OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide

Literature Number: SPRU679A
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

GPIOs are programmable inputs or outputs. They generate a level interrupt, the sources for which can be masked from within the GPIO module. Under software control, the GPIOs can be individually dedicated to the DSP or the MPU.

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.

OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide (literature number SPRU671)

OMAP5910 Dual-Core Processor DSP Subsystem Reference Guide (literature number SPRU672)

OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide (literature number SPRU673)

OMAP5910 Dual-Core Processor System DMA Controller Reference Guide (literature number SPRU674)

OMAP5910 Dual-Core Processor LCD Controller Reference Guide (literature number SPRU675)

OMAP5910 Dual-Core Processor Universal Asynchronous Receiver/Transmitter (UART) Devices Reference Guide (literature number SPRU676)

OMAP5910 Dual-Core Processor Universal Serial Bus (USB) and Frame Adjustment Counter (FAC) Reference Guide (literature number SPRU677)

OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide (literature number SPRU678)

OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide (literature number SPRU679)

OMAP5910 Dual-Core Processor MMC/SD Reference Guide (literature number SPRU680)

OMAP5910 Dual-Core Processor Inter-Integrated Circuit (I2C) Controller Reference Guide (literature number SPRU681)

OMAP5910 Dual-Core Processor Timer Reference Guide (literature number SPRU682)

OMAP5910 Dual-Core Processor Inter-Processor Communication Reference Guide (literature number SPRU683)

OMAP5910 Dual-Core Processor Camera Interface Reference Guide (literature number SPRU684)

OMAP5905 Dual-Core Processor Multichannel Serial Interface (MCSI) Reference Guide (literature number SPRU685)

OMAP5910 Dual-Core Processor Micro-Wire Interface Reference Guide (literature number SPRU686)

OMAP5910 Dual-Core Processor Real-Time Clock (RTC) Reference Guide (literature number SPRU687)

OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide (literature number SPRU688)

OMAP5910 Dual-Core Processor PWL, PWT, and LED Peripheral Reference Guide (literature number SPRU689)

OMAP5910 Dual-Core Processor Multichannel Buffered Serial Port (McBSP) Reference Guide (literature number SPRU708)

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

1 General-Purpose I/O 11
1.1 Input/Outputs of the GPIO Module 11
1.2 GPIO Port Registers 11

2 MPU I/O 16
2.1 MPU I/O Interrupts 16
2.2 MPU I/O Clocks and Reset 16
2.3 MPUIO Keyboard Interface 18
2.4 MPUIO General-Purpose I/O Interface 19
2.5 GPIO Interrupt Reset 20
2.6 GPIO Interrupt Masking 21
2.7 Event Capture Module 22
2.8 MPU I/O Registers 24

Figures

1 GPIO Module Architecture 12
2 MPU I/O Environment 17
3 Keyboard Process Block Diagram 19
4 GPIO Process 20
5 GPIO_INT Register Read Timing 21
6 MPU I/O Input Masking Timing 22
7 GPIO_CLK Timing 23
8 Event Capture Process 24

Tables

1 GPIO Port Registers 12
2 Data Input Register (DATA_INPUT_REG) 13
3 Data Output Register (DATA_OUTPUT_REG) 13
4 Direction Control Register (DIRECTION_CONTROL_REG) 14
5 Interrupt Control Register (INTERRUPT_CONTROL_REG) 14
6 Interrupt Mask Register (INTERRUPT_MASK_REG) 14
7 Interrupt Status Register (INTERRUPT_STATUS_REG) 15
8 MPU GPIO Pin Control Register (PIN_CONTROL_REG) 15
9 DSP GPIO Pin Control Status Register (PIN_CONTROL_STATUS_REG) 15
10 Keyboard Scanning Sequence 18
11 MPU Input/Output Registers 24
12 General-Purpose Input Register (INPUT_LATCH) 25
13 Output Register (OUTPUT_REG) 25
14 Input/Output Control Register (IO_CNTL) 25
15 Keyboard Row Inputs Register (KBR_LATCH) 26
16 Keyboard Column Outputs Register (KBC_REG) 26
17 GPIO Event Mode Register (GPIO_EVENT_MODE_REG) 26
18 GPIO Interrupt Edge Register (GPIO_INT_EDGE_REG) 26
19 Keyboard Interrupt Register (KBD _INT) 27
20 GPIO Interrupt Register (GPIO_INT) 27
21 Keyboard Mask Interrupt Register (KBD_ MASKIT) 27
22 GPIO Mask Interrupt Register (GPIO_MASKIT) 27
23 GPIO Debouncing Register (GPIO_DEBOUNCING_REG) 28
24 GPIO Latch Register (GPIO_LATCH_REG) 28

OMAP5910 GPIO

1 General-Purpose I/O

The GPIOs (see Figure 1) are programmable inputs or outputs. They generate a level interrupt, the sources for which can be masked from within the GPIO module. Under software control, the GPIOs can be individually dedicated to the DSP or the MPU.

GPIOs are accessible on general-purpose input and output external pins available to the user for system-level control and general-purpose functions. The signals are user-defined as either input or output. The output state can be controlled and inputs can be configured to provide an interrupt.

The MPU and the DSP have separate instances of the GPIO registers, but share the same device pins. The determination of whether MPU or DSP has control of the device pin is controlled by a single shared pin control register (at offset 0x18). In Figure 1 this register is shown as the configuration and control register. This register is read/write from/to the MPU, but read-only from the DSP. The MPU is responsible for writing to this register to assign any necessary GPIO signals to the DSP. By default, all GPIOs are assigned to the MPU.

