XPS GeneralPurpose Input/Output(GPIO)(v1.00a)

Introduction

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

Features

- PLB interface is based on PLB v4.6 specification
- Configurable as single or dual GPIO channel(s)
- Number of GPIO bits configurable from 1 to 32 bits
- Each GPIO bit can be dynamically programmed 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	Spartan™-3E, Spartan-3, Spartan-3A, Spartan-3AN, Virtex™-4, and Virtex-5	
Version of Core	xps_gpio	v1.00a
Resources Used		
	Min	Max
Slices	Refer to the Table 12, Table 13 and Table 14	
LUTs		
FFs		
Block RAMs	N/A	
Special Features	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 & Application Notes	N/A	
Additional Items	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 9.1i or later	
Verification	ModelSim SE/EE 6.0c or later	
Simulation	ModelSim SE/EE 6.0c or later	
Synthesis	XST 9.1i or later	
Support		
Support provided by Xilinx, Inc.		

Functional Description

The XPS GPIO design provides a general purpose input/output interface to a Processor Local Bus (PLB). The XPS 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 XPS 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 XPS GPIO core is comprised of modules:

- PLB Interface Module
- Interrupt Control
- GPIO core

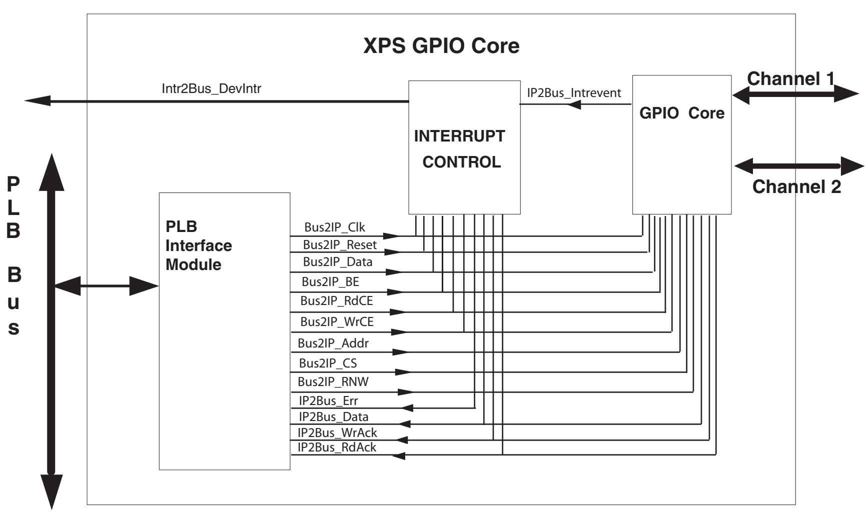

Figure 1: **XPS GPIO Block Diagram**

PLB Interface Module

PLB Interface module provides an interface between the GPIO core and the PLBV4.6 bus standard. The PLB Interface module implements the basic functionality of PLB slave operation and does the necessary protocol and timing translation between the PLB and the IPIC interfaces. The PLB Interface module allows only single beat transactions.

INTERRUPT CONTROL

Interrupt Controller provides interrupt capture support for the GPIO core. The Interrupt Controller is used to collect interrupts from the GPIO core, by which the GPIO core requests the attention of the microprocessor through the assertion of interrupt signals. The Interrupt Control module comes into the picture only when the C_INTERRUPT_PRESENT generic is set to 1.(For mor e information on the generics, refer Table 2)

GPIO Core

GPIO core provides an interface between the IPIC interface and the XPS GPIO channels. The GPIO core consists of registers and multiplexers for reading and writing the XPS 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 the 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. The control signals of the IPIC interface are not shown in Figure 2

.

