OPB IIC Bus Interface (v1.01d)

DS434 May 23, 2006 **Product Specification**

Introduction

This product specification defines the architecture and interface requirements for OPB IIC module. This includes registers that must be initialized for proper operation. OPB IIC module supports all features, except high speed mode, of the Philips IIC bus, v2.1, release January 2000. See the Specification Exceptions section for more details.

The Xilinx OPB IIC soft IP core design allows the customer to tailor the OPB IIC to suit their application by setting certain parameters to enable/disable features. The parameterizable features of the design are discussed in the OPB IIC Design Parameters section.

Features

- Master or Slave operation
- Multi-master operation
- Software selectable acknowledge bit
- Arbitration lost interrupt with automatic mode switching from Master to Slave
- Calling address identification interrupt with automatic mode switching from Master to Slave
- START and STOP signal generation/detection
- Repeated START signal generation
- Acknowledge bit generation/detection
- Bus busy detection
- Fast Mode 400 KHz operation or Standard Mode 100 KHz
- 7 Bit or 10 Bit addressing
- General Call Enable or Disable
- Transmit and Receive FIFOs - 16 bytes deep
- Throttling
- General Purpose Output, zero to 8 bits wide

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	QPro™-R Virtex™-II, QPro Virtex-II, Spartan™-II, Spartan-IIE, Spartan-3, Spartan-3E, Virtex, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-E	
Version of Core	OPB_IIC	v1.01d
Resources Used		
	Min	Max
I/O	2	2
LUTs	353	374
FFs	203	210
BRAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 6.1 or later with latest EDK	
Verification	ModelSim SE/EE 5.8 or later	
Simulation	ModelSim SE/EE 5.8 or later	
Synthesis	XST 6.1 or later	
Support		
Support provided by Xilinx, Inc.		

Functional Description

The top-level block diagram for the OPB IIC Bus Interface module is shown in Figure 1.

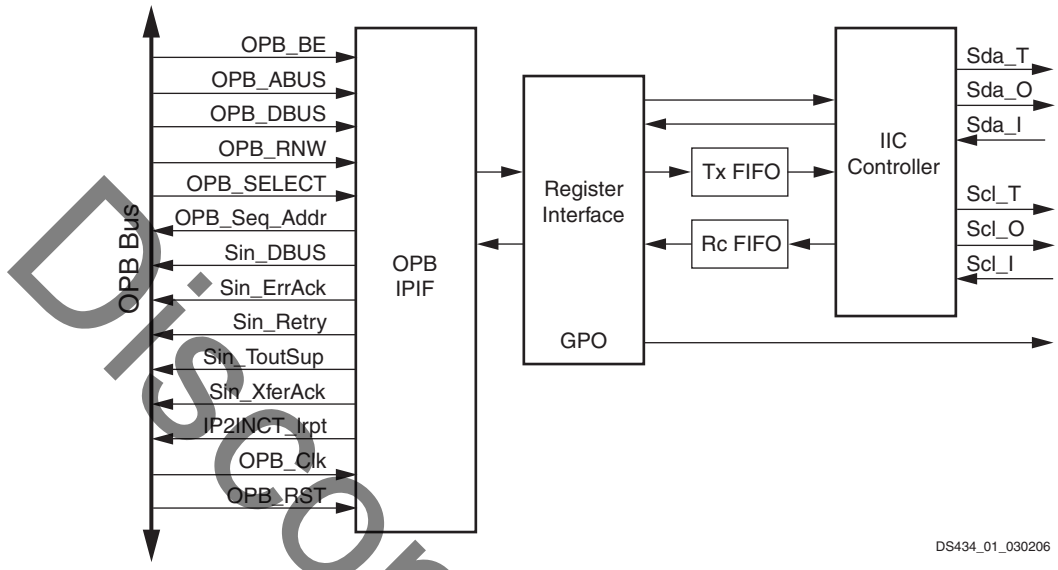

Figure 1: **OPB IIC Top Level Block Diagram**

IIC Protocol

This section describes the main protocol of the IIC bus. For more details and timing diagrams, see the Philips IIC specification.

The IIC bus consists of two wires, serial data (SDA) and serial clock (SCL), which carry information between the devices connected to the bus. The number of devices connected to the same bus is limited only by a maximum bus capacitance of 400 pF.

Both the SDA and SCL lines are bidirectional lines, connected to a positive supply voltage via a pull-up resistor. When the bus is free, both lines are High. The output stages of devices connected to the bus must have an open-drain or open-collector in order to perform the wired-AND function.

Each device on the bus has a unique address and can operate as either a transmitter or receiver. In addition, devices can also be configured as Masters or Slaves. A Master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. Any other device that is being addressed is considered a Slave.

The IIC protocol defines an arbitration procedure that insures that if more than one Master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted. The arbitration and clock synchronization procedures defined in the Philips IIC specification are supported by the OPB IIC Bus Interface module.

Data transfers on the IIC bus are initiated with a START condition and are terminated with a STOP condition. Normal data on the SDA line must be stable during the High period of the clock, including required setup and hold times.

The High or Low state of the data line can only change when SCL is Low. The START condition is a unique case and is defined by a High-to-Low transition on the SDA line while SCL is High. Likewise,

the STOP condition is a unique case and is defined by a Low-to-High transition on the SDA line while SCL is High. The definitions of data, START, and STOP ensure that the START and STOP conditions will never be confused as data is shown in Figure 2.

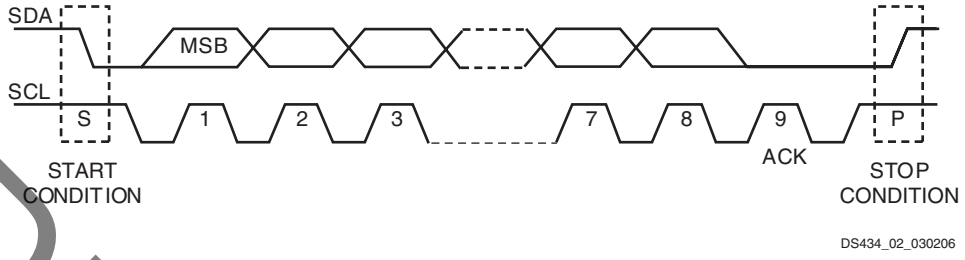

Figure 2: **Data Transfer on the IIC Bus**

Each data packet on the IIC bus consists of eight bits of data followed by an acknowledge bit so one complete data byte transfer requires nine clock pulses. Data is transferred with the most significant bit first (MSB).

