

PROIECTAREA SISTEMELOR NUMERICE

UNITATE DE EXECUȚIE CABLATĂ

Dispozitiv de împărțire binară. Unitate de execuție UE cablată.

Sinteza unității de execuție UE a unui sistem numeric se bazează pe folosirea unui limbaj de descriere. Ea va conduce la realizarea cablată a unității de execuție care rezultă din interconectarea componentelor combinaționale și secvențiale disponibile sub formă de circuite integrate MSI. Se folosesc în principal multiplexoare și unități aritmetice și logice pentru componentele combinaționale și numărătoare și registre pentru componentele secvențiale.

Se definește o metodă de sinteză pentru UE a sistemelor numerice și apoi se aplică etapele acestei metode la sinteza UE a unui sistem numeric care realizează împărțirea a 2 numere binare.

1) Metoda

Sinteza UE a unui sistem numeric care realizează un algoritm dat se efectuează în 3 etape:

1. Declararea regiștrilor și a resurselor UE, urmată de descrierea funcțională a sistemului numeric, cu ajutorul unei organigrame în care limbajul de descriere se aplică regiștrilor și resurselor UE;
2. Construirea schemei UE și declararea eventualilor regiștri și a resurselor adiționale;
3. Realizarea UE cu ajutorul componentelor combinaționale și secvențiale MSI disponibile.

Exemplificarea celor 3 etape ale metodei se face pentru sinteza UE a unui sistem numeric care împarte numărul binar X la numărul binar Y. Pentru numere întregi pozitive de 8 biți, câtul Q și restul R ale acestei împărțiri se obțin (în principiu) conform algoritmului din figura 1.

2) Declararea și descrierea funcțională

Alegerea numărului, a dimensiunii și a repartizării regiștrilor și resurselor UE se face intuitiv, pe baza algoritmului care trebuie executat. Descrierea și secvențierea transferurilor care au loc între regiștrii aleși decurge tot din algoritm. Ele determină organigrama sistemului numeric de împărțire a 2 numere binare pe 8 biți.

Legenda figurii 1:

: = atribuire

= test egalitate

◇ test

□ operație

Exemplu: $X = (F0)_H$
 $Y = (08)_H$
 $\frac{X}{Y} \Rightarrow Q = (30)_H$
 $R = 0$

