KLE Technological University

Creating Value
Leveraging Knowledge

School of
Electronics and Communication Engineering

Report on UART

By:

1. **Manjunath Inamati** **USN: 01FE21BEC356**
2. **Gautami Naragund** **USN: 01FE21BEC177**
3. **Chetan Paranatti** **USN: 01FE21BEC163**

Semester: VI, 2023-2024

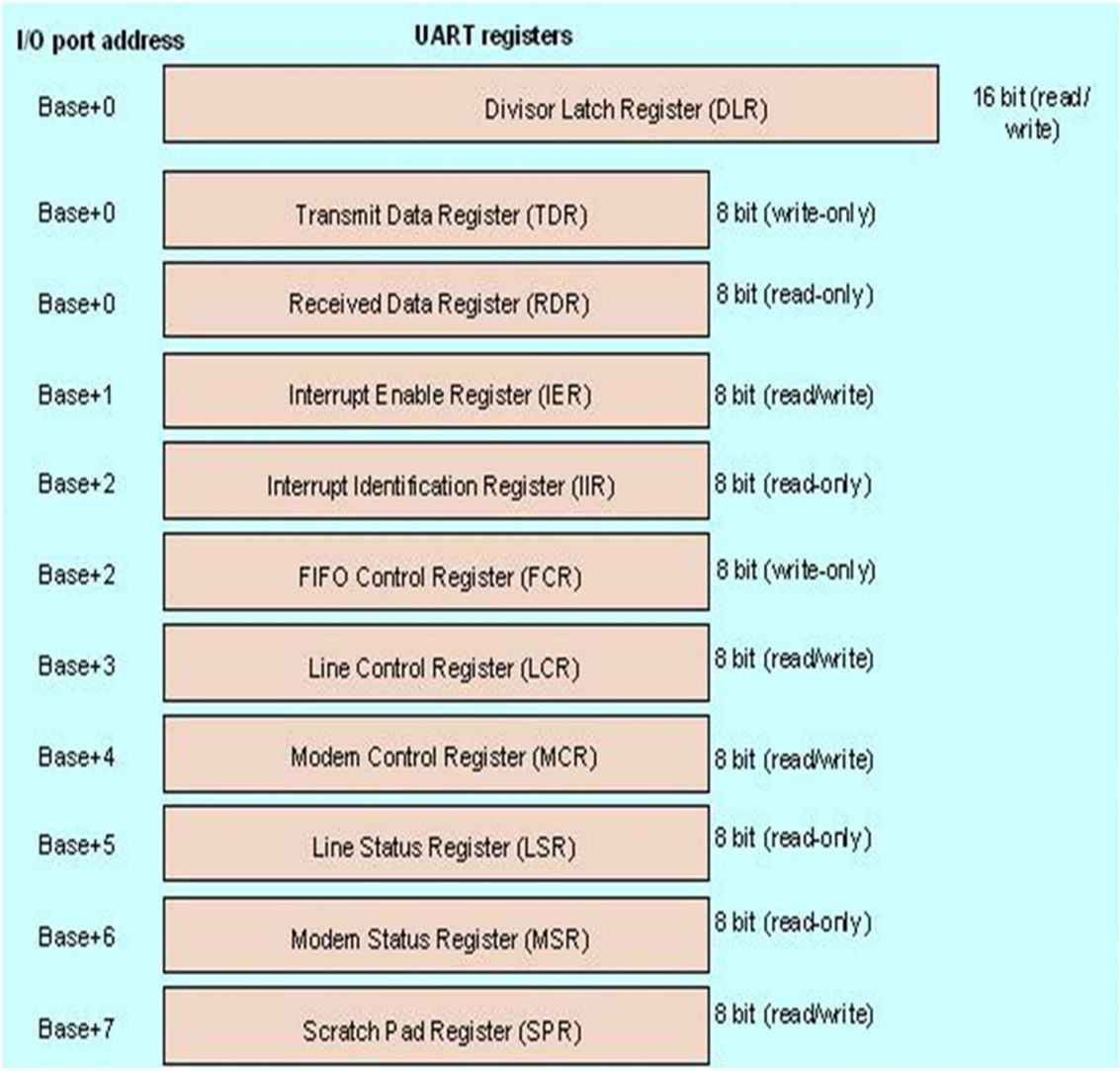
Under the Guidance of

Saroja V Siddamal

DATA Frame:

Start Bit (1 bit)	Data Frame (5 to 9 Data bits)	Parity bits (0 to 1 bit)	Stop Bits (1 to 2 bits)

UART Registers:



1. Divisor Latch Register (DLR):

Name	*Access*	*Description*
Divisor latch byte 1 (LSB)	RW	The LSB of the divisor latch.
Divisor latch byte 2 (MSB)	RW	The MSB of the divisor latch.

The divisor latches can be accessed by setting the 7th bit of LCR to '1'. You should restore this bit to '0' after setting the divisor latches in order to restore access to the other registers that occupy the same addresses. The 2 bytes form one 16-bit register, which is internally accessed as a single number. You should therefore set all 2 bytes of the register to ensure normal operation. The register is set to the default value of 0 on reset, which disables all serial I/O operations in order to ensure explicit setup of the register in the software. The value set should be equal to (system clock speed) / (16 x desired baud rate). The internal counter starts to work when the LSB of DL is written, so when setting the divisor, write the MSB first and the LSB last.

$$\text{Divisor} = \frac{\text{UART Input Clock Frequency}}{\text{Desired Baud Rate} \times 16}$$

Two 8-bit register fields (DLH and DLL), called divisor latches, hold this 16-bit divisor. DLH holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. These divisor latches must be loaded during initialization of the UART to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

Note: To access DLR, the 7th bit of LCR should be '1' for some amount of time, after which you should restore the bit to '0'.

