LogiCORE IP AXI GPIO (v1.00a)

DS744 September 21, 2010 **Product Specification**

Introduction

The AXI GPIO provides a general purpose input/output interface to the AXI (Advanced eXtensible Interface) interface. This 32-bit soft IP core is designed to interface with the AXI4-Lite interface.

Features

- Supports the AXI4-Lite interface specification
- Supports configurable single or dual GPIO channel(s)
- Supports configurable channel width for GPIO pins from 1 to 32 bits
- Supports dynamic programming of each GPIO bit as input or output
- Supports individual configuration of each channel
- Supports independent reset values for each bit of all registers
- Supports optional interrupt request generation

LogiCORE IP Facts					
Core Specifics					
Supported Device Family(1)	Spartan®-6 and Virtex®-6				
Supported User Interfaces	AXI4-Lite				
Resources	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs**	
	See Table 12 and Table 13				
Provided with Core					
Documentation	Product Specification				
Design Files	VHDL				
Example Design	Not Provided				
Test Bench	Not Provided				
Constraints File	Not Provided				
Simulation Model	Not Provided				
Design Tool Requirements					
Design Entry Tools	XPS 12.3				
Simulation	Mentor Graphics ModelSim 6.5c or later				
Synthesis Tools	XST 12.3 or later				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.

Functional Description

The AXI GPIO design provides a general purpose input/output interface to an AXI4-Lite interface. The AXI GPIO can be configured as either a single- or a dual-channel device. The width of each channel is independently configurable.

The ports are configured dynamically for input or output by enabling or disabling the three-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 subsequent sections. The AXI GPIO core is comprised of the following modules:

- AXI Interface Module
- Interrupt Controller
- GPIO core

The AXI GPIO modules are shown in the top-level block diagram in Figure 1.

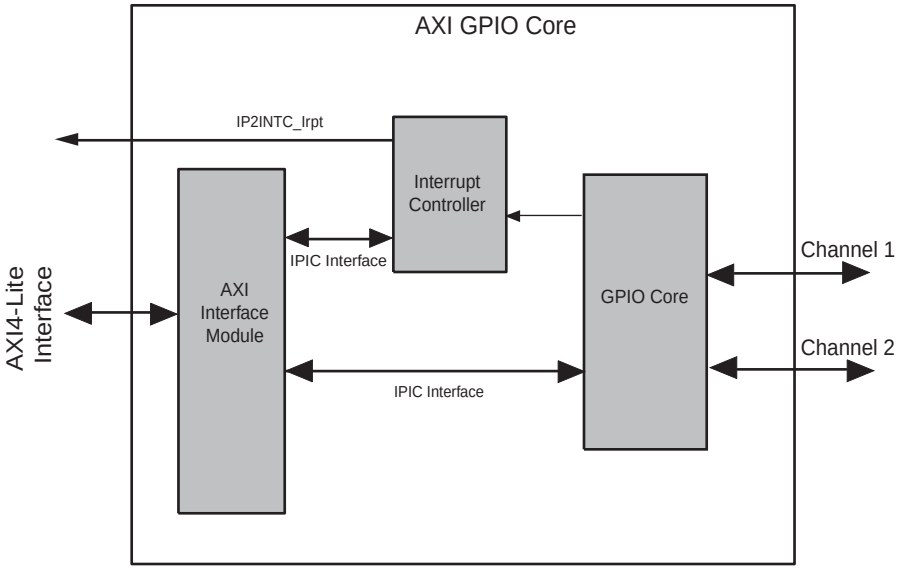

Figure 1: **Block Diagram of AXI GPIO**

AXI Interface Module

The AXI Interface Module provides a transaction interface termination from the AXI4-Lite to an internal IPIC interface using the AXI4-Lite IPIF library [Ref 3].

Interrupt Controller

The 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 by asserting interrupt signals. The Interrupt Controller will be enabled only when the C_INTERRUPT_PRESENT generic is set to 1. For more information on the generics, see Table 2, page 5.

GPIO Core

GPIO core provides an interface between the IPIC interface and the AXI GPIO channels. The GPIO core consists of registers and multiplexers for reading and writing the AXI 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 three-state buffers in the figure are not actually part of the core. The three-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.

