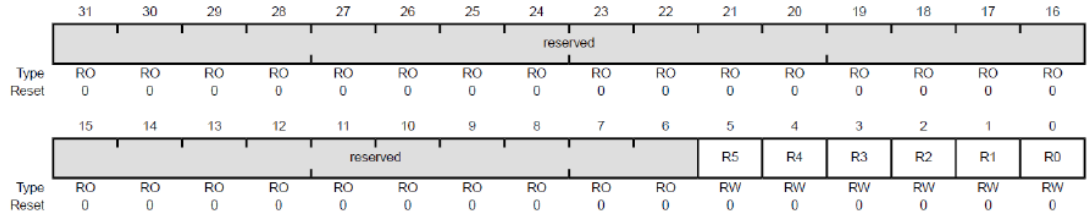
# GPIO

## GPIO pins and alternate functions

| IO | Pin | Analog Function | Digital Function (GPIOPCTL PMCx Bit Field Encoding)[a] | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 14 | 15 |
| PA0 | 17 | - | U0Rx | - | - | - | - | - | - | CAN1Rx | - | - | - |
| PA1 | 18 | - | U0Tx | - | - | - | - | - | - | CAN1Tx | - | - | - |
| PA2 | 19 | - | - | SSI0Clk | - | - | - | - | - | - | - | - | - |
| PA3 | 20 | - | - | SSI0Fss | - | - | - | - | - | - | - | - | - |
| PA4 | 21 | - | - | SSI0Rx | - | - | - | - | - | - | - | - | - |
| PA5 | 22 | - | - | SSI0Tx | - | - | - | - | - | - | - | - | - |
| PA6 | 23 | - | - | - | I2C1SCL | - | M1PWM2 | - | - | - | - | - | - |
| PA7 | 24 | - | - | - | I2C1SDA | - | M1PWM3 | - | - | - | - | - | - |
| PB0 | 45 | USB0ID | U1Rx | - | - | - | - | - | T2CCP0 | - | - | - | - |
| PB1 | 46 | USB0VBUS | U1Tx | - | - | - | - | - | T2CCP1 | - | - | - | - |
| PB2 | 47 | - | - | - | I2C0SCL | - | - | - | T3CCP0 | - | - | - | - |
| PB3 | 48 | - | - | - | I2C0SDA | - | - | - | T3CCP1 | - | - | - | - |
| PB4 | 58 | AIN10 | - | SSI2Clk | - | M0PWM2 | - | - | T1CCP0 | CAN0Rx | - | - | - |
| PB5 | 57 | AIN11 | - | SSI2Fss | - | M0PWM3 | - | - | T1CCP1 | CAN0Tx | - | - | - |
| PB6 | 1 | - | - | SSI2Rx | - | M0PWM0 | - | - | T0CCP0 | - | - | - | - |
| PB7 | 4 | - | - | SSI2Tx | - | M0PWM1 | - | - | T0CCP1 | - | - | - | - |
| PC0 | 52 | - | TCK SWCLK | - | - | - | - | - | T4CCP0 | - | - | - | - |
| PC1 | 51 | - | TMS SWDIO | - | - | - | - | - | T4CCP1 | - | - | - | - |
| PC2 | 50 | - | TDI | - | - | - | - | - | T5CCP0 | - | - | - | - |
| PC3 | 49 | - | TDO SWO | - | - | - | - | - | T5CCP1 | - | - | - | - |
| PC4 | 16 | C1- | U4Rx | U1Rx | - | M0PWM6 | - | IDX1 | WT0CCP0 | U1RTS | - | - | - |
| PC5 | 15 | C1+ | U4Tx | U1Tx | - | M0PWM7 | - | PhA1 | WT0CCP1 | U1CTS | - | - | - |
| PC6 | 14 | C0+ | U3Rx | - | - | - | - | PhB1 | WT1CCP0 | USB0EPEN | - | - | - |
| PC7 | 13 | C0- | U3Tx | - | - | - | - | - | WT1CCP1 | USB0PFLT | - | - | - |
| PD0 | 61 | AIN7 | SSI3Clk | SSI1Clk | I2C3SCL | M0PWM6 | M1PWM0 | - | WT2CCP0 | - | - | - | - |
| PD1 | 62 | AIN6 | SSI3Fss | SSI1Fss | I2C3SDA | M0PWM7 | M1PWM1 | - | WT2CCP1 | - | - | - | - |
| PD2 | 63 | AIN5 | SSI3Rx | SSI1Rx | - | M0FAULT0 | - | - | WT3CCP0 | USB0EPEN | - | - | - |
| PD3 | 64 | AIN4 | SSI3Tx | SSI1Tx | - | - | - | IDX0 | WT3CCP1 | USB0PFLT | - | - | - |
| PD4 | 43 | USB0DM | U6Rx | - | - | - | - | - | WT4CCP0 | - | - | - | - |
| PD5 | 44 | USB0DP | U6Tx | - | - | - | - | - | WT4CCP1 | - | - | - | - |
| PD6 | 53 | - | U2Rx | - | - | M0FAULT0 | - | PhA0 | WT5CCP0 | - | - | - | - |
| PD7 | 10 | - | U2Tx | - | - | - | - | PhB0 | WT5CCP1 | NMI | - | - | - |
| PE0 | 9 | AIN3 | U7Rx | - | - | - | - | - | - | - | - | - | - |
| PE1 | 8 | AIN2 | U7Tx | - | - | - | - | - | - | - | - | - | - |
| PE2 | 7 | AIN1 | - | - | - | - | - | - | - | - | - | - | - |
| PE3 | 6 | AIN0 | - | - | - | - | - | - | - | - | - | - | - |
| PE4 | 59 | AIN9 | U5Rx | - | I2C2SCL | M0PWM4 | M1PWM2 | - | - | CAN0Rx | - | - | - |
| PE5 | 60 | AIN8 | U5Tx | - | I2C2SDA | M0PWM5 | M1PWM3 | - | - | CAN0Tx | - | - | - |
| PF0 | 28 | - | U1RTS | SSI1Rx | CAN0Rx | - | M1PWM4 | PhA0 | T0CCP0 | NMI | C0o | - | - |
| PF1 | 29 | - | U1CTS | SSI1Tx | - | - | M1PWM5 | PhB0 | T0CCP1 | - | C1o | TRD1 | - |
| PF2 | 30 | - | - | SSI1Clk | - | M0FAULT0 | M1PWM6 | - | T1CCP0 | - | - | TRD0 | - |
| PF3 | 31 | - | - | SSI1Fss | CAN0Tx | - | M1PWM7 | - | T1CCP1 | - | - | TRCLK | - |
| PF4 | 5 | - | - | - | - | - | M1FAULT0 | IDX0 | T2CCP0 | USB0EPEN | - | - | - |

## Run Clock Gate Control Register for GPIO (RCGCGPIO) (address is 0x400FE608)




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 5 | R5 | RW | 0 | GPIO Port F Run Mode Clock Gating Control<br>Value Description<br>0 GPIO Port F is disabled.<br>1 Enable and provide a clock to GPIO Port F in Run mode. |
| . . . | | | | |
| 0 | R0 | RW | 0 | GPIO Port A Run Mode Clock Gating Control<br>Value Description<br>0 GPIO Port A is disabled.<br>1 Enable and provide a clock to GPIO Port A in Run mode. |

## Base addresses for GPIO ports

Port A: 0x40004000

Port B: 0x40005000

Port C: 0x40006000

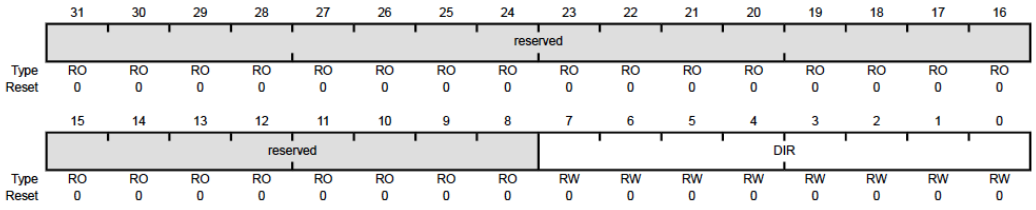
Port D: 0x40007000

Port E: 0x40024000

Port F: 0x40025000

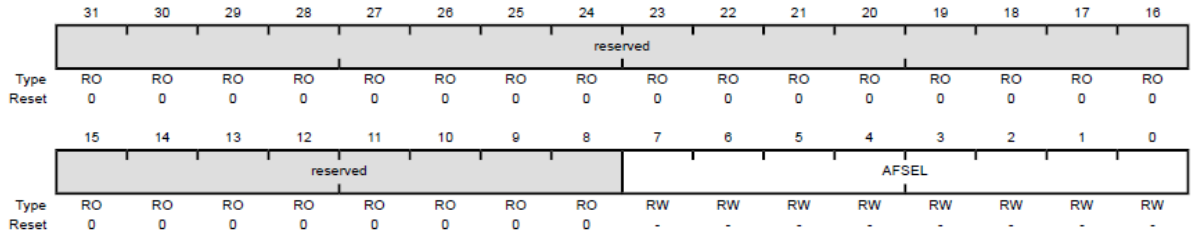
## Data Register: GPIODATA, offset 0x000

## Data direction register: GPIODIR, offset 0x400




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | DIR | RW | 0x00 | GPIO Data Direction<br>Value Description<br>0 Corresponding pin is an input.<br>1 Corresponding pins is an output. |

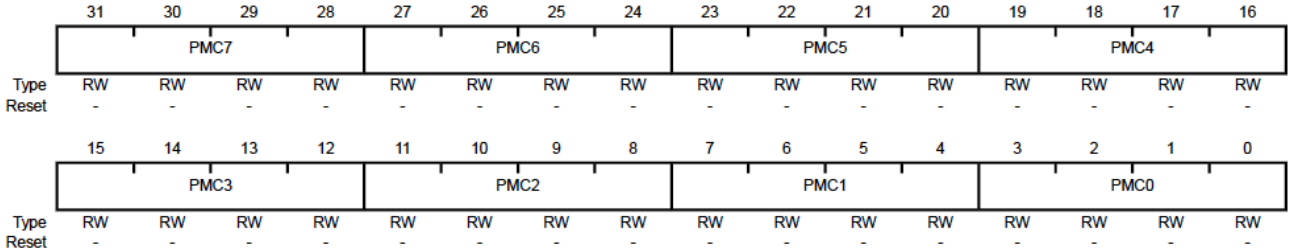
## GPIO Alternate Function Select: GPIOAFSEL, offset 0x420




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | AFSEL | RW | - | GPIO Alternate Function Select |

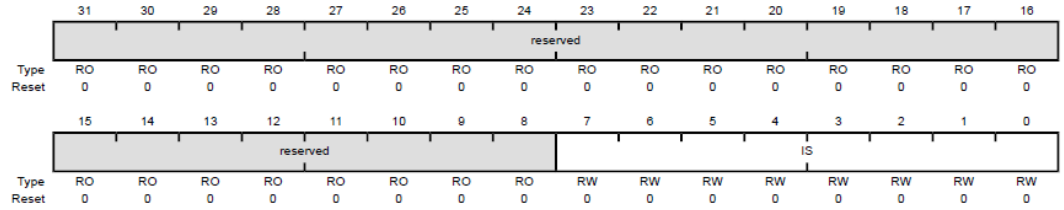
| Value | Description |
|---|---|
| 0 | The associated pin functions as a GPIO and is controlled by the GPIO registers. |
| 1 | The associated pin functions as a peripheral signal and is controlled by the alternate hardware function. |

## GPIO Port Control: GPIOPCTL, offset 0x52C




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:28 | PMC7 | RW | - | Port Mux Control 7<br>This field controls the configuration for GPIO pin 7. |
| 27:24 | PMC6 | RW | - | Port Mux Control 6<br>This field controls the configuration for GPIO pin 6. |
| 23:20 | PMC5 | RW | - | Port Mux Control 5<br>This field controls the configuration for GPIO pin 5. |
| 19:16 | PMC4 | RW | - | Port Mux Control 4<br>This field controls the configuration for GPIO pin 4. |
| 15:12 | PMC3 | RW | - | Port Mux Control 3<br>This field controls the configuration for GPIO pin 3. |
| 11:8 | PMC2 | RW | - | Port Mux Control 2<br>This field controls the configuration for GPIO pin 2. |
| 7:4 | PMC1 | RW | - | Port Mux Control 1<br>This field controls the configuration for GPIO pin 1. |
| 3:0 | PMC0 | RW | - | Port Mux Control 0<br>This field controls the configuration for GPIO pin 0. |

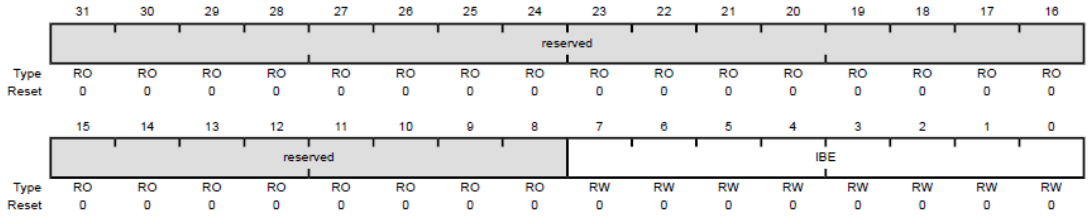
## GPIO Interrupt Sense (GPIOIS), offset 0x404




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | IS | RW | 0x00 | GPIO Interrupt Sense |

| Value | Description |
|---|---|
| 0 | The edge on the corresponding pin is detected (edge-sensitive). |
| 1 | The level on the corresponding pin is detected (level-sensitive). |

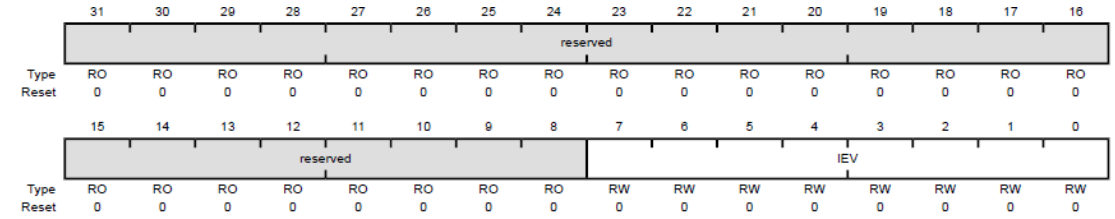
## GPIO Interrupt Both Edges:GPIOIBE, offset 0x408




