XPS General Purpose Input/Output (GPIO) (v2.00a)

DS569 December 2, 2009 **Product Specification**

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 PLBv4.6.

Features

- Connects as a 32-bit slave on PLB v4.6 bus of 32, 64 or 128 bits
- 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
- Width of each of the channels can be individually configured
- Independent reset values for each bit of all registers
- Optional interrupt request generation

LogiCORE™ IP Facts	
Core Specifics	
Supported Device Family	Spartan®-3A/3A DSP, Spartan-3, Spartan-3E, Automotive Spartan 3/3E/3A/3A DSP, Spartan-6, Virtex®-4 /4Q/4QV, Virtex-5/5FX, Virtex-6/6CX
Resources Used	
See Table 12, Table 13, Table 14, Table 15 and Table 16.	
Provided with Core	
Documentation	Product Specification
Design File Formats	VHDL
Constraints File	EDK TCL Generated
Verification	N/A
Instantiation Template	EDK
Design Tool Requirements	
Xilinx Implementation Tools	ISE® 11.4 or later
Verification	ModelSim PE/SE 6.4b or later
Simulation	ModelSim PE/SE 6.4b or later
Synthesis	XST
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 of each of the channels can be configured individually.

The ports can be configured dynamically for input or output by enabling or disabling the 3-state buffer. 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

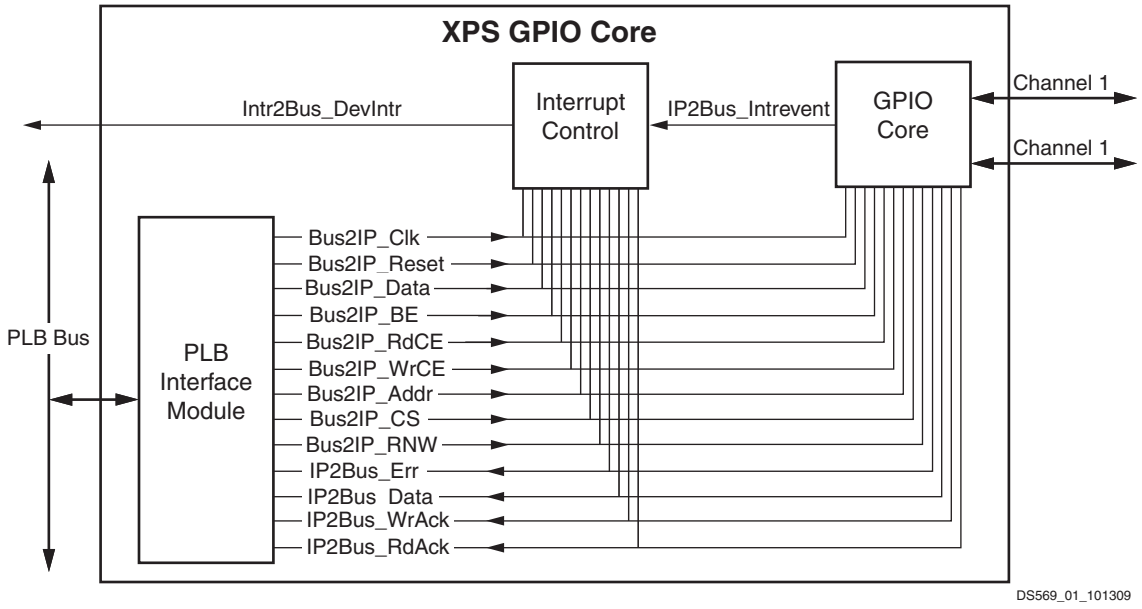

Figure 1: **XPS GPIO Block Diagram**

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 additional information on the generics, see 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.

