

# TMS320C674x/OMAP-L1x Processor General-Purpose Input/Output (GPIO)

## User's Guide



Literature Number: SPRUFL8B

June 2010



<b>Preface</b> .....	<b>7</b>
<b>1 Introduction</b> .....	<b>9</b>
1.1 Purpose of the Peripheral .....	9
1.2 Features .....	9
1.3 Functional Block Diagram .....	9
1.4 Industry Standard(s) Compliance Statement .....	9
<b>2 Architecture</b> .....	<b>10</b>
2.1 Clock Control .....	10
2.2 Signal Descriptions .....	10
2.3 Pin Multiplexing .....	10
2.4 Endianness Considerations .....	10
2.5 GPIO Register Structure .....	11
2.6 Using a GPIO Signal as an Output .....	14
2.7 Using a GPIO Signal as an Input .....	15
2.8 Reset Considerations .....	15
2.9 Initialization .....	16
2.10 Interrupt Support .....	16
2.11 EDMA Event Support .....	17
2.12 Power Management .....	17
2.13 Emulation Considerations .....	17
<b>3 Registers</b> .....	<b>18</b>
3.1 Revision ID Register (REVID) .....	19
3.2 GPIO Interrupt Per-Bank Enable Register (BINTEN) .....	20
3.3 GPIO Direction Registers (DIR $n$ ) .....	21
3.4 GPIO Output Data Registers (OUT_DATA $n$ ) .....	23
3.5 GPIO Set Data Registers (SET_DATA $n$ ) .....	25
3.6 GPIO Clear Data Registers (CLR_DATA $n$ ) .....	27
3.7 GPIO Input Data Registers (IN_DATA $n$ ) .....	29
3.8 GPIO Set Rising Edge Interrupt Registers (SET_RIS_TRIG $n$ ) .....	31
3.9 GPIO Clear Rising Edge Interrupt Registers (CLR_RIS_TRIG $n$ ) .....	33
3.10 GPIO Set Falling Edge Interrupt Registers (SET_FAL_TRIG $n$ ) .....	35
3.11 GPIO Clear Falling Edge Interrupt Registers (CLR_FAL_TRIG $n$ ) .....	37
3.12 GPIO Interrupt Status Registers (INTSTAT $n$ ) .....	39
<b>Appendix A Revision History</b> .....	<b>41</b>

## List of Figures

1	GPIO Block Diagram .....	10
2	Revision ID Register (REVID) .....	19
3	GPIO Interrupt Per-Bank Enable Register (BINTEN) .....	20
4	GPIO Banks 0 and 1 Direction Register (DIR01) .....	21
5	GPIO Banks 2 and 3 Direction Register (DIR23) .....	21
6	GPIO Banks 4 and 5 Direction Register (DIR45) .....	21
7	GPIO Banks 6 and 7 Direction Register (DIR67) .....	21
8	GPIO Bank 8 Direction Register (DIR8) .....	22
9	GPIO Banks 0 and 1 Output Data Register (OUT_DATA01) .....	23
10	GPIO Banks 2 and 3 Output Data Register (OUT_DATA23) .....	23
11	GPIO Banks 4 and 5 Output Data Register (OUT_DATA45) .....	23
12	GPIO Banks 6 and 7 Output Data Register (OUT_DATA67) .....	23
13	GPIO Bank 8 Output Data Register (OUT_DATA8) .....	24
14	GPIO Banks 0 and 1 Set Data Register (SET_DATA01) .....	25
15	GPIO Banks 2 and 3 Set Data Register (SET_DATA23) .....	25
16	GPIO Banks 4 and 5 Set Data Register (SET_DATA45) .....	25
17	GPIO Banks 6 and 7 Set Data Register (SET_DATA67) .....	25
18	GPIO Bank 8 Set Data Register (SET_DATA8) .....	26
19	GPIO Banks 0 and 1 Clear Data Register (CLR_DATA01) .....	27
20	GPIO Banks 2 and 3 Clear Data Register (CLR_DATA23) .....	27
21	GPIO Banks 4 and 5 Clear Data Register (CLR_DATA45) .....	27
22	GPIO Banks 6 and 7 Clear Data Register (CLR_DATA67) .....	27
23	GPIO Bank 8 Clear Data Register (CLR_DATA8) .....	28
24	GPIO Banks 0 and 1 Input Data Register (IN_DATA01) .....	29
25	GPIO Banks 2 and 3 Input Data Register (IN_DATA23) .....	29
26	GPIO Banks 4 and 5 Input Data Register (IN_DATA45) .....	29
27	GPIO Banks 6 and 7 Input Data Register (IN_DATA67) .....	29
28	GPIO Bank 8 Input Data Register (IN_DATA8) .....	30
29	GPIO Banks 0 and 1 Set Rise Trigger Register (SET_RIS_TRIG01) .....	31
30	GPIO Banks 2 and 3 Set Rise Trigger Register (SET_RIS_TRIG23) .....	31
31	GPIO Banks 4 and 5 Set Rise Trigger Register (SET_RIS_TRIG45) .....	31
32	GPIO Banks 6 and 7 Set Rise Trigger Register (SET_RIS_TRIG67) .....	31
33	GPIO Bank 8 Set Rise Trigger Register (SET_RIS_TRIG8) .....	32
34	GPIO Banks 0 and 1 Clear Rise Trigger Register (CLR_RIS_TRIG01) .....	33
35	GPIO Banks 2 and 3 Clear Rise Trigger Register (CLR_RIS_TRIG23) .....	33
36	GPIO Banks 4 and 5 Clear Rise Trigger Register (CLR_RIS_TRIG45) .....	33
37	GPIO Banks 6 and 7 Clear Rise Trigger Register (CLR_RIS_TRIG67) .....	33
38	GPIO Bank 8 Clear Rise Trigger Register (CLR_RIS_TRIG8) .....	34
39	GPIO Banks 0 and 1 Set Rise Trigger Register (SET_FAL_TRIG01) .....	35
40	GPIO Banks 2 and 3 Set Rise Trigger Register (SET_FAL_TRIG23) .....	35
41	GPIO Banks 4 and 5 Set Rise Trigger Register (SET_FAL_TRIG45) .....	35
42	GPIO Banks 6 and 7 Set Rise Trigger Register (SET_FAL_TRIG67) .....	35
43	GPIO Bank 8 Set Rise Trigger Register (SET_FAL_TRIG8) .....	36
44	GPIO Banks 0 and 1 Clear Rise Trigger Register (CLR_FAL_TRIG01) .....	37
45	GPIO Banks 2 and 3 Clear Rise Trigger Register (CLR_FAL_TRIG23) .....	37
46	GPIO Banks 4 and 5 Clear Rise Trigger Register (CLR_FAL_TRIG45) .....	37
47	GPIO Banks 6 and 7 Clear Rise Trigger Register (CLR_FAL_TRIG67) .....	37

48	GPIO Bank 8 Clear Rise Trigger Register (CLR_FAL_TRIG8) .....	38
49	GPIO Banks 0 and 1 Interrupt Status Register (INTSTAT01) .....	39
50	GPIO Banks 2 and 3 Interrupt Status Register (INTSTAT23) .....	39
51	GPIO Banks 4 and 5 Interrupt Status Register (INTSTAT45) .....	39
52	GPIO Banks 6 and 7 Interrupt Status Register (INTSTAT67) .....	39
53	GPIO Bank 8 Interrupt Status Register (INTSTAT8) .....	40

---

## List of Tables

1	GPIO Register Bits and Banks Associated With GPIO Signals .....	11
2	GPIO Registers .....	18
3	Revision ID Register (REVID) Field Descriptions .....	19
4	GPIO Interrupt Per-Bank Enable Register (BINTEN) Field Descriptions .....	20
5	GPIO Direction Register (DIR $n$ ) Field Descriptions .....	22
6	GPIO Output Data Register (OUT_DATA $n$ ) Field Descriptions .....	24
7	GPIO Set Data Register (SET_DATA $n$ ) Field Descriptions.....	26
8	GPIO Clear Data Register (CLR_DATA $n$ ) Field Descriptions .....	28
9	GPIO Input Data Register (IN_DATA $n$ ) Field Descriptions .....	30
10	GPIO Set Rising Edge Trigger Interrupt Register (SET_RIS_TRIG $n$ ) Field Descriptions .....	32
11	GPIO Clear Rising Edge Interrupt Register (CLR_RIS_TRIG $n$ ) Field Descriptions .....	34
12	GPIO Set Falling Edge Trigger Interrupt Register (SET_FAL_TRIG $n$ ) Field Descriptions .....	36
13	GPIO Clear Falling Edge Interrupt Register (CLR_FAL_TRIG $n$ ) Field Descriptions .....	38
14	GPIO Interrupt Status Register (INTSTAT $n$ ) Field Descriptions .....	40
15	Document Revision History .....	41

## Read This First

---

---

---

### About This Manual

Describes the general-purpose input/output (GPIO) peripheral. 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 TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x Applications Processors. Copies of these documents are available on the Internet at [www.ti.com](http://www.ti.com). *Tip:* Enter the literature number in the search box provided at [www.ti.com](http://www.ti.com).

