Implementarea unui dispozitiv de înmulţire a două numere întregi fără semn de 8 biţi

Asist.univ.drd. ing. Mihai Timiş
Universitatea Tehnică "Gh.Asachi" Iaşi

ABSTRACT. In this paper I purpose to implement a multiplication device with unsigned two 8 biti integer numbers using synchronous sequential automata. This device could be done clasic using only basculant bistabiles circuits (CBB D, T, R-S, J-K) or using printed secquentiator (PS) and Automata Execution Elements (AEE). The second option was choice because implementation is more simply and clearly. This paper is moore didactic and it is succesuffuly used on Digital Sistem lab.

1 Organigrama algoritm

În cele ce urmează este folosit un algoritm de multiplicare a două numere prin adunări repetate.

S-a dorit implementarea sistemului folosind numai registre numărătoare 74193 şi porţi logice aferente.

Fie $X=X_7X_6X_5X_4X_3X_2X_1X_0B$ şi $Y=Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0B$, deînmulţitul şi înmulţitorul.

Vom utiliza următorul algoritm:

STEP1: P:=0;
STEP2: U:=Y;
STEP3: DECU,INCP;
STEP4: Dacă U diferit de zero, se revine la STEP3;
STEP5: DECX;
STEP6: Dacă X diferit de zero, se revine la STEP2;
STEP7: STOP.

2 Interfața hardware

Automatul complex este realizat din două subsisteme care conlucrează împreună:

- Elemente de Execuție a Automatului (EEA)
- Secvențiatorul cablat (SC)

Elementele de Execuție ale Automatului sunt registre numărătoare 74193 care vor compune şi stoca rezultatul înmulțirii, fig.1.1, fig1.2.

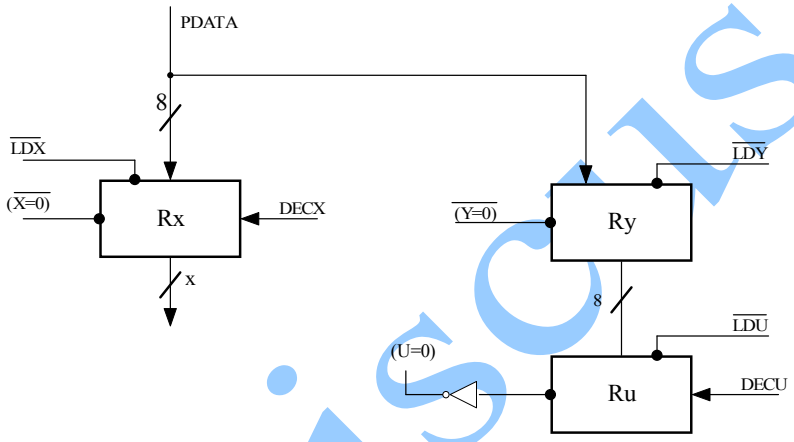

Fig. 1.1

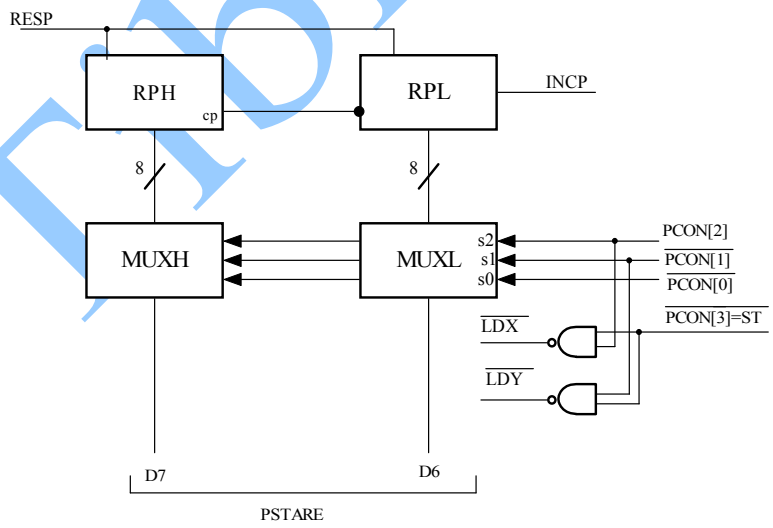

Fig. 1.2

După cum reiese din fig. 1.1, 1.2, registrele R_x, R_y sunt utilizate la încărcările operanzilor x,y, iar R_u – pentru memorarea temporală a operandului (y). Registrul RP, format din concatenarea a două registre notate cu RPH (memorează partea *High* a produsului) şi RPL (memorează partea *Low* a aceluiaşi produs).