GPIO interrupts are routed to both the MPU and DSP interrupt handlers, but a GPIO can only signal an interrupt to the processor to which it is assigned in the pin control register. By default, GPIO interrupts are disabled at both of the interrupt handlers.

There are no I/O signals associated with GPIO.5 and GPIO.10.

1.1 Input/Outputs of the GPIO Module

Some GPIO signals are multiplexed with other peripheral functions. For details on GPIO pins, see ***OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide*** (literature number SPRU671).

1.2 GPIO Port Registers

Table 1 lists the GPIO port registers. Table 2 through Table 9 describe the individual registers.

Each register exists in both the DSP GPIO and the MPU GPIO, except for the pin control (PIN_CONTROL_REG) and the pin status (PIN_CONTROL_STATUS_REG) registers.

Base Address: 0xFFFC:E000 (byte) for MPU; 0x0F000 (word) for DSP

Figure 1. GPIO Module Architecture

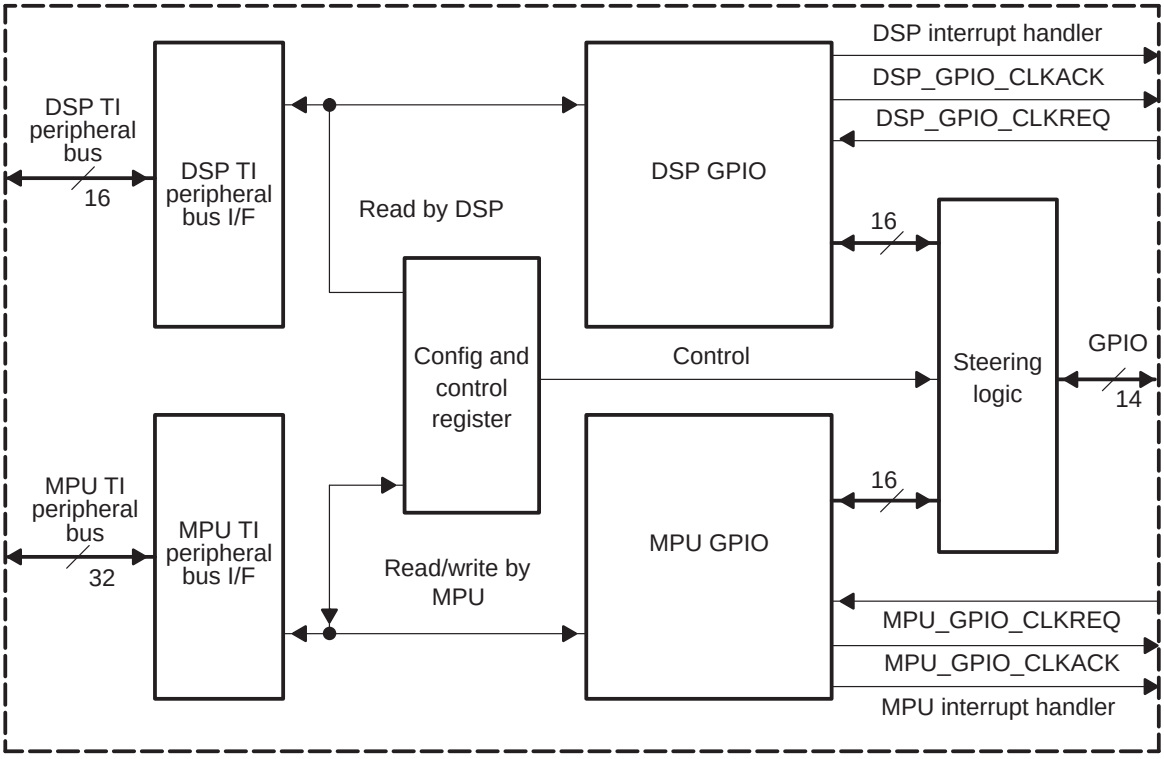


Table 1. GPIO Port Registers

Name	R/W	Size	Description	MPU Address	DSP Address
DATA_INPUT_REG	R	16 bits	Data input register	0xFFFC:E000	0x0F000
DATA_OUTPUT_REG	R/W	16 bits	Data output register	0xFFFC:E004	0x0F002
DIRECTION_CONTROL_REG	R/W	16 bits	Direction control register	0xFFFC:E008	0x0F004
INTERRUPT_CONTROL_REG	R/W	16 bits	Interrupt control register	0xFFFC:E00C	0x0F006

Table 1. GPIO Port Registers (Continued)

Name	R/W	Size	Description	MPU Address	DSP Address
INTERRUPT_MASK_REG	R/W	16 bits	Interrupt mask register	0xFFFC:E010	0x0F008
INTERRUPT_STATUS_REG	R/W	16 bits	Interrupt status register	0xFFFC:E014	0x0F00A
PIN_CONTROL_REG	R/W	16 bits	Pin control register (MPU only)	0xFFFC:E018	___
PIN_CONTROL_STATUS_REG	R	16 bits	Pin control status register (DSP only)	___	0x0F00C

The data input register is used to register the data that is read from the GPIO input pins. The input data is captured synchronously with an internal peripheral clock and can be read via the data input register. The GPIO input data is captured into this register four clock cycles after the GPIO input pin(s) change for synchronization and debouncing used to remove any input glitches. Bits not configured as input are undefined during read back. Both the MPU and DSP have read access to this register, but each processor only has read access to the individual bits controlled by that processor.

In the registers described in Table 3 through Table 7, both the MPU and the DSP have read/write access to only the bits they control within a register (bits associated with GPIO they control). The MPU (DSP) can write values to the bits not controlled by the MPU (DSP), but the written value is not valid and does not affect the configuration of the associated GPIO.

