TMS320DM647/DM648 DSP General-Purpose Input/Output (GPIO)

User's Guide

Literature Number: SPRUEK7A
October 2007

Texas Instruments

Contents

List of Figures

List of Tables

Read This First

About This Manual

Describes the general-purpose input/output (GPIO) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.

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.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Note: Acronyms 3PSW, CPSW, CPSW_3G, and 3pGSw are interchangeable and all refer to the 3 port gigabit switch.

Related Documentation From Texas Instruments

The following documents describe the TMS320DM647/DM648 Digital Signal Processor (DSP). 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.

SPRS372 — ***TMS320DM647/DM648 Digital Media Processor Data Manual*** describes the signals, specifications and electrical characteristics of the device.

SPRU732 — ***TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide*** describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C64x and TMS320C64x+ digital signal processors (DSPs) of the TMS320C6000 DSP family. The C64x/C64x+ DSP generation comprises fixed-point devices in the C6000 DSP platform. The C64x+ DSP is an enhancement of the C64x DSP with added functionality and an expanded instruction set.

SPRUEK5 — ***TMS320DM647/DM648 DSP DDR2 Memory Controller User's Guide*** describes the DDR2 memory controller in the TMS320DM647/DM648 Digital Signal Processor (DSP). The DDR2/mDDR memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices and standard Mobile DDR SDRAM devices.

SPRUEK6 — ***TMS320DM647/DM648 DSP External Memory Interface (EMIF) User's Guide*** describes the operation of the asynchronous external memory interface (EMIF) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The EMIF supports a glueless interface to a variety of external devices.

SPRUEK7 — ***TMS320DM647/DM648 DSP General-Purpose Input/Output (GPIO) User's Guide*** describes the general-purpose input/output (GPIO) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.

SPRUEK8 — ***TMS320DM647/DM648 DSP Inter-Integrated Circuit (I2C) Module User's Guide*** describes the inter-integrated circuit (I2C) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The I2C peripheral provides an interface between the DSP and other devices compliant with the I2C-bus specification and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the DSP through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

SPRUEL0 — ***TMS320DM647/DM648 DSP 64-Bit Timer User's Guide*** describes the operation of the 64-bit timer in the TMS320DM647/DM648 Digital Signal Processor (DSP). The timer can be configured as a general-purpose 64-bit timer, dual general-purpose 32-bit timers, or a watchdog timer.

SPRUEL1 — ***TMS320DM647/DM648 DSP Multichannel Audio Serial Port (McASP) User's Guide*** describes the multichannel audio serial port (McASP) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The McASP functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT).

SPRUEL2 — ***TMS320DM647/DM648 DSP Enhanced DMA (EDMA) Controller User's Guide*** describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM647/DM648 Digital Signal Processor (DSP). The EDMA3 controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DSP.

SPRUEL4 — ***TMS320DM647/DM648 DSP Peripheral Component Interconnect (PCI) User's Guide*** describes the peripheral component interconnect (PCI) port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The PCI port supports connection of the C642x DSP to a PCI host via the integrated PCI master/slave bus interface. The PCI port interfaces to the DSP via the enhanced DMA (EDMA) controller. This architecture allows for both PCI master and slave transactions, while keeping the EDMA channel resources available for other applications.

SPRUEL5 — ***TMS320DM647/DM648 DSP Host Port Interface (UHPI) User's Guide*** describes the host port interface (HPI) in the TMS320DM647/DM648 Digital Signal Processor (DSP). The HPI is a parallel port through which a host processor can directly access the CPU memory space. The host device functions as a master to the interface, which increases ease of access. The host and CPU can exchange information via internal or external memory. The host also has direct access to memory-mapped peripherals. Connectivity to the CPU memory space is provided through the enhanced direct memory access (EDMA) controller.

SPRUEL8 — ***TMS320DM647/DM648 DSP Universal Asynchronous Receiver/Transmitter (UART) User's Guide*** describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM647/DM648 Digital Signal Processor (DSP). The UART peripheral performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data received from the CPU.

SPRUEL9 — ***TMS320DM647/DM648 DSP VLYNQ Port User's Guide*** describes the VLYNQ port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The VLYNQ port is a high-speed point-to-point serial interface for connecting to host processors and other VLYNQ compatible devices. It is a full-duplex serial bus where transmit and receive operations occur separately and simultaneously without interference.

SPRUEM1 — ***TMS320DM647/DM648 DSP Video Port/VCXO Interpolated Control (VIC) Port User's Guide*** discusses the video port and VCXO interpolated control (VIC) port in the TMS320DM647/DM648 Digital Signal Processor (DSP). The video port can operate as a video capture port, video display port, or transport channel interface (TCI) capture port. The VIC port provides single-bit interpolated VCXO control with resolution from 9 bits to up to 16 bits. When the video port is used in TCI mode, the VIC port is used to control the system clock, VCXO, for MPEG transport channel.

SPRUEM2 — ***TMS320DM647/DM648 DSP Serial Port Interface (SPI) User's Guide*** discusses the Serial Port Interface (SPI) in the TMS320DM647/DM648 Digital Signal Processor (DSP). This reference guide provides the specifications for a 16-bit configurable, synchronous serial peripheral interface. The SPI is a programmable-length shift register, used for high speed communication between external peripherals or other DSPs.

SPRUEU6 — **TMS320DM647/DM648 DSP Subsystem User's Guide** describes the subsystem in the TMS320DM647/DM648 Digital Signal Processor (DSP). The subsystem is responsible for performing digital signal processing for digital media applications. The subsystem acts as the overall system controller, responsible for handling many system functions such as system-level initialization, configuration, user interface, user command execution, connectivity functions, and overall system control.

