

## Introduction

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

## Features

- Configurable as single or dual GPIO channel(s)
- PLB v34 bus interface with byte-enable support
- Each GPIO bit dynamically programmable as input or output
- Number of GPIO bits configurable from 1-32 bits
- Can be configured as inputs-only to reduce resource utilization
- Ports for both three-state and non-three-state connections
- Optional Interrupt request generation
- Independent reset values for each bit of all registers

| LogiCORE™ Facts                  |                              |        |
|----------------------------------|------------------------------|--------|
| Core Specifics                   |                              |        |
| Supported Device Family          | Virtex™-II Pro, Virtex-4     |        |
| Version of Core                  | plb_gpio                     | v1.00b |
| Resources Used (GPIO_Width = 32) |                              |        |
|                                  | Min                          | Max    |
| Slices                           | 100                          | 281    |
| LUTs                             | 46                           | 205    |
| FFs                              | 152                          | 414    |
| Block RAMs                       |                              |        |
| 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      | 5.1i or later                |        |
| Verification                     | N/A                          |        |
| Simulation                       | ModelSim SE/EE 5.6e 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 GPIO consists of registers and multiplexers for writing and reading register contents as shown in Figure 1.

## GPIO Organization

Figure 1 shows a dual channel implementation of the PLB GPIO. The tristate buffers in the figure are not actually part of the core. The tristate buffers are added in the synthesis process, usually automatically with an Add I/Os option. The interrupt enable register, global interrupt enable register, and interrupt status register are accessed through PLB\_wrDBUS and PLB\_rdDBUS.

