TEMA NR.15
SINTEZA CIRCUITELOR SECVENŢIALE.

1.1 Noţiuni de bază despre sinteza circuitelor secvenţiale.

1.2 Bistabilele.

Circuitele logice secvenţiale (CLS) se caracterizează prin faptul că în orice moment de timp vectorul de ieşire a circuitului depinde nu numai de semnalele de la intrare din acel moment (ignorînd timpul de propagare) ci şi de semnalele de la intrare aplicate în momentele de timp anterioare. Această dependenţă se datorează prezenţei în CLS a unor elemente de memorie, care reprezintă mai multe stări logice stabile, comandate prin intrări şi participă la formarea semnalelor de ieşire. *Rezultă că un CLS conţine o schemă combinaţională completată cu o structură de memorare.*

Sinteza unui CLS se efectuează în următoarele etape:

- descrierea necesităţilor ce trebuie să realizeze circuitul respectiv (prin text, desen, diagrame etc.);
- reprezentarea acestei descrieri sub forma unui tabel de tranziţie;
- deducerea funcţiilor logice şi minimizarea acestora;
- implementarea acestor funcţii minimizate sub forma unor reţele de comutare prin intermediul circuitelor integrate;

Elementul esenţial în etapa de sinteză a circuitelor logice secvenţiale constă în faptul că se va elabora de la început aşa numitul tabel de tranziţie care se deosebeşte de tabelul de adevăr prin faptul că-n acest tabel trebuie înscrisă starea momentului secvenţial în momentul curent *t*, iar în calitate de funcţie se ia starea circuitului secvenţial în momentul *t+1*. În rest sinteza are loc după aceleaşi principii ca şi sinteza circuitelor combinaţionale. Starea curentă a circuitului secvenţial se utilizează în calitate de variabilă, iar starea ulterioară în calitate de funcţie.

În calitate de valori a funcţiilor logice, care trebuie deduse şi minimizate, se iau valorile funcţiilor de ieşire la momentul ***t+1***, iar în calitate de variabile a acestor funcţii se iau valorile variabilelor de intrare la momentul ***t*** şi valorile funcţiilor de ieşire la momentul ***t***, adică toate datele care asigură tranziţia circuitului respectiv de la starea care este în momentul ***t*** la starea care trebuie să obțină în momentul ***t+1***.

2. Circuitele basculante bistabile

Circuitele basculante bistabile (CBB) sunt circuite elementare secvenţiale ce se caracterizează prin faptul că tot timpul prezintă una din două posibile stări stabile. Aceste două stări codificabile prin cele două valori binare (0 sau 1) se utilizează de regulă în calculatoarele numerice ca suport pentru memorarea unui bit.

Există mai multe tipuri de circuite basculante bistabile. Primul ca ordine a apariţiei şi cel mai simplu este bistabilul de tip RS. Funcţionarea acestuia este descrisă de următorul tabelul de tranziţie.

În acest tabel *S* (Set) este intrarea de scriere, *R* (Reset) este intrarea de ştergere, Q_t este starea bistabilului în momentul *t*, iar Q_{t+1} este starea bistabilului în momentul *t+1*, care rezultă în urma aplicării valorilor semnalelor respective la intrare şi stării bistabilului în momentul *t*. Combinaţia *S* =*R*=1 este interzisă deoarece în urma aplicării ei starea bistabilului va fi incertă din cauza asimetriei reale a elementelor logice folosite la implementarea bistabilului.

S	R	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	*
1	1	1	*

Din tabelul uşor se poate deduce că:

$$Q_{t+1} = \overline{R}(S \vee Q_t)$$

$$RS = 0$$

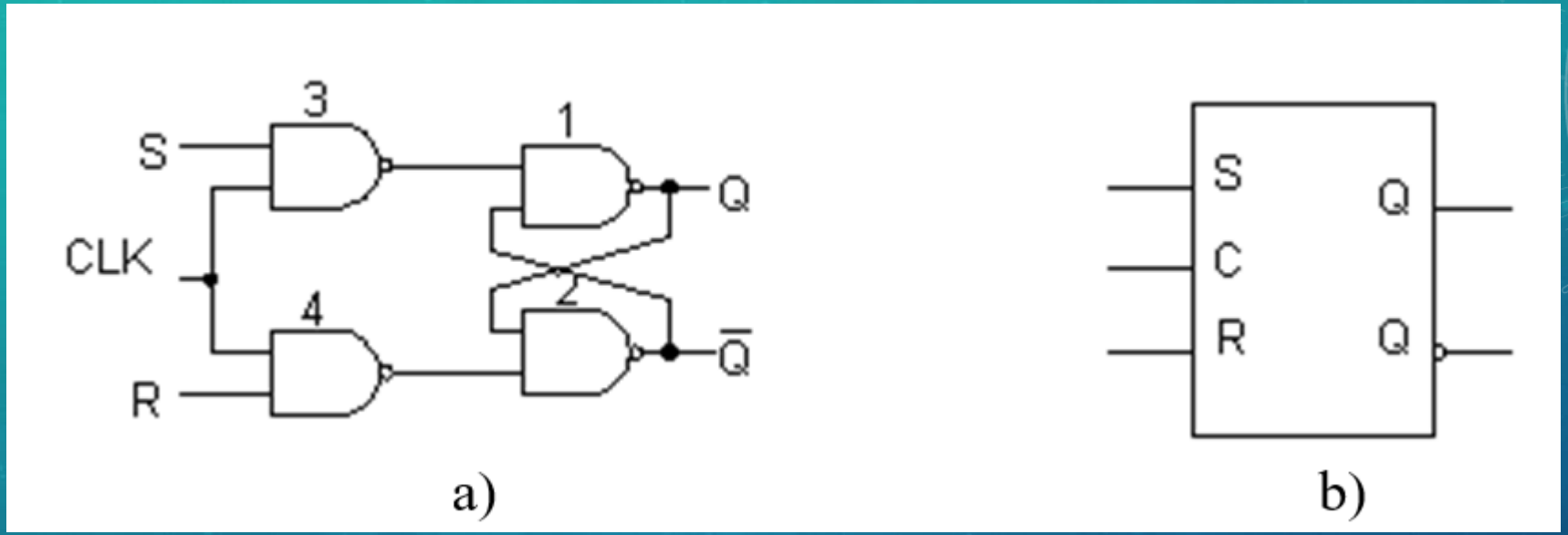
Cu ajutorul formulelor de Morgan expresia poate fi uşor transformată în una din următoarele două forme:

$$Q_{t+1} = S \vee \overline{R \vee \overline{Q}_t}$$

$$Q_{t+1} = \overline{\overline{S} \wedge \overline{\overline{R} Q_t}}$$

Expresiile date pot fi implementate cu ajutorul a două elemente logice SAU-NU ori cu ajutorul a două elemente logice ŞI-NU şi în aşa fel se obţine schema bistabilelor RS asincrone.

În foarte multe aplicaţii este utilizat bistabilul *RS* sincron.



Circuitul bistabil RS sincron: a)–schema logică; b)-simbolul.

Elementele logice 1 şi 2 formează bistabilul propriu-zis, iar elementele 3 şi 4 realizează sincronizarea cu un semnal de ceas, *CLK*. Acest semnal activează intrările *S* şi *R* numai cînd se află în starea logică 1, inhibîndu-le pe durata cît se află în stare logică 0.

Pornind de la bistabilul *RS*, se obţine bistabilul de tip *D* *în cazul cînd se stabileşte condiţia:* $S = \overline{R}$

În acest caz semnalul de la ieşire va fi identic cu cel de la intrare.

Funcţionarea bistabilului de tip *D* este descrisă de relaţia $Q_{t+1} = D$

în care *D* – intrarea de date. Această ultimă relaţie reflectă capacitatea bistabilului de tip *D* de a păstra direct o informaţie, aplicaţia principală a circuitului fiind memorarea cuvintelor binare. Cea mai simplă schemă de circuit bistabil *D* sincron:

Circuitul bistabil D sincron: a - schema; b – simbolul.

Faptul că informaţia de la intrare este transmisă la ieşire şi apoi intrarea este inactivată a condus la denumirea de latch (lacăt) atribuită bistabilelor de tip *D*.

