

# Lista subiecte LD

## PARTEA I - Logica combinationala

### 1) Combinational - SM,C1,C2, IEEE, DeMorgan

- a) Ce este un circuit combinational. Dati 3 exemple.
- b) Formatul IEEE754. Campurile unui numar IEEE754. Valoarea numarului
- c) Formatul C2. Exemplu. Avantaje si Dezavantaje.
- d) DeMorgan. 1 exemplu pentru fiecare teorema cu minim 3 termeni.
- e) Formatul BCD. Codificare. Avantaje si Dezavantaje.
- f) Formatul Exces de 3. Codificare. Avantaje si Dezavantaje.
- g) Formatul C1. Avantaje si Dezavantaje.
- h) Ce este un circuit decodicator? 2 exemple de folosire
- i) Teoremele lui DeMorgan. Demonstratie.

### 2) Porti logice

- a) Poarta NAND (cu 3 intrari) si XNOR (cu 2 intrari). Tabela de adevar si simbol.
- b) Poarta AND (cu 2 intrari) si OR (cu 3 intrari). Tabel de adevar si simbol.
- c) Poarta NOR si AND (cu 2 intrari). Tabel de adevar si simbol
- d) Poarta NAND (cu 2 intrari) si NOR (cu 3 intrari). Tabela de adevar si simbol.
- e) Poarta NOR (3 intrari) si XOR (2 intrari).Tabel de adevar si simbol.
- f) Poarta AND (cu 2 intrari) si XNOR (cu 2 intrari). Tabel de adevar si simbol.
- g) Poarta OR (2 intrari) si XNOR (2 intrari). Tabela de adevar si simbol

### 3) MSI combinational

- a) Decodicatorul. Definitie. Exemplificati prin decodicator 2-la-4 (tabel de adevar, schema logica).
- b) Multiplexorul. Definitie. Exemplificati prin multiplexor 8-la-1 (tabel de adevar, schema logica).
- c) Multiplexorul. Definitie. Exemplificati prin multiplexor 4-la-1 (tabel de adevar, schema logica).
- d) Codicator de prioritate. Exemplificare pentru un codicator cu 3 intrari.
- e) Demultiplexorul. Definitie. Exemplificati prin demultiplexor 1-la-4 ( tabel de adevar, schema logica)
- f) Decodicator. Exemplificare pentru un decodicator 2-la-4 cu 1 intrare de validare (tabel de adevar si schema logica)
- g) Multiplexorul. Definitie. Exemplificati prin multiplexor 4-la-1 cu 1 intrare de validare(tabel de adevar, schema logica).

### 4) Aplicatii cu MSI combinational

- a) Comparator pe 2 biti.
- b) Scazator folosind Ripple Carry Adder pe 2 biti.
- c) Sumator Ripple Carry Adder pe 3 biti.
- d) Comparator pe 2 biti. Exemplificare pentru mai mare sau egal.
- e) Scazator folosind Ripple Carry Adder pe 3 biti

- f) Comparator pe 2 biti pentru operatia mai mic sau egal.
- g) Sumator Carry Look Ahead pe 2 biti

### 5) Functii logice de minimizat

- a) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,4,8,10,11) + \Sigma_d(2,4,6,7)$ 
  - i) Sa se minimizeze functia
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- b) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,2,4,14) + \Sigma_d(5,6,7,10)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- c) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(2,4,8,10) + \Sigma_d(0,6,7)$ 
  - i) Sa se minimizeze functia
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- d) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,8,12,14) + \Sigma_d(1,3,9,11,15)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- e) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,1,5,10) + \Sigma_d(2,8,13)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8 si poarta NAND
- f) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,4,5,6) + \Sigma_d(2,7,12,14)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- g) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(8,12,14,15) + \Sigma_d(4,6,10,13)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- h) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(4,6,10,13) + \Sigma_d(0,8,12,14,15)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8
- i) Se da functia logica  $f(x_3, x_2, x_1, x_0) = \Sigma(0,4,6,8) + \Sigma_d(2,5,10)$ 
  - i) Sa se minimizeze functia si sa se implementeze cu porti
  - ii) Sa se implementeze folosind decodificatoare 3-la-8

