

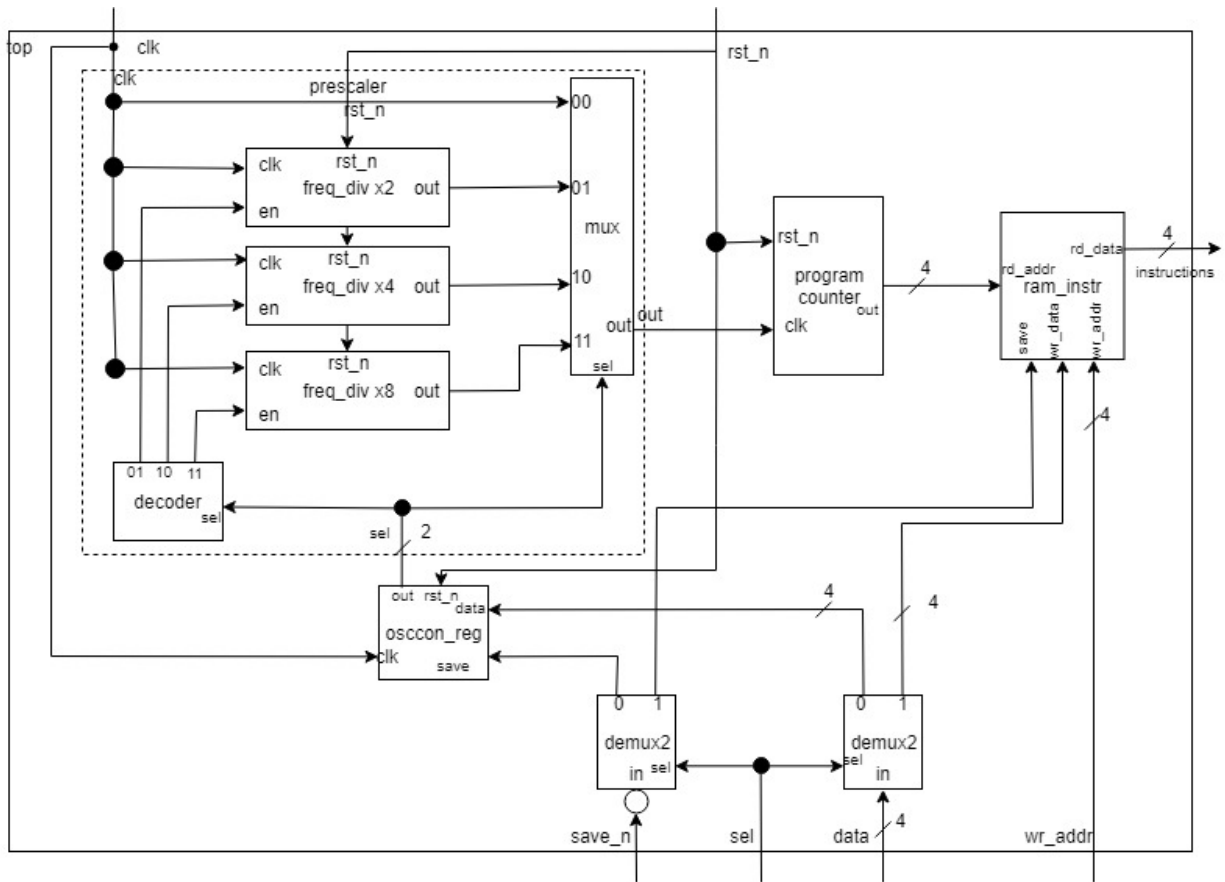
# Subiect: Prescaler circuit

Timp de lucru: 2h

## 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 este o posibila implementare a unui prescaler, circuit folosit in microprocesoare si microcontrollere pentru a avea viteze ale ceasului variabile. Selectia vitezei ceasului (viteza maxima, viteza de 2/4/8 ori mai mica) se face cu ajutorul registrului oscon.

Ceasul astfel generat este folosit de program counter (numaratorul de program) pentru a selecta ce instructiune urmeaza a fi executata din memoria de instructiuni.

OSCCON este un registru de 4b ce memoreaza valoarea din intrarea data atunci cand intrarea sa save este 1. Iesirea out este alcatuita din cei mai putin semnificativi 2b.

Dimensionati corect memoria.

Memorie: save (cu rol de write\_en). Citirea din memorie este asincrona.

RST\_N e pe logica negativa. Atentia la modulele ce il folosesc. Orice reset este sincron.

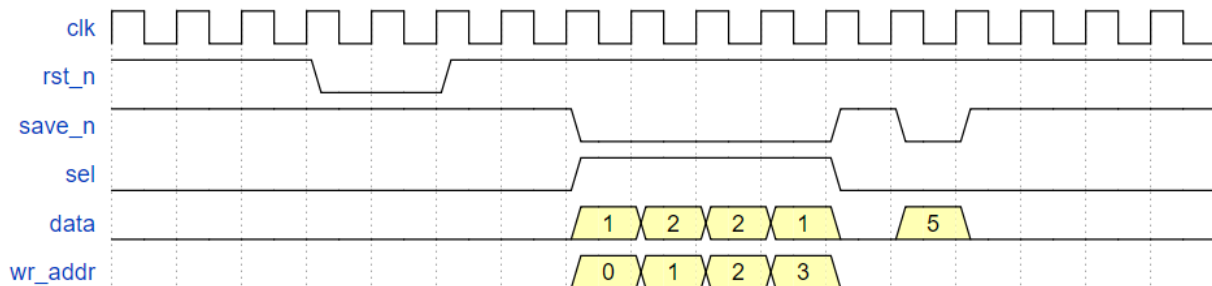
Exista o optimizare posibila majora pentru blocul/modulul de prescaler. Gasirea si implementarea acesteia => puncte bonus.

Blocurile freq\_div scot un semnal de ceas de o frecventa de 2/4/8 ori mai mica decat cea a clock-ului de intrare. Toate cele 3 freq\_div sunt conectate la acelasi reset. Am desenat asa pentru a nu incarca desenul.

Atentie la intrarea in demux2 pentru save\_n, acolo este si o negatie.

## Pentru simulare:

Formele de unda pentru intrare trebuie corespunda cu figura de mai jos:



Spatiul dintre liniile punctate (verticale) reprezinta 10 nano secunde.

Simularea se termina dupa 10\_000 ns.

## Pentru programat placa:

clk - 50MHz  
rst\_n - button [0]  
save\_n - button [1]  
sel - switch [0]  
data - switch [1:4]  
wr\_addr - switch [5:8]  
instructions - led R [3:0]

## Barem:

Total - 30p

design - 20

- top - 5p - 1p interfata + 4p instantieri si conexiuni
- prescaler - 3 - instantieri \_ conexiuni
- decoder - 1p
- mux4 - 1p
- freq\_div - 1.5p (varianta optimizata 5p primii 3 care imi zic)
- osccon\_reg - 2.5p
- demux2 - 1p
- program\_counter - 2p
- ram\_instr - 3p - 1p dimnesionare\_corecta + 2p functionalitate

simulare - 5p

- testbench 3p - 1p instantiere + 2p generarea corecta a semnalelor
- demonstratie simulare (imi aratati) - 2p

programare placa - 5p

- fisierul de constrangeri (pin planner) - 3p
- demonstratie functionare (imi aratati) - 2p