Lucrarea 7

Metode de Sinteză a Dispozitivelor Aritmetice: Unitatea de Control

Această lucrare de anvergură urmăreşte prezentarea metodelor de sinteză a unităților de control în variantă cablată. Unitățile de control reprezintă automatele cu stări finite, sinteza acestora fiind prezentată prin intermediul a patru metode diferite. Alegerea metodei se face prin evaluarea constrângerilor ținând de disponibilitatea resurselor logice combinaționale (porți logice) şi secvențiale (bistabile).

Unitatea de control reprezintă un dispozitiv secvențial care are rolul de generator al semnalelor de control respectând secvența impusă de algoritmul pe care îl implementează. Pentru implementarea şi sinteza acestor automate în variantă cablată punctul inițial îl constituie descrierea algoritmului, fie printr-un limbaj de descriere hardware, fie prin ordinogramă. Sunt utilizează în mod frecvent diverse suprapuneri de metode sau abordări ad-hoc/euristice. Există însă doar 3 metode sistematice distincte, cunoscute sub denumirile următoare:

1. metoda tabelei de stări (state table method)
2. metoda elementelor de întârziere (delay element method)
3. metoda numărătorului de secvențe (sequence counter method)

Deşi toate aceste metode au ca rezultat arhitecturi hardware identice ca şi funcționalitate, există diferențe structurale care pot impune utilizarea unei anumite metode în funcție de contextul existent. Astfel, dacă se doreşte obținerea un număr minim de bistabile, metoda state table este cea recomandată. Dacă dimpotrivă, numărul de bistabile nu reprezintă un criteriu de proiectare dar există constrîngeri privitoare la numărul de porți logice disponibile, metoda delay element este

preferabilă, în timp ce metoda sequence counter produce structuri regulate. Caracteristicile metodelor de sinteză ar putea fi sintetizate în modul următor:

- State table
  - Avantaj: metodă sistematică şi algoritmică
  - Dezavantaj: arhitectura hardware rezultată prezintă structuri complexe, neregulate; grad ridicat de dificultate la depanare
- Delay element şi sequence counter
  - Avantaj: prezintă caracter euristic, păstrează o mai mare claritate la urmărirea schemei, depanare mai facilă
  - Dezavantaj: labilitate, implică mai multe circuite decât metoda state table

În cele ce urmează se va exemplifica realizarea după fiecare metodă de sinteză a unității de control cablate pentru algoritmul de înmulțire Robertson, pentru numere întregi reprezentate pe 8 biți în complement de 2. Vor fi respectate algoritmul şi ordinograma descrise în lucrarea 6.

1. Metoda tabelei de stări (state table method)

Metoda demarează prin elaborarea aşa-numitului tabel de stări, care prezintă câte un rând distinct pentru fiecare stare internă (starea memorată a maşinii corespunzătoare acțiunii unui impuls de clock) şi prezintă câte o coloană corespunzătoare fiecărui vector de intrare posibil. Sunt considerate intrări toate semnalele care participă în mod direct la operații de testare, în cazul de față existând un număr de 3 asemenea semnale: Begin, COUNT7 şi Q[0]. Vom exemplifica aceasta metoda prin proiectarea unui automat de tip Moore, în acest caz ieşirile sale (reprezentate de semnalele de control $c_0,...,c_6$) depinzând doar de starea curentă.

Pasul 1: identificarea numărului de stări care acoperă descrierea comportamentală a maşinii secvențiale constituită de unitatea de control. Admițând că această descriere comportamentală are la modul general P stări, vor fi necesare $\lceil \log_2 P \rceil$ elemente de memorare. În cazul de față, acest pas este deja acoperit de lucrarea 6, fiind identificat un număr de 8 stări.

Pasul 2: se conferă celor $\lceil \log_2 P \rceil$ bistabile câte un cod. În cazul de față vor fi necesare 3 bistabile.
Pasul 3: se elaborează tabelul de stări cu ajutorul căruia vor fi derivate ecuațiile logice în formă minimizată corespunzătoare părții de logică combinațională a implementării automatului.