SPRUF57 — **TMS320DM647/DM648 DSP 3 Port Switch (3PSW) Ethernet Subsystem User's Guide** describes the operation of the 3 port switch (3PSW) ethernet subsystem in the TMS320DM647/DM648 Digital Signal Processor (DSP). The 3 port switch gigabit ethernet subsystem provides ethernet packet communication and can be configured as an ethernet switch (DM648 only). It provides the serial gigabit media independent interface (SGMII), the management data input output (MDIO) for physical layer device (PHY) management.

General-Purpose Input/Output (GPIO)

1 Introduction

The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an output, you can write to an internal register to control the state driven on the output pin. When configured as an input, you can detect the state of the input by reading the state of an internal register.

1.1 Purpose of the Peripheral

Most devices require some general-purpose input/output (GPIO) functionality in order to interact with other components in the system using low-speed interface pins. The control and use of the GPIO capability on this device is grouped together in the GPIO peripheral and is described in the following sections.

1.2 Features

The GPIO peripheral consists of the following features.

- Output set/clear functionality through separate data set and clear registers allows multiple software processes to control GPIO signals without critical section protection.
- Set/clear functionality through writing to a single output data register is also supported.
- Separate input/output registers
  - Output register can be read to reflect output drive status.
  - Input register can be read to reflect pin status.
- All GPIO signals can be used as interrupt sources with configurable edge detection.
- All GPIO signals can be used to generate events to the EDMA.

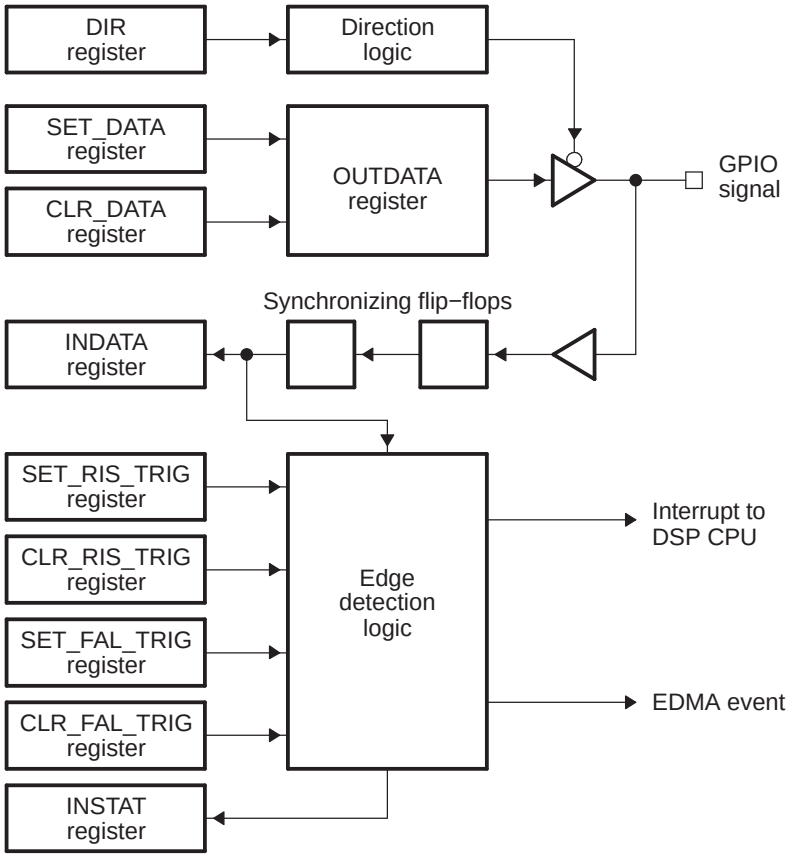
1.3 Functional Block Diagram

Figure 1 shows a block diagram of the GPIO peripheral.

1.4 Industry Standard(s) Compliance Statement

The GPIO peripheral connects to external devices. While it is possible that the software implements some standard connectivity protocol over GPIO, the GPIO peripheral itself is not compliant with any such standards.

Figure 1. GPIO Peripheral Block Diagram



2 Peripheral Architecture

The following sections describe the GPIO peripheral.

2.1 Clock Control

The input clock to the GPIO peripheral is not enabled by default. Program the local Power and Sleep Controller(LPSC) associated with the GPIO peripheral appropriately to enable the input clock to the GPIO peripheral. The input clock frequency for the GPIO peripheral will be CPU/6.

2.2 Signal Descriptions

The DM647 device supports up to 32 GPIO signals, GP[31-0]. For information on the package pinout of each GPIO signal, refer to the device data manual.

2.3 Pin Multiplexing

On the DM648 extensive pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package. Pin multiplexing is controlled using a combination of hardware configuration at device reset and software programmable register settings. Refer to the device-specific data manual to determine how pin multiplexing affects the GPIO module.

2.4 Endianness Considerations

The GPIO operation is independent of endianness; therefore, there are no endianness considerations for the GPIO module.

2.5 GPIO Register Structure

The GPIO signals are grouped into 2 banks of 16 signals per bank.

Associated with each bank of GPIO signals, there are several registers that control use of the GPIO bits, and within those registers, various control fields for each GPIO signal. The GPIO control registers are organized as 32-bit registers per pair of banks of GPIO signals.

The 32-bit register names per pair of banks of GPIO signals are all of the form *register_nameXY*, where *X* and *Y* represent the two bank numbers. The register fields associated with each GPIO are all of the form *field_nameN*, where *N* is the number of the GPIO signal. For example, for GP[0], which is located in GPIO bank 0, the control register names are of the form *register_name*01, and the register fields associated with GP[0] are all of the form *field_name*0. The GP[0] control bits are located in bit 0 of each of these registers.

Table 1 shows the banks and register control bit information associated with each GPIO pin on the device. Table 1 can be used to locate the register bits that control each GPIO signal. Detailed information regarding the specific register names for each bank and the contents and function of these registers is presented in Section 3.

