Lucrarea 9

Sinteza Dispozitivelor de Înmulțire Booth

Această lucrare extinde sinteza dispozitivelor de înmulțire în vederea obținerii unei penalități minimale asupra performanței dictată de operațiile de adunare. Algoritmul lui Booth, împreună cu varianta sa modificată, aduc îmbunătățiri semnificative procedurii lui Robertson, cu prețul unei investiții hardware pentru realizarea operațiilor de adunare/scădere.

Algoritmul lui Robertson se bazează pe o observație legată de reprezentarea numerelor în complement de 2. Această observație implică însă o penalitate în ceea ce privește performanța, fiind presupusă activarea de n ori a sumatorului paralel pentru numere reprezentate pe n biți, ceea ce afectează negativ timpul total de înmulțire.

De aceea s-a pus problema găsirii unei metode de accelerare care să depindă de particularitățile configurației binare particulare a înmulțitorului, şi anume, dacă înmulțitorul prezintă porțiuni constituite din şiruri de „0“ sau de „1“ atunci se poate beneficia de metoda lui Booth în vederea surmontării penalității de performanță care apare în mod tipic în cazul procedurii lui Robertson.

$$X = ...X^*...$$

$$X^* = x_i x_{i+1} ... x_{i+k-1} x_{i+k} x_{i+k+1} = 011...110$$

a row of 1s

Fără a pierde din generalitate să admitem că X este un număr fracționar şi vom spune că aportul lui X^* la $X \cdot Y$ este dat de următoarea relație:

Se observă că se realizează doar două activări ale sumatorului paralel.

Dacă avem $x_i x_{i-1}$ biți ai înmulțitorului şi dacă aceştia sunt 01, sugerând că înmulțitorul este parcurs de la stânga la dreapta, atunci în vederea obținerii produsului se efectuează o adunare a lui Y la produsul parțial. Dacă $x_i x_{i-1}$ sunt 10 atunci se efectuează o scădere a lui Y din produsul parțial, iar în situația în care cei doi biți sunt 00 sau 11, ne aflăm pe parcursul baleierii unui şir de 1 sau 0 şi atunci nu este necesară efectuarea vreunei operații. Algoritmul lui Booth este echivalent cu o recodificare a înmulțitorului într-o formă specială (signed digit form), în care se utilizează 3 simboluri şi anume: 1 când se face o adunare (când cei doi biți adiacenți consultați sunt 01), 0 când sunt 00 sau 11 şi $\bar{1}$ când sunt 10.

De exemplu:

$$X = -\frac{79}{128};\ X_{SM} = 1.1001111;\ X_{C2} = 1.0110001|0;$$

$$X^* = \bar{1}10\bar{1}001\bar{1} = -\frac{1}{2^7} + \frac{1}{2^6} - \frac{1}{2^4} + \frac{1}{2} - 1$$

În reprezentarea în complement de 2 a numărului, pentru a putea lua în considerare şi ultimul bit, se mai consideră implicit un bit de "bordare" având valoarea logică 0 (cel de după bara verticală). Algoritmul lui Booth are următoarea formă în limbajul de descriere Hayes:

```
Booth mutiplier(in:INBUS;out:OUTBUS);

register A[7:0],Q[7:-1],M[7:0],COUNT[2:0];
bus INBUS[7:0],OUTBUS[7:0];
BEGIN : A:=0, COUNT:=0,
INPUT : M:=INBUS; {c0}
Q:=INBUS, Q[-1]:=0; {c1}

SCAN  : if Q[0]Q[-1]=0 then A[7:0] := A[7:0] +
M[7:0],  goto TEST; {c2}
else if Q[0]Q[-1]=10 then A[7:0]:=A[7:0]-M[7:0];
{c2,c3}
TEST  : if COUNT7=1 then goto OUTPUT, RSHIFT:
A[7]:=A[7], A[6:0].Q:=A.Q[7:-1],
COUNT:=COUNT+1, goto SCAN; {c4}
```

```
OUTPUT: OUTBUS:=A, Q[0]:=0; {c5}
OUTBUS:=Q; {c6}
END. end Booth multiplier {END}
```

Schema se uşor mai complicată faţă de Robertson pentru că acum trebuie efectuate atât adunări cât şi scăderi, deci este necesară existenţa unui sumator-scăzător complet. Arhitectura este prezentată în Figura 8-1.

