Jordan University of Sciecne and technolgoy (JUST)
Computer Engineering Department
CPE 746 Design Project - New UART Module for SPAM
Fall - 2006
Dr. Lo'ai Tawalbeh

1 Introduction

1.1 Revisions

This is a revised version of the UART module (Version 1.1)

1.2 Background

The UART (universal asynchronous receiver transmitter) described below implements a full-duplex asynchronous receiver and transmitter. The transmitter converts parallel data from a CPU to a serial bit stream, inserting appropriate start, stop, and parity bits. It outputs the resulting stream on the channel transmitter serial data output.

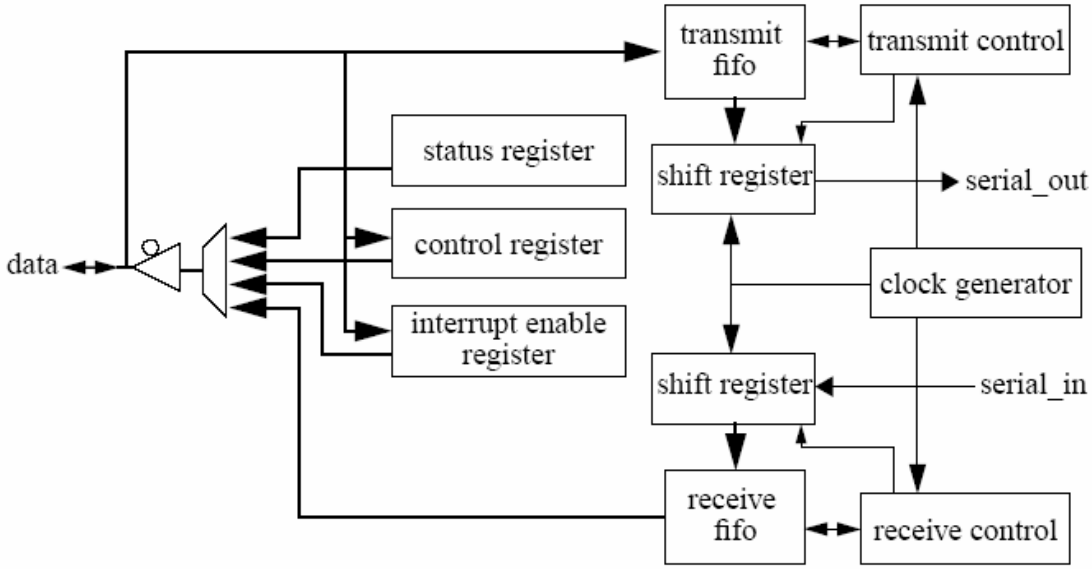
The receiver converts serial data from the channel receiver serial data input to a parallel format, checks for a start, stop, and parity bits, and transfers the assembled character onto the bus during read operations.

The UART contains two FIFOs that can each hold sixteen bytes of data. One of these will hold data for the transmitter; the other will hold the data for the receiver. These FIFOs give the attached CPU extra time to respond to interrupts from the UART.

Our project is to design a simple microcontroller (SPAM) suitable for eventual FPGA implementation.

2.0 Top Level Block Diagram

The top level block diagram of the UART is seen below.



