

## Introduction

This document describes the specifications for the General Purpose Input/Output (GPIO) core for the On Chip Processor Bus (OPB). The OPB GPIO is a 32-bit peripheral that attaches to the OPB bus.

## Features

- OPB v2.0 bus interface with byte-enable support
- Configurable as single or dual GPIO channel(s)
- Number of GPIO bits configurable from 1 to 32 bits
- Each GPIO bit dynamically programmable as input or output
- Can be configured as inputs-only on a per channel basis to reduce resource utilization
- Ports for both 3-state and non 3-state connections
- Independent reset values for each bit of all registers
- Optional interrupt request generation

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	QPro™-R Virtex™-II, QPro Virtex-II, Spartan™-II, Spartan-II-E, Spartan-3, Spartan-3E, Virtex, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-E	
Version of Core	opb_gpio	v3.01b
Resources Used (GPIO_WIDTH = 32)		
	Min	Max
Slices	84	306
LUTs	20	268
FFs	141	413
Block RAMs	N/A	N/A
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	6.2i or later	
Verification	ModelSim SE/EE 5.8b or later	
Simulation	ModelSim SE/EE 5.8b or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

© 2005 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.  
 NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

## Functional Description

The OPB GPIO design provides a general purpose input/output interface to a 32-bit On-Chip Peripheral Bus (OPB). The OPB GPIO can be configured as either a single or a dual channel device. The channel width is configurable and when both channels are enabled, the channel width remains the same for both.

The OPB GPIO design supports 3-state as well as independent input and output ports. An input port may be configured to take its external input either from the bidirectional 3-state pin or from the dedicated input only pins. For a port configured as output, the data is driven out through a 3-state buffer as well as to an output only pin. The ports can be configured dynamically for input or output by enabling or disabling the 3-state buffer.

Each channel is individually configurable as input ports only. When a channel is configured as input only, the logic required to implement the output path and three-state controls are removed resulting in reduced resource utilization.

The channels may be configured to generate an interrupt when a transition on any of their inputs occurs.

The major interfaces and modules of the design are shown in [Figure 1](#) and described in the subsequent sections. The OPB GPIO design is comprised of modules

- OPB\_IPIF
- GPIO\_CORE

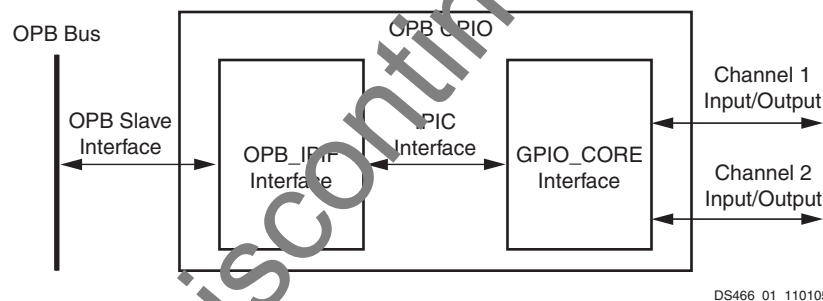


Figure 1: OPB GPIO Block Diagram

### OPB\_IPIF

OPB\_IPIF provides an interface between GPIO\_CORE and the OPB 32-bit bus standard. It supports the following functions:

- OPB slave interface: The OPB\_IPIF module implements the basic functionality of OPB slave operation and does the necessary protocol and timing translation between the OPB bus and the IPIF interface
- Interrupt support: The OPB\_IPIF module provides support for enabling interrupts, to capture an interrupt event, and to maintain the interrupt status. The registers required to support interrupts are implemented within this module.

For more information on OPB\_IPIF interface, refer OPB IPIF documents listed in the [Reference Documents](#) section of this document.

## GPIO\_CORE

GPIO\_CORE provides an interface between the IPIC interface and the OPB GPIO channels. The GPIO\_CORE consists of registers and multiplexers for reading and writing the OPB GPIO channel registers. It also includes the necessary logic to identify an interrupt event when the channel input changes.

Figure 2 shows a detailed diagram of dual channel implementation of the GPIO\_CORE. The 3-state buffers in the figure are not actually part of the core. The 3-state buffers are added in the synthesis process, usually automatically with an Add I/Os option. For the sake of simplicity, the control signals of the IPIC interface are not shown in Figure 2.

