OPB UART Lite

DS209 (v2.4) January 8, 2003 **Product Specification**

Summary

This document describes the specifications for a UART core for the On-Chip Peripheral Bus (OPB). This document applies to the following peripherals:

opb_uartlite	v1.00b

Overview

The UART Lite is a module that attaches to the OPB (On-Chip Peripheral Bus) and has the following features:

Features

- OPB V2.0 bus interface with byte-enable support
- Supports 8-bit bus interfaces
- One transmit and one receive channel (full duplex)
- 16-character transmit FIFO and 16-character receive FIFO
- Number of databits in a character is configurable (5-8)
- Parity; can be configured for odd or even
- Configurable baud rate

UART Lite Parameters

To allow you to obtain a UART Lite that is uniquely tailored for your system, certain features can be parameterized in a UART Lite design. This allows you to configure a design that only utilizes the resources required by your system, and operates with the best possible performance. The features that can be parameterized in the Xilinx UART Lite design are shown in Table 1.

Table 1: **UART Lite Parameters**

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
UART Lite Registers Base Address		Valid Address Range(2)	None(1)	std_logic_vector
UART Lite Registers High Address		Valid Address Range(2)	None(1)	std_logic_vector
Target Family		Xilinx FPGA families	virtex2	strings
Bus Width	C_OPB_AWIDTH	32	32	integer
Data Bus Width	C_OPB_DWIDTH	32	32	integer
C_CLK_FREQ	Clock frequency of the OPB system clock driving the UART Lite peripheral in Hz.	integer (ex. 125000000)	125_000_000	integer

Table 1: **UART Lite Parameters *(Continued)***

Feature/Description	Parameter Name	Allowable Values	Default Value	VHDL Type
C_BAUDRATE	Baud rate of the UART Lite in bits per second.	integer (ex. 9600)	19_200	integer
C_DATA_BITS	The number of data bits in the serial frame.	integer (5 to 8)	8	integer
C_USE_PARITY	Determines whether parity is used or not.	Integer 1 = use parity, 0 = do not use parity.	1	integer
C_ODD_PARITY	If parity is used, determines whether parity is odd or even	integer 1= odd parity, 0 = even parity.	1	integer

System Generator & CORE Generator Parameters

System Generator for Processors provides an easy way to parameterize the OPB UART Lite to your specific requirements. All XCO values must conform to the rules specified in Table 1. The following fields are provided in the System Generator graphical interface:

OPB Data Bus Width

Specifies the width in bits of the OPB data buses connected to this core. This field sets the C_OPB_DWIDTH parameter.

OPB Clock Frequency

Bus clock frequency in Hertz. Case insensitive. Syntax: <*real number*> <unit>, where unit = Hz, kHz, MHz, gHz. Example: 100.5 MHz. The VHDL genic C_OPB_CLK_PERIOD_PS will be set to equal 1/OPB clock frequency.

Data_Bits_Number

Set the number of data bits in the serial frame. This field corresponds to the VHDL generic C_DATA_BITS.

Device Block ID

Unique module identification number assigned by you. This field corresponds to the VHDL generic C_DEV_BLK_ID.

Base Address

Base address decoded by this core. This field corresponds to the VHDL generic C_BASEADDR.

Parity_Mode

Specifies odd or even parity. This field corresponds to the VHDL generic C_ODD_PARITY.

Use_Parity

Enables the inclusion of parity in this core. This field corresponds to the VHDL generic C_USE_PARITY.

Baud_Rate

Set the value of the baud rate of the UART Lite in bits per second. This field corresponds to the VHDL generic C_BAUDRATE.

End Address

High address decoded by this core. This field corresponds to the VHDL generic C_HIGHADDR.

OPB Address Bus Width

Specifies the width in bits of the OPB address buses connected to this core. This field sets the C_OPB_AWIDTH parameter.

UART Lite I/O Signals

The I/O signals for the UART Lite are listed in Table 2.

Table 2: **UART Lite I/O Signals**

Signal Name	Interface	I/O	Description
OPB_Clk	OPB	I	OPB Clock
OPB_Rst	OPB	I	OPB Reset
OPB_ABus[0:31]	OPB	I	OPB Address Bus
OPB_BE[0:3]	OPB	I	OPB Byte Enables
OPB_DBus[0:31]	OPB	I	OPB Data Bus
OPB_RNW	OPB	I	OPB Read, Not Write
OPB_select	OPB	I	OPB Select
OPB_seqAddr	OPB	I	OPB Sequential Address
UART_DBus[0:31]	OPB	O	UART Data Bus
UART_errAck	OPB	O	UART Error Acknowledge
UART_retry	OPB	O	UART Retry
UART_toutSup	OPB	O	UART Timeout Suppress
UART_xferAck	OPB	O	UART Transfer Acknowledge
Interrupt	Interrupt	O	UART Interrupt
RX	External	I	Receive Data
TX	External	O	Transmit Data

UART Lite Address Map and Register Descriptions

Register Data Types and Organization

Registers in the UART Lite are accessed as one of three types: byte (8 bits), halfword (2 bytes), and word (4 bytes). All register accesses are on word boundaries to conform to the OPB-IPIF register location convention. The addresses of the UART Lite registers are provided in the **Address Map** section.