2.1 Top Level Pin Definition

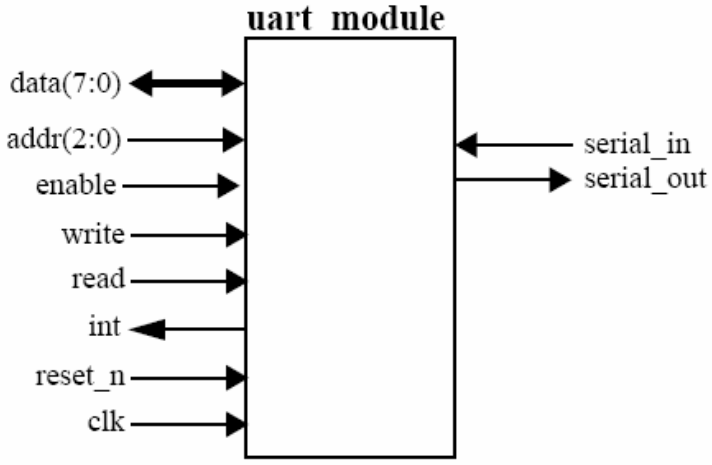


FIGURE 1. UART Module Top Level Pin Definition

2.1.1 Port Definition

Pin	Function
data(7:0)	Data bus, 8-bit, MSB= bit 7. This tri-state bus is the external data interface to the CPU data bus.
addr(2:0)	This pin selects the internal status register or the internal data register. "000" - data register "001" - control register "010" - status register "011" - interrupt enable register All other codes are undefined
read	This active high input is used to pass the value of internal registers out to the data port based on the value at the 'addr' input.
reset_n	Input pin . Active low asserted. Provides asynchronous global reset to the UART module. It clears all internal registers.
clk	The 4Mhz clock input for the UART.
write	This active high input is used to load a value present on the 'data' port into one of the internal registers, based on the value on the 'addr' input.
enable	This is an active high enable. This signal must be high to allow for a write or read to the internal registers occur.
int	The active high output indicates that an enabled interrupt source within the UART has been asserted.
serial_out	Output pin. This is the serial data output line.
serial_in	Input pin. The is the serial data input line.

TABLE 1. Port Definition for UART Module

3.0 Registers

3.1 Status Register

This eight-bit register holds the following UART status information.

7	6	5	4	3	2	1	0
RPE	ROE	RFE	INT	TXMT	TXHF	RXNMT	RXHF

Bit 7: **Receive Parity Error (RPE) -** Bit 7 indicates that the received data character does not have the correct even or odd parity, as selected by the data protocol selection in the control register. **RPE** is set to logic "1" upon detection of a parity error and is held until reset to logic "0" by the CPU reading the contents of the control register.
Bit 6: **Receive Overrun Error (ROE) -** Bit 6 indicates that the receiver FIFO was in a "full" condition when new data came in and was not able to be stored in the receive FIFO. Data was lost because apparently the CPU did not service the receive FIFO quickly enough. **ROE** is set to logic "1" immediately upon detection of the overrun condition and is held until reset by the CPU reading the contents of the control register.
Bit 5: **Receive Framing Error (RFE) -** Bit 5 indicates that the received data byte did not have a valid Stop bit. **RFE** is set to logic "1" whenever the stop bit following the last data

bit or parity bit is detected as a logic "0" bit. This bit is reset whenever the CPU reads the contents of the control register.

Bit 4: **Interrupt (INT) -** Bit 4 indicates that one of the enabled interrupt sources has become asserted. Individual interrupt sources are enabled by the contents of the interrupt enable register. **INT** is asserted logic "1" whenever an enabled interrupt occurs. The bit remains asserted until either the global interrupt enable (interrupt enable register bit 4) bit is set to logic "0" or the source of the interrupt goes away.

Bit 3: Transmit FIFO Empty (TXMT) - TXMT indicates that the transmit FIFO is empty. This bit stays asserted as long as the empty condition is met.

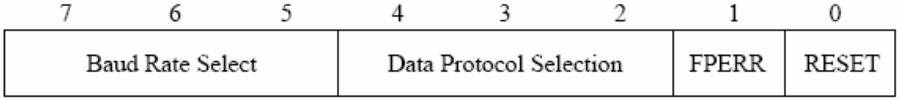
Bit 2: **Transmit FIFO Half Full (TXHF) - TXHF** indicates that the transmit FIFO is less than or equal to one-half full. This bit stays asserted as long as the half full condition is met.

Bit 1: **Receive FIFO Not Empty (RXNMT) - RXNMT** indicates that the receive FIFO has some data in it that can be read. This bit stays asserted as long as the not empty condition is met.

Bit 0: **Receive FIFO Half Full (RXHF) - RXHF** indicates that the receive FIFO is greater than or equal to one-half full. This bit stays asserted as long as the half full condition is met.

3.2 Control Register

This eight-bit register stores the following UART control and configuration information.



Bit 7-5: **Baud Rate Select -** Bits 7,6, and 5 are used to select the baud rate for reception and transmission. The baud rates are selected as follows:

Bit setting	Baud Rate
000	600
001	1200
010	2400
011	4800
100	9600
101	19,2000
110	38,4000
111	unused

FIGURE 2. Baud Rate Selection

Bit 4-2: Data Protocol Selection - Bits 4, 3, and 2 select the data protocol for the UART.

