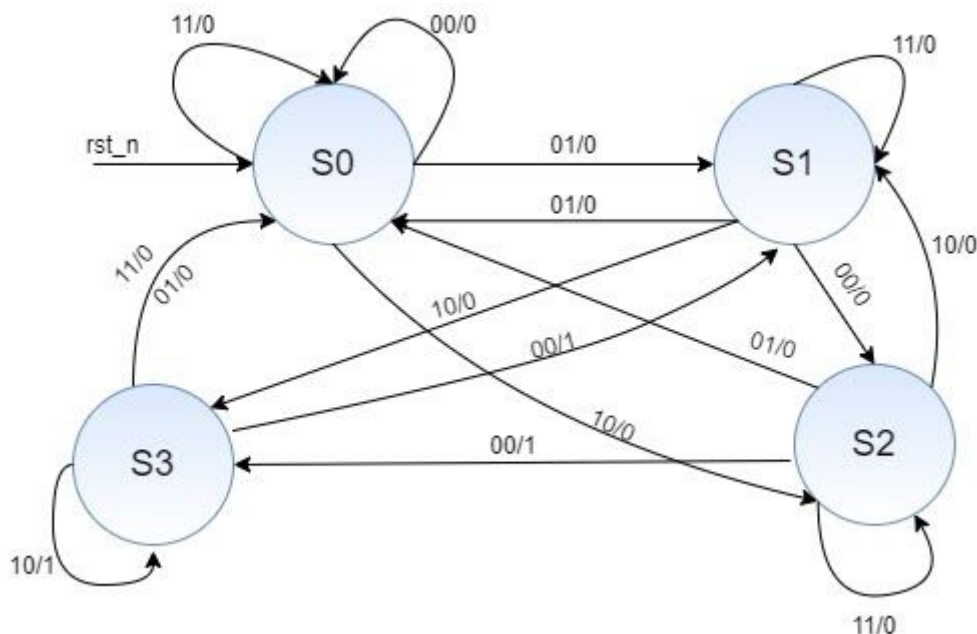


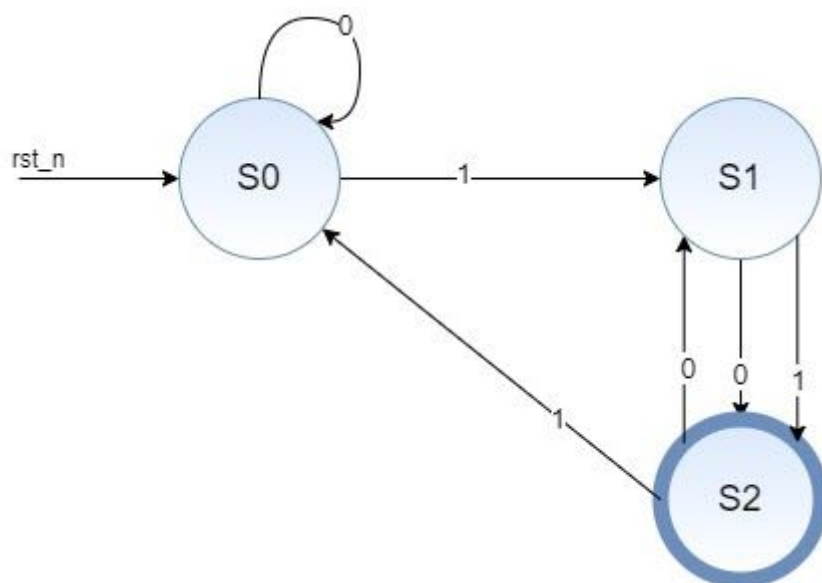
Exerciții

1. (Opțional - de implementat acasă) Implementați automatul de stări din figura de mai jos și identificați tipul acestuia:

a.



b.



2. a. Implementați modulul register pornind de la declarația din fișierul *register.v*. Semnalele *oe* și *we* reprezintă Output Enable, respectiv Write Enable.

- *oe* controlează ieșirea registrului. Când *oe* este high ieșirea este activă având valoarea memorată de registru. Când *oe* este low ieșirea va fi 0. Acest semnal trebuie să fie asincron: modificarea lui va avea efect imediat asupra ieșirii și nu se va aștepta tranziția semnalului de ceas.

- *we* controlează scrierea în registru. Când *we* este high registrul va memora valoarea aflată în semnalul de intrare. Când *we* este low valoarea registrului nu se va modifica, ignorând practic semnalul de intrare. Acest semnal trebuie să fie sincron: modificarea valorii memorate de

registru se face doar în momentul tranziției semnalului de ceas.

- Semnalul *disp_out* este folosit pentru afișare/debugging pe display, iar valoarea acestuia trebuie să fie cea memorată de registru în momentul curent. În mod normal acest semnal nu este prezent într-un calculator. Acest semnal nu trebuie să fie afectat de oe, valoarea disponibilă pe *disp_out* fiind în orice moment egală cu valoarea memorată de registru.

- Semnalul de reset *rst_n* este activ pe low (0).

- *Hint: Puteți folosi operatorul condițional.*

b. Parametrizați modulul **register** astfel încât data de intrare și ieșire din registru să aibă o dimensiune configurabilă.

- *Hint: Utilizați construcția de limbaj parameter*

c. Pornind de la interfața modulului **sequential_multiplier** din scheletul de cod, implementați un automat de stări care să folosească instanțe parametrizate ale modulului **register** pentru a îndeplini următoarele funcționalități:

- La activarea semnalului *write* să se scrie pe câte un registru (parametrizat corespunzător) valorile semnalelor *a* și *b*
- La activarea semnalului *multiply* să fie extrase valorile din cele două registre, să se înmulțească și să se adauge pe un al treilea registru.
- La activarea semnalului *display*, semnalul *out* să primească valoarea aflată pe cel de-al treilea registru.
- Prioritatea celor trei semnale este dată de ordinea în care au fost descrise (e.g. dacă *write* este activ, se ignoră semnalele *multiply* și *display*; dacă *multiply* este activ, se ignoră semnalul *display*)

3. Vi se pune la dispoziție un RAM de tip [Block Memory Generator](#) instanțiat în modulul **ram_reader**. Completați modulul astfel încât să puteți gestiona citirea din memorie de la o adresă *am_out* în momentul în care semnalul *read* este activ (1).