

S9. Construirea unităților de control

Codificarea One Hot pentru implementarea mașinilor cu stări finite

1. Scopul laboratorului

Construirea unitatilor de control utilizand codificarea One Hot

2. Desfășurarea laboratorului

2.1 Fundamente teoretice

Codificarea One Hot pentru un FSM având n stări utilizează n elemente de stocare. Fiecare element de stocare este asociat cu o stare. În consecință, la fiecare moment de timp, doar unul din cele n elemente de stocare are ieșirea activă. FSM-ul implementat folosind codificarea One Hot utilizează un număr crescut de elemente de stocare, însă beneficiază de o implementare directă și depanare facilă.

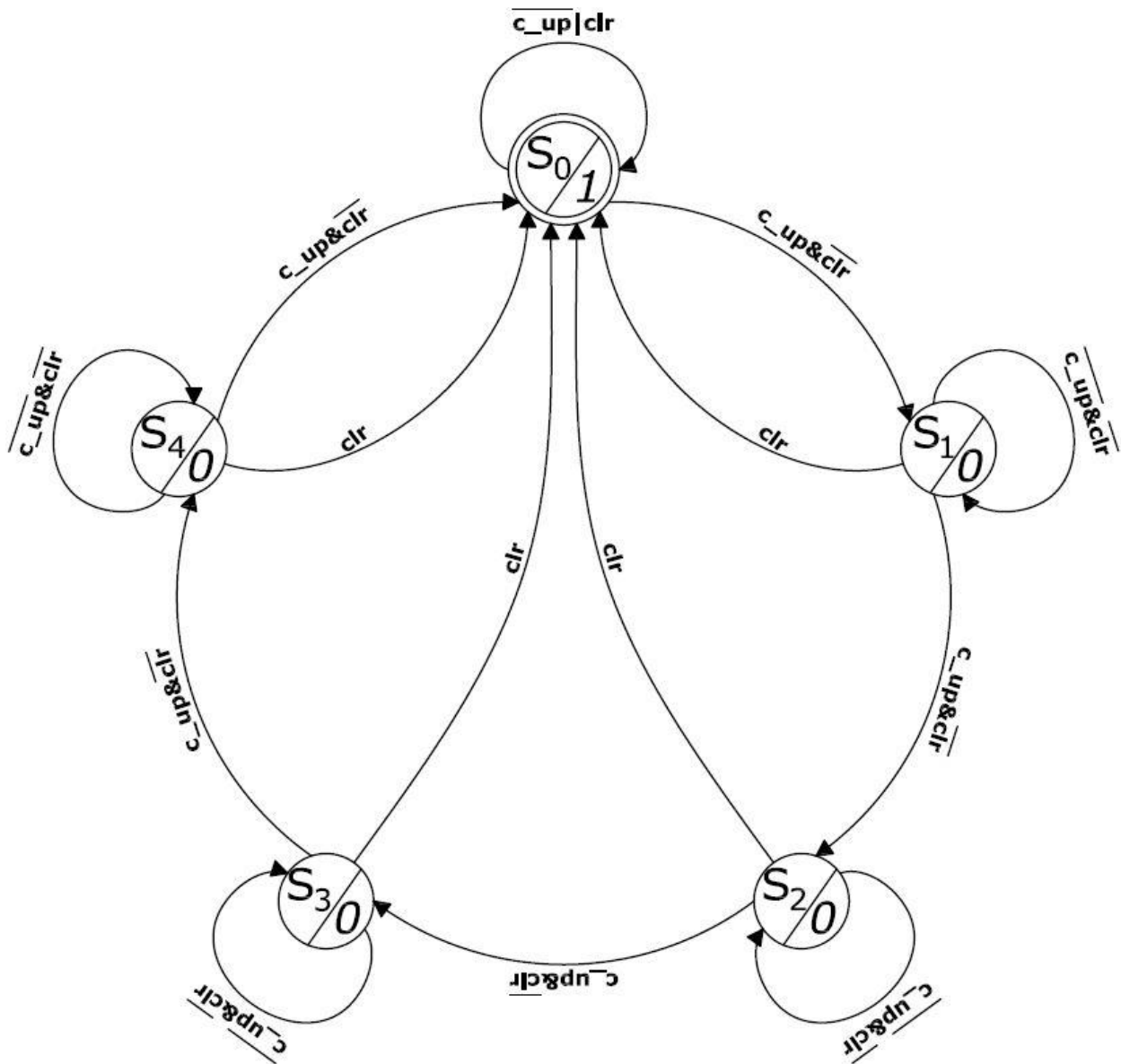
2.2.1 Numărător divide-by- n

Un numărător divide-by- n împarte frecvența semnalului de tact de la intrare cu factorul n și este construit în jurul unui numărător modulo- n . Un numărător modulo- n este compus din:

- Un registru pe r biți ($r = \lceil \log_2 n \rceil$)
- O componentă realizând incrementarea conținutului registrului
- Un semnal sincron de ștergere a registrului atunci când conținutul acestuia atinge valoarea $n-1$.

Numărătorul divide-by- n are 2 intrări sincrone: **c_up**, pentru activarea înaintării în secvența de numărare și **clr**, pentru ștergerea conținutului. De asemenea, numărătorul divide-by- n are ieșirea pe 1 bit **dclk** (divided clock), activată o dată la fiecare n cicluri de ceas.

2.2.2 Diagrama tranzițiilor de stare: numărător divide-by-5



Designul folosește 5 bistabile de tip *D*, *FF0*, *FF1*, *FF2*, *FF3* și *FF4*, cu intrările *Di* și ieșirile *Qi*, asociate celor 5 stări *S0*, *S1*, *S2*, *S3* și *S4* (așa cum este ilustrat în diagrama de mai sus). La fiecare moment doar unul din cele 5 bistabile este activ, având ieșirea *Qi* egală cu 1. Starea curentă este indicată de bistabilul cu ieșirea activă. Dacă *Q1* este activă starea curentă este *S1*, dacă *Q4* este 1 atunci *S4* este starea curentă, șamd. La