TMS320DM646x DMSoC
General-Purpose Input/Output (GPIO)

User's Guide

TEXAS INSTRUMENTS

Literature Number: SPRUEQ8A
March 2009

Contents

List of Figures

List of Tables

Read This First

About This Manual

Describes the general-purpose input/output (GPIO) in the peripheral TMS320DM646x 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.

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 TMS320DM646x Digital Media System-on-Chip (DMSoC). 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.

The current documentation that describes the DM646x DMSoC, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

SPRUEP8 — *TMS320DM646x DMSoC DSP Subsystem Reference Guide.* Describes the digital signal processor (DSP) subsystem in the TMS320DM646x Digital Media System-on-Chip (DMSoC).

SPRUEP9 — *TMS320DM646x DMSoC ARM Subsystem Reference Guide.* Describes the ARM subsystem in the TMS320DM646x 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 DSP subsystem and a majority of the peripherals and external memories.

SPRUEQ0 — *TMS320DM646x DMSoC Peripherals Overview Reference Guide.* Provides an overview and briefly describes the peripherals available on the TMS320DM646x Digital Media System-on-Chip (DMSoC).

SPRAA84 — *TMS320C64x to TMS320C64x+ CPU Migration Guide.* Describes migrating from the Texas Instruments TMS320C64x digital signal processor (DSP) to the TMS320C64x+ DSP. The objective of this document is to indicate differences between the two cores. Functionality in the devices that is identical is not included.

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.

SPRU871 — *TMS320C64x+ DSP Megamodule Reference Guide.* Describes the TMS320C64x+ digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

User's Guide

SPRUEQ8A–March 2009

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. The GPIO peripheral is accessible via the ARM926 CPU as well as the C64x CPU.

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

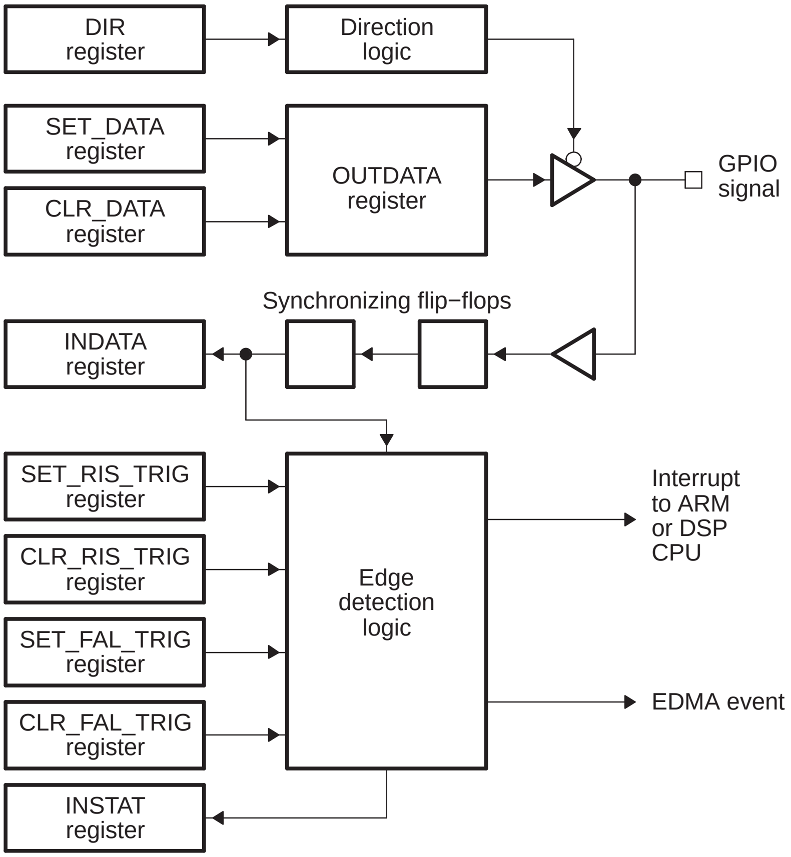
The GPIO peripheral consists of the following features.