2. Transmit Data Register (TDR):

The UART transmitter section consists of a transmit data register (TDR) and a transmit shift register (TSR). Transmitter control is a function of the line control register (LCR).
The TDR receives data from the internal data bus. When the TSR is idle the UART then moves the data from the TDR to the TSR. The UART serializes the data in the TSR and transmits the data on the TX pin.

Bit	*Description*
0-7	Data to transmit

3. Received Data Register (RDR):

The UART receiver section consists of a receiver shift register (RSR) and a received data register (RDR). The RSR receives serial data from the RX pin. Then the RSR concatenates the data and moves it into the RDR.

Bit	*Description*
0-7	Received data

4. Interrupt Enable Register (IER):

This register allows enabling and disabling interrupt generation by the UART.

Bit	*Access*	*Description*
0	RW	Received Data available interrupt. '0'-disabled '1'-enabled
1	RW	Transmit Data Register empty interrupt. '0'-disabled '1'-enabled
2	RW	Receiver Line Status Interrupt. '0'-disabled '1'-enabled
3	RW	Modem Status Interrupt '0'-disabled '1'-enabled
7-4	RW	Reserved. Should be logic '0'.

5. Interrupt Identification Register (IIR):

The IIR enables the programmer to retrieve what is the current highest priority pending interrupt. Bit 0 indicates that an interrupt is pending when it's logic '0'. When it's '1' no interrupt is pending. The following table displays the list of possible interrupts along with the bits they enable, their priority, and their source and reset control.

Bit	*Access*	*Description*
0	R	Interrupt Pending. '0'- Interuppt is pending. '1'- Interuppt is not pending.
1	R	Receiver Line Status. Overrun error, parity error and framing error. '0'- No any errors. '1'- Either overrun or parity or framing error has occurred.
2	R	Receiver Data Available. **In non-FIFO mode:** '0' = Receiver data register (RDR) is empty. '1' = Receiver data register (RDR) is not empty. **In FIFO mode:** '0' = Receiver FIFO is empty. '1' = Receiver FIFO is not empty.
3	R	Transmit Data Register (TDR) Empty **In non-FIFO mode:** '0' = Transmit data register (TDR) is not empty. '1' = Transmit data register (TDR) is empty. **In FIFO mode:** '0' = Transmitter FIFO is not empty. '1' = Transmitter FIFO is empty.
4	R	Modem Status '0'- Modem Status Register (MSR) state has not changed. '1'- Modem Status Register (MSR) state has changed.
7-5	R	Reserved.

Where CTS – Clear to send.
DSR – Data set ready.
RI – Ring Indicator.
DCD – Data carrier detect.

Bits 5: Logic '0'.
Bits 6 and 7: Logic '1' for compatibility reason.

6. FIFO Control Register (FCR):

The FCR allows the selection of UART operation mode (FIFO / Non FIFO). In addition, the FIFOs can be cleared using this register.

Bit	***Access***	***Description***
0	W	Transmitter and receiver FIFOs mode enable. '0' = Non-FIFO mode. The transmitter and receiver FIFOs are disabled, and the FIFO pointers are cleared. '1' = FIFO mode. The transmitter and receiver FIFOs are enabled.
1	W	Writing a '1' to bit 1 clears the Receiver FIFO and resets its logic. But it doesn't clear the shift register, i.e. receiving of the current character continues.
2	W	Writing a '1' to bit 2 clears the Transmitter FIFO and resets its logic. The shift register is not cleared, i.e. transmitting of the current character continues.
5-3	W	**Ignored**
7-6	W	Reserved

7. Line Control Register (LCR):

The line control register allows the specification of the format of the asynchronous data communication used. A bit in the register also allows access to the Divisor Latches, which define the baud rate. Reading from the register is allowed to check the current settings of the communication.

Bit	***Access***	***Description***
1-0	RW	Select number of bits in each character '00'- 5bits '01'- 6bits '10'- 7bits '11'- 8bits
2	RW	Specify the number of generated stop bits '0'- 1 stop bit '1'- 1.5 stop bits when 5-bit character length selected and 2 bits otherwise Note that the receiver always checks the first stop bit only.
3	RW	Parity Enable '0'-No parity '1'-Parity bit is generated on each outgoing character and is checked on each incoming one.

4	RW	Even Parity select '0'- Odd number of '1' is transmitted and checked in each word (data and parity combined). In other words, if the data has an even number of '1' in it, then the parity bit is '1'. '1'- Even number of '1' is transmitted in each word.
5	RW	Stick Parity bit. **(Ignored)**
6	RW	Ignored
7	RW	Divisor Latch Access bit. '1'- The divisor latches can be accessed '0'- The normal registers are accessed

8. Modem Control Register (MCR):

The modem control register allows transferring control signals to a modem connected to the UART.

Bit	***Access***	***Description***
0	W	Data Terminal Ready (DTR) signal control '0'- DTR is '1' '1'- DTR is '0'
1	W	Request To Send (RTS) signal control '0'- RTS is '1' '1'- RTS is '0'
2	W	Out1. In loopback mode, connected Ring Indicator (RI) signal input
3	W	Out2. In loopback mode, connected to Data Carrier Detect (DCD) input.
4	W	Loopback mode '0'- normal operation '1'- loopback mode. The signal of the transmitter shift register is internally connected to the input of the receiver shift register. The following connections are made: DTR->DSR RTS-> CTS Out1-> RI Out2-> DCD
7-5	W	Ignored