Tabelul de stări cuprinde cuprinde codificarea condițiilor în care au loc tranzițiile din starea prezentă în starea viitoare. Pe coloane se reprezintă starea curentă (în cazul nostru de la S_0 la S_7) iar pe linii sunt indicate toate combinațiile posibile ale semnalelor de intrare. În interiorul tabelului se indică starea următoare, conform Tabelului 7-1. De exemplu, dacă starea curentă este S_2 iar combinația semnalelor de stare este 010 atunci starea următoare va fi S_3 iar semnalul de control care se generează este c_1.

Semnale de control activate	**Starea prezentă**	**Semnale de intrare (Begin, Q(7), Count$_7$)**							
		000	001	010	011	100	101	110	111
Ø	S_0	S_0				S_1			
c_0	S_1	S_2							
c_1	S_2	S_4		S_3		S_4		S_3	
c_2	S_3	S_4							
c_3	S_4	S_4	S_6	S_3	S_5	S_4	S_6	S_3	S_5
c_2, c_4	S_5	S_6							
c_5	S_6	S_7							
c_6	S_7	S_0							

Tabelul 7-1. Tabela de stări corespunzătoare algoritmului lui Robertson

Din tabela de stări este apoi derivată tabela de tranziții. Diferența față de tabela de stări constă în faptul că aceasta utilizează o codificare simbolică a stărilor, pe când tabela de tranziții codifică stările prin intermediul unor variabile. În cele ce urmează se va utiliza codificarea pentru stări indicată în Tabelul 7-2.

Starea	Codificarea stării		
	X_0	X_1	X_2
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0
S_5	1	0	1
S_6	1	1	0
S_7	1	1	1

Tabelul 7-2. Codificarea stării pentru algoritmul lui Robertson

În acest moment este necesară derivarea tabelelor de excitație pentru bistabilele automatului, cu ajutorul cărora se vor determina ecuațiile corespunzătoare în formă minimizată. Tabelele de excitație sunt proprii fiecărui tip de bistabil.

Vom exemplifica sinteza unității de control cu ajutorul bistabilelor de tip JKMS, fiind necesară generarea a 3 seturi de câte 2 ecuații logice (câte un set pentru fiecare bistabil). Pentru aceasta se realizează tabelele de excitație pentru fiecare bistabil în parte. O celulă din tabela de excitație cuprinde starea următoare a unui bistabil, în funcție de starea prezentă şi semnalele de intrare. Pentru bistabilul care generează bitul cel mai semnificativ din codul stării (X_0) tabela de excitație este prezentată în Tabelul 7-3.

	Begin Q(7) Count$_7$							
(X_0)	000	001	011	010	110	111	101	100
000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
001	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
011	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
010	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
110	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
111	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
101	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
100	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**

$X_0X_1X_2$

Tabelul 7-3. Tabela de excitație pentru bistabilul X_0

Q(t)	Q(t+1)	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Din această tabelă de excitație se generează tabelele de ieşire pentru J_0 şi, respectiv, K_0, prezentate în Tabelul 7-4, respectiv 7-5.

Vom ține cont de faptul că bistabilele de tipul JK funcționează după tabelul de adevăr alăturat, în care d reprezintă termen *don't care*.

	Begin Q(7) Count$_7$								
		000	001	011	010	110	111	101	100
J_0	000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	001	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	011	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
$X_0X_1X_2$	010	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
	110	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	111	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	101	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	100	**d**	**d**	**d**	**0**	**0**	**d**	**d**	**d**

Tabelul 7-4. Tabela de ieşire pentru J_0

Prin minimizarea Tabelului 7-4 rezultă ecuația logică:

$$J_0 = X_1 \cdot \left(X_2 + \overline{Q[7]} \right)$$

	Begin Q(7) Count$_7$								
		000	001	011	010	110	111	101	100
K_0	000	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	001	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	011	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
$X_0X_1X_2$	010	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
	110	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	111	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	101	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	100	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**

Tabelul 7-5. Tabela de ieşire pentru K_0

Prin minimizarea Tabelului 7-5 rezultă ecuația logică:

$$K_0 = X_1 \cdot X_2 + Q[7] \cdot \overline{Count7} \cdot \overline{X_1} \cdot \overline{X_2}$$

Urmând procedura precedentă, în mod absolut similar rezultă tabela de excitație pentru X_1, prezentată în Tabelul 7-6.

