LogiCORE IP AXI UART Lite (v1.02a)

DS741 July 25, 2012 **Product Specification**

Introduction

The LogiCORE™ IP AXI Universal Asynchronous Receiver Transmitter (UART) Lite interface connects to the Advanced Microcontroller Bus Architecture (AMBA®) specification's Advanced eXtensible Interface (AXI) and provides the controller interface for asynchronous serial data transfer. This soft LogiCORE IP core is designed to interface with the AXI4-Lite protocol.

Features

- AXI interface is based on the AXI4-Lite specification
- One transmit and one receive channel (full duplex)
- 16-character transmit and receive FIFOs
- Configurable number of data bits (5-8) in a character
- Configurable parity bit (odd or even or none)
- Configurable baud rate

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family [1]	Zynq™-7000[2], Virtex®-7[3], Kintex™-7[3], Artix™-7[3], Virtex-6[4], Spartan®-6[5],
Supported User Interfaces	AXI4-Lite
Resources	See Table 9, Table 10, Table 10, Table 11, Table 12, and Table 13
Provided with Core	
Design Files	ISE®: VHDL Vivado™: VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	None
Simulation Model	None
Supported S/W Drivers[6]	Standalone and Linux
Tested Design Flows[7]	
Design Entry	Xilinx Platform Studio (XPS) Vivado™ Design Suite[8]
Simulation	Mentor Graphics ModelSim
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
Support	
Provided by Xilinx@ www.xilinx.com/support	

Notes:

1. For a complete list of supported derivative devices, see the Embedded Edition Derivative Device Support.
2. Supported in ISE Design Suite implementations only.
3. For more information, see DS180 *7 Series FPGAs Overview.*
4. For more information, see the DS150 *Virtex-6 Family Overview Product Specification.*
5. For more information, see DS160 *Spartan-6 Family Overview Product Specification.*
6. Standalone driver information can be found in the EDK or SDK installation directory.

   See xilinx_drivers.htm in <install_directory>/doc/usenglish. Linux OS and driver support information is available from wiki.xilinx.com.
7. For a listing of the supported tool versions, see the Xilinx Design Tools: Release Note Guide.
8. Supports only 7 series devices.

Functional Description

The AXI UART Lite:

- Performs parallel-to-serial conversion on characters received through the AXI4-Lite interface and serial-to-parallel conversion on characters received from a serial peripheral.
- Can transmit and receive 8, 7, 6, or 5 bit characters, with 1 stop bit and with odd, even, or no parity bit. The AXI UART Lite can transmit and receive independently.
- Can be configured and its status can be monitored through the internal register set.
- Generates an interrupt when the receive FIFO becomes non empty or when the transmit FIFO becomes empty. This interrupt can be masked by using an interrupt enable/disable signal. The device contains a baud rate generator and independent 16-character deep transmit and receive FIFOs.

The AXI UART Lite modules are shown in Figure 1 and described in the sections that follow.

