection input clock as shown below.

Table 4-9 Clock Failure Detection Configuration

| CCSCTL.CLKFAILMUXS | FRCLK | PLLCLK |
|--------------------|-------|--------|
| 0b (default) | X | |
| 1b | | X |

When a clock failure is detected on the selected sampled clock, the following actions are taken:

- **CCSCTL.CLKFAILIF** is set to 1b.
- A NMI of IRQ_CCS is asserted to the NVIC
- The FRCLK MUX is set to **ROSCCLK** as the clock source (**CCSCTL.FRCLKMUXSEL** = 00b)
- The SCLK MUX is set to use **FRCLK** as the clock source (**CCSCTL.SCLKMUXSEL** = 0b)

These actions will put the device into safe state where firmware may still run to handle this condition.

4.10 USART Mode

The CCSCTL register is used to configure the USART A to USART D mode to be SSP or UART. The description of how to use these mode settings can be found in:

- USART A
- USART B
- USART C
- USART D

4.11 GPIO Drive Strength

The registers to configure GPIO Drive Strength are located in the SCC module as shown below. The description of how to use these registers is given in the following GPIO sections:

- GPIOA Drive Strength
- GPIOB Drive Strength
- GPIOC Drive Strength
- GPIOD Drive Strength
- GPIOE Drive Strength
- GPIOF Drive Strength
- GPIOG Drive Strength

4.12 GPIO Pull-up and Pull-down Configuration

The registers to configure GPIO weak pull-up and pull-down as well as Schmitt Trigger Input are located in the SCC module as shown below.

The description of how to use these registers is given in the following GPIO sections:

- GPIOA Weak Pull-up and Pull-down
- GPIOB Weak Pull-up and Pull-down
- GPIOC Weak Pull-up and Pull-down
- GPIOD Weak Pull-up and Pull-down
- GPIOE Weak Pull-up and Pull-down
- GPIOF Weak Pull-up and Pull-down
- GPIOG Weak Pull-up and Pull-down

4.13 Digital Peripheral MUX

The registers to configure the Digital Peripheral MUX are located in the SCC module as shown in the tables below.

4.14 Interrupt Vector Table

The PAC55XX supports 31 user interrupts for the various peripherals. The interrupt vector table is located at address 0 in FLASH memory by default.

Each interrupt has 3 levels of priority

The interrupt vector table is shown below.

Table 4-10 PAC55XX Interrupt Vector Table

| Exception | IRQ | Addr Offset | Interrupt | Description |
|--------------------------|-----|-------------|------------------|------------------------------------|
| System Exceptions | | | | |
| | | 0000h | Initial SP Value | Initial value of the stack pointer |
| 1 | | 0004h | Reset | Reset Vector Address |
| 2 | -14 | 0008h | NMI | Non-Maskable Interrupt |

| | | | | |
|------------------------|-----|-------|--------------------|-----------------------|
| 3 | -13 | 000Ch | HardFault | |
| 4 | -12 | 0010h | MemManage | |
| 5 | -11 | 0014h | BusFault | |
| 6 | -10 | 0018h | UsageFault | |
| 7 | | 001Ch | Reserved | |
| 8 | | 0020h | Reserved | |
| 9 | | 0024h | Reserved | |
| 10 | | 0028h | Reserved | |
| 11 | -5 | 002Ch | SVCall | |
| 12 | | 0030h | Reserved for Debug | |
| 13 | | 0034h | Reserved | |
| 14 | -2 | 0038h | PendSV | |
| 15 | -1 | 003Ch | SysTick | |
| User Interrupts | | | | |
| 16 | 0 | 0040h | Memory Controller | Memory Controller IRQ |
| 17 | 1 | 0044h | WWDT | WWDT IRQ |
| 18 | 2 | 0048h | RTC | RTC with Calendar IRQ |
| 19 | 3 | 004Ch | ADC0 | ADC IRQ 0 |
| 20 | 4 | 0050h | ADC1 | ADC IRQ 1 |
| 21 | 5 | 0054h | ADC2 | ADC IRQ 2 |
| 22 | 6 | 0058h | ADC3 | ADC IRQ 3 |
| 23 | 7 | 005Ch | Timer A | Timer A IRQ |
| 24 | 8 | 0060h | Timer B | Timer B IRQ |
| 25 | 9 | 0064h | Timer C | Timer C IRQ |
| 26 | 10 | 0068h | Timer D | Timer D IRQ |
| 27 | 11 | 006Ch | Timer A QEP | Timer A QEP IRQ |
| 28 | 12 | 0070h | Timer B QEP | Timer B QEP IRQ |
| 29 | 13 | 0074h | Timer C QEP | Timer C QEP IRQ |
| 30 | 14 | 0078h | Timer D QEP | Timer D QEP IRQ |
| 31 | 15 | 007Ch | GPIOA | GPIO A IRQ |
| 32 | 16 | 0080h | GPIOB | GPIO B IRQ |
| 33 | 17 | 0084h | GPIOC | GPIO C IRQ |
| 34 | 18 | 0088h | GPIOD | GPIO D IRQ |
| 35 | 19 | 008Ch | GPIOE | GPIO E IRQ |
| 36 | 20 | 0090h | GPIOF | GPIO F IRQ |

| | | | | |
|-----------|----|-------|----------|--------------------------|
| 37 | 21 | 0094h | GPIOG | GPIO G IRQ |
| 38 | 22 | 0098h | I2C | I2C IRQ |
| 39 | 23 | 009Ch | USARTA | USART A IRQ |
| 40 | 24 | 0100h | USARTB | USART B IRQ |
| 41 | 25 | 0104h | USARTC | USART C IRQ |
| 42 | 26 | 0108h | USARTD | USART D IRQ |
| 43 | 27 | 010Ch | CAN | CAN IRQ |
| 44 | 28 | 0110h | GPTIMERA | GP Timer A IRQ |
| 45 | 29 | 0114h | GPTIMERB | GP Timer B IRQ |
| 46 | 30 | 0118h | SCC | System Clock Control IRQ |
| 47 | 31 | | Reserved | |

4.15 Register Summary

Table 4-11 SCC Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------|------------|-----------------------------------|--------|------------|
| CCSCTL | 400D 0400h | CCS Control | RW | 0000 F004h |
| CCSPLLCTL | 400D 0404h | PLL Control | RW | 0000 0000h |
| CCSROSTRIM | 400D 0408h | ROSC Trim Control | RW | 0000 007Fh |
| PAMUXSEL | 400D 040Ch | PA Peripheral MUX Select | RW | 0111 1110h |
| PBMUXSEL | 400D 0410h | PB Peripheral MUX Select | RW | 0000 0000h |
| PCMUXSEL | 400D 0414h | PC Peripheral MUX Select | RW | 0000 0000h |
| PDMUXSEL | 400D 0418h | PD Peripheral MUX Select | RW | 0000 0000h |
| PEMUXSEL | 400D 041Ch | PE Peripheral MUX Select | RW | 0000 0000h |
| PFMUXSEL | 400D 0420h | PF Peripheral MUX Select | RW | 0000 3333h |
| PGMUXSEL | 400D 0424h | PG Peripheral MUX Select | RW | 0000 0000h |
| PAPUEN | 400D 0428h | PA Weak Pull-up Enable | RW | 0000 0000h |
| PBPUEN | 400D 042Ch | PB Weak Pull-up Enable | RW | 0000 0000h |
| PCPUEN | 400D 0430h | PC Weak Pull-up Enable | RW | 0000 0000h |
| PDCPUEN | 400D 0434h | PD Weak Pull-up Enable | RW | 0000 0000h |
| PEPUEN | 400D 0438h | PE Weak Pull-up Enable | RW | 0000 0000h |
| PFPUEN | 400D 043Ch | PF Weak Pull-up Enable | RW | 0000 0000h |
| PGPUEN | 400D 0440h | PG Weak Pull-up Enable | RW | 0000 0000h |
| PAPDEN | 400D 0444h | PA Weak Pull-down Enable | RW | 0000 0000h |
| PBPDEN | 400D 0448h | PB Weak Pull-down Enable | RW | 0000 0000h |
| PCPDEN | 400D 044Ch | PC Weak Pull-down Enable | RW | 0000 0000h |
| PDPDEN | 400D 0450h | PD Weak Pull-down Enable | RW | 0000 0000h |
| PEPDEN | 400D 0454h | PE Weak Pull-down Enable | RW | 0000 0000h |
| PFPDEN | 400D 0458h | PF Weak Pull-down Enable | RW | 0000 0000h |
| PGPDEN | 400D 045Ch | PG Weak Pull-down Enable | RW | 0000 0000h |
| PADS | 400D 0460h | PA Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PBDS | 400D 0464h | PB Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PCDS | 400D 0468h | PC Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PDDS | 400D 046Ch | PD Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PEDS | 400D 0470h | PE Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PFDS | 400D 0474h | PF Drive Strength/Schmitt Trigger | RW | 0000 0000h |
| PGDS | 400D 0478h | PG Drive Strength/Schmitt Trigger | RW | 0000 0000h |

4.16 Register Detail

4.16.1 CCSCTL

Register 4-1 CCSCTL (CCS Configuration, 400D 0400h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | USDMODE | RW | 0b | USART D Mode: 0b: SSP 1b: UART |
| 30 | USCMODE | RW | 0b | USART C Mode: 0b: SSP 1b: UART |
| 29 | USBMODE | RW | 0b | USART B Mode: 0b: SSP 1b: UART |
| 28 | USAMODE | RW | 0b | USART A Mode: 0b: SSP 1b: UART |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | HCLKDIV | RW | 000b | HCLK Divider: 000b: SCLK /1 001b: SCLK /2 010b: SCLK /3 011b: SCLK /4 100b: SCLK /5 101b: SCLK /6 110b: SCLK /7 111b: SCLK /8 |
| 23 | Reserved | RO | 0b | Reserved |
| 22:20 | ACLKDIV | RW | 000b | ACLK Divider: 000b: SCLK /1 001b: SCLK /2 010b: SCLK /3 011b: SCLK /4 100b: SCLK /5 101b: SCLK /6 110b: SCLK /7 111b: SCLK /8 |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | PCLKDIV | RW | 000b | PCLK Divider: 000b: HCLK /1 001b: HCLK /2 010b: HCLK /3 011b: HCLK /4 100b: HCLK /5 101b: HCLK /6 110b: HCLK /7 |

| | | | | |
|------|----------------------|-----|-------|---|
| | | | | 111b: HCLK /8 |
| 15 | STCLKSLPEN | RW | 1b | STCLK Sleep enable: 0b: disabled: STCLK active in deep sleep mode 1b: enabled: STCLK gated in deep sleep mode |
| 14 | ADCCLKEN | RW | 1b | ADCCLK enable: 0b: disabled 1b: enabled |
| 13 | ACLKEN | RW | 1b | ACLK enable: 0b: disabled 1b: enabled |
| 12 | PCLKEN | RW | 1b | PCLK enable: 0b: disabled 1b: enabled |
| 11 | SWPOR | RW | 0b | Software POR |
| 10:9 | Reserved | RO | 0000b | Reserved |
| 8 | LDOEN | RW | 0b | 1.8V LDO Disable: 0b: Not disabled 1b: Disabled |
| 7 | CLKFAILIF | W1C | 0 | Clock Fail Interrupt Flag: 0b: no flag 1b: flag |
| 6 | CLKFAILMUXSEL | RW | 0 | Clock Fail MUX Select: 0b: FRCLK 1b: PLLCLK |
| 5 | CLKFAILEN | RW | 0 | Clock Fail Detection Enabled: 0b: disabled 1b: enabled |
| 4 | SCLKMUXSEL | RW | 0 | SCLK MUX Select: 0b: FRCLK 1b: PLLCLK |
| 3 | XTALEN | RW | 0 | Crystal Driver Enabled: 0b: disabled 1b: enabled |
| 2 | ROSCEN | RW | 1 | ROSC Enabled: 0b: disabled 1b: enabled |
| 1:0 | FRCLKMUXSEL | RW | 00b | FRCLK MUX Select: 00b: ROSC 01b: CLKREF 10b: XTALCLK 11b: EXTCLK |

4.16.2 CCSPLLCTL

Register 4-2 CCSPLLCTL (CCS PLL Configuration, 400D 0404h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:25 | Reserved | RO | 0b | Reserved |
| 24 | PLLLOCK | RO | 0b | PLL Lock Status: 0b: PLL not locked 1b: PLL locked |
| 23:22 | Reserved | RO | 0b | |
| 21:8 | PLLFBDIV | RW | 0 | PLL Feedback Divider. Must be between 4 – 16383. |
| 7:4 | PLLINDIV | RW | 0 | PLL Input Divider. Must be between 1 – 15. |
| 3:2 | PLLOUTDIV | RW | 0 | PLL Output Divider: 00b: /1 01b: /2 10b: /3 11b: /4 |
| 1 | PLLBP | RW | 0 | PLL Bypass: 0b: Bypass inactive 1b: Bypass active |
| 0 | PLLEN | RW | 0 | PLL Enabled: 0b: PLL Disabled 1b: PLL Enabled |

4.16.3 CCSROSCTRIM

Register 4-3 CCSROSCTRIM (CCS ROSC Trim Configuration, 400D 0408h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-----------|------------------|
| 31:7 | Reserved | RO | 0b | Reserved |
| 6:0 | TRIM | RW | 111 1111b | ROSC Trim Value. |

4.16.4 PAMUXSEL

Register 4-4 PAMUXSEL (PA Peripheral MUX Select, 400D 040Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOA7 [default] 001b-111b: Reserved |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 001b | Pin 6 MUX Select: 000b: GPIOA6 001b: USASS [default] 010b: USBSS 011b-111b: Reserved |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 001b | Pin 5 MUX Select: 000b: GPIOA5 001b: USAMISO [default] 010b: USBMISO 011b-111b: Reserved |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 001b | Pin 4 MUX Select: 000b: GPIOA4 001b: USAMOSI [default] 010b: USBMOSI 011b-111b: Reserved |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 001b | Pin 3 MUX Select: 000b: GPIOA3 001b: USASCLK [default] 010b: USBSCLK 011b-111b: Reserved |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 001b | Pin 2 MUX Select: 000b: GPIOA2 001b: EMUXC [default] 010b-111b: Reserved |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 001b | Pin 1 MUX Select: 000b: GPIOA1 001b: EMUXD [default] |

| | | | | |
|-----|-----------------|----|------|--|
| | | | | 010b-111b: Reserved |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOA0 [default] 001b-111b: Reserved |

4.16.5 PBMUXSEL

Register 4-5 PBMUXSEL (PB Peripheral MUX Select, 400D 0410h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOB7 [default] 001b: TAPWM7 010b: TBPWM7 011b-111b: Reserved |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOB6 [default] 001b: TAPWM6 010b: TBPWM6 011b-111b: Reserved |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOB5 [default] 001b: TAPWM5 010b: TBPWM5 011b-111b: Reserved |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 000b | Pin 4 MUX Select: 000b: GPIOB4 [default] 001b: TAPWM4 010b: TBPWM4 011b-111b: Reserved |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 000b | Pin 3 MUX Select: 000b: GPIOB3 [default] 001b: TAPWM3 010b: TBPWM3 011b-111b: Reserved |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 000b | Pin 2 MUX Select: 000b: GPIOB2 [default] 001b: TAPWM2 010b: TBPWM2 011b-111b: Reserved |
| 7 | Reserved | RO | 0b | Reserved |

| | | | | |
|-----|-----------------|----|------|--|
| 6:4 | P1 | RW | 000b | Pin 1 MUX Select: 000b: GPIOB1 [default] 001b: TAPWM1 010b: TBPWM1 011b-111b: Reserved |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOB0 [default] 001b: TAPWM0 010b: TBPWM0 |

4.16.6 PCMUXSEL

Register 4-6 PCMUXSEL (PC Peripheral MUX Select, 400D 0414h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOC7 [default] 001b: TBPWM7 010b: TCPWM7 011b: Reserved 100b: USBSS 101b: USCMISO 110b: FRCLK 111b: EMUXC |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOC6 [default] 001b: TBPWM6 010b: TCPWM6 011b: TCPHB 100b: USBSCLK 101b: USCMOSI 110b: nRESET 111b: EMUXD |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOC5 [default] 001b: TBPWM5 010b: TCPWM5 011b: TCQEPPHA 100b: USBMISO 101b: USCSS 110b: CANTXD 111b: I2CSDA |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 001b | Pin 4 MUX Select: 000b: GPIOC4 [default] 001b: TBPWM4 010b: TCPWM4 011b: TCQEPIDX 100b: USBMOSI 101b: USCCLK 110b: CANRXD 111b: I2CSCL |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 000b | Pin 3 MUX Select: 000b: GPIOC3 [default] 001b: TBPWM3 |

| | | | | |
|------|-----------------|----|------|---|
| | | | | 010b: TCPWM3 011b: Reserved 100b: USBSS 101b: USCMISO 110b: nRESET 111b: EMUXC |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 000b | Pin 2 MUX Select: 000b: GPIOC2 [default] 001b: TBPWM2 010b: TCPWM2 011b: TBQEPPHB 100b: USBCLK 101b: USCMOSI 110b: Reserved 111b: EMUXD |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 000b | Pin 1 MUX Select: 000b: GPIOC1 [default] 001b: TBPWM1 010b: TCPWM1 011b: TBQEPPHA 100b: USBMISO 101b: USSS 110b: CANTXD 111b: I2CSDA |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOC1 [default] 001b: TBPWM0 010b: TCPWM0 011b: TBQEPIDX 100b: USBMOSI 101b: USSCLK 110b: CANRXD 111b: I2CSCL |

4.16.7 PDMUXSEL

Register 4-7 PDMUXSEL (PD Peripheral MUX Select, 400D 0418h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOD7 [default] 001b: TBPWM7 010b: TCPWM7 011b: Reserved 100b: Reserved 101b: USDMISO 110b: CANRXD 111b: I2CSCL |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOC6 [default] 001b: TBPWM6 010b: TCPWM6 011b: TDQEPPHB 100b: TBQEPPHB 101b: USDMOSI 110b: CANTXD 111b: I2CSDA |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOD5 [default] 001b: TBPWM5 010b: TCPWM5 011b: TDQEPPHA 100b: TBQEPPHA 101b: USDSS 110b: CANRXD 111b: USDMISO |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 000b | Pin 4 MUX Select: 000b: GPIOD4 [default] 001b: TBPWM4 010b: TCPWM4 011b: TDQEPIDX 100b: TBQEPIDX 101b: USDCLK 110b: TRACED3 111b: USDMOSI |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 000b | Pin 3 MUX Select: 000b: GPIOD3 [default] 001b: TBPWM3 |

| | | | | |
|------|-----------------|----|------|---|
| | | | | 010b: TCPWM3 011b: Reserved 100b: TDO 101b: USCMISO 110b: FRCLK 111b: TRACED3 |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 000b | Pin 2 MUX Select: 000b: GPIOD2 [default] 001b: TBPWM2 010b: TCPWM2 011b: TDQEPPHB 100b: TDI 101b: USCMOSI 110b: nRESET 111b: Reserved |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 000b | Pin 1 MUX Select: 000b: GPIOD1 [default] 001b: TBPWM1 010b: TCPWM1 011b: TDQEPPHA 100b: TMS/SWDIO 101b: USCSS 110b: CANRXD 111b: EMUXC |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOD0 [default] 001b: TBPWM0 010b: TCPWM0 011b: TDQEPIX 100b: TCK/SWDCLK 101b: USCCLK 110b: CANTXD 111b: EMUXD |

4.16.8 PEMUXSEL

Register 4-8 PEMUXSEL (PE Peripheral MUX Select, 400D 041Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOE7 [default] 001b: TCPWM7 010b: TDPWM7 011b: Reserved 100b: USBMISO 101b: USDSS 110b: CANTXD 111b: TDO |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOE6 [default] 001b: TCPWM6 010b: TDPWM6 011b: TDQEPPHB 100b: USBMOSI 101b: USDCLK 110b: CANRXD 111b: TDI |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOE5 [default] 001b: TCPWM5 010b: TDPWM5 011b: TDQEPPHA 100b: USBSS 101b: USDMISO 110b: I2CSDA 111b: TMS/SWDIO |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 000b | Pin 4 MUX Select: 000b: GPIOE4 [default] 001b: TCPWM4 010b: TDPWM4 011b: TDQEPIDX 100b: USBCLK 101b: USDMOSI 110b: I2CSCL 111b: TCK/SWCCLK |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 000b | Pin 3 MUX Select: 000b: GPIOE3 [default] 001b: TCPWM3 |

| | | | | |
|------|-----------------|----|------|--|
| | | | | 010b: TDPWM3 011b: FRCLK 100b: nReset 101b: USCMISO 110b: CANTXD 111b: Reserved |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 000b | Pin 2 MUX Select: 000b: GPIOE2 [default] 001b: TCPWM2 010b: TDPWM2 011b: TAQEPPHA 100b: TAQEPPHB 101b: USCMOSI 110b: CANRXD 111b: EXTCLK |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 000b | Pin 1 MUX Select: 000b: GPIOE1 [default] 001b: TCPWM1 010b: TDPWM1 011b: TAQEPPHA 100b: TBQEPPHA 101b: USCSS 110b: I2CSDA 111b: EMUXD |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOE0 [default] 001b: TCPWM0 010b: TDPWM0 011b: TAQEPIDX 100b: TBQEPIDX 101b: USCCLK 110b: I2CSCL 111b: EMUXC |

4.16.9 PFMUXSEL

Register 4-9 PFMUXSEL (PF Peripheral MUX Select, 400D 0420h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOF7 [default] 001b: TCPWM7 010b: TDPWM7 011b: TDO 100b: nRESET 101b: USDMISO 110b: CANTXD 111b: I2CSDA |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOF6 [default] 001b: TCPWM6 010b: TDPWM6 011b: TDI 100b: TCQEPPHB 101b: USDMOSI 110b: CANRXD 111b: I2CSCL |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOF5 [default] 001b: TCPWM5 010b: TDPWM5 011b: TMS/SWDIO 100b: TCQEPPHA 101b: USDSS 110b: Reserved 111b: EMUXD |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 000b | Pin 4 MUX Select: 000b: GPIOF4 [default] 001b: TCPWM4 010b: TDPWM4 011b: TCK/SWDCLK 100b: TCQEPIDX 101b: USDCLK 110b: TRACED3 111b: EMUXC |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 011b | Pin 3 MUX Select: 000b: GPIOF3 001b: TCPWM3 |

| | | | | |
|------|-----------------|----|------|--|
| | | | | 010b: TDPWM3 011b: TDO [default] 100b: FRCLK 101b: USBMISO 110b: TRACED2 111b: Reserved |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 011b | Pin 2 MUX Select: 000b: GPIOF2 001b: TCPWM2 010b: TDPWM2 011b: TDI [default] 100b: TBQEPPHB 101b: USBMOSI 110b: TRACED1 111b: Reserved |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 011b | Pin 1 MUX Select: 000b: GPIOF1 001b: TCPWM1 010b: TDPWM1 011b: TMS/SWDIO [default] 100b: TBQEPPHA 101b: USBSS 110b: TRACED0 111b: Reserved |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 011b | Pin 0 MUX Select: 000b: GPIOF0 001b: TCPWM0 010b: TDPWM0 011b: TCK/SWDCLK [default] 100b: TBQEPIDX 101b: USBSCLK 110b: TRACECLK 111b: Reserved |

4.16.10 PGMUXSEL

Register 4-10 PGMUXSEL (PG Peripheral MUX Select, 400D 0424h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31 | Reserved | RO | 0b | Reserved |
| 30:28 | P7 | RW | 000b | Pin 7 MUX Select: 000b: GPIOG7 [default] 001b: Reserved 010b: TDQEPIDX 011b: Reserved 100b: Reserved 101b: USDCLK 110b: Reserved 111b: Reserved |
| 27 | Reserved | RO | 0b | Reserved |
| 26:24 | P6 | RW | 000b | Pin 6 MUX Select: 000b: GPIOG6 [default] 001b: TCPWM6 010b: TDPWM6 011b: I2CSDA 100b: Reserved 101b: USDMISO 110b: CANTXD 111b: TDQEPPHB |
| 23 | Reserved | RW | 0b | Reserved |
| 22:20 | P5 | RW | 000b | Pin 5 MUX Select: 000b: GPIOG5 [default] 001b: TCPWM5 010b: TDPWM5 011b: EMUXC 100b: Reserved 101b: USDMOSI 110b: CANRXD 111b: TDQEPPHA |
| 19 | Reserved | RO | 0b | Reserved |
| 18:16 | P4 | RW | 000b | Pin 4 MUX Select: 000b: GPIOG4 [default] 001b: TCPWM4 010b: TDPWM4 011b: EMUXD 100b: I2CSCL 101b: USDSS 110b: TRACED3 111b: TDQEPIDX |
| 15 | Reserved | RO | 0b | Reserved |
| 14:12 | P3 | RW | 000b | Pin 3 MUX Select: 000b: GPIOG3 [default] 001b: TCPWM3 |

| | | | | |
|------|-----------------|----|------|--|
| | | | | 010b: TDPWM3 011b: nRESET 100b: TDO 101b: USDMISO 110b: TRACED2 111b: Reserved |
| 11 | Reserved | RO | 0b | Reserved |
| 10:8 | P2 | RW | 000b | Pin 2 MUX Select: 000b: GPIOG2 [default] 001b: TCPWM2 010b: TDPWM2 011b: FRCLK 100b: TDI 101b: USDMOSI 110b: TRACED1 111b: TCQEPPHB |
| 7 | Reserved | RO | 0b | Reserved |
| 6:4 | P1 | RW | 000b | Pin 1 MUX Select: 000b: GPIOG1 [default] 001b: TCPWM1 010b: TDPWM1 011b: EMUXD 100b: TMS/SWDIO 101b: USDSS 110b: TRACED0 111b: TCQEPPHA |
| 3 | Reserved | RO | 0b | Reserved |
| 2:0 | P0 | RW | 000b | Pin 0 MUX Select: 000b: GPIOG0 [default] 001b: TCPWM0 010b: TDPWM0 011b: EMUXC 100b: TCK/SWCLK 101b: USDCLK 110b: TRACECLK 111b: TCQEPIX |

4.16.11 PAPUEN

Register 4-11 PAPUEN (PA Pull-up Enable, 400D 0428h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.12 PBPUE

Register 4-12 PBPUE (PB Pull-up Enable, 400D 042Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.13 PCPUEN

Register 4-13 PCPUEN (PC Pull-up Enable, 400D 0430h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.14 PDPHEN

Register 4-14 PDPHEN (PD Pull-up Enable, 400D 0034h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.15 PEPHEN

Register 4-15 PEPHEN (PE Pull-up Enable, 400D 0438h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.16 PFPUE

Register 4-16 PFPUE (PF Pull-up Enable, 400D 043Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.17 PGPUEN

Register 4-17 PGPUEN (PG Pull-up Enable, 400D 0440h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-up enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-up enabled: 0b: disabled 1b: enabled |

4.16.18 PAPDEN

Register 4-18 PAPDEN (PA Pull-down Enable, 400D 0444h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.19 PBPDEN

Register 4-19 PBPDEN (PB Pull-down Enable, 400D 0448h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.20 PCPDEN

Register 4-20 PCPDEN (PC Pull-down Enable, 400D 044Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.21 PDPDEN

Register 4-21 PDPDEN (PD Pull-down Enable, 400D 0450h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.22 PEPDEN

Register 4-22 PEPDEN (PE Pull-down Enable, 400D 0454h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.23 PFPDEN

Register 4-23 PFPDEN (PF Pull-down Enable, 400D 0458h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.24 PGPDEN

Register 4-24 PGPDEN (PG Pull-down Enable, 400D 045Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0b | Reserved |
| 7 | P7 | RW | 0b | Pin 7 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 6 | P6 | RW | 0b | Pin 6 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 5 | P5 | RW | 0b | Pin 5 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 4 | P4 | RW | 0b | Pin 4 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 3 | P3 | RW | 0b | Pin 3 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 2 | P2 | RW | 0b | Pin 2 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 1 | P1 | RW | 0b | Pin 1 Weak pull-down enabled: 0b: disabled 1b: enabled |
| 0 | P0 | RW | 0b | Pin 0 Weak pull-down enabled: 0b: disabled 1b: enabled |

4.16.25 PADS

Register 4-25 PADS (PA Drive Strength/Schmitt Trigger, 400D 0460h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt 3rigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.26 PBDS

Register 4-26 PBDS (PB Drive Strength/Schmitt Trigger, 400D 0464h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt 3rigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.27 PCDS

Register 4-27 PCDS (PC Drive Strength/Schmitt Trigger, 400D 0468h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.28 PDDS

Register 4-28 PDDS (PD Drive Strength/Schmitt Trigger, 400D 046Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt 3rigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.29 PEDS

Register 4-29 PEDS (PE Drive Strength/Schmitt Trigger, 400D 0470h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt 3rigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.30 PFDS

Register 4-30 PFDS (PF Drive Strength/Schmitt Trigger, 400D 0474h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt 3rigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

4.16.31 PGDS

Register 4-31 PGDS (PG Drive Strength/Schmitt Trigger, 400D 0478h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------|--------|-------|---|
| 31 | P7ST | RW | 0b | Pin 7 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 30:28 | P7DS | RW | 000b | Pin 7 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 37 | P6ST | RW | 0b | Pin 6 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 36:24 | P6DS | RW | 000b | Pin 6 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 23 | P5ST | RW | 0b | Pin 5 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 22:20 | P5DS | RW | 000b | Pin 5 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 19 | P4ST | RW | 0b | Pin 4 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 18:16 | P4DS | RW | 000b | Pin 4 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA |

| | | | | |
|-------|-------------|----|------|---|
| | | | | 111b: 25mA |
| 15 | P3ST | RW | 0b | Pin 3 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 14:12 | P3DS | RW | 000b | Pin 3 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 11 | P2ST | RW | 0b | Pin 2 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 10:8 | P2DS | RW | 000b | Pin 2 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 7 | P1ST | RW | 0b | Pin 1 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 6:4 | P1DS | RW | 000b | Pin 1 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |
| 3 | P0ST | RW | 0b | Pin 0 Schmitt Trigger Enable: 0b: disabled 1b: enabled |
| 2:0 | P0DS | RW | 000b | Pin 0 Drive Strength: 000b: 6mA 001b: 8mA 010b: 11mA 011b: 14mA 100b: 17mA 101b: 20mA 110b: 22mA 111b: 25mA |

5 MEMCTL

5.1 Overview

The Memory Controller (MEMCTL) allows access to the internal program FLASH, as well as INFO-1, INFO-2, INFO-3 and SRAM. The access to the memory controller is through the AHB bus for reading and writing FLASH and RAM, as well as register access for control and configuration.

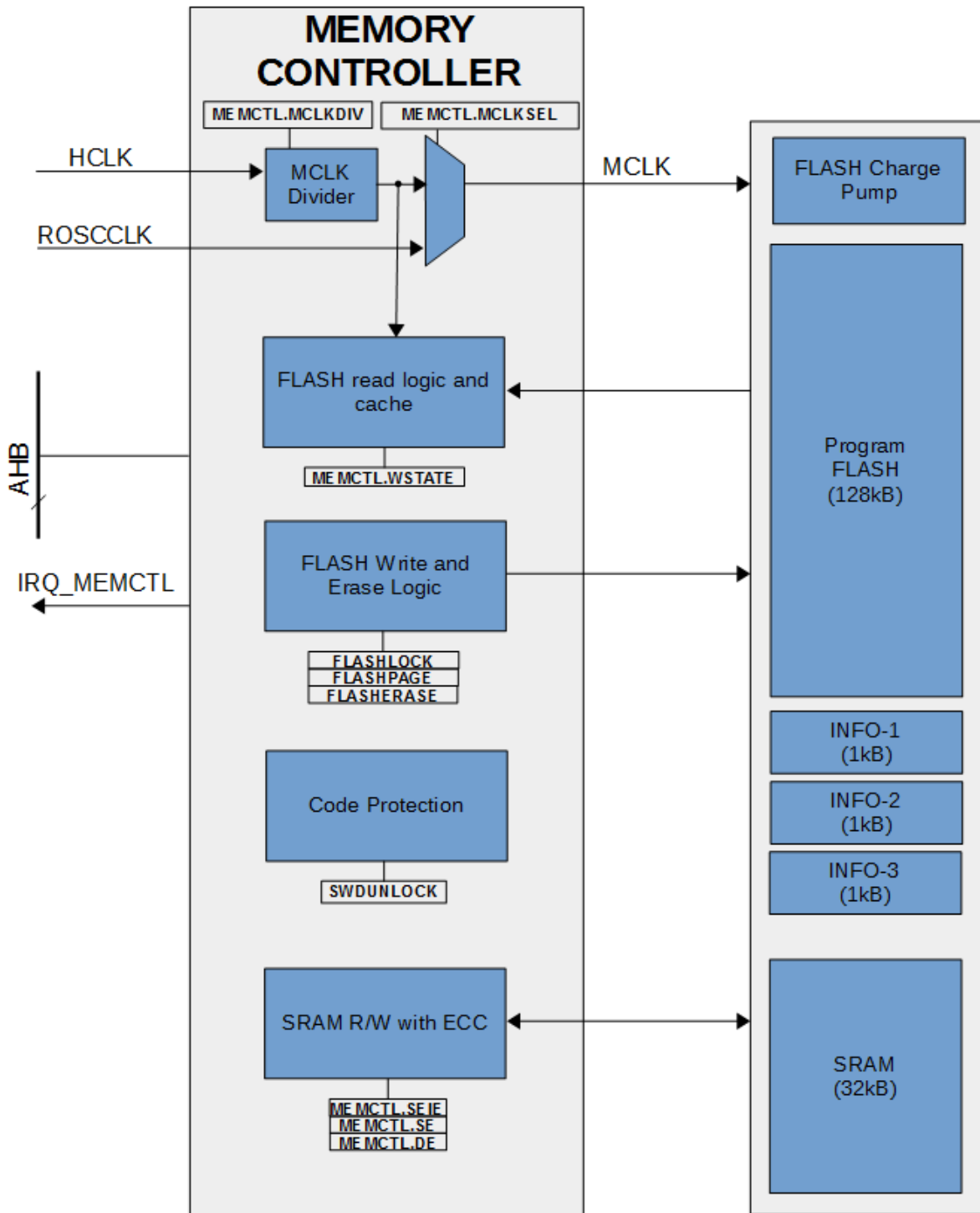
The memory controller has a single clock input (HCLK) and has a single interrupt signal to the NVIC (IRQ_MEMCTL).

5.2 Features

- FLASH control for read, erase, write operations
- FLASH read cache
- Invalid address interrupts
- SRAM control for read/write
- SECDED ECC
 - Single and Double bit error interrupts
- Code Protection

5.3 System Block Diagram

Figure 5-1 Memory Controller System Block Diagram



5.4 Functional Description

5.4.1 Clocking

The memory controller is an AHB bus client and it may be clocked with either HCLK or ROSCCLK. By default, the PAC55XX will supply a 16MHz ROSCCLK to the memory controller via HCLK.

The FLASH memory on the PAC55XX has two clocking requirements:

- FLASH read frequency must be a maximum of 25MHz
- The FLASH charge pump has an input clock (MCLK) that must obey the following clock frequencies:
 - 16MHz – 30MHz nominal MCLK for FLASH read
 - 30MHz nominal MCLK for FLASH read, write or erase

In order to support these frequencies, the memory controller can be configured as described below.

The memory controller has a clock divider for HCLK to generate MCLK (16MHz – 30MHz). In order to do this, there is a MCLK divider that the user may use to generate the proper MCLK. The MCLK divider may be configured using the **MEMCTL.MCLKDIV** field as shown below.

Note that any time the user wants to write the **MEMCTL** register, they must first write the **FLASHLOCK** with the value D513 B490h.

Table 5-1 Memory Controller MCLK Divider

| MEMCTL.MCLKDIV | MCLK Output |
|----------------|-------------|
| 0000b | HCLK /1 |
| 0001b | HCLK /2 |
| 0010b | HCLK /3 |
| 0011b | HCLK /4 |
| 0100b | HCLK /5 |
| 0101b | HCLK /6 |
| 0110b | HCLK /7 |
| 0111b | HCLK /8 |
| 1000b | HCLK /9 |
| 1001b | HCLK /10 |
| 1010b | HCLK /11 |
| 1011b | HCLK /12 |
| 1100b | HCLK /13 |
| 1101b | HCLK /14 |

| | |
|-------|----------|
| 1110b | HCLK /15 |
| 1111b | HCLK /16 |

The maximum FLASH read frequency is 25MHz. In order to control this frequency to FLASH, the memory controller has wait states that it can insert when performing transactions to FLASH. This will insert a programmable number of wait states to HCLK for FLASH access as shown in the table below.

There are always two wait states, so the **MEMCTL.WSTATE** value will add to the two wait states for accessing FLASH.

Table 5-2 Memory Controller FLASH Wait States

| MEMCTL.MCLKDIV | Wait States |
|----------------|----------------|
| 0000b | 0 wait states |
| 0001b | 1 wait states |
| 0010b | 2 wait states |
| 0011b | 3 wait states |
| 0100b | 4 wait states |
| 0101b | 5 wait states |
| 0110b | 6 wait states |
| 0111b | 7 wait states |
| 1000b | 8 wait states |
| 1001b | 9 wait states |
| 1010b | 10 wait states |
| 1011b | 11 wait states |
| 1100b | 12 wait states |
| 1101b | 13 wait states |
| 1110b | 14 wait states |
| 1111b | 15 wait states |

The table below shows some examples of how to configure the memory controller to support the FLASH read frequency and MCLK frequency for various HCLK input clock frequencies.

Table 5-3 FLASH Read Clock Configuration Examples

| HCLK | MEMCTL.MCLKDIV | MCLK | MEMCTL.WSTATE | FLASH READ FREQ | FLASH READ OK | FLASH WRITE OK |
|--------|--------------------|-------|--------------------------|-----------------------|---------------|----------------|
| 150MHz | 0100b (HCLK /5) | 30MHz | 0110b (6 wait states) | 150MHz / 6 = 25MHz | X | X |

| | | | | | | |
|--------|--------------------|-----------------------|--------------------------|------------------------|---|---|
| 125MHz | 0011b (HCLK /4) | 31.25MHz ² | 0101b (5 wait states) | 125MHz / 5 = 25MHz | X | X |
| 90MHz | 0010b (HCLK /3) | 30MHz | 0100b (4 wait states) | 90MHz / 4 = 22.5MHz | X | X |
| 60MHz | 0001b (HCLK /2) | 30MHz | 0011b (3 wait states) | 60MHz / 3 = 20MHz | X | X |
| 16MHz | 0000b (HCLK /1) | 16MHz | 0010b (2 wait states) | 30MHz / 2 = 15MHz | X | |

Note that for the last entry in the table above, 16MHz is only valid for FLASH read, not for FLASH program or erase.

5.4.2 Low-Power Operation

If FLASH is going to be unused by the application and the user wishes to conserve power, it may put the FLASH into a standby mode. If using this mode, be sure that the program does not need to execute any code from FLASH, only from RAM otherwise the system will stall.

To enable standby mode set the **MEMCTL.STDBY** to 1b. To exit standby mode set **MEMCTL.STDBY** to 0b.

Note that any time the user wants to write the **MEMCTL** register, they must first write the **FLASHLOCK** with the value D513 B490h.

To disable the 1.8V FLASH supply, the user may write the **CCSCTL.LDOEN** bit to 0b. When set to 0b, the FLASH memory will not be usable.

5.4.3 Invalid Memory Accesses

If the PAC55XX attempts to make an invalid memory access, the memory controller will set the **MEMSTATUS.INVADDR** interrupt flag to a 1b. If the **MEMCTL.INVADDRIE** is set to 1b when this happens, then the memory controller will assert the IRQ_MEMCTL interrupt signal to the NVIC.

The **MEMSTATUS.INVADDR** interrupt flag may be cleared by writing it to 1b.

Note that any time the user wants to write the **MEMCTL** register, they must first write the **FLASHLOCK** with the value D513 B490h.

5.4.4 SRAM

The PAC55XX contains 32kB of SRAM. SRAM may be accessed through AHB transactions at a rate of up to 150MHz. The memory controller supports 32-bit word aligned 32-bit, 16-bit and 8-bit read write transactions to SRAM.

The MCU may read and write data or execute code from SRAM.

² The MCLK frequency range may be from 14.4MHz – 17.6MHz for FLASH read and 27MHz – 33MHz for FLASH write, so this frequency is in the acceptable range.

The PAC55XX SRAM contains support for Single Error Correction and Dual Error Detection (SECDED) for enhanced code and data integrity checking. Each word of SRAM contains additional 7-bits for SECDED error checking.

For word that is read from SRAM, the memory controller calculates the SECDED value and compares it to the 7-bits stored with that word in SRAM to verify that it is correct. For any word that is written to SRAM, the memory controller calculates the SECDED value and stores it with that word in SRAM.

The SECDED checking may be enabled by setting **MEMCTL.ECCDIS** to 0b and disabled by setting **MEMCTL.ECCDIS** to 1b. By default, SECDED is disabled after reset.

Single-bit errors may be detected and corrected by the memory controller. If a single-bit error is detected, then the memory controller will set the **MEMSTATUS.SE** bit to 1b. If the **MEMCTL.SEIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC will be asserted. The user may clear **MEMSTATUS.SE** by writing a 1b to it.

Dual-bit errors are able to be detected, but not able to be corrected by the memory controller. If a dual-bit error is detected, the memory controller will set the **MEMSTATUS.DE** bit to 1b and assert the IRQ_MEMCTL signal to the NVIC. This interrupt signal is a NMI.

Note that any time the user wants to write the **MEMCTL** register, they must first write the **FLASHLOCK** with the value D513 B490h.

5.4.5 FLASH Read Cache

The maximum HCLK frequency in the PAC55XX is 150MHz and the maximum FLASH frequency is 25MHz. To help provide faster memory access, the memory controller provides a small cache for reading instructions to provide a faster instruction read frequency.

The cache's performance is application-dependent. In general, the cache performs better if less jump or branch instructions are performed.

If the FLASH read execution speed is not fast enough, then the user has the option of running parts of their program from SRAM, where it can execute up to 150MHz.

The FLASH read cache may be enabled by setting **MEMCTL.CACHEDIS** to 0b and disabled by setting **MEMCTL.CACHEDIS** to 1b. By default, the cache is disabled after a reset.

5.4.6 Erasing a FLASH Page

To erase one page of FLASH, the user must follow these steps:

- Write **FLASHLOCK** to the value 43DF 140Ah to allow a FLASH page erase operation
- Write the FLASH page number to the **FLASHPAGE** register
- Write **FLASHERASE** to the value 8C79 9CA7h

The memory controller will set **MEMSTATUS.EBUSY** to a 1b while the erase operation is in progress. This bit will be cleared as soon as the FLASH erase operation is complete.

5.4.7 Erase all FLASH Pages

To erase all pages of FLASH, the user must follow these steps:

- Write **FLASHLOCK** to the value 43DF 140Ah to allow FLASH all page erase operation
- Write **FLASHERASE** to the value 09EE 76C9h

The memory controller will set **MEMSTATUS.EBUSY** to a 1b while the erase operation is in progress. This bit will be cleared as soon as the FLASH erase operation is complete.

5.4.8 Erase INFO-3

To erase the INFO-3 FLASH, the user must follow the following steps:

- Write **FLASHLOCK** to the value 43DF 140Ah to allow FLASH all page erase operation
- Write **FLASEIERASE** to the value 1266 FF45h

The memory controller will set **MEMSTATUS.EBUSY** to a 1b while the erase operation is in progress. This bit will be cleared as soon as the FLASH erase operation is complete.

5.4.9 Writing FLASH

Once FLASH memory is erased, FLASH may be written using the AHB bus 32-bits a time through write commands. Before writing FLASH to an erased memory, the user should set the **FLASHLOCK** register to a value of 43DF 140Ah.

5.5 Code Protection

The PAC55XX Memory Controller provides a programmable Code Protection function that allows the user to configure several of levels of security depending on the application needs.

5.5.1 Code Protection Variables

Code Protection is configured by variables in INFO-1 and INFO-2 that are used to control the behavior and security of the various Code Protection levels.

INFO2.SECEN – Enables Code Protection

INFO2.SWDFUSE – Fuse to permanently disable SWD/JTAG access to the device

INFO-3 contains additional variables as shown below.

Table 5-4 INFO-3 Format for Code Protection

| ADDRESS | BYTE OFFSET | | | |
|----------------------------|-------------|---|----------|-------|
| | 3 | 2 | 1 | 0 |
| 0100 0800h | RMASK | | | |
| 0100 0804h | | | | |
| 0100 0808h | | | | |
| 0100 080Ch | | | | |
| 0100 0810h – 0100 081F | Reserved | | | |
| 0100 0820h | WMASK | | | |
| 0100 0824h | | | | |
| 0100 0828h | | | | |
| 0100 082Ch | | | | |
| 0100 0830h – 0100 083Ch | Reserved | | | |
| 0100 0840h | Reserved | | SECLEVEL | IMASK |

INFO3.RMASK is a 128-bit, bit-mask for each of the 128 segments of FLASH memory. The LSB of 0100 0800h represents segment 0 and the MSB of 0100 080Fh represents segment 127. This mask specifies which segments the application allows for read-access when in levels 1, 2 and 3 of code protection as described below.

INFO3.WMASK is a 128-bit, bit-mask for each of the 128 segments of FLASH memory. The LSB of 0100 0820h represents segment 0 and the MSB of 0100 082Fh represents segment 127. This mask specifies which segments the application allows for write-access when in levels 1, 2 and 3 of code protection as described below.

INFO3.IMASK[1] specifies the read access for INFO-3.

INFO3.IMASK[3] specifies the write access for INFO-3.

INFO2.SECLEVEL specifies the code protection security level as described below.

Table 5-5 Code Protection Levels

| LEVEL | INFO3.SECLEVEL | INFO2.SWDFUSE | NAME | DESCRIPTION |
|-------|----------------|-----------------------------|-----------------------|--|
| 0 | FFh | Any value except DEAD DEADh | UNLOCKED | <ul style="list-style-type: none"> No restrictions |
| 1 | FEh | Any value except DEAD DEADh | RW PROTECTION | <ul style="list-style-type: none"> SWD/JTAG Enabled No FLASH erase operations allowed If in ARM debug mode, restrict access to FLASH, according to INFO3.RMASK and INFO3.WMASK If in ARM debug mode, restrict access to INFO-2, according to INFO3.IMASK |
| 2 | FCh | Any value except DEAD DEADh | SWD DISABLED | <ul style="list-style-type: none"> SWD/JTAG Disabled No FLASH erase operations allowed If in ARM debug mode, restrict access to FLASH, according to INFO3.RMASK and INFO3.WMASK If in ARM debug mode, restrict access to INFO-2, according to INFO3.IMASK May re-enable SWD/JTAG temporarily |
| 3 | n/a | DEAD DEADh | PERMANENT SWD DISABLE | <ul style="list-style-type: none"> SWD/JTAG Permanently Disabled No FLASH erase operations allowed If in ARM debug mode, restrict access to FLASH, according to INFO3.RMASK and INFO3.WMASK If in ARM debug mode, restrict access to INFO-2, according to INFO3.IMASK |

When using code protection, the INFO-3 memory is used by the Code Protection feature and is not usable by the application. The Memory Controller may erase the contents of INFO-3 when the security levels are changed. The table below shows the format of INFO-3 when used with code protection enabled.

When code protection is enabled, segments that are marked as read-only will also be remapped to 0020 0000h from 0000 0000h.

5.5.2 Code Security Levels

To enable code security, first set the **INFO2.SECEN** to a value other than 0xFF.

*NOTE: When writing the **INFO2.SECEN** it may not be undone and code security will be enabled from that point on.*

After enabling code security, write **INFO3.SECLEVEL** or **INFO2.SWDFUSE** to the correct value to enable the desired security level.

5.5.3 Level 0 – UNLOCKED

When in level 0, Code Protection is in the *UNLOCKED* state.

In this state, there are no restrictions on read or writing transactions to FLASH memory.

5.5.4 Level 1 – RW PROTECTION

When in level 1, Code Protection is in the *RW PROTECTION* state.

In this state, SWD/JTAG remains enabled. All FLASH erase operations by the user are disabled.

After SWD/JTAG has been initialized, the Memory Controller restricts access to FLASH memory according to the **INFO3.RMASK** and **INFO3.WMASK** fields.

When the Memory Controller receives a read request for FLASH memory, if the corresponding bit for the segment of FLASH is cleared in **INFO3.RMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for FLASH memory, if the corresponding bit for the segment of FLASH is cleared in **INFO3.WMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-2, if the **INFO3.IMASK[0]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-3, if the **INFO3.IMASK[1]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-2, if the **INFO3.IMASK[2]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-3, if the **INFO3.IMASK[3]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

To revert to level 0, the user must perform a mass program and INFO-3 erase. This will erase program FLASH followed by INFO-3 FLASH.

5.5.5 Level 2 – SWD DISABLED

When in level 1, Code Protection is in the *SWD DISABLED* state.

In this state, SWD/JTAG access is disabled. All FLASH erase operations by the user are disabled.

When the ARM Cortex MCU is in debug mode, the Memory Controller restricts access to FLASH memory according to the **INFO3.RMASK** and **INFO3.WMASK** fields.

When the Memory Controller receives a read request for FLASH memory, if the corresponding bit for the segment of FLASH is not cleared in **INFO3.RMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for FLASH memory, if the corresponding bit for the segment of FLASH is not cleared in **INFO3.WMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-2, if the **INFO3.IMASK[0]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-3, if the **INFO3.IMASK[1]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-2, if the **INFO3.IMASK[2]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-3, if the **INFO3.IMASK[3]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

To revert to level 0, the user must perform a mass program and INFO-3 erase. This will erase program FLASH followed by INFO-3 FLASH.

The SWD/JTAG may be temporarily unlocked in this mode. To temporarily unlock SWD/JTAG until the device is reset, the user should write **SWDUNLOCK** with the same value as previously written.

5.5.6 Level 3 – PERMANENT SWD DISABLED

When in level 1, Code Protection is in the *PERMANENT SWD DISABLED* state. SWD and JTAG may not be enabled again.

In this state, SWD/JTAG access is disabled. All FLASH erase operations by the user are disabled.

When the ARM Cortex MCU is in debug mode, the Memory Controller restricts access to FLASH memory according to the **INFO3.RMASK** and **INFO3.WMASK** fields.

When the Memory Controller receives a read request for FLASH memory, if the corresponding bit for the segment of FLASH is not cleared in **INFO3.RMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for FLASH memory, if the corresponding bit for the segment of FLASH is not cleared in **INFO3.WMASK**, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-2, if the **INFO3.IMASK[0]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a read request for INFO-3, if the **INFO3.IMASK[1]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-2, if the **INFO3.IMASK[2]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

When the Memory Controller receives a write request for INFO-3, if the **INFO3.IMASK[3]** is not cleared, then the transaction is denied. The **MEMSTATUS.INVADDR** bit is set to 1b. If the **MEMCTL.INVADDRIE** bit is set to 1b, then the IRQ_MEMCTL signal to the NVIC is asserted.

To revert to level 0, the user must perform a mass program and INFO-3 erase. This will erase program FLASH followed by INFO-3 FLASH.

5.6 Register Summary

Table 5-6 FLASH Controller Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|---------------------------------|--------|------------|
| MEMCTL | 400D 0000h | Memory Controller Configuration | RW | 0032 0000h |
| MEMSTATUS | 400D 0004h | Memory Controller Status | RW | 0000 0000h |
| FLASHLOCK | 400D 0008h | FLASH Lock Access | RW | 0000 0000h |
| FLASHPAGE | 400D 000Ch | FLASH Page | RW | 0000 0000h |
| SWDUNLOCK | 400D 0010h | SWD Unlock | RW | 0000 0000h |
| FLASHERASE | 400D 0020h | FLASH Erase | RW | 0000 0000h |

5.7 Register Detail

5.7.1 MEMCTL

Register 5-1 MEMCTL (Memory Controller Configuration, 400D 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|---|
| 31:23 | Reserved | RO | 0 | Reserved |
| 22 | MCLKSEL | RW | 0 | MCLK MUX Select: 0b: ROSCCLK 1b: MCLK |
| 21 | CACHEDIS | RW | 1 | FLASH Read Cache Disable: 0b: Read Cache Enabled 1b: Read Cache Disabled |
| 20 | ECCDIS | RW | 1 | SECEDED (ECC) Disable: 0b: SECEDED Enabled 1b: SECEDED Disabled |
| 19 | STDBY | RW | 0 | FLASH Standby Mode: 0b: normal mode 1b: standby mode |
| 18 | INVADDRIE | RW | 0 | Invalid Memory Access Interrupt Enable: 0b: disabled 1b: enabled |
| 17 | DEIE | RW | 1 | Dual Bit Detection Interrupt Enable (ECC): 0b: disabled 1b: enabled |
| 16 | SEIE | RW | 0 | Single Bit Detection Interrupt Enable (ECC): 0b: disabled 1b: enabled |
| 15:10 | Reserved | RO | 0 | Reserved |
| 9:8 | WRITEWORDCNT | RW | 0 | Write Data Counter: Write to 0 to reset the write data buffer for FLASH. |
| 7:4 | MCLKDIV | RW | 0000b | MCLK Divider for FLASH controller: 0000b: /1 0001b: /2 0010b: /3 0011b: /4 0100b: /5 0101b: /6 0110b: /7 0111b: /8 1000b: /9 1001b: /10 1010b: /11 1011b: /12 1100b: /13 1101b: /14 |

| | | | | |
|-----|---------------|----|-------|--|
| | | | | 1110b: /15 1111b: /16 |
| 3:0 | WSTATE | RW | 0000b | FLASH Read Wait States: 0000b: 0 wait states 0001b: 1 wait state 0010b: 2 wait states 0011b: 3 wait states 0100b: 4 wait states 0101b: 5 wait states 0110b: 6 wait states 0111b: 7 wait states 1000b: 8 wait states 1001b: 9 wait states 1010b: 10 wait states 1011b: 11 wait states 1100b: 12 wait states 1101b: 13 wait states 1110b: 14 wait states 1111b: 15 wait states |

5.7.2 MEMSTATUS

Register 5-2 MEMSTATUS (Memory Controller Status, 400D 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|---|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | INVADDR | W1C | 0 | Invalid Address Fetch Flag: 0b: no flag 1b: flag |
| 17 | DE | W1C | 0 | Double-bit Detection Flag (ECC): 0b: no flag 1b: flag |
| 16 | SE | W1C | 0 | Single-bit Detection Flag (ECC): 0b: no flag 1b: flag |
| 15:10 | Reserved | RO | 0 | Reserved |
| 9:8 | WRITEWORDCNT | RO | 0 | Number of bytes written to FLASH for the write data buffer: 00b: 4 bytes 01b: 8 bytes 10b: 12 bytes 11b: 16 bytes |
| 7:2 | Reserved | RO | 0 | Reserved |
| 1 | EBUSY | RO | 0 | Erase Busy: 0b: Erase operation not in progress 1b: Erase operation in progress |
| 0 | WBUSY | RO | 0 | Write Busy: 0b: Write operation not in progress 1b: Write operation in progress |

5.7.3 FLASHLOCK

Register 5-3 FLASHLOCK (Memory Controller Status, 400D 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|---|
| 31:0 | FLASHLOCK | RW | 0 | FLASHLOCK value for Memory Controller Operations: 43DF 140Ah: Allow write an erase operations to FLASH D513 B490h: Allow write to MEMCTL register 5F72 9205h: Allow write access to INFO-1 to disable SWD |

5.7.4 FLASHPAGE

Register 5-4 FLASHPAGE (FLASH Page, 400D 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6:0 | PAGE | RW | 0 | FLASH Page Selection: 000 0000b: Page 0, 0000 0000h to 0000 03FFh 000 0001b: Page 1, 0000 0400h to 0000 0800h 000 0010b: Page 2, 0000 0800h to 0000 01BFFh ... 111 1101b: Page 125, 0001 F400h to 0001 F7FFh 111 1110b: Page 126, 0001 F800h to 0001 FBFFh 111 1111b: Page 127, 0001 FC00h to 0001 FFFFh |

5.7.5 SWDUNLOCK

Register 5-5 SWDUNLOCK (SWD Unlock, 400D 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------------------|--------|-------|---|
| 31:0 | KEY_SWDUNLOCK | WO | 0 | Writing the last written value to this register will unlock the SWD port. |

5.7.6 FLASHERASE

Register 5-6 FLASHERASE (FLASH Erase, 400D 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------|--------|-------|--|
| 31:0 | KEY | RW | 0 | Writing the value to this register will start the given erase operation: Page Erase: 8C79 9CA7h Mass Page Erase: 09EE 76C9h INFO-3 Erase: 1266 FF45h Reading this register will always return 0. |

6 DPM

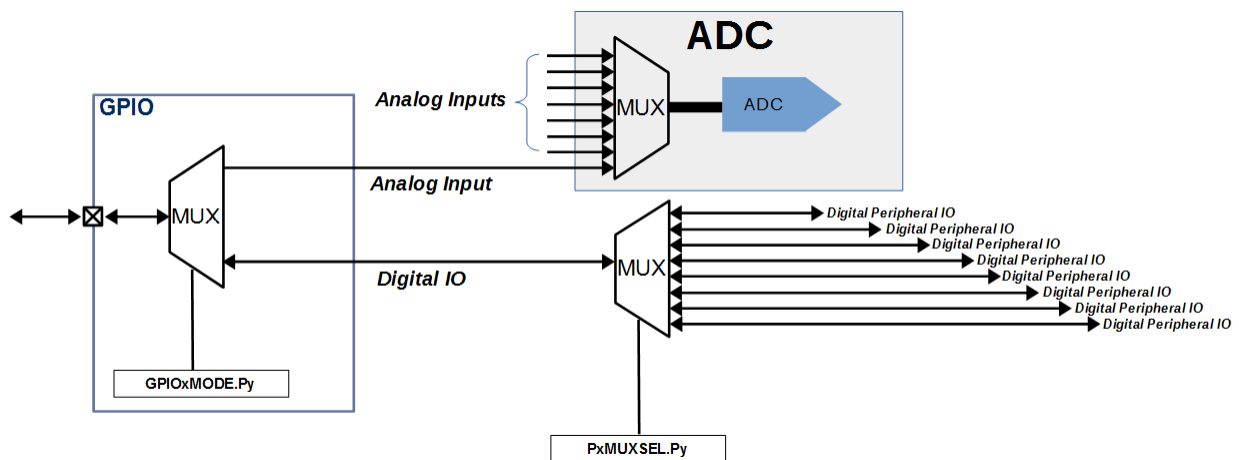
6.1 Overview

The Digital Peripheral MUX (DPM) on the PAC55XX allows a connection of its many different digital peripherals to IO pins. Each digital peripheral has several digital signals that are used in its operation. For example, a USART in SPI mode will have a SPI clock, slave select, input and output data.

In order to support the most flexible assignment of digital peripheral IOs to device pins, the PAC55XX has a highly flexible Digital Peripheral MUX. The Digital Peripheral MUX allows the assignment of peripheral functions to device pins.

Each GPIO in the PAC55XX has the capability to perform digital IO and some GPIO have the ability to act as an Analog Input for the ADC. See the diagram below for an example.

Figure 6-1 GPIO and Peripheral MUX Block Diagram



The GPIO can select if it acts as a Digital IO or Analog Input by the **GPIOxMODE.Py** register.

If **GPIOxMODE.Py** = 00b (Analog Input), then the GPIO acts as an analog input and the signal is connected to the analog MUX in the ADC.

If the **GPIOxMODE.Py** is not 00b (High-Impedance Digital Input, Push-pull Output, Open-drain Output) then the GPIO acts as a digital input and connects the signal to the Digital Peripheral MUX. This MUX can select which peripheral function is connected to the GPIO by the **PxMUXS.Py** register.

The following section shows the set of peripheral functions that are available in the Digital Peripheral MUX.

6.2 Digital Peripheral Functions

There are many digital peripherals in the PAC55XX. The sections below discuss each peripheral, and what digital IO signals are used in that peripheral.

For certain peripherals, the peripheral signals can be discretely assigned to the IO (like PWM timers). For other peripheral functions, the entire set needs to be assigned for the peripheral function to work (like a communication module).

Table 6-1 Peripheral Functions

| PERIPHERAL GROUP | SIGNAL NAME | DESCRIPTION |
|------------------|----------------|--------------------------------------|
| SYSTEM | nRESET | Active-low reset signal for MCU |
| | FRCLK | Free-running clock (output from CCS) |
| | EXTCLK | External Clock Input (input to CCS) |
| DEBUG | TMS/SWDIO | JTAG TMS or SWD IO (Debug Port) |
| | TCK/SWDCLK | JTAG clock or SWD clock (Debug Port) |
| | TDI | JTAG data input (Debug Port) |
| | TDO | JTAG data output (Debug Port) |
| ETM | TRACECLK | ETM Instruction Trace Clock (ETM) |
| | TRACED0 | ETM Instruction Trace Data 0 (ETM) |
| | TRACED1 | ETM Instruction Trace Data 1 (ETM) |
| | TRACED2 | ETM Instruction Trace Data 2 (ETM) |
| | TRACED3 | ETM Instruction Trace Data 3 (ETM) |
| EMUX | EMUXC | EMUX Clock (ADC DTSE) |
| | EMUXD | EMUX Data (ADC DTSE) |
| I2C | I2CSCL | I2C Serial Clock (I2C) |
| | I2CSDA | I2C Serial Data (I2C) |
| CAN | CANTXD | CAN controller TX data (CAN) |
| | CANRXD | CAN controller RX data (CAN) |
| USARTA | USACLK | USARTA Clock |
| | USAMOSI/UARCTX | USARTA SPI MOSI, UART TX |
| | USAMISO/UARTRX | USARTA SPI MISO, UART RX |
| | USASCLK | USARTA SPI SCLK |
| | USASS | USARTA SPI Slave Select |
| USARTB | USBCLK | USARTB Clock |
| | USBMOSI/UARCTX | USARTB SPI MOSI, UART TX |
| | USBMISO/UARTRX | USARTB SPI MISO, UART RX |
| | USBCLK | USARTB SPI SCLK |

| | | |
|---------------|----------------|---------------------------|
| | USBSS | USARTB SPI Slave Select |
| USARTC | USCCLK | USARTC Clock |
| | USCMOSI/UARTTX | USARTC SPI MOSI, UART TX |
| | USCMISO/UARTRX | USARTC SPI MISO, UART RX |
| | USCSCLK | USARTC SPI SCLK |
| | USCSS | USARTC SPI Slave Select |
| | | |
| USARTD | USDCLK | USARTD Clock |
| | USDMOSI/UARTTX | USARTD SPI MOSI, UART TX |
| | USDMISO/UARTRX | USARTD SPI MISO, UART RX |
| | USDCLK | USARTD SPI SCLK |
| | USDSS | USARTD SPI Slave Select |
| TIMERA | TAPWM0 | Timer A PWM output/CCR0 |
| | TAPWM1 | Timer A PWM output/CCR1 |
| | TAPWM2 | Timer A PWM output/CCR2 |
| | TAPWM3 | Timer A PWM output/CCR3 |
| | TAPWM4 | Timer A PWM output/CCR4 |
| | TAPWM5 | Timer A PWM output/CCR5 |
| | TAPWM6 | Timer A PWM output/CCR6 |
| | TAPWM7 | Timer A PWM output/CCR7 |
| | TAQEPPHA | Timer A QEP Phase A Input |
| | TAQEPPHB | Timer A QEP Phase B Input |
| | TAQEPIDX | Timer A QEP Index Input |
| TIMERB | TBPWM0 | Timer B PWM output/CCR0 |
| | TBPWM1 | Timer B PWM output/CCR1 |
| | TBPWM2 | Timer B PWM output/CCR2 |
| | TBPWM3 | Timer B PWM output/CCR3 |
| | TBPWM4 | Timer B PWM output/CCR4 |
| | TBPWM5 | Timer B PWM output/CCR5 |
| | TBPWM6 | Timer B PWM output/CCR6 |
| | TBPWM7 | Timer B PWM output/CCR7 |
| | TBQEPPHA | Timer B QEP Phase A Input |
| | TBQEPPHB | Timer B QEP Phase B Input |
| | TBQEPIDX | Timer B QEP Index Input |
| TIMERC | TCPWM0 | Timer C PWM output/CCR0 |
| | TCPWM1 | Timer C PWM output/CCR1 |
| | TCPWM2 | Timer C PWM output/CCR2 |

| | | |
|---------------|----------|---------------------------|
| | TCPWM3 | Timer C PWM output/CCR3 |
| | TCPWM4 | Timer C PWM output/CCR4 |
| | TCPWM5 | Timer C PWM output/CCR5 |
| | TCPWM6 | Timer C PWM output/CCR6 |
| | TCPWM7 | Timer C PWM output/CCR7 |
| | TCQEPPHA | Timer C QEP Phase A Input |
| | TCQEPPHB | Timer C QEP Phase B Input |
| | TCQEPIDX | Timer C QEP Index Input |
| TIMERD | TDPWM0 | Timer D PWM output/CCR0 |
| | TDPWM1 | Timer D PWM output/CCR1 |
| | TDPWM2 | Timer D PWM output/CCR2 |
| | TDPWM3 | Timer D PWM output/CCR3 |
| | TDPWM4 | Timer D PWM output/CCR4 |
| | TDPWM5 | Timer D PWM output/CCR5 |
| | TDPWM6 | Timer D PWM output/CCR6 |
| | TDPWM7 | Timer D PWM output/CCR7 |
| | TDQEPPHA | Timer D QEP Phase A Input |
| | TDQEPPHB | Timer D QEP Phase B Input |
| | TDQEPIDX | Timer D QEP Index Input |

6.3 Peripheral MUX Settings

The table below shows the Digital Peripheral MUX Settings for each GPIO in the PAC55XX according to the setting of the **PxMUXSEL** register for the given GPIO.

Items in the table that are in boldface type are the default values.

Table 6-2 Peripheral MUX Settings

| PORT | Pin | PxMUXSEL.Py | | | | | | | |
|------|-----|---------------|----------------|---------|----------|------------|---------|--------|---------|
| | | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
| PA | P0 | GPIOA0 | | | | | | | |
| | P1 | GPIOA1 | EMUXD | | | | | | |
| | P2 | GPIOA2 | EMUXC | | | | | | |
| | P3 | GPIOA3 | USASCLK | USBSCLK | | | | | |
| | P4 | GPIOA4 | USAMOSI | USBMOSI | | | | | |
| | P5 | GPIOA5 | USAMISO | USBMISO | | | | | |
| | P6 | GPIOA6 | USASS | USBSS | | | | | |
| | P7 | GPIOA7 | | | | | | | |
| PB | P0 | GPIOB0 | TAPWM0 | TBPWM0 | | TCPWM0 | TDPWM0 | | |
| | P1 | GPIOB1 | TAPWM1 | TBPWM1 | | TCPWM1 | TDPWM1 | | |
| | P2 | GPIOB2 | TAPWM2 | TBPWM2 | | TCPWM2 | TDPWM2 | | |
| | P3 | GPIOB3 | TAPWM3 | TBPWM3 | | TCPWM3 | TDPWM3 | | |
| | P4 | GPIOB4 | TAPWM4 | TBPWM4 | TCPWM0 | TCPWM4 | TDPWM4 | | |
| | P5 | GPIOB5 | TAPWM5 | TBPWM5 | TCPWM1 | TCPWM5 | TDPWM5 | | |
| | P6 | GPIOB6 | TAPWM6 | TBPWM6 | TCPWM2 | TCPWM6 | TDPWM6 | | |
| | P7 | GPIOB7 | TAPWM7 | TBPWM7 | TCPWM3 | TCPWM7 | TDPWM7 | | |
| PC | P0 | GPIOC0 | TBPWM0 | TCPWM0 | TBQEPIDX | USBMOSI | USCCLK | CANRXD | I2CSCL |
| | P1 | GPIOC1 | TBPWM1 | TCPWM1 | TBQEPPHA | USBMISO | USCSS | CANTXD | I2CSDA |
| | P2 | GPIOC2 | TBPWM2 | TCPWM2 | TBQEPPHB | USBSCLK | USCMOSI | | EMUXD |
| | P3 | GPIOC3 | TBPWM3 | TCPWM3 | | USBSS | USCMISO | nRESET | EMUXC |
| | P4 | GPIOC4 | TBPWM4 | TCPWM4 | TCQEPIDX | USBMOSI | USCCLK | CANRXD | I2CSCL |
| | P5 | GPIOC5 | TBPWM5 | TCPWM5 | TCQEPPHA | USBMISO | USCSS | CANTXD | I2CSDA |
| | P6 | GPIOC6 | TBPWM6 | TCPWM6 | TCQEPPHB | USBSCLK | USCMOSI | nRESET | EMUXD |
| | P7 | GPIOC7 | TBPWM7 | TCPWM7 | | USBSS | USCMISO | FRCLK | EMUXC |
| PD | P0 | GIOD0 | TBPWM0 | TCPWM0 | TDQEPIDX | TCK/SWDCLK | USCCLK | CANTXD | EMUXD |
| | P1 | GIOD1 | TBPWM1 | TCPWM1 | TDQEPPHA | TMS/SWDIO | USCSS | CANRXD | EMUXC |
| | P2 | GIOD2 | TBPWM2 | TCPWM2 | TDQEPPHB | TDI | USCMOSI | nRESET | |
| | P3 | GIOD3 | TBPWM3 | TCPWM3 | | TDO | USCMISO | FRCLK | TRACED3 |

| | | | | | | | | | |
|----|----|---------------|--------|----------|-------------------|------------|---------|----------|------------|
| | P4 | GPIOD4 | TBPWM4 | TCPWM4 | TDQEPIDX | TBQEPIX | USDCLK | TRACED3 | USDMOSI |
| | P5 | GPIOD5 | TBPWM5 | TCPWM5 | TDQEPPHA | TBQEPPHA | USDSS | CANRXD | USDMISO |
| | P6 | GPIOD6 | TBPWM6 | TCPWM6 | TDQEPPHB | TBQEPPHB | USDMOSI | CANTXD | I2CSDA |
| | P7 | GPIOD7 | TBPWM7 | TCPWM7 | | | USDMISO | CANRXD | I2CSCL |
| PE | P0 | GPIOE0 | TCPWM4 | TDPWM0 | TAQEPIX | TBQEPIX | USCLK | I2CSCL | EMUXC |
| | P1 | GPIOE1 | TCPWM5 | TDPWM1 | TAQEPPHA | TBQEPPHA | USCSS | I2CSDA | EMUXD |
| | P2 | GPIOE2 | TCPWM6 | TDPWM2 | TAQEPPHB | TBQEPPHB | USCMOSI | CANRXD | EXTCLK |
| | P3 | GPIOE3 | TCPWM7 | TDPWM3 | FRCLK | nRESET | USCMISO | CANTXD | |
| | P4 | GPIOE4 | TCPWM0 | TDPWM4 | TDQEPIDX | USBCLK | USDMOSI | I2CSCL | TCK/SWDCLK |
| | P5 | GPIOE5 | TCPWM1 | TDPWM5 | TDQEPPHA | USBSS | USDMISO | I2CSDA | TMS/SWDIO |
| | P6 | GPIOE6 | TCPWM2 | TDPWM6 | TDQEPPHB | USBMOSI | USDCLK | CANRXD | TDI |
| | P7 | GPIOE7 | TCPWM3 | TDPWM7 | | USBMISO | USDSS | CANTXD | TDO |
| PF | P0 | GPIOF0 | TCPWM0 | TDPWM0 | TMS/SWDCLK | TBQEPIX | USBCLK | TRACED2 | TRACECLK |
| | P1 | GPIOF1 | TCPWM1 | TDPWM1 | TMS/SWDIO | TBQEPPHA | USBSS | TRACED1 | TRACED0 |
| | P2 | GPIOF2 | TCPWM2 | TDPWM2 | TDI | TBQEPPHB | USBMOSI | TRACED0 | TRACED1 |
| | P3 | GPIOF3 | TCPWM3 | TDPWM3 | TDO | FRCLK | USBMISO | TRACECLK | TRACED2 |
| | P4 | GPIOF4 | TCPWM4 | TDPWM4 | TCK/SWDCLK | TCQEPIX | USDCLK | TRACED3 | EMUXC |
| | P5 | GPIOF5 | TCPWM5 | TDPWM5 | TMS/SWDIO | TCQEPPHA | USDSS | | EMUXD |
| | P6 | GPIOF6 | TCPWM6 | TDPWM6 | TDI | TCQEPPHB | USDMOSI | CANRXD | I2CSCL |
| | P7 | GPIOF7 | TCPWM7 | TDPWM7 | TDO | nRESET | USDMISO | CANTXD | I2CSDA |
| PG | P0 | GPIOG0 | TCPWM0 | TDPWM0 | EMUXC | TCK/SWDCLK | USDCLK | TRACECLK | TCQEPIX |
| | P1 | GPIOG1 | TCPWM1 | TDPWM1 | EMUXD | TMS/SWDIO | USDSS | TRACED0 | TCQEPPHA |
| | P2 | GPIOG2 | TCPWM2 | TDPWM2 | FRCLK | TDI | USDMOSI | TRACED1 | TCQEPPHB |
| | P3 | GPIOG3 | TCPWM3 | TDPWM3 | nRESET | TDO | USDMISO | TRACED2 | |
| | P4 | GPIOG4 | TCPWM4 | TDPWM4 | EMUXD | I2CSCL | USDSS | TRACED3 | TDQEPIDX |
| | P5 | GPIOG5 | TCPWM5 | TDPWM5 | EMUXC | | USDMOSI | CANRXD | TDQEPPHA |
| | P6 | GPIOG6 | TCPWM6 | TDPWM6 | I2CSDA | | USDMISO | CANTXD | TDQEPPHB |
| | P7 | GPIOG7 | | TDQEPIDX | | | USDCLK | | |

7 WWDT

7.1 Overview

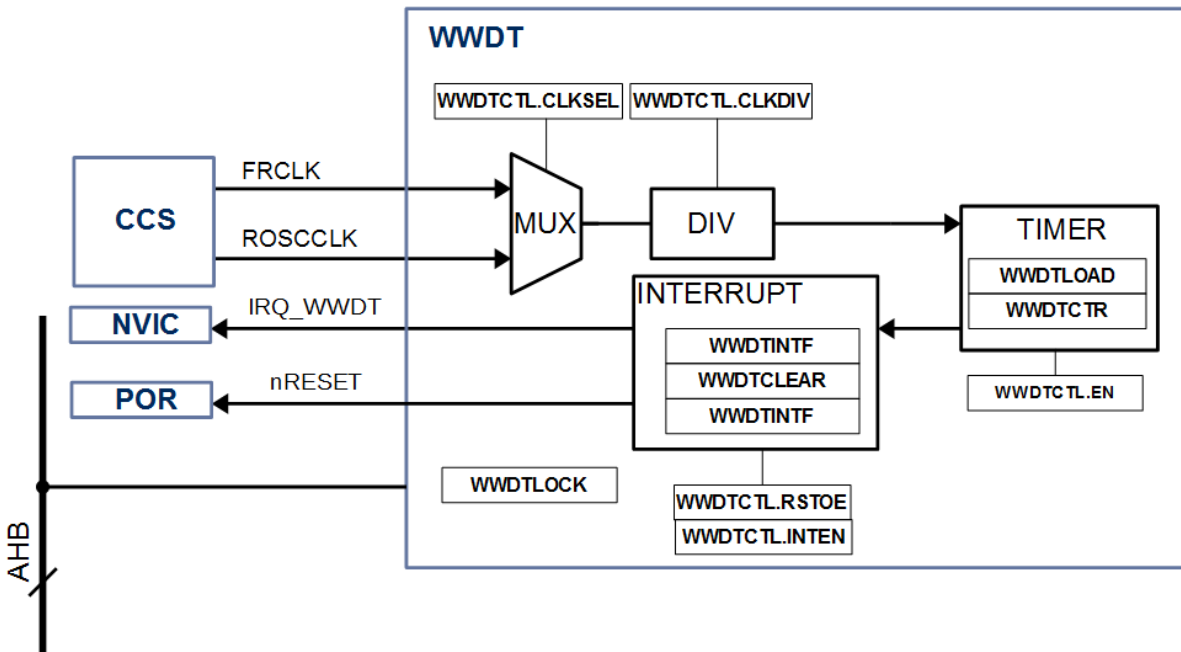
The PAC55XX contains a Windowed Watchdog Timer (WWDT) peripheral. The WWDT is an AHB client that allows the user to specify safety timeout by which the timer must be cleared, in order to avoid a POR.

7.2 Features

- 16-bit Count-down timer
- Input clock selection
- Input clock divider
- Interrupt or POR on expiration
- Register write protection

7.3 System Block Diagram

Figure 7-1 WWDT System Block Diagram



7.4 Functional Description

The WWDT is a 16-bit count-down timer. The WWDT is an AHB client and has a clock input of FRCLK or ROSCCLK. The input clock may be selected by the **WWDTCTL.CLKSEL** field.

The WWDT has an input clock divider that may be selected as shown below.

Table 7-1 WWDT Input Clock Divider

| WWDTCTL.CLKDIV | WWDT Clock |
|----------------|------------------------------|
| 0000b | WWDTCTL.CLKSEL /1 |
| 0001b | WWDTCTL.CLKSEL /2 |
| 0010b | WWDTCTL.CLKSEL /4 |
| 0011b | WWDTCTL.CLKSEL /8 |
| ... | ... |
| 1110b | WWDTCTL.CLKSEL /16384 |
| 1111b | WWDTCTL.CLKSEL /32768 |

If the user wants to write any register in the WWDT, **WWDTLOCK** must be set to 55AA 6699h.

If the user wants to read any register in the WWDT, **WWDTLOCK** must be set to 55AA 6698h.

When the timer is enabled by setting **WWDTCTL.EN** to 1b, the timer loads the value of **WWDTLOAD.CDV** into **WWDTCTR**. The timer will count down **WWDTCTR.VALUE**.

The user may clear or reset the timer at any time by writing **WWDTCLEAR** to any value. This will clear the **WWDTINTF.IF** flag and reloads the **WWDTLOAD.CDV** value into **WWDTCTR.VALUE**. If **WWDTCTR.VALUE** counts down to 0, then the **WWDTINTF.IF** is set to 1b and the IRQ_WWDT signal to the NVIC. If the **WWDTCTL.RSTEN** bit is set to 1b, then the nRESET signal is asserted, and the PAC55XX will perform a soft reset.

The WWDT makes sure the user only resets the timer by writing the **WWDTLOCK** register when the **WWDTCTR.VALUE** is < **WWDTLOAD.INTOFF**. If this is not true, then the interrupt and POR is generated as described above.

7.5 Register Summary

Table 7-2 WWDT Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|-------------------------|--------|------------|
| WWDTCTL | 400D 0800h | WWDT Control | RW | 0000 0000h |
| WWDTLOAD | 400D 0804h | WWDT Load Counter Value | RW | 0010 FFFFh |
| WWDTCTR | 400D 0808h | WWDT Counter | RO | 0000 FFFFh |
| WWDTINTF | 400D 080Ch | WWDT Interrupt Flag | RO | 0000 0000h |
| WWDTCLEAR | 400D 0810h | WWDT Clear | WO | -- |
| WWDTLOCK | 400D 0814h | WWDT Lock | RW | 0000 0000h |

7.6 Register Detail

7.6.1 WWDTCTL

Register 7-1 WWDTCTL (WWDT Control, 400D 0800h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:4 | CLKDIV | RW | 0 | WWDT Input Clock Divider: 0000b: WWDTCTL.CLKSEL /1 0001b: WWDTCTL.CLKSEL /2 0010b: WWDTCTL.CLKSEL /4 0011b: WWDTCTL.CLKSEL /8 0100b: WWDTCTL.CLKSEL /16 0101b: WWDTCTL.CLKSEL /32 0110b: WWDTCTL.CLKSEL /64 0110b: WWDTCTL.CLKSEL /128 0111b: WWDTCTL.CLKSEL /256 1000b: WWDTCTL.CLKSEL /512 1001b: WWDTCTL.CLKSEL /1024 1010b: WWDTCTL.CLKSEL /2048 1011b: WWDTCTL.CLKSEL /4096 1100b: WWDTCTL.CLKSEL /8192 1101b: WWDTCTL.CLKSEL /16384 1111b: WWDTCTL.CLKSEL /32768 |
| 3 | CLKSEL | RW | 0 | WWDT Input Clock Selection: 0b: FRCLK 1b: ROSCLK |
| 2 | RSTEN | RW | 0 | WWDT POR Enable: 0b: Do not assert nReset 1b: Assert nRESET on WWDTCTR = 0 |
| 1 | INTEN | RW | 0 | WWDT Interrupt Enable: 0b: no interrupt 1b: Assert IRQ_WWDT when WWDTINTF.IF = 1b |
| 0 | EN | RW | 0 | WWDT Enable: 0b: WWDT Disabled 1b: WWDT Enabled |

7.6.2 WWDTLOAD

Register 7-2 WWDTLOAD (WWDT Load Value, 400D 0804h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------|--------|-------|--|
| 31:16 | INTOFF | RW | 0010h | Value from which counter will decrement. |
| 15:0 | CDV | RW | FFFFh | WWDT Count-down Value. This value is loaded into WWDTCTR when the counter reaches a value of 0. When this register is written, the count |

| | | | | |
|--|--|--|--|--|
| | | | | is immediately restarted from the new value. The minimum valid value is 1. |
|--|--|--|--|--|

7.6.3 WWDTCTR

Register 7-3 WWDTCTR (WWDT Counter Value, 400D 0808h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | VALUE | RO | FFFFh | WWDT Counter Value. This is the current WWDT counter value. |

7.6.4 WWDTINTF

Register 7-4 WWDTINTF (WWDT Interrupt Flag, 400D 080Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:1 | Reserved | RO | 0 | Reserved |
| 0 | IF | RW | 0 | WWDT Interrupt Flag: 0b: no flag 1b: flag (write 1 to clear) |

7.6.5 WWDTCLEAR

Register 7-5 WWDTCLEAR (WWDT Interrupt Clear, 400D 0810h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|--------------|--------|-------|--|
| 31:0 | VALUE | WO | - | WWDT Interrupt Clear. A Write of any value to this register will clear the WWDTINTF registers. |

7.6.6 WWDTLOCK

Register 7-6 WWDTLOCK (WWDT Lock, 400D 0814h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|---------------|--------|-------|--|
| 31:1 | WDTKEY | RW | 0 | WWDT Lock. This register will allow all of the WWDT registers to be locked for writing. The possible values for this register are (all 32-bits): |

| | | | | |
|---|---------------|----|---|---|
| | | | | 55AA 6699h: All registers in the WWDT available for writing 55AA 6698h: All registers in the WWDT are read-only Any other values for this register will be ignored. |
| 0 | WDTREN | RW | 0 | Register write enable: 0b: Write access to all other registers disabled 1b: Write access to all other registers enabled |

8 RTC

8.1 Overview

The PAC55XX contains a Real-Time Clock (RTC) with Calendar and Alarm functions.

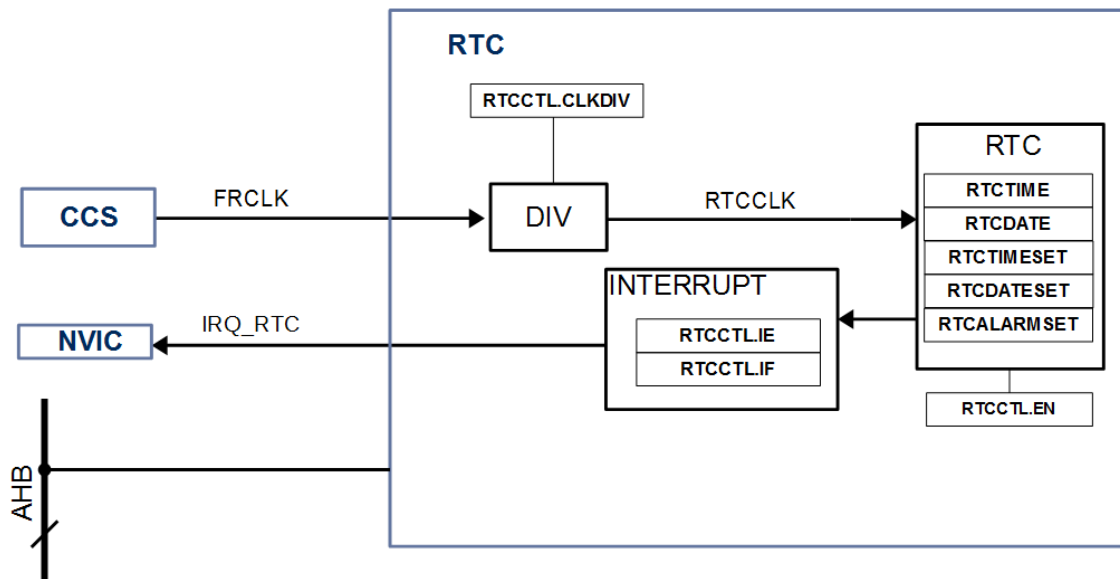
The RTC is an AHB client and is clocked using the FRCLK clock.

8.2 Features

- RTC with Calendar and Alarm Functions
- Interrupt on matching years, months, weeks, days, hours, minutes, seconds
- Enable/Disable to save power
- Input clock divider

8.3 System Block Diagram

Figure 8-1 RTC System Block Diagram



8.4 Functional Description

8.4.1 Overview

The Real-Time Clock (RTC) is a timer peripheral that allows the user to program an alarm time in the future using years, months, weeks, days, hours, minutes and seconds. The user can program the RTC to interrupt the MCU when an alarm time is reached.

The RTC is clocked by the FRCLK system clock. The RTC is intended to work with a 1MHz clock source, so the user must set the **RTCCTL.CLKDIV** field with the proper divider to generate a 1MHz RTCCLK from FRCLK. This will insure that this will count the real-time properly.

The RTC can be enabled writing setting **RTCCTL.EN** to 1b. It can be disabled by setting this bit to 0b.

The user may read the current time and date from the **RTCTIME** and **RTCDATE** read-only registers at any time.

To set the current time and date, the user may write the **RTCTIMESET** and **RTCDATESET** registers with the current date and time. To copy the time and date to the internal registers used for keeping the time, set the **RTCCTL.SETCAL** to 1b. In order for this to work, the **RTCCTL.EN** bit must already be set to 1b. The **RTCCTL.SETCAL** is always read as a 0b.

The user may set an alarm based on days, week day, hours or minutes by using the **RTCALARMSET** register.

To match the day alarm setting, set the **RTCALARMSET.DAYALARM** to the day number and **RTCALARMSET.DAYALARMEN** to 1b.

To match the week day alarm setting, set the **RTCALARMSET.DAYOFWEEKALARM** to the week day number and **RTCALARMSET.DAYOFWEEKALARMEN** to 1b.

To match the hour alarm setting, set the **RTCALARMSET.HOURALARM** to the hour number and **RTCALARMSET.HOURALARMEN** to 1b.

To match the minute setting, set the **RTCALARMSET.MINUTEALARM** to the minute number and **RTCALARMSET.MINUTEALARMEN** to 1b.

If the alarm matches the time, **RTCCTL.IF** will be set to a 1b. If the **RTCCTL.IE** is set to 1b, then the IRQ_RTC signal will be asserted to the NVIC.

