TMS320C6000 DSP General-Purpose Input/Output (GPIO) Reference Guide

Literature Number: SPRU584A
March 2004

Texas Instruments

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 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.

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.

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Preface

Read This First

About This Manual

This document describes the general-purpose input/output (GPIO) peripheral in the digital signal processors (DSPs) of the TMS320C6000™ DSP family.

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.
- When referencing specific register bits, the X in the register bit name is replaced with the bit number; for example, GPXDIR refers to the bit field of the GPIO direction register and GP15DIR refers to bit 15 of GPDIR.
- 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 C6000™ devices and related support tools. 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.

TMS320C6000 CPU and Instruction Set Reference Guide (literature number SPRU189) describes the TMS320C6000™ CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.

TMS320C6000 DSP Peripherals Overview Reference Guide (literature number SPRU190) describes the peripherals available on the TMS320C6000™ DSPs.

TMS320C6000 Technical Brief (literature number SPRU197) gives an introduction to the TMS320C62x™ and TMS320C67x™ DSPs, development tools, and third-party support.

TMS320C64x Technical Overview (SPRU395) gives an introduction to the TMS320C64x™ DSP and discusses the application areas that are enhanced by the TMS320C64x VelociTI™.

TMS320C6000 Programmer's Guide (literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

TMS320C6000 Code Composer Studio Tutorial (literature number SPRU301) introduces the Code Composer Studio™ integrated development environment and software tools.

Code Composer Studio Application Programming Interface Reference Guide (literature number SPRU321) describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

TMS320C6x Peripheral Support Library Programmer's Reference (literature number SPRU273) describes the contents of the TMS320C6000™ peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

TMS320C6000 Chip Support Library API Reference Guide (literature number SPRU401) describes a set of application programming interfaces (APIs) used to configure and control the on-chip peripherals.

Trademarks

Code Composer Studio, C6000, C62x, C64x, C67x, TMS320C6000, TMS320C62x, TMS320C64x, TMS320C67x, and VelociTI are trademarks of Texas Instruments.

Contents

Figures

Tables

This page is intentionally left blank.

General-Purpose Input/Output (GPIO)

This document describes the general-purpose input/output (GPIO) peripheral in the digital signal processors (DSPs) of the TMS320C6000™ DSP family.

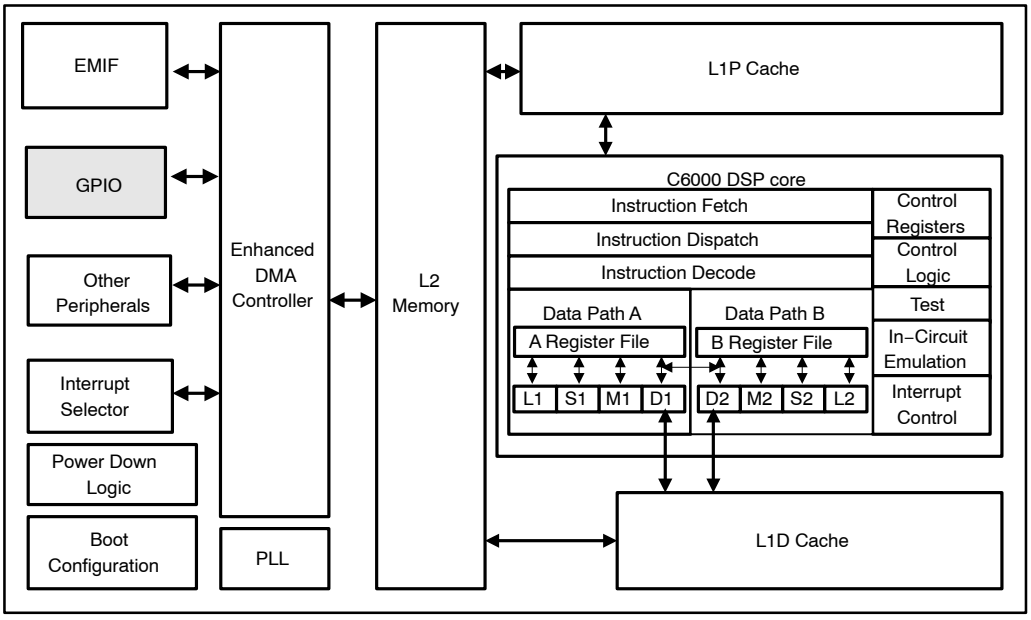
1 Overview

The general-purpose input/output (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.

In addition, the GPIO peripheral can produce CPU interrupts and EDMA events in different interrupt/event generation modes.

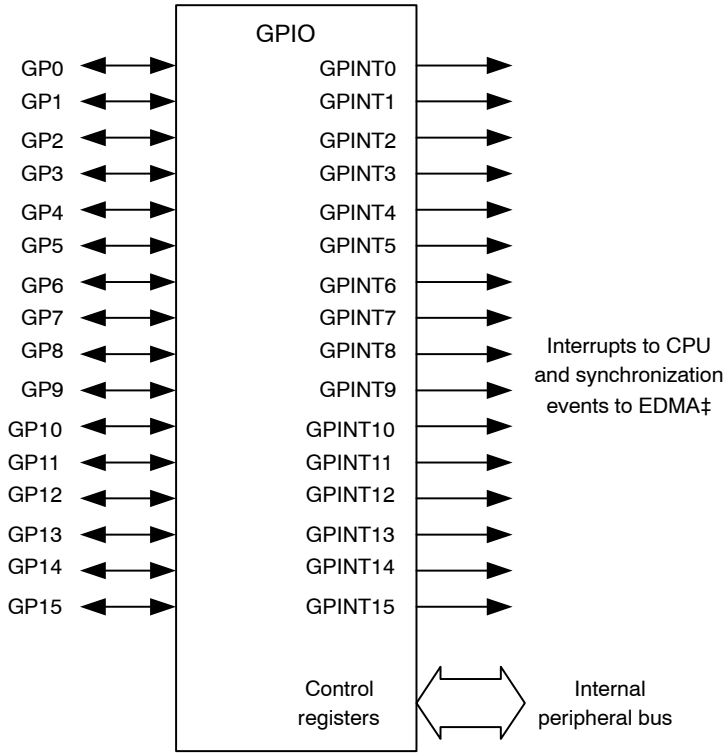
Figure 1 shows the GPIO peripheral in the TMS320C64x™ DSP. Figure 2 shows the GPIO peripheral block diagram.

Figure 1. TMS320C64x DSP Block Diagram



Some GPIO pins are MUXed with other device pins. On a given device, all bits may not be implemented for each GPIO register. Refer to the device-specific datasheet for details on specific MUXing and for the availability of the register bits. GPINT[0−15] are all synchronization events to the EDMA. However, only GPINT0 and GPINT[4−7] are available as interrupt sources to the CPU.

Figure 2. GPIO Peripheral Block Diagram



† Some of the GP*n* pins are MUXed with other device signals. The number of available GP*n* pins is device specific. Refer to the device-specific datasheet for details.

‡ All GPINT*n* are synchronization events to the EDMA. Only GPINT0 and GPINT[4−7] are available as interrupts to the CPU.

2 GPIO Function

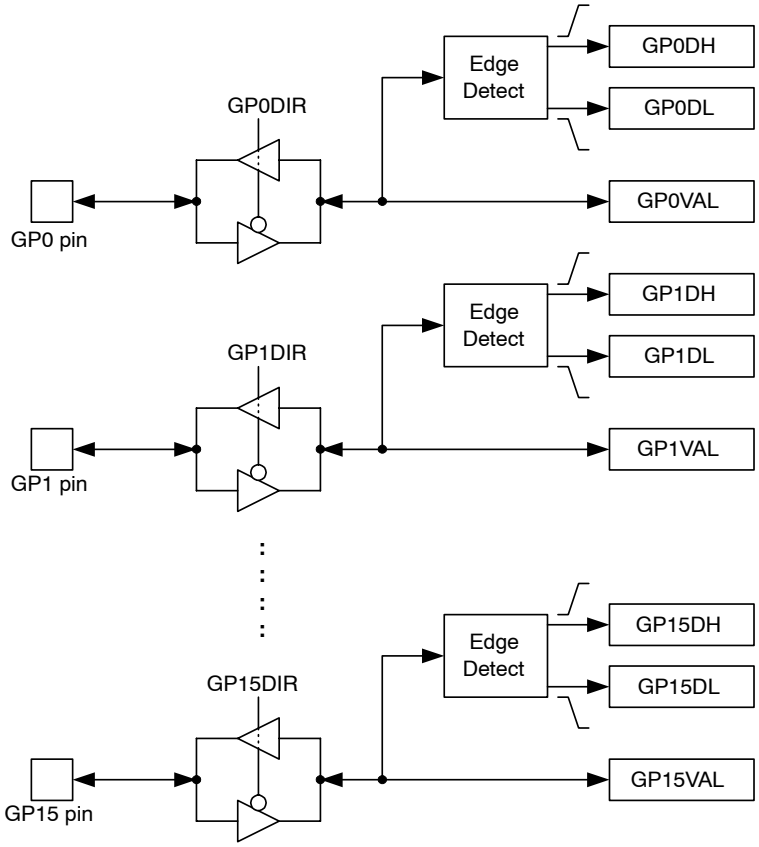
A GPIO pin can operate as a general-purpose input/output once it is enabled in the GPIO enable register (GPEN). You can independently configure each GPIO pin as either an input or an output using the GPIO direction register (GPDIR). When configured as an output (GPXDIR bit = 1), the value in the GPXVAL bit in the GPIO value register (GPVAL) is driven on the corresponding GP*n* pin. When configured as an input (GPXDIR bit = 0), the state of the input can be read from the corresponding GPXVAL bit. Refer to section 5 for details on the GPIO registers.

