

---

# Lucrarea 6

## Metode de Sinteză a Dispozitivelor Aritmetice: Platforma Hardware Hayes

Această lucrare introduce conceptele utilizate pentru descrierea unei platforme hardware după modelul Hayes. Pentru exemplificare este prezentat algoritmul de înmulțire a două numere reprezentate în complement de 2, după procedura lui James Robertson.

Descrierea algoritmului în pseudo-limbajul Hayes este urmată de explicații privitoare la paralelizarea microoperațiilor neconflictuale (care nu sunt mutual exclusive), de realizarea ordinogramei corespunzătoare și prezentarea arhitecturii complete a dispozitivului de înmulțire.

Algoritmul lui Robertson permite înmulțirea a două numere în complement de 2 (întregi sau fracționare) și se bazează pe interpretarea biților de semn și mărime:

$$X = x_{n-1}x_{n-2}\dots x_0 = \begin{cases} 0x_{n-2}\dots x_0, & X \geq 0 \\ 1x_{n-2}\dots x_0, & X < 0 \end{cases}$$

În consecință, un număr negativ (-X) exprimat în complement de 2 va putea fi scris în felul următor, derivat din corespondentul său pozitiv

$$-X = 1x_{n-2}\dots x_1x_0 = 1.00\dots 00 + \sum_{i=0}^{n-2} x_i \cdot 2^{-i-n+1} = -1 + \sum_{i=0}^{n-2} x_i \cdot 2^{-i-n+1}$$

Astfel, un număr exprimat în complement de 2 poate fi interpretat ca o parcurgere dinspre bitul de semn (partea cea mai semnificativă) către bitul cel mai puțin semnificativ.

Exemplu :

$$-X = -\frac{19}{32} = 1.01101 = -1 + 0.01101 = -1 + \frac{13}{32}$$

Reprezentarea Robertson a complementului de 2 permite tratarea unitară a numerelor, indiferent de semnul acestora, ca și când ar fi pozitive, prin parcurgerea acestora dinspre partea mai puțin semnificativă cea mai semnificativă. Bitul de semn este tratat în același mod ca și biții de mărime. Structura dispozitivului de înmulțire Robertson este similară celei pentru înmulțirea numerelor exprimate în semn-mărime, față de aceasta existând însă unele ajustări.

Fără a pierde din generalitate, vom considera înmulțirea a 2 numere fracționare reprezentate pe 8 biți. Atât timp cât produsele parțiale generate sunt pozitive, operația de deplasare la dreapta va introduce valoarea 0 în poziția cea mai semnificativă a registrului acumulator. În momentul în care produsul parțial generat devine negativ, este necesară introducerea valorii 1 în poziția cea mai semnificativă a registrului acumulator. Aceste situații se explică prin modalitatea de extindere a numărului de biți pentru un număr exprimat în complement de 2, care este echivalentă cu extinderea bitului de semn (introducerea de „leading 1s“ sau „leading 0s“).

Finalul algoritmului corespunde operării bitului de semn. În cazul în care înmulțitorul (numărul stocat în registrul Q) a fost negativ, va fi necesar un pas de corecție, prin care se efectuează o scădere, corespunzătoare reprezentării Robertson a numărului în complement de 2:

$$P = -Y + \sum_{i=0}^6 x_i \cdot 2^{i-7} \cdot Y$$

Față de dispozitivul de înmulțire a numerelor exprimate în semn-mărime (care opera doar biții de mărime, semnul fiind calculat ulterior), algoritmul lui Robertson presupune operarea tuturor biților din reprezentare (inclusiv biții de semn), ceea ce impune controlul logic al bitului care se încarcă în cel mai semnificativ bit al acumulatorului (adică bitul de semn al rezultatului parțial) iar acest lucru se face printr-un fanion:

$$F = (x_i \cdot y_7) + F$$

Considerând toate cele amintite, un dispozitiv cablat de înmulțire Robertson prezintă următoarea structură hardware:

- o magistrală de intrare pe 8 biți, necesară citirii operanzilor;

- o magistrală de ieșire pe 8 biți, prin intermediul căreia rezultatul final va fi salvat în două etape;
- 3 registre, fiecare pe 8 biți: registrul Q (multiplier) care stochează inițial înmulțitorul Y, registrul M (multiplicand) care stochează pe tot parcursul operației de înmulțire X și registrul A (acumulator) care inițial este resetat și care, concatenat cu registrul Q va stoca produsele parțiale și, în final, rezultatul înmulțirii (cei mai semnificativi biți în registrul A iar cei mai puțin semnificativi în registrul Q);
- un sumator paralel pe 8 biți;
- o unitate de control care va genera semnalele necesare controlului operațiilor necesitate de algoritm.