8.5 Register Summary

Table 8-1 RTC Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|--------------------|------------|-------------------|--------|------------|
| RTCCTL | 400D 0C00h | RTC Control | RW | 0000 0010h |
| RTCTIME | 400D 0C04h | RTC Time | RO | 0000 0000h |
| RTCDATE | 400D 0C08h | RTC Date | RO | 0001 0100h |
| RTCTIMESET | 400D 0C0Ch | RTC Time Setting | RW | 0000 0000h |
| RTCDATESET | 400D 0C10h | RTC Date Setting | RW | 0001 0100h |
| RTCALARMSET | 400D 0C14h | RTC Alarm Setting | RW | 0000 0000h |

8.6 Register Detail

8.6.1 RTCCTL

Register 8-1 RTCCTL (RTC Control, 400D 0C00h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|---------|---|
| 31 | SETCAL | RW | 0 | When RTCCTL.EN is a 1b, writing 1b to this bit sets the internal TIME/DATA registers to the value of the TIMESSET and RTCDATESET registers. RTCCTL.EN must be a 1b to write this bit. When read, this bit will always return 0b. |
| 30:9 | Reserved | RO | 0 | Reserved |
| 8:4 | CLKDIV | RW | 0 0001b | <p>Clock divider for the RTC input clock (FRCLK). The user should set this divider so that the output of the divider generates a 1MHz clock, in order for the RTC to correctly count the real-time.</p> <p>0 0000b: FRCLK /1 0 0001b: FRCLK /2 0 0010b: FRCLK /3 ... 1 1110b: FRCLK /31 1 1111b: FRCLK /32</p> |
| 3 | Reserved | RO | 0 | Reserved |
| 2 | IF | W1C | 0b | <p>Alarm match interrupt flag:</p> <p>0b: no alarm 1b: alarm</p> |
| 1 | IE | RW | 0b | <p>Alarm match interrupt enable:</p> <p>0b: not enabled 1b: enabled</p> |
| 0 | EN | RW | 0b | <p>RTC Enabled:</p> <p>0b: not enabled 1b: enabled</p> |

8.6.2 RTCTIME

Register 8-2 RTCTIME (RTC Time, 400D 0C04h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|-------------|
| 31:22 | Reserved | RO | 0 | Reserved |
| 21:16 | HOURS | RO | 0 | RTC Hours |
| 15 | Reserved | RO | 0 | Reserved |
| 14:8 | MINUTES | RO | 0 | RTC Minutes |
| 7 | Reserved | RO | 0 | Reserved |
| 6:0 | SECONDS | RO | 0 | RTC Seconds |

8.6.3 RTCDATE

Register 8-3 RTCDATE (RTC Date, 400D 0C08h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|-----------------|
| 31:24 | YEAR | RO | 0 | RTC Year |
| 23:21 | Reserved | RO | 0 | Reserved |
| 20:16 | MONTH | RO | 1 | RTC Month |
| 15:14 | Reserved | RO | 0 | Reserved |
| 13:8 | DAY | RO | 1 | RTC Day |
| 7:3 | Reserved | RO | 0 | Reserved |
| 2:0 | DAYOFWEEK | RO | 0 | RTC Day of Week |

8.6.4 RTCTIMESET

Register 8-4 RTCTIMESET (RTC Time Setting, 400D 0C0Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--------------------|
| 31:22 | Reserved | RO | 0 | Reserved |
| 21:16 | HOUR | RW | 0 | RTC Hour Setting |
| 15 | Reserved | RO | 0 | Reserved |
| 14:8 | MINUTE | RW | 0 | RTC Minute Setting |
| 7 | Reserved | RO | 0 | Reserved |
| 6:0 | SECOND | RW | 0 | RTC Second Setting |

8.6.5 RTCDATESET

Register 8-5 RTCDATESET (RTC Date Setting, 400D 0C10h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|-------------------------|
| 31:24 | YEAR | RW | 0 | RTC Year Setting |
| 23:21 | Reserved | RO | 0 | Reserved |
| 20:16 | MONTH | RW | 1 | RTC Month Setting |
| 15:14 | Reserved | RO | 0 | Reserved |
| 13:8 | DAY | RW | 1 | RTC Day Setting |
| 7:3 | Reserved | RO | 0 | Reserved |
| 2:0 | DAYOFWEEK | RW | 0 | RTC Day of Week Setting |

8.6.6 RTCALARMSET

Register 8-6 RTCALARMSET (RTC Alarm Setting, 400D 0C14h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|------------------|--------|-------|---|
| 31 | Reserved | RO | 0 | Reserved |
| 30 | DAYALARMEN | RW | 0 | Day Alarm Enabled: 0b: disabled 1b: enabled |
| 29:24 | DAYALARM | RW | 0 | Day Alarm Setting |
| 23:20 | Reserved | RO | 0 | Reserved |
| 19 | DAYOFWEEKALARMEN | RW | 0 | Day of Week Alarm Enabled: 0b: disabled 1b: enabled |
| 18:16 | DAYOFWEEKALARM | RW | 0 | Day of Week Alarm Setting |
| 15 | Reserved | RO | 0 | Reserved |
| 14 | HOURALARMEN | RW | 0 | Hour Alarm Enabled: 0b: disabled 1b: enabled |
| 13:8 | HOURALARM | RW | 0 | Hour Alarm Setting |
| 7 | MINUTEALARMEN | RW | 0 | Minute Alarm Enabled: 0b: disabled 1b: enabled |
| 6:0 | MINUTEALARM | RW | 0 | Minute Alarm Setting |

9 GPIOA

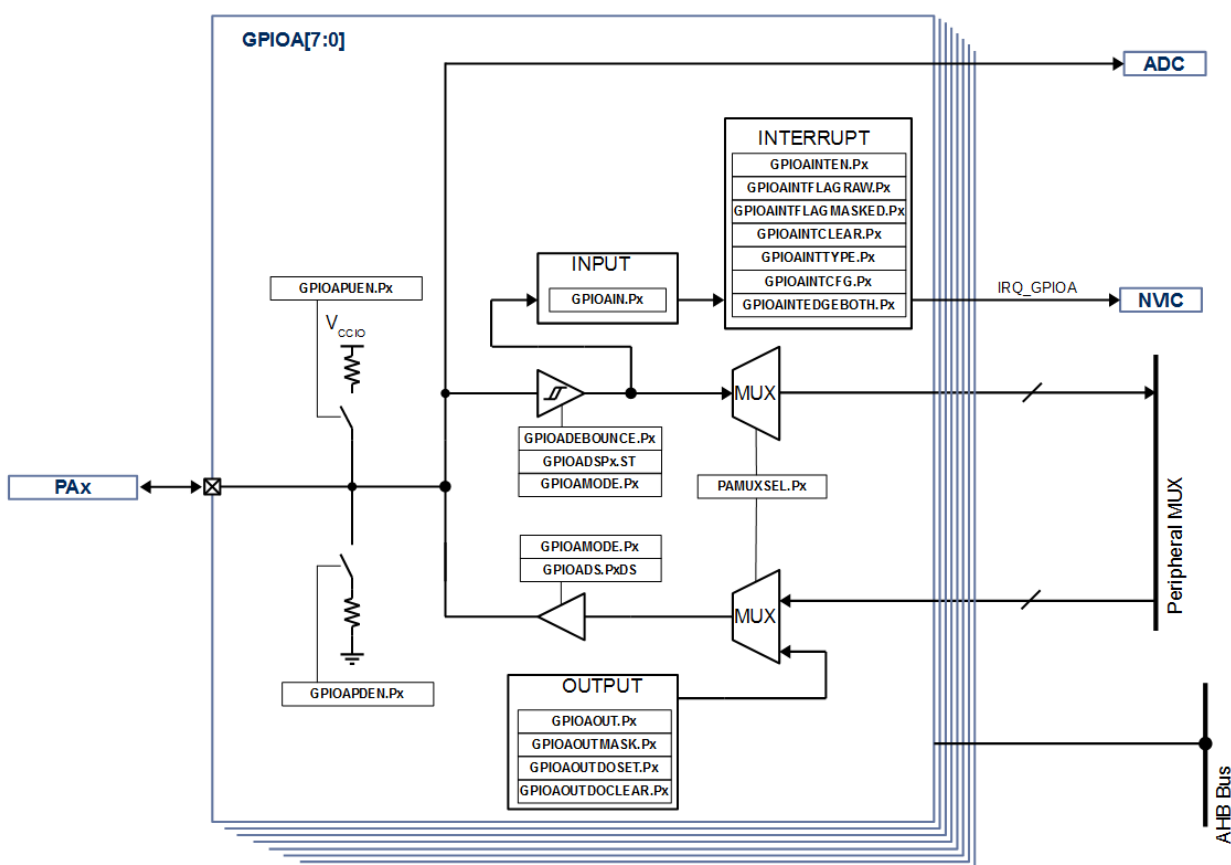
9.1 Overview

The GPIOA port contains 8 GPIO pin, GPIOA0 to GPIOA7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.³ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

9.2 GPIOA Block Diagram

Figure 9-1 GPIOA Block Diagram



³ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

9.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

9.4 Functional Description

9.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOAMODE** register as shown below.

Table 9-1 GPIOA Mode Configuration

| GPIOAMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Reserved |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

9.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOAMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOAOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOAOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOAOUTMASK.Px** is set to 1b, then any changes to **GPIOAOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOAIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

9.4.3 Digital Input

If **GPIOAMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOAIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

9.4.4 GPIOA Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOAMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 9-2 GPIOA Interrupt Configuration

| GPIOAINTTYPE | GPIOAINTCFG | GPIOAEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Level Logic Low |
| | 1b | X | Level Logic High |

Interrupts may be enabled or disabled using the **GPIOAINTEN** register. To enable interrupts on a pin, set the **GPIOAINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOAINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOAINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOAINTFLAGRAW.Px** is set to 1b and **GPIOAINTFLAGMASKED.Px** is set to 1b, then the **GPIOAINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOA signal to the NVIC is asserted.

9.4.5 GPIOA De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOAMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOAEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOAEBOUNCE.Px** to 0b.

9.4.6 GPIOA Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOA Drive strength.

Table 9-3 GPIOA Drive Strength

| PADS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |

| | |
|------|------|
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOA Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

9.4.7 GPIOA Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOAMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PADS.PxST** = 0b). To enable the input Schmitt Trigger, set **PADS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

9.4.8 GPIOA Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOAPU.Px** to 1b. To disable the pull-up, set **GPIOAPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOAPD.Px** to 1b. To disable the pull-down, set **GPIOAPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

9.5 Peripheral IO Mapping

The GPIOA peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 9-4 GPIOA Peripheral IO Mapping

| GPIOA SIGNAL | IO PIN |
|--------------|--------|
| GPIOA0 | PA0 |
| GPIOA1 | PA1 |
| GPIOA2 | PA2 |
| GPIOA3 | PA3 |
| GPIOA4 | PA4 |
| GPIOA5 | PA5 |
| GPIOA6 | PA6 |
| GPIOA7 | PA7 |

9.6 Register Summary

Table 9-5 GPIOA Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------------|------------|-------------------------------|--------|------------|
| GPIOAMODE | 400D 1400h | GPIOA Pin Mode Select | RW | 0000 FFFFh |
| GPIOAOUTMASK | 400D 1404h | GPIOA Data Output Write Mask | RW | 0000 0000h |
| GPIOAOUT | 400D 1408h | GPIOA Data Output Value | RW | 0000 00FFh |
| GPIOAIN | 400D 140Ch | GPIOA Data Input Value | RO | -- |
| GPIOAINTEN | 400D 1410h | GPIOA Interrupt Enable | RW | 0000 0000h |
| GPIOAINTFLAGRAW | 400D 1414h | GPIOA Interrupt Flag Raw | RW | 0000 0000h |
| GPIOAINTFMASKED | 400D 1418h | GPIOA Interrupt Flag Masked | RW | 0000 0000h |
| GPIOAINTCLEAR | 400D 141Ch | GPIOA Interrupt Clear | WO | 0000 0000h |
| GPIOAINTTYPE | 400D 1420h | GPIOA Interrupt Type | RW | 0000 0000h |
| GPIOAINTCFG | 400D 1424h | GPIOA Interrupt Configuration | RW | 0000 0000h |
| GPIOAINTEDGEBOTH | 400D 1428h | GPIOA Interrupt Edge Both | RW | 0000 0000h |
| GPIOADEBOUNCE | 400D 142Ch | GPIOA De-bounce Filter | RW | 0000 0000h |
| GPIOADOSET | 400D 1430h | GPIOA Data Output Set | WO | 0000 0000h |
| GPIOADOCLEAR | 400D 1434h | GPIOA Data Output Clear | WO | 0000 0000h |

9.7 Register Detail

9.7.1 GPIOAMODE

Register 9-1 GPIOAMODE (GPIOA Mode Configuration, 400D 1400h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

9.7.2 GPIOAOUTMASK

Register 9-2 GPIOAOUTMASK (GPIOA Output Mask, 400D 1404h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

9.7.3 GPIOAOUT

Register 9-3 GPIOAOUT (GPIOA Output Data Value, 400D 1408h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

9.7.4 GPIOAIN

Register 9-4 GPIOAIN (GPIOA Input Data Value, 400D 140Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

9.7.5 GPIOAINTEN

Register 9-5 GPIOAINTEN (GPIOA Interrupt Enable, 400D 1410h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

9.7.6 GPIOINTFLAGRAW

Register 9-6 GPIOINTFLAGRAW (GPIOA Interrupt Flag Raw, 400D 1414h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

9.7.7 GPIOINTFLAGMASKED

Register 9-7 GPIOINTFLAGMASKED (GPIOA Interrupt Flag Masked, 400D 1418h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

9.7.8 GPIOAINTCLEAR

Register 9-8 GPIOAINTCLEAR (GPIOA Interrupt Clear, 400D 141Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOAINTFLAGRAW.P0 |

9.7.9 GPIOAINTTYPE

Register 9-9 GPIOAINTTYPE (GPIOA Interrupt Type Configuration, 400D 1420h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

9.7.10 GPIOAINTCFG

Register 9-10 GPIOAINTCFG (GPIOA Interrupt Value Configuration, 400D 1424h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |

| | | | | |
|---|----|----|---|---|
| 2 | P2 | RW | 0 | <p>If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 2 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 2 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 1 | P1 | RW | 0 | <p>If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 1 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 1 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 0 | P0 | RW | 0 | <p>If GPIOAINTTYPE.Px = 0 (edge-triggered), Pin 0 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOAINTTYPE.Px = 1 (level-triggered), Pin 0 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |

9.7.11 GPIOAINTEDGEBOOTH

Register 9-11 GPIOAINTEDGEBOOTH (GPIOA Interrupt Both Edge Configuration, 400D 1428h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOAINTCFG.P0 1b: Trigger on both rising and falling edge |

9.7.12 GPIOA DEBOUNCE

Register 9-12 GPIOA DEBOUNCE (GPIOA De-bounce Enable, 400D 142Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

9.7.13 GPIOADSET

Register 9-13 GPIOADSET (GPIOA Data Output Set, 400D 1430h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P0 to 1b |

9.7.14 GPIOADOCLEAR

Register 9-14 GPIOADOCLEAR (GPIOA Data Output Clear, 400D 1434h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOAOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOAOUT.P0 to 0b |

10 GPIOB

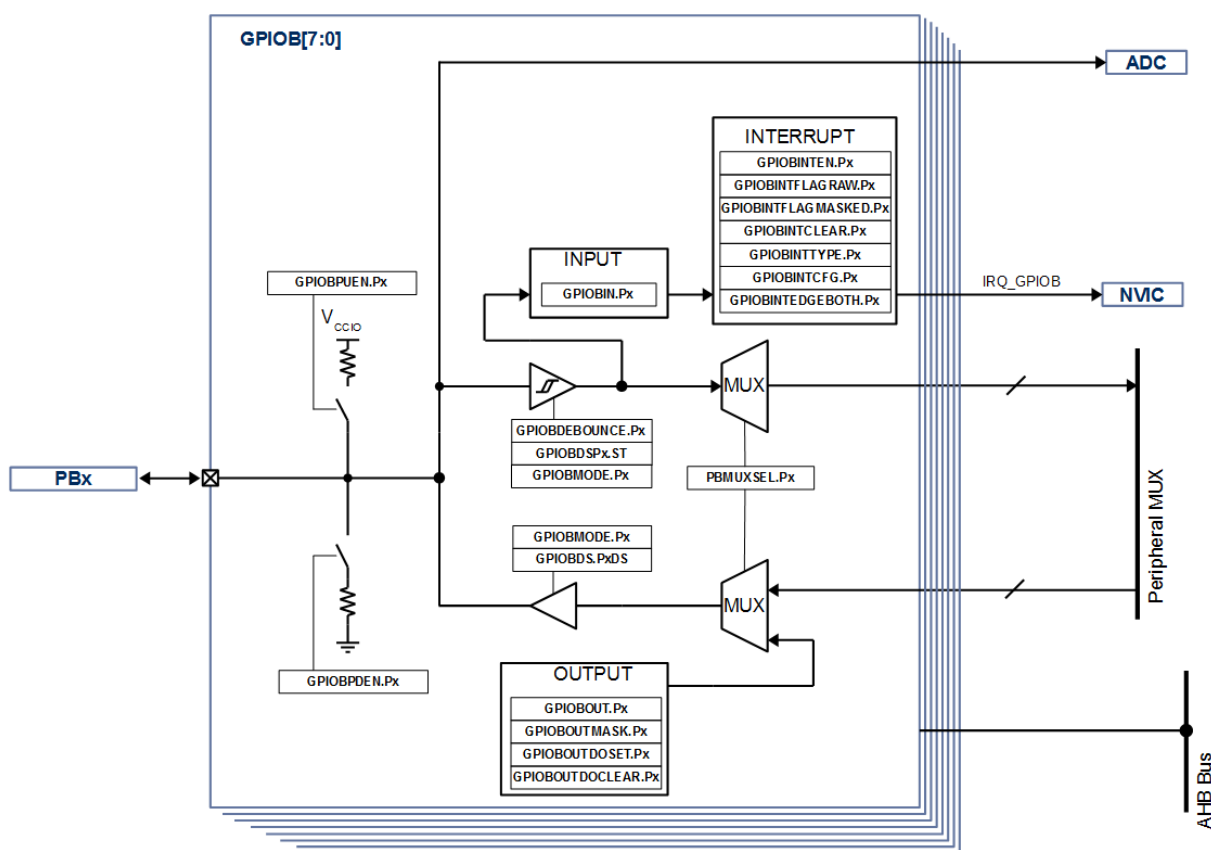
10.1 Overview

The GPIOB port contains 8 GPIO pin, GPIOB0 to GPIOB7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁴ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

10.2 GPIOB Block Diagram

Figure 10-1 GPIOB Block Diagram



⁴ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

10.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

10.4 Functional Description

10.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOBMODE** register as shown below.

Table 10-1 GPIOB Mode Configuration

| GPIOBMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Reserved |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

10.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOBMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOBOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOBOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOBOUTMASK.Px** is set to 1b, then any changes to **GPIOBOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOBIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

10.4.3 Digital Input

If **GPIOBMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOBIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

10.4.4 GPIOB Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOBMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 10-2 GPIOB Interrupt Configuration

| GPIOBINTTYPE | GPIOBINTCFG | GPIOBEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIOBINTEN** register. To enable interrupts on a pin, set the **GPIOBINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOBINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOBINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOBINTFLAGRAW.Px** is set to 1b and **GPIOBINTFLAGMASKED.Px** is set to 1b, then the **GPIOBINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOB signal to the NVIC is asserted.

10.4.5 GPIOB De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOBMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOBDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOBDEBOUNCE.Px** to 0b.

10.4.6 GPIOB Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOB Drive strength.

Table -3 GPIOB Drive Strength

| PBDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |

| | |
|------|------|
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOB Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

10.4.7 GPIOB Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOBMODE.Px = 11b**).

The Schmitt Trigger is disabled by default (**PB.PxST = 0b**). To enable the input Schmitt Trigger, set **PB.PxST = 1b**.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

10.4.8 GPIOB Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOBPU.Px** to 1b. To disable the pull-up, set **GPIOBPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOBPD.Px** to 1b. To disable the pull-down, set **GPIOBPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

10.5 Peripheral IO Mapping

The GPIOB peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 10-4 GPIOB Peripheral IO Mapping

| GPIOB SIGNAL | IO PIN |
|---------------------|---------------|
| GPIOB0 | PB0 |
| GPIOB1 | PB1 |
| GPIOB2 | PB2 |
| GPIOB3 | PB3 |
| GPIOB4 | PB4 |
| GPIOB5 | PB5 |
| GPIOB6 | PB6 |
| GPIOB7 | PB7 |

10.6 Register Summary

Table 10-5 GPIOB Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOBMODE | 400D 1800h | GPIOB Pin Mode Select | RW | 0000 FFFFh |
| GPIOBOUTMASK | 400D 1804h | GPIOB Data Output Write Mask | RW | 0000 0000h |
| GPIOBOUT | 400D 1808h | GPIOB Data Output Value | RW | 0000 00FFh |
| GPIOBIN | 400D 180Ch | GPIOB Data Input Value | RO | -- |
| GPIOBINTEN | 400D 1810h | GPIOB Interrupt Enable | RW | 0000 0000h |
| GPIOBINTFLAGRAW | 400D 1814h | GPIOB Interrupt Flag Raw | RW | 0000 0000h |
| GPIOBINTFLAGMASKED | 400D 1818h | GPIOB Interrupt Flag Masked | RW | 0000 0000h |
| GPIOBINTCLEAR | 400D 181Ch | GPIOB Interrupt Clear | WO | 0000 0000h |
| GPIOBINTTYPE | 400D 1820h | GPIOB Interrupt Type | RW | 0000 0000h |
| GPIOBINTCFG | 400D 1824h | GPIOB Interrupt Configuration | RW | 0000 0000h |
| GPIOBINTEDGEBOTH | 400D 1828h | GPIOB Interrupt Edge Both | RW | 0000 0000h |
| GPIOBDEBOUNCE | 400D 182Ch | GPIOB De-bounce Filter | RW | 0000 0000h |
| GPIOBDOSET | 400D 1830h | GPIOB Data Output Set | WO | 0000 0000h |
| GPIOBDOCLEAR | 400D 1834h | GPIOB Data Output Clear | WO | 0000 0000h |

10.7 Register Detail

10.7.1 GPIOBMODE

Register 10-1 GPIOBMODE (GPIOB Mode Configuration, 400D 1800h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

10.7.2 GPIOBOUTMASK

Register 10-2 GPIOBOUTMASK (GPIOB Output Mask, 400D 1804h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

10.7.3 GPIOBOUT

Register 10-3 GPIOBOUT (GPIOB Output Data Value, 400D 1808h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

10.7.4 GPIOBIN

Register 10-4 GPIOBIN (GPIOB Input Data Value, 400D 180Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

10.7.5 GPIOBINTEN

Register 10-5 GPIOBINTEN (GPIOB Interrupt Enable, 400D 1810h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

10.7.6 GPIOBINTFLAGRAW

Register 10-6 GPIOBINTFLAGRAW (GPIOB Interrupt Flag Raw, 400D 1814h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

10.7.7 GPIOBINTFLAGMASKED

Register 10-7 GPIOBINTFLAGMASKED (GPIOB Interrupt Flag Masked, 400D 1818h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

10.7.8 GPIOBINTCLEAR

Register 10-8 GPIOBINTCLEAR (GPIOB Interrupt Clear, 400D 181Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOBINTFLAGRAW.P0 |

10.7.9 GPIOBINTTYPE

Register 10-9 GPIOBINTTYPE (GPIOB Interrupt Type Configuration, 400D 1820h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

10.7.10 GPIOBINTCFG

Register 10-10 GPIOBINTCFG (GPIOB Interrupt Value Configuration, 400D 1824h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIOBINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low 1b: Logic high |

| | | | | |
|---|-----------|----|---|--|
| 1 | P1 | RW | 0 | <p>If GPIOBINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOINTTYPE.Px = 1b (level), Pin 1 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 0 | P0 | RW | 0 | <p>If GPIOBINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOINTTYPE.Px = 1b (level), Pin 0 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |

10.7.11 GPIOBINTEDGEBOTH

Register 10-11 GPIOBINTEDGEBOTH (GPIOB Interrupt Both Edge Configuration, 400D 1828h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOBINTCFG.P0 1b: Trigger on both rising and falling edge |

10.7.12 GPIOBDEBOUNCE

Register 10-12 GPIOBDEBOUNCE (GPIOB De-bounce Enable, 400D 182Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

10.7.13 GPIOBDOSET

Register 10-13 GPIOBDOSET (GPIOB Data Output Set, 400D 1830h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P0 to 1b |

10.7.14 GPIOBDOCLEAR

Register 10-14 GPIOBDOCLEAR (GPIOB Data Output Clear, 400D 1834h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOBOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOBOUT.P0 to 0b |

11 GPIOC

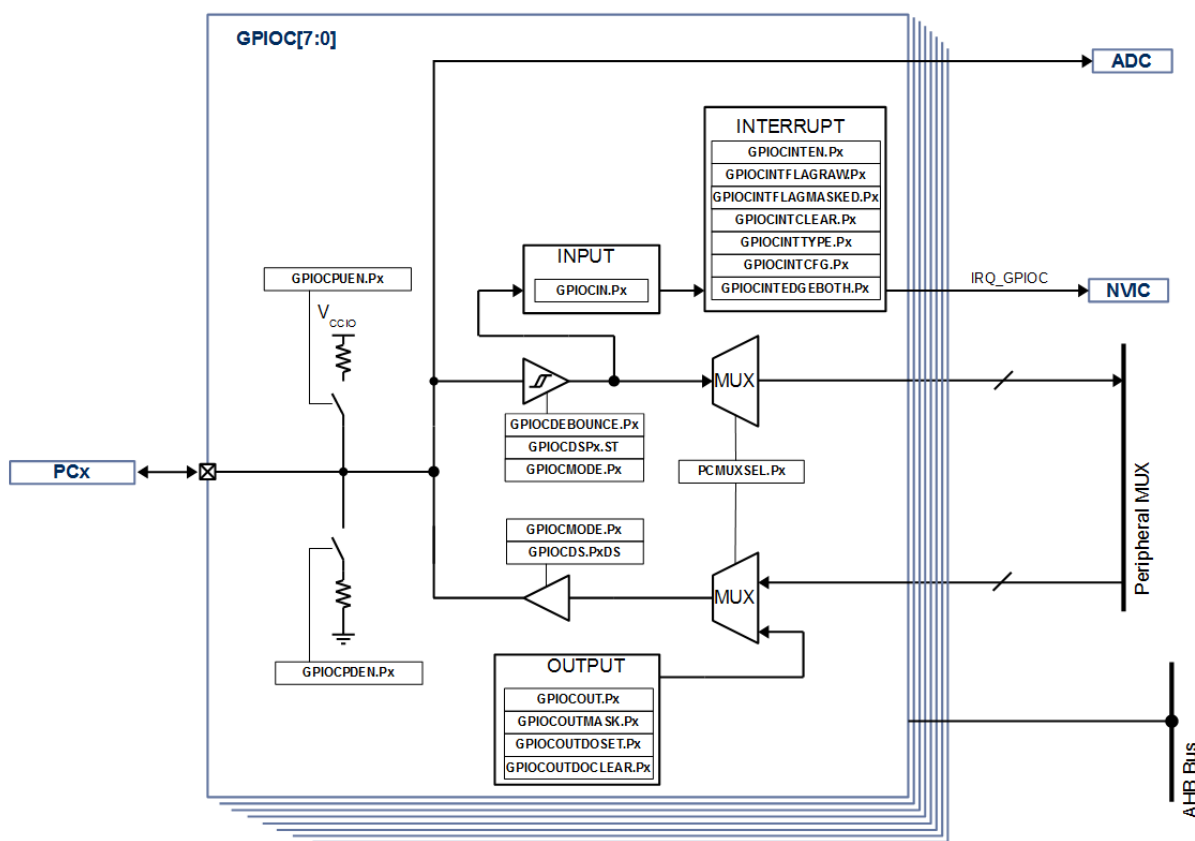
11.1 Overview

The GPIOC port contains 8 GPIO pin, GPIOC0 to GPIOC7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁵ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

11.2 GPIOC Block Diagram

Figure 11-1 GPIOC Block Diagram



⁵ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

11.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

11.4 Functional Description

11.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOCMODE** register as shown below.

Table 11-1 GPIOC Mode Configuration

| GPIOCMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Reserved |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

11.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOCMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOCOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOCOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOCOUTMASK.Px** is set to 1b, then any changes to **GPIOCOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOCIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

11.4.3 Digital Input

If **GPIOCMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOCIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

11.4.4 GPIOC Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOCMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 11-2 GPIOC Interrupt Configuration

| GPIOCINTTYPE | GPIOCINTCFG | GPIOCEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIOCINTEN** register. To enable interrupts on a pin, set the **GPIOCINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOCINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOCINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOCINTFLAGRAW.Px** is set to 1b and **GPIOCINTFLAGMASKED.Px** is set to 1b, then the **GPIOCINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOC signal to the NVIC is asserted.

11.4.5 GPIOC De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOCMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOCDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOCDEBOUNCE.Px** to 0b.

11.4.6 GPIOC Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOC Drive strength.

Table 11-3 GPIOC Drive Strength

| PCDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |

| | |
|------|------|
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOC Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

11.4.7 GPIOC Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOCMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PCDS.PxST** = 0b). To enable the input Schmitt Trigger, set **PCDS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

11.4.8 GPIOC Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOCPU.Px** to 1b. To disable the pull-up, set **GPIOCPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOCPD.Px** to 1b. To disable the pull-down, set **GPIOCPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

11.5 Peripheral IO Mapping

The GPIOC peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 11-4 GPIOC Peripheral IO Mapping

| GPIOC SIGNAL | IO PIN |
|---------------------|---------------|
| GPIOC0 | PC0 |
| GPIOC1 | PC1 |
| GPIOC2 | PC2 |
| GPIOC3 | PC3 |
| GPIOC4 | PC4 |
| GPIOC5 | PC5 |
| GPIOC6 | PC6 |
| GPIOC7 | PC7 |

11.6 Register Summary

Table 11-5 GPIOC Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOCMODE | 400D 1C00h | GPIOC Pin Mode Select | RW | 0000 FFFFh |
| GPIOCOUTMASK | 400D 1C04h | GPIOC Data Output Write Mask | RW | 0000 0000h |
| GPIOCOUT | 400D 1C08h | GPIOC Data Output Value | RW | 0000 00FFh |
| GPIOCIN | 400D 1C0Ch | GPIOC Data Input Value | RO | -- |
| GPIOCINTEN | 400D 1C10h | GPIOC Interrupt Enable | RW | 0000 0000h |
| GPIOCINTFLAGRAW | 400D 1C14h | GPIOC Interrupt Flag | RW | 0000 0000h |
| GPIOCINTFLAGMASKED | 400D 1C18h | GPIOC Interrupt Flag Masked | RW | 0000 0000h |
| GPIOCINTCLEAR | 400D 1C1Ch | GPIOC Interrupt Clear | WO | 0000 0000h |
| GPIOCINTTYPE | 400D 1C20h | GPIOC Interrupt Type | RW | 0000 0000h |
| GPIOCINTCFG | 400D 1C24h | GPIOC Interrupt Configuration | RW | 0000 0000h |
| GPIOCINTEDGEBOTH | 400D 1C28h | GPIOC Interrupt Edge Both | RW | 0000 0000h |
| GPIOCDEBOUNCE | 400D 1C2Ch | GPIOC De-bounce Filter | RW | 0000 0000h |
| GPIOCDOSET | 400D 1C30h | GPIOC Data Output Set | WO | 0000 0000h |
| GPIOCDOCLEAR | 400D 1C34h | GPIOC Data Output Clear | WO | 0000 0000h |

11.7 Register Detail

11.7.1 GPIOCMODE

Register 11-1 GPIOCMODE (GPIOC Mode Configuration, 400D 1C00h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

11.7.2 GPIOCOUTMASK

Register 11-2 GPIOCOUTMASK (GPIOC Output Mask, 400D 1C04h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

11.7.3 GPIOCOUT

Register 11-3 GPIOCOUT (GPIOC Output Data Value, 400D 1C08h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

11.7.4 GPIOCIN

Register 11-4 GPIOCIN (GPIOC Input Data Value, 400D 1C0Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

11.7.5 GPIOCINTEN

Register 11-5 GPIOCINTEN (GPIOC Interrupt Enable, 400D 1C10h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

11.7.6 GPIOCINTFLAGRAW

Register 11-6 GPIOCINTFLAGRAW (GPIOC Interrupt Flag Raw, 400D 1C14h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

11.7.7 GPIOCINTFLAGMASKED

Register 11-7 GPIOCINTFLAGMASKED (GPIOC Interrupt Flag Masked, 400D 1C18h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

11.7.8 GPIOCINTCLEAR

Register 11-8 GPIOCINTCLEAR (GPIOC Interrupt Clear, 400D 1C1Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOCINTFLAGRAW.P0 |

11.7.9 GPIOCINTTYPE

Register 11-9 GPIOCINTTYPE (GPIOC Interrupt Type Configuration, 400D 1C20h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

11.7.10 GPIOCINTCFG

Register 11-10 GPIOCINTCFG (GPIOC Interrupt Configuration, 400D 1C24h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIOCINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOCINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low 1b: Logic high |

| | | | | |
|---|-----------|----|---|---|
| 1 | P1 | RW | 0 | <p>If GPIOCINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOCINTTYPE.Px = 1b (level), Pin 1 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 0 | P0 | RW | 0 | <p>If GPIOCINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOCINTTYPE.Px = 1b (level), Pin 0 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |

11.7.11 GPIOCINTEDGEBOTH

Register 11-11 GPIOCINTEDGEBOTH (GPIOC Interrupt Both Edge Configuration, 400D 1C28h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOCINTCFG.P0 1b: Trigger on both rising and falling edge |

11.7.12 GPIOCDEBOUNCE

Register 11-12 GPIOCDEBOUNCE (GPIOC De-bounce Enable, 400D 1C2Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

11.7.13 GPIOCDOSET

Register 11-13 GPIOCDOSET (GPIOC Data Output Set, 400D 1C30h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P0 to 1b |

11.7.14 GPIOCDOCLEAR

Register 11-14 GPIOCDOCLEAR (GPIOC Data Output Clear, 400D 1C34h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOCOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOCOUT.P0 to 0b |

12 GPIOD

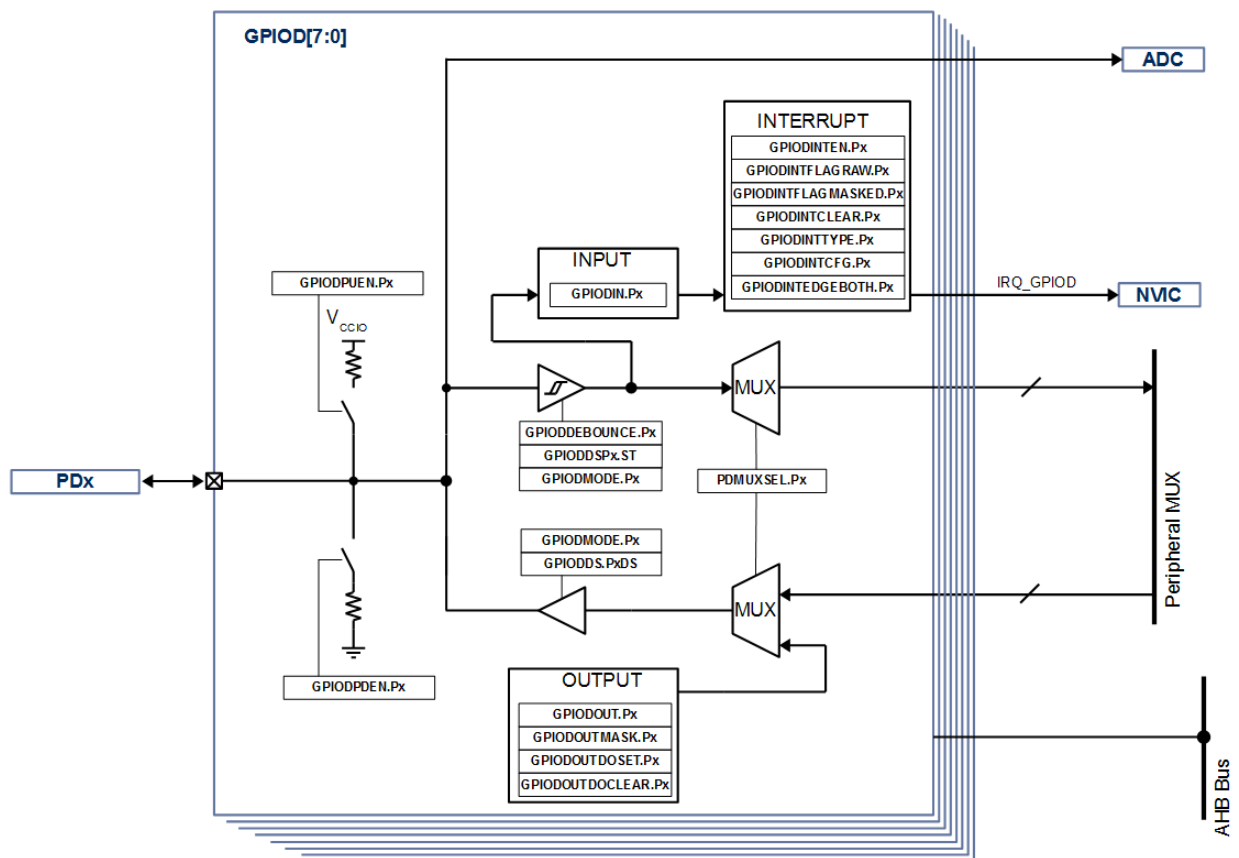
12.1 Overview

The GPIOD port contains 8 GPIO pin, GPIOD0 to GPIOD7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁶ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

12.2 GPIOD Block Diagram

Figure 12-1 GPIOD Block Diagram



⁶ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

12.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

12.4 Functional Description

12.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIODMODE** register as shown below.

Table 12-1 GPIOD Mode Configuration

| GPIODMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Analog Input |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

12.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIODMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOCOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOCOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOCOUTMASK.Px** is set to 1b, then any changes to **GPIOCOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIODIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

12.4.3 Digital Input

If **GPIODMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIODIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

12.4.4 Analog Input

Some GPIO pins may be configured for Analog Input.

To configure a GPIO pin for Analog Input, the user may set **GPIODMODE.Px** to 00b. In this mode, the user MUST be sure to set the output mode to high-impedance input by setting **GPIODMODE.Px** to 11b or the Analog Input will not work properly.

To determine if your device supports Analog Input for each GPIO pin, consult the device data sheet.

12.4.5 GPIOD Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIODMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 12-2 GPIOD Interrupt Configuration

| GPIODINTTYPE | GPIODINTCFG | GPIODEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIODINTEN** register. To enable interrupts on a pin, set the **GPIODINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIODINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIODINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIODINTFLAGRAW.Px** is set to 1b and **GPIODINTFLAGMASKED.Px** is set to 1b, then the **GPIODINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIO signal to the NVIC is asserted.

12.4.6 GPIOD De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIODMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIODDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIODDEBOUNCE.Px** to 0b.

12.4.7 GPIOD Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOD Drive strength.

Table 12-3 GPIOD Drive Strength

| PDDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOD Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

12.4.8 GPIOD Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIODMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PDDS.PxST** = 0b). To enable the input Schmitt Triger, set **PDDS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

12.4.9 GPIOD Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIODPU.Px** to 1b. To disable the pull-up, set **GPIODPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIODPD.Px** to 1b. To disable the pull-down, set **GPIODPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

12.5 Peripheral IO Mapping

The GPIOD peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 12-4 GPIOD Peripheral IO Mapping

| GPIOD SIGNAL | IO PIN |
|---------------|--------|
| GPIOD0 | PD0 |
| GPIOD1 | PD1 |
| GPIOD2 | PD2 |
| GPIOD3 | PD3 |
| GPIOD4 | PD4 |
| GPIOD5 | PD5 |
| GPIOD6 | PD6 |
| GPIOD7 | PD7 |

12.6 Register Summary

Table 12-5 GPIOD Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIODMODE | 400D 2000h | GPIOD Pin Mode Select | RW | 0000 FFFFh |
| GPIODOUTMASK | 400D 2004h | GPIOD Data Output Write Mask | RW | 0000 0000h |
| GPIODOUT | 400D 2008h | GPIOD Data Output Value | RW | 0000 00FFh |
| GPIODIN | 400D 200Ch | GPIOD Data Input Value | RO | -- |
| GPIODINTEN | 400D 2010h | GPIOD Interrupt Enable | RW | 0000 0000h |
| GPIODINTFLAGRAW | 400D 2014h | GPIOD Interrupt Flag Raw | RW | 0000 0000h |
| GPIODINTFLAGMASKED | 400D 2018h | GPIOD Interrupt Flag Masked | RW | 0000 0000h |
| GPIODINTCLEAR | 400D 201Ch | GPIOD Interrupt Clear | WO | 0000 0000h |
| GPIODINTTYPE | 400D 2020h | GPIOD Interrupt Type | RW | 0000 0000h |
| GPIODINTCFG | 400D 2024h | GPIOD Interrupt Configuration | RW | 0000 0000h |
| GPIODINTEDGEBOTH | 400D 2028h | GPIOD Interrupt Edge Both | RW | 0000 0000h |
| GPIODDEBOUNCE | 400D 202Ch | GPIOD De-bounce Filter | RW | 0000 0000h |
| GPIODDOSET | 400D 2030h | GPIOD Data Output Set | WO | 0000 0000h |
| GPIODDOCLEAR | 400D 2034h | GPIOD Data Output Clear | WO | 0000 0000h |

12.7 Register Detail

12.7.1 GPIODMODE

Register 12-1 GPIODMODE (GPIOD Mode Configuration, 400D 2000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Analog Input (ADC1) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Analog Input (ADC2) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Analog Input (ADC3) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Analog Input (ADC4) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

12.7.2 GPIODOUTMASK

Register 12-2 GPIODOUTMASK (GPIO Output Mask, 400D 2004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

12.7.3 GPIODOUT

Register 12-3 GPIODOUT (GPIOD Output Data Value, 400D 2008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

12.7.4 GPIODIN

Register 12-4 GPIODIN (GPIOD Input Data Value, 400D 200Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

12.7.5 GPIODINTEN

Register 12-5 GPIODINTEN (GPIOD Interrupt Enable, 400D 2010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

12.7.6 GPIODINTFLAGRAW

Register 12-6 GPIODINTFLAGRAW (GPIOD Interrupt Flag Raw, 400D 2014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

12.7.7 GPIODINTFLAGMASKED

Register 12-7 GPIODINTFLAGMASKED (GPIOD Interrupt Flag Masked, 400D 2018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

12.7.8 GPIODINTCLEAR

Register 12-8 GPIODINTCLEAR (GPIO Interrupt Clear, 400D 201Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIODINTFLAGRAW.P0 |

12.7.9 GPIODINTTYPE

Register 12-9 GPIODINTTYPE (GPIOD Interrupt Type Configuration, 400D 2020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

12.7.10 GPIODINTCFG

Register 12-10 GPIODINTCFG (GPIOD Interrupt Configuration, 400D 2024h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIODINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIODINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low 1b: Logic high |

| | | | | |
|---|-----------|----|---|---|
| 1 | P1 | RW | 0 | <p>If GPIODINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIODINTTYPE.Px = 1b (level), Pin 1 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 0 | P0 | RW | 0 | <p>If GPIODINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIODINTTYPE.Px = 1b (level), Pin 0 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |

12.7.11 GPIODINTEDGEBOTH

Register 12-11 GPIODINTEDGEBOTH (GPIOD Interrupt Both Edge Configuration, 400D 2028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIODINTCFG.P0 1b: Trigger on both rising and falling edge |

12.7.12 GPIODDEBOUNCE

Register 12-12 GPIODDEBOUNCE (GPIO De-bounce Enable, 400D 202Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

12.7.13 GPIODDOSET

Register 12-13 GPIODDOSET (GPIOD Data Output Set, 400D 2030h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P0 to 1b |

12.7.14 GPIODDOCLEAR

Register 12-14 GPIODDOCLEAR (GPIOD Data Output Clear, 400D 2034h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIODOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIODOUT.P0 to 0b |

13 GPIOE

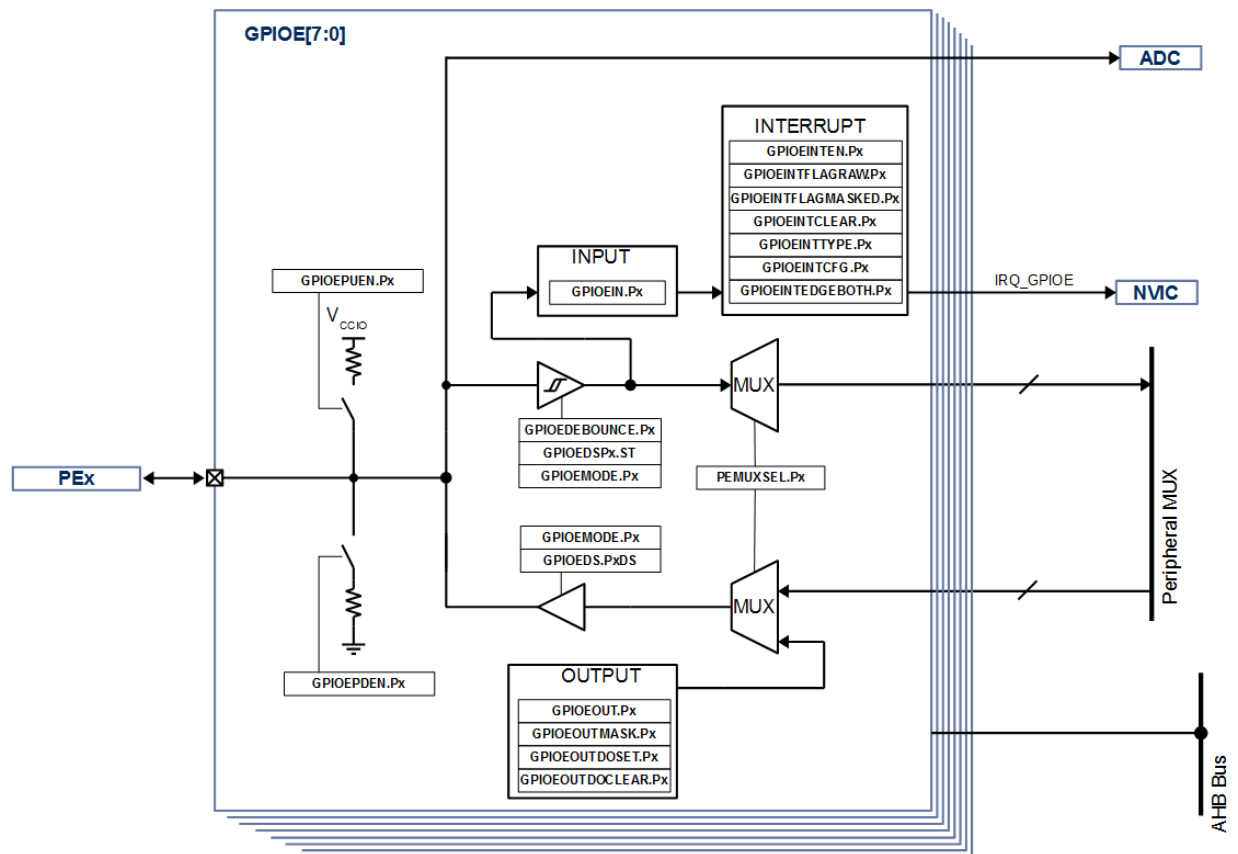
13.1 Overview

The GPIOE port contains 8 GPIO pin, GPIOE0 to GPIOE7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁷ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

13.2 GPIOE Block Diagram

Figure 13-1 GPIOE Block Diagram



⁷ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

13.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

13.4 Functional Description

13.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOEMODE** register as shown below.

Table 13-1 GPIOE Mode Configuration

| GPIOEMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Reserved |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

13.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOEMODE.Px** to either 01b (Pull-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOEOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOEOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOEOUTMASK.Px** is set to 1b, then any changes to **GPIOEOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOEIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

13.4.3 Digital Input

If **GPIOEMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOEIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

13.4.4 GPIOE Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOEMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 13-2 GPIOE Interrupt Configuration

| GPIOEINTTYPE | GPIOEINTCFG | GPIOEEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIOEINTEN** register. To enable interrupts on a pin, set the **GPIOEINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOEINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOEINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOEINTFLAGRAW.Px** is set to 1b and **GPIOEINTFLAGMASKED.Px** is set to 1b, then the **GPIOEINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOE signal to the NVIC is asserted.

13.4.5 GPIOE De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOEMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOEDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOEDEBOUNCE.Px** to 0b.

13.4.6 GPIOE Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOE Drive strength.

Table 13-3 GPIOE Drive Strength

| PEDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |

| | |
|------|------|
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOE Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

13.4.7 GPIOE Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOEMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PEDS.PxST** = 0b). To enable the input Schmitt Trigger, set **PEDS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

13.4.8 GPIOE Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOEPU.Px** to 1b. To disable the pull-up, set **GPIOEPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOEPD.Px** to 1b. To disable the pull-down, set **GPIOEPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

13.5 Peripheral IO Mapping

The GPIOE peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 13-4 GPIOE Peripheral IO Mapping

| GPIOE SIGNAL | IO PIN |
|--------------|--------|
| GPIOE0 | PE0 |
| GPIOE1 | PE1 |
| GPIOE2 | PE2 |
| GPIOE3 | PE3 |
| GPIOE4 | PE4 |
| GPIOE5 | PE5 |
| GPIOE6 | PE6 |
| GPIOE7 | PE7 |

13.6 Register Summary

Table 13-5 GPIOE Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOEMODE | 400D 2400h | GPIOE Pin Mode Select | RW | 0000 FFFFh |
| GPIOEOUTMASK | 400D 2404h | GPIOE Data Output Write Mask | RW | 0000 0000h |
| GPIOEOUT | 400D 2408h | GPIOE Data Output Value | RW | 0000 00FFh |
| GPIOEIN | 400D 240Ch | GPIOE Data Input Value | RO | -- |
| GPIOEINTEN | 400D 2410h | GPIOE Interrupt Enable | RW | 0000 0000h |
| GPIOEINTFLAGRAW | 400D 2414h | GPIOE Interrupt Flag Raw | RW | 0000 0000h |
| GPIOEINTFLAGMASKED | 400D 2418h | GPIOE Interrupt Flag Masked | RW | 0000 0000h |
| GPIOEINTCLEAR | 400D 241Ch | GPIOE Interrupt Clear | WO | 0000 0000h |
| GPIOEINTTYPE | 400D 2420h | GPIOE Interrupt Type | RW | 0000 0000h |
| GPIOEINTCFG | 400D 2424h | GPIOE Interrupt Configuration | RW | 0000 0000h |
| GPIOEINTEDGEBOTH | 400D 2428h | GPIOE Interrupt Edge Both | RW | 0000 0000h |
| GPIOEDEBOUNCE | 400D 242Ch | GPIOE De-bounce Filter | RW | 0000 0000h |
| GPIOEDOSET | 400D 2430h | GPIOE Data Output Set | WO | 0000 0000h |
| GPIOEDOCLEAR | 400D 2434h | GPIOE Data Output Clear | WO | 0000 0000h |

13.7 Register Detail

13.7.1 GPIOEMODE

Register 13-1 GPIOEMODE (GPIOE Mode Configuration, 400D 2400h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

13.7.2 GPIOEOUTMASK

Register 13-2 GPIOEOUTMASK (GPIOE Output Mask, 400D 2404h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

13.7.3 GPIOEOUT

Register 13-3 GPIOEOUT (GPIOE Output Data Value, 400E 2408h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

13.7.4 GPIOEIN

Register 13-4 GPIOEIN (GPIOE Input Data Value, 400D 240Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

13.7.5 GPIOEINTEN

Register 13-5 GPIOEINTEN (GPIOE Interrupt Enable, 400D 2410h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

13.7.6 GPIOEINTFLAGRAW

Register 13-6 GPIOEINTFLAGRAW (GPIOE Interrupt Flag Raw, 400D 2414h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

13.7.7 GPIOEINTFLAGMASKED

Register 13-7 GPIOEINTFLAGMASKED (GPIOE Interrupt Flag Masked, 400D 2418h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

13.7.8 GPIOINTCLEAR

Register 13-8 GPIOINTCLEAR (GPIOE Interrupt Clear, 400D 241Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P0 |

13.7.9 GPIOEINTTYPE

Register 13-9 GPIOEINTTYPE (GPIOE Interrupt Type Configuration, 400D 2420h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

13.7.10 GPIOINTCFG

Register 13-10 GPIOINTCFG (GPIOE Interrupt Configuration, 400D 2424h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low |

| | | | | |
|---|-----------|----|---|--|
| | | | | 1b: Logic high |
| 1 | P1 | RW | 0 | If GPIOEINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOEINTTYPE.Px = 1b (level), Pin 1 Interrupt Level: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 0 | If GPIOEINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOEINTTYPE.Px = 1b (level), Pin 0 Interrupt Level: 0b: Logic low 1b: Logic high |

13.7.11 GPIOINTEDGEBOTH

Register 13-11 GPIOINTEDGEBOTH (GPIOE Interrupt Both Edge Configuration, 400D 2428h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFGE.P0 1b: Trigger on both rising and falling edge |

13.7.12 GPIOEBOUNCE

Register 13-12 GPIOEBOUNCE (GPIOE De-bounce Enable, 400D 242Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

13.7.13 GPIOEDOSET

Register 13-13 GPIOEDOSET (GPIOE Data Output Set, 400D 2430h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P0 to 1b |

13.7.14 GPIOEDOCLEAR

Register 13-14 GPIOEDOCLEAR (GPIOE Data Output Clear, 400D 2434h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOEOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOEOUT.P0 to 0b |

14 GPIOF

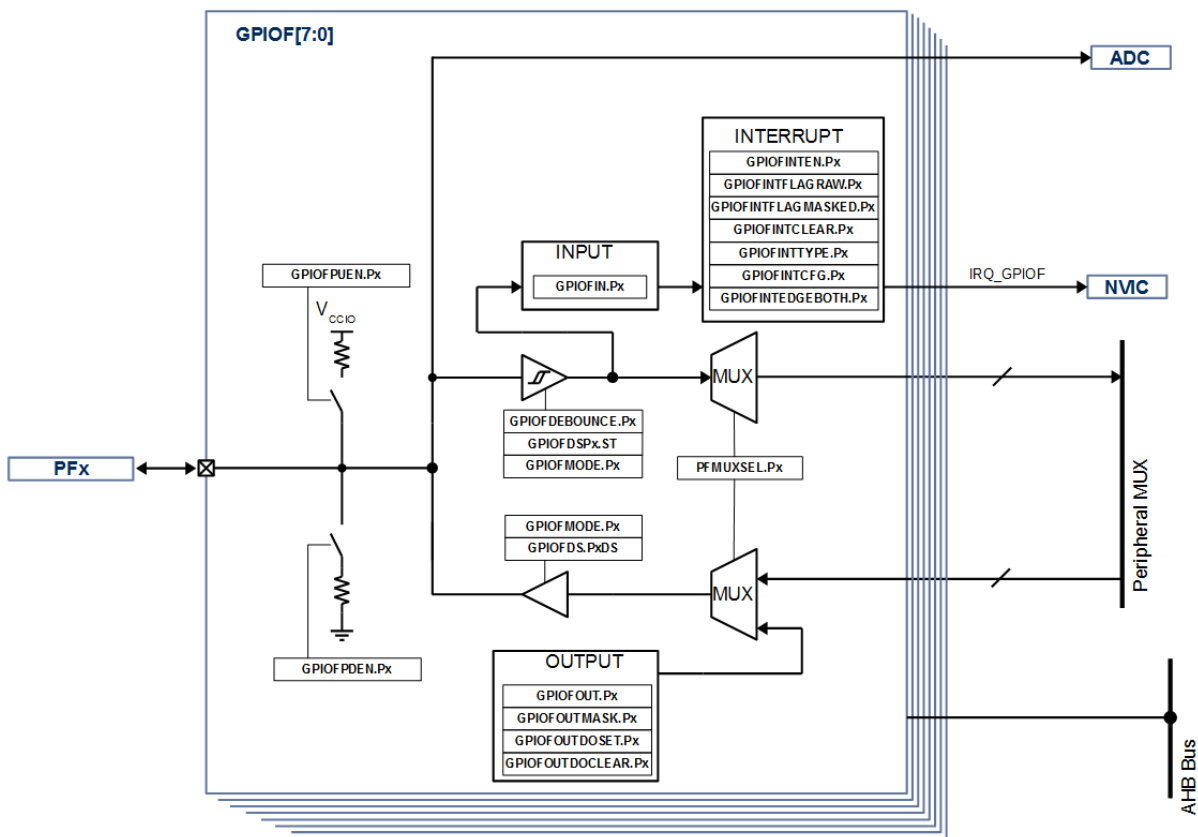
14.1 Overview

The GPIOF port contains 8 GPIO pin, GPIOF0 to GPIOF7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁸ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

14.2 GPIOF Block Diagram

Figure 14-1 GPIOF Block Diagram



⁸ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

14.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

14.4 Functional Description

14.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOFMODE** register as shown below.

Table 14-1 GPIOF Mode Configuration

| GPIOFMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Analog Input |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

14.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOFMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOFOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOFOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOFOUTMASK.Px** is set to 1b, then any changes to **GPIOFOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOFIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

14.4.3 Digital Input

If **GPIOFMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOFIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

14.4.4 Analog Input

Some GPIO pins may be configured for Analog Input.

To configure a GPIO pin for Analog Input, the user may set **GPIOFMODE.Px** to 00b. In this mode, the user MUST be sure to set the output mode to high-impedance input by setting **GPIOFMODE.Px** to 11b or the Analog Input will not work properly.

To determine if your device supports Analog Input for each GPIO pin, consult the device data sheet.

14.4.5 GPIOF Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOFMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 14-2 GPIOF Interrupt Configuration

| GPIOFINTTYPE | GPIOFINTCFG | GPIOFEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|---------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIOFLAGINTEN** register. To enable interrupts on a pin, set the **GPIOFLAGINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOFLAGINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOFINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOFINTFLAGRAW.Px** is set to 1b and **GPIOFINTFLAGMASKED.Px** is set to 1b, then the **GPIOFINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOF signal to the NVIC is asserted.

14.4.6 GPIOF De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOFMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOFDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOFDEBOUNCE.Px** to 0b.

14.4.7 GPIOF Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOF Drive strength.

Table 14-3 GPIOF Drive Strength

| PFDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOF Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

14.4.8 GPIOF Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOFMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PFDS.PxST** = 0b). To enable the input Schmitt Triger, set **PFDS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

14.4.9 GPIOF Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOFPU.Px** to 1b. To disable the pull-up, set **GPIOFPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOFPD.Px** to 1b. To disable the pull-down, set **GPIOFPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

14.5 Peripheral IO Mapping

The GPIOF peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 14-4 GPIOF Peripheral IO Mapping

| GPIOF SIGNAL | IO PIN |
|--------------|--------|
| GPIOF0 | PF0 |
| GPIOF1 | PF1 |
| GPIOF2 | PF2 |
| GPIOF3 | PF3 |
| GPIOF4 | PF4 |
| GPIOF5 | PF5 |
| GPIOF6 | PF6 |
| GPIOF7 | PF7 |

14.6 Register Summary

Table 14-5 GPIOF Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|-------------------------------|--------|------------|
| GPIOFMODE | 400D 2800h | GPIOF Pin Mode Select | RW | 0000 FFFFh |
| GPIOFOUTMASK | 400D 2804h | GPIOF Data Output Write Mask | RW | 0000 0000h |
| GPIOFOUT | 400D 2808h | GPIOF Data Output Value | RW | 0000 00FFh |
| GPIOFIN | 400D 280Ch | GPIOF Data Input Value | RO | -- |
| GPIOFINTEN | 400D 2810h | GPIOF Interrupt Enable | RW | 0000 0000h |
| GPIOFINTFLAGRAW | 400D 2814h | GPIOF Interrupt Flag | RW | 0000 0000h |
| GPIOFINTFLAGMASKED | 400D 2818h | GPIOF Interrupt Flag Masked | RW | 0000 0000h |
| GPIOFINTCLEAR | 400D 281Ch | GPIOF Interrupt Clear | WO | 0000 0000h |
| GPIOFINTTYPE | 400D 2820h | GPIOF Interrupt Type | RW | 0000 0000h |
| GPIOFINTCFG | 400D 2824h | GPIOF Interrupt Configuration | RW | 0000 0000h |
| GPIOFINTEGEBOTH | 400D 2828h | GPIOF Interrupt Edge Both | RW | 0000 0000h |
| GPIOFDEBOUNCE | 400D 282Ch | GPIOF De-bounce Filter | RW | 0000 0000h |
| GPIOFDOSET | 400D 2830h | GPIOF Data Output Set | WO | 0000 0000h |
| GPIOFDOCLEAR | 400D 2834h | GPIOF Data Output Clear | WO | 0000 0000h |

14.7 Register Detail

14.7.1 GPIOFMODE

Register 14-1 GPIOFMODE (GPIOF Mode Configuration, 400D 2800h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 7 00b: Analog Input (ADC7) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Analog Input (ADC6) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Analog Input (ADC5) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Analog Input (ADC4) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

14.7.2 GPIOFOUTMASK

Register 14-2 GPIOFOUTMASK (GPIOF Output Mask, 400D 2804h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

14.7.3 GPIOFOUT

Register 14-3 GPIOFOUT (GPIOF Output Data Value, 400E 2808h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

14.7.4 GPIOFIN

Register 14-4 GPIOFIN (GPIOF Input Data Value, 400D 280Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

14.7.5 GPIOFINTEN

Register 14-5 GPIOFINTEN (GPIOF Interrupt Enable, 400D 2810h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

14.7.6 GPIOFINTFLAGRAW

Register 14-6 GPIOFINTFLAGRAW (GPIOF Interrupt Flag Raw, 400D 2814h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

14.7.7 GPIOFINTFLAGMASKED

Register 14-7 GPIOFINTFLAGMASKED (GPIOF Interrupt Flag Masked, 400D 2818h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

14.7.8 GPIOFINTCLEAR

Register 14-8 GPIOFINTCLEAR (GPIOF Interrupt Clear, 400D 281Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOFINTFLAGRAW.P0 |

14.7.9 GPIOFINTTYPE

Register 14-9 GPIOFINTTYPE (GPIOF Interrupt Type Configuration, 400D 2820h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

14.7.10 GPIOFINTCFG

Register 14-10 GPIOFINTCFG (GPIOF Interrupt Configuration, 400D 2824h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low |

| | | | | |
|---|-----------|----|---|--|
| | | | | 1b: Logic high |
| 1 | P1 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 1 Interrupt Level: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 0 | If GPIOFINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOFINTTYPE.Px = 1b (level), Pin 0 Interrupt Level: 0b: Logic low 1b: Logic high |

14.7.11 GPIOFINTEGEBOTH

Register 14-11 GPIOFINTEGEBOTH (GPIOF Interrupt Both Edge Configuration, 400D 2828h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOFINTCFG.P0 1b: Trigger on both rising and falling edge |

14.7.12 GPIOFDEBOUNCE

Register 14-12 GPIOFDEBOUNCE (GPIOF De-bounce Enable, 400D 282Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

14.7.13 GPIOFDOSET

Register 14-13 GPIOFDOSET (GPIOF Data Output Set, 400D 2830h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P0 to 1b |

14.7.14 GPIOFDOCLEAR

Register 14-14 GPIOFDOCLEAR (GPIOF Data Output Clear, 400D 2834h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOFOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOFOUT.P0 to 0b |

15 GPIOG

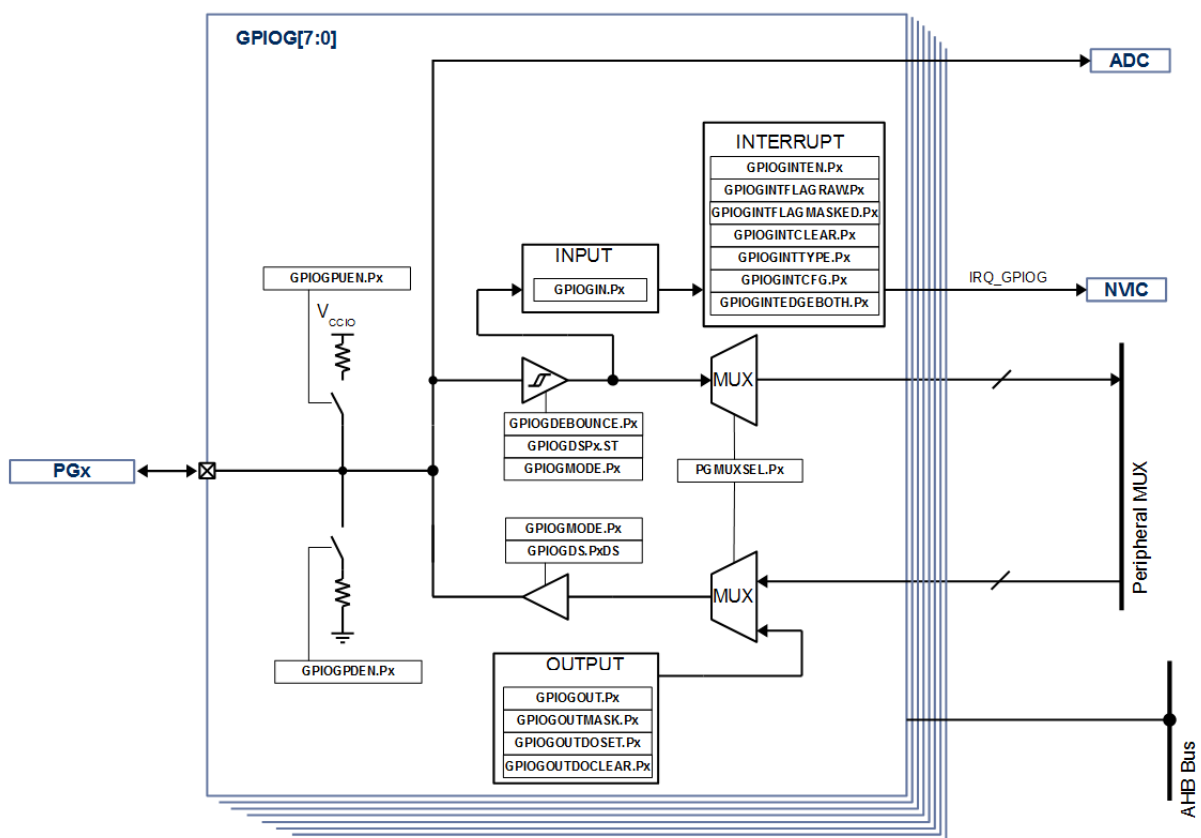
15.1 Overview

The GPIOG port contains 8 GPIO pin, GPIOG0 to GPIOG7.

Each GPIO pin may be used for Digital Input, Digital Output and some may be used for Analog Input.⁹ Each GPIO is capable of generating 3.3V output and is 3.3V input tolerant and has a set of features described below.

15.2 GPIOG Block Diagram

Figure 15-1 GPIOG Block Diagram



⁹ Consult the device data sheet for information on which GPIO pins are present on any given PAC55XX device.

15.3 Features

- 3.3V output, 3.3V input tolerant
- Configurable Pin Modes:
 - High-Impedance Digital Input
 - Push-Pull Digital Output
 - Open-Drain Digital Output
 - Analog Input
- Configurable Drive Strength up to 25mA
- Configurable Input Schmitt Trigger
- Configurable Weak Pull-up or Pull-down
- Edge or Level Sensitive Interrupts
 - Rising, Falling or both edges
- Configurable de-bouncing filter

15.4 Functional Description

15.4.1 IO Mode

Each GPIO pin supports up to 4 different modes and can be selected from the **GPIOGMODE** register as shown below.

Table 15-1 GPIOG Mode Configuration

| GPIOGMODE VALUE | DESCRIPTION |
|-----------------|------------------------------|
| 00b | Analog Input |
| 01b | Push-Pull Digital Output |
| 10b | Open-Drain Digital Output |
| 11b | High-Impedance Digital Input |

The reset value for the GPIO pin mode is High-Impedance Digital Input.

15.4.2 Digital Output

Each GPIO pin may be configured for digital output by setting the **GPIOGMODE.Px** to either 01b (Push-pull output) or to 10b (Open-drain output).

When configured for Push-pull output, the GPIO is driven to the high state using the VCCIO 3.3V power supply. When configured for Open-drain output, the GPIO is pulled up externally.

When **GPIOGOUTMASK.Px** is set to 0b, then the state of the GPIO pin will be set from **GPIOGOUT.Px** (0b: logic low, 1b: logic high). If the **GPIOGOUTMASK.Px** is set to 1b, then any changes to **GPIOGOUT.Px** have no effect.

In either of these two output modes, the user may read the digital state of the input by reading the **GPIOGIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports both digital output modes.

15.4.3 Digital Input

If **GPIOGMODE.Px** is set to 00b, then the GPIO pin is configured for high-impedance input. For application safety, this is the reset value for the GPIO pin mode.

In this mode, the user may read the state of the digital input by reading the **GPIOGIN.Px** value (0b: logic low, 1b: logic high).

Every GPIO pin supports this mode.

15.4.4 Analog Input

Some GPIO pins may be configured for Analog Input.

To configure a GPIO pin for Analog Input, the user may set **GPIOGMODE.Px** to 00b. In this mode, the user MUST be sure to set the output mode to high-impedance input by setting **GPIOGMODE.Px** to 11b or the Analog Input will not work properly.

To determine if your device supports Analog Input for each GPIO pin, consult the device data sheet.

15.4.5 GPIOG Interrupts

Each GPIO pin supports configurable interrupts. In order for interrupts to work, the GPIO must be set for high-impedance input mode (**GPIOGMODE.Px** = 11b).

The user may configure either edge or level sensitive interrupts for each GPIO pin. If the user configures edge sensitive interrupts, they may select rising, falling or both edges for edge interrupts.

See the table below for how to configure the GPIO for the various types of interrupts.

Table 15-2 GPIOG Interrupt Configuration

| GPIOGINTTYPE | GPIOGINTCFG | GPIOEDGEBOTH | INTERRUPT MODE |
|--------------|-------------|--------------|------------------------------|
| 0b | 0b | 0b | Falling Edge |
| | 1b | 0b | Rising Edge |
| | X | 1b | Both Rising and Falling Edge |
| 1b | 0b | X | Logic Low |
| | 1b | X | Logic High |

Interrupts may be enabled or disabled using the **GPIOGINTEN** register. To enable interrupts on a pin, set the **GPIOGINTEN.Px** to 0b (not masked). To disable interrupts on a pin, set the **GPIOGINTEN.Px** to 1b (masked).

If an edge or level interrupt is detected, the **GPIOGINTFLAGRAW.Px** will be set to 1b for that GPIO pin. If **GPIOGINTFLAGRAW.Px** is set to 1b and **GPIOGINTFLAGMASKED.Px** is set to 1b, then the **GPIOGINTFLAGMASKED.Px** is set to a 1b and the IRQ_GPIOG signal to the NVIC is asserted.

15.4.6 GPIOG De-bouncing

Each GPIO pin has a de-bouncing circuit that may filter noise on GPIO pins, when configured as an input (**GPIOGMODE.Px** = 11b).

To enable the de-bouncing filter, set the **GPIOGDEBOUNCE.Px** to 1b.

To disable this filter, set the **GPIOGDEBOUNCE.Px** to 0b.

15.4.7 GPIOG Drive Strength

Each GPIO pin has configurable drive strength when the ports are configured as a digital output.

See the table below for the settings for GPIOG Drive strength.

Table 15-3 GPIOG Drive Strength

| PGDS.PxDS | DRIVE STRENGTH |
|-----------|----------------|
| 000b | 6mA |
| 001b | 8mA |
| 010b | 11mA |
| 011b | 14mA |
| 100b | 17mA |
| 101b | 20mA |
| 110b | 22mA |
| 111b | 25mA |

The drive strength shown in the table above represents the minimum drive strength for the GPIO pin. For the details on the electrical characteristics of the drive strength, set the device data sheet.

Note that the register configuration for the GPIOG Drive Strength is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

15.4.8 GPIOG Schmitt Trigger

Each GPIO has a Schmitt Trigger that can be enabled when the GPIO is configured as a high-impedance digital input (**GPIOGMODE.Px** = 11b).

The Schmitt Trigger is disabled by default (**PGDS.PxST** = 0b). To enable the input Schmitt Triger, set **PGDS.PxST** = 1b.

Note that the register configuration for the GPIO Schmitt Trigger is located in the System and Clock Control (SCC) system.

The register summary for this can be located in Register Summary.

15.4.9 GPIOG Weak Pull-up and Pull-down

Each GPIO pin has a configurable weak 60k pull-up to VCCIO or pull-down to ground.

To enable the weak pull-up to VCCIO, set **GPIOGPU.Px** to 1b. To disable the pull-up, set **GPIOGPU.Px** to 0b.

To enable the weak pull-down to ground, set the **GPIOGPD.Px** to 1b. To disable the pull-down, set **GPIOGPD.Px** to 0b.

WARNING: Do not configure the weak pull-up and pull-down to be active at the same time. Doing so may cause device damage.

The register summary for this can be located in Register Summary.

15.5 Peripheral IO Mapping

The GPIOG peripheral is connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 15-4 GPIOG Peripheral IO Mapping

| GPIOG SIGNAL | IO PIN |
|---------------------|---------------|
| GPIOG0 | PG0 |
| GPIOG1 | PG1 |
| GPIOG2 | PG2 |
| GPIOG3 | PG3 |
| GPIOG4 | PG4 |
| GPIOG5 | PG5 |
| GPIOG6 | PG6 |
| GPIOG7 | PG7 |

15.6 Register Summary

Table 15-5 GPIOG Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|---------------------------|------------|------------------------------|--------|------------|
| GPIOGMODE | 400D 2C00h | GPIOG Pin Mode Select | RW | 0000 FFFFh |
| GPIOGOUTMASK | 400D 2C04h | GPIOG Data Output Write Mask | RW | 0000 0000h |
| GPIOGOUT | 400D 2C08h | GPIOG Data Output Value | RW | 0000 00FFh |
| GPIOGIN | 400D 2C0Ch | GPIOG Data Input Value | RO | -- |
| GPIOGINTEN | 400D 2C10h | GPIOG Interrupt Enable | RW | 0000 0000h |
| GPIOGINTFLAGRAW | 400D 2C14h | GPIOG Interrupt Flag Raw | RW | 0000 0000h |
| GPIOGINTFLAGMASKED | 400D 2C18h | GPIOG Interrupt Flag Masked | RW | 0000 0000h |
| GPIOGINTCLEAR | 400D 2C1Ch | GPIOG Interrupt Clear | WO | 0000 0000h |
| GPIOGINTTYPE | 400D 2C20h | GPIOG Interrupt Type | RW | 0000 0000h |
| GPIOGINTVALUE | 400D 2C24h | GPIOG Interrupt Value | RW | 0000 0000h |
| GPIOGINTEDGE | 400D 2C28h | GPIOG Interrupt Edge | RW | 0000 0000h |
| GPIOGDEBOUNCE | 400D 2C2Ch | GPIOG De-bounce Filter | RW | 0000 0000h |
| GPIOGDOSET | 400D 2C30h | GPIOG Data Output Set | WO | 0000 0000h |
| GPIOGDOCLEAR | 400D 2C34h | GPIOG Data Output Clear | WO | 0000 0000h |

15.7 Register Detail

15.7.1 GPIOGMODE

Register 15-1 GPIOGMODE (GPIOG Mode Configuration, 400D 2C00h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:14 | P7 | RW | 11b | Pin 7 Mode: 00b: Analog Input (ADC0) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 13:12 | P6 | RW | 11b | Pin 6 Mode: 00b: Analog Input (ADC2) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 11:10 | P5 | RW | 11b | Pin 5 Mode: 00b: Analog Input (ADC1) 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 9:8 | P4 | RW | 11b | Pin 4 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 7:6 | P3 | RW | 11b | Pin 3 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 5:4 | P2 | RW | 11b | Pin 2 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 3:2 | P1 | RW | 11b | Pin 1 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |
| 1:0 | P0 | RW | 11b | Pin 0 Mode: 00b: Reserved 01b: Pull-Pull Output 10b: Open-Drain Output 11b: High-Impedance Input |

15.7.2 GPIOGOUTMASK

Register 15-2 GPIOGOUTMASK (GPIOG Output Mask, 400D 2C04h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Output Data Mask: 0b: Not masked 1b: Masked |
| 6 | P6 | RW | 0 | Pin 6 Output Data Mask: 0b: Not masked 1b: Masked |
| 5 | P5 | RW | 0 | Pin 5 Output Data Mask: 0b: Not masked 1b: Masked |
| 4 | P4 | RW | 0 | Pin 4 Output Data Mask: 0b: Not masked 1b: Masked |
| 3 | P3 | RW | 0 | Pin 3 Output Data Mask: 0b: Not masked 1b: Masked |
| 2 | P2 | RW | 0 | Pin 2 Output Data Mask: 0b: Not masked 1b: Masked |
| 1 | P1 | RW | 0 | Pin 1 Output Data Mask: 0b: Not masked 1b: Masked |
| 0 | P0 | RW | 0 | Pin 0 Output Data Mask: 0b: Not masked 1b: Masked |

15.7.3 GPIOGOUT

Register 15-3 GPIOGOUT (GPIOG Output Data Value, 400E 2C08h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 1 | Pin 7 Output Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 1 | Pin 6 Output Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 1 | Pin 5 Output Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 1 | Pin 4 Output Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 1 | Pin 3 Output Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 1 | Pin 2 Output Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RW | 1 | Pin 1 Output Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RW | 1 | Pin 0 Output Data Value: 0b: Logic low 1b: Logic high |

15.7.4 GPIOGIN

Register 15-4 GPIOFIN (GPIOG Input Data Value, 400D 2C0Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Input Data Value: 0b: Logic low 1b: Logic high |
| 6 | P6 | RO | 0 | Pin 6 Input Data Value: 0b: Logic low 1b: Logic high |
| 5 | P5 | RO | 0 | Pin 5 Input Data Value: 0b: Logic low 1b: Logic high |
| 4 | P4 | RO | 0 | Pin 4 Input Data Value: 0b: Logic low 1b: Logic high |
| 3 | P3 | RO | 0 | Pin 3 Input Data Value: 0b: Logic low 1b: Logic high |
| 2 | P2 | RO | 0 | Pin 2 Input Data Value: 0b: Logic low 1b: Logic high |
| 1 | P1 | RO | 0 | Pin 1 Input Data Value: 0b: Logic low 1b: Logic high |
| 0 | P0 | RO | 0 | Pin 0 Input Data Value: 0b: Logic low 1b: Logic high |

15.7.5 GPIOGINTEN

Register 15-5 GPIOGINTEN (GPIOG Interrupt Enable, 400D 2C10h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Enable : 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Enable: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Enable: 0b: Disabled 1b: Enabled |

15.7.6 GPIOGINTFLAGRAW

Register 15-6 GPIOGINTFLAGRAW (GPIOG Interrupt Flag Raw, 400D 2C14h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag: 0b: No flag 1b: Flag |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag: 0b: No flag 1b: Flag |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag: 0b: No flag 1b: Flag |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag: 0b: No flag 1b: Flag |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag: 0b: No flag 1b: Flag |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag: 0b: No flag 1b: Flag |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag: 0b: No flag 1b: Flag |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag: 0b: No flag 1b: Flag |

15.7.7 GPIOGINTFLAGMASKED

Register 15-7 GPIOGINTFLAGMASKED (GPIOG Interrupt Flag Masked, 400D 2C18h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RO | 0 | Pin 7 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 6 | P6 | RO | 0 | Pin 6 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 5 | P5 | RO | 0 | Pin 5 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 4 | P4 | RO | 0 | Pin 4 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 3 | P3 | RO | 0 | Pin 3 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 2 | P2 | RO | 0 | Pin 2 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 1 | P1 | RO | 0 | Pin 1 Interrupt Flag Masked: 0b: Not masked 1b: Masked |
| 0 | P0 | RO | 0 | Pin 0 Interrupt Flag Masked: 0b: Not masked 1b: Masked |

15.7.8 GPIOINTCLEAR

Register 15-8 GPIOINTCLEAR (GPIO Interrupt Clear, 400D 2C1Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P7 |
| 6 | P6 | WO | 0 | Pin 6 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P6 |
| 5 | P5 | WO | 0 | Pin 5 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P5 |
| 4 | P4 | WO | 0 | Pin 4 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P4 |
| 3 | P3 | WO | 0 | Pin 3 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P3 |
| 2 | P2 | WO | 0 | Pin 2 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P2 |
| 1 | P1 | WO | 0 | Pin 1 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P1 |
| 0 | P0 | WO | 0 | Pin 0 Interrupt Clear: 0b: No effect 1b: Clear GPIOINTFLAGRAW.P0 |

15.7.9 GPIOGINTTYPE

Register 15-9 GPIOGINTTYPE (GPIOG Interrupt Type Configuration, 400D 2C20h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Type: 0b: Edge triggered 1b: Level triggered |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Type: 0b: Edge triggered 1b: Level triggered |

15.7.10 GPIOINTCFG

Register 15-10 GPIOINTCFG (GPIO Interrupt Configuration, 400D 2C24h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 7 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 7 Interrupt Level: 0b: Logic low 1b: Logic high |
| 6 | P6 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 6 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 6 Interrupt Level: 0b: Logic low 1b: Logic high |
| 5 | P5 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 5 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 5 Interrupt Level: 0b: Logic low 1b: Logic high |
| 4 | P4 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 4 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 4 Interrupt Level: 0b: Logic low 1b: Logic high |
| 3 | P3 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 3 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 3 Interrupt Level: 0b: Logic low 1b: Logic high |
| 2 | P2 | RW | 0 | If GPIOINTTYPE.Px = 0b (edge), Pin 2 Interrupt Edge: 0b: Trigger on falling edge 1b: Trigger on rising edge If GPIOINTTYPE.Px = 1b (level), Pin 2 Interrupt Level: 0b: Logic low 1b: Logic high |

| | | | | |
|---|-----------|----|---|---|
| 1 | P1 | RW | 0 | <p>If GPIOGINTTYPE.Px = 0b (edge), Pin 1 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOGINTTYPE.Px = 1b (level), Pin 1 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |
| 0 | P0 | RW | 0 | <p>If GPIOGINTTYPE.Px = 0b (edge), Pin 0 Interrupt Edge:</p> <p>0b: Trigger on falling edge 1b: Trigger on rising edge</p> <p>If GPIOGINTTYPE.Px = 1b (level), Pin 0 Interrupt Level:</p> <p>0b: Logic low 1b: Logic high</p> |

15.7.11 GPIOINTEDGEBOTH

Register 15-11 GPIOINTEDGEBOTH (GPIOG Interrupt Both Edge Configuration, 400D 2C28h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P7 1b: Trigger on both rising and falling edge |
| 6 | P6 | RW | 0 | Pin 6 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P6 1b: Trigger on both rising and falling edge |
| 5 | P5 | RW | 0 | Pin 5 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P5 1b: Trigger on both rising and falling edge |
| 4 | P4 | RW | 0 | Pin 4 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P4 1b: Trigger on both rising and falling edge |
| 3 | P3 | RW | 0 | Pin 3 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P3 1b: Trigger on both rising and falling edge |
| 2 | P2 | RW | 0 | Pin 2 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P2 1b: Trigger on both rising and falling edge |
| 1 | P1 | RW | 0 | Pin 1 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P1 1b: Trigger on both rising and falling edge |
| 0 | P0 | RW | 0 | Pin 0 Interrupt Both Edge Configuration: 0b: Trigger on rising or falling edge according to GPIOINTCFG.P0 1b: Trigger on both rising and falling edge |

15.7.12 GPIOGDEBOUNCE

Register 15-12 GPIOFDEBOUNCE (GPIOG De-bounce Enable, 400D 2C2Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | RW | 0 | Pin 7 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 6 | P6 | RW | 0 | Pin 6 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 5 | P5 | RW | 0 | Pin 5 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 4 | P4 | RW | 0 | Pin 4 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 3 | P3 | RW | 0 | Pin 3 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 2 | P2 | RW | 0 | Pin 2 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 1 | P1 | RW | 0 | Pin 1 De-bounce Enabled: 0b: Disabled 1b: Enabled |
| 0 | P0 | RW | 0 | Pin 0 De-bounce Enabled: 0b: Disabled 1b: Enabled |

15.7.13 GPIOGDOSSET

Register 15-13 GPIOGDOSSET (GPIOG Data Output Set, 400D 2C30h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P7 to 1b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P6 to 1b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P5 to 1b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P4 to 1b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P3 to 1b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P2 to 1b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P1 to 1b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P0 to 1b |

15.7.14 GPIOGDOCLEAR

Register 15-14 GPIOGDOCLEAR (GPIOG Data Output Clear, 400D 2C34h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | P7 | WO | 0 | Pin 7 Data Output Clear: 0b: No effect 1b: Set GPIOGOUT.P7 to 0b |
| 6 | P6 | WO | 0 | Pin 6 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P6 to 0b |
| 5 | P5 | WO | 0 | Pin 5 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P5 to 0b |
| 4 | P4 | WO | 0 | Pin 4 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P4 to 0b |
| 3 | P3 | WO | 0 | Pin 3 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P3 to 0b |
| 2 | P2 | WO | 0 | Pin 2 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P2 to 0b |
| 1 | P1 | WO | 0 | Pin 1 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P1 to 0b |
| 0 | P0 | WO | 0 | Pin 0 Data Output Set: 0b: No effect 1b: Set GPIOGOUT.P0 to 0b |

16 GPTIMERA

16.1 Overview

General Purpose Timer A (GPTimer A) is a general-purpose system timer.

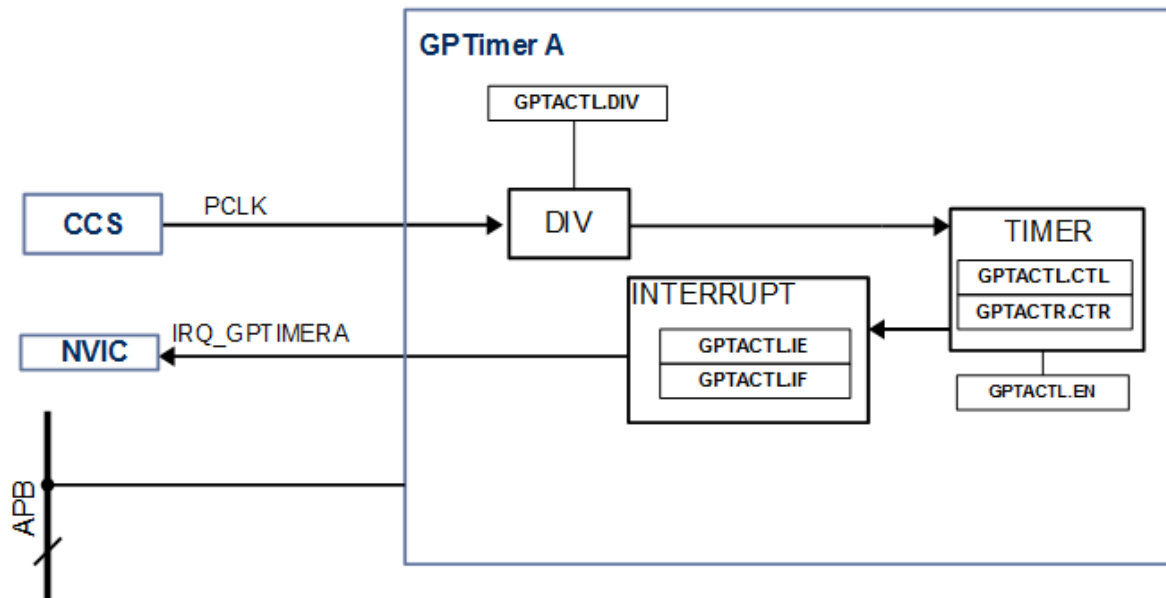
GPTimer A is an APB client.

16.2 Features

- 24-bit Count-down Timer
- Auto-reload
- Interrupt
- User-configurable count-down value
- Input clock Divider

16.3 System Block Diagram

Figure 16-1 GP Timer A System Block Diagram



16.4 Functional Description

GPTimer A is a general-purpose, 24-bit count-down system timer.

GPTimer A is an APB peripheral and the input clock is PCLK. This timer has a user-configurable clock divider that may be set as follows.