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GPXDIR bit = 0). The GPIO signal transition is reflected in the GPIO delta registers (GPDH or GPDL). The GPXDH bit in GPDH bit is set to 1 when the corresponding enabled input undergoes a low-to-high transition; the GPXDL bit in GPDL is set to 1 when the corresponding enabled input undergoes a high-to-low transition.

Figure 3 shows the general-purpose input/output and edge-detect logic of the GPIO peripheral.

To configure GP0 as a general-purpose output, the GP0M bit in the GPIO global control register (GPGC) must be cleared to 0, in addition to setting the GPDIR bit 0 in GPDIR to 1. See section 3.3 for details on GP0 configurations.

Figure 3. GPIO Functional Block Diagram



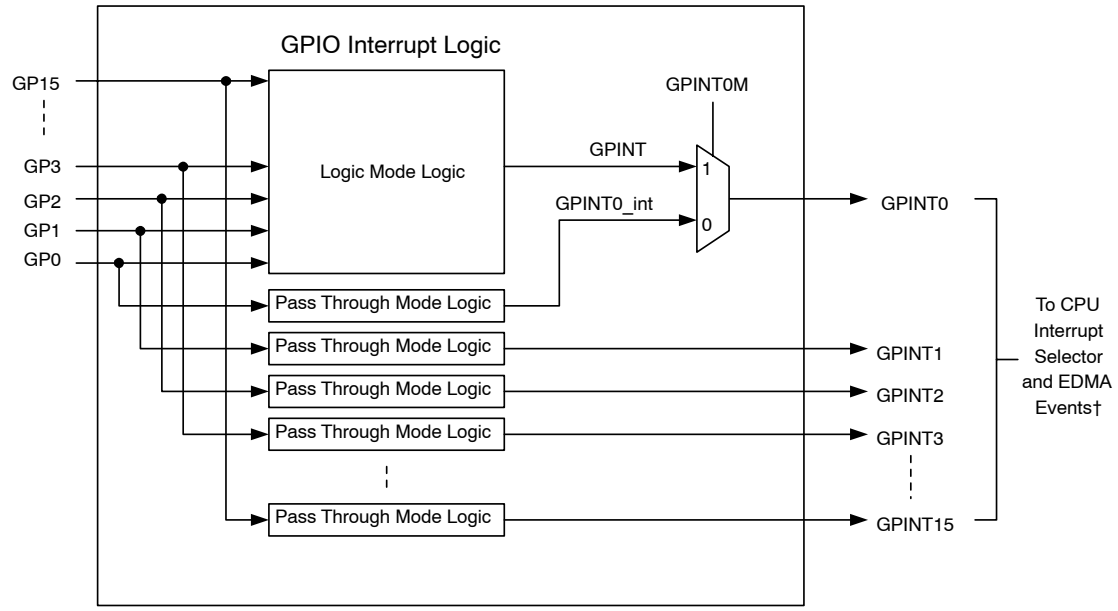
3 Interrupt and Event Generation

The GPIO peripheral generates interrupts to the CPU and synchronization events to the EDMA in two modes:

- ❑ Pass-through mode
- ❑ Logic mode

The pass-through mode allows each GP*n* signal configured as an input to directly trigger a CPU interrupt and an EDMA event. The logic mode allows you to determine which GPIO signals are used as inputs to a semi-programmable logic function. The output of this logic function, GPINT, is MUXed with the pass-through mode internal output GPINT0_int to generate a CPU interrupt and an EDMA event, GPINT0. In addition, the logic mode output GPINT is driven out of the GP0 pin for use at the board level (section 3.3). Figure 4 shows the GPIO interrupt and event generation logic.

Figure 4. GPIO Interrupt and Event Generation Block Diagram



† All GPINT*n* are synchronization events to the EDMA. Only GPINT0 and GPINT[4–7] are available as interrupts to the CPU.

3.1 Pass-Through Mode

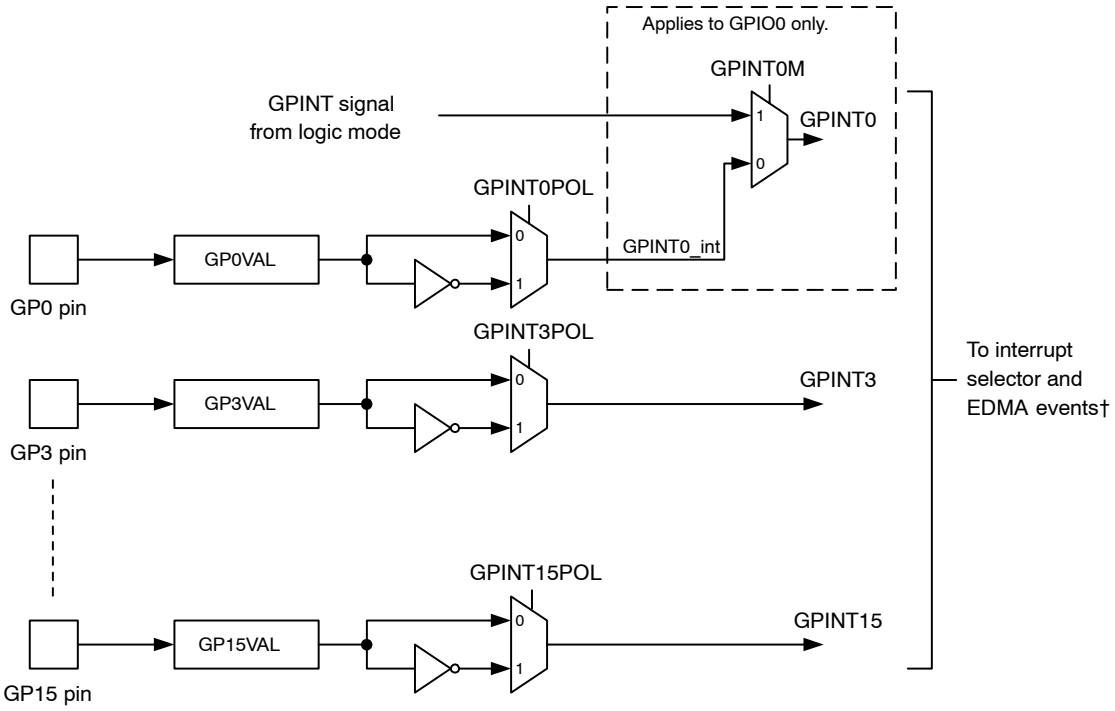
The pass-through mode applies to all GPIO signals. In pass-through mode, a transition on the GP*n* input pin generates an interrupt event to the CPU and a synchronization event to the EDMA. Note that although all GPINT*n* are synchronization events to the EDMA, only GPINT0 and GPINT[4−7] are available as interrupts to the CPU. Figure 5 shows the pass-through mode interrupt/event generation block diagram. You must configure the following bits correctly to use a GP*n* pin in the pass-through mode:

- GPXEN bit *n* = 1: enable GP*n* to function as a GPIO pin.
- GPXDIR bit *n* = 0: GP*n* pin is an input.
- Set GPINTXPOL bit *n* = 0, if an interrupt/event is desired upon a rising-edge transition on the corresponding GP*n* pin; set GPINTXPOL bit *n* = 1, if an interrupt/event is desired upon a falling-edge transition on the corresponding GP*n* pin.

