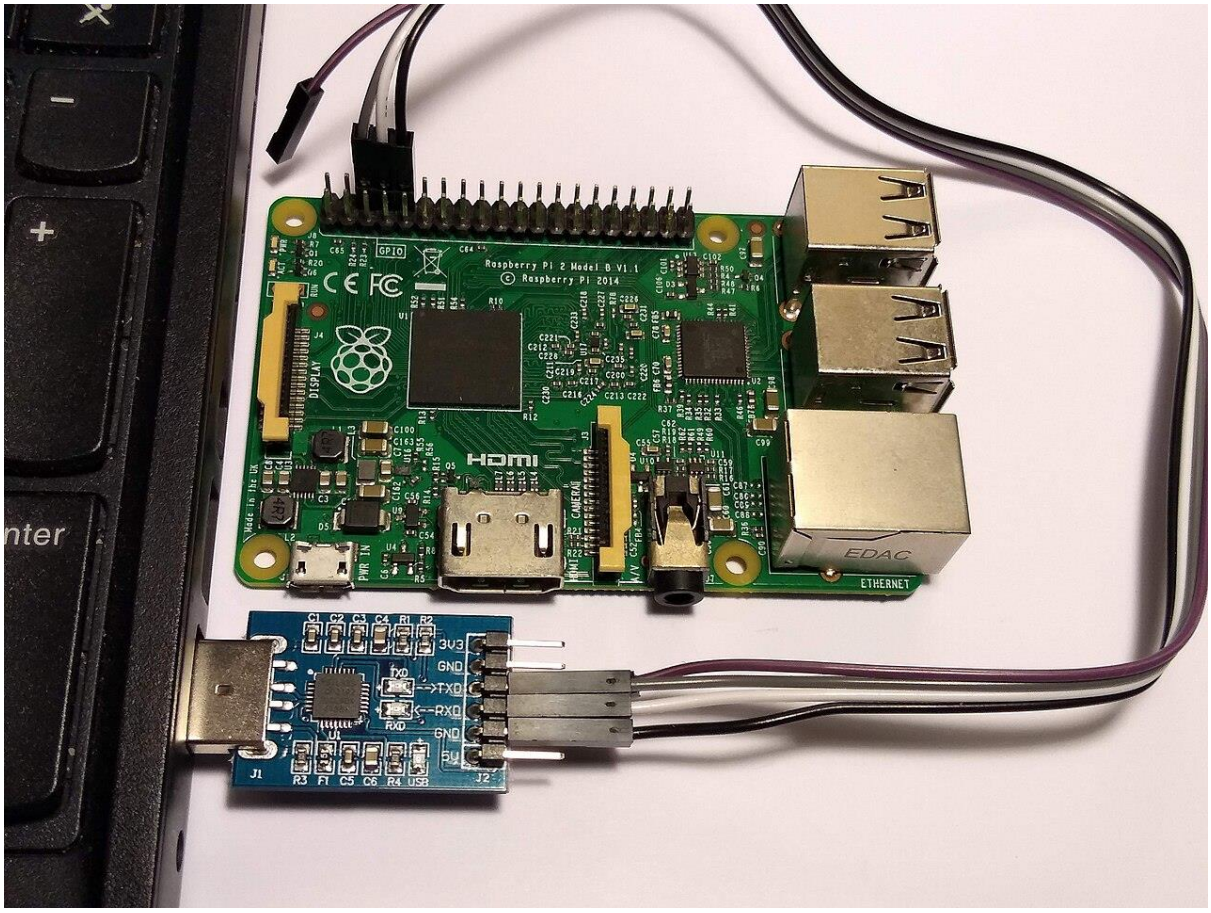


UART RX – TX Using Verilog HDL

Brief Description of all Modules

1. Baud Rate Generator (baud_gen) – Brief Description

The Baud Rate Generator produces a timing tick used to sample or transmit UART bits at the correct baud rate. It uses a programmable divisor (dvsr) to divide the main clock frequency. The internal counter runs from 0 to the divisor value and generates a tick pulse upon completing the count cycle.
This tick is used by both the transmitter and receiver for 16× oversampling, ensuring accurate timing and stable serial communication.