Existenţa restricţiei S=R=1 constituie o dificultate pentru proiectanţi, de aceea foarte des se recurge la utilizarea circuitului bistabil de tip *JK*.

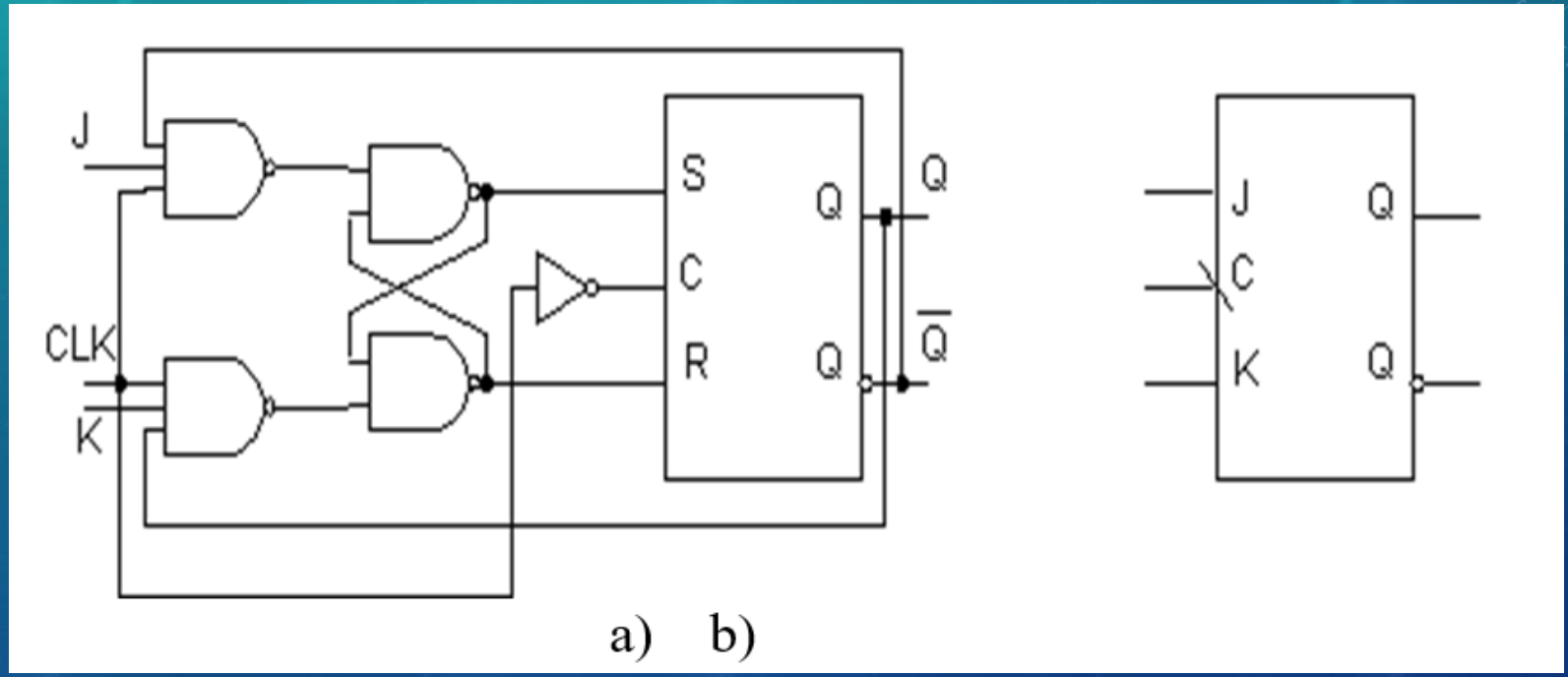
Bistabilul *JK* îşi păstrează funcţionalitatea şi în cazul cînd *R*=*S*=1. Această combinaţie a semnalelor de la intrare se foloseşte pentru a inversa starea bistabilului. Funcţionarea bistabilului *JK* este prezentată în tabelul de tranziţie, iar funcţia logică ce rezultă din acest tabel este reflectată de expresia logică

Tabelul de tranziţie al bistabilului *JK*

J	*K*	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$Q_{t+1} = J\overline{Q}_t \vee \overline{K}Q_t$$

Din tabelul se observă că intrările *J* şi *K* pot fi tratate ca intrări *S* şi respectiv *R*. Însă în cazul, cînd *J*=*K*=1, bistabilul îşi schimbă starea anterioară (0→1, respectiv 1→0). Bistabilul *JK* se realizează de obicei în baza bistabilului *RS* într-o formă cunoscută sub numele de master-slave (stăpîn-sclav). Circuitul *JK* de tip master-slave conţine două bistabile *RS*.