Prin două multiplexoare notate MUXH,MUXL- sunt prelucrate bit cu bit părţile *High*, respectiv *Low*.

Bitul trei, (3) din portul de control reprezintă semnalul *Start,*(ST), iar $PCON_2, \overline{PCON_1}\,\overline{PCON_0}$ – semnalele de selecţie a biţilor i, i+8, din produs. Pentru *RDY*, nu vom considera hard suplimentar. Astfel, *RDY*=1, dacă x=0 sau y=0.

Deci *RDY*=(x=0)+(y=0)= $\overline{(x=0)} * \overline{(y=0)}$ (2)

3 Organigrama funcţională

Descrierea funcţionării automatului este realizată prin organigrama funcţională, construită pe baza algoritmului propus, (fig.3.1).

Deci atât timp cât ST=0, ($\overline{ST}=1$), automatul rămâne în starea zero (0).

După încărcarea operanzilor (x,y), prin programul final de multiplexare, se activează ST ($\overline{ST}=0$), deci automatul va trece în starea (1), în care se generează semnalul RESP, ce are ca efect anularea produsului parţial.

În starea (2) se testează dacă cel puţin unul din operanzi este nul. În cazul când [(x=0)+(y=0)]=0, deci ambii operanzi sunt diferiţi de zero se trece în starea (3), în care se realizează u:=y.

În starea (4), se decrementează (u) şi se incrementează (p), operaţie ce continuă până ce (u) devine null, când se trece în starea (6).

Se decrementează (x) şi se reia algoritmul cu secvenţa (2) până când (x=0)=1,(se anulează x).

În final se trece în starea (7), automatul revenind în starea (0), doar după dezactivarea semnalului ST($\overline{ST}=1$).

Partea de comandă se realizează sub forma unui secvenţiator cablat, fig. 3.2 şi 3.3.

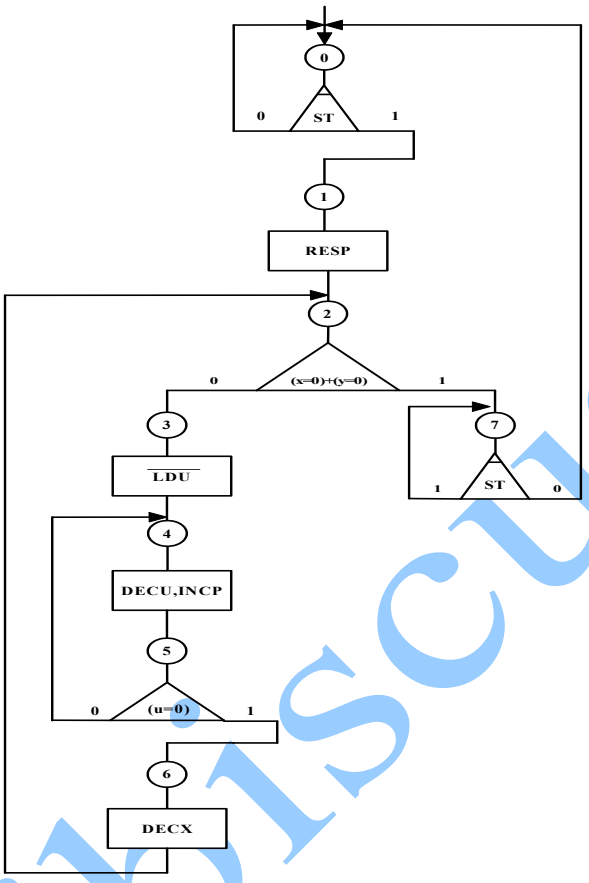

Fig. 3.1

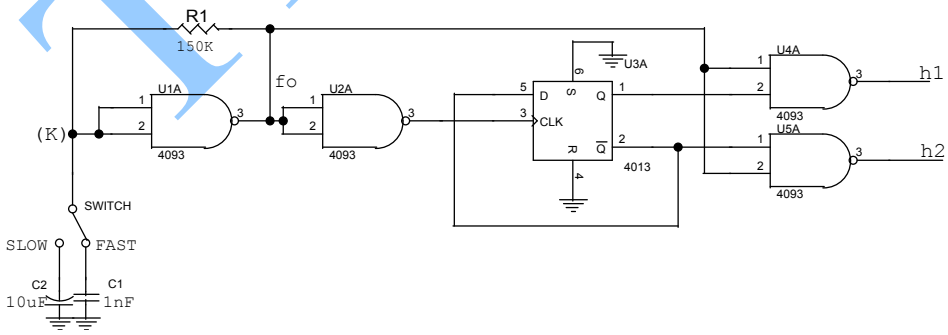

Fig.3.2.