- Supports up to 33 general-purpose pins
- 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.

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 Architecture

The following sections describe the GPIO peripheral.

2.1 Clock Control

The input clock to the GPIO peripheral represents PLL0 divided by 6. The maximum operation speed for the GPIO peripheral is 10 MHZ.

2.2 Signal Descriptions

The DM646x DMSoC supports up to 33 GPIO signals. All GPIO signals are 3.3V I/O signals. For information on the package pinout of each GPIO signal, refer to the device-specific 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. The supported 33 GPIOs are not necessarily populated in consecutive manner. 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
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	Reserved	*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	Reserved	*field_name*14	Bit 14
GP[15]	0	Reserved	*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
GP[27]	1	Reserved	*field_name*27	Bit 27
GP[28]	1	Reserved	*field_name*28	Bit 28
GP[29]	1	Reserved	*field_name*29	Bit 29
GP[30]	1	Reserved	*field_name*30	Bit 30
GP[31]	1	Reserved	*field_name*31	Bit 31
GP[32]	2	*register_name*2	*field_name*32	Bit 0
GP[33]	2	*register_name*2	*field_name*33	Bit 1
GP[34]	2	Reserved	*field_name*34	Bit 2
GP[35]	2	Reserved	*field_name*35	Bit 3
GP[36]	2	*register_name*2	*field_name*36	Bit 4

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

GPIO Signal	Bank Number	Register Pair Number	Register Field Number	Bit Number
GP[37]	2	*register_name*2	*field_name*37	Bit 5
GP[38]	2	*register_name*2	*field_name*38	Bit 6
GP[39]	2	*register_name*2	*field_name*39	Bit 7
GP[40]	2	*register_name*2	*field_name*40	Bit 8
GP[41]	2	*register_name*2	*field_name*41	Bit 9
GP[42]	2	*register_name*2	*field_name*42	Bit 10
GP[43]	2	Reserved	*field_name*43	Bit 11
GP[44]	2	Reserved	*field_name*44	Bit 12
GP[45]	2	Reserved	*field_name*45	Bit 13
GP[46]	2	Reserved	*field_name*46	Bit 14
GP[47]	2	Reserved	*field_name*47	Bit 15

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, for example, the direction register for the desired GPIO should be configured as output.

- GPIO set data register (SET_DATA) controls driving GPIO signals high. When writing 1 to the corresponding GPIO bit field of the SET_DATA register, the state of the corresponding GPIO would be driven high.
- GPIO clear data register (CLR_DATA) controls driving GPIO signals low. When writing 1 to the corresponding GPIO bit field of the CLR_DATA register, the state of the corresponding GPIO would be driven low.
- GPIO output data register (OUT_DATA) contains the current state of the output signals. When writing 1 or 0 to the corresponding GPIO bit field of the OUT_DATA register, the state of the corresponding GPIO would be driven high or low respectively. Use read-modify-write access to the GPIO OUT_DATA register in order to selectively affect the state of the desired GPIO.

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 (regardless with the GPIO direction configuration) 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 1 or 0 value to the GPIO direction register (DIR) respectively. 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-specific data manual.

To use GPIO input signals as interrupt sources, see section Section 2.7.

2.6 Reset Considerations

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

2.6.1 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 PSC (GPIO Clock Disable, PSC reset, followed by GPIO Clock Enable) will result in the default configuration registers setting.

2.6.2 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.

2.7 Interrupt Support

The GPIO peripheral can send an interrupt event to the ARM and/or the DSP.

