TMS320DM35x Digital Media System-on-Chip (DMSoC) General-Purpose Input/Output (GPIO)

Reference Guide

Literature Number: SPRUEE6B
May 2006–Revised March 2008

TEXAS INSTRUMENTS

Contents

List of Figures

List of Tables

Read This First

This document describes the General-Purpose Input/Output (GPIO) on the TMS320DM35x Digital Media System-on-Chip.

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.

Related Documentation from Texas Instruments

The following documents describe the TMS320DM35x Digital Media System-on-Chip (DMSoC). Copies of these documents are available on the internet at www.ti.com.

SPRS463 — ***TMS320DM355 Digital Media System-on-Chip (DMSoC) Data Manual*** This document describes the overall TMS320DM355 system, including device architecture and features, memory map, pin descriptions, timing characteristics and requirements, device mechanicals, etc.

SPRZ264 — ***TMS320DM355 DMSoC Silicon Errata*** Describes the known exceptions to the functional specifications for the TMS320DM355 DMSoC.

SPRUFB3 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) ARM Subsystem Reference Guide*** This document describes the ARM Subsystem in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The ARM subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the components of the ARM Subsystem, the peripherals, and the external memories.

SPRUED1 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Asynchronous External Memory Interface (EMIF) Reference Guide*** This document describes the asynchronous external memory interface (EMIF) in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.

SPRUED2 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Universal Serial Bus (USB) Controller Reference Guide*** This document describes the universal serial bus (USB) controller in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The USB controller supports data throughput rates up to 480 Mbps. It provides a mechanism for data transfer between USB devices and also supports host negotiation.

SPRUED3 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Audio Serial Port (ASP) Reference Guide*** This document describes the operation of the audio serial port (ASP) audio interface in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the ASP are the AC97 and IIS modes. In addition to the primary audio modes, the ASP supports general serial port receive and transmit operation, but is not intended to be used as a high-speed interface.

SPRUED4 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Serial Peripheral Interface (SPI) Reference Guide*** This document describes the serial peripheral interface (SPI) in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the DMSoC and external peripherals. Typical applications include an interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMs and analog-to-digital converters.

SPRUED9 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Universal Asynchronous Receiver/Transmitter (UART) Reference Guide*** This document describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM35x Digital Media System-on-Chip (DMSoC). 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.

SPRUEE0 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Inter-Integrated Circuit (I2C) Peripheral Reference Guide*** This document describes the inter-integrated circuit (I2C) peripheral in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The I2C peripheral provides an interface between the DMSoC 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 DMSoC through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

SPRUEE2 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Multimedia Card (MMC)/Secure Digital (SD) Card Controller Reference Guide*** This document describes the multimedia card (MMC)/secure digital (SD) card controller in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The MMC/SD card is used in a number of applications to provide removable data storage. The MMC/SD controller provides an interface to external MMC and SD cards. The communication between the MMC/SD controller and MMC/SD card(s) is performed by the MMC/SD protocol.

SPRUEE4 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Enhanced Direct Memory Access (EDMA) Controller Reference Guide*** This document describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The EDMA controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.

SPRUEE5 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) 64-bit Timer Reference Guide*** This document describes the operation of the software-programmable 64-bit timers in the TMS320DM35x Digital Media System-on-Chip (DMSoC). Timer 0, Timer 1, and Timer 3 are used as general-purpose (GP) timers and can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode; Timer 2 is used only as a watchdog timer. The GP timer modes can be used to generate periodic interrupts or enhanced direct memory access (EDMA) synchronization events and Real Time Output (RTO) events (Timer 3 only). The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition, such as a non-exiting code loop.

SPRUEE6 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) General-Purpose Input/Output (GPIO) Reference Guide*** This document describes the general-purpose input/output (GPIO) peripheral in the TMS320DM35x Digital Media System-on-Chip (DMSoC). 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.

SPRUEE7 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Pulse-Width Modulator (PWM) Reference Guide*** This document describes the pulse-width modulator (PWM) peripheral in the TMS320DM35x Digital Media System-on-Chip (DMSoC).

SPRUEH7 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) DDR2/Mobile DDR (DDR2/mDDR) Memory Controller Reference Guide*** This document describes the DDR2/mDDR memory controller in the TMS320DM35x Digital Media System-on-Chip (DMSoC). The DDR2/mDDR memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM and mobile DDR devices.

SPRUF71 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Video Processing Front End (VPFE) Reference Guide*** This document describes the Video Processing Front End (VPFE) in the TMS320DM35x Digital Media System-on-Chip (DMSoC).

SPRUF72 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Video Processing Back End (VPBE) Reference Guide*** This document describes the Video Processing Back End (VPBE) in the TMS320DM35x Digital Media System-on-Chip (DMSoC).

SPRUF74 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Real-Time Out (RTO) Controller Reference Guide*** This document describes the Real Time Out (RTO) controller in the TMS320DM35x Digital Media System-on-Chip (DMSoC).

SPRUFC8 — ***TMS320DM35x Digital Media System-on-Chip (DMSoC) Peripherals Overview Reference Guide*** This document provides an overview of the peripherals in the TMS320DM35x Digital Media System-on-Chip (DMSoC).

SPRAAR3 — ***Implementing DDR2/mDDR PCB Layout on the TMS320DM35x DMSoC*** This provides board design recommendations and guidelines for DDR2 and mobile DDR.

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 system on a chip (SoC) 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

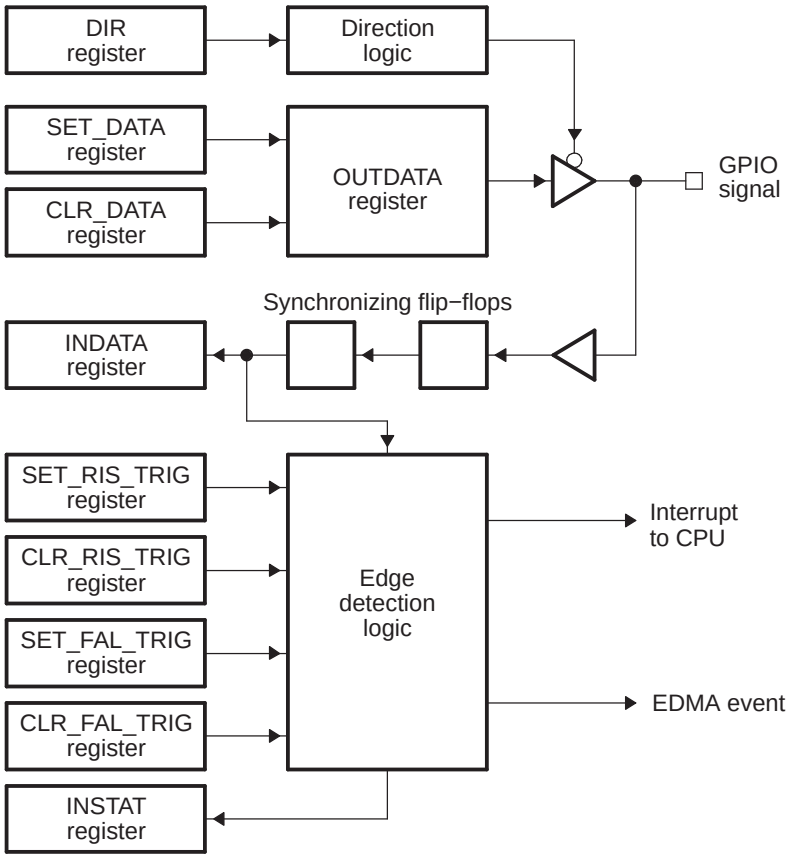
The GPIO peripheral consists of the following features.