9. Line Status Register (LSR):

Bit	***Access***	***Description***
0	R	Data-ready (DR) indicator for the receiver. **In non-FIFO mode:** '0' = Data is not ready, or the DR bit was cleared because the character was read from the received data register (RDR). '1' = Data is ready. A complete incoming character has been received and transferred into the received data register (RDR). **In FIFO mode:** '0' = Data is not ready, or the DR bit was cleared because all of the characters in the receiver FIFO have been read. '1' = Data is ready. There is at least one unread character in the receiver FIFO. If the FIFO is empty, the DR bit is set as soon as a complete

		incoming character has been received and transferred into the FIFO. The DR bit remains set until the FIFO is empty again.
1	R	Overrun Error (OE) indicator. **In non-FIFO mode:** '0' = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). '1' = Overrun error has been detected. Before the character in the received data register (RDR) could be read, it was overwritten by the next character arriving in RDR. **In FIFO mode:** '0' = No overrun error has been detected, or the OE bit was cleared because the CPU read the content of the line status register (LSR). '1' = Overrun error has been detected. If 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. An overrun error is indicated to the CPU as soon as it happens. The new character overwrites the character in the shift register, but it is not transferred to the FIFO.
2	R	Parity Error (PE) indicator. **In non-FIFO mode:** '0' = No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the received data register (RDR). '1' = A parity error has been detected with the character in the received data register (RDR). **In FIFO mode:** '0' = No parity error has been detected, or the PE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no parity error. '1' = A parity error has been detected with the character at the top of the receiver FIFO.
3	R	Framing Error (FE) indicator. **In non-FIFO mode:** '0' = No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the received data register (RDR). '1' = A framing error has been detected with the character in the received data register (RDR). **In FIFO mode:** '0' = No framing error has been detected, or the FE bit was cleared because the CPU read the erroneous data from the receiver FIFO and the next character to be read from the FIFO has no framing error. '1' = A framing error has been detected with the character at the top of the receiver FIFO.
4	R	**Ignored**
5	R	Transmit Data Register (TDR) Empty **In non-FIFO mode:** '0' = Transmit data register (TDR) is not empty. TDR has been loaded by the CPU.

		'1' = Transmit data register (TDR) is empty (ready to accept a new character). The content of TDR has been transferred to the transmitter shift register (TSR). **In FIFO mode:** '0' = Transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. The transmitter FIFO may be written to if it is not full. '1' = Transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (TSR).
6	R	Transmitter Shift Register Empty indicator. '0' = Transmitter Shift Register (TSR) contains a data character. '1' = Transmitter Shift register (TSR) is empty.
7	R	**Ignored**

10. Modem Status Register (MSR):

The register displays the current state of the modem control lines. Also, four bits also provide an indication in the state of one of the modem status lines. These bits are set to '1' when a change in the corresponding line has been detected and they are reset when the register is being read.

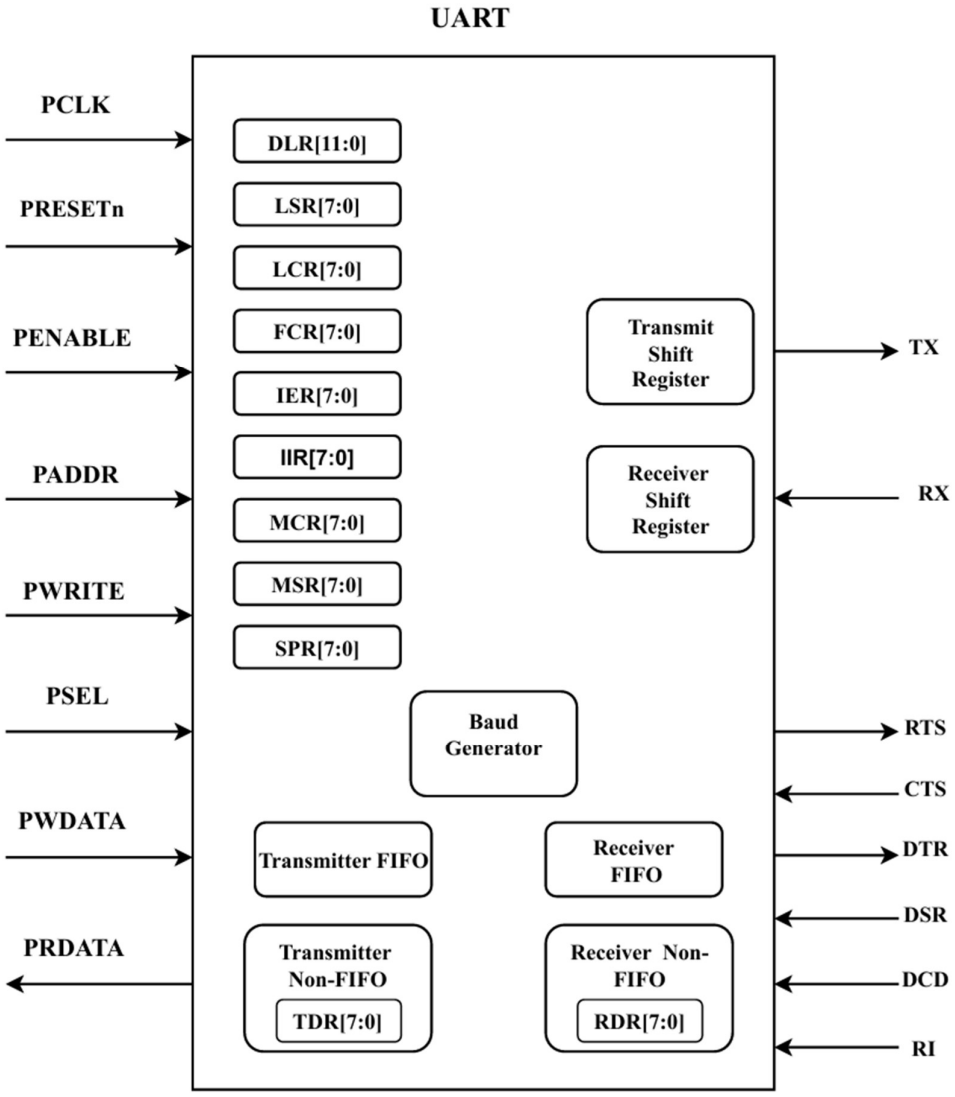
Bit	***Access***	***Description***
0	R	Delta Clear To Send (DCTS) indicator '1'- The CTS line has changed its state.
1	R	Delta Data Set Ready (DDSR) indicator '1'- The DSR line has changed its state.
2	R	Trailing Edge of Ring Indicator (TERI) detector. The RI line has changed its state from low to high state.
3	R	Delta Data Carrier Detect (DDCD) indicator '1'- The DCD line has changed its state.
4	R	Complement of the CTS input or equals to RTS in loopback mode.
5	R	Complement of the DSR input or equals to DTR in loopback mode.
6	R	Complement of the RI input or equals to Out1 in loopback mode.
7	R	Complement of the DCD input or equals to Out2 in loopback mode.