Circuitul bistabil JK sincron: a) – schema; b) – simbolul.

Analizînd modul de funcţionare al circuitului se poate constata că informaţia de la intrare este transferată la ieşire pe frontul negativ al semnalului de ceas. Asemenea circuite sunt denumite în literatura de specialitate ca bistabile flip-flop.

Bistabilul *JK* este universal, avînd aplicaţii multiple:

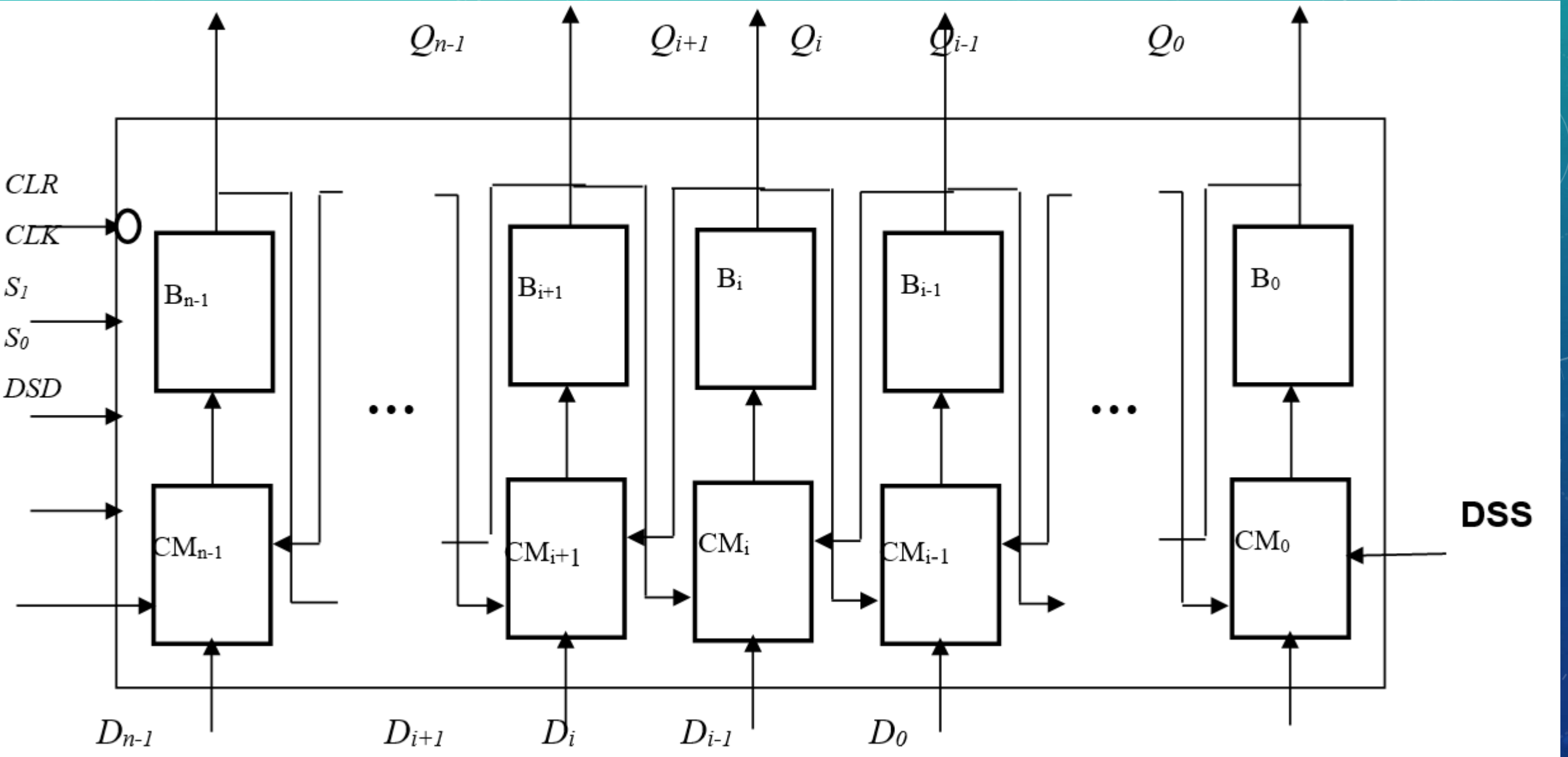
- După cum s-a menţionat mai sus, el poate funcţiona în regim de bistabil *RS*.
- Dacă intrările *J* şi *K* sunt conectate între ele printr-un inversor, circuitul devine bistabil *D*, cu intrarea *D* pe linia *J*.
- Dacă *J*=*K*=1, atunci bistabilul *JK* devine bistabil de tip *T*, care basculează la fiecare impuls de ceas.