The UART Lite registers are organized as big-endian data. The bit and byte labeling for the big-endian data types is shown in Figure 1.

.

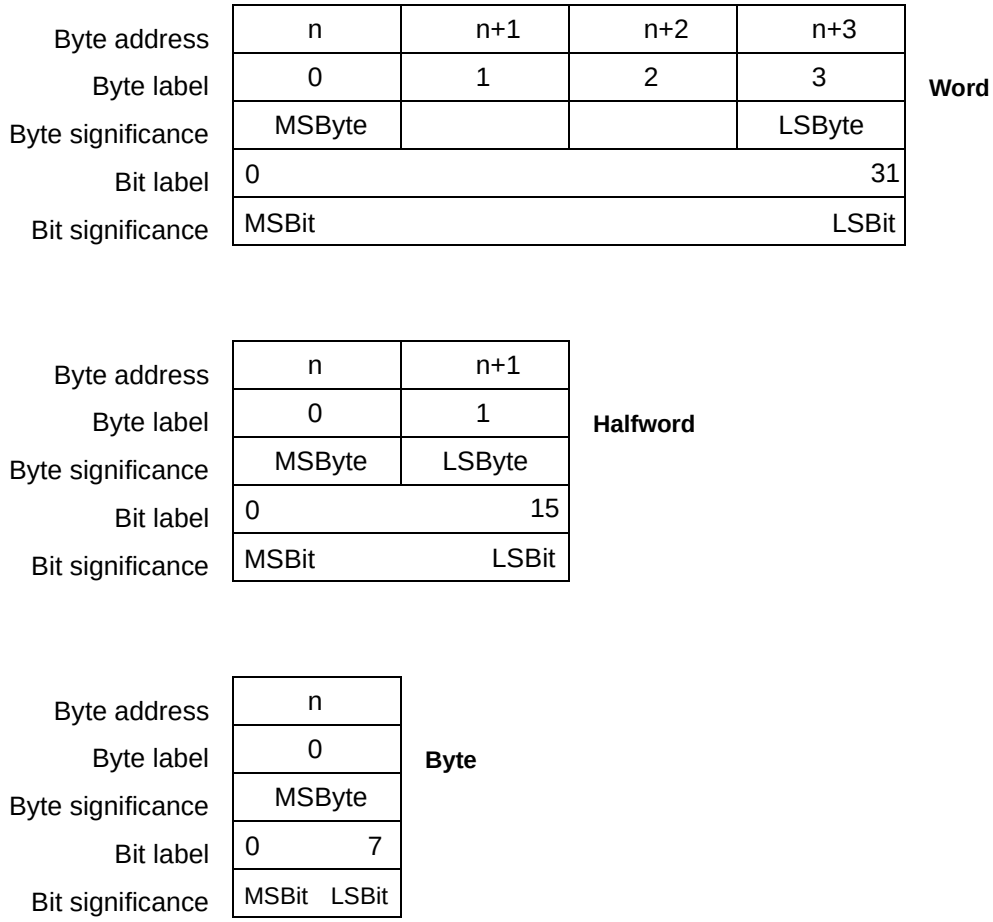

Figure 1: **Big-Endian Data Types**

Registers of the UART Lite

Information on the following registers used in assembly language programming are described in this section.

Receive FIFO	Read character from Receive FIFO
Transmit FIFO	Write character into Transmit FIFO
Status	Read from Status Register
Control	Write to Control Register

Figure 2: **UART Lite Register Set**

Status Register (STATREG)

The Status register contains the status of the receive and transmit FIFO, if interrupts are enabled, and if there are any errors.

