OMAP35x Applications Processor General-Purpose I/O (GPIO) Interface

Texas Instruments OMAP™ Family of Products

Technical Reference Manual

Literature Number: SPRUFD5A
September 2008

Contents

List of Figures

List of Tables

General-Purpose I/O (GPIO) Interface

This chapter describes the general-purpose I/O (GPIO) interface for the OMAP35x Applications Processor.

Topic **Page**

1.1 General-Purpose I/O (GPIO) Interface Overview

The general-purpose interface combines six general-purpose input/output (GPIO) banks.

Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 192 (6 x 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.

These modules do not include pad control (pull up/down control, open-drain feature). For more information, see the *System Control Module* chapter.

1.1.1 Global Features

The GPIO modules include the following global features:

- Synchronous interrupt requests in active mode from each channel are processed by two identical interrupt generation submodules used independently by the imaging video and audio accelerator (IVA2.2) and the microprocessor unit (MPU) subsystems. One of these interrupts is mapped on the IVA2.2 subsystem interrupt controller and the other on the MPU subsystem interrupt controller.
- Asynchronous wake-up requests in idle mode from input channels are merged together to issue one wake-up signal per GPIO module.
- Data input (capture)/output (drive)
- Power management support

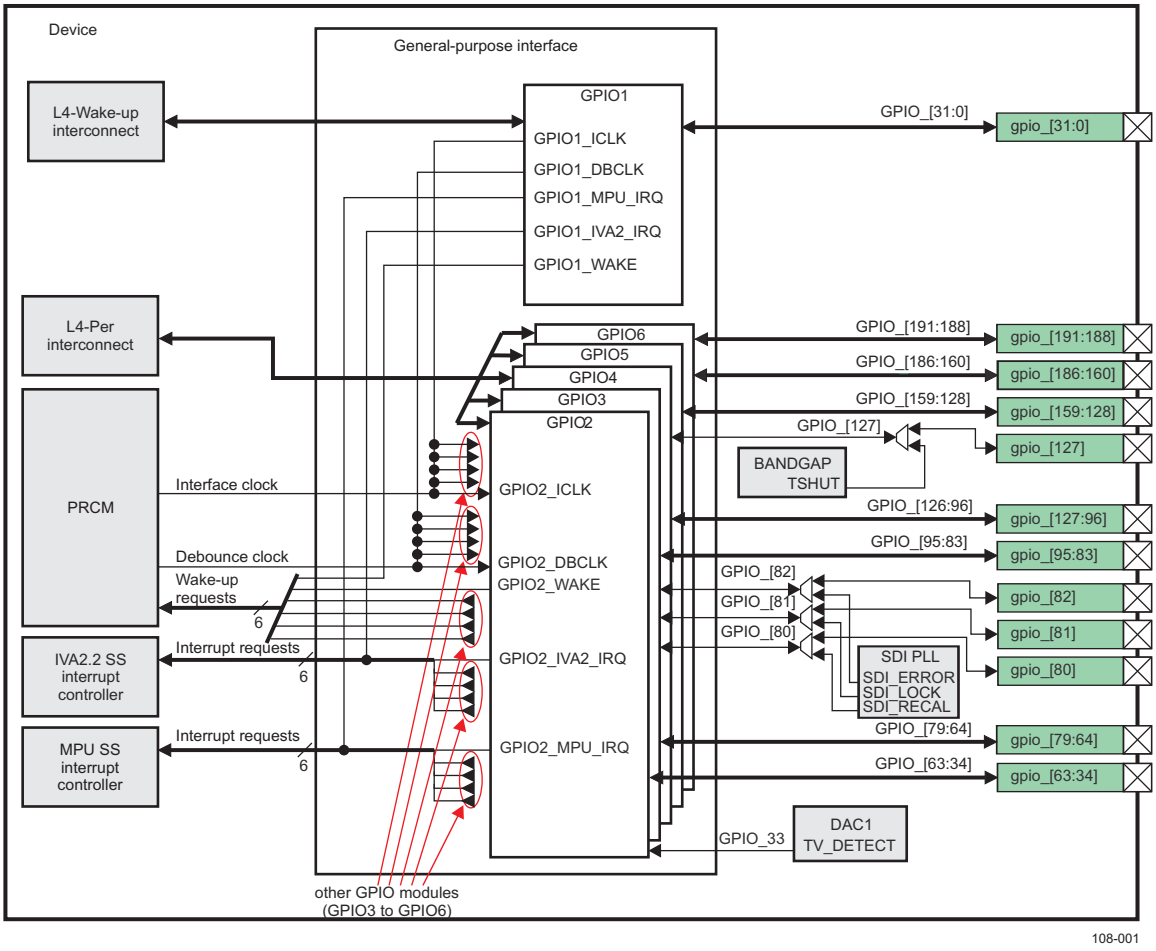
The general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance).

Each GPIO module produces a wake-up request signal to the power, reset, and clock management (PRCM) module.

Note: Some features may not be available or supported in your particular device. For more information, see Chapter 1, the *OMAP35x Family* section, and your device-specific data manual.

Figure 1-1 shows an overview of the general-purpose interface.

Figure 1-1. General-Purpose Interface Overview



Each channel in the GPIO modules has the following features:

- The GPIOi.GPIO_OE register controls the output capability for each pin.
- The output line level reflects the value written in the GPIOi.GPIO_DATAOUT register through the L4 interconnect.
- The input line can be fed to the GPIO module through an optional and configurable debounce cell. (The debouncing time value is global for all ports of one GPIO module, so up to five different debouncing time values are possible.)
- The input line value is sampled into the GPIOi.GPIO_DATAIN register and can be read through the L4 interconnect.
- In active mode, the input line can be used through level and edge detectors to trigger synchronous interrupts. The edge (rising, falling, or both) or the level (logical 0, logical 1, or both) used can be configured.
- In idle mode, the input line can be used to activate the asynchronous wake-up request (on edge detection: rising edge, falling edge, or both).

The module provides an alternative to the atomic test and set operations for the following registers:

- GPIOi.GPIO_DATAOUT
- GPIOi.GPIO_IRQENABLE1
- GPIOi.GPIO_IRQENABLE2
- GPIOi.GPIO_WAKEUPENABLE

For these registers, the modules implement the set-and-clear protocol register update (see Section 1.5.2, *Set-and-Clear Instructions*).

1.2 General-Purpose Interface Environment

The general-purpose interface combines six GPIO modules for a flexible, user-programmable, general-purpose input/output (I/O) controller. The general-purpose interface implements functions that are not implemented with the dedicated controllers in the device and require simple input and/or output software-controlled signals. The general-purpose interface allows a variety of custom connections and expands the I/O capabilities of the system to the real world.

Figure 1-2 shows a typical application using the general-purpose interface.

Figure 1-2. General-Purpose Interface Typical Application System Overview

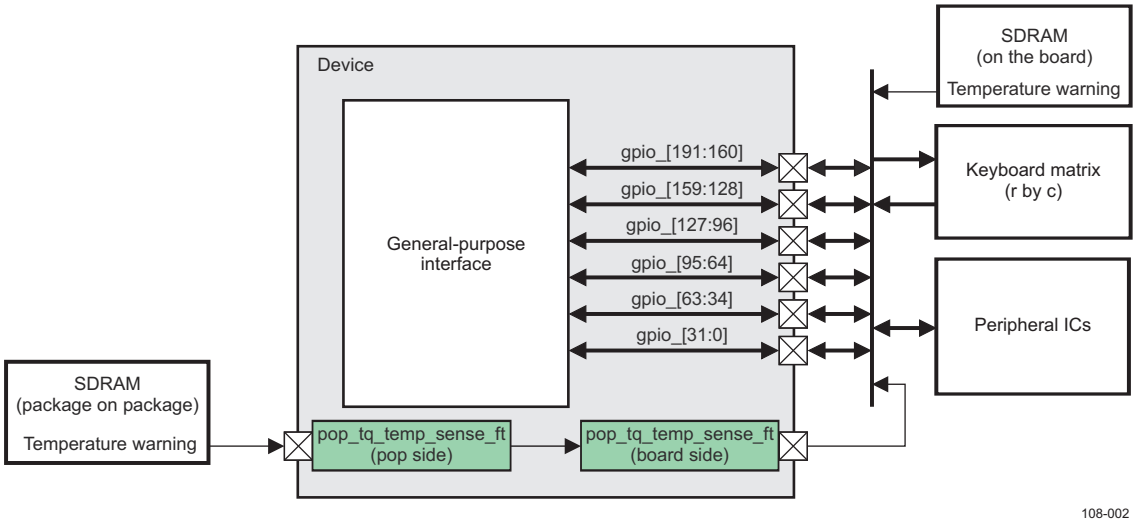


Figure 1-3. General-Purpose Interface Typical Application System Overview

Note: Temperature Sensing

Most memories provide a temperature sensor to control the auto-refresh duty cycle. The device monitors the temperature of the external memory using the pop_tq_temp_sense_ft ball and a GPIO input. To do this, pop_tq_temp_sense_ft is connected to a GPIO through the customer board. This feature is application-dependent.

CAUTION

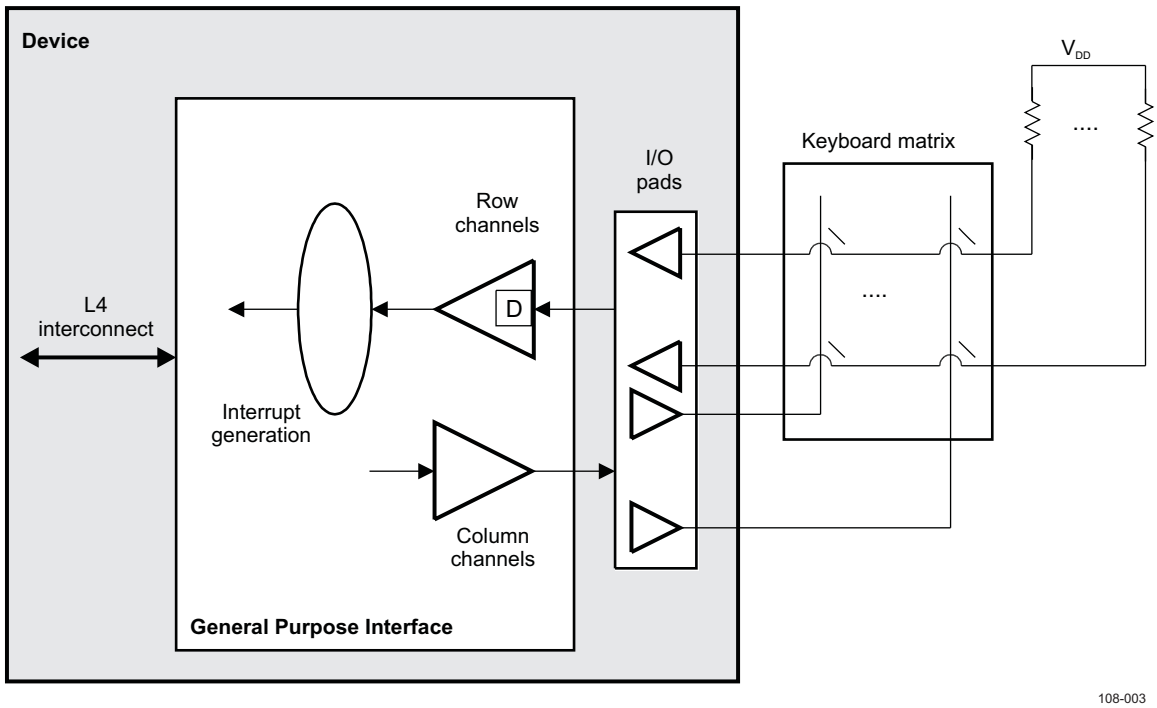
Due to buffer strength, an external serial resistor must be connected to the balls corresponding to gpio_120 to gpio_129.