Table 16-1 GPTimer A Input Clock Divider

| GPTACTL.DIV | GP Timer A Clock |
|-------------|------------------|
| 0000b | PCLK /1 |
| 0001b | PCLK /2 |
| 0010b | PCLK /4 |
| 0011b | PCLK /8 |
| ... | ... |
| 1110b | PCLK /16384 |
| 1111b | PCLK /32768 |

GPTimer A may be enabled by setting **GPTACTL.EN** to 1b. When enabled, the timer will then copy the value of **GPTACTL.CDV** to **GPTACTR.CTR** and begin counting down.

When timer counts down and reaches 0, it will set **GPTACTL.IF** to a 1b. If the **GPTACTL.IE** is set to 1b, then the timer will assert the IRQ_GPTIMERA signal to the NVIC.

16.5 Register Summary

Table 16-2 GPTimer A Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------------|------------|----------------------------|--------|------------|
| GPTACTL | 400B 0000h | GPTimer A Control | RW | 000F FF00h |
| GPTACTR | 400B 0004h | GPTimer A Count-down Value | RO | 00FF FFFFh |

16.6 Register Detail

16.6.1 GPTACTL

Register 16-1 GPTACTL (GPTimer A Control, 400B 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | CDV | RW | FFFh | Count-down value. |
| 7 | Reserved | RO | 0 | Reserved |
| 6:3 | DIV | RW | 0 | GPTimer A Clock Divider: 0000b: PCLK /1 0001b: PCLK /2 0010b: PCLK /4 0011b: PCLK /8 ... 1110b: PCLK /16384 1111b: PCLK /32768 |
| 2 | IF | W1C | 0 | Interrupt flag: 0b: no flag 1b: flag |
| 1 | IE | RW | 0 | Interrupt enable: 0b: not enabled 1b: enabled |
| 0 | EN | RW | 0 | Timer Enabled: 0b: disabled 1b: enabled |

16.6.2 GPTACTR

Register 16-2 GPTACTR (GPTimer A Counter, 400B 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|----------|----------------|
| 31:24 | Reserved | RO | 0 | Reserved |
| 23:0 | CTR | RW | FF FFFFh | Counter value. |

17 GPTIMERB

17.1 Overview

General Purpose Timer B (GPTimer B) is a general-purpose system timer.

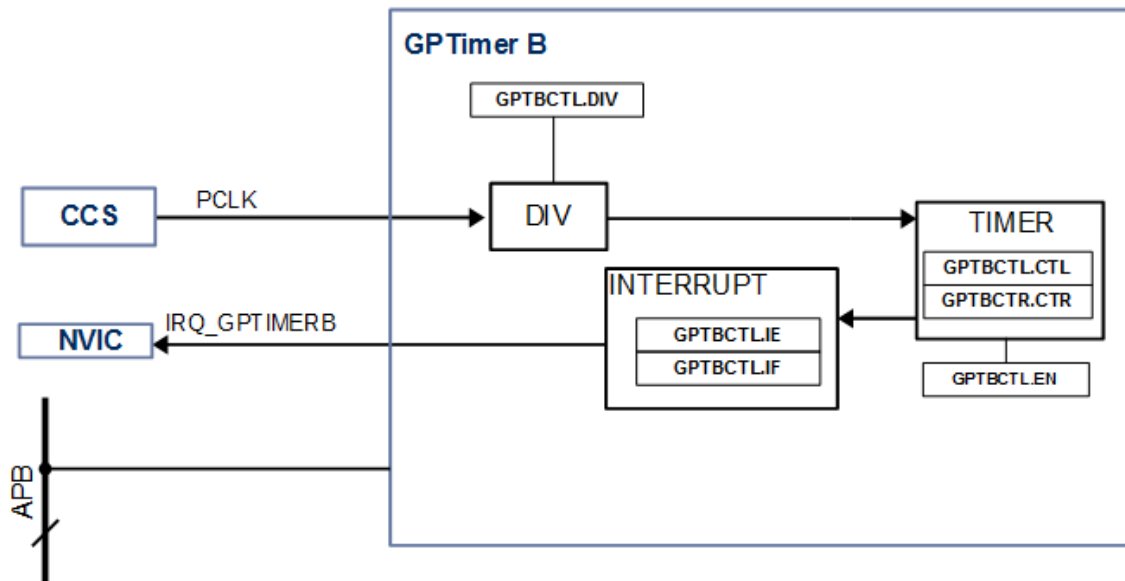
GPTimer B is an APB client.

17.2 Features

- 24-bit Count-down Timer
- Auto-reload
- Interrupt
- User-configurable count-down value
- Input clock Divider

17.3 System Block Diagram

Figure 17-1 GP Timer B System Block Diagram



17.4 Functional Description

GPTimer B is a general-purpose, 24-bit count-down system timer.

GPTimer B is an APB peripheral and the input clock is PCLK. This timer has a user-configurable clock divider that may be set as follows.

Table 17-1 GPTimer B Input Clock Divider

| GPTACTL.DIV | GP Timer A Clock |
|-------------|------------------|
| 0000b | PCLK /1 |
| 0001b | PCLK /2 |
| 0010b | PCLK /4 |
| 0011b | PCLK /8 |
| ... | ... |
| 1110b | PCLK /16384 |
| 1111b | PCLK /32768 |

GPTimer B may be enabled by setting **GPTBCTL.EN** to 1b. When enabled, the timer will then copy the value of **GPTBCTL.CDV** to **GPTBCTR.CTR** and begin counting down.

When timer counts down and reaches 0, it will set **GPTBCTL.IF** to a 1b. IF the **GPTBCTL.IE** is set to 1b, then the timer will assert the IRQ_GPTIMERB signal to the NVIC.

17.5 Register Summary

Table 17-2 GPTimer B Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------------|------------|----------------------------|--------|------------|
| GPTBCTL | 400C 0000h | GPTimer B Control | RW | 000F FF00h |
| GPTBCTR | 400C 0004h | GPTimer B Count-down Value | RO | 00FF FFFFh |

17.6 Register Detail

17.6.1 GPTBCTL

Register 17-1 GPTBCTL (GPTimer B Control, 400C 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | CDV | RW | 0 | Count-down value. |
| 7 | Reserved | RO | 0 | Reserved |
| 6:3 | DIV | RW | 0 | GPTimer B Clock Divider: 0000b: PCLK /1 0001b: PCLK /2 0010b: PCLK /4 0011b: PCLK /8 ... 1110b: PCLK /16384 1111b: PCLK /32768 |
| 2 | IF | W1C | 0 | Interrupt flag: 0b: no flag 1b: flag |
| 1 | IE | RW | 0 | Interrupt enable: 0b: not enabled 1b: enabled |
| 0 | EN | RW | 0 | Timer Enabled: 0b: disabled 1b: enabled |

17.6.2 GPTBCTR

Register 17-2 GPTBCTR (GPTimer B Counter, 400C 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|----------|----------------|
| 31:24 | Reserved | RO | 0 | Reserved |
| 23:0 | CTR | RW | FF FFFFh | Counter value. |

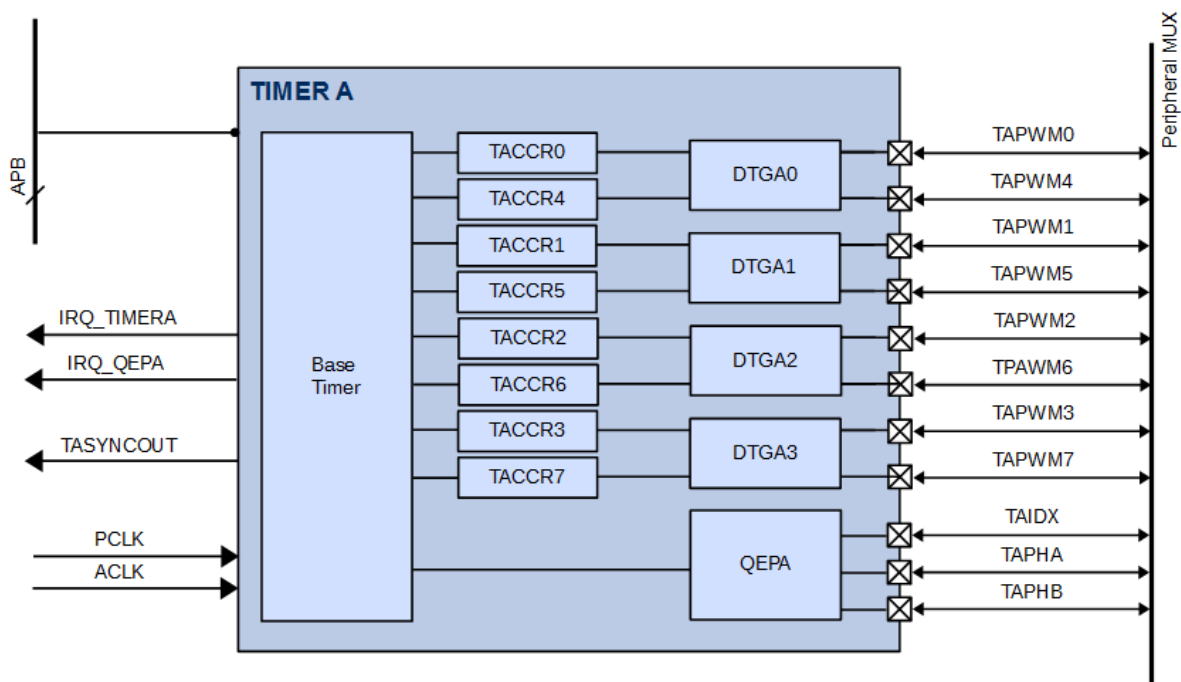
18 PWM TIMER A

18.1 Overview

All devices in the PAC55XX family of controllers have a Timer A peripheral. This peripheral is a 16-bit timer that allows support for 8 Capture and Compare Units (CCR) capable of PWM generation; capture input processing and a QEP encoder for various control applications.

Below is a simplified block diagram of the Timer peripheral.

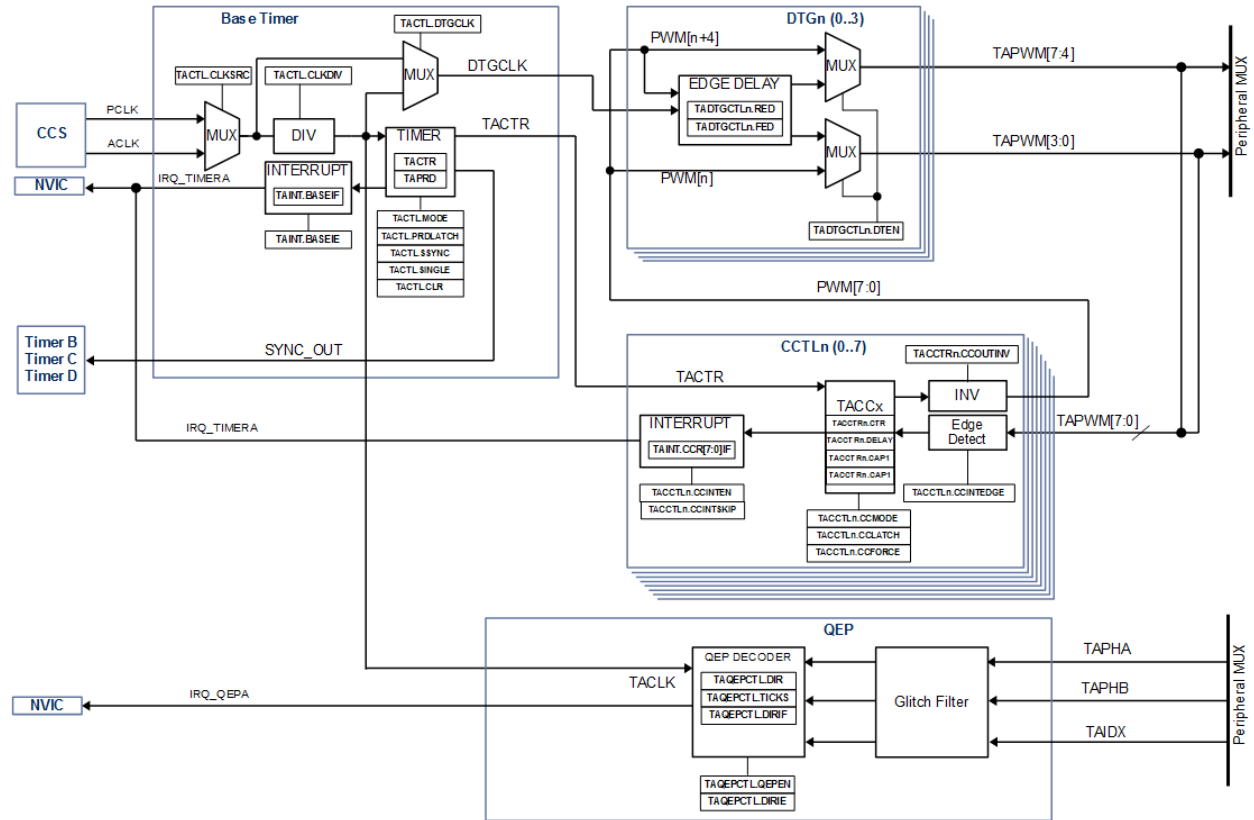
Figure 18-1 Timer A Simplified Block Diagram



The timer module is an APB bus client. There are 8 timer channel input/outputs to the digital peripheral MUX, in addition to three IOs for QEP. Each timer block has two interrupt signal outputs to the NVIC; one for timer functions and one for QEP functions. The user may select either PCLK or ACLK as the clock input for each timer module.

18.2 Timer A Block Diagram

Figure 18-2 Timer A Block Diagram



18.3 Features

The PWM Timer peripheral has the following features in the PAC55XX family of controllers.

General Features:

- Configurable input clock: PCLK or ACLK
- Up to 300MHz clock input for 3.33ns PWM edge resolution
- 3-bit input clock divider
- Latch timer period and all CCR values on command

Base Timer Features:

- Single-shot or auto-reload
- Base timer interrupts
- Timer synchronization
- Timer Modes:
 - Disabled
 - Up mode
 - Up/Down mode
 - Asymmetric Up mode
- Timer Register Latching Options
 - Latch TAPRD when counter = 0
 - Latch TAPRD when counter = period
 - Latch TAPRD immediately
- Dead-time Generator Input Clock
 - DTG clock = PCLK
 - DTG clock = ACLK

CCR/PWM Features:

- PWM output or input capture
- 8 CCR units per timer
- CCR interrupts
- CCR interrupt skips
- SW force CCR interrupt
- CCR interrupt type:
 - Rising, falling, both
- CCR latch modes:
 - Compare/PWM mode: counter = 0, counter = period, immediate
 - Capture input: rising edge, falling edge, both
- Period/duty calculation mode
- Force compare event

- Invert CCR output
- CCR phase delay for phase shifted drive topologies
- ADC trigger outputs:
 - PWM rising or falling edge

Dead-Time Generator (DTG) Features:

- DTG enable or bypass
- 12-bit rising edge delay
- 12-bit falling edge delay

QEP Decoder Features:

- QEP encoder enabled
- Direction status
- Configurable interrupts:
 - Phase A rising edge
 - Phase B rising edge
 - Index event
 - Counter wrap
- 4 Different counting modes for best resolution, range and speed performance

18.4 Functional Description

18.4.1 Timer Clock Structure

The timer peripheral input clock can be selected as either PCLK or ACLK. The input clock may be selected by the **TACTL.CLKSRC** register.

Each timer peripheral has a 4-bit divider that can be used to divide the selected input clock. The user may set this divider by the **TACTL.CLKDIV** register. There are 16 settings between /1 and /128 for the input clock divider.

The base timer also supplies the clock for the Dead-Time Generators (DTGs) to allow for different range and resolution for dead-time. The timer may select the DTGCLK to be the timer clock before or after the clock divider. The DTG input clock may be selected by the **TACTL.DTGCLK** register.

To use the timer clock before the input clock divider, set **TACTL.DTGCLK** to 0b. To use the timer clock after the input clock divider, set **TACTL.DTGCLK** to 1b.

18.4.2 Timer Counter

The base timer is a 16-bit timer that can count either up, or up then down to support both edge and center aligned and asymmetric PWM output types.

The timer period is stored in the **TAPRD** register. The current value of the timer counter is updated at every timer tick and is stored in the **TACTR** register. The **TACTR** register is a RW register, so it may be updated at any time and is changed immediately.

The **TAPRD** register and all of the **TACCTRn** registers have shadow copies that are updated at a user-specified time. See the section below on Base Timer TAPRD Latching and CCR Timer Latching for more information.

When **TACTL.MODE** = 00b, the timer is disabled. Even when disabled, all timer registers are accessible via the APB bus.

To enable the timer, set the **TACTL.MODE** to 01b (up mode), 10b (up/down mode) or 11b (up/down asymmetric mode).

When **TACTL.MODE** = 01b, the timer is configured in up mode. The timer will count from 0 to **TAPRD**. The counter is updated in **TACTR** at every timer tick. If auto-reload is active (**TACTL.SINGLE** = 0b), then the timer count will automatically count from **TAPRD** to 0 and continue counting up. If auto-reload is not active (**TACTL.SINGLE** = 1b), then the timer will count from 0 to **TAPRD** and stop. When this happens, the timer will set the **TACTL.MODE** to 00b (disabled).

When the timer is configured for up/down mode (**TACTL.MODE** = 10b) or configured for up/down asymmetric mode (**TACTL.MODE** = 11b), the timer will count from 0 to **TAPRD** and then back down to 0. If auto-reload is not active (**TACTL.SINGLE** = 1b), the time timer will stop

and set **TACTL.MODE** to 00b (disabled). If auto-reload is active (**TACTL.SINGLE** = 0b), then the timer count will then continue counting back up after it reaches 0.

18.4.3 Up Mode

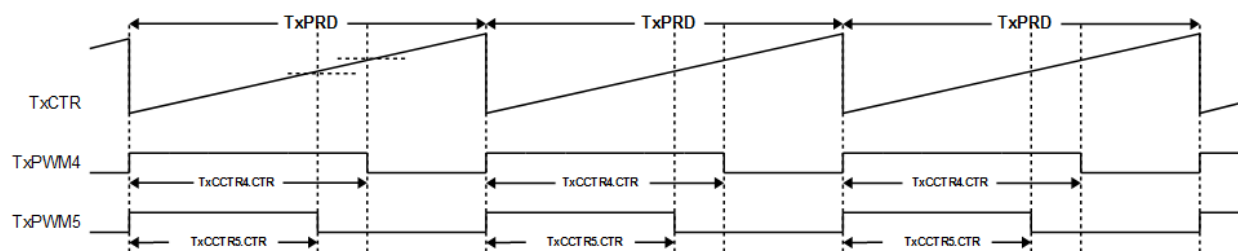
When the **TACTL.MODE** is set to 01b (up mode) the timer will begin counting up from 0. The **TACTR** will count to the value of **TAPRD** and then will reset to 0.

If the timer is configured for auto-reload mode (**TACTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it reaches 0.

The **TACTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate an edge aligned PWM as shown below.

Figure 18-3 Up Mode PWM Waveform



In this mode, when **TACTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TACTR** register reaches the **TACCTRn.CTR** threshold. When the **TACTR** counts from **TAPRD** to 0, the PWM output is transitioned to high.

18.4.4 Up/Down Mode

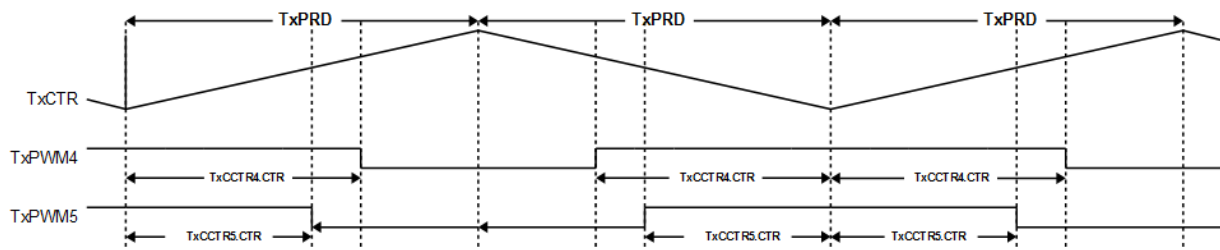
When the **TACTL.MODE** is set to 10b (up/down mode) the timer will begin counting up from 0. The **TACTR** will count to the value of **TAPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TACTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TACTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned PWM as shown below.

Figure 18-4 Up/Down Mode PWM Waveform



In this mode, when **TACTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TACTR** register reaches the **TACCTRn.CTR** threshold. When the **TACTR** counts from **TAPRD - 1** to **TAPRD**, it will begin counting down. When the **TACTR** counts down to **TACCTRn.CTR** the PWM output is transitioned to high.

18.4.5 Up/Down Asymmetric mode

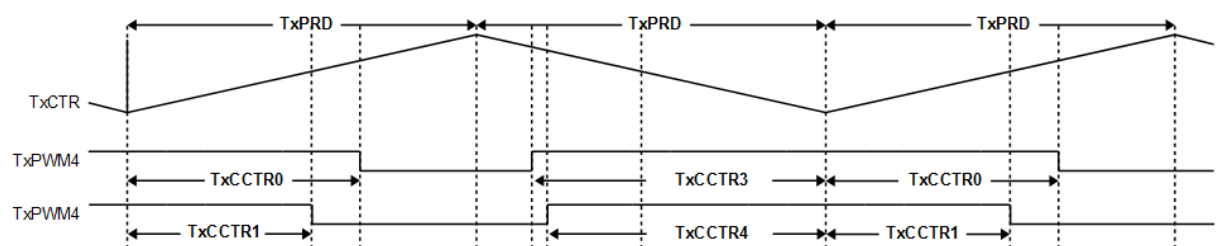
When the **TACTL.MODE** is set to 11b (up/down asymmetric mode) the timer will begin counting up from 0. The **TACTR** will count to the value of **TAPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TACTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TACTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned asymmetric PWM as shown below.

Figure 18-5 Up/Down Asymmetric Mode PWM Waveform



In this mode, when **TACTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TACTR** register reaches the **TACCTRn.CTR** threshold. When the **TACTR** counts from **TAPRD - 1** to **TAPRD**, it will begin counting down. When the **TACTR** counts down to **TACCTRn.CTR** the PWM output is transitioned to high.

In this mode, the CCR outputs are generated such that 2 **TACCTRn.CTR** registers are used to generate a single PWM output, so that the center-aligned PWM can be generated with asymmetric on-time.

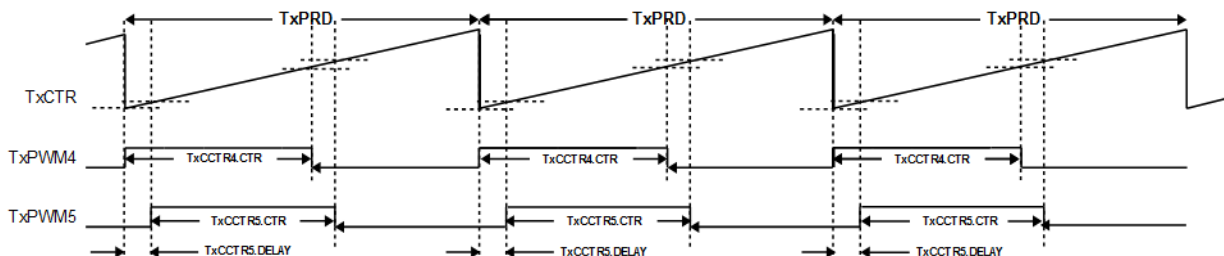
In this mode, the PWM outputs of the timer are generated as follows:

- TAPWM4 uses **TACCTR0.CTR** for the first portion of the on time and **TACCTR4.CTR** for the second portion of the on time
- TAPWM5 uses **TACCTR1.CTR** for the first portion of the on time and **TACCTR5.CTR** for the second portion of the on time
- TAPWM6 uses **TACCTR2.CTR** for the first portion of the on time and **TACCTR6.CTR** for the second portion of the on time
- TAPWM7 uses **TACCTR3.CTR** for the first portion of the on time and **TACCTR7.CTR** for the second portion of the on time

18.4.6 Up Mode with Phase Delay

When the timer is configured for up mode (**TAMODE** = 01b) the CCR allows a delay to be applied to the output PWM period, to support phase delay control topologies. If the value of **TACCTLn.DELAY** > 0, then the CCR will delay this number of ticks before starting the PWM period (transitioning the output signal from high to low).

Figure 18-6 Up Mode with Phase Delay



18.4.7 Timer Synchronization

The timer peripheral allows synchronization between the timers in the PAC55XX. The timers may be synchronized as follows:

- Timer A (master), Timer B (slave)
- Timer A (master), Timer B (slave), Timer C (slave)
- Timer A (master), Timer B (slave), Timer C (slave), Timer D (slave)

The timer master's SYNC_OUT signals are connected to the downstream timer slave's SYNC_IN signal.

When configured, the timers can synchronize their clocks so they are sharing the same time base. In order to do this, the timers must be configured with the same clock source (**TACTL.CLKSRC**) and clock divider (**TACTL.CLKDIV**).

To configure the timers for synchronization, follow these steps:

- While the timer master is disabled **TACTL.MODE** = 00b, set the timer master **TACTL.SSYNC** = 0b
- For each of the downstream timer slaves, while the timer is disabled **TACTL.MODE** = 00b, set the **TACTL.SSYNC** to 1b. Make sure that **TACTL.CLKDIV** and **TACTL.CLKSRC** for each of the timer slaves is the same as the timer master.
- Enable each of the slave timers by setting **TACTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric). The slave timers will not start counting since the **TACTL.SSYNC** is set to a 1b.
- Enable the timer master by setting **TACTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric).

At this point, the master and all configured timer slaves will begin counting.

18.4.8 Base Timer TAPRD Latching

The timer peripherals have a shadow copy of the TAPRD register that is used for the counting operations. The latching of the data from the TAPRD register to the shadow register is controlled by the timer configuration.

There are several options for latching of this register into the shadow copy. This behavior is controlled by the **TACTL.PRDLATCH** setting and depends on the setting of the **TACTL.MODE** register as well.

18.4.8.1 TAPRD Latch When TACTR = 0

- If **TACTL.MODE** = 01b (up mode) and **TACTL.PRDLATCH** = 00b (**TACTR** = 0), then the **TAPRD** register is copied into the shadow register when **TACTR** counts from **TAPRD** to 0.
- If **TACTL.MODE** = 10b (up/down mode) or if **TACTL.MODE** = 11b (up/down asymmetric mode) and **TACTL.PRDLATCH** = 00b (**TACTR** = 0), then the **TAPRD** register is copied into the shadow register when the **TACTR** counts from 1 to 0.

18.4.8.2 TAPRD Latch When TACTR = TAPRD

- If **TACTL.MODE** = 01b (up mode) or if **TACTL.MODE** = 10b (up/down mode) or if **TACTL.MODE** = 11b (up/down asymmetric mode) and **TACTL.PRDLATCH** = 01b (**TACTR** = **TAPRD**), then the **TAPRD** register is copied into the shadow register when **TACTR** counts from **TAPRD** – 1 to **TAPRD**.

18.4.8.3 TAPRD Latch Immediate

- If **TACTL.PRDLATCH** = 11b (immediate), then the **TAPRD** register is copied into the shadow register as soon as the **TAPRD** register is written.

18.4.9 CCR Timer Latching

When in compare mode, the CCR units support configurable latching of the **TACCRn.CTR** values into a shadow register that is used for timer operation. The shadow register is used for generating the PWM output when the CCR is in compare mode.

This feature can be used to control when the new duty cycle is applied, during the PWM period.

When the CCR is in compare mode (**TACCTLn.CCMODE** = 0b), the shadow register is latched as follows:

- If **TACCTLn.CCLATCH** = 00b (**TACTR** = 0)
 - If **TACTL.MODE** = 01b (up mode), **TACCTRn.CTR** is copied into the shadow register when **TACTR** counts from **TAPRD** to 0. If **TACTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode), **TACCTRn.CTR** is copied into the shadow register when **TACTR** counts from 1 to 0.
- If **TACCTLn.CCLATCH** = 01b (**TACTR** = **TAPRD**), then the **TACCTRn.CTR** is copied into the shadow register when **TACTR** counts from **TAPRD** – 1 to **TAPRD**.
- If **TACCTLn.CCLATCH** = 10b (latch immediate), then the **TACCTRn.CTR** is copied into the shadow register as soon as it is written.

When the CCR is in capture mode (**TACCTLn.CCMODE** = 1b), the configuration of **TACCTLn.CCLATCH** controls when the **TACTR** is copied into the **TACCTRn.CTR** register.

- If **TACCTLn.CCLATCH** = 00b (rising edge), then the value of **TACTR** is copied into the **TACCTL.CTR** register upon a rising edge in the input signal.
- If **TACCTLn.CCLATCH** = 01b (falling edge), then the value of the **TACTR** is copied into the **TACCTLn.CTR** register upon a falling edge in the input signal.
- If **TACCTLn.CCLATCH** = 10b (both), then the value of the **TACTR** is copied into the **TACCTLn.CTR** register on both a rising and falling edge of the input signal.
- If **TACCTLn.CCLATCH** = 11b (period/duty mode), then see the section below on how the values are latched into the **TACCTRn** register.

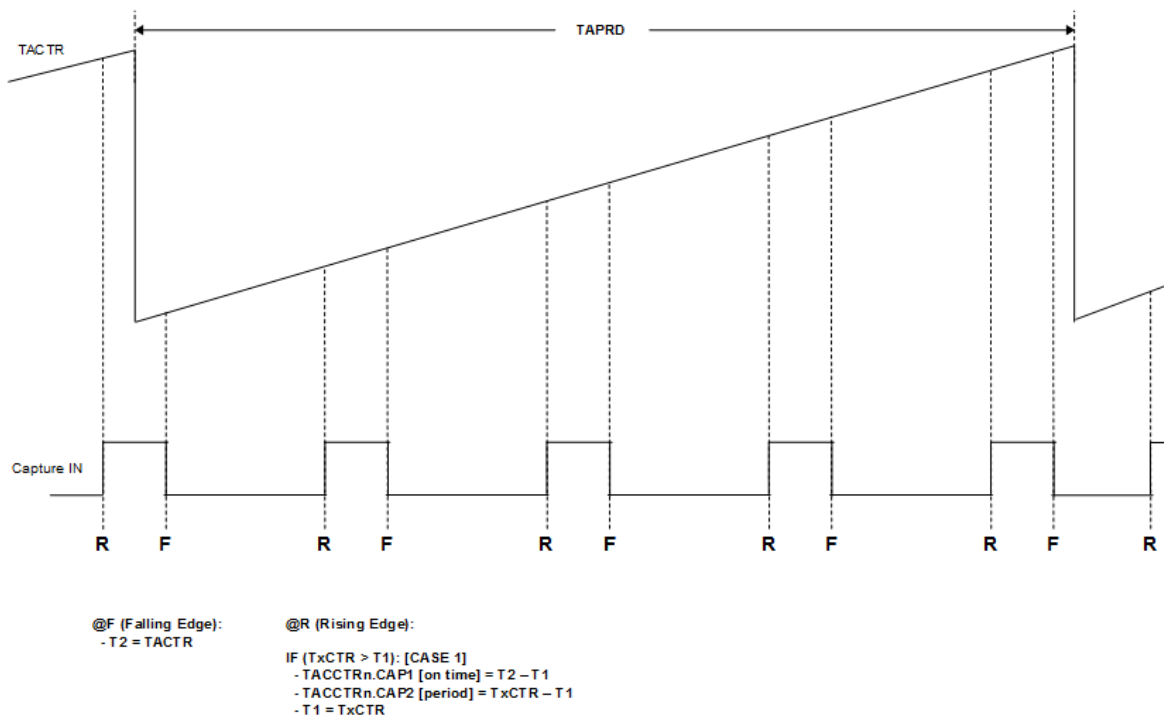
18.4.10 CCR Capture Period/Duty Mode

If the **TACCTLn.CCLATCH = 11b** (period/duty mode), then the CCR can capture the number of timer ticks for the input PWM duty cycle and period. This mode is useful for measuring input pulse widths for PPM and other type of control applications.

At each rising edge of the input signal, the following happens:

- The number of timer ticks for the last timer period's duty cycle is copied into **TACCTRn.CAP1**
- The number of timer ticks for the last timer period is copied into **TACCTRn.CAP2**

Figure 18-7 Period/Duty Capture Mode PWM Waveform



18.4.11 Timer Whole Latching

It is sometimes convenient to latch both the **TAPRD** (period) and all **TACCTRn.CTR** (duty cycle) registers into the shadow registers at one time.

If the **TACTL.LATCH** bit is written to a 1b, then all the **TAPRD** and all **TACCTRn.CTR** registers will be copied into the shadow registers on the same clock cycle.

The **TACTL.LATCH** bit is self-clearing, and will always be read as a 0b.

18.4.12 Inverting CCR PWM Output

When the CCR is in compare mode (**TACCTRn.CCMODE** = 0b), the output of the CCR (the input to the DTG) may be inverted. To invert this output, set the **TACCTRn.CCOUTINV** = 1b.

This mode is useful for some control topologies, especially full-bridge.

18.4.13 Base Timer Interrupts

The base timer may be configured to generate an interrupt to the NVIC.

If **TACTL.MODE** = 01b (up mode) and **TACTR** counts from **TAPRD** to 0 then the **TAINT.BASEIF** bit is set to 1b. If the **TACTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMER_IRQ** signal to the NVIC is asserted.

If **TACTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode) and **TACTR** counts from 1 to 0, then the **TAINT.BASEIF** bit is set to 1b. If the **TACTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERA_IRQ** signal to the NVIC is asserted.

The **TAINT.BASEIF** may be cleared by writing a 1b to it.

18.4.14 CCR Compare Interrupts

When **TACCTLn.CCMODE** = 0b (compare mode), the CCR unit may be configured to generate an interrupt to the NVIC.

When **TACTR** counts to **TACCTRn.CTR**, then the CCR interrupt flag in the **TAINT** register is set as follows:

- For CCR0, when **TACTR** counts to **TACCTR0.CTR**, then **TAINT.CCR0IF** is set to a 1b. If the **TACCTR0.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR0IF** may be cleared by writing it to a 1b.
- For CCR1, when **TACTR** counts to **TACCTR1.CTR**, then **TAINT.CCR1IF** is set to a 1b. If the **TACCTR1.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR1IF** may be cleared by writing it to a 1b.
- For CCR2, when **TACTR** counts to **TACCTR2.CTR**, then **TAINT.CCR2IF** is set to a 1b. If the **TACCTR2.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR2IF** may be cleared by writing it to a 1b.

- For CCR3, when **TACTR** counts to **TACCTR3.CTR**, then **TAINT.CCR3IF** is set to a 1b. If the **TACCTR3.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR3IF** may be cleared by writing it to a 1b.
- For CCR4, when **TACTR** counts to **TACCTR4.CTR**, then **TAINT.CCR4IF** is set to a 1b. If the **TACCTR4.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR4IF** may be cleared by writing it to a 1b.
- For CCR5, when **TACTR** counts to **TACCTR5.CTR**, then **TAINT.CCR5IF** is set to a 1b. If the **TACCTR5.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR5IF** may be cleared by writing it to a 1b.
- For CCR6, when **TACTR** counts to **TACCTR6.CTR**, then **TAINT.CCR6IF** is set to a 1b. If the **TACCTR6.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR6IF** may be cleared by writing it to a 1b.
- For CCR7, when **TACTR** counts to **TACCTR7.CTR**, then **TAINT.CCR7IF** is set to a 1b. If the **TACCTR7.CCINTEN** is set to 1b, then **TIMER_IRQ** to the NVIC is asserted. **TAINT.CCR7IF** may be cleared by writing it to a 1b.

18.4.15 CCR Capture Interrupts

When **TACCTLn.CCMODE** = 1b (capture mode), the CCR unit may be configured to generate an interrupt to the NVIC based on the input signal.

- If **TACCTLn.CCEGE** = 00b (rising edge) and a rising edge is detected in the input signal, then the **TAINT.CCRnIF** interrupt flag is set.
- If **TACCTLn.CCEDGE** = 01b (falling edge) and a falling edge is detected in the input signal, then the **TAINT.CCRnIF** interrupt flag is set.
- If the **TACCTLn.CCEDGE** = 11b (rising or falling edge) and a rising or a falling edge is detected in the input signal, then the **TAINT.CCRnIF** interrupt flag is set.

If the **TAINT.CCRnIF** interrupt flag is set and the corresponding **TACCTLn.CCINTEN** bit is set, then the **TIMERA_IRQ** signal to the NVIC will be asserted.

The **TAINT.CCRnIF** interrupt flag can be cleared by writing it to a 1b.

18.4.16 Timer IRQ signal

The timer unit has one IRQ output signal to the NVIC (**TIMERA_IRQ**).

This signal is asserted when any of the following conditions are true:

- **TAINT.BASEIF** = 1b and **TACTL.BASEIE** = 1b
- **TAINT.CCR0IF** = 1b and **TACCTL0.CCINTEN** = 1b
- **TAINT.CCR1IF** = 1b and **TACCTL1.CCINTEN** = 1b
- **TAINT.CCR2IF** = 1b and **TACCTL2.CCINTEN** = 1b
- **TAINT.CCR3IF** = 1b and **TACCTL3.CCINTEN** = 1b
- **TAINT.CCR4IF** = 1b and **TACCTL4.CCINTEN** = 1b

- **TAIN.TCCR5IF** = 1b and **TACCTL5.CCINTEN** = 1b
- **TAIN.TCCR6IF** = 1b and **TACCTL6.CCINTEN** = 1b
- **TAIN.TCCR7IF** = 1b and **TACCTL7.CCINTEN** = 1b

The TIMERA_IRQ signal is de-asserted when all of the above conditions are false.

18.4.17 Skipping CCR Interrupts

Sometimes it is useful to not generate a CCR interrupt to the NVIC every time the **TACTR** counts to the **TACCTRn.CTR** value. For example, in a control application when the PWM frequency is fast, but you only need a MCU interrupt every 5 PWM periods.

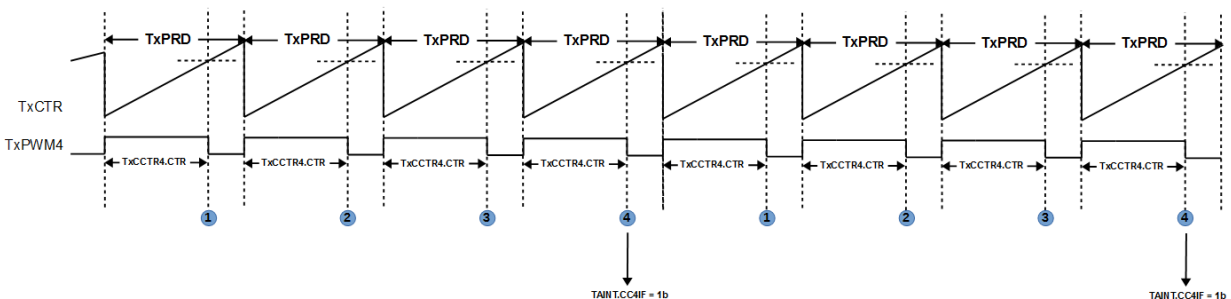
Each CCR in the timer has the ability to skip interrupts to the NVIC to allow this capability.

The CCR will generate interrupts to the NVIC according to the **TACCTL.CCINTSKIP** field as follows:

- If **TACCTLn.CCINTSKIP** = 0000b, then the CCR will generate an interrupt each time
- If **TACCTLn.CCINTSKIP** = 0001b, then the CCR will skip 1 CCR cycle before generating an interrupt to the NVIC.
- If **TACCTLn.CCINTSKIP** = 0010b, then the CCR will skip 2 CCR cycles before generating an interrupt to the NVIC.
- If **TACCTLn.CCINTSKIP** = 0011b, then the CCR will skip 3 CCR cycles before generating an interrupt to the NVIC.
- ...
- If **TACCTLn.CCINTSKIP** = 1110b, then the CCR will skip 14 CCR cycles before generating an interrupt to the NVIC.
- If **TACCTLn.CCINTSKIP** = 1111b, then the CCR will skip 15 CCR cycles before generating an interrupt to the NVIC.

The diagram below shows an example of a CCR4 with interrupt skipping configured with a value of 4 (**TACCTL.CCINTSKIP** = 4).

Figure 18-8 CCR Interrupt Skipping Diagram



18.4.18 Timer ADC Triggers

Each timer CCR unit may be configured to be an ADC trigger, to allow the DTSE to automatically begin conversion sequences.

For more information on this feature, see the section on the ADC and DTSE in this user guide.

18.4.19 Dead-Time Generators (DTG)

Each timer has 4 DTG units. Each DTG unit is capable of generating a pair of complementary signals that can be configured with dead-time to drive an inverter for half-bridge topologies.

The input clock to the DTG can be configured to be the timer clock before or after the **TACTL.CLKDIV** input clock divider. If **TACTL.DTGCLK** is 0b, then the DTGCLK is the clock before the **TACTL.CLKDIV** clock divider. If **TACTL.DTGCLK** is 1b, then the DTGCLK is the clock after the **TACTL.CLKDIV** clock divider.

When **TADTGCTLn.DTEN** = 0b, the DTG is disabled (bypassed). In this mode, the timer signals have the following behavior (see the simplified block diagram above):

DTG0:

- CCR0 output is connected to TAPWM0
- CCR4 output is connected to TAPWM4

DTG1:

- CCR1 output is connected to TAPWM1
- CCR5 output is connected to TAPWM5

DTG2:

- CCR2 output is connected to TAPWM2
- CCR6 output is connected to TAPWM6

DTG3:

- CCR3 output is connected to TAPWM3
- CCR7 output is connected to TAPWM7

When **TADTGCTL0.DTEN** = 1b, the DTG is enabled. In this mode, only one CCR output is used to generate the two complementary outputs. In this mode, the timer signals have the following behavior:

DTG0:

- CCR0 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR4 output is used to generate the two complementary output signals: TAPWM0 and TAPWM4.
- TAPWM0 becomes the low-side complementary signal.
- TAPWM4 becomes the high-side complementary signal.

- The **TADTGCTL0.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, to when the low-side signal rises).
- The **TADTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, to when the high-side signal rises).

DTG1:

- CCR1 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR5 output is used to generate the two complementary output signals: TAPWM1 and TAPWM5.
- TAPWM1 becomes the low-side complementary signal.
- TAPWM5 becomes the high-side complementary signal.
- The **TADTGCTL1.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TADTGCTL1.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG2:

- CCR2 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR6 output is used to generate the two complementary output signals: TAPWM2 and TAPWM6.
- TAPWM2 becomes the low-side complementary signal.
- TAPWM6 becomes the high-side complementary signal.
- The **TADTGCTL2.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TADTGCTL2.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG3:

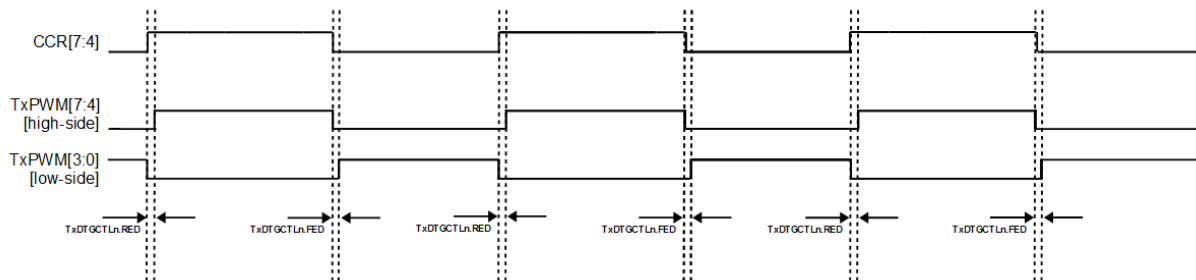
- CCR3 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR7 output is used to generate the two complementary output signals: TAPWM3 and TAPWM4.

- TAPWM3 becomes the low-side complementary signal.
- TAPWM7 becomes the high-side complementary signal.
- The **TADTGCTL3.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TADTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

The PWM waveform below shows an example of the dead-time applied to the signals through the DTG units.

18.4.20 Timer A DTG Output

Figure 18-9 Timer A DTG Output



18.5 QEP

The PAC55XX contains a QEP encoder peripheral that can be used to determine position and speed of a position-encoded motor. The QEP encoder can be enabled by setting the **TAQEPCTL.QEPEN** bit to a 1b.¹⁰

The inputs to the QEP peripheral are TBQEPPHA, TBQEPPHB and TBQEPIDX (phase A, phase B and index). These inputs are used to determine the position and direction of the motor, when the **TAQEPCTL.QEPEN** is set to a 1b. These three inputs have a glitch filter applied to them, to filter out noise and are used for calculation of the motor position and speed.

After filtering, the TBQEPPHA, TBQEPPHB and TBQEPIDX inputs are passed to the QEP engine to determine the position and speed. If the position is determined to be clockwise, then the **TAQEPCTL.DIR** is set to 0b. If the position is determined to be counter-clockwise, then the **TAQEPCTL.DIR** is set to 1b.

During operation, if the **TAQEPCTL.CNTAB** bit is 0b, only phase A edges are counted. If this bit is set to a 1b, then both phase A and phase B edges are counted. If the **TAQEPCTL.CNTEDGE** bit is set to 0b, then only the rising edges are counted. If this bit is set to a 1b, then both rising and falling edges are counted.

If the **TAQEPCTL.IDXRST** bit is set to a 1b, then an index event clears the **TAQEPCTL.TICKS** counter to 0. If this bit is set to a 0b, then an index event does not reset the **TAQEPCTL.TICKS** counter.

If a change in the direction is made (0 to 1, or 1 to 0), then the **TAQEPCTL.DIRIF** bit is set to a 1b. If the **TAQEPCTL.DIRIE** bit is set to 1b, then the QEPA_IRQ signal is asserted to the NVIC. The **TAQEPCTL.DIRIF** bit can be cleared by writing it to 1b.

When a rising edge on phase A is detected the **TAQEPCTL.PHAIF** bit is set to 1b. If the **TAQEPCTL.PHAIE** bit is set and the **TAQEPCTL.PHAIF** bit is set, then the QEPA_IRQ signal is asserted to the NVIC. The **TAQEPCTL.PHAIF** bit can be cleared by writing a 1b to it.

When a rising edge on phase B is detected the **TAQEPCTL.PHBIF** bit is set. If the **TAQEPCTL.PHBIE** bit is set and the **TAQEPCTL.PHBIF** bit is set, then the QEPA_IRQ signal is asserted to the NVIC. The **TAQEPCTL.PHBIF** bit can be cleared by writing a 1b to it.

If an overflow or underflow in the **TAQEPCTL.TICKS** is detected, the **TAQEPCTL.WRIF** bit is set (counter wrap interrupt flag). If the **TAQEPCTL.WRIE** bit is set and the **TAQEPCTL.WRIF** bits are set, then the QEPA_IRQ signal is asserted to the NVIC.

¹⁰ Before enabling the QEP peripheral, be sure to set up the Digital Peripheral MUX, so that no false edges are detected by the QEP state machine.

If an index event is detected, the **TAQEPCTL.IDXEVI** bit is set. If the **TAQEPCTL.IDXEVIF** bit is set and the **TAQEPCTL.IDXEVI** bit is set, then the QEPA_IRQ signal to the NVIC is asserted. The **TAQEPCTL.IDXEVI** bit can be cleared by writing a 1b to it.

As the QEP decoder counts edges on PH_A and is turning clockwise (CW), it increments the counter value in **TAQEPCTL.TICKS**. If the motor is turning counter-clockwise (CCW), the **TAQEPCTL.TICKS** register is decremented. The user may write the **TAQEPCTL.TICKS** register at any time.

18.6 Peripheral IO Mapping

The Timer A peripheral signal inputs and outputs (CCR, QEP) are connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 18-1 Timer A Peripheral IO Mapping

| TIMER SIGNAL | IO PIN |
|--------------|--------|
| TAPWM0 | PB0 |
| TAPWM1 | PB1 |
| TAPWM2 | PB2 |
| TAPWM3 | PB3 |
| TAPWM4 | PB4 |
| TAPWM5 | PB5 |
| TAPWM6 | PB6 |
| TAPWM7 | PB7 |
| TAQEPIDX | PE0 |
| TAQEPPHA | PE1 |
| TAQEPPHB | PE2 |

For more information on how to use the Digital Peripheral MUX to connect peripheral signals to IO, see Register Summary

18.7 Timer A Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---------------------------|--------|------------|
| TACTL | 4006 0000h | Timer A control | RW | 0000 0000h |
| TAINT | 4006 0004h | Timer A interrupt control | RW | 0000 0000h |
| TAPRD | 4006 0008h | Timer A period | RW | 0000 0000h |
| TACTR | 4006 000Ch | Timer A counter | RW | 0000 0000h |
| TAQEPCTL | 4006 0010h | Timer A QEP control | RW | 0000 0000h |
| TACCTL0 | 4006 0100h | Timer A CC control 0 | RW | 0000 0000h |
| TACCTR0 | 4006 0104h | Timer A CC counter 0 | RW | 0000 0000h |
| TACCTL1 | 4006 0108h | Timer A CC control 1 | RW | 0000 0000h |
| TACCTR1 | 4006 010Ch | Timer A CC counter 1 | RW | 0000 0000h |
| TACCTL2 | 4006 0110h | Timer A CC control 2 | RW | 0000 0000h |
| TACCTR2 | 4006 0114h | Timer A CC counter 2 | RW | 0000 0000h |
| TACCTL3 | 4006 0118h | Timer A CC control 3 | RW | 0000 0000h |
| TACCTR3 | 4006 011Ch | Timer A CC counter 3 | RW | 0000 0000h |
| TACCTL4 | 4006 0120h | Timer A CC control 4 | RW | 0000 0000h |
| TACCTR4 | 4006 0124h | Timer A CC counter 4 | RW | 0000 0000h |
| TACCTL5 | 4006 0128h | Timer A CC control 5 | RW | 0000 0000h |
| TACCTR5 | 4006 012Ch | Timer A CC counter 5 | RW | 0000 0000h |
| TACCTL6 | 4006 0130h | Timer A CC control 6 | RW | 0000 0000h |
| TACCTR6 | 4006 0134h | Timer A CC counter 6 | RW | 0000 0000h |
| TACCTL7 | 4006 0138h | Timer A CC control 7 | RW | 0000 0000h |
| TACCTR7 | 4006 013Ch | Timer A CC counter 7 | RW | 0000 0000h |
| TADTGCTL0 | 4006 0200h | Timer A DTG control 0 | RW | 0000 0000h |
| TADTGCTL1 | 4006 0204h | Timer A DTG control 1 | RW | 0000 0000h |
| TADTGCTL2 | 4006 0208h | Timer A DTG control 2 | RW | 0000 0000h |
| TADTGCTL3 | 4006 020Ch | Timer A DTG control 3 | RW | 0000 0000h |

18.8 Register Detail

18.8.1 TACTL

Register 18-1 TACTL (Timer A Control, 4006 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | BASEIE | RW | 0 | Base timer interrupt enable: 0b: not enabled 1b: enabled |
| 12 | CLR | RW | 0 | Timer Clear: 0b: do not clear 1b: clear timer counter |
| 11 | LATCH | RW | 0 | When written to 1b, this will latch the TAPRD and all TACCTRN registers on the same clock cycle. Writing this bit to 0b has no effect. This is a self-clearing bit. |
| 10 | DTGCLK | RW | 0 | DTG Clock Source: 0b: PCLK 1b: ACLK |
| 9 | CLKSRC | RW | 0 | Timer Clock Source: 0b: PCLK 1b: ACLK |
| 8:6 | CLKDIV | RW | 0 | Timer Input Clock Divider: 000b: /1 001b: /2 010b: /4 011b: /8 100b: /16 101b: /32 110b: /64 111b: /128 |
| 5 | SINGLE | RW | 0 | Single Shot Timer: 0b: disabled (auto-reload) 1b: enabled (single shot timer) |
| 4 | SSYNC | RW | 0 | Timer Slave Synchronization: 0b: disabled 1b: enabled |
| 3:2 | PRDLATCH | RW | 0 | Timer Period Latch Mode: 00b: Latch TAPRD when TACTR = 0 01b: Latch TAPRD when TACTR = TAPRD 10b: Latch TAPRD immediately upon register write 11b: reserved |
| 1:0 | MODE | RW | 0 | Timer Mode: |

| | | | | |
|--|--|--|--|--|
| | | | | 00b: disabled 01b: up mode 10b: up/down mode 11b: asymmetric mode |
|--|--|--|--|--|

18.8.2 TAINT

Register 18-2 TAINT (Timer A Interrupt Control, 4006 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:9 | Reserved | RO | 0 | Reserved |
| 8 | BASEIF | W1C | 0 | Base timer interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 7 | CCR7IF | W1C | 0 | CCR7 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 6 | CCR6IF | W1C | 0 | CCR6 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 5 | CCR5IF | W1C | 0 | CCR5 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 4 | CCR4IF | W1C | 0 | CCR4 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 3 | CCR3IF | W1C | 0 | CCR3 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 2 | CCR2IF | W1C | 0 | CCR2 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 1 | CCR1IF | W1C | 0 | CCR1 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 0 | CCR0IF | W1C | 0 | CCR0 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |

18.8.3 TAPRD

Register 18-3 TAPRD (Timer A Period, 4006 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--------------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | PERIOD | RW | 0 | Timer period value |

18.8.4 TACTR

Register 18-4 TACTR (Timer A Counter, 4006 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | COUNTER | RW | 0 | Timer counter |

18.8.5 TAQEPCTL

Register 18-5 TAQEPCTL (Timer A QEP Control, 4006 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | TICKS | RW | 0 | Number of counted phase edges according to counting mode defined by TAQEPCTL.CNTEEDGE and TAQEPCTL.CNTAB . |
| 15 | Reserved | RO | 0 | Reserved |
| 14 | CNTEEDGE | RW | 0 | Count on edge: 0b: Rising edge only 1b: Rising and falling edge |
| 13 | CNTAB | RW | 0 | Count on A/B: 0b: Count phase A only 1b: Count phase A and phase B |
| 12 | IDXEIVF | W1C | 0 | Index event interrupt flag: 0b: no index event detected 1b: index event detected |
| 11 | WRIF | W1C | 0 | Counter wrap interrupt flag: 0b: no counter wrap 1b: counter wrap |
| 10 | PHBIF | W1C | 0 | Phase B rising edge interrupt flag: 0b: no phase B rising edge detected 1b: phase B rising edge detected |
| 9 | PHAIF | W1C | 0 | Phase A rising edge interrupt flag: 0b: no phase A rising edge detected 1b: phase A rising edge detected |
| 8 | DIRIF | W1C | 0 | Direction change interrupt flag: 0b: no direction change detected 1b: direction change detected |
| 7 | IDXRST | RW | 0 | Reset counter on index: 0b: Do not reset TAQEPCTL.TICKS 1b: Reset TAQEPCTL.TICKS on all index events |
| 6 | IDXEIVIE | RW | 0 | Index event interrupt enable: 0b: disabled 1b: enabled |
| 5 | WRIE | RW | 0 | Counter wrap interrupt enable: 0b: disabled 1b: enabled |
| 4 | PHBIE | RW | 0 | Phase B rising edge interrupt enable: 0b: disabled 1b: enabled |
| 3 | PHAIE | RW | 0 | Phase A rising edge interrupt enable: |

| | | | | |
|---|--------------|----|---|---|
| | | | | 0b: disabled 1b: enabled |
| 2 | DIRIE | RW | 0 | Direction change interrupt enable: 0b: disabled 1b: enabled |
| 1 | DIR | RO | 0 | Motor direction: 0b: CW (clockwise) 1b: CCW (counter-clockwise) |
| 0 | QEPEN | RW | 0 | QEP peripheral enabled: 0b: disabled 1b: enabled |

18.8.6 TACCTL0

Register 18-6 TACCTL0 (Timer A CCR Control 0, 4006 0100h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL0.CCMODE = 0b (compare): 00b: Latch TACCTRL0 registers when TACTR = 0 01b: Latch TACCTRL0 registers when TACTR = TAPRD 10b: Latch TACCTRL0 registers immediately 11b: Reserved If TACCTL0.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR0.CTR on rising edge 01b: Latch TACTR into TACCTR0.CTR on falling edge 10b: Latch TACTR into TACCTR0.CTR on both edges 11b: Latch TACTR into TACCTR0.CAP1 and TACCTR0.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL0.CCMODE = 1b (capture), set TAINTCR0IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.7 TACCTR0

Register 18-7 TACCTR0 (Timer A CCR Counter 0, 4006 0104h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTL0.CCLATCH != 11b (period/duty mode) and TACCTL0.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTL0.CCLATCH = 11b (period/duty mode) and TACCTL0.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTL0.CCLATCH != 11b (period/duty mode) and TACCTR0.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTL0.CCLATCH != 11b (period/duty mode) and TACCTL0.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTL0.CCLATCH = 11b (period/duty mode) and TACCTL0.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.8 TACCTL1

Register 18-8 TACCTL1 (Timer A CCR Control 1, 4006 0108h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL1.CCMODE = 0b (compare): 00b: Latch TACCTL1 registers when TACTR = 0 01b: Latch TACCTL1 registers when TACTR = TAPRD 10b: Latch TACCTL1 registers immediately 11b: Reserved If TACCTL1.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR1.CTR on rising edge 01b: Latch TACTR into TACCTR1.CTR on falling edge 10b: Latch TACTR into TACCTR1.CTR on both edges 11b: Latch TACTR into TACCTR1.CAP1 and TACCTR1.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL1.CCMODE = 1b (capture), set TAINT.CCR1IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.9 TACCTR1

Register 18-9 TACCTR1 (Timer A CCR Counter 1, 4006 010Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTL1.CCLATCH != 11b (period/duty mode) and TACCTL1.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTL1.CCLATCH = 11b (period/duty mode) and TACCTL1.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTL1.CCLATCH != 11b (period/duty mode) and TACCTL1.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTL1.CCLATCH != 11b (period/duty mode) and TACCTL1.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTL1.CCLATCH = 11b (period/duty mode) and TACCTL1.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.10 TACCTL2

Register 18-10 TACCTL2 (Timer A CCR Control 2, 4006 0110h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL2.CCMODE = 0b (compare): 00b: Latch TACCTL2 registers when TACTR = 0 01b: Latch TACCTL2 registers when TACTR = TAPRD 10b: Latch TACCTL2 registers immediately 11b: Reserved If TACCTL2.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR2.CTR on rising edge 01b: Latch TACTR into TACCTR2.CTR on falling edge 10b: Latch TACTR into TACCTR2.CTR on both edges 11b: Latch TACTR into TACCTR2.CAP1 and TACCTR2.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL2.CCMODE = 1b (capture), set TAINT.CCR2IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.11 TACCTR2

Register 18-11 TACCTR2 (Timer A CCR Counter 2, 4006 0114h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL2.CCLATCH != 11b (period/duty mode) and TACCTRL2.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL2.CCLATCH = 11b (period/duty mode) and TACCTRL2.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL2.CCLATCH != 11b (period/duty mode) and TACCTRL2.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL2.CCLATCH != 11b (period/duty mode) and TACCTRL2.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL2.CCLATCH = 11b (period/duty mode) and TACCTRL2.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.12 TACCTL3

Register 18-12 TACCTL3 (Timer A CCR Control 3, 4006 0118h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL3.CCMODE = 0b (compare): 00b: Latch TACCTL3 registers when TACTR = 0 01b: Latch TACCTL3 registers when TACTR = TAPRD 10b: Latch TACCTL3 registers immediately 11b: Reserved If TACCTL3.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR3.CTR on rising edge 01b: Latch TACTR into TACCTR3.CTR on falling edge 10b: Latch TACTR into TACCTR3.CTR on both edges 11b: Latch TACTR into TACCTR3.CAP1 and TACCTR3.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL3.CCMODE = 1b (capture), set TAINT.CCR3IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.13 TACCTR3

Register 18-13 TACCTR3 (Timer A CCR Counter 3, 4006 011Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL3.CCLATCH != 11b (period/duty mode) and TACCTRL3.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL3.CCLATCH = 11b (period/duty mode) and TACCTRL3.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL3.CCLATCH != 11b (period/duty mode) and TACCTRL3.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL3.CCLATCH != 11b (period/duty mode) and TACCTRL3.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL3.CCLATCH = 11b (period/duty mode) and TACCTRL3.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.14 TACCTL4

Register 18-14 TACCTL4 (Timer A CCR Control 4, 4006 0120h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|------------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL4.CCMODE = 0b (compare): 00b: Latch TACCTL4 registers when TACTR = 0 01b: Latch TACCTL4 registers when TACTR = TAPRD 10b: Latch TACCTL4 registers immediately 11b: Reserved If TACCTL4.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR4.CTR on rising edge 01b: Latch TACTR into TACCTR4.CTR on falling edge 10b: Latch TACTR into TACCTR4.CTR on both edges 11b: Latch TACTR into TACCTR4.CAP1 and TACCTR4.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL4.CCMODE = 1b (capture), set TAINT.CCR4IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.15 TACCTR4

Register 18-15 TACCTR4 (Timer A CCR Counter 4, 4006 0124h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL4.CCLATCH != 11b (period/duty mode) and TACCTRL4.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL4.CCLATCH = 11b (period/duty mode) and TACCTRL4.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL4.CCLATCH != 11b (period/duty mode) and TACCTRL4.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL4.CCLATCH != 11b (period/duty mode) and TACCTRL4.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL4.CCLATCH = 11b (period/duty mode) and TACCTRL4.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.16 TACCTL5

Register 18-16 TACCTL5 (Timer A CCR Control 5, 4006 0128h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|------------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL5.CCMODE = 0b (compare): 00b: Latch TACCTL5 registers when TACTR = 0 01b: Latch TACCTL5 registers when TACTR = TAPRD 10b: Latch TACCTL5 registers immediately 11b: Reserved If TACCTL5.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR5.CTR on rising edge 01b: Latch TACTR into TACCTR5.CTR on falling edge 10b: Latch TACTR into TACCTR5.CTR on both edges 11b: Latch TACTR into TACCTR5.CAP1 and TACCTR5.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL5.CCMODE = 1b (capture), set TAINT.CCR5IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.17 TACCTR5

Register 18-17 TACCTR5 (Timer A CCR Counter 5, 4006 012Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL5.CCLATCH != 11b (period/duty mode) and TACCTRL5.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL5.CCLATCH = 11b (period/duty mode) and TACCTRL5.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL5.CCLATCH != 11b (period/duty mode) and TACCTRL5.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL5.CCLATCH != 11b (period/duty mode) and TACCTRL5.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL5.CCLATCH = 11b (period/duty mode) and TACCTRL5.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.18 TACCTL6

Register 18-18 TACCTL6 (Timer A CCR Control 6, 4006 0130h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL6.CCMODE = 0b (compare): 00b: Latch TACCTL6 registers when TACTR = 0 01b: Latch TACCTL6 registers when TACTR = TAPRD 10b: Latch TACCTL6 registers immediately 11b: Reserved If TACCTL6.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR6.CTR on rising edge 01b: Latch TACTR into TACCTR6.CTR on falling edge 10b: Latch TACTR into TACCTR6.CTR on both edges 11b: Latch TACTR into TACCTR6.CAP1 and TACCTR6.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL6.CCMODE = 1b (capture), set TAINT.CCR6IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.19 TACCTR6

Register 18-19 TACCTR6 (Timer A CCR Counter 6, 4006 0134h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL6.CCLATCH != 11b (period/duty mode) and TACCTRL6.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL6.CCLATCH = 11b (period/duty mode) and TACCTRL6.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL6.CCLATCH != 11b (period/duty mode) and TACCTRL6.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL6.CCLATCH != 11b (period/duty mode) and TACCTRL6.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL6.CCLATCH = 11b (period/duty mode) and TACCTRL6.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.20 TACCTL7

Register 18-20 TACCTL7 (Timer A CCR Control 7, 4006 0138h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TACCTL7.CCMODE = 0b (compare): 00b: Latch TACCTL7 registers when TACTR = 0 01b: Latch TACCTL7 registers when TACTR = TAPRD 10b: Latch TACCTL7 registers immediately 11b: Reserved If TACCTL7.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR7.CTR on rising edge 01b: Latch TACTR into TACCTR7.CTR on falling edge 10b: Latch TACTR into TACCTR7.CTR on both edges 11b: Latch TACTR into TACCTR7.CAP1 and TACCTR7.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TACCTL7.CCMODE = 1b (capture), set TAINT.CCR7IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

18.8.21 TACCTR7

Register 18-21 TACCTR7 (Timer A CCR Counter 7, 4006 013Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TACCTRL7.CCLATCH != 11b (period/duty mode) and TACCTRL7.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TACCTRL7.CCLATCH = 11b (period/duty mode) and TACCTRL7.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TACCTRL7.CCLATCH != 11b (period/duty mode) and TACCTRL7.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TACCTRL7.CCLATCH != 11b (period/duty mode) and TACCTRL7.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TACCTRL7.CCLATCH = 11b (period/duty mode) and TACCTRL7.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

18.8.22 TADTGCTL0

Register 18-22 TADTGCTL0 (Timer A Dead-Time Generator Control 0, 4006 0200h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG0. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG0. |

18.8.23 TADTGCTL1

Register 18-23 TADTGCTL1 (Timer A Dead-Time Generator Control 1, 4006 0204h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG1. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG1. |

18.8.24 TADTGCTL2

Register 18-24 TADTGCTL2 (Timer A Dead-Time Generator Control 2, 4006 0208h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG2. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG2. |

18.8.25 TADTGCTL3

Register 18-25 TADTGCTL3 (Timer A Dead-Time Generator Control 3, 4006 020Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG3. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG3. |

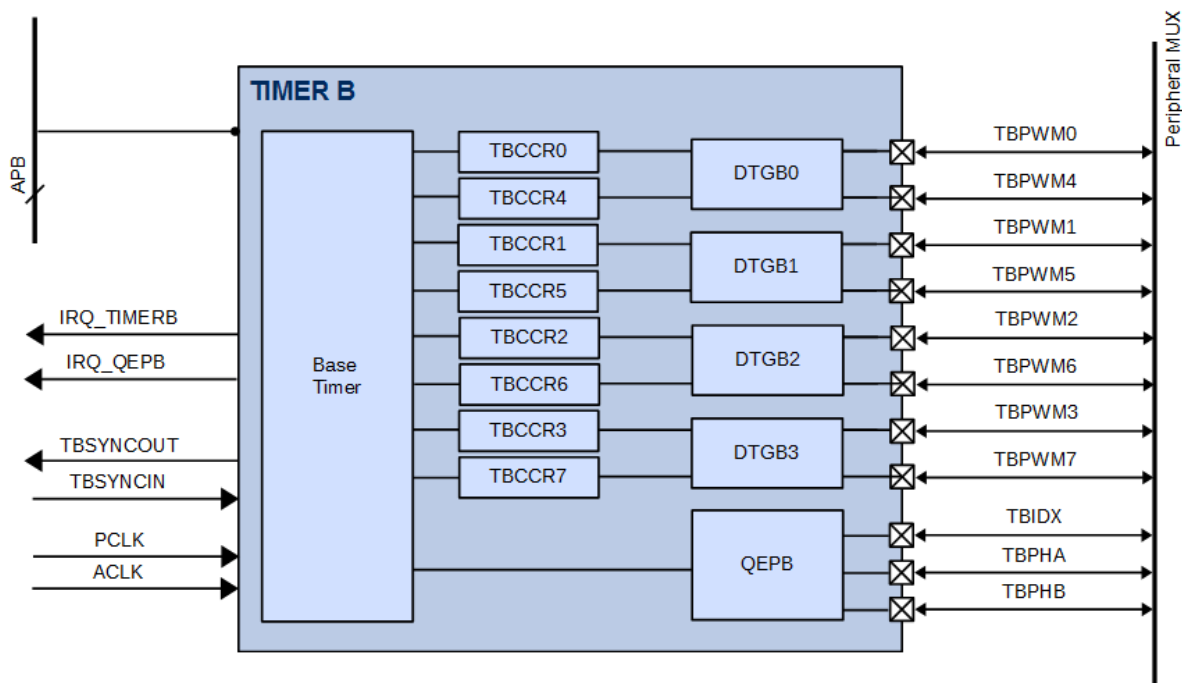
19 PWM TIMER B

19.1 Overview

All devices in the PAC55XX family of controllers have a Timer B peripheral. This peripheral is a 16-bit timer that allows support for 8 Capture and Compare Units (CCR) capable of PWM generation; capture input processing and a QEP encoder for various control applications.

Below is a simplified block diagram of the Timer peripheral.

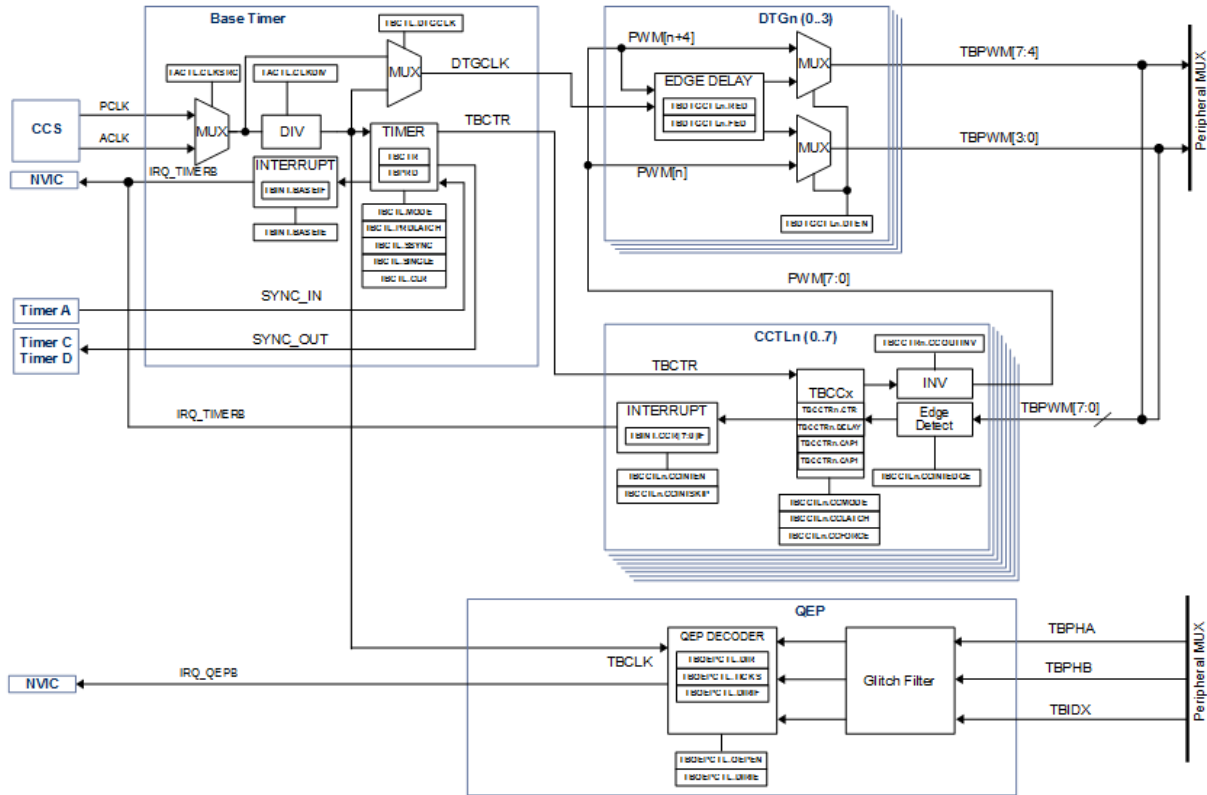
Figure 19-1 Timer B Simplified Block Diagram



The timer module is an APB bus client. There are 8 timer channel input/outputs to the digital peripheral MUX, in addition to three IOs for QEP. Each timer block has two interrupt signal outputs to the NVIC; one for timer functions and one for QEP functions. The user may select either PCLK or ACLK as the clock input for each timer module.

19.2 Timer B Block Diagram

Figure 19-2 Timer B Block Diagram



19.3 Features

The PWM Timer peripheral has the following features in the PAC55XX family of controllers.

General Features:

- Configurable input clock: PCLK or ACLK
- Up to 300MHz clock input for 3.33ns PWM edge resolution
- 3-bit input clock divider
- Latch timer period and all CCR values on command

Base Timer Features:

- Single-shot or auto-reload
- Base timer interrupts
- Timer synchronization
- Timer Modes:
 - Disabled
 - Up mode
 - Up/Down mode
 - Asymmetric Up mode
- Timer Register Latching Options
 - Latch TAPRD when counter = 0
 - Latch TAPRD when counter = period
 - Latch TAPRD immediately
- Dead-time Generator Input Clock
 - DTG clock = PCLK
 - DTG clock = ACLK

CCR/PWM Features:

- PWM output or input capture
- 8 CCR units per timer
- CCR interrupts
- CCR interrupt skips
- SW force CCR interrupt
- CCR interrupt type:
 - Rising, falling, both
- CCR latch modes:
 - Compare/PWM mode: counter = 0, counter = period, immediate
 - Capture input: rising edge, falling edge, both
- Period/duty calculation mode
- Force compare event

- Invert CCR output
- CCR phase delay for phase shifted drive topologies
- ADC trigger outputs:
 - PWM rising or falling edge

Dead-Time Generator (DTG) Features:

- DTG enable or bypass
- 12-bit rising edge delay
- 12-bit falling edge delay

QEP Decoder Features:

- QEP encoder enabled
- Direction status
- Configurable interrupts:
 - Phase A rising edge
 - Phase B rising edge
 - Index event
 - Counter wrap
- 4 Different counting modes for best resolution, range and speed performance

19.4 Functional Description

19.4.1 Timer Clock Structure

The timer peripheral input clock can be selected as either PCLK or ACLK. The input clock may be selected by the **TBCTL.CLKSRC** register.

Each timer peripheral has a 4-bit divider that can be used to divide the selected input clock. The user may set this divider by the **TBCTL.CLKDIV** register. There are 16 settings between /1 and /128 for the input clock divider.

The base timer also supplies the clock for the Dead-Time Generators (DTGs) to allow for different range and resolution for dead-time. The timer may select the DTGCLK to be the timer clock before or after the clock divider. The DTG input clock may be selected by the **TBCTL.DTGCLK** register.

To use the timer clock before the input clock divider, set **TBCTL.DTGCLK** to 0b. To use the timer clock after the input clock divider, set **TBCTL.DTGCLK** to 1b.

19.4.2 Timer Counter

The base timer is a 16-bit timer that can count either up, or up then down to support both edge and center aligned and asymmetric PWM output types.

The timer period is stored in the **TBPRD** register. The current value of the timer counter is updated at every timer tick and is stored in the **TBCTR** register. The **TBCTR** register is a RW register, so it may be updated at any time and is changed immediately.

The **TBPRD** register and all of the **TBCCTRn** registers have shadow copies that are updated at a user-specified time. See the section below on Base Timer **TAPRD** Latching and CCR Timer Latching for more information.

When **TBCTL.MODE** = 00b, the timer is disabled. Even when disabled, all timer registers are accessible via the APB bus.

To enable the timer, set the **TBCTL.MODE** to 01b (up mode), 10b (up/down mode) or 11b (up/down asymmetric mode).

When **TBCTL.MODE** = 01b, the timer is configured in up mode. The timer will count from 0 to **TBPRD**. The counter is updated in **TBCTR** at every timer tick. If auto-reload is active (**TBCTL.SINGLE** = 0b), then the timer count will automatically count from **TBPRD** to 0 and continue counting up. If auto-reload is not active (**TBCTL.SINGLE** = 1b), then the timer will count from 0 to **TBPRD** and stop. When this happens, the timer will set the **TBCTL.MODE** to 00b (disabled).

When the timer is configured for up/down mode (**TBCTL.MODE** = 10b) or configured for up/down asymmetric mode (**TBCTL.MODE** = 11b), the timer will count from 0 to **TBPRD** and then back down to 0. If auto-reload is not active (**TBCTL.SINGLE** = 1b), the time timer will stop

and set **TBCTL.MODE** to 00b (disabled). If auto-reload is active (**TBCTL.SINGLE** = 0b), then the timer count will then continue counting back up after it reaches 0.

19.4.3 Up Mode

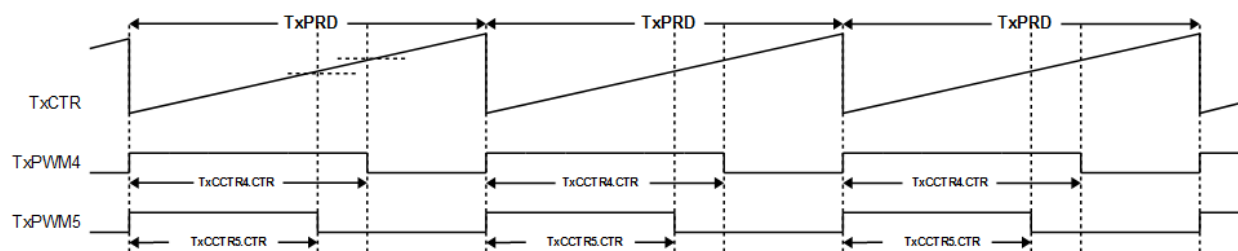
When the **TBCTL.MODE** is set to 01b (up mode) the timer will begin counting up from 0. The **TBCTR** will count to the value of **TBPRD** and then will reset to 0.

If the timer is configured for auto-reload mode (**TBCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it reaches 0.

The **TBCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate an edge aligned PWM as shown below.

Figure 19-3 Up Mode PWM Waveform



In this mode, when **TBCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TBCTR** register reaches the **TBCCTRn.CTR** threshold. When the **TBCTR** counts from **TBPRD** to 0, the PWM output is transitioned to high.

19.4.4 Up/Down Mode

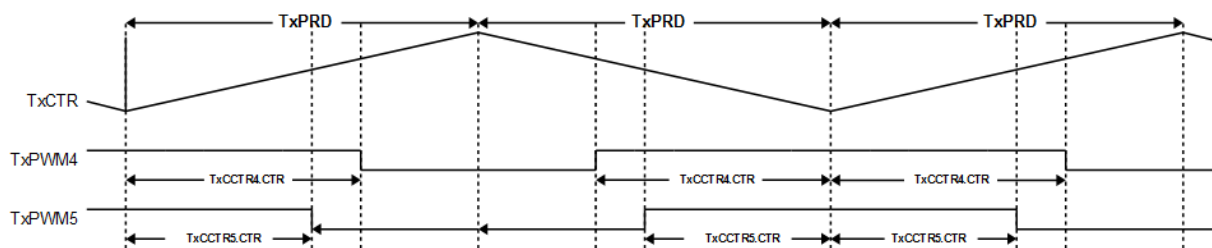
When the **TBCTL.MODE** is set to 10b (up/down mode) the timer will begin counting up from 0. The **TBCTR** will count to the value of **TBPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TBCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TBCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned PWM as shown below.

Figure 19-4 Up/Down Mode PWM Waveform



In this mode, when **TBCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TBCTR** register reaches the **TBCCTRn.CTR** threshold. When the **TBCTR** counts from **TBPRD - 1** to **TBPRD**, it will begin counting down. When the **TBCTR** counts down to **TBCCTRn.CTR** the PWM output is transitioned to high.

19.4.5 Up/Down Asymmetric mode

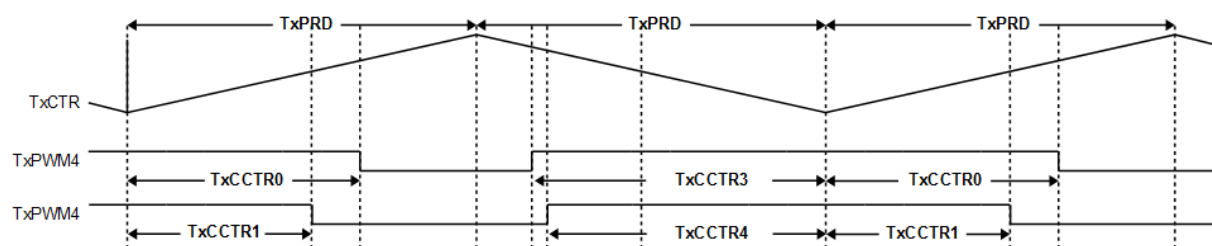
When the **TBCTL.MODE** is set to 11b (up/down asymmetric mode) the timer will begin counting up from 0. The **TBCTR** will count to the value of **TBPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TBCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TBCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned asymmetric PWM as shown below.

Figure 19-5 Up/Down Asymmetric Mode PWM Waveform



In this mode, when **TBCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TBCTR** register reaches the **TBCCTRn.CTR** threshold. When the **TBCTR** counts from **TBPRD - 1** to **TBPRD**, it will begin counting down. When the **TBCTR** counts down to **TBCCTRn.CTR** the PWM output is transitioned to high.

In this mode, the CCR outputs are generated such that 2 **TBCCTRn.CTR** registers are used to generate a single PWM output, so that the center-aligned PWM can be generated with asymmetric on-time.

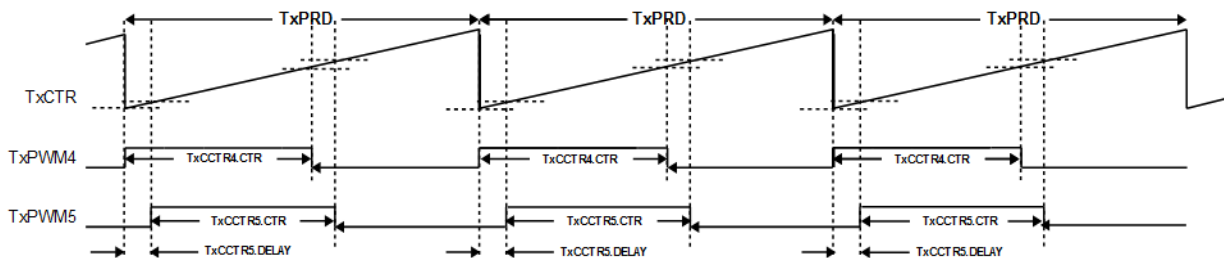
In this mode, the PWM outputs of the timer are generated as follows:

- TBPWM4 uses **TBCCTR0.CTR** for the first portion of the on time and **TBCCTR4.CTR** for the second portion of the on time
- TBPWM5 uses **TBCCTR1.CTR** for the first portion of the on time and **TBCCTR5.CTR** for the second portion of the on time
- TBPWM6 uses **TBCCTR2.CTR** for the first portion of the on time and **TBCCTR6.CTR** for the second portion of the on time
- TBPWM7 uses **TBCCTR3.CTR** for the first portion of the on time and **TBCCTR7.CTR** for the second portion of the on time

19.4.6 Up Mode with Phase Delay

When the timer is configured for up mode (**TBMODE** = 01b) the CCR allows a delay to be applied to the output PWM period, to support phase delay control topologies. If the value of **TBCCTLn.DELAY** > 0, then the CCR will delay this number of ticks before starting the PWM period (transitioning the output signal from high to low).

Figure 19-6 Up Mode with Phase Delay



19.4.7 Timer Synchronization

The timer peripheral allows synchronization between the timers in the PAC55XX. The timers may be synchronized as follows:

- Timer A (master), Timer B (slave)
- Timer A (master), Timer B (slave), Timer C (slave)
- Timer A (master), Timer B (slave), Timer C (slave), Timer D (slave)

The timer master's SYNC_OUT signals are connected to the downstream timer slave's SYNC_IN signal.

When configured, the timers can synchronize their clocks so they are sharing the same time base. In order to do this, the timers must be configured with the same clock source (**TBCTL.CLKSRC**) and clock divider (**TBCTL.CLKDIV**).

To configure the timers for synchronization, follow these steps:

- While the timer master is disabled (**TBCTL.MODE** = 00b), set the timer master **TxCTL.SSYNC** = 0b
- For each of the downstream timer slaves, while the timer is disabled **TBCTL.MODE** = 00b, set the **TBCTL.SSYNC** to 1b. Make sure that **TBCTL.CLKDIV** and **TBCTL.CLKSRC** for each of the timer slaves is the same as the timer master.
- Enable each of the slave timers by setting **TBCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric). The slave timers will not start counting since the **TBCTL.SSYNC** is set to a 1b.
- Enable the timer master by setting **TxCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric).

At this point, the master and all configured timer slaves will begin counting.

19.4.8 Base Timer TAPRD Latching

The timer peripherals have a shadow copy of the **TBPRD** register that is used for the counting operations. The latching of the data from the **TBPRD** register to the shadow register is controlled by the timer configuration.

There are several options for latching of this register into the shadow copy. This behavior is controlled by the **TBCTL.PRDLATCH** setting and depends on the setting of the **TBCTL.MODE** register as well.

19.4.8.1 TBPRD Latch When TACTR = 0

- If **TBCTL.MODE** = 01b (up mode) and **TBCTL.PRDLATCH** = 00b (**TBCTR** = 0), then the **TBPRD** register is copied into the shadow register when **TBCTR** counts from **TBPRD** to 0.
- If **TBCTL.MODE** = 10b (up/down mode) or if **TBCTL.MODE** = 11b (up/down asymmetric mode) and **TBCTL.PRDLATCH** = 00b (**TBCTR** = 0), then the **TBPRD** register is copied into the shadow register when the **TBCTR** counts from 1 to 0.

19.4.8.2 TBPRD Latch When TBCTR = TBPRD

- If **TBCTL.MODE** = 01b (up mode) or if **TBCTL.MODE** = 10b (up/down mode) or if **TBCTL.MODE** = 11b (up/down asymmetric mode) and **TBCTL.PRDLATCH** = 01b (**TBCTR** = **TBPRD**), then the **TBPRD** register is copied into the shadow register when **TBCTR** counts from **TBPRD** – 1 to **TBPRD**.

19.4.8.3 TBPRD Latch Immediate

- If **TBCTL.PRDLATCH** = 11b (immediate), then the **TBPRD** register is copied into the shadow register as soon as the **TBPRD** register is written.

19.4.9 CCR Timer Latching

When in compare mode, the CCR units support configurable latching of the **TBCCRn.CTR** values into a shadow register that is used for timer operation. The shadow register is used for generating the PWM output when the CCR is in compare mode.

This feature can be used to control when the new duty cycle is applied, during the PWM period.

When the CCR is in compare mode (**TBCCTLn.CCMODE** = 0b), the shadow register is latched as follows:

- If **TBCCTLn.CCLATCH** = 00b (**TBCTR** = 0)
 - If **TBCTL.MODE** = 01b (up mode), **TBCCRn.CTR** is copied into the shadow register when **TBCTR** counts from **TBPRD** to 0. If **TBCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode), **TBCCRn.CTR** is copied into the shadow register when **TBCTR** counts from 1 to 0.
- If **TBCCTLn.CCLATCH** = 01b (**TBCTR** = **TBPRD**), then the **TBCCRn.CTR** is copied into the shadow register when **TBCTR** counts from **TBPRD** – 1 to **TBPRD**.
- If **TBCCTLn.CCLATCH** = 10b (latch immediate), then the **TBCCRn.CTR** is copied into the shadow register as soon as it is written.

When the CCR is in capture mode (**TBCCTLn.CCMODE** = 1b), the configuration of **TBCCTLn.CCLATCH** controls when the **TBCTR** is copied into the **TBCCRn.CTR** register.

- If **TBCCTLn.CCLATCH** = 00b (rising edge), then the value of **TBCTR** is copied into the **TBCCTL.CTR** register upon a rising edge in the input signal.
- If **TBCCTLn.CCLATCH** = 01b (falling edge), then the value of the **TBCTR** is copied into the **TBCCTLn.CTR** register upon a falling edge in the input signal.
- If **TBCCTLn.CCLATCH** = 10b (both), then the value of the **TBCTR** is copied into the **TBCCTLn.CTR** register on both a rising and falling edge of the input signal.
- If **TBCCTLn.CCLATCH** = 11b (period/duty mode), then see the section below on how the values are latched into the **TBCCRn** register.