Table 1. GPIO Register Bits and Banks Associated With GPIO Signals

GPIO Signal	Bank Number	Control Registers	Register Field	Bit Number
GP[0]	0	*register_name*01	*field_name*0	Bit 0
GP[1]	0	*register_name*01	*field_name*1	Bit 1
GP[2]	0	*register_name*01	*field_name*2	Bit 2
GP[3]	0	*register_name*01	*field_name*3	Bit 3
GP[4]	0	*register_name*01	*field_name*4	Bit 4
GP[5]	0	*register_name*01	*field_name*5	Bit 5
GP[6]	0	*register_name*01	*field_name*6	Bit 6
GP[7]	0	*register_name*01	*field_name*7	Bit 7
GP[8]	0	*register_name*01	*field_name*8	Bit 8
GP[9]	0	*register_name*01	*field_name*9	Bit 9
GP[10]	0	*register_name*01	*field_name*10	Bit 10
GP[11]	0	*register_name*01	*field_name*11	Bit 11
GP[12]	0	*register_name*01	*field_name*12	Bit 12
GP[13]	0	*register_name*01	*field_name*13	Bit 13
GP[14]	0	*register_name*01	*field_name*14	Bit 14
GP[15]	0	*register_name*01	*field_name*15	Bit 15
GP[16]	1	*register_name*01	*field_name*16	Bit 16
GP[17]	1	*register_name*01	*field_name*17	Bit 17
GP[18]	1	*register_name*01	*field_name*18	Bit 18
GP[19]	1	*register_name*01	*field_name*19	Bit 19
GP[20]	1	*register_name*01	*field_name*20	Bit 20
GP[21]	1	*register_name*01	*field_name*21	Bit 21
GP[22]	1	*register_name*01	*field_name*22	Bit 22
GP[23]	1	*register_name*01	*field_name*23	Bit 23
GP[24]	1	*register_name*01	*field_name*24	Bit 24
GP[25]	1	*register_name*01	*field_name*25	Bit 25
GP[26]	1	*register_name*01	*field_name*26	Bit 26

Table 1. GPIO Register Bits and Banks Associated With GPIO Signals (continued)

GPIO Signal	Bank Number	Control Registers	Register Field	Bit Number
GP[27]	1	*register_name*01	*field_name*27	Bit 27
GP[28]	1	*register_name*01	*field_name*28	Bit 28
GP[29]	1	*register_name*01	*field_name*29	Bit 29
GP[30]	1	*register_name*01	*field_name*30	Bit 30
GP[31]	1	*register_name*01	*field_name*31	Bit 31

2.6 Using a GPIO Signal as an Output

GPIO signals are configured to operate as inputs or outputs by writing the appropriate value to the GPIO direction register (DIR). This section describes using the GPIO signal as an output signal.

2.6.1 Configuring a GPIO Output Signal

To configure a given GPIO signal as an output, clear the bit in DIR that is associated with the desired GPIO signal. For detailed information on DIR, see Section 3.

2.6.2 Controlling the GPIO Output Signal State

There are three registers that control the output state driven on a GPIO signal configured as an output:

- GPIO set data register (SET_DATA01) controls driving GPIO signals high
- GPIO clear data register (CLR_DATA01) controls driving GPIO signals low
- GPIO output data register (OUT_DATA01) contains the current state of the output signals

Reading SET_DATA01, CLR_DATA01, and OUT_DATA01 returns the output state not necessarily the actual signal state (since some signals may be configured as inputs). The actual signal state is read using the GPIO input data register (IN_DATA01) associated with the desired GPIO signal. IN_DATA01 contains the actual logic state on the external signal.

For detailed information on these registers, see Section 3.

2.6.2.1 Driving a GPIO Output Signal High

To drive a GPIO signal high, use one of the following methods:

- Write a logic 1 to the bit in SET_DATA01 associated with the desired GPIO signal(s) to be driven high. Bit positions in SET_DATA containing logic 0 do not affect the state of the associated output signals.
- Modify the bit in OUT_DATA01 associated with the desired GPIO signal by using a read-modify-write operation. The logic states driven on the GPIO output signals match the logic values written to all bits in OUT_DATA01.

For GPIO signals configured as inputs, the values written to the associated SET_DATA01, CLR_DATA01, and OUT_DATA01 bits have no effect.

2.6.2.2 Driving a GPIO Output Signal Low

To drive a GPIO signal low, use one of the following methods:

- Write a logic 1 to the bit in CLR_DATA01 associated with the desired GPIO signal(s) to be driven low. Bit positions in CLR_DATA01 containing logic 0 do not affect the state of the associated output signals.
- Modify the bit in OUT_DATA01 associated with the desired GPIO signal by using a read-modify-write operation. The logic states driven on the GPIO output signals match the logic values written to all bits in OUT_DATA01.

For GPIO signals configured as inputs, the values written to the associated SET_DATA01, CLR_DATA01, and OUT_DATA01 bits have no effect.

2.7 Using a GPIO Signal as an Input

GPIO signals are configured to operate as inputs or outputs by writing the appropriate value to the GPIO direction register (DIR01). This section describes using the GPIO signal as an input signal.

2.7.1 Configuring a GPIO Input Signal

To configure a given GPIO signal as an input, set the bit in DIR01 that is associated with the desired GPIO signal. For detailed information on DIR01, see Section 3.

2.7.2 Reading a GPIO Input Signal

The current state of the GPIO signals is read using the GPIO input data register (IN_DATA01).

- For GPIO signals configured as inputs, reading IN_DATA01 returns the state of the input signal synchronized to the GPIO peripheral clock.
- For GPIO signals configured as outputs, reading IN_DATA01 returns the output value being driven by the device.

Some signals may utilize open-drain output buffers for wired-logic operations. For open-drain GPIO signals, reading IN_DATA01 returns the wired-logic value on the signal (which will not be driven by the device alone). Information on any signals using open-drain outputs is available in the device data manual.

To use GPIO input signals as interrupt sources, see section Section 2.10.