The general-purpose interface can physically connect the device to a keyboard matrix and peripheral integrated circuits (ICs).

1.2.1 GPIO as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface. You can dedicate channels based on the keyboard matrix = * c). Figure 1-4 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.

Figure 1-4. General-Purpose Interface used as a Keyboard Interface



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 1.5.3, *Interrupt and wake-up*).

When the keyboard interrupt is received, the processor (MPU and/or IVA2.2 subsystem) 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.

1.2.2 General-Purpose Interface Functional Interfaces

1.2.2.1 Basic General-Purpose Interface Pins

Table 1-1 lists the interface pins of the general-purpose interface.

Table 1-1. General-Purpose Interface Functional Pins Description

Signal Name	I/O (1)	Description (2)	Module Reset Value
gpio_[31:0]	I/O	GPIO in configuration mode 4.	Input until software configuration
gpio_[186:34]	I/O	GPIO in configuration mode 4.	Input until software configuration
gpio_[191:188]	I/O	GPIO in configuration mode 4.	Input until software configuration

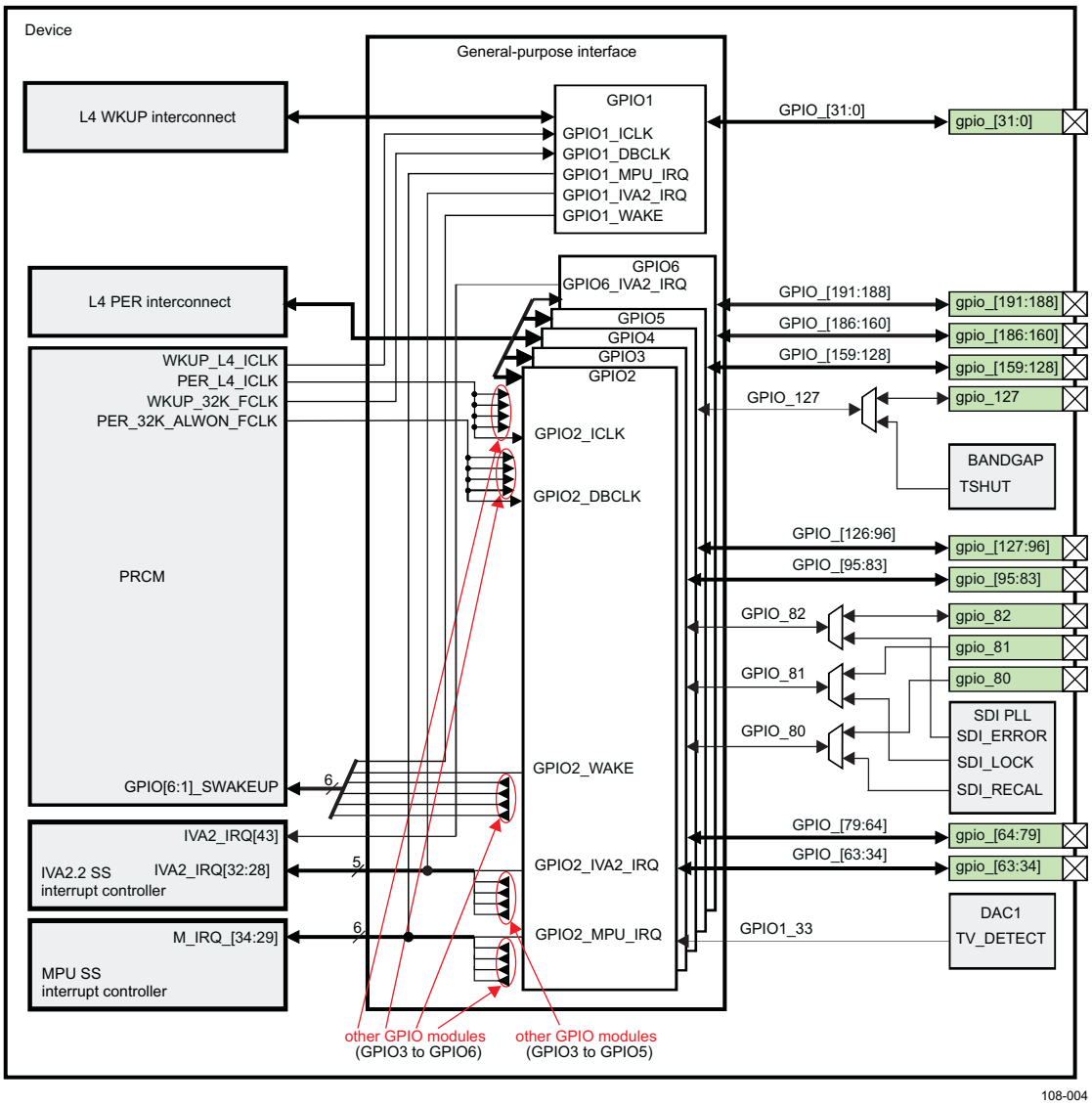
(1) I = Input, O = Output
(2) See the *System Control Module* chapter, for more information about pin configuration modes.

1.3 General-Purpose Interface Integration

1.3.1 Description

Figure 1-5 highlights the general-purpose interface integration in the device.

Figure 1-5. General-Purpose Interface Integration Overview



1.3.1.1 Clocking, Reset, and Power-Management Scheme

1.3.1.1.1 Clocking

Each GPIO module uses two clocks:

- Debounce clock: The 32-KHz debounce clock, GPIOi_DBCLK, (where I = 1, 2, 3, 4, 5, and 6, with one debounce clock per module), comes from the PRCM module and is used for the debounce cell logic (without the corresponding configuration registers). This cell can sample the input line and filters the input level using a programmed delay.

  For GPIO2 to GPIO6, this clock is controlled by the EN_GPIOi (where I = 2 to 6) bit PRCM.CM_FCLKEN_PER (0: disabled, 1: enabled the clock). For GPIO1, this clock is controlled by EN_GPIO1 bit PRCM.CM_FCLKEN_WKUP[3] (0: disabled, 1: enabled the clock) for GPIO1.

- Interface clock: The interface clock, GPIOi_ICLK (where I = 1, 2, 3, 4, 5, and 6), comes from the PRCM module and is used throughout the GPIO module (except within the debounce cell logic). The interface clock clocks the data exchanges between the L4 interconnect and the internal logic. The clock-gating features allow module power consumption to be adapted to the activity.

  For GPIO1, this clock is controlled by the EN_GPIO1 bit PRCM.CM_ICLKEN_WKUP[3] (0: disabled, 1:enabled the clock) and AUTO_GPIO1 bit PRCM.CM_AUTOIDLE_WKUP[3] (enables/disables automatic control of the interface clock). For GPIO2 to GPIO6, this clock is controlled by the EN_GPIOi (where I = 2 to 6) bit PRCM.CM_ICLKEN_PER (0: disabled, 1: enabled the clock) and AUTO_GPIOi (where I = 2 to 6) bit PRCM.CM_AUTOIDLE_PER (enables/disables automatic control of the interface clock). Table 25-2 describes the GPIO module clocks.

Table 1-2. Clocks

Attribute	Frequency	Name	Mapping	Comments
Debounce clock	32 KHz	GPIOi_DBCLK, where I = 2 to 6	PER_32K_ALWON_FCLK	Source is PRCM module.
		GPIO1_DBCLK	WKUP_32K_CLK	
Interface clock	Depends on PRCM registers settings	GPIOi_ICLK, where I = 2 to 6	PER_L4_ICLK	
		GPIO1_ICLK	WKUP_L4_ICLK	

1.3.1.1.2 Reset

The general-purpose interface can be reset by using the domain reset (hardware reset) or by setting a dedicated configuration bit (software reset) in each GPIO module.

- Hardware reset: The GPIO2 to GPIO6 modules are attached to the PER_RST reset domain. The GPIO1 module is attached to the WKUP_RST reset domain.

  The hardware reset has a global reset action on the GPIO modules of the general-purpose interface. All configuration registers and internal logic are reset when it is active (low level). In each GPIO module, the RESETDONE bit GPIOi.GPIO_SYSSTATUS[0] monitors the internal reset status; it is set when the reset completes. For more information, see the *Power, Reset, and Clock Management* chapter.

- Software reset: Each GPIO module has its own software reset using the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit (where I = 1, 2, 3, 4, 5, or 6). The software reset has the same effect as the hardware reset signal, but this reset can be applied on one module or more.

  Writing 1 to SOFTRESET bit GPIOi.GPIO_SYSCONFIG[1](where I = 1, 2, 3, 4, 5, or 6) resets the module. Bit value 1 remains until the reset is complete. When the software reset is complete, the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit is automatically reset to 0 and has the same effect as the hardware reset. The GPIOi.GPIO_SYSSTATUS[0] RESETDONE is cleared during a software reset. This bit is set to 1 when the software reset is complete.

1.3.1.1.3 Power Domain

The GPIO1 module is attached to the WKUP power domain (see the *Power, Reset, and Clock Management* chapter). This domain is composed of the logic permanently supplied to manage domain power state transitions and detect wake-up events. The WKUP power domain is continuously active. The GPIO2 to GPIO6 modules are attached to the PER power domain (see the *Power, Reset, and Clock Management* chapter). The PER power domain is not active continuously.

1.3.1.1.4 Power Management

1.3.1.1.4.1 Idle Scheme

To save dynamic consumption, an efficient idle scheme is based on the following:

- An efficient local autoclock gating for each module
- The implementation of control sideband signals between the PRCM module and each module

This enhanced idle control allows clocks to be activated and deactivated safely without requiring a complex software management.

The idle mode request, idle acknowledge, and wake-up request (GPIOi_SWAKEUP, where I = 1, 2, 3, 4, 5, and 6) are sideband signals between the PRCM module and the general-purpose interface (see Section 1.3.1.2, *Hardware Requests*).

1.3.1.1.4.2 Operating Modes

The following four operating modes are defined for the modules:

- Active mode: The module runs synchronously on the interface clock; interrupts can be generated based on the configuration and external signals.
- Idle mode: Power-saving mode with the module in a waiting state. The interface clock can be stopped, an interrupt cannot be generated, and a wake-up signal can be generated based on the configuration and external signals.

  If the debounce clock provided by the PRCM module is active, the debounce cell can sample and filter the input to generate a wake-up event. If the debounce clock is inactive, the debounce cell gates all input signals and thus cannot be used.
- Inactive mode: The module has no activity. The interface clock can be stopped, an interrupt cannot be generated, and the wake-up feature is inhibited.
- Disabled mode: The module is not used. The internal clock paths are gated, and an interrupt or wake-up request cannot be generated.

The idle and inactive modes are configured within the module and activated on request by the PRCM module (see the *Power, Reset, and Clock Management* chapter) through sideband signals (see Section 1.3.1.1.4.3, *System Power Management and Wake-up*).

The disabled mode is set by software through a dedicated configuration bit, the GPIOi.GPIO_CTRL[0] DISABLEMODULE bit (0: the module is enabled and clocks are not gated; 1: the module is disabled and clocks are gated). It unconditionally gates the internal clock paths that are not used for the L4 interconnect.