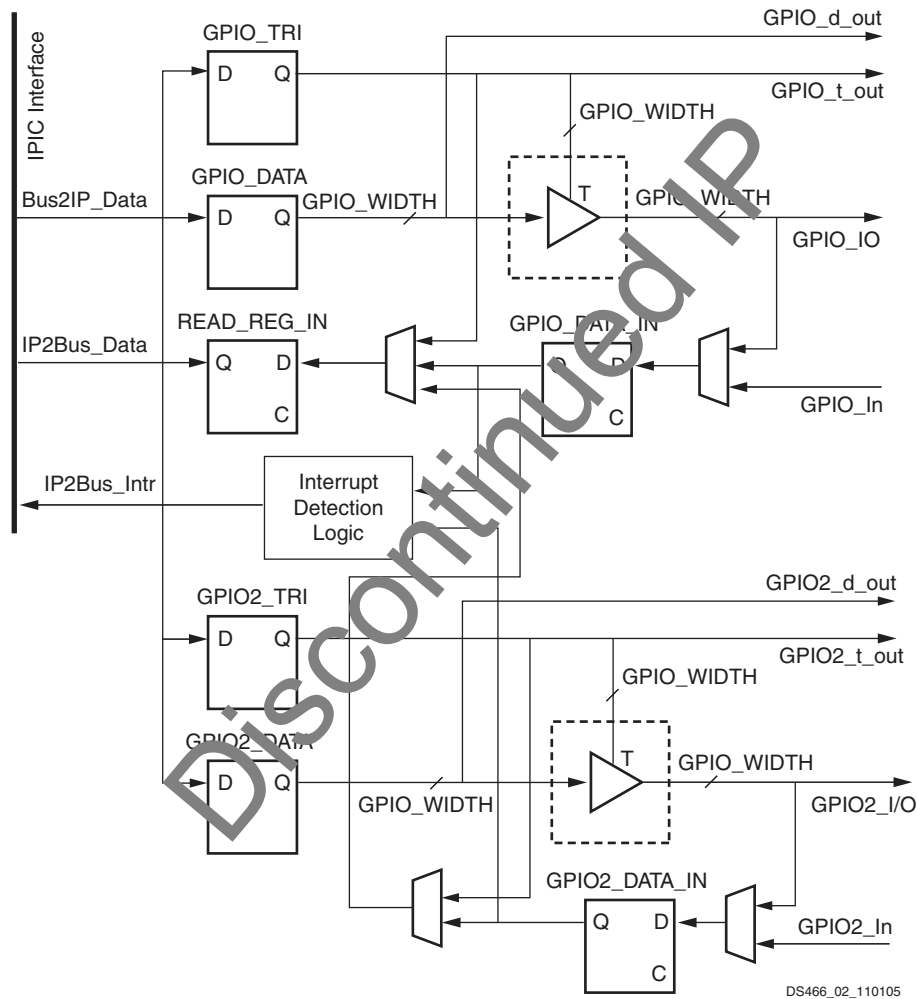


Figure 2: GPIO\_CORE Dual Channel Implementation

## OPB GPIO Design Parameters

Certain features can be parameterized in the OPB GPIO design. Some of these parameters control the interface to the OPB bus while others provide information to tailor the GPIO\_CORE logic to minimize resource utilization. The features that can be parameterized in the OPB GPIO are shown in Table 1.

Table 1: OPB GPIO Design Parameters

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
<b>GPIO</b>					
G1	OPB GPIO Base Address	C_BASEADDR	Valid OPB Address (1,2)	None (3)	std_logic_vector
G2	OPB GPIO High Address	C_HIGHADDR	Valid OPB Address (1,2)	None (3)	std_logic_vector
G3	Target FPGA Family	C_FAMILY	Any FPGA family	virtex2	string
G4	GPIO Data Bus Width	C_GPIO_WIDTH	1-32	32	integer
G5	OPB GPIO Interrupt	C_INTERRUPT_PRESENT	0/1	0	integer
G6	Inputs Only	C_ALL_INPUTS	0/1	0	integer
G7	Select GPIO_IO as input source	C_IS_BIDIR	0/1	1	integer
G8	GPIO_DATA reset value	C_DOUT_DEFAULT	Any valid std_logic_vector	00000000	std_logic_vector
G9	GPIO_TRI reset value	C_TRI_DEFAULT	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
G10	Use dual channel	C_IS_DUAL	0/1	0	integer
G11	Channel 2 inputs only	C_ALL_INPUTS_2	0/1	0	integer
G12	Channel 2 select GPIO_IO as input source	C_IS_BIDIR_2	0/1	1	integer
G13	GPIO2_DATA reset value	C_DOUT_DEFAULT_2	Any valid std_logic_vector	00000000	std_logic_vector
G14	GPIO2_TRI reset value	C_TRI_DEFAULT_2	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
G15	User ID for MMR / reset register	C_USER_ID_CODE	0-255	3	integer
<b>OPB Interface</b>					
G16	OPB Address Width	C_OPB_AWIDTH	32	32	integer
G17	OPB Data Width	C_OPB_DWIDTH	32	32	integer

