OMAP5912 Multimedia Processor General-Purpose Interface Reference Guide

Literature Number: SPRU767A
March 2004

OMAP™
TEXAS INSTRUMENTS TECHNOLOGY

TEXAS INSTRUMENTS

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 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.

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.

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
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Preface

Read This First

About This Manual

This document describes the general-purpose interface of the OMAP5912 multimedia processor.

Notational Conventions

This document uses the following conventions.

- ❑ Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5912 Multimedia Processor Device Overview and Architecture Reference Guide (literature number SPRU748) introduces the setup, components, and features of the OMAP5912 multimedia processor and provides a high-level view of the device architecture.

OMAP5912 Multimedia Processor OMAP 3.2 Subsystem Reference Guide (literature number SPRU749) introduces and briefly defines the main features of the OMAP3.2 subsystem of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor DSP Sybsystem Reference Guide (literature number SPRU750) describes the OMAP5912 multimedia processor DSP subsystem. The digital signal processor (DSP) subsystem is built around a core processor and peripherals that interface with: 1) The ARM926EJS via the microprocessor unit interface (MPUI); 2) Various standard memories via the external memory interface (EMIF); 3) Various system peripherals via the TI peripheral bus (TIPB) bridge.

OMAP5912 Multimedia Processor Clocks Reference Guide (literature number SPRU751) describes the clocking mechanisms of the OMAP5912 multimedia processor. In OMAP5912, various clocks are created from special components such as the digital phase locked loop (DPLL) and the analog phase-locked loop (APLL).

OMAP5912 Multimedia Processor Initialization Reference Guide (literature number SPRU752) describes the reset architecture, the configuration, the initialization, and the boot ROM of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Power Management Reference Guide (literature number SPRU753) describes power management in the OMAP5912 multimedia processor. The ultralow-power device (ULPD) generates and manages clocks and reset signals to OMAP3.2 and to some peripherals. It controls chip-level power-down modes and handles chip-level wake-up events. In deep sleep mode, this module is still active to monitor wake-up events.This book describes the ULPD module and outline architecture.

OMAP5912 Multimedia Processor Security Features Reference Guide (literature number SPRU754) describes the security features of teh OMAP5912 multimedia processor. The OMAP5912 security scheme relies on the OMAP3.2 secure mode. The distributed security on the OMAP3.2 platform is a Texas Instruments solution to address m-commerce and security issues within a mobile phone environment. The OMAP3.2 secure mode was developed to bring hardware robustness to the overall OMAP5912 security scheme.

OMAP5912 Multimedia Processor Direct Memory Access (DMA) Support Reference Guide (literature number SPRU755) describes the direct memory access support of the OMAP5912 multimedia processor. The OMAP5912 processor has three DMAs:

- The system DMA is embedded in OMAP3.2. It handles DMA transfers associated with MPU and shared peripherals.
- The DSP DMA is embedded in OMAP3.2. It handles DMA transfers associated with DSP peripherals.
- The generic distributed DMA (GDD) is an OMAP5912 resource attached to the SSI peripheral. It handles only DMA transfers associated with the SSI peripheral.

OMAP5912 Multimedia Processor Memory Interfaces Reference Guide (literature number SPRU756) describes the memory interfaces of the OMAP5912 multimedia processor.

- SDRAM (external memory interface fast, or EMIFF)
- Asynchronous and synchronous burst memory (external memory interface slow, or EMIFS)
- NAND flash (hardware controller or software controller)
- CompactFlash on EMIFS interface
- Internal static RAM

OMAP5912 Multimedia Processor Interrupts Reference Guide (literature number SPRU757) describes the interrupts of the OMAP5912 multimedia processor. Three level 2 interrupt controllers are used in OMAP5912:

- One MPU level 2 interrupt handler (also referred to as MPU interrupt level 2) is implemented outside of OMAP3.2 and can handle 128 interrupts.
- One DSP level 2 interrupt handler (also referred to as DSP interrupt level 2.1) is instantiated outside of OMAP3.2 and can handle 64 interrupts.
- One OMAP3.2 DSP level 2 interrupt handler (referenced as DSP interrupt level 2.0) can handle 16 interrupts.

OMAP5912 Multimedia Processor Peripheral Interconnects Reference Guide (literature number SPRU758) describes various periperal interconnects of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Timers Reference Guide (literature number SPRU759) describes various timers of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Serial Interfaces Reference Guide (literature number SPRU760) describes the serial interfaces of the OMAP5912 multimedia processor.

OMAP5912 Multimedia Processor Universal Serial Bus (USB) Reference Guide (literature number SPRU761) describes the universal serial bus (USB) host on the OMAP5912 multimedia processor. The OMAP5912 processor provides several varieties of USB functionality. Flexible multiplexing of signals from the OMAP5912 USB host controller, the OMAP5912 USB function controller, and other OMAP5912 peripherals allow a wide variety of system-level USB capabilities. Many of the OMAP5912 pins can be used for USB-related signals or for signals from other OMAP5912 peripherals. The OMAP5912 top-level pin multiplexing controls each pin individually to select one of several possible internal pin signal interconnections. When these shared pins are programmed for use as USB signals, the OMAP5912 USB signal multiplexing selects how the signals associated with the three OMAP5912 USB host ports and the OMAP5912 USB function controller can be brought out to OMAP5912 pins.

OMAP5912 Multimedia Processor Multi-channel Buffered Serial Ports (McBSPs) Reference Guide (literature number SPRU762) describes the three multi-channel buffered serial ports (McBSPs) available on the OMAP5912 device. The OMAP5912 device provides multiple high-speed multichannel buffered serial ports (McBSPs) that allow direct interface to codecs and other devices in a system.

OMAP5912 Multimedia Processor Camera Interface Reference Guide (literature number SPRU763) describes two camera inerfaces implemented in the OMAP5912 multimedia processor: compact serial camera port and camera parallel interface.

OMAP5912 Multimedia Processor Display Interface Reference Guide (literature number SPRU764) describes the display interface of the OMAP5912 multimedia processor.

- LCD module
- LCD data conversion module
- LED pulse generator
- Display interface

OMAP5912 Multimedia Processor Multimedia Card (MMC/SD/SDIO) (literature number SPRU765) describes the multimedia card (MMC) interface of the OMAP5912 multimedia processor. The multimedia card/secure data/secure digital IO (MMC/SD/SDIO) host controller provides an interface between a local host, such as a microprocessor unit (MPU) or digital signal processor (DSP), and either an MMC or SD memory card, plus up to four serial flash cards. The host controller handles MMC/SD/SDIO or serial port interface (SPI) transactions with minimal local host intervention.

OMAP5912 Multimedia Processor Keyboard Interface Reference Guide (literature number SPRU766) describes the keyboard interface of the OMAP5912 multimedia processor. The MPUIO module enables direct I/O communication between the MPU (through the public TIPB) and external devices. Two types of I/O can be used: specific I/Os dedicated for 8 x 8 keyboard connection, and general-purpose I/Os.

OMAP5912 Multimedia Processor General-Purpose Interface Reference Guide (literature number SPRU767) describes the general-purpose interface of the OMAP5912 multimedia processor. There are four GPIO modules in the OMAP5912. Each GPIO peripheral controls 16 dedicated pins configurable either as input or output for general purposes. Each pin has an independent control direction set by a programmable register. The two−edge control registers configure events (rising edge, falling edge, or both edges) on an input pin to trigger interrupts or wake−up requests (depending on the system mode). In addition, an interrupt mask register masks out specified pins. Finally, the GPIO peripherals provide the set and clear capabilities on the data output registers and the interrupt mask registers. After detection, all event sources are merged and a single synchronous interrupt (per module) is generated in active mode, whereas a unique wake−up line is issued in idle mode. Eight data output lines of the GPIO3 are ORed together to generate a global output line at the OMAP5912 boundary. This global output line can be used in conjunction with the SSI to provide a CMT−APE interface to the OMAP5912.

OMAP5912 Multimedia Processor VLYNQ Serial Communications Interface Reference Guide (literature number SPRU768) describes the VLYNQ of the OMAP5912 multimedia processor.

VLYNQ is a serial communications interface that enables the extension of an internal bus segment to one or more external physical devices. The external devices are mapped into local, physical address space and appear as if they are on the internal bus of the OMAP 5912. The external devices must also have a VLYNQ interface. The VLYNQ module serializes bus transactions in one device, transfers the serialized data between devices via a VLYNQ port, and de-serializes the transaction in the external device.

