CONTENTS:

1. **SysTick Timer Registers** ..... (Page 2)

1.1 SysTick Control and Status Register (STCTRL)

1.2 SysTick Current Register (STCURRENT)

1.3 SysTick Reload Value Register (STRELOAD)

2. **GPIO Data Structure Definition** ..... (Page 4)

3. **GPIO Registers** ..... (Page 5)

3.1 GPIO Data Register (GPIODATA)

3.2. GPIO Direction Register (GPIODIR)

3.3 GPIO Interrupt Both Edges Register (GPIOIBE)

3.4 GPIO Event Register (GPIOIEV)

3.5. GPIO Alternate Function Select Register (GPIOAFSEL)

3.6. GPIO 2mA Drive Register (GPIODR2R)

3.7 GPIO 4-mA Drive Select (GPIODR4R)

3.8 GPIO 8-mA Drive Select (GPIODR8R)

3.9. GIPO Pull Up Register (GPIOPUR)

3.10. GIPO Pull Down Register (GPIOPDR)

3.11. GPIO Digital Enable Register (GPIODEN)

3.12. GPIO Lock Register (GPIOLOCK)

3.13. GPIO Commit Register (GPIOCR)

3.14. GPIO Analog Mode Select Register (GPIOAMSEL)

3.15. GPIO Port Control Register (GPIOPCTL)

3.16. GPIO Interrupt Mask Register (GPIOIM)

3.17. GPIO Masked Interrupt Status Register (GPIOPMIS)

3.18. GPIO Raw Interrupt Status Register (GPIOPRIS)

3.19. GPIO Interrupt Clear Register (GPIOPICR)

4. **System Control Registers** ..... (Page 12)

4.1 GPIO Run Mode Clock Gating Register (RCGCGPIO)

4.2 Peripheral Ready GPIO Register (PRGPIO)

1. SysTick Timer Registers:

1.1 SysTick Control and Status Register (STCTRL)

- The SysTick **STCTRL** register enables the SysTick features.

SysTick Control and Status Register (STCTRL)
Base 0xE000.E000
Offset 0x010
Type RW, reset 0x0000.0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															COUNT
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	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													CLK_SRC	INTEN	ENABLE
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

- **ENABLE:**
  - 1 (SysTick enabled); 0 (SysTick disabled).
- **INTEN:**
  - 0 (SysTick interrupt disabled);
  - 1 (SysTick interrupt generated when count reached 0)
- **CLK_SRC:**
  - 0 (uses the internal oscillator/4);
  - 1 (uses system clock)
- **COUNT flag:**
  - 1 (when SysTick reached count of 0);
  - 0 (0 count not reached);
  - Cleared by a Read to STCURRENT register.

1.2 SysTick Current Register (STCURRENT)

- The **STCURRENT** register contains the current value of the SysTick counter.
- Writing to this register clears the register & COUNT bit of the STCTRL register.

SysTick Current Value Register (STCURRENT)
Base 0xE000.E000
Offset 0x018
Type RWC, reset -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved								CURRENT							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC
Reset	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
	CURRENT															
Type	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

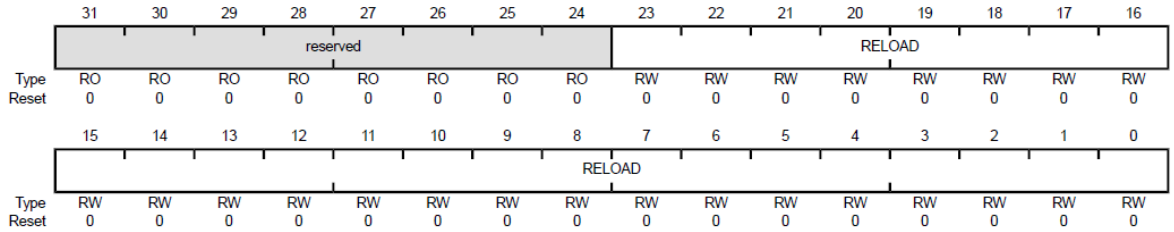
- **CURRENT:** SysTick counter current value.

1.3 SysTick Reload Value Register (STRELOAD)

- The **STRELOAD** register specifies the start value to load into the **SysTick Current Value (STCURRENT)** register when the counter reaches 0.

SysTick Reload Value Register (STRELOAD)

Base 0xE000.E000
Offset 0x014
Type RW, reset -



- **RELOAD:**
  - SysTick counter start (reload) value.