1.3.1.1.4.3 System Power Management and Wake-Up

The PRCM module can require the GPIO modules to be idled for power saving purposes.

The general-purpose interface has six identical idle mode request/acknowledge (handshake) mechanisms with the PRCM module (see Figure 1-5 and Section 1.3.1.2, *Hardware Requests*): one per GPIO module. The general-purpose interface allows the GPIO modules to enter idle mode based on the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE field.

The idle acknowledge depends on the configuration and activity of each GPIO module:

- Smart-idle mode (recommended)

  When the GPIO module is configured in smart-idle mode (GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE field [10]) and receives an idle request from the PRCM module (for GPIO2 to GPIO6: the corresponding bits in the PRCM.CM_FCLKEN_PER and PRCM.CM_ICLKEN_PER registers cleared to 0 or the corresponding bit in the PRCM.CM_AUTOIDLE_PER bit set to 1 and L4 interface clock idle transitions; for GPIO1: PRCM.CM_FCLKEN_ WKUP[3] EN_GPIO1 bit cleared to 0, PRCM.CM_ICLKEN_WKUP[3] EN_GPIO1 bit cleared to 0, or PRCM.CM_AUTOIDLE_WKUP[3] AUTO_ GPIO1 bit set to 1 and L4 interface clock idle transitions), the GPIO module checks for more activity (capture of the input GPIO pins in the GPIOi.GPIO_DATAIN register is complete with no pending interrupt; all interrupt status bits are cleared) ; and there is no access to GPIO.GPIO_DEBOUNCINGTIME register pending to be synchronized.

  Idle acknowledge is then asserted and the module enters in idle-mode. It waits for active system clock gating by the PRCM module (when all peripherals supplied by the same L4 interface clock domain are also ready for idle).

Idle mode (that is, when the PRCM module gates the interface clock), no interrupt occurs and the module is ready to issue a wake-up request.

When the expected transition occurs on an enabled GPIO input pin, the GPIO module exits from idle mode, if the GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP bit is set to 1 (wake-up capability enabled), and the corresponding bit in the PRCM.PM_WKEN_PER register is also set to 1 for the GPIO2 to GPIO6 modules, and/or the PRCM.PM_WKEN_WKUP[3] EN_GPIO1 bit is also set to 1 for GPIO1.

- Force-idle mode

  When the GPIO module is configured in force-idle mode (GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE field [00]) and receives an idle request from the PRCM module (for the GPIO2 to GPIO6: the corresponding bits in the PRCM.CM_FCLKEN_PER and PRCM.CM_ICLKEN_PER registers cleared to 0 or the corresponding bit in the PRCM.CM_AUTOIDLE_PER bit set to 1 and L4 interface clock idle transitions; for the GPIO1: PRCM.CM_FCLKEN_ WKUP[3] EN_GPIO1 bit cleared to 0, PRCM.CM_ICLKEN_WKUP[3] EN_GPIO1 bit cleared to 0, or PRCM.CM_AUTOIDLE_WKUP[3] AUTO_ GPIO1 bit set to 1 and L4 interface clock idle transitions), the GPIO module waits unconditionally for active system clock gating by the PRCM module. (This occurs only when all peripherals supplied by the same L4 interface clock domain are also ready for idle.)

  When in idle mode (that is, when the PRCM module gates the interface clock), the module (in inactive mode) has no activity, the interface clock paths are gated, an interrupt cannot be generated, and the wake-up feature is totally inhibited.

- No-idle mode

  When the GPIO module is configured in no-idle mode (GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE field [01]) and receives an idle request from the PRCM module (for the GPIO2 to GPIO6: the corresponding bits in the PRCM.CM_FCLKEN_PER and PRCM.CM_ICLKEN_PER registers cleared to 0 or the corresponding bit in the PRCM.CM_AUTOIDLE_PER bit set to 1 and L4 interface clock idle transitions; for the GPIO1: PRCM.CM_FCLKEN_ WKUP[3] EN_GPIO1 bit cleared to 0, PRCM.CM_ICLKEN_WKUP[3] EN_GPIO1 bit cleared to 0 or PRCM.CM_AUTOIDLE_WKUP[3] AUTO_ GPIO1 bit set to 1 and L4 interface clock idle transitions), the GPIO module does not go to the idle mode and the idle acknowledge is never sent.

Note: The GPIO2 to GPIO6 idle state can be checked by reading the corresponding status bits in the PRCM.CM_IDLEST_PER register (0: active, 1: idle) and is idle only when the GPIO2 to GPIO6 modules are configured in smart-idle mode and have asserted their idle acknowledge.

The GPIO1 idle state can be checked by the PRCM.CM_IDLEST_WKUP[3] ST_GPIO1 bit (0: idle, 1: active) and is idle only when the GPIO1 module is configured in smart-idle mode and has asserted its idle acknowledge.

The GPIO2 to GPIO6 wake-up status can be checked by accessing the corresponding bits in the PRCM.PM_WKST_PER register (read 0: no wake-up occurred; read 1: wake-up occurred; write 1: status bit reset).

The GPIO1 wake-up status can also be checked by the PRCM.PM_WKST_WKUP[3] ST_GPIO1 bit (read 0: no wake-up occurred; read 1: wake-up occurred; write 1: status bit reset).

1.3.1.1.4.4 Module Power Saving

The GPIO module has local power management by internal clock-gating features:

- Internal interface clock gating: The clock for the L4 interconnect logic can be gated when the module is not accessed, if the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit is set. Otherwise, this logic is free-running on the interface clock.
- Clock gating for the input data sample logic: Clock for the input data sample logic can be gated when the data in register is not accessed.
- Clock gating for the event detection logic: Each GPIO module implements four clock groups used for the logic in the synchronous events detection. Each group of eight input GPIO pins has a separate enable signal depending on the edge/level detection register setting. If a group requires no detection, the corresponding clock is gated off (see Section 1.5.1, *Power Saving by Grouping the Edge/Level Detection*). All channels are also gated using a one-out-of-N scheme. N is the GATINGRATIO field

GPIOi.GPIO_CTRL[2:1] and can take the values 1 (b00), 2 (0b01), 4 (b10), or 8 (0b11). 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 2, 4, or 8, this logic is running at the equivalent frequency of interface clock frequency divided by N.

- Inactive mode: In inactive mode, all internal clock paths are gated.
- Disabled mode: All internal clock paths not used for the L4 interconnect are gated. The GPIOi.GPIO_CTRL[0] DISABLEMODULE bit controls a clock-gating feature at the module level. When set to 1, this bit forces clock gating for all internal clock paths. Module internal activity is suspended. The L4 interconnect is not affected by this bit.

  The interface clock gating is controlled with the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit, which is used to save power 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.

1.3.1.2 Hardware Requests

1.3.1.2.1 Interrupt Requests

All interrupt sources (the 32 input GPIO channels) are merged to issue two synchronous interrupt requests in each GPIO module. Thus, the general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance).

Synchronous interrupt request lines 1 and 2 are active depending on their respective interrupt enable 1 and 2 registers (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2).

- Synchronous interrupt request line 1 is mapped on the MPU interrupt controller.
- Synchronous interrupt request line 2 is mapped on the IVA2.2 interrupt controller.

Table 1-3 lists the interrupt lines that are driven out from the general-purpose interface to the MPU subsystem and IVA2.2 subsystem interrupt controller.

Table 1-3. Interrupts

Name	Mapping	Comments
GPIO1 Module		
GPIO1_IVA2_IRQ	IVA2_IRQ[28]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO1_MPU_IRQ	M_IRQ_29	Destination is the MPU subsystem interrupt controller.
GPIO2 Module		
GPIO2_IVA2_IRQ	IVA2_IRQ[29]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO2_MPU_IRQ	M_IRQ_30	Destination is the MPU subsystem interrupt controller.
GPIO3 Module		
GPIO3_IVA2_IRQ	IVA2_IRQ[30]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO3_MPU_IRQ	M_IRQ_31	Destination is the MPU subsystem interrupt controller.
GPIO4 Module		
GPIO4_IVA2_IRQ	IVA2_IRQ[31]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO4_MPU_IRQ	M_IRQ_32	Destination is the MPU subsystem interrupt controller.
GPIO5 Module		
GPIO5_IVA2_IRQ	IVA2_IRQ[32]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO5_MPU_IRQ	M_IRQ_33	Destination is the MPU subsystem interrupt controller.
GPIO6 Module		
GPIO6_IVA2_IRQ	IVA2_IRQ[43]	Destination is the IVA2.2 subsystem interrupt controller.
GPIO6_MPU_IRQ	M_IRQ_34	Destination is the MPU subsystem interrupt controller.

1.3.1.2.1.1 Wake-Up Generation

The GPIO1 module of the general-purpose interface is attached to the WKUP power domain (see the *Power, Reset, and Clock Management* chapter) and can wake up the system.

Note: The GPIO2 to GPIO6 modules belong to the PER power domain and thus have wake-up system capability only when the PER power domain is active.

All wake-up sources (the 32 input GPIO channels) are merged together to issue a single asynchronous wake-up request in each GPIO module following the expected transition(s) (based on register programming). Each GPIO module generates a wake-up signal to the PRCM module.

Note: Only gpio_1, gpio_9, gpio_10, gpio_11, gpio_30 and gpio_31 can be used to generate a direct wake-up event. The other GPIO1 pins can not be used to generate a direct wake up event because they are connected to the device I/O pad logic in the CORE power domain (VDD2). When the CORE power domain is off, the I/O pins of the GPIO1 module, which are supplied by VDD2, cannot generate a wake-up event.

The asynchronous wake-up request line is active based on the GPIOi.GPIO_WAKEUPENABLE register (where I = 1, 2, 3, 4, 5, and 6).

CAUTION

The wake-up capabilities of the GPIO2 to GPIO6 modules are operational only when the PER power domain is active.

Table 1-4 shows the wake-up signals mapping.

Table 1-4. Wake-Up Signals

Name	Mapping	Comments
GPIOi_WAKE	GPIOi_SWAKEUP	Where I = 1, 2, 3, 4, 5, and 6. Destination is the PRCM module.

Table 1-5 describes the GPIO channels.

Table 1-5. GPIO Channel Description

Channel Number	Type (1)	Mapping	Wake-Up Feature	Comments
GPIO1 Module				
[31:0]	I/O	gpio_[31:0]	Yes	GPIO (2)
GPIO2 Module				
[0]	I	-	No	Not available on external balls. Read value is always 0.
[1]	I	TV_DETECT	Yes	Internal TV detection signal from the 10-bit composite/luma video DAC1
[31:2]	I/O	gpio_[63:34]	Yes	GPIO (2)
GPIO3 Module				
[15:0]	I/O	gpio_[79:64]	Yes	GPIO (2)
[16]	I/O	gpio_80	Yes	GPIO (2)

(1) I = Input, O = Output
(2) Configuration mode 4. See the *System Control Module* chapter.

