

Studii de caz în circuitele secvențiale

În acest capitol, vor fi ilustrate câteva dintre caracteristicile operaționale fundamentale ale rețelelor secvențiale prin considerarea a trei mari clase, foarte mult utilizate, din această categorie de rețele: registrele, numărătoarele și memoriile.

Registrele sunt alcătuite din elemente cu memorie care sunt citite sau scrise unitar.

Registrele fișier sunt mai multe registre grupate într-o singură entitate integrată.

Registre de deplasare cuprind, suplimentar față de registre, circuite care facilitează deplasarea ciclică a conținutului elementelor cu memorie.

Memoriile cu acces aleator - sunt generalizări ale registrelor fișier având foarte multe elemente de memorare.

Numărătoarele sunt registre având circuite combinaționale adiționale care permit parcurgerea, ciclică, a unei secvențe predefinite de stări. Dacă aceste stări sunt asociate unor numere binare (prin atribuirea unor ponderi convenabile elementelor cu memorie), valoarea curentă a stării poate să crească sau să scadă, numărătorul numindu-se, tradițional, crescător, respectiv, descrescător. Vor fi considerate următoarele aspecte:

- *Construcția generică a registrelor și numărătoarelor.* Utilizând bistabili ca principale elemente de construcție se vor prezenta principalele caracteristici constructive ale registrelor și numărătoarelor.
- *Proiectarea numărătoarelor.* Va fi enunțat un procedeu care va permite, pornind de la o descriere comportamentală a numărătorului, implementarea acestuia bazată pe bistabili și circuite combinaționale. Procedeu este o formă particulară, dar foarte simplă și intuitivă, a unuia mult mai general de implementare a mașinilor cu număr finit de stări.
- *Numărătoarele* disponibile, existente, ca numărătoare integrate. Vor fi trecute în revistă câteva dintre numărătoarele curent disponibile în cataloagele de circuite integrate și o parte dintre particularitățile utilizării lor.
- *Memorii cu acces aleator.* Descrierea funcționării și comportamentului în timp a componentelor secvențiale cu foarte multe elemente cu memorie.

1 Tipuri de registre și numărătoare

În această secțiune, vor fi descrise elementele fundamentale ale rețelelor secvențiale simple: registrele, numărătoarele și memoriile cu acces aleatoriu (*RAM*-urile).

1.1 Registrele

Un registru este un grup de elemente cu memorie care este citit sau scris odată, în mod unitar. Cel mai simplu mod de construire al unui registru este gruparea unui număr de bistabili D, fiecare alcătuind un rang al registrului. Toți bistabilii registrului au un mod similar de conectare. Un astfel de registru poate stoca un vector de valori binare, lungimea vectorului fiind egală cu numărul de ranguri ai registrului.

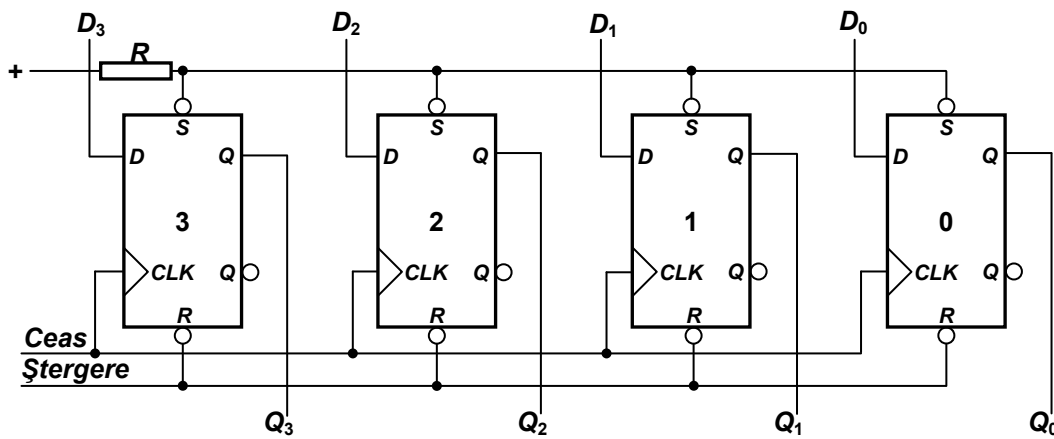


Figura 1. Registrul cu patru ranguri implementat cu bistabile D.

Fiecare bistabil din figura 1 (identificabili prin etichetele 3, 2, 1 și 0) are intrarea de ceas, activă pe frontul pozitiv al semnalului, conectată la linia comună **Ceas** și are active prin valori 0 (logică pozitivă) intrările asincrone S și R (pentru injectarea valorilor 0 și respectiv 1, în bistabile).

Deoarece intrările asincrone S din fiecare rang nu sunt utilizate, în această implementare, acestea sunt conectate printr-o rezistență, *r*, la valoarea logică 1 (plusul tensiunii de alimentare, corespunzător valorii logice 1, în logică pozitivă). Conectarea printr-o rezistență de valoare potrivită este recomandată, în general, pentru liniile de intrare legate la plusul alimentării. În acest mod se limitează curentul absorbit prin liniile de intrare respective.

Intrările asincrone R ale bistabilelor sunt reunite la borna de semnal **Ștergere** fiind folosite pentru aducerea simultană a conținutului celor patru bistabile la valoarea zero. Au fost utilizate ca ieșiri doar liniile, asertate, Q ale fiecărui bistabil și s-a procedat la etichetarea fiecărei linii de ieșire din registru corespunzător rangului respectiv (Q_3, Q_2, \dots, Q_0). De remarcat utilizarea convenției care marchează printr-un mic cerc liniile active prin tensiune coborâtă, corespunzător valorii 0 în logica pozitivă.

