

## Introduction

The Xilinx® LogiCORE™ IP Advanced eXtensible Interface General Purpose Input/Output (AXI GPIO) core provides a general purpose input/output interface to the AXI interface. This 32-bit soft Intellectual Property (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 Table	
<b>Core Specifics</b>	
Supported Device Family <sup>(1)</sup>	Zynq™-7000 <sup>(2)</sup> , Virtex®-7, Kintex™-7, Artix™-7, Virtex-6, Spartan-6
Supported User Interfaces	AXI4-Lite
Resources	See <a href="#">Table 12</a> through <a href="#">Table 16</a>
<b>Provided with Core</b>	
Design Files	ISE: VHDL Vivado™: RTL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Not Provided
Supported S/W Driver <sup>(3)</sup>	Standalone and Linux
<b>Tested Design Flows<sup>(4)</sup></b>	
Design Entry	Xilinx Platform Studio (XPS) Vivado Design Suite <sup>(5)</sup>
Simulation	Mentor Graphics ModelSim
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
<b>Support</b>	
Provided by Xilinx@ <a href="http://www.xilinx.com/support">www.xilinx.com/support</a>	

### Notes:

- For a complete list of supported derivative devices, see the [Embedded Edition Derivative Device Support](#).
- Supported in ISE Design Suite implementations only.
- Standalone driver details can be found in the EDK or SDK directory (`<install_directory>/doc/usenglish/xilinx_drivers.htm`). Linux OS and driver support information is available from [/wiki.xilinx.com](http://wiki.xilinx.com).
- For the supported versions of the tools, see the [Xilinx Design Tools: Release Notes Guide](#).
- Supports only 7 series devices.

## Functional Description

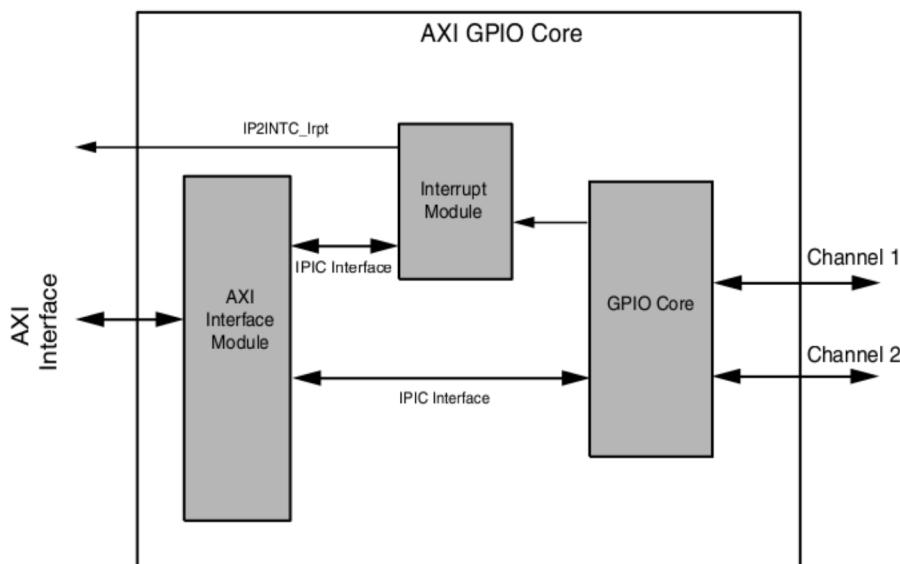
The AXI GPIO design provides a general purpose input/output interface to an AXI4-Lite interface. The AXI GPIO can be configured as either a single or a dual-channel device. The width of each channel is independently configurable.

The ports are configured dynamically for input or output by enabling or disabling the 3-state buffer. The channels can 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 IP Interconnect (IPIC) interface using the AXI4-Lite IP Interface (IPIF) library [\[Ref 2\]](#).

## 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 is enabled only when the C\_INTERRUPT\_PRESENT generic is set to 1. For more information on the generics, see [Table 2](#).

## 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 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 Input/Outputs (I/Os) option. The control signals of the IPIC interface are not shown in Figure 2.