2. UART Receiver (uart_rx) – Brief Description

The UART Receiver module implements a standard asynchronous serial data reception mechanism using a finite state machine (FSM). It supports 8 data bits and uses 16× oversampling for reliable bit detection. The design transitions through four primary states: **IDLE**, **START**, **DATA**, and **STOP**.

- In the **IDLE** state, the module waits for the start bit (logic ‘0’).
- In the **START** state, the incoming line is sampled at the midpoint of the bit period to confirm start-bit validity.
- In the **DATA** state, each bit is sampled at precise oversampled intervals and shifted into an internal register, capturing the full byte.
- In the **STOP** state, the stop bit is verified before asserting a receive-complete flag.

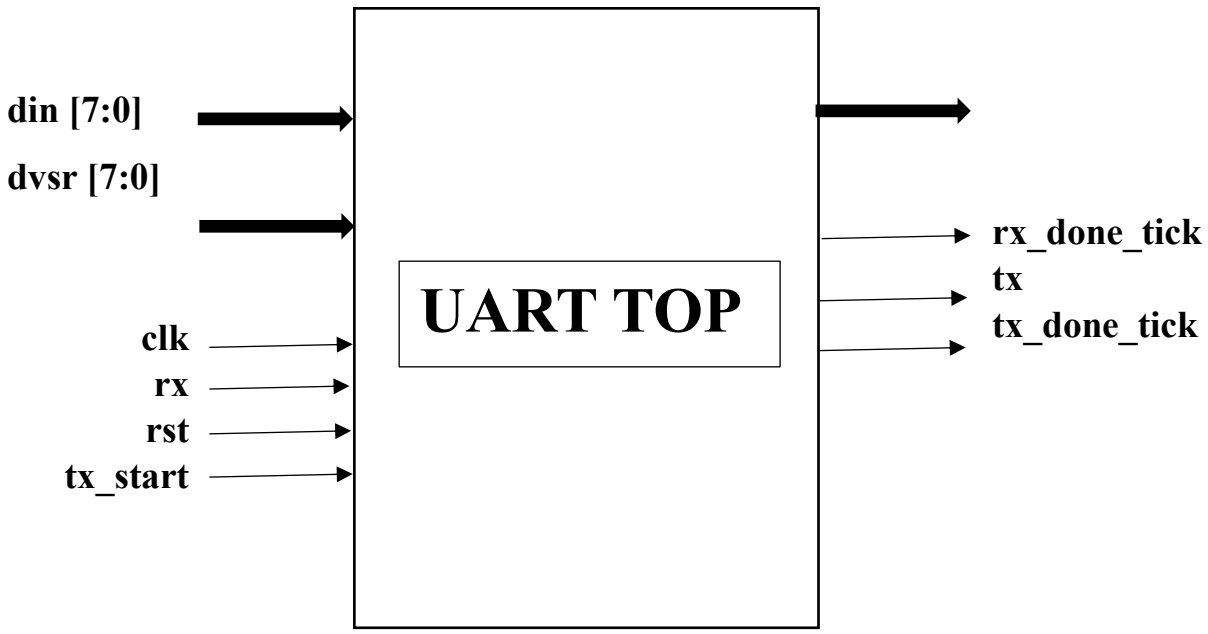
Upon successful reception of 8 bits and validation of the stop bit, the module outputs the received byte and raises the rx_done_tick signal, indicating data availability.