2.8 Reset Considerations

The GPIO peripheral has two reset sources: software reset and hardware reset.

2.8.1 Software Reset Considerations

A software reset (such as a reset initiated through the emulator) does not modify the configuration and state of the GPIO signals. A reset invoked via the Power and Sleep Controller (PSC) (GPIO clock disable, PSC reset, followed by GPIO clock enable) will result in the default configuration register settings.

2.8.2 Hardware Reset Considerations

A hardware reset does reset the GPIO configuration and data registers to their default states; therefore, affecting the configuration and state of the GPIO signals.

2.9 Initialization

The following steps are required to configure the GPIO module after a hardware reset:

1. Perform the necessary device pin multiplexing setup (see the device-specific data manual).
2. Program the Power and Sleep Controller (PSC) to enable the GPIO module. For details on the PSC, see the DSP Subsystem User's Guide.
3. Program the direction, data, and interrupt control registers to set the configuration of the desired GPIO pins (described in this document).

The GPIO module is now ready to perform data transactions.

2.10 Interrupt Support

The GPIO peripheral can send an interrupt event to the DSP CPU.

2.10.1 Interrupt Events and Requests

All GPIO signals can be configured to generate interrupts. The DM647 device supports interrupts from single GPIO signals, interrupts from banks of GPIO signals, or both. Note that the GPIO interrupts can also be used to provide synchronization events to the EDMA. See Section 2.11 for additional information.

2.10.2 Enabling GPIO Interrupt Events

GPIO interrupt events are enabled in banks of 16 by setting the appropriate bit(s) in the GPIO interrupt per-bank enable register (BINTEN). For example, to enable bank 0 interrupts (events from GP[15-0]), set bit 0 in BINTEN .

For detailed information on BINTEN, see Section 3.

See the device-specific data manual for interrupt number details

2.10.3 Configuring GPIO Interrupt Edge Triggering

Each GPIO interrupt source can be configured to generate an interrupt on the GPIO signal rising edge, falling edge, both edges, or neither edge (no event). The edge detection is synchronized to the GPIO peripheral module clock.

The following four registers control the configuration of the GPIO interrupt edge detection:

- The GPIO set rising edge interrupt register (SET_RIS_TRIG01) enables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
- The GPIO clear rising edge interrupt register (CLR_RIS_TRIG01) disables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
- The GPIO set falling edge interrupt register (SET_FAL_TRIG01) enables GPIO interrupts on the occurrence of a falling edge on the GPIO signal.
- The GPIO clear falling edge interrupt register (CLR_FAL_TRIG01) disables GPIO interrupts on the occurrence of a falling edge on the GPIO signal.

To configure a GPIO interrupt to occur only on rising edges of the GPIO signal:

- Write a logic 1 to the associated bit in SET_RIS_TRIG01.
- Write a logic 1 to the associated bit in CLR_FAL_TRIG01.

To configure a GPIO interrupt to occur only on falling edges of the GPIO signal:

- Write a logic 1 to the associated bit in SET_FAL_TRIG01.
- Write a logic 1 to the associated bit in CLR_RIS_TRIG01.

To configure a GPIO interrupt to occur on both the rising and falling edges of the GPIO signal:

- Write a logic 1 to the associated bit in SET_RIS_TRIG01.
- Write a logic 1 to the associated bit in SET_FAL_TRIG01.

To disable a specific GPIO interrupt:

- Write a logic 1 to the associated bit in CLR_RIS_TRIG01.
- Write a logic 1 to the associated bit in CLR_FAL_TRIG01.

For detailed information on these registers, see Section 3.

Note that the direction of the GPIO signal does not have to be an input for the interrupt event generation to work. When a GPIO signal is configured as an output, the software can change the GPIO signal state and, in turn, generate an interrupt. This can be useful for debugging interrupt signal connectivity.

2.10.4 GPIO Interrupt Status

The status of GPIO interrupt events can be monitored by reading the GPIO interrupt status register (INTSTAT01). Pending GPIO interrupts are indicated with a logic 1 in the associated bit position; interrupts that are not pending are indicated with a logic 0.

For individual GPIO interrupts that are directly routed to the DSP subsystem, the interrupt status can be read by reading the associated interrupt flag in the CPU. For the GPIO bank interrupts, INTSTAT01 can be used to determine which GPIO interrupt occurred. It is the responsibility of software to ensure that all pending GPIO interrupts are appropriately serviced.

Pending GPIO interrupt flags can be cleared by writing a logic 1 to the associated bit position in INTSTAT01.

For detailed information on INTSTAT, see Section 3.

2.10.5 Interrupt Multiplexing

No GPIO interrupts are multiplexed with other interrupt functions on the DM647 device.

2.11 EDMA Event Support

The GPIO peripheral can provide synchronization events to the EDMA. See the device-specific data manual for EDMA event details.

2.12 Power Management

The GPIO peripheral can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the GPIO peripheral is controlled by the processor Power and Sleep Controller (PSC). The PSC acts as a master controller for power management for all of the peripherals on the device. For detailed information on power management procedures using the PSC, see the DSP Subsystem User's Guide.

When the GPIO peripheral is placed in a low-power state by the PSC, the interrupt generation capability is suspended until the GPIO peripheral is removed from the low-power state. While in the low-power state, the GPIO signals configured as outputs are maintained at their state prior to the GPIO peripheral entering the low-power state.

2.13 Emulation Considerations

The GPIO peripheral is not affected by emulation suspend events (such as halts and breakpoints).

3 Registers

Table 2 lists the memory-mapped registers for the general-purpose input/output (GPIO). See the device-specific data manual for the memory address of these registers.