### 6) Conversoare

- a) Sa se realizeze un convertor din format SM in format C2 pe 3 biti
- b) Sa se realizeze un convertor din format Exces de 3 in format BCD
- c) Sa se realizeze un convertor din format C2 in format SM pe 3 biti
- d) Sa se realizeze un convertor din format C1 in format C2 pe 3 biti
- e) Sa se realizeze un convertor din format C1 in format SM pe 4 biti.

### 7) Aplicatii 1 Combinationale

- a) Sa se implementeze un circuit care implementeaza operatia (A ROL B), unde ROL este operatorul de rotatie la stanga. A este un numar pe 6 biti si B este un numar pe 3 biti. (COMBINATIONAL)

- b) Sa se implementeze un circuit care implementeaza operatia (A ROR B), unde ROR este operatorul de rotatie la dreapta. A este un numar pe 6 biti si B un numar pe 3 biti. (COMBINATIONAL)
- c) Sa se implementeze un circuit de shiftare pentru numere pe 8 biti, care poate sa shifteze cu 0,1,2 sau 3 pozitii folosind MUX 4-la-1. Se va puncta varianta optima din punct de vedere al costului. (COMBINATIONAL)
- d) Sa se implementeze un circuit de shiftare pentru numere pe 8 biti care poate sa shifteze cu 0,1,2 sau 3 pozitii folosind MUX 4-la-1. Se va puncta varianta optima din punct de vedere al costului. (COMBINATIONAL)

## 8) Aplicatii 2 combinationale

- a) Sa se implementeze in mod optim o unitate aritmetico-logica pe 1 bit care sa realizeze urmatoarele operatii: AND logic, XOR logic, NOR logic si diferenta. Unitatea va avea ca si intrari doua numere pe 1 bit (intrarile de date), precum si intrarile de codificare a operatiilor, iar ca iesire va avea rezultatul pe 1 bit al operatiei selectate precum si 1 bit pentru transfer (borrow-out), in cazul scaderii. Acest bit va avea valoarea 0 pentru operatii logice. (COMBINATIONAL)
- b) Sa se implementeze in mod optim o unitate aritmetico-logica pe 1 bit care sa realizeze urmatoarele operatii: AND logic, XOR logic, OR logic si suma. Unitatea va avea ca si intrari doua numere pe 1 bit (intrarile de date), precum si intrarile de codificare a operatiilor, iar ca iesire va avea rezultatul pe 1 bit al operatiei selectate precum si 1 bit pentru transfer (carry-out), in cazul adunarii. Acest bit va avea valoarea 0 pentru operatii logice. (COMBINATIONAL)

## PARTEA II - Logica secventiala

### 1) Intrebari secvential

- a) Ce este un latch?
- b) Ce este un flip-flop?
- c) Ce este un circuit secvential? Dati 3 exemple.
- d) Ce este un circuit secvential? Dati 2 exemple.
- e) Reset sincron vs reset asincron.
- f) Ce este un latch asincron? (Implementare, diagrama de timp)
- g) Ce este un edge-triggered flip-flop? Enumerati arhitecturile de edge triggered FF

### 2) Latch, Flip-flop

- a) Flip-Flop D Master-Slave. Tabel de adevar, schema, diagrama de timp.
- b) Latch D sincron. Tabel de adevar, schema, diagrama de timp.
- c) Latch JK sincron. Table de adevar, schema, diagrama de timp.
- d) Latch RS asincron. Tabel de adevar, schema. Diagrama de timp.
- e) Latch JK sincron. Tabel de adevar, schema, diagrama de timp.