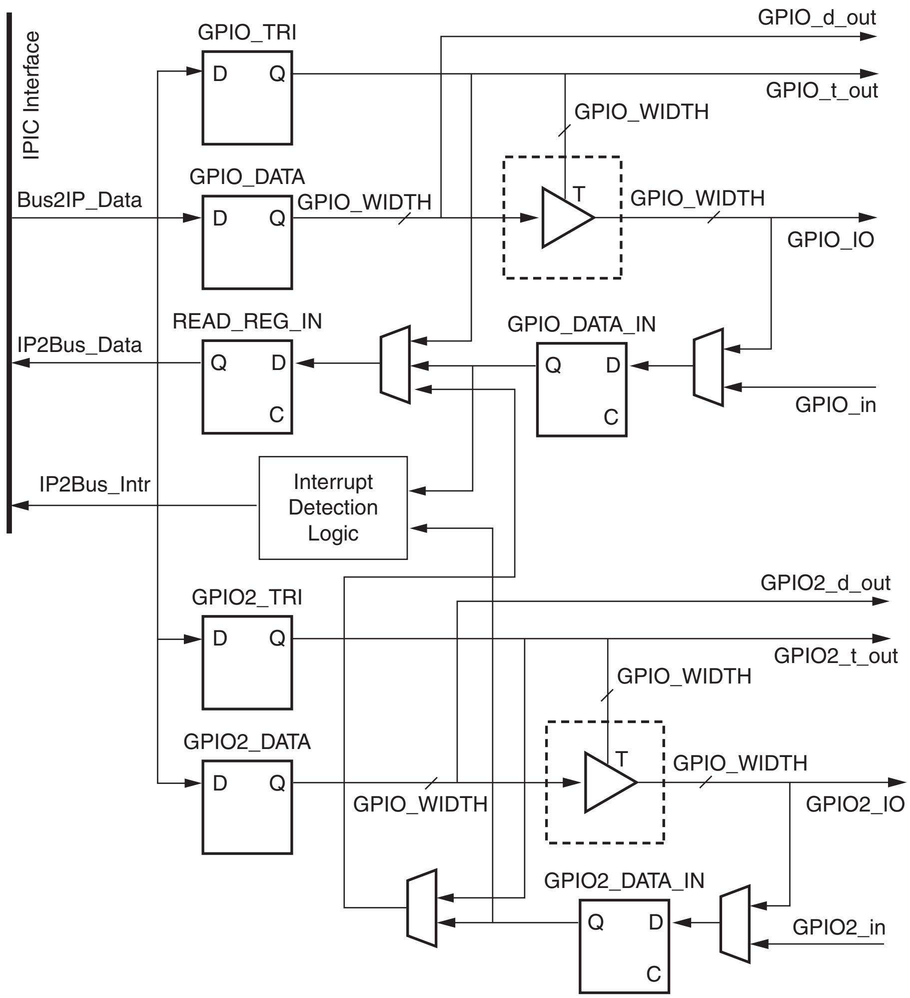

Figure 2: **GPIO_CORE Dual Channel Implementation**

XPS GPIO I/O Signals

The I/O signals for this reference design are listed in Table 1.

Table 1: **XPS GPIO I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	SPLB_Clk	PLB	I	-	PLB clock
P2	SPLB_Rst	PLB	I	-	PLB reset, active high
PLB Interface Signals					
P3	PLB_ABus[0:C_SPLB_AWIDTH-1]	PLB	I	-	PLB address bus

Table 1: **XPS GPIO I/O Signals *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P4	PLB_PAValid	PLB	I	-	PLB primary address valid
P5	PLB_MasterID[0:C_SPLB_MID_WIDTH-1]	PLB	I	-	PLB current master identifier
P6	PLB_RNW	PLB	I	-	PLB read not write
P7	PLB_BE[0:C_SPLB_DWIDTH/8 -1]	PLB	I	-	PLB byte enables
P8	PLB_Size[0:3]	PLB	I	-	PLB size of requested transfer
P9	PLB_type[0:2]	PLB	I	-	PLB transfer type
P10	PLB_wrDBus[0:C_SPLB_DWIDTH-1]	PLB	I	-	PLB write data bus
Unused PLB Interface Signals					
P11	PLB_UABus[0:C_SPLB_AWIDTH-1]	PLB	I	-	PLB Upper Address bits
P12	PLB_SAValid	PLB	I	-	PLB secondary address valid
P13	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P14	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P15	PLB_abort	PLB	I	-	PLB abort bus request
P16	PLB_busLock	PLB	I	-	PLB bus lock
P17	PLB_MSize[0:1]	PLB	I	-	PLB data bus width indicator
P18	PLB_TAttribute[0:15]	PLB	I	-	PLB transfer attribute
P19	PLB_lockerr	PLB	I	-	PLB lock error
P20	PLB_wrBurst	PLB	I	-	PLB burst write transfer
P21	PLB_rdBurst	PLB	I	-	PLB burst read transfer
P22	PLB_wrPendReq	PLB	I	-	PLB pending bus write request
P23	PLB_rdPendReq	PLB	I	-	PLB pending bus read request
P24	PLB_rdPendPri[0:1]	PLB	I	-	PLB pending write request priority
P25	PLB_wrPendPri[0:1]	PLB	I	-	PLB pending read request priority
P26	PLB_reqPri[0:1]	PLB	I	-	PLB current request priority
PLB Slave Interface Signals					
P27	Sl_addrack	PLB	O	0	Slave address acknowledge