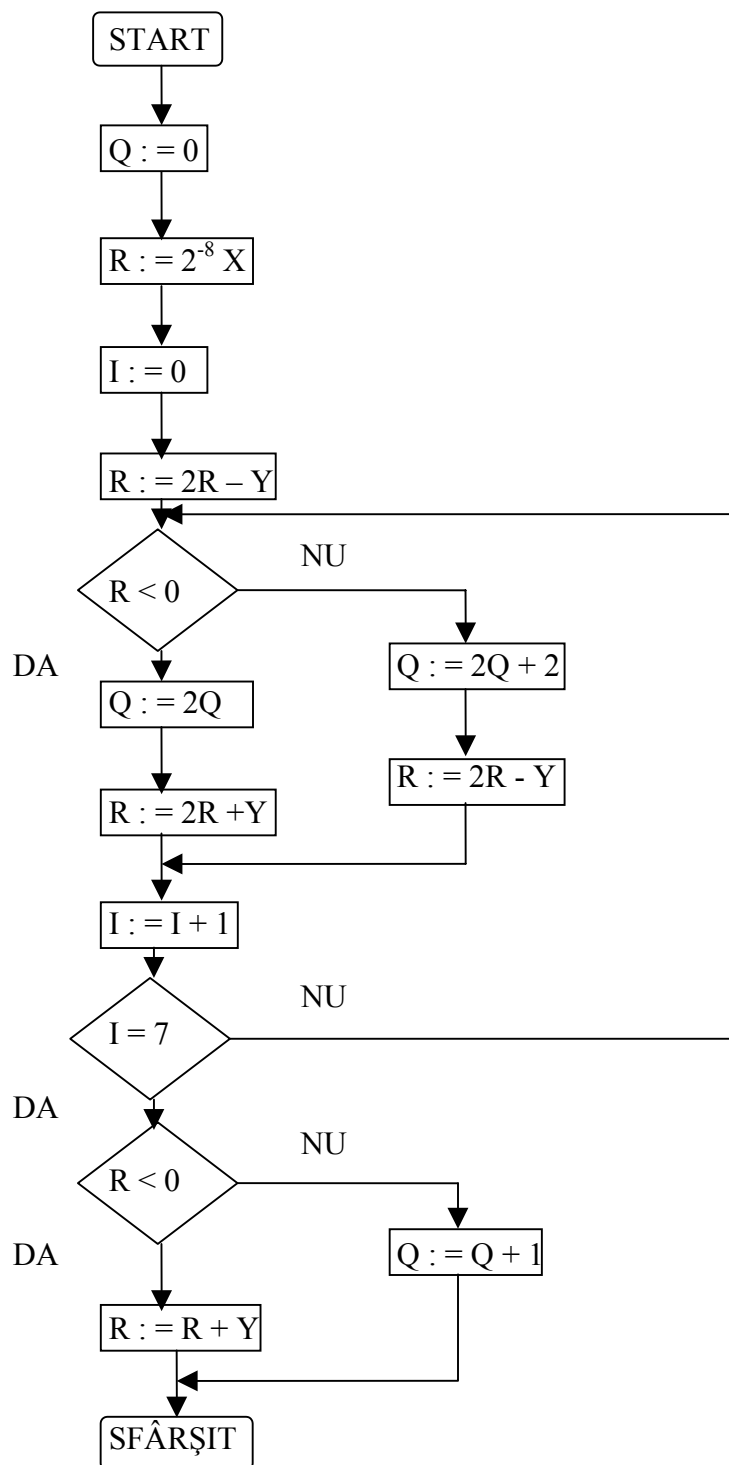


Figura 1. Algoritmul de împărțire a 2 numere binare pe 8 biți

Un sistem numeric capabil să realizeze acest algoritm de împărțire, atunci când se dă comanda de execuție E ($E = 1$), necesită regiștrii următori – vezi figura 2.

Legenda figurii 2:

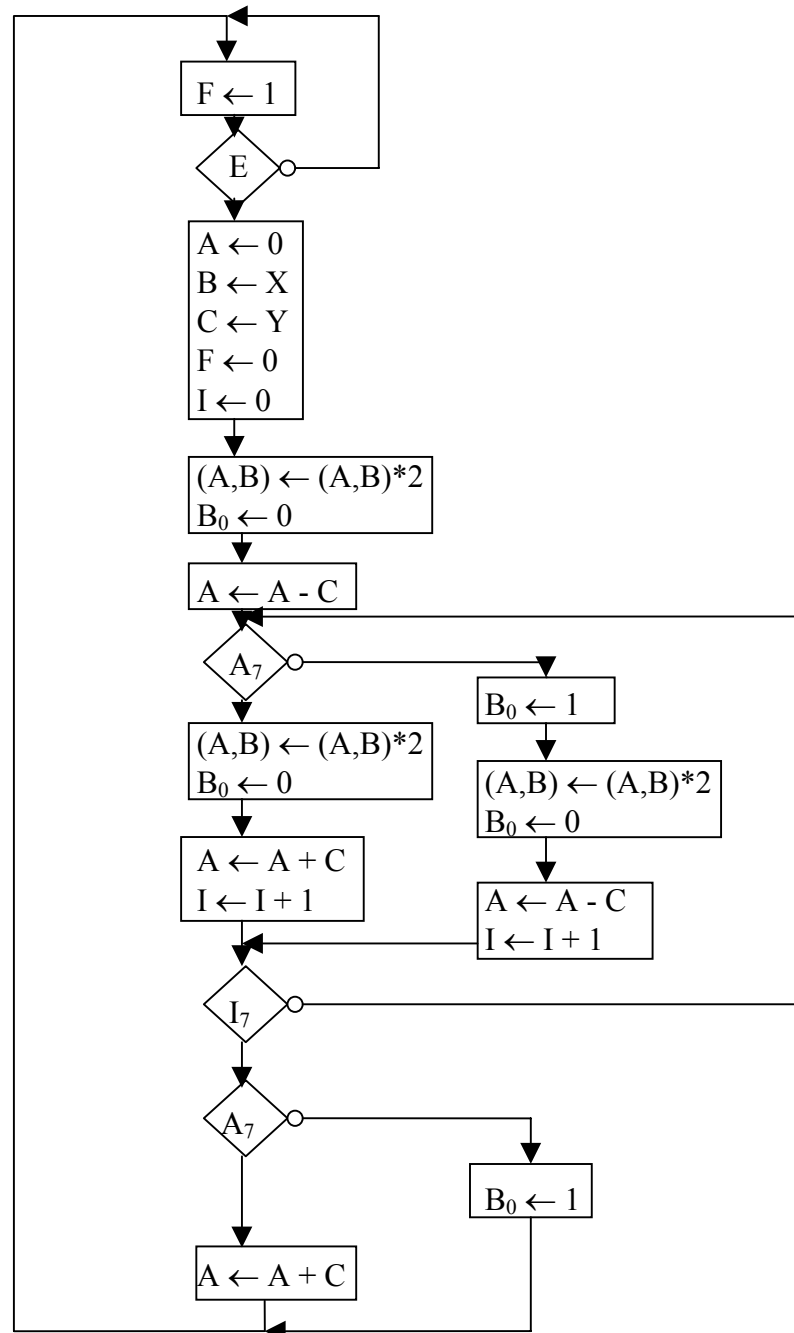
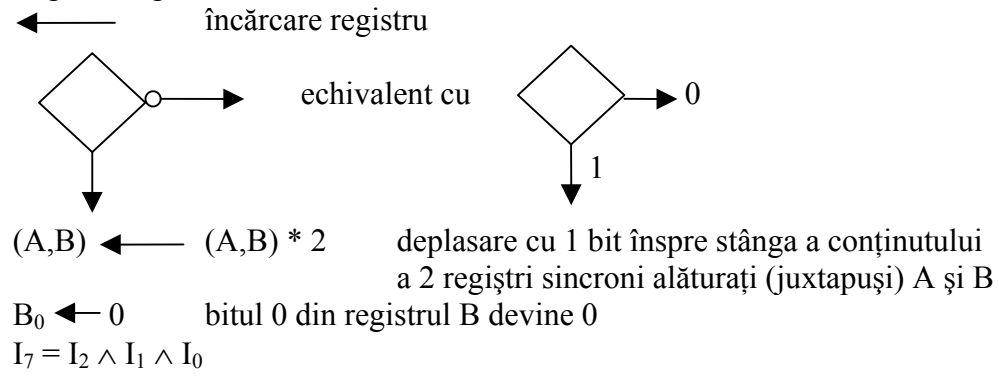


Figura 2. Organigrama

Funcționarea sistemului numeric este descrisă prin organigramă.

S-au folosit:

$A_{1 \times 8}$ = acumulator pentru rezultate intermediare și pentru restul R al împărțirii (8 biți)

$B_{1 \times 8}$ = registru pentru deîmpărțitul X și pentru rezultatul final Q (8 biți)

$C_{1 \times 8}$ = registru pentru împărțitorul Y (8 biți)

$F_{1 \times 1}$ = indicator de sfârșit al împărțirii (1 bit – flag)

$I_{1 \times 3}$ = registru index (3 biți)

3) Schema și declarația adițională

Schema UE interconectează resursele și regiștrii aleși astfel încât să asigure toate transferurile de informație descrise de organigramă. Această schemă poate eventual să conțină și regiștri și resurse adiționale.

