

Cap.1 Definitia si structurarea sistemelor digitale

1.1. Sisteme digitale (numerice).

Sistemele digitale sint sisteme care prelucreaza semnale binare numite biti.

Bitul este unitatea elementara de masura a informatiei.

Denumirea "bit" provine din prescurtarea cuvintelor englezești binary digit, insemnand cifra binara.

Bitii se noteaza uzual cu litere A, B, C... {0, 1}.

Nivelurile logice "0" si "1" au asociate valori distincte ale unor marimi electrice ca: tensiuni, curenti, sarcini. Prin distantarea acestor valori asociate pentru "0" si "1", efectele zgomotelor in sistemele digitale sint mult diminuate fata de cazul sistemelor analogice care prelucreaza semnale continue. In plus, calculul numeric poate fi realizat cu o precizie mult mai mare decit cel analogic. Datorita in principal acestor avantaje, sistemele digitale domina astazi electronica.

In aceasta lucrare se vor prezenta, cu citeva exceptii, sisteme caracterizate de corespondenta **valorilor logice cu niveluri de tensiune**.

Pentru a intelege mai bine evolutia sistemelor digitale, unitatea, dar, in acelasi timp, diversitatea acestora este necesara prezentarea unor criterii de structurare.

1.2 Bazele teoretice ale structurarii sistemelor digitale

Fie S_n multimea sistemelor de ordinul n , caracterizate printr-un set de proprietati fundamentale comune.

Elementele multimii S_n sint sisteme diferiteiate intre ele prin anumite particularitati de comportament. Aceste sisteme pot fi elementare sau obtinute din sisteme elementare prin urmatoarele tipuri de cuplaje:

1.2.1 Cuplaj paralel (fig.1.1) prin care se obtine un sistem echivalent S_{ne} , tot de ordinul n ;

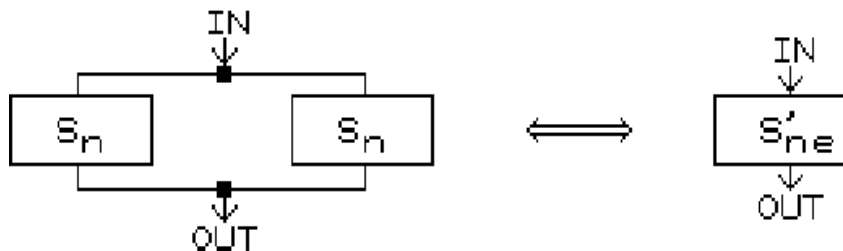


Fig. 1.1. Cuplare paralel a sistemelor logice de ordinul n . Sistemul echivalent S_{ne} este tot de ordinul n .

1.2.2 Cuplaj serie (fig.1.2) in care, prin conectarea inlantuita a doua sau mai multe sisteme de un anumit ordin, se obtine tot un sistem de acelasi ordin S_n ;

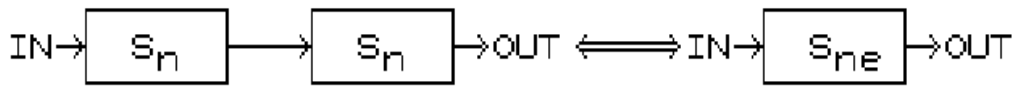


Fig. 1.2. Cuplare serie a sistemelor logice de ordinul n . Rezulta tot un sistem de ordinul n S_{ne} .

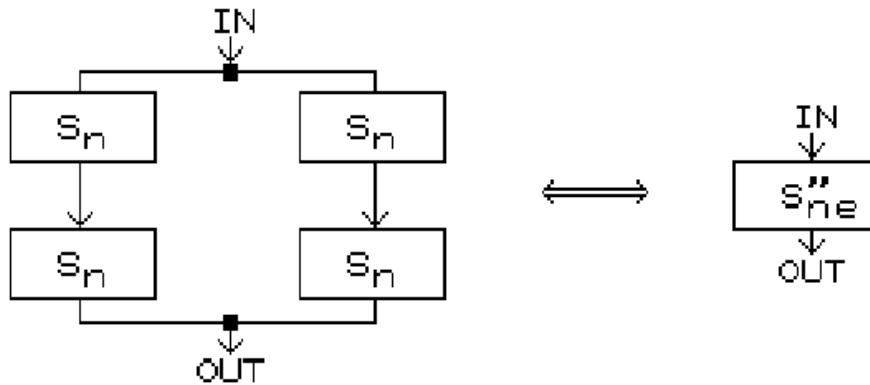


Fig. 1.3. Cuplare serie-paralel a sistemelor logice de ordinul n . Sistemul echivalent S_{ne} este tot de ordinul n .

1.2.3 cuplaj serie-paralel (fig.1.3), combinatie intre cele doua tipuri de cuplaje anterioare; si prin acest tip de conexiune se pastreaza proprietatile fundamentale ale sistemelor componente, S_{ne} , ramainind tot de ordinul n .

Spre deosebire de cuplajele 1.1 - 1.3 datorita carora se diversifica sistemele in cadrul aceluiasi ordin, cuplajul urmator permite obtinerea unor sisteme mai complexe, de ordin superior.

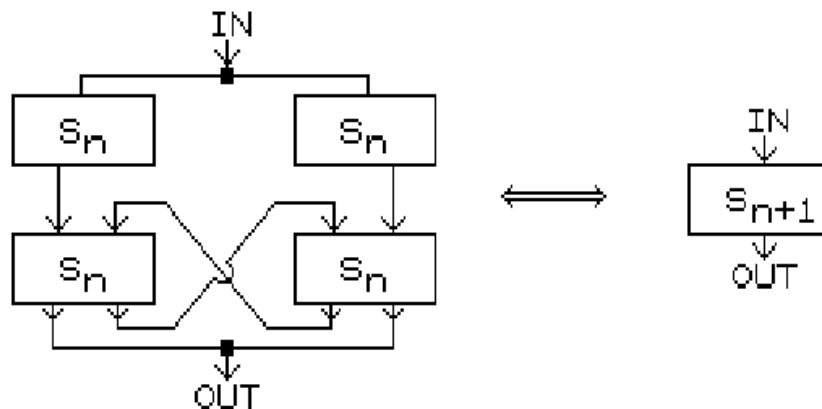


Fig. 1.4. Cuplare cu reactie a sistemelor logice de ordinul n . Rezulta un sistem superior, de ordin $n+1$.

1.4 cuplaj cu reactie (fig.1.4 si fig.1.5, singurul care realizeaza un salt calitativ in structurare: sistemul echivalent S_r are ordinul cu 1

mai mare decit ordinul maxim al sistemelor conectate in reactie

$$r = \max(n, t) + 1$$

Obs Trebuie remarcat ca, daca unul din sistemele S_n sau S_t (fig.1.5) nu realizeaza o prelucrare efectiva a datelor, reactia nu majoreaza ordinul sistemului.

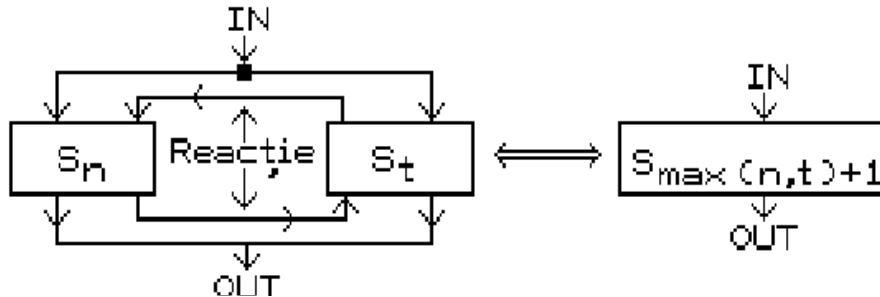


Fig. 1 5. Cuplarea cu reactie a sistemelor logice de ordine **diferite**. Se obtine un sistem de ordin $r=\max(n,r)+1$.

In cele ce urmeaza se vor da doua exemple privind rolul reactiei in sistemele logice (fig.1.6 si fig.1.7). Studiarea amanuntita a acestora se va face in capitolele urmatoare.