### 3) Automate, registrii, numaratoare

- a) Automat de tip Mealy. Dati exemplu de diagrama pentru un astfel de automat.
- b) Registru intrare seriala-iesire paralela. Exemplificare pentru registru pe 3 biti.
- c) Numarator sincron. Diagrama de timp pentru un numarator sincron descrescator pe 2 biti.
- d) Registri intrare seriala-iesire paralela. Exemplificare pentru registru pe 2 biti.
- e) Registru cu intrare seriala si iesire seriala. Schema pentru un registru pe 2 biti.
- f) Registru intrare paralela - iesire paralela. Exemplificare pentru registru pe 3 biti.
- g) Registre cu intrare paralela si iesire seriala. Schema pentru un registru pe 2 biti.

#### **4) Memorii + pila de registre**

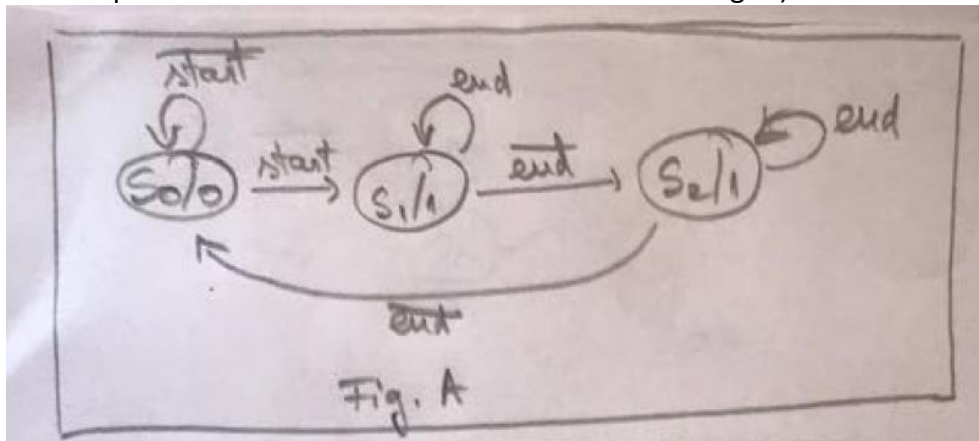
- a) SRAM vs DRAM.
- b) Memorii volatile cu acces aleator. Ce reprezinta accesul aleator? Exemple.
- c) Memorii cu acces ne-aleator
- d) Ce este o memorie cu acces ne-aleator? Exemple
- e) Ce este o memorie nevolatila. Exemple
- f) Memorii volatile cu acces aleator. Exemple
- g) Pila de registre. Arhitectura pentru o pila de 3 registre pe 8 biti.
- h) Pila de registre. Arhitectura pentru o pila de 3 registre pe 6 biti.