The transmitter releases the SDA line during the acknowledge bit and the receiver of the data transfer must drive the SDA line low during the acknowledge bit to acknowledge receipt of the data.

If a Slave-receiver does not drive the SDA line Low during the acknowledge bit, this indicates that the Slave-receiver was unable to accept the data and the Master can then generate a STOP condition to abort the transfer.

If the Master-receiver does not generate an acknowledge, this indicates to the Slave-transmitter that this byte was the last byte of the transfer.

Standard communication on the bus between a Master and a Slave is composed of four parts: START, Slave address, Data transfer, and STOP. The IIC protocol defines a data transfer format for both 7-bit and 10-bit addressing.

A seven bit address is initiated as follows. After the START condition, a Slave address is sent. This address is seven bits long followed by an eighth-bit which is the read/write bit. A "1" indicates a request for data (read) and a "0" indicates a data transmission (write).

Only the Slave with the calling address that matches the address transmitted by the Master responds by sending back an acknowledge bit by pulling the SDA line Low on the ninth clock.

For 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slave that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slave address bits 9:8, and the LSB bit (bit 8) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address.

Once successful Slave addressing is achieved, the data transfer can proceed byte-by-byte as specified by the read/write bit. The Master can terminate the communication by generating a STOP signal to free the bus. However, the Master may generate a START signal without generating a STOP signal first. This is called a repeated START.

OPB IIC Design Parameters

To obtain an OPB IIC that is uniquely tailored to the user system requirements, certain features can be parameterized in the OPB IIC design. This allows a design that utilizes only the resources required by system and runs at the best possible performance. The features that can be parameterized in the Xilinx OPB IIC design are shown in Table 1.

Table 1: **OPB IIC Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
IIC Features					
G1	Output Frequency of the SCL signal(1)	C_IIC_FREQ	Less than or equal to 400KHz for Fast Mode definition(1)	100000	Integer
G2	10 Bit Addressing	C_TEN_BIT_ADR	1 = The slave will respond to 10 bit addresses 0 = The slave will respond to 7 bit addresses	0	Integer
G3	Width of GPO	C_GPO_WIDTH	1 to 8	1	Integer
OPB/IPIF Interface					
G4	OPB Clock Frequency(1)	C_CLK_FREQ	Requirements as stated in note 1	100000000	Integer
G5	Device Block ID(5)	C_DEV_BLK_ID	See note 5.	0	Integer
G6	Module Identification Register Enable(5)	C_DEV_MIR_ENABLE	See note 5.	0	Integer
G7	OPB High Address	C_HIGHADDR	Refer to "Allowable Parameter Combinations"	None(2,3)	std_logic_vector
G8	Location of the first IPIF register.	C_BASEADDR	Valid Address range(4).	None(2,3)	std_logic_vector
G9	OPB Address Bus Width(5)	C_OPB_AWIDTH	32	32	Integer
G10	OPB Data Bus Width(5)	C_OPB_DWIDTH	32	32	Integer

Notes:

1. The OPB clock frequency must be at least 25MHz and 25X the SCL clock frequency, due to the pipe-lined design of the IIC.
2. No default value will be specified to insure that the actual value is set, i.e. if the value is not set, a compiler error will be generated.
3. For example, C_BASEADDR = 0xE0000000, C_HIGHADDR = 0xE00001FF. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1
4. Address range specified by C_BASEADDR and C_HIGHADDR must be at least 0X200 and must be a power of 2.
5. See the *Processor IP Reference Guide* under Part 1: Embedded Processor IP, under IPIF, under OPB IPIF Architecture.

Allowable Parameter Combinations

The OPB IIC is a pipe-lined synchronous design, due to the pipe-lining, the OPB Clock must be at least 25 MHz and at least 25 times faster than the SCL clock.

The address range specified by C_BASEADDR and C_HIGHADDR must be a power of 2, and C_HIGHADDR range must be at least 0x200. For example, if C_BASEADDR = 0xE0000000, C_HIGHADDR must be at least = 0xE00001FF.

OPB IIC I/O Signals

The I/O signals for the OPB IIC are listed in Table 2.

Table 2: **OPB IIC I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
IIC Signals					
P1	Sda_I	System	I	0	IIC Serial Data Input from 3-state buffer.
P2	Sda_O	System	O	0	IIC Serial Data Output to 3-state buffer
P3	Sda_T	System	O	0	IIC Serial Data Output Enable to 3-state buffer (1)
P4	Scl_I	System	I	0	IIC Serial Clock Input from 3-state buffer
P5	Scl_O	System	O	0	IIC Serial Clock Output to 3-state buffer
P6	Scl_T	System	O	0	IIC Serial Clock Output Enable to 3-state buffer (1)
P7	GPO(32 - C_GPO_WIDTH: C_OPB_DWIDTH-1)	System	O	0	General Purpose Outputs
OPB Signals					
P8	Sln_DBus(0:C_OPB_ DWIDTH-1)	OPB	O	0	IIC output data bus
P9	Sln_xferAck	OPB	O	0	IIC transfer acknowledge
P10	Sln_Retry	OPB	O	0	IIC retry
P11	Sln_ToutSup	OPB	O	0	IIC timeout suppress
P12	Sln_ErrAck	OPB	O	0	IIC error acknowledge
P13	OPB_ABus(0:C_OPB_ AWIDTH-1)	OPB	I		OPB address bus
P14	OPB_BE(0:C_OPB_ DWIDTH/8-1)	OPB	I		OPB byte enables
P15	OPB_DBus(0:C_OPB_ DWIDTH-1)	OPB	I		OPB data bus
P16	OPB_RNW	OPB	I		Read not Write (OR of all master RNW signals)
P17	OPB_select	OPB	I		Master has taken control of the bus (OR of all master selects)
P18	OPB_seqAddr	OPB	I		OPB sequential address

Table 2: **OPB IIC I/O Signals *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P19	OPB_Clk	System	I		System clock
P20	OPB_Rst	System	I		System Reset (active high)
System					
P21	IP2INTC_Irpt	System	O		System Interrupt

Notes:
1. The Sda_T and Scl_T signals are the 3-state enable signals that control the data direction for the Sda and Scl signals

OPB IIC Port Dependencies

The width of some of the OPB IIC signals depends on parameters selected in the design. The dependencies between the OPB IIC design parameters and I/O signals are shown in Table 3.

