General-Purpose Input/Output

This chapter describes the GPIO of the device.

Topic **Page**

25.1 Introduction

25.1.1 Purpose of the Peripheral

The general-purpose interface combines four general-purpose input/output (GPIO) modules. Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 128 (4 × 32) pins. These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events.

25.1.2 GPIO Features

Each GPIO module is made up of 32 identical channels. Each channel can be configured to be used in the following applications:

- Data input/output
- Keyboard interface with a de-bouncing cell
- Synchronous interrupt generation (in active mode) upon the detection of external events (signal transition(s) and/or signal level(s))
- Wake-up request generation (in Idle mode) upon the detection of signal transition(s)

Global features of the GPIO interface are:

- Synchronous interrupt requests from each channel are processed by two identical interrupt generation sub-modules to be used independently by the ARM Subsystem
- Wake-up requests from input channels are merged together to issue one wake-up signal to the system
- Shared registers can be accessed through “Set & Clear” protocol

25.1.3 Unsupported GPIO Features

The wake-up feature of the GPIO modules is only supported on GPIO0.

25.2 Integration

The device instantiates four GPIO_V2 modules. Each GPIO module provides the support for 32 dedicated pins with input and output configuration capabilities. Input signals can be used to generate interruptions and wake-up signal. Two Interrupt lines are available for bi-processor operation. Pins can be dedicated to be used as a keyboard controller.

With four GPIO modules, the device allows for a maximum of 128 GPIO pins. (The exact number available varies as a function of the device configuration and pin muxing.) GPIO0 is in the Wakeup domain and may be used to wakeup the device via external sources. GPIO[1:3] are located in the peripheral domain.

Figure 25-1 and Figure 25-2 show the GPIO integration.

Figure 25-1. GPIO0 Module Integration

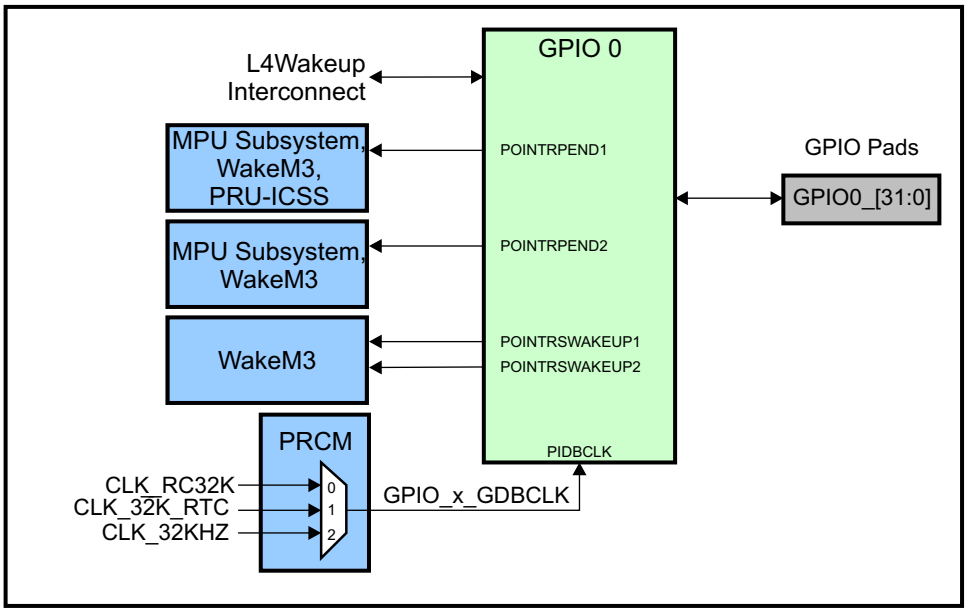
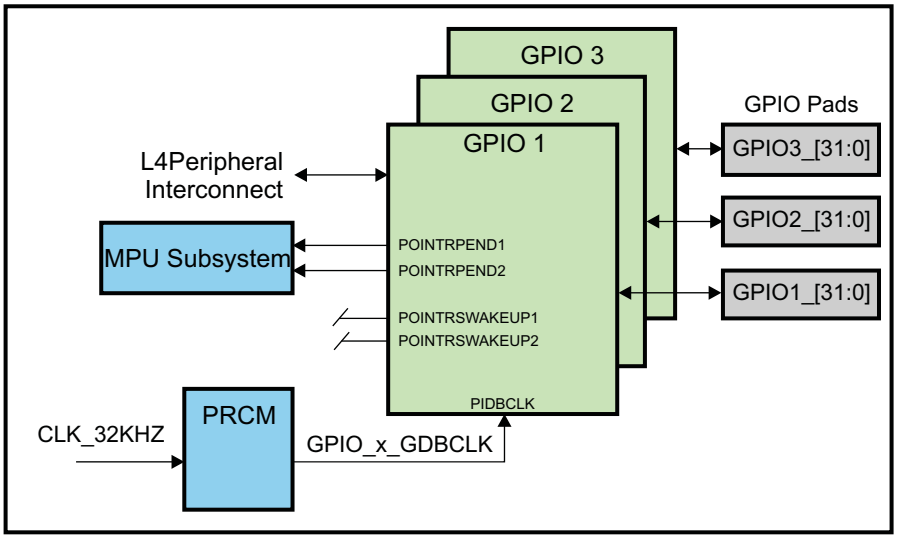


Figure 25-2. GPIO[1–3] Module Integration



25.2.1 GPIO Connectivity Attributes

The general connectivity attributes for the GPIO modules in the device are shown in Table 25-1 and Table 25-2.

Table 25-1. GPIO0 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) GPIO_0_GDBCLK (Debounce)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	Two Interrupts: INTRPEND1 (GPIOINT0A) to MPU subsystem, PRU-ICSS (POINTRPEND1), and WakeM3 INTRPEND2 (GPIOINT0B) to MPU subsystem and WakeM3
DMA Requests	Interrupt Requests are redirected as DMA requests: 1 DMA request (GPIOEVT0)
Physical Address	L4 Wakeup slave port

Table 25-2. GPIO[1:3] Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) GPIO_1_GDBCLK (GPIO1 Debounce) GPIO_2_GDBCLK (GPIO2 Debounce) GPIO_3_GDBCLK (GPIO3 Debounce)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	Two Interrupts: INTRPEND1 (GPIOINTxA) to MPU subsystem INTRPEND2 (GPIOINTxB) to MPU subsystem
DMA Requests	Interrupt Requests are redirected as DMA requests: 1 DMA request only for GPIO1 (GPIOEVT1) and GPIO2 (GPIOEVT2)
Physical Address	L4 Peripheral slave port

