UART 16550 Transceiver

October 2012 Reference Design RD1138

Introduction

The Universal Asynchronous Receiver/Transmitter (UART) performs serial-to-parallel conversion on data characters received from a peripheral device or a modem, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, framing, or break interrupt).

The UART has complete modem-control capability, and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. The register set and data transfer protocol of this design is compatible with the National Semiconductor PC16550D UART.

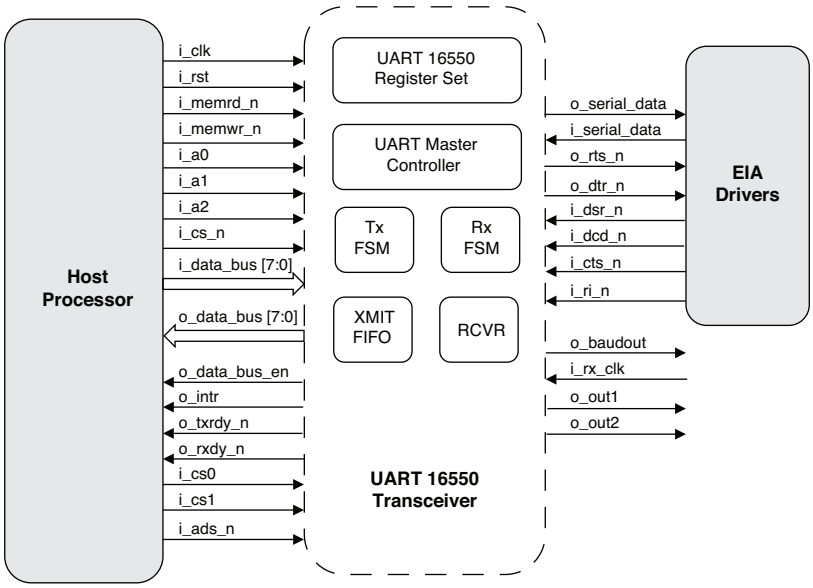
This reference design is implemented in Verilog. The Lattice iCEcube2™ Place and Route tool integrated with the Synplify Pro synthesis tool is used for implementation of the design. The design uses an iCE40™ ultra low density FPGA and can be targeted to other iCE40 family members.

Features

- Compatible with National Semiconductor PC16550D UART
- Configurable data widths of 5, 6, 7 or 8 bits
- Configurable stop bits – 1, 1.5 or 2 bits for transmit operations
- Even parity, odd parity or stick parity configuration for transmit and receive operations
- Programmable divisor latch for custom baud rates
- Interrupt generation logic with readable Interrupt Identification Register
- Verilog RTL, test bench and Aldec A-HDL script for simulation