**Notes:**

1. The range specified by C\_BASEADDR and C\_HIGHADDR must comprise a complete, contiguous power of two range such that the range = 2<sup>n</sup> and the n least significant bits of C\_BASEADDR is zero. This range needs to encompass the address range required by the OPB GPIO
2. The minimum address range specified by C\_BASEADDR and C\_HIGHADDR must be at least 0xFF
3. No default value will be specified to ensure that the actual value is set i.e. if the value is not set, a compiler error will be generated

## Allowable Parameter Combinations

The address range specified by C\_BASEADDR and C\_HIGHADDR must comprise a complete, contiguous power of two such that the range =  $2^n$ . The n least significant bits of C\_BASEADDR must be zero.

The range specified by C\_BASEADDR and C\_HIGHADDR must encompass the memory space required by the OPB GPIO. The minimum range specified by C\_BASEADDR and C\_HIGHADDR should be at least 0xFF. If C\_INTERRUPT\_PRESENT parameter is set then the address range specified by C\_BASEADDR and C\_HIGHADDR should be at least 0x1FF.

No default value will be specified for C\_BASEADDR and C\_HIIGHADDR in order to enforce that the user configures these parameters with the actual values. If the values are not set for C\_BASEADDR and C\_HIGHADDR, a compiler error will be generated.

## OPB GPIO I/O Signals

The I/O signals for this reference design are listed in [Table 2](#).

Table 2: OPB GPIO I/O Signals

Port	Signal Name	Interface	I/O	Initial State	Description
<b>OPB</b>					
P1	OPB_ABus(0:C_OPB_AWIDTH-1)	OPB	I		OPB Address Bus
P2	OPB_BE(0:C_OPB_DWIDTH/8-1)	OPB	I		OPB Byte Enables
P3	OPB_DBus(0:C_OPB_DWIDTH-1)	OPB	I		OPB Data Bus
P4	OPB_RNW	OPB	I		OPB Read, Not Write
P5	OPB_select	OPB	I		OPB Select
P6	OPB_seqAddr	OPB	I		OPB Sequential Address
P7	SIn_DBus(0:C_OPB_DWIDTH-1)	OPB	O	0	OPB GPIO Data Bus
P8	SIn_errAck	OPB	O	0	OPB GPIO Error Acknowledge
P9	SIn_retry	OPB	O	0	OPB GPIO Retry (Always inactive)
P10	SIn_toutSup	OPB	O	0	OPB GPIO Timeout Suppress (Always inactive)
P11	SIn_xferAck	OPB	O	0	OPB GPIO Transfer Acknowledge (Always inactive)
<b>System</b>					
P12	OPB_Clk	OPB	I		OPB Clock
P13	OPB_Rst	OPB	I		OPB Reset
P14	IP2INTC_Irpt	IPIF Interrupt	O	0	OPB GPIO Interrupt Active high signal
P16	GPIO_in(0 to C_GPIO_WIDTH-1)	GPIO	I		Channel 1 General purpose input
P17	GPIO_d_out(0 to C_GPIO_WIDTH-1)	GPIO	O	0 <sup>(1)</sup>	Channel 1 data register (GPIO_DATA) output

Table 2: OPB GPIO I/O Signals (Contd)