TEMA NR.16 : SINTEZA REGISTRELOR

Registrele se includ în categoria elementelor funcţionale secvenţiale şi sunt destinate memorării şi procesării cuvintelor binare.

Componenta de bază a oricărui registru sunt bistabilele. Structura generală a unui registru este constituită din ***n*** bistabile (în cazul de faţă de tip *D*), avînd un semnal de ceas CLK comun pentru toate bistabilele (B). Intrarea de ştergere $\overline{CLR}$ activă pe zero logic, este prezentă la majoritatea registrelor şi permite resetarea celor ***n*** bistabile. Intrările S_1 şi S_0 comandă cele ***n*** comutatoare logice (CM), asigurînd astfel selecţia regimului de lucru al registrului. Comutatoarele, în dependenţă de codul de selecţie, por asigura conectarea intrărilor bistabilelor în trei moduri: la ieşirea B_i din stînga, din dreapta sau la intrarea de date *D*.

În dependenţă de conectarea intrărilor bistabilelor, registrul poate încărca un cuvînt binar în cod paralel sau succesiv.



Structura generală a unui registru

În regimul de încărcare paralel, cuvîntul pentru înscriere se aplică la intrările de date D_{n-1}, …, D_0 – portul de intrare. Cuvîntul înscris este accesibil la ieşirile Q_{n-1}, …, Q_0 – portul de ieşire.

Încărcarea datelor în registru se realizează la aplicarea semnalului de ceas.

În regimul succesiv de încărcare a datelor cuvîntul binar poate fi deplasat spre dreapta sau spre stînga.

Pentru înscrierea succesivă se folosesc două intrări de date: spre dreapta *DSD* şi respectiv spre stînga *DSS*.

Notă: Registrul capabil să deplaseze datele atît la stînga, cît şi la dreapta se numeşte registru cu deplasare bidirecţională.

Acest registru poate fi utilizat nu numai pentru memorarea unui cuvînt binar, ci şi pentru procesarea lui, deoarece deplasarea spre dreapta cu ***i*** poziţii este echivalentă cu operaţia de împărţire a cuvîntului la 2^i iar deplasarea spre stînga – cu operaţia de inmulţire cu 2^i .

Cuvîntul de ***n*** biţi înscris pe intervalul de ***n*** tacte prin intrarea DSD, respectiv DSS este pierdut secvenţial bit cu bit la ieşirea Q_0, respectiv Q_{n-1} pe următorul interval de ***n*** tacte. Dacă însă se conectează ieşirea Q_0 la DSD, respectiv Q_{n-1} la *DSS* se obţine structura de registru în inel sau registru cu deplasare ciclică.
Într-un asemenea registru cuvîntul înscris iniţial este recirculat în interiorul registrului.