	Begin Q(7) Count$_7$							
X_1	000	001	011	010	110	111	101	100
000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
001	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
011	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
010	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**
110	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
111	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
101	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
100	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**

$X_0X_1X_2$

Tabelul 7-6. Tabela de excitație pentru bistabilul X_1

Tabelele de ieşiri pentru J_1 şi K_1 sunt prezentate în Tabelul 7-7, respectiv Tabelul 7-8.

	Begin Q(7) Count$_7$							
J_1	000	001	011	010	110	111	101	100
000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
001	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
011	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
010	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
110	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
111	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
101	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
100	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**

$X_0X_1X_2$

Tabelul 7-7. Tabela de ieşire pentru J_1

Din aceste tabele, după minimizare rezultă ecuațiile logice:

$$J_1 = X_2 + X_0 \cdot \left(Q[7] \oplus Count7\right)$$

$$K_1 = X_2 + \overline{X_0} \cdot \overline{Q[7]}$$

În sfârşit, tabela de excitație pentru X_2 rezultă conform Tabelului 7-9. Tabelele de ieşiri pentru J_2 şi K_2 sunt prezentate în Tabelul 7-10, respectiv Tabelul 7-11.

K_1 / $X_0X_1X_2$	Begin Q(7) Count$_7$							
	000	001	011	010	110	111	101	100
000	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
001	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
011	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
010	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
110	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
111	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
101	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
100	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**

Tabelul 7-8. Tabela de ieşire pentru K_1

X_2 / $X_0X_1X_2$	Begin Q(7) Count$_7$							
	000	001	011	010	110	111	101	100
00**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
00**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
01**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
01**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
11**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
11**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
10**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
10**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Tabelul 7-9. Tabela de excitație pentru bistabilul X_2

J_2 / $X_0X_1X_2$	Begin Q(7) Count$_7$							
	000	001	011	010	110	111	101	100
000	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
001	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
011	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
010	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**
110	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
111	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
101	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
100	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**

Tabelul 7-10. Tabela de ieşire pentru J_2

Din Tabelele 7-10 şi 7-11, după minimizare rezultă ecuaţiile logice:

$J_2 = X_0 \cdot X_1 + \overline{X_2} \cdot Q[7] \cdot (X_0 + X_1) + \overline{X_0} \cdot \overline{X_1} \cdot Begin$

$K_2 = 1$

	Begin Q(7) Count$_7$							
K_2	000	001	011	010	110	111	101	100
000	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
001	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
011	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
$X_0X_1X_2$ 010	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
110	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**
111	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
101	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
100	**d**	**d**	**d**	**d**	**d**	**d**	**d**	**d**

Tabelul 7-11. Tabela de ieşire pentru K_2

Schema hardware care rezultă pentru unitatea de control este prezentată în Figura 7-12.

2. Metoda elementelor de întârziere (delay element method)

Metoda delay-element implică existenţa câte unui element de întârziere notat cu DE pentru fiecare bloc operativ din ordinogramă. Plecând de la descrierea din ordinogramă sau de la o descriere într-un limbaj hardware, există două particularităţi specifice acestei metode:

- De fiecare dată când în ordinogramă avem de-a face cu o reunire a mai multor ramuri, în transpunerea în circuite acest lucru va fi implementat prin intermediul unei porţi OR care va prelua respectivele ramuri ca şi intrări.
- Un bloc de decizie va fi implementat conform Figurii 7-13, în care blocul decizional apare în partea stângă iar implementarea sa apare în partea dreaptă.

