
Lucrarea 11

Sinteza Dispozitivelor Combi-naționale de Înmulțire

Lucrarea prezintă variantele de implementare combinațională a algoritmilor de înmulțire prezentați în lucrările precedente. Performanța acestor structuri poate fi drastic îmbunătățită prin aplicarea pipeline-ului aritmetic, grefat pe structurile de tip Carry Save Adder. Problema rezolvată prezintă o aplicație de pipelinizare tipică

1 Structuri Combi-naționale pentru Înmulțiri Binare

Dispozitivele de înmulțire efectuează operații într-un număr de părți egale cu alt număr de biți, fiecare pe clock-ul său. Se dorește ca întreaga operație de înmulțire să se efectueze pe un singur impuls de clock.

Astfel de dispozitive foarte rapide, pot fi implementate în logică combinațională, cu respectarea algoritmilor deja prezentați, cum sunt paper-and-pencil, Robertson, sau Booth. Operația se execută cu cost dictat de complexitatea spațială. Atunci când lucrăm cu întregi fără semn (i.e. partea de mărime a numerelor reprezentate în semn-mărime), avem operanzii X , Y și produsul P :

$$X = X_{n-1} \dots X_i \dots X_0$$
$$Y = Y_{n-1} \dots Y_0$$

$$P = XY \quad (11.1)$$

Caz particular cu operanzi pe 4 biți (întregi fără semn):
 $X = X_3X_2X_1X_0$ și $Y = Y_3Y_2Y_1Y_0$ - rezultă că P va fi pe 8 biți. Produsul se va obține prin adunarea următoarelor produse parțiale:

$$\begin{aligned} P_0 &= X_0Y_0 \\ P_1 &= X_0Y_1 + X_1Y_0 \\ P_2 &= X_0Y_2 + X_1Y_1 + X_2Y_0 \\ P_3 &= X_0Y_3 + X_1Y_2 + X_2Y_1 + X_3Y_0 \\ P_4 &= X_1Y_3 + X_2Y_2 + X_3Y_1 \\ P_5 &= X_2Y_3 + X_3Y_2 \\ P_6 &= X_3Y_3 \end{aligned} \quad (11.2)$$

Structura utilizată pentru obținerea AND-urilor între biții individuali ai operanzilor este prezentată în Figura 11.1.

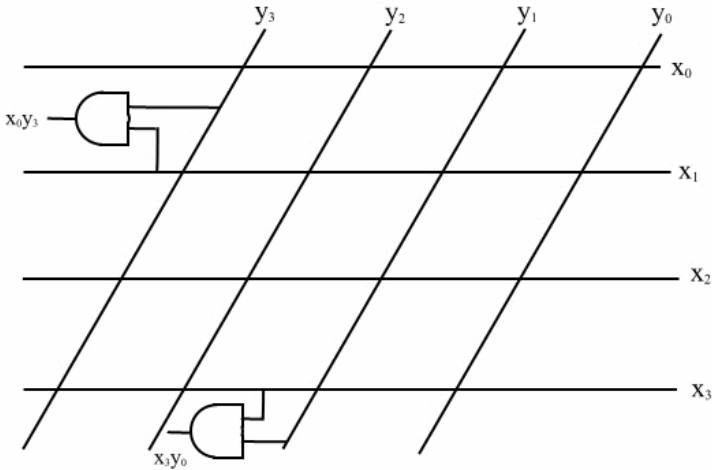


Figura 11.1 Matricea folosită la obținerea produselor de biți individuali.