Exemplul 1. Latch-ul cu ceas are proprietatea de memorare, spre deosebire de poarta SINU.

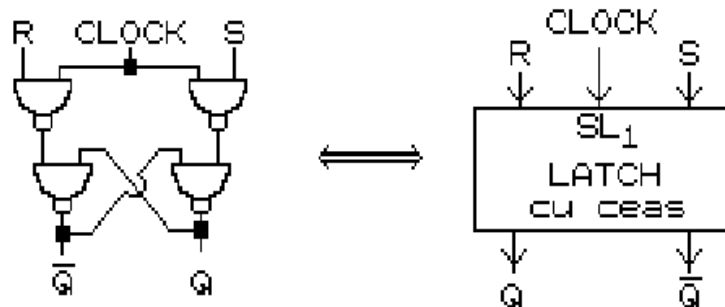


Fig. 1.6. Prin conectarea serie-parallel cu reactie a unor sisteme logice de ordinul 0, SL0 (porti SINU) se obtine un sistem logic de ordinul 1 SL1.

Exemplul 2.

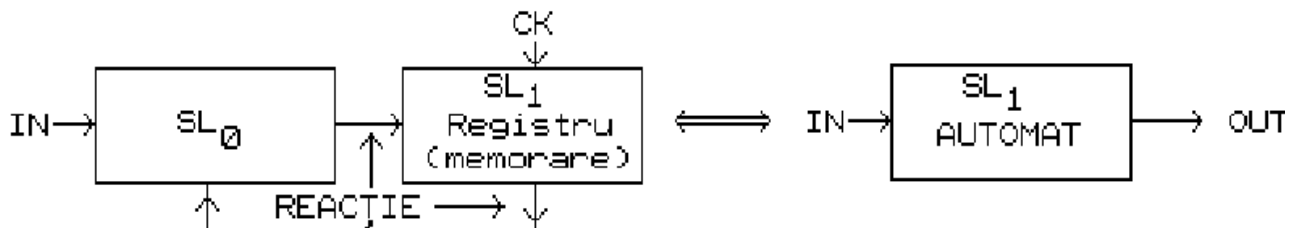


Fig. 1.7. Prin conectarea in reactie a unui SL_0 cu un SL_1 , registru (cu proprietati de memorare), se obtine un SL_2 numit **AUTOMAT** care are proprietatea de a rula algoritmi, de a controla diverse scheme complexe.

Cap.2 Clasificarea sistemelor digitale

Majoritatea sistemelor digitale cunoscute pina in prezent pot fi cuprinse in 5 ordine, SLo - SL4. Diversificarea acestora in functie de tipurile de cuplaje este ilustrata in tabel:

Tipuri de sisteme digitale (logice) SL	Sisteme elementare	Sisteme obtinute prin cuplaj paralel	Sisteme obtinute prin cuplaj serie	Sisteme obtinute prin cuplaj serie-paralel	Formalism matematic utilizat in descrierea SL
SL0 Sisteme Logice combinacionale (SLC)	Porti logice: SI NU, SAU NU, SI, SAU, SAU EX-CLUSIV etc.	-Demultiplexoare DMUX -Decodificatoare DCD	Multi-plexoare MUX	- ROM - PLA - PGA	ALGEBRA BOOLE
SL1 Sisteme de memorare	LATCH cu ceas	RAM	Bistabil MASTER-SLAVE MS	Registre	
SL2 Sisteme Logice Secventiale SLS	-Bistabilul JKMS -Numaratoare -Automate (*)	- RALU - CROM	Unitati PIPE LINE	Structuri functionale de tip PIPE LINE	(*)Teoria automatelor finite
SL3 Sisteme de procesare	-procesorul microprogramat -microprocesorul	RETELE ELEMENTARE DE PROCESARE			
SL4 Sisteme de calcul	Calculatorul	RETELE ELEMENTARE DE CALCULATOARE			

Notatiile din tabel au urmatoarele semnificatii:

ROM - Read Only Memory (Memorie numai cu citire)
 PLA - Programmable Logic Array (Suprafata logica programabila)
 PGA - Programmable Gate Array (Retea de porti programabila)
 RAM - Random Access Memory (Memorie cu acces aleator)
 RALU - Registre + ALU (Arithmetic Logic Unit - Unitate Logico-Aritmetica)
 CROM - Controlor cu ROM

Se observa ca aceasta clasificare este alcatuita dupa criteriile valabile, deoarece include toate tipurile de sisteme cunoscute pina in prezent. In plus, este o clasificare deschisa, oferind posibilitati de extindere catre sisteme logice de ordin superior celor existente. Ramine ca aceste posibilitati sa fie confirmate in viitor.

Cap.3 Sisteme logice de ordinul 0.

Sisteme logice combinacionale SLC

3.1 Proprietati fundamentale comune

3.1.1 Raspunsul la iesire Y este determinat numai de variabilele X de la intrarea sistemelor (fig.3.1).

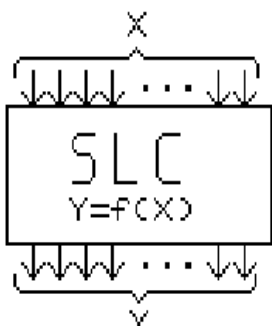


Fig. 3.1. Functia de transfer a SLC este $f:X \rightarrow Y$

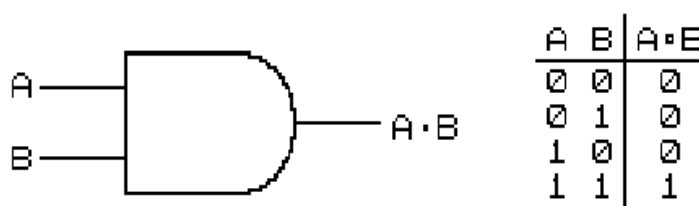


Fig. 3.2. Circuitul SI cu doua intrari.

3.1.2 Raspunsul corect apare la iesire dupa un interval de timp numit timp de propagare (T_p).

Spre deosebire de cazul ideal in care sistemele raspund instantaneu, in realitate raspunsul apare intirziat la iesire cu T_p . Pentru exemplificare se va considera un simplu circuit SI cu 2 intrari. Acesta are proprietatea de a lua valoarea logica 1 numai cind ambele variabile de intrare sint 1 (fig.3.2).

Raspunsul circuitului, in cazul ideal, este reprezentat in fig.3.3 in care se considera intrarea $B=1$ si intrarea A comutind din 0 in 1 la momentul t_0 .

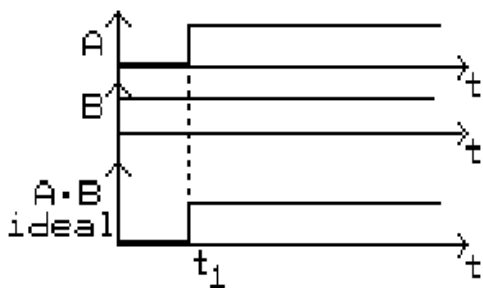


Fig. 3.3. Raspunsul circuitului SI in cazul ideal se stabileste instantaneu, in momentul t_1

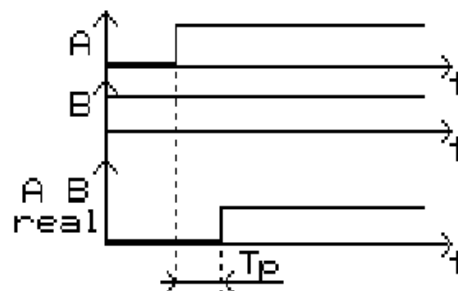
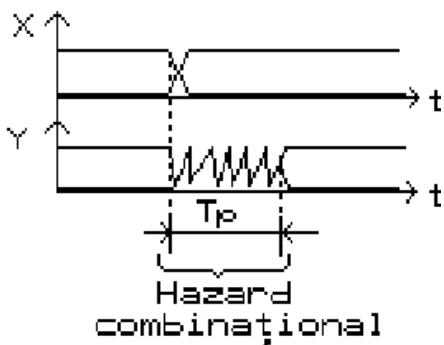


Fig. 3.4. Raspunsul real al circuitului SI apare intirziat cu timpul de propagare T_p .