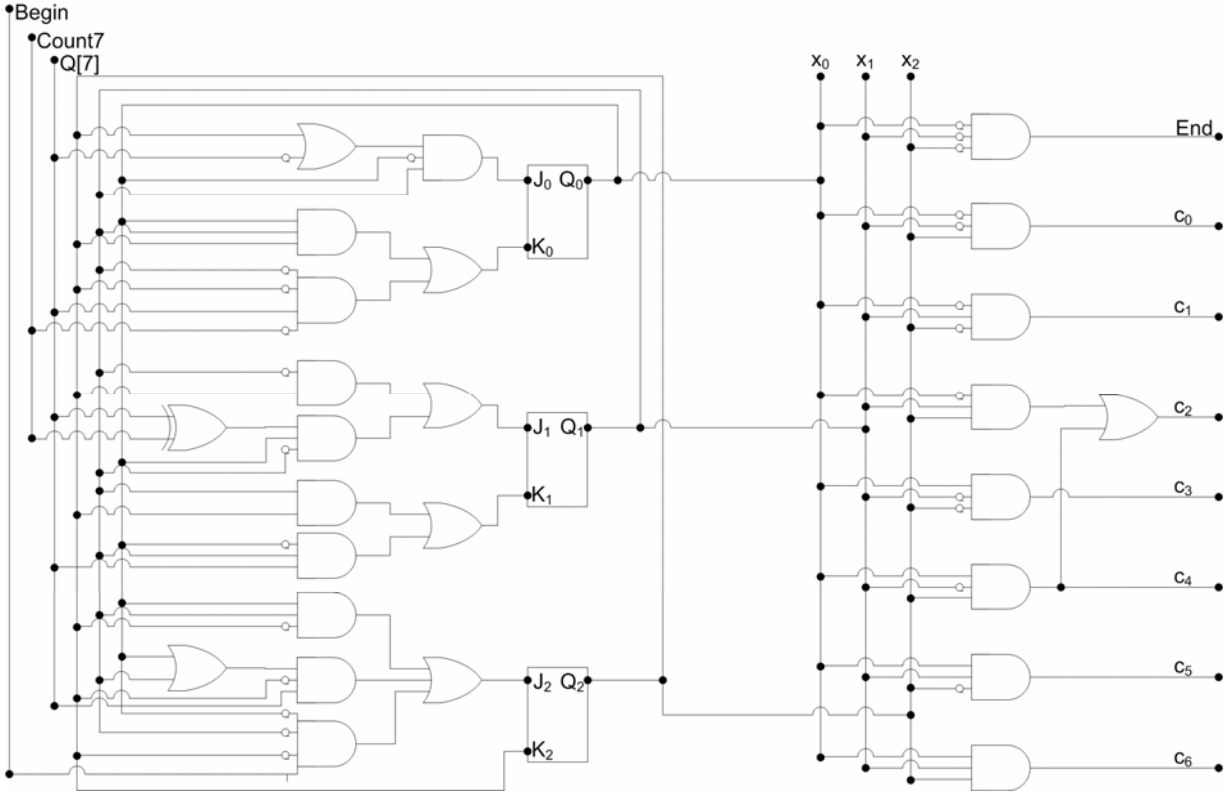

Figura 7-12. Implementarea unității de control după metoda state table

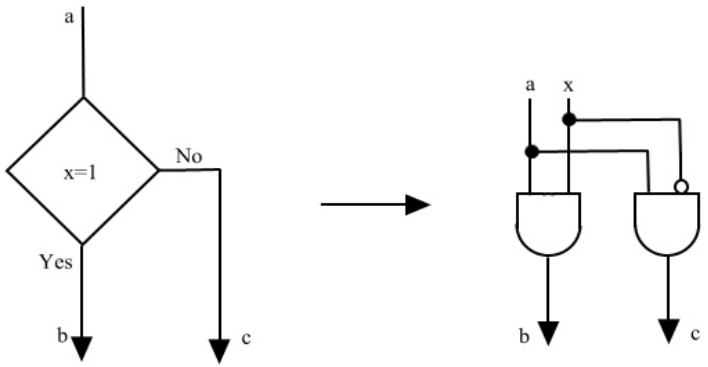

Figura 7-13. Implementarea blocurilor de decizie în metoda delay-element

Un element de întârziere DE (delay-element) nu reprezintă doar o linie de întârziere de tip pasiv ci este constituit în mod uzual dintr-un bistabil de tip D al cărui ieşire este filtrată prin semnalul de tact (Figura 7-14).