2.7.1 Interrupts, Events, and Requests

All GPIO signals can also be configured to generate interrupts. The DM646x DMSoC supports interrupts from GPIO signals at both independent and group level. A selective set of GPIO signals have dedicated events allotted for them, in addition to the bank/group level. The remaining GPIOs can only generate events allotted at bank/group level. The interrupt and EDMA events from the GPIO peripheral to ARM and DSP CPUs are listed in Table 2. The same event generated to the Interrupt controller is used to generate events to the EDMA when the corresponding bank interrupt is enabled.

Table 2. GPIO Interrupts to the ARM CPU and DSP CPU

Interrupt Source	Acronym	ARM Interrupt Number	DSP Interrupt Number	EDMA Event Number/Channel
GP[0]	GPIO0	48	64	32
GP[1]	GPIO1	49	65	33
GP[2]	GPIO2	50	66	34
GP[3]	GPIO3	51	67	35
GP[4]	GPIO4	52	68	36
GP[5]	GPIO5	53	69	37
GP[6]	GPIO6	54	70	38
GP[7]	GPIO7	55	71	39
GPIO Bank 0	GPIOBNK0	56	-	40
GPIO Bank 1	GPIOBNK1	57	-	41
GPIO Bank 2	GPIOBNK2	58	-	42

Note: The DM646x DMSoC has 33 GPIOs. The GPIOs supported are not necessarily populated among the three banks supported.

See Table 1 for details.

2.7.2 Enabling Interrupt and EDMA Events

GPIO interrupt and EDMA events are enabled in banks of 16 by setting the appropriate bit(s) in the GPIO Interrupt and EDMA Events Per-Bank Enable Register (BINTEN). To enable bank 0 interrupts or EDMA events to any member of BANK0 interrupt (GP[15-0]), set the bit to 0 in the BINTEN register.

For detailed information on BINTEN, see Section 3.

2.7.3 GPIO Interrupt and EDMA Event Generation by Edge Triggering

Each GPIO interrupt source can be configured to generate an interrupt or EDMA events 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.

The direction of the GPIO signal does not have to be an input for the interrupt and EDMA 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 or an EDMA event. This can be useful for debugging interrupt signal connectivity.

2.7.4 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 or DSP subsystem, the interrupt status can be read by reading the associated interrupt flag in the CPU. 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.

2.7.5 Interrupt Multiplexing

No GPIO interrupts are multiplexed with other interrupt functions on the DM646x DMSoC.

2.8 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 *TMS320DM646x DMSoC ARM Subsystem Reference Guide* (SPRUEP9).

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.

2.9 Emulation Considerations

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

3 Registers

Table 3 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 3. General-Purpose Input/Output (GPIO) Registers

Offset	Acronym	Register Description	Section
0h	PID	Peripheral Identification Register	Section 3.1
8h	BINTEN	GPIO Interrupt and EDMA Events 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 Bank 2	
38h	DIR2	GPIO Bank 2 Direction Register	Section 3.3
3Ch	OUT_DATA2	GPIO Bank 2 Output Data Register	Section 3.4
40h	SET_DATA2	GPIO Bank 2 Set Data Register	Section 3.5
44h	CLR_DATA2	GPIO Bank 2 Clear Data Register	Section 3.6
48h	IN_DATA2	GPIO Bank 2 Input Data Register	Section 3.7
4Ch	SET_RIS_TRIG2	GPIO Bank 2 Set Rising Edge Interrupt Register	Section 3.8
50h	CLR_RIS_TRIG2	GPIO Bank 2 Clear Rising Edge Interrupt Register	Section 3.9
54h	SET_FAL_TRIG2	GPIO Bank 2 Set Falling Edge Interrupt Register	Section 3.10
58h	CLR_FAL_TRIG2	GPIO Bank 2 Clear Falling Edge Interrupt Register	Section 3.11
5Ch	INTSTAT2	GPIO Bank 2 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 4.

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 4. 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 and EDMA Events Per-Bank Enable Register (BINTEN)

The GPIO Interrupt and EDMA Events Per-Bank Enable Register (BINTEN) is shown in Figure 3 and described in Table 5. For information on which GPIO signals are associated with each bank, see Table 1.