- 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.
- Programmable debounce circuit provided on GPIO[7:0].

1.3 Functional Block Diagram

Figure 1 shows a block diagram of the GPIO peripheral.

Figure 1. GPIO Peripheral Block Diagram



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.

2 Peripheral Architecture

The following sections describe the GPIO peripheral.

2.1 Clock Control

The input clock to the GPIO peripheral is driven by PLL1. The maximum operation speed for the GPIO peripheral is 10 MHz.

2.2 Signal Descriptions

The DM35x device supports up to 104 signals, GPIO[103:0]. These GPIOs are 3.3V I/O signals. For information on the package pinout of each GPIO signal, refer to the device data manual.

2.3 GPIO Register Structure

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

The GPIO configuration registers are organized as one 32-bit register per pair of banks. When there are an odd number of banks, the upper 16-bit of registers for the last pair are reserved and have no effect. For the interrupt configuration, the registers associated with GPIO signals that do not support interrupt capability are also reserved and have no effect. Table 1 shows the banks and register control bit information associated with each GPIO pin on the device. The table can be used to locate the register bits that control each GPIO signal. For detailed information on the GPIO registers, see section Section 3.

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

GPIO Signal	Bank Number	Register Pair Number	Register Field Number	Bit Number
GPIO0	0	*register_name*01	*field_name*0	Bit 0
GPIO1	0	*register_name*01	*field_name*1	Bit 1
GPIO2	0	*register_name*01	*field_name*2	Bit 2
GPIO3	0	*register_name*01	*field_name*3	Bit 3
GPIO4	0	*register_name*01	*field_name*4	Bit 4
GPIO5	0	*register_name*01	*field_name*5	Bit 5
GPIO6	0	*register_name*01	*field_name*6	Bit 6
GPIO7	0	*register_name*01	*field_name*7	Bit 7
GPIO8	0	*register_name*01	*field_name*8	Bit 8
GPIO9	0	*register_name*01	*field_name*9	Bit 9
GPIO10	0	*register_name*01	*field_name*10	Bit 10
GPIO11	0	*register_name*01	*field_name*11	Bit 11
GPIO12	0	*register_name*01	*field_name*12	Bit 12
GPIO13	0	*register_name*01	*field_name*13	Bit 13
GPIO14	0	*register_name*01	*field_name*14	Bit 14
GPIO15	0	*register_name*01	*field_name*15	Bit 15
GPIO16	1	*register_name*01	*field_name*16	Bit 16
GPIO17	1	*register_name*01	*field_name*17	Bit 17
GPIO18	1	*register_name*01	*field_name*18	Bit 18
GPIO19	1	*register_name*01	*field_name*19	Bit 19
GPIO20	1	*register_name*01	*field_name*20	Bit 20
GPIO21	1	*register_name*01	*field_name*21	Bit 21
GPIO22	1	*register_name*01	*field_name*22	Bit 22
GPIO23	1	*register_name*01	*field_name*23	Bit 23
GPIO24	1	*register_name*01	*field_name*24	Bit 24
GPIO25	1	*register_name*01	*field_name*25	Bit 25
GPIO26	1	*register_name*01	*field_name*26	Bit 26
GPIO27	1	*register_name*01	*field_name*27	Bit 27
GPIO28	1	*register_name*01	*field_name*28	Bit 28
GPIO29	1	*register_name*01	*field_name*29	Bit 29
GPIO30	1	*register_name*01	*field_name*30	Bit 30
GPIO31	1	*register_name*01	*field_name*31	Bit 31
GPIO32	2	*register_name*23	*field_name*32	Bit 0
GPIO33	2	*register_name*23	*field_name*33	Bit 1
GPIO34	2	*register_name*23	*field_name*34	Bit 2
GPIO35	2	*register_name*23	*field_name*35	Bit 3
GPIO36	2	*register_name*23	*field_name*36	Bit 4
GPIO37	2	*register_name*23	*field_name*37	Bit 5

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

GPIO Signal	Bank Number	Register Pair Number	Register Field Number	Bit Number
GPIO38	2	*register_name*23	*field_name*38	Bit 6
GPIO39	2	*register_name*23	*field_name*39	Bit 7
GPIO40	2	*register_name*23	*field_name*40	Bit 8
GPIO41	2	*register_name*23	*field_name*41	Bit 9
GPIO42	2	*register_name*23	*field_name*42	Bit 10
GPIO43	2	*register_name*23	*field_name*43	Bit 11
GPIO44	2	*register_name*23	*field_name*44	Bit 12
GPIO45	2	*register_name*23	*field_name*45	Bit 13
GPIO46	2	*register_name*23	*field_name*46	Bit 14
GPIO47	2	*register_name*23	*field_name*47	Bit 15
GPIO48	3	*register_name*23	*field_name*48	Bit 16
GPIO49	3	*register_name*23	*field_name*49	Bit 17
GPIO50	3	*register_name*23	*field_name*50	Bit 18
GPIO51	3	*register_name*23	*field_name*51	Bit 19
GPIO52	3	*register_name*23	*field_name*52	Bit 20
GPIO53	3	*register_name*23	*field_name*53	Bit 21
GPIO54	3	*register_name*23	*field_name*54	Bit 22
GPIO55	3	*register_name*23	*field_name*55	Bit 23
GPIO56	3	*register_name*23	*field_name*56	Bit 24
GPIO57	3	*register_name*23	*field_name*57	Bit 25
GPIO58	3	*register_name*23	*field_name*58	Bit 26
GPIO59	3	*register_name*23	*field_name*59	Bit 27
GPIO60	3	*register_name*23	*field_name*60	Bit 28
GPIO61	3	*register_name*23	*field_name*61	Bit 29
GPIO62	3	*register_name*23	*field_name*62	Bit 30
GPIO63	3	*register_name*23	*field_name*63	Bit 31
GPIO64	4	*register_name*45	*field_name*64	Bit 0
GPIO65	4	*register_name*45	*field_name*65	Bit 1
GPIO66	4	*register_name*45	*field_name*66	Bit 2
GPIO67	4	*register_name*45	*field_name*67	Bit 3
GPIO68	4	*register_name*45	*field_name*68	Bit 4
GPIO69	4	*register_name*45	*field_name*69	Bit 5
GPIO70	4	*register_name*45	*field_name*70	Bit 6
GPIO71	4	*register_name*45	*field_name*71	Bit 7
GPIO72	4	*register_name*45	*field_name*72	Bit 8
GPIO73	4	*register_name*45	*field_name*73	Bit 9
GPIO74	4	*register_name*45	*field_name*74	Bit 10
GPIO75	4	*register_name*45	*field_name*75	Bit 11
GPIO76	4	*register_name*45	*field_name*76	Bit 12
GPIO77	4	*register_name*45	*field_name*77	Bit 13
GPIO78	4	*register_name*45	*field_name*78	Bit 14
GPIO79	4	*register_name*45	*field_name*79	Bit 15
GPIO80	5	*register_name*45	*field_name*80	Bit 16
GPIO81	5	*register_name*45	*field_name*81	Bit 17
GPIO82	5	*register_name45*	*field_name*82	Bit 18
GPIO83	5	*register_name*45	*field_name*83	Bit 19