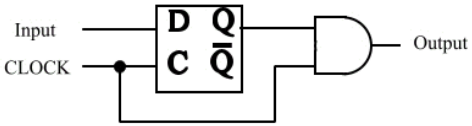

Figura 7-14. Structura unui element de întârziere DE

Toate elemente DE au un semnal comun de tact, ceea ce induce fenomenul de "alunecare de clock" (clock skew). Cu cât acest lanț de propagare al semnalului de clock este mai lung, acesta va ajunge cu întârziere, afectând elementele DE. Fiecărei stări din ordinogramă îi va corespunde câte un delay element, făcând preferabilă această metodă pentru claritatea transpunerii ordinogramei funcționale şi pentru uşurința depanării implementării. Unitatea de control pentru algoritmul lui Robertson după metoda delay-element este prezentată în Figura 7-15.

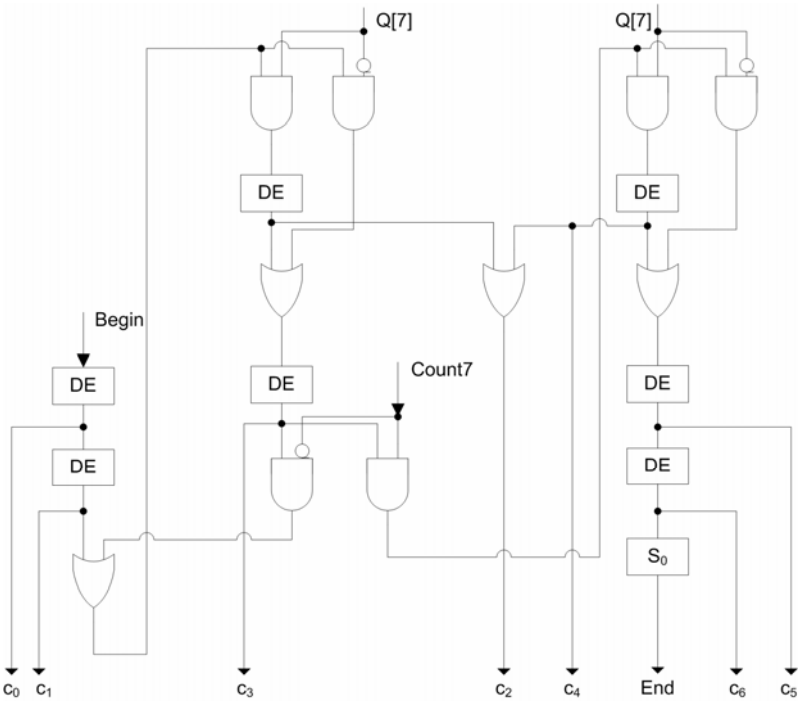

Figura 7-15. Unitatea de control pentru algoritmul lui Robertson realizată după metoda delay element

2.1 Metoda One-hot

Dacă metoda tabelei de stări prezintă avantajul unui număr minim de elemente de memorare, complexitatea părții de logică combinațională nu rezultă în mod clar, predictibil, aceasta prezentând uzual porți cu multe intrări şi conexiuni complexe, neordonate, care fac depanarea unei astfel de realizări dificilă. În consecință, metoda one-hot propune o relație de corespondență între numărul de stări din ordinogramă şi numărul de elemente de memorare. Numele metodei provine de la faptul că un singur bistabil este setat (conține valoarea logică 1), în timp ce toate celelalte sunt resetate (conțin valoarea logică 0) în orice moment. Vom ilustra proiectarea unității de control cu ajutorul acestei metode utilizând bistabile de tip D.

Numărul mai mare de elemente de memorare va fi compensat prin diminuarea în complexitate a părții de logică combinațională. Metoda permite elaborarea rapidă a ecuațiilor logice de ieşire pentru bistabile, prin identificarea condițiilor care conduc la trecerea în fiecare stare. Paşii metodei one-hot sunt următorii:

Pasul 1: identificarea numărului de stări care acoperă descrierea algoritmului şi atribuirea câte unui bistabil pentru fiecare stare.

Pasul 2: se atribuie un cod distinct fiecărui bistabil.

Pasul 3: se elaborează ecuațiile logice care determină starea de „1“ a fiecărui bistabil, răspunzându-se la întrebarea “care este condiția logică de a ajunge într-o anumită stare la momentul imediat viitor”.

