

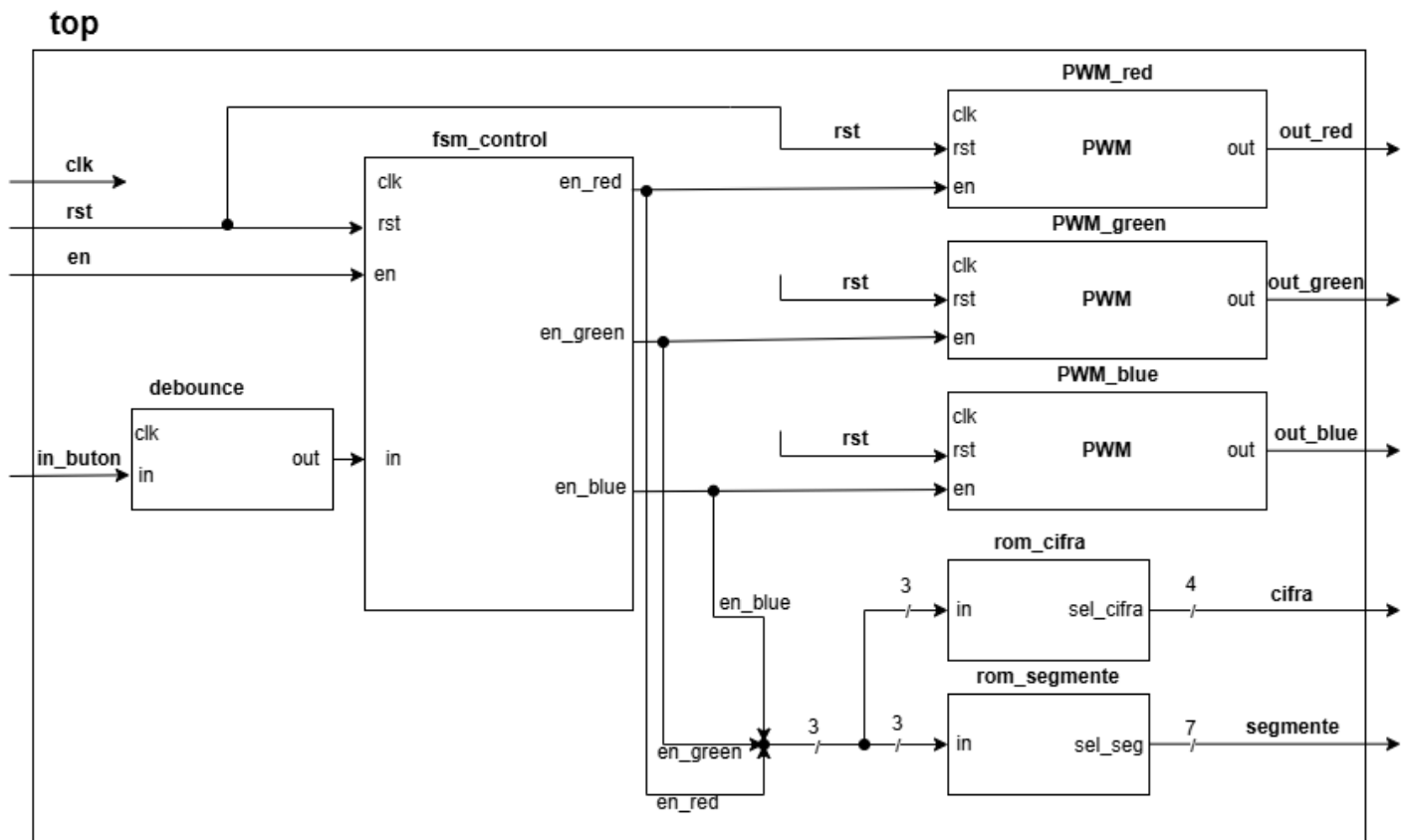
Subiect:

Timp de lucru: 1h20

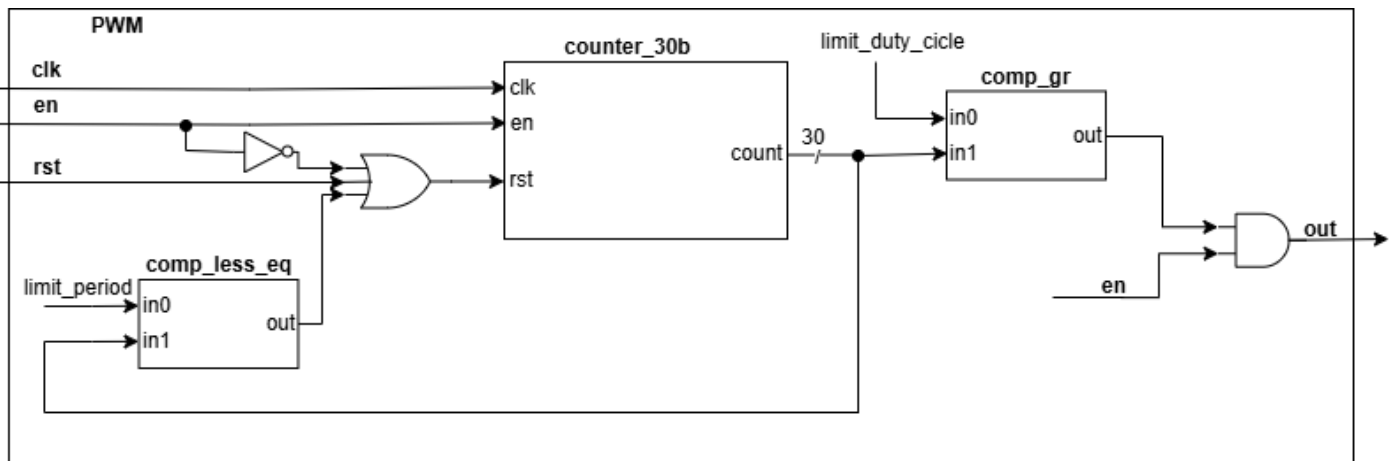
Cerinta:

Implementati schema de mai jos in verilog si efectuati simulari, sinteza si testare pe placa pentru verificarea functionalitati.

Schema bloc:

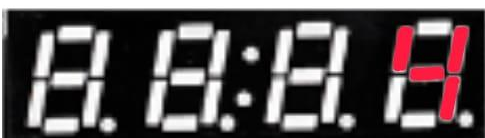


Modulul de PWM:



Descrierea schemei, funtionalitate, cerinte speciale:

Circuitul de mai sus are rolul de a a schimba culoarea afisata pe un led RGB, la o intensitate presetata, atunci cand este apasat un buton. De asemenea acesta va afisa pe 7 segments, in functie de culoare:

- **All off:**
(stare 0) 
- **Red:**
(stare 1) 
- **Green:**
(stare 2) 
- **Blue:**
(stare 3) 
- **White:**
(stare 4) 

Circuitul este alcatuit din:

- **debouce** – circuit ce are rolul de a genera un puls de lungime fix un ciclu de ceas la fiecare apasare a butonului. Se gaseste la:
https://wiki.dcae.pub.ro/index.php/Circuitul_de_debounce
- **PWM** – un modul **parametrizat**, avand ca parametri **limit_period** si **limit_duty_cicle**, are rolul de a seta intensitatea fiecărei culori din ledul RGB, format din:
 - **comp_less_eq** – un comparator de mai mic egal ($\leq$)
 - **comp_gr** - un comparator de mai mare ($>$)
 - **counter_30b** – un numerator pe 30 de biti
- **rom_cifra** – memorie ROM folosita pentru a selecta care cifra (sau cifre) este aprinsa la un moment dat
- **rom_segmente** – memorie ROM folosita pentru a selecta care segmente sunt aprinse la un moment dat
- **fsm_control** – automat care are rolul de a coordona sistemul. La apasarea butonului acesta va trece in urmatoarea **stare**, iar in functie de starea in care se afla, va genera semnale de **enable** pentru fiecare PWM in parte.

Observatii:

1. Pentru simplitatea desenului, unele fire de clock sau reset nu au fost desenate efectiv, nu uitati sa le puneti.
2. Pentru modulul de PWM, **limit_period** si **limit_duty_cicle** sunt **parametri**, **nu** semnale de intrare.

Pentru simulare:

Pentru testarea functionalitatii se va simula apasatul butonului de 3 ori.

Pentru a nu dura prea mult simularea, este recomandat sa se modifice in **top** parametrul **limit** al modulului de **debounce** si folosirea unei perioade mici de **clock**.

Observatie: modulul de debounce necesita tinerea apasata a butonului pt cel putin (**limit * perioada clock**) timp.

Nu uitati sa schimati la loc parametrul **limit** atunci cand vreti sa puneti pe placuta!

Pentru placa:

clk – clock 100 MHz

rst – SW[1]

en – SW[0]

in_buton – BTN[0]

out_red – led RGB0 red

out_green – led RGB0 green

out_blue – led RGB0 blue

cifra – display0, digits

segmente - display0, segments

Barem:

Total - 30p

design – 20p

- Top – 3p
- PWM - 2p
- rom_cifra - 2p
- rom_segmente - 3p
- counter_30b – 2p
- comp_less_eq – 1p
- comp_gr - 1p
- fsm_control – 6p
- debounce – 0p

simulare - 6p

- testbench – 4p
- demonstratie simulare – 2p
 - (forme de unda clare, corecte, grupate corespunzator. Minim top+fsm)

placa – 4p

- fisier constrangeri – 2p
- demonstratie functionalitate – 2p