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

GPIO Signal	Bank Number	Register Pair Number	Register Field Number	Bit Number
GPIO84	5	*register_name*45	*field_name*84	Bit 20
GPIO85	5	*register_name*45	*field_name*85	Bit 21
GPIO86	5	*register_name*45	*field_name*86	Bit 22
GPIO87	5	*register_name*45	*field_name*87	Bit 23
GPIO88	5	*register_name*45	*field_name*88	Bit 24
GPIO89	5	*register_name*45	*field_name*89	Bit 25
GPIO90	5	*register_name*45	*field_name*90	Bit 26
GPIO91	5	*register_name*45	*field_name*91	Bit 27
GPIO92	5	*register_name*45	*field_name*92	Bit 28
GPIO93	5	*register_name*45	*field_name*93	Bit 29
GPIO94	5	*register_name*45	*field_name*94	Bit 30
GPIO95	5	*register_name*45	*field_name*95	Bit 31
GPIO96	6	*register_name*6	*field_name*96	Bit 0
GPIO97	6	*register_name*6	*field_name*97	Bit 1
GPIO98	6	*register_name*6	*field_name*98	Bit 2
GPIO99	6	*register_name*6	*field_name*99	Bit 3
GPIO100	6	*register_name*6	*field_name*100	Bit 4
GPIO101	6	*register_name*6	*field_name*101	Bit 5
GPIO102	6	*register_name*6	*field_name*102	Bit 6
GPIO103	6	*register_name*6	*field_name*103	Bit 7

2.4 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.

2.4.1 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.

2.4.2 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_DATA) controls driving GPIO signals high
- GPIO clear data register (CLR_DATA) controls driving GPIO signals low
- GPIO output data register (OUT_DATA) contains the current state of the output signals

Reading SET_DATA, CLR_DATA, and OUT_DATA 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_DATA) associated with the desired GPIO signal. IN_DATA contains the actual logic state on the external signal.

For detailed information on these registers, see Section 3.

2.4.2.1 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_DATA 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_DATA 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_DATA.

For GPIO signals configured as inputs, the values written to the associated SET_DATA, CLR_DATA, and OUT_DATA bits have no effect.

2.4.2.2 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_DATA associated with the desired GPIO signal(s) to be driven low. Bit positions in CLR_DATA containing logic 0 do not affect the state of the associated output signals.
- Modify the bit in OUT_DATA 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_DATA.

For GPIO signals configured as inputs, the values written to the associated SET_DATA, CLR_DATA, and OUT_DATA bits have no effect.

2.5 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 (DIR). This section describes using the GPIO signal as an input signal.

2.5.1 Configuring a GPIO Input Signal

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

2.5.2 Reading a GPIO Input Signal

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

- For GPIO signals configured as inputs, reading IN_DATA returns the state of the input signal synchronized to the GPIO peripheral clock.
- For GPIO signals configured as outputs, reading IN_DATA 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_DATA 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 2.8.

2.6 Debounce

GPIO[7:0] feature a debounce circuit which provides the capability to filter out noise and spurious signals that occur on the input by ignoring switching that occurs faster than a selected rate. This can be useful in systems for debouncing mechanical switches, or for sampling the state of a slowly varying input.

This debounce capability is controlled by the eight DEBOUNCEn (n = 0-7) registers, one each for GPIO0-GPIO7. The DEBOUNCEn registers each contain two bit fields which control the debounce circuit - the ENABLE bit, and the INTERVAL bits. The ENABLE bit controls whether the debounce circuit for the corresponding GPIO pin is enabled or not, and the INTERVAL bits determine the duration of the debounce interval in clock cycles.

Once enabled, the debounce circuit stays idle until the input in question changes state. When this occurs, a counter starts counting out the debounce interval selected, and as long as the the input stays at the new state until the end of the count, the circuit will pass the new state to the rest of the GPIO logic. If the input changes state again before the end of the count, the counter is reset to zero, and starts counting again. When the debounce circuit is enabled, no input state changes are propagated to the rest of the GPIO logic unless the selected count interval has been counted. Note that this also results in a delay of the length of the interval count until any state changes are propagated to the rest of the GPIO logic. The interval counter is clocked at a rate that is 1/2 of the ARM CPU clock rate.

The DEBOUNCEn register bit field definitions and memory addresses are presented in the *TMS320DM35x DMSoC ARM Subsystem Reference Guide* (SPRUFB3).

2.7 Reset Considerations

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

2.7.1 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.

2.7.2 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.

2.8 Interrupt Support

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

2.8.1 Interrupt Events and Requests

A subset of the GPIO signals can also be configured to generate interrupts. The DM35x supports interrupts from single GPIO signals, interrupts from banks of GPIO signals, or both. The interrupt mapping from the GPIO peripheral to the ARM CPU is shown in Table 2.

Table 2. GPIO Interrupts to the ARM CPU

Interrupt Source	Acronym	ARM Interrupt Number
GPIO0	GPIO0	44
GPIO1	GPIO1	45
GPIO2	GPIO2	46
GPIO3	GPIO3	47
GPIO4	GPIO4	48
GPIO5	GPIO5	49
GPIO6	GPIO6	50
GPIO7	GPIO7	51
GPIO8	GPIO8	52
GPIO9	GPIO9	53
GPIO Bank 0	GPIOBNK0	54
GPIO Bank 1	GPIOBNK1	55
GPIO Bank 2	GPIOBNK2	56
GPIO Bank 3	GPIOBNK3	57
GPIO Bank 4	GPIOBNK4	58
GPIO Bank 5	GPIOBNK5	59

Table 2. GPIO Interrupts to the ARM CPU (continued)

Interrupt Source	Acronym	ARM Interrupt Number
GPIO Bank 6	GPIOBNK6	60

2.8.2 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 GPIO[15-0]), set bit 0 in BINTEN; to enable bank 3 interrupts (events from GPIO[63-48]), set bit 3 in BINTEN.