Vom nota cu S_i^+ faptul că la momentul imediat viitor starea curentă va deveni S_i. În cazul algoritmului Robertson, ecuațiile logice care rezultă sunt următoarele:

$$S_0^+ = S_0 \cdot \overline{Begin} + S_7$$

$$S_1^+ = S_0 \cdot Begin$$

$$S_2^+ = S_1$$

$$S_3^+ = Q[7] \cdot \left(S_2 + S_4 \cdot \overline{Count7} \right)$$

$$S_4^+ = S_3 + \overline{Q[7]} \cdot \left(S_2 + S_4 \cdot \overline{Count7} \right)$$

$$S_5^+ = S_4 \cdot Count7 \cdot Q[7]$$

$$S_6^+ = S_5 + S_4 \cdot Count7 \cdot \overline{Q[7]}$$

$$S_7^+ = S_6$$

Schema hardware care rezultă pentru unitatea de control este prezentată în Figura 7-16.

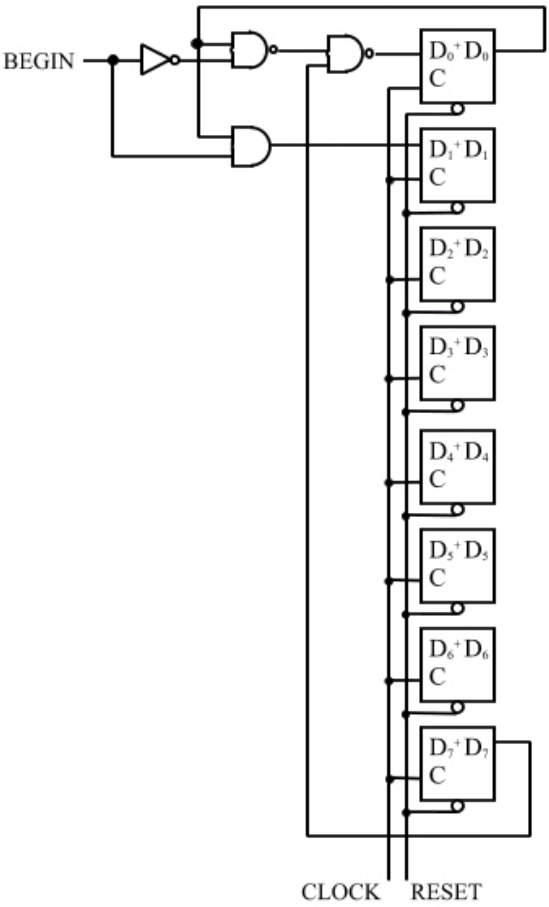

Figura 7-16. Implementarea unității de control după metoda one-hot

3. Metoda numărătorului de secvențe (sequence counter method)

Această metodă este potrivită atunci când, din punct de vedere comportamental, unitatea de control care trebuie proiectată parcurge în mod repetat cel puțin o buclă funcțională și uzitează de o schemă care poartă numele de numărător de secvență, prezentată în Figura 7-17.

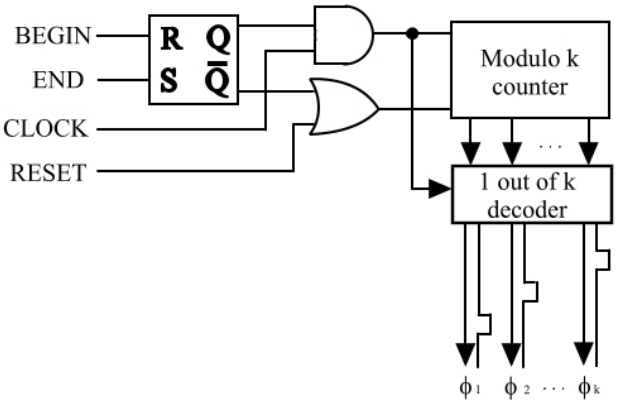

Figura 7-17. Schema de principiu a unei unități de control implementată după metoda sequence counter

Paşii care trebuie urmați în proiectare sunt următorii:

- Se analizează ordinograma în vederea identificării ciclurilor, adică a părților care se execută în mod repetat. Ciclul care prezintă cea mai lungă secvență de blocuri operative va impune numărul de faze k.
- Se partiționează ordinograma în secvențe de lungime maximă k (dacă există mai multe cicluri, fiecare astfel de ciclu va reprezenta câte o partiție), fiecare astfel de partiție fiind alocată câte unui bistabil.

