OMAP5912 Multimedia Processor Keyboard Interface Reference Guide

Literature Number: SPRU766A
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 keyboard 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

Keyboard Interface

This document describes the keyboard interface of the OMAP5912 multimedia processor.

1 MPUIO

The MPUIO module enables direct I/O communication between the MPU (through the public TIPB) and external devices (see Figure 1).

Two types of I/O can be used:

- ❑ Specific I/Os dedicated for 8 x 8 keyboard connection:
  - ■ Eight inputs (KB.R[7:0]) for row lines
  - ■ Eight outputs (KB.C[7:0]) for column lines

  The keyboard feature allows communication with a keyboard. The MPUIO supports keyboards with up to eight rows and eight columns and has the capability to detect multiple key presses. A keyboard event is signaled to the host by an interrupt.

- ❑ General-purpose I/Os:
  - ■ Five MPUIO signals (5, 4, 3, 2, and 1) are available in the default OMAP multiplexing.
  - ■ Eleven additional MPUIO signals (15, 14, 13, 12, 11, 10, 9, 8, 7, 6, and 0) can be used by configuring the OMAP multiplexing.

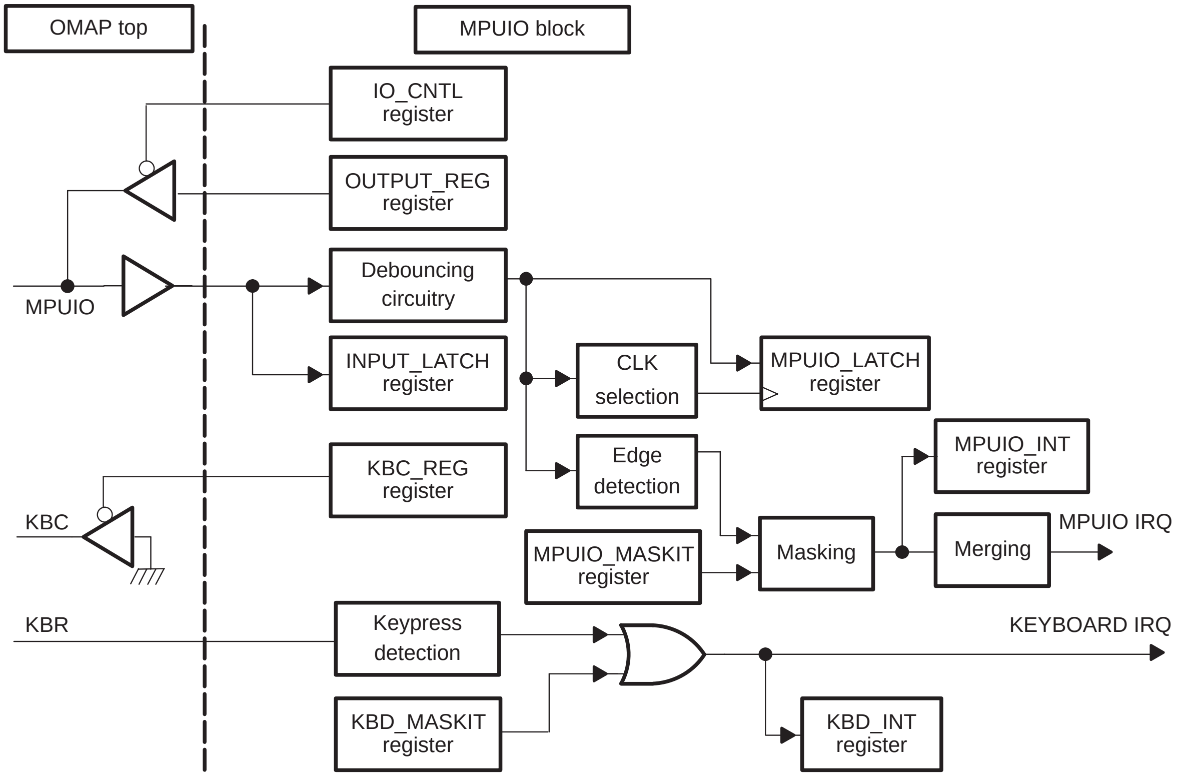
  The MPUIO feature allows communication with an external device through as many as 16 MPUIOs. These MPUIOs can be configured on a pin-by-pin basis as inputs or outputs. When configured as input, each MPUIO can be individually selected to generate interrupts on a level change (rising or falling edge). All the MPUIO inputs can be latched on this event.