Figura 2 prezintă capsula circuitului integrat 74x171, un registru cu patru bistabile de tip D, având o structură similară celei prezentate în figura 1. Acest circuit este prevăzut cu următoarele linii:

- O linie de ceas (CLK), disponibilă la pinul 12;

- O linie *CLR* (abreviere, curentă, pentru *Clear*), pinul 13, care permite inițializarea cu valoarea 0 a stării celor patru bistabile;
- Liniile de intrare D_i , $0 \leq i \leq 3$, în cele patru bistabile, respectiv pinii 11, 5, 4 și 14;
- Ambele linii de ieșire (linia Q_i și cea complementată Q_i' , $0 \leq i \leq 3$) ale fiecărui bistabil, respectiv perechile de pini (10, 9), (6, 7), (3, 2) și (15, 1).

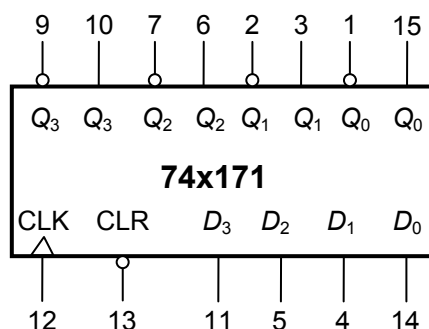


Figura 2. Conexiunile registrului integrat 7x 171.

Circuitele, precum cel din figura 1, sunt principala componentă a aproape tuturor registrelor integrate. Există, însă, multe variații ale schemei de bază, de obicei incluzând semnale de control adiționale pentru intrări și ieșiri. Astfel, o extensie poate fi implementată printr-un semnal de *încărcare*. Logica internă a registrului păstrează starea curentă, indiferent de tranzițiile impulsului ceas, atâta timp cât nu este asertată încărcarea. Atunci când aceasta este asertată, noile valori ale liniilor de intrare ajung la intrările interne ale bistabililor. Astfel, se înlocuiește starea curentă a fiecărui bistabil atunci când apare un impuls de ceas. O altă extensie a acestor circuite integrate permite ieșirilor registrului să aibă trei stări (*tri-state*) sau să fie cu colectoarele deschise (tehnologia bipolară *open-collector*, spre exemplu). Logica externă unui astfel de registru permite plasarea valorilor ieșirilor acestuia pe conexiuni partajate sau magistrale.

Componenta integrată 74x377, spre exemplu, conține opt circuite bistabile de tip *D* prevăzute cu o linie de validare, numită curent, *Enable* (permite). Denumirea acestei linii este abreviată *EN*. Este un registru alcătuit din opt elemente de memorie constituite din bistabile *D* acționate pe frontul pozitiv al semnalului de ceas (conexiunile la pinii capsulei sunt prezentate, schematic, în figura 3(a)). Semnalul de intrare *EN*, activ prin valori coborâte, are rolul validării intrărilor bistabilelor. Când semnalul de validare are valoare zero (logică pozitivă), sincron cu frontul pozitiv al impulsului de ceas, bistabilii încarcă valorile aflate pe liniile D_7 , D_6 , ..., D_0 . Altfel, semnalul de validare are valoarea unu, conținutul curent al bistabilelor este păstrat chiar dacă au loc tranziții ale impulsului de ceas.

Componenta integrată 74x374, numită *opt bistabile de tip D cu ieșiri tri-state*, conține o linie adițională activă prin valori coborâte (așa cum se poate vedea în figura 3 (b)). Când acest semnal este activ (are valoarea zero în logică pozitivă), conținutul registrului este vizibil la ieșiri. Altfel, ieșirile sunt în starea de înaltă impedanță.

Fișierele registre. Dacă registrele grupează mai multe elemente cu memorie într-o unitate funcțională, atunci fișierele registre exind gruparea la mai multe registre. Fiecare registru din fișierul de registre este numit *cuvânt* fiind identificat printr-un unic index numit și *adresă*.

Aceste componente conțin propriile decodoare interne. În conjuncție cu o adresă extern specificată, decodoarele selectează un anumit cuvânt din fișierul de registre pentru operația de scriere. La o operație de citire, cuvântul selectat este multiplexat spre ieșiri.

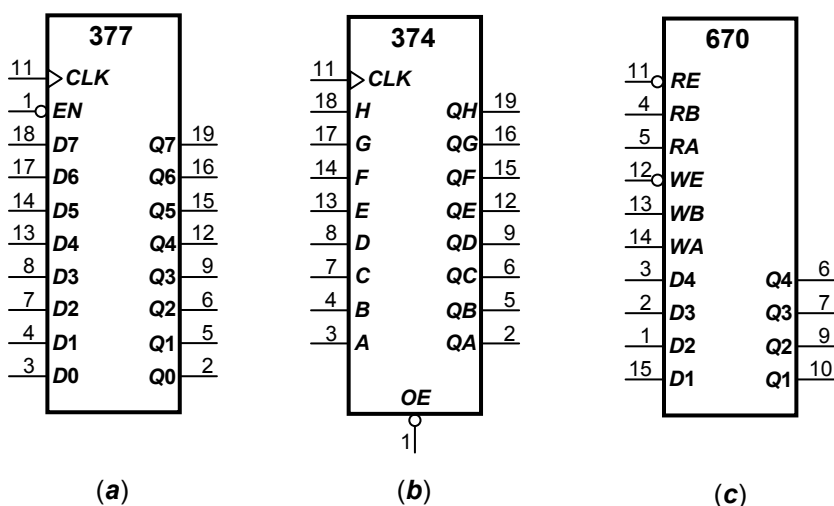


Figura 3. (a) 7x377 registrul cu liniile de intrare validate;
 (b) 7x374 registrul cu ieșiri tri-state;
 (c) fișierul de 4x4 registre cu ieșiri tri-state.

