# Circuite logice secvențiale – Lab 3

## Realizarea diferitelor tipuri de circuitelor basculante bistabile

**Tema:** Să se proiecteze şi să se realizeze simultan in interiorul unui circuit GAL un bistabil de tip D de tip *master-slave* prin sinteză logică şi un altul prin folosirea particularităților limbajului ABEL (pritr-o conectare corespunzatoare a bistabilului intern pre-existent in macrocelula).

Pentru implementarea combinationala a unui bistabil de tip D master-slave plecam de la celula de baza prezentata mai jos:

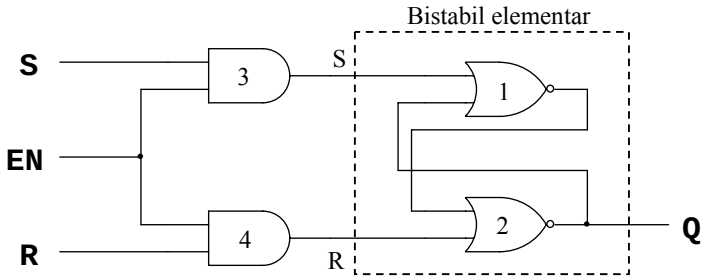


**Figura 1.** Schema logica a unui bistabil de tip RS cu intrare de validare – bistabil RS de tip latch

În **Figura 1** se recunoaşte schema fundamentală a unui bistabil RS de tip latch care este formată dintr-un bistabil elementar compus din porțile 1 şi 2 precedate de porțile de acces 3 şi 4 care introduc în plus o intrare de tip EN (Enable) sau T (tact) sau CL sau CK (clock) care are rolul de a stabili momentul în care acționează intrările RS. Circuitul global din **Figura 1** devenind astfel un circuit sincron comparativ cu bistabilul elementar (parte componentă a circuitului) care este un bistabil asincron (imediat ce o comandă este aplicată acest bistabil o execută).

Ecuațiile care descriu funcționarea circuitului din **Figura 1 s**unt date mai jos:

$$Q = \overline{\left(\overline{(Q + S \cdot EN)} + R \cdot EN\right)} = (Q + S \cdot EN) \cdot \left(\overline{R \cdot EN}\right) \quad (1)$$

$$Q = \overline{(R \cdot EN)} \cdot (S \cdot EN) + \overline{(R \cdot EN)} \cdot Q \quad (2)$$

$$Q = \overline{R} \cdot S \cdot EN + \overline{R} \cdot Q + \overline{EN} \cdot Q \quad (3)$$

Ecuația finală (3) descrie funcționarea globală a bistabilului RS de tip latch prezentat în **Figura 1**.

Pentru transformarea unui bistabil de tip RS latch într-un bistabil de tip D între intrările de set şi reset se conectează o poartă inversoare. Schema de transformare a unui bistabil RS în unul de tip D este dată în **Figura 2**.

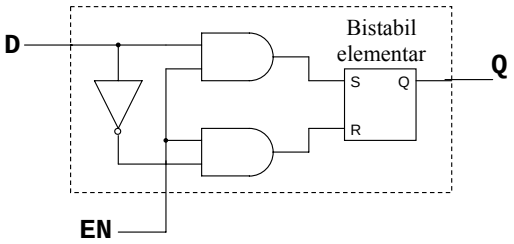


**Figura 2**. Schema de transformare a unui bistabil de tip RS în unul de tip D

Din **Figura 2** se observă că pentru a transforma un bistabil RS în unul de tip D trebuie să introducem o poartă inversoare între intrările S şi R ale bistabilului. Din punct de vedere logic putem scrie acum relația:

$$D = S = \overline{R} \quad (4)$$

Utilizându-ne de relația (4) şi introducând-o în relația (3) vom obține astfel ecuația bistabilului de tip D:

$$Q = D \cdot EN + D \cdot Q + Q \cdot \overline{EN} \quad (5)$$

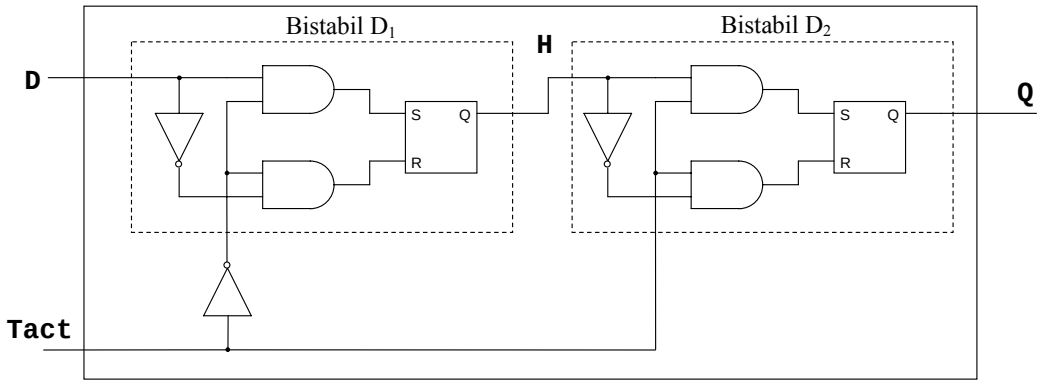


**Figura 3.** Realizarea bistabilului de tip D printr-o implementare de tip master-slave

Pentru realizarea bistabil de tip D de tip *master-slave* am ales implementarea din **Figura 3** ce utilizează transformarea unui bistabi de tip RS în unul de tip D prezentată în **Figura 2**. Utilizându-ne de relația (5) pentru descrierea funcționării bistabilului $D_2$ şi a bistabilului $D_1$ obținem:

$$Q = Q \cdot \overline{Tact} + Q \cdot H + H \cdot Tact \quad (6)$$

$$H = H \cdot Tact + H \cdot D + D \cdot \overline{Tact} \quad (7)$$

Ultimul pas care ne rămâne de făcut este să implementăm relațiile (6) şi (7) în cadrul programului ce va fi scris în ABEL. Din **Figura 3** se observa existența ieşirii H. Această ieşire, H, va fi scoasă la un pin al circuitului dar nu va fi folosita, acest lucru este obligatoriu deoarece în interiorul unui GAL de tipul 16V8 putem scrie doar relații logice ce sunt caracterizate de un nivel de SI urmat de unul de SAU.

Pentru o implementare a unui bistabil de tip D pritr-o conectare corespunzatoare a bistabilului intern pre-existent in macrocelula se prezintă în **Figura 4** schema internă a unei macrocelule în modul registru de lucru care permite accesarea bistabilului intern.

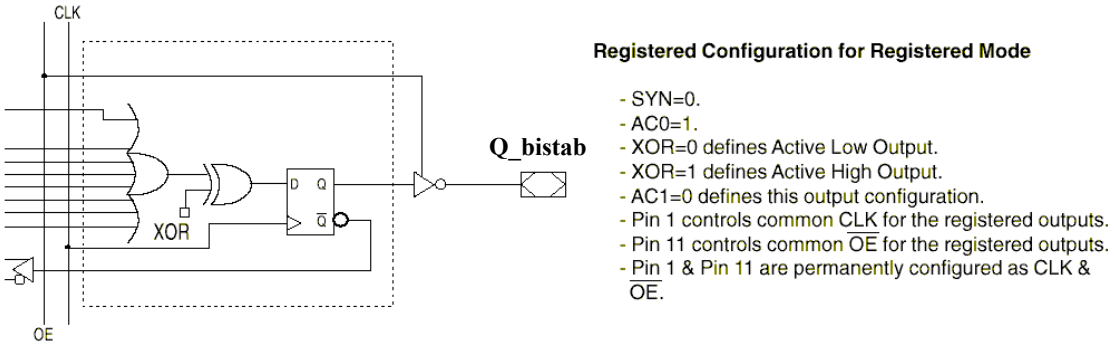


**Figura 4.** Configurația de tip registru a macrocelulei în modul *registru* de lucru

Dacă pinul de ieşire a macrocelulei este denumit Q_bistab atunci accesarea intrării D a bistabilului se face prin intermediul secvenței „Q_bistab := ....” (diferită față de cea utilizată în cadrul descrierii unei structuri combinaționale care este de tipul „Q = ....”), accesarea intrării de clock este realizată prin intermediul secvenței: „Q_bistab.clk = ...”, în timp ce pentru accesarea pinului de output enable ne folosim de următoarea secvență în ABEL: „Q_bistab.oe = ...”.