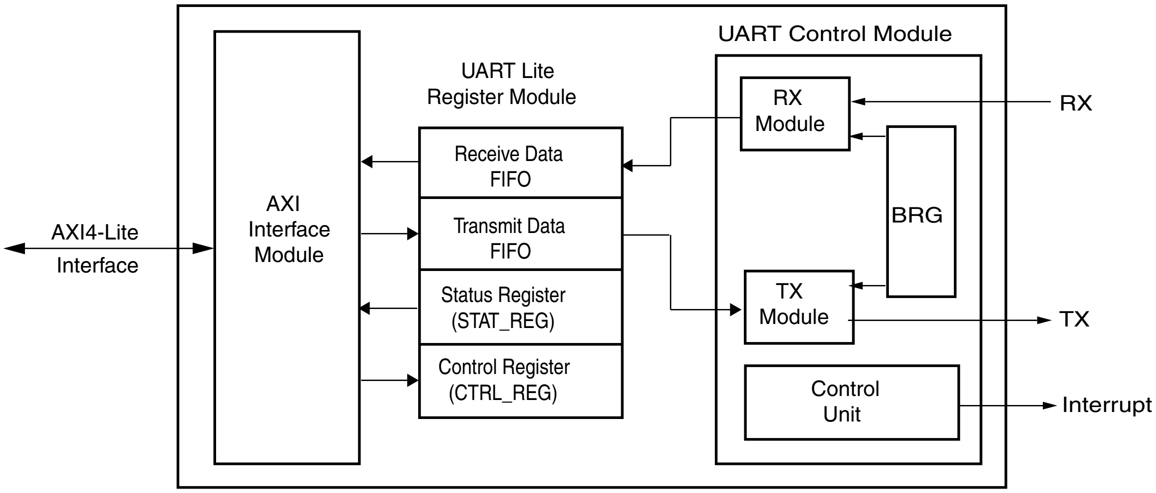

Figure 1: **Block Diagram of AXI UART Lite**

AXI Interface Module: Provides the interface to the AXI and implements AXI protocol logic. The AXI interface module is a bidirectional interface between a user IP core and the AXI4-Lite interface standard. To simplify the process of attaching AXI UART Lite to the AXI, the core makes use of a portable, pre-designed AXI interface called AXI-4-Lite IPIF, that takes care of the AXI interface signals.

UART Lite Register Module: Includes all memory mapped registers (as shown in Figure 1). It interfaces to the AXI through the AXI interface module. It consists of a status register, a control register, and a pair of transmit/receive FIFOs, both of 16-character depth. All registers are accessed directly from the AXI using the AXI interface module.

UART Control Module: Consists of an RX module, a TX module, a parameterized baud rate generator (BRG), and a control unit. This module also contains the logic to generate the interrupts.

Interrupts

If interrupts are enabled, an edge rising sensitive interrupt pulse is generated when one of the following is true:

1. When the Receive FIFO goes from empty to not empty, such as when the first valid character is received in the Receive FIFO
2. When the transmit FIFO goes from not empty to empty, such as when the last character in the transmit FIFO is transmitted.

If interrupts are disabled in between reset the RX FIFO to get the Valid Character receives interrupt.

I/O Signals

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

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

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	S_AXI_ACLK	System	Input	-	AXI clock.
P2	S_AXI_ARESETN	System	Input	-	AXI reset, active-Low.
P3	Interrupt	System	Output	0x0	Edge rising UART interrupt.
AXI Write Address Channel Signals					
P4	S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0]	AXI	Input	-	AXI write address. The write address bus gives the address of the write transaction.
P5	S_AXI_AWVALID	AXI	Input	-	Write address valid. This signal indicates that valid write address is available.
P6	S_AXI_AWREADY	AXI	Output	0x0	Write address ready. This signal indicates that the slave is ready to accept an address.
AXI Write Channel Signals					
P7	S_AXI_WDATA[C_S_AXI_DATA_WIDTH - 1: 0]	AXI	Input	-	Write data.
P8	S_AXI_WSTB[C_S_AXI_DATA_WIDTH/8-1:0] [(1)]	AXI	Input	-	Write strobes. This signal indicates which byte lanes to update in memory. [(1)]
P9	S_AXI_WVALID	AXI	Input	-	Write valid. This signal indicates that valid write data and strobes are available.
P10	S_AXI_WREADY	AXI	Output	0x0	Write ready. This signal indicates that the slave can accept the write data.