25.2.2 GPIO Clock and Reset Management

The GPIO modules require two clocks: The de-bounce clock is used for the de-bouncing cells. The interface clock provided by the peripheral bus (L4 interface) is also the functional clock and is used through the entire GPIO module (except within the de-bouncing sub-module). It clocks the OCP interface and the internal logic. For GPIO0 the debounce clock is selected from one of three sources using the CLKSEL_GPIO0_DBCLK register in the PRCM:

- The on-chip ~32.768 KHz oscillator (CLK_RC32K)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The external 32.768 KHz oscillator/clock (CLK_32K_RTC)

Table 25-3. GPIO Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
GPIO0			
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk From PRCM
Debounce Functional clock	32.768 KHz	CLK_RC32K CLK_32KHZ (PER_CLKOUTM2 / 5859.375) CLK_32K_RTC	pd_wkup_gpio0_gdbclk From PRCM
GPIO[1:3]			

Table 25-3. GPIO Clock Signals (continued)

Clock Signal	Max Freq	Reference / Source	Comments
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
Debounce Functional clock (GPIO1)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_1_gdbclk From PRCM
Debounce Functional clock (GPIO2)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_2_gdbclk From PRCM
Debounce Functional clock (GPIO3)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_3_gdbclk From PRCM

25.2.3 GPIO Pin List

Each GPIO module includes 32 interface I/Os. These signals are designated as shown in Table 25-4. Note that for this device, most of these signals will be multiplexed with functional signals from other interfaces.

Table 25-4. GPIO Pin List

Pin	Type	Description
GPIO0_[31:0]	I/O	General Purpose Input-Output pins
GPIO1_[31:0]		
GPIO2_[31:0]		
GPIO3_[31:0]		
GPIO4_[31:0]		
GPIO5_[31:0]		

25.3 Functional Description

This section discusses the operational details and basic functions of the GPIO peripheral.

25.3.1 Operating Modes

Four operating modes are defined for the module:

- **Active mode:** the module is running synchronously on the interface clock, interrupt can be generated according to the configuration and the external signals.
- **Idle mode:** the module is in a waiting state, interface clock can be stopped , no interrupt can be generated, a wake-up signal can be generated according to the configuration and external signals. Check the chip top-level functional specification for the availability of the debouncing clock while in Idle mode. If the debouncing clock is active, the debouncing cell can be used to sample and to filter the input to generate a wakeup event. Otherwise (debouncing clock inactive), the debouncing cell cannot be used, as it would gate all input signals.
- **Inactive mode:** the module has no activity, interface clock can be stopped, no interrupt can be generated, and the wake-up feature is inhibited.
- **Disabled mode:** the module is not used, internal clock paths are gated, no interrupt or wake-up request can be generated.

The Idle and Inactive modes are configured within the module and activated on request by the host processor through system interface sideband signals. The Disabled mode is set by software through a dedicated configuration bit. It unconditionally gates the internal clock paths not use for the system interface. All module registers are 8, 16 or 32-bit accessible through the OCP compatible interface (little endian encoding). In active mode, the event detection (level or transition) is performed in the GPIO module using the interface clock. The detection's precision is set by the frequency of this clock and the selected internal gating scheme.

25.3.2 Clocking and Reset Strategy

25.3.2.1 Clocks

GPIO module runs using two clocks:

- The debouncing clock is used for the debouncing sub-module logic (without the corresponding configuration registers). This module can sample the input line and filters the input level using a programmed delay.
- The interface clock provided by the peripheral bus (OCP compatible system interface). It is used through the entire GPIO module (except within the debouncing sub-module logic). It clocks the OCP interface and the internal logic. Clock gating features allow adapting the module power consumption to the activity.

25.3.2.2 Clocks, Gating and Active Edge Definitions

The interface clock provided by the peripheral bus (OCP compatible system interface) is used through the entire GPIO module. Two clock domains are defined: the OCP interface and the internal logic. Each clock domain can be controlled independently. Sampling operations for the data capture and for the events detection are done using the rising edge. The data loaded in the data output register (GPIO_DATAOUT) is set at the output GPIO pins synchronously with the rising edge of the interface clock.

Five clock gating features are available:

- Clock for the system interface logic can be gated when the module is not accessed, if the AUTOIDLE configuration bit in the system configuration register (GPIO_SYSCONFIG) is set. Otherwise, this logic is free running on the interface clock.
- Clock for the input data sample logic can be gated when the data in register is not accessed.
- Four clock groups are used for the logic in the synchronous events detection. Each 8 input GPIO_V2 pins group will have a separate enable signal depending on the edge/level detection register setting. If a group requires no detection, then the corresponding clock will be gated. All channels are also gated using a 'one out of N' scheme. N can take the values 1, 2, 4 or 8. The interface clock is enabled for

this logic one cycle every N cycles. When N is equal to 1, there is no gating and this logic is free running on the interface clock. When N is between 2 to 8, this logic is running at the equivalent frequency of interface clock frequency divided by N.

- In Inactive mode, all internal clock paths are gated.
- In Disabled mode, all internal clock paths not used for the system interface are gated. All GPIO registers are accessible synchronously with the interface clock.

25.3.2.3 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode according to the IDLEMODE field in the system configuration register (GPIO_SYSCONFIG).

- IDLEMODE = 0 (Force-Idle mode): the GPIO goes in Inactive mode independently of the internal module state and the Idle acknowledge is unconditionally sent. In Force-Idle mode, the module is in Inactive mode and its wake-up feature is totally inhibited.
- IDLEMODE = 1h (No-Idle mode): the GPIO does not go to the Idle mode and the Idle acknowledge is never sent.
- IDLEMODE = 2h (Smart-Idle mode) or IDLEMODE = 3h (Smart-Idle mode): the GPIO module evaluates its internal capability to have the interface clock switched off. Once there is no more internal activity (the data input register completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to GPIO_DEBOUNCINGTIME register pending to be synchronized), the Idle acknowledge is asserted and the GPIO enters Idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the ENAWAKEUP bit in GPIO_SYSCONFIG is set to enable the GPIO wakeup capability. When the system is awake, the Idle Request goes inactive, the Idle acknowledge and wake-up request (if it is the GPIO that triggered the system's wakeup) signals are immediately de-asserted, and the asynchronous wake-up request (if existing) is reflected into the synchronous interrupt status registers.