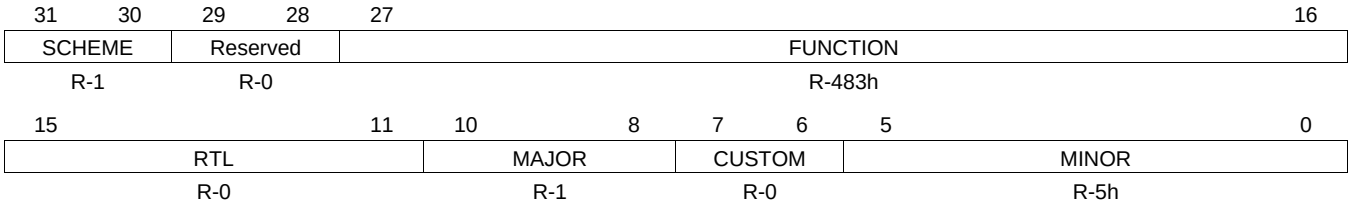
Table 2. General-Purpose Input/Output (GPIO) Registers

Offset	Acronym	Register Description	Section
0h	PID	Peripheral Identification Register	Section 3.1
8h	BINTEN	GPIO Interrupt Per-Bank Enable Register	Section 3.2
Ch	-	Reserved	-
		GPIO Banks 0 and 1	
10h	DIR01	GPIO Banks 0 and 1 Direction Register	Section 3.3
14h	OUT_DATA01	GPIO Banks 0 and 1 Output Data Register	Section 3.4
18h	SET_DATA01	GPIO Banks 0 and 1 Set Data Register	Section 3.5
1Ch	CLR_DATA01	GPIO Banks 0 and 1 Clear Data Register	Section 3.6
20h	IN_DATA01	GPIO Banks 0 and 1 Input Data Register	Section 3.7
24h	SET_RIS_TRIG01	GPIO Banks 0 and 1 Set Rising Edge Interrupt Register	Section 3.8
28h	CLR_RIS_TRIG01	GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register	Section 3.9
2Ch	SET_FAL_TRIG01	GPIO Banks 0 and 1 Set Falling Edge Interrupt Register	Section 3.10
30h	CLR_FAL_TRIG01	GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register	Section 3.11
34h	INTSTAT01	GPIO Banks 0 and 1 Interrupt Status Register	Section 3.12

3.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) contains identification data (type, class, and revision) for the peripheral. PID is shown in Figure 2 and described in Table 3.

Figure 2. Peripheral Identification Register (PID)



LEGEND: R = Read only; -*n* = value after reset

Table 3. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	1	Scheme of PID encoding. This field is fixed to 01.
29-28	Reserved	0	Reserved
27-16	FUNCTION	0-FFFh	Function. For GPIO = 483h
15-11	RTL	0-1Fh	RTL identification. For GPIO = 0
10-8	MAJOR	0-Fh	Major Revision. GPIO code revisions are indicated by a revision code taking the format MAJOR_REVISION.MINOR_REVISION. Major revision = 1h
7-6	CUSTOM	0-3h	Custom identification. For GPIO = 0
5-0	MINOR	0-Fh	Minor Revision. GPIO code revisions are indicated by a revision code taking the format MAJOR_REVISION.MINOR_REVISION. Minor revision = 5h

3.2 GPIO Interrupt Per-Bank Enable Register (BINTEN)

The GPIO interrupt per-bank enable register (BINTEN) is shown in Figure 3 and described in Table 4. For information on which GPIO signals are associated with each bank, see Table 1. Note that the bits in BINTEN control both the interrupt and EDMA events.

Figure 3. GPIO Interrupt Per-Bank Enable Register (BINTEN)

31								16
Reserved								
R-0								
15	7	6	5	4	3	2	1	0
Reserved							EN1	EN0
R-0							R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 4. GPIO Interrupt Per-Bank Enable Register (BINTEN) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
1	EN1		Bank 1 interrupt enable is used to disable or enable all GPIO pins as interrupt sources to the DSP CPU for Bank 1.
		0	Bank 1 GPIO interrupts are disabled.
		1	Bank 1 GPIO interrupts are enabled.
0	EN0		Bank 0 interrupt enable is used to disable or enable all GPIO pins as interrupt sources to the DSP CPU for Bank 0.
		0	Bank 0 GPIO interrupts are disabled.
		1	Bank 0 GPIO interrupts are enabled.

3.3 GPIO Direction Registers (DIR01)

The GPIO direction register (DIR*01*) determines if GPIO pin *n* in GPIO bank 0 and 1 is an input or an output. Each of the GPIO banks may have up to 16 GPIO pins. By default, all the GPIO pins are configured as inputs (bit value = 1). The GPIO direction register (DIR01) is shown in Figure 4 and described in Table 5. See Table 1 to determine the DIR*n* bit associated with each GPIO bank and pin number.

Figure 4. GPIO Banks 0 and 1 Direction Register (DIR01)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIR31	DIR30	DIR29	DIR28	DIR27	DIR26	DIR25	DIR24	DIR23	DIR22	DIR21	DIR20	DIR19	DIR18	DIR17	DIR16
R/W-1															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR15	DIR14	DIR13	DIR12	DIR11	DIR10	DIR9	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
R/W-1															

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 5. GPIO Direction Register (DIR*n*) Field Descriptions

Bit	Field	Value	Description
31-16	DIR*n*		Direction of GPIO pin *n*. The DIR*n* bit is used to control the direction (output = 0, input = 1) of pin *n* on GPIO bank 1. This bit field configures the GPIO pins on GPIO bank 1.
		0	GPIO pin *n* is an output.
		1	GPIO pin *n* is an input.
15-0	DIR*n*		Direction of GPIO pin *n*. The DIR*n* bit is used to control the direction (output = 0, input = 1) of pin *n* on GPIO bank 0. This bit field configures the GPIO pins on GPIO bank 0.
		0	GPIO pin *n* is an output.
		1	GPIO pin *n* is an input.