Table 1: **XPS GPIO I/O Signals *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P28	Sl_Ssize[0:1]	PLB	O	0	Slave data bus size
P29	Sl_wait	PLB	O	0	Slave wait
P30	Sl_rearbitrate	PLB	O	0	Slave bus rearbitrate
P31	Sl_wrDaclk	PLB	O	0	Slave write data acknowledge
P32	Sl_wrComp	PLB	O	0	Slave write transfer complete
P33	Sl_rdBus[0:C_SPLB_DWIDTH-1]	PLB	O	0	Slave read data bus
P34	Sl_rdDack	PLB	O	0	Slave read data acknowledge
P35	Sl_rdComp	PLB	O	0	Slave read transfer complete
P36	Sl_Mbusy[0:C_SPLB_NUM_MASTERS-1]	PLB	O	0	Slave busy
P37	Sl_MWrErr[0:C_SPLB_NUM_MASTERS-1]	PLB	O	0	Slave write error
P38	Sl_MRdErr[0:C_SPLB_NUM_MASTERS-1]	PLB	O	0	Slave read error
Unused PLB Slave Interface Signals					
P39	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P40	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P41	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P42	Sl_MIRQ[0:C_SPLB_NUM_MASTERS-1]	PLB	O	0	Master interrupt request
GPIO Interface Signals					
P43	IP2INTC_Irpt	IPIF Interrupt	O	0	XPS GPIO Interrupt Active high signal
P44	GPIO_in[0 to C_GPIO_WIDTH-1]	GPIO	I	-	Channel 1 General purpose input
P45	GPIO_d_out[0 to C_GPIO_WIDTH-1]	GPIO	O	0[1]	Channel 1 data register (GPIO_DATA) output
P46	GPIO_t_out[0 to C_GPIO_WIDTH-1]	GPIO	O	1[2]	Channel 1 3-state control register (GPIO_TRI) output
P47	GPIO_IO[0 to C_GPIO_WIDTH-1]	GPIO	I/O	Z[3]	Channel 1 General purpose I/O
P48	GPIO2_in[0 to C_GPIO_WIDTH-1]	GPIO	I	-	Channel 2 General purpose input
P49	GPIO2_d_out[0 to C_GPIO_WIDTH-1]	GPIO	O	0[4]	Channel 2 data register (GPIO2_DATA) output

Table 1: **XPS GPIO I/O Signals *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P50	GPIO2_t_out[0 to C_GPIO_WIDTH-1]	GPIO	O	1[5]	Channel 2 3-state control register (GPIO2_TRI) output
P51	GPIO2_IO[0 to C_GPIO_WIDTH-1]	GPIO	I/O	Z[6]	Channel 2 General purpose I/O

Notes:

1. GPIO_d_out has an initial value of 0 only if the default value of C_DOUT_DEFAULT is used
2. GPIO_t_out has an initial value of 1 only if the default value of C_TRI_DEFAULT is used
3. GPIO remains at high impedance only if the default value of C_TRI_DEFAULT is used
4. GPIO2_d_out has an initial value of 0 only if the default value of C_DOUT_DEFAULT_2 is used
5. GPIO2_t_out has an initial value of 1 only if the default value of C_TRI_DEFAULT_2 is used
6. GPIO2 remains at high impedance only if the default value of C_TRI_DEFAULT_2 is used

XPS GPIO Design Parameters

To allow the designer to obtain a XPS GPIO core that is uniquely tailored for the designer's system, certain features can be parameterized. Some of these parameters control the interface to the PLB interface module while others provide information to minimize resource utilization. The features that can be parameterized in the XPS GPIO are shown in Table 2