Pentru exemplul dat schema UE face apel la o resursă suplimentară:

$AU_{1 \times 8}$ = unitate aritmetică pentru adunarea sau scăderea împărțitorului cu (din) rezultatul intermediar

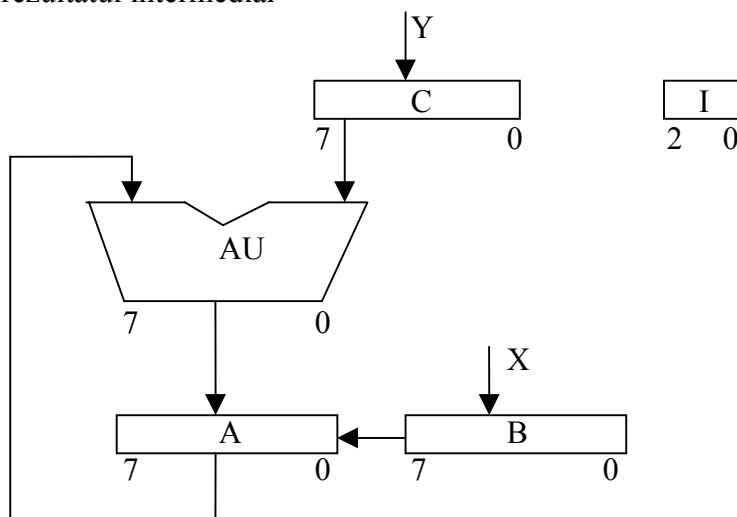


Figura 3. Schema UE

4) Realizarea schemei UE

Realizarea UE constă în alegerea circuitelor pentru resurse și regiștri pentru a se efectua operațiile din organigramă și a interconecta variabilele de informație conform schemei din figura 3.

Pentru exemplu dat:

AU

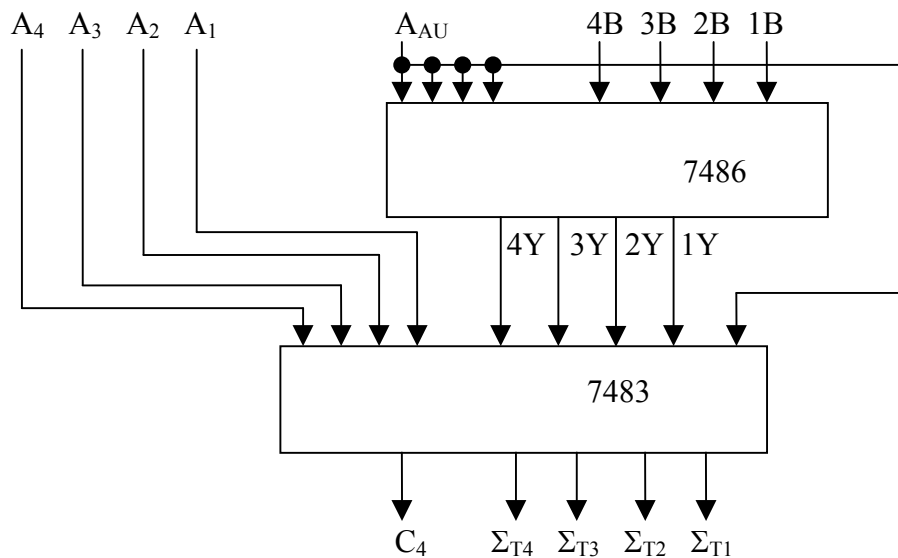
Operațiile efectuate de resursa AU se reduc la:

$$AU = A + C$$

$$AU = A - C$$

Se poate folosi un sumator-scăzător pe 8 biți. Vom utiliza 2 x (7483; 7486); (7483 este pe 4 biți).

Operație	Descriere	A_{AU}
ADD	$(C_4, \Sigma_{T4}, \Sigma_{T3}, \Sigma_{T2}, \Sigma_{T1}) = (A_4, A_3, A_2, A_1) + (B_4, B_3, B_2, B_1) + (0, 0, 0, 0)$	0
SUBTRACT	$(C_4, \Sigma_{T4}, \Sigma_{T3}, \Sigma_{T2}, \Sigma_{T1}) = (A_4, A_3, A_2, A_1) + (\overline{B_4}, \overline{B_3}, \overline{B_2}, \overline{B_1}) + (0, 0, 0, 1)$	1