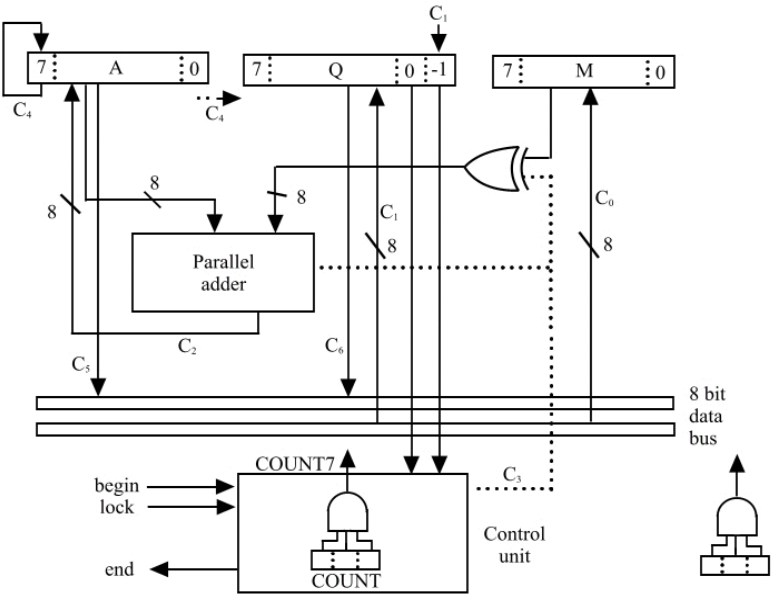

Figura 8-1. Platforma hardware pentru un dispozitiv de înmulţire Booth

Algoritmul lui Booth modificat

Când înmulţitorul este format din alternanţe de 1 şi 0 nu se beneficiază de observaţia lui Booth ci se înrăutăţeşte per global performanţa înmulţitorului, astfel încât s-a ajuns la algoritmul lui Booth modificat. La acest algoritm se ia în considerare situaţia de „0 izolat“ respectiv de „1 izolat“ şi în cazul unui „0 izolat“ se efectuează doar 0 scădere, iar în cazul lui „1 izolat“ se efectuează doar o adunare. Dacă avem biţii:

$$...x_i x_{i+1} x_{i+2}... = ...101...$$

$$...x_i x_{i+1} x_{i+2}... = ...\ 1 \quad 0 \quad 1\ ...$$

$$-2^{-i} + 2^{-(i+1)}(2-1) = -2^{-(i+1)}$$

această situație corespunde unui 0 izolat, ceea ce înseamnă că va fi necesară efectuarea doar unei scăderi.

Dacă avem biții:

$$...x_i x_{i+1} x_{i+2}... = ...010...$$

$$...x_i x_{i+1} x_{i+2}... = ...\ 0 \quad 1 \quad 0\ ...$$

$$2^{-i} - 2^{-(i+1)} = 2^{-(i+1)}$$

această situație corespunde unui 1 izolat, ceea ce înseamnă că va fi necesară efectuarea doar unei adunări.