As shown in Figure 5, to use the GP0 in pass-through mode, the GPINT0M bit in GPGC must be cleared to 0. The GPINT0_int output from the pass-through mode logic is MUXed with the GPINT output from the logic mode logic to generate the GPINT0 interrupt/event. This is shown in Figure 4 and Figure 5. Refer to section 5.8 and section 3.3 for details.

If a GP*n* is configured as an output, the corresponding GPINT*n* signal is disabled.

*Figure 5. GPINT*n *Generation in Pass-Through Mode*



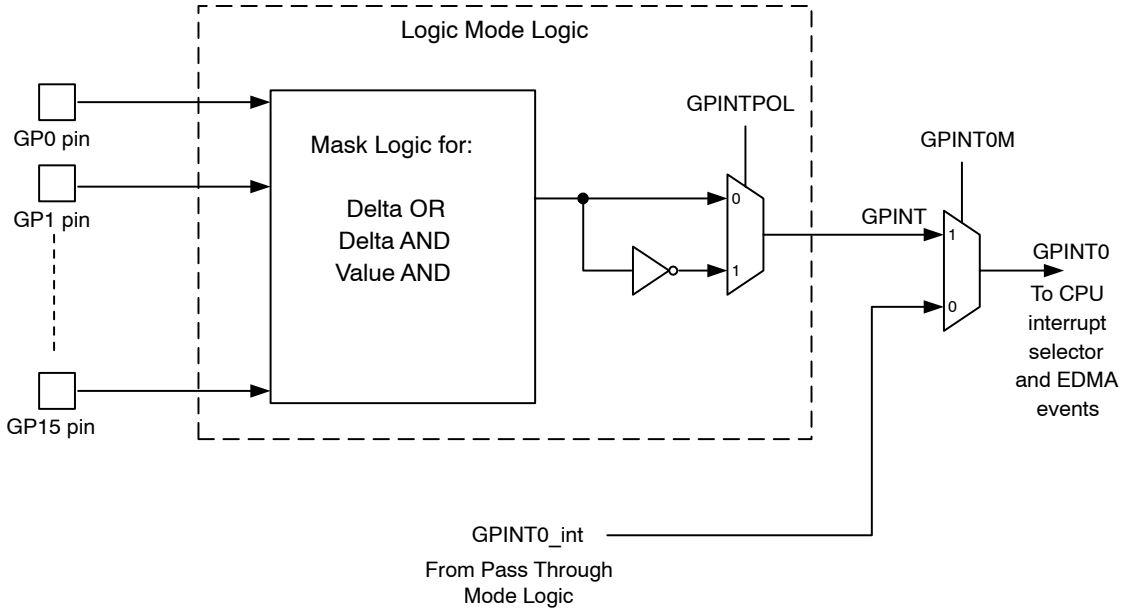
† All GPINT*n* are synchronization events to the EDMA. Only GPINT0 and GPINT[4−7] are available as interrupts to the CPU.

3.2 Logic Mode

In the logic mode, an interrupt/event is generated based on a logic combination of the GPIO inputs. The output of this logic function, GPINT, can be generated upon detection of a specific edge (rising, falling, or both) on any GPIO input signal(s), or upon detection of specific value(s) on any GPIO input signal(s). Disabled GPIO signals or enabled GPIO outputs cannot be used for interrupt/event generation. The logic mode output GPINT is MUXed with the pass-through mode output GPINT0_int to generate a CPU interrupt and an EDMA event. In order to use the logic mode to generate an interrupt/event, GPINT0M in GPGC must be set to 1. The GPINT signal can also be driven out of the GP0 pin for use at the board level. See section 3.3.

Figure 6 shows a block diagram of the logic mode logic. By default, GPINT is asserted (high) when the logic combination of the input(s) is evaluated true. By setting GPINTPOL = 1 in GPGC, GPINT is asserted (high) when the logic combination of the input(s) is evaluated false. This negative function of the GPINT is useful in indicating signal deassertions at the GPIO pins.

Figure 6. GPINT Generation in Logic Mode



The GPINT generation operates in one of three modes: Delta OR, Delta AND, or Value AND. The GPINT generation is configured via two control bits in GPGC, GPINTDV and LOGIC, in addition to the mask bits in GPHM and GPLM. The GPINTDV bit in GPGC divides the logic mode into either Delta or Value mode as follows:

- Delta Mode: Inputs to the interrupt/event mask logic are sourced from GPDH and GPDL. GPINT is caused by the logic combination of the *transition* on the GPIO pin(s).
- Value Mode: Inputs to the interrupt/event mask logic are sourced from GPVAL. GPINT is caused by the logic combination of the *value* on the GPIO pin(s).

The source to the logic mode mask logic is gated by GPHM and GPLM. In delta mode, the GPDH bit is gated with the GPHM bit and the GPDL bit is gated with the GPLM bit. In value mode, the value from the pin is gated with the GPHM bit and the inverted value from the pin is gated with the GPLM bit.

The LOGIC bit in GPGC controls whether an interrupt/event is generated based on all the mask outputs being true or any one of the mask outputs being true:

- OR Mode: Interrupt/event generated based on *any* one of the mask outputs being true.
- AND Mode: Interrupt/event generated based on *all* of the mask outputs being true.

Table 1 summarizes the three modes in logic mode and the setup of the GPINTDV and LOGIC bits in GPGC.

Table 1. Logic Mode Truth Table

GPGC Bit			
GPINTDV	**LOGIC**	**Logic Mode**	**Section**
0	0	Delta OR	3.2.1
0	1	Delta AND	3.2.2
1	0	Reserved	—
1	1	Value AND	3.2.3

In summary, GPGC must be configured as follows in logic mode:

- GPINT0M = 1 to enable logic mode interrupt/event generation. The interrupt/event signal to the DSP (GPINT0) is based on the logic function output GPINT.
- GPINTPOL = 0 if the interrupt/event is based upon the logic evaluating true; or
  GPINTPOL = 1 if the interrupt/event is based upon the logic evaluating false.
- GPINTDV = 0 for Delta Mode, or
  GPINTDV = 1 for Value Mode.
- LOGIC = 0 for OR Mode, or
  LOGIC = 1 for AND Mode.

3.2.1 Delta OR Mode (GPINTDV = 0, LOGIC = 0)

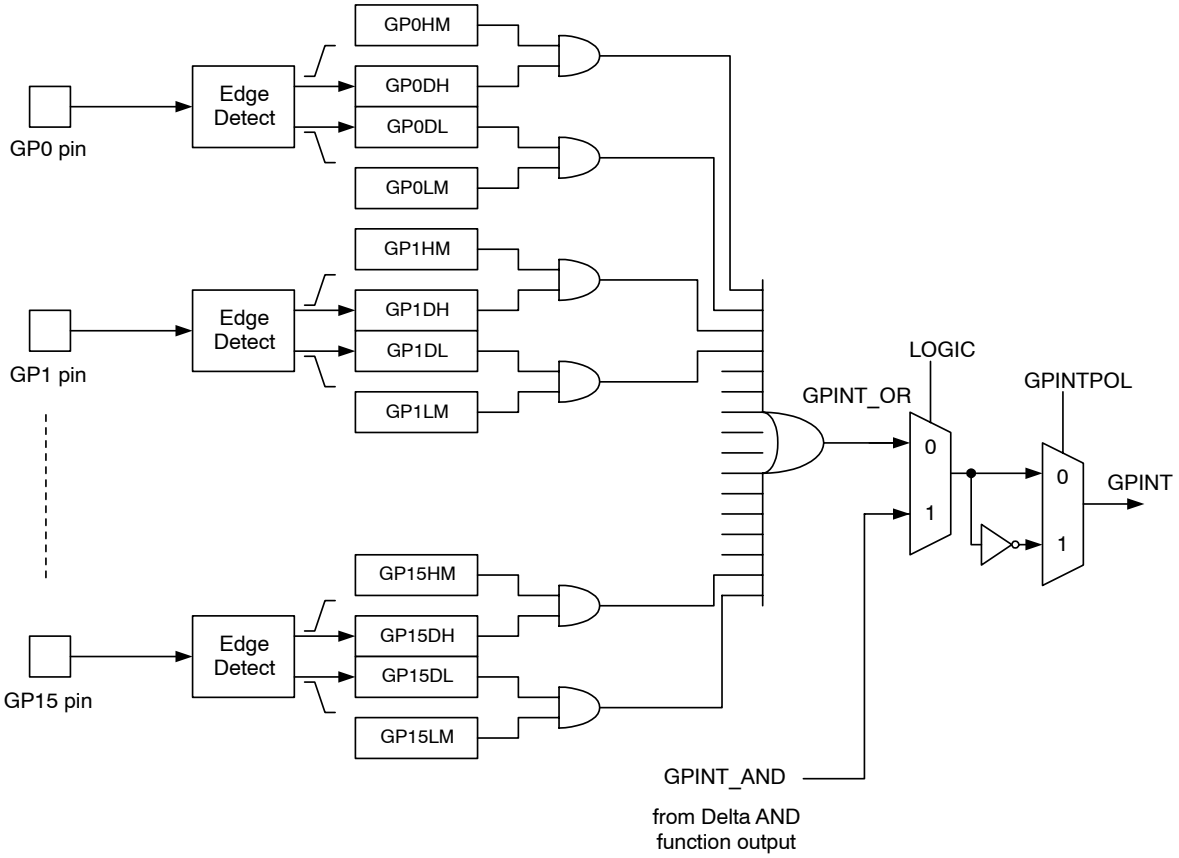
Delta OR mode allows the generation of an interrupt/event upon the first transition among a set of enabled GPIO inputs. The logic function output GPINT is driven active when any of the GPDH or GPDL bits and the corresponding GPHM or GPLM bits are set. Since the GPDH and GPDL bits operate independently from one another and have separate masks (GPHM and GPLM), an interrupt can be generated based on a signal transitioning to a specific state (high or low) or transitioning at all (either state).