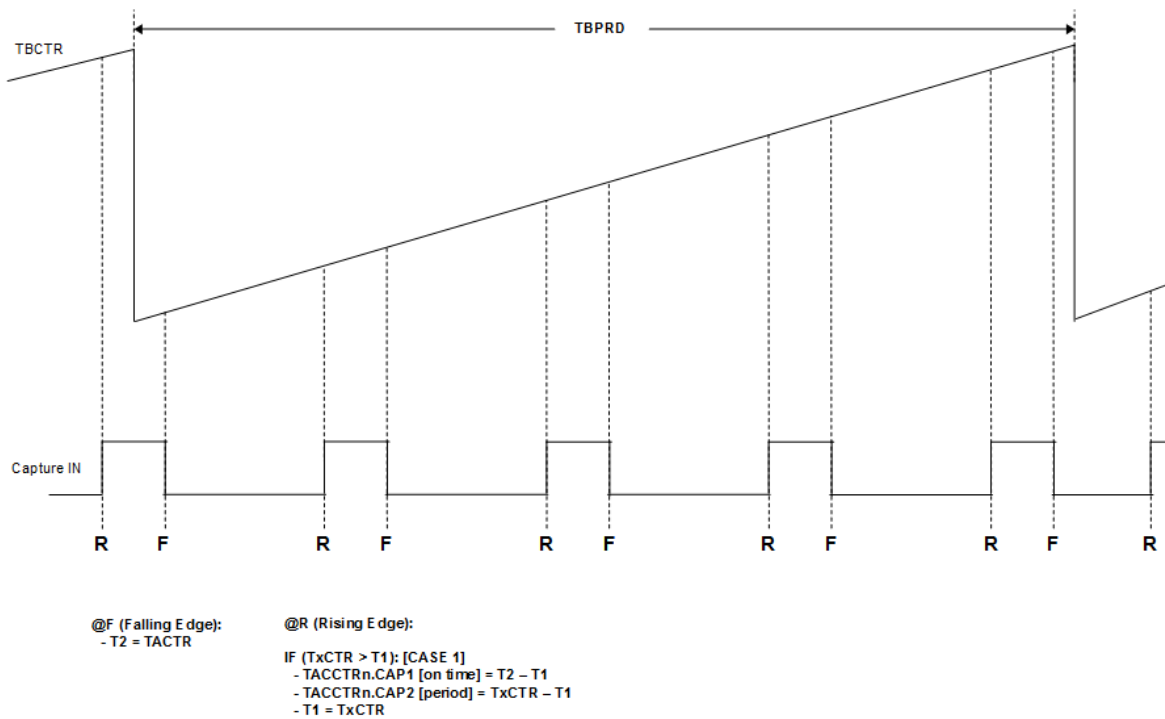
19.4.10 CCR Capture Period/Duty Mode

If the **TBCCTLn.CCLATCH = 11b** (period/duty mode), then the CCR can capture the number of timer ticks for the input PWM duty cycle and period. This mode is useful for measuring input pulse widths for PPM and other type of control applications.

At each rising edge of the input signal, the following happens:

- The number of timer ticks for the last timer period's duty cycle is copied into **TBCCTRn.CAP1**
- The number of timer ticks for the last timer period is copied into **TBCCTRn.CAP2**

Figure 19-7 Period/Duty Capture Mode PWM Waveform



19.4.11 Timer Whole Latching

It is sometimes convenient to latch both the **TBPRD** (period) and all **TBCCTRn.CTR** (duty cycle) registers into the shadow registers at one time.

If the **TBCTL.LATCH** bit is written to a 1b, then all the **TBPRD** and all **TBCCTRn.CTR** registers will be copied into the shadow registers on the same clock cycle.

The **TBCTL.LATCH** bit is self-clearing, and will always be read as a 0b.

19.4.12 Inverting CCR PWM Output

When the CCR is in compare mode (**TBCCTRn.CCMODE** = 0b), the output of the CCR (the input to the DTG) may be inverted. To invert this output, set the **TBCCTRn.CCOUTINV** = 1b.

This mode is useful for some control topologies, especially full-bridge.

19.4.13 Base Timer Interrupts

The base timer may be configured to generate an interrupt to the NVIC.

If **TBCTL.MODE** = 01b (up mode) and **TBCTR** counts from **TBPRD** to 0 then the **TBINT.BASEIF** bit is set to 1b. If the **TBCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMER_IRQ** signal to the NVIC is asserted.

If **TBCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode) and **TBCTR** counts from 1 to 0, then the **TBINT.BASEIF** bit is set to 1b. If the **TBCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERB_IRQ** signal to the NVIC is asserted.

The **TBINT.BASEIF** may be cleared by writing a 1b to it.

19.4.14 CCR Compare Interrupts

When **TBCCTLn.CCMODE** = 0b (compare mode), the CCR unit may be configured to generate an interrupt to the NVIC.

When **TBCTR** counts to **TBCCTRn.CTR**, then the CCR interrupt flag in the **TBINT** register is set as follows:

- For CCR0, when **TBCTR** counts to **TBCCTR0.CTR**, then **TBINT.CCR0IF** is set to a 1b. If the **TBCCTR0.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR0IF** may be cleared by writing it to a 1b.
- For CCR1, when **TBCTR** counts to **TBCCTR1.CTR**, then **TBINT.CCR1IF** is set to a 1b. If the **TBCCTR1.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR1IF** may be cleared by writing it to a 1b.
- For CCR2, when **TBCTR** counts to **TBCCTR2.CTR**, then **TBINT.CCR2IF** is set to a 1b. If the **TBCCTR2.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR2IF** may be cleared by writing it to a 1b.

- For CCR3, when **TBCTR** counts to **TBCCTR3.CTR**, then **TBINT.CCR3IF** is set to a 1b. If the **TBCCTR3.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR3IF** may be cleared by writing it to a 1b.
- For CCR4, when **TBCTR** counts to **TBCCTR4.CTR**, then **TBINT.CCR4IF** is set to a 1b. If the **TBCCTR4.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR4IF** may be cleared by writing it to a 1b.
- For CCR5, when **TBCTR** counts to **TBCCTR5.CTR**, then **TBINT.CCR5IF** is set to a 1b. If the **TBCCTR5.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR5IF** may be cleared by writing it to a 1b.
- For CCR6, when **TBCTR** counts to **TBCCTR6.CTR**, then **TBINT.CCR6IF** is set to a 1b. If the **TBCCTR6.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR6IF** may be cleared by writing it to a 1b.
- For CCR7, when **TBCTR** counts to **TBCCTR7.CTR**, then **TBINT.CCR7IF** is set to a 1b. If the **TBCCTR7.CCINTEN** is set to 1b, then **TIMERB_IRQ** to the NVIC is asserted. **TBINT.CCR7IF** may be cleared by writing it to a 1b.

19.4.15 CCR Capture Interrupts

When **TBCCTLn.CCMODE** = 1b (capture mode), the CCR unit may be configured to generate an interrupt to the NVIC based on the input signal.

- If **TBCCTLn.CCEGE** = 00b (rising edge) and a rising edge is detected in the input signal, then the **TBINT.CCRnIF** interrupt flag is set.
- If **TBCCTLn.CCEDGE** = 01b (falling edge) and a falling edge is detected in the input signal, then the **TBINT.CCRnIF** interrupt flag is set.
- If the **TBCCTLn.CCEDGE** = 11b (rising or falling edge) and a rising or a falling edge is detected in the input signal, then the **TBINT.CCRnIF** interrupt flag is set.

If the **TBINT.CCRnIF** interrupt flag is set and the corresponding **TBCCTLn.CCINTEN** bit is set, then the **TIMERB_IRQ** signal to the NVIC will be asserted.

The **TBINT.CCRnIF** interrupt flag can be cleared by writing it to a 1b.

19.4.16 Timer IRQ signal

The timer unit has one IRQ output signal to the NVIC (**TIMERB_IRQ**).

This signal is asserted when any of the following conditions are true:

- **TBINT.BASEIF** = 1b and **TBCTL.BASEIE** = 1b
- **TBINT.CCR0IF** = 1b and **TBCCTL0.CCINTEN** = 1b
- **TBINT.CCR1IF** = 1b and **TBCCTL1.CCINTEN** = 1b
- **TBINT.CCR2IF** = 1b and **TBCCTL2.CCINTEN** = 1b
- **TBINT.CCR3IF** = 1b and **TBCCTL3.CCINTEN** = 1b
- **TBINT.CCR4IF** = 1b and **TBCCTL4.CCINTEN** = 1b

- **TBINT.CCR5IF** = 1b and **TBCCTL5.CCINTEN** = 1b
- **TBINT.CCR6IF** = 1b and **TBCCTL6.CCINTEN** = 1b
- **TBINT.CCR7IF** = 1b and **TBCCTL7.CCINTEN** = 1b

The **TIMERB_IRQ** signal is de-asserted when all of the above conditions are false.

19.4.17 Skipping CCR Interrupts

Sometimes it is useful to not generate a CCR interrupt to the NVIC every time the **TBCTR** counts to the **TBCCTRn.CTR** value. For example, in a control application when the PWM frequency is fast, but you only need a MCU interrupt every 5 PWM periods.

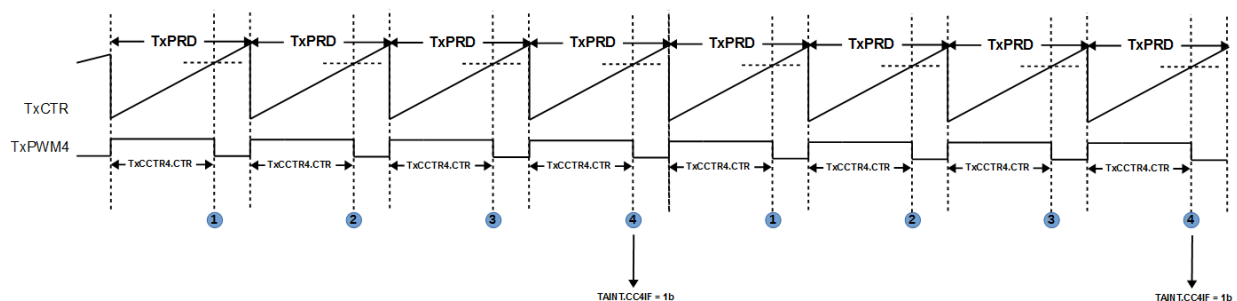
Each CCR in the timer has the ability to skip interrupts to the NVIC to allow this capability.

The CCR will generate interrupts to the NVIC according to the **TBCCTL.CCINTSKIP** field as follows:

- If **TBCCTLn.CCINTSKIP** = 0000b, then the CCR will generate an interrupt each time
- If **TBCCTLn.CCINTSKIP** = 0001b, then the CCR will skip 1 CCR cycle before generating an interrupt to the NVIC.
- If **TBCCTLn.CCINTSKIP** = 0010b, then the CCR will skip 2 CCR cycles before generating an interrupt to the NVIC.
- If **TBCCTLn.CCINTSKIP** = 0011b, then the CCR will skip 3 CCR cycles before generating an interrupt to the NVIC.
- ...
- If **TBCCTLn.CCINTSKIP** = 1110b, then the CCR will skip 14 CCR cycles before generating an interrupt to the NVIC.
- If **TBCCTLn.CCINTSKIP** = 1111b, then the CCR will skip 15 CCR cycles before generating an interrupt to the NVIC.

The diagram below shows an example of a CCR4 with interrupt skipping configured with a value of 4 (**TBCCTL.CCINTSKIP** = 4).

Figure 19-8 CCR Interrupt Skipping Diagram



19.4.18 Timer ADC Triggers

Each timer CCR unit may be configured to be an ADC trigger, to allow the DTSE to automatically begin conversion sequences.

For more information on this feature, see the section on the ADC and DTSE in this user guide.

19.4.19 Dead-Time Generators (DTG)

Each timer has 4 DTG units. Each DTG unit is capable of generating a pair of complementary signals that can be configured with dead-time to drive an inverter for half-bridge topologies.

The input clock to the DTG can be configured to be the timer clock before or after the **TBCTL.CLKDIV** input clock divider. If **TBCTL.DTGCLK** is 0b, then the DTGCLK is the clock before the **TBCTL.CLKDIV** clock divider. If **TBCTL.DTGCLK** is 1b, then the DTGCLK is the clock after the **TBCTL.CLKDIV** clock divider.

When **TBDTGCTLn.DTEN** = 0b, the DTG is disabled (bypassed). In this mode, the timer signals have the following behavior (see the simplified block diagram above):

DTG0:

- CCR0 output is connected to TBPWM0
- CCR4 output is connected to TBPWM4

DTG1:

- CCR1 output is connected to TBPWM1
- CCR5 output is connected to TBPWM5

DTG2:

- CCR2 output is connected to TBPWM2
- CCR6 output is connected to TBPWM6

DTG3:

- CCR3 output is connected to TBPWM3
- CCR7 output is connected to TBPWM7

When **TBDTGCTL0.DTEN** = 1b, the DTG is enabled. In this mode, only one CCR output is used to generate the two complementary outputs. In this mode, the timer signals have the following behavior:

DTG0:

- CCR0 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR4 output is used to generate the two complementary output signals: TBPWM0 and TBPWM4.
- TBPWM0 becomes the low-side complementary signal.
- TBPWM4 becomes the high-side complementary signal.

- The **TBDTGCTL0.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, to when the low-side signal rises).
- The **TBDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, to when the high-side signal rises).

DTG1:

- CCR1 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR5 output is used to generate the two complementary output signals: TBPWM1 and TBPWM5.
- TBPWM1 becomes the low-side complementary signal.
- TBPWM5 becomes the high-side complementary signal.
- The **TBDTGCTL1.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TBDTGCTL1.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG2:

- CCR2 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR6 output is used to generate the two complementary output signals: TBPWM2 and TAPWM6.
- TAPWM2 becomes the low-side complementary signal.
- TAPWM6 becomes the high-side complementary signal.
- The **TADTGCTL2.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TADTGCTL2.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG3:

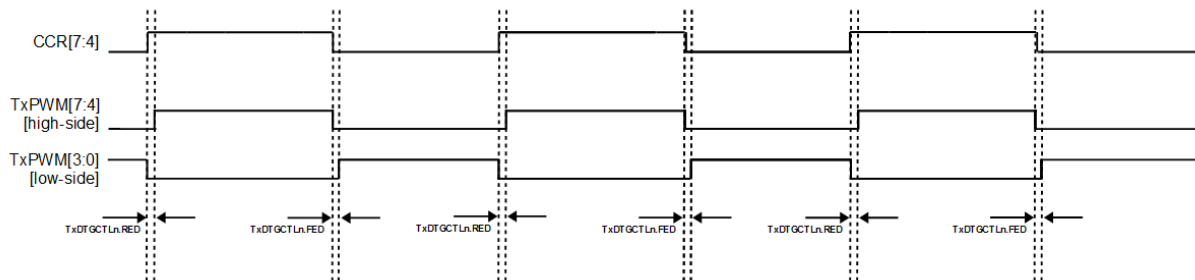
- CCR3 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR7 output is used to generate the two complementary output signals: TAPWM3 and TBPWM4.

- TBPWM3 becomes the low-side complementary signal.
- TBPWM7 becomes the high-side complementary signal.
- The **TBDTGCTL3.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TBDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

The PWM waveform below shows an example of the dead-time applied to the signals through the DTG units.

19.4.20 Timer B DTG Output

Figure 19-9 Timer B DTG Output



19.5 QEP

The PAC55XX contains a QEP encoder peripheral that can be used to determine position and speed of a position-encoded motor. The QEP encoder can be enabled by setting the **TBQEPCTL.QEPEN** bit to a 1b.¹¹

The inputs to the QEP peripheral are TBQEPPHA, TBQEPPHB and TBQEPIDX (phase A, phase B and index). These inputs are used to determine the position and direction of the motor, when the **TBQEPCTL.QEPEN** is set to a 1b. These three inputs have a glitch filter applied to them, to filter out noise and are used for calculation of the motor position and speed.

After filtering, the TBQEPPHA, TBQEPPHB and TBQEPIDX inputs are passed to the QEP engine to determine the position and speed. If the position is determined to be clockwise, then the **TBQEPCTL.DIR** is set to 0b. If the position is determined to be counter-clockwise, then the **TBQEPCTL.DIR** is set to 1b.

During operation, if the **TBQEPCTL.CNTAB** bit is 0b, only phase A edges are counted. If this bit is set to a 1b, then both phase A and phase B edges are counted. If the **TBQEPCTL.CNTEDGE** bit is set to 0b, then only the rising edges are counted. If this bit is set to a 1b, then both rising and falling edges are counted.

If the **TBQEPCTL.IDXRST** bit is set to a 1b, then an index event clears the **TBQEPCTL.TICKS** counter to 0. If this bit is set to a 0b, then an index event does not reset the **TBQEPCTL.TICKS** counter.

If a change in the direction is made (0 to 1, or 1 to 0), then the **TBQEPCTL.DIRIF** bit is set to a 1b. If the **TBQEPCTL.DIRIE** bit is set to 1b, then the QEPB_IRQ signal is asserted to the NVIC. The **TBQEPCTL.DIRIF** bit can be cleared by writing it to 1b.

When a rising edge on phase A is detected the **TBQEPCTL.PHAIF** bit is set to 1b. If the **TBQEPCTL.PHAIE** bit is set and the **TBQEPCTL.PHAIF** bit is set, then the QEPB_IRQ signal is asserted to the NVIC. The **TBQEPCTL.PHAIF** bit can be cleared by writing a 1b to it.

When a rising edge on phase B is detected the **TBQEPCTL.PHBIF** bit is set. If the **TBQEPCTL.PHBIE** bit is set and the **TBQEPCTL.PHBIF** bit is set, then the QEPB_IRQ signal is asserted to the NVIC. The **TBQEPCTL.PHBIF** bit can be cleared by writing a 1b to it.

If an overflow or underflow in the **TBQEPCTL.TICKS** is detected, the **TBQEPCTL.WRIF** bit is set (counter wrap interrupt flag). If the **TBQEPCTL.WRIE** bit is set and the **TBQEPCTL.WRIF** bits are set, then the QEPB_IRQ signal is asserted to the NVIC.

If an index event is detected, the **TBQEPCTL.IDXEVI** bit is set. If the **TBQEPCTL.IDXEVI** bit is set and the **TBQEPCTL.IDXEVI** bit is set, then the QEPB_IRQ signal to the NVIC is asserted. The **TBQEPCTL.IDXEVI** bit can be cleared by writing a 1b to it.

¹¹ Before enabling the QEP peripheral, be sure to set up the Digital Peripheral MUX, so that no false edges are detected by the QEP state machine.

As the QEP decoder counts edges on PH_A and is turning clockwise (CW), it increments the counter value in **TBQEPCTL.TICKS**. If the motor is turning counter-clockwise (CCW), the **TBQEPCTL.TICKS** register is decremented. The user may write the **TBQEPCTL.TICKS** register at any time.

19.6 Peripheral IO Mapping

The Timer B peripheral signal inputs and outputs (CCR, QEP) are connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 19-1 Timer B Peripheral IO Mapping

| TIMER SIGNAL | IO PIN |
|--------------|--------------------|
| TBPWM0 | PB3, PB7, PC0, PD0 |
| TBPWM1 | PB2, PB6, PC1, PD1 |
| TBPWM2 | PB1, PB5, PC2, PD2 |
| TBPWM3 | PB0, PB4, PC3, PD3 |
| TBPWM4 | PC4, PD4 |
| TBPWM5 | PC5, PD5 |
| TBPWM6 | PC6, PD6 |
| TBPWM7 | PC7, PD7 |
| TBQEPHA | PC1, PD5, PE1, PF1 |
| TBQEPIDX | PC0, PD4, PE0, PF0 |
| TBQEPHB | PC2, PD6, PE2, PF2 |

For more information on how to use the Digital Peripheral MUX to connect peripheral signals to IO, see Register Summary

19.7 Timer B Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|---------------------------|--------|------------|
| TBCTL | 4007 0000h | Timer B control | RW | 0000 0000h |
| TBINT | 4007 0004h | Timer B interrupt control | RW | 0000 0000h |
| TBPRD | 4007 0008h | Timer B period | RW | 0000 0000h |
| TBCTR | 4007 000Ch | Timer B counter | RW | 0000 0000h |
| TBQEPCTL | 4007 0010h | Timer B QEP control | RW | 0000 0000h |
| TBCCTL0 | 4007 0100h | Timer B CC control 0 | RW | 0000 0000h |
| TBCCTR0 | 4007 0104h | Timer B CC counter 0 | RW | 0000 0000h |
| TBCCTL1 | 4007 0108h | Timer B CC control 1 | RW | 0000 0000h |
| TBCCTR1 | 4007 010Ch | Timer B CC counter 1 | RW | 0000 0000h |
| TBCCTL2 | 4007 0110h | Timer B CC control 2 | RW | 0000 0000h |
| TBCCTR2 | 4007 0114h | Timer B CC counter 2 | RW | 0000 0000h |
| TBCCTL3 | 4007 0118h | Timer B CC control 3 | RW | 0000 0000h |
| TBCCTR3 | 4007 011Ch | Timer B CC counter 3 | RW | 0000 0000h |
| TBCCTL4 | 4007 0120h | Timer B CC control 4 | RW | 0000 0000h |
| TBCCTR4 | 4007 0124h | Timer B CC counter 4 | RW | 0000 0000h |
| TBCCTL5 | 4007 0128h | Timer B CC control 5 | RW | 0000 0000h |
| TBCCTR5 | 4007 012Ch | Timer B CC counter 5 | RW | 0000 0000h |
| TBCCTL6 | 4007 0130h | Timer B CC control 6 | RW | 0000 0000h |
| TBCCTR6 | 4007 0134h | Timer B CC counter 6 | RW | 0000 0000h |
| TBCCTL7 | 4007 0138h | Timer B CC control 7 | RW | 0000 0000h |
| TBCCTR7 | 4007 013Ch | Timer B CC counter 7 | RW | 0000 0000h |
| TBDTGCTL0 | 4007 0200h | Timer B DTG control 0 | RW | 0000 0000h |
| TBDTGCTL1 | 4007 0204h | Timer B DTG control 1 | RW | 0000 0000h |
| TBDTGCTL2 | 4007 0208h | Timer B DTG control 2 | RW | 0000 0000h |
| TBDTGCTL3 | 4007 020Ch | Timer B DTG control 3 | RW | 0000 0000h |

19.8 Register Detail

19.8.1 TBCTL

Register 19-1 TBCTL (Timer B Control, 4007 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | BASEIE | RW | 0 | Base timer interrupt enable: 0b: not enabled 1b: enabled |
| 12 | CLR | RW | 0 | Timer Clear: 0b: do not clear 1b: clear timer counter |
| 11 | LATCH | RW | 0 | When written to 1b, this will latch the TBPRD and all TBCCTRn registers on the same clock cycle. Writing this bit to 0b has no effect. This is a self-clearing bit. |
| 10 | DTGCLK | RW | 0 | DTG Clock Source: 0b: PCLK 1b: ACLK |
| 9 | CLKSRC | RW | 0 | Timer Clock Source: 0b: PCLK 1b: ACLK |
| 8:6 | CLKDIV | RW | 0 | Timer Input Clock Divider: 000b: /1 001b: /2 010b: /4 011b: /8 100b: /16 101b: /32 110b: /64 111b: /128 |
| 5 | SINGLE | RW | 0 | Single Shot Timer: 0b: disabled (auto-reload) 1b: enabled (single shot timer) |
| 4 | SSYNC | RW | 0 | Timer Slave Synchronization: 0b: disabled 1b: enabled |
| 3:2 | PRDLATCH | RW | 0 | Timer Period Latch Mode: 00b: Latch TBPRD when TBCTR = 0 01b: Latch TBPRD when TBCTR = TBPRD 10b: Latch TBPRD immediately upon register write 11b: reserved |
| 1:0 | MODE | RW | 0 | Timer Mode: |

| | | | | |
|--|--|--|--|--|
| | | | | 00b: disabled 01b: up mode 10b: up/down mode 11b: asymmetric mode |
|--|--|--|--|--|

19.8.2 TBINT

Register 19-2 TBINT (Timer B Interrupt Control, 4007 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:9 | Reserved | RO | 0 | Reserved |
| 8 | BASEIF | W1C | 0 | Base timer interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 7 | CCR7IF | W1C | 0 | CCR7 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 6 | CCR6IF | W1C | 0 | CCR6 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 5 | CCR5IF | W1C | 0 | CCR5 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 4 | CCR4IF | W1C | 0 | CCR4 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 3 | CCR3IF | W1C | 0 | CCR3 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 2 | CCR2IF | W1C | 0 | CCR2 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 1 | CCR1IF | W1C | 0 | CCR1 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 0 | CCR0IF | W1C | 0 | CCR0 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |

19.8.3 TBPRD

Register 19-3 TBPRD (Timer A Period, 4007 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--------------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | PERIOD | RW | 0 | Timer period value |

19.8.4 TBCTR

Register 19-4 TBCTR (Timer B Counter, 4007 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | COUNTER | RW | 0 | Timer counter |

19.8.5 TBQEPCTL

Register 19-5 TBQEPCTL (Timer B QEP Control, 4007 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | TICKS | RW | 0 | Number of counted phase edges according to counting mode defined by TBQEPCTL.CNTEEDGE and TBQEPCTL.CNTAB . |
| 15 | Reserved | RO | 0 | Reserved |
| 14 | CNTEEDGE | RW | 0 | Count on edge: 0b: Rising edge only 1b: Rising and falling edge |
| 13 | CNTAB | RW | 0 | Count on A/B: 0b: Count phase A only 1b: Count phase A and phase B |
| 12 | IDXEIVF | W1C | 0 | Index event interrupt flag: 0b: no index event detected 1b: index event detected |
| 11 | WRIF | W1C | 0 | Counter wrap interrupt flag: 0b: no counter wrap 1b: counter wrap |
| 10 | PHBIF | W1C | 0 | Phase B rising edge interrupt flag: 0b: no phase B rising edge detected 1b: phase B rising edge detected |
| 9 | PHAIF | W1C | 0 | Phase A rising edge interrupt flag: 0b: no phase A rising edge detected 1b: phase A rising edge detected |
| 8 | DIRIF | W1C | 0 | Direction change interrupt flag: 0b: no direction change detected 1b: direction change detected |
| 7 | IDXRST | RW | 0 | Reset counter on index: 0b: Do not reset TBQEPCTL.TICKS 1b: Reset TBQEPCTL.TICKS on all index events |
| 6 | IDXEIVIE | RW | 0 | Index event interrupt enable: 0b: disabled 1b: enabled |
| 5 | WRIE | RW | 0 | Counter wrap interrupt enable: 0b: disabled 1b: enabled |
| 4 | PHBIE | RW | 0 | Phase B rising edge interrupt enable: 0b: disabled 1b: enabled |
| 3 | PHAIE | RW | 0 | Phase A rising edge interrupt enable: |

| | | | | |
|---|--------------|----|---|---|
| | | | | 0b: disabled 1b: enabled |
| 2 | DIRIE | RW | 0 | Direction change interrupt enable: 0b: disabled 1b: enabled |
| 1 | DIR | RO | 0 | Motor direction: 0b: CW (clockwise) 1b: CCW (counter-clockwise) |
| 0 | QEPEN | RW | 0 | QEP peripheral enabled: 0b: disabled 1b: enabled |

19.8.6 TBCCTL0

Register 19-6 TBCCTL0 (Timer B CCR Control 0, 4007 0100h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL0.CCMODE = 0b (compare): 00b: Latch TBCCTRL0 registers when TBCTR = 0 01b: Latch TBCCTRL0 registers when TBCTR = TBPRD 10b: Latch TBCCTRL0 registers immediately 11b: Reserved If TBCCTL0.CCMODE = 1b (capture): 00b: Latch TACTR into TACCTR0.CTR on rising edge 01b: Latch TACTR into TACCTR0.CTR on falling edge 10b: Latch TACTR into TACCTR0.CTR on both edges 11b: Latch TACTR into TACCTR0.CAP1 and TACCTR0.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL0.CCMODE = 1b (capture), set TBINT.CCR0IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.7 TBCCTR0

Register 19-7 TBCCTR0 (Timer B CCR Counter 0, 4007 0104h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|--|
| 31:16 | DELAY | RW | 0 | If TBCCTL0.CCLATCH != 11b (period/duty mode) and TACCTL0.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTL0.CCLATCH = 11b (period/duty mode) and TBCCTL0.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTL0.CCLATCH != 11b (period/duty mode) and TBCCTRL0.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTL0.CCLATCH != 11b (period/duty mode) and TBCCTL0.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTL0.CCLATCH = 11b (period/duty mode) and TBCCTL0.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.8 TBCCTL1

Register 19-8 TBCCTL1 (Timer A CCR Control 1, 4007 0108h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL1.CCMODE = 0b (compare): 00b: Latch TBCCTL1 registers when TBCTR = 0 01b: Latch TBCCTL1 registers when TBCTR = TBPRD 10b: Latch TBCCTL1 registers immediately 11b: Reserved If TBCCTL1.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR1.CTR on rising edge 01b: Latch TBCTR into TBCCTR1.CTR on falling edge 10b: Latch TBCTR into TBCCTR1.CTR on both edges 11b: Latch TBCTR into TBCCTR1.CAP1 and TBCCTR1.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL1.CCMODE = 1b (capture), set TBINT.CCR1IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.9 TBCCTR1

Register 19-9 TBCCTR1 (Timer B CCR Counter 1, 4007 010Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTL1.CCLATCH != 11b (period/duty mode) and TBCCTL1.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTL1.CCLATCH = 11b (period/duty mode) and TBCCTL1.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTL1.CCLATCH != 11b (period/duty mode) and TBCCTL1.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTL1.CCLATCH != 11b (period/duty mode) and TBCCTL1.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTL1.CCLATCH = 11b (period/duty mode) and TBCCTL1.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.10 TBCCTL2

Register 19-10 TBCCTL2 (Timer B CCR Control 2, 4007 0110h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL2.CCMODE = 0b (compare): 00b: Latch TBCCTL2 registers when TBCTR = 0 01b: Latch TBCCTL2 registers when TBCTR = TBPRD 10b: Latch TBCCTL2 registers immediately 11b: Reserved If TBCCTL2.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR2.CTR on rising edge 01b: Latch TBCTR into TBCCTR2.CTR on falling edge 10b: Latch TBCTR into TBCCTR2.CTR on both edges 11b: Latch TBCTR into TBCCTR2.CAP1 and TBCCTR2.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL2.CCMODE = 1b (capture), set TBINT.CCR2IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.11 TBCCTR2

Register 19-11 TBCCTR2 (Timer B CCR Counter 2, 4007 0114h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL2.CCLATCH != 11b (period/duty mode) and TBCCTRL2.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL2.CCLATCH = 11b (period/duty mode) and TBCCTRL2.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL2.CCLATCH != 11b (period/duty mode) and TBCCTRL2.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL2.CCLATCH != 11b (period/duty mode) and TBCCTRL2.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL2.CCLATCH = 11b (period/duty mode) and TBCCTRL2.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.12 TBCCTL3

Register 19-12 TBCCTL3 (Timer B CCR Control 3, 4007 0118h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL3.CCMODE = 0b (compare): 00b: Latch TBCCTL3 registers when TBCTR = 0 01b: Latch TBCCTL3 registers when TBCTR = TBPRD 10b: Latch TBCCTL3 registers immediately 11b: Reserved If TBCCTL3.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR3.CTR on rising edge 01b: Latch TBCTR into TBCCTR3.CTR on falling edge 10b: Latch TBCTR into TBCCTR3.CTR on both edges 11b: Latch TBCTR into TBCCTR3.CAP1 and TBCCTR3.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL3.CCMODE = 1b (capture), set TBINT.CCR3IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.13 TBCCTR3

Register 19-13 TBCCTR3 (Timer B CCR Counter 3, 4007 011Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL3.CCLATCH != 11b (period/duty mode) and TBCCTRL3.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL3.CCLATCH = 11b (period/duty mode) and TBCCTRL3.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL3.CCLATCH != 11b (period/duty mode) and TBCCTRL3.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL3.CCLATCH != 11b (period/duty mode) and TBCCTRL3.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL3.CCLATCH = 11b (period/duty mode) and TBCCTRL3.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.14 TBCCTL4

Register 19-14 TBCCTL4 (Timer B CCR Control 4, 4007 0120h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL4.CCMODE = 0b (compare): 00b: Latch TBCCTL4 registers when TBCTR = 0 01b: Latch TBCCTL4 registers when TBCTR = TBPRD 10b: Latch TBCCTL4 registers immediately 11b: Reserved If TBCCTL4.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR4.CTR on rising edge 01b: Latch TBCTR into TBCCTR4.CTR on falling edge 10b: Latch TBCTR into TBCCTR4.CTR on both edges 11b: Latch TBCTR into TBCCTR4.CAP1 and TBCCTR4.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL4.CCMODE = 1b (capture), set TBINT.CCR4IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.15 TBCCTR4

Register 19-15 TBCCTR4 (Timer B CCR Counter 4, 4007 0124h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL4.CCLATCH != 11b (period/duty mode) and TBCCTRL4.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL4.CCLATCH = 11b (period/duty mode) and TBCCTRL4.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL4.CCLATCH != 11b (period/duty mode) and TBCCTRL4.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL4.CCLATCH != 11b (period/duty mode) and TBCCTRL4.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL4.CCLATCH = 11b (period/duty mode) and TBCCTRL4.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.16 TBCCTL5

Register 19-16 TBCCTL5 (Timer B CCR Control 5, 4007 0128h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL5.CCMODE = 0b (compare): 00b: Latch TBCCTL5 registers when TBCTR = 0 01b: Latch TBCCTL5 registers when TBCTR = TBPRD 10b: Latch TBCCTL5 registers immediately 11b: Reserved If TBCCTL5.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR5.CTR on rising edge 01b: Latch TBCTR into TBCCTR5.CTR on falling edge 10b: Latch TBCTR into TBCCTR5.CTR on both edges 11b: Latch TBCTR into TBCCTR5.CAP1 and TBCCTR5.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL5.CCMODE = 1b (capture), set TBINT.CCR5IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.17 TBCCTR5

Register 19-17 TBCCTR5 (Timer B CCR Counter 5, 4007 012Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL5.CCLATCH != 11b (period/duty mode) and TBCCTRL5.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL5.CCLATCH = 11b (period/duty mode) and TBCCTRL5.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL5.CCLATCH != 11b (period/duty mode) and TBCCTRL5.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL5.CCLATCH != 11b (period/duty mode) and TBCCTRL5.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL5.CCLATCH = 11b (period/duty mode) and TBCCTRL5.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.18 TBCCTL6

Register 19-18 TBCCTL6 (Timer B CCR Control 6, 4007 0130h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL6.CCMODE = 0b (compare): 00b: Latch TBCCTL6 registers when TBCTR = 0 01b: Latch TBCCTL6 registers when TBCTR = TBPRD 10b: Latch TBCCTL6 registers immediately 11b: Reserved If TBCCTL6.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR6.CTR on rising edge 01b: Latch TBCTR into TBCCTR6.CTR on falling edge 10b: Latch TBCTR into TBCCTR6.CTR on both edges 11b: Latch TBCTR into TBCCTR6.CAP1 and TBCCTR6.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL6.CCMODE = 1b (capture), set TBINT.CCR6IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.19 TACCTR6

Register 19-19 TBCCTR6 (Timer B CCR Counter 6, 4007 0134h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL6.CCLATCH != 11b (period/duty mode) and TBCCTRL6.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL6.CCLATCH = 11b (period/duty mode) and TBCCTRL6.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL6.CCLATCH != 11b (period/duty mode) and TBCCTRL6.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL6.CCLATCH != 11b (period/duty mode) and TBCCTRL6.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL6.CCLATCH = 11b (period/duty mode) and TBCCTRL6.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.20 TBCCTL7

Register 19-20 TBCCTL7 (Timer B CCR Control 7, 4007 0138h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TBCCTL7.CCMODE = 0b (compare): 00b: Latch TBCCTL7 registers when TBCTR = 0 01b: Latch TBCCTL7 registers when TBCTR = TBPRD 10b: Latch TBCCTL7 registers immediately 11b: Reserved If TBCCTL7.CCMODE = 1b (capture): 00b: Latch TBCTR into TBCCTR7.CTR on rising edge 01b: Latch TBCTR into TBCCTR7.CTR on falling edge 10b: Latch TBCTR into TBCCTR7.CTR on both edges 11b: Latch TBCTR into TBCCTR7.CAP1 and TBCCTR7.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TBCCTL7.CCMODE = 1b (capture), set TBINT.CCR7IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

19.8.21 TBCCTR7

Register 19-21 TBCCTR7 (Timer B CCR Counter 7, 4007 013Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TBCCTRL7.CCLATCH != 11b (period/duty mode) and TBCCTRL7.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TBCCTRL7.CCLATCH = 11b (period/duty mode) and TBCCTRL7.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TBCCTRL7.CCLATCH != 11b (period/duty mode) and TBCCTRL7.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TBCCTRL7.CCLATCH != 11b (period/duty mode) and TBCCTRL7.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TBCCTRL7.CCLATCH = 11b (period/duty mode) and TBCCTRL7.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

19.8.22 TBDTGCTL0

Register 19-22 TBDTGCTL0 (Timer B Dead-Time Generator Control 0, 4007 0200h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG0. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG0. |

19.8.23 TBDTGCTL1

Register 19-23 TBDTGCTL1 (Timer B Dead-Time Generator Control 1, 4007 0204h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG1. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG1. |

19.8.24 TBDTGCTL2

Register 19-24 TBDTGCTL2 (Timer B Dead-Time Generator Control 2, 4007 0208h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG2. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG2. |

19.8.25 TBDTGCTL3

Register 19-25 TBDTGCTL3 (Timer B Dead-Time Generator Control 3, 4007 020Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG3. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG3. |

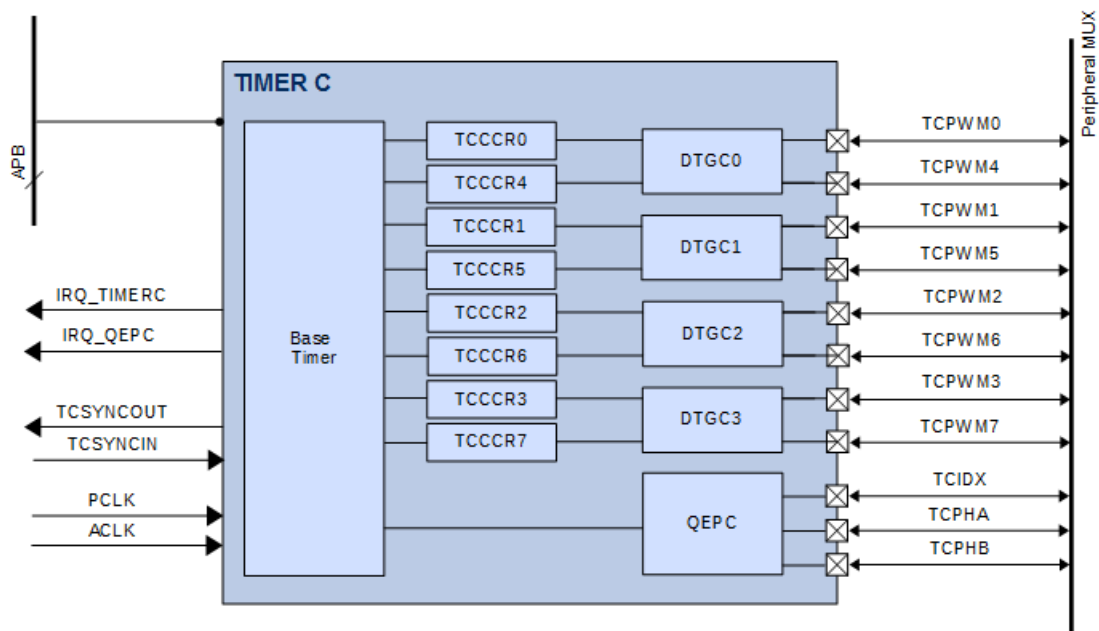
20 PWM TIMER C

20.1 Overview

All devices in the PAC55XX family of controllers have a Timer C peripheral. This peripheral is a 16-bit timer that allows support for 8 Capture and Compare Units (CCR) capable of PWM generation; capture input processing and a QEP encoder for various control applications.

Below is a simplified block diagram of the Timer peripheral.

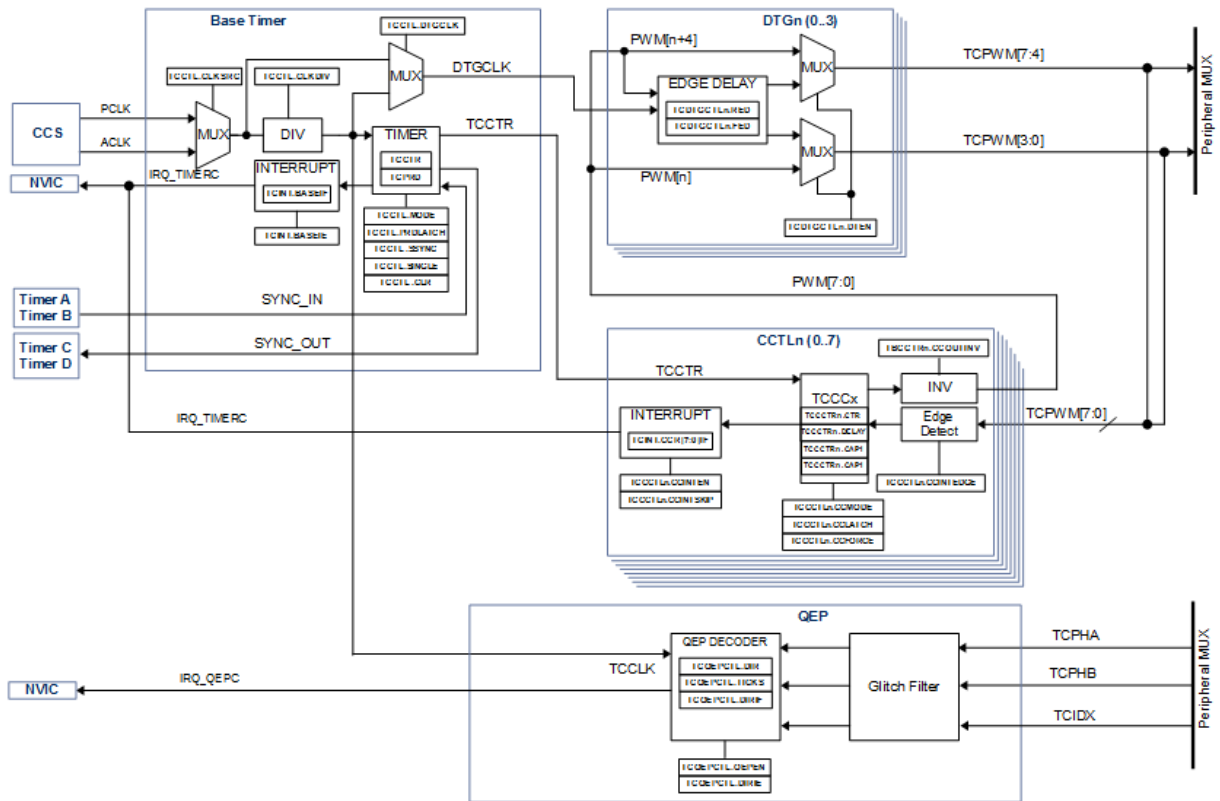
Figure 20-1 Timer C Simplified Block Diagram



The timer module is an APB bus client. There are 8 timer channel input/outputs to the digital peripheral MUX, in addition to three IOs for QEP. Each timer block has two interrupt signal outputs to the NVIC; one for timer functions and one for QEP functions. The user may select either PCLK or ACLK as the clock input for each timer module.

20.2 Timer C Block Diagram

Figure 20-2 Timer C Block Diagram



20.3 Features

The PWM Timer peripheral has the following features in the PAC55XX family of controllers.

General Features:

- Configurable input clock: PCLK or ACLK
- Up to 300MHz clock input for 3.33ns PWM edge resolution
- 3-bit input clock divider
- Latch timer period and all CCR values on command

Base Timer Features:

- Single-shot or auto-reload
- Base timer interrupts
- Timer synchronization
- Timer Modes:
 - Disabled
 - Up mode
 - Up/Down mode
 - Asymmetric Up mode
- Timer Register Latching Options
 - Latch TAPRD when counter = 0
 - Latch TAPRD when counter = period
 - Latch TAPRD immediately
- Dead-time Generator Input Clock
 - DTG clock = PCLK
 - DTG clock = ACLK

CCR/PWM Features:

- PWM output or input capture
- 8 CCR units per timer
- CCR interrupts
- CCR interrupt skips
- SW force CCR interrupt
- CCR interrupt type:
 - Rising, falling, both
- CCR latch modes:
 - Compare/PWM mode: counter = 0, counter = period, immediate
 - Capture input: rising edge, falling edge, both
- Period/duty calculation mode
- Force compare event

- Invert CCR output
- CCR phase delay for phase shifted drive topologies
- ADC trigger outputs:
 - PWM rising or falling edge

Dead-Time Generator (DTG) Features:

- DTG enable or bypass
- 12-bit rising edge delay
- 12-bit falling edge delay

QEP Decoder Features:

- QEP encoder enabled
- Direction status
- Configurable interrupts:
 - Phase A rising edge
 - Phase B rising edge
 - Index event
 - Counter wrap
- 4 Different counting modes for best resolution, range and speed performance

20.4 Functional Description

20.4.1 Timer Clock Structure

The timer peripheral input clock can be selected as either PCLK or ACLK. The input clock may be selected by the **TCCTL.CLKSRC** register.

Each timer peripheral has a 4-bit divider that can be used to divide the selected input clock. The user may set this divider by the **TCCTL.CLKDIV** register. There are 16 settings between /1 and /128 for the input clock divider.

The base timer also supplies the clock for the Dead-Time Generators (DTGs) to allow for different range and resolution for dead-time. The timer may select the DTGCLK to be the timer clock before or after the clock divider. The DTG input clock may be selected by the **TCCTL.DTGCLK** register.

To use the timer clock before the input clock divider, set **TCCTL.DTGCLK** to 0b. To use the timer clock after the input clock divider, set **TCCTL.DTGCLK** to 1b.

20.4.2 Timer Counter

The base timer is a 16-bit timer that can count either up, or up then down to support both edge and center aligned and asymmetric PWM output types.

The timer period is stored in the **TCPRD** register. The current value of the timer counter is updated at every timer tick and is stored in the **TCCTR** register. The **TCCTR** register is a RW register, so it may be updated at any time and is changed immediately.

The **TCPRD** register and all of the **TCCTRn** registers have shadow copies that are updated at a user-specified time. See the section below on Base Timer **TAPRD** Latching and CCR Timer Latching for more information.

When **TCCTL.MODE** = 00b, the timer is disabled. Even when disabled, all timer registers are accessible via the APB bus.

To enable the timer, set the **TCCTL.MODE** to 01b (up mode), 10b (up/down mode) or 11b (up/down asymmetric mode).

When **TCCTL.MODE** = 01b, the timer is configured in up mode. The timer will count from 0 to **TCPRD**. The counter is updated in **TCCTR** at every timer tick. If auto-reload is active (**TCCTL.SINGLE** = 0b), then the timer count will automatically count from **TCPRD** to 0 and continue counting up. If auto-reload is not active (**TCCTL.SINGLE** = 1b), then the timer will count from 0 to **TCPRD** and stop. When this happens, the timer will set the **TCCTL.MODE** to 00b (disabled).

When the timer is configured for up/down mode (**TCCTL.MODE** = 10b) or configured for up/down asymmetric mode (**TCCTL.MODE** = 11b), the timer will count from 0 to **TCPRD** and then back down to 0. If auto-reload is not active (**TCCTL.SINGLE** = 1b), the time timer will stop

and set **TCCTL.MODE** to 00b (disabled). If auto-reload is active (**TCCTL.SINGLE** = 0b), then the timer count will then continue counting back up after it reaches 0.

20.4.3 Up Mode

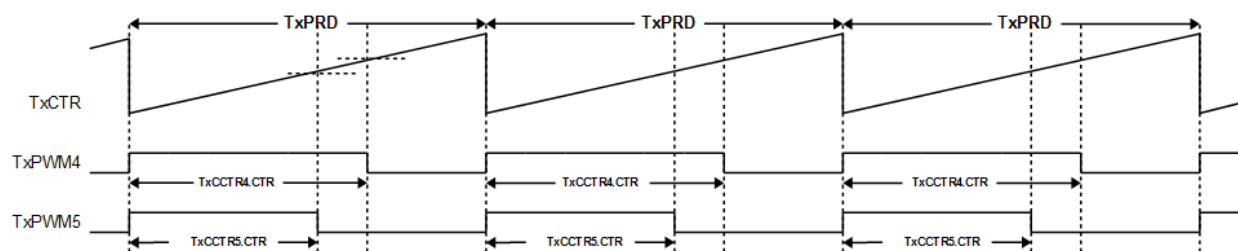
When the **TCCTL.MODE** is set to 01b (up mode) the timer will begin counting up from 0. The **TCCTR** will count to the value of **TCPRD** and then will reset to 0.

If the timer is configured for auto-reload mode (**TCCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it reaches 0.

The **TCCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate an edge aligned PWM as shown below.

Figure 20-3 Up Mode PWM Waveform



In this mode, when **TCCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TCCTR** register reaches the **TCCCTRn.CTR** threshold. When the **TCCTR** counts from **TCPRD** to 0, the PWM output is transitioned to high.

20.4.4 Up/Down Mode

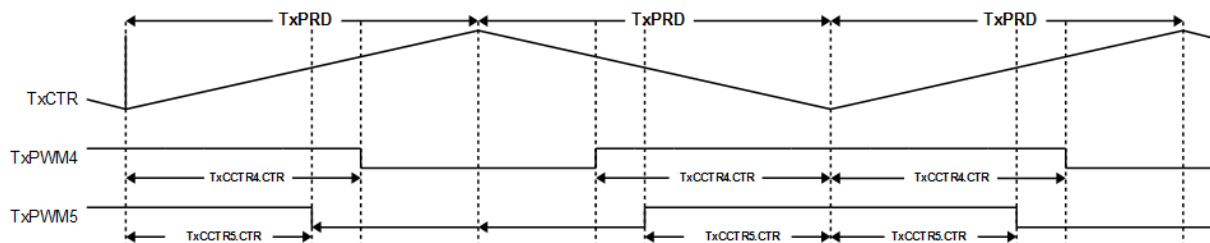
When the **TCCTL.MODE** is set to 10b (up/down mode) the timer will begin counting up from 0. The **TCCTR** will count to the value of **TCPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TCCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TCCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned PWM as shown below.

Figure 20-4 Up/Down Mode PWM Waveform



In this mode, when **TCCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TBCTR** register reaches the **TCCCTRn.CTR** threshold. When the **TCCTR** counts from **TCPRD - 1** to **TCPRD**, it will begin counting down. When the **TCCTR** counts down to **TCCCTRn.CTR** the PWM output is transitioned to high.

20.4.5 Up/Down Asymmetric mode

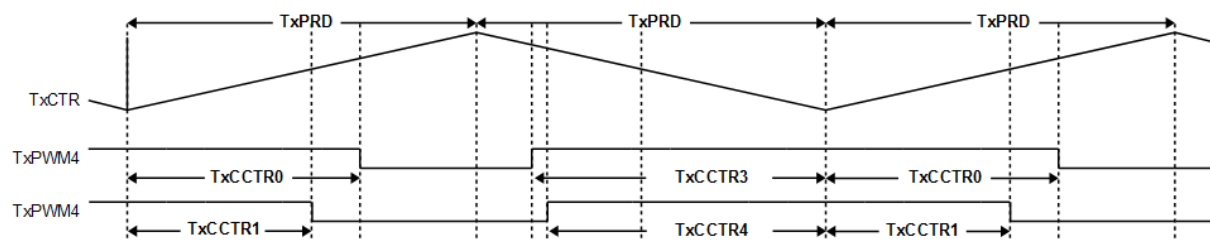
When the **TCCTL.MODE** is set to 11b (up/down asymmetric mode) the timer will begin counting up from 0. The **TCCTR** will count to the value of **TCPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TCCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TCCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned asymmetric PWM as shown below.

Figure 20-5 Up/Down Asymmetric Mode PWM Waveform



In this mode, when **TCCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TCCTR** register reaches the **TCCCTRn.CTR** threshold. When the **TCCTR** counts from **TCPRD - 1** to **TCPRD**, it will begin counting down. When the **TCCTR** counts down to **TCCCTRn.CTR** the PWM output is transitioned to high.

In this mode, the CCR outputs are generated such that 2 **TCCCTRn.CTR** registers are used to generate a single PWM output, so that the center-aligned PWM can be generated with asymmetric on-time.

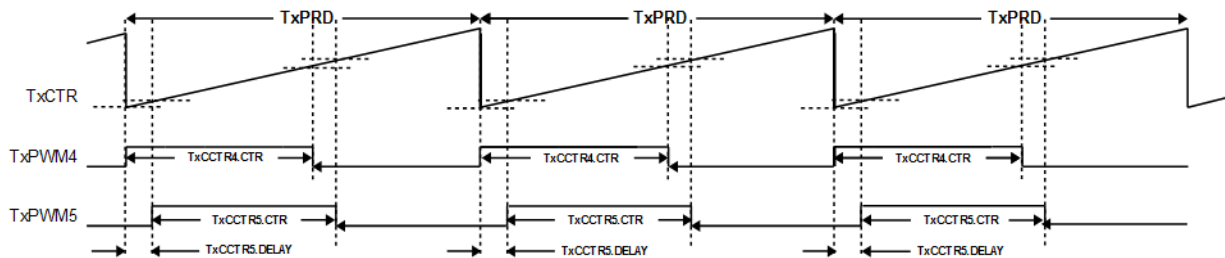
In this mode, the PWM outputs of the timer are generated as follows:

- TCPWM4 uses **TCCCTR0.CTR** for the first portion of the on time and **TCCCTR4.CTR** for the second portion of the on time
- TCPWM5 uses **TCCCTR1.CTR** for the first portion of the on time and **TCCCTR5.CTR** for the second portion of the on time
- TCPWM6 uses **TCCCTR2.CTR** for the first portion of the on time and **TCCCTR6.CTR** for the second portion of the on time
- TCPWM7 uses **TCCCTR3.CTR** for the first portion of the on time and **TCCCTR7.CTR** for the second portion of the on time

20.4.6 Up Mode with Phase Delay

When the timer is configured for up mode (**TCMODE** = 01b) the CCR allows a delay to be applied to the output PWM period, to support phase delay control topologies. If the value of **TCCCTLn.DELAY** > 0, then the CCR will delay this number of ticks before starting the PWM period (transitioning the output signal from high to low).

Figure 20-6 Up Mode with Phase Delay



20.4.7 Timer Synchronization

The timer peripheral allows synchronization between the timers in the PAC55XX. The timers may be synchronized as follows:

- Timer A (master), Timer B (slave)
- Timer A (master), Timer B (slave), Timer C (slave)
- Timer A (master), Timer B (slave), Timer C (slave), Timer D (slave)

The timer master's SYNC_OUT signals are connected to the downstream timer slave's SYNC_IN signal.

When configured, the timers can synchronize their clocks so they are sharing the same time base. In order to do this, the timers must be configured with the same clock source (**TCCTL.CLKSRC**) and clock divider (**TCCTL.CLKDIV**).

To configure the timers for synchronization, follow these steps:

- While the timer master is disabled (**TCCTL.MODE** = 00b), set the timer master **TxCTL.SSYNC** = 0b
- For each of the downstream timer slaves, while the timer is disabled **TCCTL.MODE** = 00b, set the **TCCTL.SSYNC** to 1b. Make sure that **TCCTL.CLKDIV** and **TCCTL.CLKSRC** for each of the timer slaves is the same as the timer master.
- Enable each of the slave timers by setting **TCCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric). The slave timers will not start counting since the **TCCTL.SSYNC** is set to a 1b.
- Enable the timer master by setting **TxCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric).

At this point, the master and all configured timer slaves will begin counting.

20.4.8 Base Timer TAPRD Latching

The timer peripherals have a shadow copy of the **TCPRD** register that is used for the counting operations. The latching of the data from the **TCPRD** register to the shadow register is controlled by the timer configuration.

There are several options for latching of this register into the shadow copy. This behavior is controlled by the **TCCTL.PRDLATCH** setting and depends on the setting of the **TCCTL.MODE** register as well.

20.4.8.1 TCPRD Latch When TCCTR = 0

- If **TCCTL.MODE** = 01b (up mode) and **TCCTL.PRDLATCH** = 00b (**TCCTR** = 0), then the **TCPRD** register is copied into the shadow register when **TCCTR** counts from **TCPRD** to 0.
- If **TCCTL.MODE** = 10b (up/down mode) or if **TCCTL.MODE** = 11b (up/down asymmetric mode) and **TCCTL.PRDLATCH** = 00b (**TCCTR** = 0), then the **TCPRD** register is copied into the shadow register when the **TCCTR** counts from 1 to 0.

20.4.8.2 TCPRD Latch When TCCTR = TCPRD

- If **TCCTL.MODE** = 01b (up mode) or if **TCCTL.MODE** = 10b (up/down mode) or if **TCCTL.MODE** = 11b (up/down asymmetric mode) and **TCCTL.PRDLATCH** = 01b (**TCCTR** = **TCPRD**), then the **TCPRD** register is copied into the shadow register when **TCCTR** counts from **TCPRD** – 1 to **TCPRD**.

20.4.8.3 TCPRD Latch Immediate

- If **TCCTL.PRDLATCH** = 11b (immediate), then the **TCPRD** register is copied into the shadow register as soon as the **TCPRD** register is written.

20.4.9 CCR Timer Latching

When in compare mode, the CCR units support configurable latching of the **TCCCRn.CTR** values into a shadow register that is used for timer operation. The shadow register is used for generating the PWM output when the CCR is in compare mode.

This feature can be used to control when the new duty cycle is applied, during the PWM period.

When the CCR is in compare mode (**TCCCTLn.CCMODE** = 0b), the shadow register is latched as follows:

- If **TCCCTLn.CCLATCH** = 00b (**TCCTR** = 0)
 - If **TCCTL.MODE** = 01b (up mode), **TCCCTRn.CTR** is copied into the shadow register when **TCCTR** counts from **TCPRD** to 0. If **TCCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode), **TCCCTRn.CTR** is copied into the shadow register when **TCCTR** counts from 1 to 0.
- If **TCCCTLn.CCLATCH** = 01b (**TCCTR** = **TCPRD**), then the **TCCCTRn.CTR** is copied into the shadow register when **TCCTR** counts from **TCPRD** – 1 to **TCPRD**.
- If **TCCCTLn.CCLATCH** = 10b (latch immediate), then the **TCCCTRn.CTR** is copied into the shadow register as soon as it is written.

When the CCR is in capture mode (**TCCCTLn.CCMODE** = 1b), the configuration of **TCCCTLn.CCLATCH** controls when the **TCCTR** is copied into the **TCCCTRn.CTR** register.

- If **TCCCTLn.CCLATCH** = 00b (rising edge), then the value of **TCCTR** is copied into the **TCCCTL.CTR** register upon a rising edge in the input signal.
- If **TCCCTLn.CCLATCH** = 01b (falling edge), then the value of the **TCCTR** is copied into the **TCCCTLn.CTR** register upon a falling edge in the input signal.
- If **TCCCTLn.CCLATCH** = 10b (both), then the value of the **TCCTR** is copied into the **TCCCTLn.CTR** register on both a rising and falling edge of the input signal.
- If **TCCCTLn.CCLATCH** = 11b (period/duty mode), then see the section below on how the values are latched into the **TCCCTRn** register.

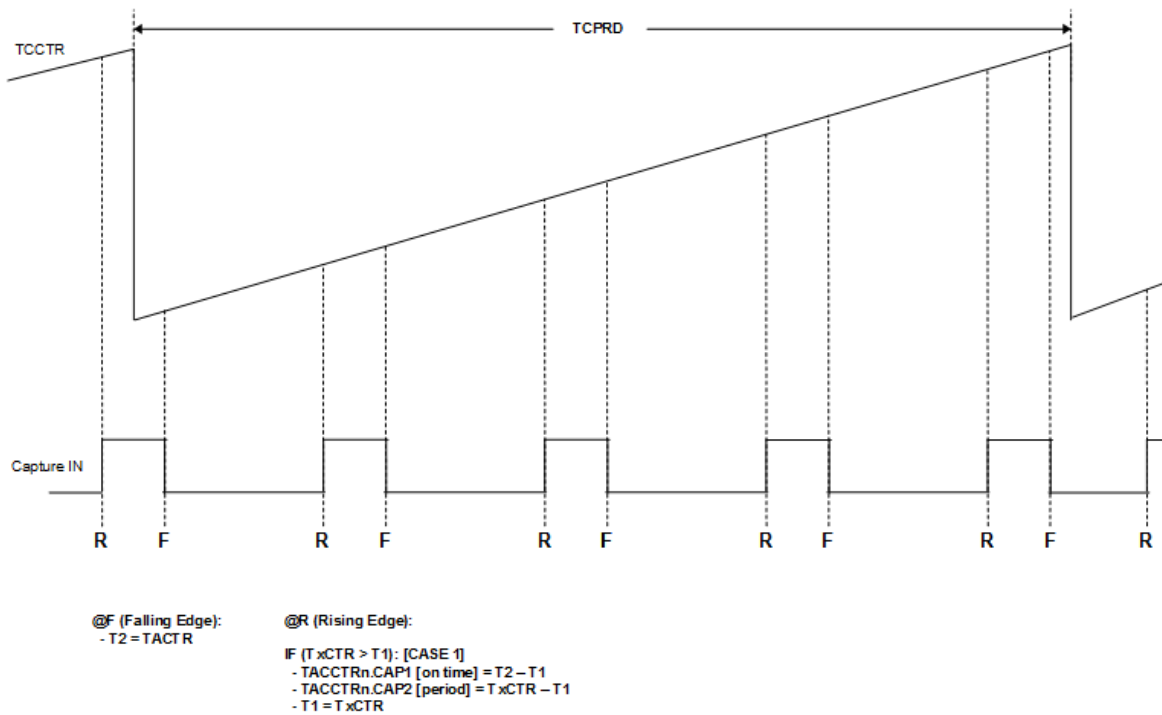
20.4.10 CCR Capture Period/Duty Mode

If the **TCCCTLn.CCLATCH = 11b** (period/duty mode), then the CCR can capture the number of timer ticks for the input PWM duty cycle and period. This mode is useful for measuring input pulse widths for PPM and other type of control applications.

At each rising edge of the input signal, the following happens:

- The number of timer ticks for the last timer period's duty cycle is copied into **TCCCTRn.CAP1**
- The number of timer ticks for the last timer period is copied into **TCCCTRn.CAP2**

Figure 20-7 Period/Duty Capture Mode PWM Waveform



20.4.11 Timer Whole Latching

It is sometimes convenient to latch both the **TCPRD** (period) and all **TCCCTRn.CTR** (duty cycle) registers into the shadow registers at one time.

If the **TCCTL.LATCH** bit is written to a 1b, then all the **TCPRD** and all **TCCCTRn.CTR** registers will be copied into the shadow registers on the same clock cycle.

The **TCCTL.LATCH** bit is self-clearing, and will always be read as a 0b.

20.4.12 Inverting CCR PWM Output

When the CCR is in compare mode (**TCCCTRn.CCMODE** = 0b), the output of the CCR (the input to the DTG) may be inverted. To invert this output, set the **TCCCTRn.CCOUTINV** = 1b.

This mode is useful for some control topologies, especially full-bridge.

20.4.13 Base Timer Interrupts

The base timer may be configured to generate an interrupt to the NVIC.

If **TCCTL.MODE** = 01b (up mode) and **TCCTR** counts from **TCPRD** to 0 then the **TCINT.BASEIF** bit is set to 1b. If the **TCCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERC_IRQ** signal to the NVIC is asserted.

If **TCCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode) and **TCCTR** counts from 1 to 0, then the **TCINT.BASEIF** bit is set to 1b. If the **TCCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERC_IRQ** signal to the NVIC is asserted.

The **TCINT.BASEIF** may be cleared by writing a 1b to it.

20.4.14 CCR Compare Interrupts

When **TCCCTLn.CCMODE** = 0b (compare mode), the CCR unit may be configured to generate an interrupt to the NVIC.

When **TCCTR** counts to **TCCCTRn.CTR**, then the CCR interrupt flag in the **TCINT** register is set as follows:

- For CCR0, when **TCCTR** counts to **TCCCTR0.CTR**, then **TCINT.CCR0IF** is set to a 1b. If the **TCCCTR0.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR0IF** may be cleared by writing it to a 1b.
- For CCR1, when **TCCTR** counts to **TCCCTR1.CTR**, then **TCINT.CCR1IF** is set to a 1b. If the **TCCCTR1.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR1IF** may be cleared by writing it to a 1b.
- For CCR2, when **TCCTR** counts to **TCCCTR2.CTR**, then **TCINT.CCR2IF** is set to a 1b. If the **TCCCTR2.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR2IF** may be cleared by writing it to a 1b.

- For CCR3, when **TCCTR** counts to **TCCCTR3.CTR**, then **TCINT.CCR3IF** is set to a 1b. If the **TCCCTR3.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR3IF** may be cleared by writing it to a 1b.
- For CCR4, when **TCCTR** counts to **TCCCTR4.CTR**, then **TCINT.CCR4IF** is set to a 1b. If the **TCCCTR4.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR4IF** may be cleared by writing it to a 1b.
- For CCR5, when **TCCTR** counts to **TCCCTR5.CTR**, then **TCINT.CCR5IF** is set to a 1b. If the **TCCCTR5.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR5IF** may be cleared by writing it to a 1b.
- For CCR6, when **TCCTR** counts to **TCCCTR6.CTR**, then **TCINT.CCR6IF** is set to a 1b. If the **TCCCTR6.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR6IF** may be cleared by writing it to a 1b.
- For CCR7, when **TCCTR** counts to **TCCCTR7.CTR**, then **TCINT.CCR7IF** is set to a 1b. If the **TCCCTR7.CCINTEN** is set to 1b, then **TIMERC_IRQ** to the NVIC is asserted. **TCINT.CCR7IF** may be cleared by writing it to a 1b.

20.4.15 CCR Capture Interrupts

When **TCCCTLn.CCMODE** = 1b (capture mode), the CCR unit may be configured to generate an interrupt to the NVIC based on the input signal.

- If **TCCCTLn.CCEGE** = 00b (rising edge) and a rising edge is detected in the input signal, then the **TCINT.CCRnIF** interrupt flag is set.
- If **TCCCTLn.CCEDGE** = 01b (falling edge) and a falling edge is detected in the input signal, then the **TCINT.CCRnIF** interrupt flag is set.
- If the **TCCCTLn.CCEDGE** = 11b (rising or falling edge) and a rising or a falling edge is detected in the input signal, then the **TCINT.CCRnIF** interrupt flag is set.

If the **TCINT.CCRnIF** interrupt flag is set and the corresponding **TCCCTLn.CCINTEN** bit is set, then the **TIMERC_IRQ** signal to the NVIC will be asserted.

The **TCINT.CCRnIF** interrupt flag can be cleared by writing it to a 1b.

20.4.16 Timer IRQ signal

The timer unit has one IRQ output signal to the NVIC (**TIMERC_IRQ**).

This signal is asserted when any of the following conditions are true:

- **TCINT.BASEIF** = 1b and **TCCTL.BASEIE** = 1b
- **TCINT.CCR0IF** = 1b and **TCCCTL0.CCINTEN** = 1b
- **TCINT.CCR1IF** = 1b and **TCCCTL1.CCINTEN** = 1b
- **TCINT.CCR2IF** = 1b and **TCCCTL2.CCINTEN** = 1b
- **TCINT.CCR3IF** = 1b and **TCCCTL3.CCINTEN** = 1b
- **TCINT.CCR4IF** = 1b and **TCCCTL4.CCINTEN** = 1b

- **TCINT.CCR5IF = 1b and TCCCTL5.CCINTEN = 1b**
- **TCINT.CCR6IF = 1b and TCCCTL6.CCINTEN = 1b**
- **TCINT.CCR7IF = 1b and TCCCTL7.CCINTEN = 1b**

The TIMERC_IRQ signal is de-asserted when all of the above conditions are false.

20.4.17 Skipping CCR Interrupts

Sometimes it is useful to not generate a CCR interrupt to the NVIC every time the **TCCTR** counts to the **TCCCTRn.CTR** value. For example, in a control application when the PWM frequency is fast, but you only need a MCU interrupt every 5 PWM periods.

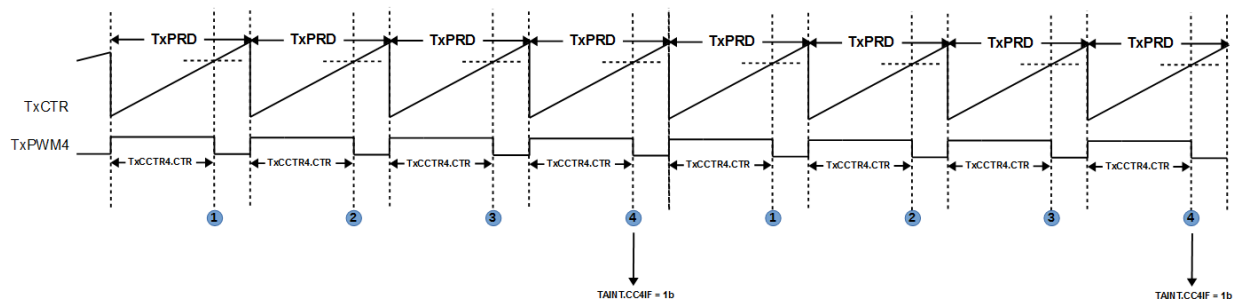
Each CCR in the timer has the ability to skip interrupts to the NVIC to allow this capability.

The CCR will generate interrupts to the NVIC according to the **TCCCTL.CCINTSKIP** field as follows:

- If **TCCCTLn.CCINTSKIP = 0000b**, then the CCR will generate an interrupt each time
- If **TCCCTLn.CCINTSKIP = 0001b**, then the CCR will skip 1 CCR cycle before generating an interrupt to the NVIC.
- If **TCCCTLn.CCINTSKIP = 0010b**, then the CCR will skip 2 CCR cycles before generating an interrupt to the NVIC.
- If **TCCCTLn.CCINTSKIP = 0011b**, then the CCR will skip 3 CCR cycles before generating an interrupt to the NVIC.
- ...
- If **TCCCTLn.CCINTSKIP = 1110b**, then the CCR will skip 14 CCR cycles before generating an interrupt to the NVIC.
- If **TCCCTLn.CCINTSKIP = 1111b**, then the CCR will skip 15 CCR cycles before generating an interrupt to the NVIC.

The diagram below shows an example of a CCR4 with interrupt skipping configured with a value of 4 (**TCCCTL.CCINTSKIP = 4**).

Figure 20-8 CCR Interrupt Skipping Diagram



20.4.18 Timer ADC Triggers

Each timer CCR unit may be configured to be an ADC trigger, to allow the DTSE to automatically begin conversion sequences.

For more information on this feature, see the section on the ADC and DTSE in this user guide.

20.4.19 Dead-Time Generators (DTG)

Each timer has 4 DTG units. Each DTG unit is capable of generating a pair of complementary signals that can be configured with dead-time to drive an inverter for half-bridge topologies.

The input clock to the DTG can be configured to be the timer clock before or after the **TCCTL.CLKDIV** input clock divider. If **TCCTL.DTGCLK** is 0b, then the DTGCLK is the clock before the **TCCTL.CLKDIV** clock divider. If **TCCTL.DTGCLK** is 1b, then the DTGCLK is the clock after the **TCCTL.CLKDIV** clock divider.

When **TCDTGCTLn.DTEN** = 0b, the DTG is disabled (bypassed). In this mode, the timer signals have the following behavior (see the simplified block diagram above):

DTG0:

- CCR0 output is connected to TCPWM0
- CCR4 output is connected to TCPWM4

DTG1:

- CCR1 output is connected to TCPWM1
- CCR5 output is connected to TCPWM5

DTG2:

- CCR2 output is connected to TCPWM2
- CCR6 output is connected to TCPWM6

DTG3:

- CCR3 output is connected to TCPWM3
- CCR7 output is connected to TCPWM7

When **TCDTGCTL0.DTEN** = 1b, the DTG is enabled. In this mode, only one CCR output is used to generate the two complementary outputs. In this mode, the timer signals have the following behavior:

DTG0:

- CCR0 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR4 output is used to generate the two complementary output signals: TCPWM0 and TCPWM4.
- TCPWM0 becomes the low-side complementary signal.
- TCPWM4 becomes the high-side complementary signal.

- The **TCDTGCTL0.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, to when the low-side signal rises).
- The **TCDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, to when the high-side signal rises).

DTG1:

- CCR1 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR5 output is used to generate the two complementary output signals: TCPWM1 and TCPWM5.
- TCPWM1 becomes the low-side complementary signal.
- TCPWM5 becomes the high-side complementary signal.
- The **TCDTGCTL1.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TCDTGCTL1.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG2:

- CCR2 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR6 output is used to generate the two complementary output signals: TCPWM2 and TCPWM6.
- TCPWM2 becomes the low-side complementary signal.
- TCPWM6 becomes the high-side complementary signal.
- The **TCDTGCTL2.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TCDTGCTL2.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG3:

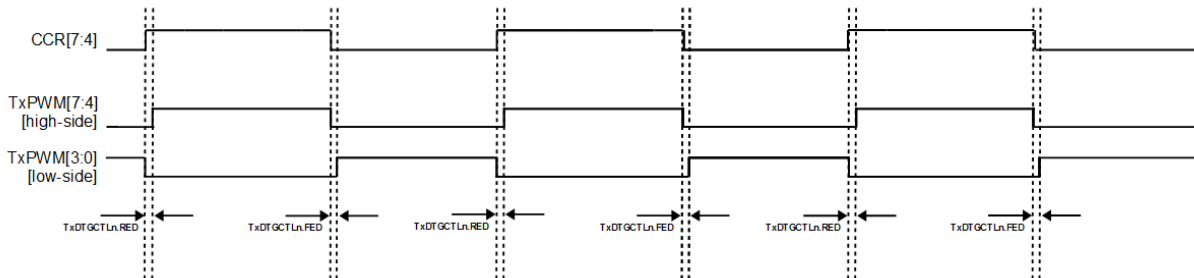
- CCR3 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR7 output is used to generate the two complementary output signals: TCPWM3 and TCPWM4.

- TCPWM3 becomes the low-side complementary signal.
- TCPWM7 becomes the high-side complementary signal.
- The **TCDTGCTL3.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TCDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

The PWM waveform below shows an example of the dead-time applied to the signals through the DTG units.

20.4.20 Timer C DTG Output

Figure 20-9 Timer C DTG Output



20.5 QEP

The PAC55XX contains a QEP encoder peripheral that can be used to determine position and speed of a position-encoded motor. The QEP encoder can be enabled by setting the **TCQEPCTL.QEPEN** bit to a 1b.¹²

The inputs to the QEP peripheral are TCQEPPHA, TCQEPPHB and TCQEPIDX (phase A, phase B and index). These inputs are used to determine the position and direction of the motor, when the **TCQEPCTL.QEPEN** is set to a 1b. These three inputs have a glitch filter applied to them, to filter out noise and are used for calculation of the motor position and speed.

After filtering, the TCQEPPHA, TCQEPPHB and TCQEPIDX inputs are passed to the QEP engine to determine the position and speed. If the position is determined to be clockwise, then the **TCQEPCTL.DIR** is set to 0b. If the position is determined to be counter-clockwise, then the **TCQEPCTL.DIR** is set to 1b.

During operation, if the **TCQEPCTL.CNTAB** bit is 0b, only phase A edges are counted. If this bit is set to a 1b, then both phase A and phase B edges are counted. If the **TCQEPCTL.CNTEDGE** bit is set to 0b, then only the rising edges are counted. If this bit is set to a 1b, then both rising and falling edges are counted.

If the **TCQEPCTL.IDXRST** bit is set to a 1b, then an index event clears the **TCQEPCTL.TICKS** counter to 0. If this bit is set to a 0b, then an index event does not reset the **TCQEPCTL.TICKS** counter.

If a change in the direction is made (0 to 1, or 1 to 0), then the **TCQEPCTL.DIRIF** bit is set to a 1b. If the **TCQEPCTL.DIRIE** bit is set to 1b, then the QEPC_IRQ signal is asserted to the NVIC. The **TCQEPCTL.DIRIF** bit can be cleared by writing it to 1b.

When a rising edge on phase A is detected the **TCQEPCTL.PHAIF** bit is set to 1b. If the **TCQEPCTL.PHAIE** bit is set and the **TCQEPCTL.PHAIF** bit is set, then the QEPC_IRQ signal is asserted to the NVIC. The **TCQEPCTL.PHAIF** bit can be cleared by writing a 1b to it.

When a rising edge on phase B is detected the **TCQEPCTL.PHBIF** bit is set. If the **TCQEPCTL.PHBIE** bit is set and the **TCQEPCTL.PHBIF** bit is set, then the QEPC_IRQ signal is asserted to the NVIC. The **TCQEPCTL.PHBIF** bit can be cleared by writing a 1b to it.

If an overflow or underflow in the **TCQEPCTL.TICKS** is detected, the **TCQEPCTL.WRIF** bit is set (counter wrap interrupt flag). If the **TCQEPCTL.WRIE** bit is set and the **TCQEPCTL.WRIF** bits are set, then the QEPC_IRQ signal is asserted to the NVIC.

If an index event is detected, the **TCQEPCTL.IDXEVI** bit is set. If the **TCQEPCTL.IDXEVI** bit is set and the **TCQEPCTL.IDXEVI** bit is set, then the QEPC_IRQ signal to the NVIC is asserted. The **TCQEPCTL.IDXEVI** bit can be cleared by writing a 1b to it.

¹² Before enabling the QEP peripheral, be sure to set up the Digital Peripheral MUX, so that no false edges are detected by the QEP state machine.

As the QEP decoder counts edges on PH_A and is turning clockwise (CW), it increments the counter value in **TCQEPCTL.TICKS**. If the motor is turning counter-clockwise (CCW), the **TCQEPCTL.TICKS** register is decremented. The user may write the **TCQEPCTL.TICKS** register at any time.