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | IBE | RW | 0x00 | GPIO Interrupt Both Edges |

| Value | Description |
|---|---|
| 0 | Interrupt generation is controlled by the **GPIO Interrupt Event (GPIOIEV)** register (see page 666). |
| 1 | Both edges on the corresponding pin trigger an interrupt. |

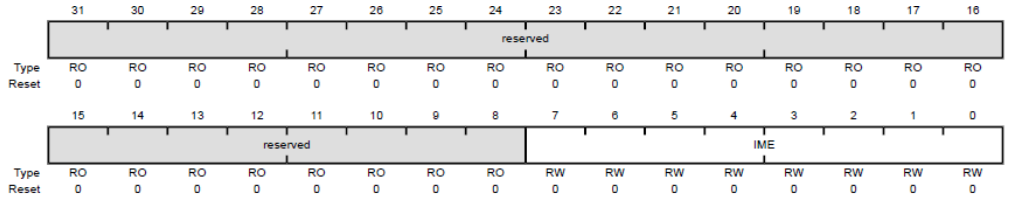
## GPIO Interrupt Event : GPIOIEV, offset 0x40C




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | IEV | RW | 0x00 | GPIO Interrupt Event |

| Value | Description |
|---|---|
| 0 | A falling edge or a Low level on the corresponding pin triggers an interrupt. |
| 1 | A rising edge or a High level on the corresponding pin triggers an interrupt. |

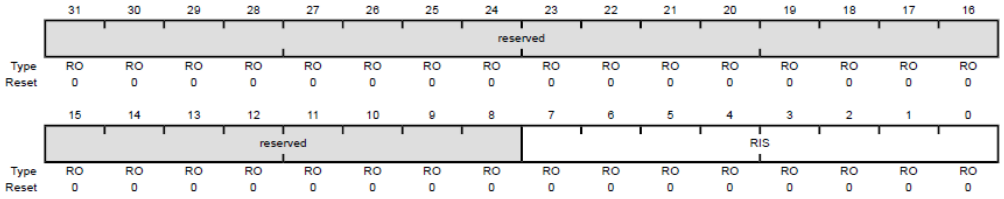
## GPIO Interrupt Mask: GPIOIM, offset 0x410




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | IME | RW | 0x00 | GPIO Interrupt Mask Enable |

| Value | Description |
|---|---|
| 0 | The interrupt from the corresponding pin is masked. |
| 1 | The interrupt from the corresponding pin is sent to the interrupt controller. |

## GPIO Raw Interrupt Status: GPIORIS, offset 0x414 (read only)



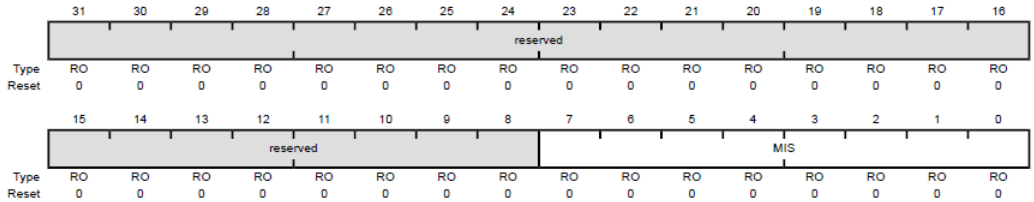

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | RIS | RO | 0x00 | GPIO Interrupt Raw Status |

| Value | Description |
|---|---|
| 0 | An interrupt condition has not occurred on the corresponding pin. |
| 1 | An interrupt condition has occurred on the corresponding pin. |

For edge-detect interrupts, this bit is cleared by writing a 1 to the corresponding bit in the **GPIOICR** register.

For a GPIO level-detect interrupt, the bit is cleared when the level is deasserted.

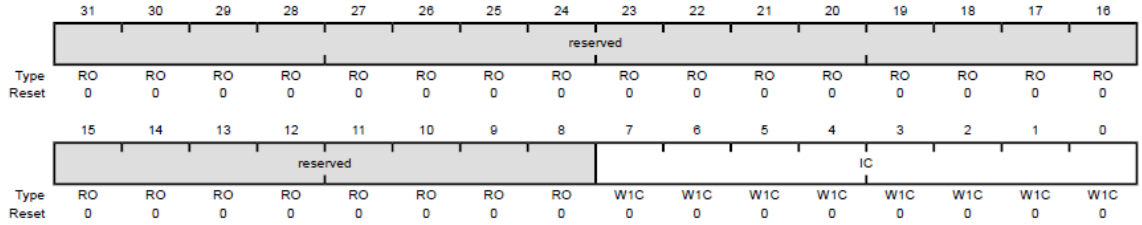
## GPIO Masked Interrupt Status: GPIOMIS, offset 0x418 (read only)




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | MIS | RO | 0x00 | GPIO Masked Interrupt Status |

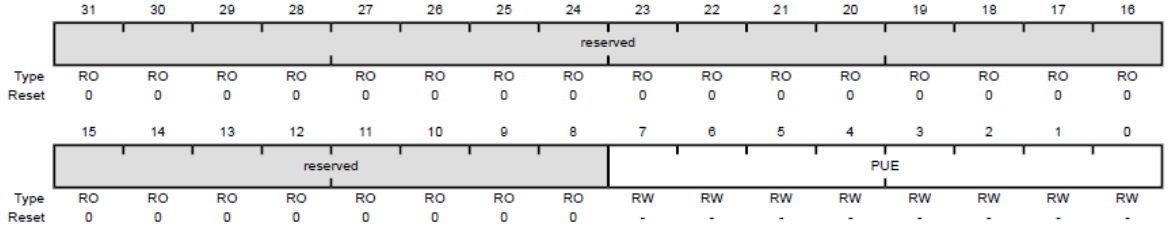
| Value | Description |
|---|---|
| 0 | An interrupt condition on the corresponding pin is masked or has not occurred. |
| 1 | An interrupt condition on the corresponding pin has triggered an interrupt to the interrupt controller. |

## GPIO Interrupt Clear: GPIOICR, offset 0x41C



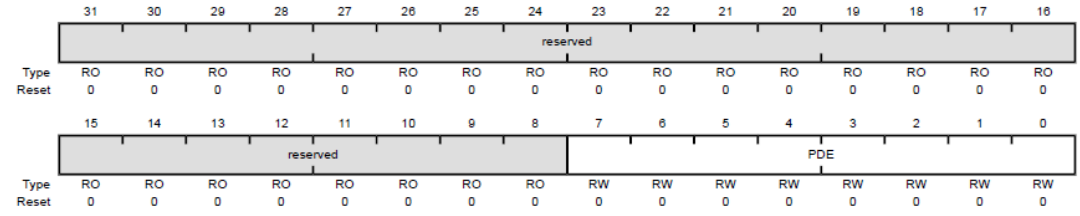

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | IC | W1C | 0x00 | GPIO Interrupt Clear<br>Value Description<br>0 The corresponding interrupt is unaffected.<br>1 The corresponding interrupt is cleared. |

## GPIO Pull-Up Select: GPIOPUR, offset 0x510



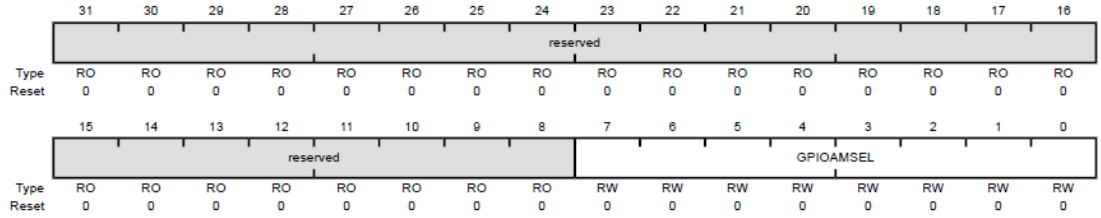

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | PUE | RW | - | Pad Weak Pull-Up Enable<br>Value Description<br>0 The corresponding pin's weak pull-up resistor is disabled.<br>1 The corresponding pin's weak pull-up resistor is enabled. |

## GPIO Pull-Down Select (GPIOPDR), offset 0x514




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | PDE | RW | 0x00 | Pad Weak Pull-Down Enable<br>Value Description<br>0 The corresponding pin's weak pull-down resistor is disabled.<br>1 The corresponding pin's weak pull-down resistor is enabled. |

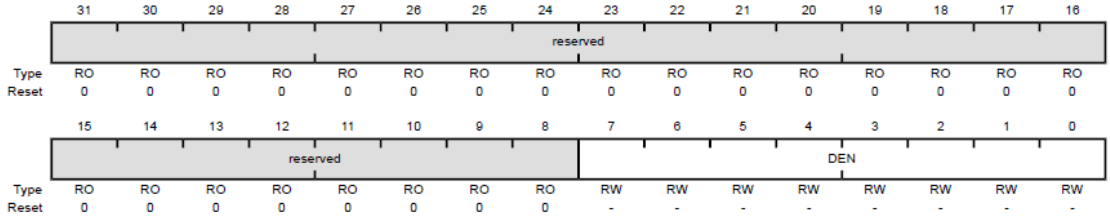
## GPIO Analog Mode Select: GPIOAMSEL, offset 0x528




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | GPIOAMSEL | RW | 0x00 | GPIO Analog Mode Select |

| Value | Description |
|---|---|
| 0 | The analog function of the pin is disabled, the isolation is enabled, and the pin is capable of digital functions as specified by the other GPIO configuration registers. |
| 1 | The analog function of the pin is enabled, the isolation is disabled, and the pin is capable of analog functions. |

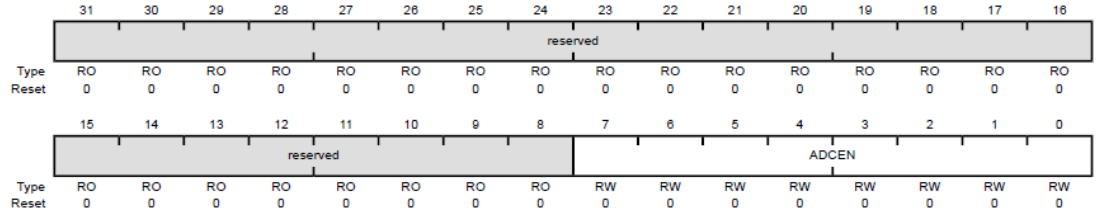
## GPIO Digital Enable: GPIODEN, offset 0x51C




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | DEN | RW | - | Digital Enable |

| Value | Description |
|---|---|
| 0 | The digital functions for the corresponding pin are disabled. |
| 1 | The digital functions for the corresponding pin are enabled. |

## GPIO ADC Control: GPIOADCCTL, offset 0x530




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:8 | reserved | RO | 0x0000.00 | Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation. |
| 7:0 | ADCEN | RW | 0x00 | ADC Trigger Enable |

| Value | Description |
|---|---|
| 0 | The corresponding pin is not used to trigger the ADC. |
| 1 | The corresponding pin is used to trigger the ADC. |

# TIMER

## SysTick

| Address | 31-24 | 23-17 | 16 | 15-3 | 2 | 1 | 0 | Name |
|---|---|---|---|---|---|---|---|---|
| $E000E010 | 0 | 0 | COUNT | 0 | CLK_SRC | INTEN | ENABLE | STCTRL |
| $E000E014 | 0 | 24-bit RELOAD value | | | | | | STRELOAD |
| $E000E018 | 0 | 24-bit CURRENT value of SysTick counter | | | | | | STCURRENT |

### SysTick Control and Status Register(STCTRL)

Enable: 0 timer disabled, 1 timer enabled

INTEN: 0 interrupt disabled, 1 interrupt is generated to the NVIC when SysTick counts to 0.

CLK_SRC: 0 (POSC) divided by 4, 1 system clock

COUNT: 0 SysTick timer has not counted to 0 yet, 1 SysTick timer has counted to 0. This bit is cleared by a read of the register or if the STCURRENT register is written with any value.

### SysTick Reload Value Register (STRELOAD)

This register specifies the start value to load into the SysTick Current Value (STCURRENT) register when the counter reaches 0.

Start value can be between 0x1 and 0x00FFFFFF (only 24 bit value an be written)

### SysTick Current Value Register (STCURRENT)

This register is write-clear. Writing to it with any value clears the register. Clearing this register also clears the COUNT bit of the STCTRL register.