Functional Description

Figure 1. System Block Diagram



Signal Descriptions

Table 1. Signal Descriptions

Signal	Width (Bits)	Type	Description
i_clk	1	Input	System clock
i_rst	1	Input	Asynchronous active high system reset
i_memrd_n	1	Input	Active low signal to read data or status from the selected UART 16550 register
i_memwr_n	1	Input	Active low signal to write data or control word to the selected UART 16550 register
i_a0	1	Input	LSB of 3-bit register select
i_a1	1	Input	Bit 1 of 3-bit register select
i_a2	1	Input	MSB of 3-bit register select
i_cs_n	1	Input	Active low chip select line
i_cs0	1	Input	Active high chip select. This pin is tied to Vcc.
i_cs1	1	Input	Active high chip select. This pin is tied to Vcc.
i_ads_n	1	Input	Active low address strobe line. This pin is tied to GND.
i_data_bus	8	Input	Input data bus
o_data_bus	8	Output	Output data bus
o_data_bus_en	1	Output	Data bus enable signal for connecting i_data_bus and o_data_bus to a tri-stated io_data_bus
o_txrdy_n	1	Output	When FCR[3] = 0 and there are no characters in the XMIT FIFO or the XMIT holding register then o_txrdy_n will be active low. Once it is activated, o_txrdy_n will be inactive after the first character is loaded into the XMIT FIFO or the holding register. When FCR[3] = 1 and there are no characters in the XMIT FIFO then o_txrdy_n will be active low and will go inactive once the XMIT FIFO is completely full.
o_rxrdy_n	1	Output	When FCR[3] = 0 and there is at least one character in the RCVR FIFO then o_rxrdy_n will be active low and will go inactive when there are no more characters in the FIFO. With FCR [3] = 1 and the trigger level or the timeout has been reached then o_rxrdy_n will be active low and will go inactive when there are no more characters in the FIFO.
o_intr	1	Output	Active high interrupt signal generated whenever corresponding interrupt types are active high in the Interrupt Enable register. This is reset low on a master reset or on an interrupt service.
o_baudout	1	Output	16x baud rate clock from the transmitter side of the design
i_serial_data	1	Input	Serial data input from the communication link
o_serial_data	1	Output	Serial data output to the communication link
o_dtr_n	1	Output	Active low data terminal ready signal. When low, this informs the modem that the UART is ready to establish a communication link. This can be made active low by programming bit 0 of the Modem Control Register. On reset, this bit is set to ‘1’.
o_rts_n	1	Output	Active low request to send signal. When low, this informs the modem that the UART is ready to exchange data. This can be made active low by programming Bit 1 of the Modem Control Register. On reset, this bit is set to ‘1’.
i_ri_n	1	Input	Active low ring indicator signal. When active low, indicates that the modem has received a telephone ringing signal.
i_dsr_n	1	Input	Active low data set ready signal. When low, this indicates that the modem is ready to establish a communication link with the UART.
i_dcd_n	1	Input	Active low data carrier detect. When low, this indicates that the modem has detected the data carrier.
i_cts_n	1	Input	Active low clear to send signal. When low, this indicates that the modem is ready to exchange data.
i_rx_clk	1	Input	Receiver clock is 16x baud rate clock for the receiver section.

Design Module Description

The UART 16550 Transceiver consists of a register set interface block, UART Master Controller block, UART Tx FSM block, UART Rx FSM block and UART transmit/receive FIFOs as shown in Figure 1.

Register Set Interface Block

The main function of the register set interface block is to receive and transmit information to the host processor using the asynchronous processor interface. The register set interface block delivers control information to the UART master controller. It also gathers status information and updates the register set. This block is responsible for generation of the interrupt.

Table 2. Internal Register List

Name	Address	DLAB	Width	Access	Description
RBR	0x00	0	8	R	Receive Buffer Register
THR	0x00	0	8	W	Transmitter Holding Register
IER	0x01	0	8	RW	Interrupt Enable Register
IIR	0x02	X	8	R	Interrupt Identification Register
FCR	0x02	X	8	W	FIFO Control Register
LCR	0x03	X	8	RW	Line Control Register
MCR	0x04	X	8	RW	Modem Control Register
LSR	0x05	X	8	R	Line Status Register
MSR	0x06	X	8	R	Modem Status Register
DLR (LSB)	0x00	1	8	RW	Divisor Latch Register (LSB)
DLR (MSB)	0x01	1	8	RW	Divisor Latch Register (MSB)

The 3-bit register select bits are used to select a UART 16550 Transceiver register for the CPU to read from or write to during data transfer. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the line control register, affects the selection of certain UART 16550 Transceiver registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

In UART 16550 mode the data received will be stored in the **Receive Buffer Register**. A read operation provides the received data. Data to be transmitted will be written to the **Transmitter Holding Register**.

The **Interrupt Enable Register** enables the four types of UART interrupts. Each interrupt can individually activate the interrupt (o_intr) output signal.

The **Interrupt Identification Register** bits, interrupt conditions and priority are summarized in Table 3. In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status, Received Data Ready, Transmitter Holding Register Empty and Modem Status.

The **Line Control Register** configures character length, number stop bits and parity bit.

The **Modem Control Register** controls the interface with the modem or data set (or a peripheral device emulating a modem).

The **Line Status Register** provides status information to the CPU concerning data transfer.

The **Modem Status Register** provides the current state of the control lines from the modem (or peripheral device) to the CPU. In addition to this current state information, four bits of the Modem Status Register provide change information. These bits are set to a logic ‘1’ whenever a control input from the modem changes state. They are re-set to logic ‘0’ whenever the CPU reads the Modem Status Register.