OMAP5912 includes one VLYNQ module connected on OCPT2 target port and OCPI initiator port. These connections are configured via a static switch, which selects either SSI or VLYNQ module. This switch, forbids the simultaneous use of GDD/SSI and VLYNQ. The switch is controlled by the VLYNQ_EN bit in the OMAP5912 configuration control register (CONF_5912_CTRL).

OMAP5912 Multimedia Processor Pinout Reference Guide (literature number SPRU769) provides the pinout of the OMAP5912 multimedia processor. After power-up reset, the user can change the configuration of the default interfaces. If another interface is available on top of the default, it is possible to enable a new interface for each ball by setting the corresponding 3-bit field of the associated FUNC_MUX_CTRL register. It is also possible to configure on-chip pullup/pulldown. This document also describes the various power domains so that the user can apply the different interfaces seamlessly with external components.

OMAP5912 Multimedia Processor Window Tracer (WT) Reference Guide (literature number SPRU770) describes the window tracer module used to capture the memory transactions from four interfaces: EMIFF, EMIFS, OCP-T1, and OCP-T2. This module is located in the OMAP3.2 traffic controller (TC).

OMAP5912 Multimedia Processor Real-Time Clock Reference Guide (literature number SPRUxxx) describes the real-time clock of the OMAP5912 multimedia processor. The real-time clock (RTC) block is an embedded real-time clock module directly accessible from the TIPB bus interface.

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

General-Purpose Interface

This document describes the general-purpose interface of the OMAP5912 multimedia processor.

1 GPIO Peripheral

The general-purpose input/output (GPIO) peripheral can be used for the following types of applications:

❑ Input/output data

❑ Generation of an interrupt in active mode upon the detection of external events

❑ Generation of a wake-up request in idle mode upon the detection of external events

There are four GPIO modules in the OMAP5912 (see Figure 1).

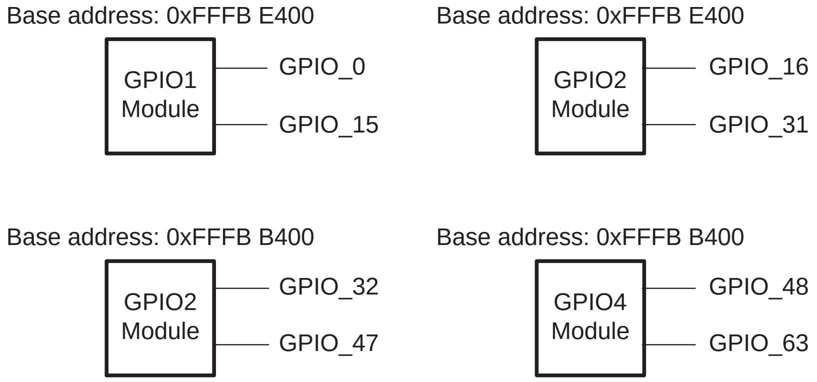
Each GPIO peripheral controls 16 dedicated pins configurable either as input or output for general purposes. Each pin has an independent control direction set by a programmable register. The two−edge control registers configure events (rising edge, falling edge, or both edges) on an input pin to trigger interrupts or wake−up requests (depending on the system mode). In addition, an interrupt mask register masks out specified pins. Finally, the GPIO peripherals provide the set and clear capabilities on the data output registers and the interrupt mask registers. After detection, all event sources are merged and a single synchronous interrupt (per module) is generated in active mode, whereas a unique wake−up line is issued in idle mode. Eight data output lines of the GPIO3 are ORed together to generate a global output line at the OMAP5912 boundary. This global output line can be used in conjunction with the SSI to provide a CMT−APE interface to the OMAP5912.

All wake−up events of each GPIO module are merged and connected to the MPU interrupt handler to enable the system wake−up. Then, the MPU is in charge.

From one event detection, the GPIO is able to generate two distinct interrupts with independent status and mask registers. One of these interrupts is mapped on the DSP interrupt handler and the other on the MPU interrupt handler.

Some GPIO lines are instanciated more than once at the OMAP5912 level to offer greater flexibility. You must ensure that not more than one GPIO line is enabled at any given time.

Figure 1. GPIO Modules



1.1 Functional Description

Each GPIO pin has an independent control direction set by a programmable register that enables the 16 GPIO pins to be configured individually either as input or output.

The GPIO has two registers to control the expected transition (rising and/or falling edge) on the input line which activates the interrupt line(s). Through these registers, the user can also disable the edge detection individually for each input GPIO.

This peripheral also provides an interrupt enable register for each synchronous interrupt line to enable or disable the source of interrupt (input GPIO). All peripheral registers are 16- or 32-bit accessible with an OCP interface compatible.

To avoid the atomic test and set usage for a register update, the peripheral offers the set and clear protocol register update.

In active mode, the edge detection is performed synchronously with the GPIO functional clock. The precision of the detection is then set by the frequency of this functional clock: the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the functional clock period. This minimum pulse width is required before and after any expected level transition detection.

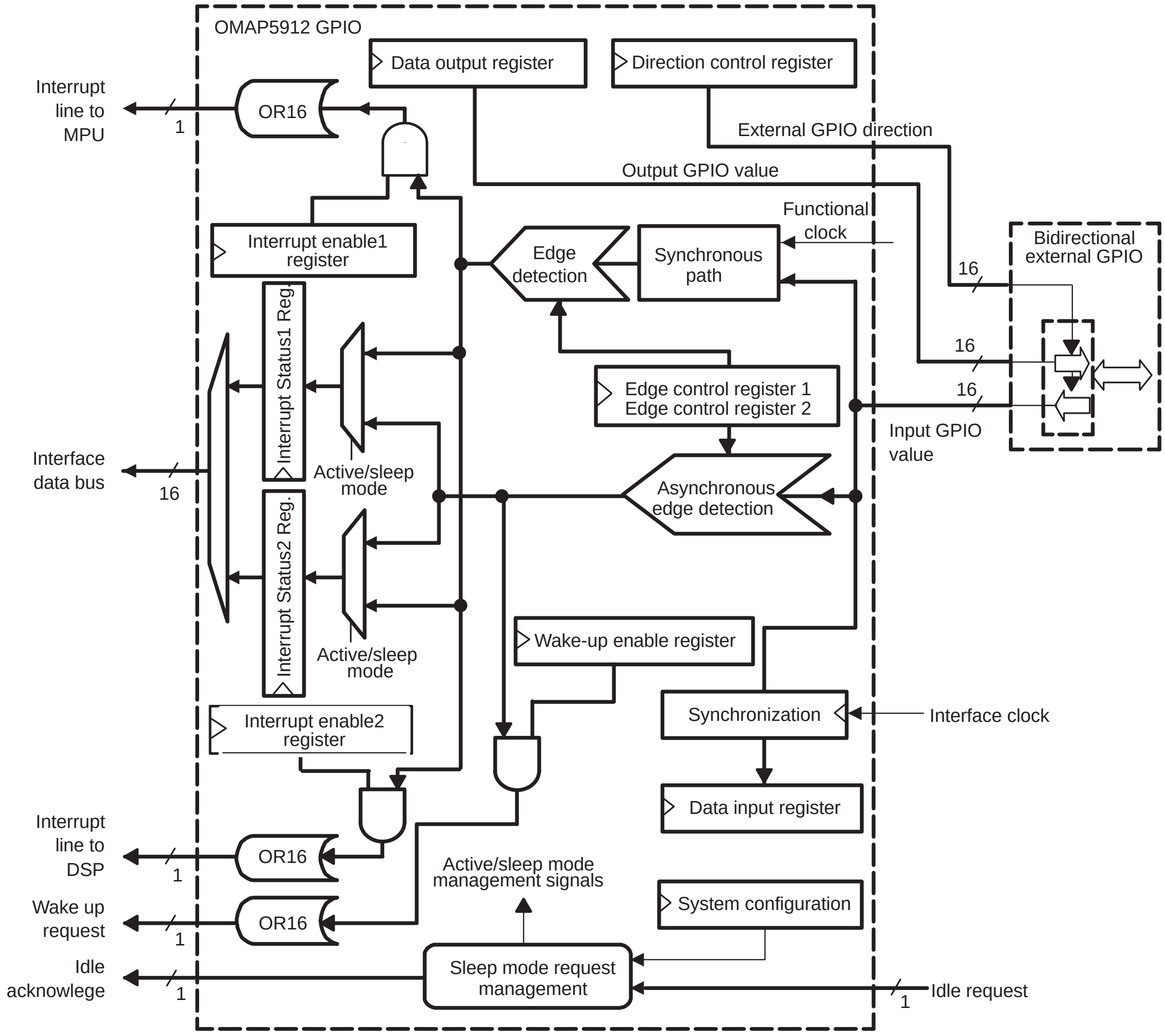
After the detection, all interrupt lines are merged together in two symmetrical paths to issue two synchronous interrupt request lines. Each interrupt line owns its dedicated interrupt status register to determine which source has activated the interrupt request. For each interrupt line, an interrupt enable register can mask the interrupt request activation without affecting the interrupt status register update in case of expected transition on one input GPIO pin.