20.6 Peripheral IO Mapping

The Timer C peripheral signal inputs and outputs (CCR, QEP) are connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

Table 20-1 Timer C Peripheral IO Mapping

| TIMER SIGNAL | IO PIN |
|--------------|-------------------------|
| TCPWM0 | PC0, PD0, PE0, PF0, PG0 |
| TCPWM1 | PC1, PD1, PE1, PF1, PG1 |
| TCPWM2 | PC2, PD2, PE2, PF2, PG2 |
| TCPWM3 | PC3, PD3, PE3, PF3, PG3 |
| TCPWM4 | PC4, PD4, PE4, PF4, PG4 |
| TCPWM5 | PC5, PD5, PE5, PF5, PG5 |
| TCPWM6 | PC6, PD6, PE6, PF6, PG6 |
| TCPWM7 | PC7, PD7, PE7, PF7 |
| TCQEPIDX | PC4, PF4, PG0 |
| TCQEPHA | PC5, PF5, PG1 |
| TCQEPPHB | PC6, PF6, PG2 |

For more information on how to use the Digital Peripheral MUX to connect peripheral signals to IO, see Register Summary

20.7 Timer C Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------|------------|---------------------------|--------|------------|
| TCCTL | 4008 0000h | Timer C control | RW | 0000 0000h |
| TCINT | 4008 0004h | Timer C interrupt control | RW | 0000 0000h |
| TCPRD | 4008 0008h | Timer C period | RW | 0000 0000h |
| TCCTR | 4008 000Ch | Timer C counter | RW | 0000 0000h |
| TCQEPCTL | 4008 0010h | Timer C QEP control | RW | 0000 0000h |
| TCCCTL0 | 4008 0100h | Timer C CC control 0 | RW | 0000 0000h |
| TCCCTR0 | 4008 0104h | Timer C CC counter 0 | RW | 0000 0000h |
| TCCCTL1 | 4008 0108h | Timer C CC control 1 | RW | 0000 0000h |
| TCCCTR1 | 4008 010Ch | Timer C CC counter 1 | RW | 0000 0000h |
| TCCCTL2 | 4008 0110h | Timer C CC control 2 | RW | 0000 0000h |
| TCCCTR2 | 4008 0114h | Timer C CC counter 2 | RW | 0000 0000h |
| TCCCTL3 | 4008 0118h | Timer C CC control 3 | RW | 0000 0000h |
| TCCCTR3 | 4008 011Ch | Timer C CC counter 3 | RW | 0000 0000h |
| TCCCTL4 | 4008 0120h | Timer C CC control 4 | RW | 0000 0000h |
| TCCCTR4 | 4008 0124h | Timer C CC counter 4 | RW | 0000 0000h |
| TCCCTL5 | 4008 0128h | Timer C CC control 5 | RW | 0000 0000h |
| TCCCTR5 | 4008 012Ch | Timer C CC counter 5 | RW | 0000 0000h |
| TCCCTL6 | 4008 0130h | Timer C CC control 6 | RW | 0000 0000h |
| TCCCTR6 | 4008 0134h | Timer C CC counter 6 | RW | 0000 0000h |
| TCCCTL7 | 4008 0138h | Timer C CC control 7 | RW | 0000 0000h |
| TCCCTR7 | 4008 013Ch | Timer C CC counter 7 | RW | 0000 0000h |
| TCDTGCTL0 | 4008 0200h | Timer C DTG control 0 | RW | 0000 0000h |
| TCDTGCTL1 | 4008 0204h | Timer C DTG control 1 | RW | 0000 0000h |
| TCDTGCTL2 | 4008 0208h | Timer C DTG control 2 | RW | 0000 0000h |
| TCDTGCTL3 | 4008 020Ch | Timer C DTG control 3 | RW | 0000 0000h |

20.8 Register Detail

20.8.1 TCCTL

Register 20-1 TCCTL (Timer C Control, 4008 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | BASEIE | RW | 0 | Base timer interrupt enable: 0b: not enabled 1b: enabled |
| 12 | CLR | RW | 0 | Base timer clear: 0b: do not clear 1b: clear timer counter |
| 11 | LATCH | RW | 0 | When written to 1b, this will latch the TCPRD and all TCCTRn registers on the same clock cycle. Writing this bit to 0b has no effect. This is a self-clearing bit. |
| 10 | DTGCLK | RW | 0 | DTG Clock Source: 0b: PCLK 1b: ACLK |
| 9 | CLKSRC | RW | 0 | Timer Clock Source: 0b: PCLK 1b: ACLK |
| 8:6 | CLKDIV | RW | 0 | Timer Input Clock Divider: 000b: /1 001b: /2 010b: /4 011b: /8 100b: /16 101b: /32 110b: /64 111b: /128 |
| 5 | SINGLE | RW | 0 | Single Shot Timer: 0b: disabled (auto-reload) 1b: enabled (single shot timer) |
| 4 | SSYNC | RW | 0 | Timer Slave Synchronization: 0b: disabled 1b: enabled |
| 3:2 | PRDLATCH | RW | 0 | Timer Period Latch Mode: 00b: Latch TCPRD when TCCTR = 0 01b: Latch TCPRD when TCCTR = TCPRD 10b: Latch TCPRD immediately upon register write 11b: reserved |
| 1:0 | MODE | RW | 0 | Timer Mode: |

| | | | | |
|--|--|--|--|--|
| | | | | 00b: disabled 01b: up mode 10b: up/down mode 11b: asymmetric mode |
|--|--|--|--|--|

20.8.2 TCINT

Register 20-2 TCINT (Timer C Interrupt Control, 4008 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:9 | Reserved | RO | 0 | Reserved |
| 8 | BASEIF | W1C | 0 | Base timer interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 7 | CCR7IF | W1C | 0 | CCR7 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 6 | CCR6IF | W1C | 0 | CCR6 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 5 | CCR5IF | W1C | 0 | CCR5 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 4 | CCR4IF | W1C | 0 | CCR4 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 3 | CCR3IF | W1C | 0 | CCR3 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 2 | CCR2IF | W1C | 0 | CCR2 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 1 | CCR1IF | W1C | 0 | CCR1 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 0 | CCR0IF | W1C | 0 | CCR0 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |

20.8.3 TCPRD

Register 20-3 TCPRD (Timer C Period, 4008 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--------------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | PERIOD | RW | 0 | Timer period value |

20.8.4 TCCTR

Register 20-4 TCCTR (Timer C Counter, 4008 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | COUNTER | RW | 0 | Timer counter |

20.8.5 TCQEPCTL

Register 20-5 TCQEPCTL (Timer C QEP Control, 4008 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | TICKS | RW | 0 | Number of counted phase edges according to counting mode defined by TCQEPCTL.CNTEEDGE and TCQEPCTL.CNTAB . |
| 15 | Reserved | RO | 0 | Reserved |
| 14 | CNTEEDGE | RW | 0 | Count on edge: 0b: Rising edge only 1b: Rising and falling edge |
| 13 | CNTAB | RW | 0 | Count on A/B: 0b: Count phase A only 1b: Count phase A and phase B |
| 12 | IDXEIVF | W1C | 0 | Index event interrupt flag: 0b: no index event detected 1b: index event detected |
| 11 | WRIF | W1C | 0 | Counter wrap interrupt flag: 0b: no counter wrap 1b: counter wrap |
| 10 | PHBIF | W1C | 0 | Phase B rising edge interrupt flag: 0b: no phase B rising edge detected 1b: phase B rising edge detected |
| 9 | PHAIF | W1C | 0 | Phase A rising edge interrupt flag: 0b: no phase A rising edge detected 1b: phase A rising edge detected |
| 8 | DIRIF | W1C | 0 | Direction change interrupt flag: 0b: no direction change detected 1b: direction change detected |
| 7 | IDXRST | RW | 0 | Reset counter on index: 0b: Do not reset TCQEPCTL.TICKS 1b: Reset TCQEPCTL.TICKS on all index events |
| 6 | IDXEIVIE | RW | 0 | Index event interrupt enable: 0b: disabled 1b: enabled |
| 5 | WRIE | RW | 0 | Counter wrap interrupt enable: 0b: disabled 1b: enabled |
| 4 | PHBIE | RW | 0 | Phase B rising edge interrupt enable: 0b: disabled 1b: enabled |
| 3 | PHAIE | RW | 0 | Phase A rising edge interrupt enable: |

| | | | | |
|---|--------------|----|---|---|
| | | | | 0b: disabled 1b: enabled |
| 2 | DIRIE | RW | 0 | Direction change interrupt enable: 0b: disabled 1b: enabled |
| 1 | DIR | RO | 0 | Motor direction: 0b: CW (clockwise) 1b: CCW (counter-clockwise) |
| 0 | QEPEN | RW | 0 | QEP peripheral enabled: 0b: disabled 1b: enabled |

20.8.6 TCCCTL0

Register 20-6 TCCCTL0 (Timer C CCR Control 0, 4008 0100h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL0.CCMODE = 0b (compare): 00b: Latch TCCCTRL0 registers when TCCTR = 0 01b: Latch TCCCTRL0 registers when TCCTR = TCPRD 10b: Latch TCCCTRL0 registers immediately 11b: Reserved If TCCCTL0.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR0.CTR on rising edge 01b: Latch TCCTR into TCCCTR0.CTR on falling edge 10b: Latch TCCTR into TCCCTR0.CTR on both edges 11b: Latch TCCTR into TCCCTR0.CAP1 and TCCCTR0.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL0.CCMODE = 1b (capture), set TCINT.CCR0IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.7 TCCCTR0

Register 20-7 TCCCTR0 (Timer C CCR Counter 0, 4008 0104h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTL0.CCLATCH != 11b (period/duty mode) and TCCCTL0.CCMODE = 1b (capture mode), this is the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTL0.CCLATCH = 11b (period/duty mode) and TCCCTL0.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTL0.CCLATCH != 11b (period/duty mode) and TCCCTR0.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTL0.CCLATCH != 11b (period/duty mode) and TCCCTL0.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTL0.CCLATCH = 11b (period/duty mode) and TCCCTL0.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.8 TCCCTL1

Register 20-8 TCCCTL1 (Timer C CCR Control 1, 4008 0108h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL1.CCMODE = 0b (compare): 00b: Latch TCCCTL1 registers when TCCTR = 0 01b: Latch TCCCTL1 registers when TCCTR = TCPRD 10b: Latch TCCCTL1 registers immediately 11b: Reserved If TCCCTL1.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR1.CTR on rising edge 01b: Latch TCCTR into TCCCTR1.CTR on falling edge 10b: Latch TCCTR into TCCCTR1.CTR on both edges 11b: Latch TCCTR into TCCCTR1.CAP1 and TCCCTR1.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL1.CCMODE = 1b (capture), set TCINT.CCR1IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.9 TCCCTR1

Register 20-9 TCCCTR1 (Timer C CCR Counter 1, 4008 010Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTL1.CCLATCH != 11b (period/duty mode) and TCCCTL1.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTL1.CCLATCH = 11b (period/duty mode) and TCCCTL1.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTL1.CCLATCH != 11b (period/duty mode) and TCCCTL1.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTL1.CCLATCH != 11b (period/duty mode) and TCCCTL1.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTL1.CCLATCH = 11b (period/duty mode) and TCCCTL1.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.10 TCCCTL2

Register 20-10 TCCCTL2 (Timer C CCR Control 2, 4008 0110h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL2.CCMODE = 0b (compare): 00b: Latch TCCCTL2 registers when TCCTR = 0 01b: Latch TCCCTL2 registers when TCCTR = TCPRD 10b: Latch TCCCTL2 registers immediately 11b: Reserved If TCCCTL2.CCMODE = 1b (capture): 00b: Latch TCBCTR into TCCCTR2.CTR on rising edge 01b: Latch TCCTR into TCCCTR2.CTR on falling edge 10b: Latch TCCTR into TCCCTR2.CTR on both edges 11b: Latch TCCTR into TCCCTR2.CAP1 and TCCCTR2.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL2.CCMODE = 1b (capture), set TCINT.CCR2IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.11 TCCCTR2

Register 20-11 TCCCTR2 (Timer C CCR Counter 2, 4008 0114h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL2.CCLATCH != 11b (period/duty mode) and TCCCTRL2.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL2.CCLATCH = 11b (period/duty mode) and TCCCTRL2.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL2.CCLATCH != 11b (period/duty mode) and TCCCTRL2.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL2.CCLATCH != 11b (period/duty mode) and TCCCTRL2.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL2.CCLATCH = 11b (period/duty mode) and TCCCTRL2.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.12 TCCCTL3

Register 20-12 TCCCTL3 (Timer C CCR Control 3, 4008 0118h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL3.CCMODE = 0b (compare): 00b: Latch TCCCTL3 registers when TCCTR = 0 01b: Latch TCCCTL3 registers when TCCTR = TCPRD 10b: Latch TCCCTL3 registers immediately 11b: Reserved If TCCCTL3.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR3.CTR on rising edge 01b: Latch TCCTR into TCCCTR3.CTR on falling edge 10b: Latch TCCTR into TCCCTR3.CTR on both edges 11b: Latch TCCTR into TCCCTR3.CAP1 and TCCCTR3.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL3.CCMODE = 1b (capture), set TCINT.CCR3IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.13 TCCCTR3

Register 20-13 TCCCTR3 (Timer C CCR Counter 3, 4008 011Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL3.CCLATCH != 11b (period/duty mode) and TCCCTRL3.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL3.CCLATCH = 11b (period/duty mode) and TCCCTRL3.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL3.CCLATCH != 11b (period/duty mode) and TCCCTRL3.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL3.CCLATCH != 11b (period/duty mode) and TCCCTRL3.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL3.CCLATCH = 11b (period/duty mode) and TCCCTRL3.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.14 TCCCTL4

Register 20-14 TCCCTL4 (Timer C CCR Control 4, 4008 0120h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL4.CCMODE = 0b (compare): 00b: Latch TCCCTL4 registers when TCCTR = 0 01b: Latch TCCCTL4 registers when TCCTR = TCPRD 10b: Latch TCCCTL4 registers immediately 11b: Reserved If TCCCTL4.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR4.CTR on rising edge 01b: Latch TCCTR into TCCCTR4.CTR on falling edge 10b: Latch TCCTR into TCCCTR4.CTR on both edges 11b: Latch TCCTR into TCCCTR4.CAP1 and TCCCTR4.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL4.CCMODE = 1b (capture), set TCINT.CCR4IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.15 TCCCTR4

Register 20-15 TCCCTR4 (Timer C CCR Counter 4, 4008 0124h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL4.CCLATCH != 11b (period/duty mode) and TCCCTRL4.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL4.CCLATCH = 11b (period/duty mode) and TCCCTRL4.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL4.CCLATCH != 11b (period/duty mode) and TCCCTRL4.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL4.CCLATCH != 11b (period/duty mode) and TCCCTRL4.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL4.CCLATCH = 11b (period/duty mode) and TCCCTRL4.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.16 TCCCTL5

Register 20-16 TCCCTL5 (Timer C CCR Control 5, 4008 0128h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL5.CCMODE = 0b (compare): 00b: Latch TCCCTL5 registers when TCCTR = 0 01b: Latch TCCCTL5 registers when TCCTR = TCPRD 10b: Latch TCCCTL5 registers immediately 11b: Reserved If TCCCTL5.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR5.CTR on rising edge 01b: Latch TCCTR into TCCCTR5.CTR on falling edge 10b: Latch TCCTR into TCCCTR5.CTR on both edges 11b: Latch TCCTR into TCCCTR5.CAP1 and TCCCTR5.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL5.CCMODE = 1b (capture), set TCINT.CCR5IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.17 TCCCTR5

Register 20-17 TCCCTR5 (Timer C CCR Counter 5, 4008 012Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL5.CCLATCH != 11b (period/duty mode) and TCCCTRL5.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL5.CCLATCH = 11b (period/duty mode) and TCCCTRL5.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL5.CCLATCH != 11b (period/duty mode) and TCCCTRL5.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL5.CCLATCH != 11b (period/duty mode) and TCCCTRL5.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL5.CCLATCH = 11b (period/duty mode) and TCCCTRL5.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.18 TCCCTL6

Register 20-18 TCCCTL6 (Timer C CCR Control 6, 4008 0130h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL6.CCMODE = 0b (compare): 00b: Latch TCCCTL6 registers when TCCTR = 0 01b: Latch TCCCTL6 registers when TCCTR = TCPRD 10b: Latch TCCCTL6 registers immediately 11b: Reserved If TCCCTL6.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR6.CTR on rising edge 01b: Latch TCCTR into TCCCTR6.CTR on falling edge 10b: Latch TCCTR into TCCCTR6.CTR on both edges 11b: Latch TCCTR into TCCCTR6.CAP1 and TCCCTR6.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL6.CCMODE = 1b (capture), set TCINT.CCR6IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.19 TCCCTR6

Register 20-19 TCCCTR6 (Timer C CCR Counter 6, 4008 0134h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL6.CCLATCH != 11b (period/duty mode) and TCCCTRL6.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL6.CCLATCH = 11b (period/duty mode) and TCCCTRL6.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL6.CCLATCH != 11b (period/duty mode) and TCCCTRL6.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL6.CCLATCH != 11b (period/duty mode) and TCCCTRL6.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL6.CCLATCH = 11b (period/duty mode) and TCCCTRL6.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.20 TCCCTL7

Register 20-20 TCCCTL7 (Timer C CCR Control 7, 4008 0138h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|------------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TCCCTL7.CCMODE = 0b (compare): 00b: Latch TCCCTL7 registers when TCCTR = 0 01b: Latch TCCCTL7 registers when TCCTR = TCPRD 10b: Latch TCCCTL7 registers immediately 11b: Reserved If TCCCTL7.CCMODE = 1b (capture): 00b: Latch TCCTR into TCCCTR7.CTR on rising edge 01b: Latch TCCTR into TCCCTR7.CTR on falling edge 10b: Latch TCCTR into TCCCTR7.CTR on both edges 11b: Latch TCCTR into TCCCTR7.CAP1 and TCCCTR7.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TCCCTL7.CCMODE = 1b (capture), set TCINT.CCR7IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

20.8.21 TCCCTR7

Register 20-21 TCCCTR7 (Timer C CCR Counter 7, 4008 013Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TCCCTRL7.CCLATCH != 11b (period/duty mode) and TCCCTRL7.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TCCCTRL7.CCLATCH = 11b (period/duty mode) and TCCCTRL7.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TCCCTRL7.CCLATCH != 11b (period/duty mode) and TCCCTRL7.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TCCCTRL7.CCLATCH != 11b (period/duty mode) and TCCCTRL7.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TCCCTRL7.CCLATCH = 11b (period/duty mode) and TCCCTRL7.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

20.8.22 TCDTGCTL0

Register 20-22 TCDTGCTL0 (Timer C Dead-Time Generator Control 0, 4008 0200h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG0. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG0. |

20.8.23 TCDTGCTL1

Register 20-23 TCDTGCTL1 (Timer C Dead-Time Generator Control 1, 4008 0204h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG1. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG1. |

20.8.24 TCDTGCTL2

Register 20-24 TCDTGCTL2 (Timer C Dead-Time Generator Control 2, 4008 0208h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG2. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG2. |

20.8.25 TCDTGCTL3

Register 20-25 TCDTGCTL3 (Timer C Dead-Time Generator Control 3, 4008 020Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG3. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG3. |

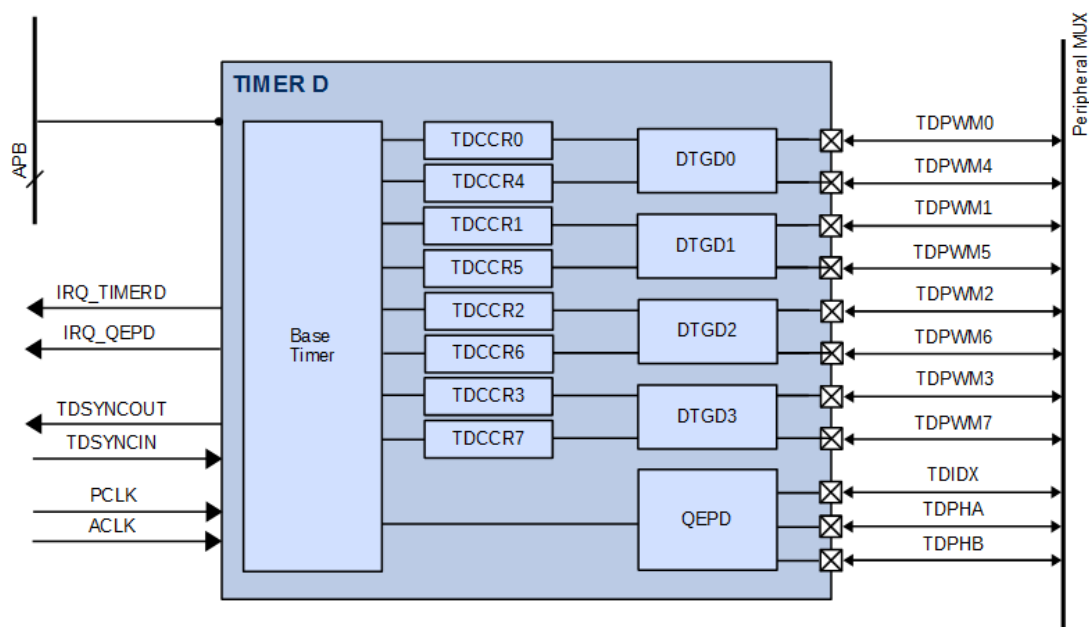
21 PWM TIMER D

21.1 Overview

All devices in the PAC55XX family of controllers have a Timer D peripheral. This peripheral is a 16-bit timer that allows support for 8 Capture and Compare Units (CCR) capable of PWM generation; capture input processing and a QEP encoder for various control applications.

Below is a simplified block diagram of the Timer peripheral.

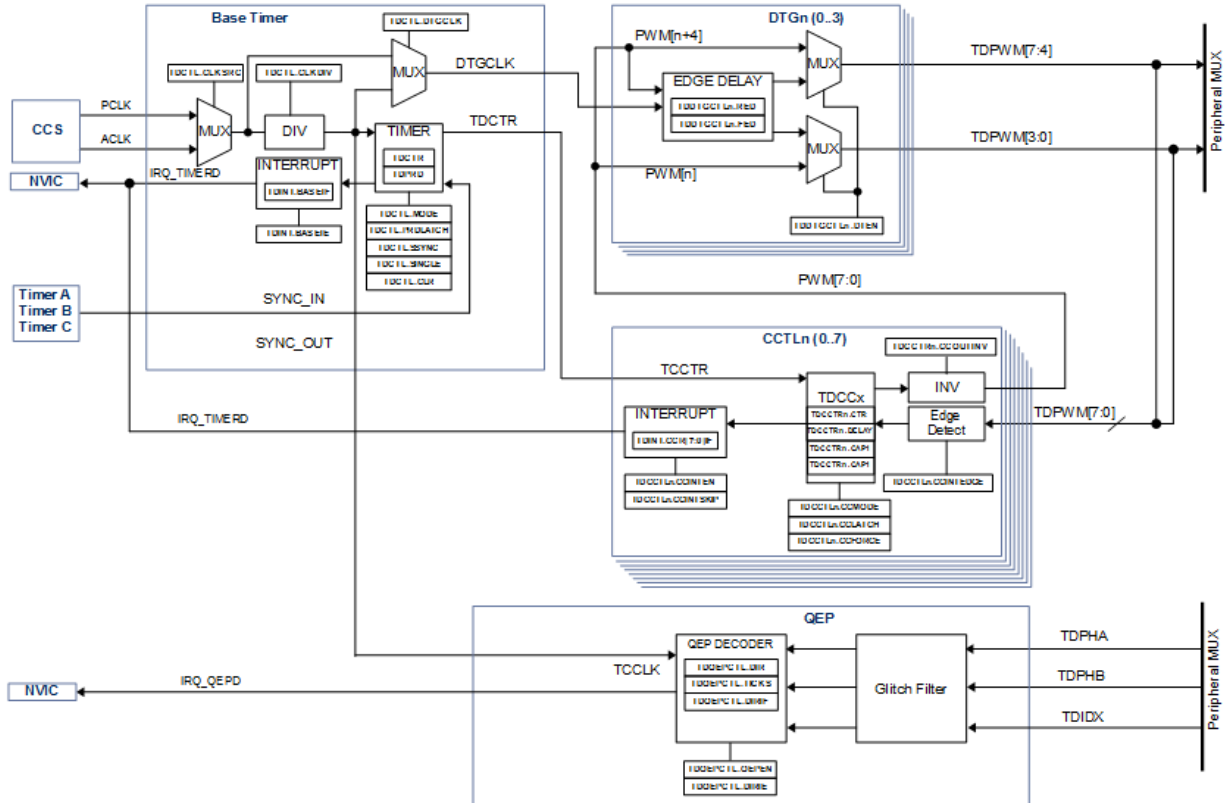
Figure 21-1 Timer D Simplified Block Diagram



The timer module is an APB bus client. There are 8 timer channel input/outputs to the digital peripheral MUX, in addition to three IOs for QEP. Each timer block has two interrupt signal outputs to the NVIC; one for timer functions and one for QEP functions. The user may select either PCLK or ACLK as the clock input for each timer module.

21.2 Timer D Block Diagram

Figure 21-2 Timer D Block Diagram



21.3 Features

The PWM Timer peripheral has the following features in the PAC55XX family of controllers.

General Features:

- Configurable input clock: PCLK or ACLK
- Up to 300MHz clock input for 3.33ns PWM edge resolution
- 3-bit input clock divider
- Latch timer period and all CCR values on command

Base Timer Features:

- Single-shot or auto-reload
- Base timer interrupts
- Timer synchronization
- Timer Modes:
 - Disabled
 - Up mode
 - Up/Down mode
 - Asymmetric Up mode
- Timer Register Latching Options
 - Latch TAPRD when counter = 0
 - Latch TAPRD when counter = period
 - Latch TAPRD immediately
- Dead-time Generator Input Clock
 - DTG clock = PCLK
 - DTG clock = ACLK

CCR/PWM Features:

- PWM output or input capture
- 8 CCR units per timer
- CCR interrupts
- CCR interrupt skips
- SW force CCR interrupt
- CCR interrupt type:
 - Rising, falling, both
- CCR latch modes:
 - Compare/PWM mode: counter = 0, counter = period, immediate
 - Capture input: rising edge, falling edge, both
- Period/duty calculation mode
- Force compare event

- Invert CCR output
- CCR phase delay for phase shifted drive topologies
- ADC trigger outputs:
 - PWM rising or falling edge

Dead-Time Generator (DTG) Features:

- DTG enable or bypass
- 12-bit rising edge delay
- 12-bit falling edge delay

QEP Decoder Features:

- QEP encoder enabled
- Direction status
- Configurable interrupts:
 - Phase A rising edge
 - Phase B rising edge
 - Index event
 - Counter wrap
- 4 Different counting modes for best resolution, range and speed performance

21.4 Functional Description

21.4.1 Timer Clock Structure

The timer peripheral input clock can be selected as either PCLK or ACLK. The input clock may be selected by the **TDCTL.CLKSRC** register.

Each timer peripheral has a 4-bit divider that can be used to divide the selected input clock. The user may set this divider by the **TDCTL.CLKDIV** register. There are 16 settings between /1 and /128 for the input clock divider.

The base timer also supplies the clock for the Dead-Time Generators (DTGs) to allow for different range and resolution for dead-time. The timer may select the DTGCLK to be the timer clock before or after the clock divider. The DTG input clock may be selected by the **TDCTL.DTGCLK** register.

To use the timer clock before the input clock divider, set **TDCTL.DTGCLK** to 0b. To use the timer clock after the input clock divider, set **TDCTL.DTGCLK** to 1b.

21.4.2 Timer Counter

The base timer is a 16-bit timer that can count either up, or up then down to support both edge and center aligned and asymmetric PWM output types.

The timer period is stored in the **TDPRD** register. The current value of the timer counter is updated at every timer tick and is stored in the **TDCTR** register. The **TDCTR** register is a RW register, so it may be updated at any time and is changed immediately.

The **TDPRD** register and all of the **TDCCTRn** registers have shadow copies that are updated at a user-specified time. See the section below on Base Timer **TDPRD** Latching and CCR Timer Latching for more information.

When **TDCTL.MODE** = 00b, the timer is disabled. Even when disabled, all timer registers are accessible via the APB bus.

To enable the timer, set the **TDCTL.MODE** to 01b (up mode), 10b (up/down mode) or 11b (up/down asymmetric mode).

When **TDCTL.MODE** = 01b, the timer is configured in up mode. The timer will count from 0 to **TDPRD**. The counter is updated in **TDCTR** at every timer tick. If auto-reload is active (**TDCTL.SINGLE** = 0b), then the timer count will automatically count from **TDPRD** to 0 and continue counting up. If auto-reload is not active (**TDCTL.SINGLE** = 1b), then the timer will count from 0 to **TDPRD** and stop. When this happens, the timer will set the **TDCTL.MODE** to 00b (disabled).

When the timer is configured for up/down mode (**TDCTL.MODE** = 10b) or configured for up/down asymmetric mode (**TDCTL.MODE** = 11b), the timer will count from 0 to **TDPRD** and then back down to 0. If auto-reload is not active (**TDCTL.SINGLE** = 1b), the time timer will stop

and set **TDCTL.MODE** to 00b (disabled). If auto-reload is active (**TDCTL.SINGLE** = 0b), then the timer count will then continue counting back up after it reaches 0.

21.4.3 Up Mode

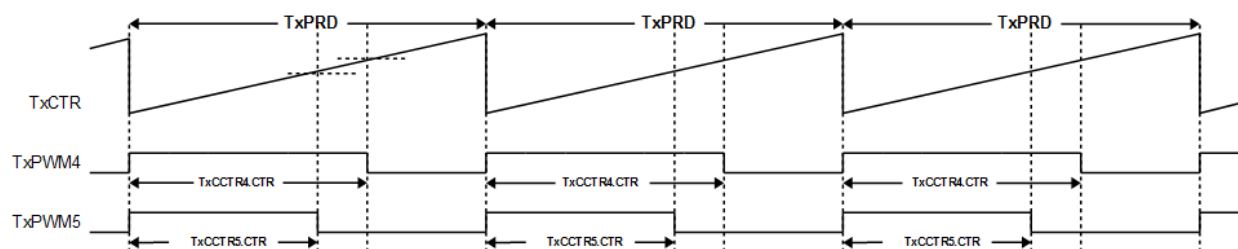
When the **TDCTL.MODE** is set to 01b (up mode) the timer will begin counting up from 0. The **TDCTR** will count to the value of **TDPRD** and then will reset to 0.

If the timer is configured for auto-reload mode (**TDCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it reaches 0.

The **TDCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate an edge aligned PWM as shown below.

Figure 21-3 Up Mode PWM Waveform



In this mode, when **TDCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TDCTR** register reaches the **TDCCTRn.CTR** threshold. When the **TDCTR** counts from **TDPRD** to 0, the PWM output is transitioned to high.

21.4.4 Up/Down Mode

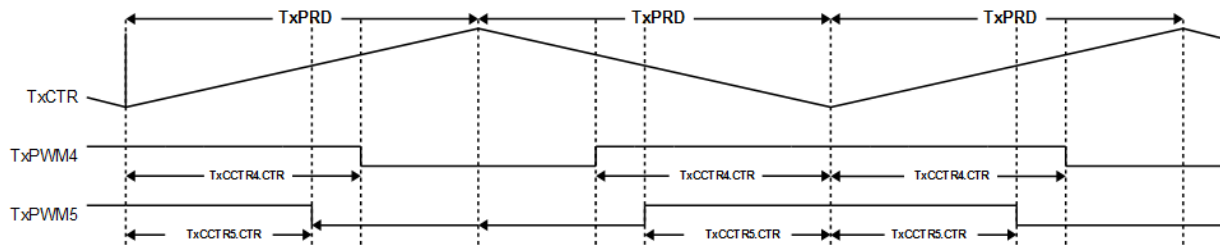
When the **TDCTL.MODE** is set to 10b (up/down mode) the timer will begin counting up from 0. The **TDCTR** will count to the value of **TDPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TDCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TDCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned PWM as shown below.

Figure 21-4 Up/Down Mode PWM Waveform



In this mode, when **TDCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TDCTR** register reaches the **TDCCTRn.CTR** threshold. When the **TDCTR** counts from **TDPRD - 1** to **TDPRD**, it will begin counting down. When the **TDCTR** counts down to **TDCCTRn.CTR** the PWM output is transitioned to high.

21.4.5 Up/Down Asymmetric mode

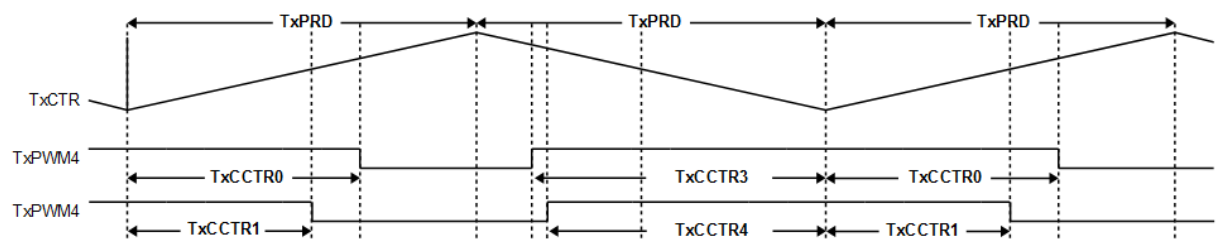
When the **TDCTL.MODE** is set to 11b (up/down asymmetric mode) the timer will begin counting up from 0. The **TDCTR** will count to the value of **TDPRD** and then will begin counting back down to 0.

If the timer is configured for auto-reload mode (**TDCTL.SINGLE** = 0b), then the timer will continue counting up on the next timer clock cycle. If the timer is not configured for auto-reload, then the timer will stop when it counts from 1 to 0.

The **TDCTR** register may be read or written at any time. This feature may be used to “pre-load” the timer counter with a user-specified value before it is enabled and begins counting.

This timer mode can generate a center aligned asymmetric PWM as shown below.

Figure 21-5 Up/Down Asymmetric Mode PWM Waveform



In this mode, when **TDCTR** starts counting from 0, the PWM output is high. The PWM output is transitioned to low then the **TDCTR** register reaches the **TDCCTRn.CTR** threshold. When the **TDCTR** counts from **TDPRD - 1** to **TDPRD**, it will begin counting down. When the **TDCTR** counts down to **TDCCTRn.CTR** the PWM output is transitioned to high.

In this mode, the CCR outputs are generated such that 2 **TDCCTRn.CTR** registers are used to generate a single PWM output, so that the center-aligned PWM can be generated with asymmetric on-time.

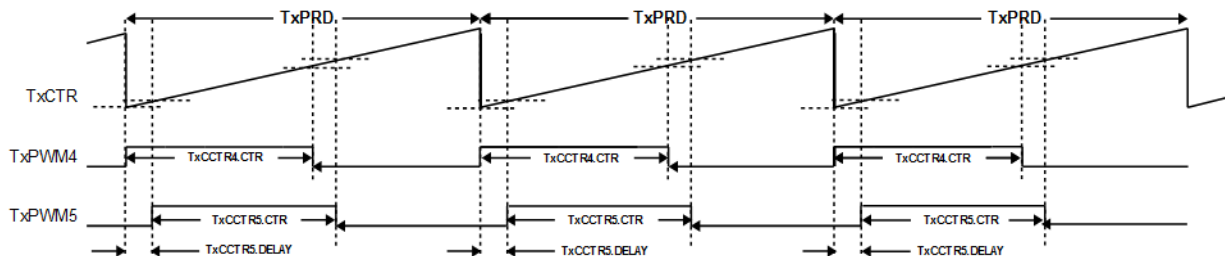
In this mode, the PWM outputs of the timer are generated as follows:

- TDPWM4 uses **TDCCTR0.CTR** for the first portion of the on time and **TDCCTR4.CTR** for the second portion of the on time
- TDPWM5 uses **TDCCTR1.CTR** for the first portion of the on time and **TDCCTR5.CTR** for the second portion of the on time
- TDPWM6 uses **TDCCTR2.CTR** for the first portion of the on time and **TDCCTR6.CTR** for the second portion of the on time
- TDPWM7 uses **TDCCTR3.CTR** for the first portion of the on time and **TDCCTR7.CTR** for the second portion of the on time

21.4.6 Up Mode with Phase Delay

When the timer is configured for up mode (**TDMODE** = 01b) the CCR allows a delay to be applied to the output PWM period, to support phase delay control topologies. If the value of **TDCCTLn.DELAY** > 0, then the CCR will delay this number of ticks before starting the PWM period (transitioning the output signal from high to low).

Figure 21-6 Up Mode with Phase Delay



21.4.7 Timer Synchronization

The timer peripheral allows synchronization between the timers in the PAC55XX. The timers may be synchronized as follows:

- Timer A (master), Timer B (slave)
- Timer A (master), Timer B (slave), Timer C (slave)
- Timer A (master), Timer B (slave), Timer C (slave), Timer D (slave)

The timer master's SYNC_OUT signals are connected to the downstream timer slave's SYNC_IN signal.

When configured, the timers can synchronize their clocks so they are sharing the same time base. In order to do this, the timers must be configured with the same clock source (**TDCTL.CLKSRC**) and clock divider (**TDCTL.CLKDIV**).

To configure the timers for synchronization, follow these steps:

- While the timer master is disabled (**TDCTL.MODE** = 00b), set the timer master **TxCTL.SSYNC** = 0b
- For each of the downstream timer slaves, while the timer is disabled **TDCTL.MODE** = 00b, set the **TDCTL.SSYNC** to 1b. Make sure that **TDCTL.CLKDIV** and **TDCTL.CLKSRC** for each of the timer slaves is the same as the timer master.
- Enable each of the slave timers by setting **TDCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric). The slave timers will not start counting since the **TDCTL.SSYNC** is set to a 1b.
- Enable the timer master by setting **TxCTL.MODE** to 01b (up), 10b (up/down) or 11b (up/down asymmetric).

At this point, the master and all configured timer slaves will begin counting.

21.4.8 Base Timer TAPRD Latching

The timer peripherals have a shadow copy of the **TDPRD** register that is used for the counting operations. The latching of the data from the **TDPRD** register to the shadow register is controlled by the timer configuration.

There are several options for latching of this register into the shadow copy. This behavior is controlled by the **TDCTL.PRDLATCH** setting and depends on the setting of the **TDCTL.MODE** register as well.

21.4.8.1 TDPRD Latch When TCCTR = 0

- If **TDCTL.MODE** = 01b (up mode) and **TDCTL.PRDLATCH** = 00b (**TDCTR** = 0), then the **TDPRD** register is copied into the shadow register when **TDCTR** counts from **TCPRD** to 0.
- If **TDCTL.MODE** = 10b (up/down mode) or if **TCCTL.MODE** = 11b (up/down asymmetric mode) and **TCCTL.PRDLATCH** = 00b (**TCCTR** = 0), then the **TCPRD** register is copied into the shadow register when the **TCCTR** counts from 1 to 0.

21.4.8.2 TDPRD Latch When TCCTR = TCPRD

- If **TDCTL.MODE** = 01b (up mode) or if **TDCTL.MODE** = 10b (up/down mode) or if **TDCTL.MODE** = 11b (up/down asymmetric mode) and **TDCTL.PRDLATCH** = 01b (**TDCTR** = **TDPRD**), then the **TDPRD** register is copied into the shadow register when **TDCTR** counts from **TDPRD** – 1 to **TDPRD**.

21.4.8.3 TDPRD Latch Immediate

- If **TDCTL.PRDLATCH** = 11b (immediate), then the **TDPRD** register is copied into the shadow register as soon as the **TDPRD** register is written.

21.4.9 CCR Timer Latching

When in compare mode, the CCR units support configurable latching of the **TDCCRn.CTR** values into a shadow register that is used for timer operation. The shadow register is used for generating the PWM output when the CCR is in compare mode.

This feature can be used to control when the new duty cycle is applied, during the PWM period.

When the CCR is in compare mode (**TDCCTLn.CCMODE** = 0b), the shadow register is latched as follows:

- If **TDCCTLn.CCLATCH** = 00b (**TDCTR** = 0)
 - If **TDCTL.MODE** = 01b (up mode), **TDCCTRn.CTR** is copied into the shadow register when **TDCTR** counts from **TDPRD** to 0. If **TDCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode), **TDCCTRn.CTR** is copied into the shadow register when **TDCTR** counts from 1 to 0.
- If **TDCCTLn.CCLATCH** = 01b (**TDCTR** = **TCPRD**), then the **TDCCTRn.CTR** is copied into the shadow register when **TDCTR** counts from **TDPRD** – 1 to **TDPRD**.
- If **TDCCTLn.CCLATCH** = 10b (latch immediate), then the **TDCCTRn.CTR** is copied into the shadow register as soon as it is written.

When the CCR is in capture mode (**TDCCTLn.CCMODE** = 1b), the configuration of **TDCCTLn.CCLATCH** controls when the **TDCTR** is copied into the **TDCCTRn.CTR** register.

- If **TDCCTLn.CCLATCH** = 00b (rising edge), then the value of **TDCTR** is copied into the **TDCCTL.CTR** register upon a rising edge in the input signal.
- If **TDCCTLn.CCLATCH** = 01b (falling edge), then the value of the **TDCTR** is copied into the **TDCCTLn.CTR** register upon a falling edge in the input signal.
- If **TDCCTLn.CCLATCH** = 10b (both), then the value of the **TDCTR** is copied into the **TDCCTLn.CTR** register on both a rising and falling edge of the input signal.
- If **TDCCTLn.CCLATCH** = 11b (period/duty mode), then see the section below on how the values are latched into the **TDCCTRn** register.

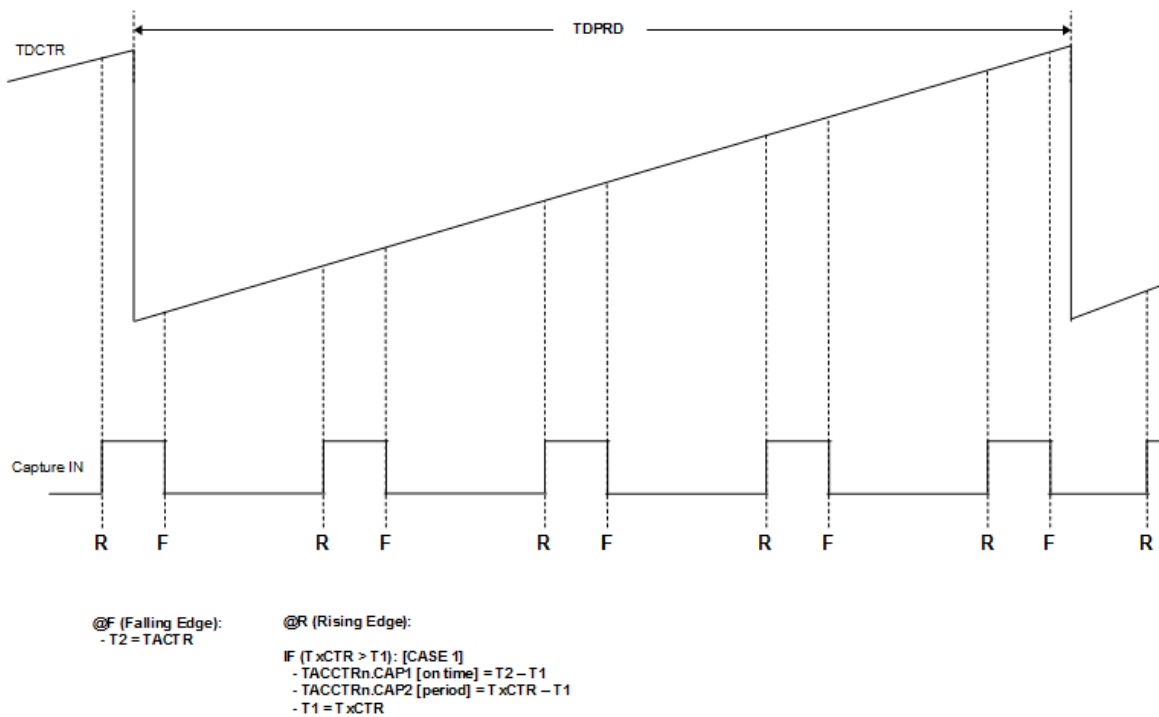
21.4.10 CCR Capture Period/Duty Mode

If the **TDCCTLn.CCLATCH = 11b** (period/duty mode), then the CCR can capture the number of timer ticks for the input PWM duty cycle and period. This mode is useful for measuring input pulse widths for PPM and other type of control applications.

At each rising edge of the input signal, the following happens:

- The number of timer ticks for the last timer period's duty cycle is copied into **TDCCTRn.CAP1**
- The number of timer ticks for the last timer period is copied into **TDCCTRn.CAP2**

Figure 21-7 Period/Duty Capture Mode PWM Waveform



21.4.11 Timer Whole Latching

It is sometimes convenient to latch both the **TDPRD** (period) and all **TDCCTRn.CTR** (duty cycle) registers into the shadow registers at one time.

If the **TDCTL.LATCH** bit is written to a 1b, then all the **TDPRD** and all **TDCCTRn.CTR** registers will be copied into the shadow registers on the same clock cycle.

The **TDCTL.LATCH** bit is self-clearing, and will always be read as a 0b.

21.4.12 Inverting CCR PWM Output

When the CCR is in compare mode (**TDCCTRn.CCMODE** = 0b), the output of the CCR (the input to the DTG) may be inverted. To invert this output, set the **TDCCTRn.CCOUTINV** = 1b.

This mode is useful for some control topologies, especially full-bridge.

21.4.13 Base Timer Interrupts

The base timer may be configured to generate an interrupt to the NVIC.

If **TDCTL.MODE** = 01b (up mode) and **TDCTR** counts from **TDPRD** to 0 then the **TCINT.BASEIF** bit is set to 1b. If the **TDCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERD_IRQ** signal to the NVIC is asserted.

If **TDCTL.MODE** = 10b (up/down mode) or 11b (up/down asymmetric mode) and **TDCTR** counts from 1 to 0, then the **TDINT.BASEIF** bit is set to 1b. If the **TDCTL.BASEIE** interrupt enable bit is set to a 1b, then the **TIMERD_IRQ** signal to the NVIC is asserted.

The **TDINT.BASEIF** may be cleared by writing a 1b to it.

21.4.14 CCR Compare Interrupts

When **TDCCTLn.CCMODE** = 0b (compare mode), the CCR unit may be configured to generate an interrupt to the NVIC.

When **TDCTR** counts to **TDCCTRn.CTR**, then the CCR interrupt flag in the **TDINT** register is set as follows:

- For CCR0, when **TDCTR** counts to **TDCCTR0.CTR**, then **TDINT.CCR0IF** is set to a 1b. If the **TDCCTR0.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TCINT.CCR0IF** may be cleared by writing it to a 1b.
- For CCR1, when **TDCTR** counts to **TDCCTR1.CTR**, then **TDINT.CCR1IF** is set to a 1b. If the **TDCCTR1.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TDINT.CCR1IF** may be cleared by writing it to a 1b.
- For CCR2, when **TDCTR** counts to **TDCCTR2.CTR**, then **TDINT.CCR2IF** is set to a 1b. If the **TDCCTR2.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TDINT.CCR2IF** may be cleared by writing it to a 1b.

- For CCR3, when **TDCTR** counts to **TDCCTR3.CTR**, then **TDINT.CCR3IF** is set to a 1b. If the **TDCCTR3.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TCINT.CCR3IF** may be cleared by writing it to a 1b.
- For CCR4, when **TDCTR** counts to **TDCCTR4.CTR**, then **TDINT.CCR4IF** is set to a 1b. If the **TCCCTR4.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TCINT.CCR4IF** may be cleared by writing it to a 1b.
- For CCR5, when **TDCTR** counts to **TDCCTR5.CTR**, then **TDINT.CCR5IF** is set to a 1b. If the **TDCCTR5.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TDINT.CCR5IF** may be cleared by writing it to a 1b.
- For CCR6, when **TDCTR** counts to **TDCCTR6.CTR**, then **TDINT.CCR6IF** is set to a 1b. If the **TCCCTR6.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TCINT.CCR6IF** may be cleared by writing it to a 1b.
- For CCR7, when **TDCTR** counts to **TDCCTR7.CTR**, then **TDINT.CCR7IF** is set to a 1b. If the **TDCCTR7.CCINTEN** is set to 1b, then **TIMERD_IRQ** to the NVIC is asserted. **TDINT.CCR7IF** may be cleared by writing it to a 1b.

21.4.15 CCR Capture Interrupts

When **TDCCTLn.CCMODE** = 1b (capture mode), the CCR unit may be configured to generate an interrupt to the NVIC based on the input signal.

- If **TDCCTLn.CCEGE** = 00b (rising edge) and a rising edge is detected in the input signal, then the **TDINT.CCRnIF** interrupt flag is set.
- If **TDCCTLn.CCEDGE** = 01b (falling edge) and a falling edge is detected in the input signal, then the **TDINT.CCRnIF** interrupt flag is set.
- If the **TDCCTLn.CCEDGE** = 11b (rising or falling edge) and a rising or a falling edge is detected in the input signal, then the **TDINT.CCRnIF** interrupt flag is set.

If the **TDINT.CCRnIF** interrupt flag is set and the corresponding **TDCCTLn.CCINTEN** bit is set, then the **TIMERD_IRQ** signal to the NVIC will be asserted.

The **TDINT.CCRnIF** interrupt flag can be cleared by writing it to a 1b.

21.4.16 Timer IRQ signal

The timer unit has one IRQ output signal to the NVIC (**TIMERD_IRQ**).

This signal is asserted when any of the following conditions are true:

- **TDINT.BASEIF** = 1b and **TDCTL.BASEIE** = 1b
- **TDINT.CCR0IF** = 1b and **TDCCTL0.CCINTEN** = 1b
- **TDINT.CCR1IF** = 1b and **TDCCTL1.CCINTEN** = 1b
- **TDINT.CCR2IF** = 1b and **TDCCTL2.CCINTEN** = 1b
- **TDINT.CCR3IF** = 1b and **TDCCTL3.CCINTEN** = 1b
- **TDINT.CCR4IF** = 1b and **TDCCTL4.CCINTEN** = 1b

- **TDINT.CCR5IF** = 1b and **TDCCTL5.CCINTEN** = 1b
- **TDINT.CCR6IF** = 1b and **TDCCTL6.CCINTEN** = 1b
- **TDINT.CCR7IF** = 1b and **TDCCTL7.CCINTEN** = 1b

The **TIMERD_IRQ** signal is de-asserted when all of the above conditions are false.

21.4.17 Skipping CCR Interrupts

Sometimes it is useful to not generate a CCR interrupt to the NVIC every time the **TDCTR** counts to the **TDCCTRn.CTR** value. For example, in a control application when the PWM frequency is fast, but you only need a MCU interrupt every 5 PWM periods.

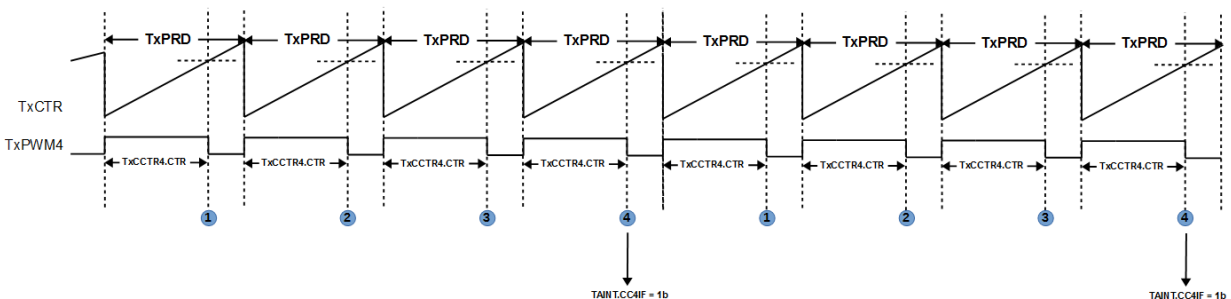
Each CCR in the timer has the ability to skip interrupts to the NVIC to allow this capability.

The CCR will generate interrupts to the NVIC according to the **TDCCTL.CCINTSKIP** field as follows:

- If **TDCCTLn.CCINTSKIP** = 0000b, then the CCR will generate an interrupt each time
- If **TDCCTLn.CCINTSKIP** = 0001b, then the CCR will skip 1 CCR cycle before generating an interrupt to the NVIC.
- If **TDCCTLn.CCINTSKIP** = 0010b, then the CCR will skip 2 CCR cycles before generating an interrupt to the NVIC.
- If **TDCCTLn.CCINTSKIP** = 0011b, then the CCR will skip 3 CCR cycles before generating an interrupt to the NVIC.
- ...
- If **TDCCTLn.CCINTSKIP** = 1110b, then the CCR will skip 14 CCR cycles before generating an interrupt to the NVIC.
- If **TDCCTLn.CCINTSKIP** = 1111b, then the CCR will skip 15 CCR cycles before generating an interrupt to the NVIC.

The diagram below shows an example of a CCR4 with interrupt skipping configured with a value of 4 (**TDCCTL.CCINTSKIP** = 4).

Figure 21-8 CCR Interrupt Skipping Diagram



21.4.18 Timer ADC Triggers

Each timer CCR unit may be configured to be an ADC trigger, to allow the DTSE to automatically begin conversion sequences.

For more information on this feature, see the section on the ADC and DTSE in this user guide.

21.4.19 Dead-Time Generators (DTG)

Each timer has 4 DTG units. Each DTG unit is capable of generating a pair of complementary signals that can be configured with dead-time to drive an inverter for half-bridge topologies.

The input clock to the DTG can be configured to be the timer clock before or after the **TDCTL.CLKDIV** input clock divider. If **TDCTL.DTGCLK** is 0b, then the DTGCLK is the clock before the **TDCTL.CLKDIV** clock divider. If **TDCTL.DTGCLK** is 1b, then the DTGCLK is the clock after the **TDCTL.CLKDIV** clock divider.

When **TDDTGCTLn.DTEN** = 0b, the DTG is disabled (bypassed). In this mode, the timer signals have the following behavior (see the simplified block diagram above):

DTG0:

- CCR0 output is connected to TDPWM0
- CCR4 output is connected to TDPWM4

DTG1:

- CCR1 output is connected to TDPWM1
- CCR5 output is connected to TDPWM5

DTG2:

- CCR2 output is connected to TDPWM2
- CCR6 output is connected to TDPWM6

DTG3:

- CCR3 output is connected to TDPWM3
- CCR7 output is connected to TDPWM7

When **TDDTGCTL0.DTEN** = 1b, the DTG is enabled. In this mode, only one CCR output is used to generate the two complementary outputs. In this mode, the timer signals have the following behavior:

DTG0:

- CCR0 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR4 output is used to generate the two complementary output signals: TDPWM0 and TDPWM4.
- TDPWM0 becomes the low-side complementary signal.
- TDPWM4 becomes the high-side complementary signal.

- The **TDDTGCTL0.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, to when the low-side signal rises).
- The **TDDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, to when the high-side signal rises).

DTG1:

- CCR1 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR5 output is used to generate the two complementary output signals: TDPWM1 and TDPWM5.
- TDPWM1 becomes the low-side complementary signal.
- TDPWM5 becomes the high-side complementary signal.
- The **TDDTGCTL1.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TDDTGCTL1.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG2:

- CCR2 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR6 output is used to generate the two complementary output signals: TDPWM2 and TDPWM6.
- TDPWM2 becomes the low-side complementary signal.
- TDPWM6 becomes the high-side complementary signal.
- The **TDDTGCTL2.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TDDTGCTL2.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

DTG3:

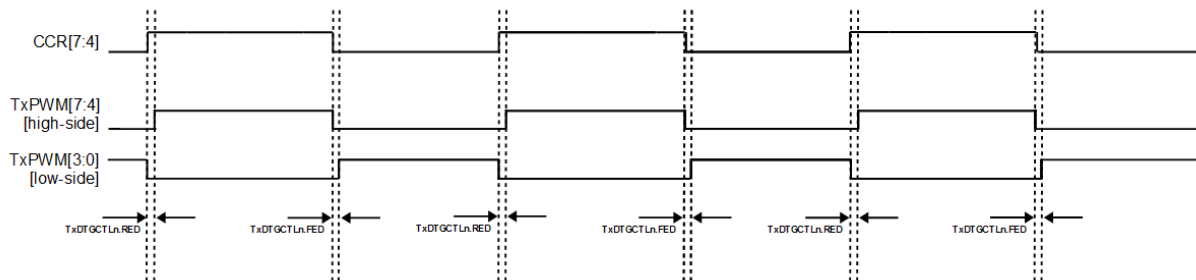
- CCR3 output is unused. This CCR output may still be used to generate CCR interrupts or ADC timer triggers, but will not be connected to the Digital Peripheral MUX.
- CCR7 output is used to generate the two complementary output signals: TDPWM3 and TDPWM4.

- TDPWM3 becomes the low-side complementary signal.
- TDPWM7 becomes the high-side complementary signal.
- The **TDDTGCTL3.FED** specifies the number of DTGCLK ticks that are applied for falling edge delay (the delay from when the high-side signal falls, and when the low-side signal rises).
- The **TDDTGCTL0.RED** specifies the number of DTGCLK ticks that are applied for rising edge delay (the delay from when the low-side signals falls, and when the high-side signal rises).

The PWM waveform below shows an example of the dead-time applied to the signals through the DTG units.

21.4.20 Timer D DTG Output

Figure 21-9 Timer D DTG Output



21.5 QEP

The PAC55XX contains a QEP encoder peripheral that can be used to determine position and speed of a position-encoded motor. The QEP encoder can be enabled by setting the **TDQEPCTL.QEPEN** bit to a 1b.¹³

The inputs to the QEP peripheral are TDQEPPHA, TDQEPPHB and TDQEPIDX (phase A, phase B and index). These inputs are used to determine the position and direction of the motor, when the **TDQEPCTL.QEPEN** is set to a 1b. These three inputs have a glitch filter applied to them, to filter out noise and are used for calculation of the motor position and speed.

After filtering, the TDQEPPHA, TDQEPPHB and TDQEPIDX inputs are passed to the QEP engine to determine the position and speed. If the position is determined to be clockwise, then the **TDQEPCTL.DIR** is set to 0b. If the position is determined to be counter-clockwise, then the **TDQEPCTL.DIR** is set to 1b.

During operation, if the **TDQEPCTL.CNTAB** bit is 0b, only phase A edges are counted. If this bit is set to a 1b, then both phase A and phase B edges are counted. If the **TDQEPCTL.CNTEDGE** bit is set to 0b, then only the rising edges are counted. If this bit is set to a 1b, then both rising and falling edges are counted.

If the **TDQEPCTL.IDXRST** bit is set to a 1b, then an index event clears the **TDQEPCTL.TICKS** counter to 0. If this bit is set to a 0b, then an index event does not reset the **TDQEPCTL.TICKS** counter.

If a change in the direction is made (0 to 1, or 1 to 0), then the **TDQEPCTL.DIRIF** bit is set to a 1b. If the **TDQEPCTL.DIRIE** bit is set to 1b, then the QEPD_IRQ signal is asserted to the NVIC. The **TDQEPCTL.DIRIF** bit can be cleared by writing it to 1b.

When a rising edge on phase A is detected the **TDQEPCTL.PHAIF** bit is set to 1b. If the **TDQEPCTL.PHAIE** bit is set and the **TDQEPCTL.PHAIF** bit is set, then the QEPD_IRQ signal is asserted to the NVIC. The **TDQEPCTL.PHAIF** bit can be cleared by writing a 1b to it.

When a rising edge on phase B is detected the **TDQEPCTL.PHBIF** bit is set. If the **TDQEPCTL.PHBIE** bit is set and the **TDQEPCTL.PHBIF** bit is set, then the QEPD_IRQ signal is asserted to the NVIC. The **TDQEPCTL.PHBIF** bit can be cleared by writing a 1b to it.

If an overflow or underflow in the **TDQEPCTL.TICKS** is detected, the **TDQEPCTL.WRIF** bit is set (counter wrap interrupt flag). If the **TDQEPCTL.WRIE** bit is set and the **TDQEPCTL.WRIF** bits are set, then the QEPD_IRQ signal is asserted to the NVIC.

If an index event is detected, the **TDQEPCTL.IDXEVI** bit is set. If the **TDQEPCTL.IDXEVI** bit is set and the **TDQEPCTL.IDXEVI** bit is set, then the QEPD_IRQ signal to the NVIC is asserted. The **TDQEPCTL.IDXEVI** bit can be cleared by writing a 1b to it.

¹³ Before enabling the QEP peripheral, be sure to set up the Digital Peripheral MUX, so that no false edges are detected by the QEP state machine.

As the QEP decoder counts edges on PH_A and is turning clockwise (CW), it increments the counter value in **TDQEPCTL.TICKS**. If the motor is turning counter-clockwise (CCW), the **TDQEPCTL.TICKS** register is decremented. The user may write the **TDQEPCTL.TICKS** register at any time.

21.6 Peripheral IO Mapping

The Timer D peripheral signal inputs and outputs (CCR, QEP) are connected to the digital peripheral MUX so they can be used by various IO pins.

The table below shows which peripheral outputs may be connected to which IO pins.

21.7 Timer D Peripheral IO Mapping

| TIMER SIGNAL | IO PIN |
|--------------|-------------------------|
| TDPWM0 | PE0, PF0, PG0 |
| TDPWM1 | PE1, PF1, PG1 |
| TDPWM2 | PE2, PF2, PG2 |
| TDPWM3 | PE3, PF3, PG3 |
| TDPWM4 | PE4, PF4, PG4 |
| TDPWM5 | PE5, PF5, PG5 |
| TDPWM6 | PE6, PF6, PG6 |
| TDPWM7 | PE7, PF7 |
| TDQEPIDX | PD0, PD4, PE4, PG4, PG7 |
| TDQEPPHA | PD1, PD5, PE5, PG5 |
| TDQEPPHB | PD2, PD6, PE6, PG6 |

For more information on how to use the Digital Peripheral MUX to connect peripheral signals to IO, see Register Summary

Table 21-1 Timer D Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|---------------------------|--------|------------|
| TDCTL | 4009 0000h | Timer D control | RW | 0000 0000h |
| TDINT | 4009 0004h | Timer D interrupt control | RW | 0000 0000h |
| TDPRD | 4009 0008h | Timer D period | RW | 0000 0000h |
| TDCTR | 4009 000Ch | Timer D counter | RW | 0000 0000h |
| TDQEPCTL | 4009 0010h | Timer D QEP control | RW | 0000 0000h |
| TDCCTL0 | 4009 0100h | Timer D CC control 0 | RW | 0000 0000h |
| TDCCTR0 | 4009 0104h | Timer D CC counter 0 | RW | 0000 0000h |
| TDCCTL1 | 4009 0108h | Timer D CC control 1 | RW | 0000 0000h |
| TDCCTR1 | 4009 010Ch | Timer D CC counter 1 | RW | 0000 0000h |
| TDCCTL2 | 4009 0110h | Timer D CC control 2 | RW | 0000 0000h |
| TDCCTR2 | 4009 0114h | Timer D CC counter 2 | RW | 0000 0000h |
| TDCCTL3 | 4009 0118h | Timer D CC control 3 | RW | 0000 0000h |
| TDCCTR3 | 4009 011Ch | Timer D CC counter 3 | RW | 0000 0000h |
| TDCCTL4 | 4009 0120h | Timer D CC control 4 | RW | 0000 0000h |
| TDCCTR4 | 4009 0124h | Timer D CC counter 4 | RW | 0000 0000h |
| TDCCTL5 | 4009 0128h | Timer D CC control 5 | RW | 0000 0000h |
| TDCCTR5 | 4009 012Ch | Timer D CC counter 5 | RW | 0000 0000h |

| | | | | |
|------------------|------------|-----------------------|----|------------|
| TDCCTL6 | 4009 0130h | Timer D CC control 6 | RW | 0000 0000h |
| TDCCTR6 | 4009 0134h | Timer D CC counter 6 | RW | 0000 0000h |
| TDCCTL7 | 4009 0138h | Timer D CC control 7 | RW | 0000 0000h |
| TDCCTR7 | 4009 013Ch | Timer D CC counter 7 | RW | 0000 0000h |
| TDDTGCTL0 | 4009 0200h | Timer D DTG control 0 | RW | 0000 0000h |
| TDDTGCTL1 | 4009 0204h | Timer D DTG control 1 | RW | 0000 0000h |
| TDDTGCTL2 | 4009 0208h | Timer D DTG control 2 | RW | 0000 0000h |
| TDDTGCTL3 | 4009 020Ch | Timer D DTG control 3 | RW | 0000 0000h |

21.8 Register Detail

21.8.1 TDCTL

Register 21-1 TDCTL (Timer D Control, 4009 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | BASEIE | RW | 0 | Base timer interrupt enable: 0b: not enabled 1b: enabled |
| 12 | CLR | RW | 0 | Base timer clear: 0b: do not clear 1b: clear timer counter |
| 11 | LATCH | RW | 0 | When written to 1b, this will latch the TDPRD and all TDCCTRn registers on the same clock cycle. Writing this bit to 0b has no effect. This is a self-clearing bit. |
| 10 | DTGCLK | RW | 0 | DTG Clock Source: 0b: PCLK 1b: ACLK |
| 9 | CLKSRC | RW | 0 | Timer Clock Source: 0b: PCLK 1b: ACLK |
| 8:6 | CLKDIV | RW | 0 | Timer Input Clock Divider: 000b: /1 001b: /2 010b: /4 011b: /8 100b: /16 101b: /32 110b: /64 111b: /128 |
| 5 | SINGLE | RW | 0 | Single Shot Timer: 0b: disabled (auto-reload) 1b: enabled (single shot timer) |
| 4 | SSYNC | RW | 0 | Timer Slave Synchronization: 0b: disabled 1b: enabled |
| 3:2 | PRDLATCH | RW | 0 | Timer Period Latch Mode: 00b: Latch TDPRD when TDCTR = 0 01b: Latch TDPRD when TDCTR = TDPRD 10b: Latch TDPRD immediately upon register write 11b: reserved |
| 1:0 | MODE | RW | 0 | Timer Mode: |

| | | | | |
|--|--|--|--|--|
| | | | | 00b: disabled 01b: up mode 10b: up/down mode 11b: asymmetric mode |
|--|--|--|--|--|

21.8.2 TDINT

Register 21-2 TDINT (Timer D Interrupt Control, 4009 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:9 | Reserved | RO | 0 | Reserved |
| 8 | BASEIF | W1C | 0 | Base timer interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 7 | CCR7IF | W1C | 0 | CCR7 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 6 | CCR6IF | W1C | 0 | CCR6 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 5 | CCR5IF | W1C | 0 | CCR5 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 4 | CCR4IF | W1C | 0 | CCR4 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 3 | CCR3IF | W1C | 0 | CCR3 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 2 | CCR2IF | W1C | 0 | CCR2 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 1 | CCR1IF | W1C | 0 | CCR1 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |
| 0 | CCR0IF | W1C | 0 | CCR0 interrupt flag: 0b: no interrupt flag 1b: interrupt flag |

21.8.3 TCPRD

Register 21-3 TDPRD (Timer D Period, 4009 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--------------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | PERIOD | RW | 0 | Timer period value |

21.8.4 TDCTR

Register 21-4 TDCTR (Timer D Counter, 4009 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---------------|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | COUNTER | RW | 0 | Timer counter |

21.8.5 TDQEPCTL

Register 21-5 TDQEPCTL (Timer D QEP Control, 4009 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | TICKS | RW | 0 | Number of counted phase edges according to counting mode defined by TDQEPCTL.CNTEDGE and TDQEPCTL.CNTAB . |
| 15 | Reserved | RO | 0 | Reserved |
| 14 | CNTEDGE | RW | 0 | Count on edge: 0b: Rising edge only 1b: Rising and falling edge |
| 13 | CNTAB | RW | 0 | Count on A/B: 0b: Count phase A only 1b: Count phase A and phase B |
| 12 | IDXEIVF | W1C | 0 | Index event interrupt flag: 0b: no index event detected 1b: index event detected |
| 11 | WRIF | W1C | 0 | Counter wrap interrupt flag: 0b: no counter wrap 1b: counter wrap |
| 10 | PHBIF | W1C | 0 | Phase B rising edge interrupt flag: 0b: no phase B rising edge detected 1b: phase B rising edge detected |
| 9 | PHAIF | W1C | 0 | Phase A rising edge interrupt flag: 0b: no phase A rising edge detected 1b: phase A rising edge detected |
| 8 | DIRIF | W1C | 0 | Direction change interrupt flag: 0b: no direction change detected 1b: direction change detected |
| 7 | IDXRST | RW | 0 | Reset counter on index: 0b: Do not reset TDQEPCTL.TICKS 1b: Reset TDQEPCTL.TICKS on all index events |
| 6 | IDXEIVIE | RW | 0 | Index event interrupt enable: 0b: disabled 1b: enabled |
| 5 | WRIE | RW | 0 | Counter wrap interrupt enable: 0b: disabled 1b: enabled |
| 4 | PHBIE | RW | 0 | Phase B rising edge interrupt enable: 0b: disabled 1b: enabled |
| 3 | PHAIE | RW | 0 | Phase A rising edge interrupt enable: |

| | | | | |
|---|--------------|----|---|---|
| | | | | 0b: disabled 1b: enabled |
| 2 | DIRIE | RW | 0 | Direction change interrupt enable: 0b: disabled 1b: enabled |
| 1 | DIR | RO | 0 | Motor direction: 0b: CW (clockwise) 1b: CCW (counter-clockwise) |
| 0 | QEPEN | RW | 0 | QEP peripheral enabled: 0b: disabled 1b: enabled |

21.8.6 TDCCTL0

Register 21-6 TDCCTL0 (Timer D CCR Control 0, 4009 0100h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL0.CCMODE = 0b (compare): 00b: Latch TDCCTRL0 registers when TDCTR = 0 01b: Latch TDCCTRL0 registers when TDCTR = TDPRD 10b: Latch TDCCTRL0 registers immediately 11b: Reserved If TDCCTL0.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR0.CTR on rising edge 01b: Latch TDCTR into TDCCTR0.CTR on falling edge 10b: Latch TDCTR into TDCCTR0.CTR on both edges 11b: Latch TDCTR into TDCCTR0.CAP1 and TDCCTR0.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL0.CCMODE = 1b (capture), set TDINT.CCR0IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.7 TDCCTR0

Register 21-7 TDCCTR0 (Timer D CCR Counter 0, 4009 0104h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTL0.CCLATCH != 11b (period/duty mode) and TDCCTL0.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTL0.CCLATCH = 11b (period/duty mode) and TDCCTL0.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTL0.CCLATCH != 11b (period/duty mode) and TDCCTR0.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTL0.CCLATCH != 11b (period/duty mode) and TDCCTL0.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTL0.CCLATCH = 11b (period/duty mode) and TDCCTL0.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.8 TDCCTL1

Register 21-8 TDCCTL1 (Timer D CCR Control 1, 4009 0108h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|------------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL1.CCMODE = 0b (compare): 00b: Latch TDCCTL1 registers when TDCTR = 0 01b: Latch TDCCTL1 registers when TDCTR = TDPRD 10b: Latch TDCCTL1 registers immediately 11b: Reserved If TDCCTL1.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR1.CTR on rising edge 01b: Latch TDCTR into TDCCTR1.CTR on falling edge 10b: Latch TDCTR into TDCCTR1.CTR on both edges 11b: Latch TDCTR into TDCCTR1.CAP1 and TDCCTR1.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL1.CCMODE = 1b (capture), set TDINT.CCR1IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.9 TDCCTR1

Register 21-9 TDCCTR1 (Timer D CCR Counter 1, 4009 010Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTL1.CCLATCH != 11b (period/duty mode) and TDCCTL1.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTL1.CCLATCH = 11b (period/duty mode) and TDCCTL1.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTL1.CCLATCH != 11b (period/duty mode) and TDCCTL1.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTL1.CCLATCH != 11b (period/duty mode) and TDCCTL1.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTL1.CCLATCH = 11b (period/duty mode) and TDCCTL1.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.10 TDCCTL2

Register 21-10 TDCCTL2 (Timer D CCR Control 2, 4009 0110h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL2.CCMODE = 0b (compare): 00b: Latch TDCCTL2 registers when TDCTR = 0 01b: Latch TDCCTL2 registers when TDCTR = TDPRD 10b: Latch TDCCTL2 registers immediately 11b: Reserved If TDCCTL2.CCMODE = 1b (capture): 00b: Latch TDBCTR into TDCCTR2.CTR on rising edge 01b: Latch TDCTR into TDCCTR2.CTR on falling edge 10b: Latch TDCTR into TDCCTR2.CTR on both edges 11b: Latch TDCTR into TDCCTR2.CAP1 and TDCCTR2.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL2.CCMODE = 1b (capture), set TDINT.CCR2IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.11 TDCCTR2

Register 21-11 TDCCTR2 (Timer D CCR Counter 2, 4009 0114h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL2.CCLATCH != 11b (period/duty mode) and TDCCTRL2.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL2.CCLATCH = 11b (period/duty mode) and TDCCTRL2.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL2.CCLATCH != 11b (period/duty mode) and TDCCTRL2.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL2.CCLATCH != 11b (period/duty mode) and TDCCTRL2.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL2.CCLATCH = 11b (period/duty mode) and TDCCTRL2.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.12 TDCCTL3

Register 21-12 TDCCTL3 (Timer D CCR Control 3, 4009 0118h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL3.CCMODE = 0b (compare): 00b: Latch TDCCTL3 registers when TDCTR = 0 01b: Latch TDCCTL3 registers when TDCTR = TDPRD 10b: Latch TDCCTL3 registers immediately 11b: Reserved If TDCCTL3.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR3.CTR on rising edge 01b: Latch TDCTR into TDCCTR3.CTR on falling edge 10b: Latch TDCTR into TDCCTR3.CTR on both edges 11b: Latch TDCTR into TDCCTR3.CAP1 and TDCCTR3.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL3.CCMODE = 1b (capture), set TDINT.CCR3IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.13 TDCCTR3

Register 21-13 TDCCTR3 (Timer D CCR Counter 3, 4009 011Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL3.CCLATCH != 11b (period/duty mode) and TDCCTRL3.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL3.CCLATCH = 11b (period/duty mode) and TDCCTRL3.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL3.CCLATCH != 11b (period/duty mode) and TDCCTRL3.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL3.CCLATCH != 11b (period/duty mode) and TDCCTRL3.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL3.CCLATCH = 11b (period/duty mode) and TDCCTRL3.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.14 TDCCTL4

Register 21-14 TDCCTL4 (Timer D CCR Control 4, 4009 0120h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL4.CCMODE = 0b (compare): 00b: Latch TDCCTL4 registers when TDCTR = 0 01b: Latch TDCCTL4 registers when TDCTR = TDPRD 10b: Latch TDCCTL4 registers immediately 11b: Reserved If TDCCTL4.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR4.CTR on rising edge 01b: Latch TDCTR into TDCCTR4.CTR on falling edge 10b: Latch TDCTR into TDCCTR4.CTR on both edges 11b: Latch TDCTR into TDCCTR4.CAP1 and TDCCTR4.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL4.CCMODE = 1b (capture), set TDINT.CCR4IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.15 TDCCTR4

Register 21-15 TDCCTR4 (Timer D CCR Counter 4, 4009 0124h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL4.CCLATCH != 11b (period/duty mode) and TDCCTRL4.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL4.CCLATCH = 11b (period/duty mode) and TDCCTRL4.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL4.CCLATCH != 11b (period/duty mode) and TDCCTRL4.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL4.CCLATCH != 11b (period/duty mode) and TDCCTRL4.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL4.CCLATCH = 11b (period/duty mode) and TDCCTRL4.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.16 TDCCTL5

Register 21-16 TDCCTL5 (Timer D CCR Control 5, 4009 0128h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL5.CCMODE = 0b (compare): 00b: Latch TDCCTL5 registers when TDCTR = 0 01b: Latch TDCCTL5 registers when TDCTR = TDPRD 10b: Latch TDCCTL5 registers immediately 11b: Reserved If TDCCTL5.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR5.CTR on rising edge 01b: Latch TDCTR into TDCCTR5.CTR on falling edge 10b: Latch TDCTR into TDCCTR5.CTR on both edges 11b: Latch TDCTR into TDCCTR5.CAP1 and TDCCTR5.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL5.CCMODE = 1b (capture), set TDINT.CCR5IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.17 TDCCTR5

Register 21-17 TDCCTR5 (Timer D CCR Counter 5, 4009 012Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL5.CCLATCH != 11b (period/duty mode) and TDCCTRL5.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL5.CCLATCH = 11b (period/duty mode) and TDCCTRL5.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL5.CCLATCH != 11b (period/duty mode) and TDCCTRL5.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL5.CCLATCH != 11b (period/duty mode) and TDCCTRL5.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL5.CCLATCH = 11b (period/duty mode) and TDCCTRL5.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.18 TDCCTL6

Register 21-18 TDCCTL6 (Timer D CCR Control 6, 4009 0130h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL6.CCMODE = 0b (compare): 00b: Latch TDCCTL6 registers when TDCTR = 0 01b: Latch TDCCTL6 registers when TDCTR = TDPRD 10b: Latch TDCCTL6 registers immediately 11b: Reserved If TDCCTL6.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR6.CTR on rising edge 01b: Latch TDCTR into TDCCTR6.CTR on falling edge 10b: Latch TDCTR into TDCCTR6.CTR on both edges 11b: Latch TDCTR into TDCCTR6.CAP1 and TDCCTR6.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL6.CCMODE = 1b (capture), set TDINT.CCR6IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.19 TDCCTR6

Register 21-19 TDCCTR6 (Timer D CCR Counter 6, 4009 0134h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL6.CCLATCH != 11b (period/duty mode) and TDCCTRL6.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL6.CCLATCH = 11b (period/duty mode) and TDCCTRL6.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL6.CCLATCH != 11b (period/duty mode) and TDCCTRL6.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL6.CCLATCH != 11b (period/duty mode) and TDCCTRL6.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL6.CCLATCH = 11b (period/duty mode) and TDCCTRL6.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.20 TDCCTL7

Register 21-20 TDCCTL7 (Timer D CCR Control 7, 4009 0138h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:8 | CCINTSKIP | RW | 0 | CC Interrupt Skip Counter: 0000b: Don't skip CCR matches before interrupt 0001b: Skip 1 CCR match before interrupt 0010b: Skip 2 CCR matches before interrupt 0011b: Skip 3 CCR matches before interrupt ... 1110b: Skip 14 CCR matches before interrupt 1111b: Skip 15 CCR matches before interrupt |
| 7 | CCFORCE | RW | 0 | Write this bit to a 1b to force this compare event. This is a self-clearing bit. |
| 6:5 | CCLATCH | RW | 0 | CCR register latch mode. If TDCCTL7.CCMODE = 0b (compare): 00b: Latch TDCCTL7 registers when TDCTR = 0 01b: Latch TDCCTL7 registers when TDCTR = TDPRD 10b: Latch TDCCTL7 registers immediately 11b: Reserved If TDCCTL7.CCMODE = 1b (capture): 00b: Latch TDCTR into TDCCTR7.CTR on rising edge 01b: Latch TDCTR into TDCCTR7.CTR on falling edge 10b: Latch TDCTR into TDCCTR7.CTR on both edges 11b: Latch TDCTR into TDCCTR7.CAP1 and TDCCTR7.CAP2 according to period/duty calculation |
| 4 | CCOUTINV | RW | 0 | Invert CCR output: 0b: Do not invert CCR output 1b: Invert CCR output |
| 3:2 | CCINTEDGE | RW | 0 | When TDCCTL7.CCMODE = 1b (capture), set TDINT.CCR7IF = 1b when input signal detects: 00b: rising edge 01b: falling edge 10b: either rising or falling edges 11b: reserved |
| 1 | CCINTEN | RW | 0 | CCR interrupt enable: 0b: disabled 1b: enabled |
| 0 | CCMODE | RW | 0 | CCR mode: 0b: compare 1b: capture |

21.8.21 TDCCTR7

Register 21-21 TDCCTR7 (Timer D CCR Counter 7, 4009 013Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------|--------|-------|---|
| 31:16 | DELAY | RW | 0 | If TDCCTRL7.CCLATCH != 11b (period/duty mode) and TDCCTRL7.CCMODE = 1b (capture mode), this the delay count to use before compare events that can be used for PWM output of phase shifted topologies. |
| 31:16 | CAP2 | RW | 0 | If TDCCTRL7.CCLATCH = 11b (period/duty mode) and TDCCTRL7.CCMODE = 1b (capture mode), this is the 2 nd capture value used for this calculation (number of ticks in duty cycle) |
| 15:0 | CTR | RW | 0 | If TDCCTRL7.CCLATCH != 11b (period/duty mode) and TDCCTRL7.CCMODE = 0b (compare mode), this is the compare value used for this CCR. If TDCCTRL7.CCLATCH != 11b (period/duty mode) and TDCCTRL7.CCMODE = 1b (capture mode), this is the capture value for this CCR. |
| 15:0 | CAP1 | RW | 0 | If TDCCTRL7.CCLATCH = 11b (period/duty mode) and TDCCTRL7.CCMODE = 1b (capture mode), this is the capture value used for this calculation (number of ticks in period). |

21.8.22 TDDTGCTL0

Register 21-22 TDDTGCTL0 (Timer D Dead-Time Generator Control 0, 4009 0200h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG0. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG0. |

21.8.23 TDDTGCTL1

Register 21-23 TDDTGCTL1 (Timer D Dead-Time Generator Control 1, 4009 0204h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG1. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG1. |

21.8.24 TDDTGCTL2

Register 21-24 TDDTGCTL2 (Timer D Dead-Time Generator Control 2, 4009 0208h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG2. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG2. |

21.8.25 TDDTGCTL3

Register 21-25 TDDTGCTL3 (Timer D Dead-Time Generator Control 3, 4009 020Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | DTEN | RW | 0 | Dead-time generator enable: 0b: not enabled 1b: enabled |
| 30:28 | Reserved | RO | 0 | Reserved |
| 27:16 | FED | RW | 0 | Falling edge delay: The number of DTGCLK ticks used for falling edge dead-time delay for DTG3. |
| 15:12 | Reserved | RW | 0 | Reserved |
| 11:0 | RED | RW | 0 | Rising edge delay: The number of DTGCLK ticks used for rising edge dead-time delay for DTG3. |

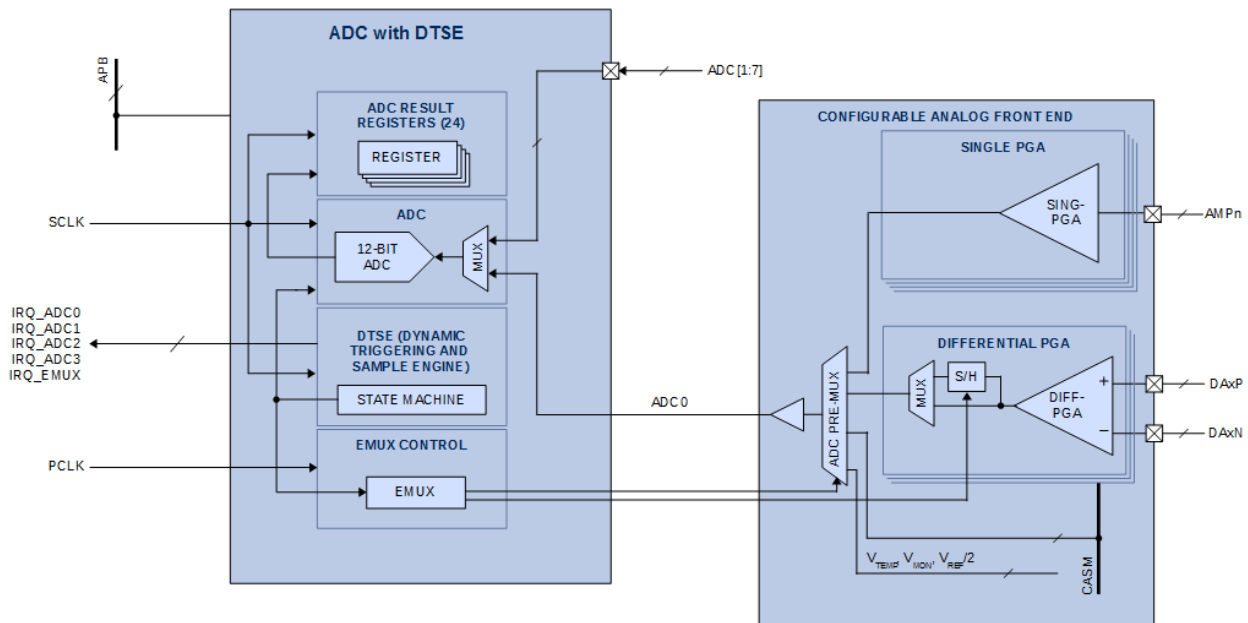
22 ADC AND DTSE

22.1 Overview

The PAC55XX contains a powerful Analog to Digital Converter (ADC) and Dynamic Triggering and Sampling Engine (DTSE).

The ADC is an 8-channel 12-bit 2.5MSPS SAR architecture Analog to Digital Converter. The ADC is controlled by a highly flexible and programmable sequencing engine called the DTSE.

Figure 22-1 ADC and DTSE Simplified Block Diagram



The ADC is an APB bus client. The ADC is clocked by the SCLK system clock and the DTSE and EMUX are clocked by the PCLK peripheral clock. The ADC may be programmed to generate up to 4 different interrupts to the NVIC.

The ADC works with the Configurable Analog Front-End™ (CAFE) to provide additional ADC channels as well as signal conditioning capabilities.

22.2 Features

The PAC55XX family ADC and DTSE peripheral has the following features:

- ADC Features:
 - 2.5 MSPS, 12-bit SAR ADC
 - 8 Input Channels
 - Expandable through the CAFE
 - Up to 40MHz Input Clock

- 4-bit Input Clock Divider
- Power-down mode
- EMUX serial port for low-latency communication to the CAFE
 - Up to 25MHz
 - Input clock divider
 - Busy status bit
 - Interrupt for EMUX done
- Manual or automatic (DTSE) conversion mode
- DTSE Features:
 - Up to 32 Input triggers
 - PWM output or capture input triggers for start of sequence
 - Software triggers for conversions
 - Multiple triggers may share the same conversion series for maximum flexibility and efficiency
 - Any conversion sequence may generate up to 4 IRQ signals to the NVIC
 - Automatic copying of conversion results to dedicated results registers (24)
 - Sending of EMUX command by hardware using sequencing engine
 - Inserting of conversion delays by hardware using sequencing engine

22.3 Functional Description

22.3.1 ADC

The ADC is a 12-bit 2.5MSPS SAR ADC. The ADC is powered by the VCORE 1.2V input for the digital logic (DTSE) and VCC33 3.3V input for the analog portion.

The PAC55XX generates its own voltage reference for the ADC, which can be either 2.5V or 3.0V.

The ADC has 8 analog input channels named ADC0 to ADC7. ADC0 is a special-purpose channel which is used to provide an analog input from the CAFE sub-system, which can perform additional signal conditioning, sample and hold via programmable gain single ended and differential amplifiers. For more information on the capabilities of the PAC55XX CAFE system, see the device Analog User Guide.

22.3.2 Enabling the ADC

To enable the ADC, set the **ADCCTL.ENABLE** to 1b. The ADC will not be able to perform any conversions in manual or DTSE modes unless this bit is set to 1b.

22.3.3 ADC Power-Down Mode

When not in use, it is recommended that the ADC put into 'power-down' mode. Even when the ADC is not being used by the rest of the PAC55XX, the analog portion of the ADC may still consume power.

In order to turn off the analog portion of the ADC when it is not in use, set **ADCCTL.ENABLE** to 0b. When this bit is set to 0b, the ADC registers will still be accessible, but the ADC will not be able to be used.

During normal operation, the **ADCCTL.ENABLE** will need to be set to 1b.

22.3.4 ADC Clock

The ADC and DTSE peripheral uses an input clock for ADC conversions and to run the DTSE. The ADC and DTSE use the SCLK system clock as the input clock.

The maximum clock frequency that the ADC supports for conversions is 40MHz. The SCLK input clock may be much faster than this (up to 150MHz), so there is an ADC clock divider to reduce the input frequency to a maximum of 40MHz.

The **ADCCTL.ADCDIV** is a 4-bit divider that may be configured to divide the SCLK input clock from between 1 to 16. The table below shows how to configure the divider to the desired setting.

Table 22-1 ADC Input Clock Divider

| ADCCTL.ADCDIV | ADCCLK |
|---------------|---------|
| 000b | SCLK /1 |
| 001b | SCLK /2 |
| 010b | SCLK /3 |
| 011b | SCLK /4 |
| 100b | SCLK /5 |
| 101b | SCLK /6 |
| 110b | SCLK /7 |
| 111b | SCLK /8 |

22.3.5 EMUX

In order to enable low-latency communication with the CAFE, the ADC and DTSE uses an EMUX peripheral. The EMUX is a dedicated, fast, low-latency serial interface used for quickly communicating data needed for signal sampling to the CAFE.

The EMUX may be used manually or in DTSE mode.

When the ADC is configured in manual mode (**ADCCTL.MODE** = 0b), the EMUX must also be set in manual mode. To set the EMUX into manual mode, set the **EMUXCTL.EMUXMODE** to 0b. In this mode, whenever the **EMUXDATA** register is written, the EMUX will write this data onto the bus to the CAFE sub-system.

When the ADC is configured in DTSE mode (**ADCCTL.MODE** = 01b), the EMUX must also be set into DTSE mode. To set the EMUX into DTSE mode, set the **EMUXCTL.EMUXMODE** to 1b.

In this mode, the EMUX is written to automatically by the DTSE. Writing data into the **EMUXDATA** register has no effect.

When the EMUX is busy, the **EMUXCTL.EMUXBUSY** bit will be set to 1b. If the EMUX is not in use, then this bit will be a 0b.

The EMUX clock may be as high as 25MHz and is supplied from the PCLK peripheral clock. Since the PCLK frequency may be much higher than the EMUX clock, the EMUX has an input clock divider that may be used to reduce the input clock rate to 25MHz or lower.

The **EMUXCTL.EMUXDIV** contains the value to be used for the EMUX input clock divider. See the table below for how this may be configured to produce the desired EMUX clock divider.

Table 22-2 EMUX Input Clock Divider

| EMUXCTL.EMUXDIV | EMUXCLK |
|-----------------|---------|
| 000b | PCLK /1 |
| 001b | PCLK /2 |
| 010b | PCLK /3 |
| 011b | PCLK /4 |
| 100b | PCLK /5 |
| 101b | PCLK /6 |
| 110b | PCLK /7 |
| 111b | PCLK /8 |

22.3.6 EMUX Interrupts

The EMUX may be configured to generate an interrupt when it has completed writing data.

Whenever the EMUX completes writing data, it will set the **ADCINT.INTFEMUX** to a 1b. If the EMUX interrupt is enabled (**ADCCTL.INTENEMUX** = 1b), then the IRQ_ADC0 signal to the NVIC will be asserted.

Writing **ADCINT.INTFEMUX** to a 1b will clear this bit and de-assert the IRQ_ADC0, if asserted.

22.3.7 Manual ADC Conversions

To perform manual ADC conversions, follow the steps below:

- Enable the ADC by setting **ADCCTL.ENABLE** to 1b.
- Set the ADC mode to manual mode by setting **ADCCTL.MODE** to 0b.
- Select the ADC input channel by writing the **ADCCTL.CHANNEL** to the desired ADC input channel number (ADC0 to ADC7).
- Start the ADC conversion by setting the **ADCCTL.START** to a 1b. This bit is self-clearing and will always be read as 0b.

- The ADC may have its channel selected and start the conversion at the same time by writing both the **ADCCTL.START** and **ADCCTL.CHANNEL** fields in the same register write to **ADCCTL**.
 - At this point, the ADC will set the **ADCCTL.ADBUSY** bit to 1b. This bit will be 1b until the conversion has completed and the ADC has written the digital result into **ADCRES**.
 - The ADC will set the **ADCCTL.ADBUSY** while the conversion is in progress.
 - When the conversion has completed, the ADC will clear the **ADCCTL.ADBUSY** bit and set the **ADCINT.FMAN** bit to 1b indicating that the conversion has completed.

The user may read the result of the ADC conversion from the **ADCRES** register.

22.3.8 Manual ADC Conversions with Repeat

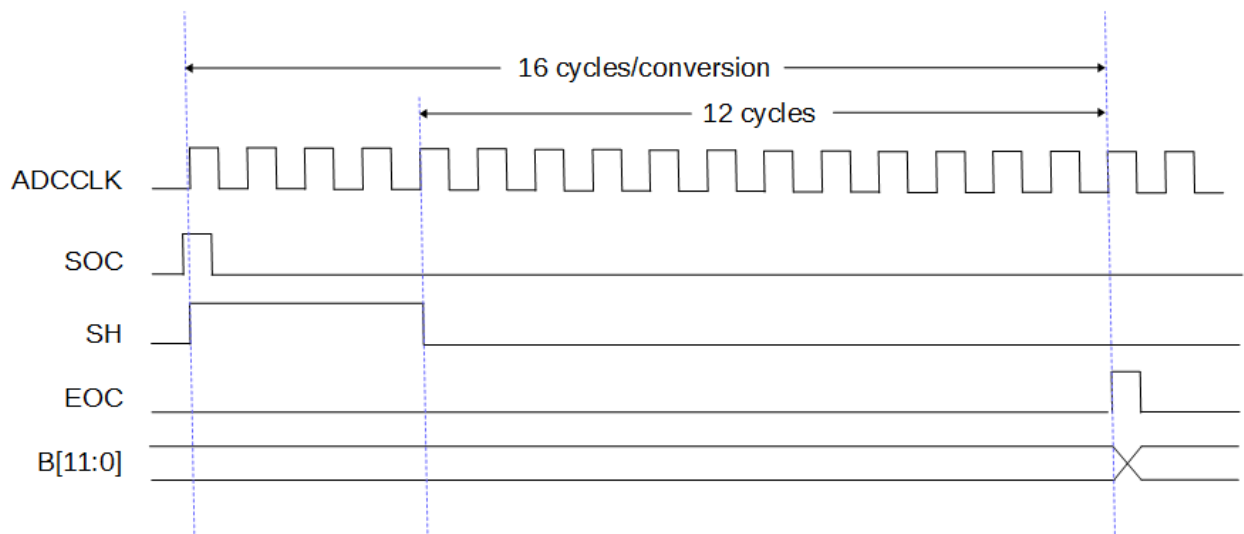
The ADC may continuously repeat ADC conversions in manual mode by setting the **ADCCTL.REPEAT** to a 1b.

When **ADCCTL.REPEAT** is set to a 1b and the **ADCCTL.START** it written to a 1b, the ADC will continue converting the channel in **ADCTL.CHANNEL** until the **ADCCTL.START** bit is cleared.

22.3.9 ADC Conversion Timing

The diagram below shows the timing of a single ADC conversion.

Figure 22-2 ADC Conversion Timing Diagram (single conversion)



The SOC (start of conversion) signal to the ADC is asserted when the **ADCCTL.START** bit is set. The ADC then begins its sample and hold of 4 ADC clock cycles and asserts the SH (sample and hold) signal.

After the sampling time has ended, the ADC begins the conversion process and writing the result to the **ADCRES** register. During this time, the **ADCRES** register's contents are invalid, until the EOC (end of conversion) signal is asserted.

When EOC is asserted, the ADC sets the **ADCCTL.ADBUSY** to 0b and the **ADCINT.INTFMAN** to 1b.

22.3.10 ADC Manual Mode Interrupts

When the ADC is configured for manual mode (**ADCCTL.MODE** = 0b), when the ADC completes a conversion, the **ADCINT.INTFMAN** is set to 1b.

If the **ADCCTL.INTENMAN** is set to 1b and the **ADCINT.INTFMAN** is set to 1b, the ADC will assert the IRQ_ADC0 signal to the NVIC. The **ADCINT.INTFMAN** may be cleared when written to a 1b.

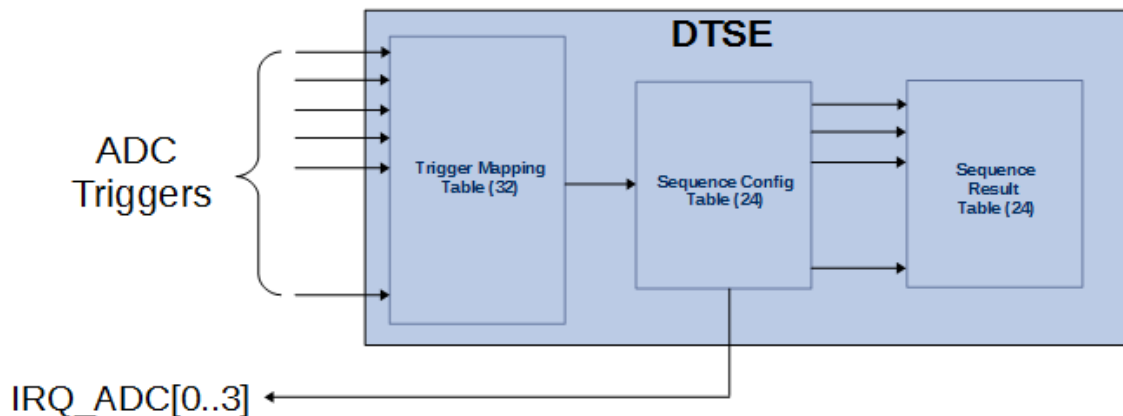
The IRQ_ADC0 signal will be de-asserted when the **ADCINT.INTFMAN** bit is clear and **ADCINT.ADCIRQ0IF** is cleared.

22.4 DTSE

22.4.1 Overview

The Dynamic Triggering and Sampling Engine (DTSE) is a power and highly configurable auto-sequencer that allows application-specific automatic ADC conversions without MCU intervention.