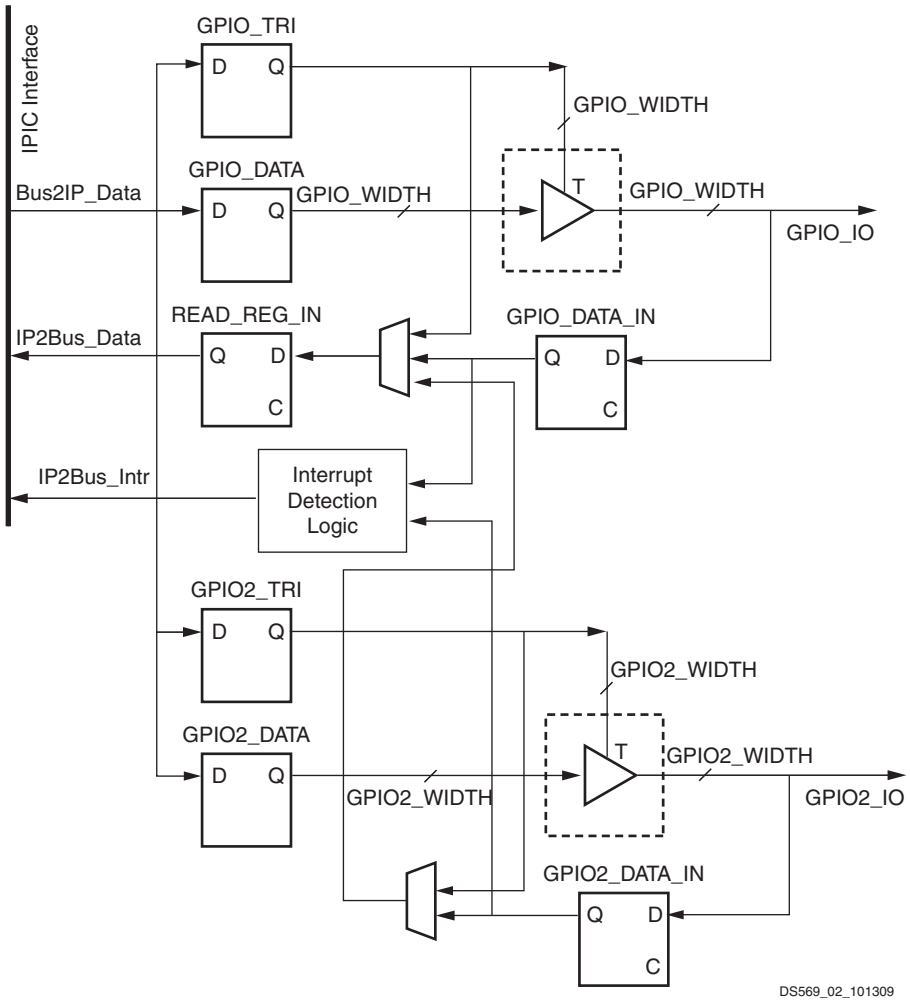

Figure 2: **GPIO_CORE Dual Channel Implementation**

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
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

Table 1: XPS GPIO I/O Signals *(Cont'd)*

Port	Signal Name	Interface	I/O	Initial State	Description
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
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_IO[0 to C_GPIO_WIDTH-1]	GPIO	I/O	Z (1)	Channel 1 General purpose I/O
P45	GPIO2_IO[0 to C_GPIO2_WIDTH-1]	GPIO	I/O	Z(2)	Channel 2 General purpose I/O

1. GPIO remains at high impedance only if the default value of C_TRI_DEFAULT is used
2. 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	spartan3, spartan3e, spartan3a, spartan3adsp, aspartan3, aspartan3e, aspartan3a, aspartan3adsp,spartan6, virtex4, qvirtex4, qvvirtex4, virtex5, virtex5fx, virtex6, virtex6cx	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
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 Channel1 Data Bus Width	C_GPIO_WIDTH	1-32	32	integer
G12	GPIO Channel2 Data Bus Width	C_GPIO2_WIDTH	1-32	32	integer
G13	XPS GPIO Interrupt	C_INTERRUPT_ PRESENT	0 = Interrupt control module is not present 1 = Interrupt control module is present	0	integer