Complexitatea în termeni de spațiu pentru această soluție este $O(n)$. Porțile ce realizează produsele de 1 bit pot fi distribuite pe celule de înmulțire, fiecare conținând o celulă sumator complet și o poartă *AND*. Se utilizează o matrice formată din celule sumator complet, care de fapt implementează suma elementelor din Ecuația 11.2 (Figura 11.2):

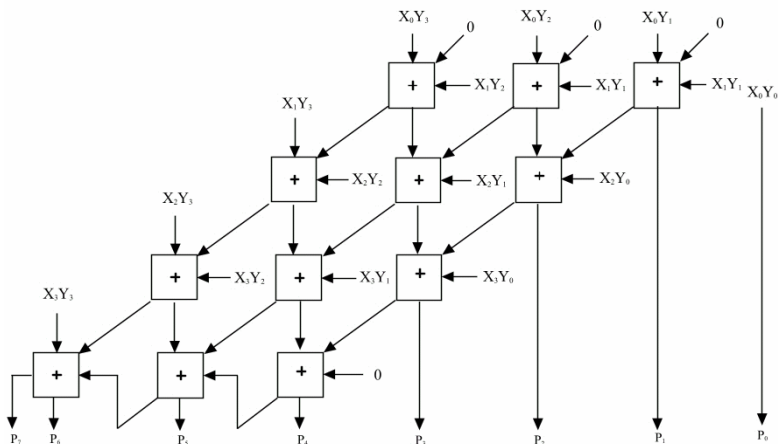


Figura 11.2 Înmulțitor paper-and-pencil pe 4 biți.

2 Implementarea Algoritmului lui Booth

Pentru reprezentarea în complement de doi, este dificil de implementat algoritmul Booth și avem nevoie și de posibilitatea scăderii.

$$\begin{aligned}
 Z &= a + H(b+c) \\
 C_{out} &= (a+D)(b+c) + bc
 \end{aligned}
 \tag{11.3}$$

Porțile ce realizează produsele de 1 bit pot fi distribuite pe celule de înmulțire care conțin fiecare o celulă sumator complet și o poartă *AND*, după cum se arată în Figura 11.3:

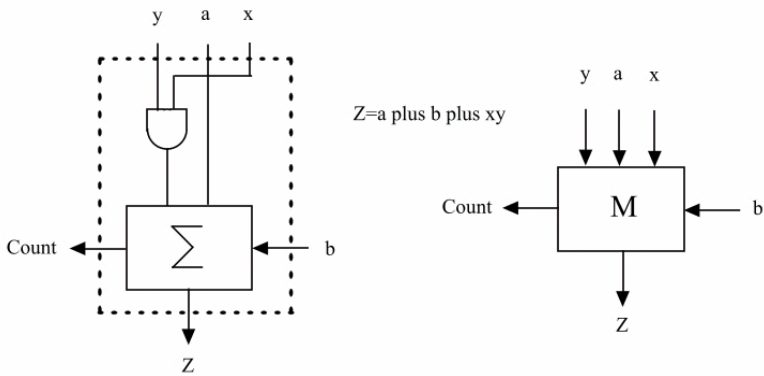


Figura 11.3 Celula de înmulțire pe 1 bit.

Dacă numerele au semn, sinteza acestui înmulțitor se complică. Este necesar să se implementeze printr-o celulă atât adunarea cât și scăderea. Astfel vom ajunge la o celulă de forma prezentată în Figura 11.4

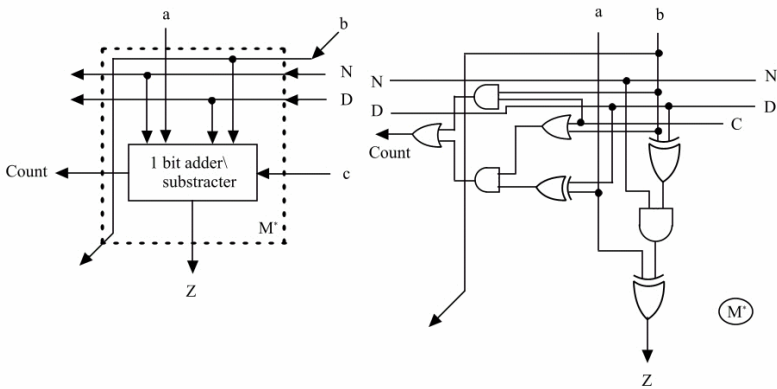


Figura 11.4 Celula de înmulțire Booth pe 1 bit.