Divisor Latch Register

The UART contains a programmable baud generator that is capable of taking any clock input (< 24 MHz). Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latch Registers must be loaded during initialization to ensure proper operation of the baud generator. Upon loading either of the Divisor Latch Registers, a 16-bit Baud Counter is immediately loaded. The divisor is calculated as follows: Divisor = (frequency input)/ (baud rate x 16). Table 4 shows the values of the Divisor Latch Register for the supported baud rates for a system clock of 18.432 MHz.

Table 3. Interrupt Identification Register

Bit3	Interrupt Identification Register			Interrupt Set Reset Functions			
	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	—	None	None	—
0	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the line status register
0	1	0	0	Second	Received data available or trigger level reached	Received data available or trigger level reached	Reading the Receive Buffer Register
1	1	0	0	Second	Character timeout indication	No characters have been input to the RCVR FIFO during the last four character times and there is at least one character in it during this time	Reading the Receive Buffer Register
0	0	1	0	Third	Transmitter holding register empty	Transmitter Holding Register empty	Writing for Transmitter Holding Register
0	0	0	0	Fourth	Modem status	Clear to Send, Data Set Ready, RI or Data Carrier detect	Reading the Modem Status Register

Table 4. Supported Baud Rates with Equivalent Divisor Latch Inputs

Baud Rate	Divisor Latch (MSB, LSB)
50	23040
75	15360
110	10473
134.5	8565
150	7680
300	3840
600	1920
1200	920
1800	640
2000	576
2400	480
3600	320
4800	240
7200	160
9600	120
19200	60
38400	30
56000	21
128000	9

Table 5. Description of Internal Register Bits

Internal Register	Bit #	Access	Description
Receive Buffer Register (0x00)	7:0	R	Holds received data
Transmitter Holding Register (0x00)	7:0	W	Holds data to be transmitted
Interrupt Enable Register(0x01)	7:4	RW	Unimplemented
	3	RW	'1' enables Modem Status Interrupt
	2	RW	'1' enables Receiver Line Status Interrupt
	1	RW	'1' enables Transmitter Holding Register Empty Interrupt
	0	RW	'1' enables Received Data Available Interrupt
Interrupt Identification Register (0x02)	7:3	R	Unimplemented
	2:0	R	'001' None '110' Receiver Line Status Interrupt '100' Received data available '010' Transmitter Holding Register Empty '000' Modem Status Interrupt Refer to Table 3 for more details about IIR

Table 5. Description of Internal Register Bits (Continued)

Internal Register	Bit #	Access	Description
FIFO Control Register	7:6	W	Set trigger level of RCVR FIFO '00' – Trigger level 1 byte '01' – Trigger level 4 bytes '10' – Trigger level 8 bytes '11' – Trigger level 14 bytes
	5:4	W	Unimplemented
	3	W	'1' will cause o_rx_rdy_n and o_tx_rdy_n pins to change modes
	2	W	'1' resets the XMIT FIFO
	1	W	'1' resets the RCVR FIFO
	0	W	'1' enables both XMIT and RCVR FIFOs
Line Control Register	7	RW	Divisor Latch Access Bit '1' to access Divisor Latch Register '0' to access RBR/THR or IER
	6	RW	Break Control Bit '1' enables break control condition
	5	RW	Sticky Parity Bit When Bits 3, 4 and 5 are logic '1', the parity bit is transmitted and checked as a logic '0'. When Bits 3 and 5 are logic '1' and Bit 4 is logic '0' then the parity bit is transmitted and checked as a logic '1'.
	4	RW	Even Parity Select bit When Bit 3 is logic '1' and bit 4 is logic '0', an odd number of logic '1's are transmitted and checked in data word bits When Bit 3 is logic '1' and Bit 4 is logic '1', an even number of logic '1's are transmitted and checked in data word bits
	3	RW	Parity Enable bit '1' enables parity
	2	RW	Specifies number of stop bits '0' – One stop bit is generated in the transmitted data '1' – If LCR [1:0] = '00' then 1.5 stop bits are generated and if LCR [1:0] = '01' or '10' or '11' then two stop bits are generated
	1:0	RW	'00' selects a character length of 5 bits '01' selects a character length of 6 bits '10' selects a character length of 7 bits '11' selects acharacter length of 8 bits
MODEM Control Register	7:2	RW	Unimplemented
	1	RW	Controls Request to Send (RTS) output
	0	RW	Controls Data Terminal Ready output