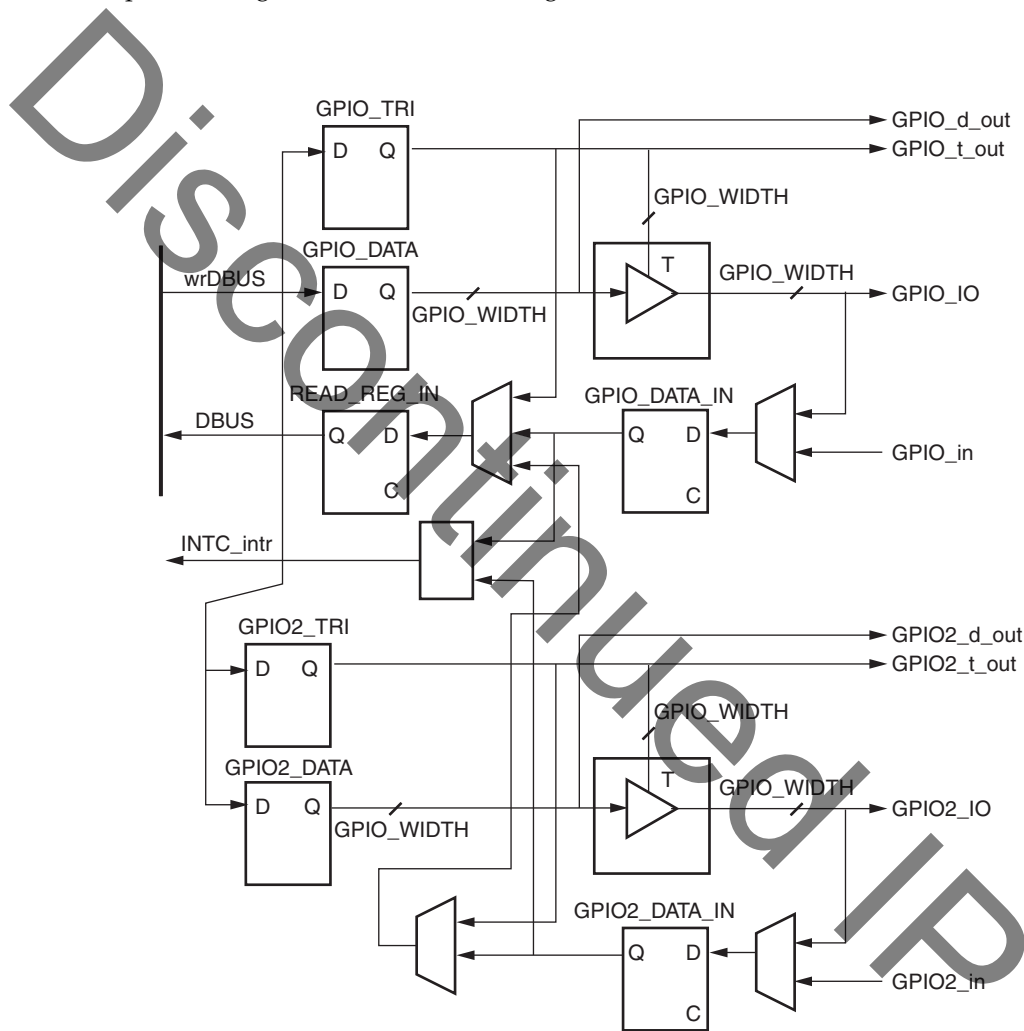


Figure 1: GPIO Block Diagram

## PLB GPIO I/O Signals

The I/O signals for this reference design are shown in [Figure 1](#) and described in [Table 1](#).

Table 1: PLB Core SSP1 Reference Design I/O Signals

| Signal Name                       | Interface | I/O | Description                                      |
|-----------------------------------|-----------|-----|--|
| PLB_abort                         | PLB       | I   | PLB abort bus request indicator                  |
| PLB_ABus(0:C_PLB_AWIDTH-1)        | PLB       | I   | PLB address bus                                  |
| PLB_BE(0:(C_PLD_DWIDTH / 8) -1)   | PLB       | I   | PLB byte enables                                 |
| PLB_busLock                       | PLB       | I   | PLB bus lock                                     |
| PLB_compress                      | PLB       | I   | PLB compressed data transfer indicator           |
| PLB_guarded                       | PLB       | I   | PLB guarded transfer indicator                   |
| PLB_lockErr                       | PLB       | I   | PLB lock error indicator                         |
| PLB_masterID(0:C_PLB_MID_WIDTH-1) | PLB       | I   | PLB current master indicator                     |
| PLB_ordered                       | PLB       | I   | PLB synchronize transfer indicator               |
| PLB_PAValiid                      | PLB       | I   | PLB primary address valid indicator              |
| PLB_rdBurst                       | PLB       | I   | PLB burst read transfer indicator                |
| PLB_abort                         | PLB       | I   | PLB abort bus request indicator                  |
| PLB_ABus(0:C_PLB_AWIDTH-1)        | PLB       | I   | PLB address bus                                  |
| PLB_BE(0:(C_PLD_DWIDTH / 8) -1)   | PLB       | I   | PLB byte enables                                 |
| PLB_busLock                       | PLB       | I   | PLB bus lock                                     |
| PLB_compress                      | PLB       | I   | PLB compressed data transfer indicator           |
| PLB_guarded                       | PLB       | I   | PLB guarded transfer indicator                   |
| PLB_lockErr                       | PLB       | I   | PLB lock error indicator                         |
| PLB_masterID(0:C_PLB_MID_WIDTH-1) | PLB       | I   | PLB current master indicator                     |
| PLB_ordered                       | PLB       | I   | PLB synchronize transfer indicator               |
| PLB_PAValiid                      | PLB       | I   | PLB primary address valid indicator              |
| PLB_rdBurst                       | PLB       | I   | PLB burst read transfer indicator                |
| PLB_rdPrim                        | PLB       | I   | PLB secondary to primary read request indicator  |
| PLB_RNW                           | PLB       | I   | PLB read not write                               |
| PLB_SAValiid                      | PLB       | I   | PLB secondary address valid indicator            |
| PLB_size(0:3)                     | PLB       | I   | PLB transfer size                                |
| PLB_type(0:2)                     | PLB       | I   | PLB transfer type                                |
| PLB_wrBurst                       | PLB       | I   | PLB burst write transfer indicator               |
| PLB_wrDBus(0:C_PLB_DWIDTH -1)     | PLB       | I   | PLB write data bus                               |
| PLB_wrPrim                        | PLB       | I   | PLB secondary to primary write request indicator |