3.4 GPIO Output Data Register (OUT_DATA01)

The GPIO output data register (OUT_DATA*01*) determines the value driven on the corresponding GPIO pin *n* in GPIO bank *0 and 1,* if the pin is configured as an output (DIR*n* = 0). Writes do not affect pins not configured as GPIO outputs. The bits in OUT_DATA*n* are set or cleared by writing directly to this register. A read of OUT_DATA*n* returns the value of the register not the value at the pin (that might be configured as an input). The GPIO output data register (OUT_DATA01) is shown in Figure 5 and described in Table 6. See Table 1 to determine the OUT_DATA*n* bit associated with each GPIO bank and pin number.

Figure 5. GPIO Banks 0 and 1 Output Data Register (OUT_DATA01)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT31	OUT30	OUT29	OUT28	OUT27	OUT26	OUT25	OUT24	OUT23	OUT22	OUT21	OUT20	OUT19	OUT18	OUT17	OUT16
R/W-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT15	OUT14	OUT13	OUT12	OUT11	OUT10	OUT9	OUT8	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
R/W-0															

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 6. GPIO Output Data Register (OUT_DATA*01*) Field Descriptions

Bit	Field	Value	Description
31-16	OUT*n*		Output drive state of GPIO pin *n*. The OUT*n* bit is used to drive the output (low = 0, high = 1) of pin *n* on GPIO bank 1 only when pin *n* is configured as an output (DIR*n* = 0). The OUT*n* bit is ignored when GPIO pin *n* is configured as an input. This bit field configures the GPIO pins on GPIO banks 1.
		0	GPIO pin *n* is driven low.
		1	GPIO pin *n* is driven high.
15-0	OUT*n*		Output drive state of GPIO pin *n*. The OUT*n* bit is used to drive the output (low = 0, high = 1) of pin *n* on GPIO bank 0 only when pin *n* is configured as an output (DIR*n* = 0). The OUT*n* bit is ignored when GPIO pin *n* is configured as an input. This bit field configures the GPIO pins on GPIO banks 0.
		0	GPIO pin *n* is driven low.
		1	GPIO pin *n* is driven high.

3.5 GPIO Set Data Register (SET_DATA01)

The GPIO set data register (SET_DATA*01*) controls driving high the corresponding GPIO pin *n* in GPIO bank *0 and 1*, if the pin is configured as an output (DIR*n* = 0). Writes do not affect pins not configured as GPIO outputs. The bits in SET_DATA01 are set or cleared by writing directly to this register. A read of the SET*n* bit returns the output drive state of the corresponding GPIO pin *n*. The GPIO set data register (SET_DATA01) is shown in Figure 6 and described in Table 7. See Table 1 to determine the SET_DATA*n* bit associated with each GPIO bank and pin number.

Figure 6. GPIO Banks 0 and 1 Set Data Register (SET_DATA01)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SET31	SET30	SET29	SET28	SET27	SET26	SET25	SET24	SET23	SET22	SET21	SET20	SET19	SET18	SET17	SET16
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
R/W-0															

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 7. GPIO Set Data Register (SET_DATA*01*) Field Descriptions

Bit	Field	Value	Description
31-16	SET*n*		Set output drive state of GPIO pin *n*. The SET*n* bit is used to set the output of pin *n* on GPIO bank 1 only when pin *n* is configured as an output (DIR*n* = 0). The SET*n* bit is ignored when GPIO pin *n* is configured as an input. Writing a 1 to the SET*n* bit sets the output drive state of the corresponding GPIO pin *n*; reading the SET*n* bit returns the output drive state of the corresponding GPIO pin *n*. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	Set GPIO pin *n* output to 1.
15-0	SET*n*		Set output drive state of GPIO pin *n*. The SET*n* bit is used to set the output of pin *n* on GPIO bank 0 only when pin *n* is configured as an output (DIR*n* = 0). The SET*n* bit is ignored when GPIO pin *n* is configured as an input. Writing a 1 to the SET*n* bit sets the output drive state of the corresponding GPIO pin *n*; reading the SET*n* bit returns the output drive state of the corresponding GPIO pin *n*. This bit field configures the GPIO pins on GPIO banks 0.
		0	No effect.
		1	Set GPIO pin *n* output to 1.

3.6 GPIO Clear Data Register (CLR_DATA01)

The GPIO clear data register (CLR_DATA*01*) controls driving low the corresponding GPIO pin *n* in GPIO bank 0 and 1, if the pin is configured as an output (DIR*n* = 0). Writes do not affect pins not configured as GPIO outputs. The bits in CLR_DATA01 are set or cleared by writing directly to this register. A read of the CLR*n* bit returns the output drive state of the corresponding GPIO pin *n*. The GPIO clear data register (CLR_DATA01) is shown in Figure 7 and described in Table 8. See Table 1 to determine the CLR_DATA*n* bit associated with each GPIO bank and pin number.

Figure 7. GPIO Banks 0 and 1 Clear Data Register (CLR_DATA01)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLR31	CLR30	CLR29	CLR28	CLR27	CLR26	CLR25	CLR24	CLR23	CLR22	CLR21	CLR20	CLR19	CLR18	CLR17	CLR16
R/W-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
R/W-0															

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 8. GPIO Clear Data Register (CLR_DATA*01*) Field Descriptions

Bit	Field	Value	Description
31-16	CLR*n*		Clear output drive state of GPIO pin *n*. The CLR*n* bit is used to clear the output of pin *n* on GPIO bank 1 only when pin *n* is configured as an output (DIR*n* = 0). The CLR*n* bit is ignored when GPIO pin *n* is configured as an input. Writing a 1 to the CLR*n* bit clears the output drive state of the corresponding GPIO pin *n*; reading the CLR*n* bit returns the output drive state of the corresponding GPIO pin *n*. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	Clear GPIO pin *n* output to 0.
15-0	CLR*n*		Clear output drive state of GPIO pin *n*. The CLR*n* bit is used to clear the output of pin *n* on GPIO bank 2*l* only when pin *n* is configured as an output (DIR*n* = 0). The CLR*n* bit is ignored when GPIO pin *n* is configured as an input. Writing a 1 to the CLR*n* bit clears the output drive state of the corresponding GPIO pin *n*; reading the CLR*n* bit returns the output drive state of the corresponding GPIO pin *n*. This bit field configures the GPIO pins on GPIO banks 0, 2, 4, and 6.
		0	No effect.
		1	Clear GPIO pin *n* output to 0.