Table 5. Description of Internal Register Bits (Continued)

Internal Register	Bit #	Access	Description
Line Status Register	7	R	Logic ‘0’
	6	R	‘1’ indicates Transmitter FIFO and Transmitter Shift Register are both empty
	5	R	‘1’ indicates Transmitter Holding Register empty
	4	R	‘1’ indicates break condition
	3	R	Framing Error Indicator ‘1’ indicates received character did not have a valid stop bit
	2	R	Parity Error Indicator ‘1’ indicates received character did not have a correct parity
	1	R	Overrun Error Indicator ‘1’ occurs when RCVR FIFO is full
Modem Status Register	0	R	‘1’ indicates Receiver Data ready
	7	R	Complement of Data Carrier Detect input
	6	R	Complement of Ring Indicator input
	5	R	Complement of Data Set Ready input
	4	R	Complement of Clear to Send input
	3	R	Delta Data Carrier Detect indicator – Indicates the DCD input has changed state
	2	R	Trailing Edge of Ring indicator detector – Indicates RI input has changed state from a low to a high state
	1	R	Delta Data Set Ready indicator – Indicates DSR input has changed state since last time it was read by the CPU
	0	R	Delta Clear to Send indicator – Indicates CTS input has changed state since last time it was read by the CPU
Divisor Latch (LSB)	7:0	RW	Holds LSB Divisor Latch value
Divisor Latch (MSB)	7:0	RW	Holds MSB Divisor Latch value

FIFO Interrupt Mode Operation

With IER0 set to ‘1’, RCVR interrupts will occur as follows:

1. The Receiver Data Available trigger will be issued to the CPU when the FIFO has reached its programmed trigger level and the interrupt is cleared as soon as the FIFO falls below the programmed trigger level.
2. The IIR Receiver Data Available indication also occurs when the FIFO trigger level is reached, and like the interrupt, it is cleared when the FIFO drops below the trigger level.
3. The Receiver Line Status interrupt (IIR06), as before, has higher priority than the Received Data Available (IIR04) interrupt.
4. The Data Ready Bit (LSR0) is set as soon as a character is transferred from the shift register to the RCVR FIFO. It is reset when the FIFO is empty.
5. A FIFO timeout interrupt will occur if all the following conditions exist:
   a. There is at least one character in the FIFO
   b. The most recent serial character received was longer than four continuous character times ago (one character is eight data bits, one start bit, one parity bit and two stop bits)
   c. The most recent CPU read of the FIFO was longer than four continuous character times ago.
   d. When a timeout interrupt has occurred, the FIFO is cleared. The timer resets when the CPU reads one character from the RCVR FIFO.

With IER1 set to '1', XMIT interrupts will occur as follows:

- The Transmitter Holding Register interrupt occurs when the XMIT FIFO is empty; it is cleared as soon as the Transmitter Holding Register is written to or the IIR is read.

The UART Master Controller block forms the heart of the design. This block decodes the control data and configures the UART Rx FSM block and the UART Tx FSM block as per configuration data. The steps involved in transmit and receive operations is briefly outlined below.

UART Transmit Operation

To start a transmit operation, the user sets Bit 1 of the Modem Control Register (MCR). This de-asserts the o_rts_n. In response to o_rts_n, the modem de-asserts the i_cts_n line. The UART 16550 IP detects this and a start condition is generated followed by the data transfer and termination of operation as per the UART protocol. Transmit data by the user will be provided as per the format shown in Figure 2.

Figure 2. Tx Data Format

5-Bit Data

0	0	0	D4	D3	D2	D1	D0

6-Bit Data

0	0	D5	D4	D3	D2	D1	D0

7-Bit Data

0	D6	D5	D4	D3	D2	D1	D0

8-Bit Data

D7	D6	D5	D4	D3	D2	D1	D0

UART Receive Operation

On receiving the i_dsr_n signal, the user sets Bit 0 of the Modem Control Register. This de-asserts the o_dtr_n signal. The UART 16550 IP then waits for the start condition on the i_serial_data line. Once the operation is terminated, the receive data is validated and the valid data is made available in the Receive Buffer Register. The format of the received data stored in the Receive Buffer Register is shown in Figure 3.