2. GPIO data structure definition:

```
typedef struct {                  /* GPIOA Structure                */
  __I  uint32_t  RESERVED[255];
  __IO uint32_t  DATA;            /* GPIO Data                      */
  __IO uint32_t  DIR;             /* GPIO Direction                 */
  __IO uint32_t  IS;              /* GPIO Interrupt Sense           */
  __IO uint32_t  IBE;             /* GPIO Interrupt Both Edges      */
  __IO uint32_t  IEV;             /* GPIO Interrupt Event           */
  __IO uint32_t  IM;              /* GPIO Interrupt Mask            */
  __IO uint32_t  RIS;             /* GPIO Raw Interrupt Status      */
  __IO uint32_t  MIS;             /* GPIO Masked Interrupt Status   */
  __O  uint32_t  ICR;             /* GPIO Interrupt Clear           */
  __IO uint32_t  AFSEL;           /* GPIO Alternate Function Select */
  __I  uint32_t  RESERVED1[55];
  __IO uint32_t  DR2R;            /* GPIO 2-mA Drive Select         */
  __IO uint32_t  DR4R;            /* GPIO 4-mA Drive Select         */
  __IO uint32_t  DR8R;            /* GPIO 8-mA Drive Select         */
  __IO uint32_t  ODR;             /* GPIO Open Drain Select         */
  __IO uint32_t  PUR;             /* GPIO Pull-Up Select            */
  __IO uint32_t  PDR;             /* GPIO Pull-Down Select          */
  __IO uint32_t  SLR;             /* GPIO Slew Rate Control Select  */
  __IO uint32_t  DEN;             /* GPIO Digital Enable   */
  __IO uint32_t  LOCK;            /* GPIO Lock */
  __IO uint32_t  CR;              /* GPIO Commit    */
  __IO uint32_t  AMSEL;           /* GPIO Analog Mode Select */
  __IO uint32_t  PCTL;            /* GPIO Port Control  */
  __IO uint32_t  ADCCTL;          /* GPIO ADC Control   */
  __IO uint32_t  DMACTL;          /* GPIO DMA Control   */
} GPIOA_Type;
```

The following are the type definitions for the GPIO ports:

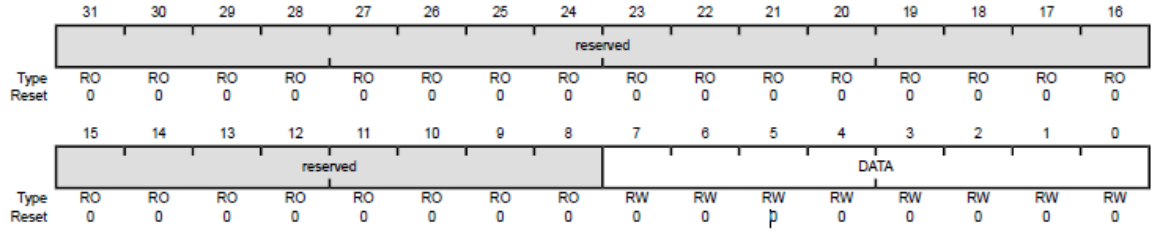
```
#define GPIOA_BASE      0x40004000UL
#define GPIOB_BASE      0x40005000UL
#define GPIOC_BASE      0x40006000UL
#define GPIOD_BASE      0x40007000UL
#define GPIOE_BASE      0x40024000UL
#define GPIOF_BASE      0x40025000UL

#define GPIOA      ((GPIOA_Type *) GPIOA_BASE)
#define GPIOB      ((GPIOA_Type *) GPIOB_BASE)
#define GPIOC      ((GPIOA_Type *) GPIOC_BASE)
#define GPIOD      ((GPIOA_Type *) GPIOD_BASE)
#define GPIOE      ((GPIOA_Type *) GPIOE_BASE)
#define GPIOF      ((GPIOA_Type *) GPIOF_BASE)
```