In idle mode (the system does not provide any other functional or interface clocks), an asynchronous path is able to detect the expected event (a transition occurring on an enabled I/O configured as input) to generate an asynchronous wake-up request. A wake-up enable register enables or disables the source of the wake up. As in active mode, all wake-up sources (input GPIO) are merged together to issue one wake-up signal to the system.

Figure 2 details the GPIO block diagram with its configuration registers and main functional paths:

- The synchronous path to generate a synchronous interrupt request upon expected edge detection on any input GPIO; the synchronous interrupt request lines 1 and 2 are active according to their respective interrupt enable 1 and 2 registers (GPIO_IRQENABLE1, GPIO_IRQENABLE2).
- The asynchronous path to generate an asynchronous wake-up request upon expected edge detection on any input GPIO; the asynchronous wake-up request line is active according to the wake-up enable register (GPIO_WAKEUPENABLE).
- The logic block managing the sleep mode request/acknowledge protocol: this part enables the synchronous path in active mode and the asynchronous path in sleep mode.

Figure 2. GPIO Block Diagram



1.2 Set and Clear Instructions

To avoid using the atomic test-and-set to update the data output register (GPIO_DATAOUT) and the interrupt enable (GPIO_IRQENABLEx) or wake-up enable (GPIO_WAKEUPENABLE) registers, the GPIO peripheral offers a set-and-clear protocol. This protocol consists of writing an operation at dedicated addresses (one address for clearing bits and one address for setting bits). A written 1 clears (or sets) bits; unaffected bits are 0. See Section 1.3 for the clear instruction example and Section 1.4 for the set instruction example.

1.3 Clear Instruction Example

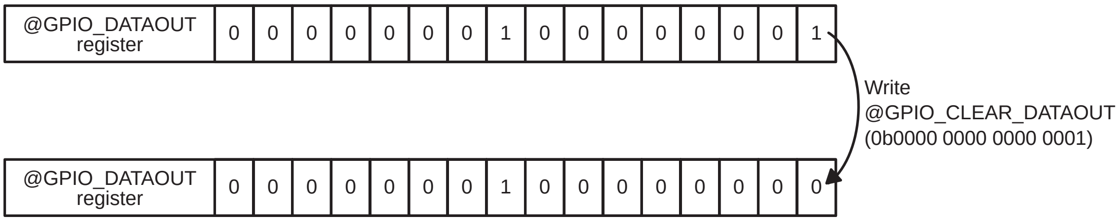
In this example, the data output register(GPIO_DATAOUT) contains the binary value 0b0000 0001 0000 0001, and the user wants to clear the bit [0].

With the clear instruction feature, the user writes 0b0000 0000 0000 0001 at the clear data output register address (GPIO_CLEAR_DATAOUT).

After this write operation, a reading of the data output register returns 0b0000 0001 0000 0000: the bit [0] has been cleared.

Similarly, the interrupt enable registers and the wakeup enable register can be cleared with GPIO_CLEAR_INTERRUPTx and GPIO_WAKEUPENABLE, respectively.

Figure 3. Write @GPIO_CLEAR_DATAOUT Register Example



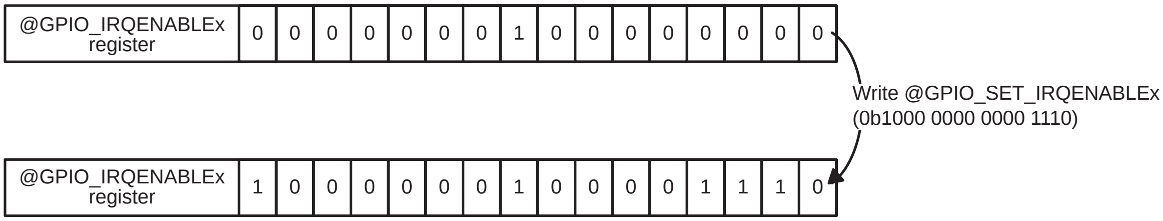
@GPIO_DATAOUT register	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
@GPIO_DATAOUT register	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

1.4 Set Instruction Example

In this example, the interrupt enable1 or 2 register contains the binary value 0b0000 0001 0000 0000, and the user wants to set the bits 15, 3, 2, and 1.

With the set instruction feature, the user writes 0b1000 0000 0000 1110 at the address of the set interrupt enable1 or 2 register (see Table 18). After this write operation, a reading of the interrupt enable1 or 2 register returns 0b1000 0001 0000 1110: the bits 15, 3, 2 , and 1 have been set.

Figure 4. Write @GPIO_SET_IRQENABLEx Register Example



@GPIO_IRQENABLEx register	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
@GPIO_IRQENABLEx register	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0

Similarly, the wakeup enable register and the data output register can be set using GPIO_SET_WAKEUP and GPIO_SET_DATAOUT, respectively.

1.5 GPIO Registers

Base address (in hexadecimal):

- Bit width: 32 bits
- Supported Accesses: 32-bit and 16-bit accesses
- Address of one register is: Base address + Offset address

Table 1 lists the 32-bit GPIO registers. Table 2 through Table 20 describe the register bits.

Table 1. GPIO Registers—General Description (1 and 2)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00		
Register	**Description**	**Offset**
GPIO_REVISION	Revision	0x0000
GPIO_SYSCONFIG	System configuration	0x0010
GPIO_SYSSTATUS	System status	0x0014
GPIO_IRQSTATUS1	Interrupt status1	0x0018
GPIO_IRQENABLE1	Interrupt enable1	0x001C
GPIO_IRQSTATUS2	Interrupt status2	0x0020
GPIO_IRQENABLE2	Interrupt enable2	0x0024
GPIO_WAKEUPENABLE	wake-up enable	0x0028
GPIO_DATAIN	Data input	0x002C
GPIO_DATAOUT	Data output	0x0030
GPIO_DIRECTION	Direction control	0x0034
GPIO_EDGE_CTRL1	Edge control 1	0x0038
GPIO_EDGE_CTRL2	Edge control 2	0x003C
GPIO_CLEAR_IRQENABLE1	Clear interrupt enable1	0x009C
GPIO_CLEAR_IRQENABLE2	Clear interrupt enable2	0x00A4
GPIO_CLEAR_WAKEUPENA	Clear wake-up enable	0x00A8
GPIO_CLEAR_DATAOUT	Clear data output	0x00B0
GPIO_SET_IRQENABLE1	Set interrupt enable1	0x00DC
GPIO_SET_IRQENABLE2	Set interrupt enable2	0x00E4

Table 1. GPIO Registers—General Description (1 and 2) (Continued)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00		
Register	**Description**	**Offset**
GPIO_SET_WAKEUPENA	Set wake-up enable	0x00E8
GPIO_SET_DATAOUT	Set data output	0x00F0

The write latency for the R/W registers is immediate, except for the direction control and the edge control 1 and 2 registers. Any write access in these three registers is resynchronized in the functional clock domain.

Table 2. Revision Register (GPIO_REVISION)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x00					
Bit	**Field Name**	**Function**		**Access (R/W)**	**Value at Reset**
31:8	Reserved	Reserved		R	0x000000
7:0	GPIO revision	Bits	RTL revision number	R	tbd
		[3:0]	Minor revision		
		[7:4]	Major revision		

This is a read-only register containing the revision number of the GPIO module. A write to this register has no affect, as the reset.

The GPIO revision 8-bit field indicates the revision number of the current module. This value is fixed by hardware.

The 4 LSBs indicate a minor revision.

The 4 MSBs indicate a major revision.