11. Scratch Pad Register (SPR):

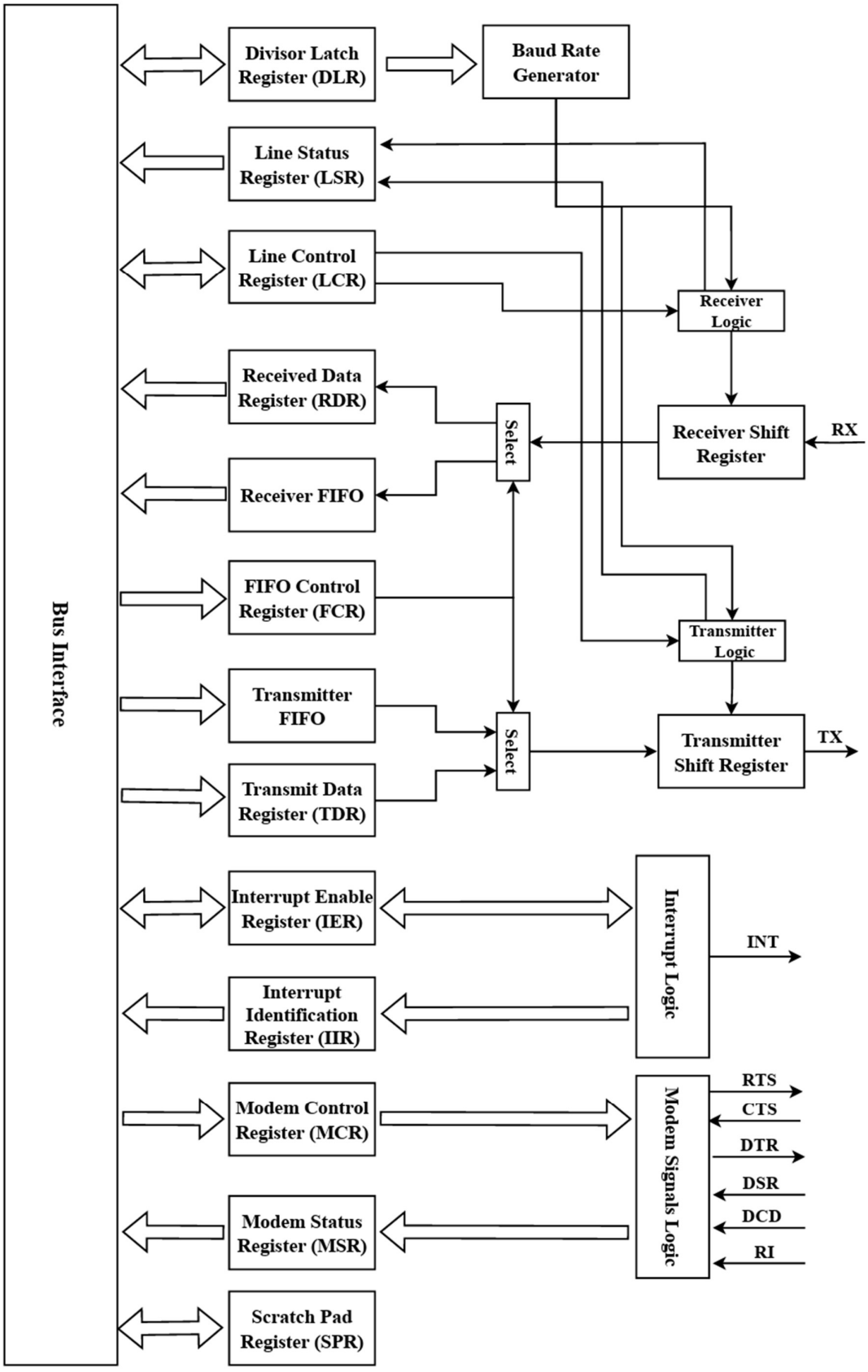
The scratch pad register (SCR) is intended for the programmer's use as a scratch pad. It temporarily holds the programmer's data without affecting UART operation.

Bits	***Access***	***Description***
0-7	RW	These bits are intended for the programmer's use as a scratch pad in the sense that it temporarily holds the programmer's data without affecting any other UART operation.

Block Diagram:



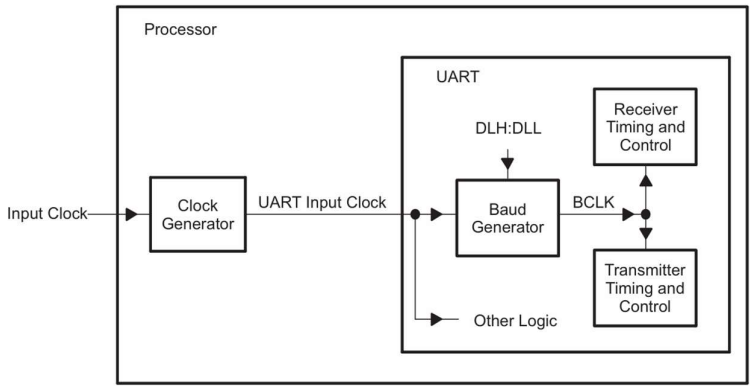
Architecture:



Clock Generation and Control:

The UART bit clock is derived from an input clock to the UART.