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

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		EN2	EN1	EN0
R-0		R/W-0	R/W-0	R/W-0

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

Table 5. GPIO Interrupt and EDMA Events Per-Bank Enable Register (BINTEN) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	EN2		Bank 2 interrupt and EDMA events enable is used to disable or enable the bank 2 interrupts and EDMA events.
		0	Bank 2 interrupts are disabled.
		1	Bank 2 interrupts are enabled.
1	EN1		Bank 1 interrupt and EDMA events enable is used to disable or enable the bank 1 interrupts and EDMA events.
		0	Bank 1 interrupts are disabled.
		1	Bank 1 interrupts are enabled.

Table 5. GPIO Interrupt and EDMA Events Per-Bank Enable Register (BINTEN) Field Descriptions (continued)

Bit	Field	Value	Description
0	EN0		Bank 0 interrupt and EDMA events enable is used to disable or enable the bank 0 interrupts and EDMA events.
		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*I* 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, DIR2 is shown in Figure 5, and described in Table 6. 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	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DIR26	DIR25	DIR24	DIR23	DIR22	DIR21	DIR20	DIR19	DIR18	DIR17	DIR16
R-1		R/W-1	R/W-1	R/W-1	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	7	6	5	4	3	2	1	0
Reserved		DIR13	DIR12	DIR11	DIR10	RSV	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
R-1		R/W-1	R/W-1	R/W-1	R/W-1	R-1	R/W-1	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; R = Read only; -*n* = value after reset

Figure 5. GPIO Bank 2 Direction Register (DIR2)

31	16
Reserved	
R-1	

15	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DIR42	DIR41	DIR40	DIR39	DIR38	DIR37	DIR36	Reserved		DIR33	DIR32
R-1		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-1		R/W-1	R/W-1

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

Table 6. GPIO Direction Register (DIRn) Field Descriptions

Bit	Field[1]	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*I* + 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 2*I*. This bit field configures the GPIO pins on GPIO banks 0 and 2.
		0	GPIO pin *n* is an output.
		1	GPIO pin *n* is an input.

[1] All of the fields are not populated. See Figure 4 and Figure 5 for the GPIO bit fields that are supported.

3.4 GPIO Output Data Register (OUT_DATA*n*)

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 6, OUT_DATA2 is shown in Figure 7, and described in Table 7. See Table 1 to determine the OUT_DATA*n* bit associated with each GPIO bank and pin number.

Figure 6. GPIO Banks 0 and 1 Output Data Register (OUT_DATA01)

31	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		OUT26	OUT25	OUT24	OUT23	OUT22	OUT21	OUT20	OUT19	OUT18	OUT17	OUT16
R-0		R/W-1	R/W-1	R/W-1	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	7	6	5	4	3	2	1	0
Reserved		OUT13	OUT12	OUT11	OUT10	RSV	OUT8	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	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; R = Read only; -*n* = value after reset

Figure 7. GPIO Bank 2 Output Data Register (OUT_DATA2)

31	16
Reserved	
R-0	

15	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		OUT42	OUT41	OUT40	OUT39	OUT38	OUT37	OUT36	Reserved		OUT33	OUT32
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0		R/W-1	R/W-1

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

Table 7. GPIO Output Data Register (OUT_DATA*n*) Field Descriptions

Bit	Field[1]	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 bank 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 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 and 2.
		0	GPIO pin *n* is driven low.
		1	GPIO pin *n* is driven high.

[1] All of the fields are not populated. See Figure 6 and Figure 7 for the GPIO bit fields that are supported.

3.5 GPIO Set Data Register (SET_DATA*n*)

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 8, SET_DATA2 is shown in Figure 9, and described in Table 8. See Table 1 to determine the SET_DATA*n* bit associated with each GPIO bank and pin number.