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

**[SPRUGM5](#)** — ***TMS320C6742 DSP System Reference Guide***. Describes the C6742 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, and system configuration module.

**[SPRUGJ0](#)** — ***TMS320C6743 DSP System Reference Guide***. Describes the System-on-Chip (SoC) including the C6743 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, and system configuration module.

**[SPRUFK4](#)** — ***TMS320C6745/C6747 DSP System Reference Guide***. Describes the System-on-Chip (SoC) including the C6745/C6747 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, and system configuration module.

**[SPRUGM6](#)** — ***TMS320C6746 DSP System Reference Guide***. Describes the C6746 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, and system configuration module.

**[SPRUGJ7](#)** — ***TMS320C6748 DSP System Reference Guide***. Describes the C6748 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, and system configuration module.

**[SPRUG84](#)** — ***OMAP-L137 Applications Processor System Reference Guide***. Describes the System-on-Chip (SoC) including the ARM subsystem, DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

- [SPRUGM7](#)** — ***OMAP-L138 Applications Processor System Reference Guide***. Describes the System-on-Chip (SoC) including the ARM subsystem, DSP subsystem, system memory, device clocking, phase-locked loop controller (PLL), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.
- [SPRUFK9](#)** — ***TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide***. Provides an overview and briefly describes the peripherals available on the TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x Applications Processors.
- [SPRUFK5](#)** — ***TMS320C674x DSP Megamodule Reference Guide***. Describes the TMS320C674x 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.
- [SPRUF8](#)** — ***TMS320C674x DSP CPU and Instruction Set Reference Guide***. Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C674x digital signal processors (DSPs). The C674x DSP is an enhancement of the C64x+ and C67x+ DSPs with added functionality and an expanded instruction set.
- [SPRUG82](#)** — ***TMS320C674x DSP Cache User's Guide***. Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C674x digital signal processor (DSP) can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C674x DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.



# **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-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.

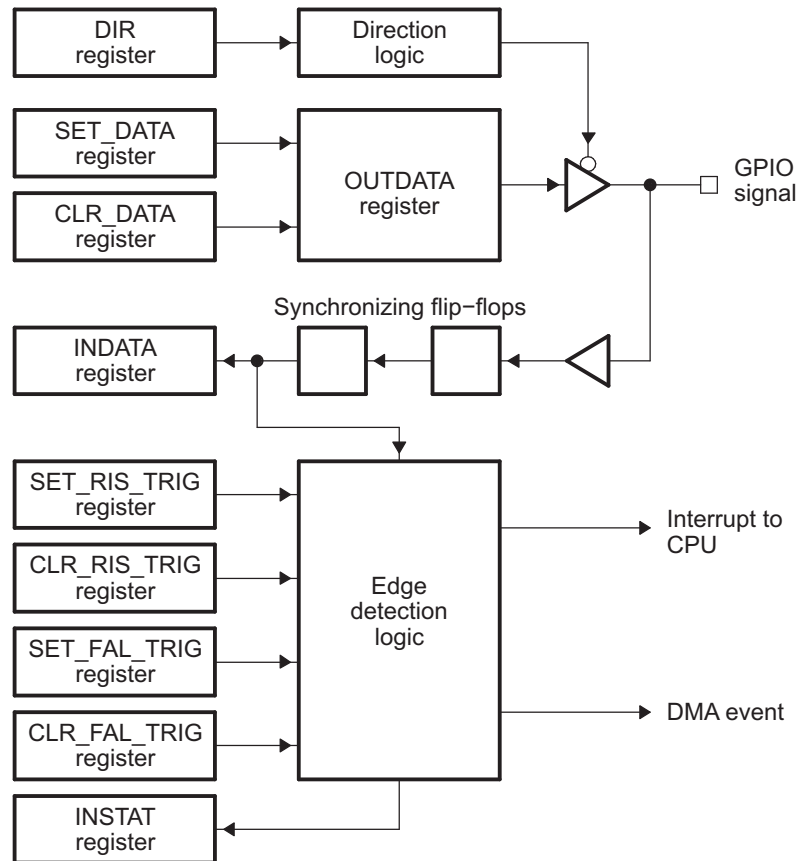
- Output set/clear functionality through separate data set and clear registers allows multiple software processes to control GPIO signals without critical section protection.
- Set/clear functionality through writing to a single output data register is also supported.
- Separate input/output registers
  - 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.

### **1.3 Functional Block Diagram**

[Figure 1](#) shows a block diagram of the GPIO peripheral.

### **1.4 Industry Standard(s) Compliance Statement**

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

**Figure 1. GPIO Block Diagram**


## 2 Architecture

The following sections describe the GPIO peripheral.

### 2.1 Clock Control

The input clock to the GPIO peripheral is indicated in the device datasheet. The maximum operating speed of the GPIO peripheral is limited by system-level latencies. More specifically, how quickly the GPIO registers can be written to or read from.

### 2.2 Signal Descriptions

The number of GPIO signals supported will vary between devices. For information on the number of signals supported and the package pinout of each GPIO signal, see your device-specific data manual.

### 2.3 Pin Multiplexing

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

### 2.4 Endianness Considerations

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

## 2.5 GPIO Register Structure

The GPIO signals are grouped by banks of 16 signals per bank. Each bank of GPIO signals has several registers with various control fields for each GPIO signal. Each 32-bit GPIO control register controls a pair of GPIO banks.

The register names for each bank of control registers (or pair of banks of GPIO bits) are all of the form *register\_nameXY*, where *X* and *Y* are the two banks of GPIO bits controlled, such as 01, 23, 45, etc. The register fields associated with each GPIO are all of the form *BkPj*, where *k* is the GPIO bank and *j* is the pin number within the GPIO bank. For example, for GP2[5], which is located in GPIO bank 2, the control register names are of the form *register\_name23*, and the register field associated with GP2[5] is GP2P5.

[Table 1](#) shows the banks and register control bit information associated with each GPIO pin for up to 144 supportable pins. The table is not indicative of how many GPIO pins are supported on a device; it is only a reference for what register and field mappings look like for the first 144 supportable GPIO pins. For devices with less than 144 GPIO pins, assume that the extraneous fields and registers listed in the table are Reserved with no function. For devices with more than 144 GPIO pins, additional control registers and fields should be appended using the same numbering scheme in the table. Detailed information regarding the specific register names for each bank and the contents and function of these registers is presented in [Section 3](#).

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

GPIO Pin Number	GPIO Signal Name	Bank Number	Control Registers	Register Bit	Register Field
1	GP0[0]	0	<i>register_name01</i>	Bit 0	GP0P0
2	GP0[1]	0	<i>register_name01</i>	Bit 1	GP0P1
3	GP0[2]	0	<i>register_name01</i>	Bit 2	GP0P2
4	GP0[3]	0	<i>register_name01</i>	Bit 3	GP0P3
5	GP0[4]	0	<i>register_name01</i>	Bit 4	GP0P4
6	GP0[5]	0	<i>register_name01</i>	Bit 5	GP0P5
7	GP0[6]	0	<i>register_name01</i>	Bit 6	GP0P6
8	GP0[7]	0	<i>register_name01</i>	Bit 7	GP0P7
9	GP0[8]	0	<i>register_name01</i>	Bit 8	GP0P8
10	GP0[9]	0	<i>register_name01</i>	Bit 9	GP0P9
11	GP0[10]	0	<i>register_name01</i>	Bit 10	GP0P10
12	GP0[11]	0	<i>register_name01</i>	Bit 11	GP0P11
13	GP0[12]	0	<i>register_name01</i>	Bit 12	GP0P12
14	GP0[13]	0	<i>register_name01</i>	Bit 13	GP0P13
15	GP0[14]	0	<i>register_name01</i>	Bit 14	GP0P14
16	GP0[15]	0	<i>register_name01</i>	Bit 15	GP0P15
17	GP1[0]	1	<i>register_name01</i>	Bit 16	GP1P0
18	GP1[1]	1	<i>register_name01</i>	Bit 17	GP1P1
19	GP1[2]	1	<i>register_name01</i>	Bit 18	GP1P2
20	GP1[3]	1	<i>register_name01</i>	Bit 19	GP1P3
21	GP1[4]	1	<i>register_name01</i>	Bit 20	GP1P4
22	GP1[5]	1	<i>register_name01</i>	Bit 21	GP1P5
23	GP1[6]	1	<i>register_name01</i>	Bit 22	GP1P6
24	GP1[7]	1	<i>register_name01</i>	Bit 23	GP1P7
25	GP1[8]	1	<i>register_name01</i>	Bit 24	GP1P8
26	GP1[9]	1	<i>register_name01</i>	Bit 25	GP1P9
27	GP1[10]	1	<i>register_name01</i>	Bit 26	GP1P10
28	GP1[11]	1	<i>register_name01</i>	Bit 27	GP1P11
29	GP1[12]	1	<i>register_name01</i>	Bit 28	GP1P12
30	GP1[13]	1	<i>register_name01</i>	Bit 29	GP1P13