NOTE: Idle mode request and Idle acknowledge are system interface sideband signals. Once the GPIO acknowledges the Sleep mode request (Idle acknowledge has been sent), the interface clock can be stopped anytime.

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode only if there is no active bit in GPIO_IRQSTATUS_RAW_n registers.

25.3.2.4 Reset

The OCP hardware Reset signal has a global reset action on the GPIO. All configuration registers, all DFFs clocked with the Interface clock or Debouncing clock and all internal state machines are reset when the OCP hardware Reset is active (low level). The RESETDONE bit in the system status register (GPIO_SYSSTATUS) monitors the internal reset status: it is set when the Reset is completed on both OCP and Debouncing clock domains. The software Reset (SOFTRESET bit in the system configuration register) has the same effect as the OCP hardware Reset signal, and the RESETDONE bit in GPIO_SYSSTATUS is updated in the same condition.

25.3.3 Interrupt and Wake-up Features

25.3.3.1 Functional Description

In order to generate an interrupt request to a host processor upon a defined event (level or logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- Interrupts for the GPIO channel must be enabled in the GPIO_IRQSTATUS_SET_0 and/or GPIO_IRQSTATUS_SET_1 registers.
- The expected event(s) on input GPIO to trigger the interrupt request has to be selected in the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers.

In order to generate a wake-up request to a host processor upon a defined event (logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- The GPIO channel must be enabled in the GPIO_IRQWAKEN register.
- The expected event(s) on input GPIO to trigger the interrupt (or the wake-up) request has to be selected in the GPIO_RISINGDETECT and GPIO_FALLINGDETECT registers. Wake-up request can only be generated on rising and/or on falling transitions.

For instance, interrupt generation on both edges on input k is configured by setting to 1 the *k*th bit in registers GPIO_RISINGDETECT and GPIO_FALLINGDETECT along with the interrupt enabling for one or both interrupt lines (GPIO_IRQSTATUS_SET_n).

NOTE: All interrupt (or wake-up) sources (the 32 input GPIO channels) are merged together to issue two synchronous interrupt requests 1 and 2, and two asynchronous wake-up requests.

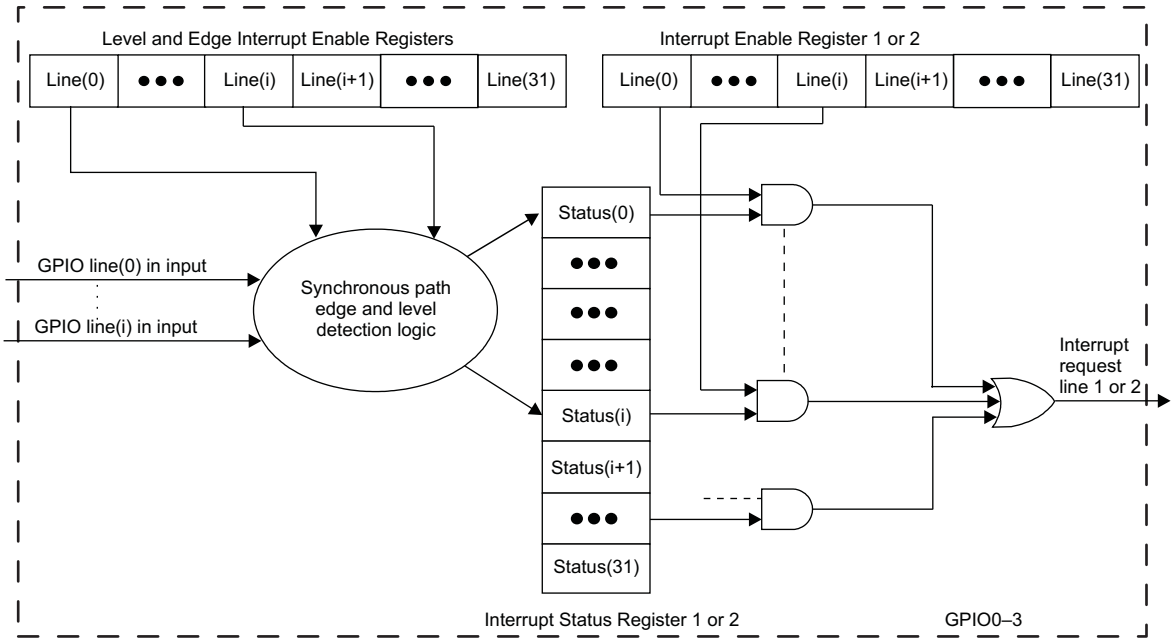
25.3.3.2 Synchronous Path: Interrupt Request Generation

In Active mode, once the GPIO configuration registers have been set to enable the interrupt generation, a synchronous path (Figure 25-3) samples the transitions and levels on the input GPIO with the internally gated interface clock. When an event matches the programmed settings, the corresponding bit in the GPIO_IRQSTATUS_RAW_n registers is set to 1 and, on the following interface clock cycle, the interrupt lines 1 and/or 2 are activated (depending on the GPIO_IRQSTATUS_SET_n registers).

Due to the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (the internally gated interface clock period is equal to N times the interface clock period). This minimum pulse width has to be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

As the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This should not exceed 3 internally gated interface clock cycles + 2 interface clock cycles when the debouncing feature is not used. When the debouncing feature is active, the latency depends on the GPIO_DEBOUNCINGTIME register value and should be less than 3 internally gated interface clock cycles + 2 interface clock cycles + GPIO_DEBOUNCINGTIME value debouncing clock cycles + 3 debouncing clock cycles.