Figure 3. Rx Data Format

5-Bit Data

0	0	0	D4	D3	D2	D1	D0

6-Bit Data

0	0	D5	D4	D3	D2	D1	D0

7-Bit Data

0	D6	D5	D4	D3	D2	D1	D0

8-Bit Data

D7	D6	D5	D4	D3	D2	D1	D0

Timing Diagrams

Figure 4. Timing Diagram of Write Cycle

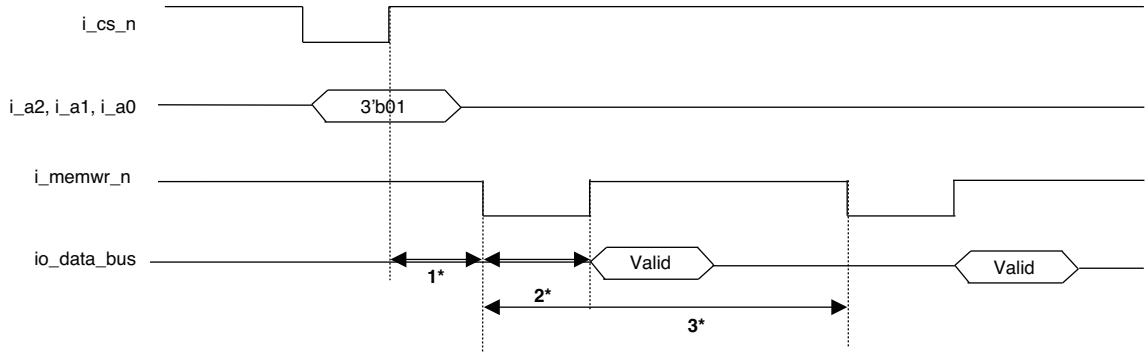


Figure 4 shows the timing diagram of a write to an internal register (e.g., FIFO Control Register with address 3'b010) of the UART 16550 transceiver IP.

The timing to be maintained is listed below:

- 1* – Between the rising edge of i_cs_n and following the falling edge of i_memwr_n should be at least one system clock period.
- 2* – Between the falling edge of i_memwr_n and its rising edge should be at least one system clock period.
- 3* – Between the falling edge of i_memwr_n and the next falling edge should be at least four system clock periods.

Figure 5. Timing Diagram of Read Cycle

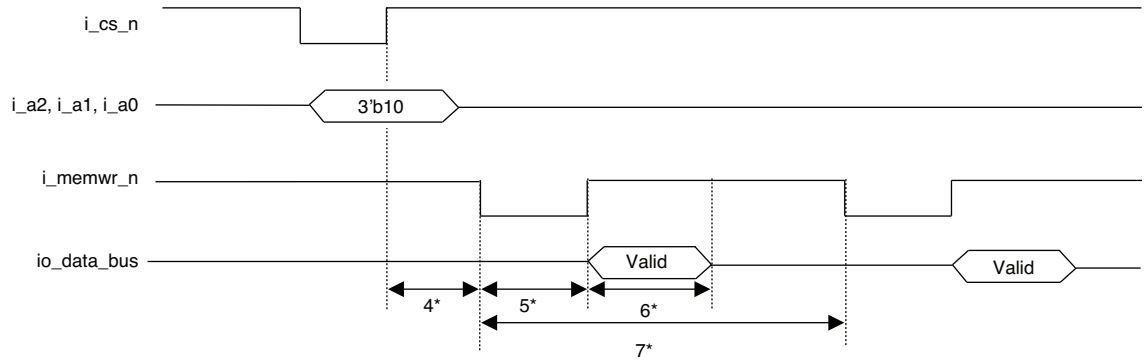


Figure 5 shows the timing diagram to the write to an internal register (e.g. Line Status Register with address 3'b101) of the UART 16550 transceiver IP.

The timing to be maintained is listed below:

- 4* – Between the rising edge of i_cs_n and the following falling edge of i_memrd_n should be at least one system clock period.
- 5* – Between the falling edge of i_memwr_n and its rising edge should be at least three system clock periods.
- 6* – Valid data availability after i_memrd_n is pulled high is a minimum of one clock period and a max of 1.5 clock periods.
- 7* – Between the falling edge of i_memwr_n and the next falling edge should be at least four system clock periods.