## 5) Aplicatii 1 Secventiale (Numaratoare)

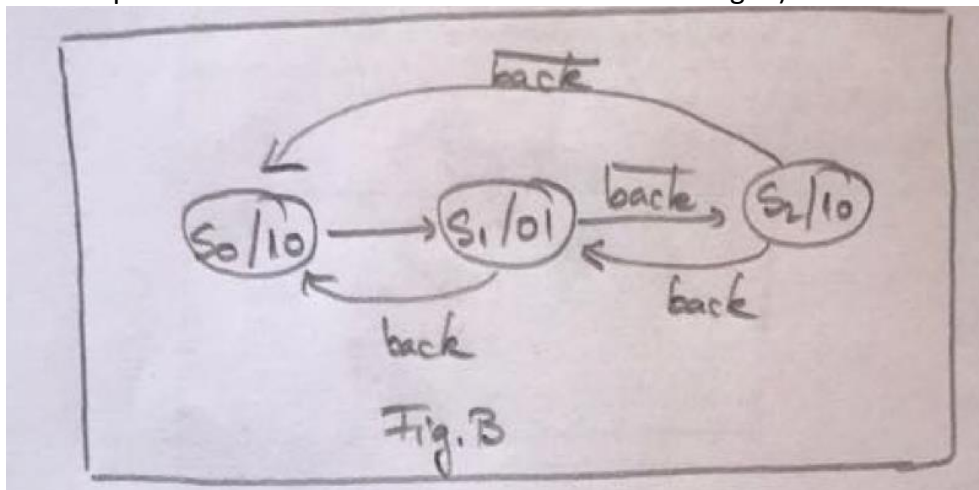
- Sa se implementeze un numarator care sa numere in mod descrescator seria lui Fibonacci pe 4 biti
- Sa se implementeze un numarator care sa numere in mod crescator o serie formata din toti multiplii numarului 3 pe 4 biti (seria va include si numarul 0).
- Sa se implementeze un numarator sincron Gray. O secventa de numere codificata Gray se caracterizeaza prin faptul ca 2 numere invecinate sunt adiacente.
- Sa se implementeze un numarator care sa numere in mod crescator seria lui Fibonacci pe 4 biti (0,1,2,...).
- Sa se implementeze un numarator care sa numere in mod descrescator o serie formata din toti multiplii reprezentati pe 4 biti ai numarului 3 (seria va include si numarul 0)
- Sa se implementeze un numarator pe 4 biti care sa numere in mod descrescator secventa formata din 0 si numerele prime.

## 6) FSM

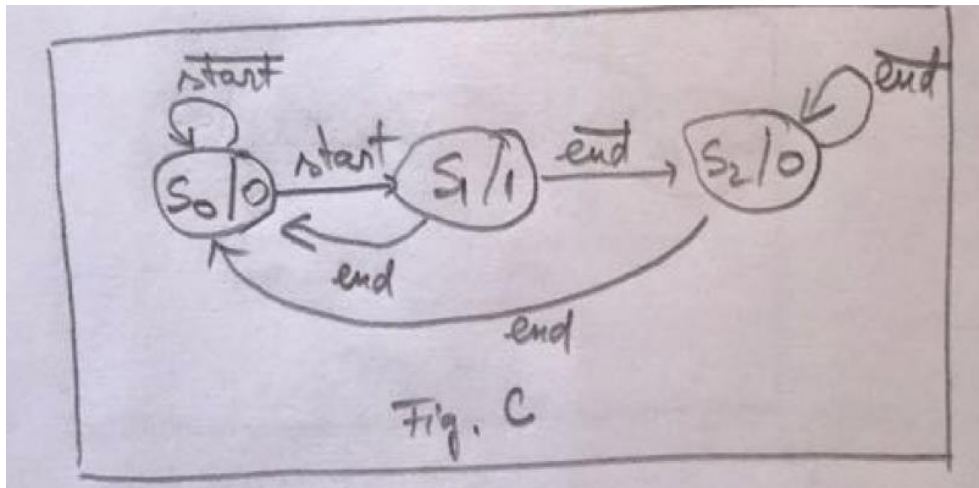
- Sa se implementeze urmatorul automat cu stari finite Fig. A)



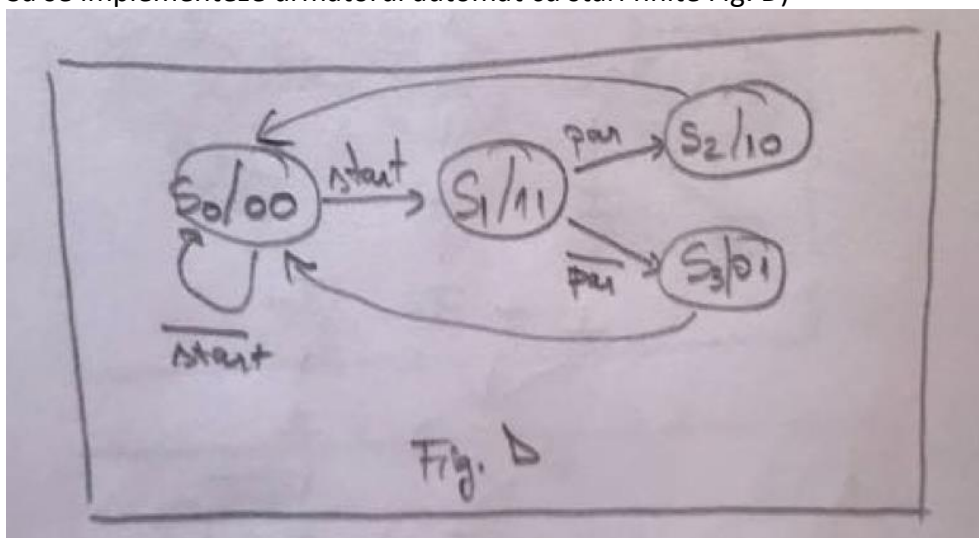
- Sa se implementeze urmatorul automat cu stari finite Fig. B)