3. UART Transmitter (uart_tx) – Brief Description

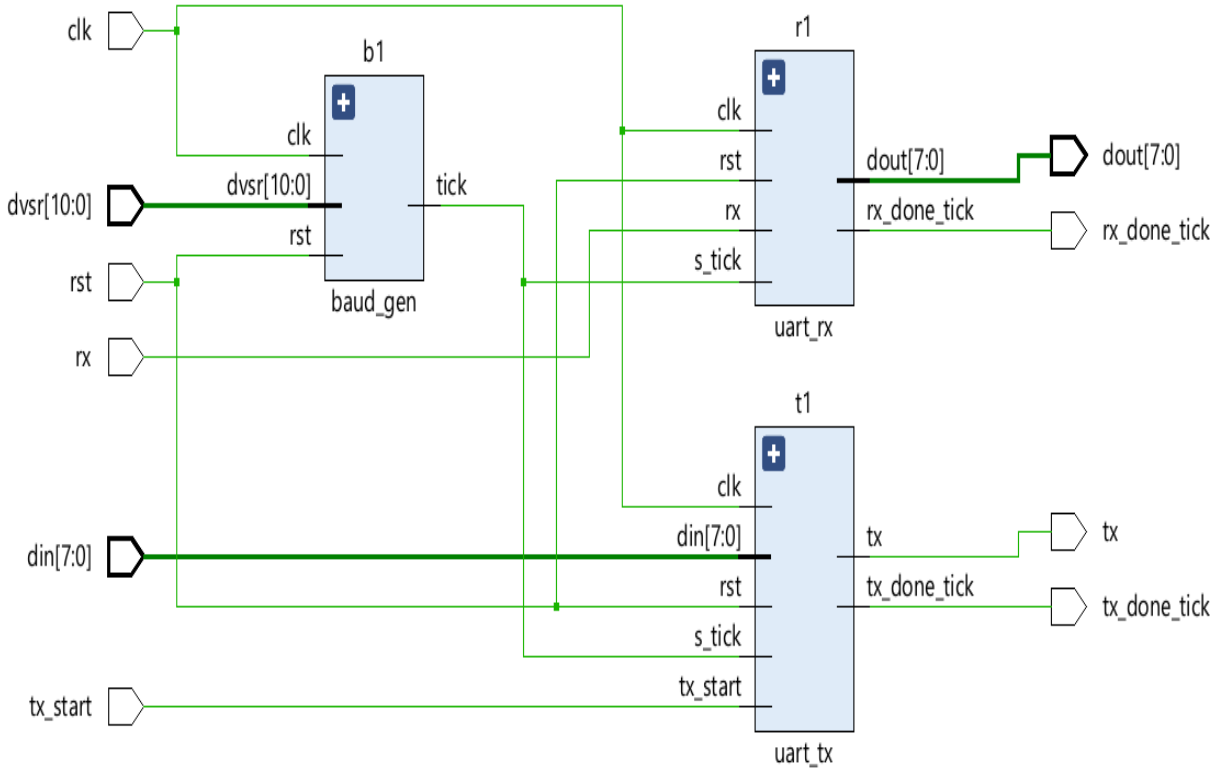
The UART Transmitter module sends an 8-bit data word serially using an FSM-based architecture. The transmitter follows four operational states: **IDLE**, **START**, **DATA**, and **STOP**.

- In the **IDLE** state, the line is held high until a transmission request (tx_start) is received.
- In the **START** state, the module drives the line low for one bit period to indicate the start bit.
- During the **DATA** state, the 8 data bits are shifted out least significant bit first, synchronized to the baud-tick generator.
- In the **STOP** state, the module drives the line high for one bit period before completing the transmission cycle.
  After sending all bits and the stop bit, the module asserts the tx_done_tick signal, indicating the end of a successful transmission.