Circuitul real are raspunsul intirziat, aparind dupa intervalul T_p (fig.3.4.).



Decalajul se datoreste timpului de propagare prin circuit a variatiilor de la intrare. Acest timp depinde de tehnologia de realizare a portii si uzual ordinul sau de marime este de citeva nanosecunde. In intervalul T_p raspunsul circuitului este incorect. In plus, la majoritatea S.L.C., raspunsul in intervalul T_p este fluctuant: pot apare mai multe valori logice aleatoare numite numite impulsuri parazite. Acest fenomen, caracteristic SLC, poarta numele de hazard combinational (HC) (fig.3.5.). Ordinul de marime al T_p , in functie de tehnologia si complexitatea circuitelor integrate, este de ns - zeci de ns.

Fig. 3.5. Raspunsul unui SLC are, in intervalul T_p , **hazard combinational**

3.1.3 Functiile de transfer ale SLC, numite functii logice, sint descrise cu ajutorul algebrei BOOLE.

Algebra BOOLE estse definita pe multimea variabilelor binare A, B, C... {0,1} cu ajutorul a doua legi de compozitie interna:
 - SI notata cu \cdot sau prin alipirea variabilelor AB
 - SAU notata cu $+$

si tinind cont si de negatia variabilelor A', B'....sau \bar{A} , \bar{B}

Postulatele si teoremele algebrei booleene

1. asociativitatea \fata de \cele doua legi de compozitie
 $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $A + (B + C) = (A + B) + C$
2. elementul neutru: 1 pt. SI, 0 pentru SAU
 $A \cdot 1 = A$ $A + 0 = A$
 Operarea cu elementele neutre inversate conduce la:
 $A \cdot 0 = 0$ $A + 1 = 1$
3. comutativitatea
 $A \cdot B = B \cdot A$ sau $AB = BA$ $A + B = B + A$
4. distributivitatea legii SI fata de SAU si a legii SAU fata de SI:
 la stinga $A \cdot (B + C) = AB + AC$ $A + BC = (A + B)(A + C)$
 la dreapta $(A + B) \cdot C = AC + BC$ $A \cdot B + C = (A + C)(B + C)$
5. principiul idempotentei
 $A \cdot A = A$ $A + A = A$
6. principiul contradictiei
 $A \cdot A' = 0$ (nu pot exista impreuna A si A')
7. principiul tertiului exclus
 $A + A' = 1$ (nu exista alta posibilitate in afara de A si A')
8. dubla negatie
 $(A')' = A$
9. principiul dualitatii: legile SI si SAU sint duale deoarece substituindu-se una cu alta se pot translata relatiile scrise cu (\cdot) in cele scrise cu $(+)$. (In relatiile ce contin 0 si 1, dualitatea implica si translatarea lui 0 in 1).

10. teoremele excluderii
 $AB+AB'=A$ $(A+B)(A+B')=A$

Obs. Se numesc astfel deoarece este exclusa variabila B.

11. teoremele absorbtiei
 $A+AB = A$ $A(A+B) = A$

Obs. Este absorbita variabila B.

12. teoremele semiabsorbției
 $A+A'B=A+B$ $A(A'+B)=AB$

Obs: Este absorbita variabila A'

13. teoremele lui De Morgan (TDM)
 $ABC\dots Z=(A'+B'+C'+\dots Z)'$ $A+B+C+\dots Z=(A'B'C'\dots Z)'$

Aceste teoreme, TDM, au o importanta deosebita, deoarece, pe baza lor se demonstreaza ca orice functie logica se poate exprima numai prin SAU NU si respectiv SI NU.

Algoritmul de aplicare al TDM contine urmatoarele trei etape:

- se neaga variabilele unite prin (\cdot) , respectiv $(+)$
- se transforma legile de operare prin dualitate $(\cdot)\leftrightarrow(+)$
- se neaga intreaga expresie a functiei.

Exemplu: Se se transforme cu TDM expresia $((AC)'+B)'$.

Solutie: $((AC)'+B)'=((AC)''B')''=ACB'$

S-a aplicat TDM iar apoi dubla negatie atat pentru variabila AC cit si pentru functie.

3.2 SLC elementare. Porti logice. Implementari cu porti.

3.2.1 Porti logice

Portile logice sint destinate realizarii unor functii logice elementare.

Prin considerarea functiilor logice de o variabila $f(A)$ se pot defini circuitele DA (buffer) si NU (inversor)

A	f1(A)	f2(A)	f3(A)	f4(A)
0	0	0	1	1
1	0	1	0	1

DA NU

Functiile f1 si f4 sint functii nesemnificative, independente de A.

f2 este functia de adevar DA, numita si buffer, deoarece se utilizeaza pentru amplificarea unor marimi electrice ca tensiuni, curenti. f2 se noteaza cu simbolul din fig.3.6.

f3(A) este functia negatie (inversare) si se noteaza ca in figura 3.7 prin adaugarea cerculetului simbolizind inversarea.

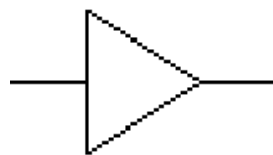


Fig. 3.6.
 Simbolul functiei logice DA (buffer).

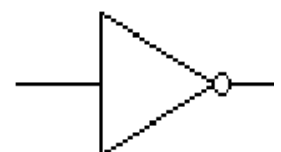


Fig. 3.7.
 Simbolul logic al inversorului (NON).

Cu ajutorul functiilor logice de 2 variabile (B,A) se pot defini alte tipuri de porti, cu doua intrari.

B	A	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

f9 este functia SI (AND) reprezentata prin simbolul din fig.3.8. Aceasta ia valoarea logica 1 numai cind ambele variabile sint 1.

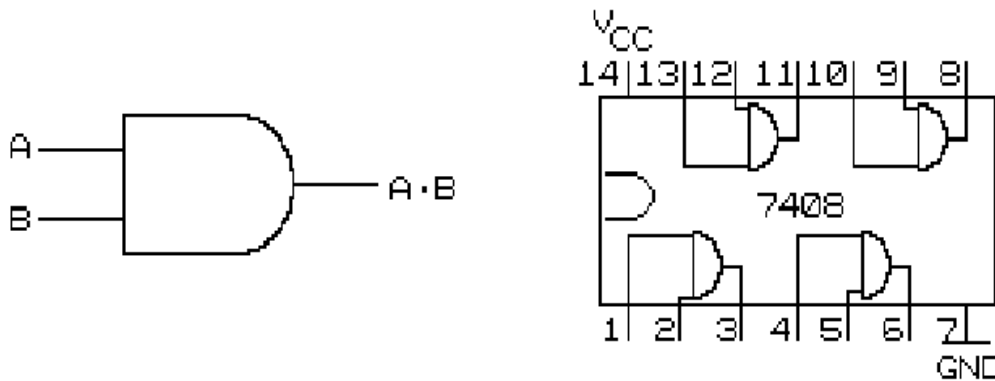


Fig.3.8. Functia SI (AND).

f8 este functia SI NU (NAND) care se reprezinta prin adaugarea la simbolul anterior a cerculetului marcind inversarea (fig.3.9).

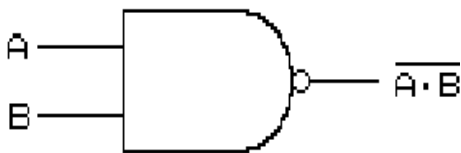


Fig. 3.9. Simbolul logic al functiei SI NU (NAND).



Fig. 3.10. Simbolul logic al functiei SAU.

f15 este functia SAU (OR) care are proprietatea de a lua valoarea logica 1 cind cel puțin una dintre variabile este 1. In fig.3.10 este prezentata poarta SAU.

f2 este functia SAU NU (NOR) si este reprezentata in fig.3.11.



Fig. 3.11. Simbolul logic al functiei SAU NU (NOR).



