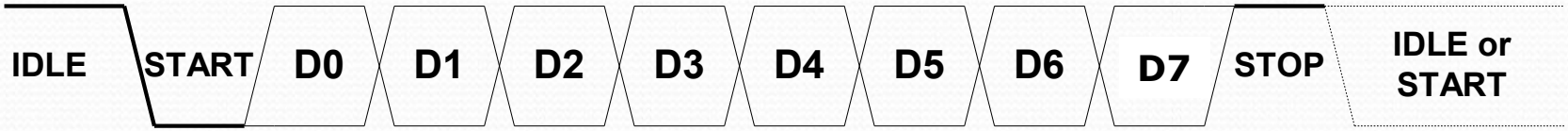
Exercise 13 (UART Transmitter)(HW3 Prob4):

- 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

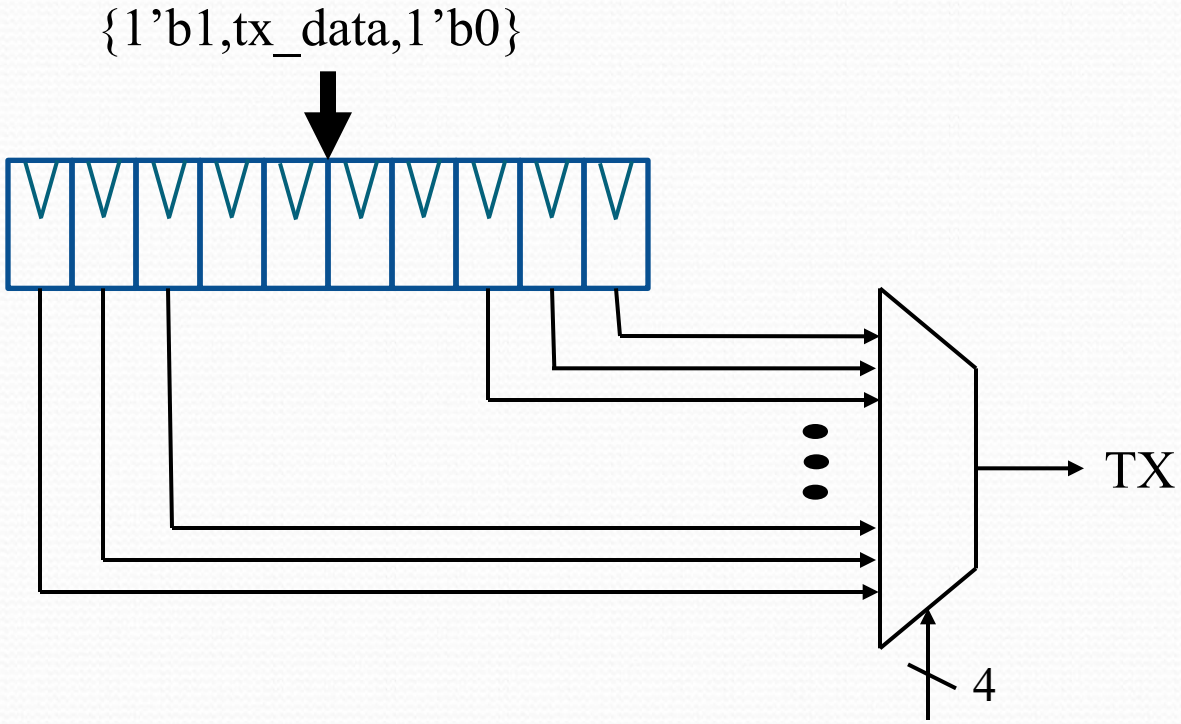
$\frac{1}{Baud}$

Baud rate will be 9600with 50MHz clock ➔ 5208 divider ➔ 13-bit