Example:

0x10 => version 1.0

This register sets various parameters that control the idle mode of the GPIO module.

The autoidle field (bit 0) sets the internal interface clock-gating strategy in active mode. The soft reset field (bit 1) resets the whole GPIO module (the same effect as the OCP hardware reset). The ENAWAKEUP field (bit 2) enables or disables the wake-up request generation upon the expected transition happening on the GPIO input pins. The idle mode field (bits [4:3]) controls the power-saving management.

Table 3. System Configuration Register (GPIO_SYSCONFIG)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x10					
Bit	**Field Name**	**Function**		**R/W**	**Reset**
31:5	Reserved	Reserved		R/W	0x000...0
4:3	IDLEMODE	Value	Power management, REQ/ACK control	R/W	00
		00	Force idle. An idle request is acknowledged unconditionally.		
		01	No idle. An idle request is never acknowledged.		
		10	Smart idle. The acknowledgement to an idle request is given based on the internal activity (see Section 1.8.2).		
		11	Reserved		
2	ENAWAKEUP	0: Wake-up generation is disabled. 1: Wake-up capability is enabled upon expected transition on input GPIO pin.		R/W	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0 : Normal mode 1 : The module is reset.		R/W	0
0	AUTOIDLE	0: Internal interface OCP clock is free−running 1: Automatic internal OCP clock gating, based on the OCP interface activity		R/W	0

Table 4. System Status Register (GPIO_SYSSTATUS)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x14					
Bit	**Field Name**	**Function**		**R/W**	**Reset**
31:1	Reserved	Reserved		R	0x000…0
0	RESETDONE	Value	Internal reset monitoring	R	−
		0	Internal reset is ongoing.		
		1	Reset completed		

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

Table 5. Interrupt Status Registers(1 and 2) (GPIO_IRQSTATUSx)

Base Address=0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address=0x18, 0x20			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No interrupt set 1: Interrupt set	R/W	0x0000

This register determines which of the input GPIO pins triggered the interrupt line 1 or 2 request, or the wake-up. As illustrated in Table 12, bit 0 corresponds to PIGPIOPINSI[1], and bit 1 to PIGPIOPINSI[2], and so on.

When a bit in this register is set to 1, it indicates that the corresponding GPIO pin is requesting the interrupt or wake up. If the user wants to reset a bit in this register, a 1 must be written to the appropriate bit. However, the user cannot generate an interrupt by writing a 1 to the interrupt status1 or 2 register. If the user writes a 0 to a bit in this register, the value remains unchanged. The interrupt status 1 or 2 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 status 1 and 2 registers is set when the GPIO peripheral is awakened.

Table 6. Interrupt Enable Registers (1 and 2)(GPIO_IRQENABLEx)

Base Address=0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address=0x1C, 0x24			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: Disabled 1: Enabled	R/W	0x0000

This register allows the user to mask the expected transition on input GPIO from generating an interrupt request on line1 or 2. The interrupt enable registers are programmed synchronously with the interface clock.

A feature enables the user to set or clear a bit of these registers with a single write access to the corresponding set interrupt enable 1 or 2 register, or to the clear interrupt enable 1 or 2 register address (see Table 15, and Table 18.)

Table 7. Wake-up Enable Register (GPIO_WAKEUPENABLE)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x28			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: Wake-up generation is disabled. 1: Wake-up generation is enabled.	R/W	0x0000

Note: In force idle mode, the module wake-up feature is inhibited.

This register allows the user to mask the expected transition on input GPIO from generating a wake-up request. It is programmed synchronously with the interface clock before any idle mode request comes from the host processor.

A feature enables the user to set or clear a bit 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 1.3 and Table 16 and Table 19)

Table 8. Data Input Register (GPIO_DATAIN)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x2C			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R	0x0000
15:0	Received data	R	0x0000

This register registers the data that is read from the GPIO pins. It is a read-only register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. Any change on the GPIO input pin is captured into this register after two interface clock cycles (seeing the required cycles to synchronize and write the data). See Figure 7. The data input register copies the PIGPIOPINSI[15:0], regardless of the GPIO_DIRECTION register.

Table 9. Data Output Register (GPIO_DATAOUT)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x30			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	Data to transmit	R/W	0x0000

This register sets the value to the GPIO output pins. Data is written to the register synchronously with the interface clock. It is possible to set or clear a bit of this register with a single write access to the set output data register, or to the clear output data register address (see Table 17 and Table 20.)

Table 10. Direction Control Register (GPIO_DIRECTION)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x34			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: Output 1: Input	R/W	0xFFFF

This register configures the GPIO pins for either input or output. At reset, all of the GPIO pins are configured as inputs. Data is written to this register synchronously with the interface clock.

Table 11. Edge Control Register 1 (GPIO_EDGE_CTRL1)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x38					
Bit	**Function**			**R/W**	**Reset**
15:0	bit[2n+1]	bit[2n]	Input GPIO[n+1] pins	R/W	0x0000
	0	0	No edge detection		
	0	1	Falling edge detection		
	1	0	Rising edge detection		
	1	1	Both edges detection		

Note: n ∈ [7:0]; GPIO pins 1 to 8.

This register allows the user to define, for each external GPIO[8:1] pin configured as input, the expected edge to trigger an interrupt or a wake-up request. The request can be generated either from a high-to-low transition (bits are 01), a low-to-high transition (bits are 10), or both transitions (bits are 11) occurring on an external GPIO pin configured as input. To disable the edge detection capability, the relevant bits corresponding to the GPIO must be reset (00).

As described in Table 12 , each edge control 1 register bit corresponds to a PIGPIOPINSI[8:1] pin.

Table 12. Edge Ctrl1 Register Bits Correspondences With PIGPIOPINSI[8:1]

EDGE CTRL 1 Register Bits and Corresponding PIGPIOPINSI[n]							
PIGPIOPINSI[4]		**PIGPIOPINSI[3]**		**PIGPIOPINSI[2]**		**PIGPIOPINSI[1]**	
bit[7]	bit[6]	bit[5]	bit[4]	bit[3]	bit[2]	bit[1]	bit[0]
PIGPIOPINSI[8]		**PIGPIOPINSI[7]**		**PIGPIOPINSI[6]**		**PIGPIOPINSI[5]**	
bit[15]	bit[14]	bit[13]	bit[12]	bit[11]	bit[10]	bit[9]	bit[8]

Table 13. Edge Control Register 2 (GPIO_EDGE_CTRL2)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0x3C					
Bit	**Function**			**R/W**	**Reset**
15:0	bit[2n+1]	bit[2n]	Input GPIO[n+9] pins	R/W	0x0000
	0	0	No edge detection		
	0	1	Falling edge detection		
	1	0	Rising edge detection		
	1	1	Both edges detection		

Note: n ∈ [7:0]; GPIO pins 9 to 16.

This register allows the user to define, for each external GPIO[16:9] pin configured as input, the expected edge to trigger an interrupt or a wake-up request. The request can be generated either from a high-to-low transition (bits are 01), a low-to-high transition (bits are 10), or both transitions (bits are 11) occurring on an external GPIO pin configured as input. To disable the edge detection capability, the relevant bits corresponding to the GPIO must be reset (00).

As described in Table 14, each edge control 2 register bit corresponds to a PIGPIOPINSI[16:9] pin.

Table 14. Edge Ctrl2 Register Bits Correspondences With PIGPIOPINSI[16:9]

EDGE CTRL 2 Register Bits and Corresponding PIGPIOPINSI[n]							
PIGPIOPINSI[12]		**PIGPIOPINSI[11]**		**PIGPIOPINSI[10]**		**PIGPIOPINSI[9]**	
bit[7]	bit[6]	bit[5]	bit[4]	bit[3]	bit[2]	bit[1]	bit[0]
PIGPIOPINSI[16]		**PIGPIOPINSI[15]**		**PIGPIOPINSI[14]**		**PIGPIOPINSI[13]**	
bit[15]	bit[14]	bit[13]	bit[12]	bit[11]	bit[10]	bit[9]	bit[8]

To avoid any inconsistency between the edge detection and the bit(s) in the interrupt status registers when the edge control registers are updated to change the expected transition, you must:

- Modify the edge detection register and then clear the interrupt status registers 1 and 2.

  Or

- Stop the edge detection by setting the relevant GPIO line as output, modify the edge detection register, and clear the interrupt status registers 1 and 2.