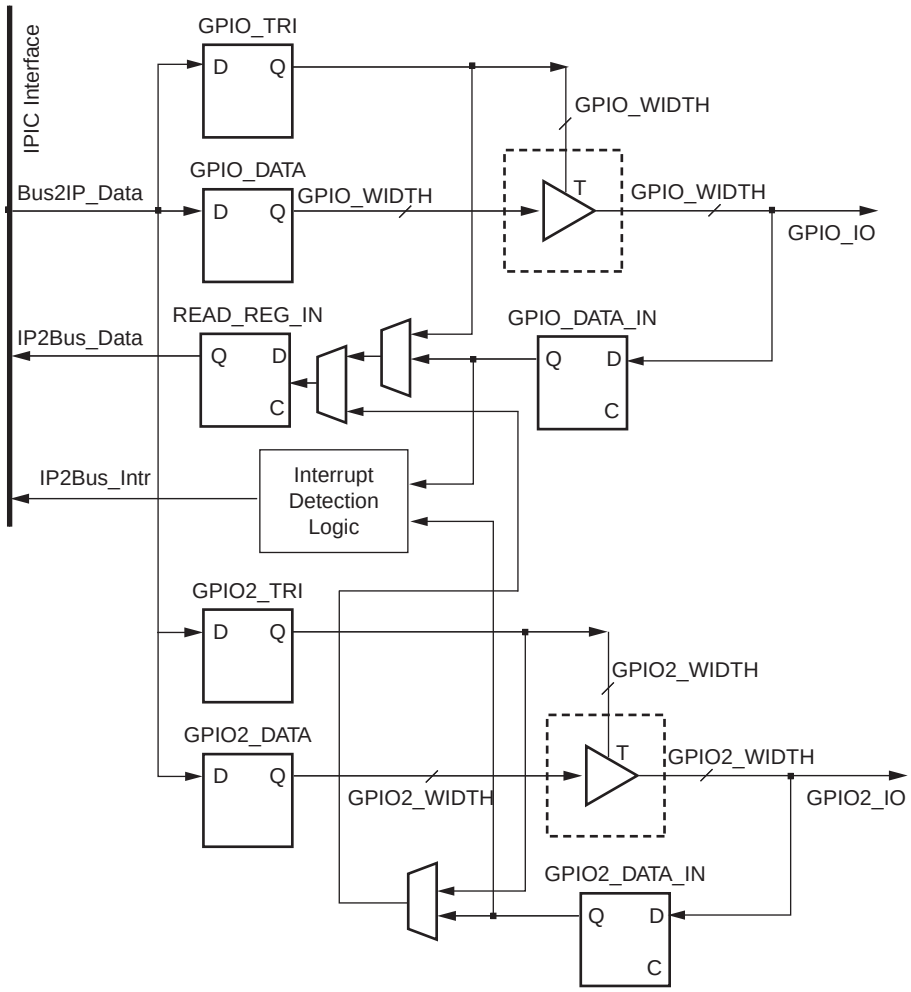

Figure 2: **GPIO core dual channel implementation**

I/O Signals

The AXI GPIO I/O signals are listed and described in Table 1.

Table 1: **I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Global System Signals					
P1	S_AXI_ACLK	AXI	I	-	AXI Clock
P2	S_AXI_ARESETN	AXI	I	-	AXI Reset. This signal is active low.
AXI Write Address Channel Signals					
P3	S_AXI_AWADDR [C_S_AXI_ADDR_WIDTH-1:0]	AXI	I	-	AXI write address. The write address bus gives the address of the write transaction.
P4	S_AXI_AWVALID	AXI	I	-	Write address valid. This signal indicates that valid write address and control information are available.
P5	S_AXI_AWREADY	AXI	O	0x0	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AXI Write Data Channel Signals					
P6	S_AXI_WDATA [C_S_AXI_DATA_WIDTH - 1: 0]	AXI	I	-	Write data.
P7	S_AXI_WSTB [C_S_AXI_DATA_WIDTH/8-1:0]	AXI	I	-	Write strobes. This signal indicates which byte lanes to update in memory.
P8	S_AXI_WVALID	AXI	I	-	Write valid. This signal indicates that valid write data and strobes are available
P9	S_AXI_WREADY	AXI	O	0x0	Write ready. This signal indicates that the slave can accept the write data.
AXI Write Response Channel Signals					
P10	S_AXI_BRESP[1:0]	AXI	O	0x0	Write response. This signal indicates the status of the write transaction: • 00 - OKAY • 10 - SLVERR
P11	S_AXI_BVALID	AXI	O	0x0	Write response valid. This signal indicates that a valid write response is available.
P12	S_AXI_BREADY	AXI	I	-	Response ready. This signal indicates that the master can accept the response information.
AXI Read Address Channel Signals					
P13	S_AXI_ARADDR [C_S_AXI_ADDR_WIDTH -1:0]	AXI	I	-	Read address. The read address bus gives the address of a read transaction.
P14	S_AXI_ARVALID	AXI	I	-	Read address valid. When high, this signal indicates that the read address and control information is valid and will remain stable until the address acknowledgement signal, S_AXI_ARREADY, is high.
P15	S_AXI_ARREADY	AXI	O	0x1	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals.