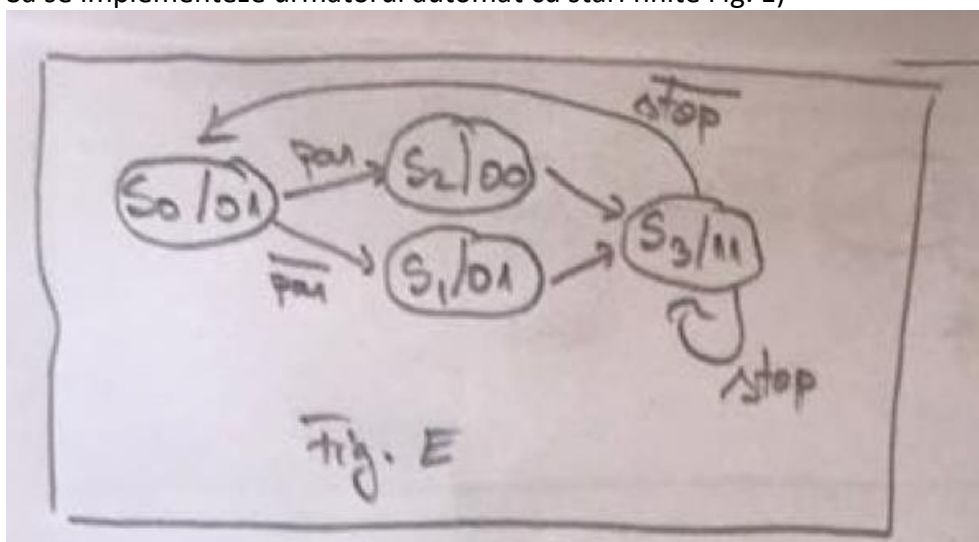
- Sa se implementeze urmatorul automat cu stari finite Fig. C)



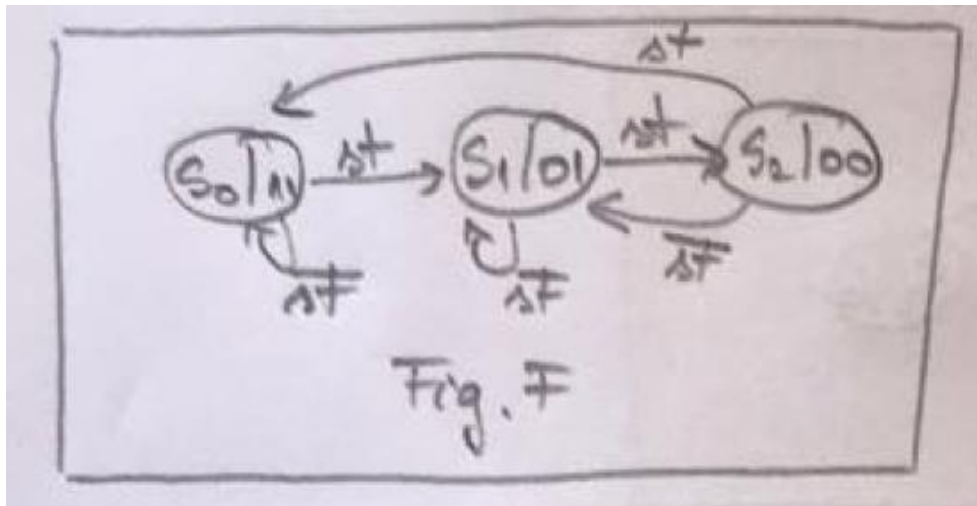
d) Sa se implementeze urmatorul automat cu stari finite Fig. D)