## General Purpose Timers

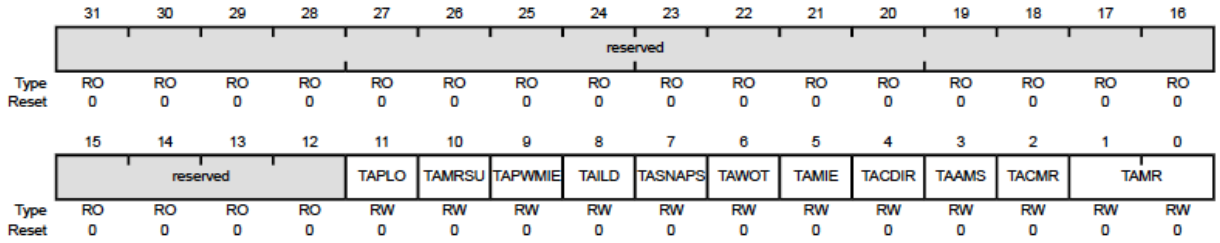
### Base addresses of six 16/32 bit Timers and six 32/64 bit Timers:

- 16/32-bit Timer 0: 0x4003.0000
- 16/32-bit Timer 1: 0x4003.1000
- 16/32-bit Timer 2: 0x4003.2000
- 16/32-bit Timer 3: 0x4003.3000
- 16/32-bit Timer 4: 0x4003.4000
- 16/32-bit Timer 5: 0x4003.5000
- 32/64-bit Wide Timer 0: 0x4003.6000
- 32/64-bit Wide Timer 1: 0x4003.7000
- 32/64-bit Wide Timer 2: 0x4004.C000
- 32/64-bit Wide Timer 3: 0x4004.D000
- 32/64-bit Wide Timer 4: 0x4004.E000
- 32/64-bit Wide Timer 5: 0x4004.F000

### Timer Configuration : CFG, offset 0x000

| Value of Bit field 2:0 | Description |
|---|---|
| 0x0 | For a 16/32-bit timer, this value selects the 32-bit timer configuration.<br>For a 32/64-bit wide timer, this value selects the 64-bit timer configuration. |
| 0x1 | For a 16/32-bit timer, this value selects the 32-bit real-time clock (RTC) counter configuration.<br>For a 32/64-bit wide timer, this value selects the 64-bit real-time clock (RTC) counter configuration. |
| 0x4 | For a 16/32-bit timer, this value selects the 16-bit timer configuration.<br>For a 32/64-bit wide timer, this value selects the 32-bit timer configuration. |

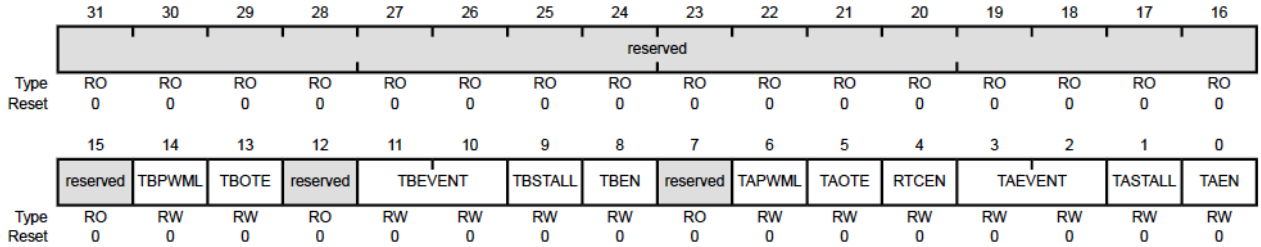
### Timer A/B Mode : TAMR/ TBMR, offset 0x004/0x008




| Bit field | Value | Description |
|---|---|---|
| 1:0 | 0x1 | One shot timer mode |
| | 0x2 | Periodic timer mode |
| | 0x3 | Capture mode |
| 2 | 0 | Edge count mode |
| | 1 | Edge time mode |
| 3 | 0 | Capture or compare mode |
| | 1 | PWM |
| 4 | 0 | Timer counts down |

| | | |
|---|---|---|
| | 1 | Timer counts up |
| 5 | 0 | The match interrupt is disabled for match events. |
| | 1 | An interrupt is generated when the match value in the TAMATCHR register is reached in the one-shot and periodic modes. |
| 6 | 0 | Timer A begins counting as soon as it is enabled |
| | 1 | If Timer A is enabled (TAEN is set in the CTL register), Timer A does not begin counting until it receives a trigger from the timer in the previous position in the daisy chain |
| 7 | 0 | Snap shot mode is disabled |
| | 1 | If Timer A is configured in the periodic mode, the actual free-running, capture or snapshot value of Timer A is loaded at the time-out event/capture or snapshot event into the Timer A ( TAR) register. If the timer prescaler is used, the prescaler snapshot is loaded into the Timer A ( TAPR). |
| 8 | 0 | Update the TAR and TAV registers with the value in the TAILR register on the next cycle. Also update the TAPS and TAPV registers with the value in the TAPR register on the next cycle. |
| | 1 | Update the TAR and TAV registers with the value in the TAILR register on the next timeout. Also update the TAPS and TAPV registers with the value in the TAPR register on the next timeout. |
| 9 | 0 | Capture event interrupt is disabled. |
| | 1 | Capture event interrupt is enabled. |
| 10 | 0 | Update the TAMATCHR register and the TAPR register, if used, on the next cycle. |
| | 1 | Update the TAMATCHR register and the TAPR register, if used, on the next time out. |
| 11 | 0 | Legacy operation with CCP pin driven Low when the TAILR is reloaded after the timer reaches 0. |
| | 1 | CCP is driven High when the TAILR is reloaded after the timer reaches 0. |

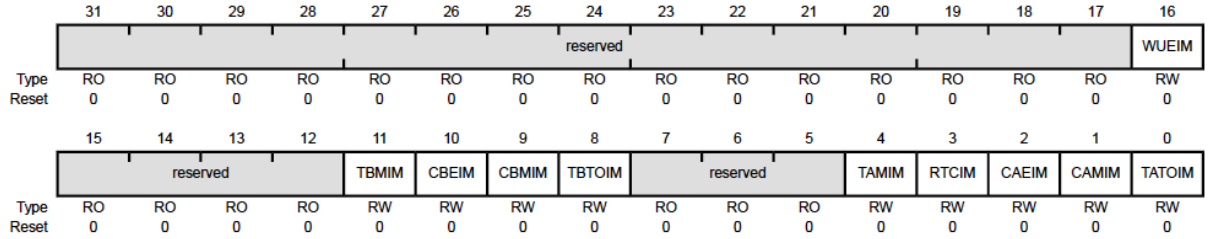
**Timer Control : CTL, offset 0x00C**




| Bit field | Value | Description |
|---|---|---|
| 14 | 0 | Output is unaffected. |
| | 1 | Output is inverted. |
| 13 | 0 | The output Timer B ADC trigger is disabled. |
| | 1 | The output Timer B ADC trigger is enabled. |
| 11:10 | 0x0 | Timer B Event Mode, Positive edge |
| | 0x1 | Timer B Event Mode, Negative edge |
| | 0x3 | Timer B Event Mode, both edges |

| | | |
|---|---|---|
| 9 | 0 | Timer B continues counting while the processor is halted by the debugger. |
| | 1 | Timer B freezes counting while the processor is halted by the debugger. |
| 8 | 0 | Timer B is disabled. |
| | 1 | Timer B is enabled and begins counting or the capture logic is enabled based on the CFG register. |
| 6 | 0 | Timer A PWM Output Level is unaffected. |
| | 1 | Timer A PWM Output Level is inverted. |
| 5 | 0 | The output Timer A ADC trigger is disabled. |
| | 1 | The output Timer A ADC trigger is enabled. |
| 4 | 0 | RTC counting freezes while the processor is halted by the debugger. |
| | 1 | RTC counting continues while the processor is halted by the debugger. |
| 3:2 | 0x0 | Timer A Event Mode, Positive edge |
| | 0x1 | Timer A Event Mode, Negative edge |
| | 0x3 | Timer A Event Mode, both edges |
| 1 | 0 | Timer A continues counting while the processor is halted by the debugger. |
| | 1 | Timer A freezes counting while the processor is halted by the debugger. |
| 0 | 0 | Timer A is disabled. |
| | 1 | Timer A is enabled and begins counting or the capture logic is enabled based on the CFG register. |

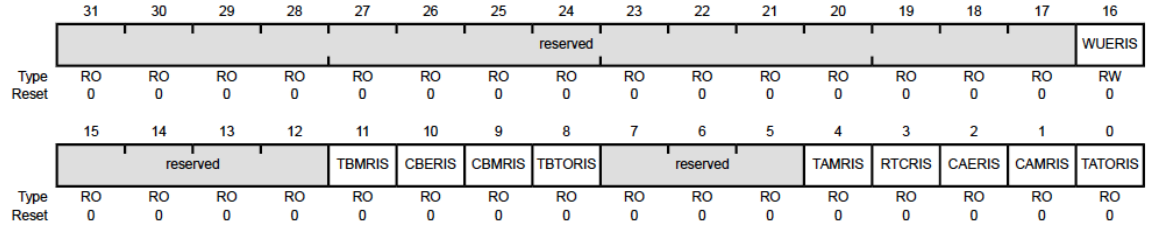
**Timer Interrupt Mask: IMR, offset 0x018**




| bit | function |
|---|---|
| 16 | 32/64-Bit Wide Write Update (WUE) Error Interrupt Mask |
| 11 | Timer B Match (TBM) Interrupt Mask |
| 10 | Timer B Capture Mode Event (CBE) Interrupt Mask |
| 9 | Timer B Capture Mode Match (CBM) Interrupt Mask |
| 8 | Timer B Time-Out (TBTO) Interrupt Mask |
| 4 | Timer A Match (TAM) Interrupt Mask |
| 3 | RTC Interrupt Mask |
| 2 | Timer A Capture Mode Event (CAE) Interrupt Mask |
| 1 | Timer A Capture Mode Match (CAM) Interrupt Mask |
| 0 | Timer A Time-Out (TATO) Interrupt Mask |

0: Interrupt is disabled, 1: Interrupt is enabled

**Timer Raw Interrupt Status : RIS, offset 0x01C**

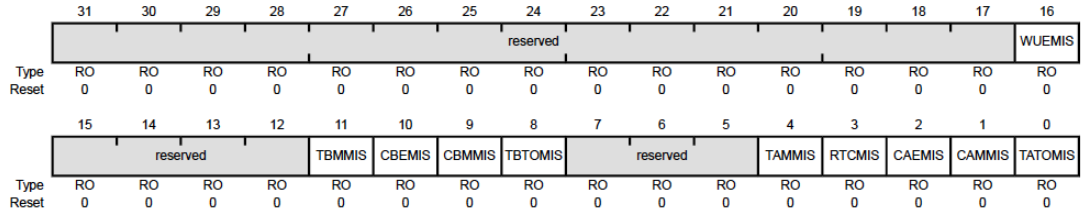



0: Match/capture/time-out etc. events not occurred.

1: Match/capture/time-out etc. events occurred.

(TBM, CBE, etc. corresponds to the same abbreviations as the above table)

**Timer Masked Interrupt Status : MIS, offset 0x020**

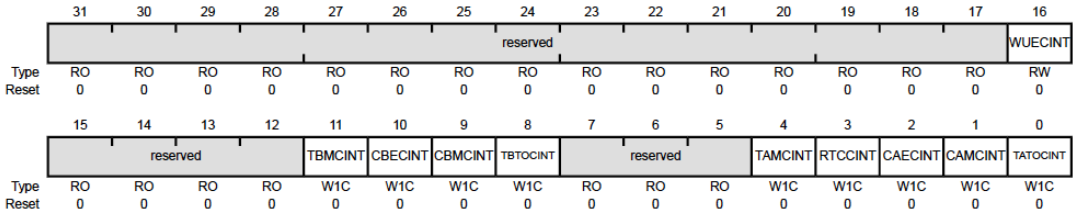



0: Match/capture/time-out etc. interrupts has not occurred or masked.

1: Unmasked Match/capture/time-out etc. interrupt has occurred.

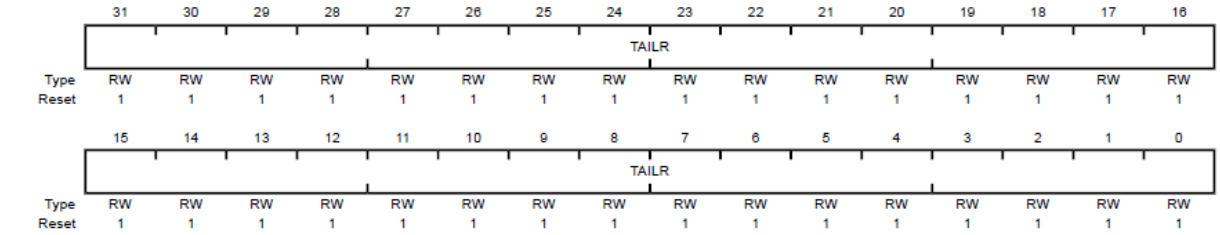
(TBM, CBE, etc. corresponds to the same abbreviations as the above table)

**Timer Interrupt Clear: ICR, offset 0x024**




Writing a 1 to this a bit clears the respective bit in RIS and MRIS registers.

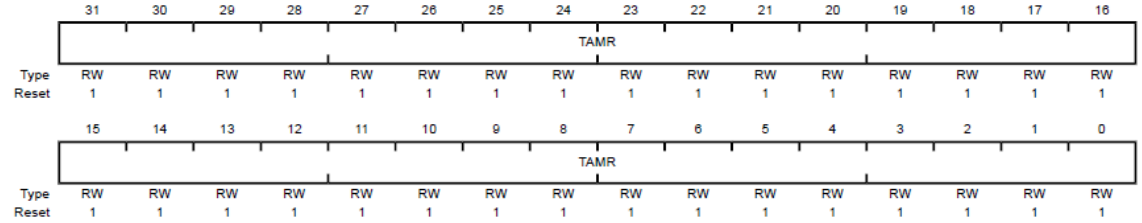
**Timer A/B Interval Load: TAILR/ TBILR, offset 0x028/0x02C**




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:0 | TAILR | RW | 0xFFFF.FFFF | GPTM Timer A Interval Load Register<br>Writing this field loads the counter for Timer A. . |

(Same for Timer B)

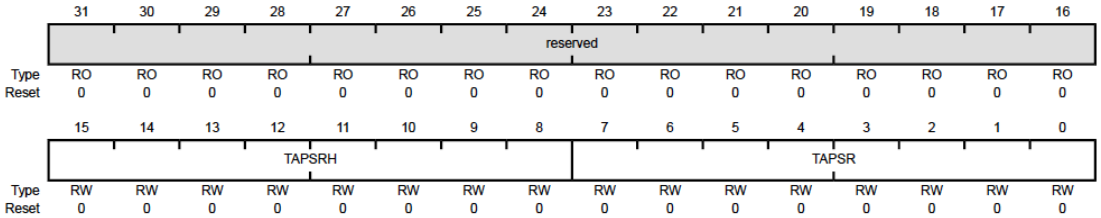
**Timer A/B Match: TAMATCHR/ TBMATCHR, offset 0x030/0x034**




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:0 | TAMR | RW | 0xFFFF.FFFF | GPTM Timer A Match Register<br>This value is compared to the **GPTMTAR** register to determine match events. |

(Same for Timer B)

**Timer A/B Prescale: TAPR/ TBPR, offset 0x038/0x03C**

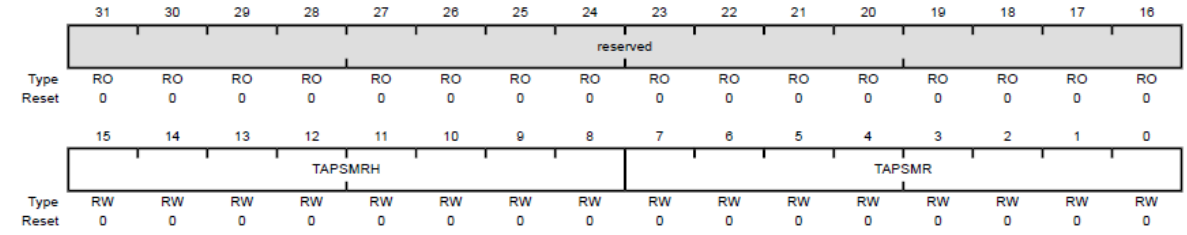



TAPSRH: GPTM Timer A Prescale High Byte. For the 16/32-bit GPTM, this field is reserved. For the 32/64-bit Wide GPTM, this field contains the upper 8-bits of the 16-bit prescaler.

