General-Purpose Input/Output

The general-purpose input/output (GIO) module provides the TMS470R3x family of devices with input/output (I/O) capability. The I/O pins are bidirectional and bit-programmable. The GIO module supports up to thirty two external interrupts; these interrupt pins are multiplexed with I/O pins within the GIO ports A - D .

Topic **Page**

History and References

Revision History

Date	Version	Author	Description
05/26/2003	0.10	Prabhu M	- Intial Version (Move from R1x to R3x)
03/16/2004	0.20	Deepak Garg	- Added Register GIOPSL and GIOP-ULDIS.
04/22/2004	0.30	Deepak Garg	- Added Register GIOINTDET.
06/03/2004	0.40	Deepak Garg	- Reserved bits read as ‘0’ (Changed from R-U to R-0 in the register description).

1 Overview

The GIO module has the following features:

- Each I/O pin is controlled by bits in five registers:
  - Data direction (GIODIR)
  - Data input (GIODIN)
  - Data output (GIODOUT)
  - Data set (GIODSET)
  - Data clear (GIODCLR)
  - Open Drain (GIOPDR)
  - Pull Disable (GIOPULDIS)
  - Pull Select (GIOPSL)

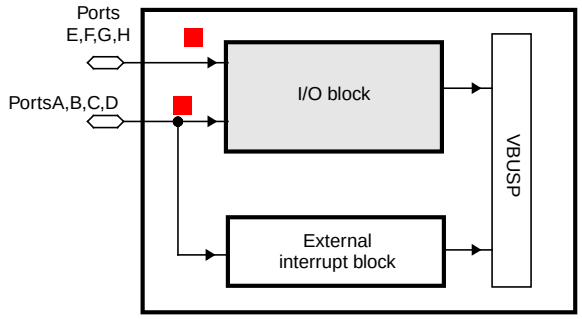
- The interrupts have the following characteristics:
  - Programmable interrupt detection either on both edge or on single edge (set in GIOINTDET)
  - Programmable edge-detection (set in GIOPOL register)
  - Individual interrupt flags (set in GIOFLG register)
  - Individual enables (set and cleared through GIOENASET and GIOENACLR registers respectively)
  - Programmable priority (set through GIOPRYSET/GIOLVLSET and GIOPRYCLR/GIOLVLCLR registers)

- Internal pullup/pulldown allows you to leave unused I/O pins unconnected.

2 Functional Description of GIO Module

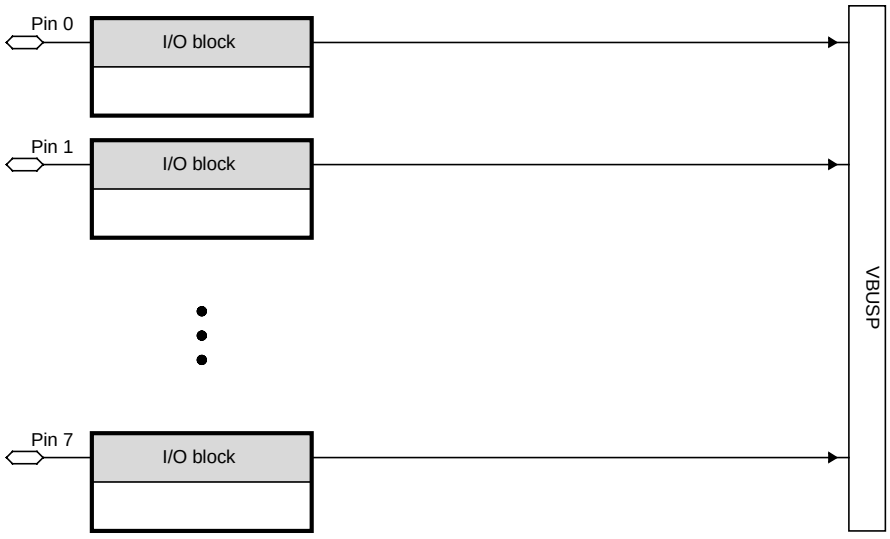
The GIO module (see Figure 1) is composed of two separate components: an input/output (I/O) block and an external interrupt block.

Figure 1. GIO Module Diagram



The pins for ports E, F, G, and H handle the standard I/O functions and are connected directly to the expansion bus. See Figure 2. Ports B, C and D can also handle interrupts along with the standard I/O functions. The interrupt handling feature for ports B, C and D is device specific. Port A is always capable of handling interrupts.

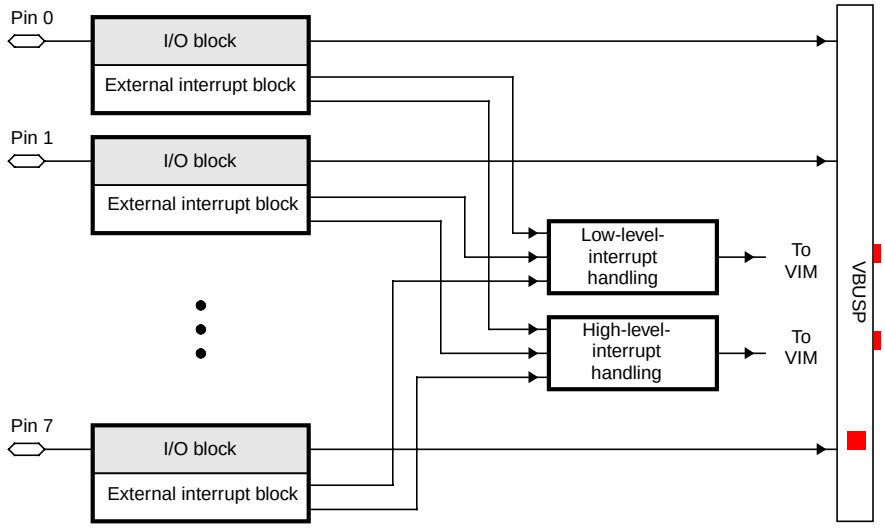
Figure 2. GIO Ports E, F, G, and H Diagram



The pins on port A (and optionally port B, C and D), shown in Figure 3, are all interrupt-capable pins and can be used to handle either general I/O functions or external interrupt signals. The GIO module can support up to thirty two external interrupt pins; however, the actual number of pins is device-specific .

The port A pins (and optionally port B, C and D pins, which is device specific) are connected to both an I/O block and an external interrupt block. Each of the eight I/O blocks is connected to the VBUSP bus, whereas the interrupt blocks are physically attached to a single high-level-interrupt-handling block and to a single low-level-interrupt-handling block. The high-level-interrupt-handling block and the low-level-interrupt-handling block each send one signal to the vectored interrupt manager (VIM) in the system module for processing.

Figure 3. GIO Port A / B / C / D Module Diagram



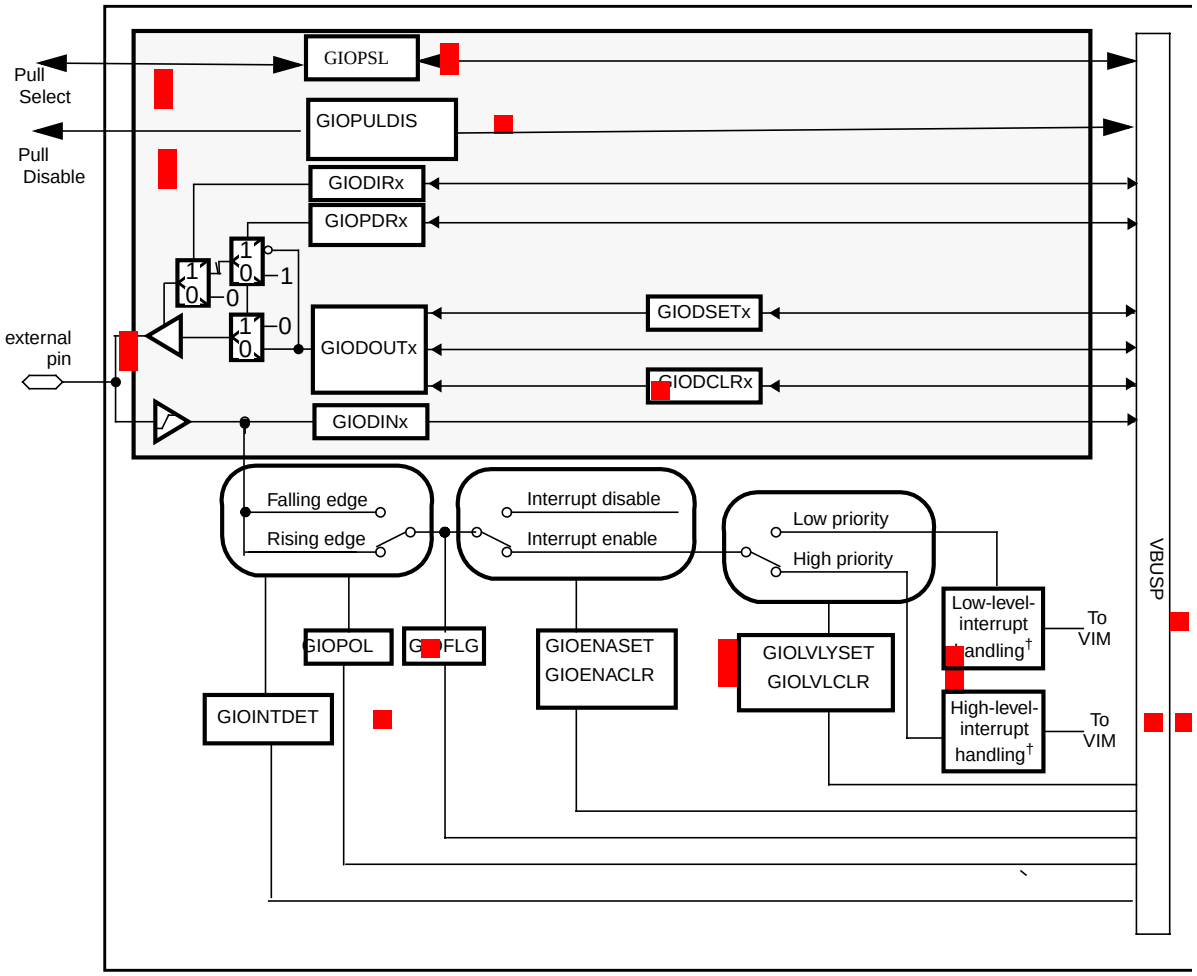
Note: Not all devices have thirty two external interrupts. The exact number of external is device specific . Consult the device specification for details.