Table 2: **XPS GPIO Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	spartan3e, spartan3, spartan3a, spartan3an, virtex4, virtex5	virtex5	string
PLB Parameters					
G2	PLB GPIO Base Address	C_BASEADDR	Valid Address	None [1]	std_logic_vector
G3	PLB GPIO High Address	C_HIGHADDR	Valid Address[1]	None[1]	std_logic_vector
G4	PLB address width	C_SPLB_AWIDTH	32	32	integer
G5	PLB data width	C_SPLB_DWIDTH	32, 64, 128	32	integer
G6	Selects point-to-point or shared PLB topology	C_SPLB_P2P	0 = Shared Bus Topology 1 = Point-to-Point Bus Topology	0	integer

Table 2: **XPS GPIO Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G7	PLB Master ID Bus Width	C_SPLB_MID _WIDTH	$\log_2$(C_SPLB_ NUM_MASTERS) with a minimum value of 1	1	integer
G8	Number of PLB Masters	C_SPLB_NUM _MASTERS	1-16	1	integer
G9	Width of the Slave Data Bus	C_SPLB_NATIVE _DWIDTH	32	32	integer
G10	Selects the transactions as being single beat or burst	C_SPLB_SUPPORT _BURSTS	0 = Supports only single beat transactions	0	integer
GPIO Parameters					
G11	GPIO Data Bus Width	C_GPIO_WIDTH	1-32	32	integer
G12	XPS GPIO Interrupt	C_INTERRUPT_ PRESENT	0 = Interrupt control module is not present 1 = Interrpt control module is present	0	integer
G13	Inputs Only	C_ALL_INPUTS	0 = IO ports of channel1 are configured as inputs or outputs 1 = All the IO ports of channel1 are configured as inputs only	0	integer
G14	Select GPIO_IO as input source	C_IS_BIDIR	0 = GPIO_IO port of channel1 is configured as output only 1 = GPIO_IO port of channel1 is configured as a bi-directional input/output port	1	integer
G15	GPIO_DATA reset value	C_DOUT_DEFAULT	Any valid std_logic_vector	00000000	std_logic_ vector
G16	GPIO_TRI reset value	C_TRI_DEFAULT	Any valid std_logic_vector	FFFFFFFF	std_logic_ vector
G17	Use dual channel	C_IS_DUAL	0 = Single channel is enabled 1 = Both the channels are enabled	0	integer

Table 2: **XPS GPIO Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G18	Channel 2 inputs only	C_ALL_INPUTS_2	0 = IO ports of channel2 are configured as inputs or outputs 1 = All the IO ports of channel2 are configured as inputs only	0	integer
G19	Channel 2 select GPIO_IO as input source	C_IS_BIDIR_2	0 = GPIO_IO port of channel2 is configured as output only 1 = GPIO_IO port of channel2 is configured as a bi-directional input/output port	1	integer
G20	GPIO2_DATA reset value	C_DOUT_DEFAULT_2	Any valid std_logic_vector	00000000	std_logic_vector
G21	GPIO2_TRI reset value	C_TRI_DEFAULT_2	Any valid std_logic_vector	FFFFFFFF	std_logic_vector

Notes:

1. The range specified by C_BASEADDR and C_HIGHADDR must be sized and aligned to some power of 2, 2^n . Then, the n least significant bits of C_BASEADDR is zero. This range needs to encompass the addresses needed by the XPS GPIO registers

Allowable Parameter Combinations

The range specified by C_BASEADDR and C_HIGHADDR must encompass the memory space required by the XPS 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_HIGHADDR in order to enforce that the user configures these parameters with actual values. If actual values are not set for C_BASEADDR and C_HIGHADDR, a elaboration error will be generated.

Parameter - Port Dependencies

The width of the XPS 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 XPS GPIO design parameters and the I/O ports are shown in Table 3.

Table 3: **Parameter-Port Dependencies**

Generic or Port	Parameter	Affects	Depends	Relationship Description
Design Parameters				
G4	C_SPLB_AWDITH	P3	-	Width of the PLB address bus
G5	C_SPLB_DWIDTH	P7, P10, P33	-	Number of byte enables decoded Width of the PLB data bus Width of the slave read data bus
G7	C_SPLB_MID_DWIDTH	P5	-	Width of Master ID Bus
G8	C_SPLB_NUM_MASTERS	P36,P37, P38	-	The number of Master Devices connected to PLBv46 bus
G11	C_GPIO_WIDTH	P44, P45, P46, P47, P48, P49, P50, P51	-	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
G13	C_ALL_INPUTS	P45, P46, P47	-	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
G14	C_IS_BIDIR	P44, P47	-	GPIO_IO is used for input rather than GPIO_in
G17	C_IS_DUAL	P48, P49, P50, P51	-	When C_IS_DUAL is 1, channel 2 is created
G18	C_ALL_INPUTS_2	P49, P50, P51	-	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
G19	C_IS_BIDIR_2	P48, P51	-	GPIO2_IO is used for input rather than GPIO2_in
I/O Signals				
P3	PLB_ABus	-	G11	Width varies with the width of the PLB Address Bus
P5	PLB_MasterID	-	G7	Width varies with the MID width
P7	PLB_BE	-	G5	Width varies with the width of the PLB Data Bus