  A simple filtering circuitry is implemented on the MPUIO input to allow debouncing.

The MPUIO module functional clock domain is clocked by the OMAP 32-kHz clock. This clock is always fed into the block, regardless of the state of the chip (awake, asleep, or idle). This allows external event latching and interrupt generation even when the system is in idle mode, to wake up the system via interrupt.

The MPUIO module interfaces with the host through a TIPB bus. The MPU peripheral clock resynchronizes register access to the module and avoids time-out on the TIPB bus caused by the functional clock being too slow.

Figure 1. MPUIO Module



1.1 MPUIO Function

The MPUIO module function allows:

- ❑ Individual configuration of 16 pins as input or output
- ❑ Capture of the 16-pin MPUIO bus on an edge of any one of the MPUIOs (rising- or falling-edge sensitivity can be chosen). The MPUIO selected to trigger the capture is the MPUIO_CLK.
- ❑ Interruption generation on the selected edge of any MPUIO. The interrupt generated is low-level sensitive, that is, it stays asserted low until cleared or masked.

1.1.1 MPUIO Debouncing

When MPUIO pins are configured in input, they can be connected to an external mechanical module. Debouncing circuitry is added on each MPUIO input. This circuitry allows an MPUIO pulse shorter than a given time to be ignored. Thus, bounce oscillations can be filtered out.

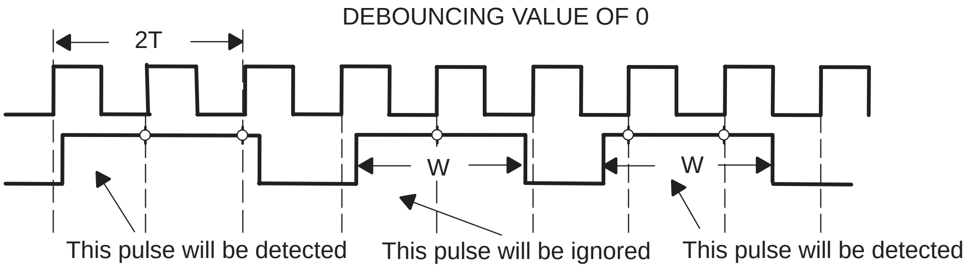
The debouncing time is programmed through the MPUIO_DEBOUNCING_REG register. It is the same for all MPUIO pins. When reference is made in all subsequent chapters to an event on an MPUIO input, it must be understood as an event after debouncing.

Because of internal resynchronization, the incoming MPUIO input must be sampled (DEBOUNCING_VALUE+2) times on the 32-kHz clock to be taken into account. This means that:

- Any pulse shorter than (DEBOUNCING_VALUE+1) 32-kHz clock periods is ignored.
- Any pulse longer than (DEBOUNCING_VALUE+2) 32-kHz clock periods is taken into account.
- Any pulse between these two durations is detected or not according to whether it can be sampled (DEBOUNCING_VALUE+2) times (see Figure 2).

This implies that any incoming MPUIO pulse shorter than two 32-kHz clock periods can be ignored even if debouncing is turned off (debouncing value equals 0).

Figure 2. Debouncing Value


DEBOUNCING VALUE OF 0
2T
W
W
This pulse will be detected
This pulse will be ignored
This pulse will be detected