2.1 GIO Block Diagram

The GIO block diagram (see Figure 4) represents the flow of information through a pin. The shaded area corresponds to the I/O block; the unshaded area corresponds to the external interrupt block.

Because ports E, F, G, and H are not interrupt-capable, the block diagram for those ports reduces to the shaded portion of the block diagram. (See Figure 5 on page 10.)

Figure 4. GIO Block Diagram



† A single low-level-interrupt-handling block and a single high-level-interrupt-handling block service all of the interrupt-capable external pins, but only one pin can be serviced by an interrupt block at a time.

The high-level and the low-level-interrupt-handling blocks each contain an offset register and an emulation register. The high-level interrupts are denoted as level A (GIOOFFA/GIOOFFSET0 and GIOEMUA/GIOMON0); the low-level interrupts are denoted as level B (GIOOFFB/GIOOFFSET1 and GIOEMUB/GIOMON1). For details on interrupt levels, see section 2.4.2, *Interrupts and Interrupt Levels*, page 14.

2.2 GIO Internal Registers

A general representation of the GIO internal registers is shown in Table 1. The page column provides a cross-reference to additional information on the individual registers. For a more detailed description of the individual bits.

Table 1. GIO Internal Registers

Offset Address†	Mnemonic	Name	Description	Page
0x00	GIOGCR0	GIO Global Control Register	Controls the module level rest	25
0x04	GIOPWDN	GIO Power Down	Controls the module power-down status	32
0x08	GIOINTDET	GIO Interrupt detect edge	cause flag to set on both edge if set	34
0x0C	GIOPOL	GIO Interrupt Polarity	Causes flag to set on falling or rising edge	35
0x10	GIOENASET	GIO Interrupt Enable	Configures corresponding pins as interrupts	36
0x14	GIOENACLR			
0x18	GIOLVLSET	GIO Interrupt Priority	Sets interrupts for high or low priority	38
0x1C	GIOLVLCLR			
0x20	GIOFLG	GIO Interrupt Flag	Indicates that an appropriate transition edge has occurred	40

Table 1. GIO Internal Registers

0x24	GIOOFFA / GIOOFFSET 0	GIO Offset A	Provides offset that represents the pending external high-priority interrupt	41
0x2C	GIOEMUA / GIOMON0	GIO Emulation A	Reflects contents of GIOOFFA register	43
0x28	GIOOFFB / GIOOFFSET 1	GIO Offset B	Provides offset that represents the pending external low-priority interrupt	44
0x30	GIOEMUB / GIOMON1	GIO Emulation B	Reflects contents of GIOOFFB register	46
0x34	GIODIRx	GIO Data Direction	Configures corresponding pin as an input or output	47
0x38	GIODINx	GIO Data Input	Reflects current value on input pins	48
0x3C	GIODOUTx	GIO Data Output	Specifies value output to pins	49
0x40	GIODSETx	GIO Data Set	Sets bits in the GIODOUTx register	50
0x44	GIODCLRx	GIO Data Clear	Clears bits in the GIODOUTx register	51
0x48	GIOPDRx	GIO Open Drain Register	Configures the Open Drain capability of the pin	52
0x4C	GIOPULD-ISx	GIO Pull Disable Regis-ter	Disable Pull Control Capability at pin	53
0x50	GIOPSL	GIO Pull Select Register	Select the Pull Type at the Pin	54

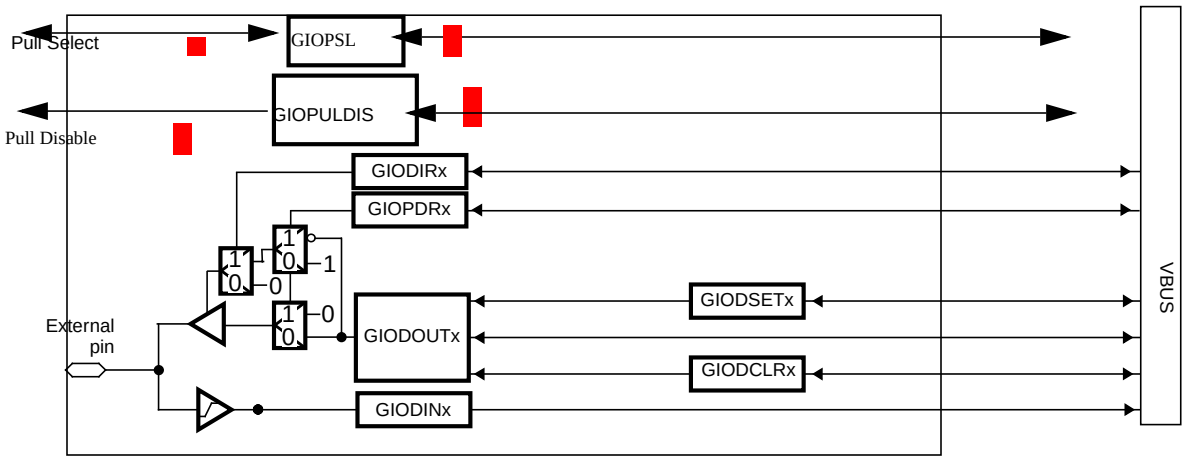
The physical address of these registers is device-specific. If the GIO memory begins at 0xFFF7EC00, the GIOENASET register is located at 0xFFF7EC0C. See the specific device data sheet to verify the register addresses.

The shaded registers (GIODIRx, GIODINx, GIODOUTx, GIODSETx, GIODCLRx, GIOPDRx, GIOPULDISx,GIOPSLx) exist for each I/O port. The beginning offset addresses for each sequence of seven registers are: 0x34, 0x54, 0x74, 0x94, 0xB4, 0xD4, 0xF4, 0x114. See the specific device data sheet to verify the register addresses.

2.3 I/O Block

Each pin serviced by port A, B, C, D, E, F, G and H contains its individual I/O block. See Figure 5.

Figure 5. I/O Block

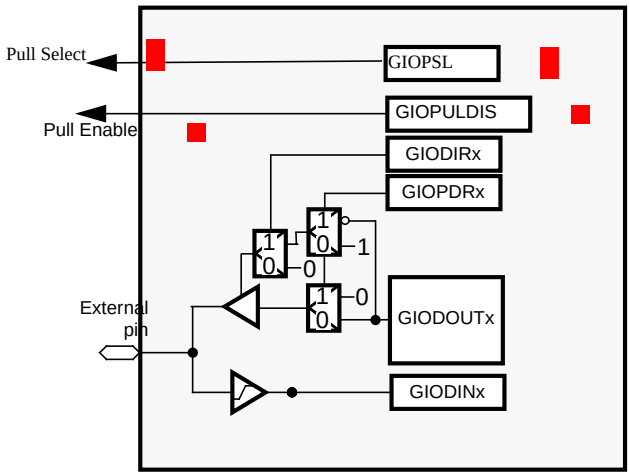


2.3.1 I/O Function

The GIO module sends data to the external pin through the output buffer and receives data from the external pin through the input buffer (see Figure 6). The three registers are:

- GIODIRx – controls the direction that information is sent. The GIODIRx register determines whether or not values in the data output register are sent to the external pin. The input buffer is always enabled. Therefore, the information that is sent to the external pin is also received in the input buffer.
- GIODOUTx – controls what information is sent to the external pin.
- GIODINx – receives the information from the external pin.
- GIOPDRx - controls the open drain configuration of the pi.
- GIOPULDISx - Disable the Pull Control Capability at Pin.
- GIOPSLx - Select the Pull Type at Pin

Figure 6. I/O Buffers



A high voltage (V_{IH} or greater) applied to the pin causes a high value (1) in the data input register (GIODINx). When a low voltage (V_{IL} or less) is applied to the pin, the data input register reads a low value (0). The pull functionality for the pin can be enabled or disabled by writing a zero (0) or a one (1) to the corresponding bit of the GIPPULDISx register. Pull Down or Pull Up can be selected by writing a zero (0) or one (1) to the corresponding bit of the GIOPSLx register.