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

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Write Response Channel Signals					
P11	S_AXI_BRESP[1:0]	AXI	Output	0x0	Write response. This signal indicates the status of the write transaction. “00“ - OKAY “10“ - SLVERR “11“ - DECERR (Not generated in the core)
P12	S_AXI_BVALID	AXI	Output	0x0	Write response valid. This signal indicates that a valid write response is available.
P13	S_AXI_BREADY	AXI	Input	-	Response ready. This signal indicates that the master can accept the response information.
AXI Read Address Channel Signals					
P14	S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH -1:0]	AXI	Input	-	Read address. The read address bus gives the address of a read transaction.
P15	S_AXI_ARVALID	AXI	Input	-	Read address valid. This signal indicates, when High, that the read address is valid and will remain stable until the address acknowledgement signal, S_AXI_ARREADY, is high.
P16	S_AXI_ARREADY	AXI	Output	0x1	Read address ready. This signal indicates that the slave is ready to accept an address.
AXI Read Data Channel Signals					
P17	S_AXI_RDATA[C_S_AXI_DATA_WIDTH -1:0]	AXI	Output	0x0	Read data
P18	S_AXI_RRESP[1:0]	AXI	Output	0x0	Read response. This signal indicates the status of the read transfer. “00“ - OKAY “10“ - SLVERR “11“ - DECERR (Not generated in the core)
P19	S_AXI_RVALID	AXI	Output	0x0	Read valid. This signal indicates that the required read data is available and the read transfer can complete.
P20	S_AXI_RREADY	AXI	Input	-	Read ready. This signal indicates that the master can accept the read data and response information.
UART Lite Interface Signals					
P21	RX	UART Lite	Input	-	Receive data
P22	TX	UART Lite	Output	0x1	Transmit data

Notes:

1. This signal is not used. The AXI UART Lite assumes that all byte lanes are active.

Design Parameters

To allow you to create the AXI UART Lite that is uniquely tailored for your system, certain features can be parameterized in the AXI UART Lite design. This allows you to have a design that only utilizes the resources required by the system and operating at the best possible performance. The AXI UART Lite design parameters are shown in Table 2.

Table 2: **Design Parameters**

Generic	Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameters					
G1	Target FPGA family	C_FAMILY	virtex7, artix7, kintex7, zynq, virtex6, spartan6	virtex6	string
G2	System clock frequency (in Hz) driving the UART Lite peripheral	C_S_AXI_ACLK_FREQ_HZ	integer (ex. 100000000)	100_ 000_ 000	integer
AXI Parameters					
G3	AXI address bus width	C_S_AXI_ADDR_WIDTH	4	4	integer
G4	AXI data bus width	C_S_AXI_DATA_WIDTH	32	32	integer
G5	AXI interface type	C_S_AXI_PROTOCOL	AXI4LITE	AXI4LITE	string
UART Lite Parameters					
G6	Baud rate of the UART Lite in bits per second	C_BAUDRATE	integer (ex. 128000)	9600[1]	integer
G7	The number of data bits in the serial frame	C_DATA_BITS	5 - 8	8	integer
G8	Determines whether parity is used or not	C_USE_PARITY	0 = Do not use parity 1 = Use parity	0	integer
G9	If parity is used, determines whether parity is odd or even	C_ODD_PARITY	0 = Even parity 1 = Odd parity	0	integer

Notes:

1. With a baud rate of 115200, the sample clock is 16 * 115200 = 1.8432 MHz. With the System clock C_S_AXI_ACLK_FREQ_HZ running at 10 MHz, the integer ratio for driving the sample clock is 5 (rounding of [10/1.8432]). The AXI UART Lite would then divide the System clock by 5 resulting in 2 MHz for the sample clock. The baud rate error is (1.8432 - 2) /1.8432 => -8.5% which is outside the tolerance for most UARTs. The issue is that the higher the baud rate and the lower the C_S_AXI_ACLK_FREQ_HZ, the greater the error in the generated baud rate of the AXI UART Lite. AXI UART Lite requires that the baud error should be less than 3%.

Dependencies between Parameters and I/O Signals

The width of some of the AXI UART Lite signals depends on parameters selected in the design. The dependencies between the AXI UART Lite core design parameters and I/O signals are described in Table 3.