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

GPIO Pin Number	GPIO Signal Name	Bank Number	Control Registers	Register Bit	Register Field
31	GP1[14]	1	<i>register_name01</i>	Bit 30	GP1P14
32	GP1[15]	1	<i>register_name01</i>	Bit 31	GP1P15
33	GP2[0]	2	<i>register_name23</i>	Bit 0	GP2P0
34	GP2[1]	2	<i>register_name23</i>	Bit 1	GP2P1
35	GP2[2]	2	<i>register_name23</i>	Bit 2	GP2P2
36	GP2[3]	2	<i>register_name23</i>	Bit 3	GP2P3
37	GP2[4]	2	<i>register_name23</i>	Bit 4	GP2P4
38	GP2[5]	2	<i>register_name23</i>	Bit 5	GP2P5
39	GP2[6]	2	<i>register_name23</i>	Bit 6	GP2P6
40	GP2[7]	2	<i>register_name23</i>	Bit 7	GP2P7
41	GP2[8]	2	<i>register_name23</i>	Bit 8	GP2P8
42	GP2[9]	2	<i>register_name23</i>	Bit 9	GP2P9
43	GP2[10]	2	<i>register_name23</i>	Bit 10	GP2P10
44	GP2[11]	2	<i>register_name23</i>	Bit 11	GP2P11
45	GP2[12]	2	<i>register_name23</i>	Bit 12	GP2P12
46	GP2[13]	2	<i>register_name23</i>	Bit 13	GP2P13
47	GP2[14]	2	<i>register_name23</i>	Bit 14	GP2P14
48	GP2[15]	2	<i>register_name23</i>	Bit 15	GP2P15
49	GP3[0]	3	<i>register_name23</i>	Bit 16	GP3P0
50	GP3[1]	3	<i>register_name23</i>	Bit 17	GP3P1
51	GP3[2]	3	<i>register_name23</i>	Bit 18	GP3P2
52	GP3[3]	3	<i>register_name23</i>	Bit 19	GP3P3
53	GP3[4]	3	<i>register_name23</i>	Bit 20	GP3P4
54	GP3[5]	3	<i>register_name23</i>	Bit 21	GP3P5
55	GP3[6]	3	<i>register_name23</i>	Bit 22	GP3P6
56	GP3[7]	3	<i>register_name23</i>	Bit 23	GP3P7
57	GP3[8]	3	<i>register_name23</i>	Bit 24	GP3P8
58	GP3[9]	3	<i>register_name23</i>	Bit 25	GP3P9
59	GP3[10]	3	<i>register_name23</i>	Bit 26	GP3P10
60	GP3[11]	3	<i>register_name23</i>	Bit 27	GP3P11
61	GP3[12]	3	<i>register_name23</i>	Bit 28	GP3P12
62	GP3[13]	3	<i>register_name23</i>	Bit 29	GP3P13
63	GP3[14]	3	<i>register_name23</i>	Bit 30	GP3P14
64	GP3[15]	3	<i>register_name23</i>	Bit 31	GP3P15
65	GP4[0]	4	<i>register_name45</i>	Bit 0	GP4P0
66	GP4[1]	4	<i>register_name45</i>	Bit 1	GP4P1
67	GP4[2]	4	<i>register_name45</i>	Bit 2	GP4P2
68	GP4[3]	4	<i>register_name45</i>	Bit 3	GP4P3
69	GP4[4]	4	<i>register_name45</i>	Bit 4	GP4P4
70	GP4[5]	4	<i>register_name45</i>	Bit 5	GP4P5
71	GP4[6]	4	<i>register_name45</i>	Bit 6	GP4P6
72	GP4[7]	4	<i>register_name45</i>	Bit 7	GP4P7
73	GP4[8]	4	<i>register_name45</i>	Bit 8	GP4P8
74	GP4[9]	4	<i>register_name45</i>	Bit 9	GP4P9
75	GP4[10]	4	<i>register_name45</i>	Bit 10	GP4P10
76	GP4[11]	4	<i>register_name45</i>	Bit 11	GP4P11
77	GP4[12]	4	<i>register_name45</i>	Bit 12	GP4P12

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

GPIO Pin Number	GPIO Signal Name	Bank Number	Control Registers	Register Bit	Register Field
78	GP4[13]	4	<i>register_name45</i>	Bit 13	GP4P13
79	GP4[14]	4	<i>register_name45</i>	Bit 14	GP4P14
80	GP4[15]	4	<i>register_name45</i>	Bit 15	GP4P15
81	GP5[0]	5	<i>register_name45</i>	Bit 16	GP5P0
82	GP5[1]	5	<i>register_name45</i>	Bit 17	GP5P1
83	GP5[2]	5	<i>register_name45</i>	Bit 18	GP5P2
84	GP5[3]	5	<i>register_name45</i>	Bit 19	GP5P3
85	GP5[4]	5	<i>register_name45</i>	Bit 20	GP5P4
86	GP5[5]	5	<i>register_name45</i>	Bit 21	GP5P5
87	GP5[6]	5	<i>register_name45</i>	Bit 22	GP5P6
88	GP5[7]	5	<i>register_name45</i>	Bit 23	GP5P7
89	GP5[8]	5	<i>register_name45</i>	Bit 24	GP5P8
90	GP5[9]	5	<i>register_name45</i>	Bit 25	GP5P9
91	GP5[10]	5	<i>register_name45</i>	Bit 26	GP5P10
92	GP5[11]	5	<i>register_name45</i>	Bit 27	GP5P11
93	GP5[12]	5	<i>register_name45</i>	Bit 28	GP5P12
94	GP5[13]	5	<i>register_name45</i>	Bit 29	GP5P13
95	GP5[14]	5	<i>register_name45</i>	Bit 30	GP5P14
96	GP5[15]	5	<i>register_name45</i>	Bit 31	GP5P15
97	GP6[0]	6	<i>register_name67</i>	Bit 0	GP6P0
98	GP6[1]	6	<i>register_name67</i>	Bit 1	GP6P1
99	GP6[2]	6	<i>register_name67</i>	Bit 2	GP6P2
100	GP6[3]	6	<i>register_name67</i>	Bit 3	GP6P3
101	GP6[4]	6	<i>register_name67</i>	Bit 4	GP6P4
102	GP6[5]	6	<i>register_name67</i>	Bit 5	GP6P5
103	GP6[6]	6	<i>register_name67</i>	Bit 6	GP6P6
104	GP6[7]	6	<i>register_name67</i>	Bit 7	GP6P7
105	GP6[8]	6	<i>register_name67</i>	Bit 8	GP6P8
106	GP6[9]	6	<i>register_name67</i>	Bit 9	GP6P9
107	GP6[10]	6	<i>register_name67</i>	Bit 10	GP6P10
108	GP6[11]	6	<i>register_name67</i>	Bit 11	GP6P11
109	GP6[12]	6	<i>register_name67</i>	Bit 12	GP6P12
110	GP6[13]	6	<i>register_name67</i>	Bit 13	GP6P13
111	GP6[14]	6	<i>register_name67</i>	Bit 14	GP6P14
112	GP6[15]	6	<i>register_name67</i>	Bit 15	GP6P15
113	GP7[0]	7	<i>register_name67</i>	Bit 16	GP7P0
114	GP7[1]	7	<i>register_name67</i>	Bit 17	GP7P1
115	GP7[2]	7	<i>register_name67</i>	Bit 18	GP7P2
116	GP7[3]	7	<i>register_name67</i>	Bit 19	GP7P3
117	GP7[4]	7	<i>register_name67</i>	Bit 20	GP7P4
118	GP7[5]	7	<i>register_name67</i>	Bit 21	GP7P5
119	GP7[6]	7	<i>register_name67</i>	Bit 22	GP7P6
120	GP7[7]	7	<i>register_name67</i>	Bit 23	GP7P7
121	GP7[8]	7	<i>register_name67</i>	Bit 24	GP7P8
122	GP7[9]	7	<i>register_name67</i>	Bit 25	GP7P9
123	GP7[10]	7	<i>register_name67</i>	Bit 26	GP7P10
124	GP7[11]	7	<i>register_name67</i>	Bit 27	GP7P11