Table 2. Data Input Register (DATA_INPUT_REG)

Bits	Description	Access (R/W)	Reset Value
15−0	Receive data	R	0x0000

The data output register is used for setting the state on the GPIO output pins.

Table 3. Data Output Register (DATA_OUTPUT_REG)

Bits	Description	Access (R/W)	Reset Value
15−0	Data to transmit	R/W	0xFFFF

The direction control register is used to configure each of the GPIO pins for either input or output. At reset, all of the GPIO pins are configured as inputs.

Table 4. Direction Control Register (DIRECTION_CONTROL_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	Output	R/W	0xFFFF
	1	Input		

The interrupt control register (INTERRUPT_CONTROL_REG) allows the user to define interrupt generation for the controlling processor on either a high-to-low transition or a low-to-high transition.

Table 5. Interrupt Control Register (INTERRUPT_CONTROL_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	Interrupt generated on high to low transition	R/W	0xFFFF
	1	Interrupt generated on low to high transition		

The interrupt mask register (INTERRUPT_MASK_REG) allows the user to mask (disable) certain input pins from generating an interrupt request.

Table 6. Interrupt Mask Register (INTERRUPT_MASK_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	Enables interrupt	R/W	0xFFFF
	1	Disables interrupt		

The interrupt status register is used to determine which of the input pins requests an interrupt. Bit 0 corresponds to GPIO0 and so forth. If the value is a 1, then that pin is requesting the interrupt. The processor services the interrupt and resets the appropriate bit in the status register. If the user wants to reset the status bit, then a 1 must be written to the appropriate bit. The user

can not generate an interrupt by writing a 1 to the *interrupt status register*. If the user writes a 0 to a bit in the status register, the value remains unchanged.

Table 7. Interrupt Status Register (INTERRUPT_STATUS_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	No interrupt request	R/W	0x0000
	1	An interrupt has been requested		

The MPU GPIO pin control register (PIN_CONTROL_REG) is only accessible by the MPU. The MPU acts as master control and is responsible for assigning the GPIO I/O pins to either the MPU GPIO or the DSP GPIO. At reset, all pins are configured for MPU GPIO.

Table 8. MPU GPIO Pin Control Register (PIN_CONTROL_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	DSP GPIO pin	R/W	0xFFFF
	1	MPU GPIO pin		

The DSP GPIO pin control status register (PIN_CONTROL_STATUS_REG) is only accessibly by the DSP. This is a read-only register that allows the DSP to determine how the MPU has configured the GPIO pins.

Table 9. DSP GPIO Pin Control Status Register (PIN_CONTROL_STATUS_REG)

Bits	Value	Description	Access (R/W)	Reset Value
15−0	0	DSP GPIO pin	R	0xFFFF
	1	MPU GPIO pin		

2 MPU I/O

The MPU I/O module enables direct I/O communications between the MPU (through the public TIPB) and external devices (see Figure 2).

Two types of I/Os can be used:

- Specific I/Os dedicated for an 8x8 or 6x5 keyboard matrix:
  - Eight inputs (KB.R[7:0]) for row lines
  - Eight outputs (KB.C[7:0]) for column lines
- General-purpose I/Os:
  - Five MPU I/O signals (5, 4, 3, 2, and 1) are available in the default OMAP5910 multiplexing.
  - Five additional MPU I/O signals (12, 11, 7, 6, and 0) can be used by configuring the OMAP5910 multiplexing. For more detail, see ***OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide*** (literature number SPRU671).

2.1 MPU I/O Interrupts

The MPU I/O module generates two interrupts:

- The keyboard interrupt (KEYBOARD_INT), used to detect a key press, connected to the MPU interrupt handler level2, IRQ_01 (edge-sensitive)
- The GPIO interrupt (GPIOS_INT), used to detect an edge on one MPUIO input, connected to the MPU interrupt handler level2, IRQ_05 (level-sensitive).