Tabelul de funcţionare al registrului de patru biţi

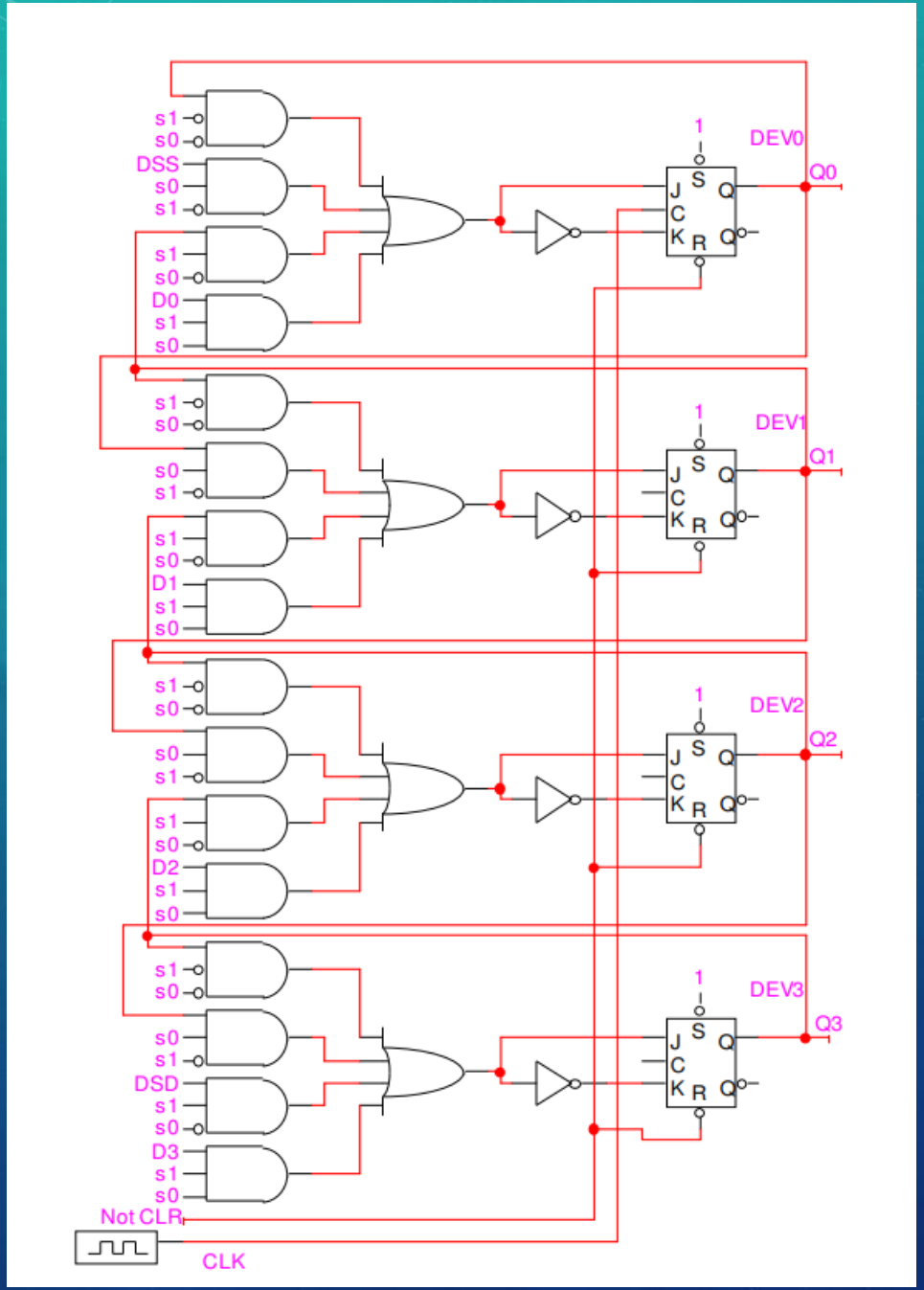
Regim de lucru	Intrări					Ieşiri				
	s_1	s_0	D_i	*DSD*	*DSS*	Q_3	Q_2	Q_1	Q_0	*t*
Păstrare	0	0	*	*	*	Q_3	Q_2	Q_1	Q_0	***t*** **+** ***1***
Deplasare stînga	*0*	1	*	*	*DSS*	Q_2	Q_1	Q_0	*DSS*	
Deplasare dreapta	1	*0*	*	*DSD*	*	*DSD*	Q_3	Q_2	Q_1	
Încărcare paralelă	1	1	D_i	*	*	D_3	D_2	D_1	D_0	