Table 1: **I/O Signal Description *(Cont'd)***

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Read Data Channel Signals					
P16	S_AXI_RDATA [C_S_AXI_DATA_WIDTH -1:0]	AXI	O	0x0	Read data.
P17	S_AXI_RRESP[1:0]	AXI	O	0x0	Read response. This signal indicates the status of the read transfer. • 00 - OKAY • 10 - SLVERR
P18	S_AXI_RVALID	AXI	O	0x0	Read valid. This signal indicates that the required read data is available and the read transfer can complete.
P19	S_AXI_RREADY	AXI	I	-	Read ready. This signal indicates that the master can accept the read data and response information.
System Interface Signals					
P20	IP2INTC_Irpt	System	O	0x0	AXI GPIO Interrupt. Active high, level sensitive signal.
GPIO Interface Signals					
P21	GPIO_IO_I [C_GPIO_WIDTH-1 : 0]	GPIO	I	-	Channel 1 general purpose input pins.
P22	GPIO_IO_O [C_GPIO_WIDTH-1 : 0]	GPIO	O	0x0	Channel 1 general purpose output pins.
P23	GPIO_IO_T [C_GPIO_WIDTH-1 : 0]	GPIO	O	0x0	Channel 1 general purpose three-state signal.
P24	GPIO2_IO_I [C_GPIO2_WIDTH-1 : 0]	GPIO	I	-	Channel 2 general purpose Input pins.
P25	GPIO2_IO_O [C_GPIO2_WIDTH-1 : 0]	GPIO	O	0x0	Channel 2 general purpose output pins.
P26	GPIO2_IO_T [C_GPIO2_WIDTH-1 : 0]	GPIO	O	0x0	Channel 2 general purpose three-state signal.

Design Parameters

To obtain a AXI 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 AXI interface module while others provide information to minimize resource utilization. Table 2 shows the features that can be parameterized in the AXI GPIO.

Table 2: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameter					
G1	Target FPGA family	C_FAMILY	spartan6, virtex6		string
AXI Parameters					
G2	AXI Base Address	C_BASEADDR	Valid Address(1)	0xffffffff(2)	std_logic_vector

Table 2: **Design Parameters *(Cont'd)***

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G3	AXI High Address	C_HIGHADDR	Valid Address(1)	0x00000000(2)	std_logic_vector
G4	AXI Address Bus Width	C_S_AXI_ADDR_WIDTH	32	32	integer
G5	AXI Data Bus Width	C_S_AXI_DATA_WIDTH	32	32	integer
G6	AXI interface type	C_S_AXI_PROTOCOL	AXI4LITE	AIX4LITE	string
GPIO Parameters					
G7	GPIO Channel 1 Data Bus Width	C_GPIO_WIDTH	1-32	32	integer
G8	GPIO Channel 2 Data Bus Width	C_GPIO2_WIDTH	1-32	32	integer
G9	AXI GPIO Interrupt	C_INTERRUPT_PRESENT	0 = Interrupt Controller module is not present 1 = Interrupt Controller module is present	0	integer
G10	GPIO_DATA Reset Value	C_DOUT_DEFAULT	Any valid std_logic_vector	0x00000000	std_logic_vector
G11	GPIO_TRI Reset Value	C_TRI_DEFAULT	Any valid std_logic_vector	0xFFFFFFFF	std_logic_vector
G12	Use Dual Channel	C_IS_DUAL	0 = Single channel is enabled 1 = Both channels are enabled	0x0	integer
G13	GPIO2_DATA Reset Value	C_DOUT_DEFAULT_2	Any valid std_logic_vector	0x00000000	std_logic_vector
G14	GPIO2_TRI Reset Value	C_TRI_DEFAULT_2	Any valid std_logic_vector	0xFFFFFFFF	std_logic_vector
G15	Future Usage	C_ALL_INPUTS	0,1	0x0	Integer
G16	Future Usage	C_ALL_INPUTS_2	0,1	0x0	Integer

1. The user must set the values. The C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1.
2. An invalid default value is used to ensure that the actual value is set. If the value is not set, a compiler error will be generated.
3. C_HIGHADDR - C_BASEADDR must be a power of 2 greater than or equal to C_BASEADDR + 0xFFF.