3.7 GPIO Input Data Register (IN_DATA01)

The current state of the GPIO signals is read using the GPIO input data register (IN_DATA*01*).

- For GPIO signals configured as inputs, reading IN_DATA*01* returns the state of the input signal synchronized to the GPIO peripheral clock.
- For GPIO signals configured as outputs, reading IN_DATA*01* returns the output value being driven by the device.

The GPIO input data register (IN_DATA01) is shown in Figure 8 and described in Table 9. See Table 1 to determine the IN_DATA*n* bit associated with each GPIO bank and pin number.

Figure 8. GPIO Banks 0 and 1 Input Data Register (IN_DATA01)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN31	IN30	IN29	IN28	IN27	IN26	IN25	IN24	IN23	IN22	IN21	IN20	IN19	IN18	IN17	IN16
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN15	IN14	IN13	IN12	IN11	IN10	IN9	IN8	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 9. GPIO Input Data Register (IN_DATA*01*) Field Descriptions

Bit	Field	Value	Description
31-16	IN*n*		Status of GPIO pin *n*. Reading the IN*n* bit returns the state of pin *n* on GPIO bank 1. This bit field returns the status of the GPIO pins on GPIO banks 1.
		0	GPIO pin *n* is logic low.
		1	GPIO pin *n* is logic high.
15-0	IN*n*		Status of GPIO pin *n*. Reading the IN*n* bit returns the state of pin *n* on GPIO bank 0. This bit field returns the status of the GPIO pins on GPIO banks 0.
		0	GPIO pin *n* is logic low.
		1	GPIO pin *n* is logic high.

3.8 GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG01)

The GPIO set rising edge interrupt register (SET_RIS_TRIG01) enables a rising edge on the GPIO pin to generate a GPIO interrupt. The GPIO set rising edge interrupt register (SET_RIS_TRIG01) is shown in Figure 9 and described in Table 10. See Table 1 to determine the SET_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 9. GPIO Banks 0 and 1 Set Rising Edge Interrupt Register (SET_RIS_TRIG01)

31	30	29	28	27	26	25	24
SETRIS31	SETRIS30	SETRIS29	SETRIS28	SETRIS27	SETRIS26	SETRIS25	SETRIS24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
SETRIS23	SETRIS22	SETRIS21	SETRIS20	SETRIS19	SETRIS18	SETRIS17	SETRIS16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
SETRIS15	SETRIS14	SETRIS13	SETRIS12	SETRIS11	SETRIS10	SETRIS9	SETRIS8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
SETRIS7	SETRIS6	SETRIS5	SETRIS4	SETRIS3	SETRIS2	SETRIS1	SETRIS0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 10. GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG*01*) Field Descriptions

Bit	Field	Value	Description
31-16	SETRIS*n*		Enable rising edge interrupt detection on GPIO pin *n*. Reading the SETRIS*n* bit returns the state of pin *n* on GPIO bank 1. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	Interrupt is caused by a low-to-high transition on GPIO pin *n*.
15-0	SETRIS*n*		Enable rising edge interrupt detection on GPIO pin *n*. Reading the SETRIS*n* bit returns the state of pin *n* on GPIO bank 0. This bit field configures the GPIO pins on GPIO banks 0.
		0	No effect.
		1	Interrupt is caused by a low-to-high transition on GPIO pin *n*.

3.9 GPIO Clear Rising Edge Interrupt Register (CLR_RIS_TRIG01)

The GPIO clear rising edge interrupt register (CLR_RIS_TRIG*01*) disables a rising edge on the GPIO pin from generating a GPIO interrupt. The GPIO clear rising edge interrupt register (CLR_RIS_TRIG01) is shown in Figure 10 and described in Table 11. See Table 1 to determine the CLR_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 10. GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG01)

31	30	29	28	27	26	25	24
CLRRIS31	CLRRIS30	CLRRIS29	CLRRIS28	CLRRIS27	CLRRIS26	CLRRIS25	CLRRIS24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
CLRRIS23	CLRRIS22	CLRRIS21	CLRRIS20	CLRRIS19	CLRRIS18	CLRRIS17	CLRRIS16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
CLRRIS15	CLRRIS14	CLRRIS13	CLRRIS12	CLRRIS11	CLRRIS10	CLRRIS9	CLRRIS8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
CLRRIS7	CLRRIS6	CLRRIS5	CLRRIS4	CLRRIS3	CLRRIS2	CLRRIS1	CLRRIS0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 11. GPIO Clear Rising Edge Interrupt Register (CLR_RIS_TRIG*01*) Field Descriptions

Bit	Field	Value	Description
31-16	CLRRIS*n*		Disable rising edge interrupt detection on GPIO pin *n*. Reading the CLRRIS*n* bit returns the complement state of pin *n* on GPIO bank 1. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	No interrupt is caused by a low-to-high transition on GPIO pin *n*.
15-0	CLRRIS*n*		Disable rising edge interrupt detection on GPIO pin *n*. Reading the CLRRIS*n* bit returns the complement state of pin *n* on GPIO bank 0. This bit field configures the GPIO pins on GPIO banks 0.
		0	No effect.
		1	No interrupt is caused by a low-to-high transition on GPIO pin *n*.