Figure 25-3. Interrupt Request Generation



25.3.3.3 Asynchronous Path: Wake-up Request Generation

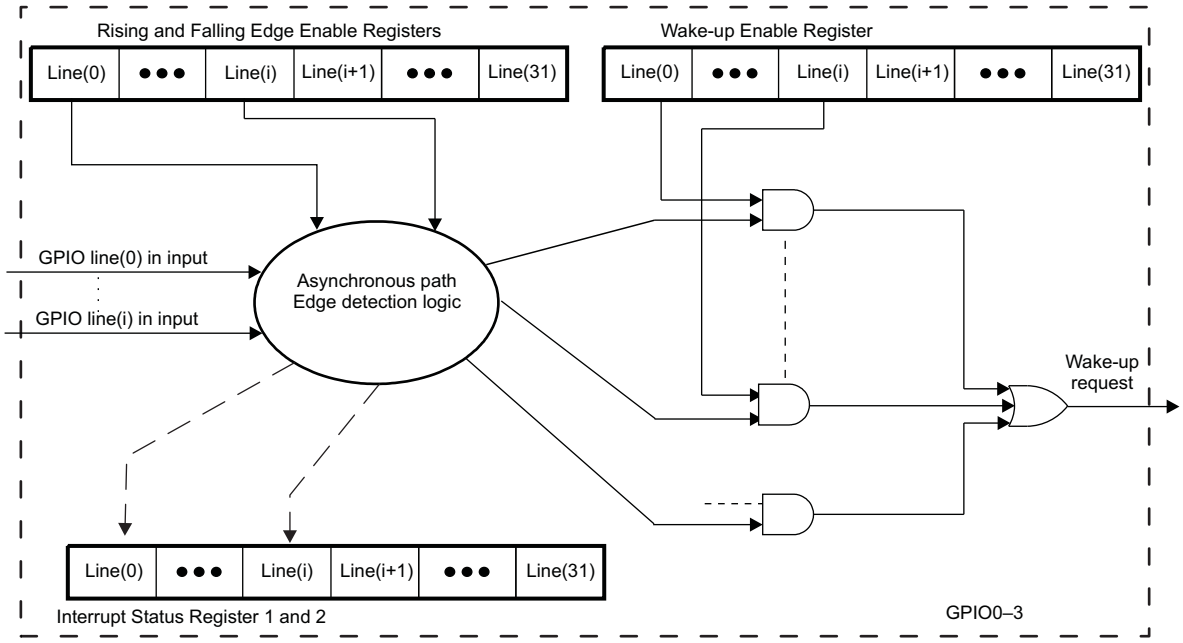
In Idle mode (interface clock is shut down, the GPIO configuration registers have been previously programmed), an asynchronous path (Figure 25-4) detects the expected transition(s) on input GPIO (according to the registers programming) and activates an asynchronous wake-up request if the wakeup enable register is set. As shown in , there is only one external wake-up line, since all wake-up sources are merged together. Once the system is in wake up, the interface clock is re-started and, according to the input GPIO pin that triggered the wake-up request, the corresponding bits in the GPIO_IRQSTATUS_RAW_n registers are synchronously set to 1; on the following internal clock cycle, the interrupt lines 1 and/or 2 are active (active high) when the corresponding bits in GPIO_IRQSTATUS_SET_n registers are set.

NOTE: If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request since there is no sampling operation.

If debouncing is used, the minimum pulse width is set by the debouncing specified time.

The ENAWAKEUP bit in the system configuration register (GPIO_SYSCONFIG) enables or disables the GPIO wake-up feature globally: if this bit is 0, the GPIO_IRQWAKEN has no effect.

Figure 25-4. Wake-Up Request Generation



25.3.3.4 Interrupt (or Wake-up) Line Release

When the host processor receives an interrupt request issued by the GPIO module, it can read the corresponding GPIO_IRQSTATUS_n register to find out which input GPIO has triggered the interrupt (or the wake-up request). After servicing the interrupt (or acknowledging the wake-up request), the processor resets the status bit and releases the interrupt line by writing a 1 in the corresponding bit of the GPIO_IRQSTATUS_n register. If there is still a pending interrupt request to serve (all bits in the GPIO_IRQSTATUS_RAW_n register not masked by the GPIO_IRQSTATUS_SET_n, which are not cleared by setting the GPIO_IRQSTATUS_CLR_n), the interrupt line will be re-asserted.

25.3.4 General-Purpose Interface Basic Programming Model

25.3.4.1 Power Saving by Grouping the Edge/Level Detection

Each GPIO module implements four gated clocks used by the edge/level detection logic to save power. Each group of eight input GPIO pins generates a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no edge/level detection, then the corresponding clock is gated (cut off). Grouping the edge/level enable can save the power consumption of the module as described in the following example.

If any of the registers:

- GPIO_LEVELDETECT0
- GPIO_LEVELDETECT1
- GPIO_RISINGDETECT
- GPIO_FALLINGDETECT

are set to 0101 0101h, then all clocks are active (power consumption is high);

are set to 0000 00FFh, then a single clock is active.

NOTE: When the clocks are enabled by writing to the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers, the detection starts after 5 clock cycles. This period is required to clean the synchronization edge/level detection pipeline.

The mechanism is independent of each clock group. If the clock has been started before a new setting is performed, the following is recommended: first, set the new detection required; second, disable the previous setting (if necessary). In this way, the corresponding clock is not gated and the detection starts immediately.

25.3.4.2 Set and Clear Instructions

The GPIO module implements the set-and-clear protocol register update for the data output and interrupt enable, and wake-up enable registers. This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s).

Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear (recommended): Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing a 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.

25.3.4.2.1 Clear Instruction

25.3.4.2.1.1 Clear Interrupt Enable Registers (GPIO_IRQSTATUS_CLR_0 and GPIO_IRQSTATUS_CLR_1):

- A write operation in the clear interrupt enable1 (or enable2) register clears the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

25.3.4.2.1.2 Clear Wake-up Enable Register (GPIO_CLEARWKUENA):

- A write operation in the clear wake-up enable register clears the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear wake-up enable register returns the value of the wake-up enable register.

25.3.4.2.1.3 Clear Data Output Register (GPIO_CLEARDATAOUT):

- A write operation in the clear data output register clears the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear data output register returns the value of the data output register.

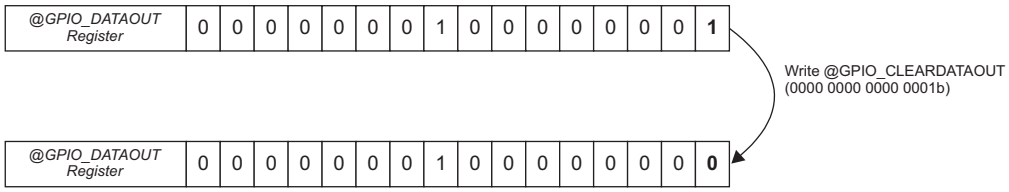
25.3.4.2.1.4 Clear Instruction Example

Assume the data output register (or one of the interrupt/wake-up enable registers) contains the binary value, 0000 0001 0000 0001h, and you want to clear bit 0.

With the clear instruction feature, write 0000 0000 0000 0001h at the address of the clear data output register (or at the address of the clear interrupt/wake-up enable register). After this write operation, a reading of the data output register (or the interrupt/wake-up enable register) returns 0000 0001 0000 0000h; bit 0 is cleared.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 25-5. Write @ GPIO_CLEARDATAOUT Register Example



25.3.4.2.2 Set Instruction

25.3.4.2.2.1 Set Interrupt Enable Registers (GPIO_IRQSTATUS_SET_0 and GPIO_IRQSTATUS_SET_1):

- A write operation in the set interrupt enable1 (or enable2) register sets the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

25.3.4.2.2.2 Set Wake-up Enable Register (GPIO_SETWKUENA):

- A write operation in the set wake-up enable register sets the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set wake-up enable register returns the value of the wake-up enable register.