În aceste condiții, structura utilizată pentru efectuarea înmulțirii se modifică față de algoritmul paper-and-pencil, capătând forma pe care o prezentăm în Figura 11.5.

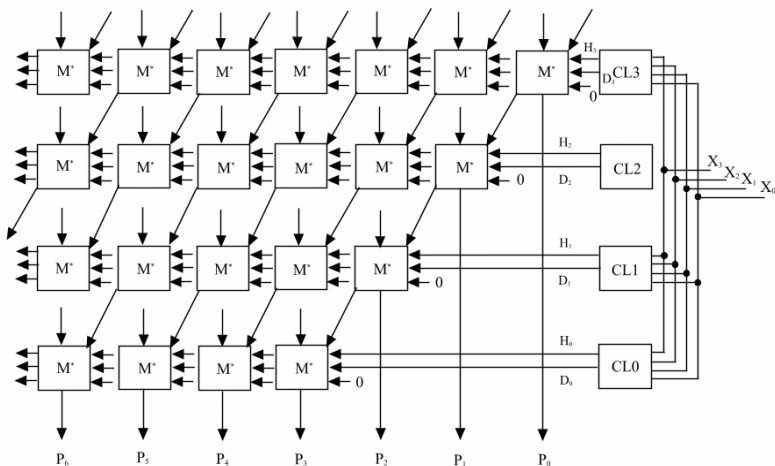


Figura 11.4 Dispozitivul matriceal de înmulțire Booth pentru operanzi 4 biți.

Ecuțiile care stau la baza acestei structuri sunt:

$$\begin{aligned}
 H_3 &= X_2X_3 + X_3X_2 = X_3 + X_2 \\
 D_3' &= X_3X_2 \\
 H_2' &= X_2 + X_1 \\
 D_2' &= X_2X_1 \\
 H_1' &= X_1 + X_0 \\
 D_1' &= X_1X_0 \\
 H_0' &= X_0 + X_0 \\
 D_0' &= X_0X_0 = X_0
 \end{aligned}
 \tag{11.4}$$

3 Aplicații

Problema 11.1 (Patterson & Hennesy – Computer Organization & Design. The Hardware/Software Interface) Se consideră un înmulțitor combinațional de tip paper-and-pencil pe 8 biți $A \times B$, ($A = (a_7a_6 \dots a_0)$, $B = (b_7b_6 \dots b_0)$) după cum se prezintă în Figura 11.5,

unde sunt figurate produsele de forma $b_i A$, $i = 0..7$. Într-o primă instanță produsele de această formă sunt aplicate intrărilor corespunzătoare din figură. Apoi, după ce cel de-al treilea nivel CSA termină calculul corespunzător primei faze, sunt aplicate următoarele produse de forma $b_i A$, cele din paranteze. După cea de-a doua trecere prin cele 3 niveluri CSA, se aplică sumatorul paralel PA, care poate fi CLA, RCA, etc. În schema dată, faptul că o matrice aritmetică poate fi pipelinizată nu este exploatat. Se cere proiectarea unei structuri similare prin simpla modificare a legăturilor marcate cu (*) și (**). Această nouă structură va trebui să funcționeze mai rapid decât cea din Figura 11.5. Explicați de ce.

