

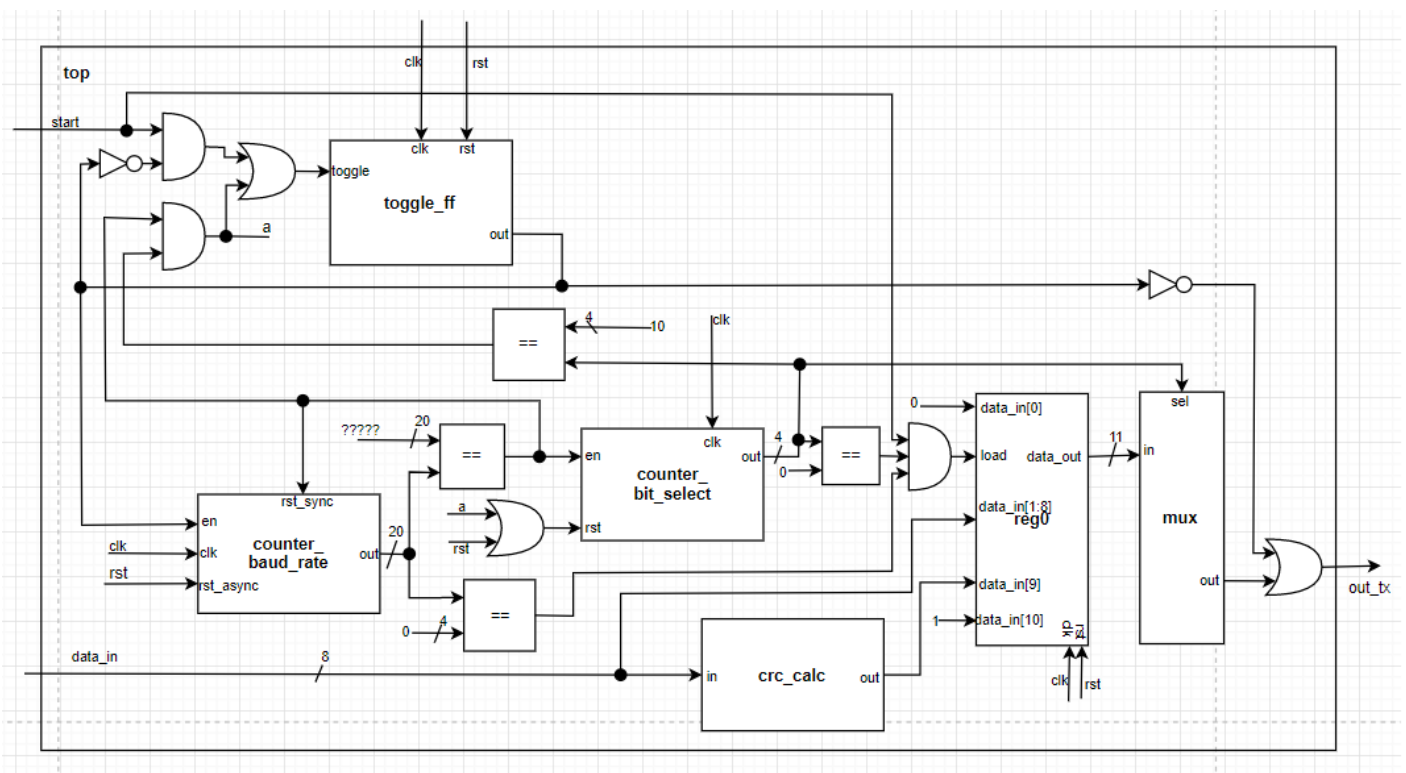
Subiect: UART - TX module

Timp de lucru: 1h30m

Cerinta:

Implementati schema de mai jos in verilog, efectuati simulari pentru verificarea functionalitati acesteia si sintetizati circuitul pe placa FPGA.

Schema bloc:



Descrierea schemei + cerinte speciale:

Circuitul reprezinta un convertor de date paralel->serial, ce transmite datele la iesire conform protocolului UART. Conform acestui protocol, din 8 biti de date la intrare, for iese 11 biti de date, unul cate unul. Un astfel de "pachet" de date este compus din: (LSB) bit de start + 8 biti de date (lsb first) + bit de paritate (crc) + bit de stop. Bitul de start este mereu 0, bitul de stop este mereu 1, bitul de paritate este calculat ca XOR intre toti bitii de date;

Circuitul salveaza datele de la intrare si incepe transmisiunea in momentul in care primeste semnal de "start".

Protocolul UART trimite bitii unul dupa altul cu o viteza semnificativ mai mica decat viteza ceasului sistemului. Una din vitezele standard de transmisune (baud rate) este de 9600 baud ceea ce inseamna ca intr-o secunda se transmit 9600 de biti. Primul numarator este folosit pentru a semnaliza cand a trecut timpul rezervat unui bit si trebuie sa inceapa transmisiunea urmatorului iar al doilea numarator numara efectiv care este bitul curent ce trebuie transmis.

Pentru a nu avea probleme in cazul schimbarii datelor de la intrare in mijlocul transmisiunii, cand se primeste semnal de "start", datele de intrare sunt salvate in registrul reg0 pentru a fi apoi transmise din acesta.

"?????" - Inlocuiti semnul intrebării cu valoarea corectă pentru a avea un baud rate de aprox 9600;

toggle_ff - bistabil de tip toggle, ce isi schimba valoarea pe ceas daca intrarea toggle este activa.

counter_baud_rate - numarator. Atentie la cele 2 surse de reset, una este sincrona, una asincrona

counter_bit_select - numarator

crc_calc - calculeaza paritatea ca fiind XOR pe biti asupra tuturor bitilor de la intrare.

reg0 - registru ce memoreaza (pe ceas) valoarea de la intrarea de date atunci cand primeste comanda de load.

mux - multiplexor.

== - comparator de egalitate.

Pentru simulare:

CLK are o perioada de 10 ns.

Setati data in la 254.

Generati un reset de 10 cicli de ceas.

Dupa reset, la 30ns generati un semnal de start cu durata un ciclu de ceas.

Pentru programat placa:

clk – clock 50MHz

rst - switch [0]

data_in - switch[1:8]

start - switch [9]

out_tx - led red 0

Barem:

Total - 30p

design - 20p

- top - 7p
- toggle_ff - 3p
- counter_baud_rate - 3p
- counter_bit_select - 2p
- mux - 1p
- reg0 - 2p
- crc_calc - 2p

simulare - 6p

- testbench 4p
- demonstratie simulare (imi aratati) - 2p

programare placa - 4p

- fisierul de constrangeri (pin planner) - 2p
- demonstratie functionare (imi aratati) - 2p