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

GPIO Pin Number	GPIO Signal Name	Bank Number	Control Registers	Register Bit	Register Field
125	GP7[12]	7	<i>register_name67</i>	Bit 28	GP7P12
126	GP7[13]	7	<i>register_name67</i>	Bit 29	GP7P13
127	GP7[14]	7	<i>register_name67</i>	Bit 30	GP7P14
128	GP7[15]	7	<i>register_name67</i>	Bit 31	GP7P15
129	GP8[0]	8	<i>register_name8</i>	Bit 0	GP8P0
130	GP8[1]	8	<i>register_name8</i>	Bit 1	GP8P1
131	GP8[2]	8	<i>register_name8</i>	Bit 2	GP8P2
132	GP8[3]	8	<i>register_name8</i>	Bit 3	GP8P3
133	GP8[4]	8	<i>register_name8</i>	Bit 4	GP8P4
134	GP8[5]	8	<i>register_name8</i>	Bit 5	GP8P5
135	GP8[6]	8	<i>register_name8</i>	Bit 6	GP8P6
136	GP8[7]	8	<i>register_name8</i>	Bit 7	GP8P7
137	GP8[8]	8	<i>register_name8</i>	Bit 8	GP8P8
138	GP8[9]	8	<i>register_name8</i>	Bit 9	GP8P9
139	GP8[10]	8	<i>register_name8</i>	Bit 10	GP8P10
140	GP8[11]	8	<i>register_name8</i>	Bit 11	GP8P11
141	GP8[12]	8	<i>register_name8</i>	Bit 12	GP8P12
142	GP8[13]	8	<i>register_name8</i>	Bit 13	GP8P13
143	GP8[14]	8	<i>register_name8</i>	Bit 14	GP8P14
144	GP8[15]	8	<i>register_name8</i>	Bit 15	GP8P15

## 2.6 Using a GPIO Signal as an Output

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

### 2.6.1 Configuring a GPIO Output Signal

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

### 2.6.2 Controlling the GPIO Output Signal State

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

1. GPIO set data register (SET\_DATA) controls driving GPIO signals high.
2. GPIO clear data register (CLR\_DATA) controls driving GPIO signals low.
3. 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](#).

### 2.6.2.1 Driving a GPIO Output Signal High

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

- Write a logic 1 to the bit in SET\_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.

### 2.6.2.2 Driving a GPIO Output Signal Low

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

- Write a logic 1 to the bit in CLR\_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.

## 2.7 Using a GPIO Signal as an Input

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

### 2.7.1 Configuring a GPIO Input Signal

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

### 2.7.2 Reading a GPIO Input Signal

The current state of the GPIO signals is read using the GPIO input data register (IN\_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 your device-specific data manual.

To use GPIO input signals as interrupt sources, see [Section 2.10](#).

## 2.8 Reset Considerations

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

### 2.8.1 Software Reset Considerations

A software reset (such as a reset initiated through the emulator) does not modify the configuration and state of the GPIO signals. A reset invoked via the Power and Sleep Controller (PSC) (GPIO clock disable, PSC reset, followed by GPIO clock enable) will result in the default configuration register settings. For details on the PSC, see your device-specific *System Reference Guide*.

## 2.8.2 Hardware Reset Considerations

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

## 2.9 Initialization

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

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

The GPIO module is now ready to perform data transactions.

## 2.10 Interrupt Support

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

### 2.10.1 Interrupt Events and Requests

All GPIO signals can be configured to generate interrupts. The device supports interrupts from single GPIO signals, interrupts from banks of GPIO signals, or both. The interrupt mapping from the GPIO peripheral to the CPU can be found in your device-specific *System Reference Guide*.

Note that the GPIO interrupts may also be used to provide synchronization events to the DMA controller. See your device-specific *System Reference Guide* for additional information.

### 2.10.2 Enabling GPIO Interrupt Events

GPIO interrupt events are enabled in banks of 16 by setting the appropriate bit(s) in the GPIO interrupt per-bank enable register (BINTEN). For example, to enable bank 0 interrupts (events from GP0[15-0]), set bit 0 in BINTEN; to enable bank 3 interrupts (events from GP3[15-0]), set bit 3 in BINTEN.

For detailed information on BINTEN, see [Section 3](#).

### 2.10.3 Configuring GPIO Interrupt Edge Triggering

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

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

1. The GPIO set rising edge interrupt register (SET\_RIS\_TRIG) enables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
2. The GPIO clear rising edge interrupt register (CLR\_RIS\_TRIG) disables GPIO interrupts on the occurrence of a rising edge on the GPIO signal.
3. The GPIO set falling edge interrupt register (SET\_FAL\_TRIG) enables GPIO interrupts on the occurrence of a falling edge on the GPIO signal.
4. 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.

#### 2.10.4 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 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](#).

#### 2.10.5 Interrupt Multiplexing

GPIO interrupts may be multiplexed with other interrupt functions on the device. The interrupt events are listed in your device-specific *System Reference Guide*.

#### 2.11 EDMA Event Support

The GPIO peripheral may provide synchronization events to the DMA controller. The DMA events supported are listed in your device-specific *System Reference Guide*.

#### 2.12 Power Management

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

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

#### 2.13 Emulation Considerations

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

### 3 Registers

**Table 2** lists the memory-mapped registers for the general-purpose input/output (GPIO). The table enumerates the registers required to support 144 GPIO pins, however not all devices will support 144 GPIO pins. For devices with less than 144 GPIO pins, assume that the extraneous fields and registers are Reserved and serve no function. For devices with more than 144 GPIO pins, append registers and fields as necessary using the address offset scheme in the table. See your device-specific data manual for the number of GPIO pins supported and the base memory address for these registers.

**Table 2. GPIO Registers**

Offset	Acronym	Register Description	Section
0h	REVID	Revision ID Register	<a href="#">Section 3.1</a>
8h	BINTEN	GPIO Interrupt Per-Bank Enable Register	<a href="#">Section 3.2</a>
<b>GPIO Banks 0 and 1</b>			
10h	DIR01	GPIO Banks 0 and 1 Direction Register	<a href="#">Section 3.3</a>
14h	OUT_DATA01	GPIO Banks 0 and 1 Output Data Register	<a href="#">Section 3.4</a>
18h	SET_DATA01	GPIO Banks 0 and 1 Set Data Register	<a href="#">Section 3.5</a>
1Ch	CLR_DATA01	GPIO Banks 0 and 1 Clear Data Register	<a href="#">Section 3.6</a>
20h	IN_DATA01	GPIO Banks 0 and 1 Input Data Register	<a href="#">Section 3.7</a>
24h	SET_RIS_TRIG01	GPIO Banks 0 and 1 Set Rising Edge Interrupt Register	<a href="#">Section 3.8</a>
28h	CLR_RIS_TRIG01	GPIO Banks 0 and 1 Clear Rising Edge Interrupt Register	<a href="#">Section 3.9</a>
2Ch	SET_FAL_TRIG01	GPIO Banks 0 and 1 Set Falling Edge Interrupt Register	<a href="#">Section 3.10</a>
30h	CLR_FAL_TRIG01	GPIO Banks 0 and 1 Clear Falling Edge Interrupt Register	<a href="#">Section 3.11</a>
34h	INTSTAT01	GPIO Banks 0 and 1 Interrupt Status Register	<a href="#">Section 3.12</a>
<b>GPIO Banks 2 and 3</b>			
38h	DIR23	GPIO Banks 2 and 3 Direction Register	<a href="#">Section 3.3</a>
3Ch	OUT_DATA23	GPIO Banks 2 and 3 Output Data Register	<a href="#">Section 3.4</a>
40h	SET_DATA23	GPIO Banks 2 and 3 Set Data Register	<a href="#">Section 3.5</a>
44h	CLR_DATA23	GPIO Banks 2 and 3 Clear Data Register	<a href="#">Section 3.6</a>
48h	IN_DATA23	GPIO Banks 2 and 3 Input Data Register	<a href="#">Section 3.7</a>
4Ch	SET_RIS_TRIG23	GPIO Banks 2 and 3 Set Rising Edge Interrupt Register	<a href="#">Section 3.8</a>
50h	CLR_RIS_TRIG23	GPIO Banks 2 and 3 Clear Rising Edge Interrupt Register	<a href="#">Section 3.9</a>
54h	SET_FAL_TRIG23	GPIO Banks 2 and 3 Set Falling Edge Interrupt Register	<a href="#">Section 3.10</a>
58h	CLR_FAL_TRIG23	GPIO Banks 2 and 3 Clear Falling Edge Interrupt Register	<a href="#">Section 3.11</a>
5Ch	INTSTAT23	GPIO Banks 2 and 3 Interrupt Status Register	<a href="#">Section 3.12</a>
<b>GPIO Banks 4 and 5</b>			
60h	DIR45	GPIO Banks 4 and 5 Direction Register	<a href="#">Section 3.3</a>
64h	OUT_DATA45	GPIO Banks 4 and 5 Output Data Register	<a href="#">Section 3.4</a>
68h	SET_DATA45	GPIO Banks 4 and 5 Set Data Register	<a href="#">Section 3.5</a>
6Ch	CLR_DATA45	GPIO Banks 4 and 5 Clear Data Register	<a href="#">Section 3.6</a>
70h	IN_DATA45	GPIO Banks 4 and 5 Input Data Register	<a href="#">Section 3.7</a>
74h	SET_RIS_TRIG45	GPIO Banks 4 and 5 Set Rising Edge Interrupt Register	<a href="#">Section 3.8</a>
78h	CLR_RIS_TRIG45	GPIO Banks 4 and 5 Clear Rising Edge Interrupt Register	<a href="#">Section 3.9</a>
7Ch	SET_FAL_TRIG45	GPIO Banks 4 and 5 Set Falling Edge Interrupt Register	<a href="#">Section 3.10</a>
80h	CLR_FAL_TRIG45	GPIO Banks 4 and 5 Clear Falling Edge Interrupt Register	<a href="#">Section 3.11</a>
84h	INTSTAT45	GPIO Banks 4 and 5 Interrupt Status Register	<a href="#">Section 3.12</a>