3. GPIO Registers

3.1 GPIO Data Register (GPIODATA)

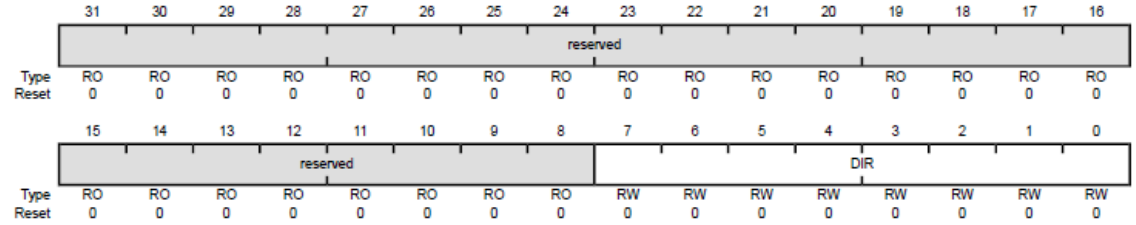
- Values written in the **GPIODATA** register are transferred onto the GPIO port pins if the respective pins have been configured as outputs through the **GPIO Direction (GPIODIR)** register.



- **DATA** = Data (8-bits) to Read or Write.

3.2. GPIO Direction Register (GPIODIR)

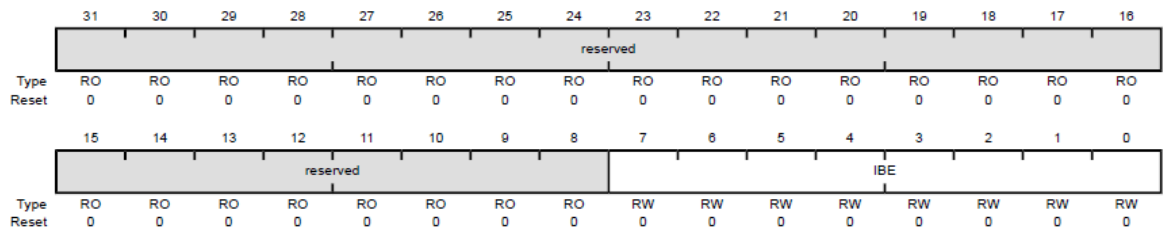
- Setting a bit in the **GPIODIR** register configures the corresponding pin to be an output, while clearing a bit configures the corresponding pin to be an input.
- All bits are cleared by a reset, meaning all GPIO pins are inputs by default.



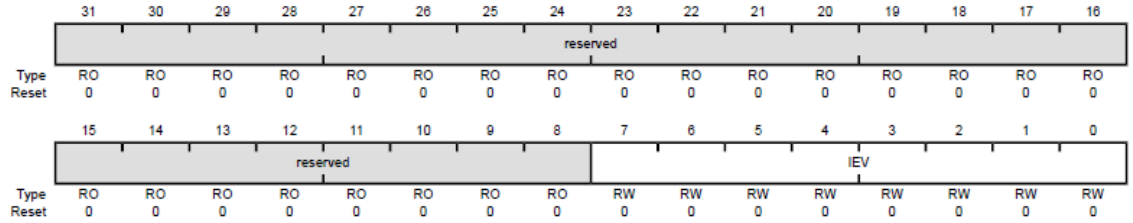
- **DIR:** '0' = Input; '1' = Output

3.3 GPIO Interrupt Both Edges Register (GPIOIBE)

- The **GPIOIBE** register allows both edges to cause interrupts.
- Setting a bit in the **GPIOIBE** register configures the corresponding pin to detect both rising and falling edges, regardless of the corresponding bit in the **GPIO Interrupt Event (GPIOIEV)** register.
- Clearing a bit configures the pin to be controlled by the **GPIOIEV** register.
- All bits are cleared by a reset.



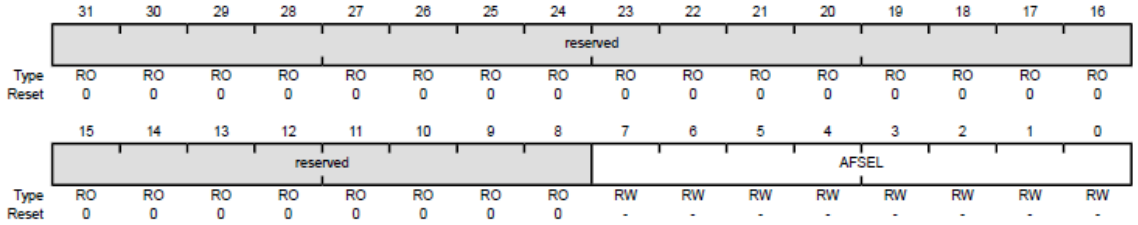
3.4 GPIO Event Register (GPIOIEV)



- Setting a bit in the GPIOIEV register configures the corresponding pin to detect rising edges or high levels, depending on the corresponding bit value in the GPIO Interrupt Sense (GPIOIS) register.
- Clearing a bit configures the pin to detect falling edges or low levels, depending on the corresponding bit value in the GPIOIS register. All bits are cleared by a reset.
- IEV bits:
    - bit = '1': detect rising edge or high level at interrupt pin.
    - bit = '0': detect falling edge or low level at interrupt pin.