Funcționarea unei astfel de structuri se poate descrie principal în două moduri distincte: fie printr-o ordinogramă (flow chart), fie printr-o descriere într-un limbaj de descriere hardware. Ordinograma este caracterizată prin două categorii de blocuri: operative (cărora li se alocă seturi de microoperații care pot fi suprapuse pe același impuls de tact) și decizionale (cărora le revine testarea condițiilor impuse de algoritm).

COUNT7 reprezintă un semnal de intrare care va fi setat (va deveni 1) în momentul în care registrul COUNT va conține valoarea 7 (corespunzător celor 7 pași de adunare/deplasare necesități de algoritm). Vom detalia structura înmulțitorului Robertson utilizând o descriere comportamentală de tip Hayes, caracterizată prin:

- declararea registrelor și magistralelor prin indicarea numelor și dimensiunilor acestora (un bistabil este echivalent unui registru de dimensiune 1);
- indicarea microoperațiilor care pot fi efectuate neconflictual în aceeași perioadă de tact prin separarea lor prin virgulă; cele care trebuie efectuate în perioade diferite de tact vor fi separate prin punct și virgulă;
- eventualele salturi sunt specificate prin „go to“ (necondiționate) sau prin „if ... then“ (condiționate);

Semnalele de control corespunzătoare seturilor de microoperații neconflictuale (sau care nu sunt mutual exclusive) sunt notate cu c1,c2,...,c6 și se regăsesc în dreptul instrucțiilor corespunzătoare. Descrierea hardware de tip Hayes pentru dispozitivul hardware va fi următoarea:

```

declare register A[7:0], Q[7:0], M[7:0], COUNT[2:0], F;
declare bus Inbus[7:0], Outbus[7:0];
Begin:  A := 0, COUNT := 0, F := 0,
Input:  M := Inbus;                                {c0}
        Q := Inbus;                                {c1}
Test1:  if Q[0] = 0 then goto Rshift;
Add:    A[7:0] := A[7:0] + M[7:0], F := Q[0] · M[7] + F;    {c2}
RShift: A[7] := F, A[6:0].Q := A.Q[7:1], COUNT := COUNT + 1,
        if COUNT7 = 0 then goto Test1;                {c3}
Test2:  if Q[0] = 0 then go to Output;
Correct: A[7:0] := A[7:0] - M[7:0], Q[0] := 0;          {c2, c4}
Output: Outbus := A;                                  {c5}
        Outbus := Q;                                  {c6}
END.    {Issue completion signal END}

```

În acest algoritm, semnalul de control  $c_0$  asigură desfășurarea operațiilor de resetare (de aducere la 0) a registrelor A, COUNT și a bistabilului F, precum și a operației de citire de pe magistrala de intrare (Inbus) a primului operand (deînmulțitul M). Semnalul de control  $c_1$  asigură desfășurarea operației de citire de pe magistrala de intrare (Inbus) a celui de-al doilea operand (înmulțitorul Q). Este evident, în acest context, că citirea celor doi operanzi de pe aceeași magistrală ilustrează o situație conflictuală. În consecință, citirea operanzilor va fi operată în mod separat, cele două operații fiind mutual exclusive. În mod analog se rezolvă toate situațiile conflictuale care pot să apară în descrierea unui algoritm.

Scopul semnalul de control  $c_2$  este desfășurarea adunării (eticheta Add) deînmulțitului (M) la acumulator (A) și a calculării noii valori pentru F. Pasul final de corecție (eticheta Correct) presupune o scădere, fiind gestionată de două semnale de control,  $c_2$  și  $c_4$ . Această situație se datorează faptului că, în reprezentarea în complement de 2, scăderea este echivalentă unei adunări, sub rezerva prelucrării scăzătorului. În timp ce adunarea este gestionată de semnalul de control  $c_2$ , complementarea scăzătorului și injectarea unui transport inițial în sumator (pentru a realiza

complementarea față de 2) este asigurată de către semnalul de control  $c_4$ . Faptul că semnalul  $c_2$  gestionează, pe lângă operația de adunare și operația de calcul a unei noi valori pentru F nu este de natură să deranjeze, întrucât această valoare nu mai este folosită.

Ordinograma pentru algoritmul Robertson, împreună cu indicarea stărilor și a semnalelor de control generate, este prezentată în Figura 6-1 iar arhitectura hardware a dispozitivului este indicată în Figura 6-2. Față de descrierea algoritmului, în ordinogramă apare un semnal de intrare suplimentar Begin care menține unitatea de control într-o stare inițială până la activarea acestui semnal.