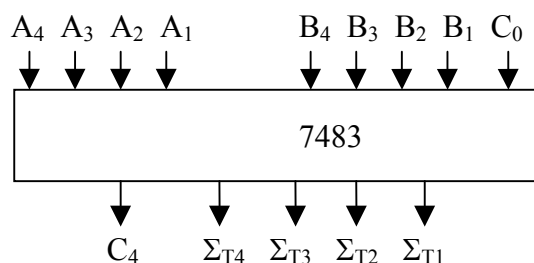
7486 SAU EXCLUSIV
cu 2 intrări

$$A \oplus 1 = \bar{A}$$

$$A \oplus 0 = A$$

4A	1	0	14	Vcc
4B	2	7486	13	2A
4Y	3		12	2B
3A	4		11	2Y
3B	5		10	1A
3Y	6		9	1B
GND	7		8	1Y

7483 SUMATOR pe 4 biți



Operație	Descriere
ADD	$(C_4, \Sigma_{T4}, \Sigma_{T3}, \Sigma_{T2}, \Sigma_{T1}) = (A_4, A_3, A_2, A_1) + (B_4, B_3, B_2, B_1) + (0, 0, 0, C_0)$

Registrul A

Operațiile efectuate sunt:

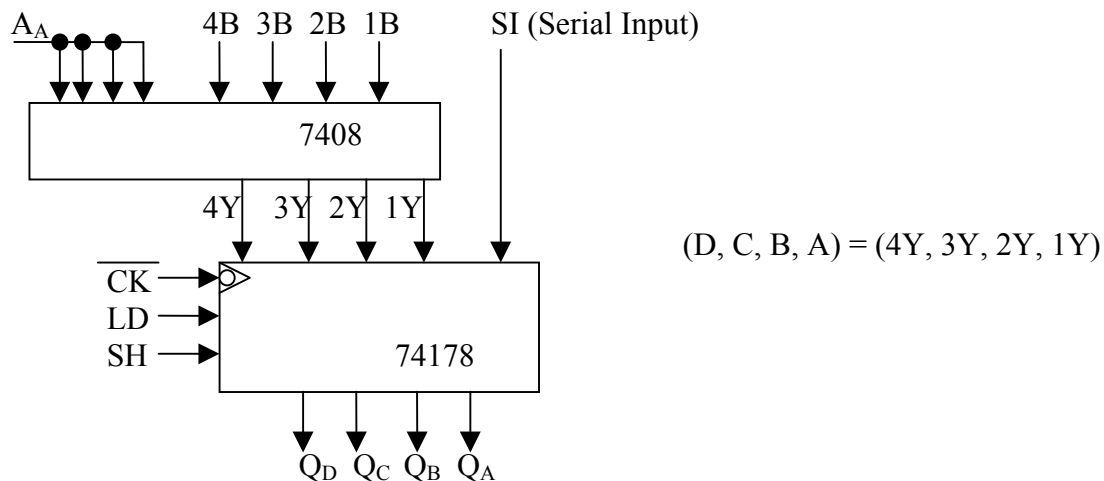
NOP (operație inefectivă)

$A \leftarrow 0$

$A \leftarrow A * 2, A_0 \leftarrow B_7$

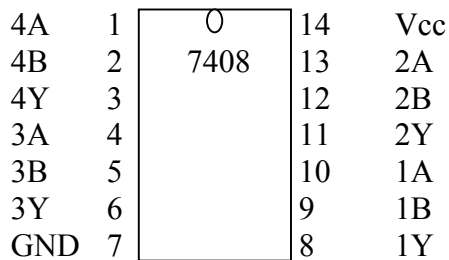
$A \leftarrow AU$

Se poate folosi un registru de deplasare pe 8 biți. Vom utiliza 2 x (7408; 74178); (74178 este pe 4 biți).

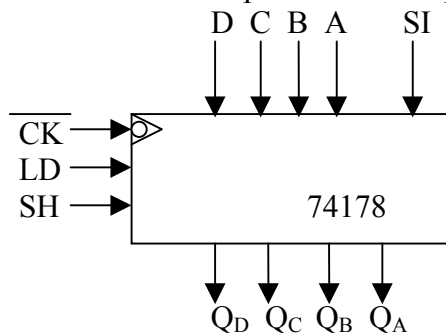


Operație	Descriere	SH	LD	A _A
HOLD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	0	0	\emptyset
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0, 0, 0, 0)$	0	1	0
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4B, 3B, 2B, 1B)$	0	1	1
SHIFT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, SI)$	1	\emptyset	\emptyset