Table 1-5. GPIO Channel Description (continued)

Channel Number	Type [1]	Mapping	Wake-Up Feature	Comments
	I	SDI_RECAL	Yes	Internal SDI_RECAL signal from the SDI PLL module [3]
[17]	I/O	gpio_81	Yes	GPIO [2]
	I	SDI_LOCK	Yes	Internal SDI_LOCK signal from the SDI PLL module [3]
[18]	I/O	gpio_82	Yes	GPIO [2]
	I	SDI_ERROR	Yes	Internal SDI_ERROR signal from the SDI PLL module [3]
[31:19]	I/O	gpio_[95:83]	Yes	GPIO [2]
GPIO4 Module				
[2:0]	I/O	gpio_[98:96]	Yes	GPIO [2]
[4:3]	I	gpio_[99:100]	Yes	GPIO [2]
	I	cam_d0 cam_d1	No [4]	Camera parallel data lines. [4]
	I	csi_dx2 csi_dy2	No[4]	Camera serial mode clock and data lines.
[15:5]	I/O	gpio_[111:101]	Yes	GPIO [2]
[19:16]	I	gpio_[115:112]	Yes	GPIO [2]
	I	csi_dx0, csi_dy0, csi_dx1, csi_dy1	No [4]	Camera serial mode clock and data lines. [4]
[30:20]	I/O	gpio_[126:116]	Yes	GPIO [2]
[31]	I/O	gpio_127	Yes	GPIO [2]
	I	TSHUT	Yes	Internal TSHUT signal from the BANDGAP module for the SRAMs LDOs [3]
GPIO5 Module				
[31:0]	I/O	gpio_[159:128]	Yes	GPIO [2]
GPIO6 Module				
[26:0]	I/O	gpio_[186:160]	Yes	GPIO [2]
[27]	I	-	No	Not available on external balls. Read value is always 0.
[31:28]	I/O	gpio_[191:188]]	Yes	GPIO [2]

(3) All configuration modes except configuration mode 4. See the *System Control Module* chapter.
(4) See Chapter 12, Camera ISP

Note: The thermal shutdown comparator output signal (TSHUT) is an output from the BANDGAP module. This signal is low during normal operation and goes high during a thermal shutdown event. When channel 31 of the GPIO4 is not connected to a ball of the device (the corresponding pin is configured in a mode different from the configuration mode 4; see \the *System Control Module* chapter, for more information about pin configuration), TSHUT is connected to channel 31 of the GPIO4, and an interrupt can be generated when a low-to-high transition occurs on TSHUT whether or not the interrupt generation for channel 31 of the GPIO4 is correctly configured.

1.4 General-Purpose Interface Functional Description

Figure 1-6 shows the general-purpose interface description.

Figure 1-6. General-Purpose Interface Description

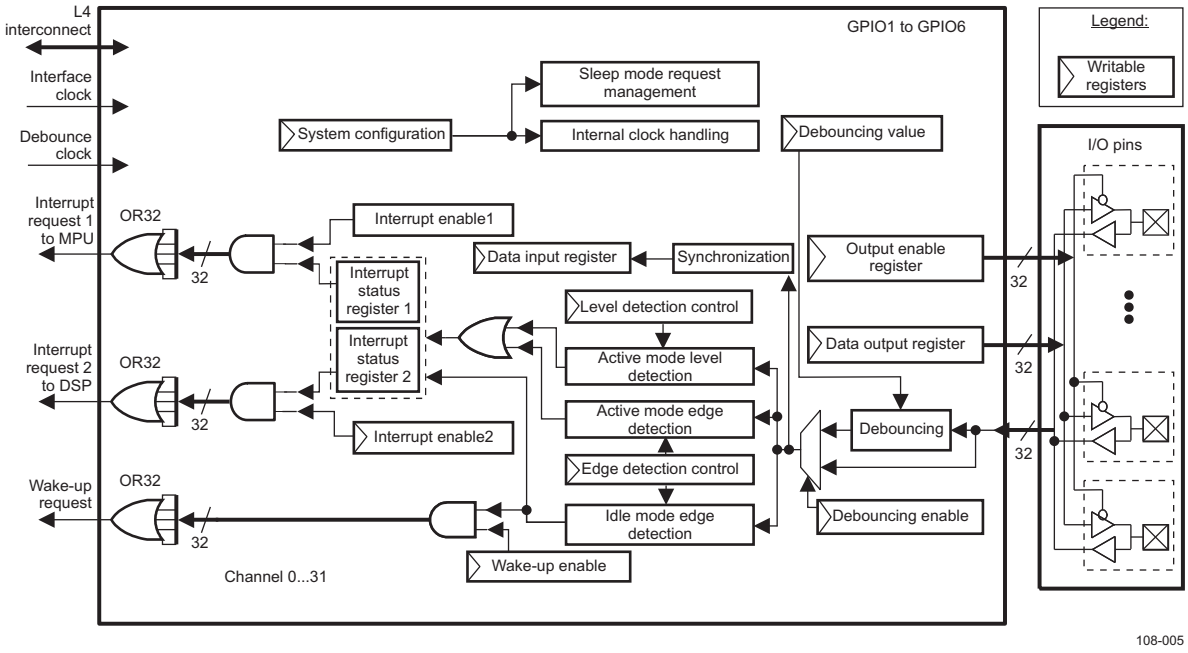
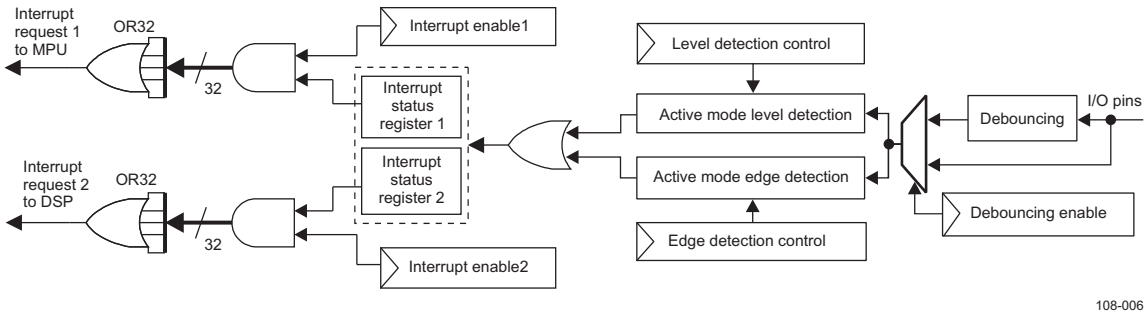


Figure 1-6 details the GPIO modules in the general-purpose interface block diagram with their configuration registers and their main functional paths:

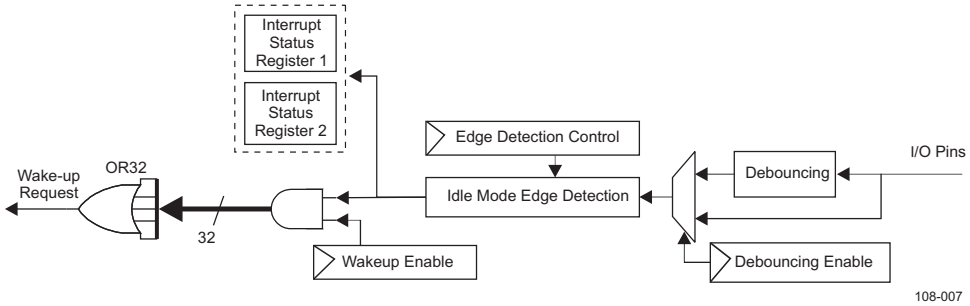
- The synchronous path (for active mode operation) used to generate a synchronous interrupt request on expected event detection on any input GPIO; the synchronous interrupt request lines 1 and 2 are active based on their respective interrupt enable 1 and 2 registers (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2). See Figure 1-7.

Figure 1-7. Synchronous Path



- The asynchronous path (for idle mode operation) used to generate an asynchronous wake-up request on the expected edge detection on any input GPIO; the asynchronous wake-up request line is active based on the wake-up enable register. See Figure 1-8.

Figure 1-8. Asynchronous Path



- The blocks handling the internal clock (clock gating) and managing the sleep mode request/acknowledge protocol (enabling the synchronous path in active mode and the asynchronous path in idle mode).

1.4.1 Interrupt and Wake-Up Features

1.4.1.1 Synchronous Path: Interrupt Request Generation

The general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance). The 12 interrupt signals are GPIOi_MPU_IRQ (used by the MPU subsystem) and GPIOi_IVA2_IRQ (used by the IVA2.2 subsystem), where I = 1, 2, 3, 4, 5, and 6.

Synchronous interrupt requests from each channel are processed by two identical interrupt generation submodules used independently by the IVA2.2 subsystem and the MPU subsystem. Each submodule controls its own synchronous interrupt request line and has its own interrupt enable (GPIOi.GPIO_IRQENABLE1 or GPIOi.GPIO_IRQENABLE2) and interrupt status (GPIOi.GPIO_IRQSTATUS1 or GPIOi.GPIO_IRQSTATUS2) registers. The interrupt enable register selects the channel(s) considered for the interrupt request generation, and the interrupt status register determines which channel(s) activate the interrupt request. Event detection on GPIO channels is reflected in the interrupt status registers independent of the content of the interrupt enable registers.

In active mode, when the GPIO configuration registers are set to enable the interrupt generation (see Section 1.5.3, *Interrupt and wake-up*), a synchronous path samples the transitions and levels on the input GPIO with the internally gated interface clock (see Section 1.3.1.1.4.4, *Module Power Saving*). When an event matches the programmed settings (see Section 1.5.3, *Interrupt and wake-up*), the corresponding bit in the interrupt status register is set to 1 and, on the following interface clock cycle, the interrupt lines 1 and/or 2 are activated (depending on the interrupt enable registers).

Because of 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 equals *N* times the interface clock period; see Section 1.3.1.1.4.4, *Module Power Saving*). This minimum pulse width must 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.

Because the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This latency must not exceed four internally gated interface clock cycles + one interface clock cycle when the debounce feature is not used.

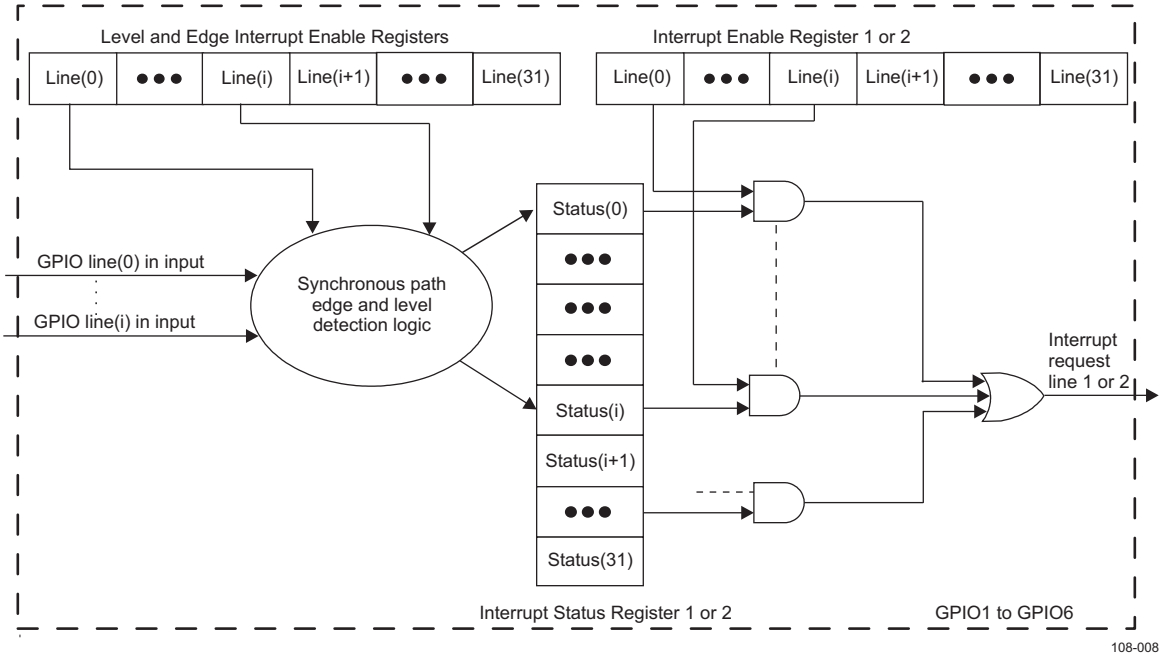
When the debounce feature is active, the latency depends on the debouncing time register (GPIOi.GPIO_DEBOUNCINGTIME) value (see Section 1.5.5, *Debouncing Time*) and is less than three internally gated interface clock cycles + two interface clock cycle + GPIOi.GPIO_DEBOUNCINGTIME register value debounce clock cycles + three debounce clock cycles.