25.3.4.2.2.3 Set Data Output Register (GPIO_SETDATAOUT):

- A write operation in the set data output register sets the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set data output register returns the value of the data output register.

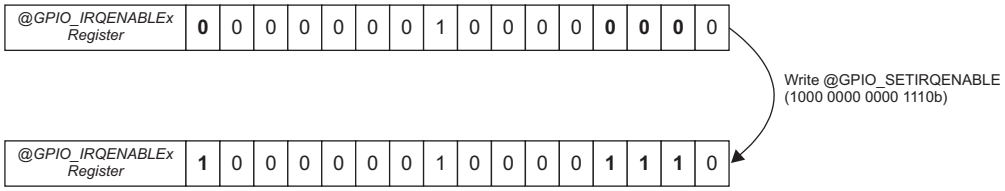
25.3.4.2.2.4 Set Instruction Example

Assume the interrupt enable1 (or enable2) register (or the data output register) contains the binary value, 0000 0001 0000 0000h, and you want to set bits 15, 3, 2, and 1.

With the set instruction feature, write 1000 0000 0000 1110h at the address of the set interrupt enable1 (or enable2) register (or at the address of the set data output register). After this write operation, a reading of the interrupt enable1 (or enable2) register (or the data output register) returns 1000 0001 0000 1110h; bits 15, 3, 2, and 1 are set.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 25-6. Write @ GPIO_SETIRQENABLEx Register Example



25.3.4.3 Data Input (Capture)/Output (Drive)

The output enable register (GPIO_OE) controls the output/input capability for each pin. At reset, all the GPIO-related pins are configured as input and output capabilities are disabled. This register is not used within the module; its only function is to carry the pads configuration.

When configured as an output (the desired bit reset in GPIO_OE), the value of the corresponding bit in the GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set and clear protocol register update feature. This feature lets you set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address. If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable and the interrupt enable registers.

When configured as an input (the desired bit set to 1 in GPIO_OE), the state of the input can be read from the corresponding bit in the GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and to write data). If the application uses a pin as an input, the application must properly configure the wake-up enable and the interrupt enable registers to the interrupt and wake-up feature as needed.

25.3.4.4 Debouncing Time

To enable the debounce feature for a pin, the GPIO configuration registers must be programmed as follows:

- The GPIO pin must be configured as input in the output enable register (write 1 to the corresponding bit of the GPIO_OE register).
- The debouncing time must be set in the debouncing value register (GPIO_DEBOUNCINGTIME). The GPIO_DEBOUNCINGTIME register is used to set the debouncing time for all input lines in the GPIO module. The value is global for all the ports of one GPIO module, so up to six different debouncing values are possible. The debounce cell is running with the debounce clock (32 kHz). This register represents the number of the clock cycle(s) (one cycle is 31 microseconds long) to be used.

  The following formula describes the required input stable time to be propagated to the debounced output:

Debouncing time = (DEBOUNCETIME + 1) × 31 µs

Where the DEBOUNCETIME field value in the GPIO_DEBOUNCINGTIME register is from 0 to 255.

- The debouncing feature must be enabled in the debouncing enable register (write 1 to the corresponding DEBOUNCEENABLE bit in the GPIO_DEBOUNCENABLE register).

25.3.4.5 GPIO as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface (Figure 25-7). You can dedicate channels based on the keyboard matrix = * c). Figure 25-7 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pull-up. Column channels are configured as outputs and drive a low level.

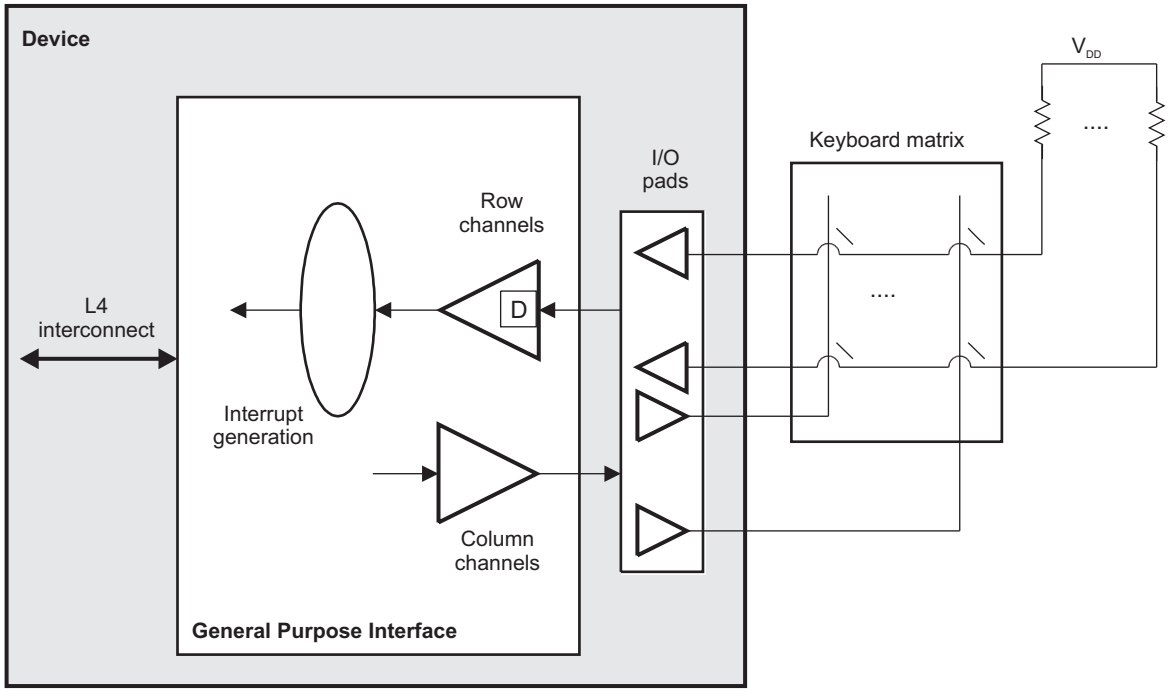
When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 25.3.3).