Figure 7 shows the block diagram of the Delta OR mode.

To generate an interrupt/event in Delta OR mode, the GPINTDV and LOGIC bits in GPGC must be configured as follows:

- GPINTDV = 0; Delta Mode
- LOGIC = 0; OR Mode

Figure 7. GPINT Generation in Delta OR Mode



In addition, GPHM and GPLM must be configured properly to enable the corresponding GPDH and GPDL bits to be inputs to the logic function. The following examples show the GPHM and GPLM setup. All the given GPIO pins in these examples are enabled as inputs (GPXEN bit = 1, GPXDIR bit = 0).

Example 1: GPINT based on a low-to-high transition on GP1

- ❏ GPHM setup: GP1HM = 1
- ❏ GPLM setup: don't care
- ❏ GPINT generation is caused by GP1DH = 1
  - ■ If GP1 is high when entering this mode, a high-to-low transition (GP1DL = 1) followed by a low-to-high transition (GP1DH = 1) on GP1 generates GPINT.
  - ■ If GP1 is low when entering this mode, a low-to-high transition (GP1DH = 1) on GP1 generates GPINT.

Example 2: GPINT based on any transition on GP1

- ❏ GPHM setup: GP1HM = 1
- ❏ GPLM setup: GP1LM = 1
- ❏ GPINT generation is caused by GP1DH = 1 or GP1DL = 1
  - ■ Regardless of the initial state of GP1, the first transition on GP1 generates GPINT. This first transition can either be a low-to-high transition (GP1DH = 1) or a high-to-low transition (GP1DL = 1).

Example 3: GPINT based on a low-to-high transition on GP1 or GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = 1
- ❏ GPLM setup: don't care
- ❏ GPINT generation is caused by GP1DH = 1 or GP2DH = 1
  - ■ The first low-to-high transition (GPXDH = 1) on either GP1 or GP2 generates GPINT.

Example 4: GPINT based on a low-to-high transition on GP1 or a high-to-low transition on GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = don't care
- ❏ GPLM setup: GP1LM = don't care, GP2LM = 1
- ❏ GPINT generation is caused by GP1DH = 1 or GP2DL = 1
  - ■ The first low-to-high transition on GP1 (GP1DH = 1) or high-to-low transition on GP2 (GP2DL = 1) generates GPINT.

Example 5: GPINT based on any transition on GP1 or GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = 1
- ❏ GPLM setup: GP1LM = 1, GP2LM = 1
- ❏ GPINT generation is caused by GP1DH, GP1DL, GP2DH, or GP2DL = 1
  - ■ The first transition on GP1 (GP1DH or GP1DL = 1) or the first transition on GP2 (GP2DH or GP2DL = 1) generates GPINT.

3.2.2 Delta AND Mode (GPINTDV = 0, LOGIC = 1)

Delta AND mode allows the generation of an interrupt/event after all of a set of specified signals have undergone some specified transitions. GPINT is driven active when both of the following conditions are true:

- ❏ All of the GPDH bits are asserted for the group of GPIO signals with the GPHM bits set.
- ❏ All of the GPDL bits are asserted for the group of GPIO signals with the GPLM bits set.

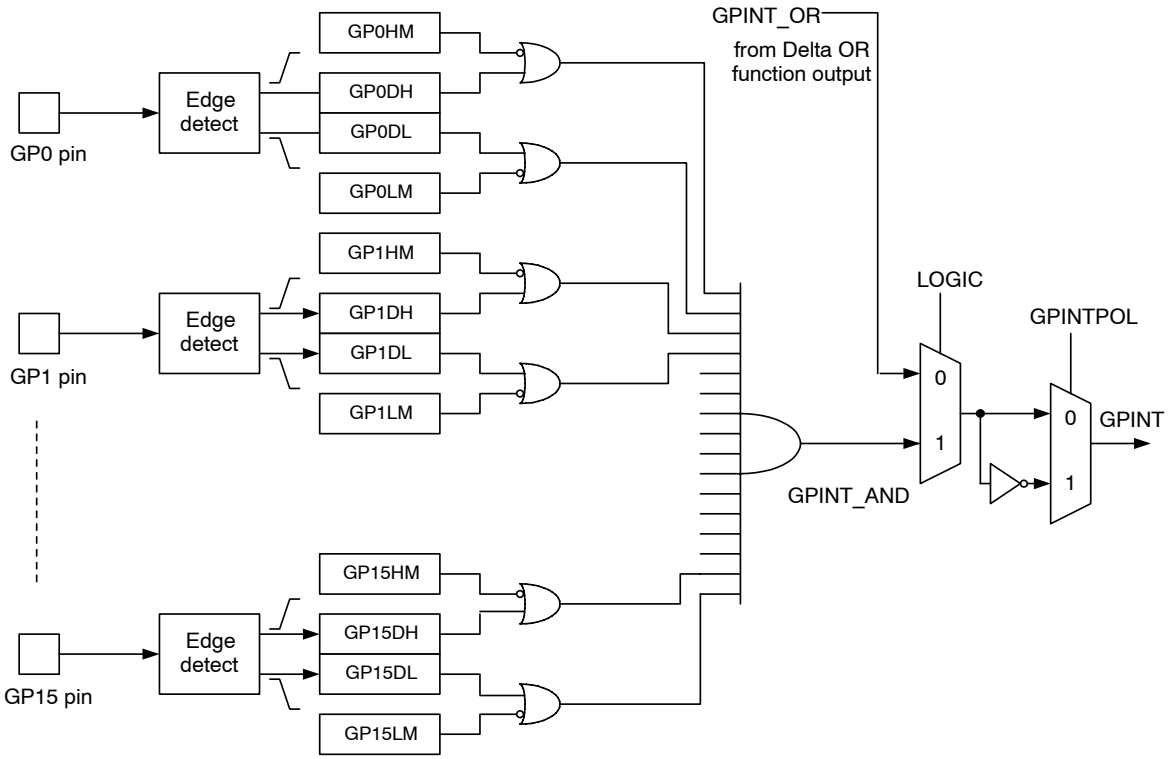
Since the GPDH and GPDL bits operate independently from one another and have separate masks (GPHM and GPLM), GPINT can be generated based on a signal transitioning from one state to another and back to the original state.

To generate an interrupt/event in Delta AND mode, the GPINTDV and LOGIC bits in GPGC must be configured as follows:

- ❏ GPINTDV = 0; Delta Mode
- ❏ LOGIC = 1; AND Mode

Figure 8 shows the functional block diagram of the Delta AND mode.

Figure 8. GPINT Generation in Delta AND Mode



Note: The functional block diagram shows the mask logic as an OR with an inverter on the mask bit. This forces the OR to evaluate true when the mask bit is disabled. This is strictly a functional block diagram. The actual implementation prevents the GPINT from being asserted in the case that all of the mask bits are disabled.

