LogiCORE IP AXI UART 16550 (v1.01a)

DS748 June 22, 2011 **Product Specification**

Introduction

The AXI Universal Asynchronous Receiver Transmitter (UART) 16550 connects to the AMBA® (Advance Microcontroller Bus Architecture) AXI (Advanced eXtensible Interface) and provides the controller interface for asynchronous serial data transfer. This soft IP core is designed to connect via an AXI4-Lite interface.

The AXI UART 16550 described in this document incorporates features described in the *National Semiconductor PC16550D UART with FIFOs Data Sheet*.

The National Semiconductor PC16550D data sheet is referenced throughout this document and should be used as the authoritative specification. Differences between the National Semiconductor PC16550D and the AXI UART 16550 data sheet are highlighted in the Specification Exceptions section.

Features

- AXI interface is based on AXI4-Lite specification
- Hardware and software register compatible with all standard 16450 and 16550 UARTs
- Supports default core configuration for 9600 baud, 8 bits data length, 1 stop bit and no parity
- Implements all standard serial interface protocols
  - 5, 6, 7 or 8 bits per character
  - Odd, Even or no parity detection and generation
  - 1, 1.5 or 2 stop bit detection and generation
  - Internal baud rate generator and separate receiver clock input
  - Modem control functions
  - Prioritized transmit, receive, line status and modem control interrupts
  - False start bit detection and recover
  - Line break detection and generation
  - Internal loopback diagnostic functionality
  - 16 character transmit and receive FIFOs

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Artix™-7, Virtex®-7, Kintex™-7, Virtex-6, Spartan®-6
Supported User Interfaces	AXI4-Lite
Resources[2, 3, 4]	
Block RAMs	For the Artix-7 FPGA, see Table 21. For the Virtex-7 FPGA, see Table 20. For the Kintex-7 FPGA, see Table 22. For the Spartan-6 FPGA, see Table 18. For the Virtex-6 FPGA, see Table 19.
LUTs	
Slices	
F_{MAX}	
Provided with Core	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	N/A
Tested Design Tools[5]	
Design Entry Tools	XPS 13.2 or later
Simulation	Mentor Graphics ModelSim v6.6d or later
Synthesis Tools	XST 13.2 or later
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete list of supported derivative devices, see the IDS Embedded Edition Derivative Device Support.
2. For more information, see DS150 Virtex-6 Family Overview.
3. For more information, see DS160 Spartan-6 Family Overview.
4. For more information, see DS180 7 Series FPGAs Overview.
5. For the supported versions of the tools, see the ISE Design Suite 13: Release Notes Guide.

Functional Description

The AXI UART 16550 implements the hardware and software functionality of the National Semiconductor 16550 UART, which works in both the 16450 and 16550 UART modes. For complete details, see the National Semiconductor data sheet.

The AXI UART 16550 performs parallel to serial conversion on characters received from the AXI master and serial to parallel conversion on characters received from a modem or serial peripheral.

The AXI UART 16550 is capable of transmitting and receiving 8, 7, 6, or 5 bit characters, with 2, 1.5 or 1 stop bits and odd, even or no parity. The AXI UART 16550 can transmit and receive independently.

The device can be configured and its status monitored via the internal register set. The AXI UART 16550 is capable of signaling receiver, transmitter and modem control interrupts. These interrupts can be masked, are prioritized and can be identified by reading an internal register.

The device contains a 16-bit, programmable, baud rate generator, and independent 16 character length transmit and receive FIFOs. The FIFOs can be enabled or disabled through software control.

The top-level block diagram for the AXI UART 16550 is shown in Figure 1.

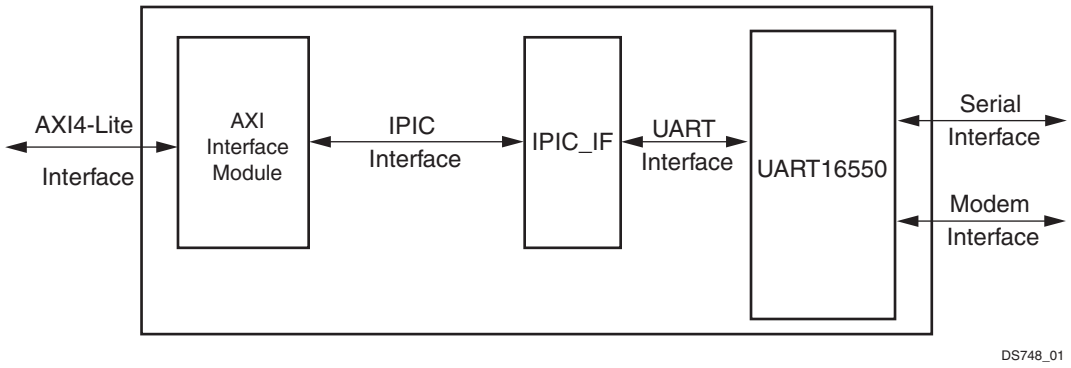

Figure 1: **Top-level Diagram**

The top level modules of the AXI UART 16550 are:

- AXI Interface Module
- IPIC_IF
- UART16550

The detailed block diagram for the AXI UART 16550 is shown in Figure 2.