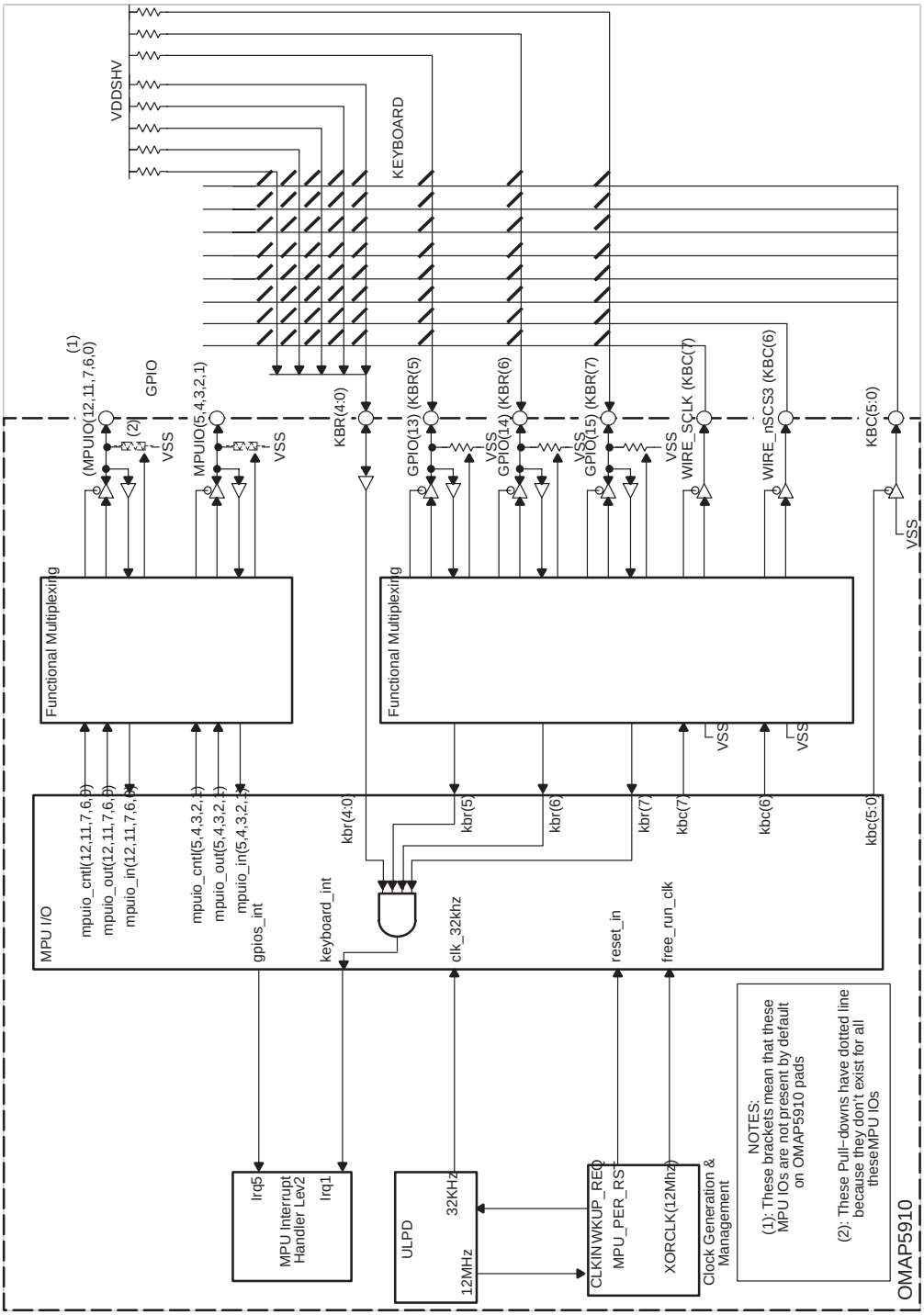
2.2 MPU I/O Clocks and Reset

The MPU I/O module has two clocks:

- The 32-kHz system clock (CLK_32KHZ), which comes, through the ULPD, from either the OMAP5910 32-kHz oscillator or the OMAP5910 CLK32K_IN CMOS input. For more detail, see *OMAP5910 Dual-Core Processor Clock Generation and System Reset Management* (literature number SPRU678).
- The 12-MHz clock (FREE_RUN_CLK), used to resynchronize the GPIO_INT register read. Comes from the MPU peripheral fixed clock (XORCLK). This clock is free running when OMAP5910 is awake.

The MPU TIPB reset (MPU_PER_RESET) resets the MPU I/O module.

Figure 2. MPU I/O Environment



2.3 MPUIO Keyboard Interface

To allow button press detection:

- All the row lines (KB.R) must have an external pullup.
- All the column lines (KB.C) drive a low level (idle state shown in Table 10).

  The output drivers of the KBC output pins are open-drain outputs and external pullup resistors are required to achieve a high state when these outputs are 3-stated.

The keyboard interrupt (keyboard_int) to the MPU is an AND of the eight row lines filtering during one 32-kHz clock period (CLK_32KHZ).

As soon as any key of the keyboard matrix is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row line, generating a keyboard interrupt (see Figure 3).

Once the keyboard interrupt is received, the MPU scans the column lines in the sequence described in the Table 10 in order to detect the key that has been pressed.

Table 10. Keyboard Scanning Sequence

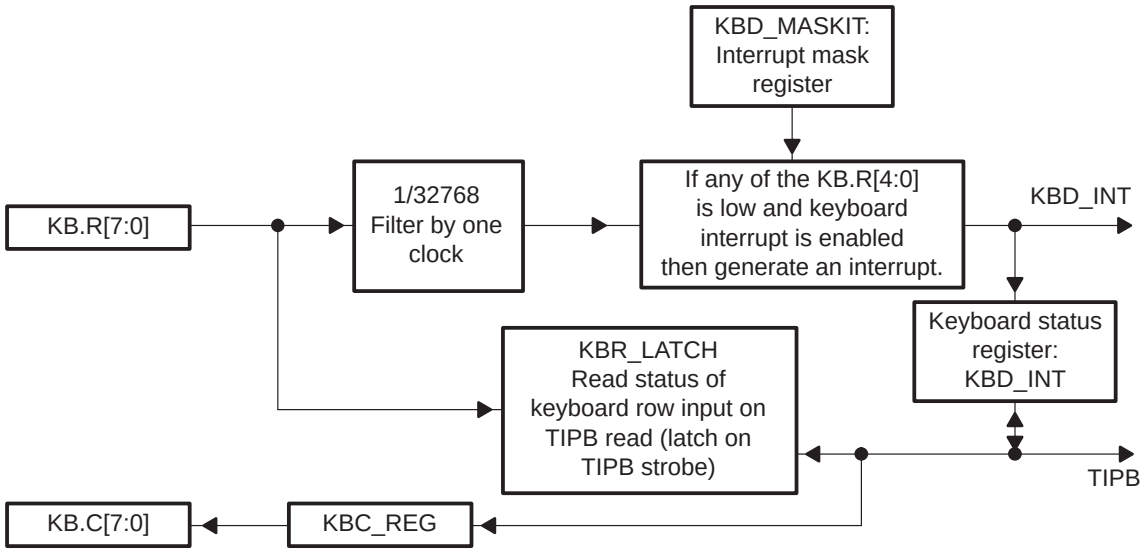
	Idle	Keyboard Scanning								Idle
KB.C[0]	0	1	0	1	1	1	1	1	1	0
KB.C[1]	0	1	1	0	1	1	1	1	1	0
KB.C[2]	0	1	1	1	0	1	1	1	1	0
KB.C[3]	0	1	1	1	1	0	1	1	1	0
KB.C[4]	0	1	1	1	1	1	0	1	1	0
KB.C[5]	0	1	1	1	1	1	1	0	1	0
KB.C[6]	0	1	1	1	1	1	1	1	0	0
KB.C[7]	0	1	1	1	1	1	1	1	1	0