The output buffer can be enabled or disabled through the data direction register (GIODIRx). To enable the output buffer, the data direction register must be set to 1.

The open drain functionality for the pin can be enable or disabled by writing a one (1) or a zero (0) respectively in the corresponding bit of the open drain register GIOPDRx).

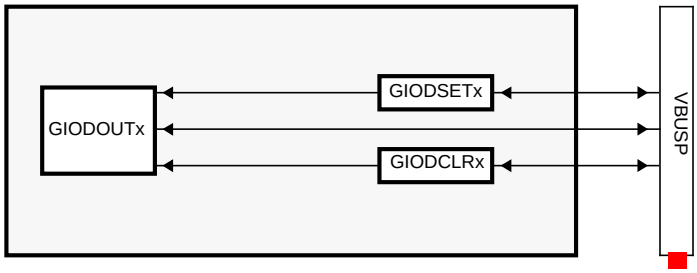
When the output buffer is enabled, writing values to the data output register (GIODOUTx) applies a voltage to the output pin. A low value (0) written to the data output register forces the pin to a low output voltage (V_{OL} or lower). A high value (1) written to the data output register forces the pin to a high output voltage (V_{OH} or higher) or to a high impedence state (z), depending on whether the GIOPDR bit is cleared or set, respectively.

2.3.2 Output Control Registers

When the output buffer is enabled, the value in the data output register (GIODOUTx) specifies the voltage applied to the external pin. The GIO

module provides three ways of communicating with the data output control register (see Figure 7).

Figure 7. Communication With the Data Output Register



- The control register bit can be written directly by moving an appropriate value to the data output register. A low value (0) written to the data output register forces the pin to a low output voltage (V_{OL} or lower), whereas a high value (1) written to the data output register forces the pin to a high output voltage (V_{OH} or higher) or to a high impedence state, depending on the value in the GIOPDRx bit.

- The data output register bit can be set to 1 by using the data set register (GIODSETx). A low value (0) written to the data set register does not affect the value in the data output register. A high value (1) written to the data set register sets the data output register (GIODOUTx) bit high (1).

- The data output register bit can be cleared to 0 by using the data clear register (GIODCLRx). A low value (0) written to the data clear register does not affect the value in the data output register. A high value (1) written to the data clear register clears (0) the corresponding bit in the data output register (GIODOUTx).

  .

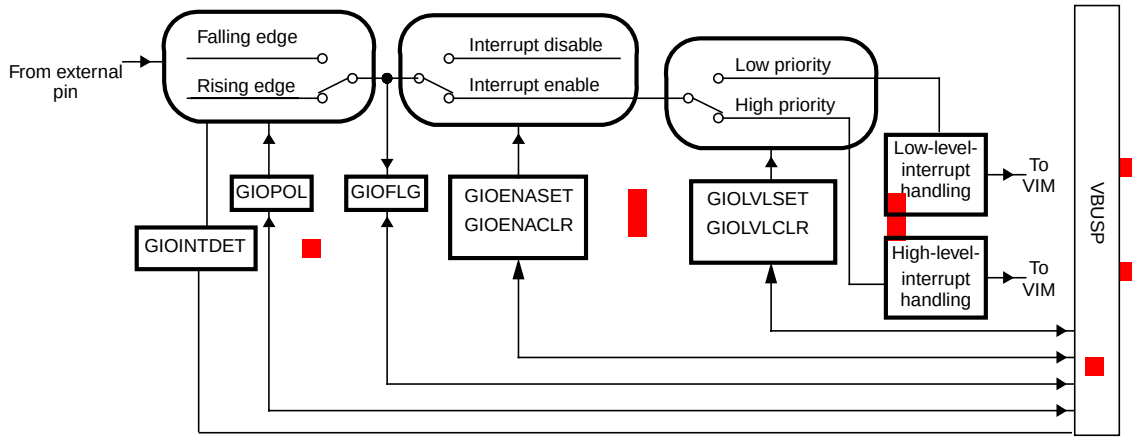
  .

  The GIODSETx and GIODCLRx registers allow improved handling of data. The data set and data clear registers remove any possibility of a read-modify-write (RMW) operation. RMW is possible when the CPU reads a register, performs some action (for example, an OR operation), and then writes the values back into the register. It is possible that the contents of the original register (GIODOUTx) can change (for example, an interrupt procedure) between the time when the CPU originally reads the register and the time when the CPU writes the new value. In this case, the new value written by the CPU overwrites the existing value, and the overwritten value is lost.

2.4 External Interrupt Block

Each interrupt-capable pin connects to the GIO module's single low-level-interrupt-handling block and single high-level-interrupt-handling block. Depending on the priority, the interrupt signal is sent through the appropriate offset register to the vectored interrupt manager (VIM) in the system module (see section 2.4.3, *High-Level-Interrupt Block and Low-Level-Interrupt Block*, page 16). Figure 8 shows the external interrupt block.

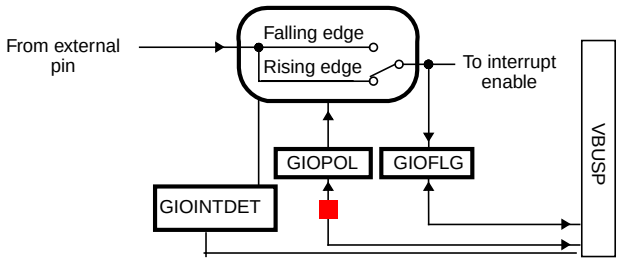
Figure 8. External Interrupt Block



2.4.1 Edge Detection and the Flag Register

The edge-detection hardware and flag register, like the input buffer, are always enabled. The GIOINTDET register specifies interrupt detection on both edge (if it is set). IF it is clear then the GIOPOL register (see Figure 9) specifies that the edge-detect flag is set on either the rising or falling edge.

Figure 9. Edge Detection and Flag Register



A rising edge occurs when the voltage on a given pin changes from a low value (V_{IL} or lower) to a high value (V_{IH} or higher). The voltage on the external pin must remain at the high level for at least one ICLK cycle to ensure recognition. (For an explanation of ICLK, see the clock portion of the *TMS470 System Module*, literature number SPNU189.)

A falling edge occurs when the voltage on the external pin changes from a high value (V_{IH} or higher) to a low value (V_{IL} or lower). The voltage on the external pin must remain at the low level for at least one ICLK cycle to ensure recognition. (GIOPOL behaves differently in a power-down state. See section 3.3, *Power-Down Modes*, page 23, for more information.)

If GIOINTDET is set, interrupt flag is set on both edge (rising as well as falling). If GIOINTDET is clear interrupt is sensed on either on rising or falling edge defined in POLARITY register as discussed above.

The corresponding flag in the GIOFLG register is set when a transition appearing on the external pin matches the edge chosen in the GIOPOL register. For example, to set the flag on a rising edge on pin 2 of GIO port A, set the bit in the polarity register (GIOPOL.2 = 1). Then, when the signal transition takes place, the GIO module will set the appropriate flag in the flag register (GIOFLG.2 = 1).

Note: Setting Flag With Interrupt Disabled

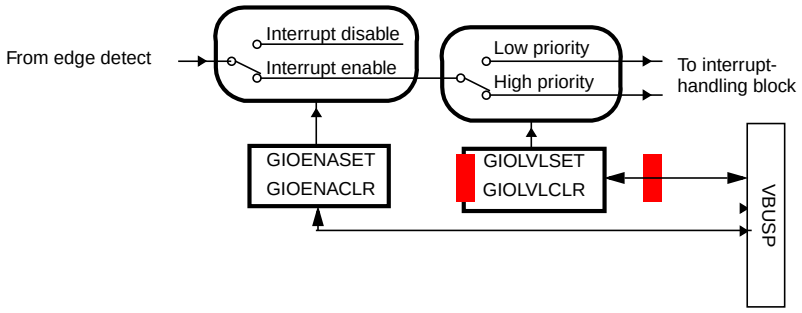
The flag can be set regardless of whether or not the interrupt is enabled. The flag register can, then, be polled instead of driving an interrupt. Additionally, you should ensure that the flag is not set prior to enabling the interrupt; specifically, you should clear the flag register before enabling the interrupt.