Table 1: PLB Core SSP1 Reference Design I/O Signals (Contd)

| Signal Name                      | Interface      | I/O | Description                             |
|----------------------------------|----------------|-----|---|
| PLB_MSize(0:1)                   | PLB            | I   | PLB master data bus size                |
| Sl_addrAck                       | PLB            | O   | Slave address acknowledge               |
| Sl_MBusy(0:C_PLB_NUM_MASTERS-1)  | PLB            | O   | Slave busy indicator                    |
| Sl_MErr(0:C_PLB_NUM_MASTERS-1)   | PLB            | O   | Slave error indicator                   |
| Sl_rdBTerm                       | PLB            | O   | Slave terminate read burst transfer     |
| Sl_rdComp                        | PLB            | O   | Slave read transfer complete indicator  |
| Sl_rdDAck                        | PLB            | O   | Slave read data acknowledge             |
| Sl_rdBUS(0:C_PLB_DWIDTH -1)      | PLB            | O   | Slave read bus                          |
| Sl_rdWdAddr(0:3)                 | PLB            | O   | Slave read word address                 |
| Sl_rearbitrate                   | PLB            | O   | Slave rearbitrate bus indicator         |
| Sl_wait                          | PLB            | O   | Slave wait indicator                    |
| Sl_wrBTerm                       | PLB            | O   | Slave terminate write burst transfer    |
| Sl_wrComp                        | PLB            | O   | Slave write transfer complete indicator |
| Sl_wrDAck                        | PLB            | O   | Slave write data acknowledge            |
| Sl_SSize(0:1)                    | PLB            | O   | Slave data bus size                     |
| PLB_pendReq                      | PLB            | I   | PLB pending bus request indicator       |
| PLB_pendPri(0:1)                 | PLB            | I   | PLB pending request priority            |
| PLB_reqPri(0:1)                  | PLB            | I   | PLB current request priority            |
| PLB_Clk                          | System         | I   | PLB Clock                               |
| PLB_Rst                          | System         | I   | PLB Reset                               |
| INTC_irpt                        | IPIF Interrupt | O   | PLBPLB GPIO Interrupt                   |
| GPIO_IO(0 to C_GPIO_WIDTH-1)     | GPIO           | I/O | General purpose I/O                     |
| GPIO_in                          | GPIO           | I   | General purpose input                   |
| GPIO_d_out(0 to C_GPIO_WIDTH-1)  | GPIO           | O   | GPIO Data register output               |
| GPIO_t_out(0 to C_GPIO_WIDTH-1)  | GPIO           | O   | GPIO Tri register output                |
| GPIO2_IO(0 to C_GPIO_WIDTH-1)    | GPIO           | I/O | General purpose I/O                     |
| GPIO2_in(0 to C_GPIO_WIDTH-1)    | GPIO           | I   | General purpose input                   |
| GPIO2_d_out(0 to C_GPIO_WIDTH-1) | GPIO           | O   | GPIO2_Data register output              |
| GPIO2_t_out(0 to C_GPIO_WIDTH-1) | GPIO           | O   | GPIO2_Tri register output               |

## PLB GPIO Design Parameters

Certain features can be parameterized in the PLBGPIO design. Some of these parameters control the interface to the PLBbus, others provide information to the GPIO logic. The PLB GPIO parameters are shown in [Table 2](#).