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

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G3	C_S_AXI_ADDR_WIDTH	P4, P14	-	Defines the width of the ports
G4	C_S_AXI_DATA_WIDTH	P7, P8, P17	-	Defines the width of the ports
I/O Signals				
P4	S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0]	-	G3	Port width depends on the generic C_S_AXI_ADDR_WIDTH
P7	S_AXI_WDATA[C_S_AXI_DATA_WIDTH-1:0]	-	G4	Port width depends on the generic C_S_AXI_DATA_WIDTH
P8	S_AXI_WSTB[C_S_AXI_DATA_WIDTH/8-1:0]	-	G4	Port width depends on the generic C_S_AXI_DATA_WIDTH
P14	S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH -1:0]	-	G3	Port width depends on the generic C_S_AXI_ADDR_WIDTH
P17	S_AXI_RDATA[C_S_AXI_DATA_WIDTH -1:0]	-	G4	Port width depends on the generic C_S_AXI_DATA_WIDTH

Register Descriptions

Table 4 shows all the AXI UART Lite registers and their addresses.

Table 4: **Registers**

Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
0x0	Rx FIFO	Read[1]	0x0	Receive data FIFO[3]
0x4	Tx FIFO	Write[2]	0x0	Transmit data FIFO[3]
0x8	STAT_REG	Read[1]	0x4	UART Lite status register
0xC	CTRL_REG	Write[2]	0x0	UART Lite control register

Notes:

1. Writing of a read only register has no effect.
2. Reading of a write only register returns zero.
3. When system reset is applied both TX FIFO and RX FIFO are reset and cleared.

Receive Data FIFO

This 16-entry deep FIFO contains data received by AXI UART Lite. The FIFO bit definitions are shown in Table 5. Reading this register will result in reading the data word from the top of the FIFO. When a read request is issued to an empty FIFO, a bus error is generated and the result is undefined. The receive data FIFO is a read-only register. Issuing a write request to the receive data FIFO does nothing but generates a successful write acknowledgement. Figure 2 shows the location for data on the AXI when C_DATA_BITS is set to 8.

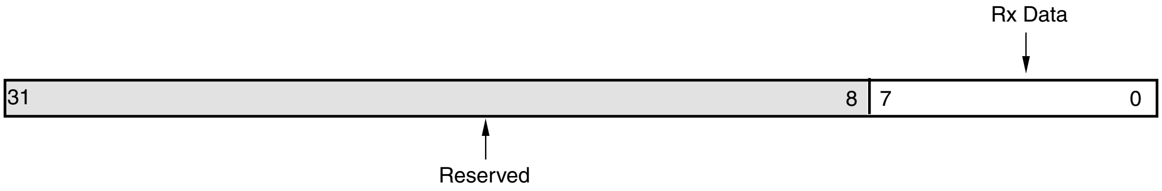

Figure 2: **Receive Data FIFO (C_DATA_BITS = 8)**

Table 5: **Receive Data FIFO Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-C_DATA_BITS	Reserved	N/A	0x0	Reserved
[C_DATA_BITS-1] - 0	Rx Data	Read	0x0	UART receive data

Transmit Data FIFO

This 16-entry deep FIFO contains data to be output by AXI UART Lite. The FIFO bit definitions are shown in Figure 3. Data to be transmitted is written into this register. When a write request is issued when the FIFO is full, a bus error (SLVERR) is generated and the data is not written into the FIFO. This is a write-only location. Issuing a read request to the transmit data FIFO generates the read acknowledgement with zero data. Table 6 shows the location for data on the AXI when C_DATA_BITS is set to 8.