For each step of the sequence, the MPU:

- Writes the specific value, with a 0 on one column bit, in the KBC_REG register
- Reads the value of the KBR_LATCH register and thus detects if one key of the concerned column line (this one which drives a low level) is pressed

At the end of the scanning sequence, the MPU is able to determine which keys have been pressed.

Figure 3. Keyboard Process Block Diagram

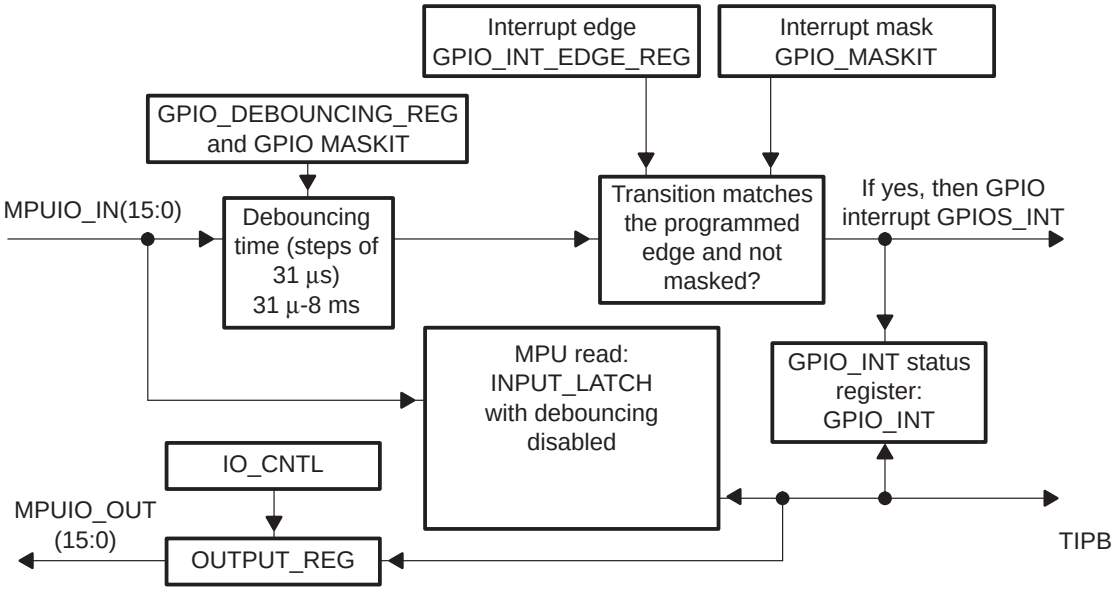


2.4 MPUIO General-Purpose I/O Interface

This interface has the following characteristics (see Figure 4):

- ❏ Every MPUIO can be configured individually either in input or in output mode.
- ❏ Interrupt generation (GPIOS_INT) on edge detection (rising or falling) after debouncing preprocessing.
- ❏ Edge detection can be used to latch all the GPIOs.
- ❏ The MPUIO interface works with the 32-kHz-system clock and consequently can be used to wake up the OMAP5910 device by generating the MPUIO interrupt.

Figure 4. GPIO Process



2.5 GPIO Interrupt Reset

The GPIO interrupt (gpios_int) is generated when one event occurs on one MPU I/O input (see Figure 5).

The edge detection and the interrupt generation are done synchronously with the 32-kHz system clock (clk_32khz).

These events (and consequently the GPIO_INT interrupt) are reset on one GPIO interrupt register (GPIO_INT) read.

Only the bits that are active after masking are reset.