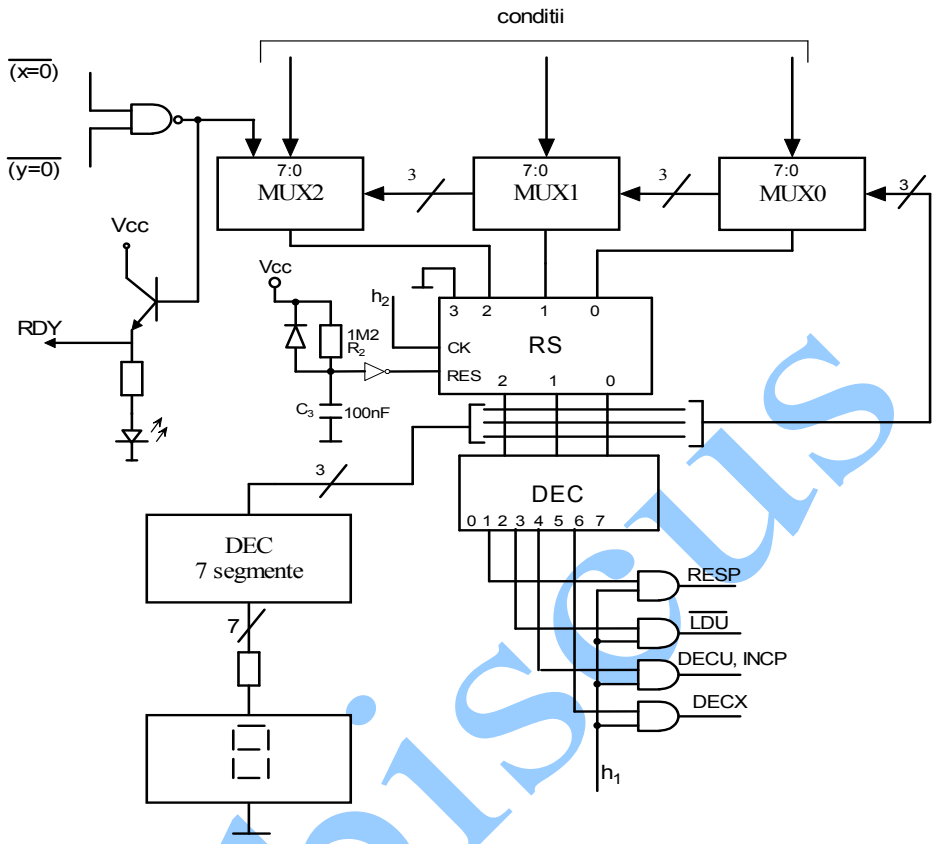

Fig.3.3

4 Calcul valori rezistențe, capacități

Resetare – calcul C_3,R_2.

Impunem $t_{ns} \geq 0.1$sec.

Notând cu V_P,V_N pragurile trigherului Schmitt inversor, rezultă:

$$t_{res}=C_3R_2\ln\frac{V_{cc}}{V_{cc}-V_P}$$, cu $V_P \cong 2.9V$ şi $C_3=100nF$ $V_N \cong 1.9V$

rezultă:

$$10^{-7}*R_2\ln\frac{5}{5-2.9} \geq 0.1\text{sec.}$$

sau

$$R_2 \geq \frac{10^{-1}}{0.868*10^{-7}}=1.15M \quad (1)$$

Se alege R2=1M2.

Oscilator

Prin (K) se stabileşte funcţia High (numai C_1) sau
Low ($C=C_1+C_2 \cong C_2$, cu $C_2>>C_1$).

Se foloseşte relaţia de calcul a perioadei semnalului (f_0):

$$T_0=2R_1C\ln\left[\left(\frac{V_P}{V_N}\right)\left(\frac{V_{cc}-V_N}{V_{cc}-V_P}\right)\right] \quad (2)$$

Pentru $f_0 \geq 2.5$kHz rezultă C=1nF, $R_1 \geq 140$K.

Se alege R_1=150K

Pentru C_1=10μF, rezultă $T_0 \cong 4$sec, ceea ce permite o vizualizare directă a procesării.