==========UART TOP DESIGN==========



RTL Schematic

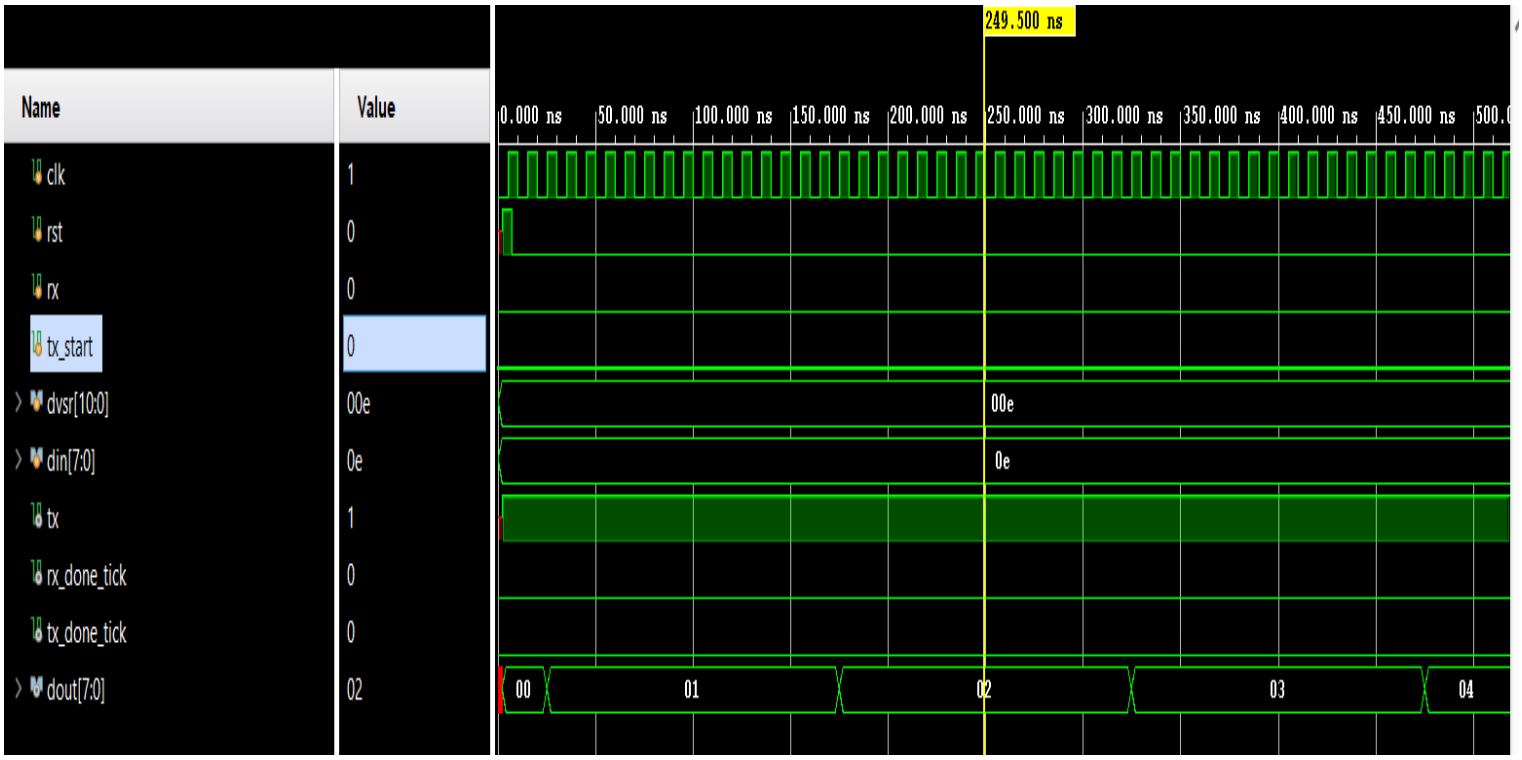


Verilog Snippet (TOP):

```
module uart_top(input clk,
input rst,
input [10:0] dvsr,
input [7:0] din,
input rx,
input tx_start,
output tx,
output [7:0] dout,
output rx_done_tick,
output tx_done_tick


  );
 wire tick;
 baud_gen b1 (clk, rst, dvsr, tick);   // baud generator for receiver as well as transmitter
 uart_rx r1 (clk, rst, rx, tick, rx_done_tick, dout);
 uart_tx t1 (clk, rst, tx_start, tick, din, tx_done_tick, tx);
endmodule
```

Simulation


Name
Value
clk
1
rst
0
rx
0
tx_start
0
dvsr[10:0]
00e
din[7:0]
0e
tx
1
rx_done_tick
0
tx_done_tick
0
dout[7:0]
02
249.500 ns
0.000 ns
50.000 ns
100.000 ns
150.000 ns
200.000 ns
250.000 ns
300.000 ns
350.000 ns
400.000 ns
450.000 ns
00e
0e
00
01
02
03
04