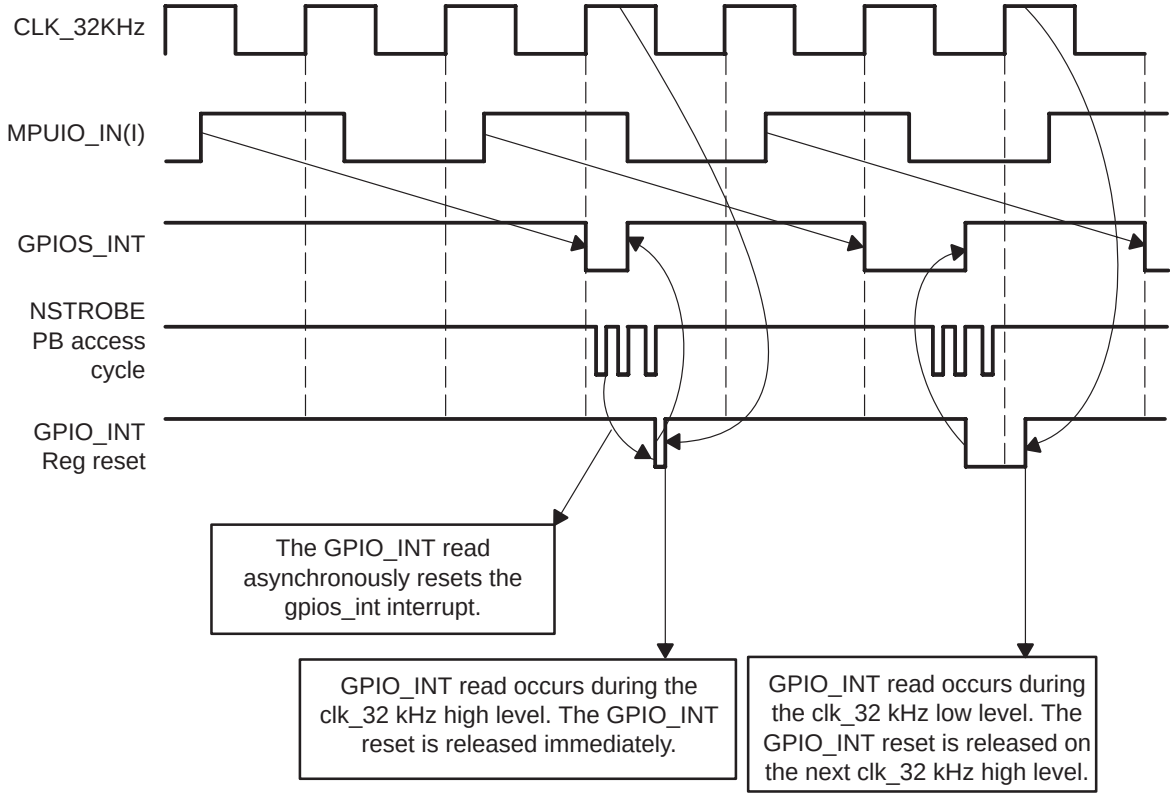
The GPIO_INT reset is synchronously asserted and synchronously released with the 32-kHz system clock. The GPIO_INT register read and the 32-kHz-system clock are resynchronized with the MPU peripheral bus fixed peripheral clock (12-MHz clock) free_run_clock command.

When the GPIO_INT read occurs:

- During a high level of the system clock, the reset is released immediately.
- During a low level of the system clock, reset is on the next high level of the system clock.

Even the worst case (reset release on the next 32-kHz cycle) supports the maximum speed of the MPU I/O module (one edge can be detected every two 32-kHz cycles with a debouncing 0).

Figure 5. GPIO_INT Register Read Timing



2.6 GPIO Interrupt Masking

MPUIO can generate the MPUIO interrupt when the edge of the signal level level changes are detected in the MPUIO input pins (*the event*). The GPIO interrupt mask register (GPIO_MASKIT) can mask the individual *event* among all the MPUIO input pins. If all *events* are masked, the MPUIO interrupt is not generated. The *event* information is not discarded while masked, but does generate an interrupt as soon as the event is unmasked.

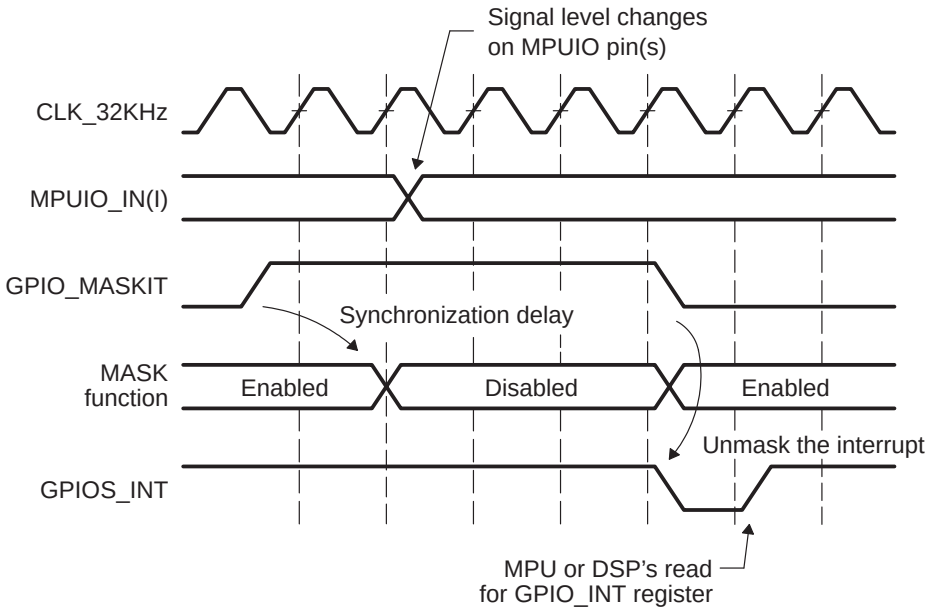
As there are two 32kHz cycle propogation delays needed to activate the masking function, the MPUIO interrupt may be generated during this period.

Setting the GPIO_MASKIT bits immediately sets to 0 the corresponding debouncing counter. If GPIO_MASKIT is set after the *event*, but before the debouncing counter expires, the GPIO_INT interrupt is generated six 32kHz-cycles later − 2 cycles to cancel debouncing, and 4 cycles to propagate the event to the interrupt generation.

Reading the GPIO_INT register does not clear masked events. When unmasked, the event generates the MPUIO interrupt, which is reset on the next GPIO_INT register read (shown in Figure 4).

Unmasked interrupts can be cleared by reading the GPIO_INT register.