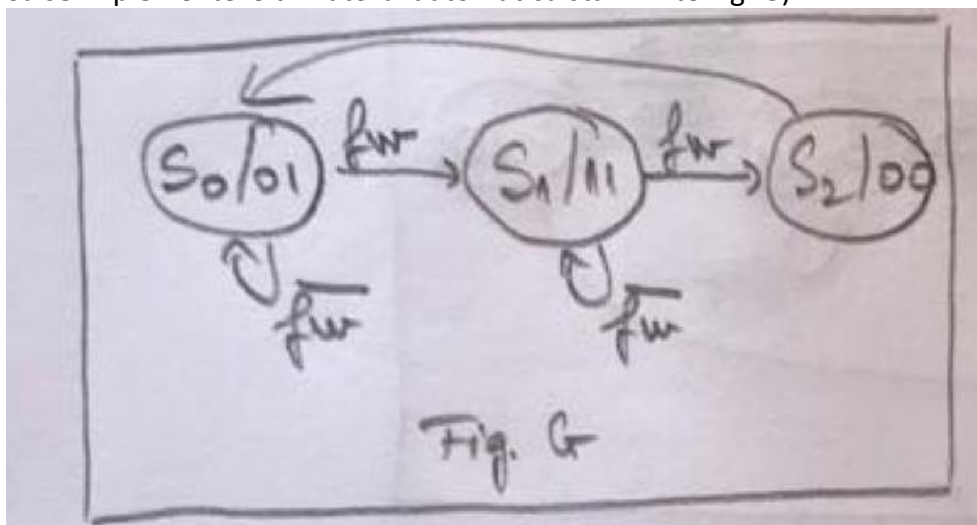
e) Sa se implementeze urmatorul automat cu stari finite Fig. E)



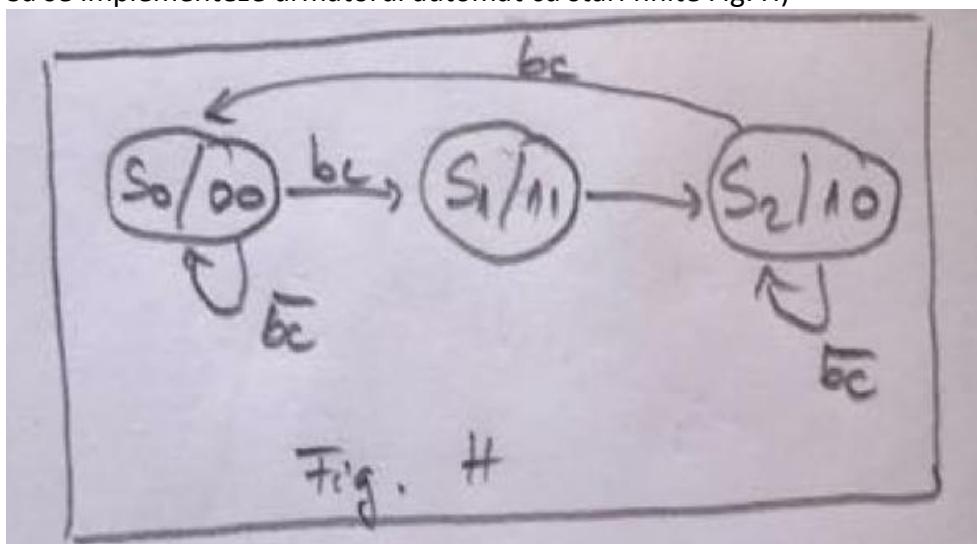
f) Sa se implementeze urmatorul automat cu stari finite Fig. F)