3.5. GPIO Alternate Function Select Register (GPIOAFSEL)

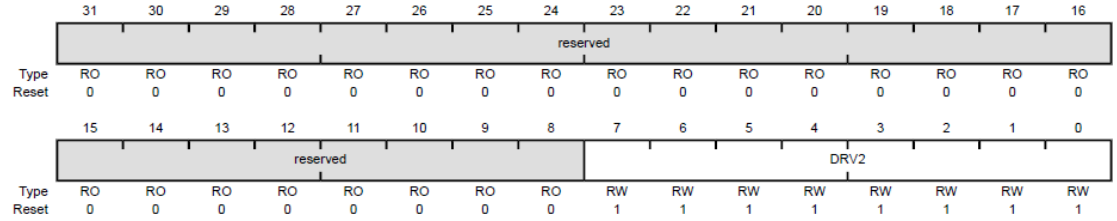
- **GPIOAFSEL** register is the mode control select register.
- If a bit is clear, the pin is used as a GPIO and is controlled by the GPIO registers.
- Setting a bit in this register configures the corresponding GPIO line to be controlled by an associated peripheral.
-



- **AFSEL[7:0]**: '0' = GPIO pin; '1' = Peripheral.
- Reset value= 0x0000.0000 (GPIO function for all 8 pins).

3.6. GPIO 2mA Drive Register (GPIODR2R)

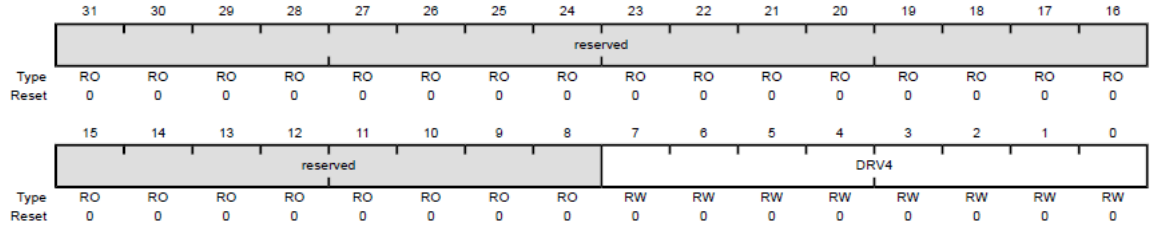
- **GPIODR2R** register is the 2-mA drive control register.



- **DRV2:** '0' = Drive is controlled by GPIODR4R or GPIODR8R registers; '1' = GPIO pin drive is 2mA.

3.7 GPIO 4-mA Drive Select (GPIODR4R)

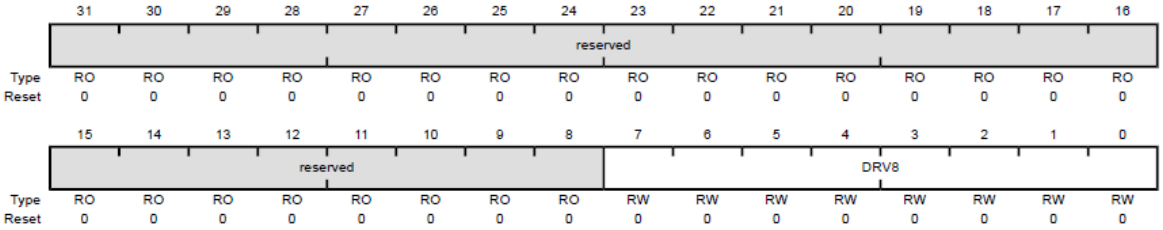
- **GPIODR4R** register is the 4-mA drive control register.



- **DRV4:** '0' = Drive is controlled by GPIODR2R or GPIODR8R registers; '1' = GPIO pin drive is 4mA.