Table 2: XPS GPIO Design Parameters

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G14	GPIO_DATA reset value	C_DOUT_DEFAULT	Any valid std_logic_vector	00000000	std_logic_vector
G15	GPIO_TRI reset value	C_TRI_DEFAULT	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
G16	Use dual channel	C_IS_DUAL	0 = Single channel is enabled 1 = Both the channels are enabled	0	integer
G17	GPIO2_DATA reset value	C_DOUT_DEFAULT_2	Any valid std_logic_vector	00000000	std_logic_vector
G18	GPIO2_TRI reset value	C_TRI_DEFAULT_2	Any valid std_logic_vector	FFFFFFFF	std_logic_vector
G19	Future Usage	C_ALL_INPUTS	0,1	0	Integer
G20	Future Usage	C_ALL_INPUTS_2	0,1	0	Integer

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

Table 3: Parameter-Port Dependencies *(Cont'd)*

Generic or Port	Parameter	Affects	Depends	Relationship Description
G11	C_GPIO_WIDTH	P44	-	The size of the registers GPIO_DATA and GPIO_TRI determines the number of GPIO_IO pins.
G12	C_GPIO2_WIDTH	P45	G17	The size of the registers GPIO2_DATA and GPIO2_TRI determines the number of GPIO2_IO pins.
G17	C_IS_DUAL	P45	-	When C_IS_DUAL is 1, channel 2 is created
G19	C_ALL_INPUTS_2	-	G17	For future usage and is valid only When C_IS_DUAL is 1
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
P10	PLB_wrDBus	-	G5	Width varies with the PLB Data Bus
P36	Sl_MBusy	-	G8	Width varies with the number of masters
P37	Sl_MWrErr	-	G8	Width varies with the number of masters
P38	Sl_MRdErr	-	G8	Width varies with the number of masters
P47	GPIO_IO	-	G11	Width depends on the GPIO width.
P51	GPIO2_IO	-	G12	Width depends on the GPIO2 width

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 and are byte accessible. 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 unimplemented register has no effect. An attempt to read the unimplemented register will return "all zero" value.

Table 5: Parameter-Register Dependency

Parameter Values	Register Retainability			
C_IS_DUAL(1)	GPIO_DATA(2)	GPIO_TRI(2)	GPIO2_DATA(2)	GPIO2_TRI(2)
0	Yes	Yes	No	No
1	Yes	Yes	Yes	Yes

1. When C_IS_DUAL = 0, the core is configured for single channel
2. Depending on the values of C_GPIO_WIDTH and C_GPIO2_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 and C_GPIO2_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 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.

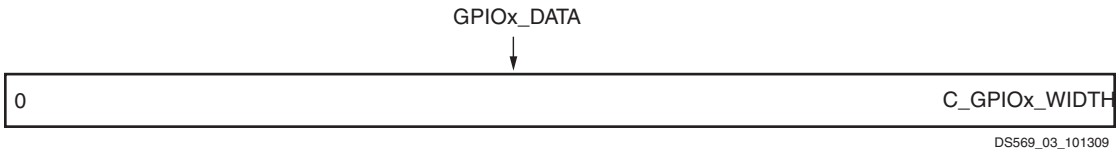

Figure 3: **XPS GPIO Data Register**

Table 6: XPS GPIO Data Register Description

Bits	Name	Core Access	Reset Value	Description
0:31	GPIOx_DATA	Read(1)/Write(2)	C_DOUT_DEFAULT C_DOUT_DEFAULT_2	XPS GPIO Data For each I/O bit Programmed as input: **R**: Read value on input pin **W**: No effect For each I/O bit Programmed as output: **R**: No effect(2) **W**: Writes value to corresponding XPS GPIO data register bit and output pin

1. When a GPIO pin is configured as input, writing to the pin has no effect in the XPS corresponding GPIO data register bit
2. When a GPIO pin is configured as an output, the corresponding data register bit 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 2, 3-state control register (GPIO2_TRI) is present only if the core is configured for dual channel (C_IS_DUAL = 1).