## Parameterization

The following characteristics of the GPIO can be parameterized:

- Base address for the GPIO registers
- Width of PLB data bus attached to the peripheral
- Width of PLB address bus attached to the peripheral
- Number of GPIO bits
- I/Os are input-only or programmable as input or output
- Single or dual channel
- Supports three-state and/or single ended I/Os
- Bitwise control of register reset values

Interrupt generation

Table 2: PLBGPIO Design Parameters

| Generic     | Feature / Description                    | Parameter Name            | Allowable Values                              | Default Value       | VHDL Type        |
|-------------|--|---------------------------|---|---------------------|------------------|
| <b>GPIO</b> |  |                           |   |                     |                  |
| G1          | GPIO Base Address                        | C_BASEADDR                | Valid PLB Address <sup>(1)</sup>              | None <sup>(2)</sup> | std_logic_vector |
| G2          | GPIO High Address                        | C_HIGHADDR                | Valid PLBAddress <sup>(1)</sup>               | None <sup>(2)</sup> | std_logic_vector |
| G3          | Target FPGA Family                       | C_FAMILY                  | Any FPGA family that is supported by the core | "virtex2"           | string           |
| G4          | GPIO Data Bus Width                      | C_GPIO_Width              | 1-32  | 32                  | integer          |
| G5          | GPIO Interrupt                           | C_INTERRUPT_PRESSENT      | 0/1   | 0                   | integer          |
| G6          | Inputs Only                              | C_ALL_INPUTS              | 0/1   | 0                   | integer          |
| G7          | Select GPIO_IO as input source           | C_IS_BIDIR <sup>(3)</sup> | 0/1   | 0                   | 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                    | "11111111"          | std_logic_vector |
| G10         | Use dual channel GPIO                    | 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   | 0                   | integer          |

Table 2: PLBGPIO Design Parameters (Contd)

| Generic  | Feature / Description  | Parameter Name    | Allowable Values                              | Default Value | VHDL Type        |
|--|------------------------|-------------------|---|---------------|------------------|
| 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                    | "11111111"    | std_logic_vector |
| <b>PLB Interface</b>   |                        |                   |   |               |                  |
| G15  | PLB Address Width      | C_PLB_AWIDTH      | 32  | 32            | integer          |
| G16  | PLB Data Width         | C_PLB_DWIDTH      | 64  | 64            | integer          |
| G17  | Number of Masters      | C_PLB_NUM_MASTERS | 1 - 8   | 8             | integer          |
| G18  | Width of Master ID Bus | C_PLB_MID_WIDTH   | roundup(log <sub>2</sub> (C_PLB_NUM_MASTERS)) | 2             | integer          |
| <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>The range specified by C_BASEADDR and C_HIGHADDR must comprise a complete, contiguous power of two range such that range = 2<sup>n</sup>, and the n least significant bits of C_BASEADDR must be zero. This range needs to encompass the address range required by the GPIO. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1.</li> <li>No default value is specified for C_BASEADDR or C_HIGHADDR to insure that the actual value is set, i.e. if the value is not set, a compiler error is generated.</li> </ol> <p>C_IS_BIDIR is used to select between driving internal logic and a pad.</p> |                        |                   |   |               |                  |

### Allowable Parameter Combinations

The address range specified by C\_BASEADDR and C\_HIGHADDR must comprise a complete, contiguous power of two range such that range = 2<sup>n</sup>, and 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 GPIO. In this reference design, the example GPIO contains 1-32-bit registers, therefore the minimum range specified by C\_BASEADDR and C\_HIGHADDR is 0x04. See the PLB IPIF Interrupt documentation for more information.

Table 3: Parameter-Port Dependencies

| Name                     | Affects                                | Depends  | Relationship Description  |
|--------------------------|--|--|---|
| <b>Design Parameters</b> |  |  |   |
| C_PLB_NUM_MASTERS        | SI_MBusy<br>SI_MErr<br>C_PLB_MID_WIDTH | 0 to C_PLB_NUM_MASTERS -1<br>0 to C_PLB_NUM_MASTERS -1                 | Number of Signals required based on the number of PLB Masters required                                |
| C_PLB_DWIDTH             | PLB_BE<br>PLB_wrDBus<br>SI_rdDBus      | 0 to C_PLB_DWIDTH/8 -1<br>0 to C_PLB_DWIDTH -1<br>0 to C_PLB_DWIDTH -1 | Number of Byte Enables Decoded<br>Width of the PLB write Data Bus<br>Width of the Slave read Data Bus |
| C_PLB_AWIDTH             | PLB_ABUS                               | 0 to C_PLB_AWIDTH -1   | Width of the PLB Address Bus  |