Table 3: **OPB IIC Parameter Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G10	C_OPB_DWIDTH	P8, P14, P15		Specifies the OPB Data Bus width
G9	C_OPB_AWIDTH	P13		Specifies the OPB Address Bus width
G7	C_HIGHADDR		G8	Specifies the OPB High address range
G3	C_GPO_WIDTH	P7		Specifies the width of the general purpose output.
I/O Signals				
P7	GPO(32 - C_GPO_WIDTH: C_OPB_DWIDTH-1)		G3	Width varies with the size of the C_GPO_WIDTH
P8	Sln_DBus(0:C_OPB_DWIDTH-1)		G10	Width varies with the size of the OPB Data bus.
P13	OPB_ABus(0:C_OPB_AWIDTH-1)		G9	Width varies with the size of the OPB Address bus.
P14	OPB_BE(0:C_OPB_DWIDTH/8-1)		G10	Width varies with the size of the OPB Data bus.
P15	OPB_DBus(0:C_OPB_DWIDTH-1)		G10	Width varies with the size of the OPB Data bus.

Notes:
1. The OPB IIC Ten Bit Slave Address Register will exist only if the parameter C_TEN_BIT_ADR is 1.
2. Please refer to the *Processor IP Reference Guide* under Part 1: Embedded Processor IP, under IPIF, under OPB IPIF Architecture, under OPB IPIF Register Descriptions for a complete description of these registers. The bit mapping for these registers is provided in Figure 13.

OPB IIC Register Descriptions

The OPB IIC contains addressable registers for read/write operations as shown in Table 4. The base address for these registers is set in the parameter C_BASEADDR. C_BASEADDR + 0x100 represents the address of the first register in the OPB IIC - in this case, the OPB IIC Control Register. The address of each register is then calculated by an offset to the base address.

The IPIF contains the interrupt registers. The base address for these registers is set by the parameter C_BASEADDR which represents the address of the first register - in this case, the ISR. The ISR is not used by the IIC design, the location of the first accessible IPIF register is the GIE, offset of 0x1C. The address of each IPIF register is then calculated by an offset to the base address.

Table 4 shows all of the OPB IIC registers and the IPIF interrupt registers and their addresses. The OPB IIC Ten Bit Slave Address register exists when parameter to select it is set to 1.

Table 4: **OPB IIC Registers**

Register Name	OPB Address	Access	Name of Clock Enable
Device Global Interrupt Enable Register (GIE) (2)	C_BASEADDR + **0x01C**	Read/Write	
IP Interrupt Status Register (IPISR)(2, 3)	C_BASEADDR + **0x020**	Read/Write	
IP Interrupt Enable Register (IPIER)(2,3)	C_BASEADDR + **0x028**	Read/Write	
IPIF Software Reset Register(2)	C_BASEADDR + **0x040**	Write Only	
Control Register (CR)	C_BASEADDR + **0x100**	Read/Write	xxx_ce0
Status Register (SR)	C_BASEADDR + **0x104**	Read	xxx_ce1
Transmit FIFO	C_BASEADDR + **0x108**	Read/Write	xxx_ce2
Receive FIFO	C_BASEADDR + **0x10C**	Read	xxx_ce3
OPB IIC Slave Address Register (ADR)	C_BASEADDR + **0x110**	Read/Write	xxx_ce4
Transmit FIFO Occupancy	C_BASEADDR + **0x114**	Read	xxx_ce5
Receive FIFO Occupancy	C_BASEADDR + **0x118**	Read	xxx_ce6
OPB IIC Ten Bit Slave Address Register(1)	C_BASEADDR + **0x11C**	Read/Write	xxx_ce7
Receive FIFO programmable depth interrupt Register	C_BASEADDR + **0x120**	Read/Write	xxx_ce8
General Purpose Output Register (GPO)	C_BASEADDR + **0x124**	Read/Write	xxx_ce9

Notes:
1. The OPB IIC Ten Bit Slave Address Register will exist only if the parameter C_TEN_BIT_ADR is 1.
2. See the *Processor IP Reference Guide* under Part 1: Embedded Processor IP, under IPIF, under OPB IPIF Architecture, under OPB IPIF Register Descriptions for a complete description of these registers.
3. The bit mapping for these registers is provided in Figure 13.

OPB IIC Control Register

Prior to setting Master Slave Mode Select (MSMS) to a 1, Tx FIFO should contain the address of the OPB IIC device. All the CR bits can be set at the same time as setting MSMS to a 1 to initiate a bus transaction.

When initiating a repeated start condition, the transmit FIFO must be empty, then the repeated start bit should be set to a 1 and then the address of the OPB IIC device should be written to the transmit FIFO.

The rest of the FIFO can be filled with data, if required. The OPB IIC Control Register is shown in Figure 3 and described in Table 5.