Table 3: **Status Register**

Bits	Name	Description	Reset Value
0-23	Reserved	Not used	0
24	PAR_ERROR	Parity Error Indicates that a parity error has occurred since the last time the status register was read. If the UART is configured without any parity handling, this bit will always be '0'. The received character will be written into the receive FIFO. The bit will be cleared when the status register is read 0 No parity error has occurred 1 A parity error has occurred	0
25	FRAME_ERROR	Frame Error Indicates that a frame error has occurred since 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 will be ignored and NOT written to the receive FIFO. The bit will be cleared when the status register is read 0 No Frame error has occurred 1 A frame error has occurred	0
26	OVERUN_ERROR	Overrun Error Indicates that a overrun error has occurred since 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 will be ignored and NOT written into the receive FIFO. The bit will be cleared when the status register is read 0 No interrupt has occurred 1 Interrupt has occurred	0

Table 3: **Status Register *(Continued)***

Bits	Name	Description	Reset Value
27	INTR_ENABLED	Interrupts is enabled Indicates that interrupts is enabled 0 Interrupt is disabled 1 Interrupt is enabled	0
28	TX_FIFO_FULL	Transmit FIFO is full Indicates if the transmit FIFO is full. 0 Transmit FIFO is not full 1 Transmit FIFO is full	
29	TX_FIFO_EMPTY	Transmit FIFO is empty Indicates if the transmit FIFO is empty. 0 Transmit FIFO is not empty 1 Transmit FIFO is empty	
30	RX_FIFO_FULL	Receive FIFO is full Indicates if the receive FIFO is full. 0 Receive FIFO is not full 1 Receive FIFO is full	
31	RX_FIFO_VALID_DATA	Receive FIFO is has valid data Indicates if the receive FIFO has valid data. 0 Receive FIFO is empty 1 Receive FIFO has valid data	

Control Register (CTRL_REG)

The Control register contains the UART Lite control.

Table 4: **Control Register (CTRL_REG)**

Bits	Name	Description	Reset Value
0-26	Reserved	Not used	0
27	ENABLE_INTR	Enable Interrupt for the UART 0 Disable interrupt signal 1 Enable interrupt signal	0
28-29	Reserved	Not used	0
30	RST_RX_FIFO	Reset/Clear the receive FIFO When written to with a '1' the receive FIFO is cleared. 0 Do nothing 1 Clear the receive FIFO	0
31	RST_TX_FIFO	Reset/Clear the transmit FIFO When written to with a '1' the transmit FIFO is cleared. 0 Do nothing 1 Clear the transmit FIFO	0

Address Map

UART_BASE_ADDRESS + 0: Read from Receive FIFO

UART_BASE_ADDRESS + 4: Write to transmit FIFO

UART_BASE_ADDRESS + 8: Read from Status Register

UART_BASE_ADDRESS + 12: Write to Control Register

Interrupts

If interrupts are enabled, an interrupt is generated when one of the following conditions is true:

1. When there exists any valid character in the receive FIFO, the interrupt stays active until the receive FIFO is empty.

2. When the transmit FIFO goes from not empty to empty, such as when the last character in the transmit FIFO is transmitted, the interrupt is only active one clock cycle.

Design Implementation

Device Utilization and Performance Benchmarks

The following table shows approximate resource utilization and performance benchmarks for the OPB UART Lite. The estimates shown are not guaranteed and can vary with FPGA family and speed grade, implementation parameters, user timing constraints, and implementation tool version. Only parameters that affect resource utilization are shown in the following table.

Table 5: **OPB UART Lite Performance and Resource Utilization Benchmarks (Virtex-II 2V1000-5)**

Parameter Values							Device Resources		f_{MAX} (MHz)
Address Bits in Decode	C_AWIDTH	C_CLK_FREQ	C_BAUDRATE	C_DATA_BITS	C_USE_PARITY	C_ODD_PARITY	Flip-Flops	4-input LUTs	f_{MAX}
24	32	100_000_000	19_200	5	FALSE	FALSE	48	88	
24	32	100_000_000	19_200	6	FALSE	FALSE	49	92	
24	32	100_000_000	19_200	7	FALSE	FALSE	50	95	
24	32	100_000_000	19_200	8	FALSE	FALSE	51	100	
24	32	40_000_000	38_400	8	FALSE	FALSE	49	97	158
24	32	100_000_000	19_200	6	TRUE	FALSE	57	108	137
24	32	100_000_000	19_200	7	TRUE	FALSE	57	108	137

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/17/01	1.0	Initial release.
10/9/01	1.01	Updated MPD parameters and added device utilization table
7/1/02	1.02	Updated with new layout and renaming of parameters
3/20/02	2.0	Updated for MDK 2.2
5/28/02	2.1	Update for EDK 1.0
7/23/02	2.2	Add XCO parameters for System Generator
10/10/02	2.3	Step the version from a to b on the core
01/08/03	2.4	Update for EDK SP3