For detailed information on BINTEN, see Section 3.

2.8.3 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_TRIG) enables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
- The GPIO clear rising edge interrupt register (CLR_RIS_TRIG) disables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
- The GPIO set falling edge interrupt register (SET_FAL_TRIG) enables GPIO interrupts on the occurrence of a falling edge on the GPIO signal.
- The GPIO clear falling edge interrupt register (CLR_FAL_TRIG) 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_TRIG
- Write a logic 1 to the associated bit in CLR_FAL_TRIG

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_TRIG
- Write a logic 1 to the associated bit in CLR_RIS_TRIG

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_TRIG
- Write a logic 1 to the associated bit in SET_FAL_TRIG

To disable a specific GPIO interrupt:

- Write a logic 1 to the associated bit in CLR_RIS_TRIG
- Write a logic 1 to the associated bit in CLR_FAL_TRIG

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.

2.8.4 GPIO Interrupt Status

The status of GPIO interrupt events can be monitored by reading the GPIO interrupt status register (INTSTAT). 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 ARM interrupt controller, the interrupt status can be read by reading the associated interrupt flag in the ARM interrupt controller. For the GPIO bank interrupts, INTSTAT 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 INTSTAT.

For detailed information on INTSTAT, see Section 3.

2.8.5 Interrupt Multiplexing

No GPIO interrupts are multiplexed with other interrupt functions on the DM35x.

2.9 EDMA Event Support

The GPIO peripheral can provide synchronization events to the EDMA. The EDMA events supported on this device are listed in Table 3.

Table 3. GPIO Synchronization Events to the EDMA

Interrupt Source	EDMA Synchronization Event Number
GPIO0	32
GPIO1	33
GPIO2	34
GPIO3	35
GPIO4	36
GPIO5	37
GPIO6	38
GPIO7	39
GPIO8	47
GPIO9	25
GPIO Bank 0	40
GPIO Bank 1	41
GPIO Bank 2	42
GPIO Bank 3	43
GPIO Bank 4	44
GPIO Bank 5	45
GPIO Bank 6	46

2.10 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 *TMS320DM35x DMSoC ARM Subsystem Reference Guide* (SPRUFB3).

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.

2.11 Emulation Considerations

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

3 Registers

Table 4 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 4. 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
		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
		GPIO Banks 2 and 3	
38h	DIR23	GPIO Banks 2 and 3 Direction Register	Section 3.3
3Ch	OUT_DATA23	GPIO Banks 2 and 3 Output Data Register	Section 3.4
40h	SET_DATA23	GPIO Banks 2 and 3 Set Data Register	Section 3.5
44h	CLR_DATA23	GPIO Banks 2 and 3 Clear Data Register	Section 3.6
48h	IN_DATA23	GPIO Banks 2 and 3 Input Data Register	Section 3.7
4Ch	SET_RIS_TRIG23	GPIO Banks 2 and 3 Set Rising Edge Interrupt Register	Section 3.8
50h	CLR_RIS_TRIG23	GPIO Banks 2 and 3 Clear Rising Edge Interrupt Register	Section 3.9
54h	SET_FAL_TRIG23	GPIO Banks 2 and 3 Set Falling Edge Interrupt Register	Section 3.10
58h	CLR_FAL_TRIG23	GPIO Banks 2 and 3 Clear Falling Edge Interrupt Register	Section 3.11
5Ch	INTSTAT23	GPIO Banks 2 and 3 Interrupt Status Register	Section 3.12
		GPIO Bank 4 and 5	
60h	DIR45	GPIO Bank 4 and 5 Direction Register	Section 3.3
64h	OUT_DATA45	GPIO Bank 4 and 5 Output Data Register	Section 3.4
68h	SET_DATA45	GPIO Bank 4 and 5 Set Data Register	Section 3.5
6Ch	CLR_DATA45	GPIO Bank 4 and 5 Clear Data Register	Section 3.6
70h	IN_DATA45	GPIO Bank 4 and 5 Input Data Register	Section 3.7
74h	SET_RIS_TRIG45	GPIO Bank 4 and 5 Set Rising Edge Interrupt Register	Section 3.8
78h	CLR_RIS_TRIG45	GPIO Bank 4 and 5 Clear Rising Edge Interrupt Register	Section 3.9
7Ch	SET_FAL_TRIG45	GPIO Bank 4 and 5 Set Falling Edge Interrupt Register	Section 3.10
80h	CLR_FAL_TRIG45	GPIO Bank 4 and 5 Clear Falling Edge Interrupt Register	Section 3.11
84h	INTSTAT45	GPIO Bank 4 and 5 Interrupt Status Register	Section 3.12
		GPIO Bank 6	
88h	DIR6	GPIO Bank 6 Direction Register	Section 3.3
8Ch	OUT_DATA6	GPIO Bank 6 Output Data Register	Section 3.4
90h	SET_DATA6	GPIO Bank 6 Set Data Register	Section 3.5
94h	CLR_DATA6	GPIO Bank 6 Clear Data Register	Section 3.6
98h	IN_DATA6	GPIO Bank 6 Input Data Register	Section 3.7
9Ch	SET_RIS_TRIG6	GPIO Bank 6 Set Rising Edge Interrupt Register	Section 3.8
A0h	CLR_RIS_TRIG6	GPIO Bank 6 Clear Rising Edge Interrupt Register	Section 3.9

Table 4. General-Purpose Input/Output (GPIO) Registers (continued)

Offset	Acronym	Register Description	Section
A4h	SET_FAL_TRIG6	GPIO Bank 6 Set Falling Edge Interrupt Register	Section 3.10
A8h	CLR_FAL_TRIG6	GPIO Bank 6 Clear Falling Edge Interrupt Register	Section 3.11
ACh	INTSTAT6	GPIO Bank 6 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 5.

Figure 2. Peripheral Identification Register (PID)

31							24	23							16
Reserved								TID							
R-0								R-0							
15							8	7							0
CID								REV							
R-0								R-0							

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

Table 5. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID	0-Fh	Identifies type of peripheral.
15-8	CID	0-Fh	Identifies class of peripheral.
7-0	REV	0-Fh	Identifies revision of peripheral.

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 6. For information on which GPIO signals are associated with each bank, see Table 1.