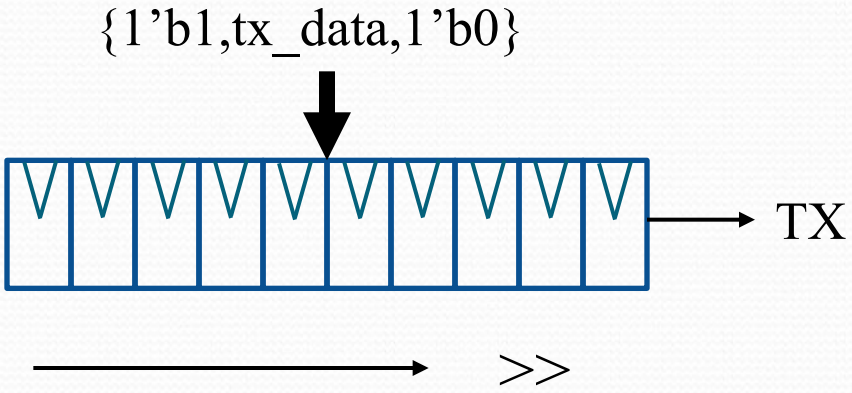


- Transmitter sits idle till told to transmit. Then will shift out a 9-bit (start bit appended) register at the baud rate interval.

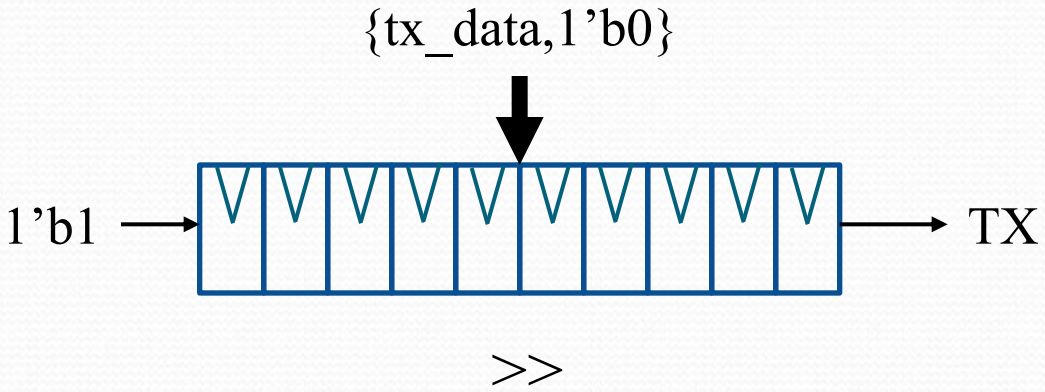
Exercise 13 (UART Transmitter)



Exercise 13 (UART Transmitter)