In addition, GPHM and GPLM must be configured properly to enable the corresponding GPDH and GPDL bits to be inputs to the logic function. The following examples show the GPHM and GPLM setup. All the given GPIO pins in these examples are enabled as inputs (GPEN bit = 1, GPDIR bit = 0).

Example 1: GPINT based on a low-to-high transition on GP1

- ❏ GPHM setup: GP1HM = 1
- ❏ GPLM setup: don't care
- ❏ GPINT generation is caused by GP1DH = 1
  - ■ If GP1 is high when entering this mode, a high-to-low transition (GP1DL = 1) followed by a low-to-high transition (GP1DH = 1) on GP1 generates GPINT.
  - ■ If GP1 is low when entering this mode, a low-to-high transition (GP1DH = 1) on GP1 generates GPINT.

Example 2: GPINT based on a low-to-high and a high-to-low transition on GP1

- ❏ GPHM setup: GP1HM = 1
- ❏ GPLM setup: GP1LM = 1
- ❏ GPINT generation is caused by GP1DH = 1 and GP1DL = 1
  - ■ If GP1 is high when entering this mode, a high-to-low transition (GP1DL = 1) followed by a low-to-high transition (GP1DH = 1) on GP1 generates GPINT.
  - ■ If GP1 is low when entering this mode, a low-to-high transition (GP1DH = 1) followed by a high-to-low transition (GP1DL = 1) on GP1 generates GPINT.

Example 3: GPINT based on a low-to-high transition on both GP1 and GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = 1
- ❏ GPLM setup: don't care
- ❏ GPINT generation is caused by GP1DH = 1 and GP2DH = 1
  - ■ Both GP1 and GP2 must undergo a low-to-high transition (GP1DH = 1 and GP2DH = 1) to generate GPINT. If either (or both) signal starts out high, GPINT is not generated until this signal undergoes a high-to-low followed by a low-to-high transition.

Example 4: GPINT based on a low-to-high transition on GP1 and a high-to-low transition on GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = don't care
- ❏ GPLM setup: GP1LM = don't care, GP2LM = 1
- ❏ GPINT generation is caused by GP1DH = 1 and GP2DL = 1
  - ■ Regardless of the initial state, GP1 must undergo a low-to-high transition (GP1DH = 1) and GP2 must undergo a high-to-low transition (GP2DL = 1).

Example 5: GPINT based on a low-to-high and high-to-low transitions on both GP1 and GP2

- ❏ GPHM setup: GP1HM = 1, GP2HM = 1
- ❏ GPLM setup: GP1LM = 1, GP2LM = 1
- ❏ GPINT generation is caused by GP1DH, GP1DL, GP2DH, and GP2DL = 1
  - ■ Regardless of the initial state, both GP1 and GP2 must undergo transitions from original state back to original state.

3.2.3 Value AND Mode (GPINTDV = 1, LOGIC = 1)

Value AND mode allows the generation of an interrupt/event based on a set of signals matching some given values. GPINT is driven active when both of the following conditions are true:

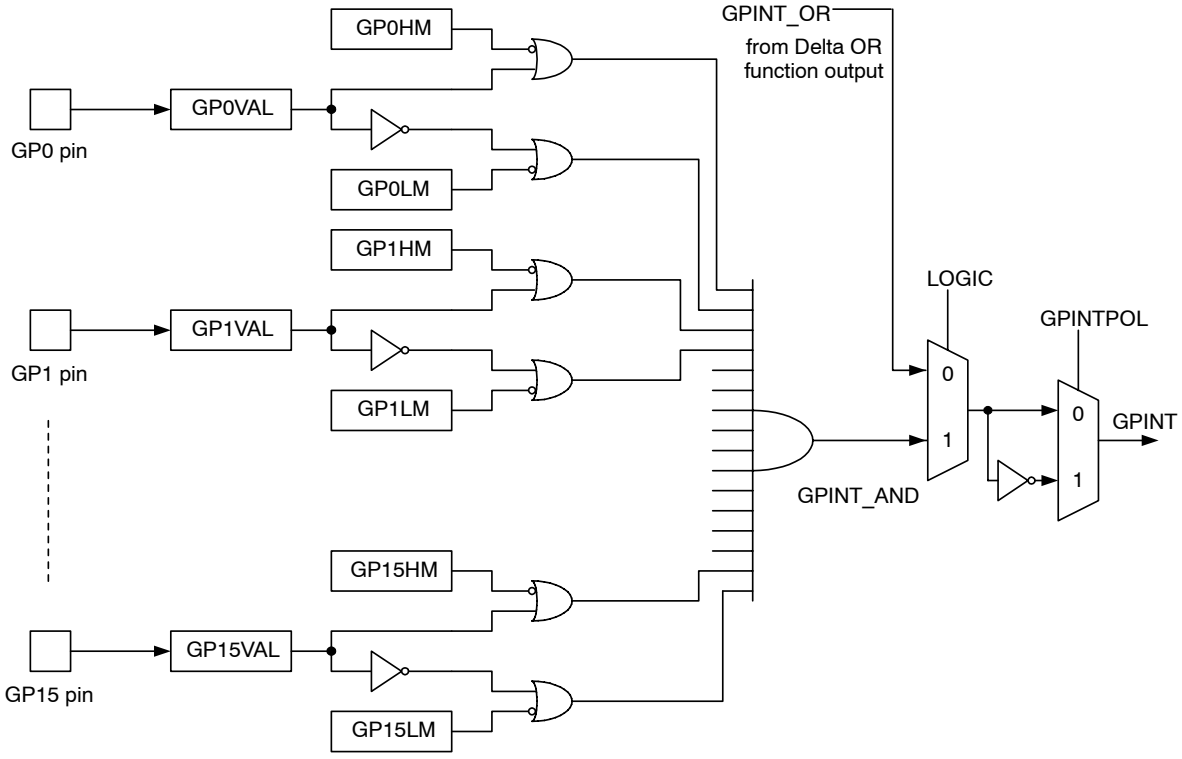
- ❏ All of the GPVAL bits are high for the group of GPIO signals with the GPHM bits set.
- ❏ All of the GPVAL bits are low for the group of GPIO signals with the GPLM bits set.

To generate an interrupt/event in Value AND mode, the GPINTDV and LOGIC bits in GPGC must be configured as follows:

- ❏ GPINTDV = 1; Value Mode
- ❏ LOGIC = 1; AND Mode

Figure 9 shows the functional block diagram of the Value AND Mode.

Figure 9. GPINT Generation in Value AND Mode



Note: The functional block diagram shows the mask logic as an OR with an inverter on the mask bit. This forces the OR to evaluate true when the mask bit is disabled. This is strictly a functional block diagram. The actual implementation prevents the GPINT from being asserted in the case that all the mask bits are disabled.

In addition, GPHM and GPLM must be configured properly to enable the corresponding GPVAL bits to be inputs to the logic function. GPINT is not generated if any given GPIO signal has both GPHM and GPLM bits asserted. This is because no GP*n* (and the corresponding GPVAL bit *n*) can be both high and low simultaneously. The following examples show the GPHM and GPLM setup. All the given GPIO pins in these examples are enabled as inputs (GPEN bit = 1, GPDIR bit = 0).

Example 1: GPINT based on GP1 = 1

- GPHM setup: GP1HM = 1
- GPLM setup: GP1LM = 0
- GPINT generation is caused by GP1 = 1
  - If GP1 is high (GPXVAL = 1) when entering this mode, GPINT is immediately asserted.
  - If GP1 is low (GPXVAL = 0) when entering this mode, a low-to-high transition (GPXVAL = 1) on GP1 generates GPINT.

Example 2: GPINT is not generated when GP*n*HM = GP*n*LM = 1

- GPHM setup: GP1HM = 1
- GPLM setup: GP1LM = 1
- GPINT is not generated because GP1 can never simultaneously be both low (GPXVAL = 0) and high (GPXVAL = 1)

Example 3: GPINT based on GP1 = GP2 = 1

- GPHM setup: GP1HM = 1, GP2HM = 1
- GPLM setup: GP1LM = 0, GP2LM = 0
- GPINT generation is caused by GP1VAL = 1 and GP2VAL = 1
  - If GP1 = GP2 = 1 when entering this mode, GPINT is immediately asserted.
  - If GP1 = 1 and GP2 = 0 when entering this mode, a low-to-high transition on GP2 (GP2VAL = 1), as GP1 stays high, generates GPINT.
  - If GP1 = GP2 = 0 when entering this mode, GPINT is generated when both GP1 and GP2 become high. If GP1 transitions high (GP1VAL = 1) then low (GP1VAL = 0) before GP2 transitions high, no GPINT is generated.