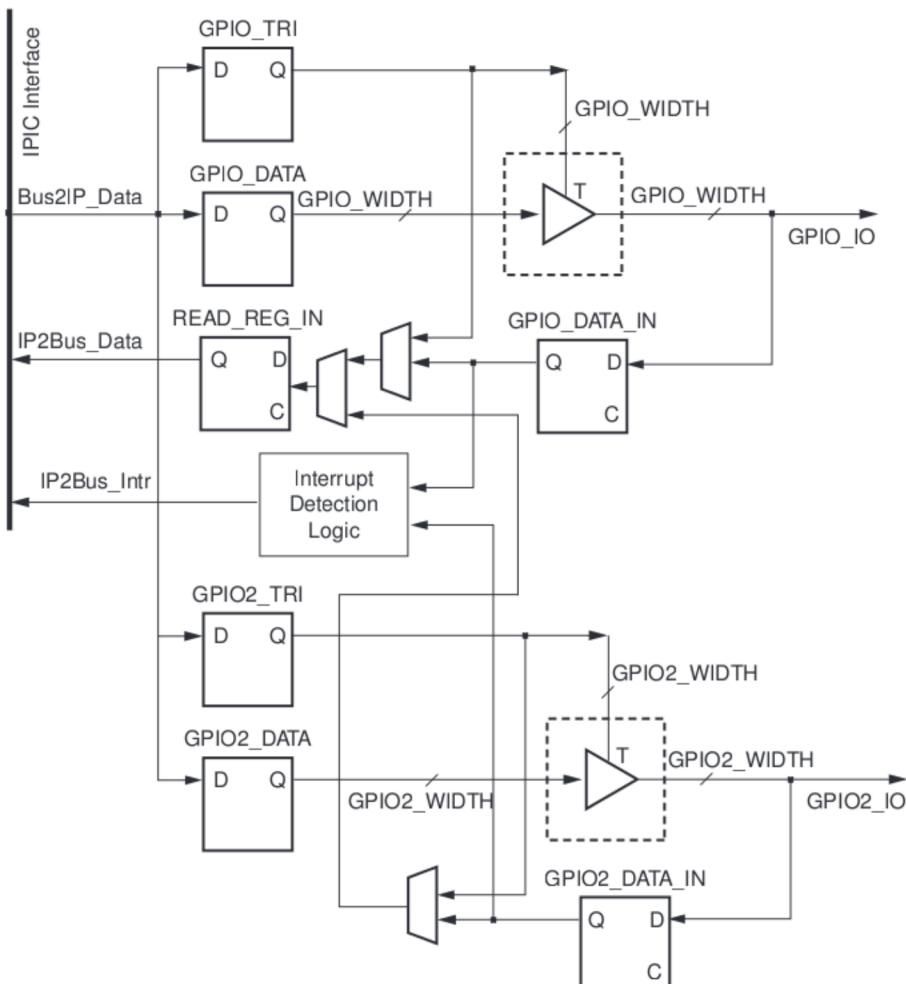


Figure 2: **GPIO Core Dual-Channel Implementation**

## 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
<b>AXI Global System Signals</b>					
P1	S_AXI_ACLK	AXI	I	-	AXI Clock
P2	S_AXI_ARESETN	AXI	I	-	AXI Reset. This signal is active-Low.
<b>AXI Write Address Channel Signals</b>					
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.
<b>AXI Write Data Channel Signals</b>					
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.
<b>AXI Write Response Channel Signals</b>					
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.
<b>AXI Read Address Channel Signals</b>					
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 remains 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
<b>AXI Read Data Channel Signals</b>					
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.
<b>System Interface Signals</b>					
P20	IP2INTC_Irpt	System	O	0x0	AXI GPIO Interrupt. Active-High, level sensitive signal.
<b>GPIO Interface Signals</b>					
P21	GPIO_IO_I [C_GPIO_WIDTH - 1:0] <sup>(1)</sup>	GPIO	I	-	Channel 1 general purpose input pins.
P22	GPIO_IO_O [C_GPIO_WIDTH - 1:0] <sup>(2)</sup>	GPIO	O	0x0	Channel 1 general purpose output pins.
P23	GPIO_IO_T [C_GPIO_WIDTH - 1:0] <sup>(3)</sup>	GPIO	O	0x0	Channel 1 general purpose 3-state signal.
P24	GPIO2_IO_I [C_GPIO2_WIDTH - 1:0] <sup>(4)</sup>	GPIO	I	-	Channel 2 general purpose Input pins.
P25	GPIO2_IO_O [C_GPIO2_WIDTH - 1:0] <sup>(5)</sup>	GPIO	O	0x0	Channel 2 general purpose output pins.
P26	GPIO2_IO_T [C_GPIO2_WIDTH - 1:0] <sup>(6)</sup>	GPIO	O	0x0	Channel 2 general purpose 3-state signal.