When the keyboard interrupt is received, the processor can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

Figure 25-7. General-Purpose Interface Used as a Keyboard Interface



25.4 GPIO Registers

25.4.1 GPIO Registers

Table 25-5 lists the memory-mapped registers for the GPIO. All register offset addresses not listed in Table 25-5 should be considered as reserved locations and the register contents should not be modified.

Table 25-5. GPIO Registers

Offset	Acronym	Register Name	Section
0h	GPIO_REVISION		Section 25.4.1.1
10h	GPIO_SYSCONFIG		Section 25.4.1.2
20h	GPIO_EOI		Section 25.4.1.3
24h	GPIO_IRQSTATUS_RAW_0		Section 25.4.1.4
28h	GPIO_IRQSTATUS_RAW_1		Section 25.4.1.5
2Ch	GPIO_IRQSTATUS_0		Section 25.4.1.6
30h	GPIO_IRQSTATUS_1		Section 25.4.1.7
34h	GPIO_IRQSTATUS_SET_0		Section 25.4.1.8
38h	GPIO_IRQSTATUS_SET_1		Section 25.4.1.9
3Ch	GPIO_IRQSTATUS_CLR_0		Section 25.4.1.10
40h	GPIO_IRQSTATUS_CLR_1		Section 25.4.1.11
44h	GPIO_IRQWAKEN_0		Section 25.4.1.12
48h	GPIO_IRQWAKEN_1		Section 25.4.1.13
114h	GPIO_SYSSTATUS		Section 25.4.1.14
130h	GPIO_CTRL		Section 25.4.1.15
134h	GPIO_OE		Section 25.4.1.16
138h	GPIO_DATAIN		Section 25.4.1.17
13Ch	GPIO_DATAOUT		Section 25.4.1.18
140h	GPIO_LEVELDETECT0		Section 25.4.1.19
144h	GPIO_LEVELDETECT1		Section 25.4.1.20
148h	GPIO_RISINGDETECT		Section 25.4.1.21
14Ch	GPIO_FALLINGDETECT		Section 25.4.1.22
150h	GPIO_DEBOUNCENABLE		Section 25.4.1.23
154h	GPIO_DEBOUNCINGTIME		Section 25.4.1.24
190h	GPIO_CLEARDATAOUT		Section 25.4.1.25
194h	GPIO_SETDATAOUT		Section 25.4.1.26

25.4.1.1 GPIO_REVISION Register (offset = 0h) [reset = 50600801h]

GPIO_REVISION is shown in Figure 25-8 and described in Table 25-6.

The GPIO revision register is a read only register containing the revision number of the GPIO module. A write to this register has no effect, the same as the reset.

Figure 25-8. GPIO_REVISION Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-1h		R-60h			
23	22	21	20	19	18	17	16
FUNC							
R-60h							
15	14	13	12	11	10	9	8
RTL					MAJOR		
R-1h					R-0h		
7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-1h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-6. GPIO_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between old Scheme and current.
29-28	RESERVED	R	1h	
27-16	FUNC	R	60h	Indicates a software compatible module family.
15-11	RTL	R	1h	RTL version
10-8	MAJOR	R	0h	Major Revision
7-6	CUSTOM	R	0h	Indicates a special version for a particular device.
5-0	MINOR	R	1h	Minor Revision

25.4.1.2 GPIO_SYSCONFIG Register (offset = 10h) [reset = 0h]

GPIO_SYSCONFIG is shown in Figure 25-9 and described in Table 25-7.

The GPIO_SYSCONFIG register controls the various parameters of the L4 interconnect. When the AUTOIDLE bit is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 25-9. GPIO_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-7. GPIO_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-3	IDLEMODE	R/W	0h	Power Management, Req/Ack control. 0h = Force-idle. An idle request is acknowledged unconditionally 1h = No-idle. An idle request is never acknowledged 2h = Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module 3h = Smart Idle Wakeup (GPIO0 only)
2	ENAWAKEUP	R/W	0h	0h = Wakeup generation is disabled. 1h = Wakeup capability is enabled upon expected transition on input GPIO pin.
1	SOFTRESET	R/W	0h	Software reset. This bit is automatically reset by the hardware. During reads, it always returns 0. 0h = Normal mode 1h = The module is reset
0	AUTOIDLE	R/W	0h	Internal interface clock gating strategy 0h = Internal Interface OCP clock is free-running 1h = Automatic internal OCP clock gating, based on the OCP interface activity

25.4.1.3 GPIO_EOI Register (offset = 20h) [reset = 0h]

GPIO_EOI is shown in Figure 25-10 and described in Table 25-8.

This module supports DMA events with its interrupt signal. This register must be written to after the DMA completes in order for subsequent DMA events to be triggered from this module.

Figure 25-10. GPIO_EOI Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DMAEvent_Ack
R-0h							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-8. GPIO_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DMAEvent_Ack	R/W	0h	Write 0 to acknowledge DMA event has been completed. Module will be able to generate another DMA event only when the previous one has been acknowledged using this register. Reads always returns 0.

25.4.1.4 GPIO_IRQSTATUS_RAW_0 Register (offset = 24h) [reset = 0h]

GPIO_IRQSTATUS_RAW_0 is shown in Figure 25-11 and described in Table 25-9.

The GPIO_IRQSTATUS_RAW_0 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 25-11. GPIO_IRQSTATUS_RAW_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-9. GPIO_IRQSTATUS_RAW_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n status. 0h = No effect. 1h = IRQ is triggered.

25.4.1.5 GPIO_IRQSTATUS_RAW_1 Register (offset = 28h) [reset = 0h]

GPIO_IRQSTATUS_RAW_1 is shown in Figure 25-12 and described in Table 25-10.

The GPIO_IRQSTATUS_RAW_1 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 25-12. GPIO_IRQSTATUS_RAW_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-10. GPIO_IRQSTATUS_RAW_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n status. 0h = No effect. 1h = IRQ is triggered.

25.4.1.6 GPIO_IRQSTATUS_0 Register (offset = 2Ch) [reset = 0h]

GPIO_IRQSTATUS_0 is shown in Figure 25-13 and described in Table 25-11.

The GPIO_IRQSTATUS_0 register provides core status information for the interrupt handling, showing all active events which have been enabled. The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 25-13. GPIO_IRQSTATUS_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W1C-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-11. GPIO_IRQSTATUS_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W1C	0h	Interrupt n status. 0h (W) = No effect. 0h (R) = IRQ is not triggered. 1h (W) = Clears the IRQ. 1h (R) = IRQ is triggered.