Example 4: GPINT based on GP1 = 1 and GP2 = 0

- ❏ GPHM setup: GP1HM = 1, GP2HM = 0
- ❏ GPLM setup: GP1LM = 0, GP2LM = 1
- ❏ GPINT generation is caused by GP1VAL = 1 and GP2VAL = 0
  - ■ As in previous examples, both GP1 and GP2 must simultaneously be at the defined state: GP1VAL = 1 and GP2VAL = 0.

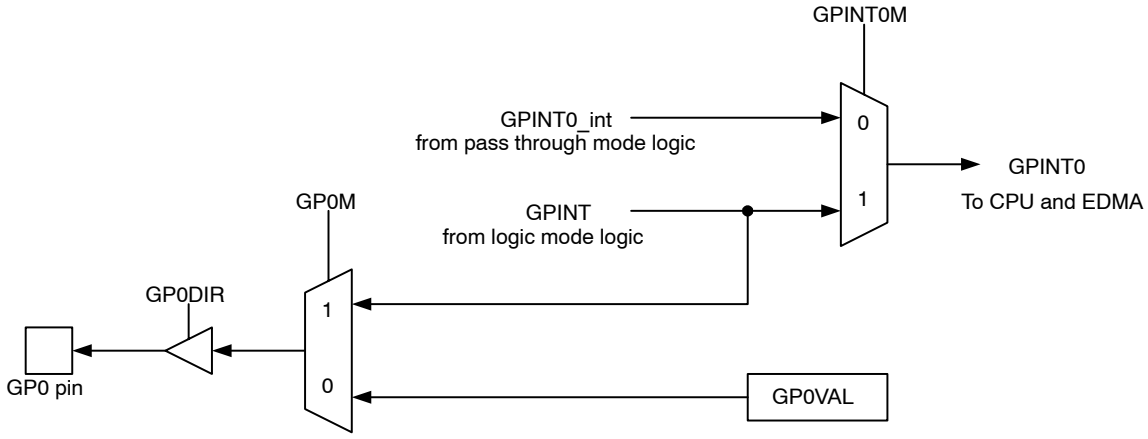
3.3 GPINT Muxing With GP0 and/or GPINT0

The logic function output signal GPINT can be used by both the DSP and an external device as follows:

- ❏ GPINT can generate a CPU interrupt and an EDMA event via GPINT0.
- ❏ If GP0 is configured as an output, GPINT can be driven out on GP0 to be used by external devices.

Figure 10 shows the connection of the GPINT signal.

Figure 10. GPINT Connection to GP0 and GPINT0



When GP0 is configured as an output (GP0DIR = 1), the GP0M bit controls whether the GP0 signal operates in GPIO mode or in logic mode. In GPIO mode (GP0M = 0), the value of the GP0VAL bit is driven out on GP0. In logic mode (GP0M = 1), GPINT is driven out on GP0. When GP0 is configured as an input, GP0M has no effect.

The GPINT0M bit controls whether the GPINT0 signal operates in pass-through mode or in logic mode. In pass-through mode (GPINT0M = 0), the GPINT0_int value from the pass-through mode logic is used to generate an interrupt/event to the CPU and EDMA. See section 3.1 and Figure 5 for details on the GPINT0_int signal. In logic mode (GPINT0M = 1), the logic mode output GPINT is used instead to generate an interrupt/event to the CPU and EDMA.

If GP0 is configured as an output, pass-through mode is disabled but logic mode is still supported and GPINT can be generated. No GPINT0_int is generated.

4 Interrupts and Events

The GPIO peripheral generates interrupts to the CPU and synchronization events to the EDMA via the internal GPINT*n* signals. The GPIO interrupts/events are summarized in Table 2. GPINT[1−15] can only be used in pass-through mode, but GPINT0 can be used in either pass-through mode or logic mode. All GPINT*n* are available as synchronization events to the EDMA. Only GPINT0 and GPINT[4−7] are available as interrupt sources to the CPU.

Table 2. GPIO Interrupts to CPU and Events to EDMA

Interrupt/Event Name	Description
GPINT0	GPINT0 is the interrupt/event output from pass-through mode or logic mode. In pass-through mode, GPINT0 reflects the value of GP0 or $\overline{GP0}$ (GPINT0_int). In logic mode, GPINT0 reflects the logic function output GPINT.
GPINT[1−15]	GPINT[1−15] are the interrupt outputs from pass-through mode. GPINT[1−15] reflect the value of GP[1−15] or $\overline{GP[1-15]}$ in pass-through mode.

5 Registers

The GPIO peripheral is configured through the registers listed in Table 3. See the device-specific datasheet for the memory address of these registers.

Table 3. GPIO Registers

Acronym	Register Name	Section
GPEN	GPIO Enable Register	5.1
GPDIR	GPIO Direction Register	5.2
GPVAL	GPIO Value Register	5.3
GPDH	GPIO Delta High Register	5.4
GPDL	GPIO Delta Low Register	5.5
GPHM	GPIO High Mask Register	5.6
GPLM	GPIO Low Mask Register	5.7
GPGC	GPIO Global Control Register	5.8
GPPOL	GPIO Interrupt Polarity Register	5.9

5.1 GPIO Enable Register (GPEN)

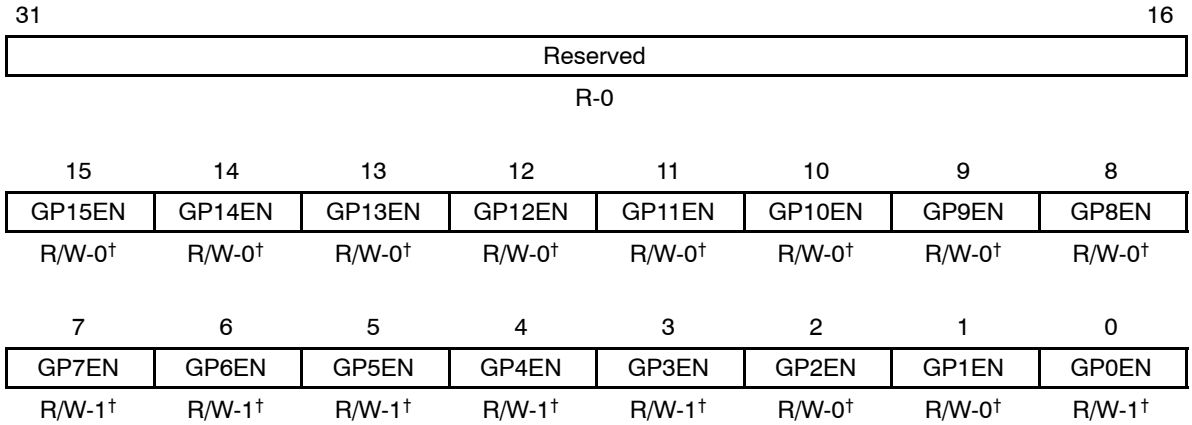
The GPIO enable register (GPEN) enables the GPIO pins for general-purpose input/output functions. To use any of the GPIO pins in general-purpose input/output mode, the corresponding GPXEN bit must be set to 1. The GPEN is shown in Figure 11 and described in Table 4.

Some GPIO signals are MUXed with other device signals. For these MUXed signals, the signal functionality is controlled by the following:

- **Device configuration inputs:** At reset, device configuration inputs select the MUXed signal to operate as either a GPIO pin or in the other mode.
- **GPEN bit fields:** A GPXEN bit *n* set to 1 indicates that the GP*n* pin operates as a GPIO signal controlled by the remaining GPIO registers. A GPXEN bit *n* cleared to 0 indicates that the GP*n* pin is disabled as a GPIO pin; it operates in the other mode.

For details on signal configuration, refer to your device-specific datasheet.

Figure 11. GPIO Enable Register (GPEN)



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

† The default values are device specific. Refer to the device-specific datasheet for the default values. The default values shown are for C64x devices.

Table 4. GPIO Enable Register (GPEN) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXEN	OF(*value*)	0–FFFFh	GPIO mode enable bit. A 16-bit unsigned value used to disable (bit value = 0) or enable (bit value = 1) a GP*n* pin as a general-purpose input/output pin.
			0	GP*n* pin is disabled as general-purpose input/output pin. It does not function as a GPIO pin and defaults to a high-impedance state.
			1	GP*n* pin is enabled as general-purpose input/output pin. It defaults to a high-impedance state.