Codul programului (atat pentru realizarea combinationala cat si secventiala) este următorul:

```
MODULE Secv

TITLE 'Doua implementari de bistabile de tip D'

declarations

@alternate

"declarare pini de intrare
 D1, Tact1, D2, Tact2, /OE pin
   2,     1,     6,     7,     11;

"seclarare pini de iesire
  H, Q2inv, Q2, Q1inv pin
 12,    14,    15,     18   istype 'com';

  Q1 pin 19 istype 'reg';

equations

 Q1.clk = Tact1;
 Q1 := D1;
 Q1inv = /Q1;

 H  = D2 & /Tact2 + D2 & H + H & Tact2;
 Q2 =  H & Tact2  + H & Q2 +  Q2 & /Tact2;
 Q2inv = /Q2;

 [Q1inv, Q2, H, Q2inv].oe = OE;

END
```

## Dezvoltarea unui numarator realizat pe baza diagramei de stare

**Tema:** Să se construiască cu ajutorul diagramei de stare un numărător pe 3 biți. Acest numărător va oferi posibilitatea alegerii sensului de numărare.

Intrările în sistem se vor nota cu x în timp ce ieşirile cu z. Diagrama de stare pentru numărotorul considerat este:

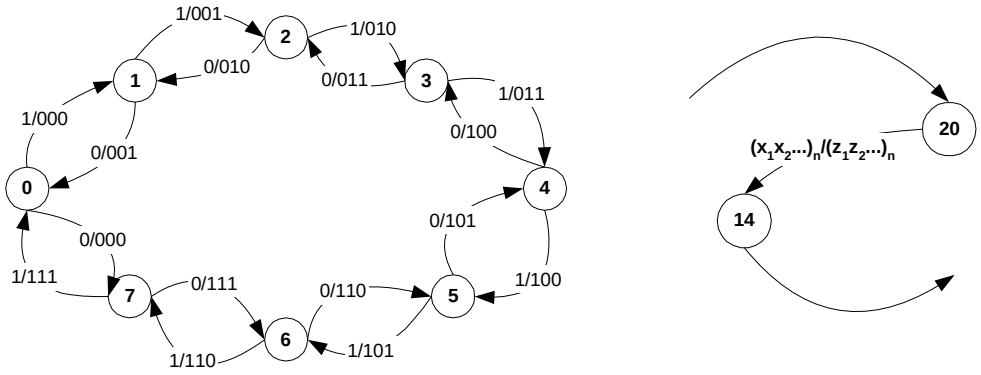


**Figura 5.** Graful de fluiență pentru numarătorul de 3 biți considerat

**Tabelul 1.** Matricea de fluenta

| $(Q_2Q_1Q_0)_n \setminus x_n$ | 0 | 1 |
|---|---|---|
| 000 | 111/000 | 001/000 |
| 001 | 000/001 | 010/001 |
| 010 | 001/010 | 011/010 |
| 011 | 010/011 | 100/011 |
| 100 | 011/100 | 101/100 |
| 101 | 100/101 | 110/101 |
| 110 | 101/110 | 111/110 |
| 111 | 110/111 | 000/111 |

În tabelul de fluență anterior în coloanele doi (pentru starea logica 0 pe intrarea care determină sensunsul de numărare) şi trei (pentru starea logica 1 pe intrarea care determină sensunsul de numărare) informația reprezintă:

$$(Q_2Q_1Q_0)_{n+1}/(Q_2Q_1Q_0)_n.$$

Pentru bistabilul de tip D avem ecuația de funcționare: $Q_{n+1}=D_n$.