Se spune că se apelează la o nouă recodificare, canonică, a înmulțitorului, care se efectuează prin intermediul unui fanion (flag) F. Acesta va fi inițial pus pe 0 şi 1 se va modifica valoare doar dacă la parcurgerea de la dreapta spre stânga al înmulțitorului se întâlneşte un şir de 1 (2 sau mai mulți biți având valoarea 1) şi va fi repus pe 0 doar dacă se întâlneşte un şir de 0 (2 sau mai mulți biți având valoarea 0). Dacă atribuim flag-ului F o valoare booleană, recodificarea Booth canonică se efectuează pe baza Tabelului 9-1:

x_3	x_2	x_1	x_0	x'_3	x'_2	x'_1	x'_0	H_3	D_3	H_2	D_2	H_1	D_1	H_0	D_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0
0	0	1	1	0	1	0	$\bar{1}$	0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	1	0	1	0	0	1	0	0	0	1	0
0	1	1	0	1	0	$\bar{1}$	0	1	0	0	0	1	1	0	0
0	1	1	1	1	0	0	$\bar{1}$	1	0	0	0	0	0	1	1
1	0	0	0	$\bar{1}$	0	0	0	1	1	0	0	0	0	0	0
1	0	0	1	$\bar{1}$	0	0	$\bar{1}$	1	1	0	0	0	0	1	0
1	0	1	0	$\bar{1}$	0	1	0	1	1	0	0	1	0	0	0
1	0	1	1	0	$\bar{1}$	0	$\bar{1}$	0	0	1	1	0	0	1	1
1	1	0	0	0	$\bar{1}$	0	0	0	0	1	1	0	0	0	0
1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	0
1	1	1	0	0	0	$\bar{1}$	0	0	0	0	0	1	1	0	0

1	1	1	1	0	0	0	$\bar{1}$	0	0	0	0	0	0	1	1

Tabelul 9-1: Recodificarea Booth canonică

De exemplu, recodificarea canonică a aceluiaşi număr rezultă în felul următor:

$X_{C2} = 1|10110001$

$X^{**} = 0\bar{1}0\bar{1}0001 = -\frac{1}{2} - \frac{1}{2^3} + \frac{1}{2^7} = -\frac{79}{128}$

$F = 11110000|0$

În limbajul de descriere Hayes algoritmul lui Booth modificat va arăta astfel:

```
modified Booth algorithm(in:INBUS;out:OUTBUS);
register A[7:0],Q[8:0],M[7:0],COUNT[2:0],F,OVR;
bus INBUS[7:0],OUTBUS[7:0];
BEGIN:      A:=0, COUNT:=0, F:=0,
INPUT:      M:=INBUS; {c0}
            Q[7:0]:=INBUS[7:0], Q[8]:=INBUS[7]; {c1}
ZEROTEST:   if Q=0 then goto OUTPUT, if M≠ 0 then goto SCAN,
            Q:=0, goto OUTPUT; {c2}
SCAN:       if F=0 then begin
                  if Q[0]Q[1]=01 then A[7:0]:=A[7:0]+M[7:0],
                  OVR:= ν ; {c3}
                  if Q[0]Q[1]=11 then A[7:0]:=A[7:0]-M[7:0],
                  OVR:= ν , F:=1; end
            else goto TEST; {c3,c4,c5}
            if F=1 then begin
                  if Q[0]Q[1]=00 then A[7:0]:=A[7:0]+M[7:0],
                  OVR:= ν , F:=0; {c3,c6}
                  if Q[0]Q[1]=10 then A[7:0]:=A[7:0]-M[7:0],
                  OVR:= ν ; end {c3,c4}
TEST:       if COUNT7=1 then go to OUTPUT;
```

RSHIFT: $A[7] = A[7] \cdot M[7] \cdot \overline{OVR} + F \cdot \overline{M[7]} + F \cdot M[7] \cdot OVR$, A[6:0].Q:=A.Q[8:1], COUNT:=COUNT+1, go to SCAN;**{c7}**

OUTPUT: OUTBUS:=A, Q[1]:=0; **{c8}**
OUTBUS:=Q; **{c9}**

END. {end mBooth multiplier}.

Platforma hardware de tip Hayes este prezentată în Figura 9-2.