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

31															16
Reserved															
R-0															

15								7	6	5	4	3	2	1	0
Reserved									EN6	EN5	EN4	EN3	EN2	EN1	EN0
R-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; R = Read only; -*n* = value after reset

Table 6. GPIO Interrupt Per-Bank Enable Register (BINTEN) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6	EN6		Bank 6 interrupt enable is used to disable or enable the bank 6 interrupts.
		0	Bank 6 interrupts are disabled.
		1	Bank 6 interrupts are enabled.
5	EN5		Bank 5 interrupt enable is used to disable or enable the bank 5 interrupts.
		0	Bank 5 interrupts are disabled.
		1	Bank 5 interrupts are enabled.
4	EN4		Bank 4 interrupt enable is used to disable or enable the bank 4 interrupts.
		0	Bank 4 interrupts are disabled.
		1	Bank 4 interrupts are enabled.
3	EN3		Bank 3 interrupt enable is used to disable or enable the bank 3 interrupts.
		0	Bank 3 interrupts are disabled.
		1	Bank 3 interrupts are enabled.
2	EN2		Bank 2 interrupt enable is used to disable or enable the bank 2 interrupts.
		0	Bank 2 interrupts are disabled.
		1	Bank 2 interrupts are enabled.
1	EN1		Bank 1 interrupt enable is used to disable or enable the bank 1 interrupts.
		0	Bank 1 interrupts are disabled.
		1	Bank 1 interrupts are enabled.
0	EN0		Bank 0 interrupt enable is used to disable or enable the bank 0 interrupts.
		0	Bank 0 interrupts are disabled.
		1	Bank 0 interrupts are enabled.

3.3 GPIO Direction Registers (DIRn)

The GPIO direction register (DIR*n*) determines if GPIO pin *n* in GPIO bank *l* 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, DIR23 is shown in Figure 5, DIR45 is shown in Figure 6, DIR6 is shown in Figure 7, and described in Table 7. 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
DIR31	DIR30	DIR29	DIR28	DIR27	DIR26	DIR25	DIR24
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
23	22	21	20	19	18	17	16
DIR23	DIR22	DIR21	DIR20	DIR19	DIR18	DIR17	DIR16
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
15	14	13	12	11	10	9	8
DIR15	DIR14	DIR13	DIR12	DIR11	DIR10	DIR9	DIR8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
7	6	5	4	3	2	1	0
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

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

Figure 5. GPIO Banks 2 and 3 Direction Register (DIR23)

31	30	29	28	27	26	25	24
DIR63	DIR62	DIR61	DIR60	DIR59	DIR58	DIR57	DIR56
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
23	22	21	20	19	18	17	16
DIR55	DIR54	DIR53	DIR52	DIR51	DIR50	DIR49	DIR48
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
15	14	13	12	11	10	9	8
DIR47	DIR46	DIR45	DIR44	DIR43	DIR42	DIR41	DIR40
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
7	6	5	4	3	2	1	0
DIR39	DIR38	DIR37	DIR36	DIR35	DIR34	DIR33	DIR32
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

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

Figure 6. GPIO Banks 4 and 5 Direction Register (DIR45)

31	30	29	28	27	26	25	24
DIR95	DIR94	DIR93	DIR92	DIR91	DIR90	DIR89	DIR88
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
23	22	21	20	19	18	17	16
DIR87	DIR86	DIR85	DIR84	DIR83	DIR82	DIR81	DIR80
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
15	14	13	12	11	10	9	8
DIR79	DIR78	DIR77	DIR76	DIR75	DIR74	DIR73	DIR72
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
7	6	5	4	3	2	1	0
DIR71	DIR80	DIR69	DIR68	DIR67	DIR66	DIR65	DIR64
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

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

Figure 7. GPIO Bank 6 Direction Register (DIR6)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8								
Reserved															
R-0															
7	6	5	4	3	2	1	0								
DIR103	DIR102	DIR101	DIR100	DIR99	DIR98	DIR97	DIR96								
R/W-1															

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

Table 7. 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 2*l* + 1. This bit field configures the GPIO pins on GPIO banks 1 and 3.
		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 2*l*. This bit field configures the GPIO pins on GPIO banks 0, 2, and 4.
		0	GPIO pin *n* is an output.
		1	GPIO pin *n* is an input.

3.4 GPIO Output Data Register (OUT_DATAn)

The GPIO output data register (OUT_DATA*n*) determines the value driven on the corresponding GPIO pin *n* in GPIO bank *I*, 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 8, OUT_DATA23 is shown in Figure 9, OUT_DATA45 is shown in Figure 10, OUT_DATA6 is shown in Figure 11, and described in Table 8. See Table 1 to determine the OUT_DATA*n* bit associated with each GPIO bank and pin number.

Figure 8. GPIO Banks 0 and 1 Output Data Register (OUT_DATA01)

31	30	29	28	27	26	25	24
OUT31	OUT30	OUT29	OUT28	OUT27	OUT26	OUT25	OUT24
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
OUT23	OUT22	OUT21	OUT20	OUT19	OUT18	OUT17	OUT16
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
OUT15	OUT14	OUT13	OUT12	OUT11	OUT10	OUT9	OUT8
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
OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
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

Figure 9. GPIO Banks 2 and 3 Output Data Register (OUT_DATA23)

31	30	29	28	27	26	25	24
OUT63	OUT62	OUT61	OUT60	OUT59	OUT58	OUT57	OUT56
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
OUT55	OUT54	OUT53	OUT52	OUT51	OUT50	OUT49	OUT48
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
OUT47	OUT46	OUT45	OUT44	OUT43	OUT42	OUT41	OUT40
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
OUT39	OUT38	OUT37	OUT36	OUT35	OUT34	OUT33	OUT32
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

Figure 10. GPIO Banks 4 and 5 Output Data Register (OUT_DATA45)

31	30	29	28	27	26	25	24
OUT95	OUT94	OUT93	OUT92	OUT91	OUT90	OUT89	OUT88
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
OUT87	OUT86	OUT85	OUT85	OUT83	OUT82	OUT81	OUT80
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
OUT79	OUT78	OUT77	OUT76	OUT75	OUT74	OUT73	OUT72
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
OUT71	OUT70	OUT69	OUT68	OUT67	OUT66	OUT65	OUT64
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

Figure 11. GPIO Bank 6 Output Data Register (OUT_DATA6)

31															16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved							
R-0							
7	6	5	4	3	2	1	0
OUT103	OUT102	OUT101	OUT100	OUT99	OUT98	OUT97	OUT96
R/W-0							

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

Table 8. GPIO Output Data Register (OUT_DATA*n*) 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 2*I* + 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 and 3.
		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 2*I* 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, 2, and 4.
		0	GPIO pin *n* is driven low.
		1	GPIO pin *n* is driven high.

3.5 GPIO Set Data Register (SET_DATAn)

The GPIO set data register (SET_DATA*n*) controls driving high the corresponding GPIO pin *n* in GPIO bank *I*, 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_DATA*n* 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 12, SET_DATA23 is shown in Figure 13, SET_DATA45 is shown in Figure 14, SET_DATA6 is shown in Figure 15, and described in Table 9. See Table 1 to determine the SET_DATA*n* bit associated with each GPIO bank and pin number.