Fig. 3.12. Simbolul logic al functiei SAU EXCLUSIV (XOR).

f7 este functia SAU EXCLUSIV (XOR sau EOR) care are proprietatea de a lua valoarea logica 1 numai cind una din variabile este 1. Functia se numeste astfel deoarece se obtine din SAU prin excluderea valorii 1 pentru ultima combinatie a variabilelor A=B=1 (vezi tabelul).

Functia SAU EXCLUSIV (fig.3.12) are numeroase aplicatii care rezulta din urmatoarele interpretari ale acesteia:

- XOR este suma modulo 2 (adunarea binara);
- XOR este functia de anticoincidenta: semnalizeaza prin "1" faptul ca variabilele de intrare nu coincid ca valori;
- XOR este functia de complementare comandata. Aceasta proprietate rezulta din tabelul functiei astfel:

B A	B⊕A	
Pentru B= 0 { 0 0	0	=rezulta B⊕A=A, adica variabila A este transmisa
0 1	1	transparent prin poarta
Pentru B=1 { 1 0	1	=rezulta B⊕A=A, adica variabila A este transmisa
1 1	0	complementata prin poarta

B se numeste variabila de comanda a transmiterii, complementate sau necomplementate, a variabilei A prin poarta.

f10 este functia SAU EXCLUSIV NEGAT si are proprietatea de a semnaliza prin "1" coincidenta variabilelor.

Pentru functii logice cu mai multe variabile se pot scrie tabele similar. Astfel, pentru n variabile de intrare:

- 2^n este numarul combinatiilor (configuratiilor) de intrare

- 2^{2^n} este numarul functiilor ce se pot obtine cu cele n variabile.

Pentru 3 variabile C, B, A se obtin 8 configuratii de intrare si 256 functii f(C,B,A). Nu toate dintre acestea sint semnificative si majoritatea nu au asociate porti logice unice, se implementeaza (se realizeaza cu circuite integrate) cu ajutorul mai multor porti elementare, de tipul celor prezentate anterior. In afara de acestea se definesc ca circuite elementare si extensiile pentru mai multe variabile de intrare ale functiilor logice SI, SI NU s.a.

Portile logice integrate se gasesc incapsulate in cip-uri cu 14 conexiuni. Doua conexiuni sint utilizate pentru alimentare +5V(Vcc) si masa (GND - Ground), restul de 12 conexiuni ramaind pentru porti. Tipul acestora este specificat prin coduri

74XX , la IPRS CDB 4XX E

De exemplu, 7400 sau CDB400E este un CI care contine 4 porti SINU cu 2 intrari

7410 sau CDB410E contine 3 porti SINU cu 3 intrari.

Implementarea cu porti a functiilor logice

Funcitiile logice cu 3 si mai multe variabile nu au, in mare majoritate, asociata o singura poarta logica, ci un ansamblu de porti. Pentru implementarea acestor functii este necesara specificarea lor intr-o forma concreta, operanta.

3.2.2 Modalitati de exprimare a functiilor booleene

1) Tabelul de adevar - este o reprezentare care arata dependenta valorilor functiei (in dreapta) de multimea, ordonata crescator, a configuratiilor de intrare (in stanga).

De exemplu, pentru o functie de 3 variabile C,B,A tabelul este:

	2^2	2^1	2^0		
	C	B	A		f_{166}
0.	0	0	0		$a_0 = 0$
1.	0	0	1		$a_1 = 1$
2.	0	1	0		$a_2 = 1$
3.	0	1	1		$a_3 = 0$
4.	1	0	0		$a_4 = 0$
5.	1	0	1		$a_5 = 1$
6.	1	1	0		$a_6 = 0$
7.	1	1	1		$a_7 = 1$

Functia f este definita de vectorul $(a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0)$ unde $a_k=0,1$

S-a aratat ca exista $2^{2^3} = 256$ de functii distincte

2. Forme canonice - se numesc astfel deoarece contin termeni canonici (mintermeni sau maxtermeni) : acei termeni care depind de toate variabilele de intrare, adevarate sau negate.

O functie este exprimata in forma canonica cu mintermeni astfel:

$$f_n = \sum_{k=0}^{2^n-1} a_k P_k \text{ unde :}$$

n este numarul de variabile ale functiei

a_k sint coeficientii vectorului functiei

P_k sunt termenii canonici de tip mintermeni. Acestia se noteaza si cu m_k .

Pentru o functie de 3 variabile, termenii canonici sint:

$$\begin{aligned} P_0 &= C'B'A' & P_1 &= C'B'A & P_2 &= C'BA' & P_3 &= C'BA \\ P_4 &= CB'A' & P_5 &= CB'A & P_6 &= CBA' & P_7 &= CBA \end{aligned}$$

Se observa ca mintermenii se obtin reunind variabilele prin functia SI. Variabilele se considera adevarate sau negate, astfel incit pentru fiecare configuratie de intrare, SI-ul sa ia valoarea logica 1.

Exemplul considerat in tabelul de adevar anterior, f_{166} are urmatoarea forma cu mintermeni:

$$f_{166} = P_1 + P_2 + P_3 + P_7 = C'B'A + C'BA' + CB'A + CBA$$

In expresia functiei ramin doar termenii canonici validati de coeficientii $a_k = 1$, ca urmare a aplicarii postulatelor algebrei booleene: $0 \cdot P_k = 0$ $1 \cdot P_k = P_k$.

Forma canonica cu maxtermeni a unei functii cu n variabile este:

$$f_n = \prod_{k=1}^{2^n-1} (a_k + S_k)$$

unde S_k sint maxtermenii obtinuti prin conectarea cu SAU a variabilelor adevarate, sau negate, astfel incit SAU-ul sa ia valoarea logica 0.

Pentru o functie cu 3 variabile C, B, A, maxtermenii sunt:

$S_0 = C+B+A$	$S_1 = C+B+A'$	$S_2 = C+B'+A$	$S_3 = C+B'+A'$
$S_4 = C'+B+A$	$S_5 = C'+B+A'$	$S_6 = C'+B'+A$	$S_7 = C'+B'+A'$

Aplicind postulatele $0+S_k = S_k$ $1+S_k = 1$ $1 \cdot S_k = S_k$
 rezulta $f=S_0S_3S_4S_6$

3. Diagramele Veitch-Karnaugh sint reprezentari grafice ale functiilor, in care sint marcati prin coeficientii a_k , termenii canonici obtinuti prin intersectia unor suprafete asociate atit variabilelor, cit si negatelor acestora.

In diagramele Veitch suprafetele sint notate prin variabile (fig.3.13 si fig.3.14)

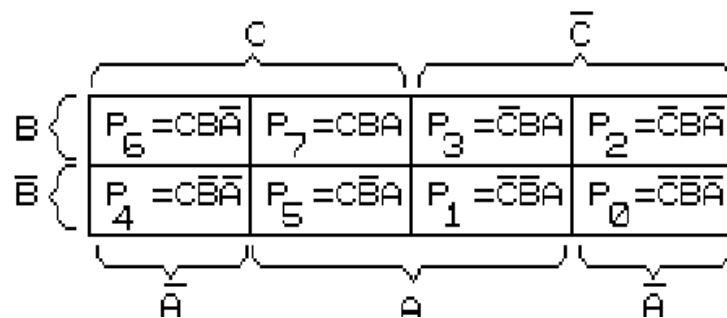


Fig. 3.13. Diagrama VEITCH pentru functii de 3 variabile exprimate in forma canonica cu mintermeni.

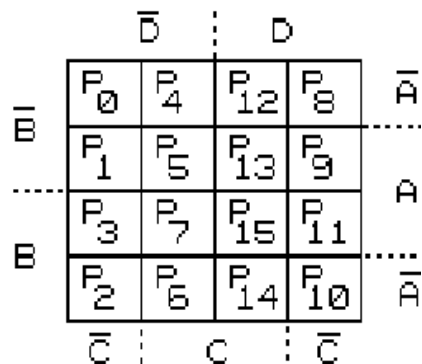


Fig. 3.14. Diagrama VEITCH pentru functii de 4 variabile in forma canonica cu mintermeni.