Circuitul 74x670, spre exemplu, a cărui capsulă este înfățișată în figura 3 (c), este un fișier de 4x4 registre cu ieșiri tri-state. Acest dispozitiv conține 16 bistabile *D*, organizate în patru cuvinte, a câte patru bistabile fiecare. Liniile de intrare *RB*, *RA* și *WB*, *WA* codifică prin valori binare cu câte două ranguri selecția pentru citire și respectiv scriere a unuia din cele patru registre. Liniile de intrare: *Read Enable (RE)* și *Write Enable (WE)* sunt active prin valori coborâte. Din cauza semnalelor de activare independente este posibilă, simultan, scrierea și citirea acestui fișier de registre (dar, bineînțeles, pentru registre cuvânt diferite). Este interesant de remarcat că acest dispozitiv nu are un semnal de ceas individual. Acesta folosește semnalul *WE* și valorile biților de adresă pentru a să genereze intrările de ceas ale bistabilelor interne. Deoarece componentei îi lipsește un ceas explicit aceasta trebuie folosită cu precauție. Liniile adresei de scriere trebuie să fie stabile înainte de apariția semnalului *WE*, altfel se poate scrie, în mod eronat, un cuvânt. Orice semnal parazit care apare pe linia *WE* poate, de asemenea, să producă scrieri nedorite.

Memoriile cu acces aleator. Registrele oferă modalități convenabile de stocare a unor volume mici de informație, în mod curent, între 4 și 16 biți. Cu toate acestea multe sisteme digitale necesită substanțial mai multe elemente de memorare decât acestea. Memoriile cu acces aleator (RAM) oferă, în acest sens, o soluție. Prin utilizarea de metode foarte eficiente pentru implementarea elementelor de memorare

prin tranzistoare, memoriile cu acces aleator generalizează conceptul de fișier de registre făcând disponibile mult mai multe cuvinte într-un singur circuit integrat. O astfel de memorie, mai mică, poate avea 256 cuvinte de câte 4 biți (în total 1024 de elemente de memorare) iar un dispozitiv modern, sofisticat, conține mai mult de 4 milioane de elemente de memorare (sunt curent întâlnite dispozitive cu 16 milioane de biți).

Memoriile cu acces aleator foarte rapide dar de capacitate mică sunt la fel de dificil de proiectat ca și memoriile foarte mari, acestea din urmă fiind, însă, mai lente.

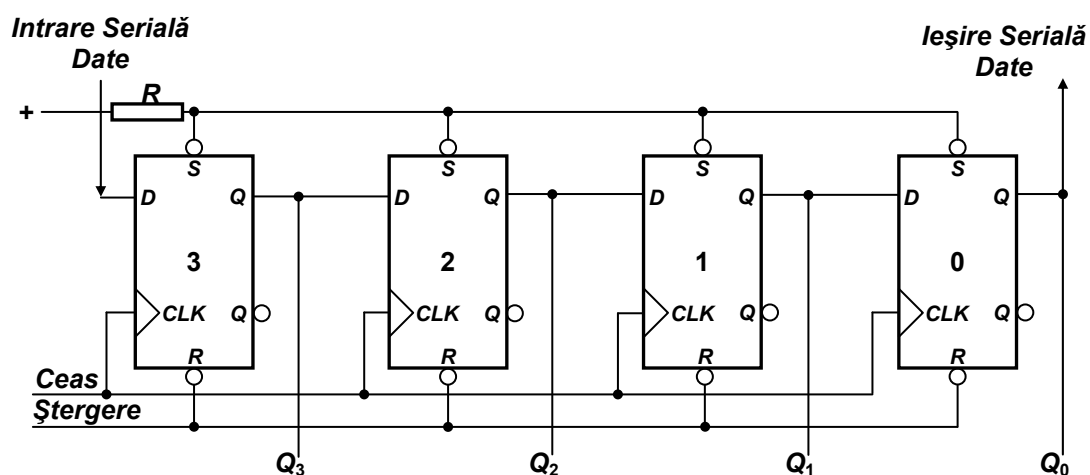


Figura 4. Un registru de deplasare cu patru ranguri implementat prin bistabile D.

1.2 Registrele de deplasare

Aceste registre pot fi folosite și pentru alte aplicații, nu numai pentru stocarea unor vectori de valori binare. Sunt adesea folosite pentru a circula, sau deplasa, valorile din elementele de memorare constitutive ale registrului. În această subsecțiune abordarea se va orienta asupra componentelor registre care au atât facilități de memorare cât și de deplasare a informației. Astfel de registre sunt numite registre de deplasare.

Figura 4 prezintă logica unui registru de deplasare circulară spre dreapta construit cu patru bistabili de tip D. Datele circulă, în interiorul acestui registru, de la stânga la dreapta. La fiecare impuls de ceas, conținutul bistabilelor 2, 1 și 0 este înlocuit prin conținutul bistabilului respectiv aflat în stânga sa. Registrul este încărcat cu date prin linia serială de date conectată la intrarea D a bistabilului 3 (numită în figura 5 prin *Intrare Serială Date*). Registrul oferă posibilitatea conectării ieșirii sale seriale (notată prin *Ieșire Serială Date* în figura 4) cu un dispozitiv digital, eventual cu propria intrare serială de date. Așa cum a fost conectat, acest registru are facilitatea inițializării celor patru ranguri (bistabile) prin valoarea 0 (Linia *Ștergere* activă prin valori zero).

În figura 5 se poate vedea o variantă a registrului de deplasare din figura 4. Acesta este astfel conectat încât să poată fi încărcat cu date în paralel în toate rangurile sale.

Încărcarea paralelă se face în două etape. În prima etapă, întreg registrul este inițializat prin valori zero acționând asupra liniei Ștergere (linia Ștergere este activă prin valoarea zero). Cea de-a doua etapă încarcă, în paralel, selectiv, doar valorile nenule în registrul de deplasare (liniile de intrare I_3' , I_2' , I_1' și I_0' sunt active prin valori zero).

În această configurație, așa cum este conectat registrul, bistabilul aflat în capătul din stânga, al registrului (rangul 3), primește la intrările sale date de la bistabilul aflat în capătul din dreapta (rangul 0). Aceasta conexiune constituie deplasarea circulară a informației binare din registru.