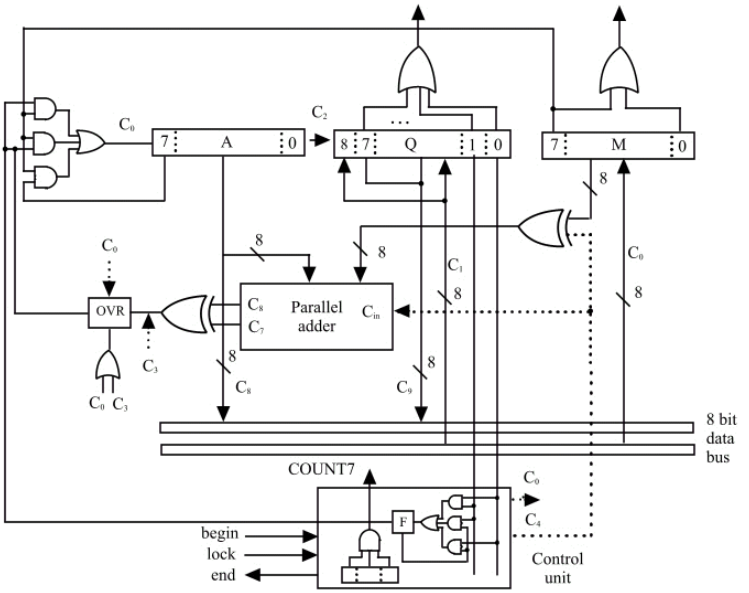

Figura 9-2. Platforma hardware pentru un dispozitiv de înmulțire Booth modificat

Pentru algoritmul lui Booth modificat rezultă ordinograma din Figura 9-3, în care s-a apelat la aglomerarea tuturor operațiilor care nu sunt mutual exclusive. Sinteza unității de control după metoda one-hot va fi realizată conform următoarelor ecuații:

$$S_0^+ = S_0 \cdot \text{Begin} + S_{10}$$

$$S_1^+ = S_0 \cdot \overline{\text{Begin}}$$

$$S_2^+ = S_1$$

$$S_3^+ = S_2 \cdot \overline{Q_{=0}} \cdot M_{=0}$$

$$S_4^+ = \overline{F} \cdot Q_7 \cdot \overline{Q_6} \cdot (S_8 + S_2 \cdot \overline{Q_{=0}} \cdot \overline{M_{=0}})$$

$$S_5^+ = \overline{F} \cdot Q_7 \cdot Q_6 \cdot (S_8 + S_2 \cdot \overline{Q_{=0}} \cdot \overline{M_{=0}})$$

$$S_6^+ = F \cdot \overline{Q_7} \cdot \overline{Q_6} \cdot (S_8 + S_2 \cdot \overline{Q_{=0}} \cdot \overline{M_{=0}})$$

$$S_7^+ = F \cdot \overline{Q_7} \cdot Q_6 \cdot (S_8 + S_2 \cdot \overline{Q_{=0}} \cdot \overline{M_{=0}})$$

$$S_8^+ = \overline{\text{Count7}} \cdot (S_4 + S_5 + S_6 + S_7 + $$
$$+ (S_8 + S_2 \overline{Q_{=0}} \overline{M_{=0}}) \cdot (\overline{F} \cdot \overline{Q_7} + F \cdot Q_7))$$

$$S_9^+ = \text{Count7} \cdot (S_4 + S_5 + S_6 + S_7 + $$
$$+ (S_8 + S_2 \overline{Q_{=0}} \overline{M_{=0}}) \cdot (\overline{F}\,\overline{Q_7} + FQ_7) + S_2 Q_{=0} + S_3)$$

$$S_{10}^+ = S_9$$

Prezentăm în continuare o parte a sintezei unității de control pentru un dispozitiv de înmulțire care operează după algoritmul lui Booth modificat conform metodei state table. Codificând fiecare stare cu $x_3 x_2 x_1 x_0$, tabela de stări rezultă într-o manieră simplă şi directă din ordinogramă şi este prezentată în Figura 9-4.