Observatie: Conteaza ordinea variabilelor. De obicei in schemele cu circuite integrate A este 2^0 .

Asocierea variabilelor cu suprafetele este arbitrara.

In diagramele Karnaugh variabilele sint notate in binar, in cod GRAY (de la o configuratie la cealalta se modifica doar o variabila) pentru a se asigura adiacenta suprafetelor (fig.3.15).

		CA			
	B	10	11	01	00
1			1	1	
0			1		1

Fig. 3.15. Diagrama KARNAUGH pentru functii de trei variabile echivalenta cu diagrama VITCH din fig 3.13.

Se poate gasi intotdeauna o diagrama Karnaugh echivalenta cu una Veitch. De aceea, uzual, cele doua se reprezinta suprapus:

-diagrama Karnaugh faciliteaza introducerea coeficientilor a_k pentru functii descrise prin tabel de adevar.

-diagrama Veitch este utila in extragerea termenilor minimizati.

A reprezenta o functie in diagrame VK inseamna a plasa coeficientii a_k in pozitiile corespunzatoare P_k .

Pentru forma cu mintermeni se iau in considerare doar acei $a_k=1$, iar pentru forma cu maxtermeni, acei $a_k=0$. De exemplu, functia $f(C,B,A)=P_0 + P_3 + P_6$ este reprezentata in diagrama VK in figura 3.16.

		BA		A	\bar{A}
	C	00	01	11	10
C 1					1
\bar{C} 0		1		1	
		\bar{B}		B	

Fig.3.16 Reprezentarea prin diagrame VK a functiei exprimita cu mintermeni $f = P_0 + P_3 + P_6$

4. Forme elementare (minimizate) sint forme ce contin un numar mai redus de termeni fata de cele canonice, iar termenii nu mai sint canonici, depind de un numar mai mic de variabile.

Formele elementare se obtin din cele canonice prin minimizare. Aceasta presupune utilizarea a doua metode:

- a. reducerea cu ajutorul postulatelor si teoremelor algebrei booleene;
- b. reducerea prin procedee grafice, cu diagrame VK.

Pentru exemplificare se va considera functia de vot majoritar, f_{VM} , functie definita pe un numar impar de variabile, care ia valoarea logica a majoritatii variabilelor. Consideram aceasta functie pentru 3 variabile, descrisa prin tabel de adevar.

	C	B	A	f_{VM}
0.	0	0	0	0
1.	0	0	1	0
2.	0	1	0	0
3.	0	1	1	1
4.	1	0	0	0
5.	1	0	1	1
6.	1	1	0	1
7.	1	1	1	1

In forma canonica cu mintermeni functia se exprima astfel:

$$f_{VM} = P_3 + P_5 + P_6 + P_7 = C'BA + CB'A + CBA' + CBA$$

a. Pentru minimizare cu metoda a. se face urmatorul artificiu: se aduna de doua ori in expresia functiei (CBA), conform principiului $A+A = A$.

$$f_{VM} = C'BA + (CBA) + CB'A + (CBA) + CBA' + CBA$$

Se grupeaza apoi termenii doi cite doi, aplicandu-se teoremele excluderii:

$$f_{VM} = (C'+C)BA + (B'+B)CA + CB(A'+A)$$

Deci s-a redus forma canonica de la 4 termeni de 3 variabile fiecare, la 3 termeni de 2 variabile fiecare.

b. Minimizarea cu aceasta metoda este mai simpla deoarece nu presupune artificii de calcul, rezultatele obtinute fiind echivalente. Pentru aceasta se reprezinta functia prin diagrama VK.

		CA			
		00	01	11	10
B	1		1	1	1
	0		1		

Fig.3.17 Minimizarea cu diagrama VK a functiei de vot majoritar de 3 variabile

Se observa ca aplicarii teoremelor excluderii ii corespunde grafic acoperirea 1-urilor din diagrama VK (a 0-urilor cind se lucreaza cu forma cu maxtermeni) cu suprafete in care se elimina variabilele care apar atat adevarate cit si negate. Prin notarea suprafetelor cu acelasi tip de linii (continua, punctata si ondulata) cu care s-au subliniat termenii carora li s-au aplicat teoremele excluderii, se evidentiaza echivalenta celor doua metode.

Se observa ca un 1 poate fi acoperit de mai multe ori, ca suprafetele de acoperire trebuie sa fie cit mai mari posibil, pentru a se elimina cit mai multe variabile. De asemenea, dimensiunile suprafetelor nu pot fi arbitrare: trebuie sa fie puteri ale lui 2, pentru a permite aplicarea teoremelor excluderii.

In plus, deoarece initial suprafetele asociate variabilelor pot fi arbitrar alese, rezulta ca lateralele diagramei VK sint adiacente.

Din cele prezentate anterior rezulta ca algoritmul de minimizare cu diagrame VK este urmatorul: se acopera toate 1-urile din diagrame cu suprafete maxime, dreptunghiulare, avind dimensiuni puteri ale lui 2.

In figurile 3.18 - 3.21 se prezinta citeva exemple de minimizare pentru functii de 3 si 4 variabile.

Fig.3.18 Minimizarea functiei
 $f = CB'A' + CB'A + CBA' + CBA = C$

Dimensiunea suprafetei de acoperire este 4x1

Dispar variabilele B si A care intervin in suprafata, atat adevarate cit si negate.

		BA			
		00	01	11	10
C	0	1	1	1	1
	1				

$f = C\bar{B}\bar{A} + C\bar{B}A + CB\bar{A} + CBA = C$

Fig.3.19 Minimizarea functiei $f=C'B'A'+C'BA'+CB'A'+CBA' = A$.

In acest caz intervine adiacenta suprafetelor laterale iar dimensiunea suprafetei de acoperire este 2x2

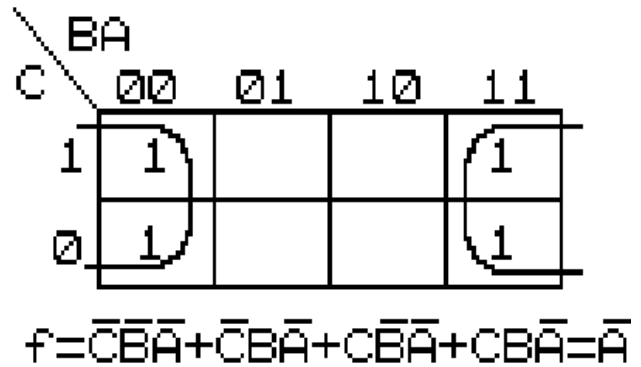


Fig.3.20 Minimizarea functiei $f = m_0 + m_1 + m_3 + m_4 + m_5 + m_7 = D'B + D'A$.

Dimensiunea 2x3 nu este acceptabila pentru ca 3 nu este putere a lui 2; se acopera 1-urile cu doua suprafete avind ca dimensiune 2x2

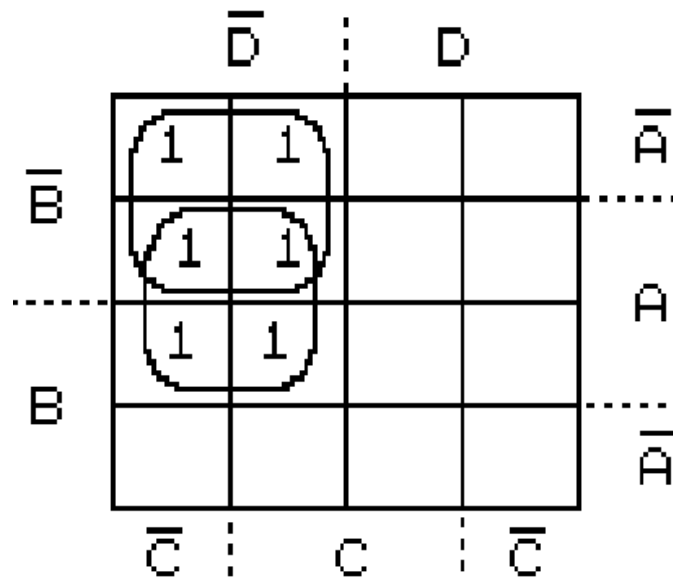
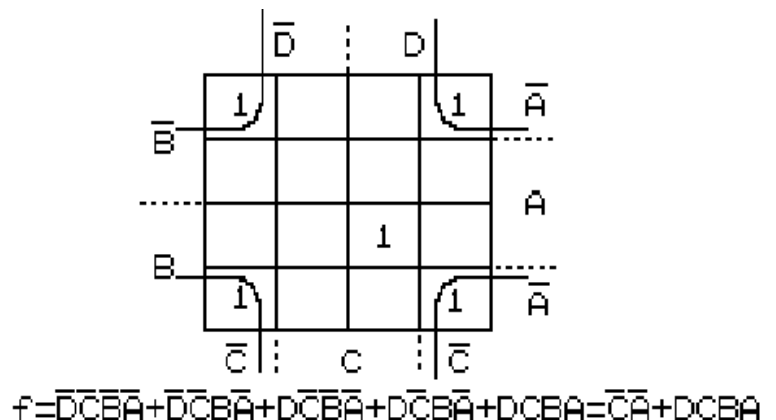