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

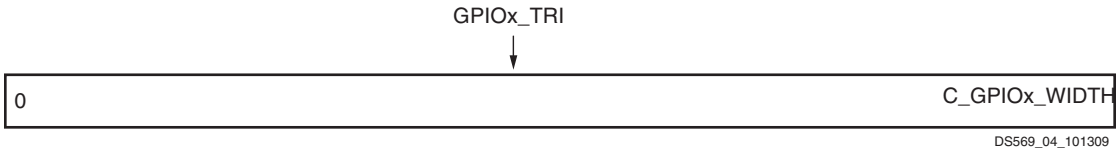

Figure 4: **XPS GPIO 3-State Register**

Table 7: XPS GPIO 3-State Register Description

Bits	Name	Core Access	Reset Value	Description
0:31	GPIOx_TRI	Read/Write	C_TRI_DEFAULT C_TRI_DEFAULT_2	XPS GPIO 3-state control. Each I/O pin of the XPS 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

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
GIER	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)

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.

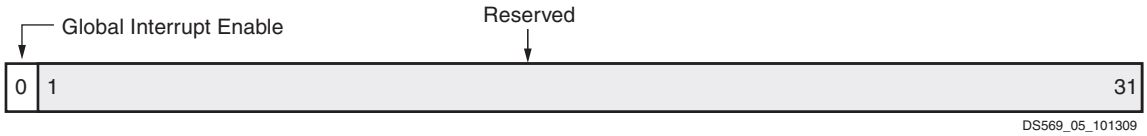

Figure 5: **Global Interrupt Enable Register**

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 behavior.

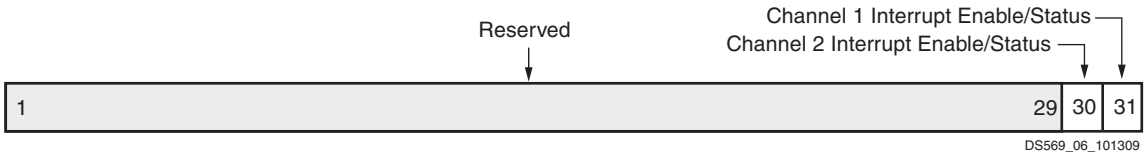

Figure 6: **IP Interrupt Enable and IP Interrupt Status Register**

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

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), the width of each channel can be different, as defined by the C_GPIO_WIDTH and C_GPIO2_WIDTH generics.

The XPS GPIO has a 3-state I/O capability. 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.

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 takes input from the bi-directional (GPIOx_IO) pins.

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 target technology is an FPGA listed in the Supported Device Family field of the LogiCORE IP Facts table.

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 the subsequent tables.

For the benchmarks and the resource utilization for the Virtex-4 device, see Table 12.

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

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	
0	0	32	32	173	264	110	254
0	0	16	32	100	182	90	275
0	1	32	16	213	351	206	234
0	1	1	1	61	100	66	252
1	0	32	32	222	394	220	232
1	0	5	28	153	262	175	227
1	0	28	5	145	262	175	224
1	1	32	32	294	548	331	197
1	1	15	28	233	416	279	205

1. These benchmark designs contain the XPS GPIO module only without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user condition

For the benchmarks and the resource utilization for the or the Virtex-5 device, see Table 13.

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

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slice	Slice Flip-Flops	LUTs	
0	0	32	32	77	264	109	285
0	0	16	32	55	182	89	317
0	1	32	16	99	351	200	246
0	1	1	1	40	100	60	293
1	0	32	32	110	394	219	267
1	0	5	28	80	262	174	289
1	0	28	5	80	262	174	277
1	1	32	32	152	548	323	239
1	1	15	28	121	416	270	256

1. 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

For the benchmarks and the resource utilization for the Virtex-6 device, see Table 14.