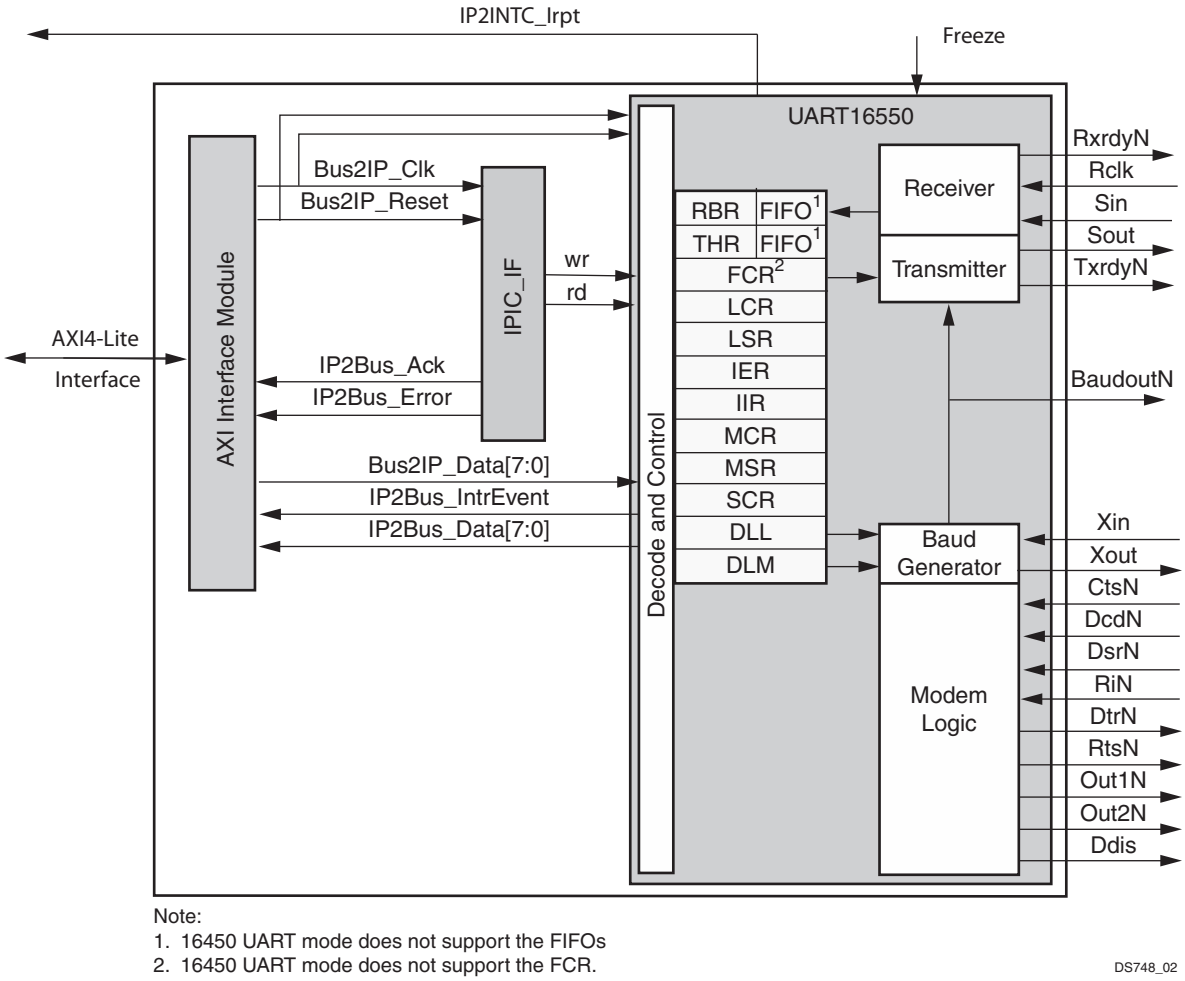

Figure 2: **Top-level Diagram**

The AXI UART 16550 modules are described in these sections:

AXI Interface Module: The 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 AXI interface standard. To simplify the process of attaching the AXI UART 16550 to the AXI, the core makes use of a portable, pre-designed AXI interface called AXI Lite IPIF that manages the AXI interface signals.

IPIC_IF: The IPIC_IF Module provide interface between the AXI interface Module and the UART16550 register interface. This module has the logic for generation of acknowledgment signals for read and write transactions to UART. The AXI Lite IPIF specification is listed in Reference Documents.

UART16550: The UART16550 Module consists of register interface, Receiver, Transmitter, programmable Baud Generator and Modem logic modules. The UART16550 module of AXI UART 16550 can be configured for 16450 or 16550 mode of operation. This is accomplished by the usage of generic C_IS_A_16550.

Interrupts

The AXI UART 16550 core provides separate interrupt enable and interrupt identification registers. If interrupts are enabled, a level sensitive interrupt is generated for these conditions:

- Receiver line status
- Received data available
- Character Timeout
- Transmitter holding register empty
- Modem status

Receiver line status

The Receiver Line Status Interrupt is generated for these conditions:

- Overrun error - An interrupt is generated when Receive buffer is not read by the master and the next character is transferred to the Receive buffer register. In FIFO mode; an overrun interrupt is generated only when FIFO is full and the next character is completely received in the shift register.
- Parity Error - This interrupt is generated when the receive character has an invalid parity bit.
- Framing Error - This interrupt is generated if the received character has an invalid stop bit.
- Line Break - This interrupt is generated when the Receiver detects logic 0 for longer than a full word transmission time.
- The Receiver Line Status Interrupt is cleared by reading the LSR register.

Received data available

The Received data available interrupt is generated when the Receiver FIFO trigger level is reached. This interrupt is cleared when the Receiver FIFO drops below the trigger level.

Character Timeout

The Character Timeout interrupt is generated when no character has been removed from, or input to, the receiver FIFO during the last 4 character time and there is at least one character in the FIFO during this time. The character time considered for timeout (Start + 8 bit data + Parity + 2 Stop bit) is constant for all configurations. This interrupt is cleared by reading Receiver Buffer Register.

Transmitter holding register empty

The Transmitter holding register empty interrupt is generated when the character is transferred from the Transmitter holding register to the Transmitter shift register. In the FIFO mode, this interrupt is generated when Transmitter FIFO becomes empty. This interrupt is cleared by reading the IIR or writing into the Transmitter holding register.

Modem Status

This interrupt is generated for these modem status conditions:

- Clear to send
- Data Set Ready
- Ring Indicator
- Data Carrier Detect

This interrupt is cleared by reading the Modem Status Register.

I/O Signals

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

Table 1: **I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	S_AXI_ACLK	System	I	-	AXI Clock
P2	S_AXI_ARESETN	System	I	-	AXI Reset signal, active Low
P3	IP2INTC_Irpt	System	O	0	Device interrupt output to microprocessor interrupt input or system interrupt controller (active High)
P4	Freeze	System	I	-	Freezes UART for software debug (active High)
AXI Write Address Channel Signals					
P5	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.
P6	S_AXI_AWVALID	AXI	I	-	Write address valid. This signal indicates that valid write address is available.
P7	S_AXI_AWREADY	AXI	O	0	Write address ready. This signal indicates that the slave is ready to accept an address.
AXI Write Channel Signals					
P8	S_AXI_WDATA[C_S_AXI_ DATA_WIDTH - 1: 0]	AXI	I	-	Write data
P9	S_AXI_WSTB[C_S_AXI_ DATA_WIDTH/8-1:0] (1)	AXI	I	-	Write strobes. This signal indicates which byte lanes to update in memory.
P10	S_AXI_WVALID	AXI	I	-	Write valid. This signal indicates that valid write data and strobes are available.
P11	S_AXI_WREADY	AXI	O	0	Write ready. This signal indicates that the slave can accept the write data.
AXI Write Response Channel Signals					
P12	S_AXI_BRESP[1:0](2)	AXI	O	0	Write response. This signal indicates the status of the write transaction. “00“ - OKAY “10“ - SLVERR
P13	S_AXI_BVALID	AXI	O	0	Write response valid. This signal indicates that a valid write response is available.
P14	S_AXI_BREADY	AXI	I	-	Response ready. This signal indicates that the master can accept the response information.
AXI Read Address Channel Signals					
P15	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.
P16	S_AXI_ARVALID	AXI	I	-	Read address valid. When High, this signal indicates that the read address is valid and remains stable until the address acknowledgement signal, S_AXI_ARREADY, is High.
P17	S_AXI_ARREADY	AXI	O	1	Read address ready. This signal indicates that the slave is ready to accept an address.