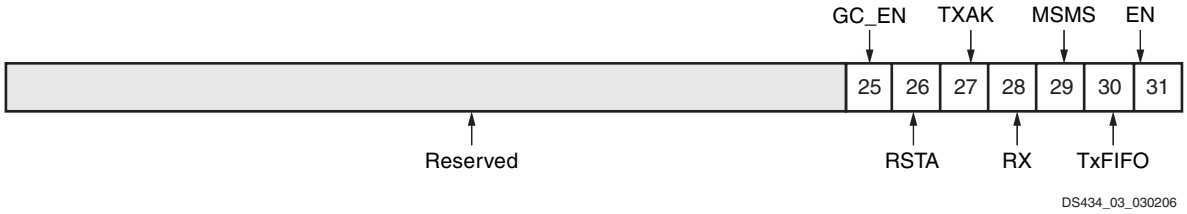

Figure 3: **OPB IIC Control Register**

Table 5: **OPB IIC Control Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0- 24	Reserved		0	**Reserved.**
25	GC_EN	Read/Write	0	**General Call Enable.** Setting this bit high will allow the OPB IIC to respond to a general call address. "0" - General Call Disabled. "1" - General Call Enabled.
26	RSTA	Read/Write	0	**Repeated Start.** Writing a "1" to this bit generates a repeated START condition on the bus if the OPB IIC Bus Interface is the current bus Master. Attempting a repeated START at the wrong time, if the bus is owned by another Master, results in a loss of arbitration. This bit is reset when the repeated start occurs. This bit must be set prior to writing the new address to the Tx FIFO or DTR.
27	TXAK	Read/Write	0	**Transmit Acknowledge Enable.** This bit specifies the value driven onto the SDA line during acknowledge cycles for both Master and Slave receivers. "1" - ACK bit = "1" - no acknowledge. "0" - ACK bit = "0" - acknowledge. Because Master receivers indicate the end of data reception by not acknowledging the last byte of the transfer, this bit is the means to end a Master receiver transfer. As a slave this bit must be set prior to receiving the byte to not acknowledge.
28	TX	Read/Write	0	**Transmit/Receive Mode Select.** This bit selects the direction of Master/Slave transfers. "1" selects an OPB IIC transmit. "0" selects an OPB IIC receive. This bit does not control the Read/Write bit that is sent on the bus with the address. The Read/Write bit that is sent with an address must be the LSB of the address written into the transmit FIFO.

Table 5: **OPB IIC Control Register Bit Definitions *(Contd)***

Bit(s)	Name	Core Access	Reset Value	Description
29	MSMS	Read/Write	0	**Master/Slave Mode Select.** When this bit is changed from "0" to "1", the OPB IIC Bus Interface generates a START condition in Master mode. When this bit is cleared, a STOP condition is generated and the OPB IIC Bus Interface switches to Slave mode. When this bit is cleared by the hardware, because arbitration for the bus has been lost, a STOP condition is not generated.
30	Tx FIFO Reset	Read/Write	0	**Transmit FIFO Reset.** This bit must be set if arbitration is lost or if a transmit error occurs to flush the FIFO. "1" resets the Transmit FIFO. "0" Transmit FIFO normal operation.
31	EN	Read/Write	0	**OPB IIC Enable.** This bit must be set before any other CR bits have any effect. "1" enables the OPB IIC controller. "0" resets and disables the OPB IIC controller.

Status Register (SR)

This register contains the status of the OPB IIC Bus Interface. The read-only SR register is shown in Figure 4. All bits are cleared upon reset.

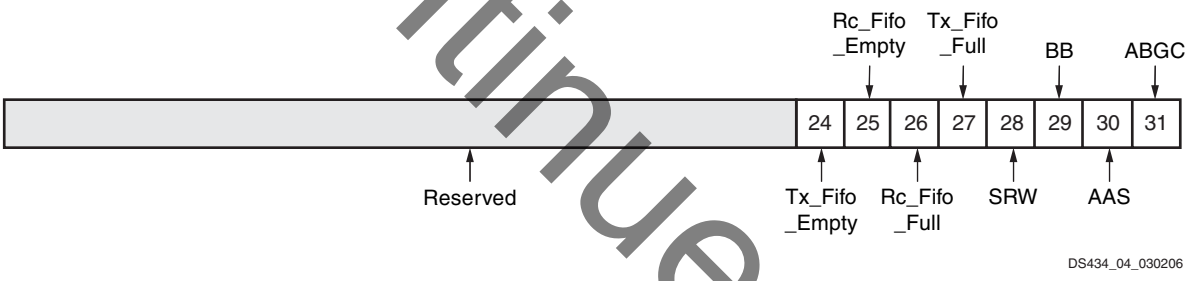

Figure 4: **Status Register (SR)**

Table 6: **Status Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 23	N/A	N/A	N/A	**Reserved**.
24	Tx_FIFO_ Empty	Read	1	**Transmit FIFO empty.** This bit will be set high when the transmit FIFO is empty.
25	Rc_FIFO_ Empty	Read	1	**Receive FIFO empty.** This bit will be set high when the receive FIFO is empty.
26	Rc_FIFO_ Full	Read	0	**Receive FIFO full.** This bit will be set high when the receive FIFO is full. This bit will only be set when all sixteen location in the FIFO are full, regardless of the value written into Rc_FIFO_PIRQ.
27	Tx_FIFO_ Full	Read	0	**Transmit FIFO full.** This bit will be set high when the transmit FIFO is full.

Table 6: **Status Register Bit Definitions *(Contd)***

Bit(s)	Name	Access	Reset Value	Description
28	SRW	Read	0	**Slave Read/Write.** When the IIC Bus Interface has been addressed as a Slave (AAS is set), this bit indicates the value of the read/write bit sent by the Master. This bit is only valid when a complete transfer has occurred and no other transfers have been initiated. "1" indicates Master reading from Slave. "0" indicates Master writing to Slave.
29	BB	Read	0	**Bus Busy.** This bit indicates the status of the IIC bus. This bit is set when a START condition is detected and cleared when a STOP condition is detected. "1" indicates the bus is busy. "0" indicates the bus is idle.
30	AAS	Read	0	**Addressed as Slave.** When the address on the IIC bus matches the Slave address in the Address Register (ADR), the IIC Bus Interface is being addressed as a Slave and switches to Slave mode. If 10-bit addressing is selected this device will only respond to a 10-bit address or general call if enabled. This bit is cleared when a stop condition is detected or a repeated start occurs. "1" indicates being addressed as a slave. "0" indicates not being addressed as a slave.
31	ABGC	Read	0	**Addressed By a General Call.** This bit is set high when another master has issued a general call and the general call enable bit is set high, CR(1) = '1'.

Transmit FIFO

This FIFO contains data to be transmitted on the OPB IIC bus. In transmit mode, data written into this FIFO is output on the IIC bus. Reading of this location will result in reading the current byte being output from the FIFO. Attempting to write to a full FIFO is not recommended and results in that data byte being lost. The Transmit FIFO is shown in Figure 5 and described in Table 7.