3.8 GPIO 8-mA Drive Select (GPIODR8R)

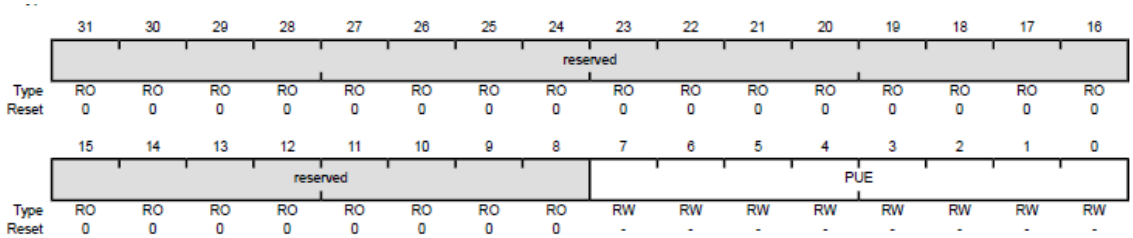
- **GPIODR8R** register is the 8-mA drive control register.



- **DRV8:** '0' = Drive is controlled by GPIODR2R or GPIODR4R registers; '1' = GPIO pin drive is 8mA.

3.9. GIPO Pull Up Register (GPIOPUR)

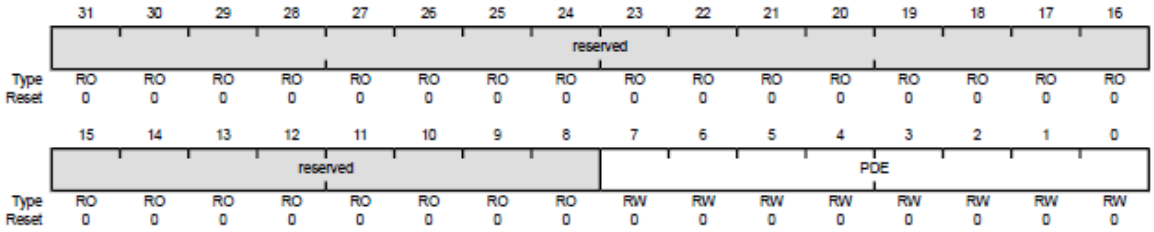
- **GPIOPUR** register is the pull-up control register.
- When a bit is set, a weak pull-up resistor on the corresponding GPIO signal is enabled.



- **PUE:** '1' enables pull-up resistor on the corresponding GPIO pin. '0' disables pull-up.
- Setting a bit in the GPIOPUR automatically clears the corresponding bit in the GPIOPDR.

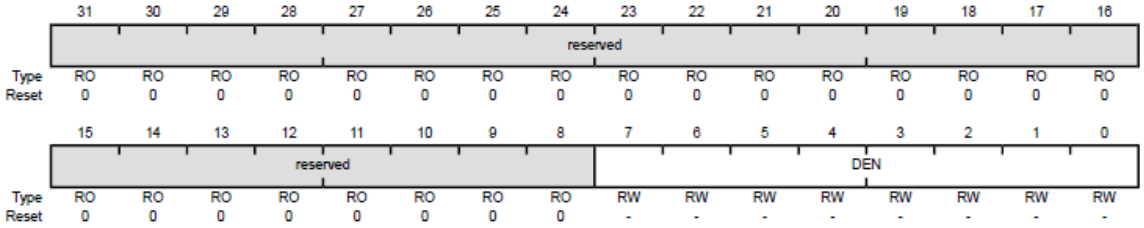
3.10. GIPO Pull Down Register (GPIOPDR)

- **GPIOPDR** register is the pull-down control register. When a bit is set, a weak pull-down resistor on the corresponding GPIO signal is enabled.
- Setting a bit in **GPIOPDR** automatically clears the corresponding bit in the **GPIO Pull-Up Select** register.



3.11. GPIO Digital Enable Register (GPIODEN)

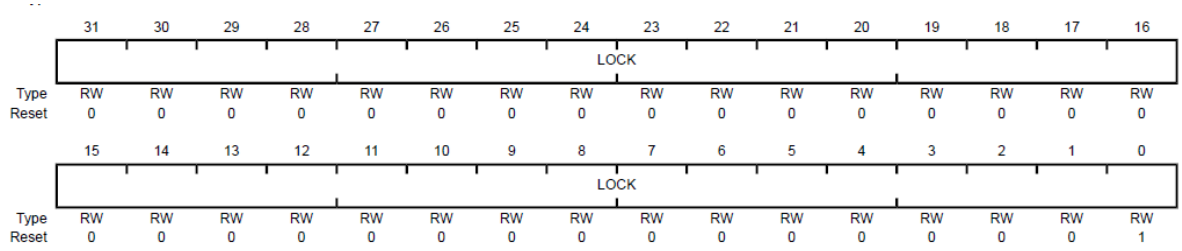
- **GPIODEN** register is the digital enable register.