The protocols are selected as follows:

Bit Setting	Data Bits	Parity	Stop Bits
000	7	even	2
001	7	odd	2
010	7	even	1
011	7	odd	1
100	8	none	2
101	8	none	1
110	8	even	1
111	8	odd	1

FIGURE 3. Data Protocol Selection

Bit 1: **Force Parity Error (FPERR) -** When asserted logic "1', this bit will invert the correct state of the parity bit on each outgoing data byte. Its purpose is to check the function of the parity check logic.
Bit 0: **UART Reset (RESET) -** When asserted to logic "1" all internal state of the UART is reset. This includes the transmit and receive FIFOs. This bit is intended as a software reset. It provides the software a way to reset the UART when data or framing errors have occurred. This bit will automatically be reset to logic "0" after being set because of the reset performed, also resets the reset bit.

3.3 Data Register

This eight-bit register is where outgoing or incoming data is either written or read respectively. When this register is read, receive data from the receive FIFO is returned on the data bus. When this register is written, the data is written into the transmit FIFO. Therefore as such, there is no actual, physical data register, but rather a source and sink of data presented to the programmer.

3.4 Interrupt Enable Register

This eight bit register enables the individual interrupt sources within the UART. The Interrupt Enable bits are assigned as follows:

7	6	5	4	3	2	1	0
RPE_EN	ROE_EN	RRE_EN	GI_EN	TXMT_EN	TXHF_EN	RXNMT	RXHF_EN

Bit 7: - Receive Parity Error Interrupt Enable (RPE_EN)
Bit 6: - Receive Overrun Interrupt Enable (ROI_EN)
Bit 5: - Receive Framing Error Interrupt Enable (RFE_EN)
Bit 4: - Global Interrupt Enable (GI_EN)
Bit 3: - Transmit FIFO Empty Interrupt Enable (TXMT_EN)
Bit 2: - Transmit FIFO 1/2 Full Interrupt Enable (TXHF_EN)
Bit 1: - Receive FIFO Not Empty Interrupt Enable (RXNT_EN)
Bit 0: - Receive FIFO 1/2 Full Interrupt Enable (RXHF_EN)

For each interrupt source inside the UART there is an individual interrupt enable bit. The bit position of the interrupt enable bits corresponds to the bit position of the interrupt

sources in the status register. Bit 4 in the status register is the global interrupt indication bit. It corresponds to the global interrupt enable in the interrupt enable register.

4.0 System initial state

When the global reset signal *reset_n* is asserted, all internal state including data storage and state machines will be reset to a known state of '0' or '1'. The UART serial data output will be asserted logic "1". When reset_n is unasserted and prior to any system clocks being supplied, UART output signals will assume the following states:

pin	state
data_out(7:0)	0xZZ (high Z)
serial_out	1 (deasserted)

TABLE 2. UART Output Pin Reset State

5.0 Clock Input

The UART will operate with a single input clock of 4Mhz. All other internal clocks required will be derived from this clock

6.0 UART Transmission

The UART transmitter converts parallel data from the CPU to a serial bit stream on *serial_out*. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit (LSB) is sent first.

Parity, when included in the transmission, will be computed only from the data bits.

After the stop bits are sent, if no new character is in the transmitter FIFO, the output *serial_out* remains high and the transmit FIFO empty bit in the status register is set. Transmission resumes and the transmit FIFO empty bit is cleared when the CPU loads a new character into the transmit FIFO.

If the UART is reset through software command, the transmit operation stops immediately.

6.1 Writing to the UART module

For the CPU to write to the UART, data is placed on the 'data' port as the destination address is asserted on the addr port. The write input is asserted for one cycle to complete the write cycle. See the timing diagram below.

If a write is attempted to the data register when the transmit FIFO is full, the new data is not written into the FIFO. Data in the transmit FIFO is preserved and the Transmit FIFO

Full bit will remains high.