Figure 22-3 DTSE Simplified Block Diagram



There are three main components to the DTSE:

- Trigger Mapping Table
- Sequence Configuration Table
- Sequence Result Table

The DTSE allows the user to specify up to 32 possible input triggers (any PWM output from any timer) to start a series of ADC conversions. The *Trigger Mapping Table* allows the user to configure which of these triggers will trigger a given series of conversions.

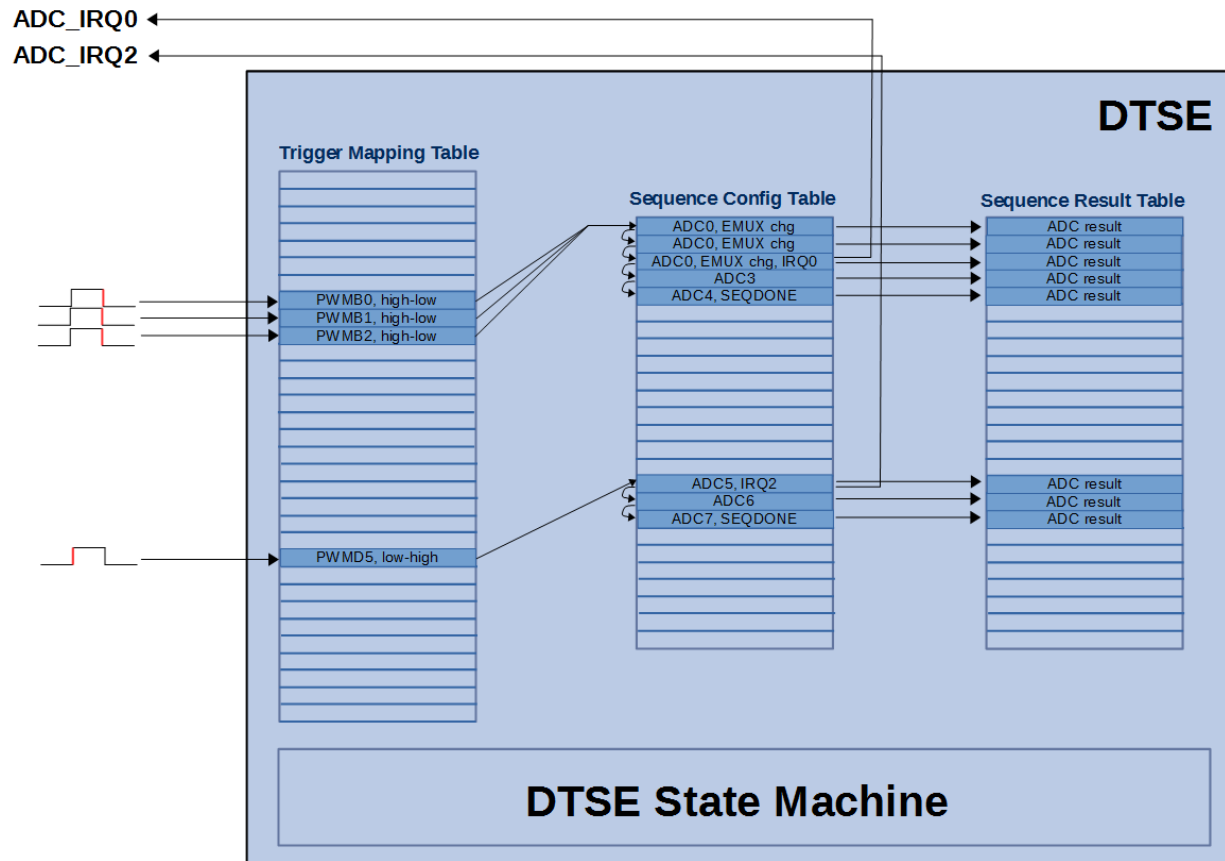
The *Sequence Configuration Table* allows the user to specify a series of conversions that the DTSE will perform automatically, without MCU intervention. This table allows the user to specify information about a single conversion such as the ADC channel, ADC sampling delay, if an interrupt should be asserted, etc.

The *Sequence Result Table* are where the digital ADC conversion results are copied using DMA. There is a dedicated results register for each ADC sequence in the Sequence Configuration Table.

22.4.2 DTSE Conversion Sequence Example

The figure below shows an example of a DTSE conversion sequence.

Figure 22-4 DTSE Conversion Sequence Example



In this example, the *Trigger Mapping Table* is configured for 4 input triggers: a high to low transition on PWMB0, PWMB1 and PWMB2 and a low to high trigger on PWMD5.

There are two series of conversions configured in the *Sequence Configuration Table*, one for PWMB0/PWMB1/PWMB2 triggers and one for PWMD5 triggers. The first series is 5 conversion sequences long, and asserts IRQ_ADC0 after the third conversion. The second series is 3 conversion sequence long and asserts IRQ_ADC2 after the first conversion in the series.

In both cases, the ADC results are copied to the results table after each individual conversion has been completed.

22.4.3 Trigger Mapping Table

The *Trigger Mapping Table* can be configured to hunt for triggers from any of the PWM outputs (or capture inputs) from Timers A, B, C or D. The user may configure the behavior of all of the DTSE triggers using the **DTSETRIGENTxTOy** registers.

Each trigger entry can be configured to look for the following signal transitions by writing the **DTSETRIGENTxTOy.TRIGNEDGE** field.

- None (00b)
- Low to high edge (01b)
- High to low edge (10b)
- Either Low to high or high to low edge (11b)

The user can force a trigger event in software by writing the **DTSETRIGENTxTOy.FORCE** field to a 1b. This will force the DTSE to process the sequence as if the trigger configured in **DTSETRIGENTxTOy.TRIGEDGE** was received.

The table below shows the *Trigger Mapping Table* and how to configure each of the PWM triggers.

Table 22-3 DTSE Trigger Mapping Table

| REGISTER | FIELD | DTSE TRIGGER |
|-------------------|------------|---------------------|
| DTSETRIGENT0TO3 | TRIG0EDGE | Timer A, PWMA0/CCR0 |
| | TRIG1EDGE | Timer A, PWMA1/CCR1 |
| | TRIG2EDGE | Timer A, PWMA2/CCR2 |
| | TRIG3EDGE | Timer A, PWMA3/CCR3 |
| DTSETRIGENT4TO7 | TRIG4EDGE | Timer A, PWMA4/CCR4 |
| | TRIG5EDGE | Timer A, PWMA5/CCR5 |
| | TRIG6EDGE | Timer A, PWMA6/CCR6 |
| | TRIG7EDGE | Timer A, PWMA7/CCR7 |
| DTSETRIGENT8TO11 | TRIG8EDGE | Timer B, PWMB0/CCR0 |
| | TRIG9EDGE | Timer B, PWMB1/CCR1 |
| | TRIG10EDGE | Timer B, PWMB2/CCR2 |
| | TRIG11EDGE | Timer B, PWMB3/CCR3 |
| DTSETRIGENT12TO15 | TRIG12EDGE | Timer B, PWMB4/CCR4 |
| | TRIG13EDGE | Timer B, PWMB5/CCR5 |
| | TRIG14EDGE | Timer B, PWMB6/CCR6 |
| | TRIG15EDGE | Timer B, PWMB7/CCR7 |
| DTSETRIGENT16TO19 | TRIG16EDGE | Timer C, PWMC0/CCR0 |
| | TRIG17EDGE | Timer C, PWMC1/CCR1 |
| | TRIG18EDGE | Timer C, PWMC2/CCR2 |
| | TRIG19EDGE | Timer C, PWMC3/CCR3 |
| DTSETRIGENT20TO23 | TRIG20EDGE | Timer C, PWMC4/CCR4 |
| | TRIG21EDGE | Timer C, PWMC5/CCR5 |

| | | |
|--------------------------|-------------------|---------------------|
| | TRIG22EDGE | Timer C, PWMC6/CCR6 |
| | TRIG23EDGE | Timer C, PWMC7/CCR7 |
| DTSETRIGENT24TO27 | TRIG24EDGE | Timer D, PWMD0/CCR0 |
| | TRIG25EDGE | Timer D, PWMD1/CCR1 |
| | TRIG26EDGE | Timer D, PWMD2/CCR2 |
| | TRIG27EDGE | Timer D, PWMD3/CCR3 |
| DTSETRIGENT28TO31 | TRIG28EDGE | Timer D, PWMD4/CCR4 |
| | TRIG29EDGE | Timer D, PWMD5/CCR5 |
| | TRIG30EDGE | Timer D, PWMD6/CCR6 |
| | TRIG31EDGE | Timer D, PWMD7/CCR7 |

If the DTSE detects an input trigger, then it uses the **DTSETRIGENTxTOY.TRIGNCFGIDX** to access the *Sequence Configuration Table* entry to perform the first conversion.

For example, if the *Trigger Mapping Table* has the following configuration:

- **DTSETRIG8TO11.TRIG10EDGE** = 10b
- **DTSETRIG8TO11.TRIG10CFGIDX** = 101b

If the DTSE detects a high-to-low transition on PWMB2, then the DTSE will use index 5 for the *Sequence Configuration Table* to determine which ADC conversion to perform to start the series.

22.4.4 Sequence Configuration Table

The *Sequence Configuration Table* contains a series of 24 entries. Each entry in this table configures a single ADC conversion. It is possible to chain together up to 24 ADC conversions in single series if all entries in this table are used.

Each entry in the Sequence Configuration Table allows the user to configure the following information:

- ADC Channel to convert
- EMUX Command Control (when to send EMUX command)
- Delay after EMUX command
- EMUX data to send
- IRQ enable
- IRQ number to assert
- Last sequence in the conversion
- No conversion flag

The *Trigger Mapping Table* will point to the first sequence index in this table to convert. After each conversion the DTSE determines if it will perform additional conversions (**DTSESEQCFGn.SEQDONE** = 0b) or if it has completed ADC conversions for the series (**DTSESEQCFGn.SEQDONE** = 1b).

If the DTSE will perform more conversions, then it will advance to the next entry in the *Sequence Configuration Table*.

If the user does not want to perform an ADC conversion (user just wants to send the EMUX data), then the user should set the **DTSSEQCFGn.NOCVT** = 1b.

The table below shows a single entry from the *Sequence Configuration Table* and what the different fields control.

Table 22-4 DTSE Sequence Configuration Table Entry

| FIELD | DESCRIPTION |
|----------------|---|
| IRQEN | IRQ Enabled. If set to a 1b, this will cause the DTSE to assert the IRQ signal to the NVIC according to the setting of the IRQNUM field in this register. |
| IRQNUM | IRQ Number to assert. If the IRQEN field is set to 1b, the DTSE will assert an interrupt to the NVIC according to the contents of this field as shown below. 00b: IRQ_ADC0 01b: IRQ_ADC1 10b: IRQ_ADC2 11b: IRQ_ADC3 |
| EMUXD | EMUX Data to send. If the EMUXC field is set to 01b (send before sample) or 10b (send after conversion), the DTSE will write this field to EMUX according to the time specified in EMUXC . |
| SEQDONE | DTSE sequence done. 0b: Sequence not complete. Advance to the next entry in this table for the next conversion. 1b: Sequence is complete. |
| DELAY | DTSE ADC delay. This is the number of ADCCLK delay cycles that the DTSE will force after the EMUX command is sent, after the sample and hold phase. |
| CHANNEL | ADC Channel. This is the ADC channel to convert for this conversion sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |

| | |
|--------------|--|
| EMUXC | EMUX Control. This field indicates when the DTSE will send the EMUX command for this conversion: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |
| NOCVT | No Conversion. 0b: Perform ADC conversion as configured in this entry 1b: Do not perform ADC conversion (only send EMUX data) |

22.4.5 Result Table

When the DTSE *Sequence Configuration Table* has completed a conversion, the DTSE writes the digital result of the conversion into the **DTSERESn** register and sets the appropriate **ADCINT.INFn** field.

For example, if entry 11 in the *Sequence Configuration Table* has just been converted, the DTSE will write the ADC result into the **DTSERES11** register and set the **ADCINT.INTF11** bit to 1b.

22.4.6 DTSE Conversion State Machine

When the DTSE is enabled and configured, the following steps are executed by the DTSE when performing a series of conversions:

1. The DTSE hunts for triggers that are configured in the *Trigger Mapping Table*
2. When a trigger is received, the DTSE checks the **DTSETRIGENTxTOy.TRIGEDGE** to see if the edge of the trigger is configured.
3. If the trigger is received, the DTSE looks up the entry in the *Sequence Configuration Table* according to the index found in the *Trigger Mapping Table* (**DTSETRIGENTxTOy.TRIGNCFGIDX**).
4. Set **ADCCTL.ADBUSY** to 1b
5. Label: *do_conversion*
 - a. If **DTSESEQCFGn.EMUXC** = 01b (send EMUX command before sampling), send the EMUX data from **DTSESEQCFGn.EMUXD**.
 - b. Set the ADC channel to **DTSESEQCFGn.CHANNEL** start the ADC conversion.
 - c. Wait for sample and hold phase to complete
 - d. Delay for **DTSESEQCFGn.DELAY ADCCLK** periods.
 - e. Wait for ADC conversion to complete
 - f. Write the result of this conversion to the **DTSERESn** register.
 - g. If **DTSESEQCFGn.EMUXC** = 10b (send EMUX command after conversion), send the EMUX data from **DTSESEQCFGn.EMUXD**.
 - h. Set the **ADCINT.INTFn** flag for this sequence entry
 - i. If the **DTSESEQCFGn.IRQEN** is set to 1b, assert the IRQ signal as specified in **DTSESEQCFGn.IRQNUM**.

- j. If **DTSESEQCFGn.SEQDONE** = 0b, advance to next entry in the *Sequence Configuration Table* and goto *do_conversion* above. Otherwise, continue below.
6. Set **ADCCTL.ADBUSY** = 0

22.4.7 Clearing DTSE Interrupts

The **ADCINT** register may be used to clear any of the interrupts (IRQ_ADC0, IRQ_ADC1, IRQ_ADC2 or IRQ_ADC3).

To clear any of these IRQs, clear the interrupt flags as shown below:

- To clear IRQ_ADC0, set **ADCINT.IRQ0IF** to 1b
- To clear IRQ_ADC1, set **ADCINT.IRQ1IF** to 1b
- To clear IRQ_ADC2, set **ADCINT.IRQ2IF** to 1b
- To clear IRQ_ADC3, set **ADCINT.IRQ3IF** to 1b

22.5 Peripheral IO Mapping

The ADC and DTSE ADC inputs are connected by an Analog MUX so they can be used by various IO pins.

The tables below shows which IO pins are available for each ADC channel.

Table 22-5 ADC Peripheral IO Mapping

| TIMER SIGNAL | IO PIN |
|--------------|----------|
| ADC0 | PG7 |
| ADC1 | PD3, PG5 |
| ADC2 | PD2, PG6 |
| ADC3 | PD1 |
| ADC4 | PD0, PF4 |
| ADC5 | PF5 |
| ADC6 | PF6 |
| ADC7 | PF7 |

For more information on how to use the Digital Peripheral MUX to connect peripheral signals to IO, see Digital Peripheral MUX.

22.6 Register Summary

Table 22-6 ADC and DTSE Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|-----------------------------|--------|------------|
| EMUXCTL | 4000 0000h | EMUX Control | RW | 0000 0000h |
| EMUXDATA | 4000 0004h | EMUX Data | RW | 0000 0000h |
| ADCCTL | 4000 0008h | ADC Control | RW | 0000 0000h |
| ADCRES | 4000 000Ch | ADC Result | RW | 0000 0000h |
| ADCINT | 4000 0010h | ADC Interrupt Control | RW | 0000 0000h |
| DTSETRIGENT0TO3 | 4000 0040h | DTSE Trigger Entry 0 to 3 | RW | 0000 0000h |
| DTSETRIGENT4TO7 | 4000 0044h | DTSE Trigger Entry 4 to 7 | RW | 0000 0000h |
| DTSETRIGENT8TO11 | 4000 0048h | DTSE Trigger Entry 8 to 11 | RW | 0000 0000h |
| DTSETRIGENT12TO15 | 4000 004Ch | DTSE Trigger Entry 12 to 15 | RW | 0000 0000h |
| DTSETRIGENT16TO19 | 4000 0050h | DTSE Trigger Entry 16 to 19 | RW | 0000 0000h |
| DTSETRIGENT20TO23 | 4000 0054h | DTSE Trigger Entry 20 to 23 | RW | 0000 0000h |
| DTSETRIGENT24TO27 | 4000 0058h | DTSE Trigger Entry 24 to 27 | RW | 0000 0000h |
| DTSETRIGENT28TO31 | 4000 005Ch | DTSE Trigger Entry 28 to 31 | RW | 0000 0000h |
| DTSESEQCFG0 | 4000 0080h | DTSE Sequence Config 0 | RW | 0000 0000h |
| DTSESEQCFG1 | 4000 0084h | DTSE Sequence Config 1 | RW | 0000 0000h |
| DTSESEQCFG2 | 4000 0088h | DTSE Sequence Config 2 | RW | 0000 0000h |
| DTSESEQCFG3 | 4000 008Ch | DTSE Sequence Config 3 | RW | 0000 0000h |
| DTSESEQCFG4 | 4000 0090h | DTSE Sequence Config 4 | RW | 0000 0000h |
| DTSESEQCFG5 | 4000 0094h | DTSE Sequence Config 5 | RW | 0000 0000h |
| DTSESEQCFG6 | 4000 0098h | DTSE Sequence Config 6 | RW | 0000 0000h |
| DTSESEQCFG7 | 4000 009Ch | DTSE Sequence Config 7 | RW | 0000 0000h |
| DTSESEQCFG8 | 4000 00A0h | DTSE Sequence Config 8 | RW | 0000 0000h |
| DTSESEQCFG9 | 4000 00A4h | DTSE Sequence Config 9 | RW | 0000 0000h |
| DTSESEQCFG10 | 4000 00A8h | DTSE Sequence Config 10 | RW | 0000 0000h |
| DTSESEQCFG11 | 4000 00ACh | DTSE Sequence Config 11 | RW | 0000 0000h |
| DTSESEQCFG12 | 4000 00B0h | DTSE Sequence Config 12 | RW | 0000 0000h |
| DTSESEQCFG13 | 4000 00B4h | DTSE Sequence Config 13 | RW | 0000 0000h |
| DTSESEQCFG14 | 4000 00B8h | DTSE Sequence Config 14 | RW | 0000 0000h |
| DTSESEQCFG15 | 4000 00BCh | DTSE Sequence Config 15 | RW | 0000 0000h |
| DTSESEQCFG16 | 4000 00C0h | DTSE Sequence Config 16 | RW | 0000 0000h |
| DTSESEQCFG17 | 4000 00C4h | DTSE Sequence Config 17 | RW | 0000 0000h |
| DTSESEQCFG18 | 4000 00C8h | DTSE Sequence Config 18 | RW | 0000 0000h |

| | | | | |
|---------------------|------------|-------------------------|----|------------|
| DTSESEQCFG19 | 4000 00CCh | DTSE Sequence Config 19 | RW | 0000 0000h |
| DTSESEQCFG20 | 4000 00D0h | DTSE Sequence Config 20 | RW | 0000 0000h |
| DTSESEQCFG21 | 4000 00D4h | DTSE Sequence Config 21 | RW | 0000 0000h |
| DTSESEQCFG22 | 4000 00D8h | DTSE Sequence Config 22 | RW | 0000 0000h |
| DTSESEQCFG23 | 4000 00DCh | DTSE Sequence Config 23 | RW | 0000 0000h |
| DTSERES0 | 4000 0100h | DTSE Result 0 | RO | 0000 0000h |
| DTSERES1 | 4000 0104h | DTSE Result 1 | RO | 0000 0000h |
| DTSERES2 | 4000 0108h | DTSE Result 2 | RO | 0000 0000h |
| DTSERES3 | 4000 010Ch | DTSE Result 3 | RO | 0000 0000h |
| DTSERES4 | 4000 0110h | DTSE Result 4 | RO | 0000 0000h |
| DTSERES5 | 4000 0114h | DTSE Result 5 | RO | 0000 0000h |
| DTSERES6 | 4000 0118h | DTSE Result 6 | RO | 0000 0000h |
| DTSERES7 | 4000 011Ch | DTSE Result 7 | RO | 0000 0000h |
| DTSERES8 | 4000 0120h | DTSE Result 8 | RO | 0000 0000h |
| DTSERES9 | 4000 0124h | DTSE Result 9 | RO | 0000 0000h |
| DTSERES10 | 4000 0128h | DTSE Result 10 | RO | 0000 0000h |
| DTSERES11 | 4000 012Ch | DTSE Result 11 | RO | 0000 0000h |
| DTSERES12 | 4000 0130h | DTSE Result 12 | RO | 0000 0000h |
| DTSERES13 | 4000 0134h | DTSE Result 13 | RO | 0000 0000h |
| DTSERES14 | 4000 0138h | DTSE Result 14 | RO | 0000 0000h |
| DTSERES15 | 4000 013Ch | DTSE Result 15 | RO | 0000 0000h |
| DTSERES16 | 4000 0140h | DTSE Result 16 | RO | 0000 0000h |
| DTSERES17 | 4000 0144h | DTSE Result 17 | RO | 0000 0000h |
| DTSERES18 | 4000 0148h | DTSE Result 18 | RO | 0000 0000h |
| DTSERES19 | 4000 014Ch | DTSE Result 19 | RO | 0000 0000h |
| DTSERES20 | 4000 0150h | DTSE Result 20 | RO | 0000 0000h |
| DTSERES21 | 4000 0154h | DTSE Result 21 | RO | 0000 0000h |
| DTSERES22 | 4000 0158h | DTSE Result 22 | RO | 0000 0000h |
| DTSERES23 | 4000 015Ch | DTSE Result 23 | RO | 0000 0000h |

22.7 Register Detail

22.7.1 EMUXCTL

Register 22-1 EMUXCTL (EMUX Control, 4000 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:6 | Reserved | RO | 0 | Reserved |
| 5 | EMUXMODE | RW | 0 | EMUX Mode: 0b: Write EMUX data from EMUXDATA register 1b: Write EMUX data from DTSE sequencer commands |
| 4 | EMUXBUSY | RO | 0 | EMUX Busy Status: 0b: EMUX not busy 1b: EMUX busy |
| 3:0 | EMUXDIV | RW | 0 | EMUX Clock Divider: 0000b: PCLK /1 0001b: PCLK /2 0010b: PCLK /3 0011b: PCLK /4 0100b: PCLK /5 0101b: PCLK /6 0110b: PCLK /7 0111b: PCLK /8 1000b: PCLK /9 1001b: PCLK /10 1010b: PCLK /11 1011b: PCLK /12 1100b: PCLK /13 1101b: PCLK /14 1110b: PCLK /15 1111b: PCLK /16 |

22.7.2 EMUXDATA

Register 22-2 EMUXDATA (EMUX Data, 4000 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | EMUXDATA | RW | 0 | When EMUXCTL.EMUXMODE = 0b, the EMUX writes this data onto the EMUX. Writing this data when EMUXCTL.EMUXMODE = 1b has no effect. |

22.7.3 ADCCTL

Register 22-3 ADCCTL (ADC Configuration, 4000 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15 | INTENCOL | RW | 0 | Sequence Collision Interrupt Enable: 0b: not enabled 1b: enabled |
| 14 | INTENEMUX | RW | 0 | EMUX Complete Interrupt Enable: 0b: not enabled 1b: enabled |
| 13 | INTENMAN | RW | 0 | ADC Manual Mode Conversion Complete: 0b: not enabled 1b: enabled |
| 12 | ENABLE | RW | 0 | ADC Enable: 0b: ADC module disabled (PD mode) 1b: ADC module enabled |
| 11 | START | W1C | 0 | ADC Start Conversion: 1b: Start the ADC conversion or conversion sequence This bit is automatically cleared when set. |
| 10 | REPEAT | RW | 0 | Enabled Repeated Conversions: 0b: Disable repeated conversions 1b: Enable repeated conversions |
| 9:8 | MODE | RW | 0 | ADC Mode: 00b: Manual mode 01b: DTSE mode |
| 7 | ADBUSY | RO | 0 | ADC Busy: 0b: ADC not performing conversion 1b: ADC performing conversion |
| 6:4 | CHANNEL | RW | 0 | ADC Channel. When ADCCTL.MODE = 0b (manual mode), this field sets the ADC channel to convert. 000b: ADC0 010b: ADC1 011b: ADC2 100b: ADC3 101b: ADC4 110b: ADC5 111b: ADC6 |
| 3:0 | ADCDIV | RW | 0 | ADC Clock Divider: 0000b: SCLK /1 0001b: SCLK /2 0010b: SCLK /3 0011b: SCLK /4 0100b: SCLK /5 |

| | | | | |
|--|--|--|--|--|
| | | | | 0101b: SCLK /6 0110b: SCLK /7 0111b: SCLK /8 1000b: SCLK /9 1001b: SCLK /10 1010b: SCLK /11 1011b: SCLK /12 1100b: SCLK /13 1101b: SCLK /14 1110b: SCLK /15 1111b: SCLK / 16 |
|--|--|--|--|--|

22.7.4 ADCRES

Register 22-4 ADCRES (ADC Result, 4000 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | When ADCCTL.MODE = 0b (manual mode) and an ADC conversion has completed, the 12-bit digitized results is available here. |

22.7.5 ADCINT

Register 22-5 ADCINT (ADC Interrupt Control, 4000 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-----|-----------|--------|-------|---|
| 31 | ADCIRQ3IF | W1C | 0 | ADCIRQ3 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 30 | ADCIRQ2IF | W1C | 0 | ADCIRQ2 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 29 | ADCIRQ1IF | W1C | 0 | ADCIRQ1 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 28 | ADCIRQ0IF | W1C | 0 | ADCIRQ0 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 27 | Reserved | RO | 0 | Reserved |
| 26 | INTFCOL | W1C | 0 | DTSE Sequence Collision Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 25 | INTFEMUX | W1C | 0 | DTSE EMUX Complete Interrupt Flag- 0b: no interrupt flag 1b: interrupt flag |
| 24 | INTFMAN | W1C | 0 | DTSE Manual Mode Conversion Complete Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 23 | INTF23 | W1C | 0 | DTSE Sequence 23 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 22 | INTF22 | W1C | 0 | DTSE Sequence 22 Interrupt Flag: 0b: no interrupt flag |

| | | | | |
|----|---------------|-----|---|---|
| | | | | 1b: interrupt flag |
| 21 | INTF21 | W1C | 0 | DTSE Sequence 21 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 20 | INTF20 | W1C | 0 | DTSE Sequence 20 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 19 | INTF19 | W1C | 0 | DTSE Sequence 19 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 18 | INTF18 | W1C | 0 | DTSE Sequence 18 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 17 | INTF17 | W1C | 0 | DTSE Sequence 17 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 16 | INTF16 | W1C | 0 | DTSE Sequence 16 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 15 | INTF15 | W1C | 0 | DTSE Sequence 15 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 14 | INTF14 | W1C | 0 | DTSE Sequence 14 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 13 | INTF13 | W1C | 0 | DTSE Sequence 13 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 12 | INTF12 | W1C | 0 | DTSE Sequence 12 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 11 | INTF11 | W1C | 0 | DTSE Sequence 11 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 10 | INTF10 | W1C | 0 | DTSE Sequence 10 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 9 | INTF9 | W1C | 0 | DTSE Sequence 9 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 8 | INTF8 | W1C | 0 | DTSE Sequence 8 Interrupt Flag: |

| | | | | |
|---|--------------|-----|---|--|
| | | | | 0b: no interrupt flag 1b: interrupt flag |
| 7 | INTF7 | W1C | 0 | DTSE Sequence 7 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 6 | INTF6 | W1C | 0 | DTSE Sequence 6 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 5 | INTF5 | W1C | 0 | DTSE Sequence 5 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 4 | INTF4 | W1C | 0 | DTSE Sequence 4 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 3 | INTF3 | W1C | 0 | DTSE Sequence 3 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 2 | INTF2 | W1C | 0 | DTSE Sequence 2 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 1 | INTF1 | W1C | 0 | DTSE Sequence 1 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |
| 0 | INTF0 | W1C | 0 | DTSE Sequence 0 Interrupt Flag: 0b: no interrupt flag 1b: interrupt flag |

22.7.6 DTSETRIGENT0TO3

Register 22-6 DTSETRIGENT0TO3 (DTSE Trigger Entry 0 to 3, 4000 0040h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------------|--------|-------|--|
| 31 | FORCE3 | W1C | 0 | Force Trigger 3. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 3. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG3EDGE | RW | 0 | DTSE Trigger 3 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG3CFGIDX | RW | 0 | DTSE Trigger 3 Sequence Configuration Entry Index. |
| 23 | FORCE2 | W1C | 0 | Force Trigger 2. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 2. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG2EDGE | RW | 0 | DTSE Trigger 2 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG2CFGIDX | RW | 0 | DTSE Trigger 2 Sequence Configuration Entry Index. |
| 15 | FORCE1 | W1C | 0 | Force Trigger 1. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 1. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG1EDGE | RW | 0 | DTSE Trigger 1 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG1CFGIDX | RW | 0 | DTSE Trigger 1 Sequence Configuration Entry Index. |
| 7 | FORCE0 | W1C | 0 | Force Trigger 0. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 0. This bit is cleared when set and writing 0 to this bit has no effect. |

| | | | | |
|-----|--------------------|----|---|--|
| 6:5 | TRIG0EDGE | RW | 0 | DTSE Trigger 0 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG0CFGIDX | RW | 0 | DTSE Trigger 0 Sequence Configuration Entry Index. |

22.7.7 DTSETRIGENT4TO7

Register 22-7 DTSETRIGENT4TO7 (DTSE Trigger Entry 4 to 7, 4000 0044h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|--------------------|--------|-------|--|
| 31 | FORCE7 | W1C | 0 | Force Trigger 7. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 7. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG7EDGE | RW | 0 | DTSE Trigger 7 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG7CFGIDX | RW | 0 | DTSE Trigger 7 Sequence Configuration Entry Index. |
| 23 | FORCE6 | W1C | 0 | Force Trigger 6. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 6. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG6EDGE | RW | 0 | DTSE Trigger 6 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG6CFGIDX | RW | 0 | DTSE Trigger 6 Sequence Configuration Entry Index. |
| 15 | FORCE5 | W1C | 0 | Force Trigger 5. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 5. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG5EDGE | RW | 0 | DTSE Trigger 5 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge |

| | | | | |
|------|--------------------|-----|---|--|
| | | | | 11b: trigger both rising and falling edges |
| 12:8 | TRIG5CFGIDX | RW | 0 | DTSE Trigger 5 Sequence Configuration Entry Index. |
| 7 | FORCE4 | W1C | 0 | Force Trigger 4. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 4. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG4EDGE | RW | 0 | DTSE Trigger 4 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG4CFGIDX | RW | 0 | DTSE Trigger 4 Sequence Configuration Entry Index. |

22.7.8 DTSETRIGENT8TO11

Register 22-8 DTSETRIGENT8TO11 (DTSE Trigger Entry 8 to 11, 4000 0048h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|--|
| 31 | FORCE11 | W1C | 0 | Force Trigger 11. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 11. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG11EDGE | RW | 0 | DTSE Trigger 11 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG11CFGIDX | RW | 0 | DTSE Trigger 11 Sequence Configuration Entry Index. |
| 23 | FORCE10 | W1C | 0 | Force Trigger 10. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 10. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG10EDGE | RW | 0 | DTSE Trigger 10 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG10CFGIDX | RW | 0 | DTSE Trigger 10 Sequence Configuration Entry Index. |

| | | | | |
|-------|--------------------|-----|---|--|
| 15 | FORCE9 | W1C | 0 | Force Trigger 9. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 9. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG9EDGE | RW | 0 | DTSE Trigger 9 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG9CFGIDX | RW | 0 | DTSE Trigger 9 Sequence Configuration Entry Index. |
| 7 | FORCE8 | W1C | 0 | Force Trigger 8. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 8. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG8EDGE | RW | 0 | DTSE Trigger 8 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG8CFGIDX | RW | 0 | DTSE Trigger 8 Sequence Configuration Entry Index. |

22.7.9 DTSETRIGENT12TO15

Register 22-9 DTSETRIGENT12TO15 (DTSE Trigger Entry 12 to 15, 4000 004Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|--|
| 31 | FORCE15 | W1C | 0 | Force Trigger 15. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 15. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG15EDGE | RW | 0 | DTSE Trigger 15 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG15CFGIDX | RW | 0 | DTSE Trigger 15 Sequence Configuration Entry Index. |
| 23 | FORCE14 | W1C | 0 | Force Trigger 14. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 14. This bit is cleared when set and writing 0 to this bit has no effect. |

| | | | | |
|-------|---------------------|-----|---|--|
| | | | | effect. |
| 22:21 | TRIG14EDGE | RW | 0 | DTSE Trigger 14 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG14CFGIDX | RW | 0 | DTSE Trigger 14 Sequence Configuration Entry Index. |
| 15 | FORCE13 | W1C | 0 | Force Trigger 13. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 13. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG13EDGE | RW | 0 | DTSE Trigger 13 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG13CFGIDX | RW | 0 | DTSE Trigger 13 Sequence Configuration Entry Index. |
| 7 | FORCE12 | W1C | 0 | Force Trigger 12. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 12. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG12EDGE | RW | 0 | DTSE Trigger 12 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG12CFGIDX | RW | 0 | DTSE Trigger 12 Sequence Configuration Entry Index. |

22.7.10 DTSETRIGENT16TO19

Register 22-10 DTSETRIGENT16TO19 (DTSE Trigger Entry 16 to 19, 4000 0050h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-------------------|--------|-------|--|
| 31 | FORCE19 | W1C | 0 | Force Trigger 19. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 19. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG19EDGE | RW | 0 | DTSE Trigger 19 Edge Configuration: 00b: unused |

| | | | | |
|-------|---------------------|-----|---|--|
| | | | | 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG19CFGIDX | RW | 0 | DTSE Trigger 19 Sequence Configuration Entry Index. |
| 23 | FORCE18 | W1C | 0 | Force Trigger 18. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 18. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG18EDGE | RW | 0 | DTSE Trigger 18 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG18CFGIDX | RW | 0 | DTSE Trigger 18 Sequence Configuration Entry Index. |
| 15 | FORCE17 | W1C | 0 | Force Trigger 17. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 17. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG17EDGE | RW | 0 | DTSE Trigger 17 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG17CFGIDX | RW | 0 | DTSE Trigger 17 Sequence Configuration Entry Index. |
| 7 | FORCE16 | W1C | 0 | Force Trigger 16. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 16. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG16EDGE | RW | 0 | DTSE Trigger 16 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG16CFGIDX | RW | 0 | DTSE Trigger 16 Sequence Configuration Entry Index. |

22.7.11 DTSETRIGENT20TO23

Register 22-11 DTSETRIGENT20TO23 (DTSE Trigger Entry 20 to 23, 4000 0054h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-----|------|--------|-------|-------------|
|-----|------|--------|-------|-------------|

| | | | | |
|-------|---------------------|-----|---|---|
| 31 | FORCE23 | W1C | 0 | <p>Force Trigger 23.</p> <p>When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 23.</p> <p>This bit is cleared when set and writing 0 to this bit has no effect.</p> |
| 30:29 | TRIG23EDGE | RW | 0 | <p>DTSE Trigger 23 Edge Configuration:</p> <p>00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges</p> |
| 28:24 | TRIG23CFGIDX | RW | 0 | <p>DTSE Trigger 23 Sequence Configuration Entry Index.</p> |
| 23 | FORCE22 | W1C | 0 | <p>Force Trigger 22.</p> <p>When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 22.</p> <p>This bit is cleared when set and writing 0 to this bit has no effect.</p> |
| 22:21 | TRIG22EDGE | RW | 0 | <p>DTSE Trigger 22 Edge Configuration:</p> <p>00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges</p> |
| 20:16 | TRIG22CFGIDX | RW | 0 | <p>DTSE Trigger 22 Sequence Configuration Entry Index.</p> |
| 15 | FORC21 | W1C | 0 | <p>Force Trigger 21.</p> <p>When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 21.</p> <p>This bit is cleared when set and writing 0 to this bit has no effect.</p> |
| 14:13 | TRIG21EDGE | RW | 0 | <p>DTSE Trigger 21 Edge Configuration:</p> <p>00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges</p> |
| 12:8 | TRIG21CFGIDX | RW | 0 | <p>DTSE Trigger 21 Sequence Configuration Entry Index.</p> |
| 7 | FORCE20 | W1C | 0 | <p>Force Trigger 20.</p> <p>When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 20.</p> <p>This bit is cleared when set and writing 0 to this bit has no effect.</p> |
| 6:5 | TRIG20EDGE | RW | 0 | <p>DTSE Trigger 20 Edge Configuration:</p> <p>00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges</p> |

| | | | | |
|-----|---------------------|----|---|---|
| 4:0 | TRIG20CFGIDX | RW | 0 | DTSE Trigger 20 Sequence Configuration Entry Index. |
|-----|---------------------|----|---|---|

22.7.12 DTSETRIGENT24TO27

Register 22-12 DTSETRIGENT24TO27 (DTSE Trigger Entry 24 to 27, 4000 0058h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|--|
| 31 | FORCE27 | W1C | 0 | Force Trigger 27. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 27. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG27EDGE | RW | 0 | DTSE Trigger 27 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG27CFGIDX | RW | 0 | DTSE Trigger 27 Sequence Configuration Entry Index. |
| 23 | FORCE26 | W1C | 0 | Force Trigger 26. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 26. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG26EDGE | RW | 0 | DTSE Trigger 26 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG26CFGIDX | RW | 0 | DTSE Trigger 26 Sequence Configuration Entry Index. |
| 15 | FORCE25 | W1C | 0 | Force Trigger 25. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 25. This bit is cleared when set and writing 0 to this bit has no effect. |
| 14:13 | TRIG25EDGE | RW | 0 | DTSE Trigger 25 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG25CFGIDX | RW | 0 | DTSE Trigger 25 Sequence Configuration Entry Index. |
| 7 | FORCE24 | W1C | 0 | Force Trigger 24. When ADCCTL.MODE = 01b (DTSE mode) and this bit is |

| | | | | |
|-----|---------------------|----|---|---|
| | | | | written to a 1b, it forces DTSE trigger 24. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG24EDGE | RW | 0 | DTSE Trigger 24 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG24CFGIDX | RW | 0 | DTSE Trigger 24 Sequence Configuration Entry Index. |

22.7.13 DTSETRIGENT28TO31

Register 22-13 DTSETRIGENT28TO31 (DTSE Trigger Entry 28 to 31, 4000 005Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------------|--------|-------|--|
| 31 | FORCE31 | W1C | 0 | Force Trigger 31. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 31. This bit is cleared when set and writing 0 to this bit has no effect. |
| 30:29 | TRIG31EDGE | RW | 0 | DTSE Trigger 31 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 28:24 | TRIG31CFGIDX | RW | 0 | DTSE Trigger 31 Sequence Configuration Entry Index. |
| 23 | FORCE30 | W1C | 0 | Force Trigger 30. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 30. This bit is cleared when set and writing 0 to this bit has no effect. |
| 22:21 | TRIG30EDGE | RW | 0 | DTSE Trigger 30 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 20:16 | TRIG30CFGIDX | RW | 0 | DTSE Trigger 30 Sequence Configuration Entry Index. |
| 15 | FORCE29 | W1C | 0 | Force Trigger 29. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 29. This bit is cleared when set and writing 0 to this bit has no effect. |

| | | | | |
|-------|---------------------|-----|---|--|
| 14:13 | TRIG29EDGE | RW | 0 | DTSE Trigger 29 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 12:8 | TRIG29CFGIDX | RW | 0 | DTSE Trigger 29 Sequence Configuration Entry Index. |
| 7 | FORCE28 | W1C | 0 | Force Trigger 28. When ADCCTL.MODE = 01b (DTSE mode) and this bit is written to a 1b, it forces DTSE trigger 28. This bit is cleared when set and writing 0 to this bit has no effect. |
| 6:5 | TRIG28EDGE | RW | 0 | DTSE Trigger 28 Edge Configuration: 00b: unused 01b: trigger rising edge 10b: trigger falling edge 11b: trigger both rising and falling edges |
| 4:0 | TRIG28CFGIDX | RW | 0 | DTSE Trigger 28 Sequence Configuration Entry Index. |

22.7.14 DTSESEQCFG0

Register 22-14 DTSESEQCFG0 (DTSE Sequence Configuration 0, 4000 0080h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG0.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG0.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG0.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG0.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.15 DTSESEQCFG1

Register 22-15 DTSESEQCFG1 (DTSE Sequence Configuration 1, 4000 0084h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG1.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG1.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG1.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG1.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.16 DTSESEQCFG2

Register 22-16 DTSESEQCFG2 (DTSE Sequence Configuration 2, 4000 0088h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG2.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG2.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG2.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG2.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.17 DTSESEQCFG3

Register 22-17 DTSESEQCFG3 (DTSE Sequence Configuration 3, 4000 008Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG3.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG3.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG3.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG3.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.18 DTSESEQCFG4

Register 22-18 DTSESEQCFG4 (DTSE Sequence Configuration 4, 4000 0090h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG4.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG4.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG4.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG4.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.19 DTSESEQCFG5

Register 22-19 DTSESEQCFG5 (DTSE Sequence Configuration 5, 4000 0094h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG5.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG5.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG5.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG5.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.20 DTSESEQCFG6

Register 22-20 DTSESEQCFG6 (DTSE Sequence Configuration 6, 4000 0098h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG6.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG6.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG6.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG6.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.21 DTSESEQCFG7

Register 22-21 DTSESEQCFG7 (DTSE Sequence Configuration 7, 4000 009Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG7.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG7.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG7.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG7.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.22 DTSESEQCFG8

Register 22-22 DTSESEQCFG8 (DTSE Sequence Configuration 8, 4000 00A0h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG8.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG8.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG8.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG8.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.23 DTSESEQCFG9

Register 22-23 DTSESEQCFG9 (DTSE Sequence Configuration 9, 4000 00A4h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG9.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG9.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG9.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG9.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.24 DTSESEQCFG10

Register 22-24 DTSESEQCFG10 (DTSE Sequence Configuration 10, 4000 00A8h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG10.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG10.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG10.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG10.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.25 DTSESEQCFG11

Register 22-25 DTSESEQCFG11 (DTSE Sequence Configuration 11, 4000 00ACh)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG11.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG11.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG11.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG11.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.26 DTSESEQCFG12

Register 22-26 DTSESEQCFG12 (DTSE Sequence Configuration 12, 4000 00B0h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG12.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG12.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG12.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG12.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.27 DTSESEQCFG13

Register 22-27 DTSESEQCFG13 (DTSE Sequence Configuration 13, 4000 00B4h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG13.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG13.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG13.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG13.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.28 DTSESEQCFG14

Register 22-28 DTSESEQCFG14 (DTSE Sequence Configuration 14, 4000 00B8h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG14.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG14.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG14.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG14.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.29 DTSESEQCFG15

Register 22-29 DTSESEQCFG15 (DTSE Sequence Configuration 15, 4000 00BCh)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG15.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG15.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG15.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG15.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.30 DTSESEQCFG16

Register 22-30 DTSESEQCFG16 (DTSE Sequence Configuration 16, 4000 00C0h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG16.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG16.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG16.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG16.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.31 DTSESEQCFG17

Register 22-31 DTSESEQCFG17 (DTSE Sequence Configuration 17, 4000 00C4h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG17.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG17.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG17.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG17.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.32 DTSESEQCFG18

Register 22-32 DTSESEQCFG18 (DTSE Sequence Configuration 18, 4000 00C8h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG8.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG8.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG8.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG8.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.33 DTSESEQCFG19

Register 22-33 DTSESEQCFG19 (DTSE Sequence Configuration 19, 4000 00CCh)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG19.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG19.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG19.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG19.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.34 DTSESEQCFG20

Register 22-34 DTSESEQCFG20 (DTSE Sequence Configuration 20, 4000 00D0h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG20.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG20.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG20.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG20.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.35 DTSESEQCFG21

Register 22-35 DTSESEQCFG21 (DTSE Sequence Configuration 21, 4000 00D4h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG21.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG21.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG21.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG21.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.36 DTSESEQCFG22

Register 22-36 DTSESEQCFG22 (DTSE Sequence Configuration 22, 4000 00D8h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG22.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG22.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG22.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG22.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.37 DTSESEQCFG23

Register 22-37 DTSESEQCFG23 (DTSE Sequence Configuration 23, 4000 00DCh)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:19 | Reserved | RO | 0 | Reserved |
| 18 | IRQEN | RW | 0 | Assert IRQ after converting this sequence. 0b: Do not assert IRQ 1b: Assert IRQ |
| 17:16 | IRQNUM | RW | 0 | IRQ number to assert. If DTSESEQCFG23.IRQEN = 1b, assert this IRQ signal to the NVIC when this sequence has completed conversion: 00b: ADC0_IRQ 01b: ADC1_IRQ 10b: ADC2_IRQ 11b: ADC3_IRQ |
| 15:8 | EMUXD | RW | 0 | EMUX data to send. If DTSESEQCFG23.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this 8-bit data is written to the EMUX. If DTSESEQCFG23.EMUXC = 00b, then EMUX data is not sent. |
| 7 | SEQDONE | RW | 0 | Final sequence of series. Indicates if this sequence is the final sequence for this conversion series. 0b: Not the final sequence, proceed to next sequence entry 1b: Final sequence in this series |
| 6:5 | DELAY | RW | 0 | Conversion Delay. If DTSESEQCFG23.EMUXC = 01b (Send before sample and hold) or 10b (Send after sample and hold), this is the number of ADCCLK delay clocks to add after the EMUX command is sent before the next action is taken. |
| 4:2 | CHANNEL | RW | 0 | ADC Channel. This is the ADC channel to convert during this sequence: 000b: ADC0 001b: ADC1 010b: ADC2 011b: ADC3 100b: ADC4 101b: ADC5 110b: ADC6 111b: ADC7 |
| 1:0 | EMUXC | RW | 0 | EMUX Control. This controls the behavior of the EMUX during this conversion sequence: 00b: Do not send EMUX command 01b: Send EMUX command before sample and hold 10b: Send EMUX command after conversion complete 11b: Reserved |

22.7.38 DTSERES0

Register 22-38 DTSERES0 (DTSE Conversion Result 0, 4000 0100h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 0. |

22.7.39 DTSERES1

Register 22-39 DTSERES1 (DTSE Conversion Result 1, 4000 0104h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 1. |

22.7.40 DTSERES2

Register 22-40 DTSERES2 (DTSE Conversion Result 2, 4000 0108h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 2. |

22.7.41 DTSERES3

Register 22-41 DTSERES3 (DTSE Conversion Result 3, 4000 010Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 3. |

22.7.42 DTSERES4

Register 22-42 DTSERES4 (DTSE Conversion Result 4, 4000 0110h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 4. |

22.7.43 DTSERES5

Register 22-43 DTSERES5 (DTSE Conversion Result 5, 4000 0114h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 5. |

22.7.44 DTSERES6

Register 22-44 DTSERES6 (DTSE Conversion Result 6, 4000 0118h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 6. |

22.7.45 DTSERES7

Register 22-45 DTSERES7 (DTSE Conversion Result 7, 4000 011Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 7. |

22.7.46 DTSERES8

Register 22-46 DTSERES8 (DTSE Conversion Result 8, 4000 0120h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 0. |

22.7.47 DTSERES9

Register 22-47 DTSERES9 (DTSE Conversion Result 9, 4000 0124h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|--|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 9. |

22.7.48 DTSERES10

Register 22-48 DTSERES10 (DTSE Conversion Result 10, 4000 0128h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 10. |

22.7.49 DTSERES11

Register 22-49 DTSERES11 (DTSE Conversion Result 11, 4000 012Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 11. |

22.7.50 DTSERES12

Register 22-50 DTSERES12 (DTSE Conversion Result 12, 4000 0130h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 12. |

22.7.51 DTSERES13

Register 22-51 DTSERES13 (DTSE Conversion Result 13, 4000 0134h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 13. |

22.7.52 DTSERES14

Register 22-52 DTSERES14 (DTSE Conversion Result 14, 4000 0138h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 14. |

22.7.53 DTSERES15

Register 22-53 DTSERES15 (DTSE Conversion Result 15, 4000 013Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|----------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 15. |

22.7.54 DTSERES16

Register 22-54 DTSERES16 (DTSE Conversion Result 16, 4000 1240h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 16. |

22.7.55 DTSERES17

Register 22-55 DTSERES17 (DTSE Conversion Result 17, 4000 0144h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 17. |

22.7.56 DTSERES18

Register 22-56 DTSERES18 (DTSE Conversion Result 18, 4000 0148h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 18. |

22.7.57 DTSERES19

Register 22-57 DTSERES19 (DTSE Conversion Result 19, 4000 014Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 19. |

22.7.58 DTSERES20

Register 22-58 DTSERES20 (DTSE Conversion Result 20, 4000 0150h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 20. |

22.7.59 DTSERES21

Register 22-59 DTSERES21 (DTSE Conversion Result 21, 4000 0154h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 21. |

22.7.60 DTSERES22

Register 22-60 DTSERES22 (DTSE Conversion Result 22, 4000 0158h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 22. |

22.7.61 DTSERES23

Register 22-61 DTSERES23 (DTSE Conversion Result 23, 4000 015Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:12 | Reserved | RO | 0 | Reserved |
| 11:0 | RES | RO | 0 | DTSE conversion result for sequence 23. |

23 CRC

23.1 Overview

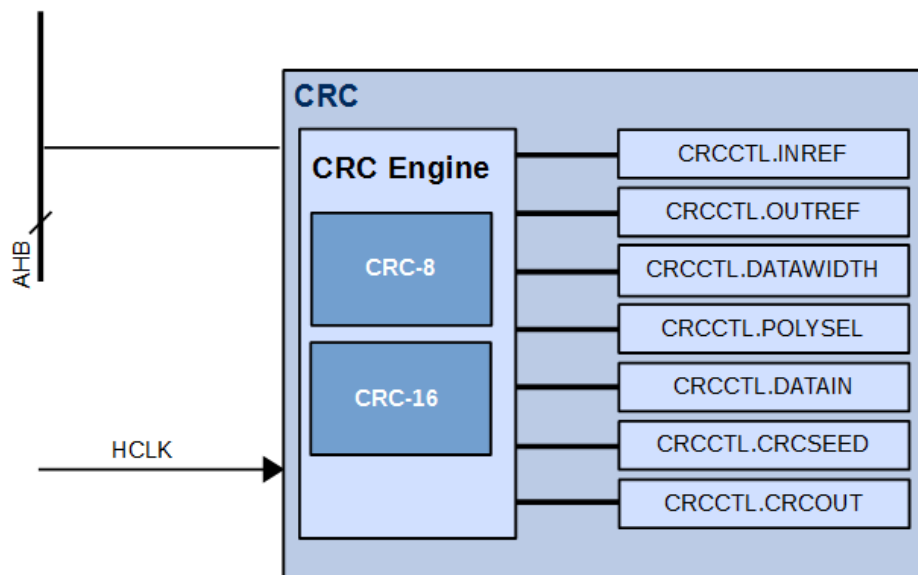
The PAC55XX contains a CRC peripheral that can be used by applications to perform data integrity validation. Examples of these types of applications could be:

- Memory Testing
- IEC60730 Class B Safety
- Communication Protocol CRC

This peripheral allows the user to write data into a CRC accumulator using the AHB bus, and the peripheral can quickly accumulate the CRC based on a selection of standard CRC polynomials.

23.2 Block Diagram

Figure 23-1 CRC Block Diagram



23.3 Features

- Polynomial Selection
 - CRC-16 (2) or CRC-8
- Configurable Data Input Width:
 - 8-bit or 32-bit
- Configurable Data Input Reflection
- Configurable Data Output Reflection
- User Defined Seed Value

23.4 Functional Description

23.4.1 CRC Polynomial

The user may select the CRC polynomial used for the CRC calculation from three choices. The table below shows the available CRC polynomials and the **CRCCTL.POLYSEL** value to choose for a given polynomial.

Table 23-1 CRC Polynomial Selections

| NAME | POLYNOMIAL REPRESENTATION | CRCCTL.POLYSEL | NOTES |
|---------------------------|---------------------------|----------------|--------------------------------------|
| CRC-16-CCITT | 0x1021 | 00b | Also known as CRC-CCITT |
| CRC-16-IBM | 0x8005 | 01b | Also known as CRC-16 and CRC-16-ANSI |
| CRC-8-Dallas/Maxim | 0x31 | 10b | |

The user may set the seed for the CRC calculation by writing it to the **CRCSEED** register.

In order to save firmware instructions and processing time, data may be written into the CRC engine either as 8-bit data or 32-bit data into the **CRCDATAIN** register. To enter data into this register as 32-bit data, set the **CRCCTL.DATAWIDTH** to 0b. To enter data into this register as 8-bit data, set the **CRCCTL.DATAWIDTH** to 1b.

The CRC engine can reflect the input data or output CRC value. To reflect the input data from the **CRCDATIN** register, set the **CRCCTL.INREF** bit to 1b. To reflect the output data to the **CRCOUT** register, set the **CRCCTL.OUTREF** bit to 1b.

The CRC will accumulate into the read-only **CRCOUT** register.

23.5 Register Summary

Table 23-2 CRC Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-------------------|------------|-----------------------|--------|------------|
| CRCCTL | 400D 1000h | CRC Control | RW | 0000 0000h |
| CRCDATAIN | 400D 1004h | CRC Data Input Value | RW | 0000 0000h |
| CRCSEED | 400D 1008h | CRC Seed Value | RW | 0000 0000h |
| CRCDATAOUT | 400D 100Ch | CRC Data Output Value | RO | 0000 0000h |

23.6 Register Detail

23.6.1 CRCCTL

Register 23-1 CRCCTL (CRC Control, 400D 1000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | INREF | RW | 0 | Reflect DATAIN input data to CRC engine: 0b: Do not reflect DATAIN input data 1b: Reflect DATAIN input data |
| 3 | OUTREF | RW | 0 | Reflect DATAOUT output data from CRC engine: 0b: Do not reflect DATAOUT output data 1b: Reflect DATAOUT output data |
| 2 | DATAWIDTH | RW | 0b | DATAIN input data width: 0b: 32-bits – DATAIN [31:0] 1b: 8-bits – DATAIN [7:0] |
| 1:0 | POLYSEL | RW | 00b | CRC Polynomial Select: 00b: CRC-16-CCITT 01b: CRC-16-IBM 10b: CRC-8-Dallas/Maxim 11b: Reserved |

23.6.2 CRCDATAIN

Register 23-2 CRCDATAIN (CRC Data Input, 400D 1004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------|--------|-------|---|
| 31:0 | | RW | 0 | CRC data input value. When CRCCTL.DATAWIDTH is set to 0b, data written to 31:0 used for CRC accumulation. When CRCCTL.DATAWIDTH is set to 1b, data written 7:0 used for CRC accumulation. |

23.6.3 CRCSEED

Register 23-3 CRCSEED (CRC Seed Value, 400D 1008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | CRCSEED | RW | 0 | CRC seed value. When CRCCTL.POLYSEL set to 00b (CRC-16-CCITT) or set to 01b (CRC-16-IBM), seed is located in 15:0. |

| | | | | |
|--|--|--|--|--|
| | | | | When CRCCTL.POLYSEL is set to 10b (CRC-8-Dallas/Maxim), seed is located in 7:0. |
|--|--|--|--|--|

23.6.4 CRCOUT

Register 23-4 CRCOUT (CRC Data Output, 400D 100Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | CRCOUT | RO | 0 | CRC data output value. When CRCCTL.POLYSEL set to 00b (CRC-16-CCITT) or set to 01b (CRC-16-IBM), data output value is located in 15:0. When CRCCTL.POLYSEL is set to 10b (CRC-8-Dallas/Maxim), data output value is located in 7:0. |

24 USART A

24.1 Overview

The PAC55XX family contains support for four Universal Synchronous Asynchronous Receive Transmit (USART) peripherals.

A USART is a serial communications engine that may be configured for UART or SSP through a mode selection register.

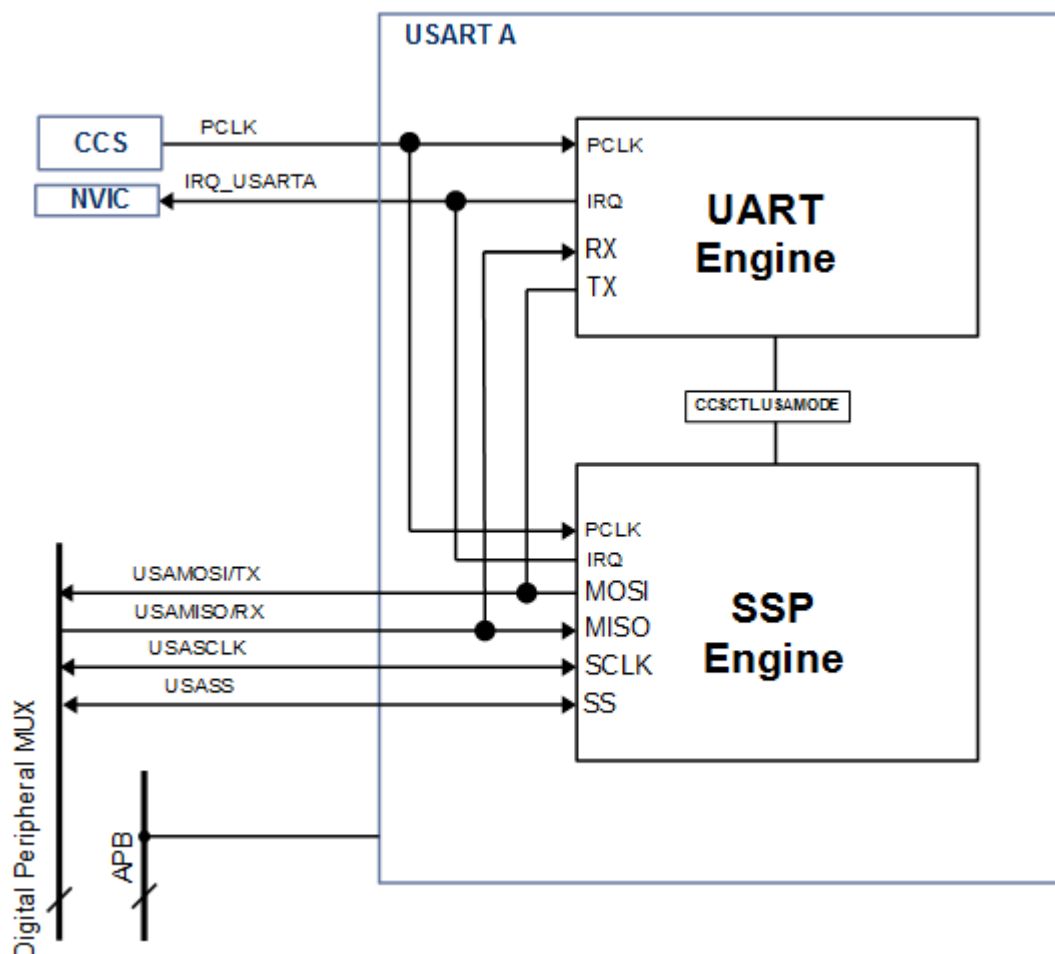
The USART is an APB client and is clocked by the Peripheral Clock (PCLK) system clock.

24.2 Features

- **UART Mode:**
 - 5-bit to 8-bit data interface
 - Optional parity
 - 1, 1.5 or 2 stop bits
 - 16-bit programmable baud-rate generator
 - 8-bit scratch pad
 - Independent RX and TX FIFOs
 - Configurable RX and TX interrupts
- **SSP Mode:**
 - Motorola SPI, TI Synchronous serial or National Semiconductor Microwire support
 - Master and Slave mode support
 - Independent RX and TX FIFOs
 - Configurable clock pre-scaler
 - Programmable Interrupts

24.3 System Block Diagram

Figure 24-1 USART A System Block Diagram



24.4 Functional Description

The USART may be configured for either UART or SSP mode via the **CCCTL.USAMODE** register field.

In UART mode, the peripheral can receive or transmit UART serial data over the USAMOSI/TX (UART TX) or USAMISO/RX serial lines. In this mode, the USART signals for USASCLK and USASS are unused. The UART may assert the IRQ_USARTA0 signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In UART mode, only the UART registers may be accessed, starting at register offset 0.

In SSP mode, the peripheral can receive or transmit serial data over the USAMOSI/TX (SPI MOSI) or USAMISO/RX (SPI MISO) serial lines. The USASCLK and USASS are used by the peripheral in either master or slave mode for the SSP clock and Slave Select signals. The SSP

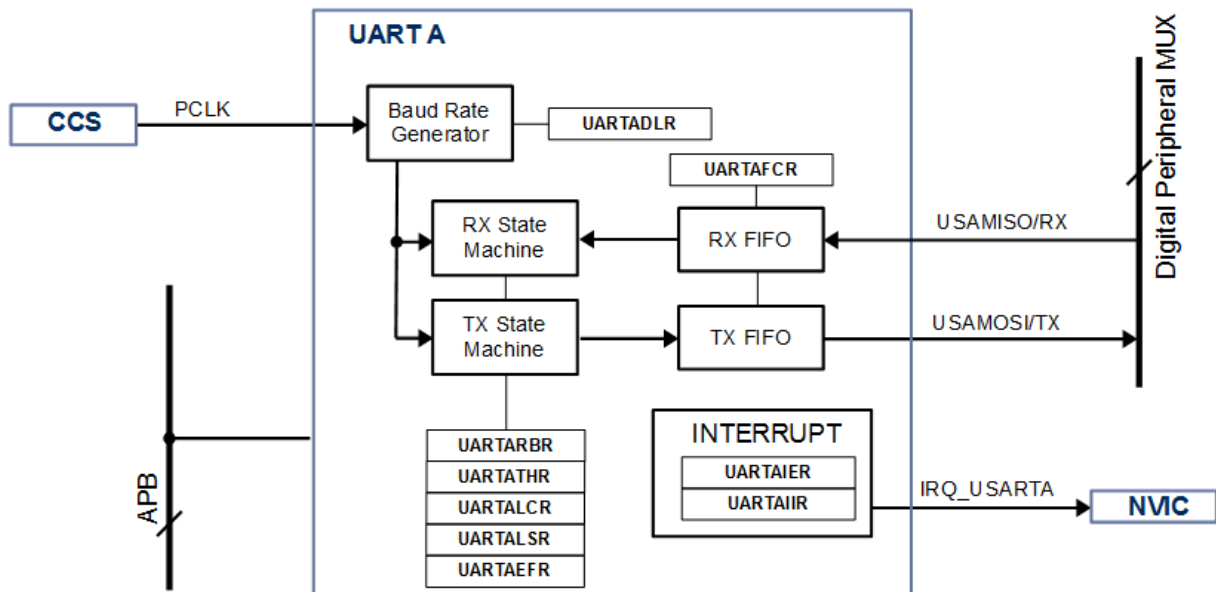
may assert the IRQ_USARTA signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In SPI mode, only the SPI registers may be accessed, starting at register offset 0.

24.5 UART Mode Functional Description

24.5.1 UART Mode System Block Diagram

Figure 24-2 UART A Mode System Block Diagram



24.5.2 Mode Configuration

The UART mode of the USART must first be selected by writing **CCSCTL.USAMODE** to 1b. This will set the USART into UART mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USAMODE** is 1b are below.

Table 24-1 USART A UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|--|
| UARTARBR | UART A Receive Buffer Register |
| UARTATHR | UART A Transmit Holding Register |
| UARTADLR | UART A Divisor Latch Register |
| UARTAIER | UART A Interrupt Enable Register |
| UARTAIIR | UART A Interrupt Identification Register |
| UARTAFCR | UART A FIFO Control Register |
| UARTALCR | UART A Line control Register |
| UARTALSR | UART A Line Status Register |
| UARTASCR | UART A Scratch Pad Register |

24.5.3 Baud Rate Configuration

To configure the baud rate for RX/TX operations, the **UARTADLR** may be used.

To baud rate may be configured by the following formula:

$$\text{Baud Rate} = \text{PCLK} / (16 * \text{UARTADLR})$$

Note that UART peripherals sometimes need an accurate timing base in order to function properly. Because of that, it is suggested that the CLKREF, EXTCLK or XTALCLK be used as the clock source for PCLK when using the UART.

The ROSCCLK may not provide enough accuracy for UART applications.

24.5.4 FIFO Reset

The RX and TX FIFOs may be reset independently by using the **UARTAFC** register.

To change any of the bits in the **UARTAFCR**, the **UARTAFCR.FIFOEN** must be set to 1b (**UARTAFCR** access).

To reset the RX FIFO, write the **UARTAFCR.RXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the RX state machine.

To reset the TX FIFO, write the **UARTAFCR.TXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the TX state machine.

When the **UARTAFCR** register is done being used to configure the FIFOs, set the **UARTAFCR.FIFOEN** to 1b (application mode).

24.5.5 UART Configuration

In UART mode, the UART may be have the parity, break control, stop bits and word length configured by the user.

The parity checking may be enabled or disabled. To enable parity checking, set the **UARTALCR.PEN** to 1b. To disable, set this field to 0b. If enabled, the user may configure the type of parity checking as follows.

Table 24-2 UART A Parity Modes

| UARTALCR.PEN | UARTALCR.PSEL | Parity | Description |
|--------------|---------------|----------------|--|
| 0b | n/a | None | |
| 1b | 00b | Odd | The number of ones in the transmitted character plus the attached parity bit will be odd. |
| | 01b | Even | The number of ones in the transmitted character plus the attached parity bit will be even. |
| | 10b | Forced 1 stick | Force 1 for parity |
| | 11b | Forced 0 stick | Force 0 for parity |

To set the number of stop bits, the user may write the **UARTALCR.SBS** field. To select 1 stop bits, set **UARTALCR.SBS** to 0b. To select 2 stop bits, set **UARTALCR.SBS** to 1b. If the **UARTALCR.WLS** is set to 00b (5-bit character length) and **UARTALCR.SBS** is set to 1b, then there will be 1.5 stop bits.

To configure break control, the user can set the **UARTALCR.BCON** field. To disable break transmission, set **UARTALCR.BCON** to 0b. To force the TX signal to logic 0, set the **UARTALCR.BCON** to 1b.

To set the word length for the UART, the user can set the **UARTALCR.WLS** field as shown below.

Table 24-3 UART A Word Length

| UARTALCR.WLS | UART word length |
|--------------|------------------|
| 00b | 5-bit |
| 01b | 6-bit |
| 10b | 7-bit |
| 11b | 8-bit |

24.5.6 UART Scratch Pad

There is an 8-bit general purpose register that may be used in the USART.

The user may use the **USASCR** register for an 8-bit scratch pad.

24.5.7 UART Interrupts

The status of interrupts may be read at any time by the **UARTAIR** register. If the **UARTAIR.INTSTATUS** field is set to 0b, at least one interrupt is pending. The interrupt type can be read by the **UARTAIR.INTID** field as shown below.

Table 24-4 UART A Interrupt ID

| UARTAIR.INTID | Interrupt Type |
|---------------|---------------------------------|
| 000b | Reserved |
| 001b | TX Holding Register Empty |
| 010b | Receive Data Available |
| 011b | Receive Line Status |
| 100b | Reserved |
| 101b | Reserved |
| 110b | Receive FIFO character time-out |
| 111b | Reserved |

24.5.8 Transmit Operation

Before any data operation, the user must be sure to set the **UARTAFCR.FIFOEN** to 0b (normal mode).

Transmission may be initiated by writing **UARTATHR** with the data desired to be sent. The data will be written into the TX FIFO and then transferred to the TX shift register together with a start bit, parity and stop bits as configured in the **UARTALCR** register, LSB first.

The baud rate used will be set by the **UARTADLR** register.

24.5.9 Transmit Interrupts

The UART may be configured to generate interrupts after characters have been successfully transmitted. The UART may be configured to generate an interrupt after 1, 4, 8 or 14 characters have been transmitted. By default, the interrupt is configured for 1 character.

To set the interrupt threshold for 4, 8 or 14 characters, the user should set the **UARTAEFR.ENMODE** = 1b and set the **UARTAFCR.TXTL** to the interrupt FIFO depth that is desired.

To enable transmit interrupts, set the **UARTAIER.THREIE** (TX holding register interrupt enable). When the transmit holding register has been emptied and the TX FIFO is empty, the **UARTALSR.THRE** will be set to 1b. This field will be cleared the next time the **UARTATHR** register is written with data to transmit.

If the **UARTAIER.THREIE** is set to 1b and the **UARTALSR.THRE** interrupt flag is set, the **IRQ_USARTA** signal to the NVIC will be asserted.

24.5.10 Receive Operation

Before any data operation, the user must be sure to set the **UARTAFCR.FIFOEN** to 0b (normal mode).

Data is sampled into the RX shift register at a sampling rate of $PCLK / 16$. A filter is used to remove spurious inputs that last for less than two periods of the sampling rate. When the complete character has been clocked into the receiver, it is loaded into the RX FIFO. The UART RX FIFO is 16 bytes deep. The first entry into the RX FIFO will be available to read from the **UARTARBR** register.

The receiver also checks the parity and stop bits as specified by the **UARTALCR** register.

24.5.11 Receive Interrupts

The UART may be configured to generate interrupts after a character has been successfully received.

To enable receive interrupts, set the **UARTAIER.RBRIE** (RX buffer register interrupt enable). When the **UARTARBR** (receive buffer register) has been filled from the RX FIFO, the **UARTALSR.RDR** will be set to 1b. This field will be cleared the next time the **UARTARBR** register is read.

If the **UARTAIER.RBRIE** is set to 1b and the **UARTALSR.RDR** interrupt flag is set, the **IRQ_USARTA** signal to the NVIC will be asserted.

24.6 UART Register Summary

Table 24-5 USART A Register Summary (UART Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|--|--------|------------|
| UARTARBR | 4002 0000h | UART A Receive Buffer Register | RO | -- |
| UARTATHR | 4002 0004h | UART A Transmit Holding Register | WO | -- |
| UARTADLR | 4002 0008h | UART A Divisor Latch Register | RW | 0000 0001h |
| UARTAIER | 4002 000Ch | UART A Interrupt Enable Register | RW | 0000 0000h |
| UARTAIIR | 4002 0010h | UART A Interrupt Identification Register | RO | 0000 0001h |
| UARTAFCR | 4002 0014h | UART A FIFO Control Register | RW | 0000 0000h |
| UARTALCR | 4002 0018h | UART A Line control Register | RW | 0000 0000h |
| UARTALSR | 4002 0020h | UART A Line Status Register | RO | 0000 0060h |
| UARTASCR | 4002 0028h | UART A Scratch Pad Register | RW | -- |
| UARTAEFR | 4002 002Ch | UART A Enhanced Mode Register | RW | 0000 000h |

24.7 UART Register Detail

24.7.1 UARTARBR

Register 24-1 UARTARBR (UART A Receive Buffer Register, 4002 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | RBR | RO | 0 | Contains the oldest received character in the UART RX FIFO. |

24.7.2 UARATHR

Register 24-2 UARATHR (UART A Transmit Holding Register, 4002 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | THR | WO | - | Writing to the UARATHR causes the data to be stored in the UART transmit FIFO. The character will be sent when it reaches the bottom of the FIFO and the transmitter is available. |

24.7.3 UARTADLR

Register 24-3 UARTADLR (UART A Divisor Latch Register, 4002 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|------------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | DLR | RW | 0000 0001b | Sets the baud rate for the module. Baud rate = PCLK / (16 * UARTADLR) |

24.7.4 UARAIER

Register 24-4 UARAIER (UART A Interrupt Enable Register, 4002 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:3 | Reserved | RO | 0 | Reserved |
| 2 | RLSIE | RW | 0 | RX Line Status Interrupt Enable. Enables the UART RX line status interrupts. The status of this interrupts can be read from UARTALSR[4:1] . 0b: Disable the RX line status interrupts 1b: Enable the RX line status interrupts |
| 1 | THRIE | RW | 0 | TX Holding Register Empty Interrupt Enable. Enables the THRE interrupt for the UART. The status of this interrupt can be read from UARTALSR.THRE . 0b: Disable the THRE interrupts 1b: Enable the THRE interrupts |
| 0 | RBRIE | RW | 0 | RX Buffer Register Interrupt Enable. Enables the Receive Data Available interrupt for the UART. It also controls the character receive time-out interrupt. 0b: Disable the RBR interrupts 1b: Enable the RBR interrupts |

24.7.5 UARAIIR

Register 24-5 UARAIIR (UART A Interrupt Identification Register, 4002 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3:1 | INTID | RO | 000b | Interrupt identification: 000b: Reserved 001b: TX Holding Register Empty 010b: Receive Data Available 011b: Receive Line Status 100b: Reserved 101b: Reserved 110b: Receive FIFO Character Time-out 111b: Reserved |
| 0 | INTSTATUS | RO | 1b | Interrupt status. Note that this bit is active low. The pending interrupt can be determined through the UARAIIR.INTID field. 0b: At least one interrupt is pending 1b: No interrupt is pending |

Note that this register is cleared on read.

24.7.6 UARTAFCR

Register 24-6 UARTAFCR (UART A FIFO Control Register, 4002 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:6 | RXTL | RW | 0 | RX Trigger Level. These two bits determine how many receive UART FIFO characters must be written before an interrupt is activated. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 5:4 | TXTL | RW | 0 | TX Trigger Level. These two bits determine how may transmit UART FIFO characters must be written before an interrupt is activated. This field cannot be set unless UARTAEFR.ENMODE = 1b. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 3 | Reserved | RO | 0 | Reserved |
| 2 | TXFIFORST | RW | 0 | TX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART TX FIFO. This bit is self-clearing |
| 1 | RXFIFORST | RW | 0 | RX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART RX FIFO. This bit is self-clearing |
| 0 | FIFOEN | RW | 0 | FIFO Enable: 0b: UART FIFOs are disabled. Reset UARTAFCR settings to their default values. 1b: UART RX and TX FIFOs are enabled. UARTAFCR[7:1] are accessible when this bit is set. Any transition on this bit will automatically clear the UART RX AND TX FIFOs. |

24.7.7 UARTALCR

Register 24-7 UARTALCR (UART A Line Control Register, 4002 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6 | BCON | RW | 0 | Break Control: 0b: Disable break transmission 1b: Enable break transmission. Output pin UARTATXD is forced to logic 0 when this bit is set. |
| 5:4 | PSEL | RW | 0 | Parity Select: 00b: Odd parity. The number of 1s in the transmitted character and the attached parity will be odd. 01b: Even parity. The number of 1s in the transmitted character and the attached parity will be even. 10b: Forced 1 stick parity. 11b: Forced 0 stick parity. |
| 3 | PEN | RW | 0 | Parity Enable: 0b: Disable parity generation and checking 1b: Enable parity generation and checking |
| 2 | SBS | RW | 0 | Stop Bit Select: 0b: 1 stop bit 1b: 2 stop bits, 1.5 stop bits if (UARTALCR.WLS = 00b) |
| 1:0 | WLS | RW | 0 | Word Length Select: 00b: 5-bit character length 01b: 6-bit character length 10b: 7-bit character length 11b: 8-bit character length |

24.7.8 UARTALSR

Register 24-8 UARTALSR (UART A Line Status Register, 4002 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | RXFE | RO | 0 | <p>Error in RX FIFO.</p> <p>This bit is set when a character with a RX error such as framing, parity or break is loaded into UARTARBR. This bit is cleared when the UARTALSR register is read, and there are no subsequent errors in the UART FIFO.</p> |
| 6 | TEMT | RO | 0 | <p>Transmitter Empty.</p> <p>This bit is set when both UARTATHR and are empty; this bit is cleared when either the THR contain valid data.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 5 | THRE | RO | 0 | <p>Transmitter Holding Register Empty.</p> <p>This bit is set immediately upon detection of an empty UART THR and is cleared on a THR write.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 4 | BI | RO | 0 | <p>Break Interrupt.</p> <p>When UARTARX is held in the spacing state (all zeroes) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until UARTARX goes to the marking statue (all ones). A read of the UARTALSR register clears this status bit. The time of break detection is dependents on UARTAFCR.FIFOEN.</p> <p>The break interrupt is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Break interrupt status is inactive 1b: Break interrupt status is active</p> |
| 3 | FE | RO | 0 | <p>Framing Error.</p> <p>When the stop bit of a received character is a logic 0, a framing error occurs. Reading the UARTALSR register will clear this bit. The time of the framing error is dependent upon UARTAFCR[3]. Upon detection of a framing error, the RX will attempt to re-synchronize the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no framing error.</p> <p>A framing error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Framing error status is inactive 1b: Framing error status is active</p> |
| 2 | PE | RO | 0 | <p>Parity Error.</p> <p>When the parity bit of a received character is in the wrong</p> |

| | | | | |
|---|------------|----|---|---|
| | | | | <p>state, a parity error occurs. Reading the UARTALSR register will clear this bit. The time of the parity error detection is dependent on UARTAFCR.FIFOEN.</p> <p>A parity error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Parity error status is inactive 1b: Parity error status is active</p> |
| 1 | OE | RO | 0 | <p>Overrun Error.</p> <p>The overrun error condition is set as soon as it occurs. Reading the UARTALSR register will clear this bit. This bit is set when the UART RSR has a new character assembled and the UART RBR FIFO is full. In this case, the UART RBR FIFO will not be overwritten and the character in the UART RSR will be lost.</p> <p>0b: Overrun error status is inactive 1b: Overrun error status is active</p> |
| 0 | RDR | RO | 0 | <p>Receiver Data Ready.</p> <p>This bit is set when the RBR holds an unread character and is cleared when the UART RBR FIFO is empty.</p> <p>0b: RBR is empty 1b: RBR contains valid data</p> |

24.7.9 UARTASCR

Register 24-9 UARTASCR (UART A Scratch Pad Register, 4002 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|-------------------------------|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | PAD | RW | - | A read-able, write-able byte. |

24.7.10 UARTAEFR

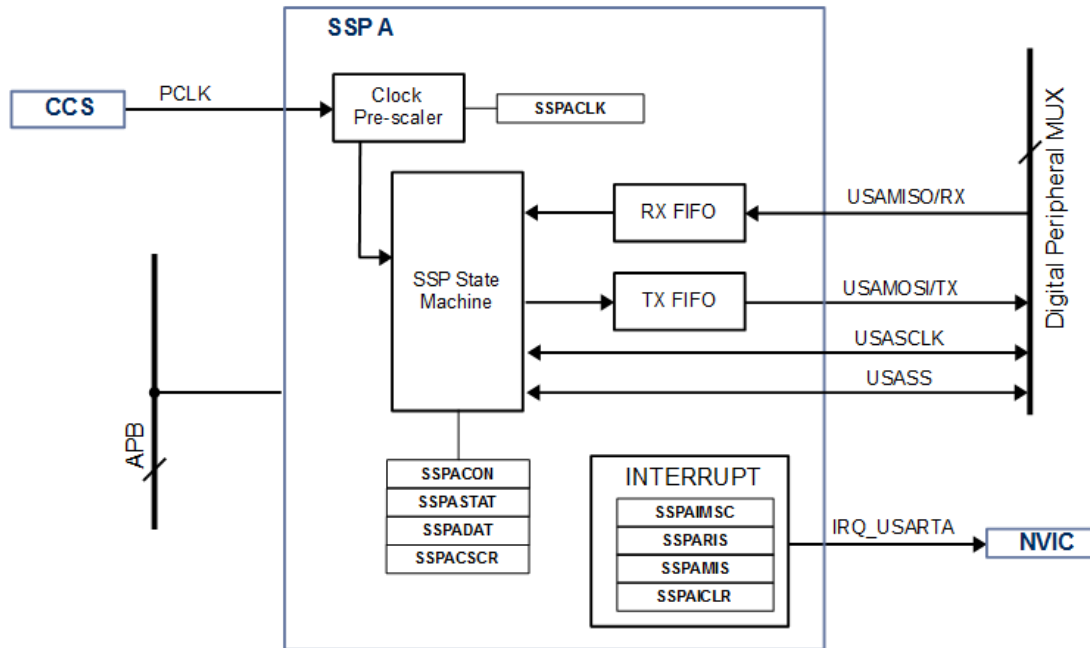
Register 24-10 UARTAEFR (UART A Enhanced Feature Register, 4002 002Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | ENMODE | RW | 0 | Enhanced Mode 0b – disabled 1b – enabled |
| 3:0 | Reserved | RO | 0 | Reserved |

24.8 SSP Functional Description

24.8.1 SSP Mode System Block Diagram

Figure 24-3 SSP A System Block Diagram



24.8.2 Mode Configuration

The SSP mode of the USART must first be selected by writing **CCSCTL.USAMODE** to 0b. This will set the USART into SSP mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USAMODE** is 0b are below.

Table 24-6 USART A UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|---|
| SSPACON | SSP A Control Register |
| SSPASTAT | SSP A Status Register |
| SSPADAT | SSP A Data Register |
| SSPACLK | SSP A Clock Control Register |
| SSPAIMSC | SSP A Interrupt Mask Set and Clear Register |
| SSPARIS | SSP A Raw Interrupt Status Register |
| SSPAMIS | SSP A Masked Interrupt Status Register |
| SSPAICLR | SSP A Interrupt Clear Register |
| SSPASSCR | SSP A Slave Select Configuration Register |

24.8.3 SSP Overview

When in SSP mode, the USART may be a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- Motorola-style SPI
- TI-style Synchronous Serial Interface
- National Semiconductor-style Microwire

In both master and slave modes, the SSP performs parallel to serial conversion on data into 32-bit wide, 8-location deep RX and TX FIFOs.

The SSP may be configured to generate interrupts for servicing the TX and RX FIFOs and error conditions such as FIFO overwrite and timeout.

24.8.4 Clock Configuration

When configured for master mode, the SSP peripheral may configure the clock pre-scaler to generate the desired SCLK output clock frequency. The input clock on the SSP peripheral is the PCLK (peripheral clock) system clock.

The SCLK is generated from the PCLK input and the **SSPACLK.M** and **SSPACLK.N** parameters. The SCLK clock frequency is generated according to the following formula:

$$F_{\text{SSPCLK}} = \text{PCLK} / ((\text{SSPACLK.M} + 1) * \text{SSPACLK.N})$$

In this formula, **SSPACLK.N** must be an even value from 2 to 254.

24.8.5 SSP Clock Constraints

The SSP clock configuration must be configured according to the guidelines below, in order for the peripheral to operate correctly.

Here are the clock constraints for the input clock SSPCLK (after the pre-scaler) and output SSP clock, USACLK:

- When in master mode, the minimum frequency of the input SSPCLK must be at least 2X the output SSP clock (USASCLK)
- When in slave mode, the minimum frequency of the input SSPCLK must be at least 12X the input SSP clock (USASCLK)

The maximum frequency of the SSPCLK should be:

- When in master mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USASCLK}}$ (SSP clock output)
- When in slave mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USASCLK}}$ (SSP clock input)

24.8.6 SSP Configuration

The SSP may be configured to support various features of master and slave based serial interfaces.

To enable the SSP controller, set **SSPACON.SSPEN** to 1b. When disabled, the SSP may still have registers configured for the configuration mode and interrupts, but will not operate on serial data until enabled. Once the SSP has been configured, set this bit to 1b to begin processing.

To configure the SSP mode, set **SSPACON.FRFB** (frame format) to the desired value as shown below.

Table 24-7 SSP A Frame Format

| SSPACON.FRFB | Frame Format |
|--------------|------------------------------------|
| 00b | SPI (Motorola) |
| 01b | Synchronous Serial Format (TI) |
| 10b | Microwire (National Semiconductor) |
| 11b | Reserved |

To select master mode, set **SSPACON.MS** to 0b. In this mode, the controller is the bus master. It will drive the SCLK, MOSI and SS signals and will receive data on the MISO line.

To select slave mode, set **SSPACON.MS** to 1b. In this mode, the controller is the bus slave and will drive the MISO line and receive input from SCLK, MOSI and SS.

To change the serial to parallel endian order, the user may use the **SSPACON.LSBFIRST** field. To configure the SSP for LSB first, set **SSPACON.LSBFIRST** to 1b. To configure the SSP for MSB first, set **SSPACON.LSBFIRST** to a 0b.

To enable loop-back mode, set the **SSPACON.LBM** to 1b. In this mode, serial input is taken from the serial output instead of from the digital peripheral MUX.

While in slave mode, the behavior of the MISO output can be controlled. When **SSPACON.SOD** is set to 0b, the SSP can drive the MISO in slave mode. When this bit is 1b, the SSP will not drive the MISO output in slave mode.

The clock phase of the SSP may also be configured. To configure the SSP for capturing data on the first clock edge transition, set **SSPACON.CPH** to 0b. To configure the SSP for capturing data on the second clock edge transition, set **SSPACON.CPH** to 1b.

When the **SSPACON.FRFB** is 00b (SPI), the clock output polarity may be configured. To configure the SSP clock to be active high, set **SSPACON.CPO** to 0b. To configure the SSP clock to be active low, set **SSPACON.CPO** to 1b.

The SSP may also configure the size of the data word size transferred in each frame. To configure the data size, see the table below.

Table 24-8 SSP A Data Size Select

| SSPACON.DSS | Data Size |
|-------------|-----------|
| 0000b | Reserved |
| 0001b | Reserved |
| 0010b | Reserved |
| 0011b | 4-bit |
| 0100b | 5-bit |
| 0101b | 6-bit |
| 0110b | 7-bit |
| 0111b | 8-bit |
| 1000b | 9-bit |
| 1001b | 10-bit |
| 1010b | 11-bit |
| 1011b | 12-bit |
| 1100b | 13-bit |
| 1101b | 14-bit |
| 1110b | 15-bit |
| 1111b | 16-bit |

24.8.7 SSP Slave Select Configuration

The SSP allows the behavior of the SS (slave select) signal to be configured by using the **SSPASSCR** register as follows when configured as a bus master.

The **SSPASSCR.SELCS** field must always be set to a 00b for proper operation.

To configure the SSP to automatically control the behavior of the SS signal, set **SSPASSCR.SWSEL** to 0b. If the user wants to control the SS signal by software, set **SSPASSCR.SWSEL** to 1b.

To configure the SS signal behavior after a frame transfer, use the **SSPASSCR.SPHDONTCARE** field. To configure the SSP to not pull the SS signal high after a frame transfer, set **SSPASSCR.SPHDONTCARE** to 0b. To configure the SSP to pull the SS signal high after a frame transfer, set the **SSPASSCR.SPHDONTCARE** to 1b.

24.8.8 SSP Interrupts

The SSP may configure interrupts for the following conditions.

When the SSP detects that the TX FIFO is half-full or less, the **SSPARIS.TXRIS** and **SSPAMIS.TXMIS** are set to 1b. During this condition, if the **SSPAIMSC.TXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTA signal to the NVIC.

When the SSP detects that the RX FIFO is half-full or more, the **SSPARIS.RXRIS** and **SSPAMIS.RXMIS** are set to 1b. During this condition, if the **SSPAIMSC.RXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTA signal to the NVIC.

The SSP contains a time for reading the RX FIFO. If the RX FIFO is not empty, and $32 F_{SSPCLK}$ periods have gone by, the **SSPARIS.RTIM** and **SSPAMIS.RTMIS** bits are set to 1b. During this condition, if the **SSPAIMSC.RTIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTA signal to the NVIC. This condition may be cleared by writing **SSPAICLR.RTIC** to 1b.

If the SSP detects the RX FIFO is full, when a character is attempted to be inserted into it, it sets the **SSPARIS.RORIM** and **SSPAMIS.ROMIS** to 1b. During this condition, if the **SSPAIMSC.RORIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTA signal to the NVIC. This condition may be cleared by writing **SSPAICLR.RORIC** to 1b.

24.8.9 Master Mode Operation

When configured as a bus master (**SSPACON.MS** = 0b), the SSP will insert a character into the TX FIFO when it is written into the **SSPADAT** register. The SSP will read a character from the TX FIFO and performs a parallel to serial conversion on it. Then the serial data stream and frame control signal is synchronized to the clock and are output through the USAMOSI/TX pin to the attached slaves. While the data is being transmit to the slaves, the USASCLK and USASS behave as configured in the **SSPACON** and **SSPACSCR** registers.