Table 15. Clear Interrupt Enable Registers (1 and 2) (GPIO_CLEAR_IRQENABLEx)

Base Address = 0xFFFB E400,0xFFFB EC00,0xFFFB B400,0xFFFB BC00, Offset Address = 0x9C,0xA4			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Clear the corresponding bit in the relevant interrupt enable register	R/W	0x0000

A write operation in this register clears the corresponding bit in the interrupt enable1, or 2, register when the written bit is 1; a written bit 0 has no effect (see Section 1.4).

A read of the clear interrupt enable1 or 2 register returns the value of the interrupt enable 1 or 2 register.

Table 16. Clear Wake-up Enable Register (GPIO_CLEAR_WAKEUPENA)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0xA8			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Clear the corresponding bit in the wake-up enable register	R/W	0x0000

A write operation in this 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 (see Section 1.4).

A read of the clear wake-up enable register returns the value of the wake-up enable register.

Table 17. Clear Data Output Register (GPIO_CLEAR_DATAOUT)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0xB0			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Clear the corresponding bit in the data output register	R/W	0x0000

A write operation in this register clears the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect (see Section 1.4).

A read of the clear data output register returns the value of the data output register.

Table 18. Set Interrupt Enable Registers (1 and 2)(GPIO_SET_IRQENABLEx)

Base Address =0xFFFB E400,0xFFFB EC00,0xFFFB B400,0xFFFB BC00, Offset Address = 0xDC,0xE4			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Set the corresponding bit in the relevant interrupt enable register	R/W	0x0000

A write operation in this register sets the corresponding bit in the interrupt enable1 or 2 register when the written bit is 1; a written bit at 0 has no effect (see Section 1.5).

A read of the set interrupt enable1 or 2 register returns the value of the interrrupt enable1 or 2 register.

Table 19. Set Wake-up Enable Register (GPIO_SET _WAKEUPENA)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0xE8			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Set the corresponding bit in the wake-up enable register	R/W	0x0000

A write operation in this 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 (see Section 1.5).

A read of the set wake-up enable register returns the value of the wake-up enable register.

Table 20. Set Data Output Register (GPIO_SET_DATAOUT)

Base Address = 0xFFFB E400, 0xFFFB EC00, 0xFFFB B400, 0xFFFB BC00, Offset Address = 0xF0			
Bit	**Function**	**R/W**	**Reset**
31:16	Reserved	R/W	0x0000
15:0	0: No effect 1: Set the corresponding bit in the data output register	R/W	0x0000

A write operation in this register sets the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect (see Section 1.5).

A read of the set data output register returns the value of the data output register.

1.6 Interrupt and Wake-up Features

In order to generate an interrupt or a wake-up request to a host processor upon a defined logic transition occurring on a GPIO pin, the GPIO configuration registers must be programmed as follows:

❏ The GPIO pin must be configured as an input in the direction register (see Table 10).

❏ The GPIO pin must be enabled in the interrupt enable 1 and/or 2 register, or in the wake-up enable register (see Table 6 and Table 7).

❏ The expected transition on input GPIO triggering the interrupt or the wake-up request must be set in the edge control 1 or 2 registers (see Table 11 and Table 13).

Because of the sample operation with the functional clock, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the functional clock period. This minimum pulse width must be met before and after any expected level transition detection.

All interrupt or wake-up sources (the 16 input GPIO pins) are merged (see Figure 5 and Figure 6) to issue two synchronous interrupt requests 1 and 2, and a single asynchronous wake-up request.

1.6.1 Synchronous Path: Interrupt Request Generation

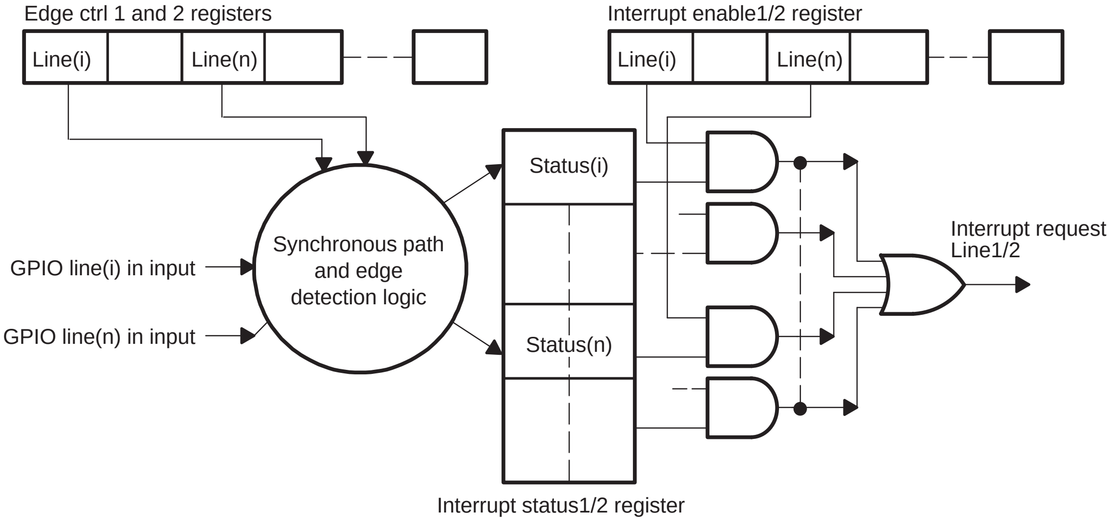
In active mode (functional and interface clocks are running), once the GPIO configuration registers have been set to enable the interrupt generation, a synchronous path samples the transitions on the input GPIO with the functional clock. Because of the sample operation, the minimum pulse width on the input GPIO is two times the functional clock period. When a transition matches with the edge control 1 and 2 registers programming (see Table 11 and Table 13), the corresponding bit in the interrupt status1 and 2 registers is set to 1, synchronously with the interface clock. On the following interface clock cycle, the interrupt lines 1 and/or 2 are active (low) (see Figure 7), depending on the interrupt enable1 and 2 registers.

The timing latency between the expected edge occurring at the input GPIO and the activation of the interrupt line(s) varies between four functional + five interface OCP clock cycles and nine functional + 11 interface OCP clock cycles. This timing latency depends on whether the ongoing edge detection process occurred during the processing (functional domain to interface OCP domain synchronization) of previous edge detection or not.

For any isolated expected edge detection, the interrupt line is active (low level) four functional + five interface OCP clock cycles after the transition occurred (see Figure 7).

In case of multiple and consecutive expected edge detections on different input GPIOs, the maximum timing latency to activate the interrupt line(s) is nine functional + 11 interface OCP clock cycles after the transition occurred (see Figure 8).

Figure 5. Interrupt Request Generation

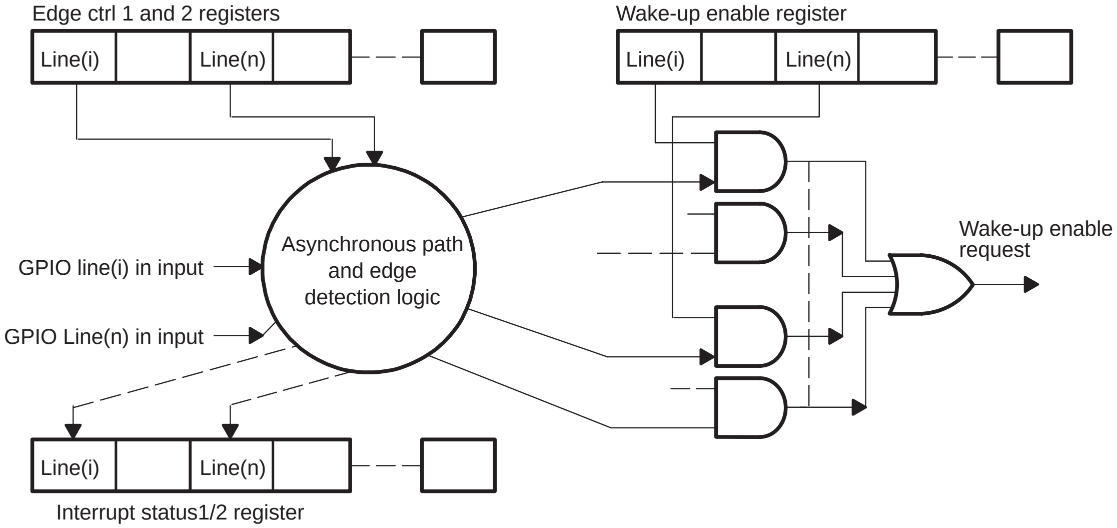

Edge ctrl 1 and 2 registers
Line(i)
Line(n)
Interrupt enable1/2 register
Line(i)
Line(n)
Status(i)
Synchronous path and edge detection logic
GPIO line(i) in input
GPIO line(n) in input
Status(n)
Interrupt request Line1/2
Interrupt status1/2 register