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

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Read Data Channel Signals					
P18	S_AXI_RDATA[C_S_AXI_DATA_WIDTH -1:0]	AXI	O	0	Read data
P19	S_AXI_RRESP[1:0][(2)]	AXI	O	0	Read response. This signal indicates the status of the read transfer. "00" - OKAY "10" - SLVERR
P20	S_AXI_RVALID	AXI	O	0	Read valid. This signal indicates that the required read data is available and the read transfer can complete
P21	S_AXI_RREADY	AXI	I	-	Read ready. This signal indicates that the master can accept the read data and response information
UART Interface Signals					
P22	BaudoutN	Serial	O	1	16 x clock signal from the transmitter section of the UART
P23	Rclk	Serial	I	-	Receiver 16x clock (Optional, can be driven externally under control of the C_HAS_EXTERNAL_RCLK parameter)
P24	Sin	Serial	I	-	Serial data input
P25	Sout	Serial	O	1	Serial data output
P26	Xin	Serial	I	-	Baud rate generator reference clock (Optional, can be driven externally under control of the C_HAS_EXTERNAL_XIN parameter)
P27	Xout	Serial	O	0	If C_HAS_EXTERNAL_XIN = 0, Xout is 0, if C_HAS_EXTERNAL_XIN = 1 Xout can be used as reference feedback clock for Baud rate generator
P28	CtsN	Modem	I	-	Clear to send (active Low). When Low, this indicates that the MODEM or data set is ready to exchange data.
P29	DcdN	Modem	I	-	Data carrier detect (active Low). When Low, indicates that the data carrier has been detected by the MODEM or data set.
P30	DsrN	Modem	I	-	Data set ready (active Low). When Low, this indicates that the MODEM or data set is ready to establish the communication link with the UART.
P31	DtrN	Modem	O	1	Data terminal ready (active Low). When Low, this informs the MODEM or data set that the UART is ready to establish a communication link.
P32	RiN	Modem	I	-	Ring indicator (active Low). When Low, this indicates that a telephone ringing signal has been received by the MODEM or data set.
P33	RtsN	Modem	O	1	Request to send (active Low). When Low, this informs the MODEM or data set that the UART is ready to exchange data.
P34	Ddis	User	O	1	Driver disable. This goes Low when CPU is reading data from UART.
P35	Out1N	User	O	1	User controlled output
P36	Our2N	User	O	1	User controlled output

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

Port	Signal Name	Interface	I/O	Initial State	Description
P37	RxrdyN	User	O	1	DMA control signal
P38	TxrdyN	User	O	0	DMA control signal

Notes:

1. This signal is not used. The AXI UART 16550 assumes that all byte lanes are active.
2. For these signals, the IP core does not generate the Decode Error ("11") response. Other responses such as "00" (OKAY) and "10" (SLVERR) are generated by the core based on certain conditions.

Design Parameters

To allow the user to create an AXI UART 16550 that is uniquely tailored for the user's system, certain features are parameterizable in the AXI UART 16550 design. This allows the user to have a design that utilizes only the resources required by the system and runs at the highest possible performance. The parameterizable features in the AXI UART 16550 core are as shown in Table 2.

Table 2: **Design Parameters**

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameters					
G1	Target FPGA family	C_FAMILY	virtex6,spartan6	virtex6	string
G2	System clock frequency (in Hz) driving the 16550 UART peripheral	C_S_AXI_ACLK_FREQ_HZ	integer (ex.100000000)	100_000_000	integer
AXI Parameters					
G3	AXI Base Address	C_BASEADDR	Valid Address(1)	0xFFFFFFFF(3)	std_logic_vector
G4	AXI High Address	C_HIGHADDR	Valid Address(2)	0x00000000(3)	std_logic_vector
G5	AXI address bus width	C_S_AXI_ADDR_WIDTH	32	32	integer
G6	AXI data bus width	C_S_AXI_DATA_WIDTH	32	32	integer
16550 UART Interface					
G7	External xin clock	C_HAS_EXTERNAL_XIN	0 : xin is open(4)(5) 1 : xin is externally driven	0	integer
G8	External Receiver clock	C_HAS_EXTERNAL_RCLK	0 : rclk is open 1 : rclk is externally driven	0	integer
G9	Select 16450/16550 UART	C_IS_A_16550	0 : 16450 mode 1 : 16550 mode	1	integer
G10	External xin clock frequency in Hz.	C_EXTERNAL_XIN_CLK_HZ(6)	Valid xin clock frequency in Hz.	25000000	integer

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

Generic	Parameter Description	Parameter Name	Allowable Values	Default Value	VHDL Type

Notes:

1. The user must set the values. The C_BASEADRR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1.
2. C_HIGHADDR - C_BASEADDR must be a power of 2. C_HIGHADDR must be greater than or equal to C_BASEADDR + 0xFFF.
3. An invalid default value is used to require that an actual value is set, otherwise a compiler error occurs.
4. When C_HAS_EXTERNAL_XIN=0, this core uses S_AXI_ACLK as a reference clock for the baud calculation. User must use S_AXI_ACLK frequency to calculate baud divisor value for DLL and DLM register configuration.
5. The external xin input clock must be less than half of S_AXI_ACLK.
6. External xin clock frequency. User must configure this parameter when external xin is used. (C_HAS_EXTERNAL_XIN is '1').

Parameter - Port Dependencies

The dependencies between the AXI UART 16550 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-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G5	C_S_AXI_ADDR_WIDTH	P5, P15	-	Defines the width of the ports
G6	C_S_AXI_DATA_WIDTH	P8, P9, P18	-	Defines the width of the ports
I/O Signals				
P5	S_AXI_AWADDR[C_S_AXI_ ADDR_WIDTH-1:0]	-	G5	Port width depends on the generic C_S_AXI_ADDR_WIDTH
P8	S_AXI_WDATA[C_S_AXI_DATA_ WIDTH-1:0]	-	G6	Port width depends on the generic C_S_AXI_DATA_WIDTH
P9	S_AXI_WSTB[C_S_AXI_DATA_ WIDTH/8-1:0]	-	G6	Port width depends on the generic C_S_AXI_DATA_WIDTH
P15	S_AXI_ARADDR[C_S_AXI_ ADDR_WIDTH -1:0]	-	G5	Port width depends on the generic C_S_AXI_ADDR_WIDTH
P18	S_AXI_RDATA[C_S_AXI_DATA_ WIDTH -1:0]	-	G6	Port width depends on the generic C_S_AXI_DATA_WIDTH
P23	Rclk	-	G12, P45	If C_HAS_EXTERNAL_RCLK = 0 baudoutN is used as 16x receiver clock, C_HAS_EXTERNAL_RCLK = 1, rclk is used as 16x receiver clock.
P26	Xin	-	G11	When C_HAS_EXTERNAL_XIN = 0, xin is unconnected, C_HAS_EXTERNAL_XIN = 1, xin is driven externally.