The processor clock generator receives a signal from an external clock source and produces a UART input clock with a programmed frequency. The UART contains a programmable baud generator that takes an input clock and divides it by a divisor in the range between 1 and (216 - 1) to produce a baud clock (BCLK). The frequency of BCLK is sixteen times (16×) the baud rate (each received or transmitted bit lasts 16 BCLK cycles). When the UART is receiving, the bit is sampled in the 8th BCLK cycle for 16× over sampling mode.

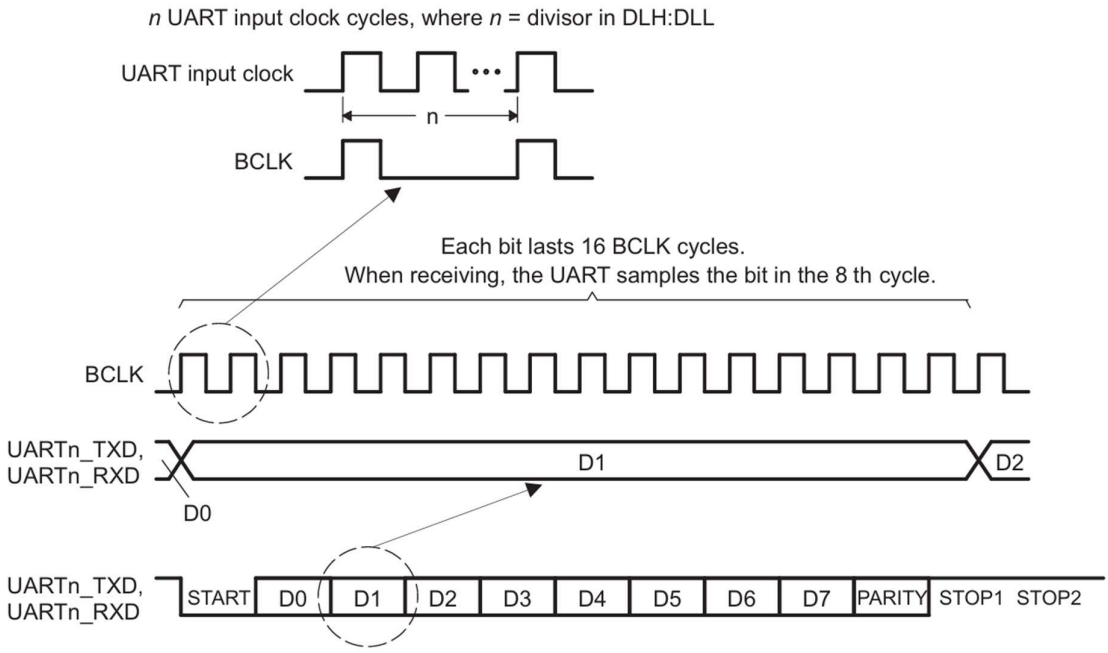


$$\text{Divisor} = \frac{\text{UART Input Clock Frequency}}{\text{Desired Baud Rate} \times 16}$$

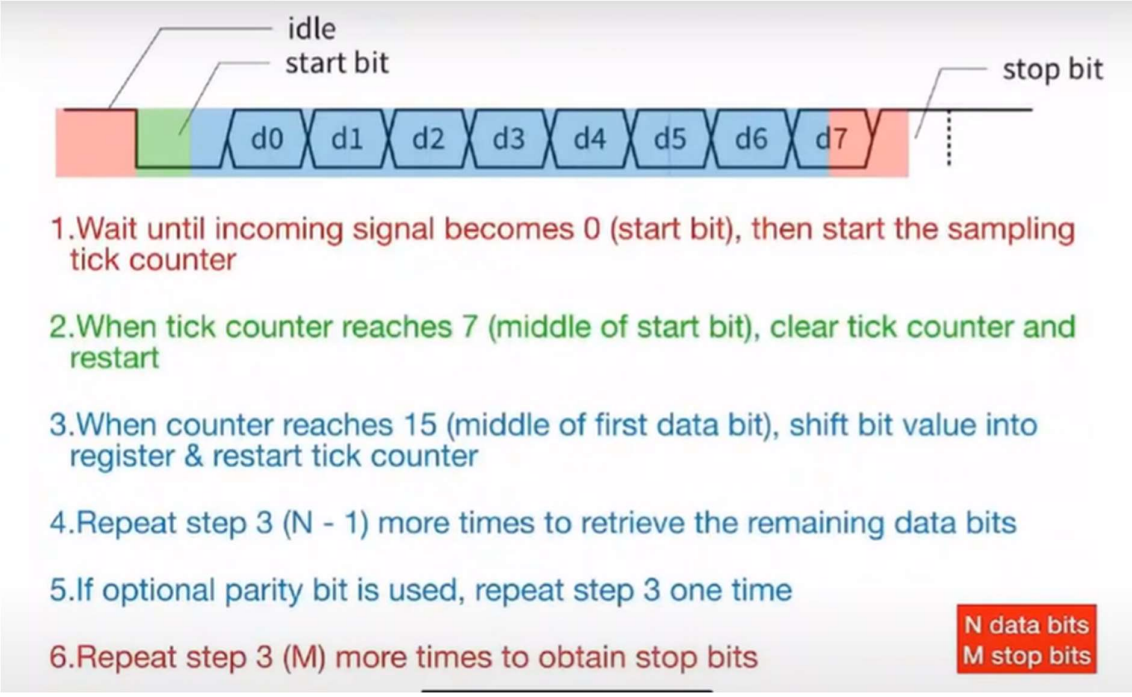
Baud Rate	Divisor value	Actual Baud rate	error
9600	156	9615.3846	15.3846
19200	78	19230.7692	30.7692
38400	39	38461.53846	61.5846
57600	26	57692.30769	92.30769
115200	13.	115384.6154	184.6153

Two 8-bit register fields (DLH and DLL), called divisor latches, hold this 16-bit divisor. DLH holds the most significant bits of the divisor, and DLL holds the least significant bits of the divisor. These divisor latches must be loaded during initialization of the UART to ensure desired operation of the baud generator. Writing to the divisor latches results in two wait states being inserted during the write access while the baud generator is loaded with the new value.