**Table 2. GPIO Registers (continued)**

Offset	Acronym	Register Description	Section
<b>GPIO Banks 6 and 7</b>			
88h	DIR67	GPIO Banks 6 and 7 Direction Register	<a href="#">Section 3.3</a>
8Ch	OUT_DATA67	GPIO Banks 6 and 7 Output Data Register	<a href="#">Section 3.4</a>
90h	SET_DATA67	GPIO Banks 6 and 7 Set Data Register	<a href="#">Section 3.5</a>
94h	CLR_DATA67	GPIO Banks 6 and 7 Clear Data Register	<a href="#">Section 3.6</a>
98h	IN_DATA67	GPIO Banks 6 and 7 Input Data Register	<a href="#">Section 3.7</a>
9Ch	SET_RIS_TRIG67	GPIO Banks 6 and 7 Set Rising Edge Interrupt Register	<a href="#">Section 3.8</a>
A0h	CLR_RIS_TRIG67	GPIO Banks 6 and 7 Clear Rising Edge Interrupt Register	<a href="#">Section 3.9</a>
A4h	SET_FAL_TRIG67	GPIO Banks 6 and 7 Set Falling Edge Interrupt Register	<a href="#">Section 3.10</a>
A8h	CLR_FAL_TRIG67	GPIO Banks 6 and 7 Clear Falling Edge Interrupt Register	<a href="#">Section 3.11</a>
ACH	INTSTAT67	GPIO Banks 6 and 7 Interrupt Status Register	<a href="#">Section 3.12</a>
<b>GPIO Bank 8</b>			
B0h	DIR8	GPIO Bank 8 Direction Register	<a href="#">Section 3.3</a>
B4h	OUT_DATA8	GPIO Bank 8 Output Data Register	<a href="#">Section 3.4</a>
B8h	SET_DATA8	GPIO Bank 8 Set Data Register	<a href="#">Section 3.5</a>
BCh	CLR_DATA8	GPIO Bank 8 Clear Data Register	<a href="#">Section 3.6</a>
C0h	IN_DATA8	GPIO Bank 8 Input Data Register	<a href="#">Section 3.7</a>
C4h	SET_RIS_TRIG8	GPIO Bank 8 Set Rising Edge Interrupt Register	<a href="#">Section 3.8</a>
C8h	CLR_RIS_TRIG8	GPIO Bank 8 Clear Rising Edge Interrupt Register	<a href="#">Section 3.9</a>
CCh	SET_FAL_TRIG8	GPIO Bank 8 Set Falling Edge Interrupt Register	<a href="#">Section 3.10</a>
D0h	CLR_FAL_TRIG8	GPIO Bank 8 Clear Falling Edge Interrupt Register	<a href="#">Section 3.11</a>
D4h	INTSTAT8	GPIO Bank 8 Interrupt Status Register	<a href="#">Section 3.12</a>

### 3.1 Revision ID Register (REVID)

The revision ID register (REVID) contains the peripheral version information. REVID is shown in [Figure 2](#) and described in [Table 3](#).

**Figure 2. Revision ID Register (REVID)**

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

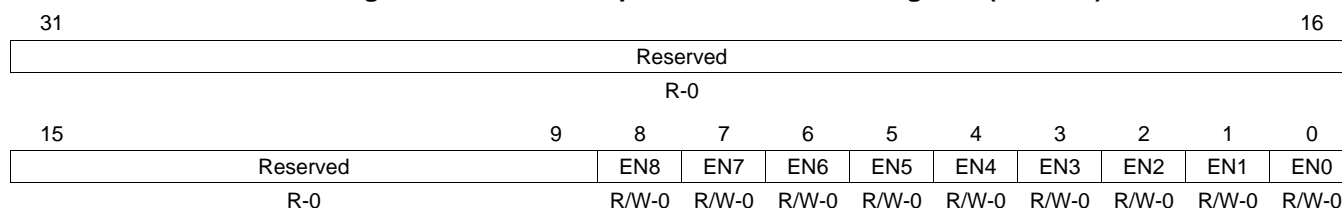
**Table 3. Revision ID Register (REVID) Field Descriptions**

Bit	Field	Value	Description
31-0	REV	4483 0105h	Peripheral Revision

### 3.2 GPIO Interrupt Per-Bank Enable Register (BINTEN)

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

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



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

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

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	EN8	0 1	Bank 8 interrupt enable is used to disable or enable the bank 8 interrupts (events from GP8[15-0]). Bank 8 interrupts are disabled. Bank 8 interrupts are enabled.
7	EN7	0 1	Bank 7 interrupt enable is used to disable or enable the bank 7 interrupts (events from GP7[15-0]). Bank 7 interrupts are disabled. Bank 7 interrupts are enabled.
6	EN6	0 1	Bank 6 interrupt enable is used to disable or enable the bank 6 interrupts (events from GP6[15-0]). Bank 6 interrupts are disabled. Bank 6 interrupts are enabled.
5	EN5	0 1	Bank 5 interrupt enable is used to disable or enable the bank 5 interrupts (events from GP5[15-0]). Bank 5 interrupts are disabled. Bank 5 interrupts are enabled.
4	EN4	0 1	Bank 4 interrupt enable is used to disable or enable the bank 4 interrupts (events from GP4[15-0]). Bank 4 interrupts are disabled. Bank 4 interrupts are enabled.
3	EN3	0 1	Bank 3 interrupt enable is used to disable or enable the bank 3 interrupts (events from GP3[15-0]). Bank 3 interrupts are disabled. Bank 3 interrupts are enabled.
2	EN2	0 1	Bank 2 interrupt enable is used to disable or enable the bank 2 interrupts (events from GP2[15-0]). Bank 2 interrupts are disabled. Bank 2 interrupts are enabled.
1	EN1	0 1	Bank 1 interrupt enable is used to disable or enable the bank 1 interrupts (events from GP1[15-0]). Bank 1 interrupts are disabled. Bank 1 interrupts are enabled.
0	EN0	0 1	Bank 0 interrupt enable is used to disable or enable the bank 0 interrupts (events from GP0[15-0]). Bank 0 interrupts are disabled. Bank 0 interrupts are enabled.