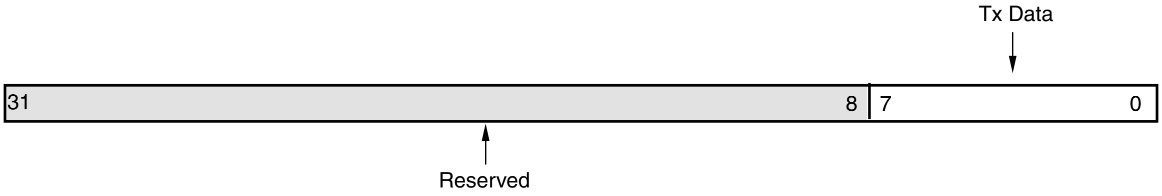

Figure 3: **Transmit Data FIFO (C_DATA_BITS = 8)**

Table 6: **Transmit Data FIFO Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-C_DATA_BITS	Reserved	N/A	0x0	Reserved
[C_DATA_BITS-1] - 0	Tx Data	Write	0x0	UART transmit data

Control Register (CTRL_REG)

The control register contains the enable interrupt bit and reset pin for the receive and transmit data FIFO. This is a write-only register. Issuing a read request to the control register generates the read acknowledgement with zero data. Figure 4 shows the bit assignment of CTRL_REG. Table 7 describes this bit assignment.

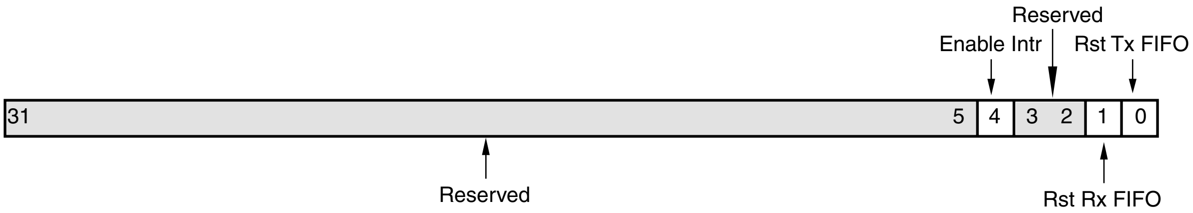

Figure 4: **Control Register**

Table 7: **Control Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31 - 5	Reserved	N/A	0x0	Reserved
4	Enable Intr	Write	0x0	Enable interrupt for the UART Lite '0' = Disable interrupt signal '1' = Enable interrupt signal
3 - 2	Reserved	N/A	0x0	Reserved
1	Rst Rx FIFO	Write	0x0	Reset/clear the receive FIFO Writing a '1' to this bit position clears the receive FIFO '0' = Do nothing '1' = Clear the receive FIFO
0	Rst Tx FIFO	Write	0x0	Reset/clear the transmit FIFO Writing a '1' to this bit position clears the transmit FIFO '0' = Do nothing '1' = Clear the transmit FIFO

Status Register (STAT_REG)

The status register contains the status of the receive and transmit data FIFOs, if interrupts are enabled, and if there are any errors. This is a read-only register. If a write request is issued to the status register, it will do nothing but generates a write acknowledgement. Bit assignment in the STAT_REG is shown in Figure 5 and described in Table 8.