Figure 8. GPIO Banks 0 and 1 Set Data Register (SET_DATA01)

31	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SET26	SET25	SET24	SET23	SET22	SET21	SET20	SET19	SET18	SET17	SET16
R-0		R/W-1	R/W-1	R/W-1	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	7	6	5	4	3	2	1	0
Reserved		SET13	SET12	SET11	SET10	RSV	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	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; R = Read only; -*n* = value after reset

Figure 9. GPIO Bank 2 Set Data Register (SET_DATA2)

31	16
Reserved	
R-0	

15	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SET42	SET41	SET40	SET39	SET38	SET37	SET36	Reserved		SET33	SET32
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0		R/W-1	R/W-1

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

Table 8. GPIO Set Data Register (SET_DATA*n*) Field Descriptions

Bit	Field[1]	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*I* + 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 bank 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 2*I* 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 and 2.
		0	No effect.
		1	Set GPIO pin *n* output to 1.

[1] All of the fields are not populated. See Figure 8 and Figure 9 for the GPIO bit fields that are supported.

3.6 GPIO Clear Data Register (CLR_DATA*n*)

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 10, CLR_DATA2 is shown in Figure 11, and described in Table 9. See Table 1 to determine the CLR_DATA*n* bit associated with each GPIO bank and pin number.

Figure 10. GPIO Banks 0 and 1 Clear Data Register (CLR_DATA01)

31	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CLR26	CLR25	CLR24	CLR23	CLR22	CLR21	CLR20	CLR19	CLR18	CLR17	CLR16
R-0		R/W-1	R/W-1	R/W-1	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	7	6	5	4	3	2	1	0
Reserved		CLR13	CLR12	CLR11	CLR10	RSV	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	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; R = Read only; -*n* = value after reset

Figure 11. GPIO Bank 2 Clear Data Register (CLR_DATA2)

31	16
Reserved	
R-0	

15	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CLR42	CLR41	CLR40	CLR39	CLR38	CLR37	CLR36	Reserved		CLR33	CLR32
R-0		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0		R/W-1	R/W-1

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

Table 9. GPIO Clear Data Register (CLR_DATA*n*) Field Descriptions

Bit	Field[1]	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/ + 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 bank 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*I* 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 and 2.
		0	No effect.
		1	Clear GPIO pin *n* output to 0.

[1] All of the fields are not populated. See Figure 10 and Figure 11 for the GPIO bit fields that are supported.

3.7 GPIO Input Data Register (IN_DATA*n*)

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 12, IN_DATA2 is shown in Figure 13, and described in Table 10. See Table 1 to determine the IN_DATA*n* bit associated with each GPIO bank and pin number.

Figure 12. GPIO Banks 0 and 1 Input Data Register (IN_DATA01)

31 27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IN26	IN25	IN24	IN23	IN22	IN21	IN20	IN19	IN18	IN17	IN16
R-0	R/W-1	R/W-1	R/W-1	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	7	6	5	4	3	2	1	0
Reserved	IN13	IN12	IN11	IN10	RSV	IN8	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
R-0	R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	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; R = Read only; -*n* = value after reset

Figure 13. GPIO Bank 2 Input Data Register (IN_DATA2)

31 16
Reserved
R-0

15 11	10	9	8	7	6	5	4	3 2	1	0
Reserved	IN42	IN41	IN40	IN39	IN38	IN37	IN36	Reserved	IN33	IN32
R-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R-0	R/W-1	R/W-1

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

Table 10. GPIO Input Data Register (IN_DATA*n*) Field Descriptions

Bit	Field[1]	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 bank 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 2*I*. This bit field returns the status of the GPIO pins on GPIO banks 0 and 2.
		0	GPIO pin *n* is logic low.
		1	GPIO pin *n* is logic high.

[1] All of the fields are not populated. See Figure 12 and Figure 13 for the GPIO bit fields that are supported.