1.6.2 Asynchronous Path: Wake-up Request Generation

In idle mode (all clocks are shut down, the GPIO configuration registers have been previously programmed), an asynchronous path detects the expected transition on the input GPIO, according to edge control 1 and 2 register programming (see Table 11 and Table 13) and sends an asynchronous wake-up request if the wake-up enable register is set (the wake-up line is active high). As shown in Figure 6, there is only one external wake-up line, because the sources of all the wake-ups are merged together. Once the system is awakened, the interface clock is restarted and, according to the input GPIO pin that triggered the wake-up request, the corresponding bits in the interrupt status1 and 2 registers are synchronously set to 1. On the following interface clock cycle, the interrupt lines 1 and/or 2 are active (low) when the corresponding interrupt enable1 and 2 registers are set.

The EnaWakeUp bit of the GPIO_SYSCONFIG register (see Table 3) allows the GPIO wake-up feature to be enabled or disabled globally. If this bit is 0, the wake-up enable register has no effect.

Figure 6. Wake-up Request Generation



1.6.3 Interrupt (or Wake-up) Line Release

When the host processor receives an interrupt request issued by the GPIO peripheral, it can read the corresponding interrupt status register to find out which input GPIO triggered the interrupt or wake-up requeues. 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 interrupt status register (see Figure 7). If an interrupt request is stillpending (the interrupt status register bits are not cleared), the interrupt line is inactive (high) for two cycles before being reasserted (see Figure 8).

1.7 Timing Diagrams

In Figure 7 and Figure 8, the enable register is set (the I/O events are unmasked from triggering an interrupt request).

Figure 7. Timing Diagram of a Single Interrupt

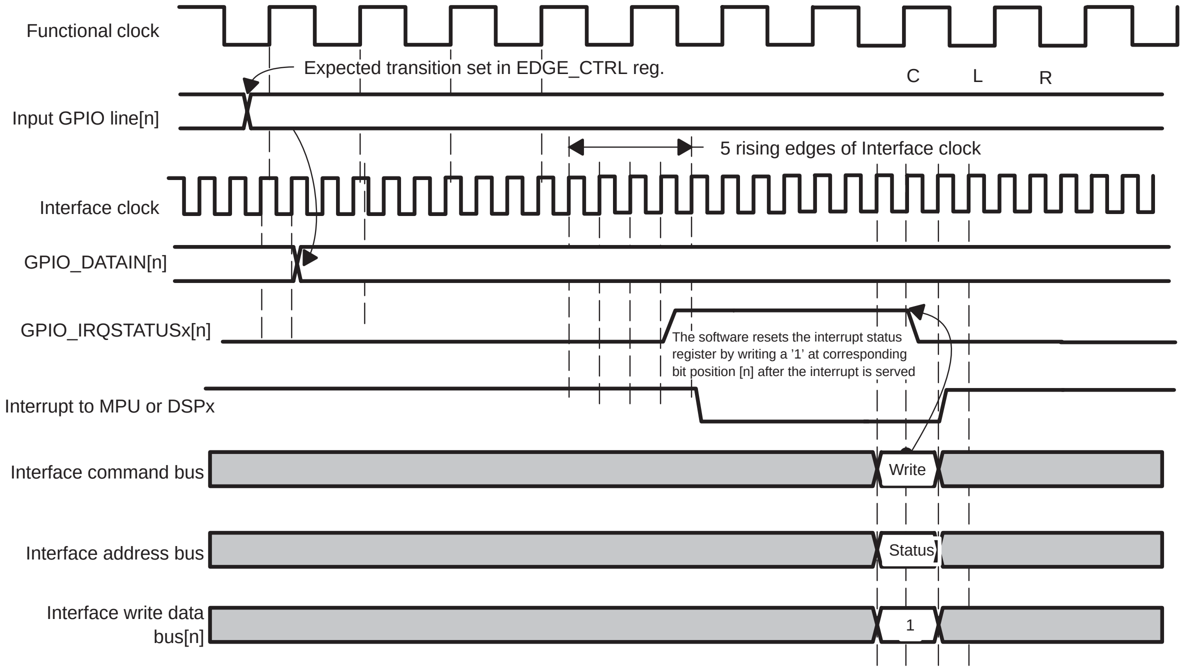
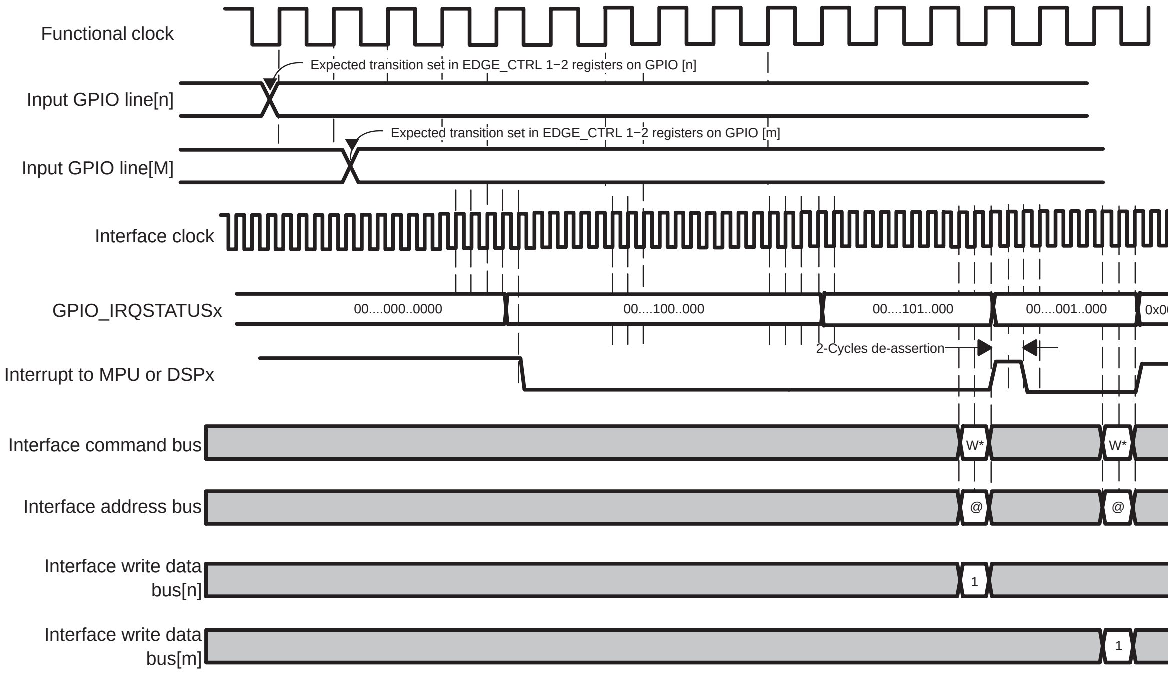

Functional clock
Expected transition set in EDGE_CTRL reg.
C
L
R
Input GPIO line[n]
5 rising edges of Interface clock
Interface clock
GPIO_DATAIN[n]
GPIO_IRQSTATUSx[n]
The software resets the interrupt status register by writing a '1' at corresponding bit position [n] after the interrupt is served
Interrupt to MPU or DSPx
Interface command bus
Write
Interface address bus
Status
Interface write data bus[n]
1

Figure 8. Timing Diagram of Two Consecutive Interrupts



* The software resets the interrupt status register by writing a 1 at the corresponding bit position [n])or[m]) after the interrupt is served.

1.8 Clocking and Reset Strategy

1.8.1 Clocks and Active Edge Definitions

The clock used to sample the GPIO input pins before writing in the data input register (GPIO_DATAIN) is the interface clock (using the rising edges).

On the synchronous path (active mode), the GPIO input transitions sample with the rising edges of the functional clock. Therefore, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the functional clock period.