TAPSRL: GPTM Timer A Prescale Low Byte. For the 16/32-bit GPTM, this field contains the entire 8-bit prescaler. For the 32/64-bit Wide GPTM, this field contains the lower 8-bits of the 16-bit prescaler.

(Same for Timer B)

**Timer A/B Prescale Match: TAPMR/TBPMR, offset 0x040/0x044**

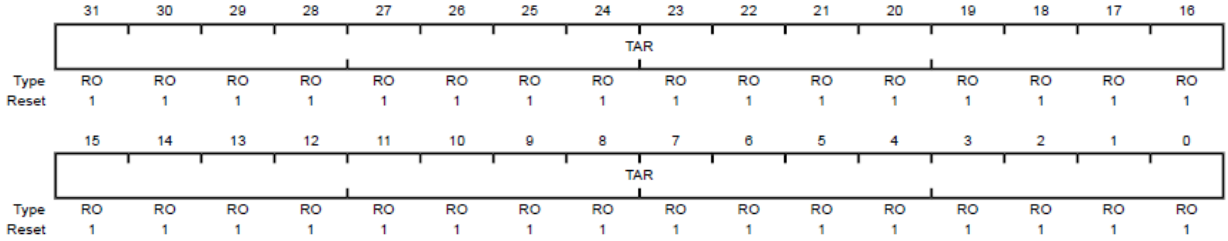



TAPSRH: GPTM Timer A  Match Prescale High Byte. For the 16/32-bit GPTM, this field is reserved. For the 32/64-bit Wide GPTM, this field contains the upper 8-bits of the 16-bit match prescaler.

TAPSRL: GPTM Timer A Match Prescale Low Byte. For the 16/32-bit GPTM, this field contains the entire 8-bit match prescaler. For the 32/64-bit Wide GPTM, this field contains the lower 8-bits of the 16-bit match prescaler.

(Same for Timer B)

**Timer A/B: TAR/ TBR, offset 0x048/0x04C**

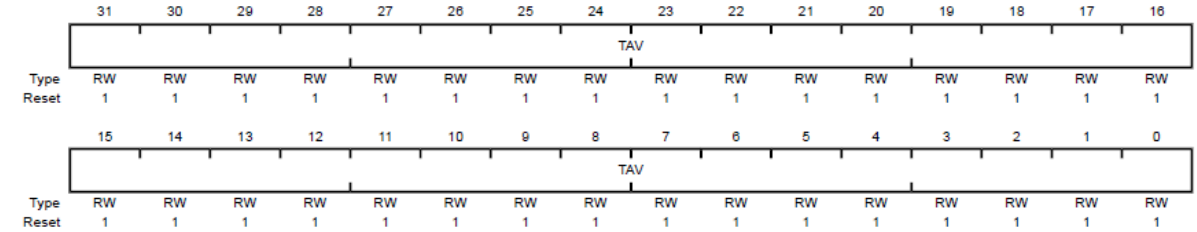



| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:0 | TAR | RO | 0xFFFF.FFFF | GPTM Timer A Register<br>A read returns the current value of the **GPTM Timer A Count Register**, in all cases except for Input Edge Count and Time modes. In the Input Edge Count mode, this register contains the number of edges that have occurred. In the Input Edge Time mode, this register contains the time at which the last edge event took place. |



(Same for Timer B)

**Timer A/B Value : TAV/ TBV, offset 0x050/0x054**




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:0 | TAV | RW | 0xFFFF.FFFF | GPTM Timer A Value<br>A read returns the current, free-running value of Timer A in all modes. When written, the value written into this register is loaded into the **GPTMTAR** register on the next clock cycle. |

(Same for Timer B)

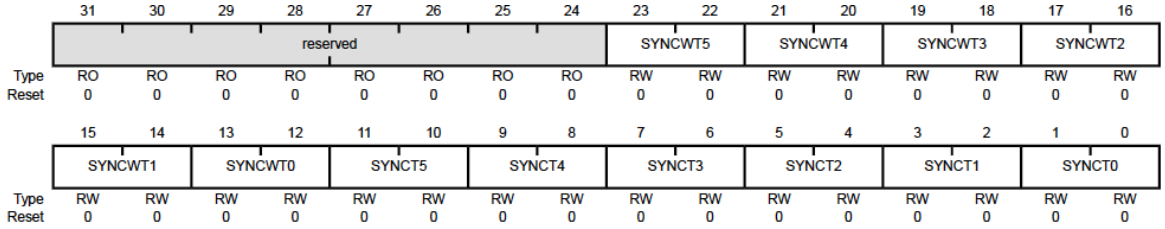
**Timer Peripheral Properties ( PP), offset 0xFC0**

Bit field 3:0 defines the size where

0 Timer A and Timer B counters are 16 bits each with an 8-bit prescale counter.
1 Timer A and Timer B counters are 32 bits each with a 16-bit prescale counter.

**Timer Synchronize ( SYNC), offset 0x010**

This register is only implemented on timer Module 0 only. This register allows software to synchronize a number of timers.




Two bits for each timer to synchronize. For a timer Timer n, bit values and descriptions are as follows:

0x0 Timer n is not affected.
0x1 A timeout event for Timer A of Timer n is triggered.
0x2 A timeout event for Timer B of Timer n is triggered.
0x3 A timeout event for both Timer A and Timer B of Timer n is triggered.

# Analog-to-Digital Converter

## Base addresses of the ADC modules are:

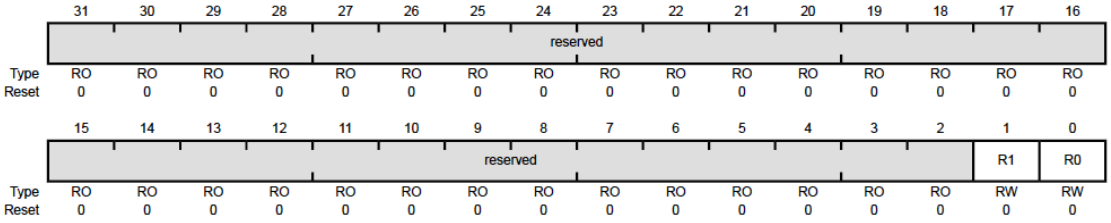
ADC0: 0x4003.8000

ADC1: 0x4003.9000

## Sample sequencers:

| Sequencer | Number of Samples | Depth of FIFO |
|---|---|---|
| SS3 | 1 | 1 |
| SS2 | 4 | 4 |
| SS1 | 4 | 4 |
| SS0 | 8 | 8 |

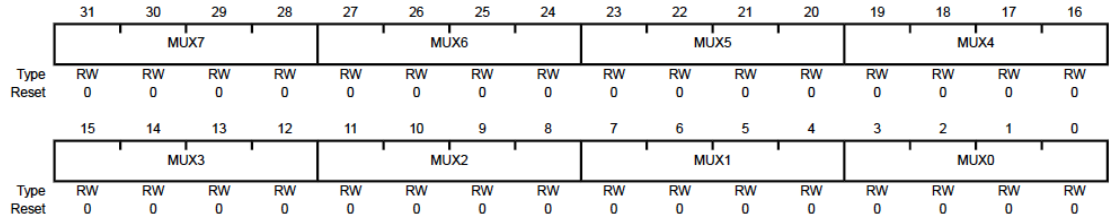
## Analog-to-Digital Converter Run Mode Clock Gating Control: RCGCADC, 0x400FE638




Rn=0 ADC module n is disabled.

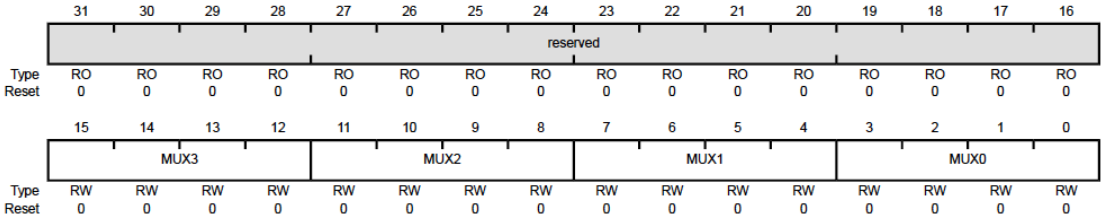
Rn=1 Enable and provide a clock to ADC module n in Run mode.

## ADC Sample Sequence Input Multiplexer Select 0: ADCSSMUX0, offset 0x040

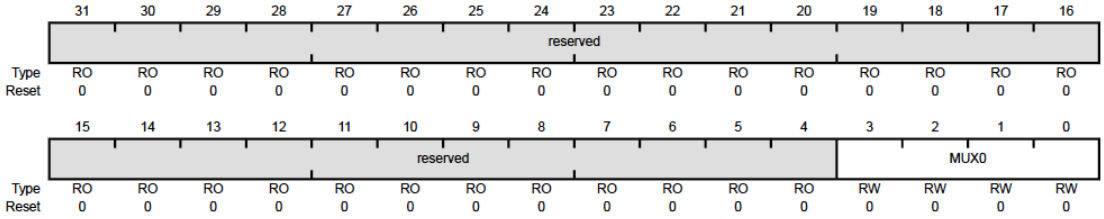



## ADC Sample Sequence Input Multiplexer Select 1 (ADCSSMUX1), offset 0x060

## ADC Sample Sequence Input Multiplexer Select 2 (ADCSSMUX2), offset 0x080

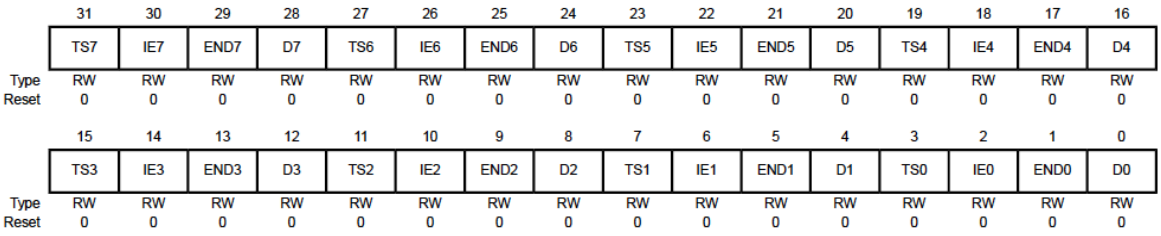



## ADC Sample Sequence Input Multiplexer Select 3 (ADCSSMUX3), offset 0x0A0



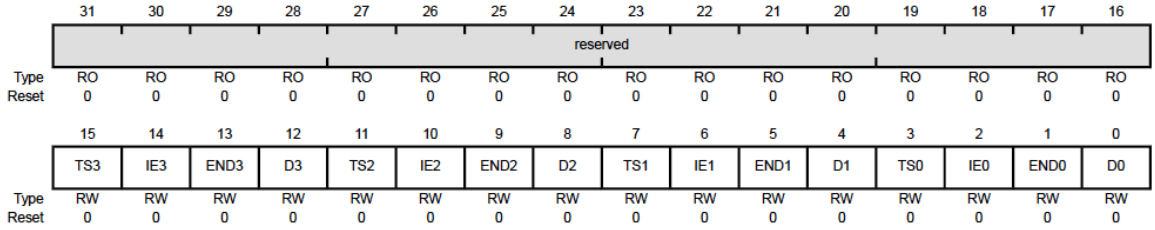

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 31:28 | MUX7 | RW | 0x0 | 8th Sample Input Select<br>The MUX7 field is used during the eighth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. The value set here indicates the corresponding pin, for example, a value of 0x1 indicates the input is AIN1. |
| 27:24 | MUX6 | RW | 0x0 | 7th Sample Input Select<br>The MUX6 field is used during the seventh sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 23:20 | MUX5 | RW | 0x0 | 6th Sample Input Select<br>The MUX5 field is used during the sixth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 19:16 | MUX4 | RW | 0x0 | 5th Sample Input Select<br>The MUX4 field is used during the fifth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 15:12 | MUX3 | RW | 0x0 | 4th Sample Input Select<br>The MUX3 field is used during the fourth sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 11:8 | MUX2 | RW | 0x0 | 3rd Sample Input Select<br>The MUX2 field is used during the third sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 7:4 | MUX1 | RW | 0x0 | 2nd Sample Input Select<br>The MUX1 field is used during the second sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |
| 3:0 | MUX0 | RW | 0x0 | 1st Sample Input Select<br>The MUX0 field is used during the first sample of a sequence executed with the sample sequencer. It specifies which of the analog inputs is sampled for the analog-to-digital conversion. |

## ADC Sample Sequence Control 0 (ADCSSCTL0), offset 0x044

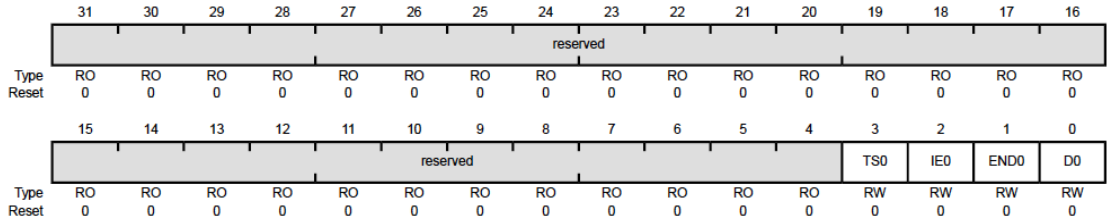



## ADC Sample Sequence Control 1 (ADCSSCTL1), offset 0x064

## ADC Sample Sequence Control 2 (ADCSSCTL2), offset 0x084.




## ADC Sample Sequence Control 3 (ADCSSCTL3), offset 0x0A4




| | | | | |
|---|---|---|---|---|
| 3 | TS0 | RW | 0 | 1st Sample Temp Sensor Select |

| Value | Description |
|---|---|
| 0 | The input pin specified by the **ADCSSMUXn** register is read during the first sample of the sample sequence. |
| 1 | The temperature sensor is read during the first sample of the sample sequence. |

| | | | | |
|---|---|---|---|---|
| 2 | IE0 | RW | 0 | 1st Sample Interrupt Enable |

| Value | Description |
|---|---|
| 0 | The raw interrupt is not asserted to the interrupt controller. |
| 1 | The raw interrupt signal (INR0 bit) is asserted at the end of the first sample's conversion. If the MASK0 bit in the **ADCIM** register is set, the interrupt is promoted to the interrupt controller. |