25.4.1.7 GPIO_IRQSTATUS_1 Register (offset = 30h) [reset = 0h]

GPIO_IRQSTATUS_1 is shown in Figure 25-14 and described in Table 25-12.

The GPIO_IRQSTATUS_1 register provides core status information for the interrupt handling, showing all active events which have been enabled. The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

Figure 25-14. GPIO_IRQSTATUS_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W1C-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-12. GPIO_IRQSTATUS_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W1C	0h	Interrupt n status. 0h (W) = No effect. 0h (R) = IRQ is not triggered. 1h (W) = Clears the IRQ. 1h (R) = IRQ is triggered.

25.4.1.8 GPIO_IRQSTATUS_SET_0 Register (offset = 34h) [reset = 0h]

GPIO_IRQSTATUS_SET_0 is shown in Figure 25-15 and described in Table 25-13.

All 1-bit fields in the GPIO_IRQSTATUS_SET_0 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 25-15. GPIO_IRQSTATUS_SET_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-13. GPIO_IRQSTATUS_SET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n enable 0h = No effect. 1h = Enable IRQ generation.

25.4.1.9 GPIO_IRQSTATUS_SET_1 Register (offset = 38h) [reset = 0h]

GPIO_IRQSTATUS_SET_1 is shown in Figure 25-16 and described in Table 25-14.

All 1-bit fields in the GPIO_IRQSTATUS_SET_1 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 25-16. GPIO_IRQSTATUS_SET_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-14. GPIO_IRQSTATUS_SET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n enable 0h = No effect. 1h = Enable IRQ generation.

25.4.1.10 GPIO_IRQSTATUS_CLR_0 Register (offset = 3Ch) [reset = 0h]

GPIO_IRQSTATUS_CLR_0 is shown in Figure 25-17 and described in Table 25-15.

All 1-bit fields in the GPIO_IRQSTATUS_CLR_0 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 25-17. GPIO_IRQSTATUS_CLR_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-15. GPIO_IRQSTATUS_CLR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n enable 0h = No effect. 1h = Disable IRQ generation.

25.4.1.11 GPIO_IRQSTATUS_CLR_1 Register (offset = 40h) [reset = 0h]

GPIO_IRQSTATUS_CLR_1 is shown in Figure 25-18 and described in Table 25-16.

All 1-bit fields in the GPIO_IRQSTATUS_CLR_1 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

Figure 25-18. GPIO_IRQSTATUS_CLR_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-16. GPIO_IRQSTATUS_CLR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Interrupt n enable 0h = No effect. 1h = Disable IRQ generation.

25.4.1.12 GPIO_IRQWAKEN_0 Register (offset = 44h) [reset = 0h]

GPIO_IRQWAKEN_0 is shown in Figure 25-19 and described in Table 25-17.

Per-event wakeup enable vector (corresponding to first line of interrupt). Every 1-bit field in the GPIO_IRQWAKEN_0 register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). This register allows the user to mask the expected transition on input GPIO from generating a wakeup request. The GPIO_IRQWAKEN_0 is programmed synchronously with the interface clock before any Idle mode request coming from the host processor. Note: In Force-Idle mode, the module wake-up feature is totally inhibited. The wake-up generation can also be gated at module level using the EnaWakeUp bit from GPIO_SYSCONFIG register.

Figure 25-19. GPIO_IRQWAKEN_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-17. GPIO_IRQWAKEN_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE	R/W	0h	Wakeup Enable for Interrupt Line 0h = Disable wakeup generation. 1h = Enable wakeup generation.

25.4.1.13 GPIO_IRQWAKEN_1 Register (offset = 48h) [reset = 0h]

GPIO_IRQWAKEN_1 is shown in Figure 25-20 and described in Table 25-18.

Per-event wakeup enable vector (corresponding to second line of interrupt). Every 1-bit field in the GPIO_IRQWAKEN_1 register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). This register allows the user to mask the expected transition on input GPIO from generating a wakeup request. The GPIO_IRQWAKEN_1 is programmed synchronously with the interface clock before any Idle mode request coming from the host processor. Note: In Force-Idle mode, the module wake-up feature is totally inhibited. The wake-up generation can also be gated at module level using the EnaWakeUp bit from GPIO_SYSCONFIG register.

Figure 25-20. GPIO_IRQWAKEN_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-18. GPIO_IRQWAKEN_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE	R/W	0h	Wakeup Enable for Interrupt Line 0h = Disable wakeup generation. 1h = Enable wakeup generation.

25.4.1.14 GPIO_SYSSTATUS Register (offset = 114h) [reset = 0h]

GPIO_SYSSTATUS is shown in Figure 25-21 and described in Table 25-19.

The GPIO_SYSSTATUS register provides the reset status information about the GPIO module. It is a read-only register; a write to this register has no effect.

Figure 25-21. GPIO_SYSSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESETDONE
R-0h							R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-19. GPIO_SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RESETDONE	R	0h	Reset status information. 0h = Internal Reset is on-going. 1h = Reset completed.

25.4.1.15 GPIO_CTRL Register (offset = 130h) [reset = 0h]

GPIO_CTRL is shown in Figure 25-22 and described in Table 25-20.

The GPIO_CTRL register controls the clock gating functionality. The DISABLEMODULE bit controls a clock gating feature at the module level. When set, this bit forces the clock gating for all internal clock paths. Module internal activity is suspended. System interface is not affected by this bit. System interface clock gating is controlled with the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG). This bit is to be used for power saving when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

Figure 25-22. GPIO_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					GATINGRATIO		DISABLEMODULE
R-0h					R/W-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-20. GPIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-1	GATINGRATIO	R/W	0h	Gating Ratio. Controls the clock gating for the event detection logic. 0h = Functional clock is interface clock. 1h = Functional clock is interface clock divided by 2. 2h = Functional clock is interface clock divided by 4. 3h = Functional clock is interface clock divided by 8.
0	DISABLEMODULE	R/W	0h	Module Disable 0h = Module is enabled, clocks are not gated. 1h = Module is disabled, clocks are gated.

25.4.1.16 GPIO_OE Register (offset = 134h) [reset = FFFFFFFFh]

GPIO_OE is shown in Figure 25-23 and described in Table 25-21.

The GPIO_OE register is used to enable the pins output capabilities. At reset, all the GPIO related pins are configured as input and output capabilities are disabled. This register is not used within the module, its only function is to carry the pads configuration. When the application is using a pin as an output and does not want interrupt generation from this pin, the application can/has to properly configure the Interrupt Enable registers.