Allowable Parameter Combinations

The range specified by C_BASEADDR and C_HIGHADDR must encompass the memory space required by the AXI GPIO. The minimum range specified by C_BASEADDR and C_HIGHADDR should be at least 0xFFF.

For example, if C_BASEADDR is 0xE0000000, C_HIGHADDR must be at least equal to 0xE00000FFF.

Parameter I/O Signal Dependencies

The dependencies between the AXI GPIO core design parameters and I/O signals are described in Table 3. In addition, when certain features are parameterized out of the design, the related logic will no longer be a part of the design. The unused input signals and related output signals are set to a specified value.

Table 3: **Parameter I/O Signal Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G4	C_S_AXI_ADDR_WIDTH	P3, P13	-	Defines the width of the ports.
G5	C_S_AXI_DATA_WIDTH	P6, P7, P16	-	Defines the width of the ports.
I/O Signals				
P3	S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0]	-	G4	Port width depends on the generic C_S_AXI_ADDR_WIDTH.
P6	S_AXI_WDATA[C_S_AXI_DATA_WIDTH-1:0]	-	G5	Port width depends on the generic C_S_AXI_DATA_WIDTH.
P7	S_AXI_WSTB[C_S_AXI_DATA_WIDTH/8-1:0]	-	G5	Port width depends on the generic C_S_AXI_DATA_WIDTH.
P13	S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH -1:0]	-	G4	Port width depends on the generic C_S_AXI_ADDR_WIDTH.
P16	S_AXI_RDATA[C_S_AXI_DATA_WIDTH -1:0]	-	G5	Port width depends on the generic C_S_AXI_DATA_WIDTH.
P20	IP2INTC_Irpt	-	G10	Will be used only when C_INTERRUPT_PRESENT is 1.
P21	GPIO_IO_I	-	G8	Width depends on the GPIO Input width.
P22	GPIO_IO_O	-	G8	Width depends on the GPIO output width.
P23	GPIO_IO_T	-	G8	Width depends on the GPIO tri-state pin width.
P24	GPIO2_IO_I	-	G9,G13	Width depends on the GPIO2 Input width and will be enabled only when C_IS_DUAL is 1.
P25	GPIO2_IO_O	-	G9,G13	Width depends on the GPIO2 Input width and will be enabled only when C_IS_DUAL is 1.
P26	GPIO2_IO_T	-	G9,G13	Width depends on the GPIO2 Input width and will be enabled only when C_IS_DUAL is 1.

Registers

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

Table 4: **Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x00	GPIO_DATA	Read/Write	0x0	Channel 1 AXI GPIO Data Register.
C_BASEADDR + 0x04	GPIO_TRI	Read/Write	0x0	Channel 1 AXI GPIO Three-state Register.
C_BASEADDR + 0x08	GPIO2_DATA	Read/Write	0x0	Channel 2 AXI GPIO Data Register.
C_BASEADDR + 0x0C	GPIO2_TRI	Read/Write	0x0	Channel 2 AXI GPIO Three-state Register.

Depending on the value of certain configuration parameters, some of these registers are removed. A write to an unimplemented register has no effect. An attempt to read the unimplemented register will return an "all zero" value. The register dependencies of these parameters are described in Table 5.

Table 5: **Parameter-Register Dependency**

Parameter Values	Register Retainability			
C_IS_DUAL	GPIO_DATA(1)	GPIO_TRI(1)	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 three-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.

AXI GPIO Data Register (GPIOx_DATA)

The AXI 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 AXI GPIO data register.

There are two AXI4-Lite GPIO data registers (GPIO_DATA and GPIO2_DATA), one corresponding to each channel. The channel 1 data register (GPIO_DATA) is always present; the channel 2 data register (GPIO2_DATA) is present only if the core is configured for dual channel (C_IS_DUAL = 1).