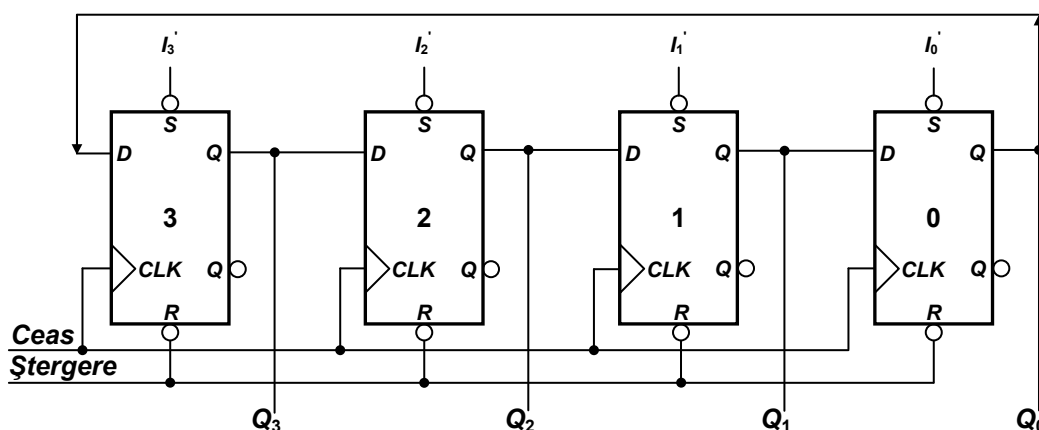


Figura 5. Un registru de deplasare circulară a datelor, având încărcarea paralelă.

Bistabilul aflat în capătul din stânga, al registrului, primește la intrările sale date de la bistabilul aflat în capătul din dreapta. Aceasta constituie deplasarea circulară a informației binare din registru.

Datorită faptului că timpul de propagare a unui bistabil este mai mare decât timpul de păstrare, valorile stocate sunt deplasate corect de la un rang la altul al registrului.

Operația de deplasare pas cu pas a registrului este ilustrată grafic în figura 6 pentru o configurație inițială cu valorile: $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$, $Q_4 = 0$ urmată de aplicarea unei secvențe de trei impulsuri ceas, pentru deplasarea informației în registrul respectiv.

Ceas	Q_3	Q_2	Q_1	Q_0
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0
	0	0	0	1

Figura 6. Deplasarea datelor în registrul de deplasare din figura 5.

Registrul de deplasare cu utilizări multiple

Pentru a ilustra acest concept se va considera registrul 74x194, numit și *registrul universal, bidirecțional de deplasare cu patru ranguri*.

MODURILE DE UTILIZARE ALE CIRCUITULUI SN 74/54194													
		INTRĂRI								IEȘIRI			
C	Mod	C	Serial		Paralel								
L	S_1	S_0	L	LSI	RSI	A	B	C	D	Q_a	Q_b	Q_c	Q_d
R		K											
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	↑	X	X	X	X	X	X	Q_{a0}	Q_{b0}	Q_{c0}	Q_{d0}
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	Q_{a0}	Q_{b0}	Q_{c0}
H	L	H	↑	X	L	X	X	X	X	L	Q_{a0}	Q_{b0}	Q_{c0}
H	H	L	↑	H	X	X	X	X	X	Q_{b0}	Q_{c0}	Q_{d0}	H
H	H	L	↑	L	X	X	X	X	X	Q_{b0}	Q_{c0}	Q_{d0}	L
H	L	L	X	X	X	X	X	X	X	Q_{a0}	Q_{b0}	Q_{c0}	Q_{d0}

Figura 8 prezintă simbolul și tabelul de funcționare al registrului universal 194. Acest dispozitiv poate opera în patru moduri distincte, determinate de valorile intrărilor de control S_1 și S_0 :

- Păstrează datele ($S_1S_0 = 00$),
- Deplasează spre dreapta, Q_A spre Q_D , ($S_1S_0 = 01$),
- Deplasează spre stânga, Q_D spre Q_A , ($S_1S_0 = 10$), și
- Încarcă paralel de la liniile de intrare A, B, C, D ($S_1S_0 = 11$).

Registrul are, adițional, un semnal asincron de inițializare *CLR* (activ prin valori coborâte) care încarcă valoarea 0 în toate bistabilele, și două linii de intrare seriale de deplasare denumite LSI și RSI.

Încărcarea paralelă este condiționată de valorile liniilor ($S_1S_0 = 11$) și are loc pe durata frontului crescător a impulsului de ceas. În același timp, valoarea aflată pe linia de intrare A este stocată în bistabilul Q_A , valoarea aflată pe linia de intrare B este stocată în bistabilul Q_B și așa mai departe. Această operație poartă numele de *încărcare sincronă* deoarece se produce sincron cu apariția impulsului de ceas.

Valorile liniilor de selecție pentru deplasarea la dreapta ($S_1S_0 = 01$) produc efectiv o deplasare la dreapta a conținutului registrului la apariția frontului pozitiv al impulsului de ceas. Pe frontul crescător al impulsului de ceas, valoarea pe linia de intrare *RSI* este încărcată în Q_A , valoarea din Q_A se încarcă în Q_B , valoarea din Q_B se stochează în Q_C iar valoarea din Q_C trece în Q_D . Valoarea anterioară din Q_D se va pierde. Se poate construi un registru cu deplasare circulară spre dreapta prin conectarea ieșirii Q_D la intrarea RSI.

Dacă $S_1 = 1$ și $S_0 = 0$ atunci se specifică o deplasare la stânga. În acest caz, valoarea liniei LSI se încarcă în Q_D , valoarea anterioară din Q_D se stochează în Q_C , vechea valoare din Q_C se va deplasa în Q_B , iar valoarea inițială din Q_B trece în Q_A , toate aceste deplasări având loc pe durata frontului crescător a impulsului de ceas. Se poate implementa, similar modului descris anterior, un registru de deplasare spre stânga, circular, prin conectarea liniei Q_A cu linia LSI.

Dacă atât linia S_1 cât și linia S_0 sunt zero, atunci registrul de deplasare păstrează starea curentă. Liniile de ieșire nu-și schimbă valorile chiar dacă au loc tranziții pozitive ale semnalului de ceas.