Figure 12. GPIO Banks 0 and 1 Set Data Register (SET_DATA01)

31	30	29	28	27	26	25	24
SET31	SET30	SET29	SET28	SET27	SET26	SET25	SET24
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
SET23	SET22	SET21	SET20	SET19	SET18	SET17	SET16
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
SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8
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
SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
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

Figure 13. GPIO Banks 2 and 3 Set Data Register (SET_DATA23)

31	30	29	28	27	26	25	24
SET63	SET62	SET61	SET60	SET59	SET58	SET57	SET56
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
SET55	SET54	SET53	SET52	SET51	SET50	SET49	SET48
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
SET47	SET46	SET45	SET44	SET43	SET42	SET41	SET40
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
SET39	SET38	SET37	SET36	SET35	SET34	SET33	SET32
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

Figure 14. GPIO Banks 4 and 5 Set Data Register (SET_DATA45)

31	30	29	28	27	26	25	24
SET95	SET94	SET93	SET92	SET91	SET90	SET89	SET88
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
SET87	SET86	SET85	SET84	SET83	SET82	SET81	SET80
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
SET79	SET78	SET77	SET76	SET75	SET74	SET73	SET72
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
SET71	SET70	SET69	SET68	SET67	SET66	SET65	SET64
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

Figure 15. GPIO Bank 6 Set Data Register (SET_DATA6)

31															16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved							
R-0							
7	6	5	4	3	2	1	0
SET103	SET102	SET101	SET100	SET99	SET98	SET97	SET96
R/W-0							

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

Table 9. GPIO Set Data Register (SET_DATA*n*) 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 2*l* + 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 and 3.
		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 2*l* 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, 2, and 4.
		0	No effect.
		1	Set GPIO pin *n* output to 1.

3.6 GPIO Clear Data Register (CLR_DATAn)

The GPIO clear data register (CLR_DATA*n*) controls driving low the corresponding GPIO pin *n* in GPIO bank *I*, 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_DATA*n* 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 16, CLR_DATA23 is shown in Figure 17, CLR_DATA45 is shown in Figure 18 CLR_DATA6 is shown in Figure 19, and described in Table 10. See Table 1 to determine the CLR_DATA*n* bit associated with each GPIO bank and pin number.

Figure 16. GPIO Banks 0 and 1 Clear Data Register (CLR_DATA01)

31	30	29	28	27	26	25	24
CLR31	CLR30	CLR29	CLR28	CLR27	CLR26	CLR25	CLR24
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
CLR23	CLR22	CLR21	CLR20	CLR19	CLR18	CLR17	CLR16
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
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8
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
CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
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

Figure 17. GPIO Banks 2 and 3 Clear Data Register (CLR_DATA23)

31	30	29	28	27	26	25	24
CLR63	CLR62	CLR61	CLR60	CLR59	CLR58	CLR57	CLR56
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
CLR55	CLR54	CLR53	CLR52	CLR51	CLR50	CLR49	CLR48
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
CLR47	CLR46	CLR45	CLR44	CLR43	CLR42	CLR41	CLR40
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
CLR39	CLR38	CLR37	CLR36	CLR35	CLR34	CLR33	CLR32
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

Figure 18. GPIO Banks 4 and 5 Clear Data Register (CLR_DATA45)

31	30	29	28	27	26	25	24
CLR95	CLR94	CLR93	CLR92	CLR91	CLR90	CLR89	CLR88
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
CLR87	CLR86	CLR85	CLR84	CLR83	CLR82	CLR81	CLR80
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
CLR79	CLR78	CLR77	CLR76	CLR75	CLR74	CLR73	CLR72
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
CLR71	CLR70	CLR69	CLR68	CLR67	CLR66	CLR65	CLR64
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

Figure 19. GPIO Bank 6 Clear Data Register (CLR_DATA6)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8								
Reserved															
R-0															
7	6	5	4	3	2	1	0								
CLR103	CLR102	CLR101	CLR100	CLR99	CLR698	CLR97	CLR96								
R/W-0															

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

Table 10. GPIO Clear Data Register (CLR_DATA*n*) 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 2*l* + 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 and 3.
		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, and 4.
		0	No effect.
		1	Clear GPIO pin *n* output to 0.

3.7 GPIO Input Data Register (IN_DATAn)

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

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

The GPIO input data register (IN_DATA01) is shown in Figure 20, IN_DATA23 is shown in Figure 21, IN_DATA45 is shown in Figure 22, IN_DATA6 is shown in Figure 23, and described in Table 11. See Table 1 to determine the IN_DATA*n* bit associated with each GPIO bank and pin number.

Figure 20. GPIO Banks 0 and 1 Input Data Register (IN_DATA01)

31	30	29	28	27	26	25	24
IN31	IN30	IN29	IN28	IN27	IN26	IN25	IN24
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
IN23	IN22	IN21	IN20	IN19	IN18	IN17	IN16
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
IN15	IN14	IN13	IN12	IN11	IN10	IN9	IN8
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
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
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

Figure 21. GPIO Banks 2 and 3 Input Data Register (IN_DATA23)

31	30	29	28	27	26	25	24
IN63	IN62	IN61	IN60	IN59	IN58	IN57	IN56
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
IN55	IN54	IN53	IN52	IN51	IN50	IN49	IN48
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
IN47	IN46	IN45	IN44	IN43	IN42	IN41	IN40
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
IN39	IN38	IN37	IN36	IN35	IN34	IN33	IN32
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

Figure 22. GPIO Banks 4 and 5 Input Data Register (IN_DATA45)

31	30	29	28	27	26	25	24
IN95	IN94	IN93	IN92	IN91	IN90	IN89	IN88
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
IN87	IN86	IN85	IN84	IN83	IN82	IN81	IN80
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
IN79	IN78	IN77	IN76	IN75	IN74	IN73	IN72
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
IN71	IN70	IN69	IN68	IN67	IN66	IN65	IN64
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

Figure 23. GPIO Bank 6 Input Data Register (IN_DATA6)

31															16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved							
R-0							
7	6	5	4	3	2	1	0
IN103	IN102	IN101	IN100	IN99	IN98	IN97	IN96
R/W-0							

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

Table 11. GPIO Input Data Register (IN_DATA*n*) 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 2*I* + 1. This bit field returns the status of the GPIO pins on GPIO banks 1 and 3.
		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 2*I*. This bit field returns the status of the GPIO pins on GPIO banks 0, 2 and 4.
		0	GPIO pin *n* is logic low.
		1	GPIO pin *n* is logic high.