Synchronous interrupt request line 1 is mapped on the MPU interrupt controller.

Synchronous interrupt request line 2 is mapped on the IVA2.2 interrupt controller.

Figure 1-9 shows an overview of the interrupt request generation.

Figure 1-9. Interrupt Request Generation



1.4.1.2 Asynchronous Path: Wake-Up Request Generation

The general-purpose interface has six wake-up lines (one wake-up line per GPIO module instance) connected to the PRCM module.

Asynchronous wake-up requests from input channels are merged to issue one wake-up signal to the system per GPIO module. The wake-up enable register (GPIOi.GPIO_WAKEUPENABLE) selects the channel(s) considered for the wake-up request generation. The asynchronous wake-up request is reflected into the synchronous interrupt status registers (GPIOi.GPIO_IRQSTATUS1 and GPIOi.GPIO_IRQSTATUS2).

In idle mode (the interface clock is shut down and the GPIO configuration registers are programmed; see Section 1.5.3, *Interrupt and wake-up*), an asynchronous path detects the expected transition(s) on a GPIO input (based on register programming) and activates an asynchronous wake-up request by the sideband signal (GPIOi_SWAKEUP, where I = 1, 2, 3, 4, 5, and 6), if the wake-up enable register is set.

When the system is awakened, the interface clock is restarted and synchronously set to 1 based on the input GPIO pin triggering the wake-up request and the corresponding bit in the interrupt status registers (GPIOi.GPIO_IRQSTATUS1 and GPIOi.GPIO_IRQSTATUS2). On the following internal clock cycle, the interrupt lines 1 and/or 2 are active (active low) when the corresponding bits are set in the interrupt enable registers (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2).

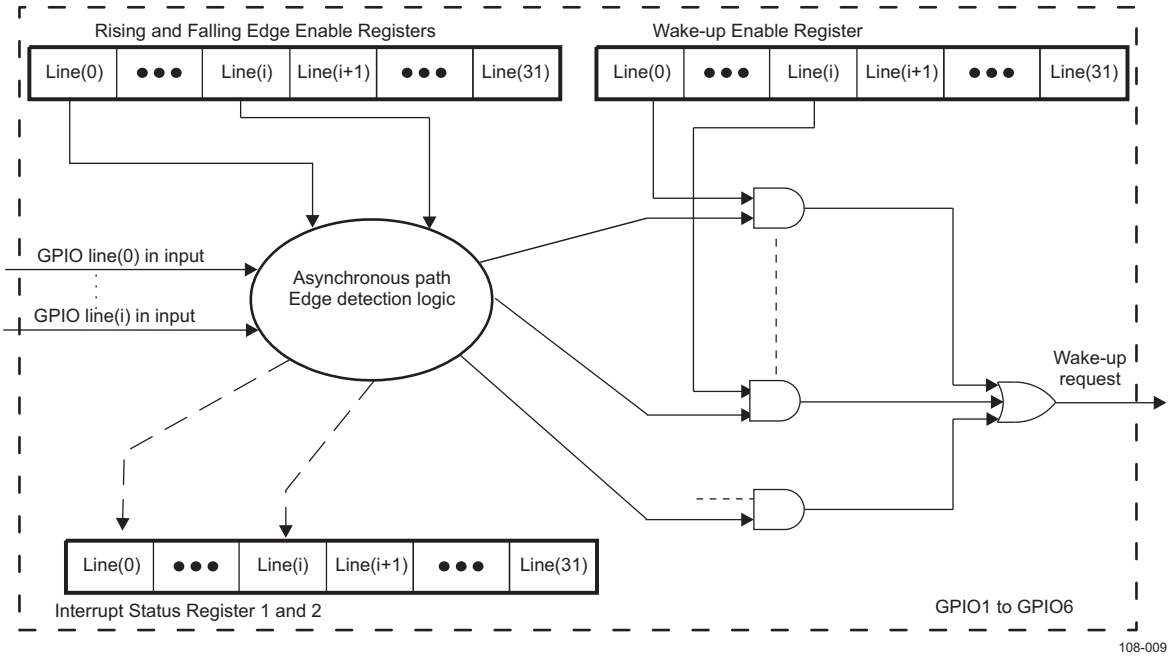
Note: When debouncing is not enabled, a minimum input pulse width does not trigger the wake-up request because there is no sampling operation.

When debouncing is enabled, the minimum pulse width is set by the specified debouncing time.

The GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP bit enables or disables the GPIO wake-up feature globally. If the bit is 0, the wake-up enable register (GPIOi.GPIO_WAKEUPENABLE) has no effect.

Figure 1-10 shows an overview of the wake-up request generation.

Figure 1-10. Wake-Up Request Generation



1.4.1.3 Interrupt (or Wake-Up) Line Release

When the host processor (the MPU and/or IVA2.2 subsystem in the device) receives an interrupt request issued by the GPIO module, it reads the corresponding interrupt status register (GPIOi.GPIO_IRQSTATUS1 or GPIOi.GPIO_IRQSTATUS2) to determine which GPIO input 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 1 in the corresponding bit of the interrupt status register. If there is still a pending interrupt request to serve (all bits in the interrupt status register that are not masked by the interrupt enable register are not cleared), the interrupt line is reasserted.

Note: The status bit must be reset to re-enter idle mode.

1.5 General-Purpose Interface Basic Programming Model

1.5.1 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:

GPIOi.GPIO_LEVELDETECT0
GPIOi.GPIO_LEVELDETECT1
GPIOi.GPIO_RISINGDETECT
GPIOi.GPIO_FALLINGDETECT

are set to 0x01 01 01 01, then all clocks are active (power consumption is high).

are set to 0x00 00 00 FF, then a single clock is active (power saving).

Note: When the clocks are enabled by writing to the GPIOi.GPIO_LEVELDETECT0, GPIOi.GPIO_LEVELDETECT1, GPIOi.GPIO_RISINGDETECT, and GPIOi.GPIO_FALLINGDETECT registers, the detection starts after five 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 and 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.

1.5.2 Set-and-Clear Instructions

1.5.2.1 Description

The GPIO module implements the set-and-clear protocol register update for the GPIOi.GPIO_DATAOUT, GPIOi.GPIO_IRQENABLE1, GPIOi.GPIO_IRQENABLE2, and GPIOi.GPIO_WAKEUPENABLE 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.

1.5.2.2 Clear Instruction

1.5.2.2.1 Clear Registers Addresses

Clear interrupt enable registers (GPIOi.GPIO_CLEARIRQENABLE1 and GPIOi.GPIO_CLEARIRQENABLE2).

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.

Clear wake-up enable register (GPIOi.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.

Clear data output register (GPIOi.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.

1.5.2.2.2 Clear Instruction Example

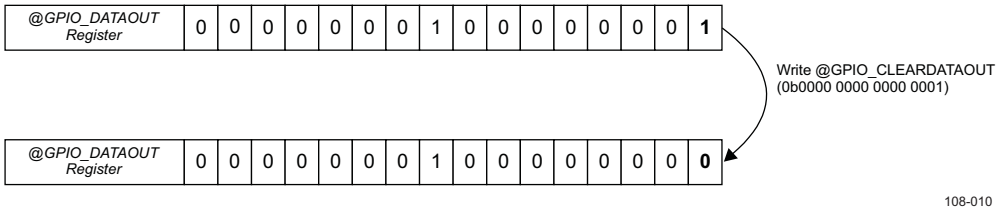
Assume the data output register (or one of the interrupt/wake-up enable register) contains the binary value, 0x0000 0001 0000 0001, and you want to clear the bit 0.

With the clear instruction feature, write 0x0000 0000 0000 0001 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 0x0000 0001 0000 0000; the bit 0 is cleared.

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

Figure 1-11 shows an example of a clear instruction.

Figure 1-11. Write @ GPIO_CLEARDATAOUT Register Example



1.5.2.3 Set Instruction

1.5.2.3.1 Set Registers Addresses

Set interrupt enable registers (GPIOi.GPIO_SETIRQENABLE1 and GPIOi.GPIO_SETIRQENABLE2).

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.

Set wake-up enable register (GPIOi.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.

Set data output register (GPIOi.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.

1.5.2.3.2 Set Instruction Example

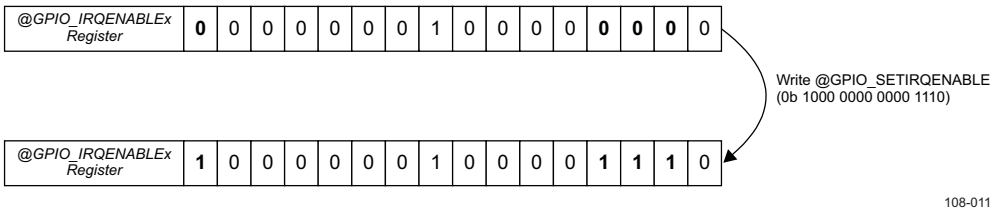
Assume the interrupt enable1 (or enable2) register (or the data output register) contains the binary value, 0x0000 0001 0000 0000, and you want to set the bits 15, 3, 2, and 1.

With the set instruction feature, write 0x1000 0000 0000 1110 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 0x1000 0001 0000 1110; the bits 15, 3, 2, and 1 are set.

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

Figure 1-12 shows an example of a set instruction.

Figure 1-12. Write @ GPIO_SETIRQENABLEx Register Example



The set wake-up enable register offers the same feature with the wake-up enable register.

1.5.3 Interrupt and Wake-up

1.5.3.1 Involved Configuration Registers

- Interrupt enable registers (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2)

  The interrupt enable1 (or interrupt enable2) register allows masking of the expected transition on input GPIO to prevent the generation of an interrupt request on line1 (or line2). The interrupt enable registers are programmed synchronously with the interface clock.

  These registers can be accessed with direct read/write operations or using the alternate set and clear protocol register update feature. This feature enables to set or clear specific bits of these registers with a single write access to the corresponding set interrupt enable1 (or interrupt enable2) registers (or to the clear interrupt enable1 [or interrupt enable2] registers) address (see Section 1.5.2, *Set-and-Clear Instructions*).

- Wake-up enable register (GPIOi.GPIO_WAKEUPENABLE)

  The wake-up enable register allows masking of the expected transition on input GPIO to prevent the generation of a wake-up request. The wake-up enable register is programmed synchronously with the interface clock before any idle mode request coming from the host processor.

  This register can be accessed with direct read/write operations or by using the alternate set and clear protocol register update feature. This feature allows setting or clearing specific bits of this register with a single write access to the set wake-up enable register (or to the clear wake-up enable register) address (see Section 1.5.2, *Set-and-Clear Instructions*).