The AXI GPIO Data Register is shown in Figure 3, and Table 6 details this register's functionality.

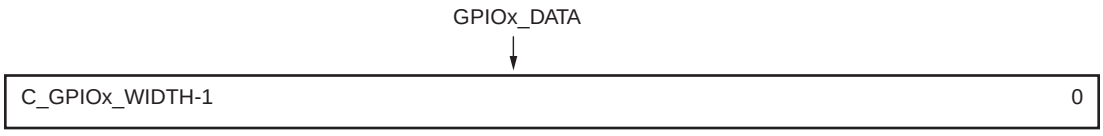

Figure 3: **AXI GPIO Data Register**

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

Bits	Name	Core Access	Reset Value	Description
C_GPIOx_WIDTH-1:0	GPIOx_DATA	Read/Write	C_DOUT_DEFAULT C_DOUT_DEFAULT_2	AXI 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. • **W**: Writes value to corresponding AXI GPIO data register bit and output pin.

AXI GPIO Three-State Register (GPIOx_TRI)

The AXI GPIO three-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 AXI GPIO three-state control registers (GPIO_TRI and GPIO2_TRI), one corresponding to each channel. The channel 2 three-state control register (GPIO2_TRI) is present only if the core is configured for dual channel (C_IS_DUAL = 1).

The AXI GPIO three-state Register is shown in Figure 4; and the register's functionality is described in Table 7.

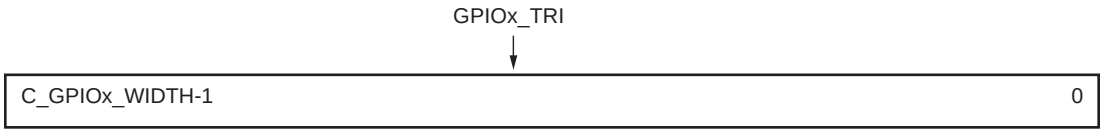

Figure 4: **AXI GPIO Three-State Register**

Table 7: **AXI GPIO Three-State Register Description**

Bits	Name	Core Access	Reset Value	Description
C_GPIOx_WIDTH-1:0	GPIOx_TRI	Read/Write	C_TRI_DEFAULT C_TRI_DEFAULT_2	AXI GPIO Three-state Control. Each I/O pin of the AXI 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.

Interrupts

The AXI GPIO core can be configured under the control of the C_INTERRUPT_PRESENT generic to generate a level 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, it is indicated to the Interrupt Controller module. The Interrupt Controller module implements the necessary registers to enable and maintain the status of the interrupts. To

support interrupt capability for channels, the Interrupt Controller module implements the following registers:

- Global Interrupt Enable register (GIE): Provides the master enable/disable for the interrupt output to the processor or Interrupt Controller. See "Global Interrupt Enable Register (GIE)" for more details.
- IP Interrupt Enable register (IP IER): Implements the independent interrupt enable bit for each channel. See "IP Interrupt Enable (IP IER) and IP Status Registers (IP ISR)" for more details.
- IP Interrupt Status register (IP ISR): Implements the 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. The IP ISR can also be manually set to generate an interrupt for testing purposes. See "IP Interrupt Enable (IP IER) and IP Status Registers (IP ISR)" for more details.

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

Table 8: **Interrupt Registers**

Register Name	Description	AXI4-Lite Address	Default Value (hex)	Access
GIER	Global Interrupt Enable Register	C_BASEADDR + 0x11C	0x0	Read/Write
IP IER	IP Interrupt Enable Register	C_BASEADDR + 0x128	0x0	Read/Write
IP ISR	IP Interrupt Status Register	C_BASEADDR + 0x120	0x0	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 valid 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.

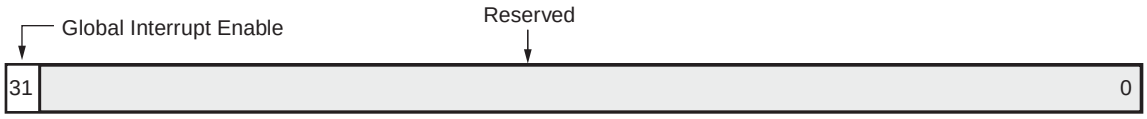

Figure 5: **Global Interrupt Enable Register**

Table 9: **Global Interrupt Enable Register Description**

Bit(s)	Name	Core Access	Reset Value	Description
31	Global Interrupt Enable	Read/Write	0	Master enable for the device interrupt output to the system interrupt controller: • 1 - Enabled • 0 - Disabled
30 - 0	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 valid 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 AXI4-Lite clock, and therefore the change in the input port must be stable for at least one clock period to guarantee interrupt capture. Each IP ISR register bit can be set or cleared via software by the Toggle-On-Write behavior.