**Notes:**

1. If GPIO\_IO\_I is used in the Microprocessor Hardware Specification file (MHS), then only inputs are allowed.
2. If GPIO\_IO\_O is used in the MHS, then only outputs are allowed.
3. If the GPIO\_IO\_T is used in the MHS, all the three pins are used along with a 3-state buffer. The insertion of the 3-state buffer is automatically done by the tool, as the information exists in MPD.
4. If GPIO2\_IO\_I is used in the MHS, then only inputs are allowed.
5. If GPIO2\_IO\_O is used in the MHS, then only outputs are allowed.
6. If the GPIO2\_IO\_T is used in the MHS, all the three pins are used along with a 3-state buffer. The insertion of the 3-state buffer is automatically done by the tool, as the information exists in MPD.

## 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.

## Inferred Parameters

In addition to the parameters listed in [Table 2](#), there are also parameters that are inferred for each AXI interface in the Embedded Development Kit (EDK) tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect.

For a complete list of the interconnect settings related to the AXI interface, see DS768, *LogiCORE IP AXI Interconnect Data Sheet*.

**Table 2: Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
<b>System Parameter</b>					
G1	Target FPGA family	C_FAMILY	artix7, virtex7, kintex7, spartan6, virtex6	virtex6	string
<b>AXI Parameters</b>					
G2	AXI Base Address	C_BASEADDR	Valid Address <sup>(1)</sup>	0xFFFFFFFF <sup>(2)</sup>	std_logic_vector
G3	AXI High Address	C_HIGHADDR	Valid Address <sup>(1)</sup>	0x00000000 <sup>(2)</sup>	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	AXI4LITE	string
<b>GPIO Parameters</b>					
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

Table 2: Design Parameters (Cont'd)

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G15	Future Usage	C_ALL_INPUTS	0, 1	0x0	Integer
G16	Future Usage	C_ALL_INPUTS_2	0, 1	0x0	Integer

**Notes:**

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 is generated.
3. C\_HIGHADDR - C\_BASEADDR must be a power of 2 greater than or equal to C\_BASEADDR + 0xFFFF.

## 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 0xFFFF.

For example, if C\_BASEADDR is 0xE0000000, C\_HIGHADDR must be at least equal to 0xE0000FFF.

## 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 is no longer 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
<b>Design Parameters</b>				
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.
<b>I/O Signals</b>				
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	Is 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 3-state pin width.

**Table 3: Parameter I/O Signal Dependencies (Cont'd)**

Generic or Port	Name	Affects	Depends	Relationship Description
P24	GPIO2_IO_I	-	G9, G13	Width depends on the GPIO2 Input width and are enabled only when C_IS_DUAL is 1.
P25	GPIO2_IO_O	-	G9, G13	Width depends on the GPIO2 Input width and are enabled only when C_IS_DUAL is 1.
P26	GPIO2_IO_T	-	G9, G13	Width depends on the GPIO2 Input width and are 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 3-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 3-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 returns 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	

### Notes:

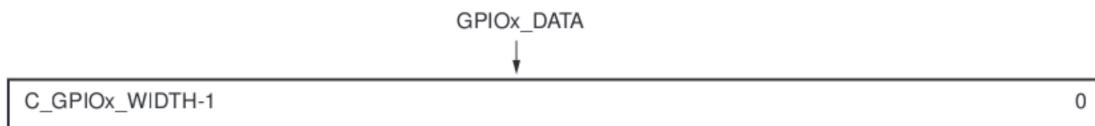
- When C\_IS\_DUAL = 0, the core is configured for single channel.
- Depending on the values of C\_GPIO\_WIDTH and C\_GPIO2\_WIDTH, the data registers and the 3-state control registers (GPIO\_DATA, GPIO\_TRI, GPIO2\_DATA and GPIO2\_TRI), when implemented, get trimmed to the size of value specified by C\_GPIO\_WIDTH and C\_GPIO2\_WIDTH.