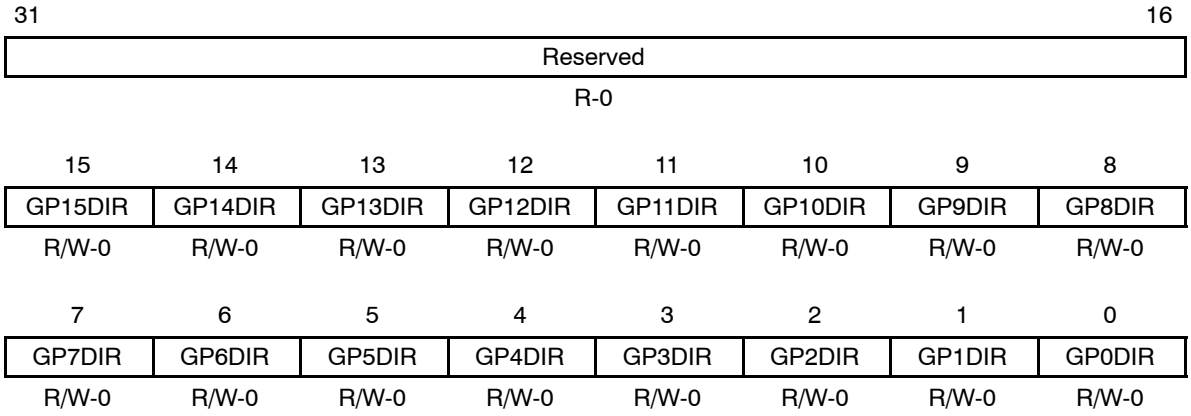
† For CSL implementation, use the notation GPIO_GPEN_GPXEN_*symval*

5.2 GPIO Direction Register (GPDIR)

The GPIO direction register (GPDIR) determines if a given GPIO pin is an input or an output. GPDIR only applies if the corresponding GPIO signal is enabled via the GPXEN bits in GPEN. The GPDIR is shown in Figure 12 and described in Table 5. By default, all the GPIO pins are configured as input pins.

When GPIO pins are configured as output pins, these pins do not have high-impedance capability. At reset, GPIO output pins default to the value in the GPIO value register (GPVAL), see section 5.3. If it is necessary to drive the GPIO output to the high-impedance state, the GPIO pins can be configured as an input pin and then changed to an output pin.

Figure 12. GPIO Direction Register (GPDIR)



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

Table 5. GPIO Direction Register (GPDIR) Field Descriptions

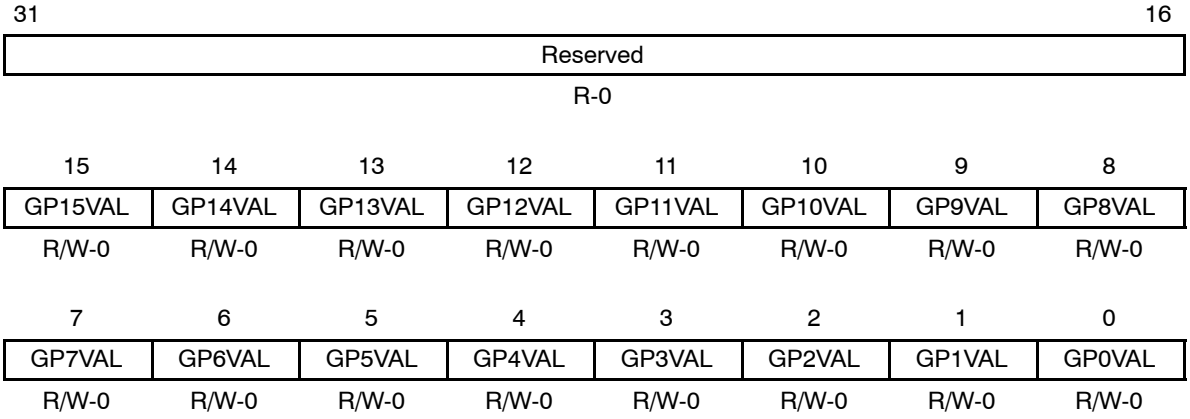
Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXDIR	OF(*value*)	0–FFFFh	GP*n* direction bit. A 16-bit unsigned value used to control the direction (input = 0, output = 1) of a GP*n* pin. Applies when the corresponding GPXEN bit in GPEN is set to 1.
			0	GP*n* pin is an input.
			1	GP*n* pin is an output.

† For CSL implementation, use the notation GPIO_GPDIR_GPXDIR_*symval*

5.3 GPIO Value Register (GPVAL)

The GPIO value register (GPVAL) indicates the value to be driven on a given GPIO output pin or the value detected on a given GPIO input pin. The GPVAL is shown in Figure 13 and described in Table 6.

Figure 13. GPIO Value Register (GPVAL)



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

Table 6. GPIO Value Register (GPVAL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXVAL	OF(*value*)	0–FFFFh	GP*n* value bit. A 16-bit unsigned value used to determine the value detected at a GP*n* input/output pin. Applies when the corresponding GPXEN bit in GPEN is set to 1.
				When GP*n* pin is an input (GPXDIR = 0).
			0	A value of 0 is latched from the GP*n* input pin.
			1	A value of 1 is latched from the GP*n* input pin.
				When GP*n* pin is an output (GPXDIR = 1).
			0	GP*n* signal is driven low.
			1	GP*n* signal is driven high.

† For CSL implementation, use the notation GPIO_GPVAL_GPXVAL_*symval*

5.4 GPIO Delta High Register (GPDH)

The GPIO delta high register (GPDH) indicates whether a given GPIO input has undergone a low-to-high transition. If the given GPIO pin is configured as an output, the corresponding bit in GPDH maintains its previous value. Writing a 1 to the corresponding bit clears the bit, writing a 0 has no effect. The GPDH is shown in Figure 14 and described in Table 7.

Figure 14. GPIO Delta High Register (GPDH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
GP15DH	GP14DH	GP13DH	GP12DH	GP11DH	GP10DH	GP9DH	GP8DH
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
GP7DH	GP6DH	GP5DH	GP4DH	GP3DH	GP2DH	GP1DH	GP0DH
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 = Read only; R/W = Read/Write; -*n* = value after reset

Table 7. GPIO Delta High Register (GPDH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXDH	OF(*value*)	0–FFFFh	GP*n* delta high bit. A 16-bit unsigned value used to determine if a low-to-high transition is detected on the GP*n* input pin. Applies when the corresponding GP*n* pin is enabled as an input (GPXEN = 1 and GPXDIR = 0).
			0	A low-to-high transition is not detected on GP*n* pin.
			1	A low-to-high transition is detected on GP*n* pin.

† For CSL implementation, use the notation GPIO_GPDH_GPXDH_*symval*

5.5 GPIO Delta Low Register (GPDL)

The GPIO delta low register (GPDL) indicates whether a given GPIO input has undergone a high-to-low transition. If the given GPIO pin is configured as an output, the corresponding bit in GPDL maintains its previous value. Writing a 1 to the corresponding bit clears the bit, writing a 0 has no effect. The GPDL is shown in Figure 15 and described in Table 8.

Figure 15. GPIO Delta Low Register (GPDL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
GP15DL	GP14DL	GP13DL	GP12DL	GP11DL	GP10DL	GP9DL	GP8DL
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
GP7DL	GP6DL	GP5DL	GP4DL	GP3DL	GP2DL	GP1DL	GP0DL
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 = Read only; R/W = Read/Write; -*n* = value after reset

Table 8. GPIO Delta Low Register (GPDL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXDL	OF(*value*)	0–FFFFh	GP*n* delta low bit. A 16-bit unsigned value used to determine if a high-to-low transition is detected on the GP*n* input pin. Applies when the corresponding GP*n* pin is enabled as an input (GPXEN = 1 and GPXDIR = 0).
			0	A high-to-low transition is not detected on GP*n* pin.
			1	A high-to-low transition is detected on GP*n* pin.

† For CSL implementation, use the notation GPIO_GPDL_GPXDL_*symval*

5.6 GPIO High Mask Register (GPHM)

The GPIO high mask register (GPHM) is used to enable a given GPIO input to cause a CPU interrupt or an EDMA event generation. If a GPHM bit (GP*n*HM) is disabled, the value or transition on the corresponding GP*n* pin does not cause an interrupt/event generation. If the mask bit is enabled, the corresponding GP*n* input may cause an interrupt/event to be generated depending on the interrupt mode selected in GPGC. Refer to section 3 for details on the function of GPHM in interrupt/event generation. The GPHM is shown in Figure 16 and described in Table 9.