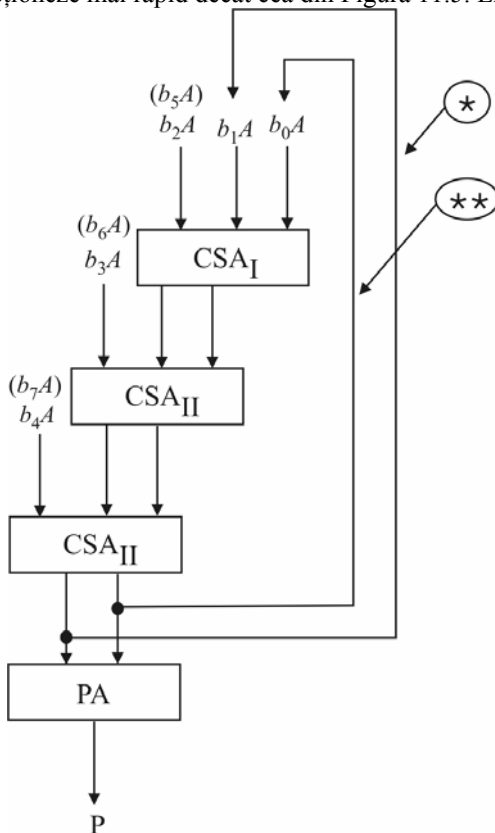


Figura 11.5 Structura înmulțitorului pe 8 biți.

În prima variantă, structura este parcursă, pentru obținerea produsului, în două cicluri, de forma următoare, defalcată pe slot-uri de timp.

CSA_I	CSA_{II}	CSA_{III}		CSA_I	CSA_{II}	CSA_{III}	PA
T_0	T_1	T_2	T_3	T_4	T_5	T_6	

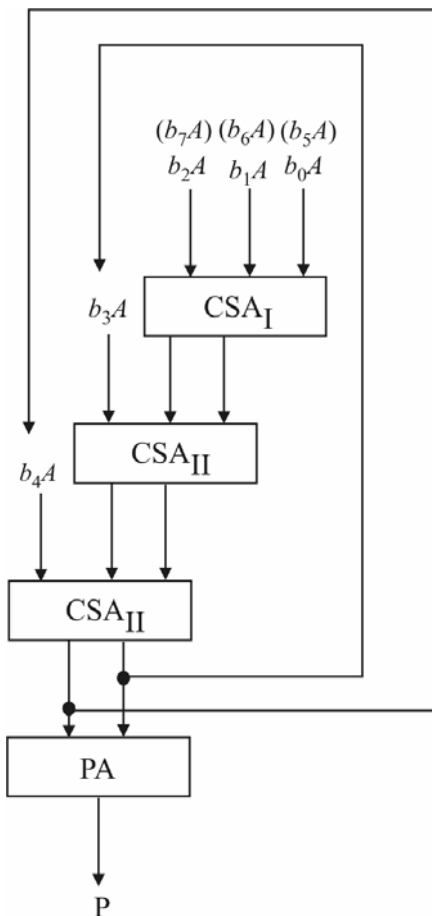


Figura 11.6 Soluția pipelinizată.

În cea de-a doua variantă (Figura 11.6), parcurgerea pentru a doua oară a nivelului CSA_I se poate suprapune peste prima parcurgere a lui CSA_{III} . Corespunzător, vom avea următoarea diagramă de timp :

CSA_I	CSA_{II}	CSA_{III}				
		CSA_I	CSA_{II}	CSA_{III}	PA	
T_0	T_1	T_2	T_3	T_4	T_5	

Prima soluție revendică 7 sloturi de timp. Prin comparație, cea de-a doua soluție revendică doar 6 sloturi de timp.

Problema 11.2 (propusă) Să se estimeze în termeni de τ întârzierea dispozitivului de la problema anterioară, în cele două.

Problema 11.3 (propusă) Să se proiecteze un dispozitiv de înmulțire binară combinațional, după algoritmul lui Robertson. Se cere, de asemenea, estimarea întârzierii acestui dispozitiv, în termeni de τ , unde τ este întârzierea pe porțile *AND* și *OR*, iar 2τ este întârzierea pe o poartă *XOR*.

Problema 11.3 (propusă) Să se proiecteze un dispozitiv combinațional de înmulțire a două numere în format BCD, pe 4 cifre zecimale (dimensiunea operandului).