## 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*

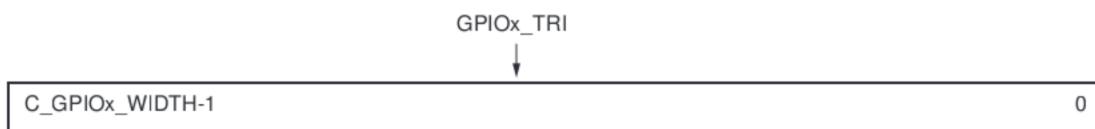
Bits	Name	Core Access	Reset Value	Description
C_GPIOx_WIDTH - [1:0]	GPIOx_DATA	Read/Write	C_DOUT_DEFAULT C_DOUT_DEFAULT_2	AXI GPIO Data. For each I/O bit programmed as input: <ul style="list-style-type: none"> <li><b>R:</b> Read value on input pin.</li> <li><b>W:</b> No effect.</li> </ul> For each I/O bit programmed as output: <ul style="list-style-type: none"> <li><b>R:</b> No effect.</li> <li><b>W:</b> Writes value to corresponding AXI GPIO data register bit and output pin.</li> </ul>

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

The AXI 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 AXI 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 AXI GPIO 3-state Register is shown in [Figure 4](#); 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 3-state Control. Each I/O pin of the AXI GPIO is individually programmable as an input or output. For each of the bits: <ul style="list-style-type: none"> <li>0 = I/O pin configured as output.</li> <li>1 = I/O pin configured as input.</li> </ul>

## 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*

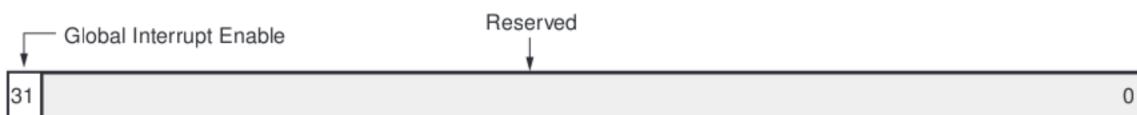
Register Name	Description	AXI4-Lite Address	Default Value (hex)	Access
GIER	Global Interrupt Enable Register	C_BASEADDR + 0x11C	0x0	Read/Write
IP IER	IP Interrupt Enable Register	C_BASEADDR + 0x128	0x0	Read/Write
IP ISR	IP Interrupt Status Register	C_BASEADDR + 0x120	0x0	Read/TOW <sup>1</sup>

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:** 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**

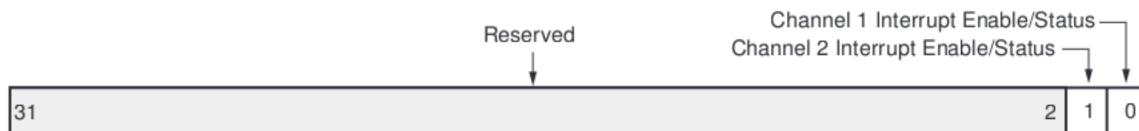
Bit(s)	Name	Core Access	Reset Value	Description
31	Global Interrupt Enable	Read/Write	0	Master enable for the device interrupt output to the system interrupt controller: 0 = Disabled 1 = Enabled
30 - 0	Reserved	N/A	0	Reserved. Set to zeroes on a read.

## 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 through 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 zeroes on a read.
1	Channel 2 Interrupt Enable	Read/Write	0	Enable Channel 2 Interrupt. 0 = Disabled (masked) 1 = Enabled
0	Channel 1 Interrupt Enable	Read/Write	0	Enable Channel 1 Interrupt. 0 = Disabled (masked) 1 = Enabled

Table 11: IP Interrupt Status Register Description

Bit(s)	Name	Core Access	Reset Value	Description
31–2	Reserved	N/A	0	Reserved. Set to zeroes on a read.
1	Channel 2 Interrupt Status	Read/TOW <sup>(1)</sup>	0	Channel 2 Interrupt Status 0 = No Channel 2 input interrupt 1 = Channel 2 input interrupt
0	Channel 1 Interrupt Status	Read/TOW <sup>(1)</sup>	0	Channel 1 Interrupt Status 0 = No Channel 1 input interrupt 1 = 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 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\_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 3-state buffers. An input port takes input from the GPIOx\_IO\_I signal of the bidirectional (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 causes 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 can find the following steps helpful in accessing the AXI GPIO core:

1. For input ports when the channel is configured for interrupts, use the following steps:
  - a. Configure the port as input by writing the corresponding bit in GPIOx\_TRI register with the value of 1.
  - b. 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.
  - c. 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.
2. For input ports when the channel is not configured for interrupt, use the following steps:
  - a. Configure the port as input by writing the corresponding bit in GPIOx\_TRI register with the value of 1.
  - b. 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 listed in the Supported Device Family field of the [LogiCORE IP Facts Table](#).

### Device Utilization and Performance Benchmarks

#### Core Performance

Because the AXI GPIO core is going to 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 varies from the results reported here.

[Table 12](#) shows the parameter values and resource utilization benchmarks for Virtex-7 devices.

**Table 12: Performance and Resource Utilization Benchmarks for Virtex-7 FPGA (XC7V855TFFG1157-3)**

Parameter Values (Other Parameters at Default Value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Flip-Flops	LUTs	F <sub>Max</sub> (MHz)
0	0	32	32	34	174	124	539.665
0	0	16	32	21	94	72	620.347
0	1	32	16	40	179	135	544.070
0	1	32	32	36	179	135	567.215
0	1	1	1	9	24	28	693.001
1	0	32	32	66	302	230	561.798
1	0	1	1	8	23	25	645.578
1	0	5	28	40	174	134	605.327
1	0	28	5	41	174	131	562.746
1	1	32	32	65	307	249	530.786
1	1	15	28	52	219	171	570.776
1	1	1	1	12	28	33	621.891

[Table 13](#) shows the parameter values and resource utilization benchmarks for Kintex-7 FPGA and Zynq-7000 device.

**Table 13: Performance and Resource Utilization Benchmarks for Kintex-7 FPGA<sup>1</sup> and Zynq-7000 Device**

Parameter Values (Other Parameters at Default Value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Flip-Flops	LUTs	F <sub>Max</sub> (MHz)
0	0	32	32	33	174	128	536.769
0	0	16	32	21	94	72	628.931
0	1	32	16	41	179	130	604.595
0	1	32	32	37	179	127	591.017

**Table 13: Performance and Resource Utilization Benchmarks for Kintex-7 FPGA<sup>1</sup> and Zynq-7000 Device**

Parameter Values (Other Parameters at Default Value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Flip-Flops	LUTs	F <sub>Max</sub> (MHz)
0	1	1	1	8	24	28	662.252
1	0	32	32	63	302	242	637.755
1	0	1	1	7	23	25	643.087
1	0	5	28	40	174	134	601.685
1	0	28	5	41	174	130	554.017
1	1	32	32	67	307	233	524.934
1	1	15	28	51	219	174	555.864
1	1	1	1	10	28	33	754.148

1. Kintex-7 FPGA (XC7K410TFFG676-3)

Table 14 shows the parameter values and resource utilization benchmarks for Artix-7 FPGA and Zynq-7000 device.

**Table 14: Performance and Resource Utilization Benchmarks for Artix-7 FPGA<sup>1</sup> and Zynq-7000 Device**

Parameter Values (Other Parameters at Default Value)				Device Resources			Performance
C_IS_DUAL	C_INTERRUPT_PRESENT	C_GPIO_WIDTH	C_GPIO2_WIDTH	Slices	Flip-Flops	LUTs	F <sub>Max</sub> (MHz)
0	0	32	32	35	174	120	455.581
0	0	16	32	22	94	72	473.037
0	1	32	16	41	179	133	453.104
0	1	32	32	39	179	119	461.467
0	1	1	1	9	24	28	498.753
1	0	32	32	66	302	230	453.926
1	0	1	1	8	23	25	465.983
1	0	5	28	41	174	132	443.853
1	0	28	5	43	174	134	426.985
1	1	32	32	65	307	241	393.546
1	1	15	28	49	219	172	439.174
1	1	1	1	10	28	33	432.900

1. Artix-7 FPGA (XC7A355TDIE)

The AXI GPIO resource utilization for various parameter combinations measured with a Virtex-6 FPGA as the target device is detailed in Table 15.