| $(Q_2Q_1Q_0)_n \setminus x_n$ | $D_{2n}$ | | $D_{1n}$ | | $D_{0n}$ | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 0 | 1 |
| 000 | 1 | 0 | 1 | 0 | 1 | 1 |
| 001 | 0 | 0 | 0 | 1 | 0 | 0 |
| 010 | 0 | 0 | 0 | 1 | 1 | 1 |
| 011 | 0 | 1 | 1 | 0 | 0 | 0 |
| 100 | 0 | 1 | 1 | 0 | 1 | 1 |
| 101 | 1 | 1 | 0 | 1 | 0 | 0 |
| 110 | 1 | 1 | 0 | 1 | 1 | 1 |
| 111 | 1 | 0 | 1 | 0 | 0 | 0 |

| $x_n$ | $D_{2n}$ | | | | $D_{1n}$ | | | | $D_{0n}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(Q_2Q_1Q_0)_n$ | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
| 00 | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **0** |
| 01 | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **0** |
| 11 | **1** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **0** |
| 10 | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **0** |

Ecuațiile rezultante sunt date mai jos:

$$D_2 = \overline{Q_2}\cdot\overline{Q_1}\cdot\overline{Q_0}\cdot\overline{i\_d} + Q_2\cdot Q_1\cdot\overline{i\_d} + Q_2\cdot\overline{Q_0}\cdot i\_d + Q_2\cdot\overline{Q_1}\cdot Q_0 + Q_2\cdot Q_0\cdot\overline{i\_d} + \overline{Q_2}\cdot Q_1\cdot Q_0\cdot i\_d$$

$$D_1 = \overline{Q_1}\cdot\overline{Q_0}\cdot\overline{i\_d} + Q_1\cdot\overline{Q_0}\cdot i\_d + \overline{Q_1}\cdot Q_0\cdot i\_d + Q_1\cdot Q_0\cdot i\_d$$

$$D_0 = \overline{Q_0}$$

Implemetarea programului ce va genera numărătorul folosindu-ne de diagrama de stare este:

**MODULE** Numarat

TITLE 'Numarator pe 3 biti cu pin de selectare a sensului'

**declarations**

@alternate

clk **pin** 1;
i_d **pin** 9;
/OE **pin** 11;

```
Q2, Q1, Q0 pin 19, 18, 17 istype 'reg';

equations

Q2 := /Q2&/Q1&/Q0&/i_d + Q2&Q1&/i_d + Q2&/Q0&i_d + Q2&/Q1&Q0 + Q2&Q0&/i_d
+ /Q2&Q1&Q0&i_d;
Q1 := /Q1&/Q0&/i_d + Q1&/Q0&i_d + /Q1&Q0&i_d + Q1&Q0&/i_d;
Q0 := /Q0;

[Q2, Q1, Q0].clk = clk;
[Q2, Q1, Q0].oe  = OE;

END
```

## Realizarea unui numărător prin utilizarea facilităților oferite de limbajul ABEL

**Temă:** Să se construiască un numărător universal binar pe patru biți cu posibilitatea: încărcării paralel a datelor, selectarea sensului de numărare cât şi a posibilității de validare a acestuia.

```
MODULE Numar

TITLE 'Numarator universal pe 4 biti'

 declarations

 X, C, Z = .x., .c., .z.;

"declararea pinilor de intrare
 d3, d2, d1, d0 pin 2, 3, 4, 5;
 clk     pin 1;
 cnten   pin 6;
 ld      pin 7;
 i_d     pin 8;
/OE  pin 11;


"declararea pinilor de ieşire
 q3, q2, q1, q0 pin 19, 18, 17, 16 istype 'reg';

"grupare pini
 data     = [d3..d0];
 contor   = [q3..q0];

"cuvinte de control
                MOD  = [cnten, ld, i_d];
LOAD   =   (MOD == [  X,   1,  X]);
STOP   =   (MOD == [  0,   0,  X]);
INC    =   (MOD == [  1,   0,   1]);
DEC    =   (MOD == [  1,   0,   0]);

equations

          when   LOAD  then contor := data;
  else    when    INC   then contor := contor + 1;
  else    when    DEC   then contor := contor - 1;
  else    when    STOP  then contor := contor;

  contor.clk = clk;
  contor.oe  = OE;


END
```