Figure 6. Modem-Side Receive Operation Timing Diagram

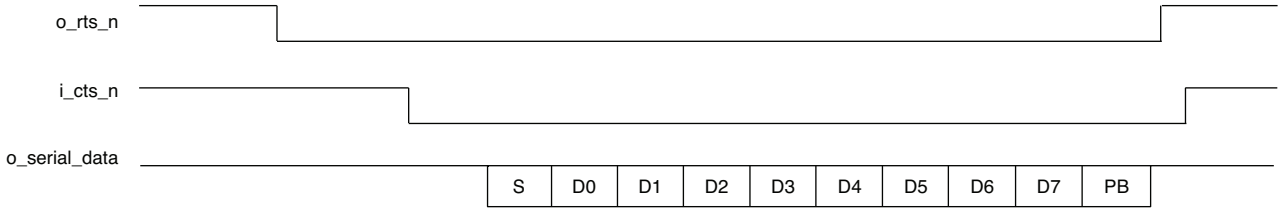


Figure 6 shows the receive operation from the modem side. In response to the o_rts_n de-asserted by the UART, the modem de-asserts the i_cts_n line. The master then sends the data on the o_serial_data line as per the UART data format as shown in Figure 8.

Figure 7. Modem-Side Transmit Operation Timing Diagram

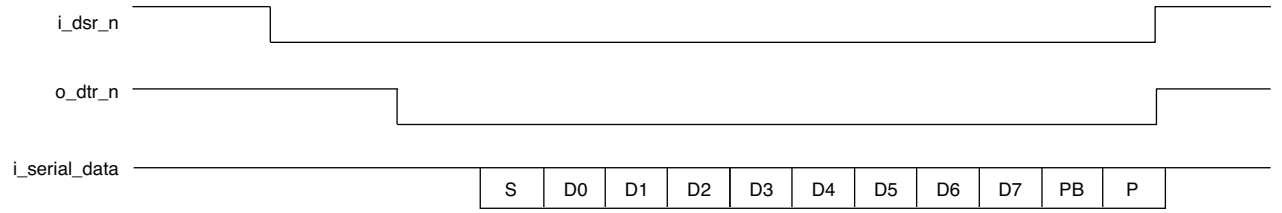
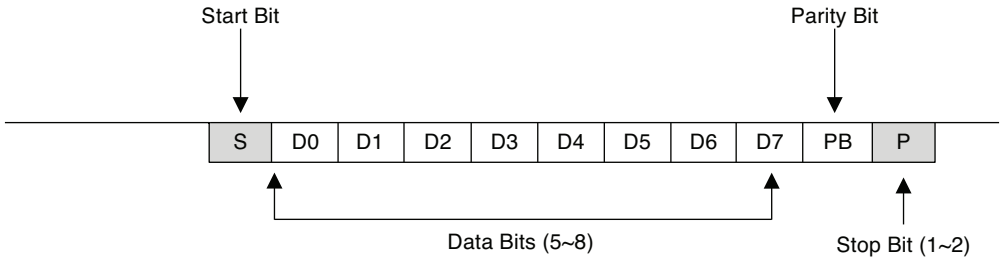


Figure 7 shows the transmit operation from the modem side. The slave de-asserts the i_dsr_n line signalling to the UART that it has data to send. The UART then de-asserts the o_dtr_n line when it is ready to receive data. The modem then initiates the transaction on the i_serial_data.

Figure 8. UART Data Format



Operation Sequence

The UART 16550 Transceiver is used for implementing the interface for serial communications. It is frequently used in implementation of serial ports which are then used for connections with modems, serial mice, printers and similar peripherals using the RS-232 interface.

Figure 1 shows one of the possible scenarios in which the UART 16550 Transceiver can be used.

The procedures for the transmit and receive operations are outlined below.

Transmit Operation

Let us consider an example of a transmit operation at a baud rate of 9600. The data format is a data width of 5 bits with even parity enabled and 1.5 stop bits; interrupt is disabled. The system clock is 18.432 MHz. All write and read to/from the internal register set are as shown in Figures 4 and 5.