The edge-detection hardware responds to voltages on the external pin and does not discriminate between the source of these voltages. Therefore, if the output is enabled, the interrupt will respond to the correct edge though it is generated by the output buffer. Typically, the output buffer should be disabled when the flag is used.

2.4.2 Interrupts and Interrupt Levels

The interrupt flag is set when a transition on external pin if GIOINTDET is set else flag is set when a transition on the external pin matches the edge chosen in the GIOPOL register. An interrupt can be generated from the set flag if the interrupt is enabled (see Figure 10).

Figure 10. Interrupt Enable and Priority Level



The external interrupt can be enabled or disabled by writing a one (1) into the GIOENASET and GIOENACLR register bits respectively. Writing a zero (0) into these registers will have no effect. These two registers will be physically implemented as a single register (GIOENA). If the interrupt is enabled, the signal with an appropriate edge leads to an interrupt. If multiple interrupts occur simultaneously, the GIO module must prioritize the interrupts so that they can be handled in order.

The order in which simultaneous GIO interrupts are processed is determined by the following criteria:

1) The GIO priority control registers (GIOLVLSET and GIOLVLCLR) provide a software-implemented prioritization scheme. Each pin can be set as either a high-priority or low-priority interrupt. Interrupts with a high priority are serviced before ones with a low priority.
2) The handling of interrupts with the same priority is determined by the interrupt with the lowest bit value which has the highest priority. This prioritization is hardwired into the module.

When multiple interrupts occur simultaneously (see Example 1), the GIO module sends the lowest bit (the highest priority based on the first criterion) for each level (the second criterion) to the vectored interrupt manager (VIM) in the system module for processing. (For additional information on the VIM, see the *TMS470 System Module*, literature number SPNU___.)

Note: Wakeup Condition

GIOA Interrupts are also used to awaken the device from halt and standby modes.

Example 1. Determining Interrupt Priority

Pin	3	2	1	0
Priority	High	Low	High	Low
Order	2	4	1	3

In Example 1, four interrupts occur simultaneously on GIO port A, pins 3–0 and are handled as described in the following:

1) The interrupts on pin 1 and pin 0 are both sent to the VIM for servicing because they have the lowest values among the high-level and low-level interrupts. Assuming that the VIM is set to service the high priority before low priority, pin 1 is serviced first because it is the high-priority interrupt.
2) Next, the interrupts on pin 3 and pin 0 are sent to the VIM for servicing. The interrupt on pin 3 is serviced because it is the only remaining high-priority interrupt.
3) The interrupt on pin 0 is serviced third because it has the lowest bit value.
4) The interrupt on pin 2 is serviced last because it is the only remaining interrupt.

2.4.3 High-Level-Interrupt Block and Low-Level-Interrupt Block

The interrupt-handling blocks each contain two registers — an offset register and an emulation register. The high-level interrupts are denoted as level A and consequently, the registers within the high-level-interrupt-handling block are GIOOFFA/GIOOFFSET0 and GIOEMUA/GIOMON0 (see Figure 11). Likewise, the registers within the low-level-interrupt-handling block are GIOOFFB/GIOOFFSET1 and GIOEMUB/GIOMON1 (Figure 12).

Figure 11. High-Level-Interrupt-Handling Block

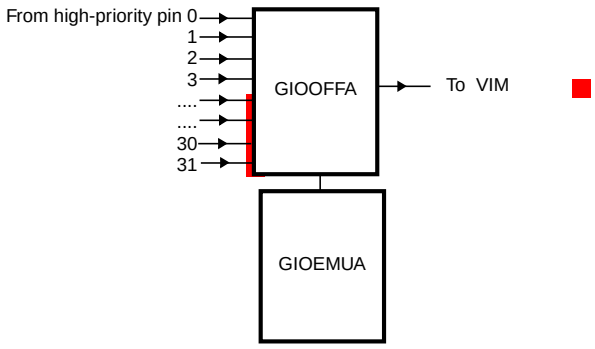
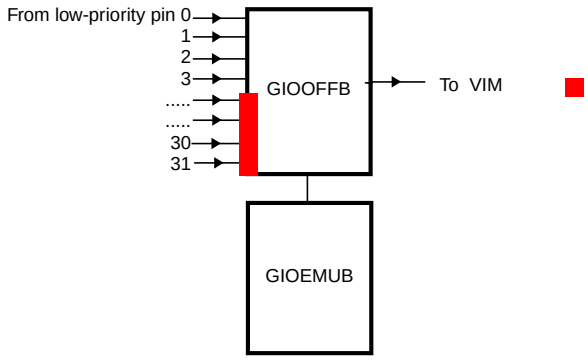


Figure 12. Low-Level-Interrupt-Handling Block



The read-only registers, GIOOFFA/GIOOFFSET0 and GIOOFFB/ GIOOFFSET1, generate a numerical offset value that represents the pending external interrupt (see Table 2). The offset can be used to locate the position of the interrupt routine in the vector table. A read of the offset register clears the offset register and the corresponding flag bit in the GIOFLG register.

The high-level offset register, GIOOFFA/GIOOFFSET0, receives signals from each active interrupt that is configured as high priority. The GIOOFFA/ GIOOFFSET0 displays the high-level interrupt with the highest priority (that is, the interrupt that was generated by the lowest bit in the flag register). Similarly, the GIOOFFB/GIOOFFSET1 displays the low-level interrupt with the highest priority.

The emulation control registers (GIOEMUA/GIOMON0 and GIOEMUB/ GIOMON0) mirror the offset registers. The emulation registers contain a numerical offset value that represents the pending external interrupt (see Table 2). A read of the emulation registers does not clear any register or any bit. These registers allow the device emulator to read and display the offset register values without affecting interrupt execution.

Table 2. GIO Offset Table †

GIOOFF .5	GIOOFF .4	GIOOFF .3	GIOOFF .2	GIOOFF .1	GIOOFF .0	Pending External Interrupt
0	0	0	0	0	0	No interrupt
0	0	0	0	0	1	Interrupt 0
0	0	0	0	1	0	Interrupt 1
0	0	0	0	1	1	Interrupt 2
0	0	0	1	0	0	Interrupt 3
0	1	1	1	1	0	Interrupt 29
0	1	1	1	1	1	Interrupt 30
1	0	0	0	0	0	Interrupt 31

† The interrupt values in this table are the same for GIO offset A and GIO offset B.

Example 2 illustrates how to identify the interrupt being serviced. This example assumes that interrupts 0 and 2 occur simultaneously (both low priority) on a device with three interrupt-capable pins. Reading the flag control register, GIOFLG, returns a value of xxxxx101. The first five values are indeterminate. Reading the offset B control register, GIOOFFB, returns a value of 00000001, indicating that the interrupt on pin 0 is being processed (see Table 2).

Example 2. Reading the Offset Register to Determine Serviced Interrupt

Bits	7	6	5	4	3	2	1	0
GIOENA	Reserved					1	1	1
GIOPOL	Reserved					1	0	0
GIOFLG	Reserved					1	0	1
GIOPRY	Reserved					0	0	0

Bits	7	6	5	4	3	2	1	0
GIOOFFA	0	0	0	0	0	0	0	0
GIOEMUA	0	0	0	0	0	0	0	0
GIOOFFB	0	0	0	0	0	0	0	1
GIOEMUB	0	0	0	0	0	0	0	1

2.4.4 Special Considerations for Interrupts

Please note that interrupts are subject to the following special considerations:

- On devices where fewer than thirty two interrupts are available, the unused control register bits are reserved.
- To use the enabled pins as interrupts, the I/O function of the pins is typically set as input. If the pin's I/O function is set as output, the signal feeds directly into the input. In this case, interrupts are only generated when the device toggles the data output register, thereby creating the appropriate interrupt edge. (See Example , *Eg.6 Reading Port B Input Register*, on page 64.)
- The interrupt flag can be set even though the interrupt is not enabled. Therefore, you must clear the flag register before enabling the interrupts to ensure that a spurious interrupt is not generated. (See Example , *Eg.5 Clearing Interrupt Flags and Setting Interrupts*, on page 63.)
- If interrupts are enabled when GIODIN0 is read, the least significant bits (those bits corresponding to the enabled interrupts) must be masked to avoid ambiguous results. For example, if pins 2:0 are configured as interrupts and pins 7:3 are configured as inputs (V_{IH} applied), then a read of GIODIN0 will read 11111xxx, where the x values are interrupt levels and not inputs. Mask the input register against (in this case) 11111000. (See Example 3, *Setting Interrupts and Configuring Pins for Output*, on page 59.)

3 Device Modes of Operation

The GIO module behaves differently in different modes of operation. Three main operating modes are discussed:

- Operating modes
  - Normal (user) mode
  - Privilege modes
- Emulation mode
- Power-down mode

3.1 Operating Modes

Operating modes can be broken down into two subgroups: normal mode and privilege modes.