Register Descriptions

AXI 16550 Interface

The internal registers of the AXI UART 16550 are offset from the base address C_BASEADDR. Additionally, some of the internal registers are accessible only when bit 7 of the Line Control Register (LCR) is set. The AXI UART 16550 internal register set is described in Table 4.

Table 4: **Registers**

Register Name	LCR(7)+ C_BASEADDR + Address	Access
Receiver Buffer Register (RBR)	0 + C_BASEADDR + 0x1000	Read
Transmitter Holding Register (THR)	0 + C_BASEADDR + 0x1000	Write
Interrupt Enable Register (IER)	0 + C_BASEADDR + 0x1004	Read/Write
Interrupt Identification Register (IIR)	0 + C_BASEADDR + 0x1008	Read
FIFO Control Register (FCR)(3)	X + C_BASEADDR + 0x1008	Write
FIFO Control Register(2), (3)	1 + C_BASEADDR + 0x1008	Read
Line Control Register (LCR)	X(1) + C_BASEADDR + 0x100C	Read/Write
Modem Control Register (MCR)	X(1) + C_BASEADDR + 0x1010	Read/Write
Line Status Register (LSR)	X(1) + C_BASEADDR + 0x1014	Read/Write
Modem Status Register (MSR)	X(1) + C_BASEADDR + 0x1018	Read/Write
Scratch Register (SCR)	X(1) + C_BASEADDR + 0x101C	Read/Write
Divisor Latch (Least Significant Byte) Register (DLL)	1 + C_BASEADDR + 0x1000	Read/Write
Divisor Latch (Most Significant Byte) Register (DLM)	1 + C_BASEADDR + 0x1004	Read/Write

Notes:

1. X denotes a ‘don’t care’
2. FIFO Control Register is write-only in the National PC16550D
3. 16450 UART mode implementation does not include this register

Register Logic

This section tabulates the internal AXI UART 16550 registers, including their reset values (if any). See the National Semiconductor PC16550D UART with FIFOs data sheet (June, 1995) for a more detailed description of the register behavior.

Receiver Buffer Register

This 32-bit read register is shown in Figure 3. The Receiver Buffer Register contains the last received character. The bit definitions for the register are shown in Table 5. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 3: **Receiver Buffer Register (RBR)**

Table 5: **Receiver Buffer Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-0	RBR	Read	"00000000"	Last received character

Transmitter Holding Register

This 32-bit write register is shown in Figure 4. The Transmitter Holding Register contains the character to be transmitted next. The bit definitions for the register are shown in Table 6. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 4: **Transmitter Holding Register (THR)**

Table 6: **Transmitter Holding Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-0	THR	Write	"11111111"	Holds the character to be transmitted next

Interrupt Enable Register

This 32-bit read/write register is shown in Figure 5. The Interrupt Enable Register contains the bits which enable interrupts. The bit definitions for the register are shown in Table 7. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 5: **Interrupt Enable Register (IER)**

Table 7: **Interrupt Enable Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-4	N/A	Read/Write	"0000"(1)	Always returns "0000"
3	EDSSI	Read/Write	'0'	Enable Modem Status Interrupt '0' = Disables Modem Status Interrupts. '1' = Enables Modem Status Interrupts.
2	ELSI	Read/Write	'0'	Enable Receiver Line Status Interrupt '0' = Disables Receiver Line Status Interrupts. '1' = Enables Receiver Line Status Interrupts.
1	ETBEI	Read/Write	'0'	Enable Transmitter Holding Register Empty Interrupt '0' = Disables Transmitter Holding Register Empty Interrupts. '1' = Enables Transmitter Holding Register Interrupts.
0	ERBFI	Read/Write	'0'	Enable Received Data Available Interrupt '0' = Disables Received Data Available Interrupts '1' = Enables Received Data Available Interrupts

1. Reading these bits always returns "0000"

Interrupt Identification Register

This 32-bit read register is shown in Figure 6. The Interrupt Identification Register contains the priority interrupt identification. The bit definitions for the register are shown in Table 8. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 6: **Interrupt Identification Register (IIR)**

Table 8: **Interrupt Identification Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-6	FIFOEN(1)	Read	"00"	FIFOs Enabled. Always zero if not in FIFO mode.
5-4	N/A	Read	"00"(2)	Always returns "00"
3-1	INTID2	Read	"000"	Interrupt ID "011" = Receiver Line Status (Highest)(4) "010" = Received Data Available (Second) "110" = Character Timeout (Second) "001" = Transmitter Holding Register Empty (Third) "000" = Modem Status (Fourth)
0	INTPEND(3)	Read	'1'	0 - Interrupt is pending 1- No interrupt is pending

Notes:

1. Bits are always zero in 16450 UART mode
2. Reading these bits always return "00"
3. If INTPEND = '0', interrupt is pending. See National Semiconductor PC16550D data sheet for more details
4. Line status interrupt is generated for framing, parity, overrun error and break condition.

FIFO Control Register

This is 32-bit write/read register is shown in Figure 7. The FIFO Control Register contains the FIFO configuration bits. The bit definitions for the register are shown in Table 9. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4. The DMA mode signalling information is given in Table 10.

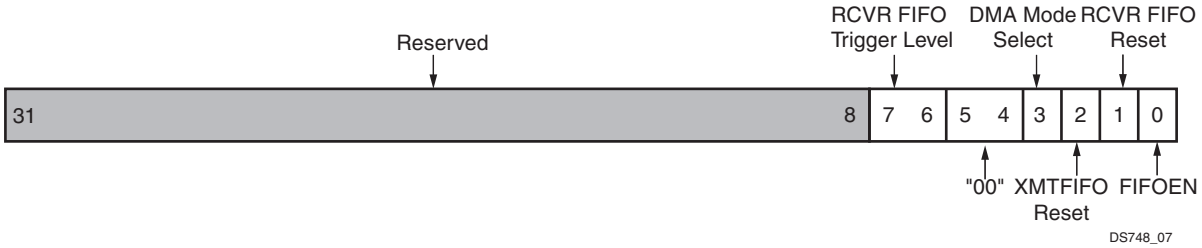

Figure 7: **FIFO Control Register (FCR)**

Table 9: **FIFO Control Register Bit Definitions** (1)

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-6	RCVR FIFO Trigger Level	Read/Write	"00"	RCVR FIFO Trigger Level. "00" = 1 byte "01" = 4 bytes "10" = 8 bytes "11" = 14 bytes
5-4	Reserved	Read/Write	N/A	Reserved
3	DMA Mode Select	Read/Write	’0’	DMA Mode Select ’0’ = Mode 0 ’1’ = Mode 1
2	XMIT FIFO Reset	Read/Write	’0’	Transmitter FIFO Reset ’1’ = Resets XMIT FIFO.
1	RCVR FIFO Reset	Read/Write	’0’	Receiver FIFO Reset ’1’ = Resets RCVR FIFO.
0	FIFOEN	Read/Write	’0’	FIFO Enable ’1’ = Enables FIFOs.