Port	Signal Name	Interface	I/O	Initial State	Description
P18	GPIO_t_out(0 to C_GPIO_WIDTH-1)	GPIO	O	1 <sup>(2)</sup>	Channel 1 3-state control register (GPIO_TRI) output
P19	GPIO_IO(0 to C_GPIO_WIDTH-1)	GPIO	I/O	Z <sup>(3)</sup>	Channel 1 General purpose I/O
P20	GPIO2_in(0 to C_GPIO_WIDTH-1)	GPIO	I		Channel 2 General purpose input
P21	GPIO2_d_out(0 to C_GPIO_WIDTH-1)	GPIO	O	0 <sup>(4)</sup>	Channel 2 data register (GPIO2_DATA) output
P22	GPIO2_t_out(0 to C_GPIO_WIDTH-1)	GPIO	O	1 <sup>(5)</sup>	Channel 2 3-state control register (GPIO2_TRI) output
P19	GPIO_IO(0 to C_GPIO_WIDTH-1)	GPIO	I/O	Z <sup>(6)</sup>	Channel 2 General purpose I/O

Notes:

- GPIO\_d\_out has an initial value of 0 only if the default value of C\_DOUT\_DEFAULT is used.
- GPIO\_t\_out has an initial value of 1 only if the default value of C\_TRI\_DEFAULT is used.
- GPIO remains at high impedance only if the default value of C\_TRI\_DEFAULT is used.
- GPIO2\_d\_out has an initial value of 0 only if the default value of C\_DOUT\_DEFAULT\_2 is used.
- GPIO2\_t\_out has an initial value of 1 only if the default value of C\_TRI\_DEFAULT\_2 is used.
- GPIO2 remains at high impedance only if the default value of C\_TRI\_DEFAULT\_2 is used.

### Parameter - Port Dependencies

The width of the OPB GPIO channel registers depends on some of the parameters. In addition, when certain features are parameterized away, the related logic is removed. The dependencies between the OPB GPIO design parameters and the I/O ports are shown in Table 3.

Table 3: Parameter-Port Dependencies

Name	Affects	Depends	Relationship Description
C_OPB_DWIDTH	OPB_BE OPB_DBus Slr_DBus	0 to (C_OPB_DWIDTH/8) -1 0 to C_OPB_DWIDTH -1 0 to C_OPB_DWIDTH -1	Number of byte enables decoded Width of the OPB data bus Width of the slave read data bus
C_OPB_AWIDTH	OPB_ABus	0 to C_OPB_AWIDTH -1	Width of the OPB address bus
C_GPIO_WIDTH	GPIO_DATA GPIO_TRI GPIO2_DATA GPIO2_TRI	1 to C_GPIO_WIDTH	The size of the registers GPIO_DATA and GPIO_TRI determines the number of GPIO_in, GPIO_IO, GPIO_d_out and GPIO_t_out pins. Similarly, the size of the registers GPIO2_DATA and GPIO2_TRI determines the number of GPIO2_in, GPIO2_IO, GPIO2_d_out and GPIO2_t_out pins
C_IS_DUAL	GPIO2_in GPIO2_IO GPIO2_d_out GPIO2_t_out	1 to C_GPIO_WIDTH	When C_IS_DUAL is 1, channel 2 is created.
C_ALL_INPUTS	GPIO_IO GPIO_d_out GPIO_t_out	1 to C_GPIO_WIDTH	Eliminates the logic required for GPIO_d_out and GPIO_t_out. Both GPIO_d_out and GPIO_t_out are driven low. GPIO_IO ports are driven to high impedance

Table 3: Parameter-Port Dependencies (Contd)

Name	Affects	Depends	Relationship Description
C_ALL_INPUTS_2	GPIO2_IO GPIO2_d_out GPIO2_t_out	1 to C_GPIO_WIDTH	Eliminates the logic required for GPIO2_d_out and GPIO2_t_out. Both GPIO2_d_out and GPIO2_t_out are driven low. GPIO2_IO ports are driven to high impedance
C_IS_BIDIR	GPIO_IO GPIO_in	1 to C_GPIO_WIDTH	GPIO_IO is used for input rather than GPIO_in
C_IS_BIDIR_2	GPIO2_IO GPIO2_in	1 to C_GPIO_WIDTH	GPIO2_IO is used for input rather than GPIO2_in

## OPB GPIO Registers

There are four internal registers in the OPB GPIO design as shown in Table 4. These registers are implemented in the GPIO\_CORE interface module. The memory map of the OPB GPIO design is determined by setting the C\_BASEADDR parameter. The internal registers of the OPB GPIO are at a fixed offset from the base address. The OPB GPIO internal registers and their offset are listed in Table 4.