It is legal to have multiple samples within a sequence generate interrupts.

| | | | | |
|---|---|---|---|---|
| 1 | END0 | RW | 0 | 1st Sample is End of Sequence |

| Value | Description |
|---|---|
| 0 | Another sample in the sequence is the final sample. |
| 1 | The first sample is the last sample of the sequence. |

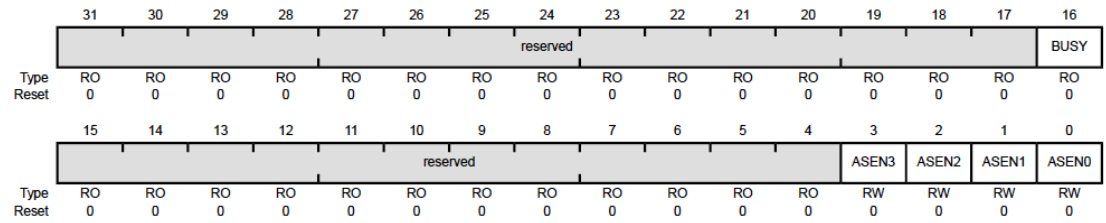
It is possible to end the sequence on any sample position. Software must set an ENDn bit somewhere within the sequence. Samples defined after the sample containing a set ENDn bit are not requested for conversion even though the fields may be non-zero.

| | | | | |
|---|---|---|---|---|
| 0 | D0 | RW | 0 | 1st Sample Differential Input Select |

| Value | Description |
|---|---|
| 0 | The analog inputs are not differentially sampled. |
| 1 | The analog input is differentially sampled. The corresponding **ADCSSMUXn** nibble must be set to the pair number "i", where the paired inputs are "2i and 2i+1". |

Because the temperature sensor does not have a differential option, this bit must not be set when the TS0 bit is set.

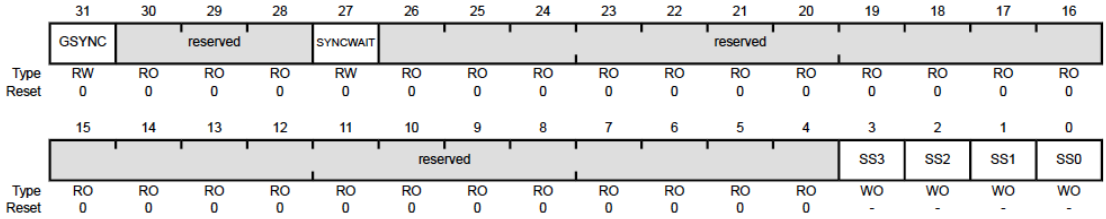
## ADC Active Sample Sequencer (ADCACTSS), offset 0x000



ASENn=0, sample sequencer n is disabled,
ASENn=1, sample sequencer n is enabled.

BUSY=0, ADC is idle,
BUSY=1, ADC is busy.

**ADC Processor Sample Sequence Initiate (ADCPSSI), offset 0x028**

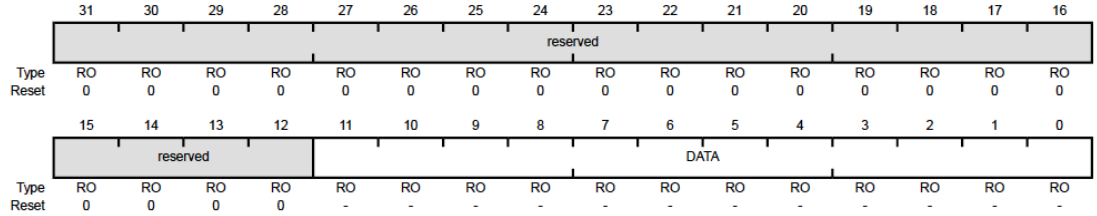



GSYNC bit is cleared once sampling has been initiated.
GSYNC=1 This bit initiates sampling in multiple ADC modules at the same time. Any ADC module that has been initialized by setting an SSn bit and the SYNCWAIT bit starts sampling once this bit is written.

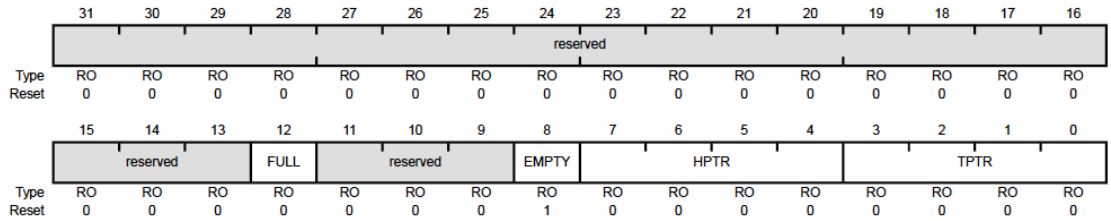
SYNCWAIT=0 Sampling begins when a sample sequence has been initiated.
SYNCWAIT=1 This bit allows the sample sequences to be initiated, but delays sampling until the GSYNC bit is set.

SSn=0 No effect.
SSn=1 Begin sampling on Sample Sequencer n, if the sequencer is enabled in the ADCACTSS register.

**ADC Sample Sequence Result FIFO 0 (ADCSSFIFO0), offset 0x048**
**ADC Sample Sequence Result FIFO 1 (ADCSSFIFO1), offset 0x068**
**ADC Sample Sequence Result FIFO 2 (ADCSSFIFO2), offset 0x088**
**ADC Sample Sequence Result FIFO 3 (ADCSSFIFO3), offset 0x0A8**




**ADC Sample Sequence FIFO 0 Status (ADCSSFSTAT0), offset 0x04C**
**ADC Sample Sequence FIFO 1 Status (ADCSSFSTAT1), offset 0x06C**
**ADC Sample Sequence FIFO 2 Status (ADCSSFSTAT2), offset 0x08C**
**ADC Sample Sequence FIFO 3 Status (ADCSSFSTAT3), offset 0x0AC**

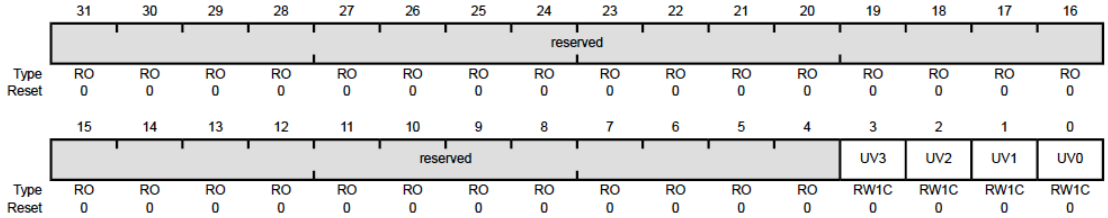



FULL=0 The FIFO is not currently full.
FULL=1 The FIFO is currently full.

EMPTY=0 The FIFO is not currently empty.
EMPTY=1 The FIFO is currently empty.

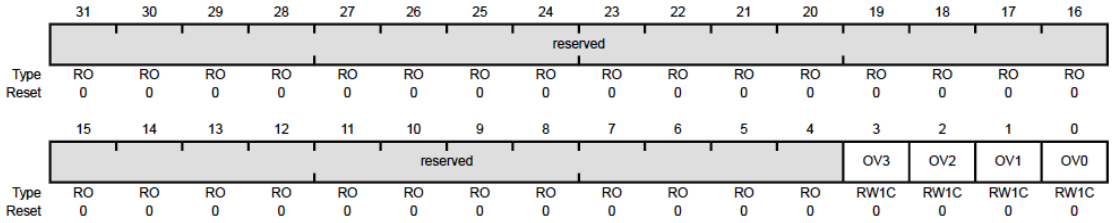
HPTR / TPTR: the current "head" / "tail" pointer index for the FIFO, that is, the next entry to be written.

## ADC Underflow Status (ADCUSTAT), offset 0x018



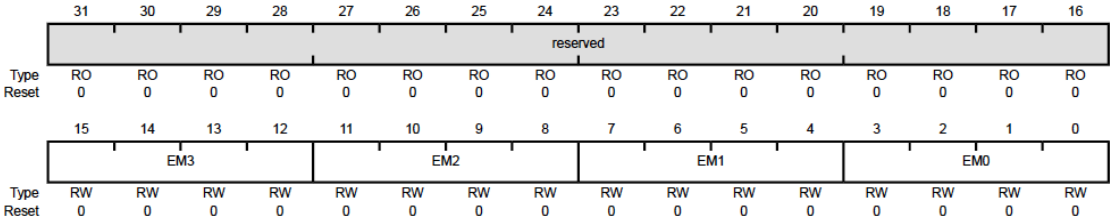

UVn=0 The FIFO has not underflowed.
UVn=1 The FIFO for the Sample Sequencer has hit an underflow condition, meaning that the FIFO is empty and a read was requested. The problematic read does not move the FIFO pointers, and 0s are returned.
This bit is cleared by writing a 1.

## ADC Overflow Status (ADCOSTAT), offset 0x010




OVn=0 The FIFO has not overflowed.
OVn=1 The FIFO for Sample Sequencer n has hit an overflow condition, meaning that the FIFO is full and a write was requested. When an overflow is detected, the most recent write is dropped.
This bit is cleared by writing a 1.

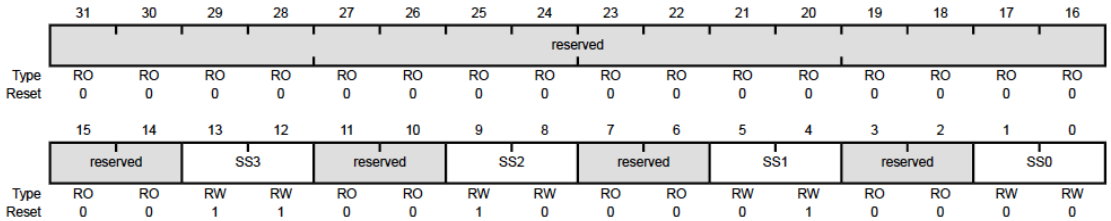
## ADC Event Multiplexer Select (ADCEMUX), offset 0x014



| Name | Type | Reset | Description |
|---|---|---|---|
| EMn | RW | 0x0 | SSn Trigger Select<br>This field selects the trigger source for Sample Sequencern .<br>The valid configurations for this field are: |

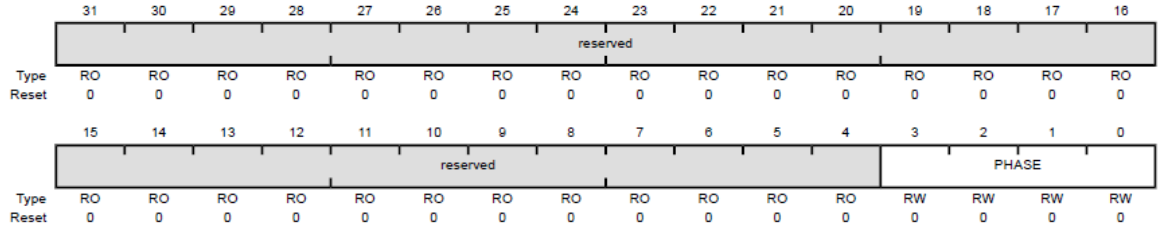
| Value | Event |
|---|---|
| 0x0 | Processor (default)<br>The trigger is initiated by setting the SSn bit in the **ADCPSSI** register. |
| 0x1 | Analog Comparator 0<br>This trigger is configured by the **Analog Comparator Control 0 (ACCTL0)** register |
| 0x2 | Analog Comparator 1<br>This trigger is configured by the **Analog Comparator Control 1 (ACCTL1)** register |
| 0x3 | reserved |
| 0x4 | External (GPIO Pins)<br>This trigger is connected to the GPIO interrupt for the corresponding GPIO (see "ADC Trigger Source" on page 655).<br>**Note:** GPIOs that have AINx signals as alternate functions can be used to trigger the ADC. However, the pin cannot be used as both a GPIO and an analog input. |
| 0x5 | Timer<br>In addition, the trigger must be enabled with the TnOTE bit in the **GPTMCTL** register |
| 0x6 | PWM generator 0<br>The PWM generator 0 trigger can be configured with the **PWM0 Interrupt and Trigger Enable (PWM0INTEN)** register |
| 0x7 | PWM generator 1<br>The PWM generator 1 trigger can be configured with the **PWM1INTEN** register |
| 0x8 | PWM generator 2<br>The PWM generator 2 trigger can be configured with the **PWM2INTEN** register |
| 0x9 | PWM generator 3<br>The PWM generator 3 trigger can be configured with the **PWM3INTEN** register |
| 0xA-0xE | reserved |
| 0xF | Always (continuously sample) |

## ADC Sample Sequencer Priority (ADCSSPRI), offset 0x020




SSn: This field contains a binary-encoded value that specifies the priority encoding of Sample Sequencer n. A priority encoding of 0x0 is highest and 0x3 is lowest.