1.2 Interrupt Handling

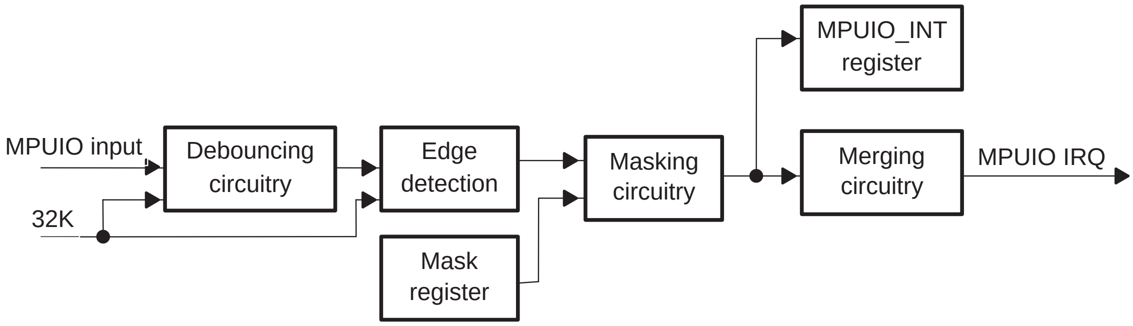
1.2.1 Interrupt Generation

Each MPUIO can be configured to generate an interrupt, either on the rising or falling edge, by unmasking interrupt generation in the MPUIO_MASKIT register. Edge sensitivity is programmed in MPUIO_INT_EDGE_REG.

On the specified edge of an incoming MPUIO, the event is registered in the MPUIO_INT register. If the interrupt generation for this MPUIO is enabled (unmasked), the MPUIO generates a low-level-sensitive interrupt to the host. Reading the MPUIO_INT register immediately resets all the pending interrupt bits and deasserts MPUIO interrupt.

Interrupt masking occurs after storage into the MPUIO interrupt register, so unmasking a pending interrupt immediatly asserts the MPUIOS_INT signal (see Figure 3).

Figure 3. MPUIOS_INT Generation



The debouncing filtering induces a delay between GPIO incoming pulse to the interrupt generation. The induced delay is more than (GPIO_DEBOUNCING_REG + 3) and less than (GPIO_DEBOUNCING_REG + 4) 32-kHz clock cycles.

For example, assuming a debouncing value of 0x000, the propagation time of an interrupt (from GPIO input edge to GPIOS_INT assertion) is between three and four 32-kHz clock cycles (between 93.75 and 125 µs).

For high debouncing values, this delay can be quite large. Therefore, when a GPIO interrupt is masked out, the debouncing value is set to 0 (GPIO_DEBOUNCING_REG is assumed to be 0) for this GPIO only. This feature ensures low-interrupt latency when an interrupt is masked and then unmasked, for slowly bouncing external devices.

1.2.2 Interrupt Acknowledgement

Acknowledging an MPUIO interrupt is done by reading the MPUIO_INT register. This immediately resets the active bits in this register and deasserts the MPUIO interrupt.

However, owing to internal reset resynchronization, the MPUIO lines being reset are not able to generate interrupts before the next 32-kHz clock rising edge. Reset of interrupts is asynchronously asserted but released synchronously with 32-kHz clock.

1.3 Interrupt Masking

Each incoming MPUIO line can be individually masked through the MPUIO_MASKIT register. When an MPUIO is masked, it does not generate an interrupt to the host.

Masking is done asynchronously, so masking all incoming MPUIOs immediately deasserts the MPUIOS_INT interrupt, if it was asserted.

When an MPUIO is masked, its debouncing value is forced to zero. This means the correct edge on this MPUIO generates an update of the MPUIO_INT register after a maximum of six 32-kHz clock cycles. As a side effect, no filtering is done on this MPUIO.

If an MPUIO can generate interrupts at a higher frequency than $\frac{CLK_{32kHz}}{6}$, masking and then unmasking it dynamically can result in interrupt loss.

1.4 MPUIO Event Capture

The MPUIO event capture allows latching the input value present on the MPUIO ports each time a rising or a falling edge occurs on a selected MPUIO port, here called MPUIO_CLK (see Figure 4).

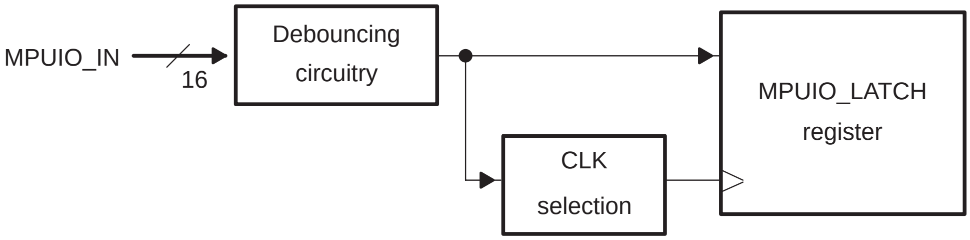
In all respects, MPUIO_CLK behaves as a regular MPUIO. It is debounced as all the others, and, if not masked, the MPUIO_CLK selected edge normally generates an interrupt to the host.