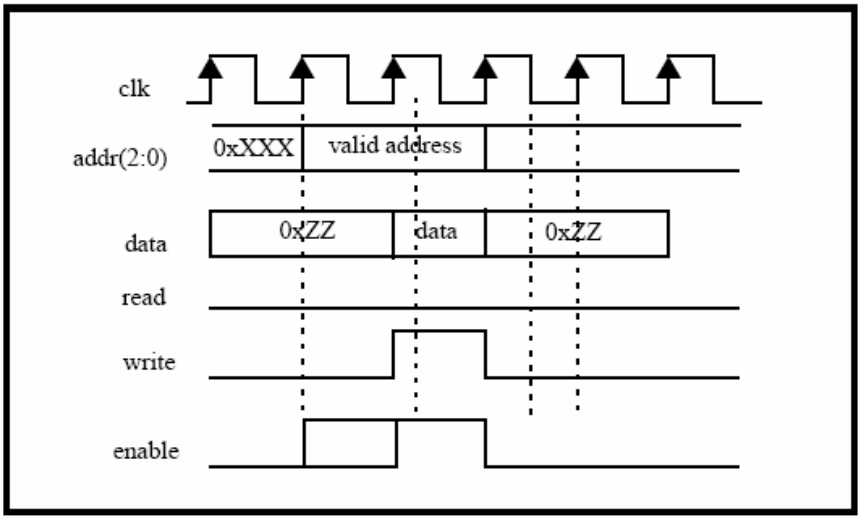


FIGURE 4. Writing to the UART module

7.0 UART Reception

When the UART receiver logic detects a high-to-low transition of the start bit on *serial_in*, the state of serial_in is sampled after a one-half bit frame delay. If the state of *serial_in* is still low, then a valid start has been detected, if not the receiver logic begins searching again for a valid low level on *serial_in*. This behavior is taken so that noise pulses on the serial_in line may be detected and not mistaken as actual start bits.

If receiver logic finds serial_in still low after the one-half bit frame, a valid start bit is assumed and the receiver continues sampling the input at one-bit time intervals, at the approximate center of the bit, until the proper number of data bits and parity if any, is assembled and one stop bit is detected. The LSB is received first. The data is then transferred to the receive FIFO. If the byte is less than 8 bits, then bit 7 will be forced to be zero when the data is written into the receive FIFO.

When parity checking is enabled, parity will be computed from the data bits and the parity bit. The modulo 2 addition of all the data bits plus the parity bit should be equal to '1' if odd parity is expected and '0', if even parity is expected.

After the stop bit is detected, the receiver immediately looks for the next start bit. However, if the byte is received without a stop bit, this is a framing error and the framing error status bit will be set. In this case, the receiver will immediately begin looking for a transition to logic '0' signifying the start of a new byte.

7.1 Reading the UART module

When the CPU reads from the UART it is similar to the write operation, except that the read input is asserted. The UART will only drive the data lines when enable and read is asserted logic one. The protocol for reading to the UART is shown below.

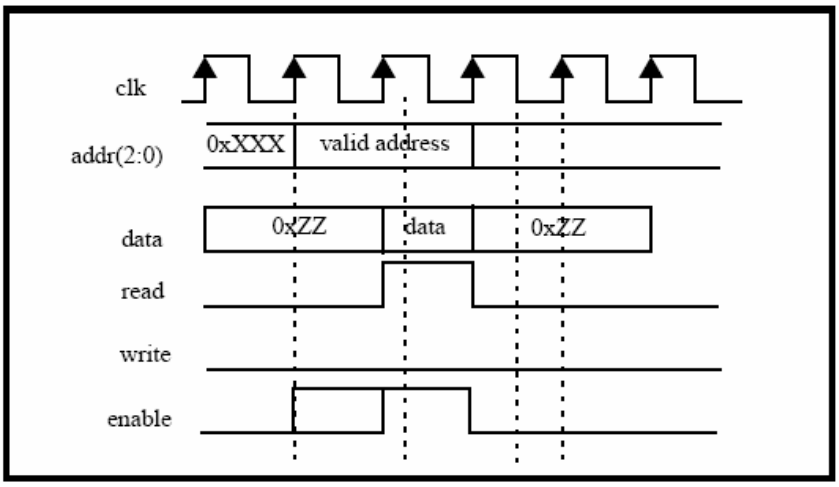


FIGURE 5. Reading from the UART module

If a read is attempted when the receive FIFO is empty, then the last data value received will be returned, and the Receive FIFO Empty bit will remain set.

8.0 Transmit and Receive Protocols

As soon as data appears in the transmit FIFO, the data is sent. Data is transmitted until the transmit FIFO is empty. Incoming data is available to CPU as soon as the receive FIFO is written with incoming data. Reception continues until the receive FIFO is full, then any new incoming data is ignored.