$$J_3 = \overline{K}_3 = \bar{s}_1\bar{s}_0 Q_3 \vee \bar{s}_1 s_0 Q_2 \vee s_1\bar{s}_0 DSD \vee s_1 s_0 D_3,$$

$$J_2 = \overline{K}_2 = \bar{s}_1\bar{s}_0 Q_2 \vee \bar{s}_1 s_0 Q_1 \vee s_1\bar{s}_0 Q_3 \vee s_1 s_0 D_2,$$

$$J_1 = \overline{K}_1 = \bar{s}_1\bar{s}_0 Q_1 \vee \bar{s}_1 s_0 Q_0 \vee s_1\bar{s}_0 Q_2 \vee s_1 s_0 D_1,$$

$$J_0 = \overline{K}_0 = \bar{s}_1\bar{s}_0 Q_0 \vee \bar{s}_1 s_0 DSS \vee s_1\bar{s}_0 Q_1 \vee s_1 s_0 D_0.$$



Registru cu deplasare bidirecţională

2. Sinteza registrului de deplasare

RG deplasarea stânga, ciclică, 4 biți, bistabile JK

	t				t+1				t+1							
	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	0	0	0	*	0	*	0	*	0	*
1	0	0	0	1	0	0	1	0	0	*	0	*	1	*	*	1
2	0	0	1	0	0	1	0	0	0	*	1	*	*	1	0	*
3	0	0	1	1	0	1	1	0	0	*	1	*	*	0	*	1
4	0	1	0	0	1	0	0	0	1	*	*	1	0	*	0	*
5	0	1	0	1	1	0	1	0	1	*	*	1	1	*	*	1
6	0	1	1	0	1	1	0	0	1	*	*	0	*	1	0	*
7	0	1	1	1	1	1	1	0	1	*	*	0	*	0	*	1
8	1	0	0	0	0	0	0	1	*	1	0	*	0	*	1	*
9	1	0	0	1	0	0	1	1	*	1	0	*	1	*	*	0
10	1	0	1	0	0	1	0	1	*	1	1	*	*	1	1	*
11	1	0	1	1	0	1	1	1	*	1	1	*	*	0	*	0
12	1	1	0	0	1	0	0	1	*	0	*	1	0	*	1	*
13	1	1	0	1	1	0	1	1	*	0	*	1	1	*	*	0
14	1	1	1	0	1	1	0	1	*	0	*	0	*	1	1	*
15	1	1	1	1	1	1	1	1	*	0	*	0	*	0	*	0