Din tabela de stări rezultă tabelele de tranziții, prezentate pentru x_0, respectiv x_1, în Figura 9-5, respectiv Figura 9-6. După minimizare, rezultă următoarele ecuații pentru x_0, respectiv x_1 :

$$x_0^+ = \overline{x_2} \cdot x_1 \cdot x_0 + \overline{x_3} \cdot \overline{x_1} \cdot \overline{x_0} \cdot Q_6 (F \oplus Q_7) + \text{Count7} \cdot x_2 +$$
$$+ \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot [Q_{=0} + \overline{Q_{=0}} \cdot M_{=0} + \overline{Q_{=0}} \cdot \overline{M_{=0}} (...)] +$$
$$+ \overline{\text{Begin}} \cdot \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + \text{Count7} \cdot (\overline{F} \cdot \overline{Q_7} + F \cdot Q_7) \cdot x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0}$$

$$x_1^+ = \overline{x_2} \cdot \overline{x_1} \cdot x_0 + F \cdot \overline{Q_7} \cdot x_3 \cdot \overline{x_1} + \overline{Q_{=0}} \cdot (M_{=0} + \overline{M_{=0}} \cdot F \cdot \overline{Q_7}) \cdot \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0}$$

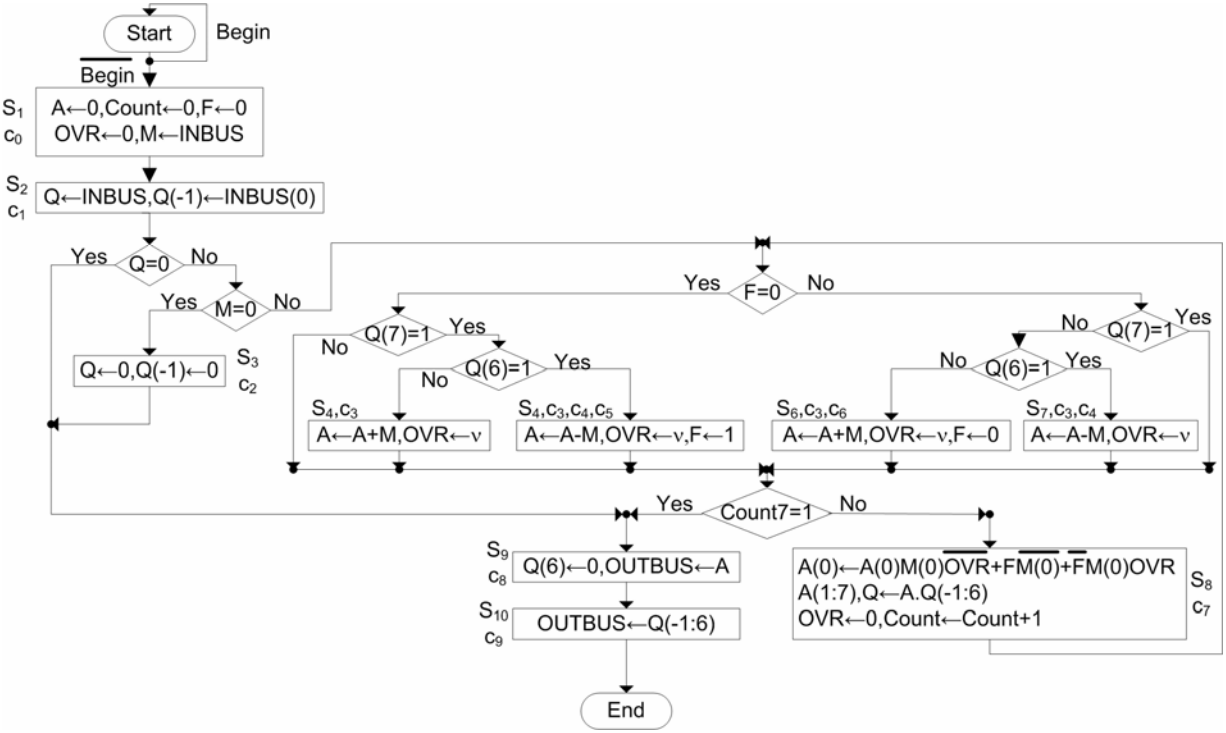