Table 14: Performance and Resource Utilization Benchmarks for Virtex-6 (XC6VLX130-1-TFF1156)

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	
0	0	32	32	55	264	182	383
0	0	16	32	43	182	134	415
0	1	32	16	89	351	271	317
0	1	1	1	32	100	71	303
1	0	32	32	88	394	303	347
1	0	5	28	74	262	211	287
1	0	28	5	72	262	217	311
1	1	32	32	128	548	441	290
1	1	15	28	107	416	340	309

1. 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

For the benchmarks and the resource utilization for the Spartan-6 device, see Table 15.

Table 15: Performance and Resource Utilization Benchmarks For Spartan-3 (xc3s1600e-fg484-4)

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	
0	0	32	32	141	264	110	160
0	0	16	32	99	182	90	166
0	1	32	16	324	351	206	135
0	1	1	1	71	100	66	166
1	0	32	32	321	394	220	146
1	0	5	28	142	262	175	176
1	0	28	5	201	262	175	147
1	1	32	32	400	548	331	156
1	1	15	28	281	416	280	137

1. 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

For the benchmarks and the resource utilization for the Spartan-6 device, see Table 16.

Table 16: Performance and Resource Utilization Benchmarks for Spartan-6 (XC6SLX16-2-CSG324)

Parameter Values				Device Resources			F_{MAX} (MHz)
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	
0	0	32	32	56	264	169	178
0	0	16	32	39	182	116	188
0	1	32	16	82	351	236	149
0	1	1	1	71	100	66	160
1	0	32	32	75	394	250	168
1	0	5	28	66	262	188	155
1	0	28	5	70	262	188	148
1	1	32	32	109	548	359	141
1	1	15	28	102	416	292	133

1. 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

System Performance

To measure the system performance (F_{MAX}) of this core, this core was added to a Virtex-4 system, a Virtex-5 system, and a Spartan-3A system as the Device Under Test (DUT) as shown in Figure 1, Figure 2, and Figure 3.