intrările *Di* se conectează ecuațiile booleene care activează stările *Si*. Designul FSM-ului se reduce la scrierea acestor ecuații. Exemplu: Din diagrama tranzițiilor, starea următoare este *S1* dacă starea curentă este *S0*, *c_up* este 1 și *clr* este 0, sau starea curentă este *S1* și nici *c_up*, nici *clr*, nu sunt 1. Deci vom putea scrie corespunzător:

$$D_1 = Q_0 \cdot c_up \cdot \overline{clr} + Q_1 \cdot (\overline{clr} \cdot \overline{c_up})$$

Celelalte 4 intrări sunt construite similar.

3. Numărător divide-by-5: codificare One Hot și cod Verilog

Lucrare de laborator

Completați șablonul de mai jos cu ecuațiile booleene lipsă deduse din diagrama tranzițiilor de stare de la pagina 2:

```

1  module d5cntr(
2      input clk,
3      input rst_b,  //asynch
4      input clr,
5      input c_up,
6      output dclk
7  );

9      //vector of inputs D for all 5 flip flops (FFs); din[0] is input to FF0;
10     //FF0 indicates whether the current state is S0 or not
11     reg [4:0] din;
12     //next value for all 5 FFs; corresponds to the next state
13     wire [4:0] din_nxt;

15     //boolean equation setting the next state to S1:
16     //      D1      =      Q0      * c_up * (~clr) +      Q1      * (~c_up) * (~clr)
17     assign din_nxt[1] = (din[0] & c_up & (~clr)) | (din[1] & (~c_up) & (~clr));
18     //boolean equations for activating states S0, S2, S3, S4
19     //...

21     //output activation
22     assign dclk = din[0]; //dclk activated once every 5 clock cycles;
23     //one can choose any state to activate the output

25     //update the current states vector
26     always @ (posedge clk, negedge rst_b)
27         if (! rst_b) din <= 5'd1; //set current state to S0 by
28         //setting FF0 and clearing all other FFs
29         else din <= din_nxt;
30 endmodule

```

4. Referințe bibliografice

[Vlad12] M. Vlăduțiu, “Computer Arithmetic: Algorithms and Hardware Implementations” Springer, 2012.