Fig.3.21 Minimizarea functiei $f=D'C'B'A'+D'C'BA'+DC'B'A'+DC'BA'+DCBA = C'A'+DCBA$

1-urile din colturi sint adiacente, deci se pot acoperi cu aceeași suprafata. Coeficientul a = 1 nu se poate acoperi decit cu propria lui suprafata pentru ca nu are vecini pe orizontala si verticala.



In situatiile in care pentru o anumita functie exista mai multe expresii minime, se extrag toate acestea si se utilizeaza cea care convine cel mai mult in implementare. In figura 3.22 este dat un astfel de exemplu.

Fig.3.22 Minimizarea functiei

$$C'B'A+C'BA'+CB'A'+CB'A+CBA'$$

$a_4=1$ poate fi acoperit cu 2

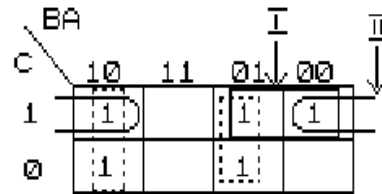
suprafate: I =CB' sau II =CA'.

Rezulta doua expresii ale

functiei minime

I $f = BA'+B'A+CB'$

II $f= BA'+B'A+CA'$



$$f = \bar{C}\bar{B}A + \bar{C}B\bar{A} + C\bar{B}\bar{A} + C\bar{B}A + C\bar{B}\bar{A} = \bar{B}\bar{A} + \bar{B}A + C\bar{B} = \bar{B}\bar{A} + \bar{B}A + C\bar{A}$$

4.1. Minimizarea functiilor cu coeficientii vectorului dependenti de variabilele de intrare

Acestea sint functii cu variabile numeroase de intrare care au proprietatea de a se putea exprima comprimat ca in exemplul urmator:

	C	B	A	f
0.	0	0	0	E
1.	0	0	1	1
2.	0	1	0	1
3.	0	1	1	(DE)'
4.	1	0	0	E
5.	1	0	1	1
6.	1	1	0	1
7.	1	1	1	0

Aceasta functie, dependenta de 6 variabile (A-F) este exprimata canonic doar cu ajutorul a 3 dintre ele si anume C,B,A, restul intervenind in coeficientii a_k astfel:

$$a_0 = E, \quad a_3 = (DE)', \quad a_4 = E$$

Forma canonica in mintermeni, raportata la variabilele C, B, A este:

$$f = EC'B'A' + C'B'A + C'BA' + (DE)'C'BA + ECB'A' + CB'A + CBA'$$

Se observa ca, luind in considerare ansamblul variabilelor functiei, expresia nu este canonica.

Pentru minimizare se reprezinta functia in diagrame Veitch-Karnaugh (VK) asa cum arata figura 3.23.

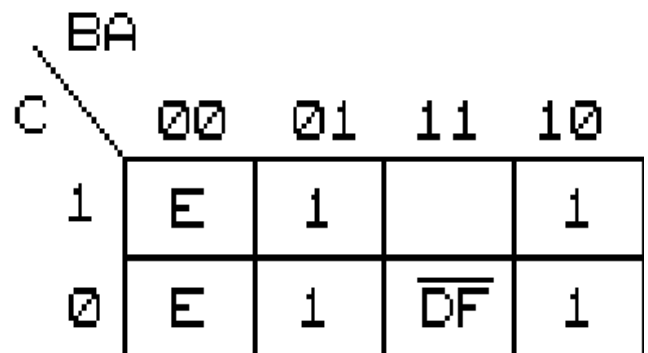


Fig.3.23 Reprezentarea in diagrame VK a functiei f(A-F).

Algoritmul de minimizare, in acest caz, contine urmatoarele etape:

I - se neglijeaza toate variabilele incluse in diagrama(fig.3.24), minimizindu-se l-urile cu metoda anterioara

II. - se considera, pe rind, coeficientii a_k neglijati anterior. In aceasta etapa, 1-urile pot lua orica valoare (0 sau a_k) astfel incit sa rezulte suprafete de acoperire maxime.

Functia data (fig.3.23) are doua tipuri de coeficienti a_k dependenti de variabile:

$$a_0 = a_4 = E$$

$$a_3 = (DF)'$$

		BA			
		00	01	11	10
C	1		1		1
	0		1		1

Fig.3.24 Se minimizeaza partial functia neglijind variabilele incluse in diagrama

		BA			
		00	01	11	10
C	1	E	E		
	0	E	E		
		$E\bar{B}$			

		CA			
		00	01	11	10
B	1	E			E
	0	E			E
		$F\bar{A}$			

Fig.3.25 Pentru acoperirea $a_0 = a_4 = E$ se considera ca 1-urile adiacente care ajuta la minimizare sint egale cu E

		BA			
		00	01	11	10
C	1				
	0		DF	DF	

		BA			
		00	01	11	10
C	1				
	0			DF	DF

$f = B'A + BA' + EB' + (DF)'C'B$
 $f = B'A + BA' + EB' + (DF)'C'A$
 $f = B'A + BA' + EA' + (DF)'C'B$
 $f = B'A + BA' + EA' + (DF)'C'A$

Fig.3.26 Pentru acoperirea lui $a_4 = (DF)'$ se considera ca 1-urile adiacente care ajuta la minimizare, sint egale cu $(DF)'$.

4.2. Minimizarea functiilor incomplet specificate

Acestea sint functii cu vectorul caracteristic nespecificat complet: anumiti coeficienti $a_k = x$ ($x = 0$ sau 1) au valori indiferente (don't care) din punctul de vedere al comportarii functiei. De aceea, in procesul de minimizare, coeficientii respectivi vor fi alesi cu valoarea logica necesara cresterii suprafetelor de acoperire.

Consideram pentru exemplificare functia logica de 4 variabile definita printr-un vector dependent de una din variabile (D) si incomplet specificat.

		BA			
		00	01	11	10
C	1	D	1		
	0	X	X	X	

C	B	A	f
0	0	0	0
0	0	1	X
0	1	0	X
0	1	1	X
1	0	0	0
1	0	1	D
1	1	0	0
1	1	1	1

Fig.3.27 Diagrama VK si tabelul de adevar pentru functia f incomplet specificata.

Parcurgerea celor doua etape ale minimizarii este prezentata in figura 3.28:

Etapa I de minimizare: Se neglijeaza variabila D si se dau X-urilor valori convenabile pentru marirea suprafetei de acoperire

Etapa II de minimizare: 1 si X sint inlocuite cu valorile D sau 0, pentru obtinerea suprafetei maxime de acoperire

		BA			
		00	01	11	10
C	1		1		
	0	X=0	X=1	X=0	

a.

		BA			
		00	01	11	10
C	1	D	D		
	0	X=D	X=D	X=0	

b.