O aplicație simplă a registrelor de deplasare. Se consideră problema comunicării, printr-o linie telefonică, dintre un calculator și un terminal. Terminalul așteaptă să apară datele în format paralel, octet după octet. Dar datele sunt trimise de la calculator pe linie în mod serial. Registrele de deplasare joacă un rol cheie în asemenea sisteme de comunicație deoarece acestea pot face conversia dintre formatele paralel și serial. Se va puncta proiectarea acestei conversii utilizând circuitul SN 74/54194. O implementare parțială a acestui subsistem este arătată în figura 9.

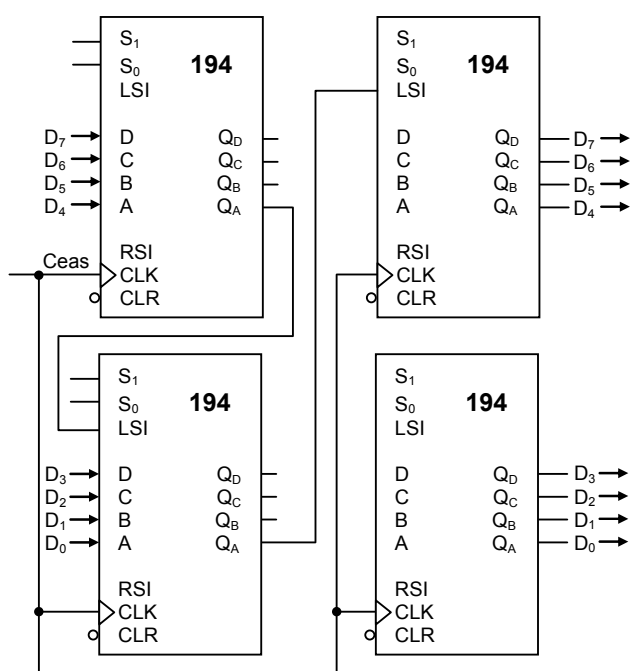


Figura 9. Transmisie serială – paralelă pe 8 biți realizată cu circuitul SN 74/54194.

Pe partea emițătorului, sunt încărcate cuvinte de date având 8 biți (octeți) în paralel în registrul de deplasare, conectat cascadat, prin aducerea ambelor linii S_1 , S_0 la valoarea 11, ($(S_1S_0) = (11)$).

Atât emițătorul cât și transmițătorul vor intra în modul de deplasare spre stânga atunci când $(S_1S_0) = (01)$. Deplasarea are loc pe durata a opt impulsuri de ceas, transmițând D_0, D_1, \dots , până la D_7 , câte un bit o dată pentru fiecare impuls de ceas.

De îndată ce toți cei opt biți au fost trimiși, se poate plasa receptorul în starea de așteptare aducând liniile (S_1S_0) la valorile (00).

1.3 Numărătoarele

Numărătoarele sunt circuite logice secvențiale care funcționează printr-o secvență predefinită de stări.

Numărătoarele crescătoare și descrescătoare. Un numărător crescător pornește de la o valoare anumită spre o valoare mai mare în secvența răspuns la impulsurile de numărare. Ieșirile numărătorului sunt identice pentru starea bistabililor interni. Odata ce cea mai mare valoare posibilă a fost atinsă, secvența se resetează la valoarea cea mai mică. De exemplu, un numărător crescător, binar de trei biți începe în starea 000, trece prin secvența: 001, 010, 011, 100, 101, 110, 111; și se întoarce la: 000, 001, și așa mai departe.

Asa cum se poate ușor imagina, un numărător descrescător funcționează în mod similar, exceptând faptul că secvența de valori descrește de la valori mari, la valori mici. Numărătorul ciclează de la valoarea cea mai mică posibilă înapoi la cea mai mare valoare din secvență.

Numărătoarele binare, decadice și în cod Gray. Chiar dacă au fost considerate, până acum, doar numărătoarele binare, un numărător poate fi proiectat să treacă, succesiv, prin orice secvență periodică. Alternative cunoscute ale numărătoarelor binare sunt numărătoarele decadice. Acestea trec prin stări binare codificând valori zecimale: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000 și 1001.

În anumite aplicații, pentru evitarea apariției hazardurilor, este important că numai un singur bit al numărătorului să se schimbe la un moment dat. Numărătoarele binare și decadice nu au această proprietate. Un numărător în cod Gray are această însușire. Un numărător crescător, cu 4 biți în cod Gray va trece prin stările: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000 după care secvența se reia.

Numărătoarele în inel. Implementarea registrului de deplasare din figura 4 poate fi deasemenea folosită ca un tip de numărător primar, numit *numărătorul în inel*. Registrul de deplasare trece prin stările: 1000, 0100, 0010, 0001 și apoi se repetă. Numărătorul în inel folosește un minimum de resurse pentru implementarea sa, dar nu codează stările la fel de eficient ca numărătoarele descrise anterior. Numărătorul în inel cu patru bistabile trece prin doar patru stări, în comparație cu cele 16 stări ale numărătorului binar cu 4 elemente.

Un *numărător Johnson* (cunoscut, deasemenea, sub denumirea *numărător Mobius*) necesită la fel de multe resurse ca și numărătorul în inel dar poate prezenta de două ori mai multe stări. Ideea este să se răsucescă ieșirile de reacție ale bistabilului din extrema dreaptă pentru producerea unui deplasator și numărător în inel. Această schimbare completează bitul deplasat în afară, prin capătul din dreapta înainte să fie re-inserat în registrul de deplasare stânga - dreapta.

Se va considera o implementare a numărătorului Johnson cu patru ranguri implementat cu bistabile D master/slave ca în figura 10.