Table 4: OPB GPIO Registers

Register Name	Description	OPB Address	Access
GPIO_DATA	Channel 1 OPB GPIO Data Register	C_BASEADDR + 0x00	Read/Write
GPIO_TRI	Channel 1 OPB GPIO 3-state Register	C_BASEADDR + 0x04	Read/Write
GPIO2_DATA	Channel 2 OPB GPIO Data Register	C_BASEADDR + 0x08	Read/Write
GPIO2_TRI	Channel 2 OPB GPIO 3-state Register	C_BASEADDR + 0x0C	Read/Write

Depending on the value of certain configuration parameters, some of these registers are removed. The parameter - register dependency is described in Table 5. A write to an unimplemented register has no effect. An attempt to read the unimplemented register will return unknown values.

Table 5: Parameter-Register Dependency

Parameter Values			Register Retainability			
C_IS_DUAL	C_ALL_INPUTS	C_ALL_INPUTS_2	GPIO_DATA	GPIO_TRI	GPIO2_DATA	GPIO2_TRI
0	1	X (1)	Yes	No	No	No
0	0	X (1)	Yes	Yes	No	No
0	1	X (1)	Yes	No	No	No
0	0	X (1)	Yes	Yes	No	No
1	1	1	Yes	No	Yes	No

Table 5: Parameter-Register Dependency (Contd)

Parameter Values			Register Retainability			
C_IS_DUAL	C_ALL_INPUTS	C_ALL_INPUTS_2	GPIO_DATA	GPIO_TRI	GPIO2_DATA	GPIO2_TRI
1	0	1	Yes	Yes	Yes	No
1	1	0	Yes	No	Yes	Yes
1	0	0	Yes	Yes	Yes	Yes
1	1	1	Yes	No	Yes	No
1	0	1	Yes	Yes	Yes	No
1	1	0	Yes	No	Yes	Yes
1	0	0	Yes	Yes	Yes	Yes

**Notes:**

- When C\_IS\_DUAL = 0, the core is configured for single channel and hence the parameter C\_ALL\_INPUTS\_2 has no effect
- Depending on the value of C\_GPIO\_WIDTH, the data registers and the 3-state control registers (GPIO\_DATA, GPIO\_TRI, GPIO2\_DATA and GPIO2\_TRI) when implemented, get trimmed to the size of value specified by C\_GPIO\_WIDTH

**OPB GPIO Data Register (GPIOx\_DATA)**

OPB GPIO data register is used to read the input ports and write to the output ports. When a port is configured as input, writing to the port has no effect. When a port is configured as output, reading the port returns the value of the corresponding bit in the OPB GPIO data register.

There are two OPB GPIO data registers (GPIO\_DATA and GPIO2\_DATA), one corresponding to each channel. The channel 1 data register (GPIO\_DATA) is always present while the channel 2 data register (GPIO2\_DATA) is present only if the core is configured for dual channel i.e. C\_IS\_DUAL = 1.

The OPB GPIO Data Register is shown in Figure 3 and Table 6 details its functionality.



Figure 3: OPB GPIO Data Register



Table 6: OPB GPIO Data Register Description

Bits	Name	Core Access	Description	Reset Value
0:31	GPIOx_DATA	Read/Write	OPB GPIO Data For I/O Programmed as inputs: <b>R</b> : Reads value on input pin <b>W</b> : No effect For I/O Programmed as outputs: <b>R</b> : Reads value in OPB GPIO data register <b>W</b> : Writes value to OPB GPIO data register and output pin	C_DOUT_DEFAULT C_DOUT_DEFAULT_2

### OPB GPIO 3-State Register (GPIOx\_TRI)

The OPB GPIO 3-state register is used to configure the ports dynamically as input or output. When a bit within this register is set, the corresponding I/O port is configured as input port. When a bit is reset, the corresponding I/O port is configured as output port.

There are two OPB GPIO 3-state control registers (GPIO\_TRI and GPIO2\_TRI), one corresponding to each channel. The channel 1 3-state control register (GPIO\_TRI) is present when channel 1 is not configured for all input ports (C\_ALL\_INPUTS = 0). The channel 2 3-state control register (GPIO2\_TRI) is present only if the core is configured for dual channel (C\_IS\_DUAL = 1) and channel 2 is not configured for all input ports (C\_ALL\_INPUTS\_2 = 0).

The OPB GPIO 3-state Register is shown in Figure 4 and its functionality is described in Table 7.