Figure 5: **Transmit FIFO**

Table 7: **OPB IIC Transmit FIFO Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 23				Reserved
24 - 31	D0 - D7	Read/Write	Indeterminate(1)	OPB IIC Transmit Data

Notes:
1. The value that was available before the reset occurred will still appear on the FIFO outputs

Receive FIFO

This FIFO contains the data received from the OPB IIC bus. The received OPB IIC data is placed in this FIFO after each complete transfer. The Receive FIFO Occupancy Register (Rc_FIFO_OCY) must be equal to the Programmable Read Depth Register (Rc_FIFO_PIRQ) before throttling will occur. The receive FIFO is read only. Reading this FIFO when it is empty results in indeterminate data being read. The Receive FIFO is shown in Figure 6 and described in Table 8.

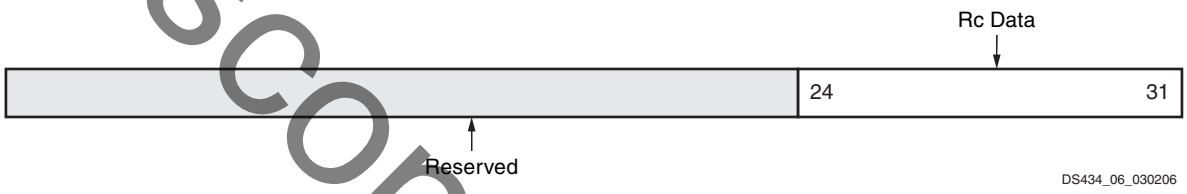

Figure 6: **Receive FIFO**

Table 8: **OPB IIC Receive FIFO Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 23				Reserved
24 - 31	D0 - D7	Read	Indeterminate(1)	OPB IIC Receive Data

Notes:
1. The value that was available before the reset occurred will still appear on the FIFO outputs

Address Register (ADR)

This field contains the specific Slave address (7-bit address) to be used by the OPB IIC Bus Interface. The read/write address register is shown in Figure 7 and described in Table 9.

Figure 7: **Address Register (ADR)**

Table 9: **Address Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 23	Reserved			Reserved.
24-30	Slave Address	Read/Write	0x00	Address used by the IIC Bus Interface when in Slave mode
31	Reserved			Reserved.

Ten Bit Address Register (TEN_ADR)

This field contains the specific top three most significant bits (MSBs) of the 10-bit Slave address to be used by the OPB IIC Bus Interface. This register is read/write. This register exists only if 10-bit addressing has been selected. The TEN_ADR register is shown in Figure 8 and described in Table 10.

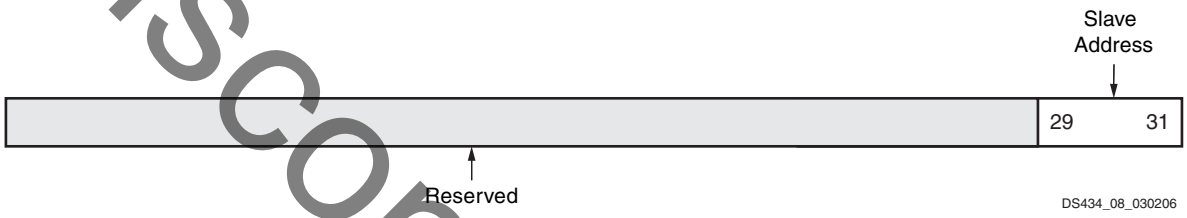

Figure 8: **Ten Bit Address Register (TEN_ADR)**

Table 10: **Ten Bit Address Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 28	Reserved			Reserved.
29 - 31	MSB of Slave Address	Read/Write	0x0	MSBs of the 10-bit Address used by the OPB IIC Bus Interface when in Slave mode.

Transmit FIFO Occupancy Register (Tx_FIFO_OCY)

This field contains the occupancy value for the Transmit FIFO. Reading this register cannot be used to determine if the FIFO is empty, the Transmit FIFO Empty Interrupt conveys that information. The value read is the binary count value, therefore reading all zeros implies that the first location is filled and reading all ones implies that all sixteen locations are filled. The Tx_FIFO_OCY register is shown in Figure 9 and described in Table 11.

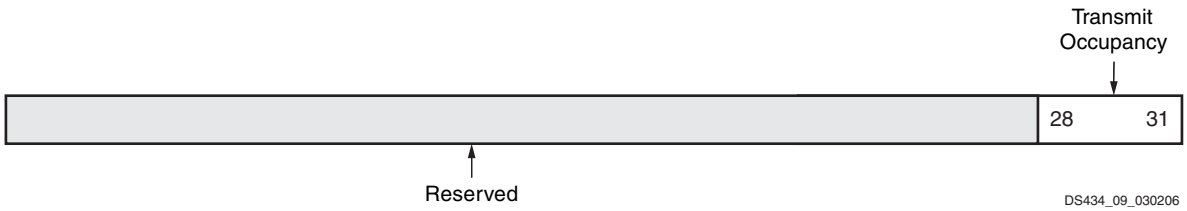

Figure 9: **Transmit FIFO Occupancy Register (Tx_FIFO_OCY)**

Table 11: **Transmit FIFO Occupancy Register Bit Definitions**

Bit(s)	Name	Access	Reset Value	Description
0 - 27	Reserved			Reserved.
28 -31	Occupancy Value	Read	0x0	Bit 28 is the MSB. A value of 1001 implies that 10 locations in the FIFO are full.

Receive FIFO Occupancy Register (Rc_FIFO_OCY)

This field contains the occupancy value for the Receive FIFO. This register is read only. Reading this register cannot be used to determine if the FIFO is empty, Rc_FIFO_Empty, bit(1) in the status register is used to convey that information. The value read is the binary count value, therefore reading all zeros implies that the first location is filled and reading all ones implies that all sixteen locations are filled. The Rc_FIFO_OCY register is shown in Figure 10 and described in Table 12.