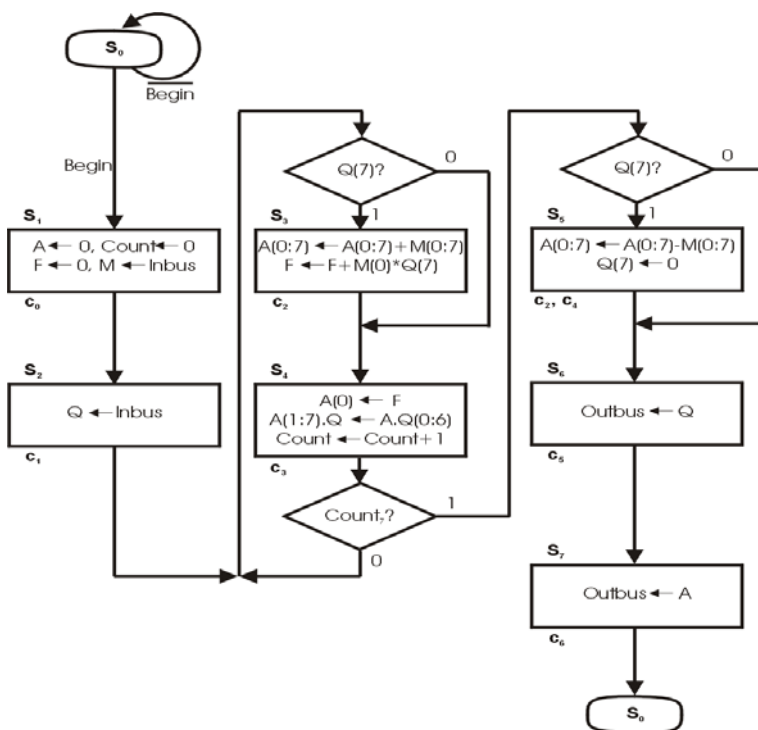


Figure 6-1: Ordinograma pentru algoritmul lui Robertson

Condiționarea unei anumite operații de către un semnal de control se realizează prin filtrarea cu ajutorul unei porți AND a intrării, respectiv

ieșirii dorite. De exemplu, semnalul  $c_0$  autorizează citirea operandului M de pe magistrala de intrare, implementarea având structura prezentată în Figura 6-3.

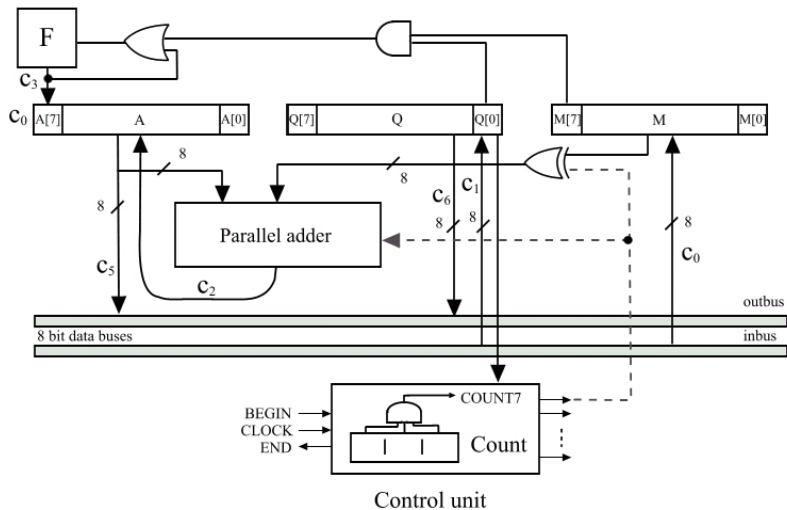


Figure 6-2: Platforma hardware pentru algoritmul lui Robertson

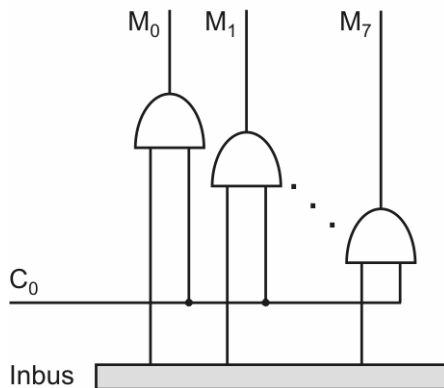


Figure 6-3: Implementarea semnalului de control  $c_0$