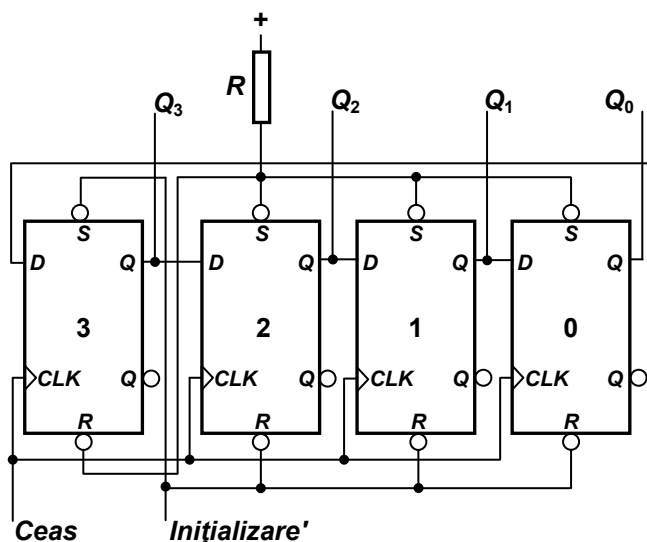


Figura 10. Un numărător Johnson cu patru ranguri implementat prin bistabile *D*.

Se folosesc liniile de intrare asincrone *Set (S)* și *Reset (R)* pentru forțarea bistabilelor în starea inițială 1000 când este aplicată comanda de inițializare (***Inițializare'***) a numărătorului (comanda este activă prin valoarea logică zero).

Rezistorul *R* (de ordinul 1 Kohm) este utilizat pentru inactivarea intrării *Reset* a bistabilului 3 dar și pentru inactivarea intrărilor *Set* ale bistabilelor 2, 1 și 0.

Stările succesive ale numărătorului Johnson sunt prezentate în diagrama din figura 11.

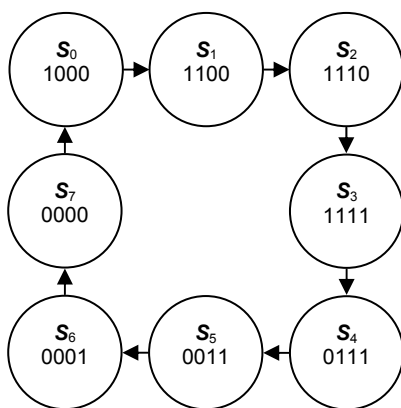


Figura 11. Diagrama completă a stărilor circuitului din figura 10.

Numărătorul trece prin stările: 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000 și repetă, apoi, secvența. Se poate remarca faptul că această secvență satisface aceeași proprietate ca și codul Gray - doar un singur bit își schimbă valoarea de la o stare la alta.

PROIECTAREA NUMĂRĂTOARELOR SINCRONE. Numărătoarele sunt printre cele mai simple automate cu stări finite. Vor fi considerate, în continuare, numărătoarele sincrone. Numărătoarele sincrone își schimbă starea la fiecare impuls de ceas. Pentru o mai ușoară înțelegere s-a ales, ca în cele ce urmează, să se analizeze un numărător binar, crescător, cu trei ranguri. Fiecare stare a acestui numărător se va identifica prin valorile binare ale celor trei ranguri componente ale numărătorului. Diagrama graf a tranziției stărilor este prezentată în figura 12.

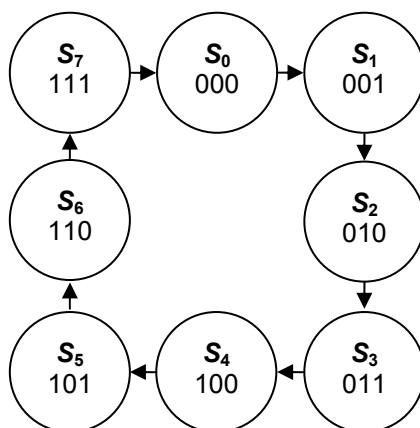


Figura 12. Diagrama de stări a unui numărător binar cu trei ranguri.

Echivalentă, funcțional, diagramei de stări din figura 1 este descrierea tranzițiilor prezentată în tabelul 1. Descrierea tabelară a tranzițiilor unui automat se numește *tabelul tranzițiilor stărilor* sau, mai scurt, *tabelul stărilor*.

Tabelul 1.

Tranzițiile numărătorului binar cu trei ranguri							
Starea curentă	Starea viitoare			Starea viitoare	Starea viitoare		
	Q^t				Q^{t+1}		
	Q_2	Q_1	Q_0		Q_2	Q_1	Q_0
S_0	0	0	0	S_1	0	0	1
S_1	0	0	1	S_2	0	1	0
S_2	0	1	0	S_3	0	1	1
S_3	0	1	1	S_4	1	0	0
S_4	1	0	0	S_5	1	0	1
S_5	1	0	1	S_6	1	1	0
S_6	1	1	0	S_7	1	1	1
S_7	1	1	1	S_0	0	0	0

Fiecare linie din tabelul stărilor corespunde unei tranziții din diagrama de stări (graful stărilor). Fiecare rang al stării numărătorului corespunde unui bistabil. Pentru acest numărător vor fi necesare trei bistabile, notate prin Q_0 , Q_1 și Q_2 .

Din rațiuni de simplitate, ale acestui exemplu, s-a ales ca numărătorul să fie implementat prin bistabile D . Motivul este evident, bistabilul D are proprietatea că la fiecare impuls de ceas starea sa viitoare (0 sau 1) este determinată exclusiv prin valoarea binară aplicată liniei sale de date (D).

Circuitul logic al stării viitoare determină, în funcție de valorile curente ale bistabililor numărătorului, valorile viitoare ale fiecărui bistabil al numărătorului. Valorile viitoare ale stărilor vor fi încărcate în bistabilele respective la următorul impuls de ceas.

		Q ₂	
		0	1
Q ₁ Q ₀	00		1
	01		1
	11	1	
	10		1

$$Q_2^+ = Q_2' Q_1 Q_0 + Q_2 Q_1' + Q_2 Q_0'$$

Figura 13 (a).

		Q ₂	
		0	1
Q ₁ Q ₀	00		
	01	1	1
	11		
	10	1	1

$$Q_1^+ = Q_1' Q_0 + Q_1 Q_0'$$

Figura 13 (b).