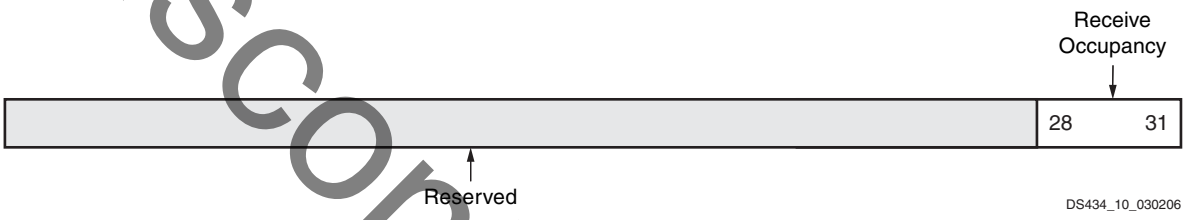

Figure 10: **Receive FIFO Occupancy Register (Rc_FIFO_OCY)**

Table 12: **Receive FIFO Occupancy Register Bit Definition**

Bit(s)	Name	Access	Reset Value	Description
0 - 27	Reserved			Reserved
28 - 31	Occupancy Value	Read	0x0	Bit 28 is the MSB. A value of 1001 implies that 10 locations in the FIFO are full.

Receive FIFO Programmable Depth Interrupt Register (Rc_FIFO_PIRQ)

This field contains the value which will cause the receive FIFO Interrupt to be set. When this value is equal to the Rc_FIFO_OCY value, the receive FIFO interrupt will be set and remain set until the equality is no longer true. A read from the receive FIFO will cause the IIC Receive FIFO Interrupt to be cleared. When the Rc_FIFO_PIRQ is equal to the Rc_FIFO_OCY throttling will also occur to prevent the transmitter from transmitting. The read/write Rc_FIFO_PIRQ register is shown in Figure 11 and described in Table 13.

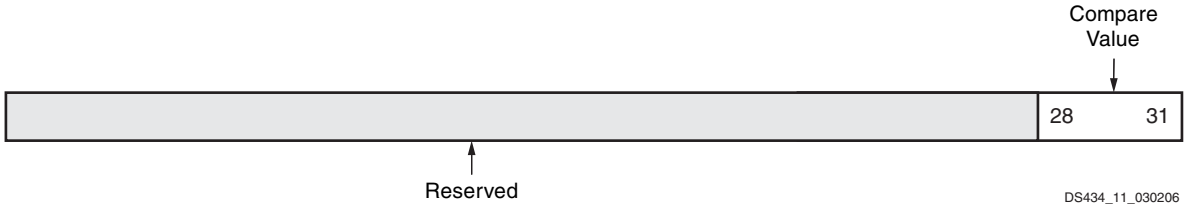

Figure 11: **Receive FIFO Programmable Depth Interrupt Register (Rc_FIFO_PIRQ)**

Table 13: **Receive FIFO Programmable Depth Interrupt Register Bit Definition**

Bit(s)	Name	Access	Reset Value	Description
0 - 27	Reserved			
28 - 31	Compare Value	Read/Write	0x0	Bit 28 is the MSB. A value of 1001 implies when ten locations in the receive FIFO have been filled the receive FIFO interrupt will be set.

General Purpose Output Register (GPO)

If C_GPO_WIDTH is equal to one, the only bit populated in the register is GPO(31), the LSB. If C_GPO_WIDTH is equal to 8, then bits 24 through 31 in the GPO are populated. This register is read/write and can be used to set memory protection, or for any other general output usage. The GPO register is shown in Figure 12 and described in Table 14.

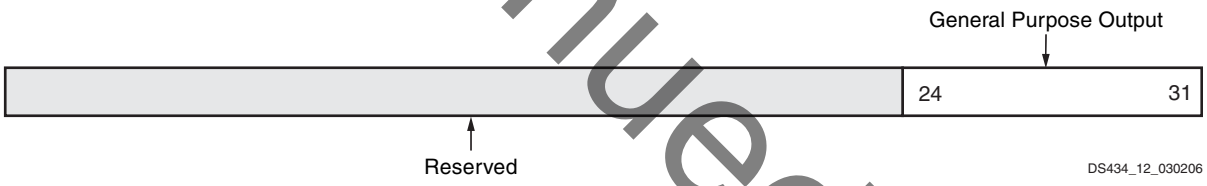

Figure 12: **General Purpose Output Register (GPO)**

Table 14: **General Purpose Output Register Bit Definition**

Bit(s)	Name	Access	Reset Value	Description
(32 - C_GPO_WIDTH) to (C_OPB_DWIDTH - 1)	General Purpose Outputs	Read/Write	0x0	GPO -- The LSB, bit 31 will be the first bit populated.

OPB IIC Interrupt Descriptions

Interrupts

The interrupt signals generated by the OPB IIC are managed by the Interrupt Source Controller in the IIC IPIF module. This interface provides many of the features commonly provided for interrupt handling. The IPIER and IPISR contain the bit mapping as shown in Figure 13. See the *Processor IP Reference Guide* under Part 1: Embedded Processor IP, under IPIF, under OPB IPIF Architecture, under OPB IPIF Register Descriptions for a complete description of the GIE, IPISR and IPIER. The OPB IIC has

eight unique interrupts that are sent to the IPIF. The number in the parenthesis is the IPIF interrupt bit number.

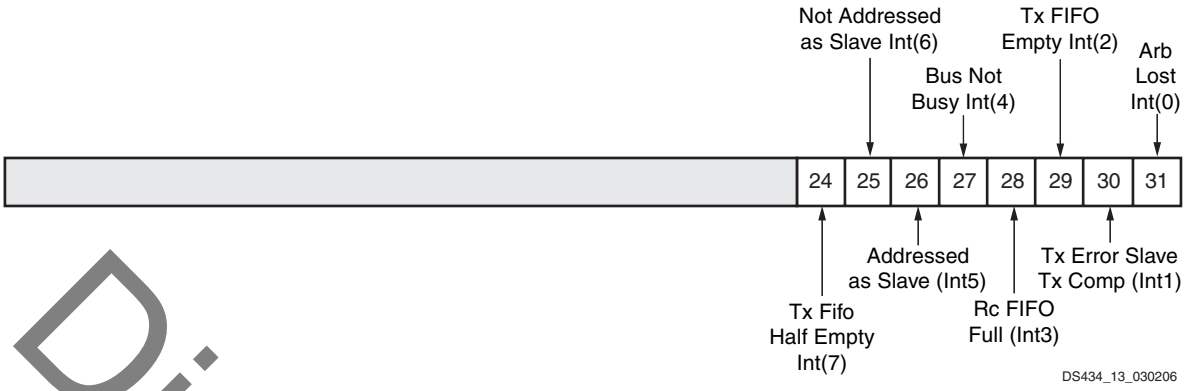

Figure 13: **Interrupt Mapping**

Interrupt(0) -- Arbitration Lost

Interrupt(0) is the Arbitration Lost interrupt. When arbitration for the OPB IIC bus is lost this interrupt is set. The only action required by software to clear this interrupt condition is to clear the IPIF interrupt register.