3.1.1 Normal (User) Mode

Most application programs operate within user mode. The device can enter into normal mode in one of two ways:

- ❏ Drive the $\overline{\text{TRANS}}$ signal low.
- ❏ Load the binary value 10000 into control bits of the current program status register (CPSR), CPSR.4:0 = 10000b.

3.1.2 Privilege Modes

Privilege modes encompass the modes in which the device handles interrupts, aborts, and supervisor mode. The device can enter privilege modes in one of two ways:

- ❏ Drive the $\overline{\text{TRANS}}$ signal high.
- ❏ Load the current program status register control bits accordingly. See *TMS470R1x: 32-Bit RISC Microcontroller Family User's Guide* (SPNU134A) for more details.

3.2 Emulation Mode

Emulation mode is used by debugger tools to stop the CPU at breakpoints in order to read registers. When the device is in emulation mode, it pulls the SUSPEND signal high.

Note: Emulation Mode

Emulation mode is separate from the GIO emulation registers (GIOEMUA/GIOMON0 and GIOEMUB/GIOMON1). Emulation mode is a mode of operation of the device.

During emulation mode:

- External interrupts are not captured because the VIM is unable to service interrupts.
- Any register can be read without affecting the state of the system.
- A write to a register affects the state of the system.

3.3 Power-Down Modes

The GIO module has two power-down modes: module-level power down and device-level power down.

3.3.1 Module-Level Power Down

The GIO module can be placed into a power-down state (GIOPWDN.0 = 1) only in privilege mode. As a safety feature, the peripheral power-down override in the system module must be set (CLKCNTL.7 = 1) as well as the POWERDOWN bit in the GIO module.

In the power-down state, the clock signal to the GIO module is disabled. Thus, there is no switching, and the only current draw comes from leakage current. In low-power mode, interrupt pins become level-sensitive rather than edge-sensitive. The polarity bit changes function from falling edge and rising edge to low and high. A corresponding level on an interrupt pin pulls the module out of low-power mode.

3.3.2 Device-Level Power Down

The entire device can be placed in low-power mode. When the device is switched to a low-power mode (halt or standby), the clocking to all peripheral modules is stopped. In low-power mode, interrupt pins become level-sensitive rather than edge-sensitive, and a corresponding level on an interrupt pin pulls the device out of low-power mode.

4 Pullup/Pulldown Function

GIO module pins can have either an active pullup or active pulldown that makes it possible to leave the pins unconnected externally. The pins can be programmed to have the active pull capability by writing a zero (0) to the corresponding bit in the GIOPULDISx register. Writing a one (1) to the corresponding bit in the GIOPULDISx register disables the active pull functionality of the pin.

The pins can be programmed to have active pull up capability by writing One (1) to the corresponding bit in GIOPSLx register. Writing zero (0) will activate Pull Down capability. This pull up/down are active on pin only when corresponding bit is zero (0) in GIOPULDISx register.

The pullup/pulldown is deactivated when a bidirectional pin is configured as an output. At system reset, the pullup functionality of all the pins are enabled. Please see the specific data sheet for the current supplied by the pullup/ pulldown.

5 Open Drain Function

- The GIO pin can be configured to include an open drain functionality when they are configured as output pins. This is done by writing a one (1) into the corresponding bit of the GIOPDRx register. When the open drain functionality is enabled (GIOPDR.x = 1), a low value (0) written to the data output register forces the pin to a low output voltage (V_{OL} or lower), whereas a high value (1) written to the data output register forces the pin to a high impedence state. The open drain functionality is disabled when the pin is configured as an input pin.

6 GIO Control Registers

Table shows the GIO register arrangement. It assumes 8 ports, including 4 interrupt capable ports, with 8 pins per port. These registers are accessible in 8-, 16-, and 32-bit reads or writes. Consult the device-specific data sheet to verify the pin configuration.

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x00	GIOGCR0	Reserved															
		Reserved															RESET
0x04	GIOPWDN	Reserved															
		Reserved															PWDN
0x08	GIOINT-DET	GIOINTDET 3								GIOINTDET 2							
		GIOINTDET 1								GIOINTDET 0							
0x0C	GIOPOL	GIOPOL 3								GIOPOL 2							
		GIOPOL 1								GIOPOL 0							
0x10	GIOENA-SET	GIOENASET 3								GIOENASET 2							
		GIOENASET 1								GIOENASET 0							
0x14	GIOENA-CLR	GIOENACLR 3								GIOENACLR 2							
		GIOENACLR 1								GIOENACLR 0							
0x18	GIOLVL SET	GIOLVLSET 3								GIOLVLSET 2							
		GIOLVLSET 1								GIOLVLSET 0							

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x1C	GIOLVL-CLR	GIOLVLCLR 3								GIOLVLCLR 2							
		GIOLVLCLR 1								GIOENACLR 0							
0x20	GIOFLG	GIOFLG 3								GIOFLG 2							
		GIOFLG 1								GIOFLG 0							
0x24	GIOAFFSET0	Reserved															
		Reserved										GIOOFFSET0					
0x28	GIOAFFSET1	Reserved															
		Reserved										GIOOFFSET1					
0x2C	GIOMON0	Reserved															
		Reserved										GIOMON0					
0x30	GIOMON1	Reserved															
		Reserved										GIOMON1					
0x34	GIODIR0	Reserved															
		Reserved								GIODIR0							

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x38	GIODIN0	Reserved															
		Reserved								GIODIN0							
0x3C	GIODOUT0	Reserved															
		Reserved								GIODOUT0							
0x40	GIOSET0	Reserved															
		Reserved								GIOSET0							
0x44	GIOCLR0	Reserved															
		Reserved								GIOCLR0							
0x48	GIOPDR0	Reserved															
		Reserved								GIOPDR0							
0x4C	GIOPUL DIS0	Reserved															
		Reserved								GIOPULDIS0							
0x50	GIOPSl0	Reserved															
		Reserved								GIOPSL0							

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x54	GIODIR1	Reserved															R
		Reserved								GIODIR1							R
0x58	GIODIN1	Reserved															R
		Reserved								GIODIN1							R
0x5C	GIODOUT1	Reserved															R
		Reserved								GIODOUT1							R
0x60	GIOSET1	Reserved															R
		Reserved								GIOSET1							R
0x64	GIOCLR1	Reserved															R
		Reserved								GIOCLR1							R
0x68	GIOPDR1	Reserved															R
		Reserved								GIOPDR1							R
0x6C	GIOPULDIS1	Reserved															R
		Reserved								GIOPULDIS1							R
0x70	GIOPSL1	Reserved															R
		Reserved								GIOPSL1							R

Offset Address† / Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
...																
0x114 GIODIR7	Reserved															■
	Reserved								GIODIR7							■
0x118 GIODIN7	Reserved															■
	Reserved								GIODIN7							■
0x11C GIODOUT7	Reserved															■
	Reserved								GIODOUT7							■
0x120 GIOSET7	Reserved															■
	Reserved								GIOSET7							■
0x124 GIOCLR7	Reserved															■
	Reserved								GIOCLR7							■
0x128 GIOPDR7	Reserved															■
	Reserved								GIOPDR7							■
0x12C GIOPULDIS7	Reserved															■
	Reserved								GIOPULDIS7							■

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x130 GIOPSL7	Reserved															
	Reserved								GIOPSL7							

6.1 GIO Global Control Register (GIOGCR0)

The GIOGCR register contains one bit that controls the module reset status. Writing one (1) into this bit brings the module out of rest and writing zero (0) into this bit puts the module in the reset state. After system reset, this bit has to be set to one (1) before normal operations can begin on this module.

RP = Read in all modes /write in privilege mode only, -n = Value after reset (x = indeterminate)

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x00 GIOGCR0	Reserved															
	R -0															
	Reserved															RESET
	R - 0															RP-0

Bits 31–1 **Reserved.**

Read values are “0” and write values have no effect.

Bit 0 **GIOGCR0.** GIO global control bit.

Writing to the GIOGCR0 bit is only allowed in privilege mode. Reading of the GIOGCR0 bit is allowed in all modes.

Privilege mode (write):

1 = Normal operation
0 = Reset State

User mode (write):

Writes have no effect in user mode.

User or privilege mode (read):

1 = Normal operation
0 = Reset State

6.2 GIO Power Down (GIOPWDN)

The GIOPWDN register contains one bit that controls the module power-down status. In power-down mode, GIO internal clocks are stopped, which leaves the module in a static state where it consumes the lowest possible current.

RP = Read in all modes /write in privilege mode only, -n = Value after reset (x = indeterminate)

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x04 GIOPWDN	Reserved															
	R - 0															
	Reserved															PWDN
	R - 0															RP-0

Bits 31–1 **Reserved.**

Read values are “0” and write values have no effect.

Bit 0 **GIOPWDN.** GIO power down.

Writing to the GIOPWDN bit is only allowed in privilege mode. Reading of the GIOPWDN bit is allowed in all modes.