The master receive logic performs serial to parallel conversion on the incoming synchronous USAMISO/RX data stream, and stores the character in the RX FIFO. The master may read first entry in the RX FIFO by reading the **SSPADAT** register.

24.8.10 Slave Mode Operation

When configured as a bus slave (**SSPACON.MS** = 1b), the SCLK is provided by the attached master. The user can write the next character to send into **SSPADAT** and it will get inserted into the TX FIFO. The slave transmit logic reads a value from the TX FIFO, performs parallel to serial conversion and outputs the serial data stream on the USAMISO/RX pin to the attached master. The slave receive logic performs serial to parallel conversion on the incoming USAMOSI/TX data stream, extracting and storing values into the RX FIFO. The first character in the RX FIFO may be read by reading the **SSPADAT** register.

24.8.11 SSP Status

The SSP module maintains status information on the operation of the module. The table below shows the available SSP status conditions.

Table 24-9 SSP A Status

| SSPASTAT field | Description |
|----------------|--|
| BSY | If set, the SSP controller is currently sending/receiving a frame and/or the TX FIFO is not empty. |
| RFF | If set, the RX FIFO is full. |
| RNE | If set, the RX FIFO is not empty. |
| TNF | If set, the TX FIFO is full. |
| TNE | If set, the TX FIFO is not empty. |

24.9 SSP Register Summary

Table 24-10 USART A Register Summary (SSP Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------------|------------|---|--------|------------|
| SSPACON | 4002 0044h | SSP A Control Register | RW | 0000 0000h |
| SSPASTAT | 4002 0048h | SSP A Status Register | RO | 0000 0003h |
| SSPADAT | 4002 0008h | SSP A Data Register | RW | 0000 0000h |
| SSPACLK | 4002 000Ch | SSP A Clock Control Register | RW | 0000 0000h |
| SSPAIMSC | 4002 0010h | SSP A Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPARIS | 4002 0014h | SSP A Raw Interrupt Status Register | RO | 0000 0008h |
| SSPAMIS | 4002 0018h | SSP A Masked Interrupt Status Register | RO | 0000 0000h |
| SSPAICLR | 4002 001Ch | SSP A Interrupt Clear Register | RW | 0000 0000h |
| SSPASSCR | 4002 0028h | SSP A Slave Select Configuration Register | RW | 0000 0000h |

24.10 SSP Register Detail

24.10.1 SSPACON

Register 24-11 SSPACON (SSP A Control Register, 4002 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | LSBFIRST | RW | 0 | Endian Order: 0b: MSB transmit / receive first 1b: LSB transmit / receive first |
| 12 | LBM | RW | 0 | Loopback Mode: 0b: Normal operation 1b: Loopback mode. Serial Input is taken from the serial output (MOSI or MISO) rather than the serial in (MISO or MOSI respectively) |
| 11 | SSPEN | RW | 0 | SSP Enable: 0b: SSP Controller is disabled 1b: SSP Controller is enabled. It will interact with other devices on the serial bus. Software should write the appropriate control information to the other SPI/SSP registers and interrupt controller registers, before setting this bit. |
| 10 | MS | RW | 0 | Master/Slave Mode: 0b: The SSP controller acts as a bus master, driving the USASCLK, USAMOSI and USSS signals 1b: The SSP controller acts as a bus slave, driving the USAMISO and receiving USASCLK, USAMOSI and USASS. |
| 9 | SOD | RW | 0 | Slave Output Disable: 0b: The SSP can drive the USAMISO output in slave mode 1b: The SSP must not drive the USAMISO output in slave mode |
| 8 | CPH | RW | 0 | Clock Out Phase. This bit is only used when SSPACON.FRF = 0b (SPI). 0b: The SSP controller captures serial data on the first edge transition of the frame. 1b: The SSP controller captures serial data on the second edge transition of the frame. |
| 7 | CPO | RW | 0 | Clock Out Polarity (this bit is only used in SPI mode, when SSPACON.FRF = 00b): 0b: The SSP clock is active high. 1b: The SSP clock is active low. |
| 6:5 | FRF | RW | 0 | Frame Format: 00b: SPI 01b: TI 10b: Microwire 11b: Reserved |
| 4:0 | DSS | RW | 0 | Data Size Select: |

| | | | | |
|--|--|--|--|--|
| | | | | 0 0000b: Reserved 0 0001b: Reserved 0 0010b: Reserved 0 0011b: 4-bit transfer 0 0100b: 5-bit transfer 0 0101b: 6-bit transfer 0 0110b: 7-bit transfer 0 0111b: 8-bit transfer 0 1000b: 9-bit transfer 0 1001b: 10-bit transfer 0 1010b: 11-bit transfer 0 1011b: 12-bit transfer 0 1100b: 13-bit transfer 0 1101b: 14-bit transfer 0 1110b: 15-bit transfer 0 1111b: 16-bit transfer ... 1 1111b: 32-bit transfer |
|--|--|--|--|--|

24.10.2 SSPASTAT

Register 24-12 SSPASTAT (SSP A Status Register, 4002 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | BSY | RO | 0 | Busy Bit: 0b: SPI controller is idle 1b: SPI controller is sending/receiving a frame and/or the TX FIFO is not empty |
| 3 | RFF | RO | 0 | Receive FIFO Full: 0b: RX FIFO not full 1b: RX FIFO full |
| 2 | RNE | RO | 0 | Receive FIFO not empty: 0b: RX FIFO empty 1b: RX FIFO not empty |
| 1 | TNF | RO | 0 | Transmit FIFO not full: 0b: TX FIFO full 1b: TX FIFO not full |
| 0 | TFE | RO | 0 | Transmit FIFO empty: 0b: TX FIFO not empty 1b: TX FIFO empty |

24.10.3 SSPADAT

Register 24-13 SSPADAT (SSP A Data Register, 4002 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-------------|--------|-------|---|
| 31:0 | DATA | RW | 0 | <p>Software can writing data to be sent in a future frame to this register when the SSPASTAT.TNF bit is set to 1b, indicating that the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise, the data written to this register will be sent as soon as all previous data has been sent (and received). If the data length is less than 32-bits, the data must be right-justified in this register.</p> <p>Software may read data from this register when the SSPASTAT.RNE bit is set to a 1b, indicating that the RX FIFO is not empty. When software reads this register, the SPI controller returns the data from the least recent frame in the RX FIFO. If the data length is less than 32-bits, the data is right-justified in this field with the MSBs set to 0.</p> |

24.10.4 SSPACKL

Register 24-14 SSPACKL (SSP A Clock Register, 4002 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:8 | M | RW | 0 | These fields may set the SPI master clock rate by the formula: $F_{SSPCLK} = PCLK / ((SSPACLK.M + 1) * SSPACLK.N)$ N must be an even value from 2 to 254. |
| 7:0 | N | RW | 0 | |

24.10.5 SSPAIMSC

Register 24-15 SSPAIMSC (SSP A Interrupt Mask Set and Clear Enable Register, 4002 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXIM | RW | 0 | Transmit FIFO Interrupt Mask: 0b: TX FIFO half-empty or less condition interrupt disabled 1b: TX FIFO half-empty or less condition interrupt enabled |
| 2 | RXIM | RW | 0 | Receive FIFO Interrupt Mask: 0b: RX FIFO half-full or more condition interrupt disabled 1b: RX FIFO half-full or more condition interrupt enabled |
| 1 | RTIM | RW | 0 | Receive Timeout Interrupt Mask: 0b: RX FIFO not empty and no read prior to timeout period interrupt disabled 1b: RX FIFO not empty and no read prior to timeout period interrupt enabled |
| 0 | RORIM | RW | 0 | Receive Overrun Interrupt Mask: 0b: RX FIFO written to while full condition interrupt disabled 1b: RX FIFO written to while full condition interrupt enabled |

24.10.6 SSPARIS

Register 24-16 SSPARIS (SSP A Raw Interrupt Status Register, 4002 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXRIS | RO | 1 | TX FIFO Raw Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXRIS | RO | 0 | RX FIFO Raw Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTRIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | RORIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

24.10.7 SSPAMIS

Register 24-17 SSPAMIS (SSP A Masked Interrupt Status Register, 4002 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXMIS | RO | 1 | TX FIFO Masked Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXMIS | RO | 0 | RX FIFO Masked Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTMIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | ROMIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

24.10.8 SSPAICLR

Register 24-18 SSPAICLR (SSP A Interrupt Clear Register, 4002 001Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:2 | Reserved | RO | 0 | Reserved |
| 1 | RTIC | RW | 0 | RX FIFO Timeout Interrupt Clear: 0b: No effect 1b: Clears the SSPAXIS.RTRIS interrupt flag This bit is self-clearing. |
| 0 | RORIC | RW | 0 | RX FIFO Overwrite Interrupt Clear: 0b: No effect 1b: Clears the SSPAXIS.RORIS interrupt flag This bit is self-clearing. |

24.10.9 SSPASSCR

Register 24-19 SSPASSCR (SSP A Slave Select Configuration Register, 4002 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|--------------------|--------|-------|---|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | SPHDONTCARE | RW | 0 | Slave Select Pull High: 0b: USASS cannot pull high after frame transfer 1b: USASS must pull high after frame transfer |
| 3 | SWSS | RW | 0 | Slave Select State: 0b: Set USASS to low 1b: Set USASS to high |
| 2 | SWSEL | RW | 0 | Slave Select Software Control: 0b: USASS is automatically controlled by the SPI module 1b: USASS is software controlled by SSPASSCR.SWCS |
| 1:0 | SELSS | RW | 0 | Slave Select Signal Control: 00b: USASS is enabled 01b: Reserved 10b: Reserved 11b: Reserved |

25 USART B

25.1 Overview

The PAC55XX family contains support for four Universal Synchronous Asynchronous Receive Transmit (USART) peripherals.

A USART is a serial communications engine that may be configured for UART or SSP through a mode selection register.

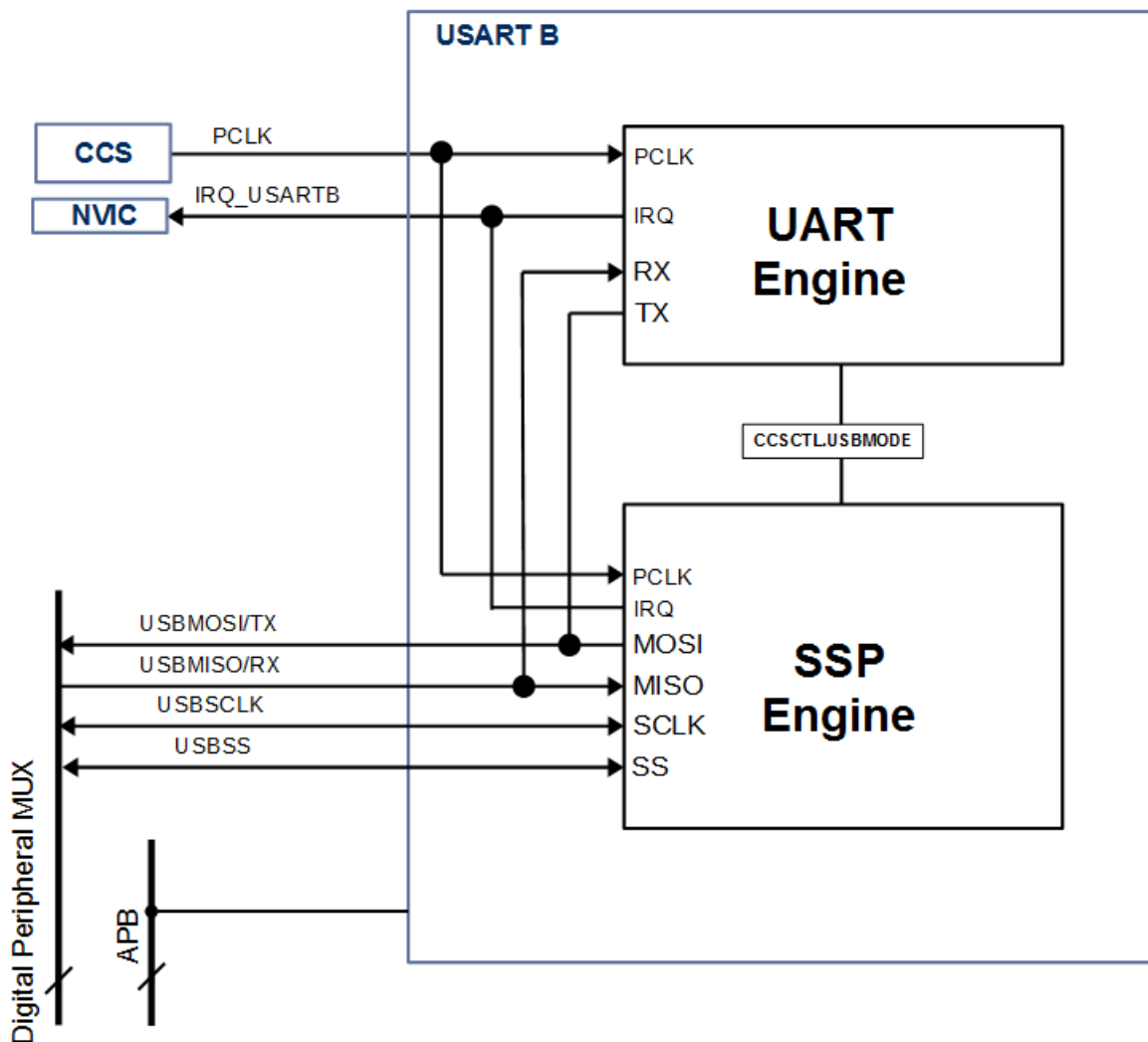
The USART is an APB client and is clocked by the Peripheral Clock (PCLK) system clock.

25.2 Features

- **UART Mode:**
 - 5-bit to 8-bit data interface
 - Optional parity
 - 1, 1.5 or 2 stop bits
 - 16-bit programmable baud-rate generator
 - 8-bit scratch pad
 - Independent RX and TX FIFOs
 - Configurable RX and TX interrupts
- **SSP Mode:**
 - Motorola SPI, TI Synchronous serial or National Semiconductor Microwire support
 - Master and Slave mode support
 - Independent RX and TX FIFOs
 - Configurable clock pre-scaler
 - Programmable Interrupts

25.3 System Block Diagram

Figure 25-1 USART B System Block Diagram



25.4 Functional Description

The USART may be configured for either UART or SSP mode via the **CCSCTL.USBMODE** register.

In UART mode, the peripheral can receive or transmit UART serial data over the USBMOSI/TX (UART TX) or USBMISO/RX serial lines. In this mode, the USART signals for USBSS and USBCLK are unused. The UART may assert the IRQ_USARTB signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In UART mode, only the UART registers may be accessed, starting at register offset 0.

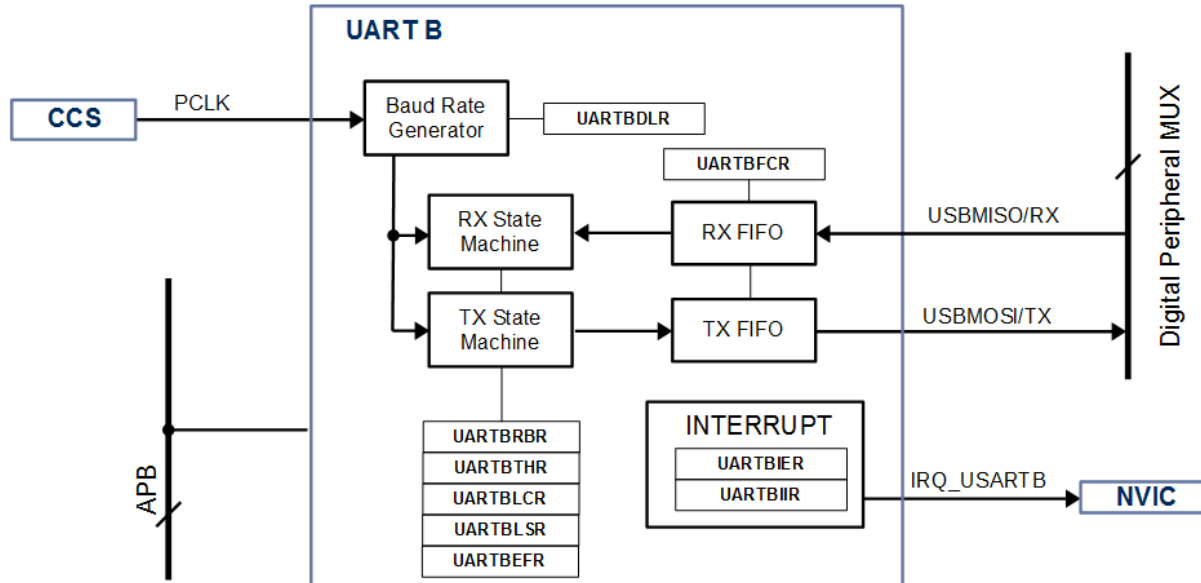
In SSP mode, the peripheral can receive or transmit serial data over the USBMOSI/TX (SPI MOSI) or USBMISO/RX (SPI MISO) serial lines. The USBSCLK and USBSS are used by the peripheral in either master or slave mode for the SSP clock and Slave Select signals. The SSP may assert the IRQ_USARTB signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In SPI mode, only the SPI registers may be accessed, starting at register offset 0.

25.5 UART Mode Functional Description

25.5.1 UART B Mode System Block Diagram

Figure 25-2 UART B Mode System Block Diagram



25.5.2 Mode Configuration

The UART mode of the USART must first be selected by writing **CCSCTL.USBMODE** to 1b. This will set the USART into UART mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USBMODE** is 1b are below.

Table 25-1 USART B UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|--|
| UARTBRBR | UART B Receive Buffer Register |
| UARTBTHR | UART B Transmit Holding Register |
| UARTBDLR | UART B Divisor Latch Register |
| UARTBIER | UART B Interrupt Enable Register |
| UARTBIIR | UART B Interrupt Identification Register |
| UARTBFCR | UART B FIFO Control Register |
| UARTBLCR | UART B Line control Register |
| UARTBLSR | UART B Line Status Register |
| UARTBSCR | UART B Scratch Pad Register |

25.5.3 Baud Rate Configuration

To configure the baud rate for RX/TX operations, the **UARTBDLR** may be used.

To baud rate may be configured by the following formula:

$$\text{Baud Rate} = \text{PCLK} / (16 * \text{UARTBDLR})$$

Note that UART peripherals sometimes need an accurate timing base in order to function properly. Because of that, it is suggested that the CLKREF, EXTCLK or XTALCLK be used as the clock source for PCLK when using the UART.

The ROSCCLK may not provide enough accuracy for UART applications.

25.5.4 FIFO Reset

The RX and TX FIFOs may be reset independently by using the **UARTBFC** register.

To change any of the bits in the **UARTBFCR**, the **UARTBFCR.FIFOEN** must be set to 1b (**UARTBFCR** access).

To reset the RX FIFO, write the **UARTBFCR.RXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the RX state machine.

To reset the TX FIFO, write the **UARTBFCR.TXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the TX state machine.

When the **UARTBFCR** register is done being used to configure the FIFOs, set the **UARTBFCR.FIFOEN** to 1b (application mode).

25.5.5 UART Configuration

In UART mode, the UART may be have the parity, break control, stop bits and word length configured by the user.

The parity checking may be enabled or disabled. To enable parity checking, set the **UARTBLCR.PEN** to 1b. To disable, set this field to 0b. If enabled, the user may configure the type of parity checking as follows.

Table 25-2 UART A Parity Modes

| UARTBLCR.PEN | UARTBLCR.PSEL | Parity | Description |
|--------------|---------------|----------------|--|
| 0b | n/a | None | |
| 1b | 00b | Odd | The number of ones in the transmitted character plus the attached parity bit will be odd. |
| | 01b | Even | The number of ones in the transmitted character plus the attached parity bit will be even. |
| | 10b | Forced 1 stick | Force 1 for parity |
| | 11b | Forced 0 stick | Force 0 for parity |

To set the number of stop bits, the user may write the **UARTBLCR.SBS** field. To select 1 stop bits, set **UARTBLCR.SBS** to 0b. To select 2 stop bits, set **UARTBLCR.SBS** to 1b. If the **UARTBLCR.WLS** is set to 00b (5-bit character length) and **UARTBLCR.SBS** is set to 1b, then there will be 1.5 stop bits.

To configure break control, the user can set the **UARTBLCR.BCON** field. To disable break transmission, set **UARTBLCR.BCON** to 0b. To force the TX signal to logic 0, set the **UARTBLCR.BCON** to 1b.

To set the word length for the UART, the user can set the **UARTBLCR.WLS** field as shown below.

Table 25-3 UART B Word Length

| UARTBLCR.WLS | UART word length |
|--------------|------------------|
| 00b | 5-bit |
| 01b | 6-bit |
| 10b | 7-bit |
| 11b | 8-bit |

25.5.6 UART Scratch Pad

There is an 8-bit general purpose register that may be used in the USART.

The user may use the **USBSCR** register for an 8-bit scratch pad.

25.5.7 UART Interrupts

The status of interrupts may be read at any time by the **UARTBIIR** register. If the **UARTBIIR.INTSTATUS** field is set to 0b, at least one interrupt is pending. The interrupt type can be read by the **UARTBIIR.INTID** field as shown below.

Table 25-4 UART B Interrupt ID

| UARTBIIR.INTID | Interrupt Type |
|----------------|---------------------------------|
| 000b | Reserved |
| 001b | TX Holding Register Empty |
| 010b | Receive Data Available |
| 011b | Receive Line Status |
| 100b | Reserved |
| 101b | Reserved |
| 110b | Receive FIFO character time-out |
| 111b | Reserved |

25.5.8 Transmit Operation

Before any data operation, the user must be sure to set the **UARTBFCCR.FIFOEN** to 0b (normal mode).

Transmission may be initiated by writing **UARTBTHR** with the data desired to be sent. The data will be written into the TX FIFO and then transferred to the TX shift register together with a start bit, parity and stop bits as configured in the **UARTBLCR** register, LSB first.

The baud rate used will be set by the **UARTBDLR** register.

25.5.9 Transmit Interrupts

The UART may be configured to generate interrupts after characters have been successfully transmitted. The UART may be configured to generate an interrupt after 1, 4, 8 or 14 characters have been transmitted. By default, the interrupt is configured for 1 character.

To set the interrupt threshold for 4, 8 or 14 characters, the user should set the **UARTBEFR.ENMODE** = 1b and set the **UARTBFCCR.TXTL** to the interrupt FIFO depth that is desired.

To enable transmit interrupts, set the **UARTBIER.THREIE** (TX holding register interrupt enable). When the transmit holding register has been emptied and the TX FIFO is empty, the **UARTBLSR.THRE** will be set to 1b. This field will be cleared the next time the **UARTBTHR** register is written with data to transmit.

If the **UARTBIER.THREIE** is set to 1b and the **UARTBLSR.THRE** interrupt flag is set, the **IRQ_USARTB** signal to the NVIC will be asserted.

25.5.10 Receive Operation

Before any data operation, the user must be sure to set the **UARTBFCR.FIFOEN** to 0b (normal mode).

Data is sampled into the RX shift register at a sampling rate of $PCLK / 16$. A filter is used to remove spurious inputs that last for less than two periods of the sampling rate. When the complete character has been clocked into the receiver, it is loaded into the RX FIFO. The UART RX FIFO is 16 bytes deep. The first entry into the RX FIFO will be available to read from the **UARTBRBR** register.

The receiver also checks the parity and stop bits as specified by the **UARTBLCR** register.

25.5.11 Receive Interrupts

The UART may be configured to generate interrupts after a character has been successfully received.

To enable receive interrupts, set the **UARTBIER.RBRIE** (RX buffer register interrupt enable). When the **UARTBRBR** (receive buffer register) has been filled from the RX FIFO, the **UARTBLSR.RDR** will be set to 1b. This field will be cleared the next time the **UARTBRBR** register is read.

If the **UARTBIER.RBRIE** is set to 1b and the **UARTBLSR.RDR** interrupt flag is set, the **IRQ_USARTB** signal to the NVIC will be asserted.

25.6 UART Register Summary

Table 25-5 USART B Register Summary (UART Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|--|--------|------------|
| UARTARBR | 4003 0000h | UART A Receive Buffer Register | RO | -- |
| UARTATHR | 4003 0004h | UART A Transmit Holding Register | WO | -- |
| UARTADLR | 4003 0008h | UART A Divisor Latch Register | RW | 0000 0001h |
| UARTAIER | 4003 000Ch | UART A Interrupt Enable Register | RW | 0000 0000h |
| UARTAIIR | 4003 0010h | UART A Interrupt Identification Register | RO | 0000 0001h |
| UARTAFCR | 4003 0014h | UART A FIFO Control Register | RW | 0000 0000h |
| UARTALCR | 4003 0018h | UART A Line control Register | RW | 0000 0000h |
| UARTALSR | 4003 0020h | UART A Line Status Register | RO | 0000 0060h |
| UARTASCR | 4003 0028h | UART A Scratch Pad Register | RW | -- |
| UARTAEFR | 4002 002Ch | UART A Enhanced Mode Register | RW | 0000 000h |

25.7 UART Register Detail

25.7.1 UARTBRBR

Register 25-1 UARTBRBR (UART B Receive Buffer Register, 4003 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | RBR | RO | 0 | Contains the oldest received character in the UART RX FIFO. |

25.7.2 UARBTHR

Register 25-2 UARBTHR (UART B Transmit Holding Register, 4003 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | THR | WO | - | Writing to the UARBTHR causes the data to be stored in the UART transmit FIFO. The character will be sent when it reaches the bottom of the FIFO and the transmitter is available. |

25.7.3 UARTBDLR

Register 25-3 UARTBDLR (UART B Divisor Latch Register, 4003 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|------------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | DLR | RW | 0000 0001b | Sets the baud rate for the module. Baud rate = PCLK / (16 * UARTBDLR) |

25.7.4 UARTBIER

Register 25-4 UARTBIER (UART B Interrupt Enable Register, 4003 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:3 | Reserved | RO | 0 | Reserved |
| 2 | RLSIE | RW | 0 | RX Line Status Interrupt Enable. Enables the UART RX line status interrupts. The status of this interrupts can be read from UARTBLSR[4:1] . 0b: Disable the RX line status interrupts 1b: Enable the RX line status interrupts |
| 1 | THRIE | RW | 0 | TX Holding Register Empty Interrupt Enable. Enables the THRE interrupt for the UART. The status of this interrupt can be read from UARTBLSR.THRE . 0b: Disable the THRE interrupts 1b: Enable the THRE interrupts |
| 0 | RBRIE | RW | 0 | RX Buffer Register Interrupt Enable. Enables the Receive Data Available interrupt for the UART. It also controls the character receive time-out interrupt. 0b: Disable the RBR interrupts 1b: Enable the RBR interrupts |

25.7.5 UARTBIIR

Register 25-5 UARTBIIR (UART B Interrupt Identification Register, 4003 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3:1 | INTID | RO | 000b | Interrupt identification: 000b: Reserved 001b: TX Holding Register Empty 010b: Receive Data Available 011b: Receive Line Status 100b: Reserved 101b: Reserved 110b: Receive FIFO Character Time-out 111b: Reserved |
| 0 | INTSTATUS | RO | 1b | Interrupt status. Note that this bit is active low. The pending interrupt can be determined through the UARTBIIR.INTID field. 0b: At least one interrupt is pending 1b: No interrupt is pending |

Note that this register is cleared on read.

25.7.6 UARTBFCR

Register 25-6 UARTBFCR (UART B FIFO Control Register, 4003 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:6 | RXTL | RW | 0 | RX Trigger Level. These two bits determine how many receive UART FIFO characters must be written before an interrupt is activated. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 5:4 | TXTL | RW | 0 | TX Trigger Level. These two bits determine how may transmit UART FIFO characters must be written before an interrupt is activated. This field cannot be set unless UARTBEFR.ENMODE = 1b. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 3 | Reserved | RO | 0 | Reserved |
| 2 | TXFIFORST | RW | 0 | TX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART TX FIFO. This bit is self-clearing |
| 1 | RXFIFORST | RW | 0 | RX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART RX FIFO. This bit is self-clearing |
| 0 | FIFOEN | RW | 0 | FIFO Enable: 0b: UART FIFOs are disabled. Reset UARTAFCR settings to their default values. 1b: UART RX and TX FIFOs are enabled. UARTBFCR[7:1] are accessible when this bit is set. Any transition on this bit will automatically clear the UART RX AND TX FIFOs. |

25.7.7 UARTBLCR

Register 25-7 UARTBLCR (UART B Line Control Register, 4003 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6 | BCON | RW | 0 | Break Control: 0b: Disable break transmission 1b: Enable break transmission. Output pin UARTBTXD is forced to logic 0 when this bit is set. |
| 5:4 | PSEL | RW | 0 | Parity Select: 00b: Odd parity. The number of 1s in the transmitted character and the attached parity will be odd. 01b: Even parity. The number of 1s in the transmitted character and the attached parity will be even. 10b: Forced 1 stick parity. 11b: Forced 0 stick parity. |
| 3 | PEN | RW | 0 | Parity Enable: 0b: Disable parity generation and checking 1b: Enable parity generation and checking |
| 2 | SBS | RW | 0 | Stop Bit Select: 0b: 1 stop bit 1b: 2 stop bits, 1.5 stop bits if (UARTBLCR.WLS = 00b) |
| 1:0 | WLS | RW | 0 | Word Length Select: 00b: 5-bit character length 01b: 6-bit character length 10b: 7-bit character length 11b: 8-bit character length |

25.7.8 UARTBLSR

Register 25-8 UARTBLSR (UART B Line Status Register, 4003 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | RXFE | RO | 0 | <p>Error in RX FIFO.</p> <p>This bit is set when a character with a RX error such as framing, parity or break is loaded into UARTBRBR. This bit is cleared when the UARTBLSR register is read, and there are no subsequent errors in the UART FIFO.</p> |
| 6 | TEMT | RO | 0 | <p>Transmitter Empty.</p> <p>This bit is set when both UARTBTHR is empty; this bit is cleared when either the THR contains valid data.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 5 | THRE | RO | 0 | <p>Transmitter Holding Register Empty.</p> <p>This bit is set immediately upon detection of an empty UART THR and is cleared on a THR write.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 4 | BI | RO | 0 | <p>Break Interrupt.</p> <p>When UARTBRX is held in the spacing state (all zeroes) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until UARTBRX goes to the marking statue (all ones). A read of the UARTBLSR register clears this status bit. The time of break detection is dependents on UARTBFGR.FIFOEN.</p> <p>The break interrupt is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Break interrupt status is inactive 1b: Break interrupt status is active</p> |
| 3 | FE | RO | 0 | <p>Framing Error.</p> <p>When the stop bit of a received character is logic 0, a framing error occurs. Reading the UARTBLSR register will clear this bit. The time of the framing error is dependent upon UARTBFGR[3]. Upon detection of a framing error, the RX will attempt to re-synchronize the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no framing error.</p> <p>A framing error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Framing error status is inactive 1b: Framing error status is active</p> |
| 2 | PE | RO | 0 | <p>Parity Error.</p> <p>When the parity bit of a received character is in the wrong</p> |

| | | | | |
|---|------------|----|---|---|
| | | | | <p>state, a parity error occurs. Reading the UARTBLSR register will clear this bit. The time of the parity error detection is dependent on UARTBFCR.FIFOEN.</p> <p>A parity error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Parity error status is inactive 1b: Parity error status is active</p> |
| 1 | OE | RO | 0 | <p>Overrun Error.</p> <p>The overrun error condition is set as soon as it occurs. Reading the UARTBLSR register will clear this bit. This bit is set when the UART RSR has a new character assembled and the UART RBR FIFO is full. In this case, the UART RBR FIFO will not be overwritten and the character in the UART RSR will be lost.</p> <p>0b: Overrun error status is inactive 1b: Overrun error status is active</p> |
| 0 | RDR | RO | 0 | <p>Receiver Data Ready.</p> <p>This bit is set when the RBR holds an unread character and is cleared when the UART RBR FIFO is empty.</p> <p>0b: RBR is empty 1b: RBR contains valid data</p> |

25.7.9 UARTBSCR

Register 25-9 UARTBSCR (UART B Scratch Pad Register, 4003 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|-------------------------------|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | PAD | RW | - | A read-able, write-able byte. |

25.7.10 UARTBEFR

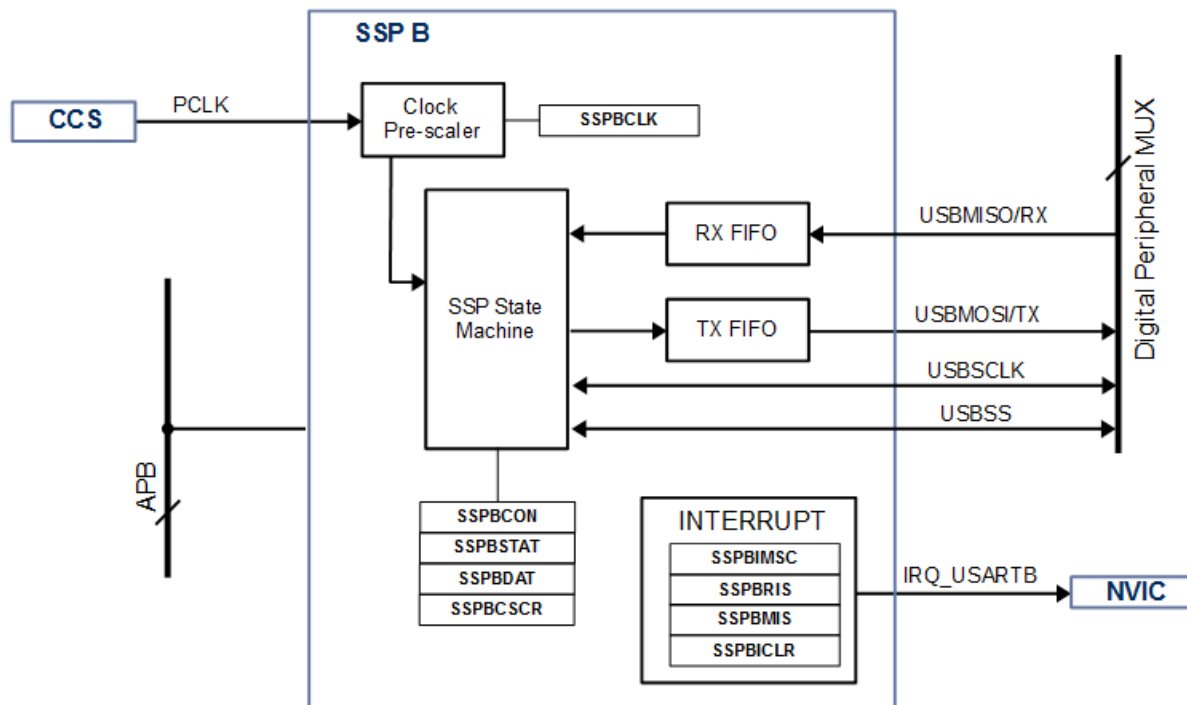
Register 25-10 UARTBEFR (UART B Enhanced Feature Register, 4003 002Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | ENMODE | RW | 0 | Enhanced Mode 0b – disabled 1b – enabled |
| 3:0 | Reserved | RO | 0 | Reserved |

25.8 SSP Functional Description

25.8.1 SSP Mode System Block Diagram

Figure 25-3 SSP B System Block Diagram



25.8.2 Mode Configuration

The SSP mode of the USART must first be selected by writing **CCSCTL.USBMODE** to 0b. This will set the USART into SSP mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USBMODE** is 0b are below.

Table 25-6 USART B UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|---|
| SSPBCON | SSP B Control Register |
| SSPBSTAT | SSP B Status Register |
| SSPBDAT | SSP B Data Register |
| SSPBCLK | SSP B Clock Control Register |
| SSPBIMSC | SSP B Interrupt Mask Set and Clear Register |
| SSPBRIS | SSP B Raw Interrupt Status Register |
| SSPBMIS | SSP B Masked Interrupt Status Register |
| SSPBICLR | SSP B Interrupt Clear Register |

| | |
|-----------------|---|
| SSPBSSCR | SSP B Slave Select Configuration Register |
|-----------------|---|

25.8.3 SSP Overview

When in SSP mode, the USART may be a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- Motorola-style SPI
- TI-style Synchronous Serial Interface
- National Semiconductor-style Microwire

In both master and slave modes, the SSP performs parallel to serial conversion on data into 32-bit wide, 8-location deep RX and TX FIFOs.

The SSP may be configured to generate interrupts for servicing the TX and RX FIFOs and error conditions such as FIFO overwrite and timeout.

25.8.4 Clock Configuration

When configured for master mode, the SSP peripheral may configure the clock pre-scaler to generate the desired SCLK output clock frequency. The input clock on the SSP peripheral is the PCLK (peripheral clock) system clock.

The SCLK is generated from the PCLK input and the **SSPBCLK.M** and **SSPBCLK.N** parameters. The SCLK clock frequency is generated according to the following formula:

$$F_{SSPCLK} = PCLK / ((SSPBCLK.M + 1) * SSPBCLK.N)$$

In this formula, **SSPBCLK.N** must be an even value from 2 to 254.

25.8.5 SSP Clock Constraints

The SSP clock configuration must be configured according to the guidelines below, in order for the peripheral to operate correctly.

Here are the clock constraints for the input clock SSPCLK (after the pre-scaler) and output SSP clock, USBCLK:

- When in master mode, the minimum frequency of the input SSPCLK must be at least 2X the output SSP clock (USBCLK)
- When in slave mode, the minimum frequency of the input SSPCLK must be at least 12X the input SSP clock (USBCLK)

The maximum frequency of the SSPCLK should be:

- When in master mode, $F_{SSPCLK(max)} \leq 254 * 128 * F_{USBCLK}$ (SSP clock output)
- When in slave mode, $F_{SSPCLK(max)} \leq 254 * 128 * F_{USBCLK}$ (SSP clock input)

25.8.6 SSP Configuration

The SSP may be configured to support various features of master and slave based serial interfaces.

To enable the SSP controller, set **SSPBCON.SSPEN** to 1b. When disabled, the SSP may still have registers configured for the configuration mode and interrupts, but will not operate on serial data until enabled. Once the SSP has been configured, set this bit to 1b to begin processing.

To configure the SSP mode, set **SSPBCON.FRFB** (frame format) to the desired value as shown below.

Table 25-7 SSP B Frame Format

| SSPBCON.FRFB | Frame Format |
|--------------|------------------------------------|
| 00b | SPI (Motorola) |
| 01b | Synchronous Serial Format (TI) |
| 10b | Microwire (National Semiconductor) |
| 11b | Reserved |

To select master mode, set **SSPBCON.MS** to 0b. In this mode, the controller is the bus master. It will drive the SCLK, MOSI and SS signals and will receive data on the MISO line.

To select slave mode, set **SSPBCON.MS** to 1b. In this mode, the controller is the bus slave and will drive the MISO line and receive input from SCLK, MOSI and SS.

To change the serial to parallel endian order, the user may use the **SSPBCON.LSBFIRST** field. To configure the SSP for LSB first, set **SSPBCON.LSBFIRST** to 1b. To configure the SSP for MSB first, set **SSPBCON.LSBFIRST** to a 0b.

To enable loop-back mode, set the **SSPBCON.LBM** to 1b. In this mode, serial input is taken from the serial output instead of from the digital peripheral MUX.

While in slave mode, the behavior of the MISO output can be controlled. When **SSPBCON.SOD** is set to 0b, the SSP can drive the MISO in slave mode. When this bit is 1b, the SSP will not drive the MISO output in slave mode.

The clock phase of the SSP may also be configured. To configure the SSP for capturing data on the first clock edge transition, set **SSPBCON.CPH** to 0b. To configure the SSP for capturing data on the second clock edge transition, set **SSPBCON.CPH** to 1b.

When the **SSPBCON.FRFB** is 00b (SPI), the clock output polarity may be configured. To configure the SSP clock to be active high, set **SSPBCON.CPO** to 0b. To configure the SSP clock to be active low, set **SSPBCON.CPO** to 1b.

The SSP may also configure the size of the data word size transferred in each frame. To configure the data size, see the table below.

Table 25-8 SSP B Data Size Select

| SSPBCON.DSS | Data Size |
|-------------|-----------|
| 0000b | Reserved |
| 0001b | Reserved |
| 0010b | Reserved |
| 0011b | 4-bit |
| 0100b | 5-bit |
| 0101b | 6-bit |
| 0110b | 7-bit |
| 0111b | 8-bit |
| 1000b | 9-bit |
| 1001b | 10-bit |
| 1010b | 11-bit |
| 1011b | 12-bit |
| 1100b | 13-bit |
| 1101b | 14-bit |
| 1110b | 15-bit |
| 1111b | 16-bit |

25.8.7 SSP Slave Select Configuration

The SSP allows the behavior of the SS (slave select) signal to be configured by using the **SSPBSSCR** register as follows when configured as a bus master.

The **SSPBSSCR.SELCS** field must always be set to a 00b for proper operation.

To configure the SSP to automatically control the behavior of the SS signal, set **SSPBSSCR.SWSEL** to 0b. If the user wants to control the SS signal by software, set **SSPBSSCR.SWSEL** to 1b.

To configure the SS signal behavior after a frame transfer, use the **SSPBSSCR.SPHDONTCARE** field. To configure the SSP to not pull the SS signal high after a frame transfer, set **SSPBSSCR.SPHDONTCARE** to 0b. To configure the SSP to pull the SS signal high after a frame transfer, set the **SSPBSSCR.SPHDONTCARE** to 1b.

25.8.8 SSP Interrupts

The SSP may configure interrupts for the following conditions.

When the SSP detects that the TX FIFO is half-full or less, the **SSPBRIS.TXRIS** and **SSPBMIS.TXMIS** are set to 1b. During this condition, if the **SSPBIMSC.TXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTB signal to the NVIC.

When the SSP detects that the RX FIFO is half-full or more, the **SSPBRIS.RXRIS** and **SSPBMIS.RXMIS** are set to 1b. During this condition, if the **SSPBIMSC.RXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTB signal to the NVIC.

The SSP contains a timer for reading the RX FIFO. If the RX FIFO is not empty, and $32 F_{SSPCLK}$ periods have gone by, the **SSPBRIS.RTIM** and **SSPBMIS.RTMIS** bits are set to 1b. During this condition, if the **SSPBIMSC.RTIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTB signal to the NVIC. This condition may be cleared by writing **SSPBICLR.RTIC** to 1b.

If the SSP detects the RX FIFO is full, when a character is attempted to be inserted into it, it sets the **SSPBRIS.RORIM** and **SSPBMIS.RORIS** to 1b. During this condition, if the **SSPBIMSC.RORIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTB signal to the NVIC. This condition may be cleared by writing **SSPBICLR.RORIC** to 1b.

25.8.9 Master Mode Operation

When configured as a bus master (**SSPBCON.MS** = 0b), the SSP will insert a character into the TX FIFO when it is written into the **SSPBDAT** register. The SSP will read a character from the TX FIFO and performs a parallel to serial conversion on it. Then the serial data stream and frame control signal is synchronized to the clock and are output through the USBMOSI/TX pin to the attached slaves. While the data is being transmit to the slaves, the USBSCLK and USBSS behave as configured in the **SSPBCON** and **SSPBCSCR** registers.

The master receive logic performs serial to parallel conversion on the incoming synchronous USBMISO/RX data stream, and stores the character in the RX FIFO. The master may read first entry in the RX FIFO by reading the **SSPBDAT** register.

25.8.10 Slave Mode Operation

When configured as a bus slave (**SSPBCON.MS** = 1b), the SCLK is provided by the attached master. The user can write the next character to send into **SSPBDAT** and it will get inserted into the TX FIFO. The slave transmit logic reads a value from the TX FIFO, performs parallel to serial conversion and outputs the serial data stream on the USBMISO/RX pin to the attached master. The slave receive logic performs serial to parallel conversion on the incoming USBMOSI/TX data stream, extracting and storing values into the RX FIFO. The first character in the RX FIFO may be read by reading the **SSPBDAT** register.

25.8.11 SSP Status

The SSP module maintains status information on the operation of the module. The table below shows the available SSP status conditions.

Table 25-9 SSP B Status

| SSPBSTAT field | Description |
|----------------|--|
| BSY | If set, the SSP controller is currently sending/receiving a frame and/or the TX FIFO is not empty. |
| RFF | If set, the RX FIFO is full. |
| RNE | If set, the RX FIFO is not empty. |
| TNF | If set, the TX FIFO is full. |
| TNE | If set, the TX FIFO is not empty |

25.9 SSP Register Summary

Table 25-10 USART B Register Summary (SSP Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------------|------------|---|--------|------------|
| SSPACON | 4003 0044h | SSP A Control Register | RW | 0000 0000h |
| SSPASTAT | 4003 0048h | SSP A Status Register | RO | 0000 0003h |
| SSPADAT | 4003 0008h | SSP A Data Register | RW | 0000 0000h |
| SSPACLK | 4003 000Ch | SSP A Clock Control Register | RW | 0000 0000h |
| SSPAIMSC | 4003 0010h | SSP A Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPARIS | 4003 0014h | SSP A Raw Interrupt Status Register | RO | 0000 0008h |
| SSPAMIS | 4003 0018h | SSP A Masked Interrupt Status Register | RO | 0000 0008h |
| SSPAICLR | 4003 001Ch | SSP A Interrupt Clear Register | RW | 0000 0000h |
| SSPASSCR | 4003 0028h | SSP A Slave Select Configuration Register | RW | 0000 0000h |

25.10 SSP Register Detail

25.10.1 SSPBCON

Register 25-11 SSPBCON (SSP B Control Register, 4003 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | LSBFIRST | RW | 0 | Endian Order: 0b: MSB transmit / receive first 1b: LSB transmit / receive first |
| 12 | LBM | RW | 0 | Loopback Mode: 0b: Normal operation 1b: Loopback mode. Serial Input is taken from the serial output (MOSI or MISO) rather than the serial in (MISO or MOSI respectively) |
| 11 | SSPEN | RW | 0 | SSP Enable: 0b: SSP Controller is disabled 1b: SSP Controller is enabled. It will interact with other devices on the serial bus. Software should write the appropriate control information to the other SPI/SSP registers and interrupt controller registers, before setting this bit. |
| 10 | MS | RW | 0 | Master/Slave Mode: 0b: The SSP controller acts as a bus master, driving the USBCLK, USBMOSI and USSS signals 1b: The SSP controller acts as a bus slave, driving the USBMISO and receiving USBCLK, USBMOSI and USBSS. |
| 9 | SOD | RW | 0 | Slave Output Disable: 0b: The SSP can drive the USBMISO output in slave mode 1b: The SSP must not drive the USBMISO output in slave mode |
| 8 | CPH | RW | 0 | Clock Out Phase. This bit is only used when SSPBCON.FRF = 0b (SPI). 0b: The SSP controller captures serial data on the first edge transition of the frame. 1b: The SSP controller captures serial data on the second edge transition of the frame. |
| 7 | CPO | RW | 0 | Clock Out Polarity (this bit is only used in SPI mode when SSPBCON.FRF = 00b) 0b: The clock is active high. 1b: The clock is active low. |
| 6:5 | FRF | RW | 0 | Frame Format: 00b: SPI 01b: TI 10b: Microwire 11b: Reserved |
| 4:0 | DSS | RW | 0 | Data Size Select: |

| | | | | |
|--|--|--|--|--|
| | | | | 0 0000b: Reserved 0 0001b: Reserved 0 0010b: Reserved 0 0011b: 4-bit transfer 0 0100b: 5-bit transfer 0 0101b: 6-bit transfer 0 0110b: 7-bit transfer 0 0111b: 8-bit transfer 0 1000b: 9-bit transfer 0 1001b: 10-bit transfer 0 1010b: 11-bit transfer 0 1011b: 12-bit transfer 0 1100b: 13-bit transfer 0 1101b: 14-bit transfer 0 1110b: 15-bit transfer 0 1111b: 16-bit transfer ... 1 1111b: 32-bit transfer |
|--|--|--|--|--|

25.10.2 SSPBSTAT

Register -12 SSPBSTAT (SSP B Status Register, 4003 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | BSY | RO | 0 | Busy Bit: 0b: SPI controller is idle 1b: SPI controller is sending/receiving a frame and/or the TX FIFO is not empty |
| 3 | RFF | RO | 0 | Receive FIFO Full: 0b: RX FIFO not full 1b: RX FIFO full |
| 2 | RNE | RO | 0 | Receive FIFO not empty: 0b: RX FIFO empty 1b: RX FIFO not empty |
| 1 | TNF | RO | 0 | Transmit FIFO not full: 0b: TX FIFO full 1b: TX FIFO not full |
| 0 | TFE | RO | 0 | Transmit FIFO empty: 0b: TX FIFO not empty 1b: TX FIFO empty |

25.10.3 SSPBDAT

Register 25-13 SSPBDAT (SSP B Data Register, 4003 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------|--------|-------|---|
| 31:0 | DATA | RW | 0 | <p>Software can writing data to be sent in a future frame to this register when the SSPSTAT.TNF bit is set to 1b, indicating that the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise, the data written to this register will be sent as soon as all previous data has been sent (and received). If the data length is less than 32-bits, the data must be right-justified in this register.</p> <p>Software may read data from this register when the SSPSTAT.RNE bit is set to a 1b, indicating that the RX FIFO is not empty. When software reads this register, the SPI controller returns the data from the least recent frame in the RX FIFO. If the data length is less than 32-bits, the data is right-justified in this field with the MSBs set to 0.</p> |

25.10.4 SSPBCLK

Register 25-14 SSPBCLK (SSP B Clock Register, 4003 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:8 | M | RW | 0 | These fields may set the SPI master clock rate by the formula: $F_{SSPCLK} = PCLK / ((SSPBCLK.M + 1) * SSPBCLK.N)$ N must be an even value from 2 to 254. |
| 7:0 | N | RW | 0 | |

25.10.5 SSPBIMSC

Register 25-15 SSPBIMSC (SSP B Interrupt Mask Set and Clear Enable Register, 4003 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXIM | RW | 0 | Transmit FIFO Interrupt Mask: 0b: TX FIFO half-empty or less condition interrupt disabled 1b: TX FIFO half-empty or less condition interrupt enabled |
| 2 | RXIM | RW | 0 | Receive FIFO Interrupt Mask: 0b: RX FIFO half-full or more condition interrupt disabled 1b: RX FIFO half-full or more condition interrupt enabled |
| 1 | RTIM | RW | 0 | Receive Timeout Interrupt Mask: 0b: RX FIFO not empty and no read prior to timeout period interrupt disabled 1b: RX FIFO not empty and no read prior to timeout period interrupt enabled |
| 0 | RORIM | RW | 0 | Receive Overrun Interrupt Mask: 0b: RX FIFO written to while full condition interrupt disabled 1b: RX FIFO written to while full condition interrupt enabled |

25.10.6 SSPBRIS

Register 25-16 SSPBRIS (SSP B Raw Interrupt Status Register, 4003 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXRIS | RO | 1 | TX FIFO Raw Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXRIS | RO | 0 | RX FIFO Raw Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTRIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | RORIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

25.10.7 SSPBMIS

Register 25-17 SSPBMIS (SSP B Masked Interrupt Status Register, 4003 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXMIS | RO | 1 | TX FIFO Masked Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXMIS | RO | 0 | RX FIFO Masked Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTMIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | ROMIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

25.10.8 SSPBICLR

Register 25-18 SSPBICLR (SSP B Interrupt Clear Register, 4003 001Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:2 | Reserved | RO | 0 | Reserved |
| 1 | RTIC | RW | 0 | RX FIFO Timeout Interrupt Clear: 0b: No effect 1b: Clears the SSPBxIS.RTRIS interrupt flag This bit is self-clearing. |
| 0 | RORIC | RW | 0 | RX FIFO Overwrite Interrupt Clear: 0b: No effect 1b: Clears the SSPBxIS.RORIS interrupt flag This bit is self-clearing. |

25.10.9 SSPBSSCR

Register 25-19 SSPBSSCR (SSP B Slave Select Configuration Register, 4003 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|--------------------|--------|-------|---|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | SPHDONTCARE | RW | 0 | Slave Select Pull High: 0b: USBSS cannot pull high after frame transfer 1b: USBSS must pull high after frame transfer |
| 3 | SWSS | RW | 0 | Slave Select State: 0b: Set USBSS to low 1b: Set USBSS to high |
| 2 | SWSEL | RW | 0 | Slave Select Software Control: 0b: USBSS is automatically controlled by the SPI module 1b: USBSS is software controlled by SSPBSSCR.SWCS |
| 1:0 | SELSS | RW | 0 | Slave Select Signal Control: 00b: USBSS is enabled 01b: Reserved 10b: Reserved 11b: Reserved |

26 USART C

26.1 Overview

The PAC55XX family contains support for four Universal Synchronous Asynchronous Receive Transmit (USART) peripherals.

A USART is a serial communications engine that may be configured for UART or SSP through a mode selection register.

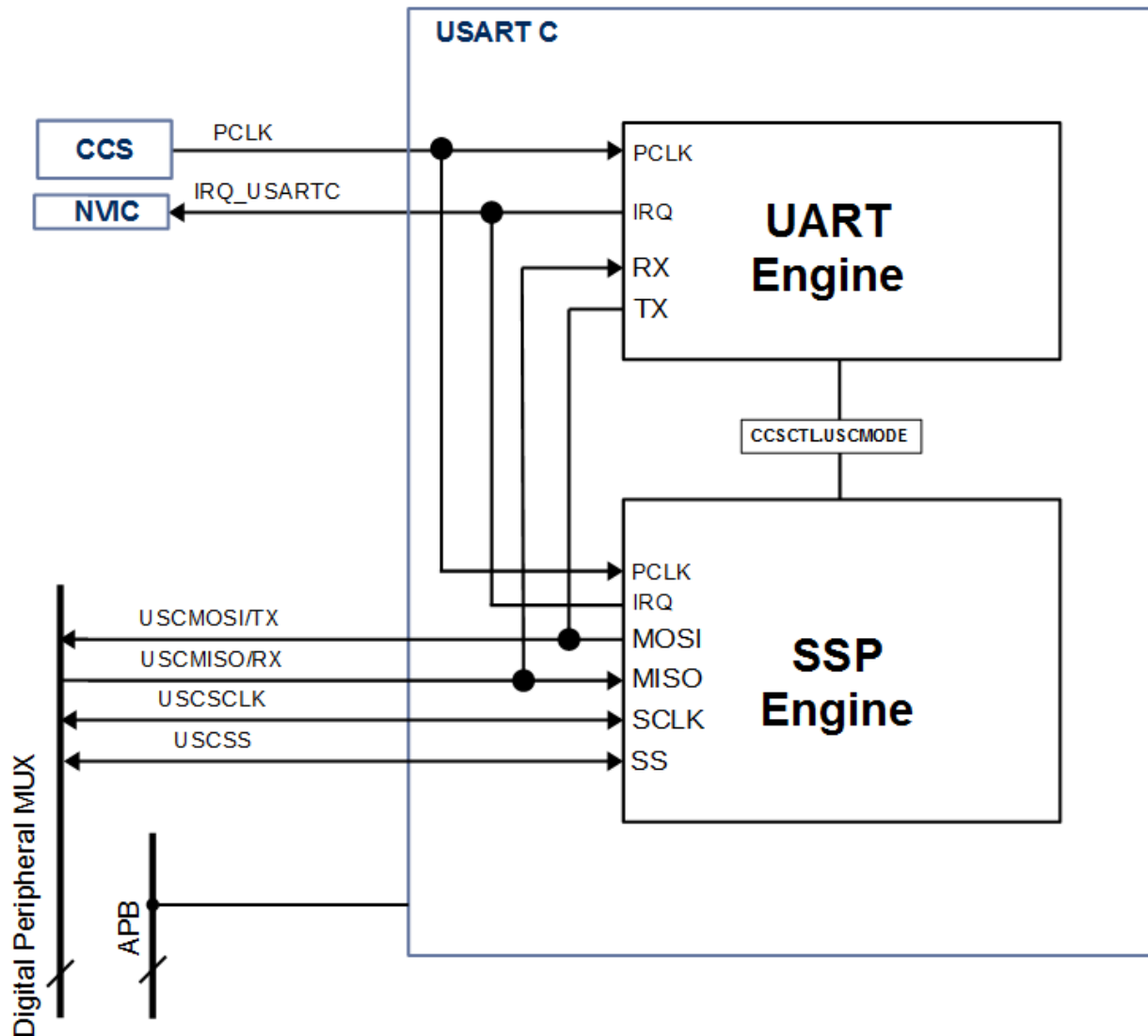
The USART is an APB client and is clocked by the Peripheral Clock (PCLK) system clock.

26.2 Features

- **UART Mode:**
 - 5-bit to 8-bit data interface
 - Optional parity
 - 1, 1.5 or 2 stop bits
 - 16-bit programmable baud-rate generator
 - 8-bit scratch pad
 - Independent RX and TX FIFOs
 - Configurable RX and TX interrupts
- **SSP Mode:**
 - Motorola SPI, TI Synchronous serial or National Semiconductor Microwire support
 - Master and Slave mode support
 - Independent RX and TX FIFOs
 - Configurable clock pre-scaler
 - Programmable Interrupts

26.3 System Block Diagram

Figure 26-1 USART C System Block Diagram



26.4 Functional Description

The USART may be configured for either UART or SSP mode via the **CCSCTL.USCMODE** register.

In UART mode, the peripheral can receive or transmit UART serial data over the USC MOSI/TX (UART TX) or USC MISO/RX serial lines. In this mode, the USART signals for USC SCLK and USC SS are unused. The UART may assert the IRQ_USARTC signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In UART mode, only the UART registers may be accessed, starting at register offset 0.

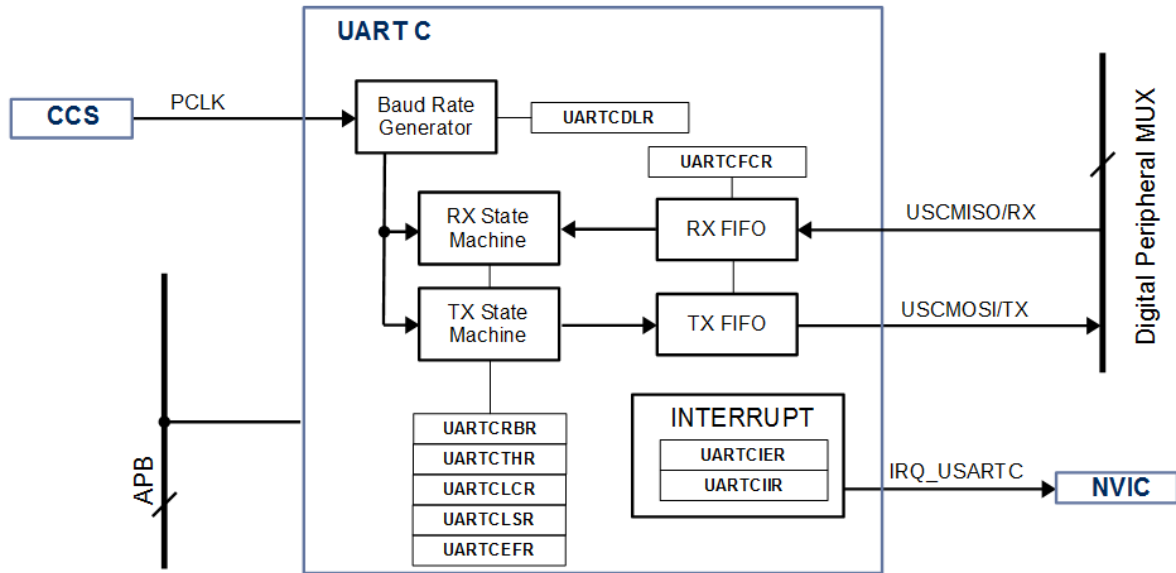
In SSP mode, the peripheral can receive or transmit serial data over the USCMOSI/TX (SPI MOSI) or USCMISO/RX (SPI MISO) serial lines. The USCCLK and USCSS are used by the peripheral in either master or slave mode for the SSP clock and Slave Select signals. The SSP may assert the IRQ_USARTC signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In SPI mode, only the SPI registers may be accessed, starting at register offset 0.

26.5 UART Mode Functional Description

26.5.1 UART C Mode System Block Diagram

Figure 26-2 UART C Mode System Block Diagram



26.5.2 Mode Configuration

The UART mode of the USART must first be selected by writing **CCSTL.USCMODE** to 1b. This will set the USART into UART mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USCMODE** is 1b are below.

Table 26-1 USART C UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|--|
| UARTCRBR | UART C Receive Buffer Register |
| UARTCTHR | UART C Transmit Holding Register |
| UARTCDLR | UART C Divisor Latch Register |
| UARTCIER | UART C Interrupt Enable Register |
| UARTCIIR | UART C Interrupt Identification Register |
| UARTCFCR | UART C FIFO Control Register |
| UARTCLCR | UART C Line control Register |
| UARTCLSR | UART C Line Status Register |
| UARTCSCR | UART C Scratch Pad Register |

26.5.3 Baud Rate Configuration

To configure the baud rate for RX/TX operations, the **UARTCDLR** may be used.

To baud rate may be configured by the following formula:

$$\text{Baud Rate} = \text{PCLK} / (16 * \text{UARTCDLR})$$

Note that UART peripherals sometimes need an accurate timing base in order to function properly. Because of that, it is suggested that the CLKREF, EXTCLK or XTALCLK be used as the clock source for PCLK when using the UART.

The ROSCCLK may not provide enough accuracy for UART applications.

26.5.4 FIFO Reset

The RX and TX FIFOs may be reset independently by using the **UARTCFC** register.

To change any of the bits in the **UARTCFCR**, the **UARTCFCR.FIFOEN** must be set to 1b (**UARTCFCR** access).

To reset the RX FIFO, write the **UARTCFCR.RXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the RX state machine.

To reset the TX FIFO, write the **UARTCFCR.TXFIFORST** bit to 1b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the TX state machine.

When the **UARTCFCR** register is done being used to configure the FIFOs, set the **UARTCFCR.FIFOEN** to 1b (application mode).

26.5.5 UART Configuration

In UART mode, the UART may be have the parity, break control, stop bits and word length configured by the user.

The parity checking may be enabled or disabled. To enable parity checking, set the **UARTCLCR.PEN** to 1b. To disable, set this field to 0b. If enabled, the user may configure the type of parity checking as follows.

Table 26-2 UART C Parity Modes

| UARTCLCR.PEN | UARTCLCR.PSEL | Parity | Description |
|--------------|---------------|----------------|--|
| 0b | n/a | None | |
| 1b | 00b | Odd | The number of ones in the transmitted character plus the attached parity bit will be odd. |
| | 01b | Even | The number of ones in the transmitted character plus the attached parity bit will be even. |
| | 10b | Forced 1 stick | Force 1 for parity |
| | 11b | Forced 0 stick | Force 0 for parity |

To set the number of stop bits, the user may write the **UARTCLCR.SBS** field. To select 1 stop bits, set **UARTCLCR.SBS** to 0b. To select 2 stop bits, set **UARTCLCR.SBS** to 1b. If the **UARTCLCR.WLS** is set to 00b (5-bit character length) and **UARTCLCR.SBS** is set to 1b, then there will be 1.5 stop bits.

To configure break control, the user can set the **UARTCLCR.BCON** field. To disable break transmission, set **UARTCLCR.BCON** to 0b. To force the TX signal to logic 0, set the **UARTCLCR.BCON** to 1b.

To set the word length for the UART, the user can set the **UARTCLCR.WLS** field as shown below.

Table 26-3 UART C Word Length

| UARTCLCR.WLS | UART word length |
|--------------|------------------|
| 00b | 5-bit |
| 01b | 6-bit |
| 10b | 7-bit |
| 11b | 8-bit |

26.5.6 UART Scratch Pad

There is an 8-bit general purpose register that may be used in the USART.

The user may use the **USCSCR** register for an 8-bit scratch pad.

26.5.7 UART Interrupts

The status of interrupts may be read at any time by the **UARTCIIR** register. If the **UARTCIIR.INTSTATUS** field is set to 0b, at least one interrupt is pending. The interrupt type can be read by the **UARTCIIR.INTID** field as shown below.

Table 26-4 UART C Interrupt ID

| UARTCIIR.INTID | Interrupt Type |
|----------------|---------------------------------|
| 000b | Reserved |
| 001b | TX Holding Register Empty |
| 010b | Receive Data Available |
| 011b | Receive Line Status |
| 100b | Reserved |
| 101b | Reserved |
| 110b | Receive FIFO character time-out |
| 111b | Reserved |

26.5.8 Transmit Operation

Before any data operation, the user must be sure to set the **UARTCFCR.FIFOEN** to 0b (normal mode).

Transmission may be initiated by writing **UARTCTHR** with the data desired to be sent. The data will be written into the TX FIFO and then transferred to the TX shift register together with a start bit, parity and stop bits as configured in the **UARTCLCR** register, LSB first.

The baud rate used will be set by the **UARTCDLR** register.

26.5.9 Transmit Interrupts

The UART may be configured to generate interrupts after characters have been successfully transmitted. The UART may be configured to generate an interrupt after 1, 4, 8 or 14 characters have been transmitted. By default, the interrupt is configured for 1 character.

To set the interrupt threshold for 4, 8 or 14 characters, the user should set the **UARTCEFR.ENMODE** = 1b and set the **UARTCFCR.TXTL** to the interrupt FIFO depth that is desired.

To enable transmit interrupts, set the **UARTCIER.THREIE** (TX holding register interrupt enable). When the transmit holding register has been emptied and the TX FIFO is empty, the **UARTCLSR.THRE** will be set to 1b. This field will be cleared the next time the **UARTCTHR** register is written with data to transmit.

If the **UARTCIER.THREIE** is set to 1b and the **UARTCLSR.THRE** interrupt flag is set, the **IRQ_USARTC** signal to the NVIC will be asserted.

26.5.10 Receive Operation

Before any data operation, the user must be sure to set the **UARTCFCR.FIFOEN** to 0b (normal mode).

Data is sampled into the RX shift register at a sampling rate of $PCLK / 16$. A filter is used to remove spurious inputs that last for less than two periods of the sampling rate. When the complete character has been clocked into the receiver, it is loaded into the RX FIFO. The UART RX FIFO is 16 bytes deep. The first entry into the RX FIFO will be available to read from the **UARTCRBR** register.

The receiver also checks the parity and stop bits as specified by the **UARTCLCR** register.

26.5.11 Receive Interrupts

The UART may be configured to generate interrupts after a character has been successfully received.

To enable receive interrupts, set the **UARTCIER.RBRIE** (RX buffer register interrupt enable). When the **UARTCRBR** (receive buffer register) has been filled from the RX FIFO, the **UARTCLSR.RDR** will be set to 1b. This field will be cleared the next time the **UARTCRBR** register is read.

If the **UARTCIER.RBRIE** is set to 1b and the **UARTCLSR.RDR** interrupt flag is set, the **IRQ_USARTC** signal to the NVIC will be asserted.

26.6 UART Register Summary

Table 26-5 USART C Register Summary (UART Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|--|--------|------------|
| UARTARBR | 4004 0000h | UART A Receive Buffer Register | RO | -- |
| UARTATHR | 4004 0004h | UART A Transmit Holding Register | WO | -- |
| UARTADLR | 4004 0008h | UART A Divisor Latch Register | RW | 0000 0001h |
| UARTAIER | 4004 000Ch | UART A Interrupt Enable Register | RW | 0000 0000h |
| UARTAIIR | 4004 0010h | UART A Interrupt Identification Register | RO | 0000 0001h |
| UARTAFCR | 4004 0014h | UART A FIFO Control Register | RW | 0000 0000h |
| UARTALCR | 4004 0018h | UART A Line control Register | RW | 0000 0000h |
| UARTALSR | 4004 0020h | UART A Line Status Register | RO | 0000 0060h |
| UARTASCR | 4004 0028h | UART A Scratch Pad Register | RW | -- |
| UARTAEFR | 4002 002Ch | UART A Enhanced Mode Register | RW | 0000 000h |

26.7 UART Register Detail

26.7.1 UARTCRBR

Register 26-1 UARTCRBR (UART C Receive Buffer Register, 4004 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | RBR | RO | 0 | Contains the oldest received character in the UART RX FIFO. |

26.7.2 UARTCTHR

Register 26-2 UARTCTHR (UART C Transmit Holding Register, 4004 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | THR | WO | - | Writing to the UARTBTHR causes the data to be stored in the UART transmit FIFO. The character will be sent when it reaches the bottom of the FIFO and the transmitter is available. |

26.7.3 UARTCDLR

Register 26-3 UARTCDLR (UART C Divisor Latch Register, 4004 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|------------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | DLR | RW | 0000 0001b | Sets the baud rate for the module. Baud rate = PCLK / (16 * UARTCDLR) |

26.7.4 UARTCIER

Register 26-4 UARTCIER (UART C Interrupt Enable Register, 4004 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:3 | Reserved | RO | 0 | Reserved |
| 2 | RLSIE | RW | 0 | RX Line Status Interrupt Enable. Enables the UART RX line status interrupts. The status of this interrupts can be read from UARTCLSR[4:1] . 0b: Disable the RX line status interrupts 1b: Enable the RX line status interrupts |
| 1 | THRIE | RW | 0 | TX Holding Register Empty Interrupt Enable. Enables the THRE interrupt for the UART. The status of this interrupt can be read from UARTCLSR.THRE . 0b: Disable the THRE interrupts 1b: Enable the THRE interrupts |
| 0 | RBRIE | RW | 0 | RX Buffer Register Interrupt Enable. Enables the Receive Data Available interrupt for the UART. It also controls the character receive time-out interrupt. 0b: Disable the RBR interrupts 1b: Enable the RBR interrupts |

26.7.5 UARTCIIR

Register 26-5 UARTCIIR (UART C Interrupt Identification Register, 4004 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3:1 | INTID | RO | 000b | Interrupt identification: 000b: Reserved 001b: TX Holding Register Empty 010b: Receive Data Available 011b: Receive Line Status 100b: Reserved 101b: Reserved 110b: Receive FIFO Character Time-out 111b: Reserved |
| 0 | INTSTATUS | RO | 1b | Interrupt status. Note that this bit is active low. The pending interrupt can be determined through the UARTCIIR.INTID field. 0b: At least one interrupt is pending 1b: No interrupt is pending |

Note that this register is cleared on read.

26.7.6 UARTCFCR

Register 26-6 UARTCFCR (UART C FIFO Control Register, 4004 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:6 | RXTL | RW | 0 | RX Trigger Level. These two bits determine how many receive UART FIFO characters must be written before an interrupt is activated. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 5:4 | TXTL | RW | 0 | TX Trigger Level. These two bits determine how may transmit UART FIFO characters must be written before an interrupt is activated. This field cannot be set unless UARTCFR.ENMODE = 1b. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 3 | Reserved | RO | 0 | Reserved |
| 2 | TXFIFORST | RW | 0 | TX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART TX FIFO. This bit is self-clearing |
| 1 | RXFIFORST | RW | 0 | RX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART RX FIFO. This bit is self-clearing |
| 0 | FIFOEN | RW | 0 | FIFO Enable: 0b: UART FIFOs are disabled. Reset UARTAFCR settings to their default values. 1b: UART RX and TX FIFOs are enabled. UARTCFR[7:1] are accessible when this bit is set. Any transition on this bit will automatically clear the UART RX AND TX FIFOs. |

26.7.7 UARTCLCR

Register 26-7 UARTCLCR (UART C Line Control Register, 4004 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6 | BCON | RW | 0 | Break Control: 0b: Disable break transmission 1b: Enable break transmission. Output pin UARTCTXD is forced to logic 0 when this bit is set. |
| 5:4 | PSEL | RW | 0 | Parity Select: 00b: Odd parity. The number of 1s in the transmitted character and the attached parity will be odd. 01b: Even parity. The number of 1s in the transmitted character and the attached parity will be even. 10b: Forced 1 stick parity. 11b: Forced 0 stick parity. |
| 3 | PEN | RW | 0 | Parity Enable: 0b: Disable parity generation and checking 1b: Enable parity generation and checking |
| 2 | SBS | RW | 0 | Stop Bit Select: 0b: 1 stop bit 1b: 2 stop bits, 1.5 stop bits if (UARTCLCR.WLS = 00b) |
| 1:0 | WLS | RW | 0 | Word Length Select: 00b: 5-bit character length 01b: 6-bit character length 10b: 7-bit character length 11b: 8-bit character length |

26.7.8 UARTCLSR

Register 26-8 UARTCLSR (UART C Line Status Register, 4004 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | RXFE | RO | 0 | <p>Error in RX FIFO.</p> <p>This bit is set when a character with a RX error such as framing, parity or break is loaded into UARTCRBR. This bit is cleared when the UARTCLSR register is read, and there are no subsequent errors in the UART FIFO.</p> |
| 6 | TEMT | RO | 0 | <p>Transmitter Empty.</p> <p>This bit is set when both UARTCTHR is empty; this bit is cleared when THR contains valid data.</p> <p>0b: THR contains valid data 1b: THR are empty</p> |
| 5 | THRE | RO | 0 | <p>Transmitter Holding Register Empty.</p> <p>This bit is set immediately upon detection of an empty UART THR and is cleared on a THR write.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 4 | BI | RO | 0 | <p>Break Interrupt.</p> <p>When UARTCRX is held in the spacing state (all zeroes) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until UARTCRX goes to the marking statue (all ones). A read of the UARTCLSR register clears this status bit. The time of break detection is dependents on UARTCFCR.FIFOEN.</p> <p>The break interrupt is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Break interrupt status is inactive 1b: Break interrupt status is active</p> |
| 3 | FE | RO | 0 | <p>Framing Error.</p> <p>When the stop bit of a received character is logic 0, a framing error occurs. Reading the UARTCLSR register will clear this bit. The time of the framing error is dependent upon UARTCFCR[3]. Upon detection of a framing error, the RX will attempt to re-synchronize the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no framing error.</p> <p>A framing error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Framing error status is inactive 1b: Framing error status is active</p> |
| 2 | PE | RO | 0 | <p>Parity Error.</p> <p>When the parity bit of a received character is in the wrong</p> |

| | | | | |
|---|------------|----|---|---|
| | | | | <p>state, a parity error occurs. Reading the UARTCLSR register will clear this bit. The time of the parity error detection is dependent on UARTCFR.FIFOEN.</p> <p>A parity error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Parity error status is inactive 1b: Parity error status is active</p> |
| 1 | OE | RO | 0 | <p>Overrun Error.</p> <p>The overrun error condition is set as soon as it occurs. Reading the UARTCLSR register will clear this bit. This bit is set when the UART RSR has a new character assembled and the UART RBR FIFO is full. In this case, the UART RBR FIFO will not be overwritten and the character in the UART RSR will be lost.</p> <p>0b: Overrun error status is inactive 1b: Overrun error status is active</p> |
| 0 | RDR | RO | 0 | <p>Receiver Data Ready.</p> <p>This bit is set when the RBR holds an unread character and is cleared when the UART RBR FIFO is empty.</p> <p>0b: RBR is empty 1b: RBR contains valid data</p> |

26.7.9 UARTCSCR

Register 26-9 UARTCSCR (UART C Scratch Pad Register, 4004 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|-------------------------------|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | PAD | RW | - | A read-able, write-able byte. |

26.7.10 UARTCEFR

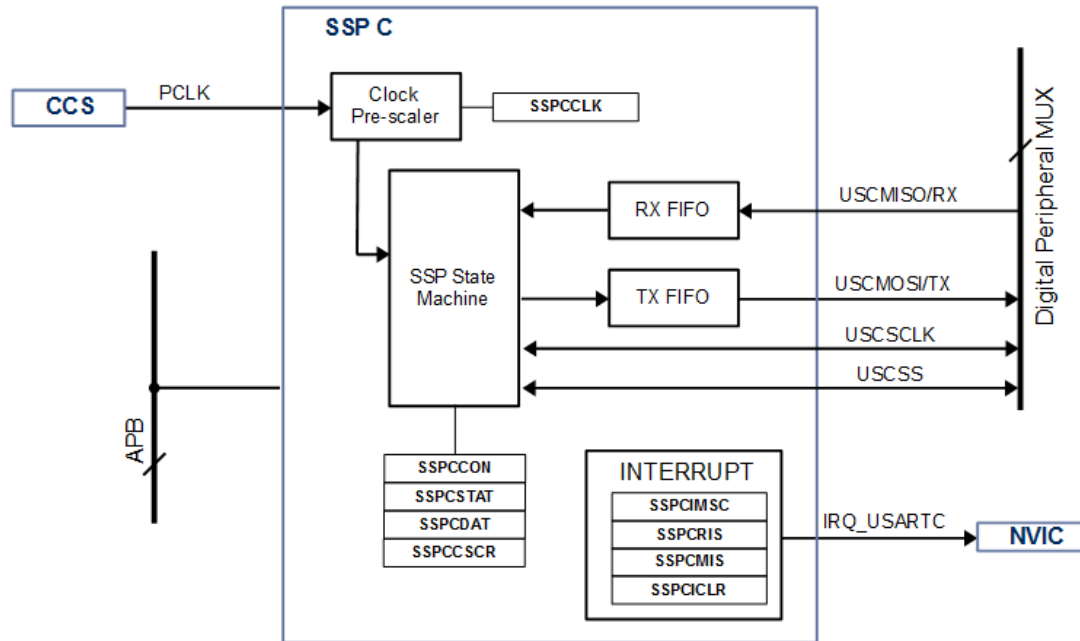
Register 26-10 UARTCEFR (UARTC Enhanced Feature Register, 4004 002Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | ENMODE | RW | 0 | Enhanced Mode 0b – disabled 1b – enabled |
| 3:0 | Reserved | RO | 0 | Reserved |

26.8 SSP Functional Description

26.8.1 SSP Mode System Block Diagram

Figure 26-3 SSP C System Block Diagram



26.8.2 Mode Configuration

The SSP mode of the USART must first be selected by writing **CCSCTL.USCMODE** to 0b. This will set the USART into SSP C mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USCMODE** is 0b are below.

Table 26-6 USART C UART Mode Registers

| REGISTER | DESCRIPTION |
|------------------|---|
| SSPCON | SSP C Control Register |
| SSPCSTAT | SSP C Status Register |
| SSPCDAT | SSP C Data Register |
| SSPCCLK | SSP C Clock Control Register |
| SSPCIMISC | SSP C Interrupt Mask Set and Clear Register |
| SSPCRIS | SSP C Raw Interrupt Status Register |
| SSPCMIS | SSP C Masked Interrupt Status Register |
| SSPCICLR | SSP C Interrupt Clear Register |
| SSPCSSCR | SSP C Slave Select Configuration Register |

26.8.3 SSP Overview

When in SSP mode, the USART may be a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- Motorola-style SPI
- TI-style Synchronous Serial Interface
- National Semiconductor-style Microwire

In both master and slave modes, the SSP performs parallel to serial conversion on data into 32-bit wide, 8-location deep RX and TX FIFOs.

The SSP may be configured to generate interrupts for servicing the TX and RX FIFOs and error conditions such as FIFO overwrite and timeout.

26.8.4 Clock Configuration

When configured for master mode, the SSP peripheral may configure the clock pre-scaler to generate the desired SCLK output clock frequency. The input clock on the SSP peripheral is the PCLK (peripheral clock) system clock.

The SCLK is generated from the PCLK input and the **SSPCCLK.M** and **SSPCCLK.N** parameters. The SCLK clock frequency is generated according to the following formula:

$$F_{\text{SSPCLK}} = \text{PCLK} / ((\text{SSPCCLK.M} + 1) * \text{SSPCCLK.N})$$

In this formula, **SSPCCLK.N** must be an even value from 2 to 254.

26.8.5 SSP Clock Constraints

The SSP clock configuration must be configured according to the guidelines below, in order for the peripheral to operate correctly.

Here are the clock constraints for the input clock SSPCLK (after the pre-scaler) and output SSP clock, USCCLK:

- When in master mode, the minimum frequency of the input SSPCLK must be at least 2X the output SSP clock (USCCLK)
- When in slave mode, the minimum frequency of the input SSPCLK must be at least 12X the input SSP clock (USCCLK)

The maximum frequency of the SSPCLK should be:

- When in master mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USCCLK}}$ (SSP clock output)
- When in slave mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USCCLK}}$ (SSP clock input)

26.8.6 SSP Configuration

The SSP may be configured to support various features of master and slave based serial interfaces.

To enable the SSP controller, set **SSPCON.SSPEN** to 1b. When disabled, the SSP may still have registers configured for the configuration mode and interrupts, but will not operate on serial data until enabled. Once the SSP has been configured, set this bit to 1b to begin processing.

To configure the SSP mode, set **SSPCON.FRFB** (frame format) to the desired value as shown below.

Table 26-7 SSP C Frame Format

| SSPCON.FRFB | Frame Format |
|-------------|------------------------------------|
| 00b | SPI (Motorola) |
| 01b | Synchronous Serial Format (TI) |
| 10b | Microwire (National Semiconductor) |
| 11b | Reserved |

To select master mode, set **SSPCON.MS** to 0b. In this mode, the controller is the bus master. It will drive the SCLK, MOSI and SS signals and will receive data on the MISO line.

To select slave mode, set **SSPCON.MS** to 1b. In this mode, the controller is the bus slave and will drive the MISO line and receive input from SCLK, MOSI and SS.

To change the serial to parallel endian order, the user may use the **SSPCON.LSBFIRST** field. To configure the SSP for LSB first, set **SSPCON.LSBFIRST** to 1b. To configure the SSP for MSB first, set **SSPCON.LSBFIRST** to a 0b.

To enable loop-back mode, set the **SSPCON.LBM** to 1b. In this mode, serial input is taken from the serial output instead of from the digital peripheral MUX.

While in slave mode, the behavior of the MISO output can be controlled. When **SSPCON.SOD** is set to 0b, the SSP can drive the MISO in slave mode. When this bit is 1b, the SSP will not drive the MISO output in slave mode.

The clock phase of the SSP may also be configured. To configure the SSP for capturing data on the first clock edge transition, set **SSPCON.CPH** to 0b. To configure the SSP for capturing data on the second clock edge transition, set **SSPCON.CPH** to 1b.

When the **SSPCON.FRFB** is 00b (SPI), the clock output polarity may be configured. To configure the SSP clock to be active high, set **SSPCON.CPO** to 0b. To configure the SSP clock to be active low, set **SSPCON.CPO** to 1b.

The SSP may also configure the size of the data word size transferred in each frame. To configure the data size, see the table below.

Table 26-8 SSP C Data Size Select

| SSPCON.DSS | Data Size |
|------------|-----------|
| 0000b | Reserved |
| 0001b | Reserved |
| 0010b | Reserved |
| 0011b | 4-bit |
| 0100b | 5-bit |
| 0101b | 6-bit |
| 0110b | 7-bit |
| 0111b | 8-bit |
| 1000b | 9-bit |
| 1001b | 10-bit |
| 1010b | 11-bit |
| 1011b | 12-bit |
| 1100b | 13-bit |
| 1101b | 14-bit |
| 1110b | 15-bit |
| 1111b | 16-bit |

26.8.7 SSP Slave Select Configuration

The SSP allows the behavior of the SS (slave select) signal to be configured by using the **SSPCSSCR** register as follows when configured as a bus master.

The **SSPCSSCR.SELCS** field must always be set to a 00b for proper operation.

To configure the SSP to automatically control the behavior of the SS signal, set **SSPCSSCR.SWSEL** to 0b. If the user wants to control the SS signal by software, set **SSPCSSCR.SWSEL** to 1b.

To configure the SS signal behavior after a frame transfer, use the **SSPCSSCR.SPHDONTCARE** field. To configure the SSP to not pull the SS signal high after a frame transfer, set **SSPCSSCR.SPHDONTCARE** to 0b. To configure the SSP to pull the SS signal high after a frame transfer, set the **SSPCSSCR.SPHDONTCARE** to 1b.

26.8.8 SSP Interrupts

The SSP may configure interrupts for the following conditions.

When the SSP detects that the TX FIFO is half-empty or less, the **SSPCRIS.TXRIS** and **SSPCMIS.TXMIS** are set to 1b. During this condition, if the **SSPCIMSC.TXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTC signal to the NVIC.

When the SSP detects that the RX FIFO is half-full or more, the **SSPCRIS.RXRIS** and **SSPCMIS.RXMIS** are set to 1b. During this condition, if the **SSPCIMSC.RXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTC signal to the NVIC.

The SSP contains a timer for reading the RX FIFO. If the RX FIFO is not empty, and $32 F_{SSPCLK}$ periods have gone by, the **SSPCRIS.RTIM** and **SSPCMIS.RTMIS** bits are set to 1b. During this condition, if the **SSPCIMSC.RTIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTC signal to the NVIC. This condition may be cleared by writing **SSPCICLR.RTIC** to 1b.

If the SSP detects the RX FIFO is full, when a character is attempted to be inserted into it, it sets the **SSPCRIS.RORIM** and **SSPCMIS.ROMIS** to 1b. During this condition, if the **SSPCIMSC.RORIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTC signal to the NVIC. This condition may be cleared by writing **SSPCICLR.RORIC** to 1b.

26.8.9 Master Mode Operation

When configured as a bus master (**SSPCON.MS** = 0b), the SSP will insert a character into the TX FIFO when it is written into the **SSPCDAT** register. The SSP will read a character from the TX FIFO and performs a parallel to serial conversion on it. Then the serial data stream and frame control signal is synchronized to the clock and are output through the USCMOSI/TX pin to the attached slaves. While the data is being transmit to the slaves, the USCCLK and USSS behave as configured in the **SSPCON** and **SSPCCSCR** registers.

The master receive logic performs serial to parallel conversion on the incoming synchronous USCMISO/RX data stream, and stores the character in the RX FIFO. The master may read first entry in the RX FIFO by reading the **SSPCDAT** register.

26.8.10 Slave Mode Operation

When configured as a bus slave (**SSPCON.MS** = 1b), the SCLK is provided by the attached master. The user can write the next character to send into **SSPCDAT** and it will get inserted into the TX FIFO. The slave transmit logic reads a value from the TX FIFO, performs parallel to serial conversion and outputs the serial data stream on the USCMISO/RX pin to the attached master. The slave receive logic performs serial to parallel conversion on the incoming USCMOSI/TX data stream, extracting and storing values into the RX FIFO. The first character in the RX FIFO may be read by reading the **SSPCDAT** register.

26.8.11 SSP Status

The SSP module maintains status information on the operation of the module. The table below shows the available SSP status conditions.