Fig.3.28 Etapele minimizarii functiei $f = X m_1 + X m_2 + X m_3 + D m_5 + m_7$

Trebuie atrasa atentia asupra faptului ca prin reducerea functiilor cu vectorul dependent de variabile de intrare, nu se obtin totdeauna forme minime ale acestora: minimizarea poate fi continuata ulterior, folosind postulatele si teoremele algebrei BOOLE.

Pentru exemplificare se considera functia a carei diagrama este prezentata in figura 3.29.

$$f = X.m_0 + D m_1 + DE.m_2 + X m_3 + D m_4 + X m_5 + DE m_6 + X m_7$$

Prin reducere cu ajutorul acoperirii suprafetelor maxime ocupate cu cele doua variabile D si DE se obtine forma:

$$f = DB' + DEB$$

Aceasta forma nu e minima; ea poate fi minimizata scotind pe D factor comun $f = DB' + DEB = D(B' + EB)$ si aplicind teorema semiabsorbției. Rezulta:

$$f_{\min} = D(B' + E)$$

		BA			
		00	01	11	10
C	1	D	X	X	DE
	0	X	D	X	DE

a.

a. Diagrama VK a functiei f.

		BA			
		00	01	11	10
C	1	D	X=D	X=0	
	0	X=D	D	X=0	

b.

b. Minimizarea f considerind variabila D.

		BA			
		00	01	11	10
C	1		X=0	X=DE	X=DE
	0	X=0		X=DE	X=DE

c.

c. Minimizarea f considerind variabila DE

Fig.3.29 Reducerea functiei f la forma $f=DB'+DEB$

5. Forme neelementare. Acestea sint expresii ale functiilor continind paranteze, negari de mai multe variabile, SAU EXCLUSIV.

De exemplu: $f = (A \circ B + CD)(C + E)$

Astfel de functii se prelucreaza cu postulatele si teoremele algebrei BOOLE pentru a se aduce la forme canonice si pentru a se minimiza.

6. Forme de descriere in limbaj natural. -Sint cele mai accesibile pentru comunicarea interumana, dar dificil de translatat in formele anterioare, singure aplicabile in lumea binara a circuitelor digitale. Metodele de translatare nu pot fi algoritmizate si solicita adesea inventivitatea proiectantului in parcurgerea urmatoarelor etape:

- E1. Stabilirea variabilelor de intrare;
- E2. Minimizarea acestor variabile;
- E3. Determinarea functiilor booleene, presupunind stabilirea corecta a relatiilor dintre variabile.

Exemplu: Sa se scrie functia de pornire a unei farfurii zburatoare cu o capacitate de 20 extraterestri si roboti. Conditii de pornire sint:

1. intreg echipajul sa se afle la bordul navei;
2. sa fie activata sursa de energie a navei;
3. extraterestrii sa fie plasati in jumatatea de control a navei si robotii in cealalta jumatate;
4. extraterestrii si robotii sa isi reia infatisarea pe care o au pe planeta lor;
5. declansarea pornirii se face cu gindul unuia dintre extraterestri;

Etapele E1 si E2 se parcurg impreuna pentru stabilirea variabilelor de intrare :

Extraterestrii E_1, E_2, \dots, E_{20} ;

Robotii nu vor fi R_1, R_2, \dots, R_{20} , ci se vor considera ca

non-extraterestri, adica $E'_1, E'_2, \dots, E'_{20}$. Astfel s-a redus cu 20 numarul variabilelor de intrare;

Sursa de energie S (1- activata, 0- neactivata);

Variabile de recunoastere a infatisarii I_1, I_2, \dots, I_{20} ;

Variabilele de detectare a pozitiei in nava: J - jumatatea de control, cu extraterestri, J' - cea cu roboti;

Gindurile extraterestrilor: G_1, G_2, \dots, G_{20} ;

Functia de pornire exprima relatiile logice dintre cele 62 de variabile:

$$P = I_1 \cdot I_2 \cdot \dots \cdot I_{20} \cdot S \cdot (G_1 + G_2 + \dots + G_{20}) (E_1 \cdot J + E'_1 \cdot J') \cdot \dots \cdot (E_{20} J + E'_{20} J') =$$

$$= I_1 \cdot I_2 \cdot \dots \cdot I_{20} \cdot S \cdot (G_1 + G_2 + \dots + G_{20}) (E_1 \text{ XOR } J)' \cdot \dots \cdot (E_{20} \text{ XOR } J)'$$

3.2.3. Teorema implementarii cu SI NU a functiilor logice

"Implementarea" este un cuvint preluat din limba engleza si inseamna realizarea functiilor logice cu circuite integrate, CI.

Tipurile de porti care se utilizeaza se aleg in functie de urmatoarele criterii:

- viteza de raspuns a schemei (timp de propagare T_p);
- minimizarea numarului de CI
- tehnologia de realizare a CI utilizate.

De exemplu, in tehnologie TTL, prima care s-a impus in CI, portile cele mai rapide si cu cea mai mare diversitate in privinta numarului de intrari sint cele SI NU. Aceasta deoarece s-au putut realiza tehnologic cu schema cea mai simpla (vezi paragraful 3.2.4.3). De aceea implementarile presupun prelucrari cu teoremele lui De Morgan, TDM, ale functiilor in sensul exprimarii doar dependent de SI NU.

De exemplu, functia $f = BA + CA + C'B$, prelucrata cu TDM pentru desfiintarea SAU-rilor devine:

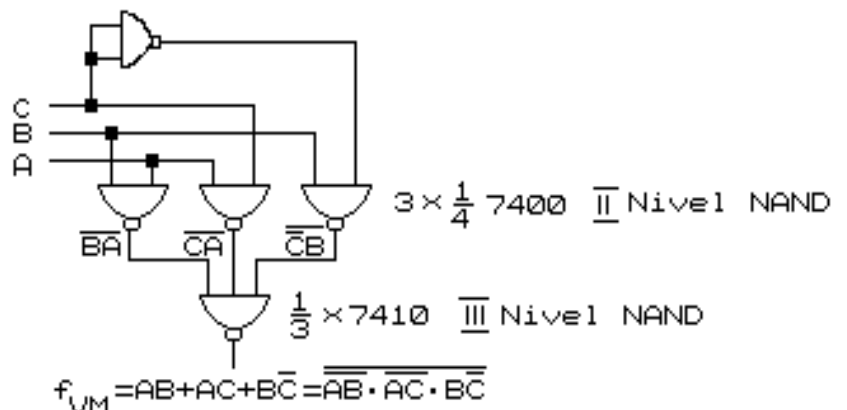
$$f = ((BA)' \cdot (CA)' \cdot (C'B)')'$$

si este exprimata prin functii SI NU de 1, 2 si 3 variabile.

Fig.3.30 Implementarea cu SI NU-uri a functiei de vot majoritar minimizata: sint necesare 2 CI.

Trebuie observat ca orice functie, in forma canonica cu mintermeni sau minimizata din aceasta, se poate implementa cu 2 niveluri de SI NU-uri:

- I nivel corespunde realizarii termenilor
- al II-lea nivel corespunde insumarii termenilor.



In cazul general mai este necesar la intrare un nivel de inversoare, pentru negarea variabilelor care intervin astfel, in termenii canonici.

3.2.4. Tehnologia de realizare a portilor logice.

3.2.4.1. Porti cu tranzistoare.

Stramosii portilor integrate, portile cu tranzistoare prezinta un interes nu numai istoric, ci si in privinta intelegerii structurilor interne ale CI. De aceea se vor prezenta pe scurt citeva aspecte ale functionarii portilor cu tranzistoare.