  Note: It must be a correlation between Wake-up enable and interrupt enable registers. If a GPIO pin has a Wake-up configured on it, it should also have the corresponding interrupt enabled (on one of the 2 interrupt lines). Otherwise, it is possible to have a Wake-up event, but after exiting the Idle state, no interrupt will be generated, thus the corresponding bit from the interrupt status register will not be cleared, and the module will not acknowledge a future Idle Request.

- Interrupt status registers (GPIOi.GPIO_IRQSTATUS1 and GPIOi.GPIO_IRQSTATUS2).

  The interrupt status1 (or interrupt status2) register determines which of the input GPIO pins triggered

the interrupt line1 (or interrupt line2) request (or the wake-up line).

When a bit in this register is set to 1, it indicates that the corresponding GPIO pin is requesting the interrupt (or the wake-up). To reset a bit in this register, write 1 to the appropriate bit. However, an interrupt cannot be generated by writing 1 to the interrupt status1 (or interrupt status2) register.

If 0 is written to a bit in this register, the value remains unchanged. The interrupt status1 (or interrupt status2) register is synchronous with the interface clock. In idle mode, the event is detected via an asynchronous path, and the corresponding bit in the interrupt status1 and interrupt status2 registers are set when the GPIO module is awake.

Note: The wake-up capabilities of the GPIO2 to GPIO6 modules are operational only when the PER power domain is active.

1.5.3.2 Description

To generate interrupt request to a host processor (the MPU and/or DSP subsystem in the device) at a defined event (level or edge logic transition) occurring on a GPIO pin (interrupt source), the GPIO configuration registers must be programmed as follows:

1. The GPIO channel must be configured as input by the output enable register (write 1 to the corresponding bit of the GPIOi.GPIO_OE register).
2. The expected event(s) on the GPIO input to trigger the interrupt request must be selected in the low-level interrupt enable register (write 1or 0 to the corresponding bit of GPIOi.GPIO_LEVELDETECT0), and/or high-level interrupt enable register (write 1 or 0 to the corresponding bit of GPIOi.GPIO_LEVELDETECT1), and/or rising-edge interrupt/wake-up enable register (write 1or 0 to the corresponding bit of GPIOi.GPIO_RISINGDETECT), and/or falling edge interrupt/wake-up enable register (write 1or 0 to the corresponding bit of GPIOi.GPIO_FALLINGDETECT).

   Note: Interrupt generation on both edges on one input is configured by setting the corresponding bit to 1 in the rinsing detect enabling register (GPIOi.GPIO_RISINGDETECT) and falling detect enabling register (GPIOi.GPIO_FALLINGDETECT) along with the interrupt enable by setting the corresponding bit to 1 in one or both interrupt enable registers (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2).

   Enabling at the same time high level detection and low level detection for one given pin makes a constant interrupt generator.

3. Interrupts from the GPIO channel must be enabled in the interrupt 1 enable register (write 1 to the corresponding bit of GPIOi.GPIO_IRQENABLE1 register) and/or the interrupt 2 enable register (write 1 to the corresponding bit of GPIOi.GPIO_IRQENABLE2 register).

To configure a GPIO module to sent a wake-up request to the PRCM at a defined event (logic transition) occurring on a GPIO pin (wake-up source), the GPIO configuration registers must be programmed as follows:

1. The GPIO pin must be configured as input by the output enable register (write 1 to the corresponding bit of the GPIOi.GPIO_OE register).
2. The expected event(s) on the GPIO input to trigger the wake-up request must be selected in the rising-edge interrupt/wake-up enable register (write 1 or 0 to the corresponding bit of GPIOi.GPIO_RISINGDETECT) and/or falling-edge interrupt/wake-up enable register (write 1 or 0 to the corresponding bit of GPIOi.GPIO_FALLINGDETECT). The wake-up request can only be generated on edge transitions.
3. The GPIO channel must be enabled in the wake-up enable register (write 1 to the corresponding bit of the GPIOi.GPIO_WAKEUPENABLE).
4. The wake-up request generation on the expected transition occurring on the GPIO input pins must enable for the module (write 1 to the corresponding bit of the GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP) .

CAUTION

For each GPIO channel used, do not forget to configure the corresponding pad configuration registers in the *System Control Module* chapter.

After servicing the interrupt, the status bit in the interrupt status register (GPIOi.GPIO_IRQSTATUS1 or GPIOi.GPIO_IRQSTATUS2) must be reset and the interrupt line released (by writing 1 in the corresponding bit of the interrupt status register) before enabling an interrupt for the GPIO channel in the interrupt enable register (GPIOi.GPIO_IRQENABLE1 or GPIOi.GPIO_IRQENABLE2) to prevent the occurrence of unexpected interrupts when enabling an interrupt for the GPIO channel.

1.5.4 Data Input (Capture)/Output (Drive)

The output enable register (GPIOi.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 the GPIOi.GPIO_OE register), the value of the corresponding bit in the GPIOi.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 output data register (GPIOi.GPIO_SETDATAOUT) or to the clear output data register (GPIOi.GPIO_CLEARDATAOUT) address (see Section 1.5.2, *Set-and-Clear Instructions*). 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 (GPIOi.GPIO_ WAKEUPENABLE) and the interrupt enable (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2) registers.

When configured as an input (the desired bit set to 1 in the GPIOi.GPIO_OE register), the state of the input can be read from the corresponding bit in the GPIOi.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 (see Section 1.5.2, *Set-and-Clear Instructions*). 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 (GPIOi.GPIO_WAKEUPENABLE) and the interrupt enable (GPIOi.GPIO_IRQENABLE1 and GPIOi.GPIO_IRQENABLE2) registers to the interrupt and wake-up feature as needed.

1.5.5 Debouncing Time

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

1. The GPIO pin must be configured as input in the output enable register (write 1 to the corresponding bit of the GPIOi.GPIO_OE register)
2. The debouncing time must be set in the debouncing time register (GPIOi.GPIO_DEBOUNCINGTIME)

   The debouncing value register (GPIOi.GPIO_DEBOUNCINGTIME) 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:

   Required input line stable = (GPIOi.GPIO_DEBOUNCINGTIME[7:0] DEBOUNCVAL field value + 1) x 31 s.

   where GPIOi.GPIO_DEBOUNCINGTIME[7:0] field DEBOUNCVAL value is from 0 to 255.
3. The debouncing feature must be enabled in the debouncing enable register (write 1 to the corresponding bit of the GPIOi.GPIO_DEBOUNCENABLE register)

1.6 General-Purpose Interface Registers

This section summarizes the hardware interface for the GPIO product. Each module instance within the design is shown, together with the module register map and bit definitions for each bit field.

Table 1-6 shows the base address and address space for the GPIO module instances.

Table 1-6. Instance Summary

Module Name	Base Address	Size
GPIO1	0x4831 0000	4K bytes
GPIO2	0x4905 0000	4K bytes
GPIO3	0x4905 2000	4K bytes
GPIO4	0x4905 4000	4K bytes
GPIO5	0x4905 6000	4K bytes
GPIO6	0x4905 8000	4K bytes

1.6.1 General-Purpose Interface Register Mapping Summary

All module registers are 8-, 16-, or 32-bit accessible through the L4 interconnect (little endian encoding). Access to registers is direct; no shadow registers are implemented.

Table 1-7 through Table 1-8 describe the GPIO register offset addresses.

Table 1-7. GPIO1 to GPIO3 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (GPIO1)	Physical Address (GPIO2)	Physical Address (GPIO3)
GPIO_SYSCONFIG	RW	32	0x010	0x4831 0010	0x4905 0010	0x4905 2010
GPIO_SYSSTATUS	R	32	0x014	0x4831 0014	0x4905 0014	0x4905 2014
GPIO_IRQSTATUS1	RW	32	0x018	0x4831 0018	0x4905 0018	0x4905 2018
GPIO_IRQENABLE1	RW	32	0x01C	0x4831 001C	0x4905 001C	0x4905 201C
GPIO_WAKEUPENABLE	RW	32	0x020	0x4831 0020	0x4905 0020	0x4905 2020
GPIO_IRQSTATUS2	RW	32	0x028	0x4831 0028	0x4905 0028	0x4905 2028
GPIO_IRQENABLE2	RW	32	0x02C	0x4831 002C	0x4905 002C	0x4905 202C
GPIO_CTRL	RW	32	0x030	0x4831 0030	0x4905 0030	0x4905 2030
GPIO_OE	RW	32	0x034	0x4831 0034	0x4905 0034	0x4905 2034
GPIO_DATAIN	R	32	0x038	0x4831 0038	0x4905 0038	0x4905 2038
GPIO_DATAOUT	RW	32	0x03C	0x4831 003C	0x4905 003C	0x4905 203C
GPIO_LEVELDETECT0	RW	32	0x040	0x4831 0040	0x4905 0040	0x4905 2040
GPIO_LEVELDETECT1	RW	32	0x044	0x4831 0044	0x4905 0044	0x4905 2044
GPIO_RISINGDETECT	RW	32	0x048	0x4831 0048	0x4905 0048	0x4905 2048
GPIO_FALLINGDETECT	RW	32	0x04C	0x4831 004C	0x4905 004C	0x4905 204C
GPIO_DEBOUNCENABLE	RW	32	0x050	0x4831 0050	0x4905 0050	0x4905 2050
GPIO_DEBOUNCINGTIME	RW	32	0x054	0x4831 0054	0x4905 0054	0x4905 2054
GPIO_CLEARIRQENABLE1	RW	32	0x060	0x4831 0060	0x4905 0060	0x4905 2060
GPIO_SETIRQENABLE1	RW	32	0x064	0x4831 0064	0x4905 0064	0x4905 2064
GPIO_CLEARIRQENABLE2	RW	32	0x070	0x4831 0070	0x4905 0070	0x4905 2070
GPIO_SETIRQENABLE2	RW	32	0x074	0x4831 0074	0x4905 0074	0x4905 2074
GPIO_CLEARWKUENA	RW	32	0x080	0x4831 0080	0x4905 0080	0x4905 2080
GPIO_SETWKUENA	RW	32	0x084	0x4831 0084	0x4905 0084	0x4905 2084
GPIO_CLEARDATAOUT	RW	32	0x090	0x4831 0090	0x4905 0090	0x4905 2090
GPIO_SETDATAOUT	RW	32	0x094	0x4831 0094	0x4905 0094	0x4905 2094