- **DEN:** '1' implies pin set to GPIO function.

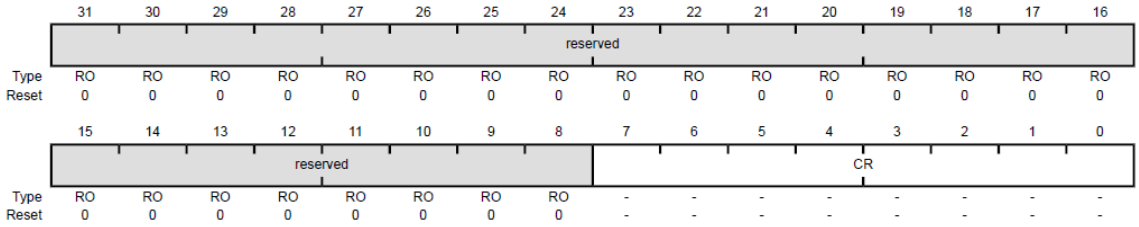
3.12. GPIO Lock Register (GPIOLOCK)

- **GPIOLOCK** register enables write access to the **GPIOCR** register.
- Writing 0x4C4F.434B to the **GPIOLOCK** register unlocks the **GPIOCR** register.
- Writing any other value to the **GPIOLOCK** register re-enables the locked state.
- Reading the **GPIOLOCK** register returns the lock status rather than the 32-bit value that was previously written.



3.13. GPIO Commit Register (GPIOCR)

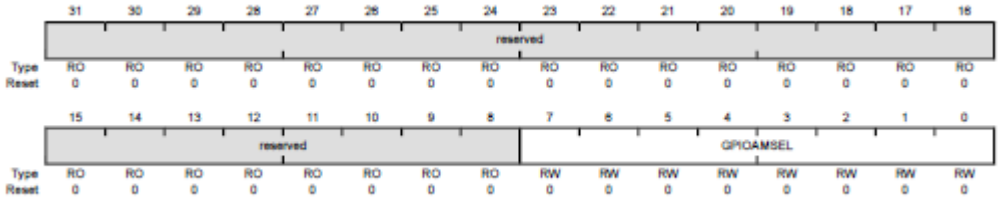
- **GPIOCR** register determines which bits of the **GPIOAFSEL**, **GPIOPUR**, **GPIOPDR**, and **GPIODEN** registers are committed when a write to these registers is performed. If a bit in the **GPIOCR** register is cleared, the data being written to the corresponding bit in the **GPIOAFSEL**, **GPIOPUR**, **GPIOPDR**, or **GPIODEN** registers cannot be committed and retains its previous value.
- If a bit in the **GPIOCR** register is set, the data being written to the corresponding bit of the **GPIOAFSEL**, **GPIOPUR**, **GPIOPDR**, or **GPIODEN** registers is committed to the register and reflects the new value.
- The contents of the **GPIOCR** register can only be modified if the status in the **GPIOLOCK** register is unlocked.



- If CR bit = 0: GPIO port bit cannot be written.
- If CR bit=1: GPIO port pin can be committed.

3.14. GPIO Analog Mode Select Register (GPIOAMSEL)

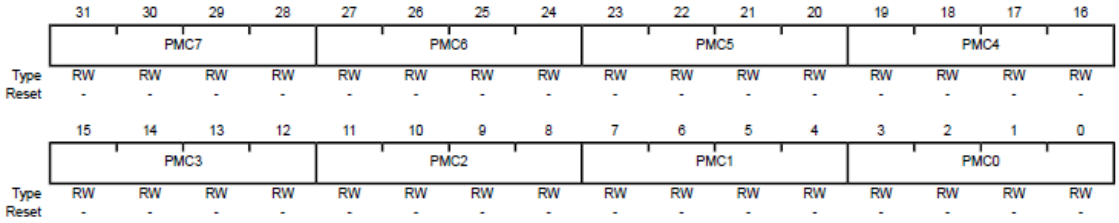
- **GPIOAMSEL** register controls isolation circuits to the analog side of a unified I/O pad.



- If GPIOAMSEL bit = 0: analog function of GPIO port bit is disabled.
- If GPIOAMSEL bit = 1: analog function of GPIO port bit is enabled.