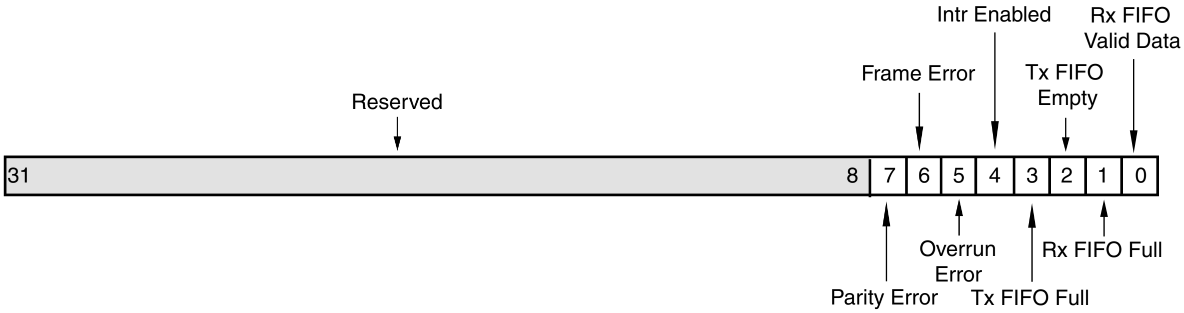

Figure 5: **Status Register**

Table 8: **Status Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31 - 8	Reserved	N/A	0x0	Reserved
7	Parity Error	Read	0x0	Indicates that a parity error has occurred after the last time the status register was read. If the UART is configured without any parity handling, this bit is always '0'. The received character is written into the receive FIFO. This bit is cleared when the status register is read '0' = No parity error has occurred '1' = Parity error has occurred
6	Frame Error	Read	0x0	Indicates that a frame error has occurred after the last time the status register was read. Frame error is defined as detection of a stop bit with the value '0'. The receive character is ignored and not written to the receive FIFO. This bit is cleared when the status register is read '0' = No frame error has occurred '1' = Frame error has occurred
5	Overrun Error	Read	0x0	Indicates that a overrun error has occurred after the last time the status register was read. Overrun is when a new character has been received but the receive FIFO is full. The received character is ignored and not written into the receive FIFO. This bit is cleared when the status register is read. '0' = No overrun error has occurred '1' = Overrun error has occurred
4	Intr Enabled	Read	0x0	Indicates that interrupts is enabled. '0' = Interrupt is disabled '1' = Interrupt is enabled
3	Tx FIFO Full	Read	0x0	Indicates if the transmit FIFO is full. '0' = Transmit FIFO is not full '1' = Transmit FIFO is full

Table 8: **Status Register Bit Definitions *(Cont'd)***

Bit(s)	Name	Core Access	Reset Value	Description
2	Tx FIFO Empty	Read	0x1	Indicates if the transmit FIFO is empty. '0' = Transmit FIFO is not empty '1' = Transmit FIFO is empty
1	Rx FIFO Full	Read	0x0	Indicates if the receive FIFO is full. '0' = Receive FIFO is not full '1' = Receive FIFO is full
0	Rx FIFO Valid Data	Read	0x0	Indicates if the receive FIFO has valid data. '0' = Receive FIFO is empty '1' = Receive FIFO has valid data

Response Signaling

The AXI UART Lite core generates SLVERR when one of the following conditions is true:

- A read request is issued to an empty receive data FIFO.
- A write request is issued when the transmit data FIFO is full.

For all other requests, OKAY response is passed. The AXI UART Lite never generates DECERR.

Design Implementation

Target Technology

The intended target technology is Zynq-7000, Virtex-7, Kintex-7, Artix-7, Virtex-6, and Spartan-6 FPGAs.

Device Utilization and Performance Benchmarks

Core Performance

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

Table 9: **Performance and Resource Utilization Benchmarks on Virtex-7 (xc7v855tffg1157-3)**

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	5	0	0	41	77	100	213
32	100_000_000	19_200	6	1	0	46	84	110	207
32	100_000_000	19_200	7	1	1	59	84	114	226
32	100_000_000	9600	8	0	0	49	79	119	244
32	40_000_000	38_400	8	0	0	44	77	107	234
32	100_000_000	19_200	6	1	0	46	84	110	207
32	100_000_000	19_200	7	1	1	59	84	114	226

The AXI UART Lite resource utilization for various parameter combinations measured with a Kintex-7 device as the target are detailed in Table 10.

Table 10: **Performance and Resource Utilization Benchmarks on Kintex-7[1] FPGA and Zynq-7000 Device**

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	5	0	0	42	77	100	200
32	100_000_000	19_200	6	1	0	44	84	111	204
32	100_000_000	19_200	7	1	1	47	84	114	225
32	100_000_000	9600	8	0	0	56	79	119	271
32	40_000_000	38_400	8	0	0	52	77	107	274
32	100_000_000	19_200	6	1	0	44	84	111	204
32	100_000_000	19_200	7	1	1	47	84	114	225

1. Kintex-7 FPGA (xc7k410tffg676-3)

The AXI UART Lite resource utilization for various parameter combinations measured with an Artix-7 device as the target are detailed in Table 11.