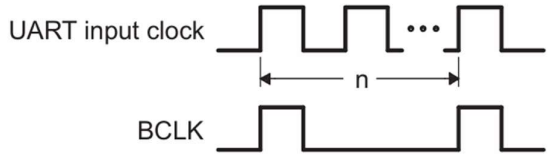
The below figure summarizes the relationship between the transferred data bit, BCLK, and the UART input clock. Note that the timing relationship in Figure shows that each bit lasts for 16 BCLK cycles. This is in case of 16× oversampling mode.


n UART input clock cycles, where n = divisor in DLH:DLL
UART input clock
n
BCLK
Each bit lasts 16 BCLK cycles.
When receiving, the UART samples the bit in the 8 th cycle.
BCLK
UARTn_TXD,
UARTn_RXD
D1
D2
D0
UARTn_TXD,
UARTn_RXD
START
D0
D1
D2
D3
D4
D5
D6
D7
PARITY
STOP1
STOP2

Oversampling:



Baud Generator:



Here

UART Input clock frequency (Fclk) = 24 MHz
Input clock time (Tclk) = 1/Fclk = 41.5 n sec
N = Divisor

$$\text{Divisor} = \frac{\text{UART Input Clock Frequency}}{\text{Desired Baud Rate} \times 16}$$

Baud Rate	Divisor value	Actual Baud rate	error
9600	156	9615.3846	15.3846
19200	78	19230.7692	30.7692
38400	39	38461.53846	61.5846
57600	26	57692.30769	92.30769
115200	13.	115384.6154	184.6153

For oversampling concept Tb should be sampled 16 times, so we need 16 clock ticks from Baud Generator in each Tb.

Let baud rate = 9600 bpsec (9600 bits transmitted per sec).
Time taken to transmit one bit (Tb) = 1/9600 = 104166.6 n sec.

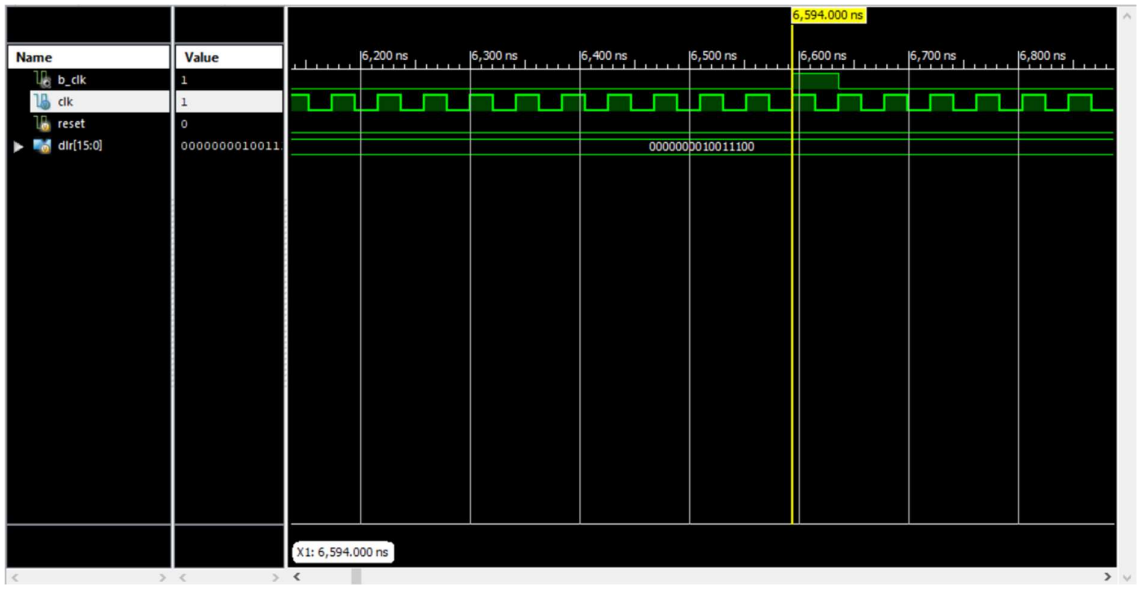
So Bclk = Tb/16 = 6510.4 n sec

Fclk = 24 MHz
Tclk = 41.5 n sec
N = 156

When Tclk reaches count 156 we get one Bclk from baud generator.
Bclk = 156 * 40.5
= 6318 n sec ≈ 6500 n sec
So we get 16 samples from the Divisor values for their respective baud rate in the above fig.