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

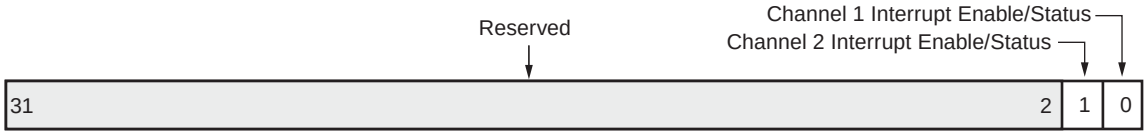

Figure 6: **IP Interrupt Enable and IP Interrupt Status Register**

Table 10: **IP Interrupt Enable Register Description**

Bit(s)	Name	Core Access	Reset Value	Description
31 - 2	Reserved	N/A	0	Reserved. Set to zeros on a read.
1	Channel 2 Interrupt Enable	Read/Write	0	Enable Channel 2 Interrupt. • 1 - Enabled • 0 - Disabled (masked)
0	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
31 - 2	Reserved	N/A	0	Reserved. Set to zeros on a read.
1	Channel 2 Interrupt Status	Read/TOW(1)	0	Channel 2 Interrupt Status. • 1 - Channel 2 input interrupt • 0 - No Channel 2 input interrupt
0	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.

Operation

The AXI 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. GPIOx_IO is the bidirectional bus formed with GPIOx_IO_I, GPIOx_IO_O, and GPIOx_IO_T pins.

The AXI GPIO has a three-state I/O capability. The GPIOx_TRI register is used to enable the three-state buffers which enable three-state outputs on the GPIOx_IO pins. The GPIOx_TRI register is also driven out of the dedicated GPIOx_IO_T 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 three-state buffers. An input port takes input from the GPIOx_IO_I signal of 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 a level interrupt. There are independent interrupt enable and interrupt status bits for each channel if dual channel operation is used.

User Application Tips

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

For input ports when the channel is configured for interrupts, use the following steps:

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 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, use the following steps:

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, use the following steps:

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-6 and Spartan-6 FPGAs.

Device Utilization and Performance Benchmarks

Core Performance

Since the AXI 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 the AXI GPIO core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI GPIO design will vary from the results reported here.

The AXI GPIO resource utilization for various parameter combinations measured with a Virtex-6 FPGA as the target device is detailed in Table 12.

Table 12: **Performance/Resource Utilization Benchmarks on Virtex-6 (XC6VLX130T-1-FF1156)**

Parameter Values (other parameters at default value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	32	32	57	264	169	366
0	0	16	32	47	188	143	406
0	1	32	16	93	357	267	310
0	1	1	1	35	105	75	298
1	0	32	32	92	262	293	340
1	0	5	28	76	262	223	275
1	0	28	5	74	416	225	306
1	1	32	32	135	560	450	285
1	1	15	28	107	416	350	302

The AXI GPIO resource utilization for various parameter combinations measured with a Spartan-6 FPGA as the target device is detailed in Table 13.

Table 13: **Performance/Resource Utilization Benchmarks on Spartan-6 (XC6SLX16-2-CSG324)**

Parameter Values (other parameters at default value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	32	32	58	274	172	166
0	0	16	32	44	184	120	184
0	1	32	16	85	355	246	144
0	1	1	1	71	104	70	156
1	0	32	32	78	402	255	160
1	0	5	28	68	266	194	150
1	0	28	5	72	266	194	140
1	1	32	32	112	550	365	138
1	1	15	28	105	420	301	135

System Performance

To measure the system performance (F_{MAX}), this core was added to a Spartan-6 FPGA system and a Virtex-6 FPGA system as the Device Under Test (DUT), as shown in Figure 7.

Because the AXI GPIO core is 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 design will vary from the results reported here.