Privilege mode (write):

0 = Normal operation; clocks enabled to GIO module
1 = Power-down mode

User mode (write):

Writes have no effect in user mode.

User or privilege mode (read):

0 = Normal operation; clocks enabled to GIO module
1 = Power-down mode

Note: Power-Down Override

The clock control register (CLKCNTL) in the system module contains a peripheral power-down override bit (CLKCNTL.7 = 1) that must be set in order for the GIOPWDN bit to have any effect.

6.3 GIO Interrupt Detect (GIOINTDET)

The GIODETINT register provides the flexibility to recognise the interrupt at both edge if it is set else interrupt is recognised on risingt edge or falling edge depends upon the POLARITY register —that sets the flag. In order to ensure recognition of the signal as an edge, the signal must maintain the new level for at least one ICLK cycle.

R = Read, W = Write, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x08 GIOINT-DET	GIOINTDET 3								GIOINTDET 2							
	RW-0								**RW-0**							
	GIOINTDET 1								GIOINTDET 0							
	RW-0								**RW-0**							

Bits 31–0 **GIOINTDET.** Interrupt detection select.

User or privilege mode (read/write):

0 = Flag set on falling/rising edge on the corresponding pin (either on rising or falling depends upon the polarity set in polarity register
1 = Flag set on rising and falling edge on the corresponding pin (Both edge)

6.4 GIO Interrupt Polarity (GIOPOL)

The GIOPOL register controls the polarity—rising edge (low to high) or falling edge (high to low)—that sets the flag. In order to ensure recognition of the signal as an edge, the signal must maintain the new level for at least one ICLK cycle.

R = Read, W = Write, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x0C GIOPOL	GIOPOL 3								GIOPOL 2							
	RW-0								**RW-0**							
	GIOPOL 1								GIOPOL 0							
	RW-0								**RW-0**							

Bits 31–0 **GIOPOL.** Interrupt polarity select.

User or privilege mode (read/write):

0 = Flag set on falling edge on the corresponding pin
1 = Flag set on rising edge on the corresponding pin

Low-power mode (halt or standby):

0 = Interrupt triggered on low level
1 = Interrupt triggered on high level

6.5 GIO Interrupt Enable (GIOENASET and GIOENACLR)

The GIOENASET and GIOENACLR register controls which interrupt-capable pins are configured as interrupts.

6.5.1 GIOENASET

Writing one (1) into the corresponding bit of the GIOENASET register enables the interrupt for that pin, writing a zero (0) has no effect.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x10	GIOENA-SET	GIOENASET 3								GIOENASET 2							
		RW-0								RW-0							
		GIOENASET 1								GIOENASET 0							
		RW-0								RW-0							

Bits 31–0 **GIOENASET.** Interrupt enable.

User or privilege mode (read/write):

0 = No Effect
1 = Interrupt enabled

Note: Enabling Interrupt at the Device Level

Two bits must be set within the VIM (Vectored interrupt manager) in the interrupt mask register (REQMASK). The REQMASK register must be configured to enable the appropriate interrupts. Additionally, the CPU must be configured to recognize interrupt requests.

6.5.2 GIOENACLR

Writing one (1) into the corresponding bit of the GIOENACLR register disables the interrupt for that pin, writing a zero (0) has no effect.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x14	GIOENA-CLR	GIOENACLR 3								GIOENACLR 2							
		RW-0								**RW-0**							
		GIOENACLR 1								GIOENACLR 0							
		RW-0								**RW-0**							

Bits 31–0 **GIOENACLR.** Interrupt disable.

User or privilege mode (read/write):

0 = No Effect
1 = Interrupt disabled

6.6 GIO Interrupt Priority (GIOLVLSET and GIOLVLCLR)

The GIOLVLSET and GIOLVLCLR registers configure the interrupts as high priority (A/0) or low priority (B/1). Each interrupt can be individually configured as high priority or low priority.

- ❏ The high-priority interrupts are recorded to GIOOFFA and GIOEMUA.
- ❏ The low-priority interrupts are recorded to GIOOFFB and GIOEMUB.

6.6.1 GIOLVLSET

The GIOLVLSET register is used to configure an interrupt as a high level or high priority interrupt. An interrupt can be configured as a high priority interrupt by writing a one (1) into the corresponding bit of the GIOLVLSET register. Writing a zero has no effect.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x18	GIOLV-LSET	GIOLVLSET 3								GIOLVLSET 2							
		RW-0								RW-0							
		GIOLVLSET 1								GIOLVLSET 0							
		RW-0								RW-0							

Bits 31–0 **GIOLVLSET.** GIO high priority interrupt.

User or privilege mode (read/write):

0 = No Effect
1 = Interrupt set as a high priority interrupt

6.6.2 GIOLVLCLR

The GIOLVLCLR register is used to configure an interrupt as a low level or low priority interrupt. An interrupt can be configured as a low priority interrupt by writing a one (1) into the corresponding bit of the GIOLVLCLR register. Writing a zero has no effect.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x1C	GIOLVL-CLR	GIOLVLCLR 3								GIOLVLCLR 2							
		RW-0								**RW-0**							
		GIOLVLCLR1								GIOLVLCLR 0							
		RW-0								**RW-0**							

Bits 31–0 **GIOLVLCLR.** GIO low priority interrupt.

User or privilege mode (read/write):

0 = No Effect
1 = Interrupt set as a low priority interrupt

6.7 GIO Interrupt Flag (GIOFLG)

The GIOFLG register contains flags indicating that the transition edge (type set in GIOPOL) has occurred. The flag is also cleared by reading the appropriate offset register. (See section 2.4.3, *High-Level-Interrupt Block and Low-Level-Interrupt Block*, page 16.)

R = Read, C = Write clears the bit, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x20 GIOFLG	GIOFLG 3								GIOFLG 2							
	RC-0								RC-0							
	GIOFLG1								GIOFLG 0							
	RC-0								RC-0							

Bits 31–0 **GIOFLG.** GIO flag.

When the GIOFLG bit is set to 1, the selected transition on the corresponding pin has occurred.

User or privilege mode (read):

0 = Transition has not occurred since last clear
1 = Transition has occurred since last clear

User or privilege mode (write):

0 = No changes in flag register
1 = Clears the corresponding bit to 0

6.8 GIO Offset A (GIOOFFA/GIOOFFSET0)

The GIOOFFA / GIOOFFSET0 register provides a numerical offset value that represents the pending external interrupt with high priority, as shown in Table 3. The offset value can be used to locate the position of the interrupt routine in a vector table. Reading this register clears it and the corresponding flag bit in the GIOFLG register. However, in emulation mode, a read to this register does not clear the corresponding flag bit.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x24	GIOOFFSET0/ GIOOFF-SETA	Reserved								Reserved							
		R-0								R-0							
		Reserved										GIOOFFSET0					
		R-0								R-0	R-0	R - 0					

Bits 31–6 **Reserved.**

Always read 0. Writes to these bits have no effect.

Bits 5–0 **GIOOFFA.** GIO offset A.

GIOOFFA.5:0 index the currently pending high-priority interrupt. Bit values and the interrupts are listed in Table 3.

- Any mode (write):

  Writes to these bits have no effect.

- User or privilege mode (read):

  A read of these bits determines the pending external interrupt; this register and the flag bit (in the GIOFLG register) are also cleared.

- Emulation mode (read):

  A read of these bits determines the pending external interrupt, but the corresponding flag bit is not cleared.

Table 3. External Interrupt With a High Priority

GIOOFFA 5	GIOOFFA 4	GIOOFFA 3	GIOOFFA 2	GIOOFFA 1	GIOOFFA 0	Pending External Interrupt
0	0	0	0	0	0	No interrupt
0	0	0	0	0	1	Interrupt 0
0	0	0	0	1	0	Interrupt 1
0	0	0	0	1	1	Interrupt 2
0	0	0	1	0	0	Interrupt 3
0	1	1	1	1	0	Interrupt 29
0	1	1	1	1	1	Interrupt 30
1	0	0	0	0	0	Interrupt 31

6.9 GIO Emulation A (GIOEMUA)

The GIOEMUA register, a read-only register, is provided for use by the debug monitor in normal operation. The contents of this register are identical to the contents of GIOOFFA. The corresponding flag in the GIOFLG register is not cleared when the GIOEMUA register is read.

R = Read, W = Write, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x2C GIOEMUA/ GIOMON0	Reserved								Reserved							
	R-0								R-0							
	Reserved										GIOMON0					
	R-0								R-0	R-0	R - 0					

Bits 31–6 **Reserved.**

Always read 0. Writes to these bits have no effect.

Bits 5–0 **GIOEMUA.** GIO emulation A.

GIOEMUA.5:0 index the currently pending high-priority interrupt. Bit values and the interrupts are listed in Table 3.

❏ Any mode (write):

Writes to these bits have no effect.

❏ User or privilege mode (read):

A read of these bits determines the pending external interrupt.