Figura 9-3. Ordinograma corespunzătoare algoritmului Booth modificat

	FQ_6Q_7	000	001	011	010	110	111	101	100
	Begin								
S_0	0	S_1							
	1	S_0							
S_1		S_2							
	$Q_{=0}M_{=0}$Count7								
S_2	1 x x	S_9							
	0 1 x	S_3							
	0 0 0	S_8	S_4	S_5	S_8	S_7	S_8	S_8	S_6
	0 0 1	S_9			S_9		S_9	S_9	
S_3		S_9							
S_4	Count7								
↓	0	S_8							
S_7	1	S_9							
	Count7								
S_8	0	S_8	S_4	S_5	S_8	S_7	S_8	S_8	S_6
	1	S_9			S_9		S_9	S_9	
S_9		S_{10}							
S_{10}		S_{10}							

Figura 9-4. Tabela de stări corespunzătoare algoritmului Booth modificat

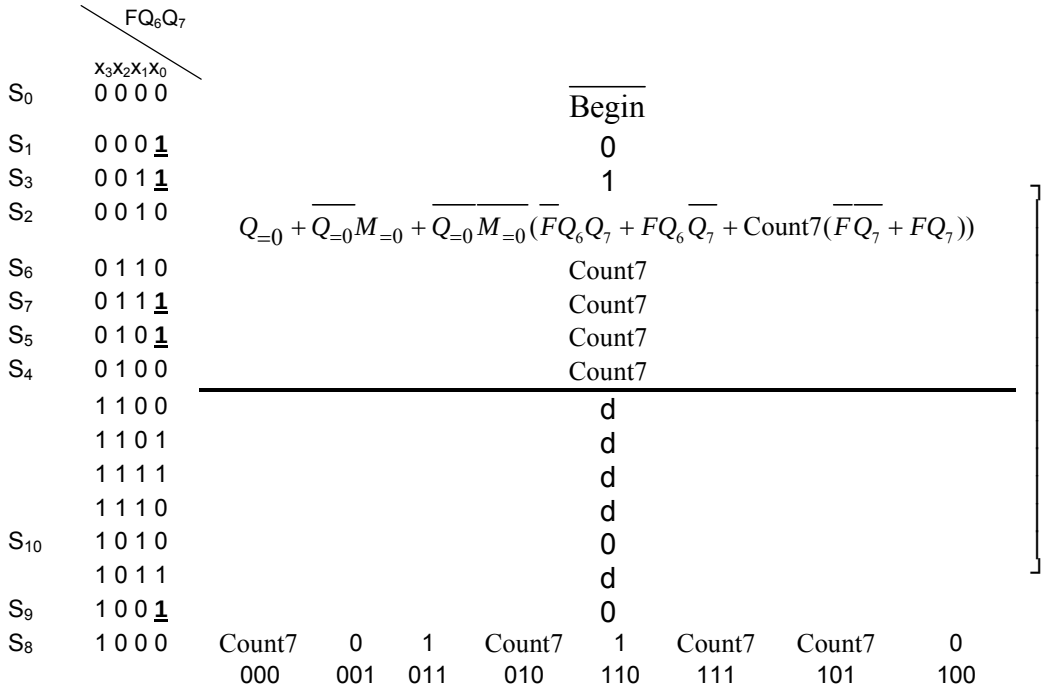

Figura 9-5. Tabela de tranziții pentru x_0

	FQ_6Q_7 / $x_3x_2x_1x_0$	
S_0	0000	0
S_1	0001	1
S_3	00**1**1	0
S_2	00**1**0	$\overline{Q_{=0}}M_{=0} + \overline{Q_{=0}}\,\overline{M_{=0}} + F\overline{Q_7}$
S_6	01**1**0	0
S_7	01**1**1	0
S_5	0101	0
S_4	0100	0
	1100	d
	1101	d
	1111	d
	1110	d
S_{10}	10**1**0	0
	1011	d
S_9	1001	1
S_8	1000	$F\overline{Q_7}$

Figura 9-6. Tabela de tranziții pentru x_1