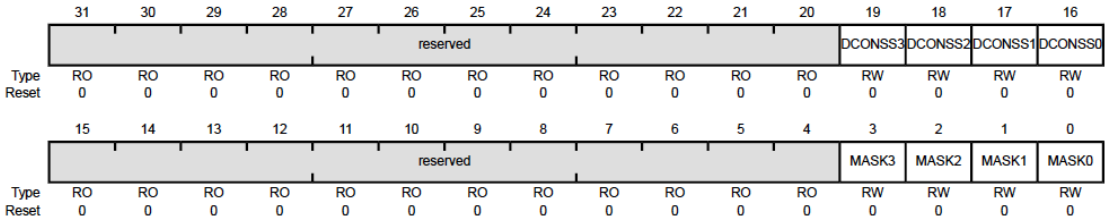
## ADC Sample Phase Control: ADCSPC




| Name | Type | Reset | Description |
|---|---|---|---|
| PHASE | RW | 0x0 | Phase Difference<br>This field selects the sample phase difference from the standard sample time. |

| Value | Description |
|---|---|
| 0x0 | ADC sample lags by 0.0° |
| 0x1 | ADC sample lags by 22.5° |
| 0x2 | ADC sample lags by 45.0° |
| 0x3 | ADC sample lags by 67.5° |
| 0x4 | ADC sample lags by 90.0° |
| 0x5 | ADC sample lags by 112.5° |
| 0x6 | ADC sample lags by 135.0° |
| 0x7 | ADC sample lags by 157.5° |
| 0x8 | ADC sample lags by 180.0° |
| 0x9 | ADC sample lags by 202.5° |
| 0xA | ADC sample lags by 225.0° |
| 0xB | ADC sample lags by 247.5° |
| 0xC | ADC sample lags by 270.0° |
| 0xD | ADC sample lags by 292.5° |
| 0xE | ADC sample lags by 315.0° |
| 0xF | ADC sample lags by 337.5° |

## ADC Interrupt Mask (ADCIM), offset 0x008



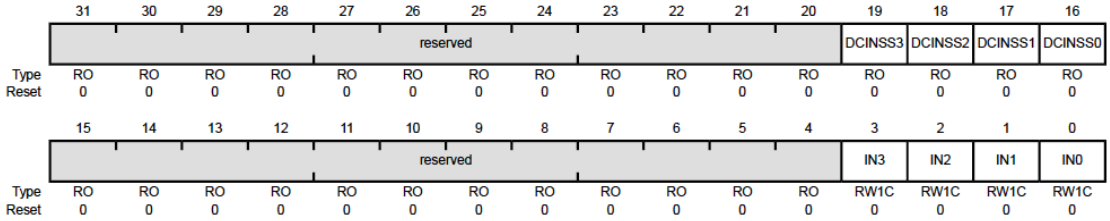

DCONSSn=0 The status of the digital comparators does not affect the SSn interrupt status.

DCONSSn=1 The raw interrupt signal from the digital comparators (INRDC bit in the ADCRIS register) is sent to the interrupt controller on the SSn interrupt line.

MASKn=0 The status of Sample Sequencer n does not affect the SSn interrupt status.

MASKn=1 The raw interrupt signal from Sample Sequencer 3 (ADCRIS register INRn bit) is sent to the interrupt controller.

## ADC Interrupt Status and Clear (ADCISC), offset 0x00C

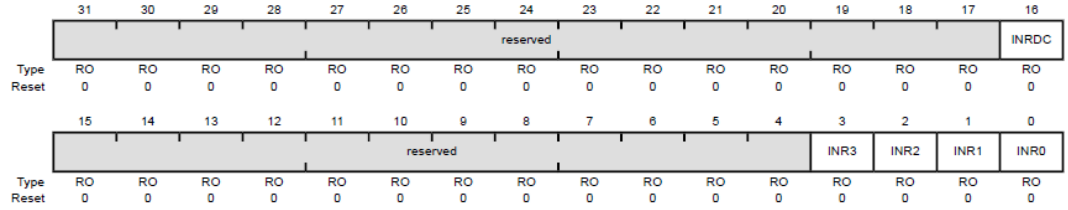



DCINSSn=0 No interrupt has occurred or the interrupt is masked. Both the INRDC bit in the ADCRIS register and the DCONSSn bit in the ADCIM register are set, providing a level-based interrupt to the interrupt controller.

DCINSSn=1 This bit is cleared by writing a 1 to it. Clearing this bit also clears the INRDC bit in the ADCRIS register.

INn=0 No interrupt has occurred or the interrupt is masked.
INn=1 Both the INRn bit in the ADCRIS register and the MASKn bit in the ADCIM register are set, providing a level-based interrupt to the interrupt controller.
This bit is cleared by writing a 1. Clearing this bit also clears the INRn bit in the ADCRIS register.

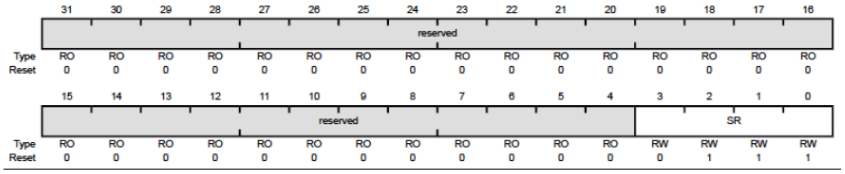
## ADC Raw Interrupt Status (ADCRIS), offset 0x004




INRn = 0 An interrupt has not occurred.
INRn = 1 A sample has completed conversion and if the respective ADCSSCTLn IEx bit is set, enabling a raw interrupt.

This bit is cleared by writing a 1 to the INn bit in the ADCISC register.

## ADC Peripheral Configuration (ADCPC), offset 0xFC4

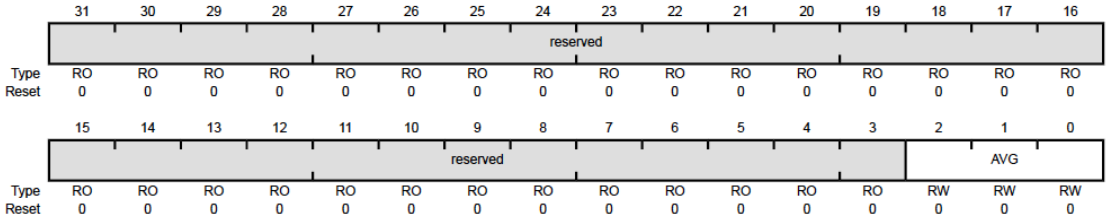



ADC Sample Rate

This field specifies the number of ADC conversions per second and is used in Run, Sleep, and Deep-Sleep modes. The field encoding is based on the legacy RCGC0 register encoding. The programmed sample rate cannot exceed the maximum sample rate specified by the MSR field in the ADCPP register. The SR field is encoded as follows:

| Value | Description |
|---|---|
| 0x0 | Reserved |
| 0x1 | 125 ksps |
| 0x2 | Reserved |
| 0x3 | 250 ksps |
| 0x4 | Reserved |
| 0x5 | 500 ksps |
| 0x6 | Reserved |
| 0x7 | 1 Msps |
| 0x8 - 0xF | Reserved |

## ADC Sample Averaging Control (ADCSAC), offset 0x030




| | |
|---|---|
| 0x0 | No hardware oversampling |
| 0x1 | 2x hardware oversampling |
| 0x2 | 4x hardware oversampling |
| 0x3 | 8x hardware oversampling |
| 0x4 | 16x hardware oversampling |
| 0x5 | 32x hardware oversampling |
| 0x6 | 64x hardware oversampling |
| 0x7 | reserved |

# Serial I/O

## UART

UART0 base: 0x4000.C000
UART1 base: 0x4000.D000
UART2 base: 0x4000.E000
UART3 base: 0x4000.F000
UART4 base: 0x4001.0000
UART5 base: 0x4001.1000
UART6 base: 0x4001.2000
UART7 base: 0x4001.3000

### Run Clock Gate Control Register for UART (RCGCUART) (address is 0x400FE618)

Base 0x400F.E000
Offset 0x618
Type RW, reset 0x0000.0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | | | | | | | | | | | | |
| RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | | | | | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| RO | RO | RO | RO | RO | RO | RO | RO | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### UART Data (UARTDR), offset 0x000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | | | | | | | | | | | | |
| RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | OE | BE | PE | FE | DATA | | | | | | | |
| RO | RO | RO | RO | RO | RO | RO | RO | RW | RW | RW | RW | RW | RW | RW | RW |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### UART Receive Status/Error Clear (UARTRSR/UARTECR), offset 0x004

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | | | | | | | | | | | | |
| RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reserved | | | | | | | | | | | | OE | BE | PE | FE |
| RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 3 | OE | RO | 0 | UART Overrun Error |

| Value | Description |
|---|---|
| 0 | No data has been lost due to a FIFO overrun. |
| 1 | New data was received when the FIFO was full, resulting in data loss. |

This bit is cleared by a write to **UARTECR**.

The FIFO contents remain valid because no further data is written when the FIFO is full, only the contents of the shift register are overwritten. The CPU must read the data in order to empty the FIFO.

| | | | | |
|---|---|---|---|---|
| 2 | BE | RO | 0 | UART Break Error |

| Value | Description |
|---|---|
| 0 | No break condition has occurred |
| 1 | A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). |

This bit is cleared to 0 by a write to **UARTECR**.

In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.

| | | | | |
|---|---|---|---|---|
| 1 | PE | RO | 0 | UART Parity Error |

| Value | Description |
|---|---|
| 0 | No parity error has occurred |
| 1 | The parity of the received data character does not match the parity defined by bits 2 and 7 of the **UARTLCRH** register. |

This bit is cleared to 0 by a write to **UARTECR**.

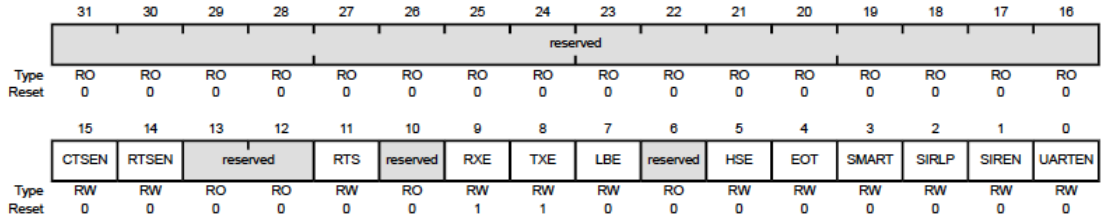
| | | | | |
|---|---|---|---|---|
| 0 | FE | RO | 0 | UART Framing Error |

| Value | Description |
|---|---|
| 0 | No framing error has occurred |
| 1 | The received character does not have a valid stop bit (a valid stop bit is 1). |

This bit is cleared to 0 by a write to **UARTECR**.

In FIFO mode, this error is associated with the character at the top of the FIFO.

## UART Control (UARTCTL), offset 0x030

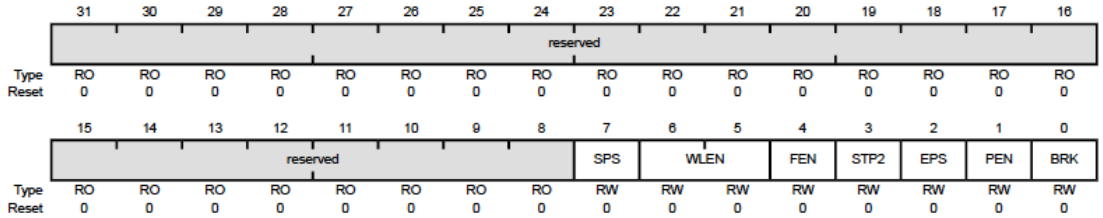



| | | | | |
|---|---|---|---|---|
| 15 | CTSEN | RW | 0 | Enable Clear To Send |

| Value | Description |
|---|---|
| 0 | CTS hardware flow control is disabled. |
| 1 | CTS hardware flow control is enabled. Data is only transmitted when the `U1CTS` signal is asserted. |

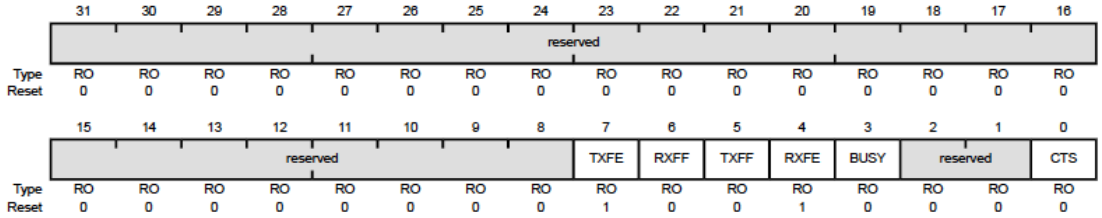
| | | | | |
|---|---|---|---|---|
| 14 | RTSEN | RW | 0 | Enable Request to Send<br><br>Value Description<br>0 RTS hardware flow control is disabled.<br>1 RTS hardware flow control is enabled. Data is only requested (by asserting U1RTS) when the receive FIFO has available entries. |
| 11 | RTS | RW | 0 | Request to Send<br>When RTSEN is clear, the status of this bit is reflected on the U1RTS signal. If RTSEN is set, this bit is ignored on a write and should be ignored on read. |
| 9 | RXE | RW | 1 | UART Receive Enable<br><br>Value Description<br>0 The receive section of the UART is disabled.<br>1 The receive section of the UART is enabled.<br><br>If the UART is disabled in the middle of a receive, it completes the current character before stopping.<br><br>**Note:** To enable reception, the UARTEN bit must also be set. |
| 8 | TXE | RW | 1 | UART Transmit Enable<br><br>Value Description<br>0 The transmit section of the UART is disabled.<br>1 The transmit section of the UART is enabled.<br><br>If the UART is disabled in the middle of a transmission, it completes the current character before stopping.<br><br>**Note:** To enable transmission, the UARTEN bit must also be set. |
| 7 | LBE | RW | 0 | UART Loop Back Enable<br><br>Value Description<br>0 Normal operation.<br>1 The UnTx path is fed through the UnRx path. |
| 5 | HSE | RW | 0 | High-Speed Enable<br><br>Value Description<br>0 The UART is clocked using the system clock divided by 16.<br>1 The UART is clocked using the system clock divided by 8.<br><br>**Note:** System clock used is also dependent on the baud-rate divisor configuration (see page 914) and page 915).<br><br>The state of this bit has no effect on clock generation in ISO 7816 smart card mode (the SMART bit is set). |
| 4 | EOT | RW | 0 | End of Transmission<br>This bit determines the behavior of the TXRIS bit in the **UARTRIS** register.<br><br>Value Description<br>0 The TXRIS bit is set when the transmit FIFO condition specified in **UARTIFLS** is met.<br>1 The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer. |
| 0 | UARTEN | RW | 0 | UART Enable<br><br>Value Description<br>0 The UART is disabled.<br>1 The UART is enabled.<br><br>If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. |