Figure 16. GPIO High Mask Register (GPHM)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
GP15HM	GP14HM	GP13HM	GP12HM	GP11HM	GP10HM	GP9HM	GP8HM
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
GP7HM	GP6HM	GP5HM	GP4HM	GP3HM	GP2HM	GP1HM	GP0HM
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 = Read only; R/W = Read/Write; -*n* = value after reset

Table 9. GPIO High Mask Register (GPHM) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXHM	OF(*value*)	0–FFFFh	GP*n* high mask bit. A 16-bit unsigned value used to disable (bit value = 0) or enable (bit value = 1) an interrupt/event generation based on either the corresponding GPXDH bit or GPXVAL bit. Applies when the corresponding GP*n* pin is enabled as an input (GPXEN = 1 and GPXDIR = 0).
			0	Interrupt/event generation is disabled for GP*n* pin. The value or transition on GP*n* pin does not cause an interrupt/event generation.
			1	Interrupt/event generation is enabled for GP*n* pin.

† For CSL implementation, use the notation GPIO_GPHM_GPXHM_*symval*

5.7 GPIO Low Mask Register (GPLM)

The GPIO low mask register (GPLM) is used to enable a given GPIO input to cause a CPU interrupt or an EDMA event generation. If a GPLM bit (GP*n*LM) is disabled, the value or transition on the corresponding GP*n* pin does not cause an interrupt/event generation. If the mask bit is enabled, the corresponding GP*n* input may cause an interrupt/event to be generated depending on the interrupt mode selected in GPGC. Refer to section 3 for details on the function of GPLM in interrupt/event generation. The GPLM is shown in Figure 17 and described in Table 10.

Figure 17. GPIO Low Mask Register (GPLM)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
GP15LM	GP14LM	GP13LM	GP12LM	GP11LM	GP10LM	GP9LM	GP8LM
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
GP7LM	GP6LM	GP5LM	GP4LM	GP3LM	GP2LM	GP1LM	GP0LM
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 = Read only; R/W = Read/Write; -*n* = value after reset

Table 10. GPIO Low Mask Register (GPLM) Field Descriptions

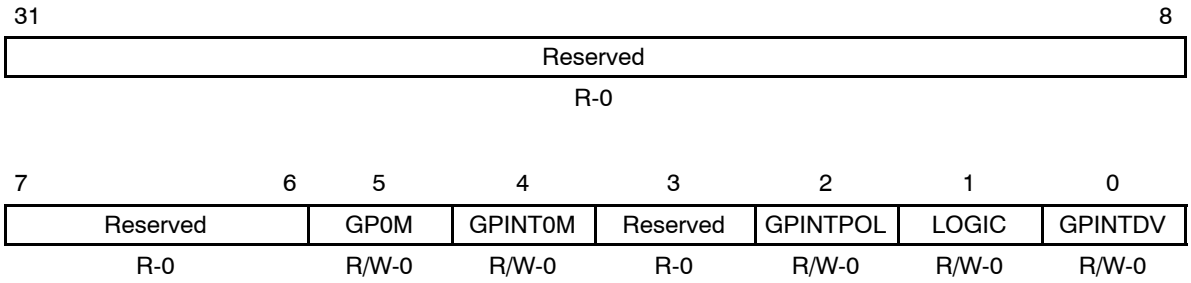
Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15–0	GPXLM	OF(*value*)	0–FFFFh	GP*n* low mask bit. A 16-bit unsigned value used to disable (bit value = 0) or enable (bit value = 1) an interrupt/event generation based on either the corresponding GPXDL bit or GPXVAL bit. Applies when the corresponding GP*n* pin is enabled as an input (GPXEN = 1 and GPXDIR = 0).
			0	Interrupt/event generation is disabled for GP*n* pin. The value or transition on GP*n* pin does not cause an interrupt/event generation.
			1	Interrupt/event generation is enabled for GP*n* pin.

† For CSL implementation, use the notation GPIO_GPLM_GPXLM_*symval*

5.8 GPIO Global Control Register (GPGC)

The GPIO global control register (GPGC) configures the interrupt/event generation of the GPIO peripheral. The GPGC is shown in Figure 18 and described in Table 11.

Figure 18. GPIO Global Control Register (GPGC)



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

Table 11. GPIO Global Control Register (GPGC) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31–6	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
5	GP0M			GP0 output mode bit. Applies only if GP0 is configured as an output (GPXDIR bit 0 = 1).
		GPIOMODE	0	GPIO mode—GP0 output is based on GP0 value (GPXVAL bit 0 value in GPVAL).
		LOGICMODE	1	Logic mode—GP0 output is based on value of internal logic mode interrupt/event signal GPINT.
4	GPINT0M			GPINT0 interrupt/event generation mode bit.
		PASSMODE	0	Pass-through mode—GPINT0 interrupt/event generation is based on GP0 input value (GPXVAL bit 0 value in GPVAL).
		LOGICMODE	1	Logic Mode—GPINT0 interrupt/event generation is based on GPINT.
3	Reserved	–	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

† For CSL implementation, use the notation GPIO_GPGC_*field_symval*

Table 11. GPIO Global Control Register (GPGC) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
2	GPINTPOL			GPINT polarity bit. Applies only to logic mode (GPINT0M = 1).
		LOGICTRUE	0	GPINT is active (high) when the logic combination of the GPIO inputs is evaluated true.
		LOGICFALSE	1	GPINT is active (high) when the logic combination of the GPIO inputs is evaluated false.
1	LOGIC			GPINT logic mode select bit. Applies only to logic mode (GPINT0M = 1).
		ORMODE	0	OR mode—GPINT is generated based on the logical-OR of all GP*n* events enabled in GPHM or GPLM.
		ANDMODE	1	AND mode—GPINT is generated based on the logical-AND of all GP*n* events enabled in GPHM or GPLM.
0	GPINTDV			GPINT delta/value mode select bit. Applies only to logic mode (GPINT0M = 1).
		DELTAMODE	0	Delta mode—GPINT is generated based on a logic combination of transitions on the GP*n* pins. The corresponding bits in GPHM and/or GPLM must be set.
		VALUEMODE	1	Value mode—GPINT is generated based on a logic combination of values on the GP*n* pins. The corresponding bits in GPHM and/or GPLM must be set.

† For CSL implementation, use the notation GPIO_GPGC_*field_symval*

5.9 GPIO Interrupt Polarity Register (GPPOL)

The GPIO interrupt polarity register (GPPOL) selects the polarity of the GPINT*n* interrupt/event signals in pass-through mode (see section 3.1). For GPINT mapping with EDMA events and external interrupts, refer to your device-specific datasheet. To use GPINT0 in pass-through mode, the GPINT0M bit in GPGC must be cleared to 0. The GPPOL is shown in Figure 19 and described in Table 12.

Figure 19. GPIO Interrupt Polarity Register (GPPOL)

31							16
Reserved							
R-0							

15	14	13	12	11	10	9	8
GPINT15POL	GPINT14POL	GPINT13POL	GPINT12POL	GPINT11POL	GPINT10POL	GPINT9POL	GPINT8POL
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
GPINT7POL	GPINT6POL	GPINT5POL	GPINT4POL	GPINT3POL	GPINT2POL	GPINT1POL	GPINT0POL
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 = Read only; R/W = Read/Write; -*n* = value after reset

Table 12. GPIO Interrupt Polarity Register (GPPOL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31–16	Reserved	–	0	Reserved. The reserved bit location is always read as zero. A value written to this field has no effect.
15–0	GPINTXPOL	OF(*value*)	0–FFFFh	GPINT*n* polarity bit. A 16-bit unsigned value used to select a rising edge (bit value = 0) or falling edge (bit value = 1) to determine when GPINT*n* is asserted. Applies only to pass-through mode (GPINT0M = 0 in GPGC).
			0	GPINT*n* is asserted (high) based on a rising edge of GP*n* (effectively based on the value of the corresponding GPXVAL bit).
			1	GPINT*n* is asserted (high) based on a falling edge of GP*n* (effectively based on the inverted value of the corresponding GPXVAL bit).

† For CSL implementation, use the notation GPIO_GPPOL_GPINTXPOL_*symval*

This page is intentionally left blank.

Revision History

Table 13 lists the changes made since the previous version of this document.

Table 13. Document Revision History

Page	Additions/Modifications/Deletions
	This document has been reviewed for accuracy and there are no changes since the previous version (July 2003) of this document.

This page is intentionally left blank.

Index

L

N

O

P

R

T

V