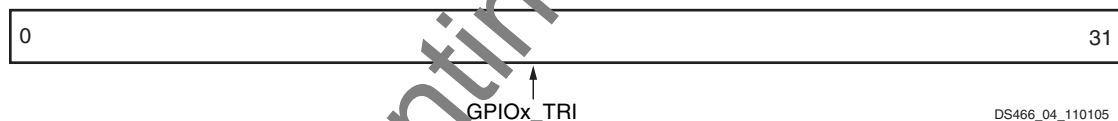


Figure 4: OPB GPIO 3-State Register

Table 7: OPB GPIO 3-State Register Description

Bits	Name	Core Access	Description	Reset Value
0:31	GPIOx_TRI	Read/Write	OPB GPIO 3-state control. Each I/O pin of the OPB GPIO is individually programmable as an input or output. For each bit: 0 I/O pin configured as output 1 I/O pin configured as input	C_TRI_DEFAULT C_TRI_DEFAULT_2

### OPB GPIO Interrupts

The OPB GPIO can be configured under the control of the C\_INTERRUPT\_PRESENT generic to generate an interrupt when a transition occurs in any of the channel inputs. The GPIO\_CORE interface module includes interrupt detection logic to identify any transition on channel inputs. When a transition is detected, the same is indicated to the OPB\_IPIF module interface. The OPB\_IPIF module implements the necessary registers to enable and maintain the status of the interrupts.

To support interrupt capability for channels, the OPB\_IPIF interface module implements the following registers:

- OPB Global Interrupt Enable register (OPB GIE)
- IP Interrupt Enable Register (IP IER)
- IP Interrupt Status Register (IP ISR)

The IP IER implements independent interrupt enable bit for each channel while the OPB Global Interrupt Enable Register provides the master enable/disable for the interrupt output to the processor. The IP ISR implements independent interrupt status bit for each channel. The IP ISR provides Read and Toggle-On-Write access. The Toggle-On-Write mechanism for the IP Interrupt Status Register avoids the requirement on the user interrupt service routine to perform Read-Modify-Write operation to clear the status bit of the interrupt. Read-Modify-Write operations can lead to inadvertent clearing of interrupts captured in the time period between the read and write operations.

Table 8 details the OPB GPIO interrupt registers and their offset from the base address of OPB GPIO memory map. Note that these registers are meaningful only if the C\_INTERRUPT\_PRESENT generic is

Table 8: OPB GPIO Interrupt Registers

Register Name	Description	OPB Address	Access
OPB GIE	OPB Global Interrupt Enable Register	C_BASEADDR + 0x11C	Read/Write
IP IER	IP Interrupt Enable Register	C_BASEADDR + 0x128	Read/Write
IP ISR	IP Interrupt Status Register	C_BASEADDR + 0x120	Read/TOW[1]

**Notes:**  
 1. Toggle-On-Write (TOW) access toggles the status of the bit when a value of "1" is written to the corresponding bit

**OPB Global Interrupt Enable Register (OPB GIE)**

The OPB Global Interrupt Enable Register provides the master enable/disable for the interrupt output to the processor. This is a single bit read/write register as shown in Figure 5. This register is meaningful only if the parameter C\_INTERRUPT\_PRESENT is 1.

Note that this bit must be set to generate interrupts, even if the interrupts are enabled in the IP Interrupt Enable Register (IP IER). The bit definition for OPB Global Interrupt Enable Register is given in Table 9.

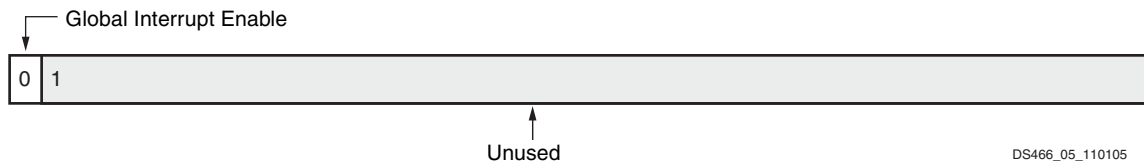


Figure 5: OPB Global Interrupt Enable Register

DS466\_05\_110105

Table 9: OPB Global Interrupt Enable Register Description

Bit(s)	Name	Core Access	Reset Value	Description
0	Global Interrupt Enable	Read/Write	0	Master enable for the device interrupt output to the system interrupt controller. <ul style="list-style-type: none"> <li>• 1 = Enabled</li> <li>• 0 = Disabled</li> </ul>
1 - 31	Unused	N/A	0	Unused. Set to zeros on a read.