Table 3: Parameter-Port Dependencies (Contd)

| Name            | Affects  | Depends                | Relationship Description  |
|-----------------|--|------------------------|---|
| C_PLB_MID_WIDTH | PLB_masterID                                       | 0 to C_PLB_MID_WIDTH-1 | Size of the masterID is decoded as the function $\log_2 \times (\text{C\_PLB\_NUM\_MASTERS})$                         |
| C_GPIO_WIDTH    | GPIO_Data<br>GPIO2_Data<br>GPIO_Tri<br>GPIO2Tri    | 1 to C_GPIO_WIDTH      | Number of GPIO_in, GPIO_IO, GPIO_d_out, GPIO_t_out, GPIO2_in, GPIO2_IO, GPIO2_d_out, GPIO2_t_out ports                |
| C_IS_DUAL       | GPIO2_in<br>GPIO2_IO<br>GPIO2_d_out<br>GPIO2_t_out | 1 to C_GPIO_WIDTH      | When C_IS_DUAL is 1, Channel 2 is created. The actual ports created is controlled by C_ALL_INPUTS_2 and C_IS_BIDIR_2. |
| C_ALL_INPUTS    | GPIO_IO<br>GPIO_d_out<br>GPIO_t_out                | 1 to C_GPIO_WIDTH      | Eliminates GPIO_IO, GPIO_d_out, GPIO_t_out ports  |
| C_ALL_INPUTS_2  | GPIO2_IO<br>GPIO2_d_out<br>GPIO2_t_out             | 1 to C_GPIO_WIDTH      | Eliminates GPIO2_IO, GPIO2_d_out, GPIO2_t_out ports   |
| C_IS_BIDIR      | GPIO_IO<br>GPIO_in                                 | 1 to C_GPIO_WIDTH      | GPIO_IO is used for input rather than GPIO_in   |
| C_IS_BIDIR_2    | GPIO2_IO<br>GPIO2_in                               | 1 to C_GPIO_WIDTH      | GPIO2_IO is used for input rather than GPIO2_in   |

## PLB GPIO Registers

There are seven PLB registers in the PLB GPIO design as shown in Table 4. The inclusion of the IP Interrupt Source Control service includes an Interrupt Status Register (ISR), and Interrupt Enable Register (IER), and a Global Interrupt Enable Register (GIE). These IP ISC registers provide read and toggle on write access.

Table 4: PLB Registers

| Register Name | Description                      | PLB Address        | Access     |
|---------------|----------------------------------|--------------------|------------|
| GIE           | Global Interrupt Enable Register | C_BASEADDR + 0x11C | Read/Write |
| IP ISR        | IP Interrupt Status Register     | C_BASEADDR + 0x120 | Read/TOW   |
| IP IER        | IP Interrupt Enable Register     | C_BASEADDR + 0x128 | Read/TOW   |
| GPIO_Data     | GPIO Data register               | C_BASEADDR + 0x00  | Read/Write |
| GPIO_TRI      | GPIO Tri register                | C_BASEADDR + 0x04  | Read/Write |
| GPIO2_Data    | Channel 2 GPIO Data register     | C_BASEADDR + 0x08  | Read/Write |
| GPIO2_TRI     | Channel 2 GPIO Tri register      | C_BASEADDR + 0x0C  | Read/Write |

### PLB Global Interrupt Enable (GIE) Register

The PLB Device Global Interrupt Enable Register provides the final enable/disable for the interrupt output to the processor. This is a single bit read/write register as shown in. NOTE: This bit must be set to generate interrupts, even if the interrupts are enabled in the IP IER.