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

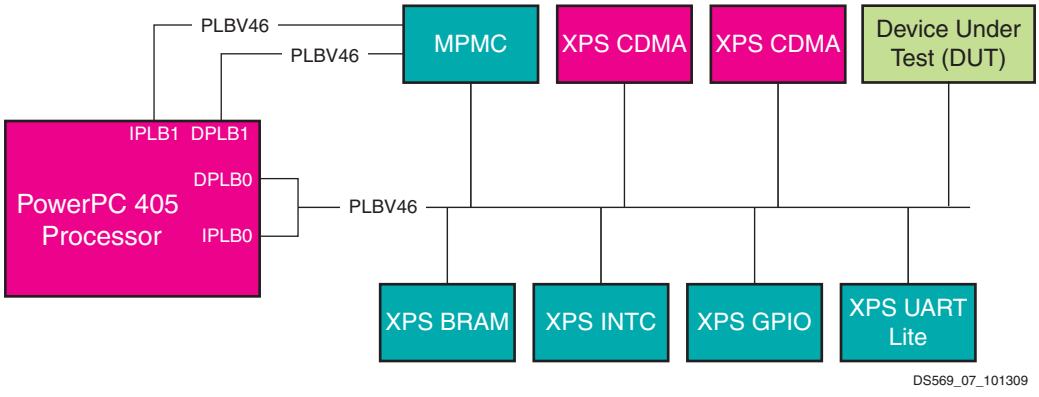

Figure 7: Virtex-4 FX FPGA System with the XPS GPIO as the DUT

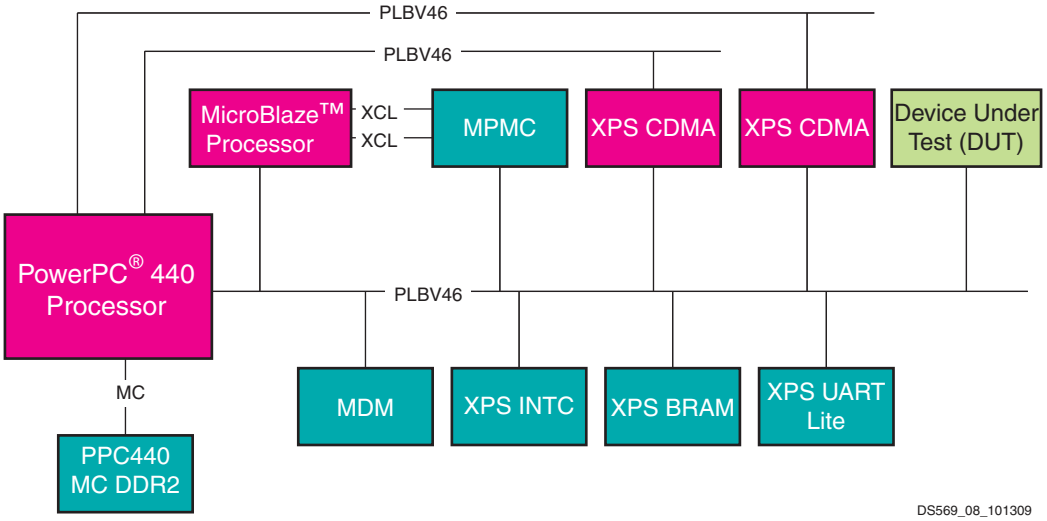

Figure 8: Virtex-5 FXT FPGA System with the XPS GPIO Core as the DUT

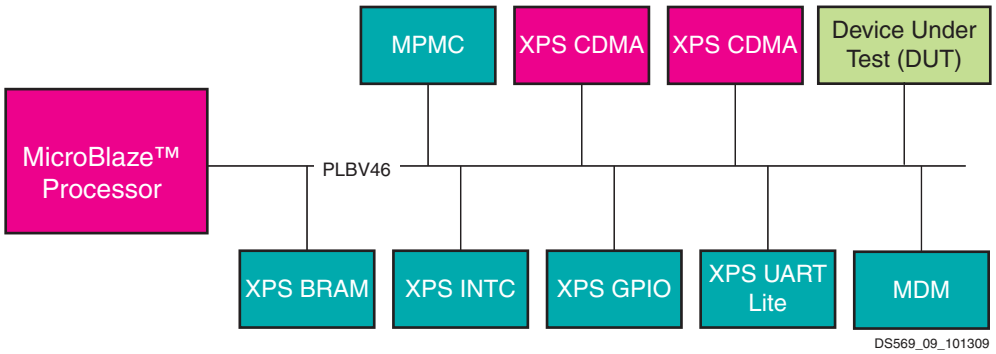

Figure 9: Spartan-3A FPGA System with the XPS GPIO Core as the DUT

The target FPGA was then filled with logic to drive the LUT and BRAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target F_{MAX} numbers are shown in Table 17.

Table 17: XPS GPIO Core System Performance

Target FPGA	Target f_{MAX} (MHz)
S3AD3400 -4	90
V4FX60 -10	100
V5FXT70 -1	120

The target F_{MAX} is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Reference Documents

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

Revision History

Date	Version	Description of Revisions
11/10/06	1.0	Initial Xilinx release
9/26/2007	1.1	Added FMax Margin System Performance section.
11/27/2007	1.2	Added Spartan-3A DSP to supported devices.
1/14/08	1.3	Added Virtex-II Pro support.
4/16/08	1.4	Added Automotive Spartan-3E, Automotive Spartan-3A, and Automotive Spartan-3A DSP support.
7/22/08	1.5	Added QPro Virtex-4 Hi-Rel and QPro Virtex-4 Rad Tolerant FPGA support.
10/13/08	1.6	Removed Virtex-II Pro and added Spartan-3 devices, added GPIO2_IO parameter description
12/08/08	1.7	Added C_ALL_INPUTS and C_ALL_INPUTS_2 parameters
01/05/09	1.8	Updated resource utlization table
4/28/09	1.9	Updated the margin system diagrams
7/17/09	1.10	Updated resource utlization tables; added Virtex-6 and Spartan-6 resource utilization tables
12/2/09	2.0	Listed supported devices families in LogiCORE Table; converted to new DS template.

Notice of Disclaimer