3.8 GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG*n*)

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 14, SET_RIS_TRIG2 is shown in Figure 15, and described in Table 11. See Table 1 to determine the SET_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 14. GPIO Banks 0 and 1 Set Rising Edge Interrupt Register (SET_RIS_TRIG01)

31				27	26	25	24
Reserved					SETRIS26	SETRIS25	SETRIS24
R-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
Reserved		SETRIS13	SETRIS12	SETRIS11	SETRIS10	Reserved	SETRIS8
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R-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; R = Read only; -*n* = value after reset

Figure 15. GPIO Bank 2 Set Rising Edge Interrupt Register (SET_RIS_TRIG2)

31	16
Reserved	
R-0	

15				11	10	9	8
Reserved					SETRIS42	SETRIS41	SETRIS40
R-0					R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
SETRIS39	SETRIS38	SETRIS37	SETRIS36	Reserved		SETRIS33	SETRIS32
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

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

Table 11. GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG*n*) Field Descriptions

Bit	Field[1]	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 2*I* + 1. This bit field configures the GPIO pins on GPIO bank 1.
		0	No effect.
		1	Interrupt is caused by a low-to-high transition on GPIO pin *n*.

[1] All of the fields are not populated. See Figure 14 and Figure 15 for the GPIO bit fields that are supported.

Table 11. GPIO Set Rising Edge Interrupt Register (SET_RIS_TRIG*n*) Field Descriptions (continued)

Bit	Field[(1)]	Value	Description
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 2*l*. This bit field configures the GPIO pins on GPIO banks 0 and 2.
		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_TRIG*n*)

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 16, CLR_RIS_TRIG2 is shown in Figure 17, and described in Table 12. See Table 1 to determine the CLR_RIS_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 16. GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG01)

31				27	26	25	24
Reserved					CLRRIS26	CLRRIS25	CLRRIS24
R-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
Reserved		CLRRIS13	CLRRIS12	CLRRIS11	CLRRIS10	Reserved	CLRRIS8
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R-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; R = Read only; -*n* = value after reset

Figure 17. GPIO Bank 2 Clear Rising Edge Interrupt Register (CLR_RIS_TRIG2)

31							16
Reserved							
R-0							

15				11	10	9	8
Reserved					CLRRIS42	CLRRIS41	CLRRIS40
R-0					R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CLRRIS39	CLRRIS38	CLRRIS37	CLRRIS36	Reserved		CLRRIS33	CLRRIS32
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

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

Table 12. GPIO Clear Rising Edge Interrupt Register (CLR_RIS_TRIG*n*) Field Descriptions

Bit	Field[1]	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 2*I* + 1. This bit field configures the GPIO pins on GPIO bank 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 2*I*. This bit field configures the GPIO pins on GPIO banks 0 and 2.
		0	No effect.
		1	No interrupt is caused by a low-to-high transition on GPIO pin *n*.

[1] All of the fields are not populated. See Figure 16 and Figure 17 for the GPIO bit fields that are supported.

3.10 GPIO Set Falling Edge Interrupt Register (SET_FAL_TRIG*n*)

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 18, SET_FAL_TRIG2 is shown in Figure 19, and described in Table 13. See Table 1 to determine the SET_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 18. GPIO Banks 0 and 1 Set Falling Edge Interrupt Register (SET_FAL_TRIG01)

31	27	26	25	24
Reserved		SETFAL26	SETFAL25	SETFAL24
R-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
Reserved		SETFAL13	SETFAL12	SETFAL11	SETFAL10	Reserved	SETFAL8
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R-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; R = Read only; -*n* = value after reset

Figure 19. GPIO Bank 2 Set Falling Edge Interrupt Register (SET_FAL_TRIG2)

31	16
Reserved	
R-0	

15	11	10	9	8
Reserved		SETFAL42	SETFAL41	SETFAL40
R-0		R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
SETFAL39	SETFAL38	SETFAL37	SETFAL36	Reserved		SETFAL33	SETFAL32
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

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

Table 13. GPIO Set Falling Edge Interrupt Register (SET_FAL_TRIG*n*) Field Descriptions

Bit	Field[1]	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 2*I* + 1. This bit field configures the GPIO pins on GPIO bank 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 2*I*. This bit field configures the GPIO pins on GPIO banks 0 and 2.
		0	No effect.
		1	Interrupt is caused by a high-to-low transition on GPIO pin *n*.