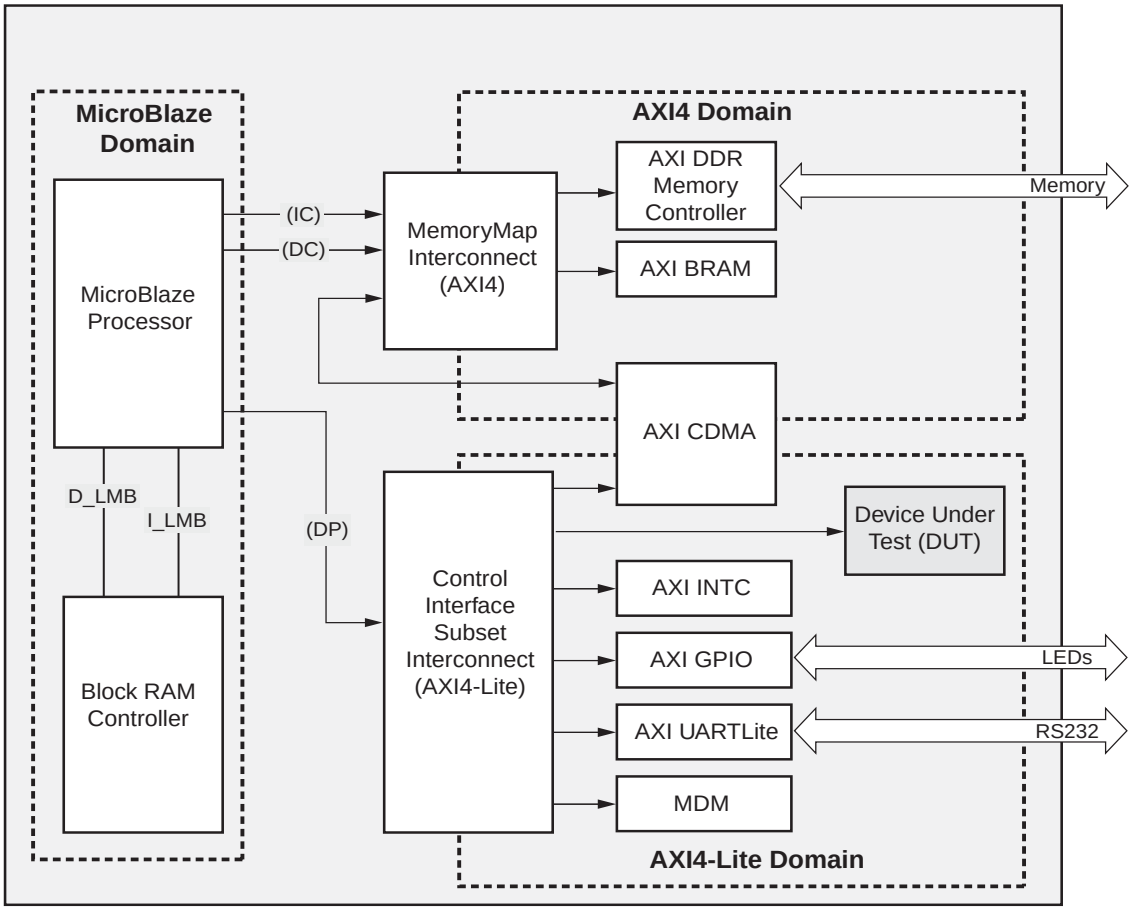

Figure 7: **Virtex-6 and Spartan-6 Devices F_{MAX} Margin System**

To measure the performance, the target FPGA was filled with logic to drive the LUT and block RAM 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 14.

Table 14: **AXI GPIO System Performance**

Target FPGA	Target F_{MAX} (MHz)
Spartan-6	110
Virtex-6	200

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

Support

Xilinx provides technical support for this LogiCORE IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

See the IP Release Notes Guide (XTP025) for further information on this core. There will be a link to all the DSP IP and then to the relevant core being designed with.

For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for the core being used. The following information is listed for each version of the core:

- New Features
- Bug Fixes
- Known Issues

Reference Documents

1. AXI4 AMBA® AXI Protocol Version: 2.0 Specification
2. *Xilinx Interrupt Control Data Sheet* (DS516)
3. *Xilinx LogiCORE IP AXI Lite IPIF (v1.00a) Data Sheet* (DS765)

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
09/21/2010	1.0	Initial Xilinx release.

Notice of Disclaimer