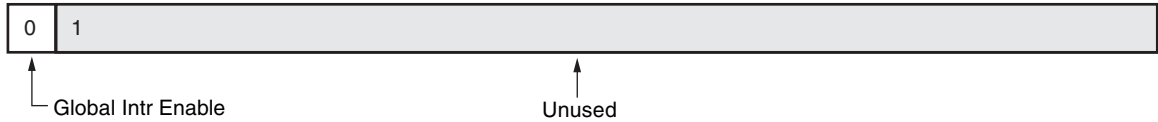


Figure 2: PLB Global Interrupt Enable(GIE) Register

Table 5: PLB 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 ISC 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.   |

### PLB Interrupt Enable Register (IER) and Interrupt Status Register (ISR)

The PLB Interrupt Enable Register (IER) and Interrupt Status Register (ISR) provide a bit per interrupt. The interrupt enables have a one-to-one correspondence with the interrupt bits in the status register. The interrupt events are registered in the ISR by the PLB clock and must be at least one PLB clock period wide to guarantee capture. Each ISR register bit can be set or cleared via software by the Toggle-On-Write implementation.

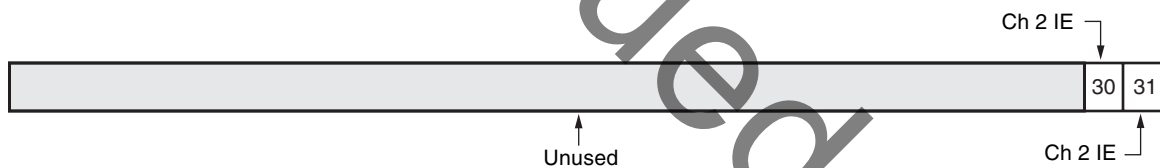


Figure 3: PLB Interrupt Enable Register (IER) and Interrupt Status Register (ISR)

Table 6: PLB 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     | Ch 2 Interrupt | Read/TOW    | 0           | <b>Channel 2 Interrupt</b> <ul style="list-style-type: none"> <li>'1' = Channel 2 input interrupt</li> <li>'0' = No Channel 2 input interrupt</li> </ul>  |
| 31     | Ch 1 Interrupt | Read/TOW    | 0           | <b>Channel 1 Interrupt.</b> <ul style="list-style-type: none"> <li>'1' = Channel 1 input interrupt</li> <li>'0' = No Channel 1 input interrupt</li> </ul> |



Table 7: PLB 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     | Ch 2   | Read/Write  | 0           | <b>Enable Channel 2 Interrupt.</b> <ul style="list-style-type: none"> <li>'1' = Enabled</li> <li>'0' = Disabled (masked)</li> </ul> |
| 31     | Ch 1   | Read/Write  | 0           | <b>Enable Channel 1 Interrupt.</b> <ul style="list-style-type: none"> <li>'1' = Enabled</li> <li>'0' = Disabled (masked)</li> </ul> |

### GPIOx Data Register (GPIO\_DATA)

GPIOx Three-state Register (GPIOx\_TRI)



Figure 4: GPIOx\_DATA

Table 8: GPIOx\_DATA Register

| Bits | Name       | Core Access | Description  | Reset Value                       |
|------|------------|-------------|--|-----------------------------------|
| 0:31 | GPIOx_DATA | Read/Write  | GPIOx Data<br>For I/O programmed as inputs:<br><b>R:</b> reads value on input pin<br><b>W:</b> no effect<br>For I/O programmed as outputs:<br><b>R:</b> reads value in GPIO data register<br><b>W:</b> writes value to GPIO data register and output pin | C_DOUT_DEFAULT<br>C_DOUT_DEFAULT2 |



Figure 5: GPIOx\_TRI

Table 9: GPIOx\_TRI Register

| Bits | Name      | Core Access | Description   | Reset Value                     |
|------|-----------|-------------|---|---------------------------------|
| 0:31 | GPIOx_TRI | Read/Write  | GPIO Three-state Control. Each I/O pin of the GPIO is individually programmable as an input or output. For each bit:<br>0 I/O pin configured as output<br>1 I/O pin configured as input | C_TRI_DEFAULT<br>C_TRI_DEFAULT2 |

## GPIO Operation

The 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. There are a number of generics which allow independent configuration of the two channels.