Relaţiile ce definesc semnalele de comandă sunt ilustrate mai jos:

$$\begin{aligned} RESP &= s_1 * h_1 \\ \overline{LDU} &= \overline{s_3 * h_1} \\ DECU &= INCP = s_4 * h_1 \\ DECX &= s_6 * h_1 \end{aligned} \quad (3)$$

Multiplexoarele MUX_i, cu i=0,1,2 sunt de tipul MMC4051.

Conform organigramei funcţionale, se construieşte tabelul de tranziţie, ca în fig.4.1.

cond. / st. curentă	ST	(u=0)	(x=0)+(y=0)	secv. urm.	cod secv. urm.	cod generic
s_0	0	-	-	s_0	000	00ST
	1	-	-	s_1	001	
s_1	-	-	-	s_2	010	010
s_2	-	-	0	s_3	011	RDY11
	-	-	1	s_7	111	
s_3	-	-	-	s_4	100	100
s_4	-	-	-	s_5	101	101
s_5	-	0	-	s_4	100	1(u=0)0
	-	1	-	s_6	110	
s_6	-	-	-	s_2	010	010
s_7	0	-	-	s_0	000	STSTST
	1	-	-	s_7	111	

Fig.4.1

Vom scrie ieşirea multiplexorului (i), de fapt $y_{i,n+1}$:

$$y_{i,n+1} = (\overline{y_2}\ \overline{y_1}\ \overline{y_0})I_{0,i} + (\overline{y_2}\ \overline{y_1}\ y_0)I_{1,i} + ... + (y_2 y_1 y_0)I_{7,i} \qquad (4)$$

$$y_{i,n+1} = s_0 I_{0,i} + s_1 I_{i,1} + ... + s_7 I_{7,i} = \sum_{k=0}^{7} s_k I_{k,i} \text{ cu } i = 0,1,2 \qquad (5)$$

Relația (4) se mai poate scrie:

$$Y = I * S, \text{ cu } y = \begin{bmatrix} y_{2,n+1} \\ y_{1,n+1} \\ y_{0,n+1} \end{bmatrix} \qquad (6)$$

$$I = \begin{bmatrix} I_{0,2} & I_{1,2} & \dots & I_{7,2} \\ I_{0,1} & I_{1,1} & \dots & I_{7,1} \\ I_{0,0} & I_{1,0} & \dots & I_{7,0} \end{bmatrix} \qquad s = \begin{bmatrix} s_0 \\ s_1 \\ . \\ . \\ s_7 \end{bmatrix} \qquad (7)$$

Din (7) şi tabelul de tranziție, rezultă:

$$I = \begin{bmatrix} I_{0,2} & I_{1,2} & I_{2,2} & I_{3,2} & I_{4,2} & I_{5,2} & I_{6,2} & I_{7,2} \\ I_{0,1} & I_{1,1} & I_{2,1} & I_{3,1} & I_{4,1} & I_{5,1} & I_{6,1} & I_{7,1} \\ I_{0,0} & I_{1,0} & I_{2,0} & I_{3,0} & I_{4,0} & I_{5,0} & I_{6,0} & I_{7,0} \end{bmatrix} = \begin{bmatrix} 0 & 0 & RDY & 1 & 1 & 1 & 0 & ST \\ 0 & 1 & 1 & 0 & 0 & (u=0) & 1 & ST \\ ST & 0 & 1 & 0 & 1 & 0 & 0 & ST \end{bmatrix}$$

de unde rezultă:

$I_{0,2} = I_{1,2} = I_{6,2} = 0$; $I_{2,2} = RDY$; $I_{3,2} = I_{4,2} = I_{5,2} = 1$; $I_{7,2} = ST$

$I_{0,1} = I_{3,1} = I_{4,1} = 0$; $I_{1,1} = I_{2,1} = I_{6,1} = 1$; $I_{5,1} = (u = 0)$; $I_{7,1} = ST$

$I_{0,0} = I_{7,0} = ST$; $I_{1,0} = I_{3,0} = I_{5,0} = I_{6,0} = 0$; $I_{2,0} = I_{4,0} = 1$

5 Interfața software

Semnificația portului (fig.5.1):

- prin P_k se înțelege bitul de rang k din produs, $0 \le k \le 15$.
- Ordinea de conectare a automatului :
  1. se comută V_{cc} = ON;
  2. se conectează portul paralel;
  3. se intră în program;

4. se introduc operanzii;
5. se aşteaptă rezultatul.

