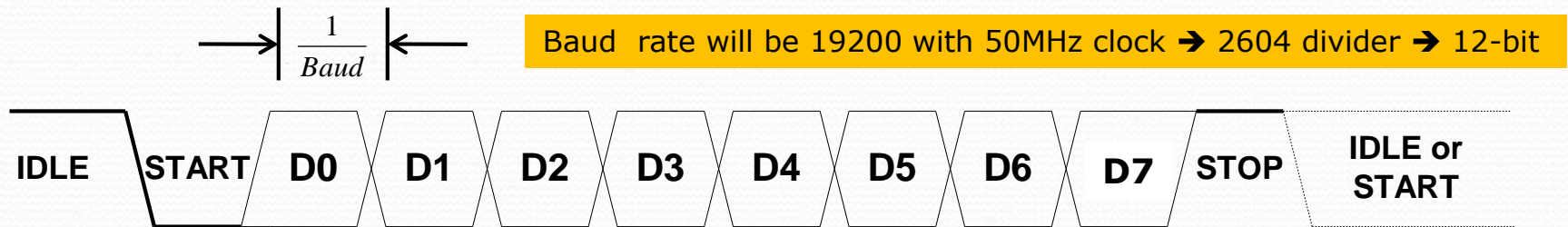


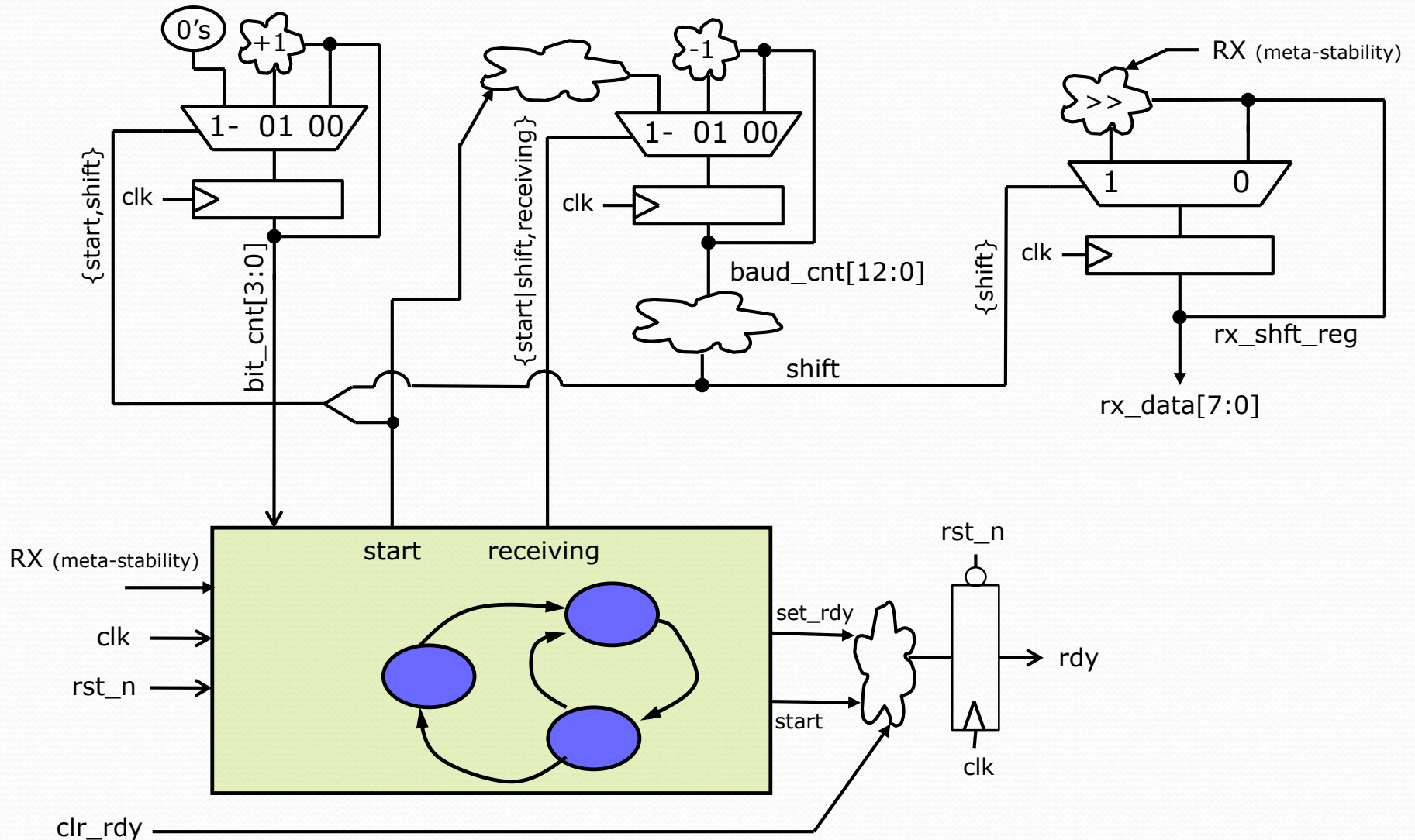
Exercise 12 (UART Reciver)(HW3 Prob4):

- RS-232 signal phases
 - Idle
 - Start bit
 - Data (8-data for our project)
 - Parity (no parity for our project)
 - Stop bit – channel returns to idle condition
 - Idle or Start next frame

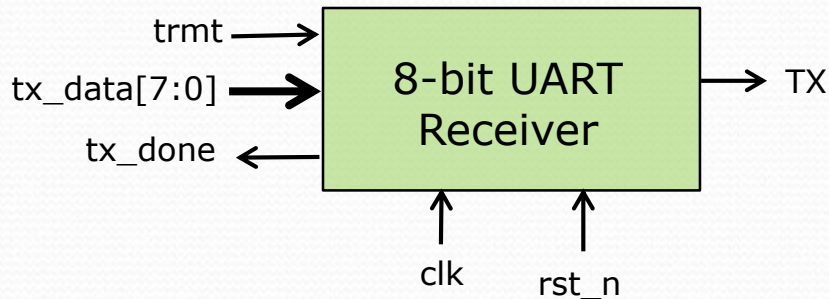


- Receiver monitors for falling edge of Start bit. Counts off $\frac{1}{2}$ a bit time and starts shifting (right shifting since LSB is first) data into a register.
- Will shift a total of 10 times. Start bit will “fall off the end” of the 9-bit shift registers. Bits [7:0] of the shift register are the received data.

Possible Topology of UART_rcv



Exercise 12 (UART Receiver):



Signal:	Dir:	Description
clk,rst_n	in	50MHz system clock & active low reset
RX	in	Serial data input
clr_rdy	In	Knocks down rdy when asserted
rx_data[7:0]	out	Byte received
rdy	out	Asserted when byte received. Stays high till start bit of next byte starts, or until clr_rdy asserted.

- HW3 Problem 4 involves making a UART transmitter (*UART_rcv.sv*). You are to start that design during this exercise.
- Submit what ever you have complete of *UART_rcv.sv* and *UART_tb.sv* to the dropbox for Exercise12.
- See next slide for structure of testbench!

Exercise 12 (UART Receiver):

Implement a the UART Receiver (**UART_rcv.sv**).

Since you have a transmitter too, it is now easy to make a self checking test bench. Architect the test bench as shown. Does the 8-bit value you transmit match the value you receive when the transmission completes?

Also test **tx_done** and **rdy** and **clr_rdy** functionality.