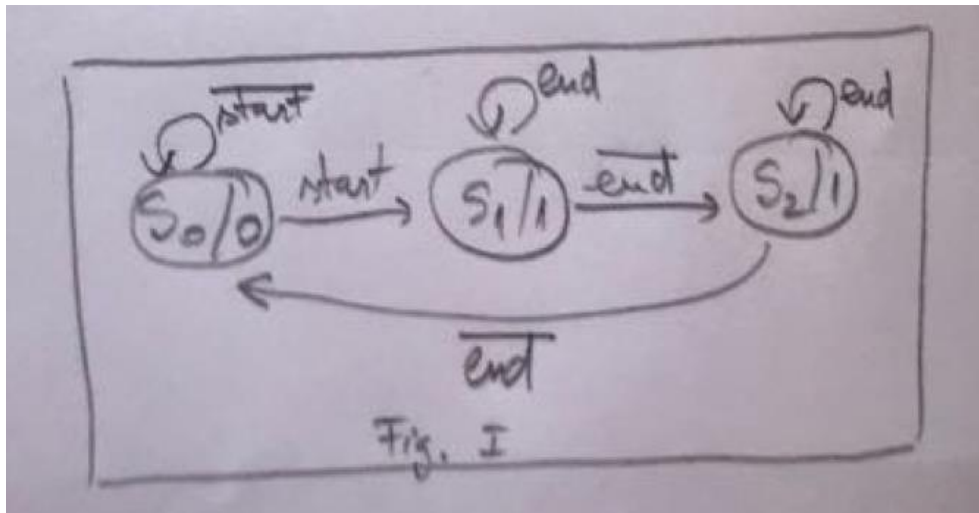
g) Sa se implementeze urmatorul automat cu stari finite Fig. G)



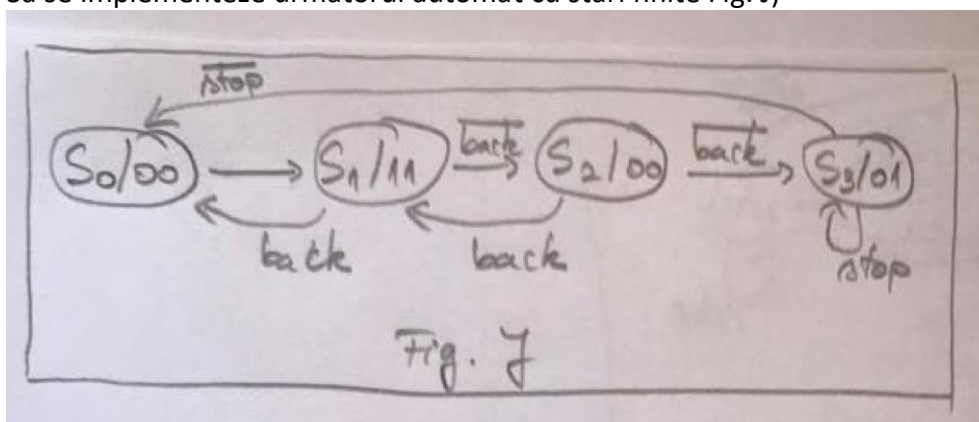
h) Sa se implementeze urmatorul automat cu stari finite Fig. H)