## UART Line Control (UARTLCRH), offset 0x02C



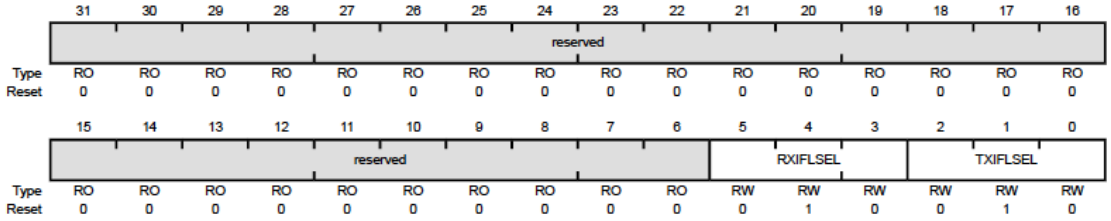

| | | | | |
|---|---|---|---|---|
| 7 | SPS | RW | 0 | UART Stick Parity Select<br>When bits 1, 2, and 7 of **UARTLCRH** are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set and 2 is cleared, the parity bit is transmitted and checked as a 1.<br>When this bit is cleared, stick parity is disabled. |
| 6:5 | WLEN | RW | 0x0 | UART Word Length<br>The bits indicate the number of data bits transmitted or received in a frame as follows:<br>Value Description<br>0x0 5 bits (default)<br>0x1 6 bits<br>0x2 7 bits<br>0x3 8 bits |
| 4 | FEN | RW | 0 | UART Enable FIFOs<br>Value Description<br>0 The FIFOs are disabled (Character mode). The FIFOs become 1-byte-deep holding registers.<br>1 The transmit and receive FIFO buffers are enabled (FIFO mode). |
| 3 | STP2 | RW | 0 | UART Two Stop Bits Select<br>Value Description<br>0 One stop bit is transmitted at the end of a frame.<br>1 Two stop bits are transmitted at the end of a frame. The receive logic does not check for two stop bits being received.<br>When in 7816 smartcard mode (the SMART bit is set in the **UARTCTL** register), the number of stop bits is forced to 2. |
| 2 | EPS | RW | 0 | UART Even Parity Select<br>Value Description<br>0 Odd parity is performed, which checks for an odd number of 1s.<br>1 Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits.<br>This bit has no effect when parity is disabled by the PEN bit. |
| 1 | PEN | RW | 0 | UART Parity Enable<br>Value Description<br>0 Parity is disabled and no parity bit is added to the data frame.<br>1 Parity checking and generation is enabled. |
| 0 | BRK | RW | 0 | UART Send Break<br>Value Description<br>0 Normal use.<br>1 A Low level is continually output on the UnTx signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods). |

## UART Flag (UARTFR), offset 0x018




| | | | | |
|---|---|---|---|---|
| 7 | TXFE | RO | 1 | UART Transmit FIFO Empty<br>The meaning of this bit depends on the state of the FEN bit in the **UARTLCRH** register.<br><br>Value Description<br>0 The transmitter has data to transmit.<br>1 If the FIFO is disabled (FEN is 0), the transmit holding register is empty.<br>If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. |
| 6 | RXFF | RO | 0 | UART Receive FIFO Full<br>The meaning of this bit depends on the state of the FEN bit in the **UARTLCRH** register.<br><br>Value Description<br>0 The receiver can receive data.<br>1 If the FIFO is disabled (FEN is 0), the receive holding register is full.<br>If the FIFO is enabled (FEN is 1), the receive FIFO is full. |
| 5 | TXFF | RO | 0 | UART Transmit FIFO Full<br>The meaning of this bit depends on the state of the FEN bit in the **UARTLCRH** register.<br><br>Value Description<br>0 The transmitter is not full.<br>1 If the FIFO is disabled (FEN is 0), the transmit holding register is full.<br>If the FIFO is enabled (FEN is 1), the transmit FIFO is full. |
| 4 | RXFE | RO | 1 | UART Receive FIFO Empty<br>The meaning of this bit depends on the state of the FEN bit in the **UARTLCRH** register.<br><br>Value Description<br>0 The receiver is not empty.<br>1 If the FIFO is disabled (FEN is 0), the receive holding register is empty.<br>If the FIFO is enabled (FEN is 1), the receive FIFO is empty. |
| 3 | BUSY | RO | 0 | UART Busy<br><br>Value Description<br>0 The UART is not busy.<br>1 The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register.<br><br>This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled). |
| 0 | CTS | RO | 0 | Clear To Send<br><br>Value Description<br>0 The UlCTS signal is not asserted.<br>1 The UlCTS signal is asserted. |

## UART Interrupt FIFO Level Select (UARTIFLS), offset 0x034



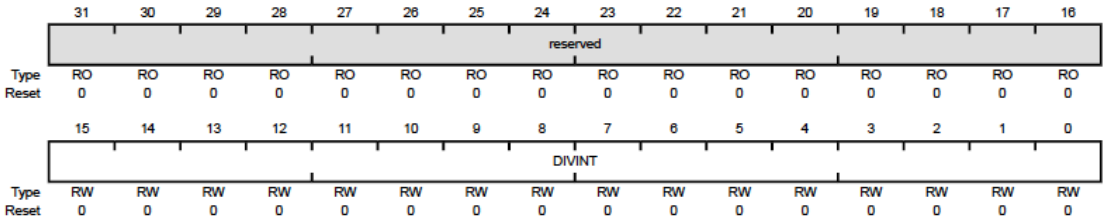

| | | | | |
|---|---|---|---|---|
| 5:3 | RXIFLSEL | RW | 0x2 | UART Receive Interrupt FIFO Level Select<br>The trigger points for the receive interrupt are as follows: |

| Value | Description |
|---|---|
| 0x0 | RX FIFO ≥ ⅛ full |
| 0x1 | RX FIFO ≥ ¼ full |
| 0x2 | RX FIFO ≥ ½ full (default) |
| 0x3 | RX FIFO ≥ ¾ full |
| 0x4 | RX FIFO ≥ ⅞ full |
| 0x5-0x7 | Reserved |

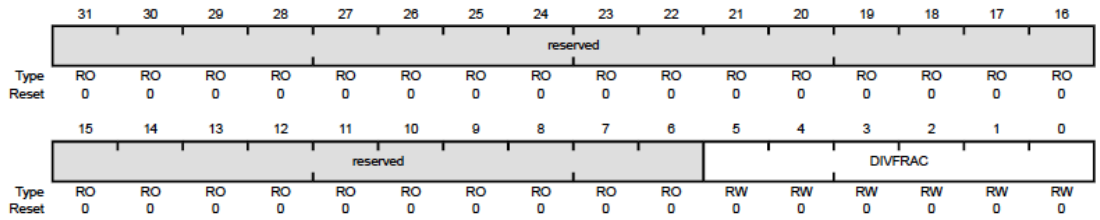
| | | | | |
|---|---|---|---|---|
| 2:0 | TXIFLSEL | RW | 0x2 | UART Transmit Interrupt FIFO Level Select<br>The trigger points for the transmit interrupt are as follows: |

| Value | Description |
|---|---|
| 0x0 | TX FIFO ≤ ⅞ empty |
| 0x1 | TX FIFO ≤ ¾ empty |
| 0x2 | TX FIFO ≤ ½ empty (default) |
| 0x3 | TX FIFO ≤ ¼ empty |
| 0x4 | TX FIFO ≤ ⅛ empty |
| 0x5-0x7 | Reserved |

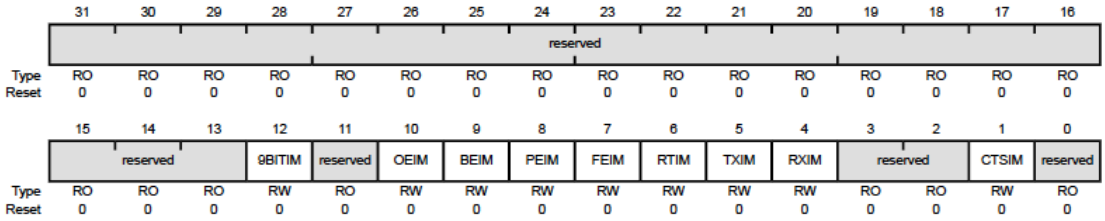
## UART Integer Baud-Rate Divisor (UARTIBRD), offset 0x024




## UART Fractional Baud-Rate Divisor (UARTFBRD), offset 0x028



## UART Interrupt Mask (UARTIM), offset 0x038




| | | | | |
|---|---|---|---|---|
| 12 | 9BITIM | RW | 0 | 9-Bit Mode Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The `9BITRIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `9BITRIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 10 | OEIM | RW | 0 | UART Overrun Error Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The `OERIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `OERIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 9 | BEIM | RW | 0 | UART Break Error Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The `BERIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `BERIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 8 | PEIM | RW | 0 | UART Parity Error Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The `PERIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `PERIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 7 | FEIM | RW | 0 | UART Framing Error Interrupt Mask |

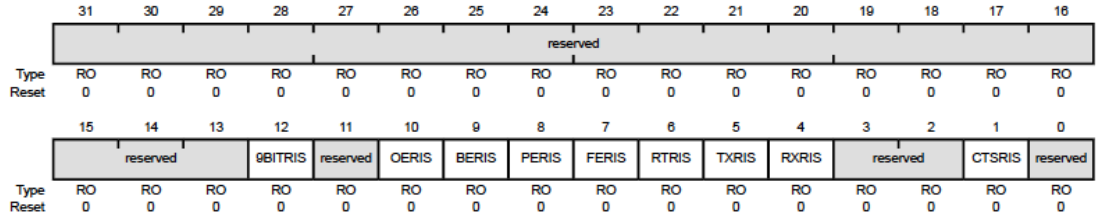
| Value | Description |
|---|---|
| 0 | The `FERIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `FERIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 6 | RTIM | RW | 0 | UART Receive Time-Out Interrupt Mask |

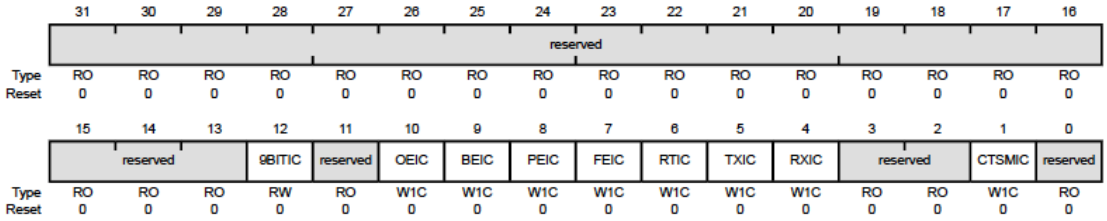
| Value | Description |
|---|---|
| 0 | The `RTRIS` interrupt is suppressed and not sent to the interrupt controller. |
| 1 | An interrupt is sent to the interrupt controller when the `RTRIS` bit in the **UARTRIS** register is set. |

| | | | | |
|---|---|---|---|---|
| 5 | TXIM | RW | 0 | UART Transmit Interrupt Mask<br><br>Value Description<br>0 The TXRIS interrupt is suppressed and not sent to the interrupt controller.<br>1 An interrupt is sent to the interrupt controller when the TXRIS bit in the **UARTRIS** register is set. |
| 4 | RXIM | RW | 0 | UART Receive Interrupt Mask<br><br>Value Description<br>0 The RXRIS interrupt is suppressed and not sent to the interrupt controller.<br>1 An interrupt is sent to the interrupt controller when the RXRIS bit in the **UARTRIS** register is set. |
| 1 | CTSIM | RW | 0 | UART Clear to Send Modem Interrupt Mask<br><br>Value Description<br>0 The CTSRIS interrupt is suppressed and not sent to the interrupt controller.<br>1 An interrupt is sent to the interrupt controller when the CTSRIS bit in the **UARTRIS** register is set.<br><br>This bit is implemented only on UART1 and is reserved for UART0 and UART2. |

## UART Raw Interrupt Status (UARTRIS), offset 0x03C

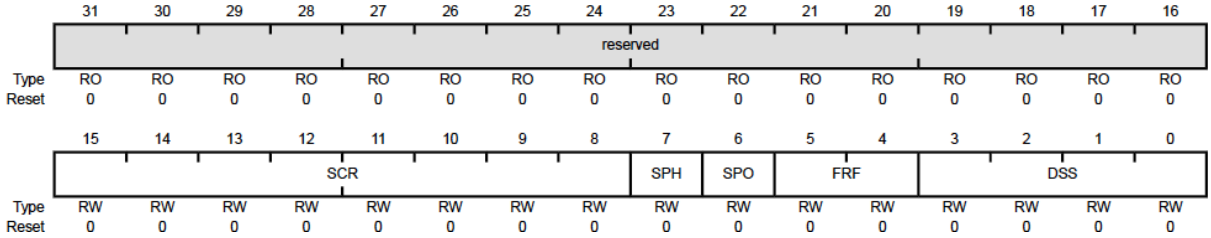



## UART Interrupt Clear (UARTICR), offset 0x044



# SSI

## SSI Control 0 (SSICR0), offset 0x000




| | | | | |
|---|---|---|---|---|
| 15:8 | SCR | RW | 0x00 | SSI Serial Clock Rate<br>This bit field is used to generate the transmit and receive bit rate of the SSI. The bit rate is:<br>`BR=SysClk/(CPSDVSR * (1 + SCR))`<br>where `CPSDVSR` is an even value from 2-254 programmed in the **SSICPSR** register, and `SCR` is a value from 0-255. |
| 7 | SPH | RW | 0 | SSI Serial Clock Phase<br>This bit is only applicable to the Freescale SPI Format.<br>The `SPH` control bit selects the clock edge that captures data and allows it to change state. This bit has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. |

| Value | Description |
|---|---|
| 0 | Data is captured on the first clock edge transition. |
| 1 | Data is captured on the second clock edge transition. |

| | | | | |
|---|---|---|---|---|
| 6 | SPO | RW | 0 | SSI Serial Clock Polarity |

| Value | Description |
|---|---|
| 0 | A steady state Low value is placed on the `SSInClk` pin. |
| 1 | A steady state High value is placed on the `SSInClk` pin when data is not being transferred. |

**Note:** If this bit is set, then software must also configure the GPIO port pin corresponding to the `SSInClk` signal as a pull-up in the **GPIO Pull-Up Select (GPIOPUR)** register.