**IP Interrupt Enable (IP IER) and IP Status Registers (IP ISR)**

The IP Interrupt Enable Register (IP IER) and IP Interrupt Status Register (IP ISR), shown in Figure 6, provide a bit per interrupt. These registers are meaningful only if the parameter C\_INTERRUPT\_PRESENT is 1.

The interrupt enable bits in the IP Interrupt Enable Register have a one-to-one correspondence with the status bits in the IP Interrupt Status Register. The interrupt events are registered in the IP Interrupt Status Register by the OPB clock and therefore the change in the input port must be stable for at least one clock period wide to guarantee interrupt capture. Each IP ISR register bit can be set or cleared via software by the Toggle-On-Write implementation.

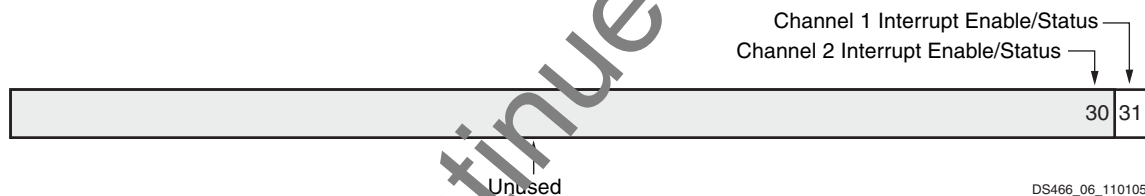


Figure 6: IP Interrupt Enable and IP Interrupt Status Register

The bit definition for IP Interrupt Enable Register and IP Interrupt Status Register are given in Table 10 and Table 11 respectively.

Table 10: IP Interrupt Enable Register Description

Bit(s)	Name	Core Access	Reset Value	Description
0 - 29	Unused	N/A	0	Unused. Set to zeros on a read.
30	Channel 2 Interrupt Enable	Read/Write	0	Enable Channel 2 Interrupt <ul style="list-style-type: none"> <li>• 1 = Enabled</li> <li>• 0 = Disabled (masked)</li> </ul>
31	Channel 1 Interrupt Enable	Read/Write	0	Enable Channel 1 Interrupt <ul style="list-style-type: none"> <li>• 1 = Enabled</li> <li>• 0 = Disabled (masked)</li> </ul>

Table 11: IP Interrupt Status Register Description

Bit(s)	Name	Core Access	Reset Value	Description
0 - 29	Unused	N/A	0	Unused. Set to zeros on a read.
30	Channel 2 Interrupt Status	Read/TOW <sup>(1)</sup>	0	Channel 2 Interrupt Status <ul style="list-style-type: none"> <li>• 1 = Channel 2 input interrupt</li> <li>• 0 = No Channel 2 input interrupt</li> </ul>
31	Channel 1 Interrupt Status	Read/TOW <sup>(1)</sup>	0	Channel 1 Interrupt Status <ul style="list-style-type: none"> <li>• 1 = Channel 1 input interrupt</li> <li>• 0 = No Channel 1 input interrupt</li> </ul>

**Notes:**

1. Toggle-On-Write (TOW) access toggles the status of the bit when a value of 1 is written to the corresponding bit

## OPB GPIO Operation

The OPB GPIO can be configured as either a single or a dual channel device using the C\_IS\_DUAL generic. When both channels are enabled with C\_IS\_DUAL, each channel has the same size, as defined by the C\_GPIO\_WIDTH size.

The OPB GPIO has a 3-state I/O capability as well as independent inputs and outputs. This allows connection to both bi-directional and conventional signals. The GPIOx\_TRI register is used to enable the 3-state buffers which enable/3-state outputs on the GPIOx\_IO pins. The GPIOx\_TRI register is also driven out of the dedicated GPIOx\_t\_out output pins. Each of the GPIOx\_IO pin has a corresponding bit in the GPIOx\_TRI register.

To configure a port as output, the corresponding bit in GPIOx\_TRI register is written as 0. A subsequent write to the GPIOx\_DATA register causes the data written to appear on the GPIOx\_IO pins for I/Os that are configured as outputs. Data written to the GPIOx\_DATA register is also driven out of the GPIOx\_d\_out output-only pin, for non 3-state connections.