Table 11: **Performance and Resource Utilization Benchmarks on Artix-7[1] FPGA and Zynq-7000 Device**

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	5	0	0	42	77	92	233
32	100_000_000	19_200	6	1	0	45	84	102	224
32	100_000_000	19_200	7	1	1	47	84	111	225
32	100_000_000	9600	8	0	0	49	79	119	251
32	40_000_000	38_400	8	0	0	46	77	112	272
32	100_000_000	19_200	6	1	0	45	84	102	224
32	100_000_000	19_200	7	1	1	47	84	111	225

[1] Artix-7 FPGA (xc7a350tfbg676-3)

The AXI UART Lite resource utilization for various parameter combinations measured with a Virtex-6 device as the target are detailed in Table 12.

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

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	5	0	0	51	76	99	270
32	100_000_000	19_200	6	1	0	49	83	106	264
32	100_000_000	19_200	7	1	1	62	83	111	264
32	100_000_000	9600	8	0	0	52	78	121	250
32	40_000_000	38_400	8	0	0	49	76	107	242

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

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	6	1	0	49	83	106	264
32	100_000_000	19_200	7	1	1	62	83	111	264

The AXI UART Lite resource utilization for various parameter combinations measured with a Spartan-6 device as the target are detailed in Table 13.

Table 13: **Performance and Resource Utilization Benchmarks on Spartan-6 (XC6SLX45T-2-FGG484)**

Parameter Values (other parameters at default value)						Device Resources			Performance
C_S_AXI_AWIDTH	C_S_AXI_ACLK_FREQ_HZ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
32	100_000_000	19_200	5	0	0	52	76	97	168
32	100_000_000	19_200	6	1	0	63	83	107	147
32	100_000_000	19_200	7	1	1	58	83	109	158
32	100_000_000	9600	8	0	0	59	78	106	167
32	40_000_000	38_400	8	0	0	58	76	99	160
32	100_000_000	19_200	6	1	0	63	83	107	147
32	100_000_000	19_200	7	1	1	58	83	109	158

System Performance

To measure the system performance (F_{MAX}) of this core, this core was added to a Virtex-6 FPGA system and a Spartan-6 FPGA system as the device under test (DUT) as illustrated in Figure 6.

Because the AXI UART Lite core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the design's FPGA resources and timing usage will vary from the results reported here.

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: **System Performance**

Target FPGA	Target F_{MAX} (MHz)		
	AXI4	AXI4-Lite	MicroBlaze™
xc6slx45t [1]	90 MHz	120 MHz	80
xc6vlx240t [2]	135 MHz	180 MHz	135

Notes:

1. Spartan-6 FPGA LUT utilization: 60%; Block RAM utilization: 70%; I/O utilization: 80%; MicroBlaze Controller not AXI4 interconnect; AXI4 interconnect configured with a single clock of 120 MHz.
2. Virtex-6 FPGA LUT utilization: 70%; Block RAM utilization: 70%; I/O utilization: 80%.