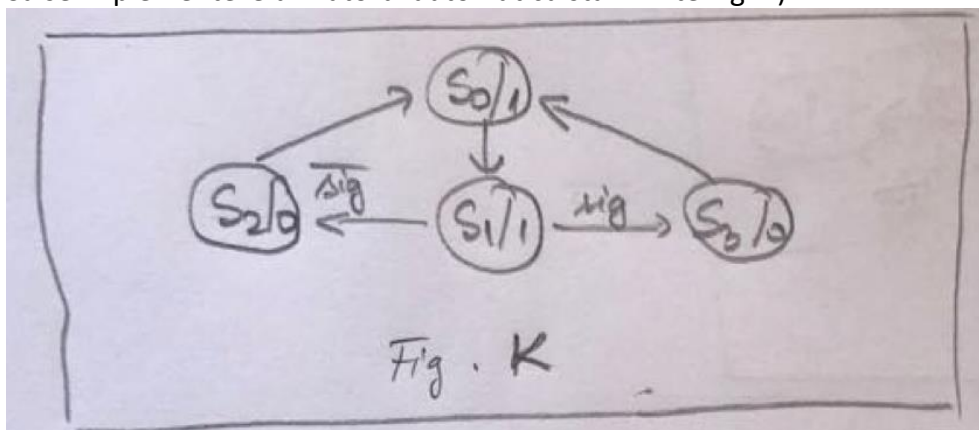
i) Sa se implementeze urmatorul automat cu stari finite Fig. I)



j) Sa se implementeze urmatorul automat cu stari finite Fig. J)

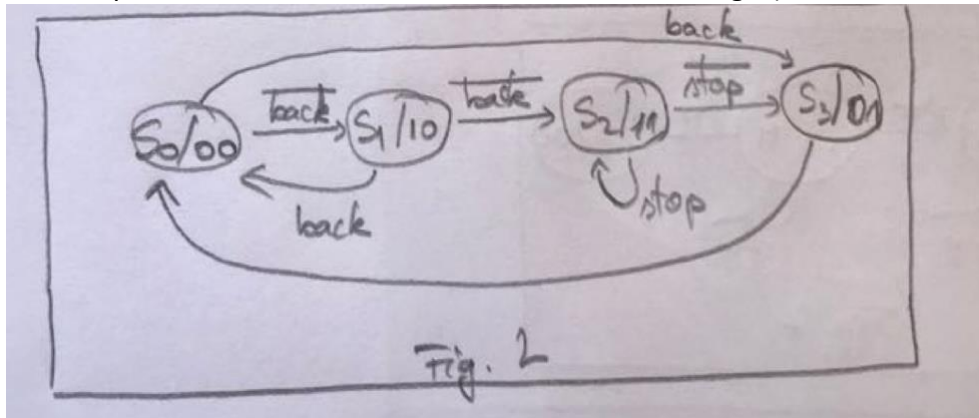


k) Sa se implementeze urmatorul automat cu stari finite Fig. K)





- l) Sa se implementeze urmatorul automat cu stari finite Fig. L)



## 7) Aplicatii 2 Secventiale

- Sa se implementeze un circuit care realizeaza inmultirea a 2 numere prin adunari repetate. Circuitul va avea in componenta doua registre, un numarator si un sumator, respectiv logica aditionala. Descrieti succint rolul si modul de functionare a fiecarei componente.
- Sa se implementeze un circuit care masoara o microsecunda. La fiecare microsecunda trecuta, va activa un semnal. Circuitul este conectat la un semnal de tact de 200MHz.
- Sa se implementeze un circuit care determina secventa maxima de biti de 0 consecutiva dintr-o serie de 7 biti ai unui sir de biti de date receptionati. Bitii sunt receptionati serial. Se va puncta varianta optima din punct de vedere al implementarii.
- Sa se proiecteze un circuit ce implementeaza o regula de prioritate round-robin pentru 4 canale. Regula round-robin se bazeaza pe faptul ca daca un canal a avut prioritate la momentul T, prioritatea lui la momentul T+1 va fi cea mai mica.
- Sa se implementeze un circuit care recunoaste o secventa de cel putin 2 de 0 logic consecutivi dintr-o secventa de 5 biti. Secventa de 5 biti se incarca in circuit in mod serial.