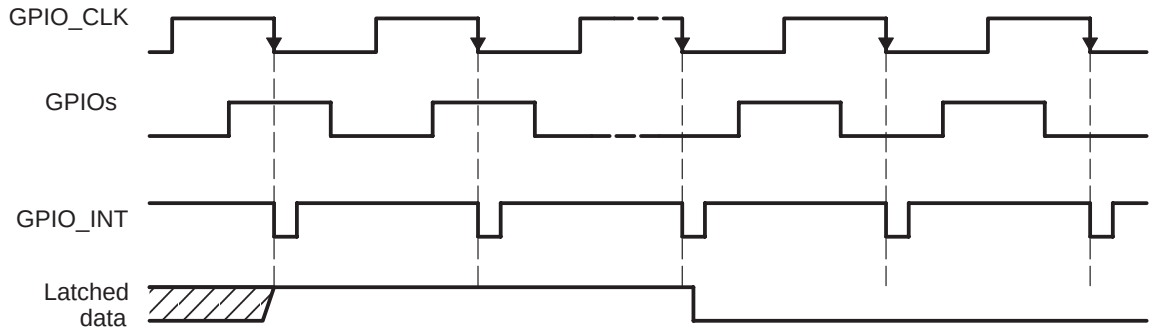
Figure 6. MPU I/O Input Masking Timing



2.7 Event Capture Module

The GPIO event capture mode allows latching the input value present on the GPIO ports each time a rising or a falling edge occurs on a selected GPIO port, here called GPIO_CLK. If not masked, the GPIO_CLK-selected edge generates an interrupt to the processor, as shown in Figure 7.

Figure 7. GPIO_CLK Timing



GPIO_CLK can be generated from an external physical module. Consequently, it may be necessary to insert a debouncing delay on this signal. The debouncing time is programmable in the GPIO debouncing register (GPIO_DEBOUNCING_REG) in steps of 31 µs.

The GPIO event mode register (GPIO_EVENT_MODE_REG) enables or disables the GPIO event mode. It also selects the external pin used as the GPIO_CLK. The GPIO interrupt edge register (GPIO_INT_EDGE) selects the GPIO_CLK falling or rising edge to generate the GPIO_INT interrupt.

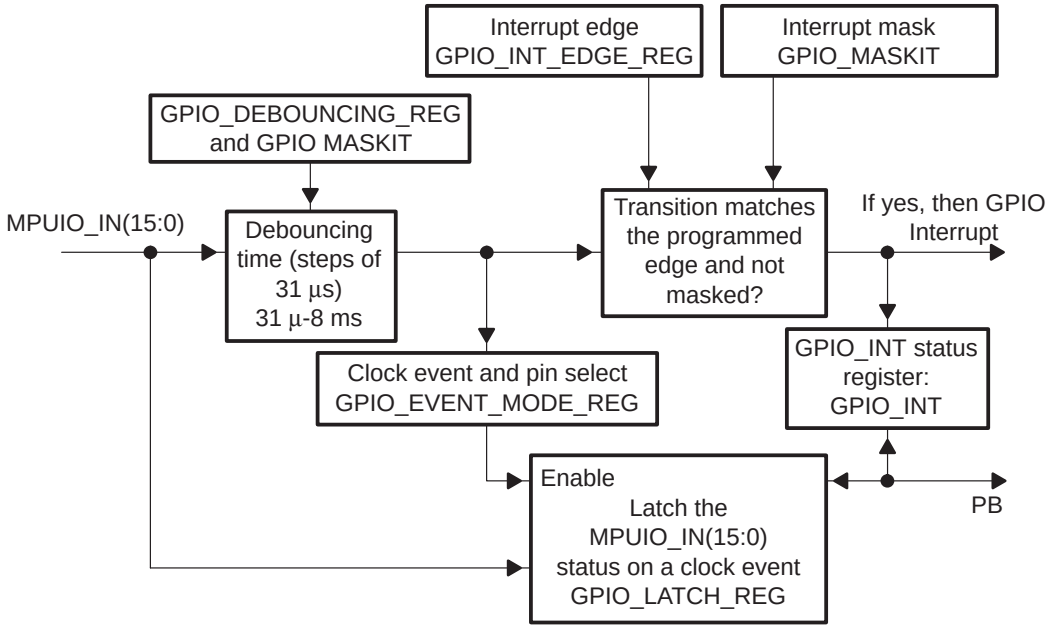
When the gpios_int interrupt (active low) is generated, the GPIO_INT register must be read by the MPU to define any active GPIO signal interrupts.

The GPIO interrupt mask register (GPIO_MASKIT) masks GPIO interrupts individually.

On the GPIO_CLK programmed edge, after the debouncing period, the MPUIO input pins are latched in the GPIO latch register (GPIO_LATCH_REG). Its value can be read after the detection of the interrupt, even if the external value has changed.

The event capture process is shown in Figure 8.

Figure 8. Event Capture Process



2.8 MPU I/O Registers

Start address in the MPU I/O range (hex): FFFB:5000

Table 11 lists the MPU I/O registers. Table 12 through Table 24 describe the individual registers.