În final, ordinograma va fi partiționată în n partiții (sau cicluri), fiecare desfăşurându-se pe un număr maxim de k faze.

În aceste condiții soluția pentru implementarea unității de control după metoda sequence counter este dictată de cei doi parametri, k şi n, obținerea lor fiind ilustrată în Figura 7-18. În acest caz, algoritmul prezintă un singur ciclu, reprezentat de o secvență de lungime k=2. După partiționare, numărul de cicluri rezultă n=4.

Aşadar, primul bistabil va fi setat atâta timp cât algoritmul se găseşte în stările S1 sau S2. În momentul în care algoritmul trece într-una din stările S3 sau S4, primul bistabil trebuie resetat iar al doilea bistabil va fi setat, indicându-se faptul că parcurgerea algoritmului a intrat în cel de-al doilea ciclu (denumit Cycle 1 to 8 în Figura 7-18).

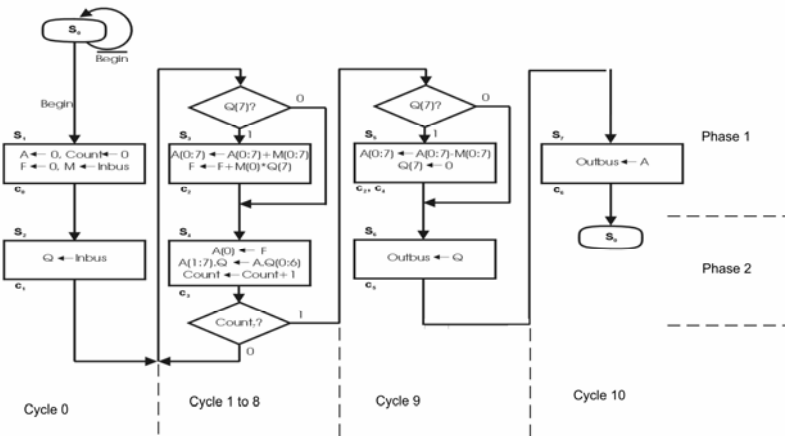

Figura 7-18. Partiționarea ordinogramei corespunzătoare algoritmului lui Robertson în cicluri şi faze

În final, implementarea unității de control după metoda sequence counter este prezentată în Figura 7-19.

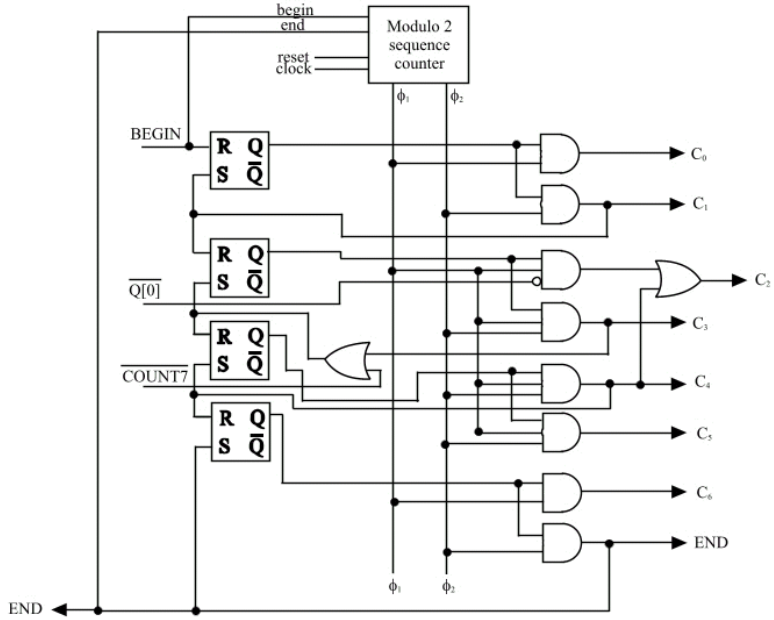

Figura 7-19. Unitatea de control pentru algoritmul lui Robertson realizată după metoda sequence counter