### 3.3 GPIO Direction Registers (DIR<sub>*n*</sub>)

The GPIO direction register (DIR<sub>*n*</sub>) determines if GPIO pin *j* in GPIO bank *k* 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, DIR67 is shown in Figure 7, DIR8 is shown in Figure 8, and described in Table 5. See Table 1 to determine the DIR<sub>*n*</sub> bit associated with each GPIO bank and pin number.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
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	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-1															

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

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

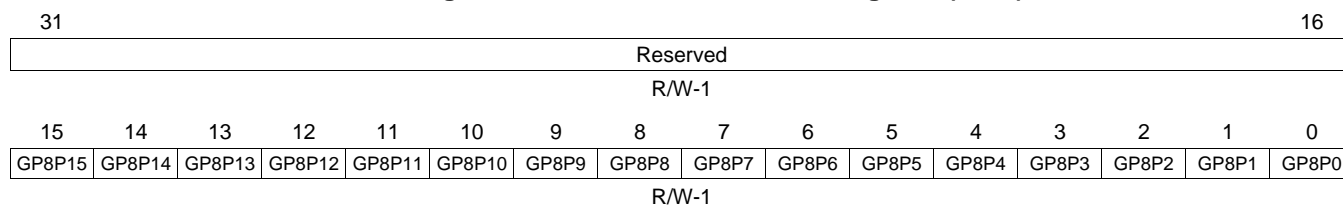
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-1															

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

**Figure 7. GPIO Banks 6 and 7 Direction Register (DIR67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-1															

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

**Figure 8. GPIO Bank 8 Direction Register (DIR8)**


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

**Table 5. GPIO Direction Register (DIR<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Direction of pin GPk[j]. The GPkPj bit is used to control the direction (output = 0, input = 1) of pin j in GPIO bankk.
		1	GPk[j] is an output.
			GPk[j] is an input.

### 3.4 GPIO Output Data Registers (OUT\_DATA<sub>n</sub>)

The GPIO output data register (OUT\_DATA<sub>n</sub>) determines the value driven on the corresponding GPIO pin *j* in GPIO bank *k*, if the pin is configured as an output (DIR<sub>n</sub> = 0). Writes do not affect pins not configured as GPIO outputs. The bits in OUT\_DATA<sub>n</sub> are set or cleared by writing directly to this register. A read of OUT\_DATA<sub>n</sub> 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 9, OUT\_DATA23 is shown in Figure 10, OUT\_DATA45 is shown in Figure 11, OUT\_DATA67 is shown in Figure 12, OUT\_DATA8 is shown in Figure 13, and described in Table 6. See Table 1 to determine the OUT\_DATA<sub>n</sub> bit associated with each GPIO bank and pin number.

**Figure 9. GPIO Banks 0 and 1 Output Data Register (OUT\_DATA01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 10. GPIO Banks 2 and 3 Output Data Register (OUT\_DATA23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 11. GPIO Banks 4 and 5 Output Data Register (OUT\_DATA45)**

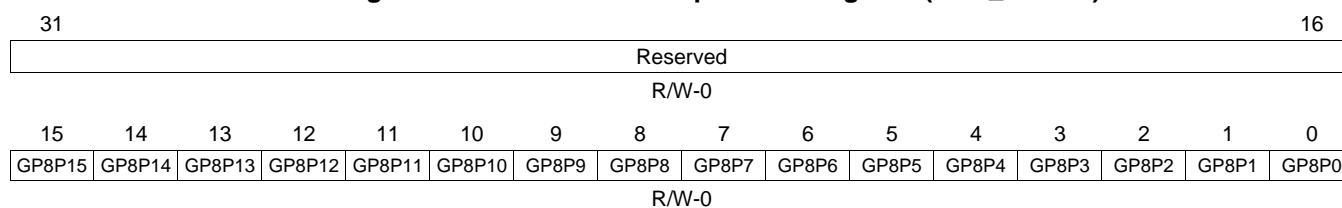
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 12. GPIO Banks 6 and 7 Output Data Register (OUT\_DATA67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 13. GPIO Bank 8 Output Data Register (OUT\_DATA8)**


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

**Table 6. GPIO Output Data Register (OUT\_DATA<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Output drive state of GPk[j]. The GPkPj bit is used to drive the output (low = 0, high = 1) of pin j in GPIO bankk. The GPkPj bit is ignored when GPk[j] is configured as an input.
		1	GPk[j] is driven low.
			GPk[j] is driven high.



### 3.5 GPIO Set Data Registers (SET\_DATA<sub>n</sub>)

The GPIO set data register (SET\_DATA<sub>n</sub>) controls driving high of the corresponding GPIO pin *j* in GPIO bank *k*, if the pin is configured as an output (DIR<sub>n</sub> = 0). Writes do not affect pins not configured as GPIO outputs. Writing a 1 to a specific bit in SET\_DATA<sub>n</sub> sets the corresponding GPIO pin *j* in GPIO bank *k*. A read of the BkPj bit returns the output drive state of the corresponding pin GPIOk[j]. The GPIO set data register (SET\_DATA01) is shown in Figure 14, SET\_DATA23 is shown in Figure 15, SET\_DATA45 is shown in Figure 16, SET\_DATA67 is shown in Figure 17, SET\_DATA8 is shown in Figure 18, and described in Table 7. See Table 1 to determine the SET\_DATA<sub>n</sub> bit associated with each GPIO bank and pin number.

**Figure 14. GPIO Banks 0 and 1 Set Data Register (SET\_DATA01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 15. GPIO Banks 2 and 3 Set Data Register (SET\_DATA23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 16. GPIO Banks 4 and 5 Set Data Register (SET\_DATA45)**

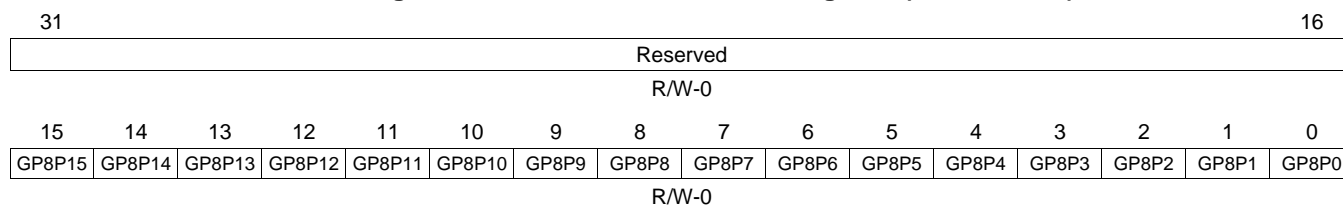
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 17. GPIO Banks 6 and 7 Set Data Register (SET\_DATA67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 18. GPIO Bank 8 Set Data Register (SET\_DATA8)**


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

**Table 7. GPIO Set Data Register (SET\_DATA<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Set the output drive state of GPk[j] to logic high. The GPkPj bit is used to drive the output high on pin j in GPIO bankk. The GPkPj bit is ignored when GPk[j] is configured as an input. Reading the GPkPj bit returns the output drive state of GPk[j].
		1	No effect.
			GPk[j] is set to output logic high.

### 3.6 GPIO Clear Data Registers (CLR\_DATA $n$ )

The GPIO clear data register (CLR\_DATA $n$ ) controls clearing low of the corresponding GPIO pin  $j$  in GPIO bank  $k$ , if the pin is configured as an output (DIR $n$  = 0). Writes do not affect pins not configured as GPIO outputs. Writing a 1 to a specific bit in CLR\_DATA $n$  resets the corresponding GPIO pin  $j$  in GPIO bank  $k$ . A read of the BkPj bit returns the output drive state of the corresponding pin GPIOk[j]. The GPIO clear data register (CLR\_DATA01) is shown in Figure 19, CLR\_DATA23 is shown in Figure 20, CLR\_DATA45 is shown in Figure 21, CLR\_DATA67 is shown in Figure 22, CLR\_DATA8 is shown in Figure 23, and described in Table 8. See Table 1 to determine the CLR\_DATA $n$  bit associated with each GPIO bank and pin number.

**Figure 19. GPIO Banks 0 and 1 Clear Data Register (CLR\_DATA01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 20. GPIO Banks 2 and 3 Clear Data Register (CLR\_DATA23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 21. GPIO Banks 4 and 5 Clear Data Register (CLR\_DATA45)**

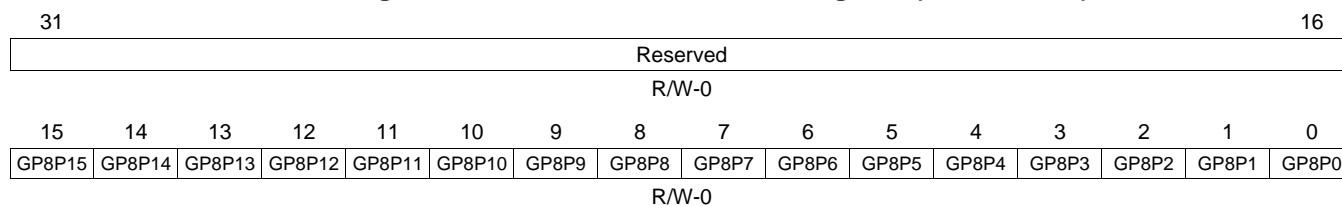
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 22. GPIO Banks 6 and 7 Clear Data Register (CLR\_DATA67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 23. GPIO Bank 8 Clear Data Register (CLR\_DATA8)**


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

**Table 8. GPIO Clear Data Register (CLR\_DATA<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkP <sub>j</sub>	0	Clear the output drive state of GPk[j] to logic low. The GPkP <sub>j</sub> bit is used to drive the output low on pin <i>j</i> in GPIO bank <i>k</i> . The GPkP <sub>j</sub> bit is ignored when GPk[j] is configured as an input. Reading the GPkP <sub>j</sub> bit returns the output drive state of GPk[j].  No effect.
		1	

### 3.7 GPIO Input Data Registers (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 24](#), IN\_DATA23 is shown in [Figure 25](#), IN\_DATA45 is shown in [Figure 26](#), IN\_DATA67 is shown in [Figure 27](#), IN\_DATA8 is shown in [Figure 28](#), and described in [Table 9](#). See [Table 1](#) to determine the IN\_DATA $n$  bit associated with each GPIO bank and pin number.