Notes:

1. FCR is not included in 16450 UART mode

Table 10: **DMA Modes signalling**

DMA Mode	TXRDYn	RXRDYn
MODE 0	In the 16450 mode or in mode 0, when there are no characters in the THR or Transmitter FIFO, this signal is Low. This signal goes High again after the first character is loaded into the THR or FIFO.	In the 16450 mode or in mode 0, when there is at least one character in the Receiver FIFO or Receiver holding register, this signal is Low. This signal goes High again when there are no characters in FIFO or receiver holding register.
MODE 1	When there are no characters in the Transmitter FIFO, this signal goes Low. This signal goes High again if the FIFO is completely full.	When the trigger level or the timeout has been reached, this signal goes Low. This signal goes High again when there are no characters in the FIFO or receiver holding register.

Line Control Register

This 32-bit write/read register is shown in Figure 8. The Line Control Register contains the serial communication configuration bits. The bit definitions for the register are shown in Table 11. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

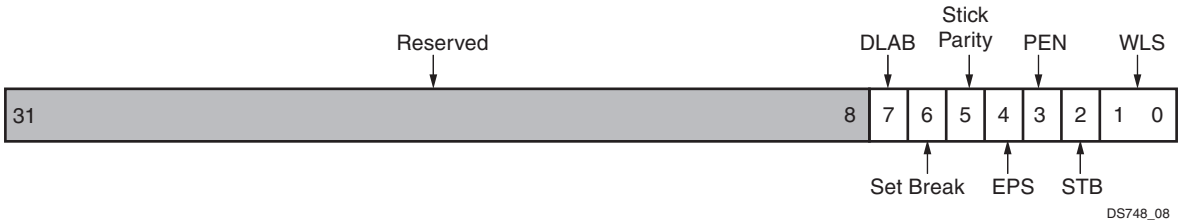

Figure 8: **Line Control Register (LCR)**

Table 11: **Line Control Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7	DLAB	Read/Write	'0'	Divisor Latch Access Bit. '1' = Allows access to the Divisor Latch Registers and reading of the FIFO Control Register. '0' = Allows access to RBR, THR, IER and IIR registers.
6	Set Break	Read/Write	'0'	Set Break '1' = Enables break condition. Sets SOUT to '0' and cause break condition. '0' = Disables break condition.
5	Stick Parity	Read/Write	'0'	Stick Parity '1' = When bits 3, 4 are logic1 the Parity bit is transmitted and checked as a logic 0. If bit 4 is a logic 0 and bit 3 is logic 1 then the Parity bit is transmitted and checked as a logic 1. '0' = Stick Parity is disabled.
4	EPS	Read/Write	'0'	Even Parity Select '1' = Selects Even parity. '0' = Selects Odd parity.
3	PEN	Read/Write	'0'	Parity Enable '1' = Enables parity. '0' = Disables parity.
2	STB	Read/Write	'0'	Number of Stop Bits '0' = 1 Stop bit '1' = 2 Stop bits or 1.5, if 5 bits/character selected. The receiver checks for 1 stop bit only regardless of the number of stop bits selected.
1-0	WLS	Read/Write	"11"	Word Length Select "00" = 5 bits/character "01" = 6 bits/character "10" = 7 bits/character "11" = 8 bits/character

Modem Control Register

This 32-bit write/read register is shown in Figure 9. The Modem Control Register contains the modem signalling configuration bits. The bit definitions for the register are shown in Table 12. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

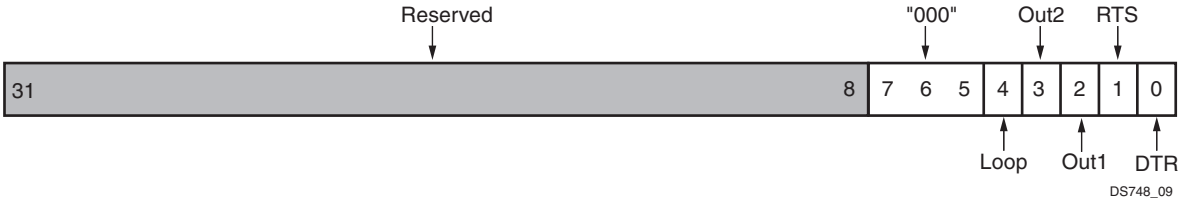

Figure 9: **Modem Control Register (MCR)**

Table 12: **Modem Control Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-5	N/A	Read/Write	"000"(1)	Always "000"
4	Loop	Read/Write	'0'	Loop Back '1' = Enables loop back.
3	Out2	Read/Write	'0'	User Output 2 '1' = Drives OUT2N Low. '0' = Drives OUT2N High.
2	Out1	Read/Write	'0'	User Output 1 '1' = Drives OUT1N Low. '0' = Drives OUT1N High.
1	RTS	Read/Write	'0'	Request To Send '1' = Drives RTSN Low. '0' = Drives RTSN High.
0	DTR	Read/Write	'0'	Data Terminal Ready '1' = Drives DTRN Low. '0' = Drives DTRN High.

Notes:

1. Reading these bits always returns "000".

Line Status Register

This 32-bit write/read register as shown in Figure 10. The Line Status Register contains the current status of receiver and transmitter. The bit definitions for the register are shown in Table 13. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 10: **Line Status Register (LSR)**

Table 13: **Line Status Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7	Error in RCVR FIFO	Read/Write	'0'	Error in RCVR FIFO(1): RCVR FIFO contains at least one receiver error (Parity, Framing, Break condition).
6	TEMT	Read/Write	'1'	Transmitter Empty: 0 - THR or Transmitter shift register contains data. 1 - THR and Transmitter shift register empty. In FIFO mode, Transmitter FIFO and shift register are both empty.
5	THRE	Read/Write	'1'	Transmitter Holding Register Empty 0 - THR or Transmitter FIFO has data to transmit. 1 - THR is empty. In FIFO mode, Transmitter FIFO is empty.
4	BI	Read/Write	'0'	Break Interrupt Set when SIN is held Low for an entire character time. (Start + data bits + Parity + Stop bits). In FIFO mode, this error is associated with a particular character in FIFO. The next character transfer is enabled if the Sin goes to marking state and receives the next valid start bit.
3	FE	Read/Write	'0'	Framing Error Character missing a stop bit. In framing error, the UART attempts to re-synchronize by assuming that the framing error was due to next character start bit, so it samples start bit twice and then takes in following data. In FIFO mode, this error is associated with a particular character in the FIFO.
2	PE	Read/Write	'0'	Parity Error Indicates that the received data character does not have correct even or odd parity as selected by the Even parity select bit. In FIFO mode, this error is associated with a particular character in the FIFO.
1	OE	Read/Write	'0'	Overrun Error RBR not read before next character is received, thereby destroying the previous character. In FIFO mode, data continues to fill the FIFO beyond the trigger level, an overrun error occurs only after the FIFO is full and the next character has been completely received in the shift register. The character in the shift register is overwritten but it is not transferred to the FIFO.