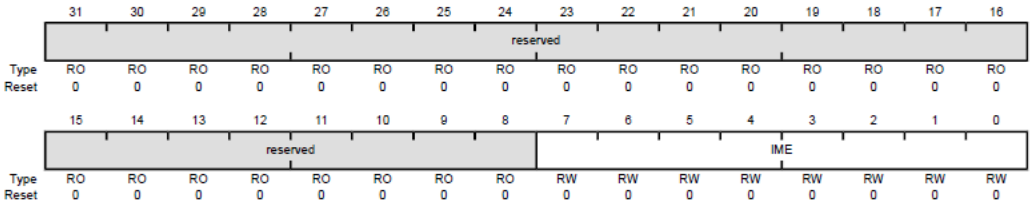
3.15. GPIO Port Control Register (GPIOPCTL)

- **GPIOPCTL** register is used in conjunction with the **GPIOAFSEL** register and selects the specific peripheral signal for each GPIO pin when using the alternate function mode.
- When a bit is set in the **GPIOAFSEL** register, the corresponding GPIO signal is controlled by an associated peripheral. The **GPIOPCTL** register selects one out of a set of peripheral functions for each GPIO.



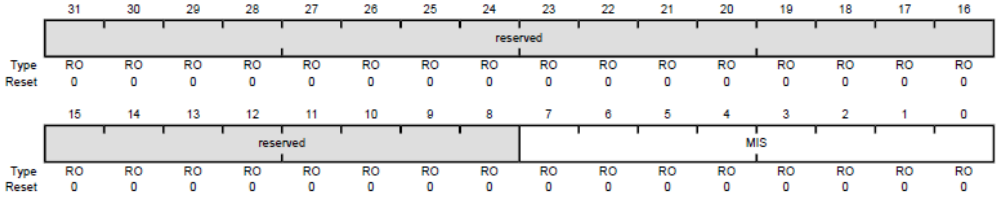
- PMC7 = Port Mux control for GPIO pin 7.
- PMC6 = Port Mux control for GPIO pin 6.
- PMC5 = Port Mux control for GPIO pin 5.
- PMC4 = Port Mux control for GPIO pin 4.
- PMC3 = Port Mux control for GPIO pin 3.
- PMC2 = Port Mux control for GPIO pin 2.
- PMC1 = Port Mux control for GPIO pin 1.
- PMC0 = Port Mux control for GPIO pin 0.

3.16. GPIO Interrupt Mask Register (GPIOIM)



- Setting a bit in the GPIOIM register allows interrupts that are generated by the corresponding pin to be sent to the NVIC.
- Clearing a bit prevents an interrupt on the corresponding pin from being sent to the NVIC.
- All bits are cleared by a reset.
- IME bits:
  - bit = '1': allows interrupt is sent to NVIC.
  - bit = '0': interrupt is masked (not sent to NVIC).

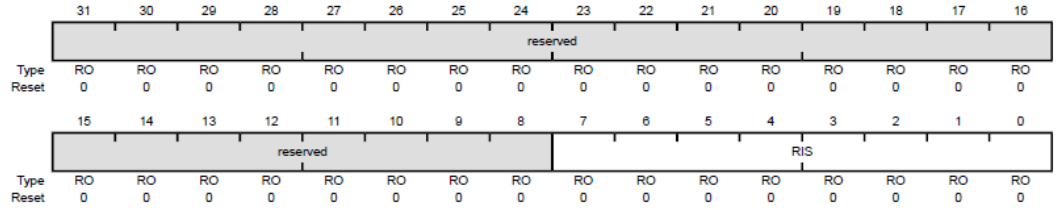
3.17. GPIO Masked Interrupt Status Register (GPIOPMIS)



- If a bit is set in this register, the corresponding interrupt has triggered an interrupt to the NVIC.

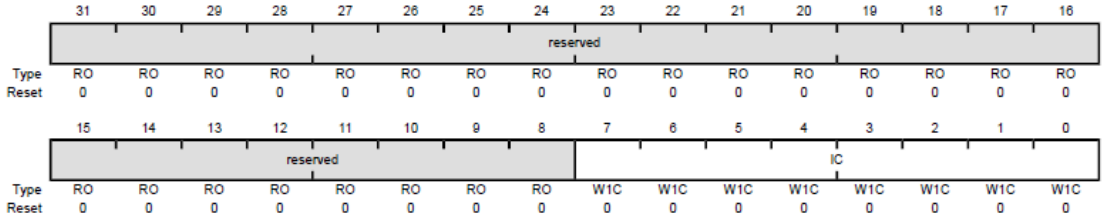
- If a bit is cleared, either no interrupt has been generated, or the interrupt is masked.
- MIS bits:
  - bit = '1': interrupt is triggered.
  - bit = '0': no interrupt triggered.