		Q ₂	
		0	1
Q ₁ Q ₀	00	1	1
	01		
	11		
	10	1	1

$$Q_0^+ = Q_0'$$

Figura 13 (c).

Dacă pentru implementarea acestui numărător sunt utilizate bistabile $J-K$ atunci este util ca tabelul stărilor să fie extins cu coloanele liniilor $J-K$ pentru fiecare bistabil. Completarea acestor coloane trebuie să se ia în considerație modul specific de funcționare al acestui tip de bistabil.

În tabelul 2 este descrisă asocierea valorilor liniilor de date J și K ale bistabilelor $J-K$ în funcție de tranziția stărilor unui astfel de bistabil.

Tabelul 2.
Valorile liniilor de date J și K ,
în funcție de tranziția stărilor
unui bistabil J - K

Q^t	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

În tabelul 3 este prezentată tranziția stărilor împreună cu valorile liniilor J și K pentru fiecare din cele trei bistabile ale numărătorului binar crescător sincron.

Tabelul 3.
Tranzițiile numărătorului binar cu trei ranguri
cu extensia necesară implementării prin bistabile J - K

Starea curentă				Starea viitoare									
Q^t				Q^{t+1}									
Q_2	Q_1	Q_0		Q_2	J_2	K_2	Q_1	J_1	K_1	Q_0	J_0	K_0	
S_0	0	0	0	S_0	0	0	X	0	0	X	1	1	X
S_1	0	0	1	S_1	0	0	X	1	1	X	0	X	1
S_2	0	1	0	S_2	0	0	X	1	X	0	1	1	X
S_3	0	1	1	S_3	1	1	X	0	X	1	0	X	1
S_4	1	0	0	S_4	1	X	0	0	0	X	1	1	X
S_5	1	0	1	S_5	1	1	0	1	1	X	0	1	X
S_6	1	1	0	S_6	1	X	0	1	X	0	1	1	X
S_7	1	1	1	S_7	0	X	1	0	X	1	0	X	1

Sunt frecvente situațiile în care un numărător parcurge o secvență de stări mai scurtă, cu mai puține stări, decât numărul maxim posibil corespunzător numărului de bistabile utilizat.

Se consideră, în acest sens, un numărător arbitrar specificat prin graful tranzițiilor de stare din figura 14.

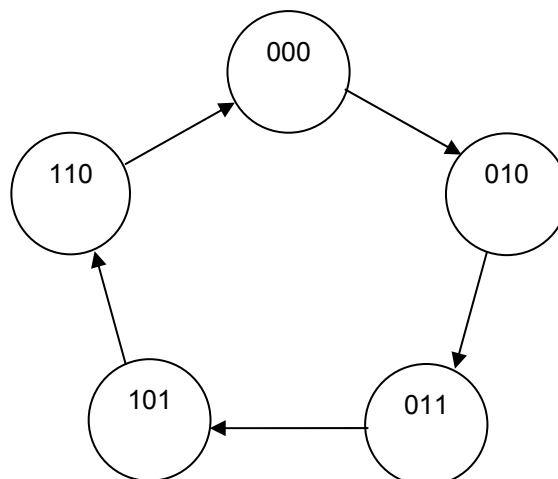


Figura 14. Diagrama de tranziție a stărilor unui numărător sincron arbitrar.

Tabelul 4.
Tranzițiile numărătorului sincron binar
arbitrar cu trei ranguri

Starea curentă Q^t			Starea viitoare Q^{t+1}		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	1	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	-	-	-
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	-	-	-

Se presupune o implementare cu bistabile D , pentru acest numărător sincron binar arbitrar.

		Q_2	
		0	1
$Q_1 Q_0$	00		x
	01	x	1
	11	1	x
	10		

$$Q_2^+ = Q_0$$

Figura 15 (a).

		Q_2	
		0	1
$Q_1 Q_0$	00	1	x
	01	x	1
	11		x
	10	1	

$$Q_1^+ = Q_1' + Q_0' Q_2'$$

Figura 15 (b).

		Q_2	
		0	1
$Q_1 Q_0$	00		x
	01	x	
	11	1	x
	10	1	

$$Q_0^+ = Q_1 Q_2'$$

Figura 15 (c).

Se poate ușor remarca faptul că nu toate combinațiile posibile de trei biți reprezintă stări valide. Stările neutilizate (001, 100 și 111) pot servi drept condiții neprecizate și pot aduce avantaje în procesul de simplificare.

Profitând de termenii neprecizați (prin interpretarea de stărilor neutilizate) în figura 15 sunt arătate diagramele Karnaugh corespunzătoare și rezultatele simplificării logice pentru cele trei variabile de stare.

Privitor la funcționarea reală a circuitelor secvențiale trebuie făcută precizarea că atunci când nu sunt prevăzute circuite, dedicate inițializării acestor circuite, starea inițială a acestora după punerea sub tensiune poate fi supusă hazardului. În cazul circuitelor secvențiale care utilizează toate codurile de stare acest fapt poate fi de mai mică importanță.

Circuitele secvențiale cu stări neutilizate explicit, dar implicit utilizate ca fiind neprecizate pot prezenta, la inițializare, situații total nedorite.

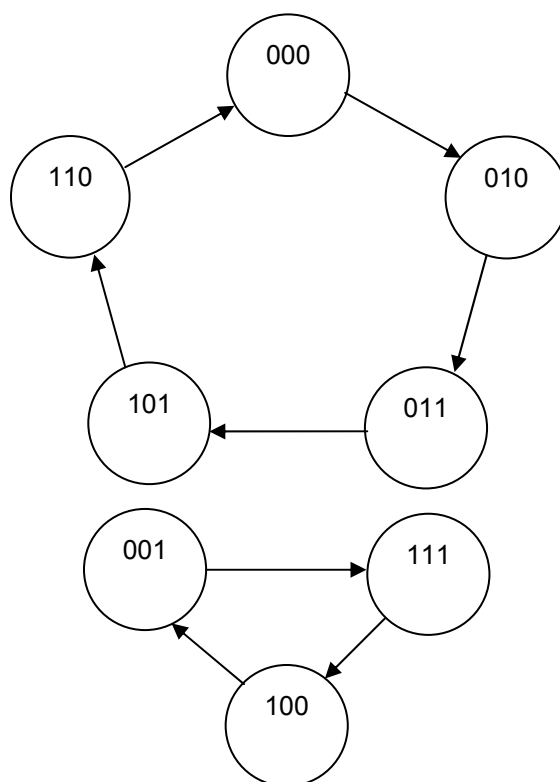


Figura 16. Diagrama de tranziție a stărilor unui numărator sincron având două cicluri de stări.