Table 13: **Line Status Register Bit Definitions *(Cont'd)***

Bit	Name	Access	Reset Value	Description
0	DR	Read/Write	'0'	Data Ready 0 - All the data in RBR or FIFO is read 1 - Complete incoming character has been received and transferred into the RBR of FIFO.

Notes:

1. The error is reported until the last character containing an error in the FIFO is read out of the FIFO.

Modem Status Register

This 32-bit write/read register is shown in Figure 11. The Modem Status Register contains the current state of the Modem Interface. The bit definitions for the register are shown in Table 14. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 11: **Modem Status Register (MSR)**

Table 14: **Modem Status Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7	DCD	Read/Write	'X'(1)	Data Carrier Detect Complement of DCDN input
6	RI	Read/Write	'X'(1)	Ring Indicator Complement of RIN input
5	DSR	Read/Write	'X'(1)	Data Set Ready Complement of DSRN input
4	CTS	Read/Write	'X'(1)	Clear To Send Complement of CTSN input
3	DDCD	Read/Write	'0'	Delta Data Carrier Detect Change in DCDN since last MSR read
2	TERI	Read/Write	'0'	Trailing Edge Ring Indicator RIN has changed from a Low to a High
1	DDSR	Read/Write	'0'	Delta Data Set Ready Change in DSRN since last MSR read
0	DCTS	Read/Write	'0'	Delta Clear To Send Change in CTSN since last MSR read

Notes:

1. X represents bit driven by external input

Scratch Register

This 32-bit write/read register is shown in Figure 12. The Scratch Register can be used to hold user data. The bit definitions for the register are shown in Table 15. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 12: **Scratch Register (SCR)**

Table 15: **Scratch Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved.
7-0	Scratch	Read/Write	"00000000"	Hold the data temporarily.

Divisor Latch (Least Significant Byte) Register

This 32-bit write/read register is shown in Figure 13. The Divisor Latch (Least Significant Byte) Register holds the least significant byte of the baud rate generator counter. The bit definitions for the register are shown in Table 16. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 13: **Divisor Latch (Least Significant Byte) Register**

Table 16: **Divisor Latch (Least Significant Byte) Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-0	DLL	Read/Write	"XXXXXXXX"(1)	Divisor Latch Least Significant Byte

Notes:

1. On reset, the DLL gets configured for 9600 baud. The DLL reset value, [LSB(divisor)] is calculated from the formula, divisor = C_S_AXI_ACLK_FREQ/(16 x 9600)).

Divisor Latch (Most Significant Byte) Register

This 32-bit write/read register is shown in Figure 14. The Divisor Latch (Most Significant Byte) Register holds the most significant byte of the baud rate generator counter. The bit definitions for the register are shown in Table 17. The offset and accessibility of this register from C_BASEADDR value is as shown in Table 4.

Figure 14: **Divisor Latch (Most Significant Byte) Register**

Table 17: **Divisor (Most Significant Byte) Register Bit Definitions**

Bit	Name	Access	Reset Value	Description
31-8	Reserved	N/A	N/A	Reserved
7-0	DLM	Read/Write	"XXXXXXXX" (1)	Divisor Latch Most Significant Byte

Notes:

1. On reset, the DLM gets configured for 9600 baud. The DLM reset value, [MSB(divisor)] is calculated from the formula, divisor = (C_S_AXI_ACLK_FREQ/(16 x 9600)).

User Application Hints

The use of the AXI UART 16550 in 16550 mode is outlined in these steps:

1. The system programmer specifies the format of the asynchronous data communications exchange: for example, Data bits (5,6,7 or 8), setting of parity ON and selecting on the even or odd parity, setting of the number stop bits for the transmission, and set the Divisor latch access bit by programming the Line Control Register.
2. Write Interrupt Enable Register to activate the individual interrupts
3. Write to the FIFO Control Register to enable the FIFO's, clear the FIFO's, and set the RCVR FIFO trigger level.
4. Write to Divisor Latch least significant byte first, then the Divisor Latch most significant byte second for proper setting of the baud rate of the UART.
5. Service the interrupts whenever an interrupt is triggered by the AXI UART 16550.

Example 1

An example use of the AXI UART 16550 with the operating mode set to the following parameters in 16550 mode is outlined in the subsequent numbered steps.

- Baud rate: 56Kbps
- System clock: 100 Mhz (C_HAS_EXTERNAL_XIN = 0)
- Enabled and Threshold settings for the FIFO receive buffer.
- Format of asynchronous data exchange 8 data bits, Even parity and 2 stop bits

1. Write 0x0000_0080 to Line Control Register. This configures the DLAB bit, which allows the writing into the Divisor Latch's Least significant and Most significant bytes.
2. Write 0x0000_006F to Divisor Latch's Least significant byte and write 0x0000_0000 to Divisor Latch's Most significant byte in that order. This configures the baud rate setup of UART to 56Kbps operation. The divisor value is calculated by using following formula:

   divisor = (C_S_AXI_ACLK_FREQ/(16 × Baud Rate))

3. Write 0x0000_001F to Line Control Register. This configures word length to 8 bits, Number of Stop Bits to 2, Parity is enabled and set to Even parity and DLAB bit is set to value 0 to enable the use of Transmit Holding register and Receive Buffer register data for transmitting and reception of data.
4. Write 0x0000_0011 to Interrupt Enable Register. This enables the Transmitter holding register empty interrupt and Receive data available interrupt.
5. Write the buffer to Transmit Holding register and read the data received from Receive Holding register by servicing the interrupts generated.

Example 2

An example use of the AXI UART 16550 when external xin clock is used (C_HAS_EXTERNAL_XIN = 1) with the operating mode set to the following parameters in 16550 mode outlined in the subsequent numbered steps.

- Baud rate: 56Kbps
- System clock: 100 Mhz
- External xin clock: 1.8432 Mhz
- Enabled and Threshold settings for the FIFO receive buffer.
- Format of asynchronous data exchange 8 data bits, Even parity and 2 stop bits

1. Write 0x0000_0080 to Line Control Register. This configures the DLAB bit which allows the writing into the Divisor Latch's Least significant and Most significant bytes.
2. Write 0x0000_0002 to the Divisor Latch's Least significant byte and write 0x0000_0000 to Divisor Latch's Most significant byte in that order. This configures the baud rate setup of the UART to 56Kbps operation. Other steps remain the same as shown in the previous example.
3. Write 0x0000_001F to Line Control Register. This configures word length to 8 bits, Number of stop bits to 2, Parity is enabled and set to Even parity, and DLAB bit is set to value 0 to enable the use of Transmit Holding register and Receive buffer register data for transmitting and reception of data.
4. Write 0x0000_0011 to Interrupt Enable Register. This enables the Transmitter holding register empty interrupt and Receive data available interrupt.
5. Write the buffer to Transmit Holding register and read the data received from Receive Holding register by servicing the interrupts generated.