The following steps are to be followed for carrying out this transaction:

1. De-assert the system reset i_rst signal.
2. Set baud rate of operation to 9600. This is a three step procedure:

- Step 1: A write to Line Control Register – The DLAB bit (i.e, Bit 7 of the Line Control Register) is first set to logic '1'. The register select inputs, i_a2, i_a1 and i_a0, are set to logic '0', logic '1' and logic '1' respectively. The data on the io_data_bus is 8'b1000_0000.
- Step 2: A write to Divisor Latch (LSB) and Divisor Latch (MSB) – To write to DLR (LSB), the register select inputs, i_a2, i_a1 and i_a0, are set to logic '0', logic '0' and logic '0'. 8'd120 is written to the Divisor Latch Register (LSB). To write to DLR (MSB), the register select inputs i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '1'. 8'd0 is written to the Divisor Latch Register (MSB).
- Step 3: A write to Line Control Register – The DLAB bit (i.e Bit 7 of the Line Control Register) is reset to logic '0'. The register select inputs, i_a2, i_a1 and i_a0, are set to logic '0', logic '1' and logic '1' respectively. The data on the io_data_bus is 8'b0000_0000.

3. A write to the Line Control Register to configure for the required data format (i.e., a data width of five bits) with even parity enabled and 1.5 stop bits.
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '1', logic '1'.
   - For 5-bit operation, Bit 0 and Bit 1 of the LCR is set to logic '0' and logic '0'.
   - For even parity enable, Bit 3 and Bit 4 of the LCR is set to logic '1'.
   - For enabling 1.5 stop bits, Bit 2 of the LCR is set to logic '1'.
   - Thus, data written to LCR is 8'b000_11_1_00.
4. A write to the Interrupt Enable Register to configure for interrupt disable
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '1'.
   - Data written to IER is 8'b0000_0000.
5. A write to the FIFO Control Register
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '1' and logic '0'.
   - Data written to FCR is 8'b0000_0001.
6. A write to Transmitter Holding Register (THR)
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '0'.
   - Data written to THR is transmit data.
7. The above write procedure to THR is repeated four more times. After this, transmit data is available in the XMIT FIFO.
8. A write to the Modem Control Register. To start the transmission procedure, set the Request to Send (RTS) bit, i.e. MCR Bit 1.
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '0' and logic '0'.
   - Data written to MCR is 8'b0000_0010.
9. When the Clear To Send (CTS) is pulled low by the modem the transmit process begins and terminates with the transmission of stop bits.
10. Line Status Register read. If bit 5 is set to one then the Transmitter Holding Register is empty.
    - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '0' and logic '1'.
    - If Bit 5 is set to '1', then data has been transmitted and the Transmitter Holding Register is now empty.
11. A write to the Modem Control Register. Reset the Request to Send (RTS) bit, i.e. MCR Bit 1.
    - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '0' and logic '0'.
    - Data written to MCR is 8'b0000_0000.

Receive Operation

Let us consider an example of a receive operation at a baud rate of 7200. The data format is a data width of eight bits with even stick parity enabled and one stop bit; interrupt is disabled. The system clock is 18.432 MHz. All write and read to/from the internal register set are as shown in Figures 4 and 5.