Denumire Port	Adresa PC	Adresa PS/1	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
PDATA	0X378	0X3bc	x x x x x x x x operanzi
PCON	0x37a	0x3be	x x x x 0 LDX $\overline{LDY}$ x
			x x x x 1 s_2 $\overline{s_1}$ $\overline{s_0}$ START+selecţie
PSTARE	0X379	0X3bd	$\overline{p_{i+8}}$ p_i 1 x x x x x RDY=1 D7 D6 din pstare
			x x 0 x x x x x RDY=0

Fig. 5.1

Deconectarea se face invers:

1. deconectare port paralel;
2. sursa V_{cc}=OFF;

În figura 5.2. este descrisă organigrama după care este realizat programul.

Pentru citire se va utiliza $s = s_2 s_1 \overline{s_0}$, ultimii 3 biţi din portul de control, dacă ST=1 (bitul 3). Se începe cu citirea biţilor 15,8 din produs, ceea ce corespunde la s=100 (selecţia canalului 7 din cele 2 multiplexoare ce formează biţii produs).

Produsul se formează conform algoritmului:

1. s=100, P_H=0, P_L=0 (partea High, respectiv Low a produsului)
2. $P_H := \overline{(PSTARE[7]} + P_H) * 2$

   $P_L := \overline{(PSTARE[6]} + P_L) * 2$
3. s: =s+1;
4. Dacă $s \neq 100$, atunci goto(2)
5. P: $=P_H * 2^7 + P_L/2$;

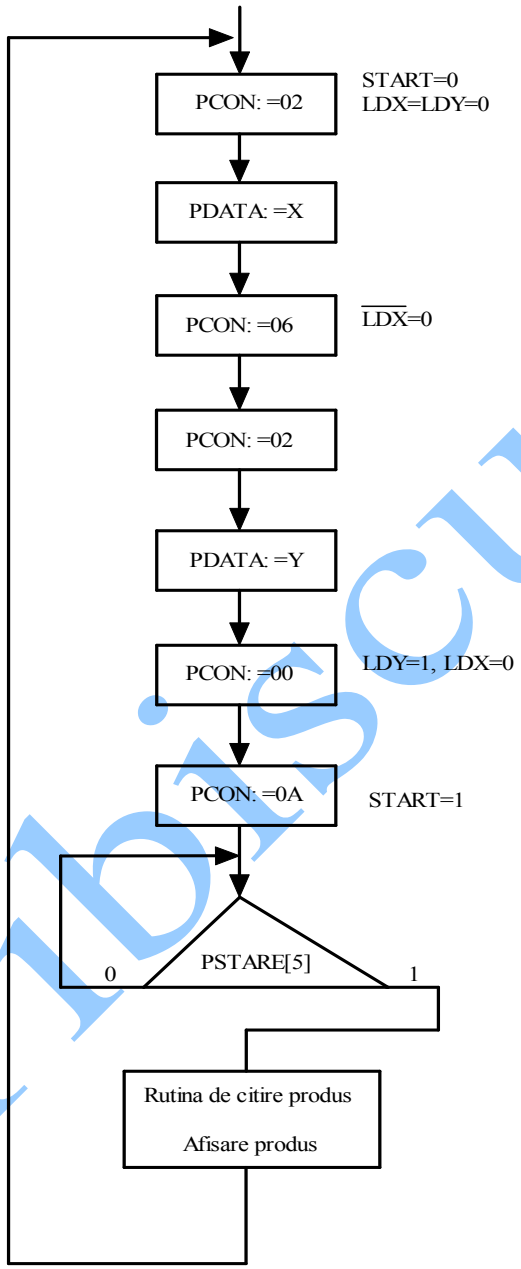

Fig. 5.2.

6 Concluzii

- Această metodă de analiză a sistemelor secvențiale sincrone cu mai multe stări se dovedeşte a fi mult mai eficientă.
- Modul de design a automatului permite interfațarea acestuia cu un PC prin intermediul portului paralel standard.
- Automatul poate fi programat folosind limbajul C sau oricare limbaj de nivel înalt.
- Lucrarea a fost concepută în scop didactic, pentru a arăta studenților modalitatea de sinteză folosind automatele complexe.

Bibliografie

[Cla93] **C. R. Clare** – "*Designing Logic Systems Using State Machine*", 1993

[VB86] **Al. Valachi, M. Birsan** – *Tehnici numerice si automate (Numerical techniques and automata)*, Junimea-Iaşi, 1986

[VH93] **Al. Valachi , Fl. Hoza** – *Analiza şi sinteza dispozitivelor numerice* – Nord-Est Iaşi, 1993

[VM00] **Al. Valachi, C. Monor** - "*Proiectarea Sistemelor Secvențiale Asincrone*" - Tempus Incot, 2000

[***95] ***, „*Principles of Fundamental Logic Design*" - New York 1995