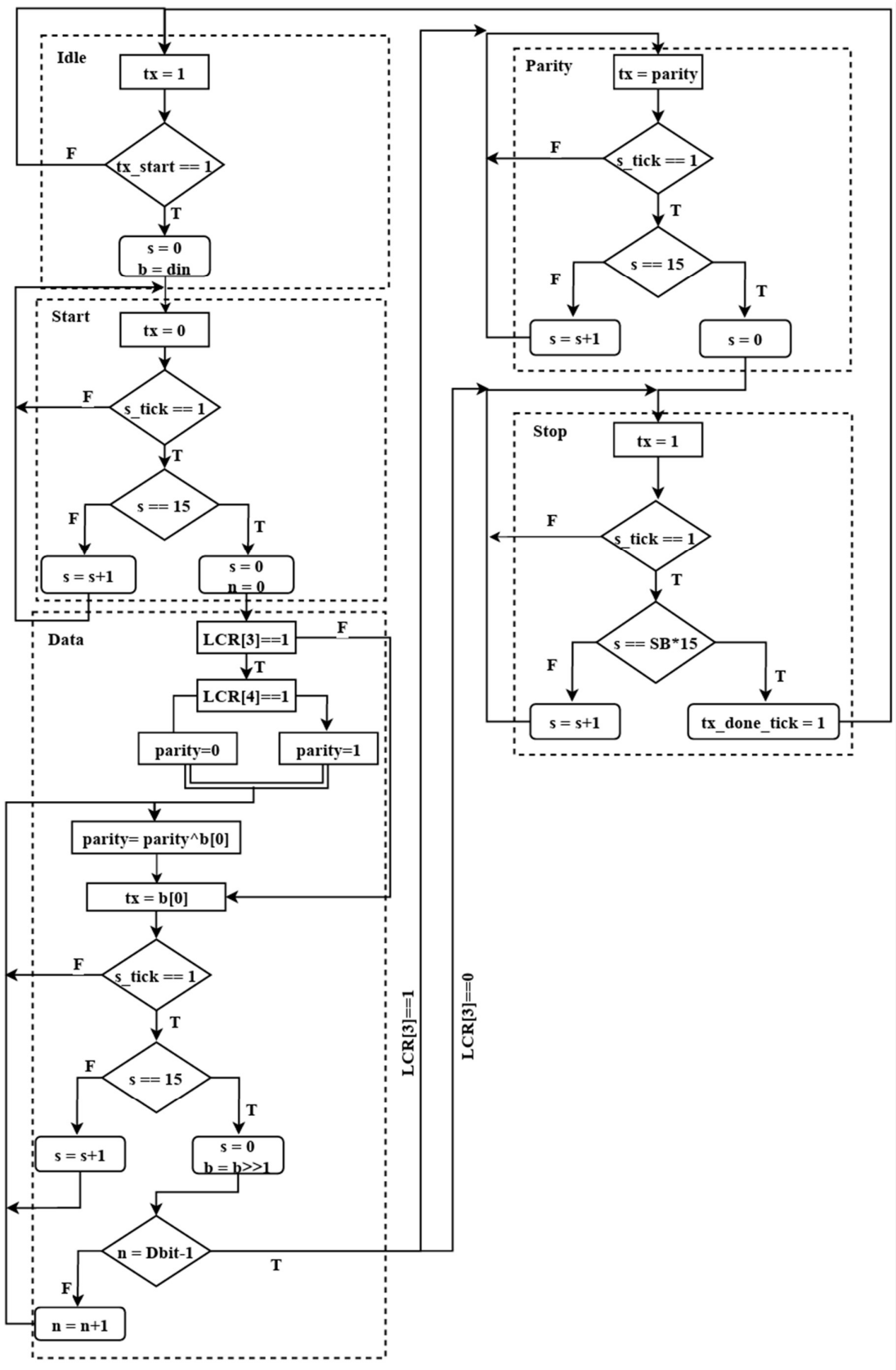
While transmitting or receiving each nit is sampled at half of bit period i.e 8^{th} Bclk.