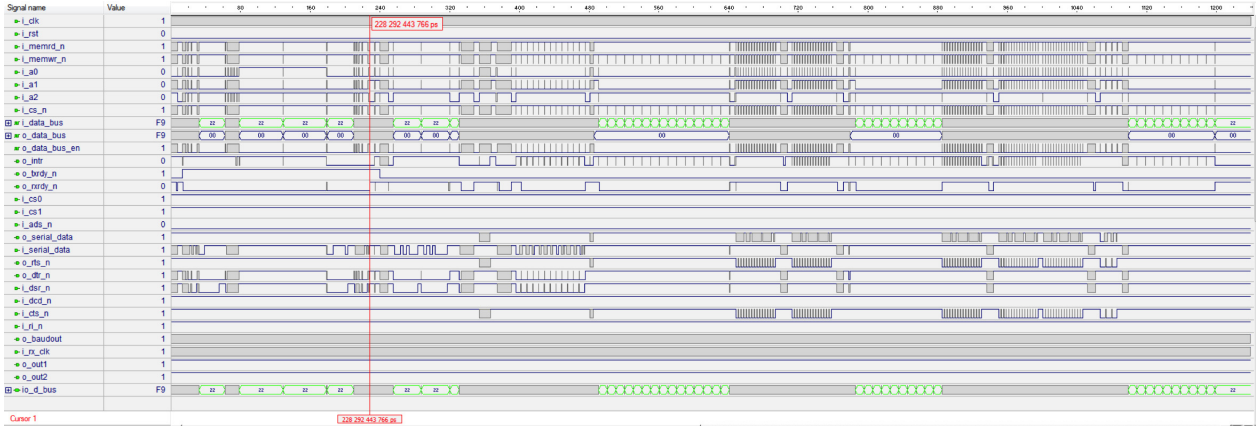
1. De-assert the system reset i_rst signal.
2. Set baud rate of operation to 7200. This is a three step procedure:
   - Step 1: A write to the Line Control Register – The DLAB bit (i.e. Bit 7 of the Line Control Register) is first set to logic '1'. Register select inputs i_a2, i_a1 and i_a0 are set to logic '0', logic '1' and logic '1' respectively. The data on the io_data_bus is 8'b1000_0000.
   - Step 2: A write to Divisor Latch(LSB) and Divisor Latch(MSB) – To write to DLR (LSB), the register select inputs i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '0'. 8'd160 is written to the Divisor Latch Register (LSB).

     To write to DLR (MSB), the register select inputs i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '1'. 8'd0 is written to the Divisor Latch Register (MSB).
   - Step 3: A write to Line Control Register – DLAB bit (i.e. Bit 7 of the Line Control Register) is reset to logic '0'. The register select inputs i_a2, i_a1 and i_a0 are set to logic '0', logic '1' and logic '1' respectively. The data on the io_data_bus is 8'b0000_0000.
3. A write to the Line Control Register to configure for the required data format (i.e. a data width of eight bits with even stick parity enabled and one stop bit).
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '1', logic '1'.
   - For 8-bit operation, Bit 0 and Bit 1 of LCR is set to logic '1' and logic '1'.
   - For even stick parity enable, Bit 3, Bit 4 and Bit 5 of LCR are set to logic '1'.
   - For enabling one stop bit, Bit 2 of LCR is set to logic '0'.
   - Thus, data written to LCR is 8'b00111_0_11.
4. A write to Interrupt Enable Register to configure for interrupt disable
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '0' and logic '1'.
   - Data written to IER is 8'b0000_0000.
5. A write to FIFO Control Register
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '0', logic '1' and logic '0'.
   - Data written to FCR is 8'b0000_0001.
6. Modem Status Register read
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '1' and logic '0'.
   - If Bit 5 is set to '1' then the modem is ready to send data.
7. A write to the Modem Control Register. A logic '1' is written to MCR Bit 0. This is Data Terminal Ready which is a response to the modem's Data Set Ready.
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '0' and logic '0'.
   - Data written to MCR is 8'b0000_0001.
8. Serial data receive operation takes place on the modem side. This is invisible on the processor side.
9. Modem Status Register read
   - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '1' and logic '0'.
   - If Bit 5 is set to '1' then the modem is not ready to send data.
10. A write to the Modem Control Register. A logic '0' is written to MCR Bit 0. This is Data Terminal Ready which is a response to the modem's Data Set Ready. This is pulled high as the modem is not ready to send new data yet.
    - Address select bits i_a2, i_a1 and i_a0 are set to logic '1', logic '0' and logic '0'.
    - Data written to MCR is 8'b0000_0000.
11. Steps 5, 6, 8 and 9 are repeated until there is no longer any data to be sent by the modem.

12. Receive Buffer Register (RBR) read. o_rxrdy_n is active low when there is data available in the FIFO to read.
    - Address select bits i_a2, i_a1 and i_a0 are set to logic ‘0’, logic ‘0’ and logic ‘0’.
    - Data is read until o_rxrdy_n goes high.

Simulation Waveforms

Figure 9. Simulation Waveforms



Implementation

This design is implemented in Verilog. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Table 6. Performance and Resource Utilization

Family	Language	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architectural Resources
iCE40[1]	Verilog	622	>100	29	N/A

1. Performance and utilization characteristics are generated using iCE40-LP1K-CM121 with iCEcube2 design software.

References

- iCE40 Family Handbook

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
October 2012	01.0	Initial release.