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

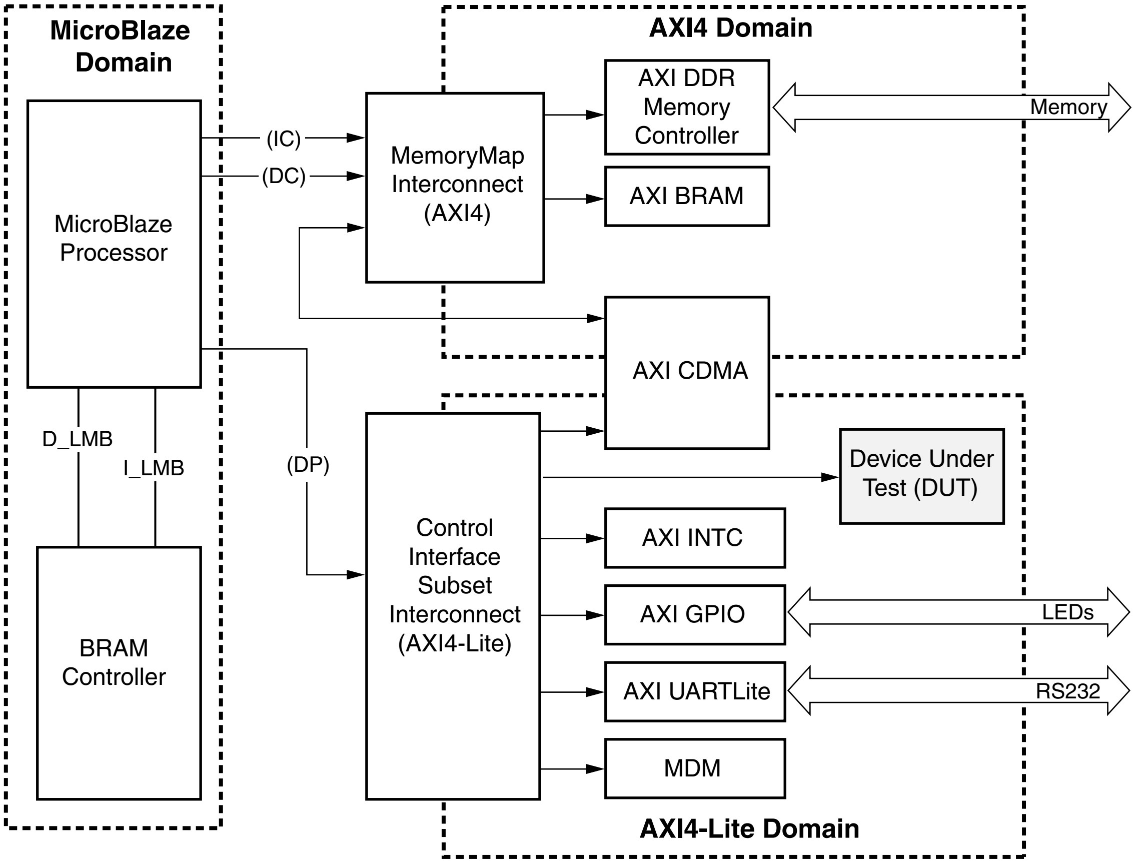

Figure 6: **Virtex-6 and Spartan-6 Devices FMAX Margin System**

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*.

Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx® Vivado Design Suite and Integrated Software Environment (ISE®) Design Suite Embedded Edition software 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 modules and software, contact your local Xilinx sales representative.

Reference Documents

To search for Xilinx documentation, go to www.xilinx.com/support.

1. ARM® AMBA® AXI Protocol Version: 2.0 Specification
2. DS765 *LogiCORE IP AXI Lite IPIF Data Sheet*
3. DS160 *Spartan-6 Family Overview*
4. DS150 *Virtex-6 Family Overview*
5. DS180 *7 Series FPGAs Overview*

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
09/21/10	1.0	Initial Xilinx release.
12/14/10	2.0	Updated core to v1.01a and ISE to v12.4.
12/04/11	3.0	Updated core to v1.02a.
6/22/11	3.1	Updated for 13.2 release.
4/24/12	3.1.1	Documentation Changes Only • Added device utilization numbers for Zynq-7000, Virtex-7, Kintex-7, and Artix-7 devices • Updated values for C_S_AXI_ADDR_WIDTH in Table 2 • Updated bit definitions in Table 5 and Table 6
7/25/12	3.2	• Updated for the 14.2/2012.2 Xilinx release • Added Vivado and Zynq-7000 device information • Removed utilization numbers for Zynq-7000 EPP • Removed the AXI parameters, C_BASEADDR and C_HIGHADDR

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.

.