3.8 GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIGn)

The GPIO set rising edge interrupt register (SET_RIS_TRIG*n*) 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 24, SET_RIS_TRIG23 is shown in Figure 25, SET_RIS_TRIG45 is shown in Figure 26, SET_RIS_TRIG6 is shown in Figure 27, and described in Table 12. See Table 1 to determine the SET_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 24. 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

Figure 25. GPIO Banks 2 and 3 Set Rising Edge Interrupt Register (SET_RIS_TRIG23)

31	30	29	28	27	26	25	24
SETRIS63	SETRIS62	SETRIS61	SETRIS60	SETRIS59	SETRIS58	SETRIS57	SETRIS56
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
SETRIS55	SETRIS54	SETRIS53	SETRIS52	SETRIS51	SETRIS50	SETRIS49	SETRIS48
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
SETRIS47	SETRIS46	SETRIS45	SETRIS44	SETRIS43	SETRIS42	SETRIS41	SETRIS40
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
SETRIS39	SETRIS38	SETRIS37	SETRIS36	SETRIS35	SETRIS34	SETRIS33	SETRIS32
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

Figure 26. GPIO Banks 4 and 5 Set Rising Edge Interrupt Register (SET_RIS_TRIG45)

31	30	29	28	27	26	25	24
SETRIS95	SETRIS94	SETRIS93	SETRIS92	SETRIS91	SETRIS90	SETRIS89	SETRIS88
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
SETRIS87	SETRIS86	SETRIS85	SETRIS84	SETRIS83	SETRIS82	SETRIS81	SETRIS80
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
SETRIS79	SETRIS78	SETRIS77	SETRIS76	SETRIS75	SETRIS74	SETRIS73	SETRIS72
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
SETRIS71	SETRIS70	SETRIS69	SETRIS68	SETRIS67	SETRIS66	SETRIS65	SETRIS64
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

Figure 27. GPIO Bank 6 Set Rising Edge Interrupt Register (SET_RIS_TRIG6)

31															16
Reserved															
R-0															
15															8
Reserved															
R-0															
7	6	5	4	3	2	1	0								
SETRIS103	SETRIS102	SETRIS101	SETRIS100	SETRIS99	SETRIS98	SETRIS97	SETRIS96								
R-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; R = Read only; -*n* = value after reset

Table 12. GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG*n*) Field Descriptions

Bit	Field	Value	Description
31-16	SETRIS*n*		Enable rising edge interrupt detection on GPIO pin *n*. Reading the SETRISn bit in either SET_RIS_TRIGn or CLR_RIS_TRIGn always returns an indication of whether the rising edge interrupt generation function is enabled for pin n on GPIO bank 2I + 1. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 1 and 3.
		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 SETRISn bit in either SET_RIS_TRIGn or CLR_RIS_TRIGn always returns an indication of whether the rising edge interrupt generation function is enabled for pin n on GPIO bank 2I. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 0, 2, and 4.
		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_TRIGn)

The GPIO clear rising edge interrupt register (CLR_RIS_TRIG*n*) 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 28, CLR_RIS_TRIG23 is shown in Figure 29, CLR_RIS_TRIG45 is shown in Figure 30, CLR_RIS_TRIG6 is shown in Figure 31, and described in Table 13. See Table 1 to determine the CLR_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 28. 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

Figure 29. GPIO Banks 2 and 3 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG23)

31	30	29	28	27	26	25	24
CLRRIS63	CLRRIS62	CLRRIS61	CLRRIS60	CLRRIS59	CLRRIS58	CLRRIS57	CLRRIS56
R/W-10	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
CLRRIS55	CLRRIS54	CLRRIS53	CLRRIS52	CLRRIS51	CLRRIS50	CLRRIS49	CLRRIS48
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
CLRRIS47	CLRRIS46	CLRRIS45	CLRRIS44	CLRRIS43	CLRRIS42	CLRRIS41	CLRRIS40
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
CLRRIS39	CLRRIS38	CLRRIS37	CLRRIS36	CLRRIS35	CLRRIS34	CLRRIS33	CLRRIS32
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

Figure 30. GPIO Banks 4 and 5 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG45)

31	30	29	28	27	26	25	24
CLRRIS95	CLRRIS94	CLRRIS93	CLRRIS92	CLRRIS91	CLRRIS90	CLRRIS89	CLRRIS88
R/W-10	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
CLRRIS87	CLRRIS86	CLRRIS85	CLRRIS84	CLRRIS83	CLRRIS82	CLRRIS81	CLRRIS80
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
CLRRIS79	CLRRIS78	CLRRIS77	CLRRIS76	CLRRIS75	CLRRIS74	CLRRIS73	CLRRIS72
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
CLRRIS71	CLRRIS70	CLRRIS69	CLRRIS68	CLRRIS67	CLRRIS66	CLRRIS65	CLRRIS64
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

Figure 31. GPIO Bank 6 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG6)

31															16
Reserved															
R-0															

15							8
Reserved							
R-0							

7	6	5	4	3	2	1	0
CLRRIS103	CLRRIS102	CLRRIS101	CLRRIS100	CLRRIS99	CLRRIS98	CLRRIS97	CLRRIS96
R-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; R = Read only; -*n* = value after reset

Table 13. GPIO Clear Rising Edge Interrupt Register (CLR_RIS_TRIG*n*) Field Descriptions

Bit	Field	Value	Description
31-16	CLRRIS*n*		Disable rising edge interrupt detection on GPIO pin *n*. Reading the CLRRISn bit in either SET_RIS_TRIGn or CLR_RIS_TRIGn always returns an indication of whether the rising edge interrupt generation function is enabled for pin n on GPIO bank 2I + 1. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 1 and 3.
		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 CLRRISn bit in either SET_RIS_TRIGn or CLR_RIS_TRIGn always returns an indication of whether the rising edge interrupt generation function is enabled for pin n on GPIO bank 2I. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 0, 2, and 4.
		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_TRIGn)

The GPIO set falling edge interrupt register (SET_FAL_TRIG*n*) 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 32, SET_FAL_TRIG23 is shown in Figure 33, SET_FAL_TRIG45 is shown in Figure 34, SET_FAL_TRIG6 is shown in Figure 35, and described in Table 14. See Table 1 to determine the SET_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 32. 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

Figure 33. GPIO Banks 2 and 3 Set Falling Edge Interrupt Register (SET_FAL_TRIG23)

31	30	29	28	27	26	25	24
SETFAL63	SETFAL62	SETFAL61	SETFAL60	SETFAL59	SETFAL58	SETFAL57	SETFAL56
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
SETFAL55	SETFAL54	SETFAL53	SETFAL52	SETFAL51	SETFAL50	SETFAL49	SETFAL48
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
SETFAL47	SETFAL46	SETFAL45	SETFAL44	SETFAL43	SETFAL42	SETFAL41	SETFAL40
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
SETFAL39	SETFAL38	SETFAL37	SETFAL36	SETFAL35	SETFAL34	SETFAL33	SETFAL32
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

Figure 34. GPIO Banks 4 and 5 Set Falling Edge Interrupt Register (SET_FAL_TRIG45)

31	30	29	28	27	26	25	24
SETFAL95	SETFAL94	SETFAL93	SETFAL92	SETFAL91	SETFAL90	SETFAL89	SETFAL88
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
SETFAL87	SETFAL86	SETFAL85	SETFAL84	SETFAL83	SETFAL82	SETFAL81	SETFAL80
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
SETFAL79	SETFAL78	SETFAL77	SETFAL76	SETFAL75	SETFAL74	SETFAL73	SETFAL72
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
SETFAL71	SETFAL70	SETFAL69	SETFAL68	SETFAL67	SETFAL66	SETFAL65	SETFAL64
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