**Figure 24. GPIO Banks 0 and 1 Input Data Register (IN\_DATA01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R-0															

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

**Figure 25. GPIO Banks 2 and 3 Input Data Register (IN\_DATA23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R-0															

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

**Figure 26. GPIO Banks 4 and 5 Input Data Register (IN\_DATA45)**

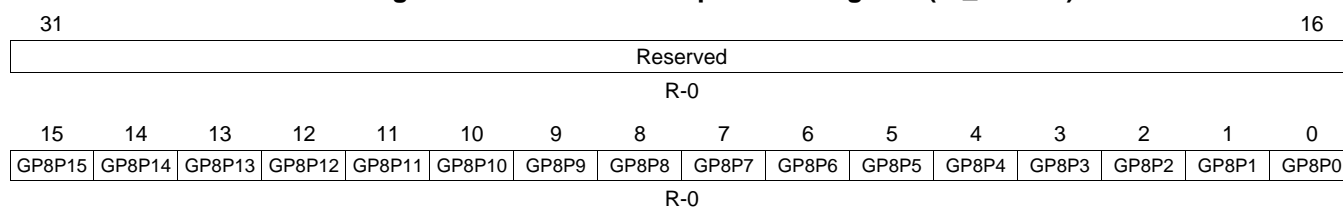
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R-0															

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

**Figure 27. GPIO Banks 6 and 7 Input Data Register (IN\_DATA67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R-0															

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

**Figure 28. GPIO Bank 8 Input Data Register (IN\_DATA8)**


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

**Table 9. GPIO Input Data Register (IN\_DATA<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0 1	Status of pin GPk[j]. Reading the GPkPj bit returns the state of pin j in GPIO bank k. GPk[j] is logic low. GPk[j] is logic high.

### 3.8 GPIO Set Rising Edge Interrupt Registers (SET\_RIS\_TRIGn)

The GPIO set rising edge trigger interrupt register (SET\_RIS\_TRIGn) enables a rising edge trigger on the GPIO pin to generate a GPIO interrupt. The GPIO set rising edge interrupt register (SET\_RIS\_TRIG01) is shown in Figure 29, SET\_RIS\_TRIG23 is shown in Figure 30, SET\_RIS\_TRIG45 is shown in Figure 31, SET\_RIS\_TRIG67 is shown in Figure 32, SET\_RIS\_TRIG8 is shown in Figure 33, and described in Table 10. See Table 1 to determine the SET\_RIS\_TRIGn bit associated with each GPIO bank and pin number.

**Figure 29. GPIO Banks 0 and 1 Set Rise Trigger Register (SET\_RIS\_TRIG01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 30. GPIO Banks 2 and 3 Set Rise Trigger Register (SET\_RIS\_TRIG23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 31. GPIO Banks 4 and 5 Set Rise Trigger Register (SET\_RIS\_TRIG45)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 32. GPIO Banks 6 and 7 Set Rise Trigger Register (SET\_RIS\_TRIG67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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





### 3.9 GPIO Clear Rising Edge Interrupt Registers (CLR\_RIS\_TRIGn)

The GPIO clear rising edge trigger interrupt register (CLR\_RIS\_TRIGn) disables the rising edge trigger on the GPIO pin to generate a GPIO interrupt. The GPIO clear rising edge interrupt register (CLR\_RIS\_TRIG01) is shown in Figure 34, CLR\_RIS\_TRIG23 is shown in Figure 35, CLR\_RIS\_TRIG45 is shown in Figure 36, CLR\_RIS\_TRIG67 is shown in Figure 37, CLR\_RIS\_TRIG8 is shown in Figure 38, and described in Table 11. See Table 1 to determine the CLR\_RIS\_TRIGn bit associated with each GPIO bank and pin number.

**Figure 34. GPIO Banks 0 and 1 Clear Rise Trigger Register (CLR\_RIS\_TRIG01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 35. GPIO Banks 2 and 3 Clear Rise Trigger Register (CLR\_RIS\_TRIG23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 36. GPIO Banks 4 and 5 Clear Rise Trigger Register (CLR\_RIS\_TRIG45)**

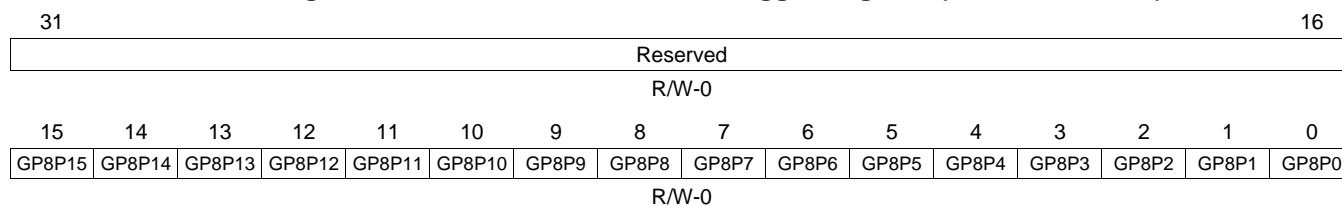
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 37. GPIO Banks 6 and 7 Clear Rise Trigger Register (CLR\_RIS\_TRIG67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 38. GPIO Bank 8 Clear Rise Trigger Register (CLR\_RIS\_TRIG8)**


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

**Table 11. GPIO Clear Rising Edge Interrupt Register (CLR\_RIS\_TRIG<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Disable rising edge interrupt detection on GPk[j]. Reading the GPkPj bit in either SET_RIS_TRIG <sub>n</sub> or CLR_RIS_TRIG <sub>n</sub> always returns an indication of whether the rising edge interrupt generation function is enabled for GPk[j]. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled.
		0	No effect.
		1	No interrupt is caused by a low-to-high transition on GPk[j].

### 3.10 GPIO Set Falling Edge Interrupt Registers (SET\_FAL\_TRIGn)

The GPIO set falling edge trigger interrupt register (SET\_FAL\_TRIGn) enables a falling edge trigger on the GPIO pin to generate a GPIO interrupt. The GPIO set falling edge interrupt register (SET\_FAL\_TRIG01) is shown in Figure 39, SET\_FAL\_TRIG23 is shown in Figure 40, SET\_FAL\_TRIG45 is shown in Figure 41, SET\_FAL\_TRIG67 is shown in Figure 42, SET\_FAL\_TRIG8 is shown in Figure 43, and described in Table 12. See Table 1 to determine the SET\_FAL\_TRIGn bit associated with each GPIO bank and pin number.

**Figure 39. GPIO Banks 0 and 1 Set Rise Trigger Register (SET\_FAL\_TRIG01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 40. GPIO Banks 2 and 3 Set Rise Trigger Register (SET\_FAL\_TRIG23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 41. GPIO Banks 4 and 5 Set Rise Trigger Register (SET\_FAL\_TRIG45)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 42. GPIO Banks 6 and 7 Set Rise Trigger Register (SET\_FAL\_TRIG67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 43. GPIO Bank 8 Set Rise Trigger Register (SET\_FAL\_TRIG8)**

31	Reserved														16
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP8P15	GP8P14	GP8P13	GP8P12	GP8P11	GP8P10	GP8P9	GP8P8	GP8P7	GP8P6	GP8P5	GP8P4	GP8P3	GP8P2	GP8P1	GP8P0
R/W-0															

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

**Table 12. GPIO Set Falling Edge Trigger Interrupt Register (SET\_FAL\_TRIG<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj		Enable falling edge trigger interrupt detection on GPk[j]. Reading the GPkPj bit in either SET_FAL_TRIG <sub>n</sub> or CLR_FAL_TRIG <sub>n</sub> always returns an indication of whether the falling edge interrupt generation function is enabled for pin GPk[j]. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled.
		0	No effect.
		1	Interrupt is caused by a high-to-low transition on GPk[j].