Table 1-8. GPIO4 to GPIO6 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (GPIO4)	Physical Address (GPIO5)	Physical Address (GPIO6)
GPIO_SYSCONFIG	RW	32	0x010	0x4905 4010	0x4905 6010	0x4905 8010
GPIO_SYSSTATUS	R	32	0x014	0x4905 4014	0x4905 6014	0x4905 8014
GPIO_IRQSTATUS1	RW	32	0x018	0x4905 4018	0x4905 6018	0x4905 8018
GPIO_IRQENABLE1	RW	32	0x01C	0x4905 401C	0x4905 601C	0x4905 801C
GPIO_WAKEUPENABLE	RW	32	0x020	0x4905 4020	0x4905 6020	0x4905 8020
GPIO_IRQSTATUS2	RW	32	0x028	0x4905 4028	0x4905 6028	0x4905 8028
GPIO_IRQENABLE2	RW	32	0x02C	0x4905 402C	0x4905 602C	0x4905 802C
GPIO_CTRL	RW	32	0x030	0x4905 4030	0x4905 6030	0x4905 8030
GPIO_OE	RW	32	0x034	0x4905 4034	0x4905 6034	0x4905 8034
GPIO_DATAIN	R	32	0x038	0x4905 4038	0x4905 6038	0x4905 8038
GPIO_DATAOUT	RW	32	0x03C	0x4905 403C	0x4905 603C	0x4905 803C
GPIO_LEVELDETECT0	RW	32	0x040	0x4905 4040	0x4905 6040	0x4905 8040
GPIO_LEVELDETECT1	RW	32	0x044	0x4905 4044	0x4905 6044	0x4905 8044
GPIO_RISINGDETECT	RW	32	0x048	0x4905 4048	0x4905 6048	0x4905 8048
GPIO_FALLINGDETECT	RW	32	0x04C	0x4905 404C	0x4905 604C	0x4905 804C
GPIO_DEBOUNCENABLE	RW	32	0x050	0x4905 4050	0x4905 6050	0x4905 8050
GPIO_DEBOUNCINGTIME	RW	32	0x054	0x4905 4054	0x4905 6054	0x4905 8054
GPIO_CLEARIRQENABLE1	RW	32	0x060	0x4905 4060	0x4905 6060	0x4905 8060
GPIO_SETIRQENABLE1	RW	32	0x064	0x4905 4064	0x4905 6064	0x4905 8064
GPIO_CLEARIRQENABLE2	RW	32	0x070	0x4905 4070	0x4905 6070	0x4905 8070
GPIO_SETIRQENABLE2	RW	32	0x074	0x4905 4074	0x4905 6074	0x4905 8074
GPIO_CLEARWKUENA	RW	32	0x080	0x4905 4080	0x4905 6080	0x4905 8080
GPIO_SETWKUENA	RW	32	0x084	0x4905 4084	0x4905 6084	0x4905 8084
GPIO_CLEARDATAOUT	RW	32	0x090	0x4905 4090	0x4905 6090	0x4905 8090
GPIO_SETDATAOUT	RW	32	0x094	0x4905 4094	0x4905 6094	0x4905 8094

1.6.2 Register Descriptions

The write latency for all the R/W registers is immediate (with respect to the interface clock)

Note: When two write accesses in the GPIO_DEBOUNCINGTIME register are performed in less than two debounce clock cycles (32 kHz) + four interface clock cycles, the first write access latency is immediate, but the second write access is acknowledged only after this interval ends.

When one single register carries an individual configuration or setting for all the channels of the module, one bit in the register is dedicated to each channel. The bit and the corresponding channel are identified with the same number: bit 0 refers to channel 0, bit 1 refers to channel 1, and so on, up to 31.

Table 1-9 through Table 1-33 describe the register bits.

1.6.2.1 GPIO_SYSCONFIG

Table 1-9. GPIO_SYSCONFIG

Address Offset	0x010		
Physical Address	0x4831 0010	**Instance**	GPIO1
	0x4905 0010		GPIO2
	0x4905 2010		GPIO3
	0x4905 4010		GPIO4
	0x4905 6010		GPIO5
	0x4905 8010		GPIO6
Description	This register controls the various parameters of the L4 interconnect.		
Type	RW		

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
RESERVED																											IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0s for future compatibility. Read returns 0	RW	0x0000000
4:3	IDLEMODE	Power Management, Req/Ack control	RW	0x0
		0x0: Force-idle. An idle request is acknowledged unconditionally		
		0x1: No-idle. An idle request is never acknowledged		
		0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module		
		0x3: reserved do not use		
2	ENAWAKEUP	wake-up capability enabled/disabled	RW	0x0
		0x0: wake-up disable		
		0x1: wake-up enable		
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always returns 0.	RW	0x0
		0x0: Normal mode		
		0x1: The module is reset		
0	AUTOIDLE	Internal interface clock gating strategy	RW	0x0
		0x0: interface clock is free-running		
		0x1: Automatic interface clock gating strategy is applied, based on the L4 interconnect activity		

1.6.2.2 GPIO_SYSSTATUS

Table 1-10. GPIO_SYSSTATUS

Address Offset	0x014		
Physical Address	0x4831 0014	**Instance**	GPIO1
	0x4905 0014		GPIO2
	0x4905 2014		GPIO3
	0x4905 4014		GPIO4
	0x4905 6014		GPIO5
	0x4905 8014		GPIO6
Description	This register provides status information about the module, excluding the interrupt status information.		
Type	R		

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
RESERVED																								RESERVED							RESETDONE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0	R	0x000000
7:1	RESERVED	Read returns 0	R	0x00
0	RESETDONE	Internal reset monitoring	R	
		0x0: Internal module reset in on-going		
		0x1: Reset completed		

1.6.2.3 GPIO_IRQSTATUS1

Table 1-11. GPIO_IRQSTATUS1

Address Offset	0x018		
Physical Address	0x4831 0018	**Instance**	GPIO1
	0x4905 0018		GPIO2
	0x4905 2018		GPIO3
	0x4905 4018		GPIO4
	0x4905 6018		GPIO5
	0x4905 8018		GPIO6
Description	This register provides IRQ 1 status information.		
Type	RW		

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
IRQSTATUS1																															

Bits	Field Name	Description	Type	Reset
31:0	IRQSTATUS1	Interrupt 1 Status Register. Write a 1 in the corresponding bit to clear it to 0. Write 0 in the corresponding bit does not affect its value. 0x0: IRQ channel N not triggered 0x1: IRQ channel N triggered	RW	0x00000000

1.6.2.4 GPIO_IRQENABLE1

Table 1-12. GPIO_IRQENABLE1

Address Offset	0x01C		
Physical Address	0x4831 001C	**Instance**	GPIO1
	0x4905 001C		GPIO2
	0x4905 201C		GPIO3
	0x4905 401C		GPIO4
	0x4905 601C		GPIO5
	0x4905 801C		GPIO6
Description	This register provides IRQ 1 enable information.		
Type	RW		

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
IRQENABLE1																															

Bits	Field Name	Description	Type	Reset
31:0	IRQENABLE1	Interrupt 1 Enable Register 0x0: disable IRQ generation for channel N 0x1: enable IRQ generation for channel N	RW	0x00000000

1.6.2.5 GPIO_WAKEUPENABLE

Table 1-13. GPIO_WAKEUPENABLE

Address Offset	0x020		
Physical Address	0x4831 0020	**Instance**	GPIO1
	0x4905 0020		GPIO2
	0x4905 2020		GPIO3
	0x4905 4020		GPIO4
	0x4905 6020		GPIO5
	0x4905 8020		GPIO6
Description	This register provides wake-up enable information.		
Type	RW		

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
WAKEUPEN																															

Bits	Field Name	Description	Type	Reset
31:0	WAKEUPEN	Wake Up Enable Register	RW	0x00000000
		0x0: disable wake-up generation for channel N		
		0x1: enable wake-up generation for channel N		

1.6.2.6 GPIO_IRQSTATUS2

Table 1-14. GPIO_IRQSTATUS2

Address Offset	0x028		
Physical Address	0x4831 0028	**Instance**	GPIO1
	0x4905 0028		GPIO2
	0x4905 2028		GPIO3
	0x4905 4028		GPIO4
	0x4905 6028		GPIO5
	0x4905 8028		GPIO6
Description	This register provides IRQ 2 status information.		
Type	RW		

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
IRQSTATUS2																															

Bits	Field Name	Description	Type	Reset
31:0	IRQSTATUS2	Interrupt 2 Status Register. Write a 1 in the corresponding bit to clear it to 0. Write 0 in the corresponding bit does not affect its value. 0x0: IRQ channel N not triggered 0x1: IRQ channel N triggered	RW	0x00000000

1.6.2.7 GPIO_IRQENABLE2

Table 1-15. GPIO_IRQENABLE2

Address Offset	0x02C		
Physical Address	0x4831 002C	**Instance**	GPIO1
	0x4905 002C		GPIO2
	0x4905 202C		GPIO3
	0x4905 402C		GPIO4
	0x4905 602C		GPIO5
	0x4905 802C		GPIO6
Description	This register provides IRQ 2 enable information.		
Type	RW		

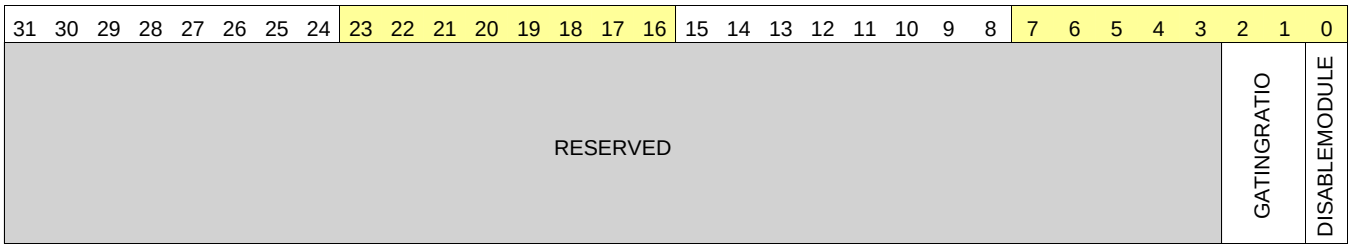
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQENABLE2																															

Bits	Field Name	Description	Type	Reset
31:0	IRQENABLE2	Interrupt 2 Enable Register 0x0: disable IRQ generation for channel N 0x1: enable IRQ generation for channel N	RW	0x00000000

1.6.2.8 GPIO_CTRL

Table 1-16. GPIO_CTRL

Address Offset	0x030		
Physical Address	0x4831 0030	**Instance**	GPIO1
	0x4905 0030		GPIO2
	0x4905 2030		GPIO3
	0x4905 4030		GPIO4
	0x4905 6030		GPIO5
	0x4905 8030		GPIO6
Description	This register controls the clock gating functionality.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Read returns 0	RW	0x00000000
2:1	GATINGRATIO	Gating Ratio	RW	0x1
		0x0: Functional clock is interface clock.		
		0x1: Functional clock is interface clock divided by 2.		
		0x2: Functional clock is interface clock divided by 4.		
		0x3: Functional clock is interface clock divided by 8.		
0	DISABLEMODULE	Module Disable	RW	0x0
		0x0: Module is enabled, clocks are not gated		
		0x1: Module is disabled, clocks are gated		

1.6.2.9 GPIO_OE

Table 1-17. GPIO_OE

Address Offset	0x034		
Physical Address	0x4831 0034	**Instance**	GPIO1
	0x4905 0034		GPIO2
	0x4905 2034		GPIO3
	0x4905 4034		GPIO4
	0x4905 6034		GPIO5
	0x4905 8034		GPIO6
Description	This register is used to enable the pins output capabilities. Its only function is to carry the pads configuration.		
Type	RW		

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																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUTEN	Output Data Enable	RW	0xFFFFFFFF
		0x0: The corresponding GPIO port is configured as output		
		0x1: The corresponding GPIO port is configured as input		

1.6.2.10 GPIO_DATAIN

Table 1-18. GPIO_DATAIN

Address Offset	0x038		
Physical Address	0x4831 0038	**Instance**	GPIO1
	0x4905 0038		GPIO2
	0x4905 2038		GPIO3
	0x4905 4038		GPIO4
	0x4905 6038		GPIO5
	0x4905 8038		GPIO6
Description	This register is used to register the data that is read from the GPIO pins.		
Type	R		

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
DATAINPUT																															

Bits	Field Name	Description	Type	Reset
31:0	DATAINPUT	Sampled Input Data	R	0x00000000

1.6.2.11 GPIO_DATAOUT

Table 1-19. GPIO_DATAOUT

Address Offset	0x03C		
Physical Address	0x4831 003C	**Instance**	GPIO1
	0x4905 003C		GPIO2
	0x4905 203C		GPIO3
	0x4905 403C		GPIO4
	0x4905 603C		GPIO5
	0x4905 803C		GPIO6
Description	This register is used for setting the value of the GPIO output pins		
Type	RW		