Table 3: **Parameter-Port Dependencies *(Contd)***

Generic or Port	Parameter	Affects	Depends	Relationship Description
P10	PLB_wrDBus	-	G5	Width varies with the PLB Data Bus
P36	Sl_MBusy	-	G8	Width varies with the number of marsters
P37	Sl_MWrErr	-	G8	Width varies with the number of masters
P38	Sl_MRdErr	-	G8	Width varies with the number of masters
P44	GPIO_in	-	G11, G14	Width depends on the GPIO width and also depends on whether the birectional generic is enabled
P45	GPIO_d_out	-	G11, G13	Width depends on the GPIO width and also depends on whether all the ports act only as inputs
P46	GPIO_t_out	-	G11, G13	Width depends on the GPIO width and also depends on whether all the ports act only as inputs
P47	GPIO_IO	-	G11, G13, G14	Width depends on the GPIO width. Also depends on whether all the ports act only as inputs and whether the bidirectional generic is enabled
P48	GPIO2_in	-	G11, G17, G19	Width depends on the GPIO width. Also depends on whether the dual channel is enabled and the bidirectional generic is enabled
P49	GPIO2_d_out	-	G11, G17, G18	Width depends on the GPIO width. Also depends on whether the dual channel is enabled and whether all the ports act only as inputs
P50	GPIO2_t_out	-	G11, G17, G18	Width depends on the GPIO width. Also depends on whether the dual channel is enabled and whether all the ports act only as input
P51	GPIO2_IO	-	G11, G17, G18, G19	Width depends on the GPIO width. Also depends on whether the dual channel is enabled and whether all the ports act only as input. Its usage changes based on whether the bidirectional generic is enabled

XPS GPIO Registers

There are four internal registers in the XPS GPIO design as shown in Table 4. The memory map of the XPS GPIO design is determined by setting the C_BASEADDR parameter. The internal registers of the XPS GPIO are at a fixed offset from the base address. The XPS GPIO internal registers and their offset are listed in Table 4.

Table 4: **XPS GPIO Registers**

Register Name	Description	PLB Address	Access
GPIO_DATA	Channel 1 XPS GPIO Data Register	C_BASEADDR + `0x00`	Read/Write
GPIO_TRI	Channel 1 XPS GPIO 3-state Register	C_BASEADDR + `0x04`	Read/Write
GPIO2_DATA	Channel 2 XPS GPIO Data register	C_BASEADDR + `0x08`	Read/Write
GPIO2_TRI	Channel 2 XPS 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 unimplimented register has no effect. An attempt to read the unimplimented 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[2]	GPIO_TRI[2]	GPIO2_DATA[2]	GPIO2_TRI[2]
0	1	X [1]	Yes	No	No	No
0	0	X[1]	Yes	Yes	No	No
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:
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. 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

XPS GPIO Data Register (GPIOx_DATA)

XPS 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 XPS GPIO data register.

There are two XPS 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 XPS GPIO Data Register is shown in Figure 3 and Table 6 details its functionality

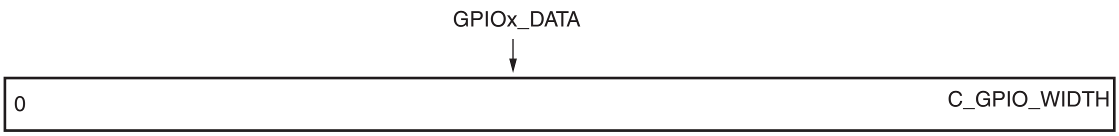

Figure 3: **XPS GPIO Data Register**

Table 6: **XPS GPIO Data Register Description**

Bits	Name	Core Access	Description	Reset Value
0:31	GPIOx_DATA	Read/Write[1]	XPS GPIO Data For I/O Programmed as inputs: **R**: Reads value on input pin **W**: No effect For I/O Programmed as outputs: **R**: Reads value on opuput pin **W**: Writes value to XPS GPIO data register and output pin	C_DOUT_DEFAULT C_DOUT_DEFAULT_2

Notes:
1. When the GPIO is configured as an output, the data register is write only, it cannot be read.