Interrupt(1) -- Transmit Error/Slave Transmit Complete

Interrupt(1) is the Transmit Error/Slave Transmit Complete. The only action required by software to clear this interrupt condition is to clear the IPIF interrupt register. The following describes the conditions that will cause this interrupt to occur:

IIC operating as a Master Transmitter, Interrupt(1) implies an error. The slave receiver did not acknowledge properly. The MSMS bit in the Control Register will remain high and the IIC will initiate a stop condition which implies that the bus will not be busy.

IIC operating as a Master Receiver, Interrupt(1) implies a transmit complete. This interrupt is caused by setting TXAK high, in the IIC, to indicate to the slave transmitter that the last byte has been transmitted.

IIC operating as a Slave Transmitter, Interrupt(1) implies a transmit complete. This interrupt is caused by the master device to indicate to the IIC that the last byte has been transmitted.

IIC operating as a Slave Receiver, Interrupt(1) implies an error. This interrupt is caused by the IIC (setting TXAK high).

Interrupt(2) -- Transmit FIFO Empty

Interrupt(2) is the Transmit FIFO Empty interrupt. This interrupt is generated only on a transmit throttle condition. Clearing this interrupt requires data to be written to the transmit FIFO or Data Transmit Register. This interrupt will be set and remain set as long as the transmit throttle condition exists.

Interrupt(3) -- Receive FIFO Full

Interrupt(3) is the Receive FIFO Full interrupt or Data Receive Register Full interrupt. This interrupt will be set and remain set as long the Rc_FIFO_PIRQ is equal to the Rc_FIFO_OCY. Clearing this interrupt requires that the Data Receive FIFO be read.

Interrupt(4) -- IIC Bus is Not Busy

Interrupt(4) is the IIC Bus is Not Busy. Interrupt(4) is set and will remain set as long as the IIC bus is not busy.

Interrupt(5) -- Addressed As Slave

Interrupt(5) is the Addressed as Slave interrupt. This interrupt will be set and remain set as long as the IIC is being addressed as a slave.

Interrupt(6) -- Not Addressed As Slave

Interrupt (6) is the Not Addressed As Slave interrupt. This interrupt allows the detection of the end of receive data for a slave receiver when there has been no stop condition (repeated start). The interrupt occurs when a start condition followed by a non-matching slave address is detected. This interrupt will be set whenever the IIC is not addressed as a slave.

Interrupt(7) -- Transmit FIFO Half Empty

Interrupt(7) is the Transmit FIFO half empty. This interrupt will be set and remain set as long as the MSB of the Tx_FIFO_OCY = 0.

Flow Description

The following is a brief discussion on setting the IIC registers in order to initiate and complete bus transactions. Similar procedures must be completed in the order shown to successfully complete an OPB IIC transaction, depending upon the users application.

IIC Master Transmitter, with a repeated start.

1. Write the IIC device address to the Tx FIFO.
2. Write data to TX FIFO.
3. Write to CR to set MSMS = 1 and TX = 1.
4. Continue writing all data to TX FIFO.
5. Wait for Transmit FIFO empty interrupt. This implies the IIC has throttled the bus.
6. Write to CR to set RSTA = 1.
7. Write IIC device address to TX FIFO.
8. Write all data except last byte to TX FIFO.
9. Wait for Transmit FIFO empty interrupt. This implies the IIC has throttled the bus.
10. Write to CR to set MSMS = 0. The IIC will generate a stop condition at the end of the last byte.
11. Write last byte of data to TX FIFO.

IIC Master Receiver, with a repeated start.

1. Write the IIC device address to the Tx FIFO, Write the programmable read depth interrupt to be 1 less then the total message length.
2. Write to CR to set MSMS = 1 and TX = 0.
3. Wait for the Receive FIFO interrupt.
4. Write to CR to set TXAK = 1, This will prevent the IIC from acknowledging the next byte received and indicate to the slave transmitter that it has transmitted the final byte of the message. Write the programmable read depth interrupt to be 1. It is important to do this prior to reading data from the FIFO, because as soon a read from the FIFO has occurred, the throttle condition will be removed and the control information must be set up to control the next byte.
5. Read all data from receive FIFO.

6. Wait for the Receive FIFO interrupt.
7. Read data from receive FIFO, Write the programmable read depth interrupt to be 1 less then the total message length.
8. Write to CR to set RSTA = 1. Write IIC device address to TX FIFO.
9. Wait for the Receive FIFO interrupt.
10. Write to CR to set TXAK = 1, and set MSMS = 0. This will prevent the IIC from acknowledging the next byte received and indicate to the slave transmitter that it has transmitted the final byte of the message and also generate a stop condition. Write the programmable read depth interrupt to be 1.
11. Wait for the Receive FIFO interrupt.
12. Read data from receive FIFO.

IIC Slave Receiver.