**Table 15: Performance and Resource Utilization Benchmarks for Virtex-6 FPGA (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 <sub>Max</sub> (MHz)
0	0	32	32	38	175	112	438
0	0	16	32	28	95	64	397
0	1	32	16	47	180	123	446

**Table 15: Performance and Resource Utilization Benchmarks for Virtex-6 FPGA (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 <sub>Max</sub> (MHz)
0	1	1	1	8	25	27	539
1	0	32	32	67	303	226	406
1	0	5	28	44	175	132	385
1	0	28	5	48	175	129	384
1	1	32	32	68	308	245	408
1	1	15	28	56	220	167	383

The AXI GPIO resource utilization for various parameter combinations measured with a Spartan-6 FPGA as the target device is detailed in [Table 16](#).

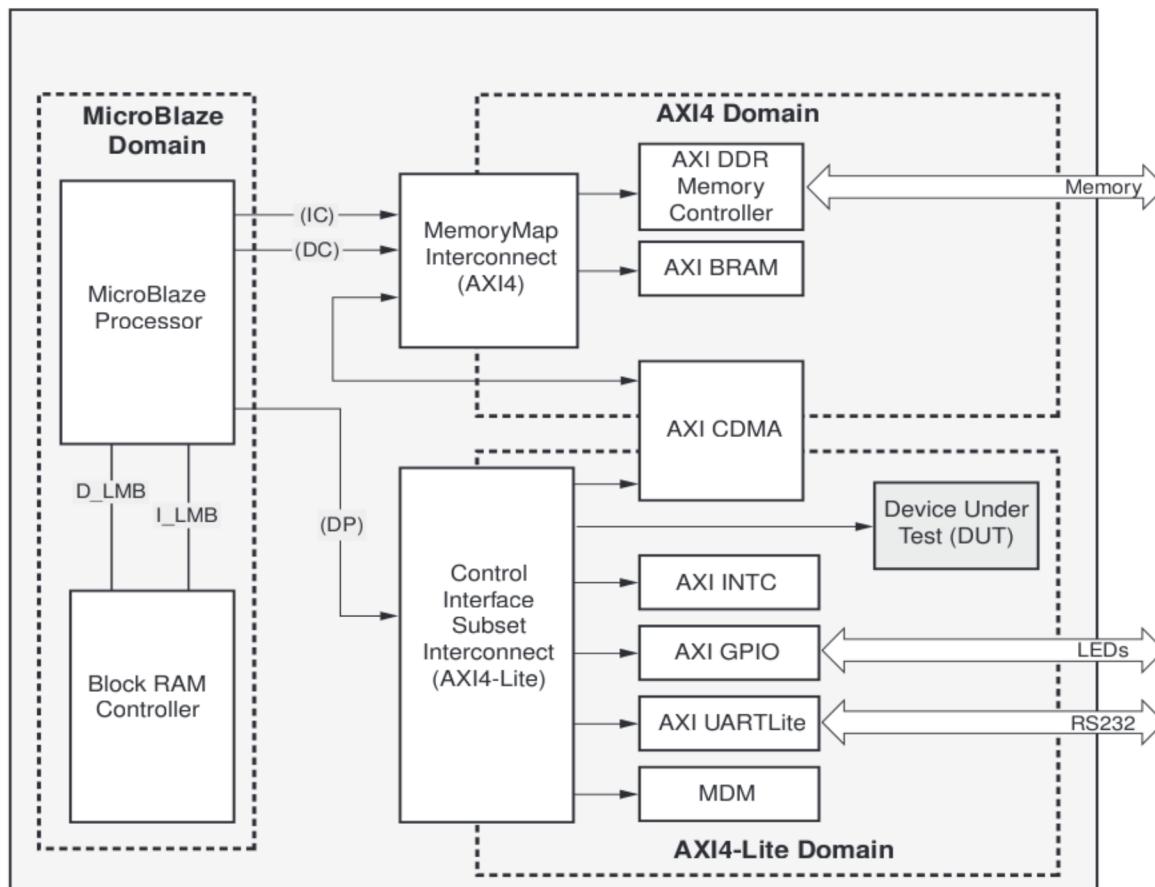
**Table 16: Performance and Resource Utilization Benchmarks for Spartan-6 FPGA (XC6SLX16-2-CSG324)**

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 <sub>Max</sub> (MHz)
0	0	32	32	33	175	104	206
0	0	16	32	24	95	64	220
0	1	32	16	46	180	107	201
0	1	1	1	10	25	27	292
1	0	32	32	60	303	194	165
1	0	5	28	43	175	117	206
1	0	28	5	44	175	113	249
1	1	32	32	63	308	205	211
1	1	15	28	53	220	156	164

## 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 varies 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 Lookup Table (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 17](#).