6.10 GIO Offset B (GIOOFFB)

The GIOOFFB register provides a numerical offset value that represents the pending external interrupt with low priority, as shown in Table 4. The offset value can be used to locate the position of the interrupt routine in a vector table. Reading this register clears it and the corresponding flag bit in the GIOFLG register. However, in emulation mode, a read to this register does not clear the corresponding flag bit.

R = Read, W = Write, -n = Value after reset

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x28	GIOOFFSET1/ GIOOFF-SETB	Reserved								Reserved							
		R-0								R-0							
		Reserved										GIOOFFSET1					
		R-0								R-0	R-0	R - 0					

Bits 31–6 **Reserved.**

Always read 0. Writes to these bits have no effect.

Bits 5–0 **GIOOFFB.** GIO offset B.

GIOOFFB.5:0 index the currently pending low-priority interrupt. Bit values and the interrupts are listed in Table 4.

- Any mode (write):

  Writes to these bits have no effect.

- User or privilege mode (read):

  A read of these bits determines the pending external interrupt; this register and the flag bit (in the GIOFLG register) are also cleared.

- Emulation mode (read):

A read of these bits determines the pending external interrupt, but the corresponding flag bit is not cleared.

Table 4. External Interrupt With a Low Priority

GIOOFFB 5	GIOOFFB 4	GIOOFFB 3	GIOOFFB 2	GIOOFFB 1	GIOOFFB 0	Pending External Interrupt
0	0	0	0	0	0	No interrupt
0	0	0	0	0	1	Interrupt 0
0	0	0	0	1	0	Interrupt 1
0	0	0	0	1	1	Interrupt 2
0	0	0	1	0	0	Interrupt 3
0	1	1	1	1	0	Interrupt 29
0	1	1	1	1	1	Interrupt 30
1	0	0	0	0	0	Interrupt 31

6.11 GIO Emulation B (GIOEMUB)

The GIOEMUB register, a read-only register, address is provided for use by the debug monitor in normal operation. The contents of this register are identical to the contents of GIOOFFB. The corresponding flag in the GIOFLG register is not cleared when the GIOEMUB register is read.

R = Read, W = Write, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x30 GIOEMUB/ GIOMON1	Reserved								Reserved							
	R-0								R-0							
	Reserved										GIOMON1					
	R-0								R-0	R-0	R - 0					

Bits 31–6 **Reserved.**

Always read 0. Writes to these bits have no effect.

Bits 5–0 **GIOEMUB.** GIO emulation B.

GIOEMUB. 5:0 index the currently pending low-priority interrupt. Bit values and the interrupts are listed in Table 4.

- Any mode (write):

  Writes to these bits have no effect.

- User or privilege mode (read):

  A read of these bits determines the pending external interrupt.

6.12 GIO Data Direction (GIODIRx)

The GIODIRx registers control whether the pins of a given port are configured as inputs or outputs.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x34 0x54 0x74 0x94	GIODIRx	Reserved								Reserved							
		R-0								R-0							
0xB4 0xD4 0xF4 0x114		Reserved								GIODIRx							
		R-0								R - 0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIODIRx. GIO data direction.

User or privilege mode (read/write):

0 = Output buffer disabled
1 = Output buffer enabled

6.13 GIO Data Input (GIODINx)

Values in the GIODINx registers reflect the current state (high = 1 or low = 0) on the pins.

R = Read, W = Write, -n = Value after reset

Offset Address† / Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x38 0x58 0x78 0x98 GIODINx	Reserved								Reserved							
	R-0								R-0							
0xB8 0xD8 0xF8 0x118	Reserved								GIODINx							
	R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIODINx. GIO data input.

User or privilege mode (read):

0 = Logic low (input voltage is V_{IL} or lower)
1 = Logic high (input voltage is V_{IH} or higher)

❑ User or privilege mode (write):

Writes to these bits have no effect.

6.14 GIO Data Output (GIODOUTx)

Values in the GIODOUTx register specify the output state (high = 1 or low = 0) of the pins if the pins are configured as outputs.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x3C 0x5C 0x7C 0x9C	GIODOUTx	Reserved								Reserved							
		R-0								R-0							
0xBC 0xDC 0xFC 0x11C		Reserved								GIODOUTx							
		R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIODOUTx. GIO data output.

User or privilege mode (read/write):

0 = Logic low (output voltage is V_{OL} or lower)
1 = Logic high (output voltage is V_{OH} or higher if GIOPDRx bit = 0 and output is in high impedence state if GIOPDRx bit = 1)

6.15 GIO Data Set (GIODSETx)

Values in the GIODSETx register set the data output control register bits to 1 regardless of the current value in the GIODOUTx bits.

R = Read, W = Write, -n = Value after reset

Offset Address† Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x40 0x60 0x80 0xA0 GIODSETx	Reserved								Reserved							
	R-0								R-0							
0xC0 0xE0 0x100 0x120	Reserved								GIODSETx							
	R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 **GIODSETx.** GIO data set.

User or privilege mode (read): GIODSETx reflects the contents of GIOOUTx.

0 = Logic low (output voltage is V_{OL} or lower)
1 = Logic high (output voltage is V_{OH} or higher)

User or privilege mode (write):

0 = Leaves the corresponding bit in GIODOUTx unchanged
1 = Sets the corresponding bit in GIODOUTx to 1

6.16 GIO Data Clear (GIODCLRx)

Values in the GIODCLRx register clear the data output register bit to 0 regardless of its current value.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x44 0x64 0x84 0xA4	GIODCLRx	Reserved								Reserved							
		R-0								R-0							
0xC4 0xE4 0x104 0x124		Reserved								GIODCLRx							
		R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIODCLRx. GIO data clear.

User or privilege mode (read): GIODCLRx reflects the contents of GIOOUTx.

0 = Logic low (output voltage is V_{OL} or lower)
1 = Logic high (output voltage is V_{OH} or higher)

User or privilege mode (write):

0 = Leaves the corresponding bit in GIODOUTx unchanged
1 = Clears the corresponding bit in GIODOUTx to 0

6.17 GIO Open Drain (GIOPDRx)

A value of one (1) in the GIODPDRx register configure the data pin for open drain capability. A value of zero (0) disables this open drain capability of the pin.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x48 0x68 0x88 0xA8	GIOPDRx	Reserved								Reserved							
		R-0								R-0							
0x C8 0x E8 0x104 0x124		Reserved								GIOPDRx							
		R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIOPDRx. GIO Open Drain.

User or privilege mode (read/write):

0 = Logic low . Open drain functionality isdisabled.
(output voltage is V_{OL} or lower if GIODOUT =0 and V_{OH} or higher if GIODOUT =1).
1 = Logic high . Open drain functionality is enabled.
(output voltage V_{OL} or lower if GIODOUT =0 and z if GIODOUT =1).

6.18 GIO Pull Disable(GIOPULDISx)

Values in the GIOPULDISx register enable or disable the pull control capability of the pins. A zero (0) in the GIOPULDISx register enables the pull functionality of the corresponding pin, while a one (1) disables this pullup functionality. R = Read, W = Write, -n = Value after reset depends upon the device

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x4C 0x6C 0x8C 0xAC	GIOPULD-ISx	Reserved								Reserved							
		R-0								R-0							
0xCC 0xEC 0x10C 0x12C		Reserved								GIOPUSDISx							
		R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives “0” and a write has no effect.

Bits 7–0 GIOPULDISx. GIO pull Disable

User or privilege mode (read/write):

0 = Logic low (pull functionality is enabled)
1 = Logic high (pull functionality is disabled)

6.19 GIO Pull Select(GIOPSLx)

Values in the GIOPSLx register Select the pull up or down functianality of pins. A zero (0) in the GIOPSLx register enables the pull up functionality of the corresponding pin, while a one (1) enable pull up functionality of the corresponding pin.

R = Read, W = Write, -n = Value after reset

Offset Address† Register		31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x50 0x70 0x90 0xB0	GIOPSLx	Reserved								Reserved							
		R-0								R-0							
0xD0 0xF0 0x110 0x130		Reserved								GIOPSLx							
		R-0								RW-0							

Bits 31-8 Reserved

Read of thses bits gives an undeterminate value and a write has no effect.

Bits 7–0 GIOPSLx. GIO Pull Select

User or privilege mode (read/write):

0 = Logic low (pull down functionality is enabled)
1 = Logic high (pull up functionality is enabled)

7 Applications

The application examples in this section assume a typical configuration of five I/O functional ports, in which only Port A has three external interrupt-capable pins, 2-0. Table 5 illustrates the GIO register.