Se consideră, spre exemplu, un numărator a cărui diagrama de tranziție este prezentată în figura 16. Se deosebesc două inele, cicluri, de stări.

Primul inel de stări corespunde funcționării specificate, nominale a acestui numărator. Cel de-al doilea, constituit din doar trei stări, este nedorit dar este constituibil, posibil, din modul în care au fost utilizate codurile stărilor neprecizate.

De remarcat că dificultatea problemei rezidă în faptul că odată inițiată o stare din cel de-al doilea ciclu, automatul rămâne captiv în acest ciclu – funcționând într-o manieră absolut neconformă specificațiilor.

Se poate vedea proiectarea numărătorului în care au fost utilizate codurile unor stări neprecizate pentru ca să se poată stabili condiția extinsă de funcționare a acestuia.

O metodă simplă poate fi inițiată pornind de la diagramele Karnaugh și prin actualizarea tabelului tranzițiilor corespunzător modului în care au fost transformați termenii neprecizați în unități (cei incluși în contururile de minimizare).

Diagrama graf extinsă, cuprinzând și acești termeni neprecizați este prezentată în figura 17.

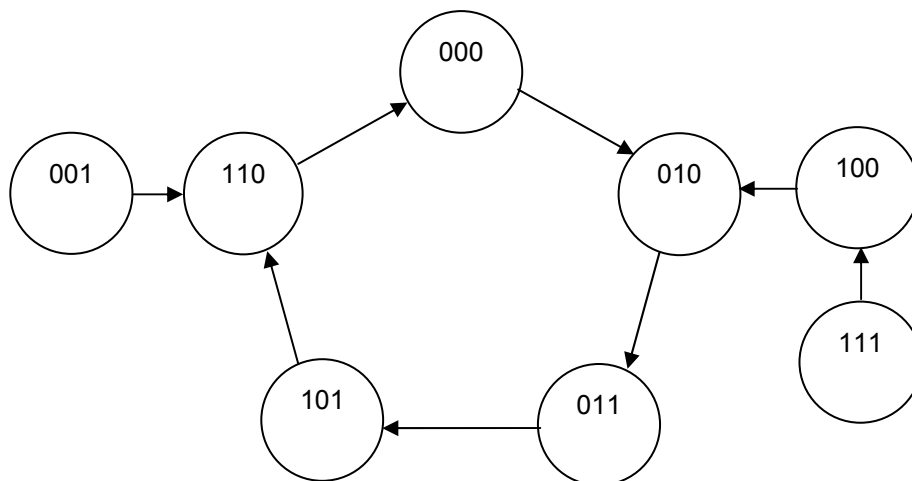


Figura 17. Diagrama de tranziție a stărilor unui numărător sincron având incluse și stările neutlizate.

Pentru proiectarea realizată pentru numărătorul respectiv s-a dovedit că și în situația în care numărătorul se inițializează într-una din stările neutlizate, după ce mult două impulsuri de ceas acesta reintră în funcționarea nominală.

Numărătoarele pot fi proiectate să funcționeze atât crescător cât și descrescător. Astfel de numărătoare sunt numite *numărătoare reversibile*. Exemplul care urmează ilustrează acest mod de proiectare.

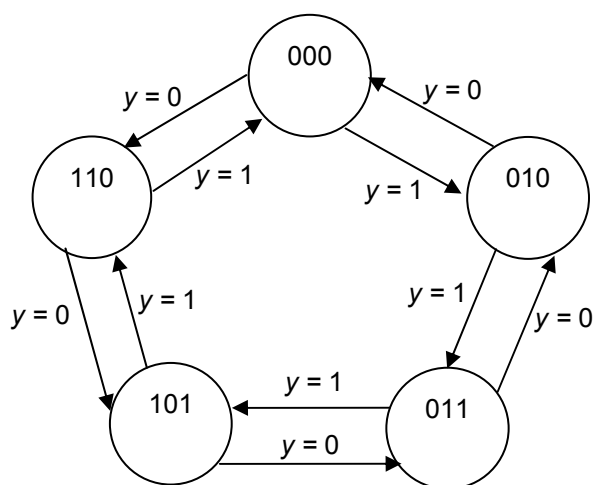


Figura 18. Diagrama de tranziție a stărilor numărătorului sincron arbitrar reversibil.

Numărătorul sincron arbitrar reversibil din figura 18 tranzitează dintr-o stare într-alta guvernat prin valoarea variabilei y .

Pentru valoarea 1 a variabilei y , diagrama de stări din figura 18 este parcursă crescător (în sensul acelor ceasornicului) în timp ce pentru valoarea 0, a variabilei y , diagrama de stări, din aceeași figură, este parcursă descrescător (în sens invers acelor ceasornicului).

Tabelul 5.
Tranzițiile numărătorului sincron binar
arbitrar, reversibil, cu trei ranguri

Starea curentă Q^t			Starea viitoare Q^{t+1}		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
$y = 1$					
0	0	0	0	1	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	-	-	-
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	-	-	-
$y = 0$					
0	0	0	1	1	0
0	0	1	-	-	-
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	-	-	-
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	-	-	-

Un tabel de tranziții al stărilor acestui numărător, cu bistabile $J-K$, este alcătuit în aceeași manieră prin care s-a determinat, anterior, tabelul 3.

Proiectarea și optimizarea acestui numărător este un exercițiu tipic al acestor circuite secvențiale.