Table 11. MPU Input/Output Registers

Register	Description	R/W	Size	Address	Offset
INPUT_LATCH	General-purpose input	R	16 bits	FFFB:5000	0x00
OUTPUT_REG	Output	R/W	16 bits	FFFB:5000	0x04
IO_CNTL	Input/Output control	R/W	16 bits	FFFB:5000	0x08
KBR_LATCH	Keyboard row inputs	R	16 bits	FFFB:5000	0x10
KBC_REG	Keyboard column outputs	R/W	16 bits	FFFB:5000	0x14
GPIO_EVENT_MODE_REG	GPIO event mode	R/W	16 bits	FFFB:5000	0x18
GPIO_INT_EDGE_REG	GPIO interrupt edge	R/W	16 bits	FFFB:5000	0x1C
KBD_INT	Keyboard interrupt	R	16 bits	FFFB:5000	0x20

Table 11. MPU Input/Output Registers (Continued)

Register	Description	R/W	Size	Address	Offset
GPIO_INT	GPIO interrupt	R	16 bits	FFFB:5000	0x24
KBD_MASKIT	Keyboard mask interrupt	R/W	16 bits	FFFB:5000	0x28
GPIO_MASKIT	GPIO mask interrupt	R/W	16 bits	FFFB:5000	0x2C
GPIO_DEBOUNCING_REG	GPIO debouncing	R/W	16 bits	FFFB:5000	0x30
GPIO_LATCH_REG	GPIO latch	R	16 bits	FFFB:5000	0x34

Table 12. General-Purpose Input Register (INPUT_LATCH)

Bits	Field	Description	Reset Value
15−0	INPUT_LATCH	General-purpose inputs	Reflects input pins

Table 13. Output Register (OUTPUT_REG)

Bits	Field	Description	Reset Value
15−0	OUTPUT_REG	General-purpose outputs	Undefined

Table 14. Input/Output Control Register (IO_CNTL)

Bits	Field	Value	Description	Reset Value
15−0	IO_CNTL		In/out control for general-purpose I/O	All bits at 1
		0	I/O is configured as output	
		1	I/O is configured as input	

Table 15. Keyboard Row Inputs Register (KBR_LATCH)

Bits	Field	Description	Reset Value
15−7	Reserved		
4−0	KBR_LATCH	Keyboard row inputs	Reflects input pins

Table 16. Keyboard Column Outputs Register (KBC_REG)

Bits	Field	Description	Reset Value
15−8	Reserved		
7−0	KBC_REG	Keyboard columns outputs	0

Table 17. GPIO Event Mode Register (GPIO_EVENT_MODE_REG)

Bits	Field	Value	Description	Reset Value
15−5	Reserved			
4−1	PIN_SELECT		Select MPUI/O_IN[15:0] pin to be the GPIO_CLK event	0000
		0000	Pin 0	
		1111	Pin 15	
0	SET_GPIO_EVENT_MODE	0	GPIO event mode disable	0
		1	GPIO event mode enable	

Table 18. GPIO Interrupt Edge Register (GPIO_INT_EDGE_REG)

Bits	Field	Value	Description	Reset Value
15−0	EDGE_SELECT[15:0]		Set interrupt on falling/rising edge	0
		0	Falling edge	
		1	Rising edge	

Table 19. Keyboard Interrupt Register (KBD _INT)

Bits	Field	Description	Reset Value
15−1	Reserved		
0	KBD_INT	Keyboard interrupt (active low)	1

Note: KBD_INT is a status bit only (duplication of the level of the corresponding interrupt signal).

Table 20. GPIO Interrupt Register (GPIO_INT)

Bits	Field	Description	Reset Value
15−0	GPIO_INT	GPIO interrupts (active high)	0

Note: GPIO_INT is reset on read access to the GPIO_INT register. The value read is the value after mask application.

Table 21. Keyboard Mask Interrupt Register (KBD_ MASKIT)

Bits	Field	Description	Reset Value
15−1	Reserved		
0	KBD_MASKIT	Mask is active at level 1, inactive at level 0	00

Table 22. GPIO Mask Interrupt Register (GPIO_MASKIT)

Bits	Field	Description	Reset Value
15−0	GPIO_MASKIT[15:0]	Mask is active at level 1, inactive at level 0	00

Table 23. GPIO Debouncing Register (GPIO_DEBOUNCING_REG)

Bits	Field	Description	Reset Value
15−9	Reserved		
8−0	GPIO_ DEBOUNCING_REG	000000000: 0 to 31 µs debouncing time 100000010: 7.97 ms to 8.0 ms debouncing time Programming step is 31 µs.	0000

Note: Because GPIO_CLK is an asynchronous signal, loading GPIO_DEBOUNCING_REG with 01 hex minimum value is recommended to ensure a 31-µs minimum debouncing time. If the value is 00 hex, the interrupt may be generated immediately when an edge is met.

Table 24. GPIO Latch Register (GPIO_LATCH_REG)

Bits	Field	Description	Reset Value
15−0	GPIO_LATCH_REG	After debouncing time, the ARMI/O_IN bus is latched in this.	00

Revision History

Table 25 lists the changes made since the previous version of this document.

Table 25. Document Revision History

Page	Additions/Modifications/Deletions
	This document has been reviewed for accuracy and there are no changes since the previous version (October 2003) of this document.

This page is intentionally left blank.

Index

C

clock, MPU, I/O 16

E

event capture, MPU I/O, MPU public peripherals 24

G

GPIO
- event capture, MPU public peripherals 20, 22
- interface 19
- interrupt masking, MPU public peripherals 21
- shared peripherals 11

I

I/O
- clocks, MPU 16
- interrupts, MPU 16
- MPU, keyboard interface 18
- reset, MPU 16

interrupt, MPU, I/O 16

K

keyboard interface, MPU I/O 18

M

MPU
- GPIO interface 19
- I/O
  - *clocks 16, 18, 19*
  - *keyboard interface 18*
  - *public peripherals 16*
  - *reset 16*
- interrupts, I/O 16

MPU public peripherals
- event capture 24
- MPU I/O
  - *GPIO event capture 20, 22*
  - *GPIO interrupt masking 21*
  - *overview 16*

MPU/DSP, shared peripherals, GPIOs 11

R

reset, MPU, I/O 16

S

shared peripherals, MPU/DSP, GPIOs 11