### 3.11 GPIO Clear Falling Edge Interrupt Registers (CLR\_FAL\_TRIGn)

The GPIO clear falling edge trigger interrupt register (CLR\_FAL\_TRIGn) disables the falling edge trigger on the GPIO pin to generate a GPIO interrupt. The GPIO clear falling edge interrupt register (CLR\_FAL\_TRIG01) is shown in Figure 44, CLR\_FAL\_TRIG23 is shown in Figure 45, CLR\_FAL\_TRIG45 is shown in Figure 46, CLR\_FAL\_TRIG67 is shown in Figure 47, CLR\_FAL\_TRIG8 is shown in Figure 48, and described in Table 13. See Table 1 to determine the CLR\_FAL\_TRIGn bit associated with each GPIO bank and pin number.

**Figure 44. GPIO Banks 0 and 1 Clear Rise Trigger Register (CLR\_FAL\_TRIG01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W-0															

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

**Figure 45. GPIO Banks 2 and 3 Clear Rise Trigger Register (CLR\_FAL\_TRIG23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W-0															

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

**Figure 46. GPIO Banks 4 and 5 Clear Rise Trigger Register (CLR\_FAL\_TRIG45)**

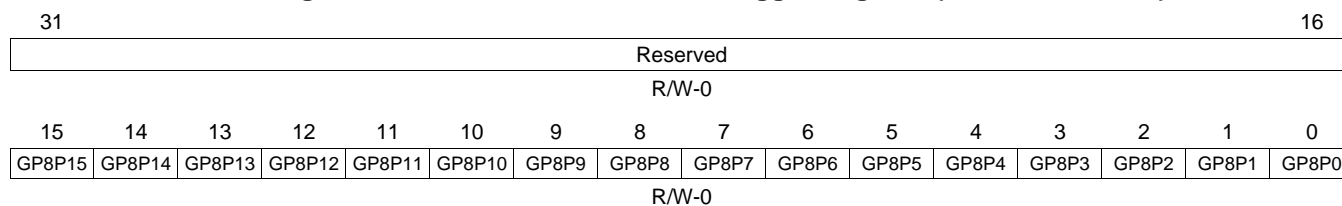
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W-0															

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

**Figure 47. GPIO Banks 6 and 7 Clear Rise Trigger Register (CLR\_FAL\_TRIG67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W-0															

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

**Figure 48. GPIO Bank 8 Clear Rise Trigger Register (CLR\_FAL\_TRIG8)**


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

**Table 13. GPIO Clear Falling Edge Interrupt Register (CLR\_FAL\_TRIG<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Disable falling edge interrupt detection on GPk[j]. Reading the GPkPj bit in either SET_FAL_TRIG <sub>n</sub> or CLR_FAL_TRIG <sub>n</sub> always returns an indication of whether the falling edge interrupt generation function is enabled for GPk[j]. Therefore, this bit will be one in both registers if the function is enabled, and zero in both registers if the function is disabled.
		0	No effect.
		1	No interrupt is caused by a high-to-low transition on GPk[j].

### 3.12 GPIO Interrupt Status Registers (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 49, INTSTAT23 is shown in Figure 50, INTSTAT45 is shown in Figure 51, INTSTAT67 is shown in Figure 52, INTSTAT8 is shown in Figure 53, and described in Table 14. See Table 1 to determine the INTSTAT $n$  bit associated with each GPIO bank and pin number.

**Figure 49. GPIO Banks 0 and 1 Interrupt Status Register (INTSTAT01)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP1P15	GP1P14	GP1P13	GP1P12	GP1P11	GP1P10	GP1P9	GP1P8	GP1P7	GP1P6	GP1P5	GP1P4	GP1P3	GP1P2	GP1P1	GP1P0
R/W1C-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP0P15	GP0P14	GP0P13	GP0P12	GP0P11	GP0P10	GP0P9	GP0P8	GP0P7	GP0P6	GP0P5	GP0P4	GP0P3	GP0P2	GP0P1	GP0P0
R/W1C-0															

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

**Figure 50. GPIO Banks 2 and 3 Interrupt Status Register (INTSTAT23)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3P15	GP3P14	GP3P13	GP3P12	GP3P11	GP3P10	GP3P9	GP3P8	GP3P7	GP3P6	GP3P5	GP3P4	GP3P3	GP3P2	GP3P1	GP3P0
R/W1C-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2P15	GP2P14	GP2P13	GP2P12	GP2P11	GP2P10	GP2P9	GP2P8	GP2P7	GP2P6	GP2P5	GP2P4	GP2P3	GP2P2	GP2P1	GP2P0
R/W1C-0															

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

**Figure 51. GPIO Banks 4 and 5 Interrupt Status Register (INTSTAT45)**

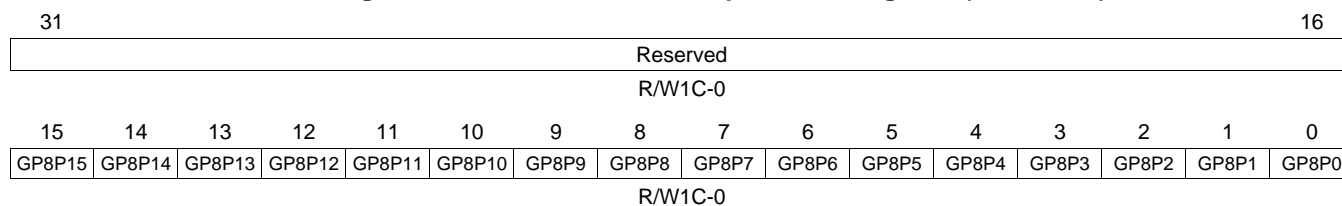
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP5P15	GP5P14	GP5P13	GP5P12	GP5P11	GP5P10	GP5P9	GP5P8	GP5P7	GP5P6	GP5P5	GP5P4	GP5P3	GP5P2	GP5P1	GP5P0
R/W1C-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP4P15	GP4P14	GP4P13	GP4P12	GP4P11	GP4P10	GP4P9	GP4P8	GP4P7	GP4P6	GP4P5	GP4P4	GP4P3	GP4P2	GP4P1	GP4P0
R/W1C-0															

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

**Figure 52. GPIO Banks 6 and 7 Interrupt Status Register (INTSTAT67)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP7P15	GP7P14	GP7P13	GP7P12	GP7P11	GP7P10	GP7P9	GP7P8	GP7P7	GP7P6	GP7P5	GP7P4	GP7P3	GP7P2	GP7P1	GP7P0
R/W1C-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP6P15	GP6P14	GP6P13	GP6P12	GP6P11	GP6P10	GP6P9	GP6P8	GP6P7	GP6P6	GP6P5	GP6P4	GP6P3	GP6P2	GP6P1	GP6P0
R/W1C-0															

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

**Figure 53. GPIO Bank 8 Interrupt Status Register (INTSTAT8)**


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

**Table 14. GPIO Interrupt Status Register (INTSTAT<sub>n</sub>) Field Descriptions**

Bit	Field	Value	Description
31-0	GPkPj	0	Interrupt status of GPk[j]. The GPkPj bit is used to monitor pending GPIO interrupts on pin j of GPIO bank k. Write a 1 to the GPkPj bit to clear the status bit; a write of 0 has no effect.
		1	No pending interrupt on GPk[j].
			Pending interrupt on GPk[j].



## Appendix A Revision History

Table 15 lists the changes made since the previous version of this document.

**Table 15. Document Revision History**

Reference	Additions/Modifications/Deletions
<a href="#">Section 2.5</a>	Changed third paragraph.
<a href="#">Table 1</a>	Added GPIO Pin Numbers 129-144.
<a href="#">Section 3</a>	Changed paragraph.
<a href="#">Table 2</a>	Added GPIO Bank 8 registers.
<a href="#">Figure 3</a>	Changed bit 8.
<a href="#">Table 4</a>	Added EN8 bit.
<a href="#">Figure 8</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 13</a>	Added figure. Subsequent figures renumbered.
<a href="#">Section 3.5</a>	Changed third sentence in paragraph.
<a href="#">Figure 18</a>	Added figure. Subsequent figures renumbered.
<a href="#">Section 3.6</a>	Changed third sentence in paragraph.
<a href="#">Figure 23</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 28</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 33</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 38</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 43</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 48</a>	Added figure. Subsequent figures renumbered.
<a href="#">Figure 53</a>	Added figure.

## 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:

<b>Products</b>		<b>Applications</b>	
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>	Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>	Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>	Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>	Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>	Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>	Energy	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>	Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>	Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>	Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>	Space, Avionics & Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>	Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
		Wireless	<a href="http://www.ti.com/wireless-apps">www.ti.com/wireless-apps</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2010, Texas Instruments Incorporated