Design Implementation

Target Technology

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

Device Utilization and Performance Benchmarks

Core Performance

Because the AXI UART 16550 core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI UART 16550 design varies from the results reported here.

The AXI UART 16550 resource utilization for various parameter combinations measured with the Spartan-6 FPGA as the target device are detailed Table 18.

Table 18: **Performance and Resource Utilization Benchmarks on the Spartan-6 FPGA (xc6slx16-csg324-2)**

Parameter Values			Device Resources			Performance
C_IS_A_16550	C_HAS_EXTERNAL_XIN	C_HAS_EXTERNAL_RCLK	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	0	115	229	250	110
0	0	1	109	230	260	110
0	1	0	120	239	292	110
0	1	1	128	237	277	110
1	0	0	160	279	348	110
1	0	1	145	281	351	110
1	1	0	162	291	388	110
1	1	1	134	281	363	110

The AXI UART 16550 resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device are detailed Table 19.

Table 19: **Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx75t-ff784-1)**

Parameter Values			Device Resources			Performance
C_IS_A_16550	C_HAS_EXTERNAL_XIN	C_HAS_EXTERNAL_RCLK	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	0	120	230	254	200
0	0	1	124	231	247	200
0	1	0	128	231	277	200
0	1	1	131	233	266	200
1	0	0	159	280	380	200
1	0	1	162	281	372	200
1	1	0	158	281	393	200
1	1	1	175	288	395	200

The AXI UART 16550 resource utilization for various parameter combinations measured with the Virtex-7 FPGA as the target device are detailed Table 20.

Table 20: **Performance and Resource Utilization Benchmarks on the Virtex-7 FPGA (xc7v855tffg1157-3)**

Parameter Values			Device Resources			Performance
C_IS_A_16550	C_HAS_EXTERNAL_XIN	C_HAS_EXTERNAL_RCLK	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	0	168	285	360	264
0	0	1	131	237	256	201
0	1	0	118	237	265	201
0	1	1	119	243	274	201
1	0	0	168	285	360	264
1	0	1	167	286	365	205
1	1	0	167	286	395	236
1	1	1	181	289	398	230

The AXI UART 16550 resource utilization for various parameter combinations measured with the Artix-7 FPGA as the target device are detailed Table 21.

Table 21: **Performance and Resource Utilization Benchmarks on the Artix-7 FPGA (xc7a355tdie-3)**

Parameter Values			Device Resources			Performance
C_IS_A_16550	C_HAS_EXTERNAL_XIN	C_HAS_EXTERNAL_RCLK	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	0	169	285	400	200
0	0	1	112	241	278	201
0	1	0	116	237	270	201
0	1	1	125	243	271	202
1	0	0	169	285	400	200
1	0	1	163	286	391	201

Table 21: **Performance and Resource Utilization Benchmarks on the Artix-7 FPGA (xc7a355tdie-3) *(Cont'd)***

1	1	0	174	286	398	201
1	1	1	151	291	401	201

The AXI UART 16550 resource utilization for various parameter combinations measured with the Kintex-7 FPGA as the target device are detailed Table 22.

Table 22: **Performance and Resource Utilization Benchmarks on the Kintex-7 FPGA (xc7k410tffg676-3)**

Parameter Values			Device Resources			Performance
C_IS_A_16550	C_HAS_EXTERNAL_XIN	C_HAS_EXTERNAL_RCLK	Slices	Slice Flip-Flops	LUTs	F_{MAX} (MHz)
0	0	0	166	285	361	209
0	0	1	121	237	260	209
0	1	0	114	237	268	204
0	1	1	120	243	264	241
1	0	0	166	285	364	209
1	0	1	184	286	359	202
1	1	0	161	286	396	221
1	1	1	164	289	399	229

System Performance

To measure the system performance (F_{MAX}) of the AXI UART 16550 core, it was added as the Device Under Test (DUT) to a Virtex-6 FPGA system as shown in Figure 15 and to a Spartan-6 FPGA system as shown in Figure 16.

Because the AXI UART 16550 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.