[1] All of the fields are not populated. See Figure 18 and Figure 19 for the GPIO bit fields that are supported.

3.11 GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG*n*)

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 20, CLR_FAL_TRIG2 is shown in Figure 21, and described in Table 14. See Table 1 to determine the CLR_FAL_TRIG*n* bit associated with each GPIO bank and pin number.

Figure 20. GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG01)

31				27	26	25	24
Reserved					CLRFAL26	CLRFAL25	CLRFAL24
R-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
Reserved		CLRFAL13	CLRFAL12	CLRFAL11	CLRFAL10	Reserved	CLRFAL8
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R-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; R = Read only; -*n* = value after reset

Figure 21. GPIO Bank 2 Clear Falling Edge Interrupt Register (CLR_FAL_TRIG2)

31							16
Reserved							
R-0							

15				11	10	9	8
Reserved					CLRFAL42	CLRFAL41	CLRFAL40
R-0					R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CLRFAL39	CLRFAL38	CLRFAL37	CLRFAL36	Reserved		CLRFAL33	CLRFAL32
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

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

Table 14. GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG*n*) Field Descriptions

Bit	Field[(1)]	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 2*I* + 1. This bit field configures the GPIO pins on GPIO bank 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 2*I*. This bit field configures the GPIO pins on GPIO banks 0 and 2.
		0	No effect.
		1	No interrupt is caused by a high-to-low transition on GPIO pin *n*.

[(1)] All of the fields are not populated. See Figure 20 and Figure 21 for the GPIO bit fields that are supported.

3.12 GPIO Interrupt Status Register (INTSTAT*n*)

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 22, INTSTAT2 is shown in Figure 23, and described in Table 15. See Table 1 to determine the INTSTAT*n* bit associated with each GPIO bank and pin number.

Figure 22. GPIO Banks 0 and 1 Interrupt Status Register (INTSTAT01)

31				27	26	25	24
Reserved					STAT26	STAT25	STAT24
R-0					R/W-0	R/W-0	R/W-0

23	22	21	20	19	18	17	16
STAT23	STAT22	STAT21	STAT20	STAT19	STAT18	STAT17	STAT16
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
Reserved		STAT13	STAT12	STAT11	STAT10	Reserved	STAT8
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

7	6	5	4	3	2	1	0
STAT7	STAT6	STAT5	STAT4	STAT3	STAT2	STAT1	STAT0
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; R = Read only; -*n* = value after reset

Figure 23. GPIO Bank 2 Interrupt Status Register (INTSTAT2)

31							16
Reserved							
R-0							

15				11	10	9	8
Reserved					STAT42	STAT41	STAT40
R-0					R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
STAT39	STAT38	STAT37	STAT36	Reserved		STAT33	STAT32
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

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

Table 15. GPIO Interrupt Status Register (INTSTAT*n*) Field Descriptions

Bit	Field[1]	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*I* + 1. This bit field returns the status of GPIO pins on GPIO bank 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 2*I*. This bit field returns the status of GPIO pins on GPIO banks 0 and 2. 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*.

[1] All of the fields are not populated. See Figure 22 and Figure 23 for the GPIO bit fields that are supported.

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
Figure 4	Changed reset value of bits 31-27 to R-1.
	Changed reset value of bits 15-14 to R-1.
	Changed reset value of bit 9 to R-1.
Figure 5	Changed reset value of bits 31-11 to R-1.
	Changed reset value of bits 3-2 to R-1.

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
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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