The MPUIO_EVENT_MODE_REG register enables or disables the MPUIO event capture mode. It also selects the external pin used as the MPUIO_CLK.

The MPUIO_INT_EDGE_REG bit that corresponds to the MPUIO_CLK determines which edge of MPUIO_CLK is used for capture.

On the MPUIO_CLK programmed edge, after the debouncing delay, the internal ARMIO_IN bus is latched in the MPUIO_LATCH_REG register. Its value can be read after the detection of the interrupt even if the external value has changed.

Figure 4. MPUIO Event Capture Mode



1.5 Keyboard Function

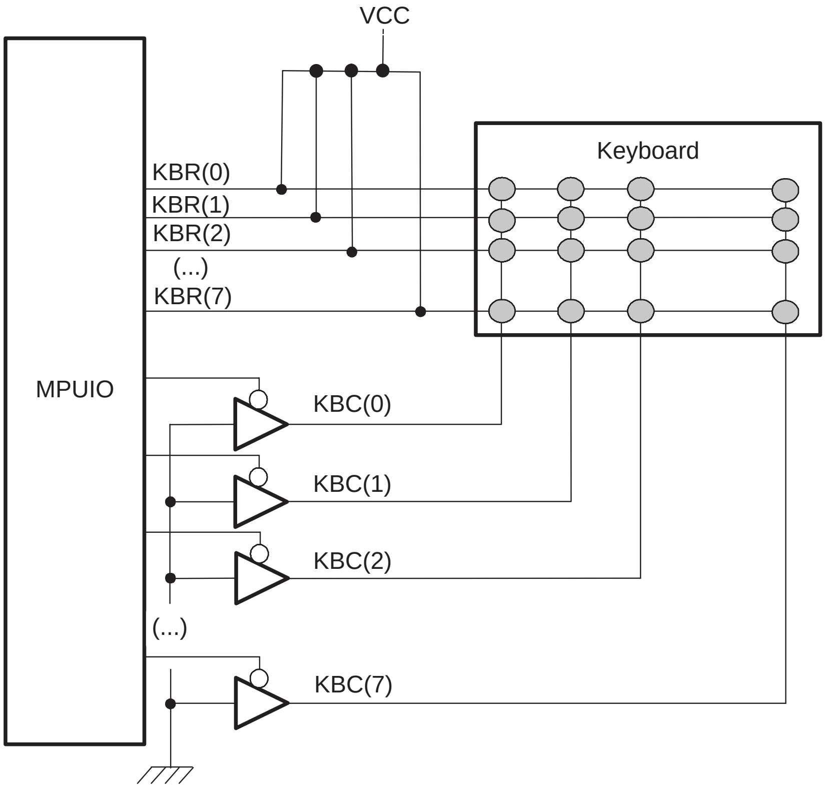
The keyboard is connected to the chip (see Figure 5) using:

- ❑ KBR(7:0) input pins for row lines
- ❑ KBC(7:0) output pins for column lines

All input pins (KBR) are externally pulled up to VCC.

By default, all output pins (KBC) drive a low level.

Figure 5. Keyboard Connection to MPUIO Module



1.6 Keyboard Scanning

If a key of the keyboard matrix is pressed, the corresponding row and column lines are shorted together, which causes a low level to be input on one of the KBR pins. This generates an interrupt to the host. On receiving the interrupt, the host must scan the column lines according to the sequence shown in Table 1.