Figure 35. GPIO Bank 6 Set Falling Edge Interrupt Register (SET_FAL_TRIG6)

31															16
Reserved															
R-0															

15							8
Reserved							
R-0							

7	6	5	4	3	2	1	0
SETFAL103	SETFAL102	SETFAL101	SETFAL100	SETFAL99	SETFAL98	SETFAL97	SETFAL96
R-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; R = Read only; -*n* = value after reset

Table 14. GPIO Set Falling Edge Interrupt Register (SET_FAL_TRIG*n*) Field Descriptions

Bit	Field	Value	Description
31-16	SETFAL*n*		Enable falling edge interrupt detection on GPIO pin *n*. Reading the SETFALn bit in either SET_FAL_TRIGn or CLR_FAL_TRIGn always returns an indication of whether the falling edge interrupt generation function is enabled for pin n on GPIO bank 2I + 1. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 1 and 3.
		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 SETFALn bit in either SET_FAL_TRIGn or CLR_FAL_TRIGn always returns an indication of whether the falling edge interrupt generation function is enabled for pin n on GPIO bank 2I. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 0, 2, and 4.
		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_TRIGn)

The GPIO clear falling edge interrupt register (CLR_FAL_TRIG*n*) 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 36, CLR_FAL_TRIG23 is shown in Figure 37, CLR_FAL_TRIG45 is shown in Figure 38, CLR_FAL_TRIG6 is shown in Figure 39, and described in Table 15. See Table 1 to determine the CLR_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 36. 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

Figure 37. GPIO Banks 2 and 3 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG23)

31	30	29	28	27	26	25	24
CLRFAL63	CLRFAL62	CLRFAL61	CLRFAL60	CLRFAL59	CLRFAL58	CLRFAL57	CLRFAL56
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
CLRFAL55	CLRFAL54	CLRFAL53	CLRFAL52	CLRFAL51	CLRFAL50	CLRFAL49	CLRFAL48
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
CLRFAL47	CLRFAL46	CLRFAL45	CLRFAL44	CLRFAL43	CLRFAL42	CLRFAL41	CLRFAL40
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
CLRFAL39	CLRFAL38	CLRFAL37	CLRFAL36	CLRFAL35	CLRFAL34	CLRFAL33	CLRFAL32
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

Figure 38. GPIO Banks 4 and 5 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG45)

31	30	29	28	27	26	25	24
CLRFAL95	CLRFAL94	CLRFAL93	CLRFAL92	CLRFAL91	CLRFAL90	CLRFAL89	CLRFAL88
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
CLRFAL87	CLRFAL86	CLRFAL85	CLRFAL84	CLRFAL83	CLRFAL82	CLRFAL81	CLRFAL80
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
CLRFAL79	CLRFAL78	CLRFAL77	CLRFAL76	CLRFAL75	CLRFAL74	CLRFAL73	CLRFAL72
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
CLRFAL71	CLRFAL70	CLRFAL69	CLRFAL68	CLRFAL67	CLRFAL66	CLRFAL65	CLRFAL64
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

Figure 39. GPIO Bank 6 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG6)

31															16
Reserved															
R-0															

15							8
Reserved							
R-0							

7	6	5	4	3	2	1	0
CLRFAL103	CLRFAL102	CLRFAL101	CLRFAL100	CLRFAL99	CLRFAL98	CLRFAL97	CLRFAL96
R-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; R = Read only; -*n* = value after reset

Table 15. GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG*n*) Field Descriptions

Bit	Field	Value	Description
31-16	CLRFAL*n*		Disable falling edge interrupt detection on GPIO pin *n*. Reading the CLRFALn bit in either SET_FAL_TRIGn or CLR_FAL_TRIGn always returns an indication of whether the falling edge interrupt generation function is enabled for pin n on GPIO bank 2I + 1. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 1 and 3.
		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 CLRFALn bit in either SET_FAL_TRIGn or CLR_FAL_TRIGn always returns an indication of whether the falling edge interrupt generation function is enabled for pin n on GPIO bank 2I. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled. This bit field configures the GPIO pins on GPIO banks 0, 2, and 4.
		0	No effect.
		1	No interrupt is caused by a high-to-low transition on GPIO pin *n*.

3.12 GPIO Interrupt Status Register (INTSTATn)

The status of GPIO interrupt events can be monitored by reading the GPIO interrupt status register (INTSTAT*n*). 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 40, INTSTAT23 is shown in Figure 41, INTSTAT45 is shown in Figure 42, INTSTAT6 is shown in Figure 43, and described in Table 16. See Table 1 to determine the INTSTAT*n* bit associated with each GPIO bank and pin number.

Figure 40. 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

Figure 41. GPIO Banks 2 and 3 Interrupt Status Register (INTSTAT23)

31	30	29	28	27	26	25	24
STAT63	STAT62	STAT61	STAT60	STAT59	STAT58	STAT57	STAT56
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
STAT55	STAT54	STAT53	STAT52	STAT51	STAT50	STAT49	STAT48
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
STAT47	STAT46	STATSTAT45	STAT44	STAT43	STAT42	STAT41	STAT40
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
STAT39	STAT38	STAT37	STAT36	STAT35	STAT34	STAT33	STAT32
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

Figure 42. GPIO Banks 4 and 5 Interrupt Status Register (INTSTAT45)

31	30	29	28	27	26	25	24
STAT95	STAT94	STAT93	STAT92	STAT91	STAT90	STAT89	STAT88
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
STAT87	STAT86	STAT85	STAT84	STAT83	STAT82	STAT81	STAT80
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
STAT79	STAT78	STATSTAT77	STAT76	STAT75	STAT74	STAT73	STAT72
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
STAT71	STAT70	STAT69	STAT68	STAT67	STAT66	STAT65	STAT64
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

Figure 43. GPIO Bank 6 Interrupt Status Register (INTSTAT6)

31															16
Reserved															
R-0															
15															8
Reserved															
R-0															
7	6	5	4	3	2	1	0								
STAT103	STAT102	STAT101	STAT100	STAT99	STAT98	STAT97	STAT96								
R-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; R = Read only; W1C = Write 1 to clear bit (writing 0 has no effect); -*n* = value after reset

Table 16. GPIO Interrupt Status Register (INTSTAT*n*) 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 2*l* + 1. This bit field returns the status of GPIO pins on GPIO banks 1 and 3. 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 2*l*. This bit field returns the status of GPIO pins on GPIO banks 0, 2, and 4. 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*.

Appendix A Revision History

Table A-1 lists the changes made since the previous version of this document.

Table A-1. Document Revision History

Reference	Additions/Modifications/Deletions
Table 12	Revised descriptions for bits 31-16 and 15-0.
Table 13	Revised descriptions for bits 31-16 and 15-0.
Table 14	Revised descriptions for bits 31-16 and 15-0.
Table 15	Revised descriptions for bits 31-16 and 15-0.

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
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
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
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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