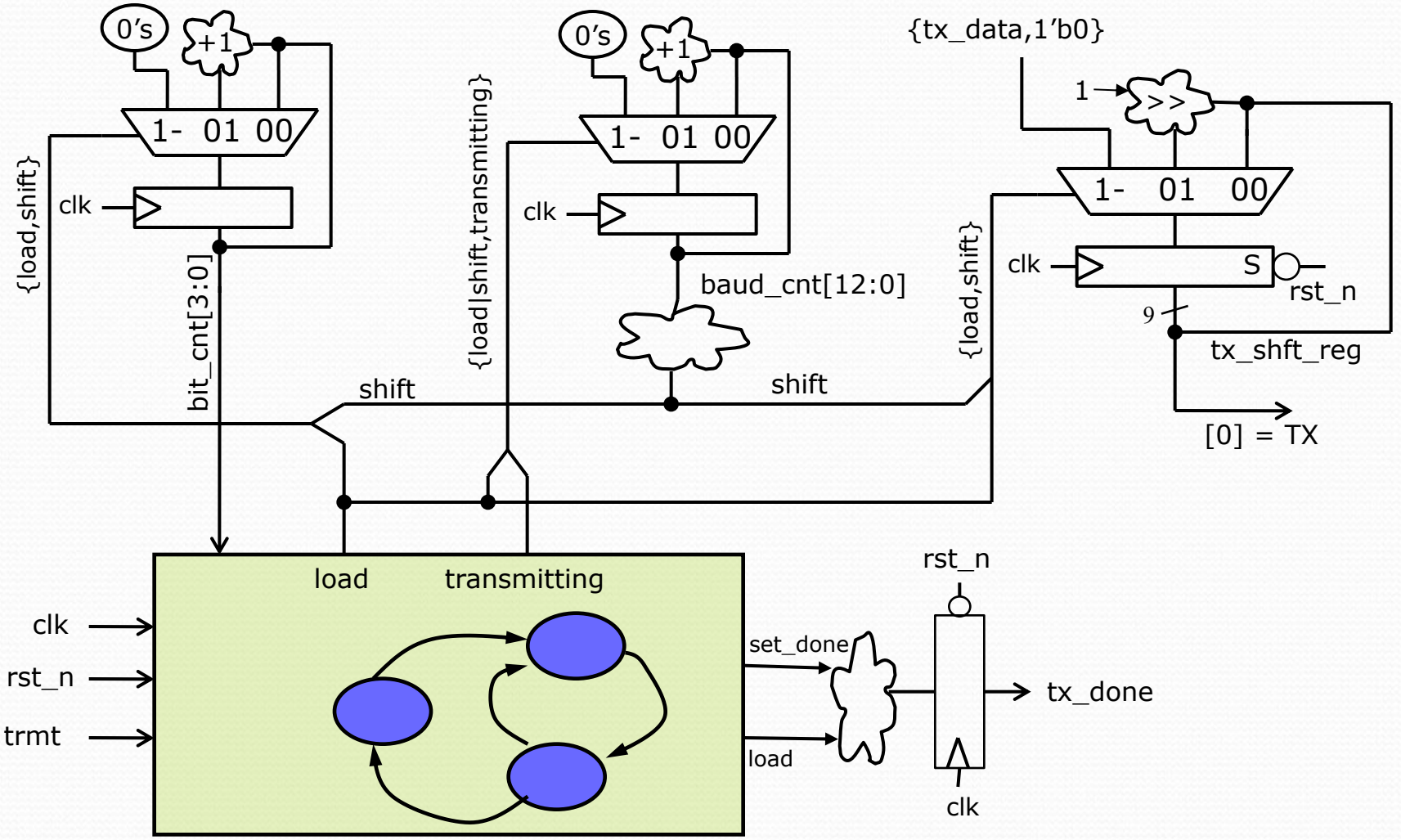


A simple shift reg is heart of the machine



Does not even have to be 10-bits, can get away with 9.

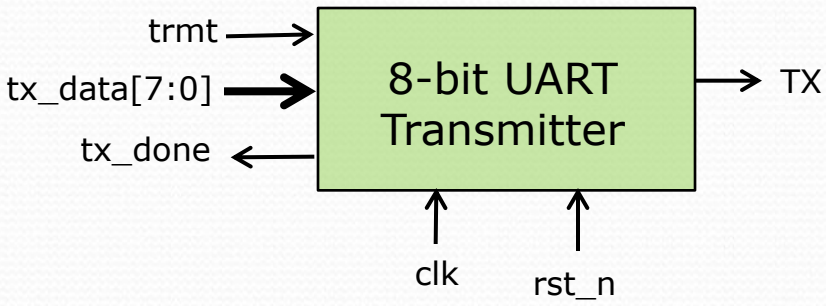
Possible Topology of UART_tx



Warning on use of LLM

- Copilot/LLM usage will likely result in a functional UART transmitter. This is the kind of well defined functional block that it does well.
- However, keep in mind your objective is not to get it done, or even necessarily to learn. Your objective is to produce code in a style that Eric likes, and that will synthesize efficiently. So….guide your LLM to produce that.

Exercise 13 (UART Transmitter):



Signal:	Dir:	Description
clk,rst_n	in	50MHz system clock & active low reset
TX	out	Serial data output
trmt	in	Asserted for 1 clock to initiate transmission
tx_data[7:0]	in	Byte to transmit
tx_done	out	Asserted when byte is done transmitting. Stays high till next byte transmitted.

- HW3 Problem 4 involves making a UART transmitter (***UART_tx.sv***). You are to start that design during this exercise.
- Make a simple test bench for it. Just instantiate your transmitter and send a few bytes. Verify correct functionality (including baud rate) by looking at the waveforms.
- Submit ***UART_tx.sv*** to the dropbox for Exercise13.