On the asynchronous path (in sleep mode, it issues a wake-up request), there is no minimum input pulse width because there is no sampling operation.

The data load in the data output register is set at the output GPIO pins synchronously with the rising edge of the interface clock.

All GPIO registers are accessible synchronously with the interface clock.

1.8.2 Sleep Mode Request and Acknowledge

When the host processor issues a sleep mode request (the idle request signal is active), the GPIO module goes into sleep mode according to the idle mode field of the system configuration register.

If the idle mode field sets the no idle mode, the GPIO does not go into sleep mode and the idle acknowledge signal is not asserted.

If the idle mode field sets the force idle mode, the GPIO goes into sleep mode independently of the internal module state, and the idle acknowledge signal is unconditionally asserted. In the force idle mode, the module is inactive and its wake-up feature is inhibited.

If the idle mode field sets the smart idle mode, the GPIO module evaluates its internal capability to have the functional and interface clocks 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), the Idle acknowledge signal is asserted and the GPIO enters into sleep mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is sent only if the field ENAWAKEUP of the system configuration register enables the GPIO wake-up capability (see Table 3). When the system is awakened, the idle request signal goes inactive and the wake-up request signal is also deasserted (if it is the GPIO that triggered the system wake-up).

Once the GPIO acknowledges the sleep mode request (the idle acknowledge signal is active), the functional and interface clocks can be stopped at any time.

1.8.3 Reset

The OCP hardware reset signal has a global reset action on the GPIO. All configuration registers, all DFFs clocked with the functional clock and all internal state machine, are reset when the OCP hardware reset is active (low level). This hardware reset signal is synchronous to the OCP interface clock and internally resynchronized with the functional clock domain to ensure a correct reset of the DFFs using the functional clock.

The reset done field of the system status register (see Table 4) monitors the internal reset status. It is set when the reset is complete.

The software reset (soft reset field of the system configuration register—see Table 3), has the same effect as the OCP hardware reset signal, and the reset done field of the system status register is updated in the same condition.

2 Pulse-Width Tone

This pulse-width tone (PWT) module generates a modulated frequency signal for the external buzzer. The frequency is programmable between 322 Hz and 4868 Hz, with 12 half-tone frequencies per octave. The volume level is also programmable. All frequencies are generated from the PWT_CLK, which is a 12-MHz clock.

The PWT module creates the output tone signal for a buzzer. The frequency and the volume of this signal are programmable.

2.1 PWT Features

The PWT module has the following features (see Figure 9):

❑ Divider generating a 1500-kHz frequency clock

❑ TIPB control interface

❑ Four dividers with ${}^{101}/_{107}$, ${}^{49}/_{55}$, ${}^{50}/_{63,}$ and ${}^{80}/_{127}$ to generate each note particularity

❑ Four dividers 1/2 and a mux to select the octave

❑ 6-bit register to control tone frequency

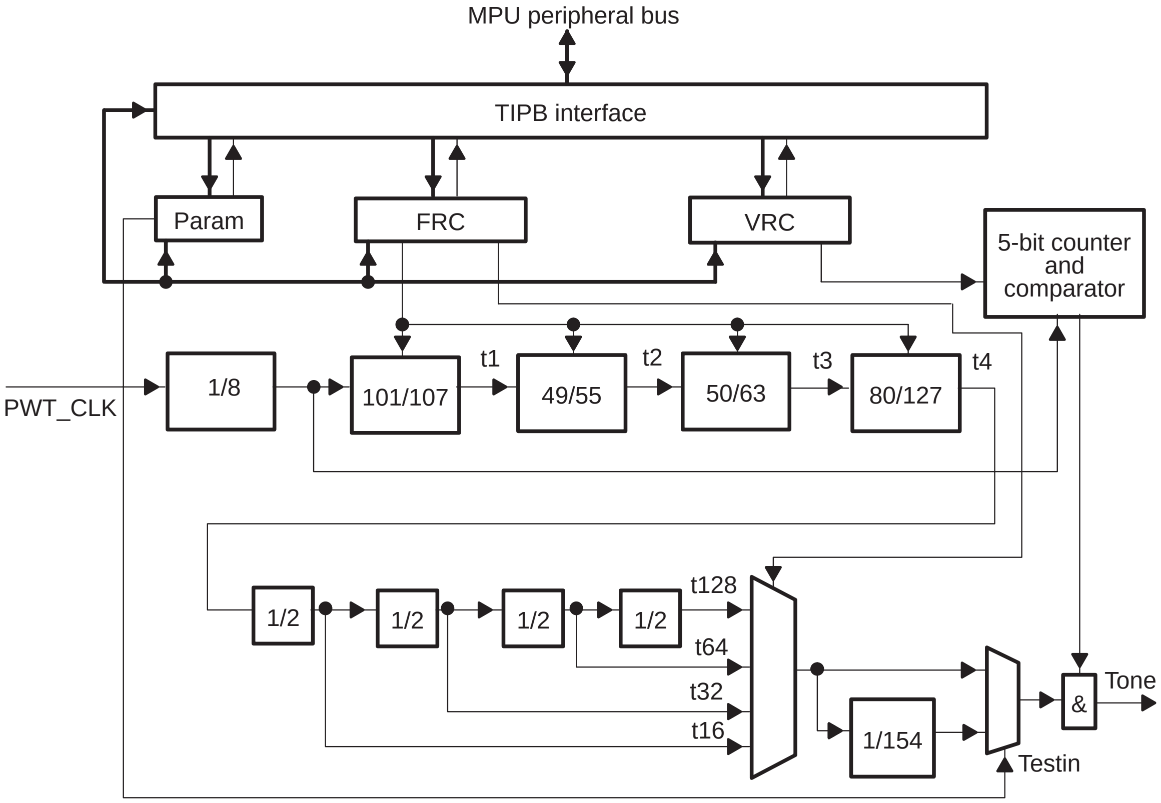
❑ 6-bit register to control tone volume

❑ 2-bit register for testing and CLK_EN

❑ 5-bit counter and comparator for creating volume pulse

❑ Divide by 1/154 to obtain the final right frequency

Figure 9. PWT Block Diagram



2.2 PWT Registers

Start address (hex): FFFB6000

Table 21 lists the 8-bit PWT registers. Table 22 through Table 24 describe the individual registers.

Table 21. PWT Registers

Name	Description	R/W	Offset
FRC	PWT frequency control	R/W	0x00
VRC	PWT volume control	R/W	0x04
GCR	PWT general control	R/W	0x08

Offset address (hex): 0x00

Table 22. PWT Frequency Control Register (FRC)

Bit	Name	Function	R/W	Reset
5:2	FRQ	Frequency selection (12 frequencies) Resynchronized writing, asynchronous reading	R/W	0000
1:0	OCT	Octave selection Resynchronized writing, asynchronous reading	R/W	00

Offset address (hex): 0x04

Table 23. PWT Volume Control Register (VRC)

Bit	Name	Function	R/W	Reset
6:1	VOL	Volume selection Resynchronized writing, asynchronous reading	R/W	000000
0	ONOFF	Switch ON/OFF tone (on: 1, off: 0). Resynchronized writing, asynchronous reading	R/W	0

Offset address (hex): 0x08

Table 24. PWT General Control Register (GCR)

Bit	Name	Function	R/W	Reset
1	TESTIN	Divider 1/154 switched ON/OFF (on: 0, off: 1). Asynchronous writing and reading	R/W	0
0	CLK_EN	PWT clock enable (clock disabled: 0, clock enabled: 1). Asynchronous writing and reading	R/W	0

2.3 PWT Programming

2.3.1 Buzzer Frequency

To obtain the required frequencies, the PWT clock is divided in a special way. Four frequency dividers with the coefficients $^{101}/_{107}$, $^{49}/_{55}$, $^{50}/_{63,}$ and $^{80}/_{127}$ are connected in series and can be enabled with the four frequency selection bits (FRQ) in the frequency register. If a divider is not enabled, the clock passes through the divider without any change so that different frequencies can be produced. After this, a multiplexer can choose between this clock, divided by 2, 4, 8, or 16. The frequency is always halved (this unit is called an octave). Because of this, the PWT has a range of four octaves.