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
DATAOUTPUT																															

Bits	Field Name	Description	Type	Reset
31:0	DATAOUTPUT	Output Data	RW	0x00000000

1.6.2.12 GPIO_LEVELDETECT0

Table 1-20. GPIO_LEVELDETECT0

Address Offset	0x040		
Physical Address	0x4831 0040	**Instance**	GPIO1
	0x4905 0040		GPIO2
	0x4905 2040		GPIO3
	0x4905 4040		GPIO4
	0x4905 6040		GPIO5
	0x4905 8040		GPIO6
Description	This register is used to enable/disable for each input lines the low-level (0) detection to be used for the interrupt request generation.		
Type	RW		

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
LOWLEVEL																															

Bits	Field Name	Description	Type	Reset
31:0	LOWLEVEL	Low Level Interrupt Enable	RW	0x00000000
		0x0: disable the IRQ assertion on low level detect		
		0x1: enable the IRQ assertion on low level detect		

1.6.2.13 GPIO_LEVELDETECT1

Table 1-21. GPIO_LEVELDETECT1

Address Offset	0x044		
Physical Address	0x4831 0044	**Instance**	GPIO1
	0x4905 0044		GPIO2
	0x4905 2044		GPIO3
	0x4905 4044		GPIO4
	0x4905 6044		GPIO5
	0x4905 8044		GPIO6
Description	This register is used to enable/disable for each input lines the high-level (1) detection to be used for the interrupt request generation.		
Type	RW		

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
HIGHLEVEL																															

Bits	Field Name	Description	Type	Reset
31:0	HIGHLEVEL	High Level Interrupt Enable	RW	0x00000000
		0x0: disable the IRQ assertion on high level detect		
		0x1: enable the IRQ assertion on high level detect		

1.6.2.14 GPIO_RISINGDETECT

Table 1-22. GPIO_RISINGDETECT

Address Offset	0x048		
Physical Address	0x4831 0048	**Instance**	GPIO1
	0x4905 0048		GPIO2
	0x4905 2048		GPIO3
	0x4905 4048		GPIO4
	0x4905 6048		GPIO5
	0x4905 8048		GPIO6
Description	This register is used to enable/disable for each input lines the rising-edge (transition 0=>1) detection to be used for the interrupt request and the wake-up generation.		
Type	RW		

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
RISINGEDGE																															

Bits	Field Name	Description	Type	Reset
31:0	RISINGEDGE	Rising Edge Interrupt/Wake-up Enable	RW	0x00000000
		0x0: disable IRQ/Wake-up on rising edge detect		
		0x1: enable IRQ/Wake-up on rising edge detect		

1.6.2.15 GPIO_FALLINGDETECT

Table 1-23. GPIO_FALLINGDETECT

Address Offset	0x04C		
Physical Address	0x4831 004C	**Instance**	GPIO1
	0x4905 004C		GPIO2
	0x4905 204C		GPIO3
	0x4905 404C		GPIO4
	0x4905 604C		GPIO5
	0x4905 804C		GPIO6
Description	This register is used to enable/disable for each input lines the falling-edge (transition 1=>0) detection to be used for the interrupt request and the wake-up generation.		
Type	RW		

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
FALLINGEDGE																															

Bits	Field Name	Description	Type	Reset
31:0	FALLINGEDGE	Falling Edge Interrupt/Wake-up Enable	RW	0x00000000
		0x0: disable IRQ/Wake-up on falling edge detect		
		0x1: enable IRQ/Wake-up on falling edge detect		

1.6.2.16 GPIO_DEBOUNCENABLE

Table 1-24. GPIO_DEBOUNCENABLE

Address Offset	0x050		
Physical Address	0x4831 0050	**Instance**	GPIO1
	0x4905 0050		GPIO2
	0x4905 2050		GPIO3
	0x4905 4050		GPIO4
	0x4905 6050		GPIO5
	0x4905 8050		GPIO6
Description	This register is used to enable/disable the debouncing feature for each input line.		
Type	RW		

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
DEBOUNCEEN																															

Bits	Field Name	Description	Type	Reset
31:0	DEBOUNCEEN	Input Debounce Enable	RW	0x00000000
		0x0: disable debouncing feature on the corresponding input port		
		0x1: enable debouncing feature on the corresponding input port		

1.6.2.17 GPIO_DEBOUNCINGTIME

Table 1-25. GPIO_DEBOUNCINGTIME

Address Offset	0x054		
Physical Address	0x4831 0054	**Instance**	GPIO1
	0x4905 0054		GPIO2
	0x4905 2054		GPIO3
	0x4905 4054		GPIO4
	0x4905 6054		GPIO5
	0x4905 8054		GPIO6
Description	This register controls debouncing time (the value is global for all ports).		
Type	RW		

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
RESERVED																								DEBOUNCEVAL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0	RW	0x000000
7:0	DEBOUNCEVAL	Input Debouncing Value in 31 microsecond steps. debouncing time = (DEBOUNCEVAL+1) x 31 s	RW	0x00

1.6.2.18 GPIO_CLEARIRQENABLE1

Table 1-26. GPIO_CLEARIRQENABLE1

Address Offset	0x060		
Physical Address	0x4831 0060	**Instance**	GPIO1
	0x4905 0060		GPIO2
	0x4905 2060		GPIO3
	0x4905 4060		GPIO4
	0x4905 6060		GPIO5
	0x4905 8060		GPIO6
Description	Clear to 0 the corresponding bits in the GPIO_IRQENABLE1 register		
Type	RW		

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
CLEARIRQEN1																															

Bits	Field Name	Description	Type	Reset
31:0	CLEARIRQEN1	Clear Interrupt Enable 1 0x0: no effect 0x1: Clear the corresponding bit in the GPIO_IRQENABLE1 register	RW	0x00000000

1.6.2.19 GPIO_SETIRQENABLE1

Table 1-27. GPIO_SETIRQENABLE1

Address Offset	0x064		
Physical Address	0x4831 0064	**Instance**	GPIO1
	0x4905 0064		GPIO2
	0x4905 2064		GPIO3
	0x4905 4064		GPIO4
	0x4905 6064		GPIO5
	0x4905 8064		GPIO6
Description	Set to 1 the corresponding bits in the GPIO_IRQENABLE1 register		
Type	RW		

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
SETIRQEN1																															

Bits	Field Name	Description	Type	Reset
31:0	SETIRQEN1	Set Interrupt Enable 1 0x0: no effect 0x1: Set the corresponding bit in the GPIO_IRQENABLE1 register	RW	0x00000000

1.6.2.20 GPIO_CLEARIRQENABLE2

Table 1-28. GPIO_CLEARIRQENABLE2

Address Offset	0x070		
Physical Address	0x4831 0070	**Instance**	GPIO1
	0x4905 0070		GPIO2
	0x4905 2070		GPIO3
	0x4905 4070		GPIO4
	0x4905 6070		GPIO5
	0x4905 8070		GPIO6
Description	Clear to 0 the corresponding bits in the GPIO_IRQENABLE2 register		
Type	RW		

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
CLEARIRQEN2																															

Bits	Field Name	Description	Type	Reset
31:0	CLEARIRQEN2	Clear Interrupt Enable 2	RW	0x00000000
		0x0: no effect		
		0x1: Clear the corresponding bit in the GPIO_IRQENABLE2 register		

1.6.2.21 GPIO_SETIRQENABLE2

Table 1-29. GPIO_SETIRQENABLE2

Address Offset	0x074		
Physical Address	0x4831 0074	**Instance**	GPIO1
	0x4905 0074		GPIO2
	0x4905 2074		GPIO3
	0x4905 4074		GPIO4
	0x4905 6074		GPIO5
	0x4905 8074		GPIO6
Description	Set to 1 the corresponding bits in the GPIO_IRQENABLE2 register		
Type	RW		

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
SETIRQEN2																															

Bits	Field Name	Description	Type	Reset
31:0	SETIRQEN2	Set Interrupt Enable 2	RW	0x00000000
		0x0: no effect		
		0x1: Set the corresponding bit in the GPIO_IRQENABLE2 register		

1.6.2.22 GPIO_CLEARWKUENA

Table 1-30. GPIO_CLEARWKUENA

Address Offset	0x080		
Physical Address	0x4831 0080	**Instance**	GPIO1
	0x4905 0080		GPIO2
	0x4905 2080		GPIO3
	0x4905 4080		GPIO4
	0x4905 6080		GPIO5
	0x4905 8080		GPIO6
Description	Clear to 0 the corresponding bits in the GPIO_WAKEUPENABLE register		
Type	RW		

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
CLEARWAKEUPEN																															

Bits	Field Name	Description	Type	Reset
31:0	CLEARWAKEUPEN	Clear Wake-up Enable 0x0: no effect 0x1: Clear the corresponding bit in the GPIO_WAKEUPENABLE register	RW	0x00000000

1.6.2.23 GPIO_SETWKUENA

Table 1-31. GPIO_SETWKUENA

Address Offset	0x084		
Physical Address	0x4831 0084	**Instance**	GPIO1
	0x4905 0084		GPIO2
	0x4905 2084		GPIO3
	0x4905 4084		GPIO4
	0x4905 6084		GPIO5
	0x4905 8084		GPIO6
Description	Set to 1 the corresponding bits in the GPIO_WAKEUPENABLE register		
Type	RW		

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
SETWAKEUPEN																															

Bits	Field Name	Description	Type	Reset
31:0	SETWAKEUPEN	Set Wake-up Enable 0x0: no effect 0x1: Set the corresponding bit in the GPIO_WAKEUPENABLE register	RW	0x00000000

1.6.2.24 GPIO_CLEARDATAOUT

Table 1-32. GPIO_CLEARDATAOUT

Address Offset	0x090		
Physical Address	0x4831 0090	**Instance**	GPIO1
	0x4905 0090		GPIO2
	0x4905 2090		GPIO3
	0x4905 4090		GPIO4
	0x4905 6090		GPIO5
	0x4905 8090		GPIO6
Description	Clear to 0 the corresponding bits in the GPIO_DATAOUT register		
Type	RW		

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
CLEARDATAOUT																															

Bits	Field Name	Description	Type	Reset
31:0	CLEARDATAOUT	Clear Data Output Register	RW	0x00000000
		0x0: no effect		
		0x1: Clear the corresponding bit in the GPIO_DATAOUT register		

1.6.2.25 GPIO_SETDATAOUT

Table 1-33. GPIO_SETDATAOUT

Address Offset	0x094		
Physical Address	0x4831 0094	**Instance**	GPIO1
	0x4905 0094		GPIO2
	0x4905 2094		GPIO3
	0x4905 4094		GPIO4
	0x4905 6094		GPIO5
	0x4905 8094		GPIO6
Description	Set to 1 the corresponding bits in the GPIO_DATAOUT register		
Type	RW		

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
SETDATAOUT																															

Bits	Field Name	Description	Type	Reset
31:0	SETDATAOUT	Set Data Output Register	RW	0x00000000
		0x0: no effect		
		0x1: Set the corresponding bit in the GPIO_DATAOUT register		

1.7 Revision History

Table 1-34 lists the changes made since the previous version of this document.

Table 1-34. Document Revision History

Reference	Additions/Modifications/Deletions
Section 1.3.1.2.1.1	Changed second note in section
Table 1-5	Changed GPIO4 Module[4:3] field.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265