Table 5. Example GIO Register Showing Reserved Bits

Register	Address	7	6	5	4	3	2	1	0
GIOGCR	0x00								
GIOPWDN	0x04								
GIOINTDET	0x08								
GIOPOL	0x0C								
GIOENASET	0x10								
GIOENACLR	0x14								
GIOPRYSET	0x18								
GIOPRYCLR	0x1C								
GIOFLG	0x20								
GIOOFFA	0x24								
GIOOFFB	0x28								
GIOEMUA	0x2C								
GIOEMUB	0x30								
GIODIRA	0x34								
GIODINA	0x38								
GIODOUTA	0x3C								
GIODSETA	0x40								
GIODCLRA	0x44								
GIOPDRA	0x48								
GIOPULDISA	0x4C								
GIOPSLA	0x50								
GIODIRB	0x54								
GIODINB	0x58								
GIODOUTB	0x5C								
GIODSETB	0x60								
GIODCLRB	0x64								
GIOPDRB	0x68								
GIOPULDISB	0x6C								
GIOPSLB	0x70								
GIODIRC	0x74								
GIODINC	0x78								
GIODOUTC	0x7C								
GIODSETC	0x80								
GIODCLRC	0x84								
GIOPDRC	0x88								
GIOPULDISC	0x8C								
GIOPSLC	0x90								
GIODIRD	0x94								
GIODIND	0x98								
GIODOUTD	0x9C								
GIODSETD	0xA0								
GIODCLRD	0xA4								
GIOPDRD	0xA8								
GIOPULDISD	0xAC								

GIOPSLD	0xB0								
GIODIRE	0xB4								
GIODINE	0xB8								
GIODOUTE	0xBC								
GIODSETE	0xC0								
GIODCLRE	0xC4								
GIOPDRE	0xC8								
GIOPULDISE	0xCC								
GIOPSLE	0xD0								
GIODIRF	0xD4								
GIODINF	0xD8								
GIODOUTF	0xDC								
GIODSETF	0xE0								
GIODCLRF	0xE4								
GIOPDRF	0xE8								
GIOPULDISF	0xEC								
GIOPSLF	0xF0								
GIODIRG	0xF4								
GIODING	0xF8								
GIODOUTG	0xFC								
GIODSETG	0x100								
GIODCLRG	0x104								
GIOPDRG	0x108								
GIOPULDISG	0x10C								
GIOPSLG	0x110								
GIODIRH	0x114								
GIODINH	0x118								
GIODOUTH	0x11C								
GIODSETH	0x120								
GIODCLRH	0x124								
GIOPDRH	0x128								
GIOPULDISH	0x12C								
GIOPSLH	0x130								

Note:

The code in Example 3 demonstrates how to set the interrupts. In the example, two of the interrupts are enabled, and the third pin of port A is configured for output. R2 keeps the same value, and the other registers act as temporary storage for addresses and values.

Example 3. Setting Interrupts and Configuring Pins for Output

```
GIO_LOC     .word 0xFFF7EC00     ;device specific address for the GIO registers.
GIOENASET .equ    0x0C           ;setting up equate statements
GIOPOL .equ    0x08
GIOFLG .equ    0x1C
GIOPRYSET .equ    0x14
GIOPRYCLR .equ  0x18
GIOOFFA .equ    0x20
GIOEMUA .equ    0x28
GIOOFFB .equ    0x24
GIOEMUB .equ    0x2C
GIODIRA .equ    0x30
GIODINA .equ    0x34
GIODOUTA .equ    0x38
GIODSETA .equ    0x3C
GIODCLRA .equ    0x40
GIOPDRA .equ   0x44
GIOPULDISA  .equ   0x48


    LDR R2, GIO_LOC              ;loads GIO base address into register 2.
                                 ; **SET THE POLARITY OF THE INTERRUPTS.**
    MOV R3, #GIOPOL              ; loads GIOPOL offset address into register 3.
    MOV R4, #0x02                ; loads a value of 0x02 into register 4.
                                 ; Sets bit 1.
    STR R4, [R2, R3]             ; sets interrupt 0 to trigger on falling edge
                                 ; and interrupt 1 to trigger on rising edge.
                                 ; **SET THE PRIORITY ON BOTH INTERRUPTS AS LOW**
    MOV R3, #GIOPRYCLR           ; loads the GIOPRY address into register 3.
    MOV R4, #0x03                ; loads a value of 0 into register 4.
    STR R4, [R2, R3]             ; priority on interrupts 0 and 1 is set LOW.
                                 ; ** SET PORT A FOR OUTPUT (EXCEPT FOR INTERRUPTS
                                 ; WHICH MUST BE CONFIGURED AS INPUTS)**
    MOV R3, #GIODIRA             ; loads GIODIRA offset address into register 3.
    MOV R4, #0xFC                ; loads a binary value of 11111100
    STR R4, [R2, R3]             ; sets pins 0 and 1 as input to insure proper
                                 ; interrupt behavior pin 2 configured as
                                 ; output
    MOV R3, #GIOFLG              ; loads GIOFLG offset address into register 3.
    MOV R4, #0xFF                ; sets 32 bits.
    STR R4, [R2, R3]             ; clears all bits of the flag register.
```

```
                    .
; **ENABLE THE FIRST TWO INTERRUPTS 1:0.**
MOV R3, #GIOENA ; loads GIOENA offset address into register 3.
MOV R4, #0x03 ; sets first 2 bits 1:0

STR R4, [R2, R3] ; enables the first two interrupts
```

The code in Example 4 demonstrates how to toggle the output buffer port B using the GIODSET and GIODCLR buffers.

EG. 4 Toggling Output Buffers

```
GIO_LOC .word 0xFFF7EC00          ; device specific address for the GIO registers.
GIOENA .equ    0x04               ;setting up equate statements
...
GIODIRB .equ 0x50
GIODINB .equ 0x54
GIODOUTB.equ 0x58
GIODSETB.equ 0x5C
GIODCLRB .equ 0x60
   LDR R2, GIO_LOC                ; loads absolute address of the GIO memory
                                  ; location (device specific).
                                  ; **SET ALL BITS IN GIODOUTB AS 1.**
   MOV R3, #GIODSETB              ; loads offset address of GIODSETB
   MOV R4, #0xFF                  ; loads a binary value of 11111111
   STR R4, [R2, R3]               ; sets all bits of GIODOUTB.
                                  ; **CONFIGURE PORT 1 AS OUTPUT.**
   MOV R3, #GIODIRB               ; loads offset address of GIODIRB.
   MOV R4, #0xFF                  ; loads a binary value of 11111111.
   STR R4, [R2, R3]               ; configures port 1 as output
                                  ; **TOGGLE BITS 0, 2, 4, 6.**
   MOV R3, #GIODSETB              ; loads offset address of GIODSETB
   MOV R4, #GIODCLRB              ; loads GIODCLRB offset address
   MOV R5, #0x55                  ; loads a binary value of 01010101
TOGGLE
   STR R5, [R2, R4]               ; clears GIODOUTB bits 0, 2, 4, 6
   STR R5, [R2, R3]               ; sets GIODOUTB bits 0, 2, 4, 6
   B TOGGLE                       ; loops back to TOGGLE
```

The code in Example demonstrates how interrupt flags should be cleared before interrupts are enabled. In this example, all three interrupt-capable pins are set as interrupts.

Eg.5 Clearing Interrupt Flags and Setting Interrupts

```
GIO_LOC.word 0xFFF7EC00          ; device specific address for GIO registers
GIOENASET .equ    0x0C           ;setting up equate statements
...
GIOFLG .equ 0x1C



   LDR R2, GIO_LOC               ;loads absolute address of the GIO memory
   MOV R3, #GIOFLG               ;loads GIOFLG offset address into R3.
   MOV R4, #0x07                 ;loads a value of 00000111 into R4.
   STR R4, [R2, R3]              ;clears the interrupt-capable bits of the
                                 ;GIOFLG control register.

   MOV R3, #GIOENA               ;loads GIOENA offset address into R3.
   MOV R4, #0x07                 ;loads 00000111 into R4.
   STR R4, [R2, R3]              ;enables pins 2:0 of port A as interrupts.
```

The code in Example demonstrates how to read the input register of port A when interrupts are enabled. Pin 0 is set as an interrupt, and pins 2 and 1 are configured as inputs.

Eg.6 Reading Port B Input Register

```
GIO_LOC.word 0xFFF7EC00          ;device specific address for GIO registers
GIOENA .equ    0x04              ;setting up equate statements
...
GIODIRB .equ 0x50
GIODINB .equ 0x54
GIODOUTB.equ 0x58
GIODSETB.equ 0x5C
GIODCLRB .equ 0x60

   LDR R2, GIO_LOC               ;loads absolute address of the GIO memory
                                 ;**MASK OUTPUTS AND INTERRUPTS SO INPUT IS UNAMBIGUOUS.*
   MOV R3, #GIODINB              ;loads the offset address of GIODINB.
   LDR R4, [R2, R3]              ;loads the value in GIODINB register into R4.
   MOV R3, #0xFE                 ;loads 11111110 into R3. This value is used
                                 ;to mask the input so that only the input
                                 ;values are read. The 1's appear in the places
                                 ;where the input register is reading input
                                 ;voltages.
   AND R4, R4, R3                ;loads masked input value into R4.
```