Table 1. Keyboard Scanning Sequence for Four Pressed Keys

Pin	Idle	Keyboard Scanning[†]									Idle
KBC(0)	0	Z	0	Z	Z	Z	Z	Z	Z	Z	0
KBC(1)	0	Z	Z	0	Z	Z	Z	Z	Z	Z	0
KBC(2)	0	Z	Z	Z	0	Z	Z	Z	Z	Z	0
KBC(3)	0	Z	Z	Z	Z	0	Z	Z	Z	Z	0
KBC(4)	0	Z	Z	Z	Z	Z	0	Z	Z	Z	0
KBC(5)	0	Z	Z	Z	Z	Z	Z	0	Z	Z	0
KBC(6)	0	Z	Z	Z	Z	Z	Z	Z	0	Z	0
KBC(7)	0	Z	Z	Z	Z	Z	Z	Z	Z	0	0

[†] Z means the output is in high impedance state (1 is written in the KBC_REG corresponding bit).

This sequence is written to allow detection of simultaneous press actions on several buttons. It ensures that any combination of pressed keys can be read, if proper hardware protection has been implemented to prevent ghostkey detection.

Writing a pattern on KBC is done by writing into the KBC_REG register. The KBR inputs are directly accessed by reading the KBR_LATCH register.

See Table 2 for values read on the KBR lines during the scanning sequence if keys at (row, column) (3,3), (3,5), (5,3), and (7,7) are simultaneously pressed.

Table 2. Keyboard Scanning Result for Four Pressed Keys

Pin	Idle	Keyboard Scanning									Idle
KBR(0)	1	1	1	1	1	1	1	1	1	1	1
KBR(1)	1	1	1	1	1	1	1	1	1	1	1
KBR(2)	1	1	1	1	1	1	1	1	1	1	1
KBR(3)	0	1	1	1	1	0[†]	1	0[‡]	1	1	0

Table 2. Keyboard Scanning Result for Four Pressed Keys (Continued)

Pin	Idle	Keyboard Scanning								Idle
KBR(4)	1	1	1	1	1	1	1	1	1	1	1
KBR(5)	0	1	1	1	1	0[†]	1	1	1	1	0
KBR(6)	1	1	1	1	1	1	1	1	1	1	1
KBR(7)	0	1	1	1	1	1	1	1	1	0[§]	0

[†] KBC(3) drives 0. As keys (3,3) and (5,3) are pressed, KBR(3) and KBR(5) read as zero.
[‡] KBC(5) drives 0. As key (3,5) is pressed, KBR(3) reads as zero.
[§] KBC(7) drives 0. As key (7,7) is pressed, KBR(7) reads as zero.

The keyboard interrupt is falling-edge sensitive (the interrupt controller must detect a falling edge on this line).

2 MPUIO Registers

Table 3 lists the 16-bit MPUIO registers. Table 4 through Table 16 describe the individual registers.

Table 3. MPU Input/Output Registers

Start Address in the MPUIO range (hex): FFFB:5000			
Register	**Description**	**R/W**	**Offset**
INPUT_LATCH	General-purpose input	R	0x00
OUTPUT_REG	General-purpose output	R/W	0x04
IO_CNTL	Input/output control	R/W	0x08
KBR_LATCH	Keyboard row inputs	R	0x10
KBC_REG	Keyboard column outputs	R/W	0x14
MPUIO_EVENT_MODE_REG	MPUIO event mode	R/W	0x18
MPUIO_INT_EDGE_REG	MPUIO interrupt edge	R/W	0x1C
KBD_INT	Keyboard interrupt	R	0x20
MPUIO_INT	MPUIO interrupt	R	0x24
KBD_MASKIT	Keyboard mask interrupt	R/W	0x28
MPUIO_MASKIT	MPUIO mask interrupt	R/W	0x2C
MPUIO_DEBOUNCING_REG	MPUIO debouncing	R/W	0x30
MPUIO_LATCH_REG	MPUIO latch	R	0x34

Table 4. General-Purpose Input Register (INPUT_LATCH)

Base Address = 0xFFFB 5000, Offset Address = 0x00			
Bit	**Name**	**Function**	**Reset**
15:0	INPUT_LATCH	General-purpose inputs	0x0000

Table 5. Output Register (OUTPUT_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x04			
Bit	**Name**	**Function**	**Reset**
15:0	OUTPUT_REG	General-purpose outputs	0x0000

Table 6. Input/Output Control Register (IO_CNTL)

Base Address = 0xFFFB 5000, Offset Address = 0x08				
Bit	**Name**	**Value**	**Function**	**Reset**
15:0	IO_CNTL		In/out control for general-purpose I/O	0xFFFF
		0	I/O is configured as output.	
		1	I/O is configured as input.	