3.10 GPIO Set Falling Edge Interrupt Register (SET_FAL_TRIG01)

The GPIO set falling edge interrupt register (SET_FAL_TRIG01) enables a falling edge on the GPIO pin to generate a GPIO interrupt. The GPIO set falling edge interrupt register (SET_FAL_TRIG01) is shown in Figure 11 and described in Table 12. See Table 1 to determine the SET_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 11. GPIO Banks 0 and 1 Set Falling Edge Interrupt Register (SET_FAL_TRIG01)

31	30	29	28	27	26	25	24
SETFAL31	SETFAL30	SETFAL29	SETFAL28	SETFAL27	SETFAL26	SETFAL25	SETFAL24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
SETFAL23	SETFAL22	SETFAL21	SETFAL20	SETFAL19	SETFAL18	SETFAL17	SETFAL16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
SETFAL15	SETFAL14	SETFAL13	SETFAL12	SETFAL11	SETFAL10	SETFAL9	SETFAL8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
SETFAL7	SETFAL6	SETFAL5	SETFAL4	SETFAL3	SETFAL2	SETFAL1	SETFAL0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 12. GPIO Set Falling Edge Interrupt Register (SET_FAL_TRIG*01*) Field Descriptions

Bit	Field	Value	Description
31-16	SETFAL*n*		Enable falling edge interrupt detection on GPIO pin *n*. Reading the SETFAL*n* bit returns the state of pin *n* on GPIO bank 1. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	Interrupt is caused by a high-to-low transition on GPIO pin *n*.
15-0	SETFAL*n*		Enable falling edge interrupt detection on GPIO pin *n*. Reading the SETFAL*n* bit returns the state of pin *n* on GPIO bank 0. This bit field configures the GPIO pins on GPIO banks 0.
		0	No effect.
		1	Interrupt is caused by a high-to-low transition on GPIO pin *n*.

3.11 GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG01)

The GPIO clear falling edge interrupt register (CLR_FAL_TRIG*01*) disables a falling edge on the GPIO pin from generating a GPIO interrupt. The GPIO clear falling edge interrupt register (CLR_FAL_TRIG01) is shown in Figure 12 and described in Table 13. See Table 1 to determine the CLR_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 12. GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG01)

31	30	29	28	27	26	25	24
CLRFAL31	CLRFAL30	CLRFAL29	CLRFAL28	CLRFAL27	CLRFAL26	CLRFAL25	CLRFAL24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
CLRFAL23	CLRFAL22	CLRFAL21	CLRFAL20	CLRFAL19	CLRFAL18	CLRFAL17	CLRFAL16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
CLRFAL15	CLRFAL14	CLRFAL13	CLRFAL12	CLRFAL11	CLRFAL10	CLRFAL9	CLRFAL8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
CLRFAL7	CLRFAL6	CLRFAL5	CLRFAL4	CLRFAL3	CLRFAL2	CLRFAL1	CLRFAL0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 13. GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG*01*) Field Descriptions

Bit	Field	Value	Description
31-16	CLRFAL*n*		Disable falling edge interrupt detection on GPIO pin *n*. Reading the CLRFAL*n* bit returns the complement state of pin *n* on GPIO bank 1. This bit field configures the GPIO pins on GPIO banks 1.
		0	No effect.
		1	No interrupt is caused by a high-to-low transition on GPIO pin *n*.
15-0	CLRFAL*n*		Disable falling edge interrupt detection on GPIO pin *n*. Reading the CLRFAL*n* bit returns the complement state of pin *n* on GPIO bank 0. This bit field configures the GPIO pins on GPIO banks 0.
		0	No effect.
		1	No interrupt is caused by a high-to-low transition on GPIO pin *n*.

3.12 GPIO Interrupt Status Register (INTSTAT01)

The status of GPIO interrupt events can be monitored by reading the GPIO interrupt status register (INTSTAT*01*). In the associated bit position, pending GPIO interrupts are indicated with a logic 1 and GPIO interrupts that are not pending are indicated with a logic 0. The GPIO interrupt status register (INTSTAT01) is shown in Figure 13. See Table 1 to determine the INTSTAT*n* bit associated with each GPIO bank and pin number.

Figure 13. GPIO Banks 0 and 1 Interrupt Status Register (INTSTAT01)

31	30	29	28	27	26	25	24
STAT31	STAT30	STAT29	STAT28	STAT27	STAT26	STAT25	STAT24
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
23	22	21	20	19	18	17	16
STAT23	STAT22	STAT21	STAT20	STAT19	STAT18	STAT17	STAT16
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
15	14	13	12	11	10	9	8
STAT15	STAT14	STAT13	STAT12	STAT11	STAT10	STAT9	STAT8
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
7	6	5	4	3	2	1	0
STAT7	STAT6	STATSTAT5	STAT4	STAT3	STAT2	STAT1	STAT0
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; W1C = Write 1 to clear bit (writing 0 has no effect); -*n* = value after reset

Table 14. GPIO Interrupt Status Register (INTSTAT*01*) Field Descriptions

Bit	Field	Value	Description
31-16	STAT*n*		Interrupt status of GPIO pin *n*. The STAT*n* bit is used to monitor pending GPIO interrupts on pin *n* of GPIO bank 1. This bit field returns the status of GPIO pins on GPIO banks 1. Write a 1 to the STAT*n* bit to clear the STAT*n* bit; a write of 0 has no effect.
		0	No pending interrupt on GPIO pin *n*.
		1	Pending interrupt on GPIO pin *n*.
15-0	STAT*n*		Interrupt status of GPIO pin *n*. The STAT*n* bit is used to monitor pending GPIO interrupts on pin *n* of GPIO bank 0. This bit field returns the status of GPIO pins on GPIO banks 0. Write a 1 to the STAT*n* bit to clear the STAT*n* bit; a write of 0 has no effect.
		0	No pending interrupt on GPIO pin *n*.
		1	Pending interrupt on GPIO pin *n*.

IMPORTANT NOTICE

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

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

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

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

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

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

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

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

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

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
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
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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