XPS GPIO 3-State Register (GPIOx_TRI)

The XPS 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 an input port. When a bit is reset, the corresponding I/O port is configured as an output port.

There are two XPS 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 inputs only(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 input only(C_ALL_INPUTS_2 = 0).

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

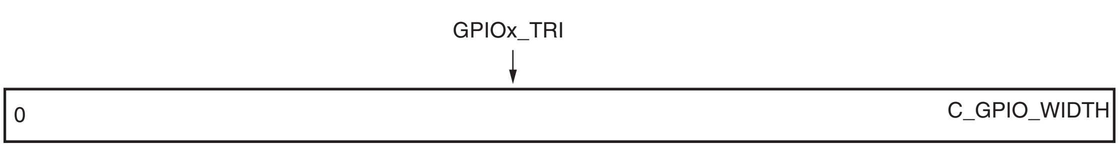

Figure 4: **XPS GPIO 3-State Register**

Table 7: **XPS GPIO 3-State Register Description**

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

XPS GPIO Interrupts

The XPS GPIO core 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 interface module includes interrupt detection logic to identify any transition on channel inputs. When a transition is detected, the same is indicated to the Interrupt Control module. The Interrupt Control module implements the necessary registers to enable and maintain the status of the interrupts. To support interrupt capability for channels, the PLB Interface module implements the following registers:

- Global Interrupt Enable register (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 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 allows interrupt service routines to clear one or more ISR bits using a single write transaction. (If needed, it also caters for setting one or more ISR bits--for testing interrupt service routines, for example.)

Table 8 details the XPS GPIO interrupt registers and their offset from the base address of XPS GPIO memory map. These registers are meaningful only if the C_INTERRUPT_PRESENT generic is set to 1.

Table 8: **XPS GPIO Interrupt Registers**

Register Name	Description	PLB Address	Access
GIE	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

Global Interrupt Enable Register (GIE)

The 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 Global Interrupt Enable Register is given in Table 9.

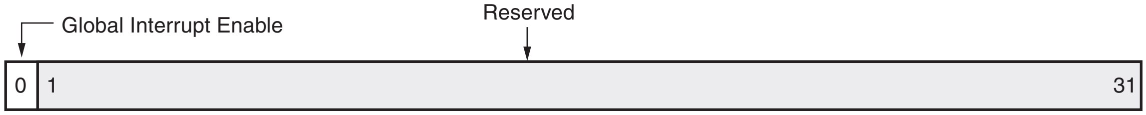

Figure 5: **Global Interrupt Enable Register**

Table 9: **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 '1' = Enabled '0' = Disabled
1 - 31	Reserved	N/A	'0'	Reserved. 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 for each of the interrupts. 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 PLB 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 behaviour.

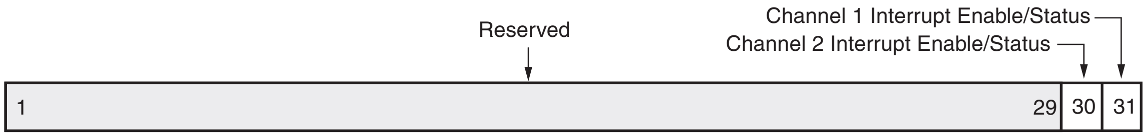

Figure 6: **IP Interrupt Enable and IP Interrupt Status Register**

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	Reserved	N/A	'0'	Reserved. Set to zeros on a read
30	Channel 2 Interrupt Enable	Read/Write	'0'	Enable Channel 2 Interrupt '1' = Enabled '0' = Disabled (masked)
31	Channel 1 Interrupt Enable	Read/Write	'0'	**Enable Channel 1 Interrupt** '1' = Enabled '0' = Disabled (masked)