1. Write to the Control Register to enable the IIC, EN = 1. If the IIC needs to recognize a general call then set EN = 1 and GC_EN = 1.
2. Write the Slave address to the ADR. In seven bit mode, a slave address of 0x7F should be written as 0xFE to the ADR.
3. Write the value to the Receive FIFO interrupt register 0x0 will cause an interrupt when 1 byte of data (not address) has been received. Because the address transmitted on the IIC bus is not stored in the receive FIFO, this interrupt will not be caused by receiving either a seven bit address or a ten bit address.
4. Wait for addressed as slave interrupt AAS.
5. Once AAS has occurred, Determine if the IIC slave is to receive or transmit data by reading the Status Register bit 4
6. Clear not addressed as slave interrupt NAS.
7. If the IIC is a slave receiver then the following interrupt processing is available for use.
8. There are two basic choices for the slave receiver interrupt processing.
   a. Set the Receive FIFO interrupt register to 0x0 and wait for either a Not Addressed as Slave NAS (no data was sent) or Rc_FIFO_PIRQ interrupt. In this mode an interrupt will occur for every byte of data received plus a NAS for the end of the transmission.
   b. Set the Receive FIFO interrupt register to 0xF and wait for either a Not Addressed as Slave NAS (some amount of data less than 16 bytes was set) or Rc_FIFO_PIRQ interrupt. In this mode if the Rc_FIFO_PIRQ interrupt occurs then 16 bytes of data exists in the FIFO to handle (Probably best to read the Receive FIFO occupancy register though not required). NAS may occur without Rc_FIFO_PIRQ interrupt. That means the Receive FIFO occupancy register should be read to indicate how many bytes of data must be handled. In this mode there will be one Rc_FIFO_PIRQ interrupt for every 16 bytes of data plus a NAS for the end of the transmission. If less than 16 bytes of data is sent then NAS will be the only interrupt.
9. In either choice above, the active interrupts need to be cleared after the data has been handled and then wait for the next interrupt.
10. Once the NAS interrupt has been received, handle the data and clear AAS.
11. Wait for AAS

IIC Slave Transmitter.

1. If the IIC is a slave transmitter then the following interrupt processing is available for use.
2. Ensure the Transmit Error/Slave Transmit Complete interrupt is cleared.

3. Once the IIC has been addressed as a slave transmitter the IIC will transmit the first byte of data in the Transmit FIFO. If no data exists in the transmit FIFO then the IIC will throttle the bus until data is written into the transmit FIFO.
4. If the protocol allows knowledge as to how much data the slave must transmit then fill up the FIFO and use the Transmit FIFO Empty or Transmit FIFO Half Empty interrupts to keep the transmit FIFO full. Wait for the Transmit Error/Slave Transmit Complete interrupt.
5. It is also possible to write one byte of data to the FIFO at a time and wait for either a Transmit FIFO empty interrupt which means the master wants more data, or for the Transmit Error/Slave Transmit Complete interrupt which indicates the master has receive all required data.
6. Once the Transmit Error/Slave Transmit Complete has occurred the NAS should also occur because the master has to either send a stop or a repeated start.
7. Once the NAS interrupt has been received, handle the data and clear AAS.

- Wait for AAS

Throttle Description

The hardware will throttle the bus to prevent either a receive overrun or a transmit underflow. The throttling is done by holding the SCL line low after the acknowledge bit has been sent.

Throttling is independent of Master or Slave operation and only depends upon transmit or receive operation. However, the Master must generate repeated start or stop conditions and that information must be available to the IIC prior to the final byte being transmitted.

Transmit throttling is done whenever Interrupt(2), Transmit FIFO Empty is set. As soon as a new byte is written into the Data Transmit FIFO, the throttle condition is removed and that byte will be transmitted onto the IIC bus.

If that byte is intended to be an address of another device, the repeated start bit must be set prior to writing the address into the Tx FIFO. This allows the IIC to generate a repeated start condition on the bus before transmitting the address of the new device.

Receive throttling is done whenever the Rc_FIFO_PIRQ is set. The throttle condition is removed when a byte from the Receive FIFO is read.

When the IIC is a master and a receive throttle condition exists the IIC will generate a stop condition if the MSMS bit is change from a 1 to a 0. This allows single byte reads from a slave device. The transmit acknowledge must be set up to no-ack the byte. This will command the slave device to release the SDA line so a stop condition can occur.

When the IIC is a master and in a receive throttle condition and the transmit FIFO is empty. Setting the repeated start in the Control Register will cause a transmit throttle condition to occur.

Design Implementation

Design Tools

Xilinx's ISE is the synthesis, map and place and route tool used for the OPB IIC design.

Target Technology

The intended target technology is Virtex and Spartan FPGA families.

Device Utilization and Performance Benchmarks

Because the OPB IIC is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the OPB IIC is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the OPB IIC design will vary from the results reported here.

To analyze the IIC timing within the FPGA, a design was created that instantiated the OPB IIC with the following parameters set.

The OPB IIC benchmarks for a Virtex-II Pro XC2VP7-6 FF672 FPGA are shown in Table 15.

Table 15: **FPGA Performance and Resource Utilization Benchmarks**

Parameter Values (For Example)			Device Resources				
C_IIC_FREQ	C_TEN_BIT_ADR	C_GPO_WIDTH	Slices	Slice Flip-Flops	4-input LUTs	GCLKs	f max (MHz)
100000	0	0	268	205	353	1	166.389
400000	0	0	284	203	356	1	167.001
400000	1	0	292	210	374	1	161.001
400000	0	1	272	205	359	1	167.392
400000	0	8	276	207	367	1	162.92

Notes:

Specification Exceptions

Exceptions to the Philips IIC-bus specification version 2.1 January 2000

High-speed mode (Hs-mode) is not currently supported by the OPB IIC IP.

3-state buffers are used to perform the wired-AND function inherent in this bus structure.

The Xilinx FPGA device ratings must not be exceeded when inter-connecting the OPB IIC to other devices.

Reference Documents

Philips IIC-bus specification version 2.1 January 2000

Revision History

Date	Version	Revision
11/11/04	1.0	Initial Xilinx release
1/19/05	1.1	Converted to new DS template; updated images to Xilinx graphic standards; reformatted tables.
4/15/05	1.1.1	Updated Figure 2 to show outline of Tx and Rc FIFO as solid lines.
5/3/05	1.1.2	Updated to incorporate CR207083; put/published 050305.
11/11/05	1.2	In Table 2, moved footnote (1) ref from Scl_O (P5) to Scl_T (P6) description; in table footnote 1, *Scl_T* was *Sc1_T*.
12/1/05	1.3	Added support for Spartan-3E; removed URL to Philips on pg 2 per CR220923.
3/2/06	1.4	Added Table 6, Status Register Bit Definitions; in Figure 11, Rc_FIFO_PIRQ Register, Compare Value was 29:31; updated figures to Xilinx Graphic Standard.
5/24/06	1.5	Entered values for Resources Used in LogiCORE Facts table.