3.18. GPIO Raw Interrupt Status Register (GPIOPRIS)



- A bit in this register is set when an interrupt condition occurs on the corresponding GPIO pin.
- If the corresponding bit in the GPIO Interrupt Mask (GPIOIM) register is set, the interrupt is sent to the NVIC.
- Bits read as '0' indicate that corresponding input pins have not initiated an interrupt.
- RIS bits:
    - bit = '1' : interrupt has occurred for corresponding GPIO pin.
    - bit = '0' : interrupt has not occurred for corresponding GPIO pin.
- The corresponding GPIOMIS bit reflects the masked value of the RIS bit.

3.19. GPIO Interrupt Clear Register (GPIOPICR)



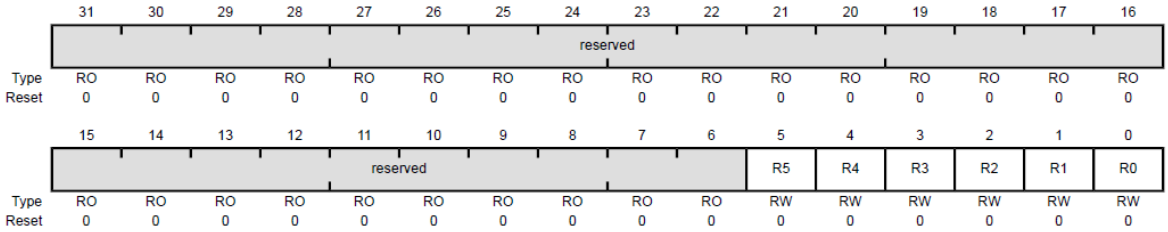
- For edge-detect interrupts, writing a 1 to the IC bit in the GPIOICR register clears the corresponding bit in the GPIORIS & GPIOMIS registers.
- If the interrupt is a level-detect, the IC bit in this register has no effect.
- Writing a 0 to any of the bits in the GPIOICR register has no effect.
- IC bits:
    - bit = '1': interrupt is cleared.
    - bit = '0': interrupt unaffected.

4. System Control Registers

4.1 GPIO Run Mode Clock Gating Register (RCGCGPIO)

- **RCGCGPIO** register provides software the capability to enable and disable GPIO modules in Run mode.
- When enabled, a module is provided a clock and accesses to module registers are allowed.
- When disabled, the clock is disabled to save power and accesses to module registers generate a bus fault.

General-Purpose Input/Output Run Mode Clock Gating Control (RCGCGPIO)
Base 0x400F.E000
Offset 0x608
Type RW, reset 0x0000.0000

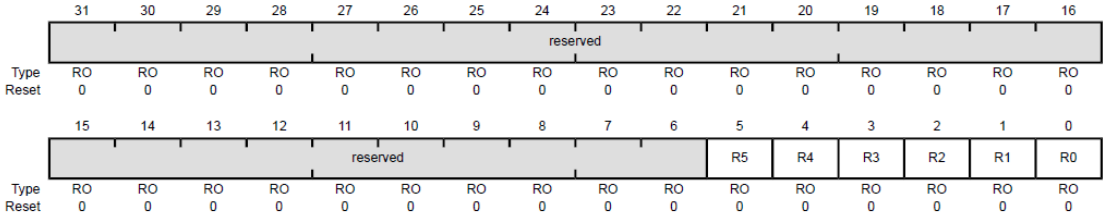


R0 bit denotes Port A.
R1 bit denotes Port B.
R2 bit denotes Port C.
R3 bit denotes Port D.
R4 bit denotes Port E.
R5 bit denotes Port F.

4.2 Peripheral Ready GPIO Register (PRGPIO)

- **PRGPIO** register indicates whether the GPIO modules are ready to be accessed.

General-Purpose Input/Output Peripheral Ready (PRGPIO)
Base 0x400F.E000
Offset 0xA08
Type RO, reset 0x0000.0000



R0 bit denotes Port A.
R1 bit denotes Port B.
R2 bit denotes Port C.
R3 bit denotes Port D.
R4 bit denotes Port E.
R5 bit denotes Port F.