Table 11: **IP Interrupt Status Register Description**

Bit(s)	Name	Core Access	Reset Value	Description
0 - 29	Reserved	N/A	'0'	Reserved. Set to zeros on a read
30	Channel 2 Interrupt Status	Read/TOW [1]	'0'	Channel 2 Interrupt Status '1' = Channel 2 input interrupt '0' = No Channel 2 input interrupt
31	Channel 1 Interrupt Status	Read/TOW [1]	'0'	Channel 1 Interrupt Status '1' = Channel 1 input interrupt '0' = No Channel 1 input interrupt

Notes:

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

XPS GPIO Operation

The XPS GPIO can be configured as either a single or a dual channel device using the C_IS_DUAL generic. When both channels are enabled (C_IS_DUAL = 1), each channel has the same size, as defined by the C_GPIO_WIDTH size.

The XPS 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 pins has a corresponding bit in the GPIOx_TRI register.

To configure a port as output, the corresponding bit in the 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 pins for non 3-state connections.

To configure a port as input, the corresponding bit in the 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 bits for each channel if dual channel operation is used.

User Application Hints

The user may find the following steps helpful in accessing the XPS 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 Global Interrupt, by setting the bit 0 of the Global Interrupt Register to 1.
3. When an interrupt is received, read the corresponding bit in the GPIOx_DATA register. Clear 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 Virtex-4, Virtex-5 and Spartan-3 family FPGA's.

Device Utilization and Performance Benchmarks

Since the GPIO Controller will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are just estimates. When the XPS GPIO Controller is combined with other designs in the system, the utilization of FPGA resources and timing will vary from the results reported here.

The XPS GPIO benchmarks and the resource utilization for various parameter combinations are detailed in Table 12, Table 13 and Table 14 for Virtex-4, Virtex-5 and Spartan-3 respectively.

Table 12: **Performance and Resource Utilization Benchmarks For Virtex-4 (Device: xc4vlx25-ff668-10)**

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slices	Slice Flip-Flops	LUTs	
0	0	1	X [1]	103	156	55	204.834
0	0	0	X [1]	180	254	95	245.761
0	1	1	X [1]	160	245	122	198.807
0	1	0	X [1]	252	340	162	180.083
1	0	1	1	118	192	121	214.832
1	0	0	1	175	288	161	200.120
1	0	0	0	259	352	171	188.147
1	1	1	1	245	349	234	183.959
1	1	0	1	311	442	274	177.620

Note:

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 XPS GPIO module without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user condition

Table 13: **Performance and Resource Utilization Benchmarks For Virtex-5 (Device: xc5vlx50-ff1153-1)**

Parameter Values				Device Resources		F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slice Flip- Flops	LUTs	
0	0	1	X [1]	156	53	219.684
0	0	0	X [1]	254	93	190.549
0	1	1	X [1]	245	114	172.831
0	1	0	X [1]	340	154	218.293

Table 13: **Performance and Resource Utilization Benchmarks For Virtex-5 (Device: xc5vlx50-ff1153-1)**

Parameter Values				Device Resources		F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slice Flip- Flops	LUTs	
1	0	1	1	192	117	241.546
1	0	0	1	288	125	223.546
1	0	0	0	352	165	233.263
1	1	1	1	349	218	148.148
1	1	0	1	442	226	172.921

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 XPS GPIO module without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user condition

Table 14: **Performance and Resource Utilization Benchmarks For Spartan-3(Device: xc3s1600e-fg484-4)**

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slices	Slice Flip-Flops	LUTs	
0	0	1	X [1]	85	156	55	131.631
0	0	0	X [1]	148	254	95	161.786
0	1	1	X [1]	202	245	122	126.502
0	1	0	X [1]	323	340	162	126.040
1	0	1	1	125	192	121	134.989
1	0	0	1	232	288	161	131.148
1	0	0	0	225	357	171	143.554

Table 14: **Performance and Resource Utilization Benchmarks For Spartan-3(Device: xc3s1600e-fg484-4)**

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_ALL_INPUTS	C_ALL_INPUTS_2	Slices	Slice Flip-Flops	LUTs	
1	1	1	1	269	349	234	125.376
1	1	0	1	325	442	274	125.094

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 XPS GPIO module without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user condition

Reference Documents

- *IBM 128-Bit Processor Local Bus, Architecture Specifications, v4.6*.

Revision History

Date	Version	Revision
02/15/07	1.0	Initial Xilinx release
4/20/07	1.1	Removed Spartan-3AF and added Spartan-3 support.