Figure 25-23. GPIO_OE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUTEN[n]																															
R/W-FFFFFFFFh																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-21. GPIO_OE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	OUTPUTEN[n]	R/W	FFFFFFFFh	Output Data Enable 0h = The corresponding GPIO port is configured as an output. 1h = The corresponding GPIO port is configured as an input.

25.4.1.17 GPIO_DATAIN Register (offset = 138h) [reset = 0h]

GPIO_DATAIN is shown in Figure 25-24 and described in Table 25-22.

The GPIO_DATAIN register is used to register the data that is read from the GPIO pins. The GPIO_DATAIN register is a read-only register. The input data is sampled synchronously with the interface clock and then captured in the GPIO_DATAIN register synchronously with the interface clock. So, after changing, GPIO pin levels are captured into this register after two interface clock cycles (the required cycles to synchronize and to write the data). When the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG) is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 25-24. GPIO_DATAIN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-22. GPIO_DATAIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATAIN	R	0h	Sampled Input Data

25.4.1.18 GPIO_DATAOUT Register (offset = 13Ch) [reset = 0h]

GPIO_DATAOUT is shown in Figure 25-25 and described in Table 25-23.

The GPIO_DATAOUT register is used for setting the value of the GPIO output pins. Data is written to the GPIO_DATAOUT register synchronously with the interface clock. This register can be accessed with direct read/write operations or using the alternate Set/Clear feature. This feature enables to set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address.

Figure 25-25. GPIO_DATAOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAOUT																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-23. GPIO_DATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATAOUT	R/W	0h	Data to set on output pins

25.4.1.19 GPIO_LEVELDETECT0 Register (offset = 140h) [reset = 0h]

GPIO_LEVELDETECT0 is shown in Figure 25-26 and described in Table 25-24.

The GPIO_LEVELDETECT0 register is used to enable/disable for each input lines the low-level (0) detection to be used for the interrupt request generation. Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 25-26. GPIO_LEVELDETECT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT0[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-24. GPIO_LEVELDETECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LEVELDETECT0[n]	R/W	0h	Low Level Interrupt Enable 0h = Disable the IRQ assertion on low-level detect. 1h = Enable the IRQ assertion on low-level detect.

25.4.1.20 GPIO_LEVELDETECT1 Register (offset = 144h) [reset = 0h]

GPIO_LEVELDETECT1 is shown in Figure 25-27 and described in Table 25-25.

The GPIO_LEVELDETECT1 register is used to enable/disable for each input lines the high-level (1) detection to be used for the interrupt request generation. Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 25-27. GPIO_LEVELDETECT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT1[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-25. GPIO_LEVELDETECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LEVELDETECT1[n]	R/W	0h	High Level Interrupt Enable 0h = Disable the IRQ assertion on high-level detect. 1h = Enable the IRQ assertion on high-level detect.

25.4.1.21 GPIO_RISINGDETECT Register (offset = 148h) [reset = 0h]

GPIO_RISINGDETECT is shown in Figure 25-28 and described in Table 25-26.

The GPIO_RISINGDETECT register is used to enable/disable for each input lines the rising-edge (transition 0 to 1) detection to be used for the interrupt request generation.

Figure 25-28. GPIO_RISINGDETECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RISINGDETECT[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-26. GPIO_RISINGDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RISINGDETECT[n]	R/W	0h	Rising Edge Interrupt Enable 0h = Disable IRQ on rising-edge detect. 1h = Enable IRQ on rising-edge detect.

25.4.1.22 GPIO_FALLINGDETECT Register (offset = 14Ch) [reset = 0h]

GPIO_FALLINGDETECT is shown in Figure 25-29 and described in Table 25-27.

The GPIO_FALLINGDETECT register is used to enable/disable for each input lines the falling-edge (transition 1 to 0) detection to be used for the interrupt request generation.

Figure 25-29. GPIO_FALLINGDETECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALLINGDETECT[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-27. GPIO_FALLINGDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FALLINGDETECT[n]	R/W	0h	Falling Edge Interrupt Enable 0h = Disable IRQ on falling-edge detect. 1h = Enable IRQ on falling-edge detect.

25.4.1.23 GPIO_DEBOUNCENABLE Register (offset = 150h) [reset = 0h]

GPIO_DEBOUNCENABLE is shown in Figure 25-30 and described in Table 25-28.

The GPIO_DEBOUNCENABLE register is used to enable/disable the debouncing feature for each input line.

Figure 25-30. GPIO_DEBOUNCENABLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEBOUNCEENABLE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-28. GPIO_DEBOUNCENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBOUNCEENABLE[n]	R/W	0h	Input Debounce Enable 0h = Disable debouncing feature on the corresponding input port. 1h = Enable debouncing feature on the corresponding input port.

25.4.1.24 GPIO_DEBOUNCINGTIME Register (offset = 154h) [reset = 0h]

GPIO_DEBOUNCINGTIME is shown in Figure 25-31 and described in Table 25-29.

The GPIO_DEBOUNCINGTIME register controls debouncing time (the value is global for all ports). The debouncing cell is running with the debouncing clock (32 kHz), this register represents the number of the clock cycle(s) (31 s long) to be used.

Figure 25-31. GPIO_DEBOUNCINGTIME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEBOUNCETIME							
R/W-0h								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-29. GPIO_DEBOUNCINGTIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DEBOUNCETIME	R/W	0h	Input Debouncing Value in 31 microsecond steps. Debouncing Value = (DEBOUNCETIME + 1) * 31 microseconds

25.4.1.25 GPIO_CLEARDATAOUT Register (offset = 190h) [reset = 0h]

GPIO_CLEARDATAOUT is shown in Figure 25-32 and described in Table 25-30.

Writing a 1 to a bit in the GPIO_CLEARDATAOUT register clears to 0 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_CLEARDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

Figure 25-32. GPIO_CLEARDATAOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-30. GPIO_CLEARDATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Clear Data Output Register 0h = No effect 1h = Clear the corresponding bit in the GPIO_DATAOUT register.

25.4.1.26 GPIO_SETDATAOUT Register (offset = 194h) [reset = 0h]

GPIO_SETDATAOUT is shown in Figure 25-33 and described in Table 25-31.

Writing a 1 to a bit in the GPIO_SETDATAOUT register sets to 1 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_SETDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

Figure 25-33. GPIO_SETDATAOUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE[n]																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 25-31. GPIO_SETDATAOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLINE[n]	R/W	0h	Set Data Output Register 0h = No effect 1h = Set the corresponding bit in the GPIO_DATAOUT register.