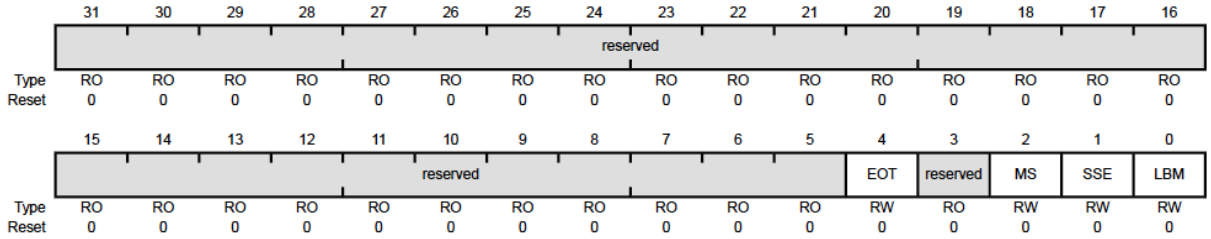
| | | | | |
|---|---|---|---|---|
| 5:4 | FRF | RW | 0x0 | SSI Frame Format Select |

| Value | Frame Format |
|---|---|
| 0x0 | Freescale SPI Frame Format |
| 0x1 | Texas Instruments Synchronous Serial Frame Format |
| 0x2 | MICROWIRE Frame Format |
| 0x3 | Reserved |

| | | | | |
|---|---|---|---|---|
| 3:0 | DSS | RW | 0x0 | SSI Data Size Select |

| Value | Data Size |
|---|---|
| 0x0-0x2 | Reserved |
| 0x3 | 4-bit data |
| 0x4 | 5-bit data |
| 0x5 | 6-bit data |
| 0x6 | 7-bit data |
| 0x7 | 8-bit data |
| 0x8 | 9-bit data |
| 0x9 | 10-bit data |
| 0xA | 11-bit data |
| 0xB | 12-bit data |
| 0xC | 13-bit data |
| 0xD | 14-bit data |
| 0xE | 15-bit data |
| 0xF | 16-bit data |

## SSI Control 1 (SSICR1), offset 0x004




| | | | | |
|---|---|---|---|---|
| 4 | EOT | RW | 0 | End of Transmission |

This bit is only valid for Master mode devices and operations (MS = 0x0).

| Value | Description |
|---|---|
| 0 | The TXRIS interrupt indicates that the transmit FIFO is half full or less. |
| 1 | The End of Transmit interrupt mode for the TXRIS interrupt is enabled. |

| | | | | |
|---|---|---|---|---|
| 2 | MS | RW | 0 | SSI Master/Slave Select |

This bit selects Master or Slave mode and can be modified only when the SSI is disabled (SSE=0).

| Value | Description |
|---|---|
| 0 | The SSI is configured as a master. |
| 1 | The SSI is configured as a slave. |

| | | | | |
|---|---|---|---|---|
| 1 | SSE | RW | 0 | SSI Synchronous Serial Port Enable |

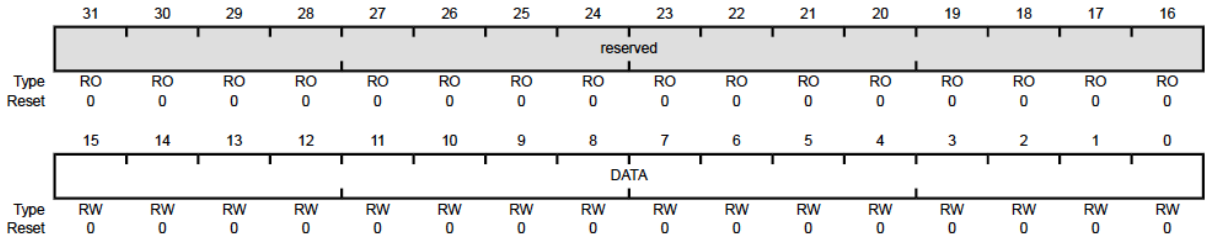
| Value | Description |
|---|---|
| 0 | SSI operation is disabled. |
| 1 | SSI operation is enabled. |

**Note:** This bit must be cleared before any control registers are reprogrammed.

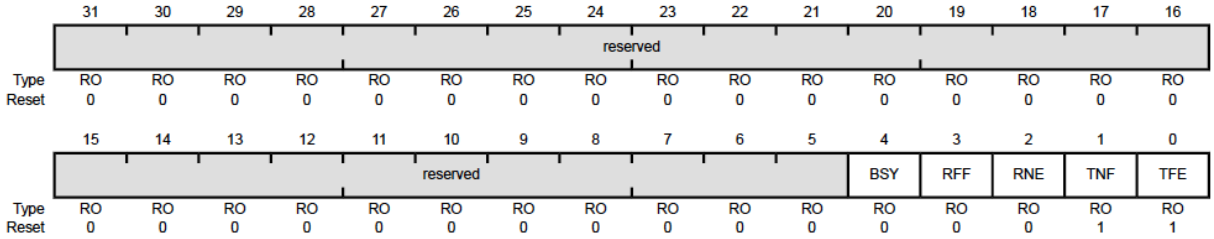
| | | | | |
|---|---|---|---|---|
| 0 | LBM | RW | 0 | SSI Loopback Mode |

| Value | Description |
|---|---|
| 0 | Normal serial port operation enabled. |
| 1 | Output of the transmit serial shift register is connected internally to the input of the receive serial shift register. |

## SSI Data (SSIDR), offset 0x008



## SSI Status (SSISR), offset 0x00C




| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 4 | BSY | RO | 0 | SSI Busy Bit |

| Value | Description |
|---|---|
| 0 | The SSI is idle. |
| 1 | The SSI is currently transmitting and/or receiving a frame, or the transmit FIFO is not empty. |

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 3 | RFF | RO | 0 | SSI Receive FIFO Full |

| Value | Description |
|---|---|
| 0 | The receive FIFO is not full. |
| 1 | The receive FIFO is full. |

| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 2 | RNE | RO | 0 | SSI Receive FIFO Not Empty |

| Value | Description |
|---|---|
| 0 | The receive FIFO is empty. |
| 1 | The receive FIFO is not empty. |

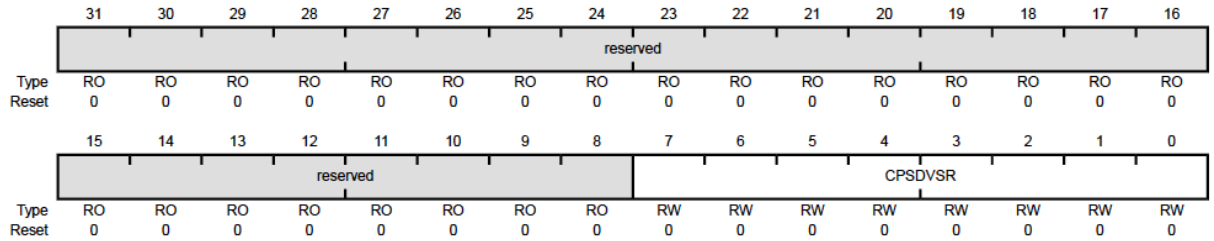
| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 1 | TNF | RO | 1 | SSI Transmit FIFO Not Full |

| Value | Description |
|---|---|
| 0 | The transmit FIFO is full. |
| 1 | The transmit FIFO is not full. |

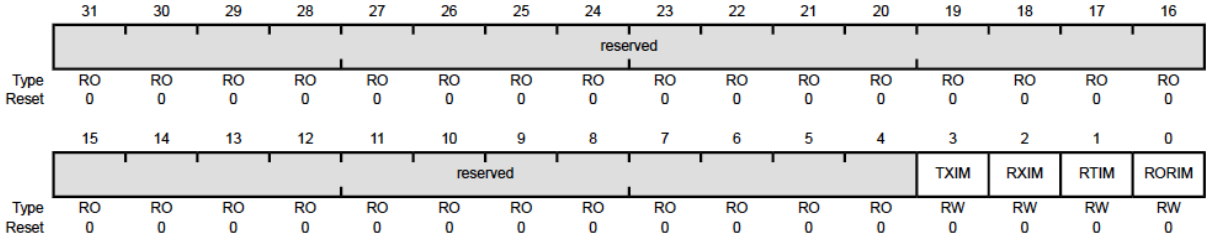
| Bit/Field | Name | Type | Reset | Description |
|---|---|---|---|---|
| 0 | TFE | RO | 1 | SSI Transmit FIFO Empty |

| Value | Description |
|---|---|
| 0 | The transmit FIFO is not empty. |
| 1 | The transmit FIFO is empty. |

## SSI Clock Prescale (SSICPSR), offset 0x010



## SSI Interrupt Mask (SSIIM), offset 0x014




| | | | | |
|---|---|---|---|---|
| 3 | TXIM | RW | 0 | SSI Transmit FIFO Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The transmit FIFO interrupt is masked. |
| 1 | The transmit FIFO interrupt is not masked. |

| | | | | |
|---|---|---|---|---|
| 2 | RXIM | RW | 0 | SSI Receive FIFO Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The receive FIFO interrupt is masked. |
| 1 | The receive FIFO interrupt is not masked. |

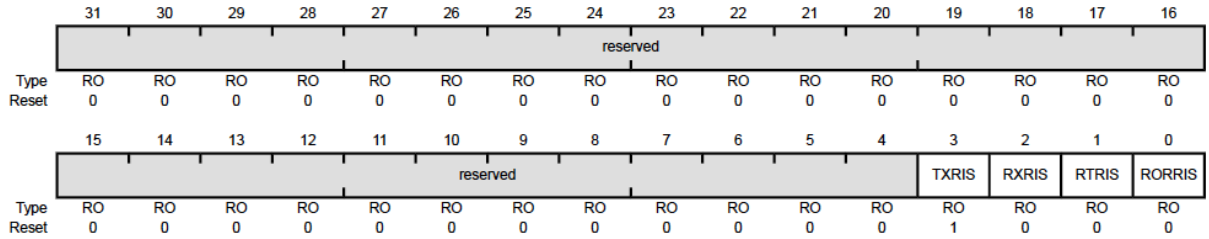
| | | | | |
|---|---|---|---|---|
| 1 | RTIM | RW | 0 | SSI Receive Time-Out Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The receive FIFO time-out interrupt is masked. |
| 1 | The receive FIFO time-out interrupt is not masked. |

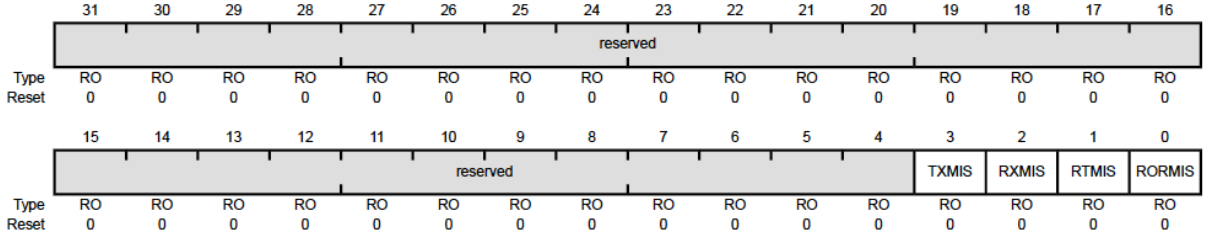
| | | | | |
|---|---|---|---|---|
| 0 | RORIM | RW | 0 | SSI Receive Overrun Interrupt Mask |

| Value | Description |
|---|---|
| 0 | The receive FIFO overrun interrupt is masked. |
| 1 | The receive FIFO overrun interrupt is not masked. |

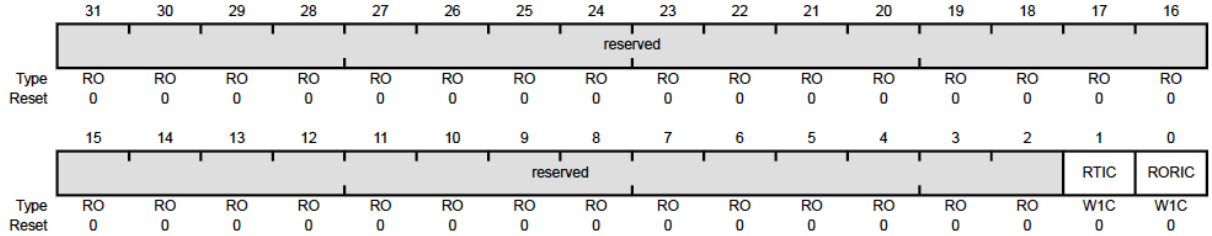
## SSI Raw Interrupt Status (SSIRIS), offset 0x018



## SSI Masked Interrupt Status (SSIMIS), offset 0x01C

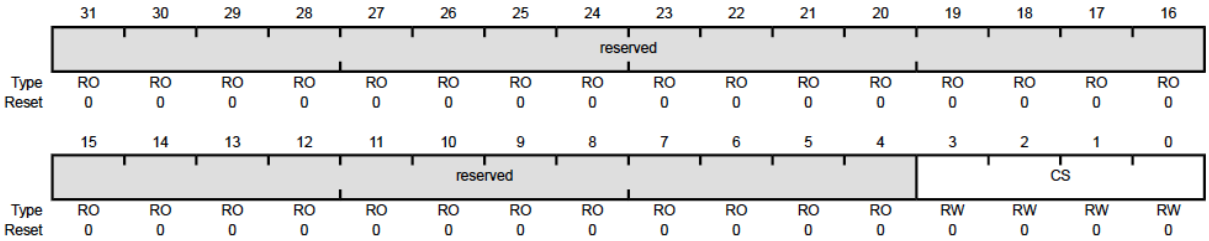



## SSI Interrupt Clear (SSIICR), offset 0x020




| | | | | |
|---|---|---|---|---|
| 1 | RTIC | W1C | 0 | SSI Receive Time-Out Interrupt Clear<br>Writing a 1 to this bit clears the `RTRIS` bit in the **SSIRIS** register and the `RTMIS` bit in the **SSIMIS** register. |
| 0 | RORIC | W1C | 0 | SSI Receive Overrun Interrupt Clear<br>Writing a 1 to this bit clears the `RORRIS` bit in the **SSIRIS** register and the `RORMIS` bit in the **SSIMIS** register. |

## SSI Clock Configuration (SSICC), offset 0xFC8




| | | | | |
|---|---|---|---|---|
| 3:0 | CS | RW | 0 | SSI Baud Clock Source<br>The following table specifies the source that generates for the SSI baud clock: |

| Value | Description |
|---|---|
| 0x0 | System clock (based on clock source and divisor factor) |
| 0x1-0x4 | reserved |
| 0x5 | PIOSC |
| 0x6 - 0xF | Reserved |