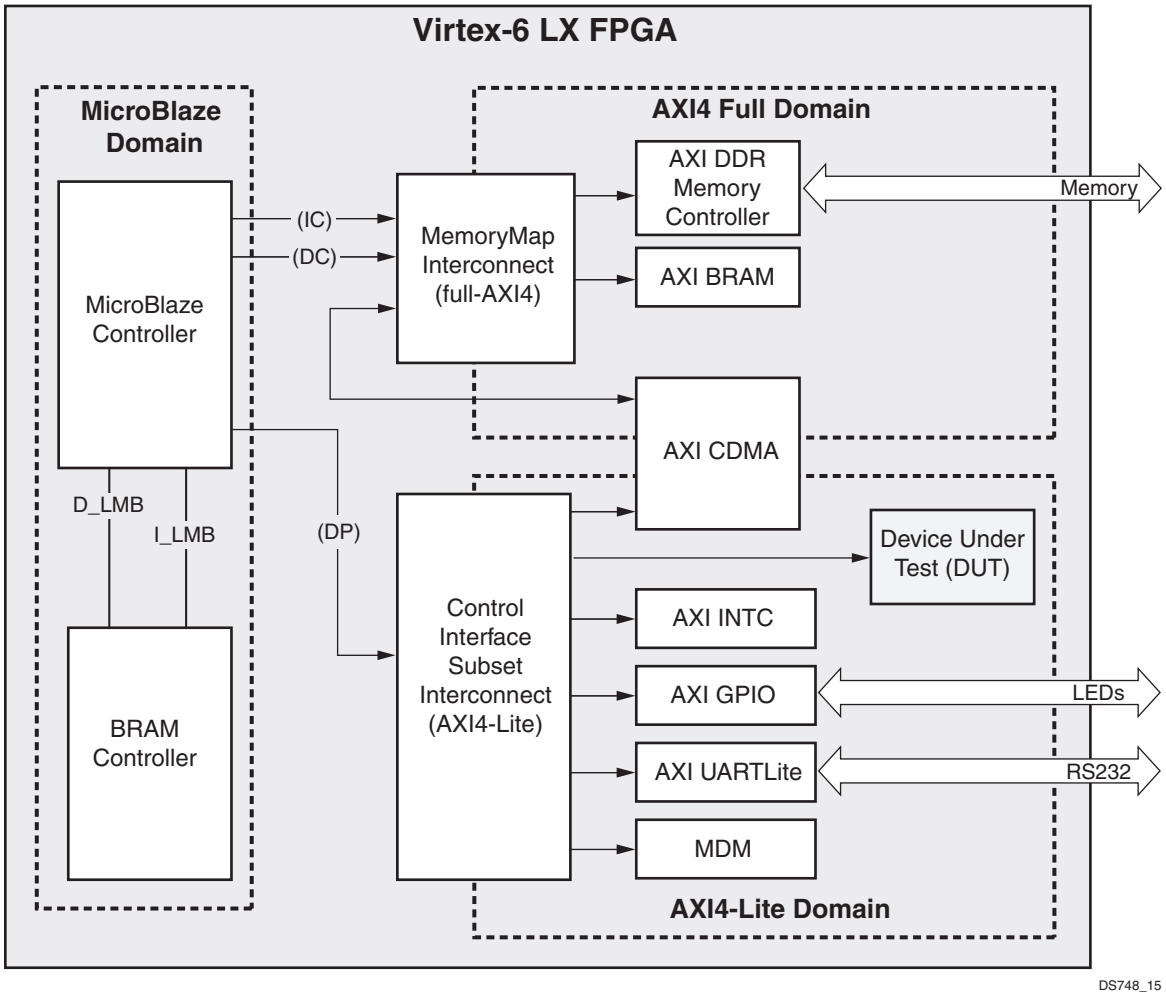

Figure 15: **Virtex-6 LX FPGA System with the AXI UART 16550 as the DUT**

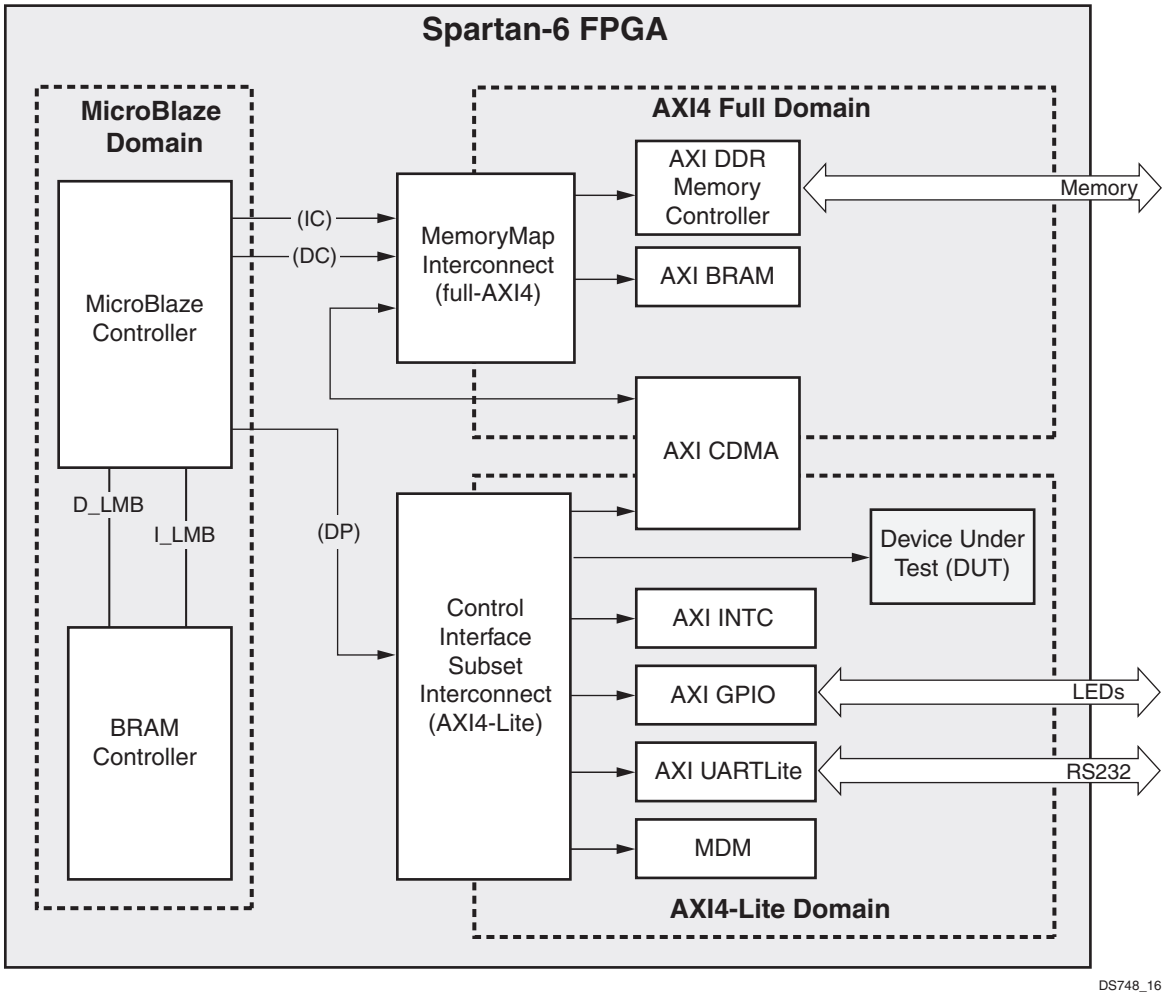

Figure 16: **Spartan-6 LXT FPGA System with the AXI UART 16550 as the DUT**

The target FPGA was then 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 23.

Table 23: **AXI UART 16550 FPGA System Performance**

Target FPGA	Target F_{MAX} (MHz)
V6LX130t-1	180
S6LX45t-2	110

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

Specification Exceptions

FIFO Control Register

The FIFO control register has been made read/write. Read access is controlled by setting Line Control Register bit 7.

System Clock

The asynchronous microprocessor interface of the National Semiconductor PC16550D is synchronized to the system clock input of the UART.

XIN Clock

If the xin input is driven externally, then the xin clock must be less than or equal to half of the system clock. (i.e. xin ≤ (S_AXI_ACLK/2)). This is mandatory for the proper functioning of the core.

Register Addresses

All internal registers reside on 32- bit word boundaries, not on 8-bit byte boundaries.

Default Register Configuration

On application of reset, the AXI UART 16550 registers is configured to support default UART configuration such as 9600 baud, 8-bit data width, 1 stop bit and no parity.

Support

Xilinx provides technical support for this LogiCORE™ 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***.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE® Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

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

The listed documents contain reference information that is important for understanding the UART design:

1. National Semiconductor PC16550D UART with FIFOs data sheet (June, 1995) (http://www.national/com/pf/PC/PCI16550.html)
2. ARM AMBA Protocol Version: 2.0 Specification
3. DS765: LogiCORE IP AXI Lite IPIF (v1.01a) Data Sheet

Revision History

Date	Version	Revision
9/21/10	1.0	Initial Xilinx release.
12/14/10	2.0	Updated for core version 1.01a.
06/22/11	2.1	Updated to ISE 13.2. Updated for Artix-7, Virtex-7, and Kintex-7.

Notice of Disclaimer