The clock behind the multiplexer is divided by 154 to get the required frequencies on the tone output. The 12 frequencies in an octave can be programmed with bits 5 to 2 of the frequency control register (FRC), and the octave can be programmed with bits 1 to 0 of the FRC. Forty-eight different frequencies can be programmed, subdivided into four octaves with twelve frequencies. The four frequency dividers with the complex coefficients $^{101}/_{107}$, $^{49}/_{55}$, $^{50}/_{63,}$ and $^{80}/_{127}$ work with the fade-out principle. To get the divider $^{101}/_{107}$ from a periodic pulse, 6 pulses every 107 pulses are fade out. Over a long time, the resulting frequency is $^{101}/_{107}$. The resulting signal has two different periods, which differ by one period of the original signal. Because of this difference, the resulting signal has jitter. To minimize this jitter, the divider works with high frequencies that result in short periods producing low jitter (see Table 25).

Table 25. Buzzer Frequencies

FRC Bits 5-2 1-0	Buzzer Frequency	FRC Bits 5-2 1-0	Buzzer Frequency
0000 00	4868 Hz e5	0000 10	1217 Hz e3
0001 00	4595 Hz dis5	0001 10	1149 Hz dis3
0010 00	4337 Hz d5	0010 10	1084 Hz d3
0011 00	4093 Hz cis5	0011 10	1023 Hz cis3
0100 00	3864 Hz c5	0100 10	966 Hz c3
0101 00	3647 Hz h4	0101 10	912 Hz h2
0110 00	3442 Hz ais4	0110 10	860 Hz ais2
0111 00	3249 Hz a4	0111 10	812 Hz a2
1000 00	3066 Hz gis4	1000 10	767 Hz gis2
1001 00	2894 Hz g4	1001 10	723 Hz g2
1010 00	2732 Hz fis4	1010 10	683 Hz fis2
1011 00	2579 Hz f4	1011 10	644 Hz f2
0000 01	2434 Hz e4	0000 11	608 Hz e2
0001 01	2297 Hz dis4	0001 11	574 Hz dis2
0010 01	2168 Hz d4	0010 11	541 Hz d2
0011 01	2046 Hz cis4	0011 11	511 Hz cis2
0100 01	1932 Hz c4	0100 11	483 Hz c2
0101 01	1824 Hz h3	0101 11	456 Hz h1
0110 01	1721 Hz ais3	0110 11	430 Hz ais1

Note: The PWT was originally designed for a 13-MHz input clock, but most wireless OMAP solutions implement PWT with a 12-MHz clock. Consequently, frequencies shown are not exact tones.

Table 25. Buzzer Frequencies (Continued)

FRC Bits 5-2 1-0	Buzzer Frequency	FRC Bits 5-2 1-0	Buzzer Frequency
0111 01	1624 Hz a3	0111 11	406 Hz a1
1000 01	1533 Hz gis3	1000 11	383 Hz gis1
1001 01	1447 Hz g3	1001 11	361 Hz g1
1010 01	1366 Hz fis3	1010 11	341 Hz fis1
1011 01	1289 Hz f3	1011 11	322 Hz f1
		11XX XX	Not allowed

Note: The PWT was originally designed for a 13-MHz input clock, but most wireless OMAP solutions implement PWT with a 12-MHz clock. Consequently, frequencies shown are not exact tones.

2.3.2 Buzzer Volume

The buzzer volume can be programmed (see Table 26) with bits 6 to 1 in the volume control register VRC. The higher the 6-bit value is, the louder the buzzer/loudspeaker. To perform this programming, a 6-bit binary counter is clocked with the PWT clock and rolls over to 0h after reaching its terminal value (3 Fh). The counter value is compared with the 6-bit value programmed in VRC. If the count value is less than or equal to VRC, the output has the value H, else L:

- Y = VOL value: 0 =<y<64
- PWT_CLK = 12 MHz
- Output signal H period = (y+1).PWT_CLK
- Output signal L period = (63-y).PWT_CLK

Table 26. Buzzer Volume

VRC Bits 6-1 ; 0	Buzzer/Loudspeaker
111111 1	Loud
000000 1	Quiet
xxxxx 0	Off

3 Pulse-Width Light

The pulse-width light (PWL) module provides control of the LCD backlighting and keypad by employing a 4096-bit random sequence generator. This voltage-level control technique decreases the spectral power at the modulator harmonic frequencies. The module uses a 32-kHz clock from ULPD.

3.1 Functional Description

The PWL module is composed of a pseudorandom 8-bit data generator and a programmable threshold comparator (see Figure 10).

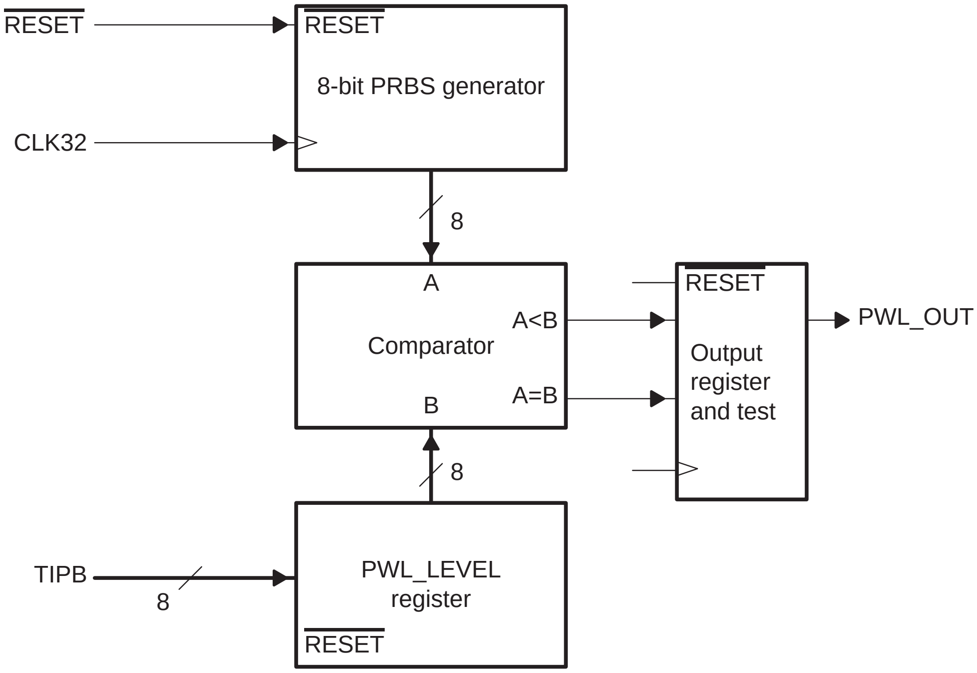
The pseudorandom 8-bit data generator is built using an LFSR. It generates a white normal-law random value between 1 and 255. The LFSR polynomial generator is P(x) = x[7] + x[3] + x[2] + x + 1.

The comparator generates:

- ❑ 0 if the random value is greater than or equal to the programmable threshold
- ❑ 1 if the random value is less than the programmable threshold

Assuming that the random sequence is normal, it generates a sequence whose mean value is proportional to the comparator threshold.

Figure 10. PWL Block Diagram


$\overline{\text{RESET}}$
$\overline{\text{RESET}}$
8-bit PRBS generator
CLK32
8
A
Comparator
A<B
A=B
B
$\overline{\text{RESET}}$
Output register and test
PWL_OUT
8
TIPB
8
PWL_LEVEL register
$\overline{\text{RESET}}$

3.2 PWL Registers

The PWL is connected to the host with a TIPB. The PWL control is done with two 8-bit registers. All TIPB accesses are done asynchronously with the 32-kHz clock, meaning that there is no TIPB wait-state insertion.

Table 27 lists the 8-bit PWL registers. Table 28 through Table 29 describe the individual registers.

Start address (hex): FFFB:5800

Table 27. PWL Registers

Name	Description	R/W	Offset
PWL_LEVEL	PWL-level	R/W	0x00
CLK_ENABLE	Clock enable	R/W	0x04

Offset address: 0x00

Table 28. PWL Level Register (PWL_LEVEL)

Bit	Name	Function	Reset
7:0	PWL_LEVEL	Defines the mean value of the PWL output signal. 0 leads to a continuous 0 output. 255 to an almost continuous 1 output: 255/256 cycles in high level.	0

Offset address: 0x04

Table 29. PWL Control Register (PWL_CTRL)

Bit	Name	Function	Reset
7:1	–	Reserved	
0	CLK_ENABLE	Internal clock is enabled when 1.	0

Index