To configure a port as input, the corresponding bit in GPIOx\_TRI register is written as 1 thereby disabling the 3-state buffers. An input port can be configured under control of the C\_IS\_BIDIRx generics, to take its external input from the bi-directional (GPIOx\_IO) pins or the dedicated input only (GPIOx\_in) pins. If C\_IS\_BIDIRx is 1, the source for inputs is the GPIOx\_IO ports. If C\_IS\_BIDIRx is 0, the source for inputs is the GPIOx\_in ports.

If only inputs are required for a channel, the C\_ALL\_INPUTSx parameter can be set to true. As a result, the GPIOx\_TRI register and the read multiplexer are removed from the logic to reduce resource utilization. The related I/O pins (GPIOx\_IO, GPIOx\_d\_out and the GPIOx\_t\_out) will be retained in the design and will be driven with the default value.

The GPIOx\_DATA and the GPIOx\_TRI registers are reset to the values set on the generics C\_DOUT\_DEFAULTx and C\_TRI\_DEFAULTx at configuration time.

If the C\_INTERRUPT\_PRESENT generic is 1, a transition on any input will cause an interrupt. There are independent interrupt enable and interrupt status bit for each channel if dual channel operation is used.

## User Application Hints

The user may find the following steps helpful in accessing the OPB GPIO core:

### For input ports when the channel is configured for interrupt

1. Configure the port as input by writing the corresponding bit in GPIOx\_TRI register with the value of 1.
2. Enable the channel interrupt by setting the corresponding bit in the IP Interrupt Enable Register; Also enable the OPB Global Interrupt Enable Register by setting bit 0.
3. When an interrupt is received, read the corresponding bit in GPIOx\_DATA register. Toggle the status in the IP Interrupt Status Register by writing the corresponding bit with the value of "1".

### For input ports when the channel is not configured for interrupt

1. Configure the port as input by writing the corresponding bit in GPIOx\_TRI register with the value of "1".
2. Read the corresponding bit in GPIOx\_DATA register.

### For output ports

1. Configure the port as output by writing the corresponding bit in GPIOx\_TRI register with a value of "0".
2. Write the corresponding bit in GPIOx\_DATA register.

## Design Implementation

### Target Technology

The intended target technology is the Virtex-4 Pro FPGA.

### Device Utilization and Performance Benchmarks

As the OPB GPIO module will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates. When the OPB GPIO module is combined with other designs, the utilization of FPGA resources and timing of the OPB GPIO design will vary.

The OPB GPIO benchmark and the resource utilization for various parameter combinations are detailed in [Table 12](#). Resource utilization is measured with Virtex-II Pro as the target device.

Table 12: OPB GPIO Performance and Resource Utilization Benchmarks

Parameter Values				Device Resources			f <sub>MAX</sub> (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slices	Slice Flip-Flops	4-input LUTs	
0	0	1	X <sup>1</sup>	89	149	23	138.2
0	0	0	X <sup>1</sup>	159	279	63	134.1
0	1	1	X <sup>1</sup>	140	206	98	124.3
0	1	0	X <sup>1</sup>	205	329	138	111.1
1	0	1	1	109	185	57	139.9
1	0	0	1	192	313	97	127.2
1	0	1	0	192	313	97	131.5
1	0	0	0	229	377	107	127.8
1	1	1	1	113	273	174	119.5
1	1	0	1	266	397	220	117.4
1	1	1	0	264	395	215	110.9
1	1	0	0	301	459	225	116.3

**Notes:**

1. When C\_IS\_DUAL = 0, the core is configured for single channel and hence the parameter C\_ALL\_INPUTS\_2 has no effect
2. These benchmark designs contain only the OPB GPIO module without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user condition
3. C\_GPIO\_WIDTH is set to 32 for all cases

## Reference Documents

The following documents contain useful information about the OPB GPIO reference design:

- *On-Chip Peripheral Bus Architectural Specifications, Version 2.0*, IBM
- *DS404 OPB IPIF Product Specification (v3.01.a)*, Xilinx

## Revision History

Date	Version	Revision
12/1/03	1.0	Initial Xilinx release.
11/1/05	1.1	Converted to new DS template; updated figures to graphic standards; reformatted tables; made minor content edits.
12/1/05	1.2	Added Spartan-3E support.