Table 26-9 SSP C Status

| SSPCSTAT field | Description |
|----------------|--|
| BSY | If set, the SSP controller is currently sending/receiving a frame and/or the TX FIFO is not empty. |
| RFF | If set, the RX FIFO is full. |
| RNE | If set, the RX FIFO is not empty. |
| TNF | If set, the TX FIFO is full. |
| TNE | If set, the TX FIFO is not empty |

26.9 SSP Register Summary

Table 26-10 USART C Register Summary (SSP Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------------|------------|---|--------|------------|
| SSPACON | 4004 0044h | SSP A Control Register | RW | 0000 0000h |
| SSPASTAT | 4004 0048h | SSP A Status Register | RO | 0000 0003h |
| SSPADAT | 4004 0008h | SSP A Data Register | RW | 0000 0000h |
| SSPACLK | 4004 000Ch | SSP A Clock Control Register | RW | 0000 0000h |
| SSPAIMSC | 4004 0010h | SSP A Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPARIS | 4004 0014h | SSP A Raw Interrupt Status Register | RO | 0000 0008h |
| SSPAMIS | 4004 0018h | SSP A Masked Interrupt Status Register | RO | 0000 0008h |
| SSPAICLR | 4004 001Ch | SSP A Interrupt Clear Register | RW | 0000 0000h |
| SSPASSCR | 4004 0028h | SSP A Slave Select Configuration Register | RW | 0000 0000h |

26.10 SSP Register Detail

26.10.1 SSPCCON

Register 26-11 SSPCCON (SSP C Control Register, 4004 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | LSBFIRST | RW | 0 | Endian Order: 0b: MSB transmit / receive first 1b: LSB transmit / receive first |
| 12 | LBM | RW | 0 | Loopback Mode: 0b: Normal operation 1b: Loopback mode. Serial Input is taken from the serial output (MOSI or MISO) rather than the serial in (MISO or MOSI respectively) |
| 11 | SSPEN | RW | 0 | SSP Enable: 0b: SSP Controller is disabled 1b: SSP Controller is enabled. It will interact with other devices on the serial bus. Software should write the appropriate control information to the other SPI/SSP registers and interrupt controller registers, before setting this bit. |
| 10 | MS | RW | 0 | Master/Slave Mode: 0b: The SSP controller acts as a bus master, driving the USCCLK, USCMOSI and USCSS signals 1b: The SSP controller acts as a bus slave, driving the USCMISO and receiving USCCLK, USCMOSI and USCSS. |
| 9 | SOD | RW | 0 | Slave Output Disable: 0b: The SSP can drive the USCMISO output in slave mode 1b: The SSP must not drive the USCMISO output in slave mode |
| 8 | CPH | RW | 0 | Clock Out Phase. This bit is only used when SSPCCON.FRF = 0b (SPI). 0b: The SSP controller captures serial data on the first edge transition of the frame. 1b: The SSP controller captures serial data on the second edge transition of the frame. |
| 7 | CPO | RW | 0 | Clock Out Polarity (this bit is only used in SPI mode when SSPCCON.FRF = 00b) 0b: The clock is active high. 1b: The clock is active low. |
| 6:5 | FRF | RW | 0 | Frame Format: 00b: SPI 01b: TI 10b: Microwire 11b: Reserved |
| 4:0 | DSS | RW | 0 | Data Size Select: |

| | | | | |
|--|--|--|--|--|
| | | | | 0 0000b: Reserved 0 0001b: Reserved 0 0010b: Reserved 0 0011b: 4-bit transfer 0 0100b: 5-bit transfer 0 0101b: 6-bit transfer 0 0110b: 7-bit transfer 0 0111b: 8-bit transfer 0 1000b: 9-bit transfer 0 1001b: 10-bit transfer 0 1010b: 11-bit transfer 0 1011b: 12-bit transfer 0 1100b: 13-bit transfer 0 1101b: 14-bit transfer 0 1110b: 15-bit transfer 0 1111b: 16-bit transfer ... 1 1111b: 32-bit transfer |
|--|--|--|--|--|

26.10.2 SSPSTAT

Register 26-12 SSPSTAT (SSP C Status Register, 4004 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | BSY | RO | 0 | Busy Bit: 0b: SPI controller is idle 1b: SPI controller is sending/receiving a frame and/or the TX FIFO is not empty |
| 3 | RFF | RO | 0 | Receive FIFO Full: 0b: RX FIFO not full 1b: RX FIFO full |
| 2 | RNE | RO | 0 | Receive FIFO not empty: 0b: RX FIFO empty 1b: RX FIFO not empty |
| 1 | TNF | RO | 0 | Transmit FIFO not full: 0b: TX FIFO full 1b: TX FIFO not full |
| 0 | TFE | RO | 0 | Transmit FIFO empty: 0b: TX FIFO not empty 1b: TX FIFO empty |

26.10.3 SSPCDAT

Register 26-13 SSPCDAT (SSP C Data Register, 4004 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------|--------|-------|---|
| 31:0 | DATA | RW | 0 | <p>Software can writing data to be sent in a future frame to this register when the SSPCSTAT.TNF bit is set to 1b, indicating that the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise, the data written to this register will be sent as soon as all previous data has been sent (and received). If the data length is less than 32-bits, the data must be right-justified in this register.</p> <p>Software may read data from this register when the SSPCSTAT.RNE bit is set to a 1b, indicating that the RX FIFO is not empty. When software reads this register, the SPI controller returns the data from the least recent frame in the RX FIFO. If the data length is less than 32-bits, the data is right-justified in this field with the MSBs set to 0.</p> |

26.10.4 SSPCCLK

Register 26-14 SSPCCLK (SSP C Clock Register, 4004 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:8 | M | RW | 0 | These fields may set the SPI master clock rate by the formula: $F_{SSPCLK} = PCLK / ((SSPCCLK.M + 1) * SSPCCLK.N)$ N must be an even value from 2 to 254. |
| 7:0 | N | RW | 0 | |

26.10.5 SSPCIMSC

Register 26-15 SSPCIMSC (SSP C Interrupt Mask Set and Clear Enable Register, 4004 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXIM | RW | 0 | Transmit FIFO Interrupt Mask: 0b: TX FIFO half-full or less condition interrupt disabled 1b: TX FIFO half-full or less condition interrupt enabled |
| 2 | RXIM | RW | 0 | Receive FIFO Interrupt Mask: 0b: RX FIFO half-full or more condition interrupt disabled 1b: TX FIFO half-full or more condition interrupt enabled |
| 1 | RTIM | RW | 0 | Receive Timeout Interrupt Mask: 0b: RX FIFO not empty and no read prior to timeout period interrupt disabled 1b: RX FIFO not empty and no read prior to timeout period interrupt enabled |
| 0 | RORIM | RW | 0 | Receive Overrun Interrupt Mask: 0b: RX FIFO written to while full condition interrupt disabled 1b: RX FIFO written to while full condition interrupt enabled |

26.10.6 SSPCRIS

Register 26-16 SSPCRIS (SSP C Raw Interrupt Status Register, 4004 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXRIS | RO | 1 | TX FIFO Raw Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXRIS | RO | 0 | RX FIFO Raw Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTRIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | RORIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

26.10.7 SSPCMIS

Register 26-17 SSPCMIS (SSP C Masked Interrupt Status Register, 4004 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXMIS | RO | 1 | TX FIFO Masked Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXMIS | RO | 0 | RX FIFO Masked Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTMIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | ROMIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

26.10.8 SSPICLR

Register 26-18 SSPICLR (SSP C Interrupt Clear Register, 4004 001Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:2 | Reserved | RO | 0 | Reserved |
| 1 | RTIC | RW | 0 | RX FIFO Timeout Interrupt Clear: 0b: No effect 1b: Clears the SSPCxIS.RTRIS interrupt flag This bit is self-clearing. |
| 0 | RORIC | RW | 0 | RX FIFO Overwrite Interrupt Clear: 0b: No effect 1b: Clears the SSPCxIS.RORIS interrupt flag This bit is self-clearing. |

26.10.9 SSPCSSCR

Register 26-19 SSPCSSCR (SSP C Slave Select Configuration Register, 4004 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|--------------------|--------|-------|---|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | SPHDONTCARE | RW | 0 | Slave Select Pull High: 0b: USCSS cannot pull high after frame transfer 1b: USCSS must pull high after frame transfer |
| 3 | SWSS | RW | 0 | Slave Select State: 0b: Set USCSS to low 1b: Set USCSS to high |
| 2 | SWSEL | RW | 0 | Slave Select Software Control: 0b: USCSS is automatically controlled by the SPI module 1b: USCSS is software controlled by SSPCSSCR.SWCS |
| 1:0 | SELSS | RW | 0 | Slave Select Signal Control: 00b: USCSS is enabled 01b: Reserved 10b: Reserved 11b: Reserved |

27 USART D

27.1 Overview

The PAC55XX family contains support for four Universal Synchronous Asynchronous Receive Transmit (USART) peripherals.

A USART is a serial communications engine that may be configured for UART or SSP through a mode selection register.

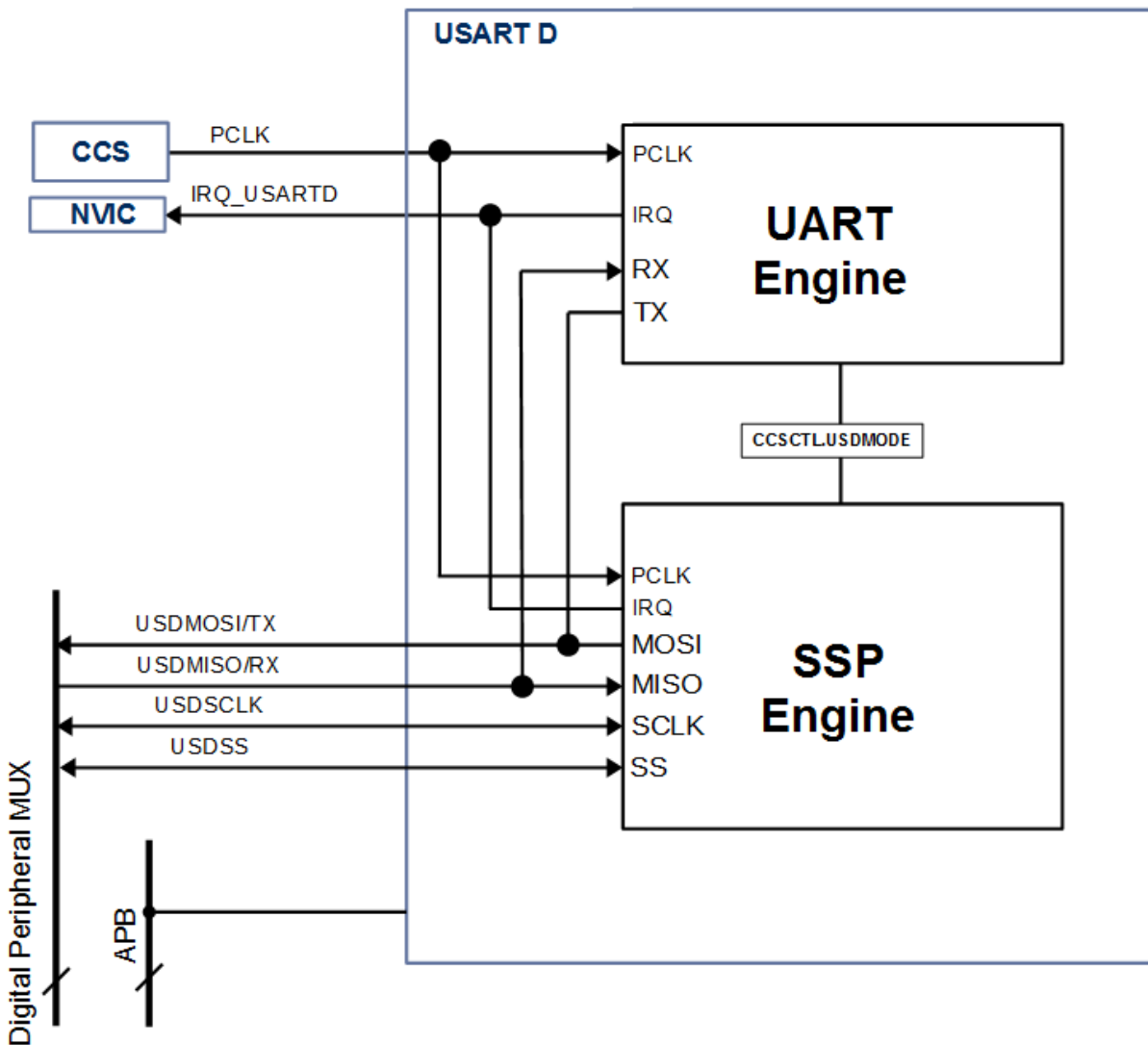
The USART is an APB client and is clocked by the Peripheral Clock (PCLK) system clock.

27.2 Features

- **UART Mode:**
 - 5-bit to 8-bit data interface
 - Optional parity
 - 1, 1.5 or 2 stop bits
 - 16-bit programmable baud-rate generator
 - 8-bit scratch pad
 - Independent RX and TX FIFOs
 - Configurable RX and TX interrupts
- **SSP Mode:**
 - Motorola SPI, TI Synchronous serial or National Semiconductor Microwire support
 - Master and Slave mode support
 - Independent RX and TX FIFOs
 - Configurable clock pre-scaler
 - Programmable Interrupts

27.3 System Block Diagram

Figure 27-1 USART D System Block Diagram



27.4 Functional Description

The USART may be configured for either UART or SSP mode via the **CCSCTL.USDMODE** register field.

In UART mode, the peripheral can receive or transmit UART serial data over the USDMOSI/TX (UART TX) or USDMISO/RX serial lines. In this mode, the USART signals for USDCLK and USBSS are unused. The UART may assert the IRQ_USARTD signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In UART mode, only the UART registers may be accessed, starting at register offset 0.

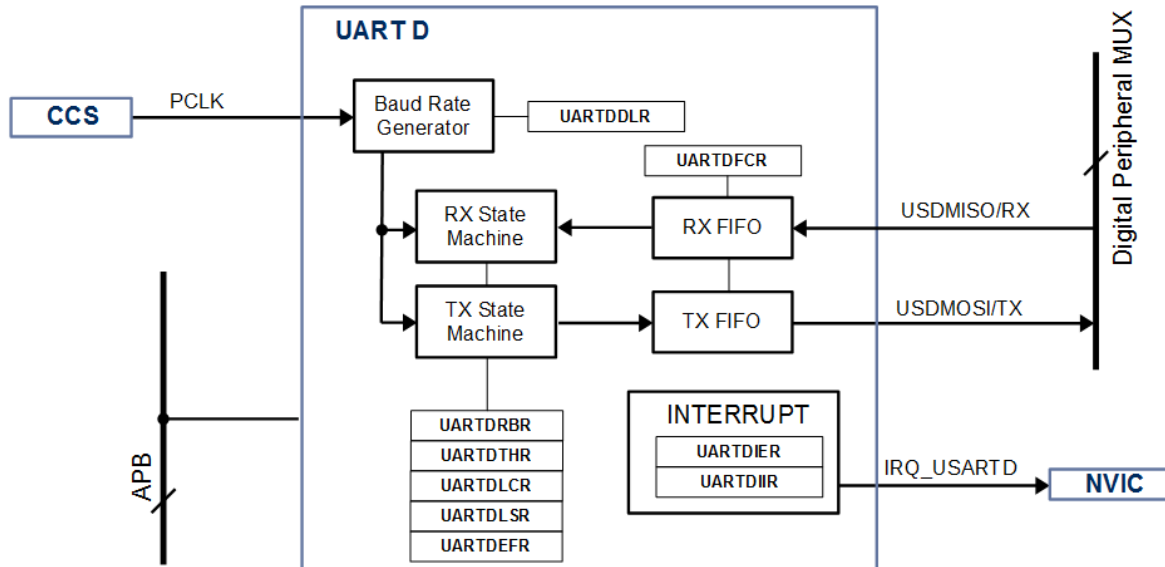
In SSP mode, the peripheral can receive or transmit serial data over the USDMOSI/TX (SPI MOSI) or USDMISO/RX (SPI MISO) serial lines. The USDCLK and USDSS are used by the peripheral in either master or slave mode for the SSP clock and Slave Select signals. The SSP may assert the IRQ_USARTD signal to the NVIC for interrupts, and is clocked using the PCLK system clock and connected to the APB bus.

In SPI mode, only the SPI registers may be accessed, starting at register offset 0.

27.5 UART Mode Functional Description

27.5.1 UART D Mode System Block Diagram

Figure 27-2 UART D Mode System Block Diagram



27.5.2 Mode Configuration

The UART mode of the USART must first be selected by writing **CCSCTL.USDMODE** to 1b. This will set the USART into UART mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USDMODE** is 1b are below.

Table 27-1 USART D UART Mode Registers

| REGISTER | DESCRIPTION |
|------------------|--|
| UARTDRBR | UART D Receive Buffer Register |
| UARTDTHR | UART D Transmit Holding Register |
| UARTDDLRL | UART D Divisor Latch Register |
| UARTDIER | UART D Interrupt Enable Register |
| UARTDIIR | UART D Interrupt Identification Register |
| UARTDFCR | UART D FIFO Control Register |
| UARTDLCR | UART D Line control Register |
| UARTDLSR | UART D Line Status Register |
| UARTDSCR | UART D Scratch Pad Register |

27.5.3 Baud Rate Configuration

To configure the baud rate for RX/TX operations, the **UARTDDLRL** may be used.

To baud rate may be configured by the following formula:

$$\text{Baud Rate} = \text{PCLK} / (16 * \text{UARTDDLRL})$$

Note that UART peripherals sometimes need an accurate timing base in order to function properly. Because of that, it is suggested that the CLKREF, EXTCLK or XTALCLK be used as the clock source for PCLK when using the UART.

The ROSCCLK may not provide enough accuracy for UART applications.

27.5.4 FIFO Reset

The RX and TX FIFOs may be reset independently by using the **UARTDFC** register.

To change any of the bits in the **UARTDFCR**, the **UARTDFCR.FIFOEN** must be set to 1b (**UARTDFCR** access).

To reset the RX FIFO, write the **UARTDFCR.RXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the RX state machine.

To reset the TX FIFO, write the **UARTDFCR.TXFIFORST** bit to 0b. This will clear all bytes in the FIFO and reset the pointer to the top of the FIFO for the TX state machine.

When the **UARTDFCR** register is done being used to configure the FIFOs, set the **UARTDFCR.FIFOEN** to 1b (application mode).

27.5.5 UART Configuration

In UART mode, the UART may be have the parity, break control, stop bits and word length configured by the user.

The parity checking may be enabled or disabled. To enable parity checking, set the **UARTDLCR.PEN** to 1b. To disable, set this field to 0b. If enabled, the user may configure the type of parity checking as follows.

Table 27-2 UART D Parity Modes

| UARTDLCR.PEN | UARTDLCR.PSEL | Parity | Description |
|--------------|---------------|----------------|--|
| 0b | n/a | None | |
| 1b | 00b | Odd | The number of ones in the transmitted character plus the attached parity bit will be odd. |
| | 01b | Even | The number of ones in the transmitted character plus the attached parity bit will be even. |
| | 10b | Forced 1 stick | Force 1 for parity |
| | 11b | Forced 0 stick | Force 0 for parity |

To set the number of stop bits, the user may write the **UARTDLCR.SBS** field. To select 1 stop bits, set **UARTDLCR.SBS** to 0b. To select 2 stop bits, set **UARTDLCR.SBS** to 1b. If the **UARTDLCR.WLS** is set to 00b (5-bit character length) and **UARTDLCR.SBS** is set to 1b, then there will be 1.5 stop bits.

To configure break control, the user can set the **UARTDLCR.BCON** field. To disable break transmission, set **UARTDLCR.BCON** to 0b. To force the TX signal to logic 0, set the **UARTDLCR.BCON** to 1b.

To set the word length for the UART, the user can set the **UARTDLCR.WLS** field as shown below.

Table 27-3 UART D Word Length

| UARTDLCR.WLS | UART word length |
|--------------|------------------|
| 00b | 5-bit |
| 01b | 6-bit |
| 10b | 7-bit |
| 11b | 8-bit |

27.5.6 UART Scratch Pad

There is an 8-bit general purpose register that may be used in the USART.

The user may use the **USDSCR** register for an 8-bit scratch pad.

27.5.7 UART Interrupts

The status of interrupts may be read at any time by the **UARTDIIR** register. If the **UARTDIIR.INTSTATUS** field is set to 0b, at least one interrupt is pending. The interrupt type can be read by the **UARTDIIR.INTID** field as shown below.

Table 27-4 UART D Interrupt ID

| UARTDIIR.INTID | Interrupt Type |
|----------------|---------------------------------|
| 000b | Reserved |
| 001b | TX Holding Register Empty |
| 010b | Receive Data Available |
| 011b | Receive Line Status |
| 100b | Reserved |
| 101b | Reserved |
| 110b | Receive FIFO character time-out |
| 111b | Reserved |

27.5.8 Transmit Operation

Before any data operation, the user must be sure to set the **UARTDFCR.FIFOEN** to 0b (normal mode).

Transmission may be initiated by writing **UARTDTHR** with the data desired to be sent. The data will be written into the TX FIFO and then transferred to the TX shift register together with a start bit, parity and stop bits as configured in the **UARTDLCR** register, LSB first.

The baud rate used will be set by the **UARTDDLRLR** register.

27.5.9 Transmit Interrupts

The UART may be configured to generate interrupts after characters have been successfully transmitted. The UART may be configured to generate an interrupt after 1, 4, 8 or 14 characters have been transmitted. By default, the interrupt is configured for 1 character.

To set the interrupt threshold for 4, 8 or 14 characters, the user should set the **UARTDEFR.ENMODE** = 1b and set the **UARTDFCR.TXTL** to the interrupt FIFO depth that is desired.

To enable transmit interrupts, set the **UARTDIER.THREIE** (TX holding register interrupt enable). When the transmit holding register has been emptied and the TX FIFO is empty, the **UARTDLSR.THRE** will be set to 1b. This field will be cleared the next time the **UARTDTHR** register is written with data to transmit.

If the **UARTDIER.THREIE** is set to 1b and the **UARTDLSR.THRE** interrupt flag is set, the **IRQ_USARTD** signal to the NVIC will be asserted.

27.5.10 Receive Operation

Before any data operation, the user must be sure to set the **UARTDFCR.FIFOEN** to 0b (normal mode).

Data is sampled into the RX shift register at a sampling rate of $PCLK / 16$. A filter is used to remove spurious inputs that last for less than two periods of the sampling rate. When the complete character has been clocked into the receiver, it is loaded into the RX FIFO. The UART RX FIFO is 16 bytes deep. The first entry into the RX FIFO will be available to read from the **UARTDRBR** register.

The receiver also checks the parity and stop bits as specified by the **UARTDLCR** register.

27.5.11 Receive Interrupts

The UART may be configured to generate interrupts after a character has been successfully received.

To enable receive interrupts, set the **UARTDIER.RBRIE** (RX buffer register interrupt enable). When the **UARTDRBR** (receive buffer register) has been filled from the RX FIFO, the **UARTDLSR.RDR** will be set to 1b. This field will be cleared the next time the **UARTDRBR** register is read.

If the **UARTDIER.RBRIE** is set to 1b and the **UARTDLSR.RDR** interrupt flag is set, the **IRQ_USARTD** signal to the NVIC will be asserted.

27.6 UART Register Summary

Table 27-5 USART D Register Summary (UART Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|----------|------------|--|--------|------------|
| UARTARBR | 4005 0000h | UART A Receive Buffer Register | RO | -- |
| UARTATHR | 4005 0004h | UART A Transmit Holding Register | WO | -- |
| UARTADLR | 4005 0008h | UART A Divisor Latch Register | RW | 0000 0001h |
| UARTAIER | 4005 000Ch | UART A Interrupt Enable Register | RW | 0000 0000h |
| UARTAIIR | 4005 0010h | UART A Interrupt Identification Register | RO | 0000 0001h |
| UARTAFCR | 4005 0014h | UART A FIFO Control Register | RW | 0000 0000h |
| UARTALCR | 4005 0018h | UART A Line control Register | RW | 0000 0000h |
| UARTALSR | 4005 0020h | UART A Line Status Register | RO | 0000 0060h |
| UARTASCR | 4005 0028h | UART A Scratch Pad Register | RW | -- |
| UARTAEFR | 4002 002Ch | UART A Enhanced Mode Register | RW | 0000 000h |

27.7 UART Register Detail

27.7.1 UARTDRBR

Register 27-1 UARTDRBR (UART D Receive Buffer Register, 4005 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | RBR | RO | 0 | Contains the oldest received character in the UART RX FIFO. |

27.7.2 UARTDTHR

Register 27-2 UARTDTHR (UART D Transmit Holding Register, 4005 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | THR | WO | - | Writing to the UARTBTHR causes the data to be stored in the UART transmit FIFO. The character will be sent when it reaches the bottom of the FIFO and the transmitter is available. |

27.7.3 UARTDDLRL

Register 27-3 UARTDDLRL (UART D Divisor Latch Register, 4005 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|------------|---|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:0 | DLR | RW | 0000 0001b | Sets the baud rate for the module. Baud rate = PCLK / (16 * UARTDDLRL) |

27.7.4 UARTDIER

Register 27-4 UARTDIER (UART D Interrupt Enable Register, 4005 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:3 | Reserved | RO | 0 | Reserved |
| 2 | RLSIE | RW | 0 | RX Line Status Interrupt Enable. Enables the UART RX line status interrupts. The status of this interrupts can be read from UARTDLSR[4:1] . 0b: Disable the RX line status interrupts 1b: Enable the RX line status interrupts |
| 1 | THRIE | RW | 0 | TX Holding Register Empty Interrupt Enable. Enables the THRE interrupt for the UART. The status of this interrupt can be read from UARTDLSR.THRE . 0b: Disable the THRE interrupts 1b: Enable the THRE interrupts |
| 0 | RBRIE | RW | 0 | RX Buffer Register Interrupt Enable. Enables the Receive Data Available interrupt for the UART. It also controls the character receive time-out interrupt. 0b: Disable the RBR interrupts 1b: Enable the RBR interrupts |

27.7.5 UARTDIIR

Register 27-5 UARTDIIR (UART D Interrupt Identification Register, 4005 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3:1 | INTID | RO | 000b | Interrupt identification: 000b: Reserved 001b: TX Holding Register Empty 010b: Receive Data Available 011b: Receive Line Status 100b: Reserved 101b: Reserved 110b: Receive FIFO Character Time-out 111b: Reserved |
| 0 | INTSTATUS | RO | 1b | Interrupt status. Note that this bit is active low. The pending interrupt can be determined through the UARTDIIR.INTID field. 0b: At least one interrupt is pending 1b: No interrupt is pending |

Note that this register is cleared on read.

27.7.6 UARTDFCR

Register 27-6 UARTDFCR (UART D FIFO Control Register, 4005 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:6 | RXTL | RW | 0 | RX Trigger Level. These two bits determine how many receive UART FIFO characters must be written before an interrupt is activated. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 5:4 | TXTL | RW | 0 | TX Trigger Level. These two bits determine how may transmit UART FIFO characters must be written before an interrupt is activated. This field cannot be set unless UARTDEFR.ENMODE = 1b. 00b: 1 character 01b: 4 characters 10b: 8 characters 11b: 14 characters |
| 3 | Reserved | RO | 0 | Reserved |
| 2 | TXFIFORST | RW | 0 | TX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART TX FIFO. This bit is self-clearing |
| 1 | RXFIFORST | RW | 0 | RX FIFO Reset. 0b: No effect 1b: Clears all bytes in the UART RX FIFO. This bit is self-clearing |
| 0 | FIFOEN | RW | 0 | FIFO Enable: 0b: UART FIFOs are disabled. Reset UARTAFCR settings to their default values. 1b: UART RX and TX FIFOs are enabled. UARTDFCR[7:1] are accessible when this bit is set. Any transition on this bit will automatically clear the UART RX AND TX FIFOs. |

27.7.7 UARTDLCR

Register 27-7 UARTDLCR (UART D Line Control Register, 4005 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6 | BCON | RW | 0 | Break Control: 0b: Disable break transmission 1b: Enable break transmission. Output pin UARTDTXD is forced to logic 0 when this bit is set. |
| 5:4 | PSEL | RW | 0 | Parity Select: 00b: Odd parity. The number of 1s in the transmitted character and the attached parity will be odd. 01b: Even parity. The number of 1s in the transmitted character and the attached parity will be even. 10b: Forced 1 stick parity. 11b: Forced 0 stick parity. |
| 3 | PEN | RW | 0 | Parity Enable: 0b: Disable parity generation and checking 1b: Enable parity generation and checking |
| 2 | SBS | RW | 0 | Stop Bit Select: 0b: 1 stop bit 1b: 2 stop bits, 1.5 stop bits if (UARTDLCR.WLS = 00b) |
| 1:0 | WLS | RW | 0 | Word Length Select: 00b: 5-bit character length 01b: 6-bit character length 10b: 7-bit character length 11b: 8-bit character length |

27.7.8 UARTDLSR

Register 27-8 UARTDLSR (UART D Line Status Register, 4005 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | RXFE | RO | 0 | <p>Error in RX FIFO.</p> <p>This bit is set when a character with a RX error such as framing, parity or break is loaded into UARTDRBR. This bit is cleared when the UARTDLSR register is read, and there are no subsequent errors in the UART FIFO.</p> |
| 6 | TEMT | RO | 0 | <p>Transmitter Empty.</p> <p>This bit is set when both UARTDTHR is empty; this bit is cleared when THR contains valid data.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 5 | THRE | RO | 0 | <p>Transmitter Holding Register Empty.</p> <p>This bit is set immediately upon detection of an empty UART THR and is cleared on a THR write.</p> <p>0b: THR contains valid data 1b: THR is empty</p> |
| 4 | BI | RO | 0 | <p>Break Interrupt.</p> <p>When UARTDRX is held in the spacing state (all zeroes) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until UARTDRX goes to the marking statue (all ones). A read of the UARTDLSR register clears this status bit. The time of break detection is dependents on UARTDFCR.FIFOEN.</p> <p>The break interrupt is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Break interrupt status is inactive 1b: Break interrupt status is active</p> |
| 3 | FE | RO | 0 | <p>Framing Error.</p> <p>When the stop bit of a received character is logic 0, a framing error occurs. Reading the UARTDLSR register will clear this bit. The time of the framing error is dependent upon UARTDFCR[3]. Upon detection of a framing error, the RX will attempt to re-synchronize the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no framing error.</p> <p>A framing error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Framing error status is inactive 1b: Framing error status is active</p> |
| 2 | PE | RO | 0 | <p>Parity Error.</p> <p>When the parity bit of a received character is in the wrong</p> |

| | | | | |
|---|------------|----|---|---|
| | | | | <p>state, a parity error occurs. Reading the UARTDLSR register will clear this bit. The time of the parity error detection is dependent on UARTDFCR.FIFOEN.</p> <p>A parity error is associated with the character at the top of the UART RBR FIFO.</p> <p>0b: Parity error status is inactive 1b: Parity error status is active</p> |
| 1 | OE | RO | 0 | <p>Overrun Error.</p> <p>The overrun error condition is set as soon as it occurs. Reading the UARTDLSR register will clear this bit. This bit is set when the UART RSR has a new character assembled and the UART RBR FIFO is full. In this case, the UART RBR FIFO will not be overwritten and the character in the UART RSR will be lost.</p> <p>0b: Overrun error status is inactive 1b: Overrun error status is active</p> |
| 0 | RDR | RO | 0 | <p>Receiver Data Ready.</p> <p>This bit is set when the RBR holds an unread character and is cleared when the UART RBR FIFO is empty.</p> <p>0b: RBR is empty 1b: RBR contains valid data</p> |

27.7.9 UARTDSCR

Register 27-9 UARTDSCR (UART D Scratch Pad Register, 4005 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|-------------------------------|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | PAD | RW | - | A read-able, write-able byte. |

27.7.10 UARTDEFR

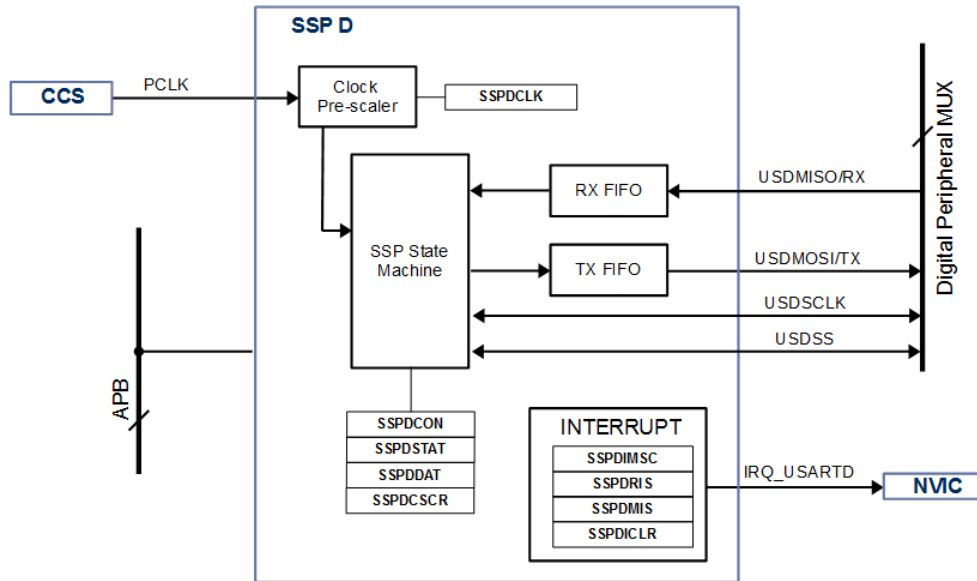
Register 27-10 UARTDEFR (UART D Enhanced Feature Register, 4005 002Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | ENMODE | RW | 0 | Enhanced Mode 0b – disabled 1b – enabled |
| 3:0 | Reserved | RO | 0 | Reserved |

27.8 SSP Functional Description

27.8.1 SSP Mode System Block Diagram

Figure 27-3 SSP D System Block Diagram



27.8.2 Mode Configuration

The SSP mode of the USART must first be selected by writing **CCSCTL.USDMODE** to 0b. This will set the USART into SSP mode and allow all configuration registers to be used in this mode.

The configuration and status registers that are available when **CCSCTL.USDMODE** is 0b are below.

Table 27-6 USART D UART Mode Registers

| REGISTER | DESCRIPTION |
|-----------------|---|
| SSPDCON | SSP D Control Register |
| SSPDSTAT | SSP D Status Register |
| SSPDDAT | SSP D Data Register |
| SSPDCLK | SSP D Clock Control Register |
| SSPDIMSC | SSP D Interrupt Mask Set and Clear Register |
| SSPDRIS | SSP D Raw Interrupt Status Register |
| SSPDMIS | SSP D Masked Interrupt Status Register |
| SSPDICLR | SSP D Interrupt Clear Register |
| SSPDSSCR | SSP D Slave Select Configuration Register |

27.8.3 SSP Overview

When in SSP mode, the USART may be a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- Motorola-style SPI
- TI-style Synchronous Serial Interface
- National Semiconductor-style Microwire

In both master and slave modes, the SSP performs parallel to serial conversion on data into 32-bit wide, 8-location deep RX and TX FIFOs.

The SSP may be configured to generate interrupts for servicing the TX and RX FIFOs and error conditions such as FIFO overwrite and timeout.

27.8.4 Clock Configuration

When configured for master mode, the SSP peripheral may configure the clock pre-scaler to generate the desired SCLK output clock frequency. The input clock on the SSP peripheral is the PCLK (peripheral clock) system clock.

The SCLK is generated from the PCLK input and the **SSPDCLK.M** and **SSPDCLK.N** parameters. The SCLK clock frequency is generated according to the following formula:

$$F_{\text{SSPCLK}} = \text{PCLK} / ((\text{SSPDCLK.M} + 1) * \text{SSPDCLK.N})$$

In this formula, **SSPDCLK.N** must be an even value from 2 to 254.

27.8.5 SSP Clock Constraints

The SSP clock configuration must be configured according to the guidelines below, in order for the peripheral to operate correctly.

Here are the clock constraints for the input clock SSPCLK (after the pre-scaler) and output SSP clock, USDCLK:

- When in master mode, the minimum frequency of the input SSPCLK must be at least 2X the output SSP clock (USDCLK)
- When in slave mode, the minimum frequency of the input SSPCLK must be at least 12X the input SSP clock (USDCLK)

The maximum frequency of the SSPCLK should be:

- When in master mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USDCLK}}$ (SSP clock output)
- When in slave mode, $F_{\text{SSPCLK}}(\text{max}) \leq 254 * 128 * F_{\text{USDCLK}}$ (SSP clock input)

27.8.6 SSP Configuration

The SSP may be configured to support various features of master and slave based serial interfaces.

To enable the SSP controller, set **SSPDCON.SSPEN** to 1b. When disabled, the SSP may still have registers configured for the configuration mode and interrupts, but will not operate on serial data until enabled. Once the SSP has been configured, set this bit to 1b to begin processing.

To configure the SSP mode, set **SSPDCON.FRFB** (frame format) to the desired value as shown below.

Table 27-7 SSP D Frame Format

| SSPDCON.FRFB | Frame Format |
|--------------|------------------------------------|
| 00b | SPI (Motorola) |
| 01b | Synchronous Serial Format (TI) |
| 10b | Microwire (National Semiconductor) |
| 11b | Reserved |

To select master mode, set **SSPDCON.MS** to 0b. In this mode, the controller is the bus master. It will drive the SCLK, MOSI and SS signals and will receive data on the MISO line.

To select slave mode, set **SSPDCON.MS** to 1b. In this mode, the controller is the bus slave and will drive the MISO line and receive input from SCLK, MOSI and SS.

To change the serial to parallel endian order, the user may use the **SSPDCON.LSBFIRST** field. To configure the SSP for LSB first, set **SSPDCON.LSBFIRST** to 1b. To configure the SSP for MSB first, set **SSPDCON.LSBFIRST** to a 0b.

To enable loop-back mode, set the **SSPDCON.LBM** to 1b. In this mode, serial input is taken from the serial output instead of from the digital peripheral MUX.

While in slave mode, the behavior of the MISO output can be controlled. When **SSPDCON.SOD** is set to 0b, the SSP can drive the MISO in slave mode. When this bit is 1b, the SSP will not drive the MISO output in slave mode.

The clock phase of the SSP may also be configured. To configure the SSP for capturing data on the first clock edge transition, set **SSPDCON.CPH** to 0b. To configure the SSP for capturing data on the second clock edge transition, set **SSPDCON.CPH** to 1b.

When the **SSPDCON.FRFB** is 00b (SPI), the clock output polarity may be configured. To configure the SSP clock to be active high, set **SSPDCON.CPO** to 0b. To configure the SSP clock to be active low, set **SSPDCON.CPO** to 1b.

The SSP may also configure the size of the data word size transferred in each frame. To configure the data size, see the table below.

Table 27-8 SSP C Data Size Select

| SSPDCON.DSS | Data Size |
|-------------|-----------|
| 0000b | Reserved |
| 0001b | Reserved |
| 0010b | Reserved |
| 0011b | 4-bit |
| 0100b | 5-bit |
| 0101b | 6-bit |
| 0110b | 7-bit |
| 0111b | 8-bit |
| 1000b | 9-bit |
| 1001b | 10-bit |
| 1010b | 11-bit |
| 1011b | 12-bit |
| 1100b | 13-bit |
| 1101b | 14-bit |
| 1110b | 15-bit |
| 1111b | 16-bit |

27.8.7 SSP Slave Select Configuration

The SSP allows the behavior of the SS (slave select) signal to be configured by using the **SSPDSSCR** register as follows when configured as a bus master.

The **SSPDSSCR.SELCS** field must always be set to a 00b for proper operation.

To configure the SSP to automatically control the behavior of the SS signal, set **SSPDSSCR.SWSEL** to 0b. If the user wants to control the SS signal by software, set **SSPDSSCR.SWSEL** to 1b.

To configure the SS signal behavior after a frame transfer, use the **SSPDSSCR.SPHDONTCARE** field. To configure the SSP to not pull the SS signal high after a frame transfer, set **SSPDSSCR.SPHDONTCARE** to 0b. To configure the SSP to pull the SS signal high after a frame transfer, set the **SSPDSSCR.SPHDONTCARE** to 1b.

27.8.8 SSP Interrupts

The SSP may configure interrupts for the following conditions.

When the SSP detects that the TX FIFO is half-full or less, the **SSPDRIS.TXRIS** and **SSPDMIS.TXMIS** are set to 1b. During this condition, if the **SSPDIMSC.TXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTD signal to the NVIC.

When the SSP detects that the RX FIFO is half-full or more, the **SSPDRIS.RXRIS** and **SSPDMIS.RXMIS** are set to 1b. During this condition, if the **SSPDIMSC.RXIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTD signal to the NVIC.

The SSP contains a time for reading the RX FIFO. If the RX FIFO is not empty, and $32 F_{SSPCLK}$ periods have gone by, the **SSPDRIS.RTIM** and **SSPDMIS.RTMIS** bits are set to 1b. During this condition, if the **SSPDIMSC.RTIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTD signal to the NVIC. This condition may be cleared by writing **SSPDICLR.RTIC** to 1b.

If the SSP detects the RX FIFO is full, when a character is attempted to be inserted into it, it sets the **SSPDRIS.RORIM** and **SSPDMIS.ROMIS** to 1b. During this condition, if the **SSPDIMSC.RORIM** interrupt enable is set to 1b, then the SSP will assert the IRQ_USARTD signal to the NVIC. This condition may be cleared by writing **SSPDICLR.RORIC** to 1b.

27.8.9 Master Mode Operation

When configured as a bus master (**SSPDCON.MS** = 0b), the SSP will insert a character into the TX FIFO when it is written into the **SSPDDAT** register. The SSP will read a character from the TX FIFO and performs a parallel to serial conversion on it. Then the serial data stream and frame control signal is synchronized to the clock and are output through the USDMOSI/TX pin to the attached slaves. While the data is being transmit to the slaves, the USDCLK and USDSS behave as configured in the **SSPDCON** and **SSPDCSCR** registers.

The master receive logic performs serial to parallel conversion on the incoming synchronous USDMISO/RX data stream, and stores the character in the RX FIFO. The master may read first entry in the RX FIFO by reading the **SSPDDAT** register.

27.8.10 Slave Mode Operation

When configured as a bus slave (**SSPDCON.MS** = 1b), the SCLK is provided by the attached master. The user can write the next character to send into **SSPDDAT** and it will get inserted into the TX FIFO. The slave transmit logic reads a value from the TX FIFO, performs parallel to serial conversion and outputs the serial data stream on the USDMISO/RX pin to the attached master. The slave receive logic performs serial to parallel conversion on the incoming USDMOSI/TX data stream, extracting and storing values into the RX FIFO. The first character in the RX FIFO may be read by reading the **SSPDDAT** register.

27.8.11 SSP Status

The SSP module maintains status information on the operation of the module. The table below shows the available SSP status conditions.

Table 27-9 SSP D Status

| SSPDSTAT field | Description |
|----------------|--|
| BSY | If set, the SSP controller is currently sending/receiving a frame and/or the TX FIFO is not empty. |
| RFF | If set, the RX FIFO is full. |
| RNE | If set, the RX FIFO is not empty. |
| TNF | If set, the TX FIFO is full. |
| TNE | If set, the TX FIFO is not empty |

27.9 SSP Register Summary

Table 27-10 USART D Register Summary (SSP Mode)

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|-----------------|------------|---|--------|------------|
| SSPACON | 4005 0044h | SSP A Control Register | RW | 0000 0000h |
| SSPASTAT | 4005 0048h | SSP A Status Register | RO | 0000 0003h |
| SSPADAT | 4005 0008h | SSP A Data Register | RW | 0000 0000h |
| SSPACLK | 4005 000Ch | SSP A Clock Control Register | RW | 0000 0000h |
| SSPAIMSC | 4005 0010h | SSP A Interrupt Mask Set and Clear Register | RW | 0000 0000h |
| SSPARIS | 4005 0014h | SSP A Raw Interrupt Status Register | RO | 0000 0008h |
| SSPAMIS | 4005 0018h | SSP A Masked Interrupt Status Register | RO | 0000 0008h |
| SSPAICLR | 4005 001Ch | SSP A Interrupt Clear Register | RW | 0000 0000h |
| SSPASSCR | 4005 0028h | SSP A Slave Select Configuration Register | RW | 0000 0000h |

27.10 SSP Register Detail

27.10.1 SSPDCON

Register 27-11 SSPDCON (SSP D Control Register, 4005 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:14 | Reserved | RO | 0 | Reserved |
| 13 | LSBFIRST | RW | 0 | Endian Order: 0b: MSB transmit / receive first 1b: LSB transmit / receive first |
| 12 | LBM | RW | 0 | Loopback Mode: 0b: Normal operation 1b: Loopback mode. Serial Input is taken from the serial output (MOSI or MISO) rather than the serial in (MISO or MOSI respectively) |
| 11 | SSPEN | RW | 0 | SSP Enable: 0b: SSP Controller is disabled 1b: SSP Controller is enabled. It will interact with other devices on the serial bus. Software should write the appropriate control information to the other SPI/SSP registers and interrupt controller registers, before setting this bit. |
| 10 | MS | RW | 0 | Master/Slave Mode: 0b: The SSP controller acts as a bus master, driving the USDCLK, USDMOSI and USDSS signals 1b: The SSP controller acts as a bus slave, driving the USDMISO and receiving USDCLK, USDMOSI and USDSS. |
| 9 | SOD | RW | 0 | Slave Output Disable: 0b: The SSP can drive the USDMISO output in slave mode 1b: The SSP must not drive the USDMISO output in slave mode |
| 8 | CPH | RW | 0 | Clock Out Phase. This bit is only used when SSPDCON.FRF = 0b (SPI). 0b: The SSP controller captures serial data on the first edge transition of the frame. 1b: The SSP controller captures serial data on the second edge transition of the frame. |
| 7 | CPO | RW | 0 | Clock Out Polarity (this is only used in SPI mode when SSPDCON.FRF = 00b) 0b: The clock is active high. 1b: The clock is active low. |
| 6:5 | FRF | RW | 0 | Frame Format: 00b: SPI 01b: TI 10b: Microwire 11b: Reserved |
| 4:0 | DSS | RW | 0 | Data Size Select: |

| | | | | |
|--|--|--|--|--|
| | | | | 0 0000b: Reserved 0 0001b: Reserved 0 0010b: Reserved 0 0011b: 4-bit transfer 0 0100b: 5-bit transfer 0 0101b: 6-bit transfer 0 0110b: 7-bit transfer 0 0111b: 8-bit transfer 0 1000b: 9-bit transfer 0 1001b: 10-bit transfer 0 1010b: 11-bit transfer 0 1011b: 12-bit transfer 0 1100b: 13-bit transfer 0 1101b: 14-bit transfer 0 1110b: 15-bit transfer 0 1111b: 16-bit transfer ... 1 1111b: 32-bit transfer |
|--|--|--|--|--|

27.10.2 SSPDSTAT

Register 27-12 SSPDSTAT (SSP D Status Register, 4005 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | BSY | RO | 0 | Busy Bit: 0b: SPI controller is idle 1b: SPI controller is sending/receiving a frame and/or the TX FIFO is not empty |
| 3 | RFF | RO | 0 | Receive FIFO Full: 0b: RX FIFO not full 1b: RX FIFO full |
| 2 | RNE | RO | 0 | Receive FIFO not empty: 0b: RX FIFO empty 1b: RX FIFO not empty |
| 1 | TNF | RO | 0 | Transmit FIFO not full: 0b: TX FIFO full 1b: TX FIFO not full |
| 0 | TFE | RO | 0 | Transmit FIFO empty: 0b: TX FIFO not empty 1b: TX FIFO empty |

27.10.3 SSPDDAT

Register 27-13 SSPDDAT (SSP D Data Register, 4005 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------|--------|-------|---|
| 31:0 | DATA | RW | 0 | <p>Software can writing data to be sent in a future frame to this register when the SSPDSTAT.TNF bit is set to 1b, indicating that the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise, the data written to this register will be sent as soon as all previous data has been sent (and received). If the data length is less than 32-bits, the data must be right-justified in this register.</p> <p>Software may read data from this register when the SSPDSTAT.RNE bit is set to a 1b, indicating that the RX FIFO is not empty. When software reads this register, the SPI controller returns the data from the least recent frame in the RX FIFO. If the data length is less than 32-bits, the data is right-justified in this field with the MSBs set to 0.</p> |

27.10.4 SSPDCLK

Register 27-14 SSPDCLK (SSP D Clock Register, 4005 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|--|
| 31:16 | Reserved | RO | 0 | Reserved |
| 15:8 | M | RW | 0 | These fields may set the SPI master clock rate by the formula: $F_{SSPCLK} = PCLK / ((SSPDCLK.M + 1) * SSPDCLK.N)$ N must be an even value from 2 to 254. |
| 7:0 | N | RW | 0 | |

27.10.5 SSPDIMSC

Register 27-15 SSPDIMSC (SSP D Interrupt Mask Set and Clear Enable Register, 4005 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXIM | RW | 0 | Transmit FIFO Interrupt Mask: 0b: TX FIFO half-full or less condition interrupt disabled 1b: TX FIFO half-full or less condition interrupt enabled |
| 2 | RXIM | RW | 0 | Receive FIFO Interrupt Mask: 0b: RX FIFO half-full or more condition interrupt disabled 1b: TX FIFO half-full or more condition interrupt enabled |
| 1 | RTIM | RW | 0 | Receive Timeout Interrupt Mask: 0b: RX FIFO not empty and no read prior to timeout period interrupt disabled 1b: RX FIFO not empty and no read prior to timeout period interrupt enabled |
| 0 | RORIM | RW | 0 | Receive Overrun Interrupt Mask: 0b: RX FIFO written to while full condition interrupt disabled 1b: RX FIFO written to while full condition interrupt enabled |

27.10.6 SSPDRIS

Register 27-16 SSPDRIS (SSP D Raw Interrupt Status Register, 4005 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXRIS | RO | 1 | TX FIFO Raw Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXRIS | RO | 0 | RX FIFO Raw Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTRIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | RORIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

27.10.7 SSPDMIS

Register 27-17 SSPDMIS (SSP D Masked Interrupt Status Register, 4005 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:4 | Reserved | RO | 0 | Reserved |
| 3 | TXMIS | RO | 1 | TX FIFO Masked Interrupt Status: 0b: TX FIFO not at least half-empty 1b: TX FIFO at least half empty |
| 2 | RXMIS | RO | 0 | RX FIFO Masked Interrupt Status: 0b: RX FIFO not at least half-full 1b: RX FIFO at least half-full |
| 1 | RTMIS | RO | 0 | RX FIFO Timeout Interrupt Status: 1b: RX FIFO is not empty and has not been read for the timeout period. The timeout period is 32-bit times at F _{SSPCLK} |
| 0 | ROMIS | RO | 0 | RX FIFO Overwrite Interrupt Status: 1b: RX FIFO was full when another frame was completely received. The preceding data frame is overwritten by the new frame when this occurs. |

27.10.8 SSPDICLR

Register 27-18 SSPDICLR (SSP D Interrupt Clear Register, 4005 001Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:2 | Reserved | RO | 0 | Reserved |
| 1 | RTIC | RW | 0 | RX FIFO Timeout Interrupt Clear: 0b: No effect 1b: Clears the SSPDxIS.RTRIS interrupt flag This bit is self-clearing. |
| 0 | RORIC | RW | 0 | RX FIFO Overwrite Interrupt Clear: 0b: No effect 1b: Clears the SSPDxIS.RORIS interrupt flag This bit is self-clearing. |

27.10.9 SSPDSSCR

Register 27-19 SSPDSSCR (SSP D Slave Select Configuration Register, 4005 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-------------|--------|-------|---|
| 31:5 | Reserved | RO | 0 | Reserved |
| 4 | SPHDONTCARE | RW | 0 | Slave Select Pull High: 0b: USDSS cannot pull high after frame transfer 1b: USDSS must pull high after frame transfer |
| 3 | SWSS | RW | 0 | Slave Select State: 0b: Set USDSS to low 1b: Set USDSS to high |
| 2 | SWSEL | RW | 0 | Slave Select Software Control: 0b: USDSS is automatically controlled by the SPI module 1b: USDSS is software controlled by SSPDSSCR.SWCS |
| 1:0 | SELSS | RW | 0 | Slave Select Signal Control: 00b: USDSS is enabled 01b: Reserved 10b: Reserved 11b: Reserved |

28 I2C

28.1 Overview

The PAC55XX contains one I2C peripheral that supports version 2.1 of the I2C specification.

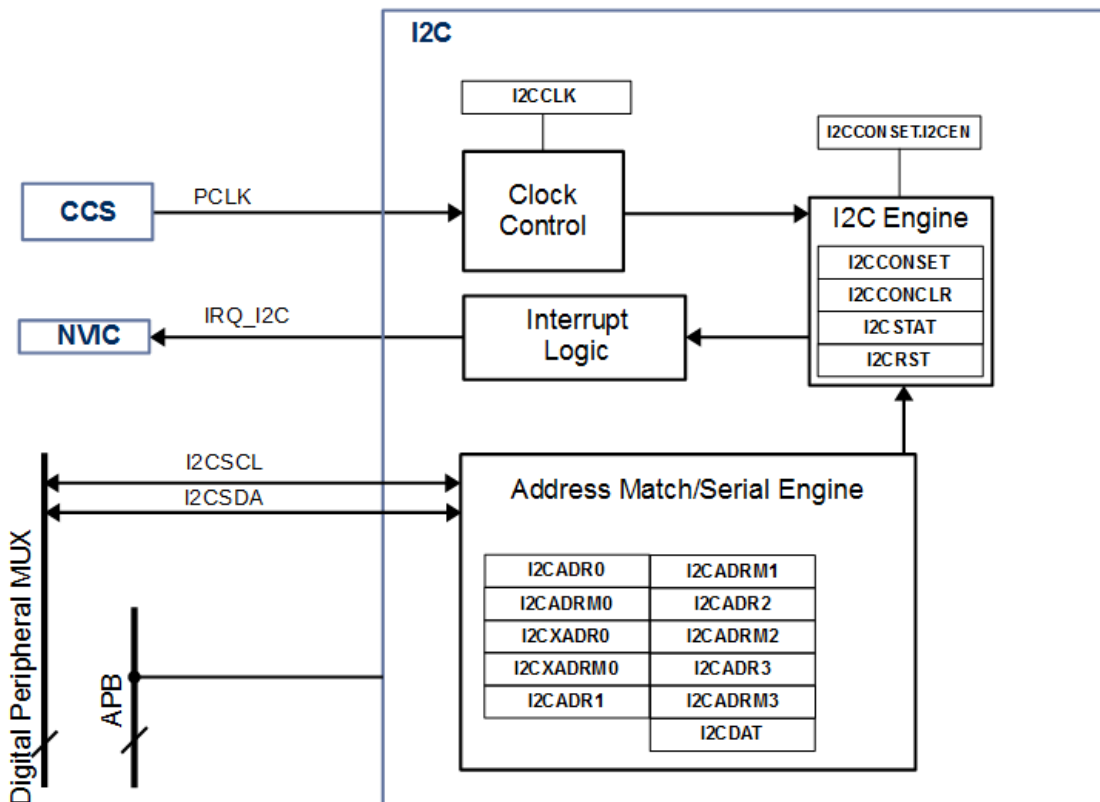
The I2C module is an APB client, and is clocked by the PCLK (peripheral clock) and can operate as a 7-bit or 10-bit master or slave controller.

28.2 Features

- I2C Master or slave peripheral
- 7-bit or 10-bit addressing modes
- Support High-speed, fast or standard speed transfer rates (3.4Mbps, 400kHz, 100kHz)
- Support for General Call Address detection
- Interrupt for transmit and receive data

28.3 System Block Diagram

Figure 28-1 I2C System Block Diagram



28.4 Functional Description

28.4.1 Enabling I2C

To enable the I2C module, set the **I2CCONSET.I2CEN** field to 1b. The module may be disabled by setting this field to a 0b. In addition, the user may set the **I2CCONCLR.I2CENC** bit to 1b to clear **I2CCONSET.I2CEN**. Writing a 0b to **I2CCONCLR.I2CENC** has no effect.

28.4.2 I2C Clocking

The I2C engine has its clock supplied from the peripheral clock (PCLK).

The I2C sampling frequency (F_{SAMP}) and the SCL clock period (F_{SCL}) may be configured using the I2CCLK register. These frequencies can be set as shown below:

- $F_{SAMP} = PCLK / (2^{I2CCLK.M})$
- $F_{SCL} = PCLK / (2^{I2CCLK.M} \times (I2CCLK.N + 1) \times 10)$

The table below shows some common configurations that can be used to generate F_{SAMP} and F_{SCL} .

Table 28-1 I2C F_{SAMP} and F_{CLK} configuration examples

| PCLK | I2CCLK.M | I2CCLK.N | F_{SAMP} | F_{SCL} |
|--------|----------|----------|------------|-----------|
| 48MHz | 2 | 2 | 12MHz | 400kHz |
| 80MHz | 2 | 1 | 20MHz | 1MHz |
| 136MHz | 2 | 0 | 34MHz | 3.4MHz |

28.4.3 I2C Master Write

To perform an I2C Write transaction as an I2C master, follow these steps:

- Write **I2CCONSET.STA** to create an I2C START condition
 - The **I2CSTAT** register will contain a 08h if successful
- Write **I2CDAT** with the I2C target address:
 - Bit 0 should contain a 0b for a write operation
 - If configured, the I2C engine should generate an interrupt after writing the address
 - The **I2CSTAT** register will contain a 18h if the address is successfully written and an ACK is received
 - The **I2CSTAT** register will contain a 20h if the address is successfully written and a NACK is received
- For each data byte to write onto the bus:
 - Write the **I2CDAT** register with the 8-bit data
 - If configured, the I2C engine should generate an interrupt after writing the data

- The **I2CSTAT** register will contain a 28h after the data is transmitted, and an ACK was received
- The **I2CSTAT** register will contain a 30h after the data is transmitted, and a NACK was received
- Write **I2CCONSET.STO** to create an I2C STOP condition
 - Note that this bit must be set before clearing the I2C interrupt flag (**I2CCONSET.SI**)

28.4.4 I2C Master Read¹⁴

- Write **I2CCONSET.STA** to create an I2C START condition
 - The **I2CSTAT** register will contain a 08h if successful
- Write **I2CDAT** with the I2C target address:
 - Bit 0 should contain a 0b for a read operation
 - If configured, the I2C engine should generate an interrupt after writing the address
 - The **I2CSTAT** register will contain a 40h if the address is successfully written and an ACK is received
 - The **I2CSTAT** register will contain a 48h if the address is successfully written and a NACK is received
- Write **I2CCONSET.STA** to create a repeated START condition
 - The I2CSTAT register will contain a 10h if the repeated START condition was successfully written
- Write **I2CDAT** with the I2C target address:
 - Bit 0 should contain a 1b for a read operation
 - If configured, the I2C engine should generate an interrupt after writing the address
 - The **I2CSTAT** register will contain a 40h if the address is successfully written and an ACK is received
 - The **I2CSTAT** register will contain a 48h if the address is successfully written and a NACK is received
- For each data byte to read from the bus:
 - If configured, wait for an I2C interrupt for a received data byte
 - Read the 8-bit data from the **I2CDAT** register
 - The **I2CSTAT** register will contain a 50h after the data is read from the bus, and an ACK was received
 - The **I2CSTAT** register will contain a 58h after the data is read from the bus, and a NACK was received
- Write **I2CCONSET.STO** to create an I2C STOP condition
 - Note that this bit must be set before clearing the I2C interrupt flag (**I2CCONSET.SI**)

¹⁴ For a single data read, a repeated start condition is not required.

28.4.5 I2C Slave Write

- Receive I2C interrupt upon received I2C slave address:
 - **I2CSTAT** will be 60h if matches slave address and write bit received, and ACK transmitted
- For every I2C interrupt after data byte received (for address phase):
 - **I2CSTAT** will be set to 80h, if ACK transmitted
 - **I2CSTAT** will be set to 88h, if NACK transmitted
- Set **I2CCONSET.AA** to 0b to NACK or 1b to ACK address
- **I2CSTAT** will be set to A0h if a repeated START was received
- For every I2C interrupt for every data byte received after address:
 - I2CSTAT will be set to 80h, if ACK transmitted
 - I2CSTAT will be set to 88h, if NACK transmitted
- Set **I2CCONSET.AA** to 0b to NACK or 1b to ACK data byte

28.4.6 I2C Slave Read

- Receive I2C interrupt upon received I2C slave address:
 - **I2CSTAT** will be A8h if matches slave address and read bit received, and ACK transmitted
- For every I2C interrupt after data byte received (for address phase):
 - **I2CSTAT** will be set to 80h, if ACK transmitted
 - **I2CSTAT** will be set to 88h, if NACK transmitted
- **I2CSTAT** will be set to A0h if a repeated START was received
- Write **I2CDAT** with data byte to write onto I2C bus
- For every I2C interrupt for every data byte received after address:
 - **I2CSTAT** will be set to 80h, if ACK transmitted
 - **I2CSTAT** will be set to 88h, if NACK transmitted

Set **I2CCONSET.AA** to 0b to NACK or 1b to ACK data byte

28.4.7 I2C Slave Addressing

The I2C engine allows up to 4 7-bit slave addresses and 1 10-bit slave address to be configured that may be used when the device is operating as a I2C slave. Each slave address register has a matching mask register that can be used to mask slave addresses with when checking for a match.

The table below shows which registers should be programmed to perform slave address matching for both 7-bit and 10-bit addressing modes.

Table 28-2 I2C Slave Address Matching

| Slave Address Register | Slave Address Mask | 7-bit Address | 10-bit Address |
|------------------------|--------------------|---------------|----------------|
| I2CADR0 | I2CADRM0 | X | |

| | | | |
|-----------------|-----------------|---|---|
| I2CADR1 | I2CADRM1 | X | |
| I2CADR2 | I2CADRM2 | X | |
| I2CXADR0 | I2CXADM0 | | X |

In each of the I2C slave address registers, **I2CxADR_x.GC** is the general call enable bit, and can be set if the registers should be configured to use the general call address.

28.4.8 I2C Interrupts

The I2C module has interrupts that may be enabled or disabled by the user. To enable interrupts, set the **I2CCONSET.I2CIE** to a 1b. To disable interrupts, set this field to a 0b. The user may also set the **I2CCONCLR.I2CIEC** bit to 1b to clear the **I2CCONSET.I2CIE** bit. Writing 0b to **I2CCONCLR.I2CIEC** has no effect.

The I2C module will generate an interrupt condition after any address or data byte is sent or received. During any of these conditions, the **I2CCONSET.SI** interrupt flag will be set. If the **I2CSETCON.I2CIE** interrupt enable bit is set to a 1b during any of these conditions, the IRQ_I2C signal to the NVIC will be asserted.

To clear the interrupt condition, write **I2CCONCLR.SIC** bit to a 1b will clear the interrupt flag in **I2CCONSET.SI** which will de-assert the IRQ_I2C signal to the NVIC.

28.4.9 I2C Status

The status of the I2C engine is kept in the **I2CSTAT** register. The contents of this register will show the status of the engine. The status codes for this register are shown below.

Table 28-3 I2C Status Codes

| I2CSTATUS Code | Description |
|----------------|--|
| 00h | Bus error (master mode only) |
| 08h | START condition transmitted |
| 10h | Repeated START condition transmitted |
| 18h | Address and Write bit transmitted, ACK received |
| 20h | Address and Write bit transmitted, NACK received |
| 28h | Data byte transmitted in master mode, ACK received |
| 30h | Data byte transmitted in master mode, NACK received |
| 38h | Arbitration lost in address or data byte |
| 40h | Address and read bit transmitted, ACK received |
| 48h | Address and read bit transmitted, NACK received |
| 50h | Data byte received in master mode, ACK transmitted |
| 58h | Data byte received in master mode, NACK transmitted |
| 60h | Slave address and write bit received, ACK transmitted |
| 68h | Arbitration lost in address as master, slave address and write bit received, ACK transmitted |
| 70h | General call address received, ACK transmitted |
| 78h | Arbitration lost in address as master, general call address received, ACK transmitted |
| 80h | Data byte received after slave address received, ACK transmitted |
| 88h | Data byte received after slave address received, NACK transmitted |
| 90h | Data byte received after general call address received, ACK transmitted |
| A0h | STOP or repeated START condition received in slave mode |
| A8h | Slave address and read bit received, ACK transmitted |
| B0h | Arbitration lost in address as master, slave address and read bit received, ACK transmitted |
| B8h | Data byte transmitted in slave mode, ACK received |
| C0h | Data byte transmitted in slave mode, NACK received |
| C8h | Last byte transmitted in slave mode, ACK received |
| D0h | Last byte transmitted in slave mode, NACK received |
| E0h | Second address byte transmitted, ACK received |
| E8h | Second address byte transmitted, NACK received |
| F8h | No relevant status information |

28.4.10 I2C Reset

The I2C module may have software reset applied to it. After a reset is commanded, all I2C state machines and registers will reset to their default values.

To execute a software reset of the module, write **I2CRST** with the value 0x07.

28.5 Register Summary

Table 28-4 I2C Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|------------------|------------|--|--------|------------|
| I2CCONSET | 4001 0000h | I2C Control Set Register | RW | 0000 0000h |
| I2CCONCLR | 4001 0004h | I2C Control Clear Register | WO | -- |
| I2CSTAT | 4001 0008h | I2C Status Register | RO | 0000 00F8h |
| I2CDAT | 4001 000Ch | I2C Data Register | RW | 0000 0000h |
| I2CCLK | 4001 0010h | I2C Clock Control Register | RW | 0000 0000h |
| I2CADR0 | 4001 0014h | I2C Slave Address Register 0 | RW | 0000 0000h |
| I2CADRM0 | 4001 0018h | I2C Slave Address Mask Register 0 | RW | 0000 00FEh |
| I2CXADR0 | 4001 001C | I2C Extended Slave Address Register 0 | RW | 0000 0000h |
| I2CXADM0 | 4001 0020h | I2C Extended Slave Address Mask Register 0 | RW | 0000 07FEh |
| I2CRST | 4001 0024h | I2C Software Reset Register | RW | 0000 0000h |
| I2CADR1 | 4001 0028h | I2C Slave Address Register 1 | RW | 0000 0000h |
| I2CADRM1 | 4001 002Ch | I2C Slave Address Mask Register 1 | RW | 0000 00FEh |
| I2CADR2 | 4001 0030h | I2C Slave Address Register 2 | RW | 0000 0000h |
| I2CADRM2 | 4001 0034h | I2C Slave Address Mask Register 2 | RW | 0000 00FEh |
| I2CADR3 | 4001 0038h | I2C Slave Address Register 3 | RW | 0000 0000h |
| I2CADRM3 | 4001 003Ch | I2C Slave Address Mask Register 3 | RW | 0000 00FEh |

28.6 Register Detail

28.6.1 I2CCONSET

Register 28-1 I2CCONSET (I2C Control Set, 4001 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:9 | Reserved | RO | 0 | Reserved |
| 8 | GCF | RO | 0 | General Call Flag: 0b: I2C general call address no match 1b: I2C general call address match |
| 7 | I2CIE | RW | 0 | Interrupt Enabled: 0b: I2C disabled 1b: I2C enabled |
| 6 | I2CEN | RW | 0 | I2C Enabled: 0b: I2C Disabled 1b: I2C Enabled |
| 5 | STA | W1C | 0 | START flag: 0b: No effect 1b: I2C enters master mode and sends START condition. This bit is self-clearing. |
| 4 | STO | W1C | 0 | STOP flag: 0b: No effect 1b: Send STOP condition (if in master mode) or behave like an I2C STOP sent (if in slave mode). This bit is self-clearing. |
| 3 | SI | RO | 0 | I2C Interrupt flag: 0b: no flag 1b: flag |
| 2 | AA | RW | 0 | Assert Acknowledge flag: 0b: NACK will be returned during the acknowledge clock pulse on the SCL line when data byte has been received 1b: ACK will be returned during the acknowledge clock pulse on the SCL line when: The address in the Slave Address Register has been received The General Call address has been received while the GC (general call) bit in the ADR register is set A data byte has been received while I2C is in master or slave mode |
| 1 | XADRF | RO | 0 | I2C Extended Slave Address Flag (10-bit addressing): 0b: No slave address match 1b: Slave address match with 10-bit address. This bit is cleared when new data is transmit/received |
| 0 | ADRF | RO | 0 | I2C Slave Address Flag: 0b: No slave address match 1b: Slave address match with 7-bit address. This bit is cleared when new data is transmit/received. |

28.6.2 I2CCONCLR

Register 28-2 I2CCONCLR (I2C Control Clear, 4001 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7 | I2CIEC | RW | 0 | I2C Interrupt Disable: 0b: No effect 1b: Clears I2CCONSET.I2CIE. This bit is self-clearing. |
| 6 | I2CENC | RW | 0 | I2C Interface Disable: 0b: No effect 1b: Clears I2CCONSET.I2CEN. This bit is self-clearing. |
| 5 | STAC | RW | 0 | START Flag Clear: 0b: No effect 1b: Clears I2CCONSET.STA. This bit is self-clearing. |
| 4 | Reserved | RO | 0 | Reserved |
| 3 | SIC | RW | 0 | I2C Interrupt Clear: 0b: No effect 1b: Clears I2CCONSET.SI. This bit is self-clearing. Clearing the I2CCONSET.SI bit will reset the I2CSTATUS register to 0xF8, so be sure not to writes this bit until ready for the I2C module to proceed. |
| 2 | AAC | RW | 0 | Assert Acknowledge Clear: 0b: No effect 1b: Clears I2CCONSET.AA. This bit is self-clearing. |
| 1:0 | Reserved | RO | 0 | Reserved |

28.6.3 I2CSTAT

Register 28-3 I2CSTAT (I2C Status, 4001 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|---------|-----------------------------|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:3 | Status | RO | 1 1111b | Status code for I2C engine. |
| 2:0 | Reserved | RO | 0 | Reserved |

28.6.4 I2CDAT

Register 28-4 I2CDAT (I2C Data, 4001 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | Data | RW | 0 | Data values received or to be transmit |

28.6.5 I2CCLK

Register 28-5 I2CCLK (I2C Clock, 4001 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|---|
| 31:7 | Reserved | RO | 0 | Reserved |
| 6:4 | M | RW | 0 | $F_{SAMP} = PCLK / 2^M$ |
| 3:0 | N | RW | 0 | $F_{SCL} = PCLK / (2^M \times (N+1) \times 10)$ |

28.6.6 I2CADR0

Register 28-6 I2CADR0 (I2C Slave Address 0, 4001 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Address | RW | 0 | The I2C device address for slave mode. |
| 0 | GC | RW | 0 | General Call Enable: 0b: General call disabled 1b: General call enabled |

28.6.7 I2CADRM0

Register 28-7 I2CADRM0 (I2C Slave Address Mask 0, 4001 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|----------|--------|-----------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Mask | RW | 111 1111b | Slave address mask bits. Bits that are set to 0 are don't care. Bits that are set to 1 should exactly match the address register. |

| | | | | |
|---|-----------------|----|---|--|
| | | | | The mask register has on effect on comparison to the GC address. |
| 0 | Reserved | RO | 0 | Reserved |

28.6.8 I2CXADR0

Register 28-8 I2CXADR0 (I2C Extended Slave Address 0, 4001 001Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31:11 | Reserved | RO | 0 | Reserved |
| 10:1 | Address | RW | 0 | The I2C device address for slave mode used for extended addressing. |
| 0 | Reserved | RO | 0 | Reserved |

28.6.9 I2CXADRM0

Register 28-9 I2CXADRM0 (I2C Extended Slave Address Mask 0, 4001 0020h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|---------------|--|
| 31:11 | Reserved | RO | 0 | Reserved |
| 10:1 | Mask | RW | 11 1111 1111b | Slave address mask bits. Bits that are set to 0 are don't care. Bits that are set to 1 should exactly match the address register. The mask register has no effect on comparison to the GC address. |
| 0 | Reserved | RO | 0 | Reserved |

28.6.10 I2CRST

Register 28-10 I2CRST (I2C Software Reset, 4001 0024h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:0 | RST | RW | 0 | I2C Software Reset. To reset, write this field to 0x07. |

28.6.11 I2CADR1

Register 28-11 I2CADR1 (I2C Slave Address 1, 4001 0028h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Address | RW | 0 | The I2C device address for slave mode. |
| 0 | GC | RW | 0 | General Call Enable: 0b: General call disabled 1b: General call enabled |

28.6.12 I2CADRM1

Register 28-12 I2CADRM1 (I2C Slave Address Mask 1, 4001 002Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-----------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Mask | RW | 111 1111b | Slave address mask bits. Bits that are set to 0 are don't care. Bits that are set to 1 should exactly match the address register. The mask register has on effect on comparison to the GC address. |
| 0 | Reserved | RO | 0 | Reserved |

28.6.13 I2CADR2

Register 28-13 I2CADR2 (I2C Slave Address 2, 4001 0030h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Address | RW | 0 | The I2C device address for slave mode. |
| 0 | GC | RW | 0 | General Call Enable: 0b: General call disabled 1b: General call enabled |

28.6.14 I2CADRM2

Register 28-14 I2CADRM2 (I2C Slave Address Mask 2, 4001 0034h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-----------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Mask | RW | 111 1111b | Slave address mask bits. Bits that are set to 0 are don't care. Bits that are set to 1 should exactly match the address register. The mask register has an effect on comparison to the GC address. |
| 0 | Reserved | RO | 0 | Reserved |

28.6.15 I2CADR3

Register 28-15 I2CADR3 (I2C Slave Address 3, 4001 0038h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-------|---|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Address | RW | 0 | The I2C device address for slave mode. |
| 0 | GC | RW | 0 | General Call Enable: 0b: General call disabled 1b: General call enabled |

28.6.16 I2CADRM3

Register 28-16 I2CADRM3 (I2C Slave Address Mask 3, 4001 003Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-----------------|--------|-----------|--|
| 31:8 | Reserved | RO | 0 | Reserved |
| 7:1 | Mask | RW | 111 1111b | Slave address mask bits. Bits that are set to 0 are don't care. Bits that are set to 1 should exactly match the address register. The mask register has an effect on comparison to the GC address. |
| 0 | Reserved | RO | 0 | Reserved |

29 CAN

The PAC55XX contains a Controller Area Network (CAN) peripheral that can be used as a serial communications interface.

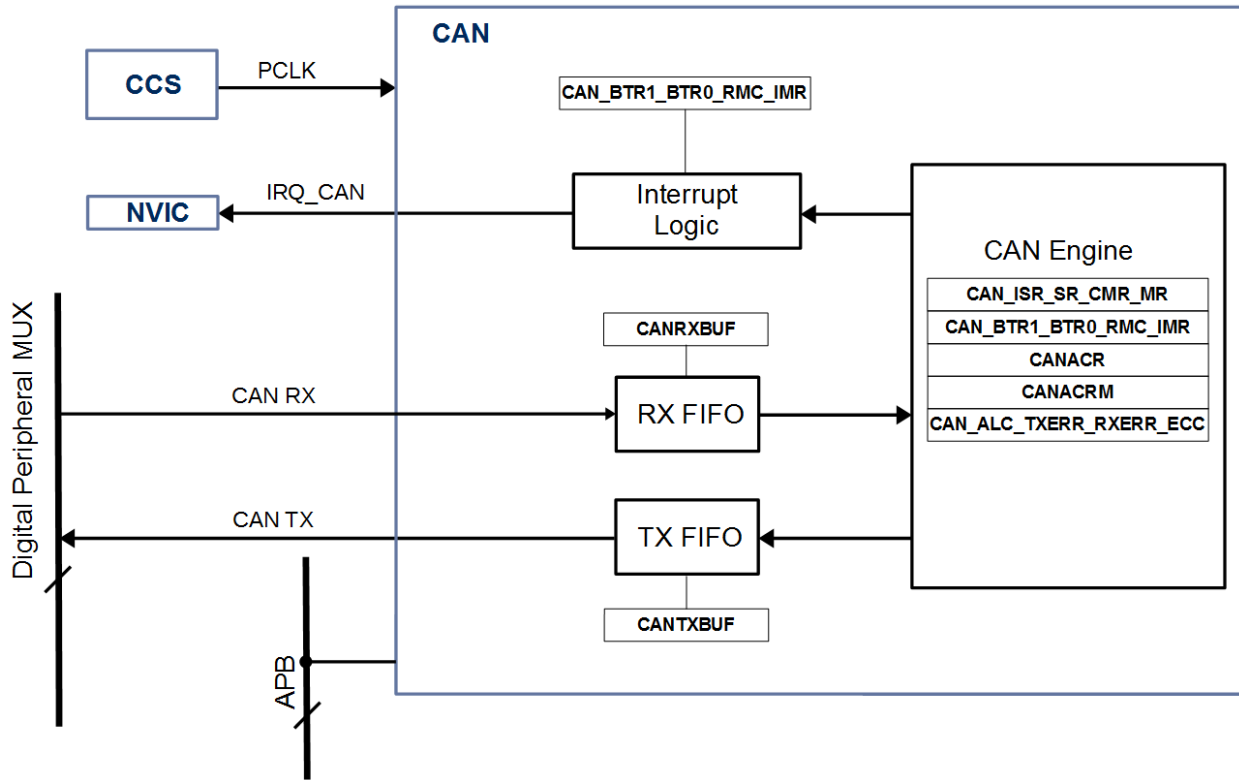
The CAN peripheral is an APB bus client and is clocked by the PCLK peripheral clock.

29.1 Features

- Conforms to Bosch CAN 2.0B Specification
 - 11 and 29-bit wide message identifiers
- Data rate up to 1Mbps
- Hardware Message Filtering (dual/single filters)
- 64-Byte Receive FIFO
- 16-Byte Transmit FIFO
- Overload Frame generated on FIFO overflow
- Normal and Listen Only mode support
- Single Shot transmission
- Ability to abort transmission
- Readable error counters
- Last error code

29.2 System Block Diagram

Figure 29-1 CAN System Block Diagram



29.3 Functional Description

TBD

29.4 Register Summary

Table 29-1 CAN Register Summary

| REGISTER | ADDRESS | DESCRIPTION | ACCESS | RESET |
|--------------------------------|------------|---|--------|-----------|
| CAN_ISR_SR_CMR_MR | 400A 0000h | CAN Mode/Command/Status/ISR Register | RW | 00000004h |
| CAN_BTR1_BTR0_RMC_IMR | 400A 0004h | CAN Bus Timing/RX Message Counter/ Interrupt Mask Register | RW | 00000000h |
| CANTXBUF | 400A 0008h | CAN Transmit Buffer Register | RW | 00000000h |
| CANRXBUF | 400A 000Ch | CAN Receive Buffer Register | RO | 00000000h |
| CANACR | 400A 0010h | CAN Acceptance Code Register | RW | 00000000h |
| CANAMR | 400A 0014h | CAN Acceptance Mask Register | RW | 00000000h |
| CAN_ALC_TXERR_RXERR_ECC | 400A 0018h | CAN ALC/TXERR/RXERR/ECC Register | RO | 00000000h |

29.5 Register Detail

29.5.1 CAN_ISR_SR_CMR_MR

Register 29-1 CAN_ISR_SR_CMR_MR (CAN Mode/Command/Status/ISR Register, 400A 0000h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-----|-----------------|--------|-------|--|
| 31 | Reserved | RO | 0 | Reserved |
| 30 | ALI | W1C | 0 | Arbitration Lost Interrupt: 0b: no interrupt 1b: interrupt |
| 29 | EWI | W1C | 0 | Error Warning Interrupt: 0b: no interrupt 1b: interrupt |
| 28 | EPI | W1C | 0 | Error Passive Interrupt: 0b: no interrupt 1b: interrupt |
| 27 | RI | W1C | 0 | Receive Interrupt: 0b: no interrupt 1b: interrupt |
| 26 | TI | W1C | 0 | Transmit Interrupt: 0b: no interrupt 1b: interrupt |
| 25 | BEI | W1C | 0 | Bus Error Interrupt: 0b: no interrupt 1b: interrupt |
| 24 | DOI | W1C | 0 | Data Overflow Interrupt: 0b: no interrupt 1b: interrupt |
| 23 | RBS | RO | 0 | 1b: Receive Buffer Status |
| 22 | DSO | RO | 0 | 1b: Data Overrun Status |
| 21 | TBS | RO | 0 | 1b: Transmit Buffer Status |
| 20 | Reserved | RO | 0 | Reserved |
| 19 | RS | RO | 0 | Receive Status |
| 18 | TS | RO | 0 | Transmit Status |
| 17 | ES | RO | 0 | Error Status |
| 16 | BS | RO | 0 | Bus Off Status |

| | | | | |
|-------|-----------------|----|---|--|
| 15:11 | Reserved | RO | 0 | Reserved |
| 10 | TR | RW | 0 | Transmit request: 0b: inactive 1b: active |
| 9 | AT | RW | 0 | Abort transmission: 0b: inactive 1b: active |
| 8:3 | Reserved | RO | 0 | Reserved |
| 2 | RM | RW | 1 | Reset mode: 0b: inactive 1b: active |
| 1 | LOM | RW | 0 | Listen only mode: 0b: inactive 1b: active |
| 0 | AFM | RW | 0 | Hardware acceptance filter scheme: 0b: Single filter 1b: Dual filter |

29.5.2 CAN_BTR1_BTR0_RMC_IMR

Register 29-2 CAN_BTR1_BTR0_RMC_IMR (CAN Bus Timing/RX Message Counter/Interrupt Mask Register, 400A 0004h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|-----------------|--------|-------|---|
| 31 | SAM | RW | 0 | Number of bus level samples: 0b: Bus is sampled once 1b: Bus is sampled three times |
| 30:28 | TSEG2 | RW | 0 | Number of clock cycles per Time Segment 2. |
| 27:24 | TSEG1 | RW | 0 | Number of clock cycles per Time Segment 1. |
| 23:22 | SJW | RW | 0 | Synchronization jump width. |
| 21:16 | BRP | RW | 0 | Baud Rate Pre-scaler. |
| 15:13 | Reserved | RO | 0 | Reserved |
| 12:8 | RMC | RO | 0 | Number of frames stored in the receive FIFO. Incremented on each successful frame reception and decremented by clearing RI interrupt. |
| 7 | Reserved | RO | 0 | Reserved |
| 6 | ALIM | RW | 0 | ALI Interrupt Mask: 0b: masked 1b: not masked |
| 5 | EWIM | RW | 0 | EWI Interrupt Mask: 0b: masked 1b: not masked |
| 4 | EPIM | RW | 0 | EPI Interrupt Mask: 0b: masked 1b: not masked |
| 3 | RIM | RW | 0 | RI Interrupt Mask: 0b: masked 1b: not masked |
| 2 | TIM | RW | 0 | TI Interrupt Mask: 0b: masked 1b: not masked |
| 1 | BEIM | RW | 0 | BEI Interrupt Mask: 0b: masked 1b: not masked |
| 0 | DOIM | RW | 0 | DOI Interrupt Mask: 0b: masked 1b: not masked |

29.5.3 CANTXBUF

Register 29-3 CANTXBUF (CAN Transmit Buffer Register, 400A 0008h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|--------------|--------|-------|----------------------|
| 31:0 | TXBUF | RW | 0 | Transmit Buffer Data |

29.5.4 CANRXBUF

Register 29-4 CANRXBUF (CAN Receive Buffer Register, 400A 000Ch)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|---------------|--------|-------|---------------------|
| 31:0 | RXBUF0 | RO | 0 | Receive Buffer Data |

29.5.5 CANACR

Register 29-5 CANACR (CAN Acceptance Code Register, 400A 0010h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|------------|--------|-------|-----------------|
| 31:0 | ACR | RW | 0 | Acceptance Code |

29.5.6 CANACRM

Register 29-6 CANACRM (CAN Acceptance Code Mask Register, 400A 0014h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|------|-------------|--------|-------|----------------------|
| 31:0 | ACRM | RW | 0 | Acceptance Code Mask |

29.5.7 CAN_ALC_TXERR_RXERR_ECC

Register 29-7 CAN_ALC_TXERR_RXERR_ECC (CAN Error Code Capture Register, 400A 0018h)

| BIT | NAME | ACCESS | RESET | DESCRIPTION |
|-------|---------------|--------|-------|---|
| 31:24 | CANALC | RO | 0 | Arbitration Lost Bit Number: 00: Arbitration lost in ID28 / 10 01: Arbitration lost in ID28 / 9 02: Arbitration lost in ID28 / 8 03: Arbitration lost in ID28 / 7 04: Arbitration lost in ID28 / 6 05: Arbitration lost in ID28 / 5 06: Arbitration lost in ID28 / 4 07: Arbitration lost in ID28 / 3 08: Arbitration lost in ID28 / 2 09: Arbitration lost in ID28 / 1 10: Arbitration lost in ID28 / 0 11: Arbitration lost in SRTR / RTR 12: Arbitration lost in IDE bit 13: Arbitration lost in ID17 ¹⁵ 14: Arbitration lost in ID16 ¹⁵ 15: Arbitration lost in ID15 ¹⁵ 16: Arbitration lost in ID14 ¹⁵ 17: Arbitration lost in ID13 ¹⁵ 18: Arbitration lost in ID12 ¹⁵ 19: Arbitration lost in ID11 ¹⁵ 20: Arbitration lost in ID10 ¹⁵ 21: Arbitration lost in ID9 ¹⁵ 22: Arbitration lost in ID8 ¹⁵ 23: Arbitration lost in ID7 ¹⁵ 24: Arbitration lost in ID6 ¹⁵ 25: Arbitration lost in ID5 ¹⁵ 26: Arbitration lost in ID4 ¹⁵ 27: Arbitration lost in ID3 ¹⁵ 28: Arbitration lost in ID2 ¹⁵ 29: Arbitration lost in ID1 ¹⁵ 30: Arbitration lost in ID0 ¹⁵ 31: Arbitration lost in RTR |
| 23:16 | TXERR | RO | 0 | Transmit error counter |
| 15:8 | RXERR | RO | 0 | Receive error counter |
| 7 | RXWRN | RO | 0 | Set when CANRXERR ≥ 96 |
| 6 | TXWRN | RO | 0 | Set when CANTXERR ≥ 96 |
| 5 | EDIR | RO | 0 | Direction of transfer while error occurred: 0b: transmission 1b: reception |
| 4 | ACKER | RO | 0 | ACK error occurred |
| 3 | FRMER | RO | 0 | Form error occurred |

¹⁵ Extended frame messages only

| | | | | |
|---|--------------|----|---|----------------------|
| 2 | CRCER | RO | 0 | CRC error occurred |
| 1 | STFER | RO | 0 | Stuff error occurred |
| 0 | BER | RO | 0 | Bit error occurred |

30 ARM CORTEX-M4F REFERENCE

The PAC55XX controller contains an ARM Cortex-M4F MCU. The ARM Cortex-M4F has several configurable options. The options listed below are present on the PAC55XX family of controllers:

- IEEE754 single-precision Floating Point Unit
- Memory Protection Unit (MPU) included
- Number of interrupt priorities: 8 (3-bit priority)
- Wakeup Interrupt Controller (WIC) included
- Sleep Mode power-saving included
- Little Endian configuration
- 24-bit SysTick timer included
- Embedded Trace Module (ETM) included
 - Instruction trace only

ARM provides a full set of documentation for the ARM Cortex-M4F MCU.

You can retrieve the full set of documentation on the ARM Cortex-M4F from here:

<http://infocenter.arm.com/help/index.jsp>

The documents that are the most important are:

- [ARM Cortex-M4 Processor Technical Reference Manual](#)
- [Coresight ETM-M4 Technical Reference Manual](#)

The [ARM Cortex-M4 Processor Technical Reference Manual](#) contains documentation for the Cortex-M4 processor, the programmer's model, instruction set, registers, memory map, floating point multimedia, trace and debug support.

The [Cortex-M4 Devices Generic User Guide](#) contains documentation for the Cortex-M4F's *Embedded Trace Macrocell*[™]

31 LEGAL INFORMATION

Copyright © 2017 Active-Semi, Inc. All rights reserved.

All information provided in this document is subject to legal disclaimers.

Active-Semi reserves the right to modify its products, circuitry or product specifications without notice. Active-Semi products are not intended, designed, warranted or authorized for use as critical components in life-support, life-critical or safety-critical devices, systems, or equipment, nor in applications where failure or malfunction of any Active-Semi product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Active-Semi accepts no liability for inclusion and/or use of its products in such equipment or applications. Active-Semi does not assume any liability arising out of the use of any product, circuit, or any information described in this document. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of Active-Semi or others. Active-Semi assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein. Customers should evaluate each product to make sure that it is suitable for their applications. Customers are responsible for the design, testing, and operation of their applications and products using Active-Semi products. Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. All products are sold subject to Active-Semi's terms and conditions of sale supplied at the time of order acknowledgment. Exportation of any Active-Semi product may be subject to export control laws.

Active-Semi[™], Active-Semi logo, Solutions for Sustainability[™], Power Application Controller[™], Micro Application Controller[™], Multi-Mode Power Manager[™], Configurable Analog Front End[™], and Application Specific Power Drivers[™] are trademarks of Active-Semi, Inc.

ARM[®] is a registered trademark and Cortex[™] is a trademark of ARM Limited. All referenced brands and trademarks are the property of their respective owners.

For more information on this and other products, contact sales@active-semi.com or visit www.active-semi.com.