A write to the GPIO\_DATA register causes the written data to appear on the GPIO\_IO ports for I/Os that are configured as outputs. The GPIO\_TRI register is used to enable the tristate buffers which enable/tristate outputs on the GPIO\_IO ports. Each bit of the GPIO\_IO port has a corresponding bit in the GPIO\_TRI register. If the GPIO is used only to generate outputs, the GPIO\_TRIx register is written as 0. If only inputs are required, the C\_ALL\_INPUTS parameter can be set to true. As a result, the GPIO\_TRI register and the read multiplexer are removed from the logic to reduce resource utilization.

The GPIO has both a three-state I/O capability as well as independent inputs and outputs. This allows connection to both bi-directional and conventional signals. Data written to the GPIO\_DATA and the GPIO\_TRI registers is also driven to the GPIOx\_d\_out and GPIOx\_t\_out pins respectively.

Under control of the C\_IS\_BIDIRx generics, the GPIO can be configured to take its external input from the bi-directional input pin or the GPIOx\_In pins. If C\_IS\_BIDIRx is 1, the source for inputs is the GPIO\_IOx ports. If C\_IS\_BIDIRx is 0, the source for inputs is the GPIO\_Inx ports.

The GPIO\_DATAx and the GPIO\_TRIx 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 Registers and Interrupt Status Registers for each channel if dual channel operation is used. The Toggle on Write mechanism for the Interrupt Status Register avoids the requirement on the User Interrupt Service routine to perform Read/Modify/Write operation to clear a single bit within the register. Read Modify/Write operations can lead to inadvertant clearing of interrupts captured in the time period between the Read and Write operations.

Table 10: PLB GPIO Resource Utilization (Virtex-II Pro)

| Parameter Values |              |              |                     | Device Resources |                  |              |
|------------------|--------------|--------------|---------------------|------------------|------------------|--------------|
| C_IS_DUAL        | C_ALL_INPUTS | C_GPIO_WIDTH | C_INTERRUPT_PRESENT | Slices           | Slice Flip Flops | 4-input LUTs |
| Yes              | Yes          | 32           | Yes                 | 171              | 227              | 170          |
| Yes              | Yes          | 32           | No                  | 119              | 188              | 79           |
| Yes              | No           | 32           | Yes                 | 281              | 414              | 205          |
| Yes              | No           | 32           | No                  | 232              | 380              | 146          |
| No               | No           | 32           | Yes                 | 207              | 316              | 161          |

Table 10: PLB GPIO Resource Utilization (Virtex-II Pro)

| Parameter Values |              |              |                     | Device Resources |                  |              |
|------------------|--------------|--------------|---------------------|------------------|------------------|--------------|
| C_IS_DUAL        | C_ALL_INPUTS | C_GPIO_WIDTH | C_INTERRUPT_PRESENT | Slices           | Slice Flip Flops | 4-input LUTs |
| No               | No           | 32           | No                  | 166              | 282              | 111          |
| No               | Yes          | 32           | Yes                 | 145              | 193              | 95           |
| No               | Yes          | 32           | No                  | 100              | 152              | 46           |

### Reference Documents

The following documents contain reference information important to understanding the PLB SSP1 Core Reference design:

- 64-Bit Processor Local Bus, Architectural Specifications, Version 3.4 IBM.
- User Core Templates Reference Guide, Xilinx
- OPB IPIF Interrupt, Xilinx.

### Revision History

The following table shows the revision history for this document.

| Date     | Version | Revision  |
|----------|---------|---|
| 11/25/03 | 1.0     | Initial Xilinx release.   |
| 03/26/04 | 2.0     | Hardware revision to 1.00b  |
| 05/05/04 | 2.2     | Updated to design parameters, parameter-port dependencies, figure 3   |
| 8/20/04  | 2.3     | Recreated and updated for Gmm; updated trademarks usage and supported device family listing.  |
| 9/7/04   | 2.3.1   | Incorporated CR191952; in Table 1, allowable values for the C_FAMILY was "any FPGA family". Incorporated CR191953; In Table 1, Default Value for C_PLB_NUM_MASTERS was 4. |
| 9/15/05  | 2.4     | Converted to new DS template; updated figures to Xilinx Graphic Standards.  |