7408 ȘI cu 2 intrări



74178 REGISTRU de deplasare sincron pe ceas. Funcționarea sa este evidentă.



Registrul B

Operațiile efectuate sunt:

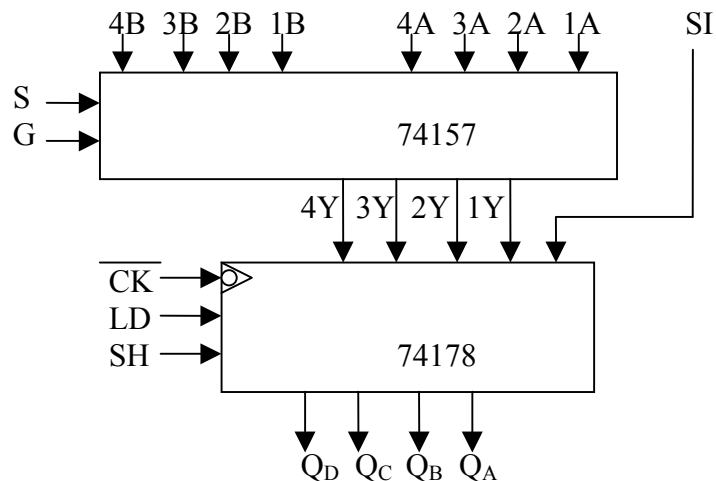
NOP (operație inefectivă)

$B \leftarrow X$

$B \leftarrow B*2, B_0 \leftarrow 0$

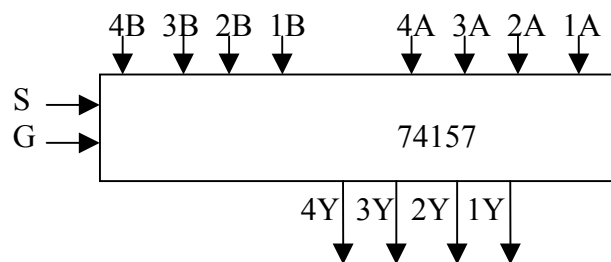
$B_0 \leftarrow 1$

Se poate folosi un registru de deplasare pe 8 biți. Se utilizează 2 x (72157; 74178); (74178 este pe 4 biți).



Operație	Descriere	SH	LD	G	S
HOLD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	0	0	\emptyset	\emptyset
LOAD A	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4A, 3A, 2A, 1A)$	0	1	0	0
LOAD B	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4B, 3B, 2B, 1B)$	0	1	0	1
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0,0,0,0)$	0	1	1	\emptyset
SHIFT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, SI)$	1	\emptyset	\emptyset	\emptyset

74157 MUX de tip 2:1 pe 4 biți



Operație	Descriere	G	S
SELECT A	$(4Y, 3Y, 2Y, 1Y) = (4A, 3A, 2A, 1A)$	0	0
SELECT B	$(4Y, 3Y, 2Y, 1Y) = (4B, 3B, 2B, 1B)$	0	1
ZERO	$(4Y, 3Y, 2Y, 1Y) = (0,0,0,0)$	1	\emptyset

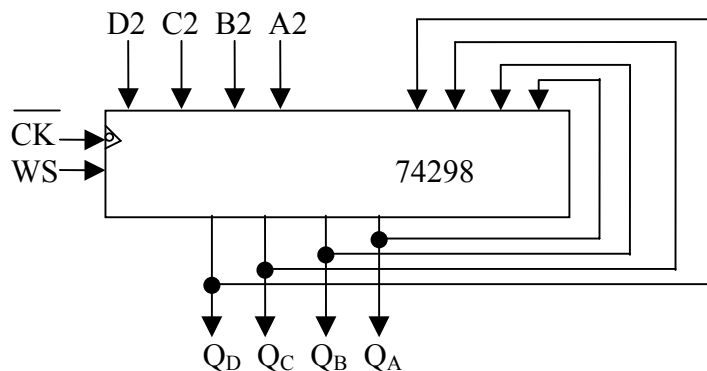
Registrul C

Operațiile efectuate sunt:

NOP

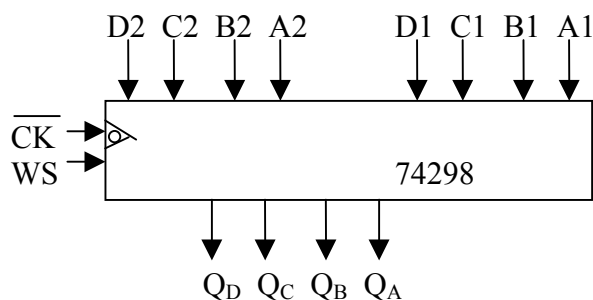
$C \leftarrow Y$

Se utilizează un registru paralel de 8 biți cu reacție 2 x (74298).



Operație	Descriere	WS
HOLD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	0
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D_2, C_2, B_2, A_2)$	1

74298 MUX 2:1 pe 4 biți cu memorare.



Operație	Descriere	WS
LOAD 1	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D_1, C_1, B_1, A_1)$	0
LOAD 2	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D_2, C_2, B_2, A_2)$	1

Registrul F

Operațiile efectuate sunt:

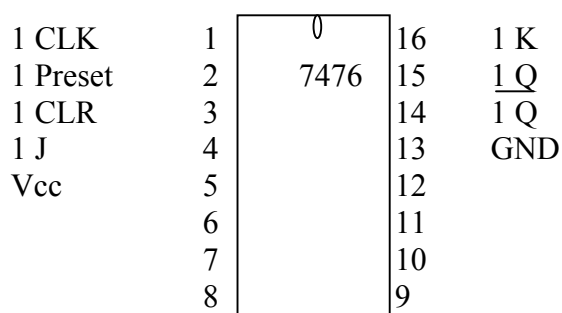
NOP

$F \leftarrow 1$

$F \leftarrow 0$

Se utilizează un bistabil de tip JK (7476/2).

7476 Bistabil JK sincron cu Set și Reset



Registrul I

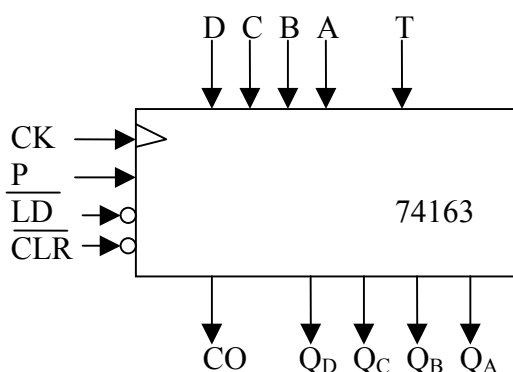
Operațiile efectuate sunt:

NOP

$I \leftarrow 0$

$I \leftarrow I + 1$

Se utilizează un numărător binar pe 4 biți 74163.



Operație	Descriere	CLR	LD	P
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0, 0, 0, 0)$ $CO = Q_D \cdot Q_C \cdot Q_B \cdot Q_A \cdot T$	0	\emptyset	\emptyset
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D, C, B, A)$ $CO = Q_D \cdot Q_C \cdot Q_B \cdot Q_A \cdot T$	1	0	\emptyset
HOLD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$ $CO = Q_D \cdot Q_C \cdot Q_B \cdot Q_A \cdot T$	1	1	0
COUNT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow [(Q_D, Q_C, Q_B, Q_A) + (0, 0, 0, T)]_{\text{mod } 16}$ $CO = Q_D \cdot Q_C \cdot Q_B \cdot Q_A \cdot T$	1	1	1

Concluzie

Alegerea resurselor și a regiștrilor s-a făcut *strict adaptat* la operațiile organigramei, nerecurgând la utilizarea unor componente universale (cum ar fi regiștri de deplasare bidirecționali 74194). Această alegere conduce la o realizare *posibilă* a UE.

UE este sincronă deoarece toate componentele primesc același semnal de tact.

TEMĂ – Să se deseneze schema globală (interconectarea tuturor componentelor) a unității de execuție UE pentru împărțirea a 2 numere binare pe 8 biți.