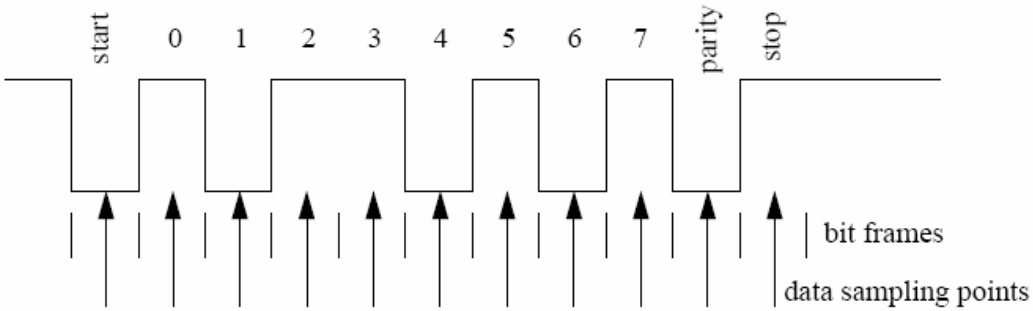


FIGURE 6. Serial Data Protocol for eight data bits, odd parity and one stop bit (8-O-1) when receiving 10101101_2

Eight possible serial protocols may be selected. The protocol 8-O-1 is shown in figure 6. This means eight data bits, odd parity and one stop bit. There is a single active low start bit, followed by 8 data bits (LSB first), a parity bit, and then one active high stop bit. If a protocol with no parity is selected, there will be no parity bit frame. If one stop bit is specified, then there will only be one stop bit frame. The value of the parity bit is determined by the transmitter to cause a odd number of bits (data plus parity) to be received.

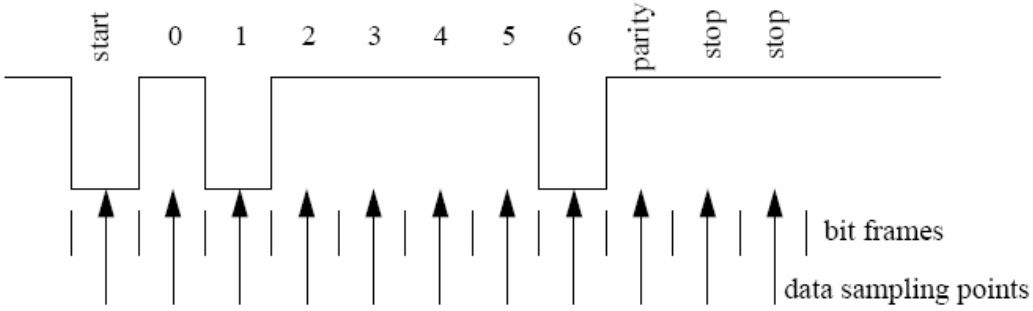


FIGURE 7. Serial Data Protocol for seven data bits, even parity and two stop bits (7-E-2) when receiving 0111101_2

In Figure 7, we see the parity bit has been set to a '1'. Since the seven data bits had five '1's a parity bit of '1' was necessary to establish even parity (even number of ones).

As previously mentioned, if no parity is to be sent, the parity bit frame is absent as shown in Figure 8 below where a 8-N-1 protocol is shown.

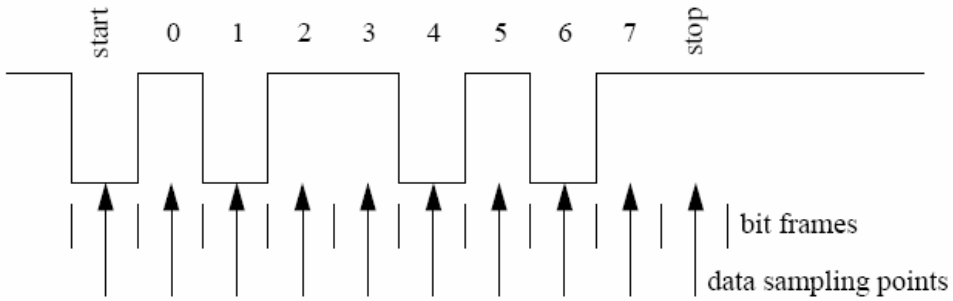


FIGURE 8. Serial Data Protocol for eight data bits, no parity and one stop bit (8-N-1) when receiving 10101101_2