*Table 17: AXI GPIO System Performance*

Target FPGA	Target $F_{Max}$ (MHz)
Artix-7	110
Virtex-7	180
Kintex-7	180
Virtex-6	180
Spartan-6	110

The target F<sub>Max</sub> is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

## References

To search for Xilinx documentation, go to [www.xilinx.com/support](http://www.xilinx.com/support).

1. [DS516, Interrupt Control Data Sheet](#)
2. [DS765, LogiCORE IP AXI-Lite IPIF \(v1.01.a\) Data Sheet](#)
3. [DS768, LogiCORE IP AXI Interconnect Data Sheet](#)
4. [UG814, Vivado Design Suite Getting Started Guide](#)
5. ARM® AMBA® AXI4 Protocol Version: 2.0 Specification  
<http://www.arm.com/products/system-ip/amba/amba-open-specifications.php>

## Technical Support

Xilinx provides technical support at [www.xilinx.com/support](http://www.xilinx.com/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 IDS Embedded Edition Derivative Device Support web page ([www.xilinx.com/ise/embedded/ddsupport.htm](http://www.xilinx.com/ise/embedded/ddsupport.htm)) for a complete list of supported derivative devices for this core.

## Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado Design Suite and ISE Design Suite Embedded Edition tools under the terms of the [Xilinx End User License](#).

Information about this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your [local Xilinx sales representative](#).

## List of Acronyms

Table 18: List of Acronyms

Acronym	Description
AMBA	Advanced Microcontroller Bus Architecture
AXI	Advanced eXtensible Interface
DSP	Digital Signal Processing
DUT	Device Under Test
Fmax	Maximum Frequency
FPGA	Field Programmable Gate Array
GIER	Global Interrupt Enable Register
GPIO	General Purpose Input/Output
I/O	Input/Output

**Table 18: List of Acronyms (Cont'd)**

<b>Acronym</b>	<b>Description</b>
IP	Intellectual Property
IP IER	IP Interrupt Enable Register
IP ISR	IP Interrupt Status Register
IPIC	IP Interconnect
IPIF	IP Interface
ISE	Integrated Software Environment
LUT	Lookup Table
MHS	Microprocessor Hardware Description
MHz	Mega Hertz
MPD	Microprocessor Peripheral Definition
RAM	Random Access Memory
TOW	Toggle-On-Write
VHDL	VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits)
XPS	Xilinx Platform Studio
XST	Xilinx Synthesis Technology

## Revision History

The following table shows the revision history for this document:

<b>Date</b>	<b>Version</b>	<b>Description of Revisions</b>
9/21/10	1.0	Initial Xilinx release.
12/14/10	2.0	Updated for core v1.01a and XPS 12.4.
6/22/11	3.0	Updated for XPS 13.2. Added support for Artix-7, Virtex-7, and Kintex-7 devices.
10/19/11	4.0	Summary of Major Changes for Core Version v1.01.b <ul style="list-style-type: none"> <li>• Updated for 13.3 tools and new core version 1.01.b</li> <li>• Removed combinatorial loop on AXI side</li> </ul> Summary of Documentation Changes <ul style="list-style-type: none"> <li>• Added List of Acronyms</li> <li>• Updated IP Facts table</li> <li>• Updated page layout with newest template</li> <li>• Added Inferred Parameters section</li> <li>• Replaced "BRAM" with "block RAM"</li> <li>• Reordered listing of devices from newest to oldest: Artix-7, Virtex-7, Kintex-7, Virtex-6, Spartan-6</li> <li>• Added Ordering Information section</li> <li>• Updated Figures 1 and 7</li> <li>• Changed "three-state" to "3-state"</li> </ul>
07/25/12	5.0	Updated for Vivado and Zynq features with minor document updates for v14.2

## Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.