Table 7. Keyboard Row Inputs Register (KBR_LATCH)

Base Address = 0xFFFB 5000, Offset Address = 0x10			
Bit	**Name**	**Function**	**Reset**
15:8	Reserved		0xFF
7:0	KBR_LATCH	Keyboard row inputs	0xFF

Table 8. Keyboard Column Outputs Register (KBC_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x14				
Bit	**Name**	**Value**	**Function**	**Reset**
15:8	Reserved			0xFF
7:0	KBC_REG		Keyboard column outputs	0x00
		0	Output drives a low level.	
		1	Output is in high-impedance state.	

Table 9. MPUIO Event Mode Register (MPUIO_EVENT_MODE_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x18				
Bit	**Name**	**Value**	**Function**	**Reset**
15:5	Reserved			0x7FF
4:1	PIN_SELECT		Select MPUIO_IN[15:0] pin to be the MPUIO_CLK event	0x0
		0000	Pin 0	
		0001	Pin 1	
		. . .		
		1111	Pin 15	
0	SET_MPUIO_EVENT_ MODE	0	MPUIO event mode disable	0x0
		1	MPUIO event mode enable	

Table 10. MPUIO Interrupt Edge Register (MPUIO_INT_EDGE_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x1C				
Bit	**Name**	**Value**	**Function**	**Reset**
15:0	EDGE_SELECT		Set interrupt on falling/rising edge	0x0
		0	Falling edge	
		1	Rising edge	

Table 11. Keyboard Interrupt Register (KBD _INT)

Base Address = 0xFFFB 5000, Offset Address = 0x20				
Bit	**Name**	**Value**	**Function**	**Reset**
15:1	Reserved			0x7FFF
0	KBD_INT		Keyboard interrupt (active low)	0x1
		0	Interrupt occurred	
		1	No interrupt pending	

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

Table 12. MPUIO Interrupt Register (MPUIO_INT)

Base Address = 0xFFFB 5000, Offset Address = 0x24			
Bit	**Name**	**Function**	**Reset**
15:0	MPUIO_INT	MPUIO interrupts (active high)	0x0

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

Even in emulation mode, the MPUIO interrupts are reset by a read in the MPUIO interrupt register (MPUIO_INT).

Table 13. Keyboard Mask Interrupt Register (KBD_ MASKIT)

Base Address = 0xFFFB 5000, Offset Address = 0x28			
Bit	**Name**	**Function**	**Reset**
15:1	Reserved		0x7FFF
0	KBD_MASKIT	Mask is active at level 1, inactive at level 0.	0x0

Table 14. MPUIO Mask Interrupt Register (MPUIO_MASKIT)

Base Address = 0xFFFB 5000, Offset Address = 0x2C			
Bit	**Name**	**Function**	**Reset**
15:0	MPUIO_MASKIT[15:0]	Mask is active at level 1, inactive at level 0.	0x0

Table 15. MPUIO Debouncing Register (MPUIO_DEBOUNCING_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x30				
Bit	**Name**	**Value**	**Function**	**Reset**
15:9	Reserved			0x7F
8:0	MPUIO_ DEBOUNCING_REG		0: 0−511 cycles of T32k debouncing time. Programming step is T32k = 32.5 µs.	0x0

Note: MPUIO_DEBOUNCING_REG ignores pulse widths less than 32 kHz. For example, when T_{32k}=31.25 µs, a value of 0x000 filters nothing, whereas a value of 0x100 ignores pulses smaller than 8 ms. *The MPUIO accommodates values between 0x000 and 0x100. If a greater value is set into the MPUIO_DEBOUNCING_REG register, the module behavior is not assured.*

Table 16. MPUIO Latch Register (MPUIO_LATCH_REG)

Base Address = 0xFFFB 5000, Offset Address = 0x34			
Bit	**Name**	**Function**	**Reset**
15:0	MPUIO_LATCH_REG	After debouncing time, the MPUIO_IN bus is latched in this register.	0x0

Index