J3

Q3Q2 \ Q1Q0	00	01	11	10
00		1	*	*
01		1	*	*
11		1	*	*
10		1	*	*

$J3 = Q2$

K3

Q3Q2 \ Q1Q0	00	01	11	10
00	*	*		1
01	*	*		1
11	*	*		1
10	*	*		1

$K3=\overline{Q2}$

J2

Q3Q2 \ Q1Q0	00	01	11	10
00		*	*	
01		*	*	
11	1	*	*	1
10	1	*	*	1

K2

Q3Q2 \ Q1Q0	00	01	11	10
00	*	1	1	*
01	*	1	1	*
11	*			*
10	*			*

$K2=\overline{Q1}$

J1

Q3Q2 \ Q1Q0	00	01	11	10
00				
01	1	1	1	1
11	*	*	*	*
10	*	*	*	*

$J1=Q0$

K1

Q3Q2 \ Q1Q0	00	01	11	10
00	*	*	*	*
01	*	*	*	*
11				
10	1	1	1	1

$K1=\overline{Q0}$

J0

Q3Q2 \ Q1Q0	00	01	11	10
00			1	1
01	*	*	*	*
11	*	*	*	*
10			1	1

$J0=Q3$

K0

Q3Q2 \ Q1Q0	00	01	11	10
00	*	*	*	*
01	1	1		
11	1	1		
10	*	*	*	*

$K0=\overline{Q3}$

$J3 = Q2$

$K3=\overline{Q2}$

J2=Q1

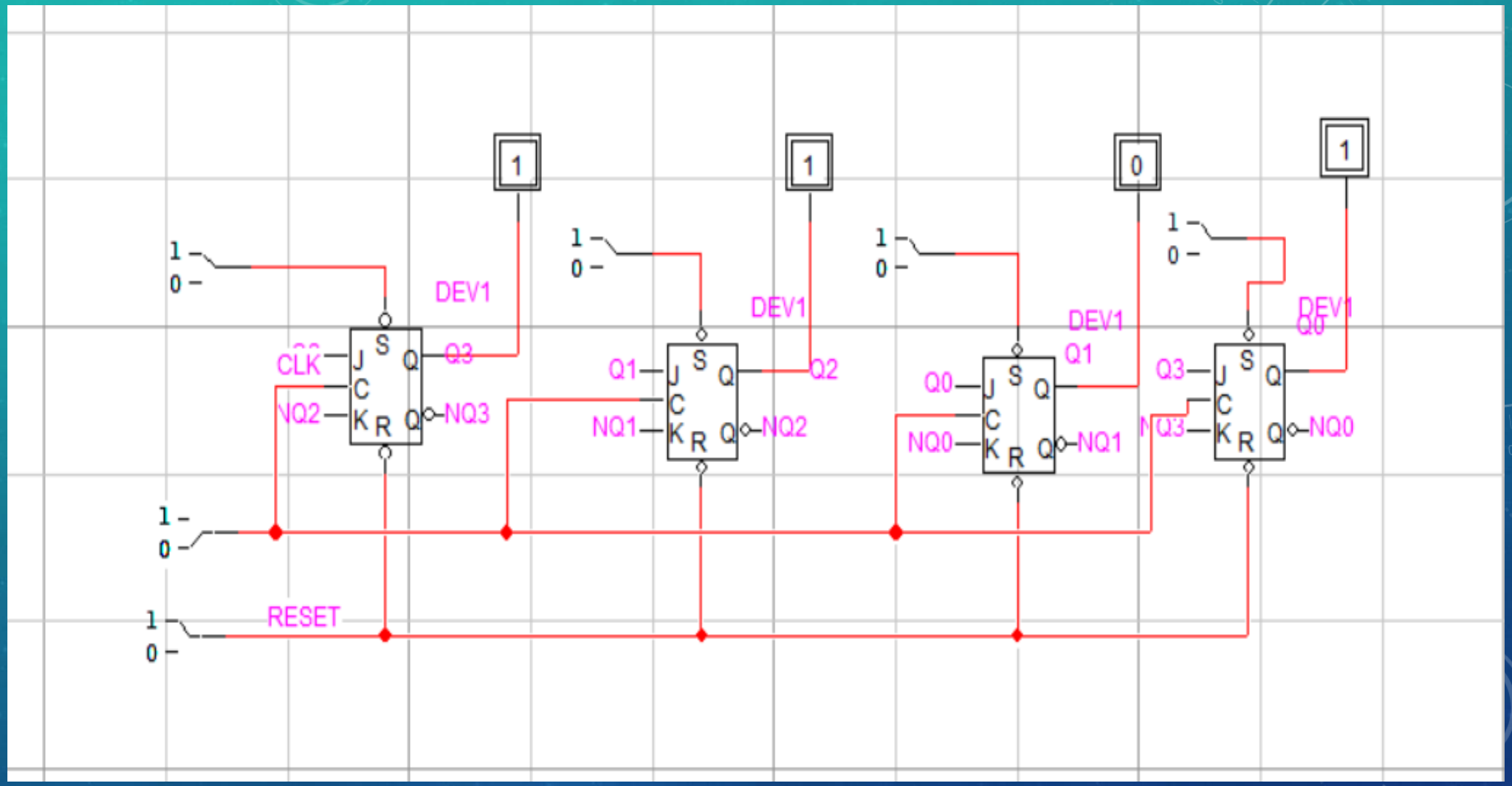
$K2=\overline{Q1}$

J1=Q0

$K1=\overline{Q0}$

J0=Q3

$K0=\overline{Q3}$



Circuitul implementat în Logik Works al RG de deplasare, stânga, ciclică, 4 biți în baza bistabile JK

Vă mulțumesc pentru atenție!


ÎNTREBĂRI