6594 n sec for 1 Bclk

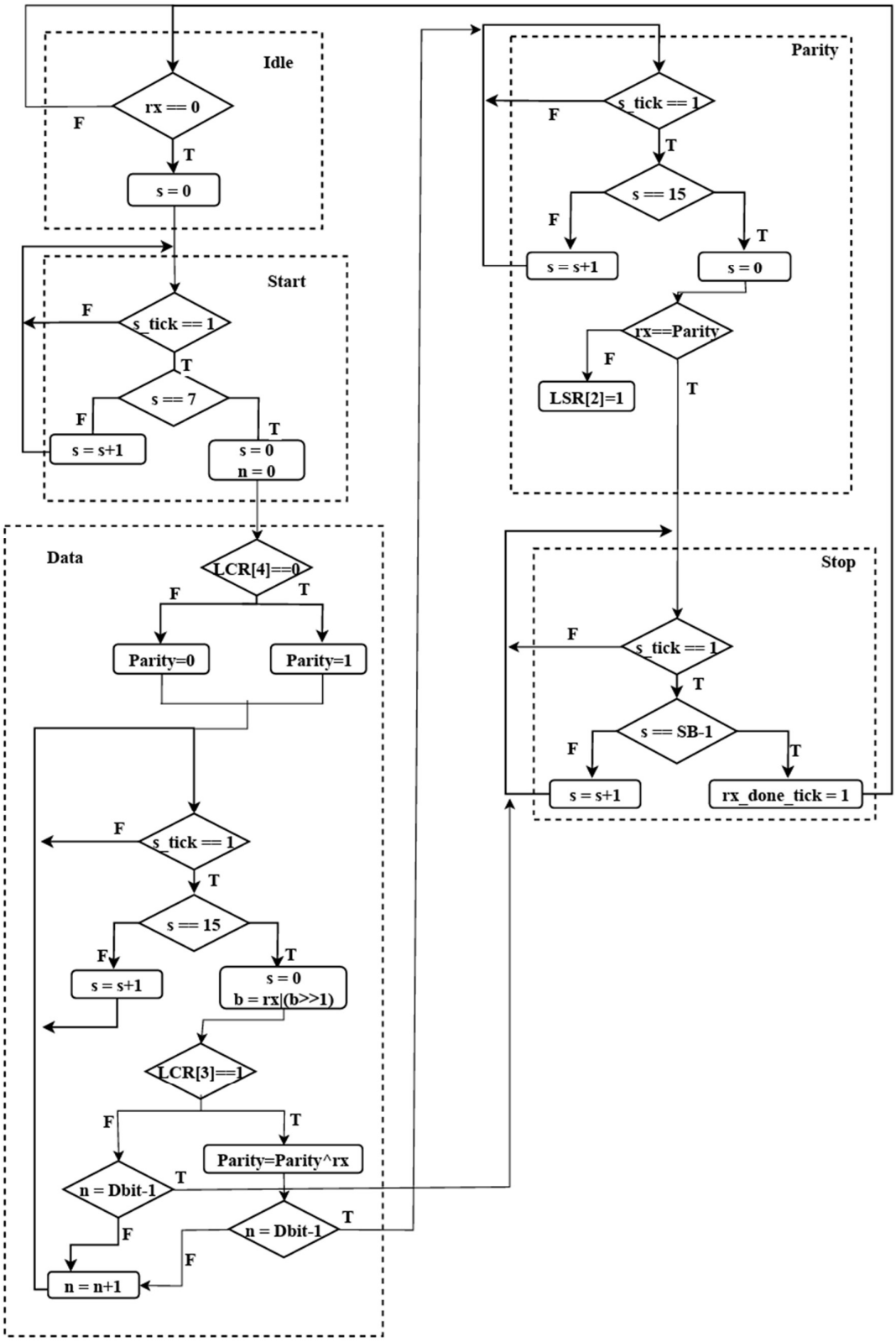


16 Bclk in 1 Tb

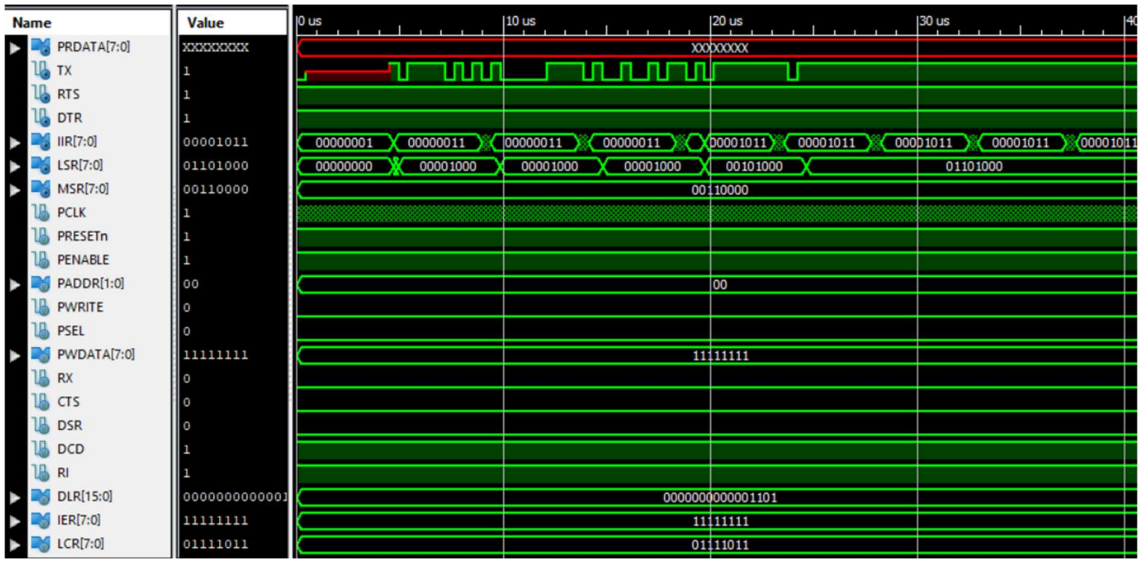
Transmitter FSM:



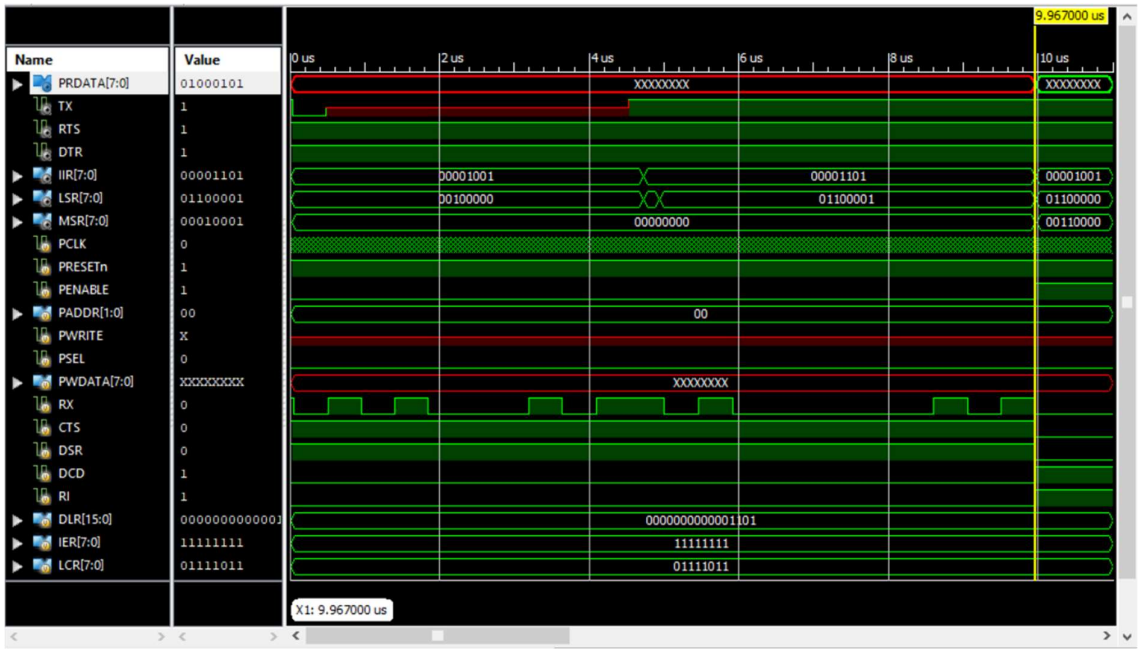
Receiver FSM:



Transmitter Results:



Receiver Results:



Loopback mode operation:

When the 4th bit of MCR is '1', Uart works in loopback mode. The RX pin receives the serial data from TX pin directly and then gets stored in RSR, which is then further read from PRDATA.