Inversorul cu tranzistor

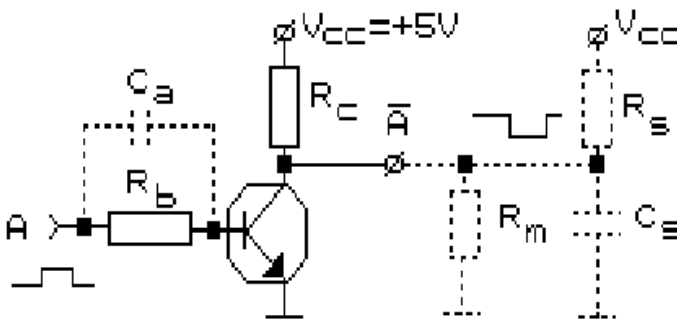


Fig.3.30 Inversorul cu tranzistor

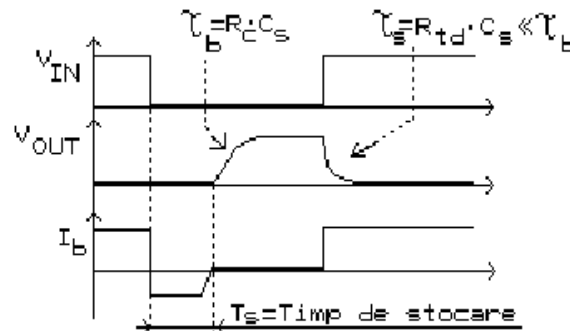


Fig.3.31 Forme de unda ale comutarii inversorului intre saturare si blocare, fara capacitatea de accelerare Ca.

Se considera in plus incarcarea rezistiva (R_s , R_m) si capacitiva (C_s), desenate punctat.

Tranzistorul poate functiona la "0" logic in doua regimuri: regiunea activa normala, RAN, sau in saturare.

Fig.3.31 Forme de unda ale comutarii inversorului intre saturare si blocare, fara capacitatea de accelerare Ca.

Comutarea in RAN are dezavantajul de a nu asigura o diferenta mare intre tensiunile asociate nivelurilor logice 1 (blocare) si 0 (RAN). Acest dezavantaj este inlaturat de comutarea in saturare, dar apare ca neplacuta intirziera blocarii datorita timpului de stocare (Timpul necesar pentru eliminarea purtatorilor minoritari acumulati in baza prin injectia suplimentara de curent la saturare : $I_B > I_C/\beta$). Efectul timpului de stocare poate fi redus prin:

- conectarea capacitatii de accelerare C_a , cu rolul de marire a vitezei de extragere a purtatorilor minoritari din baza;
- conectarea unei diode Schottky intre colectorul si baza tranzistorului pentru eliminarea rapida a sarcinii stocate in baza.

Comportarea inversorului in sarcina rezistiva

Rezistenta de sarcina catre sursa, R_s , poate varia in limite largi deoarece variatia de curent este preluata de tranzistor.

Incarcarea catre masa, R_m , nu poate fi la fel de mare ca cea catre sursa, R_s deoarece, la blocarea tranzistorului scade tensiunea asociata nivelului logic 1. De exemplu pentru $R_s = R_c$ tensiunea se injumatateste.

Concluzie: Inversorul are proprietatea de a suporta incarcari rezistive mari catre sursa, nu si catre masa.

Comportarea inversorului in sarcina capacitiva

Capacitatea de sarcina C_s influenteaza regimul dinamic al tranzistorului:

- la blocare C_s se incarca prin R_c de la sursa de alimentare V_{CC} . Durata frontului pozitiv (comutarea din 0 in 1) este proportionala cu constanta de timp $\tau_{Blocare} = R_c C_s$. Aceasta are valoare relativ mare pentru $R_c = 1k$ si nu poate fi variata in timpul comutarii;
- la saturare, C_s se descarca prin tranzistorul saturat (in principal) si, in consecinta, frontul de saturare (din 1 in 0) este mai mic, proportional cu $\tau_{Saturare} = C_s R_{tranzistor}$

Concluzie: Inversorul este un circuit care asigura fronturi de saturare bune si strica frontul de blocare.

Repetorul

(fig.3.32) este un circuit cu proprietati complementare fata de inversor: suporta bine incarcari catre masa care sint preluate de tranzistor si asigura fronturi bune de blocare. Din acest motiv, etajul de iesire al portilor integrate contine un inversor si un repetor, comandate in contratimp.

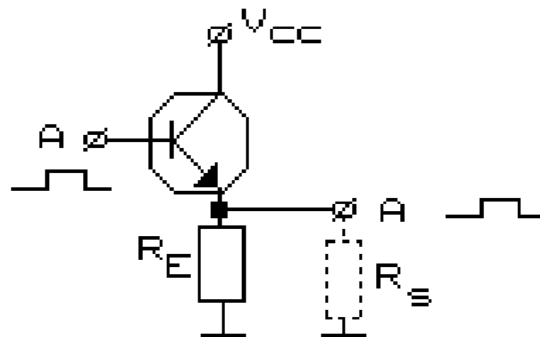


Fig.3.32 Repetorul. Circuit cu comportare in sarcina complementara cu a inversorului

Portile SI NU si SAU NU

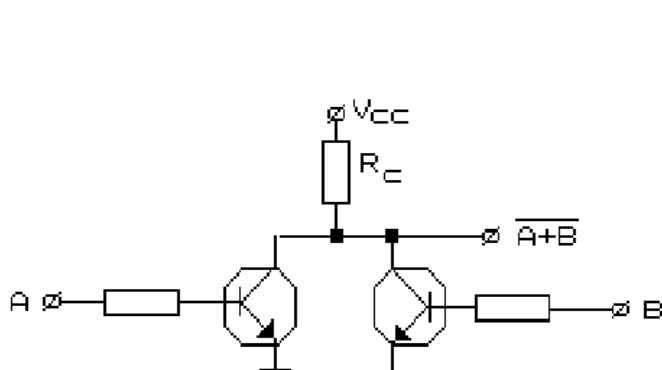


Fig.3.34 Circuit SAU NU cu tranzistoare. 1 la cel putin o intrare produce saturarea tranzistorului respectiv, deci 0 la iesire.

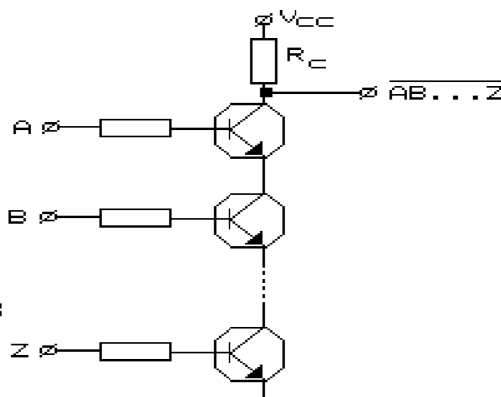


Fig.3.33 Circuit SI NU cu tranzistoare. Iesirea este "0" doar cind toate intrarile sint la 1 (toate tranzistoarele sint saturate)

3.2.4.2. Circuite integrate. Clasificare.

Portile integrate au aparut ca urmare a evolutiei tehnologice care a permis realizarea unor circuite tot mai complexe.

Clasificarea CI dupa numarul de tranzistoare pe capsula:

- circuite SSI (Small Scale Integration) avind pina la 50 de tranzistoare pe capsula. Exemplu: portile logice;
- circuite MSI (Medium Scale Integration) cu pina la 500 de tranzistoare pe capsula. Exemplu: MUX, DMUX, registre;
- circuite LSI (Large Scale Integration), continind mii de tranzistoare pe capsula. Exemplu: ROM, PLA, RAM, microprocesoare de 8 biti s.a.;
- circuite VLSI (Verry Scale Integration), continind zeci de mii de tranzistoare pe capsula. Exemplu: microprocesoare pe 16 biti, pe 32 biti;
- circuite ULSI (Ultra Large Scale Integration), continind sute de mii de tranzistoare pe capsula. Exemplu: transputere, calculatorul IBM360 intr-un cip s.a.

Clasificarea CI dupa tehnologia de realizare:

- circuite in tehnologie bipolara care sint de urmatoarele tipuri
 - DTL - Diode Transistor Logic
 - TTL - Transistor Transistor Logic
 - ECL - Emitter Coupled Logic
 - I²L - Injection Injection Logic
- circuite in tehnologie unipolara:
 - pMOS - MOS cu canal p
 - nMOS - MOS cu canal n
 - cMOS - MOS Complementar

Citeva din cele mai importante dintre aceste tipuri vor fi prezentate in cele ce urmeaza.

3.2.4.3. Circuite integrate TTL

Acestea se clasifica la rindul lor in:

- CI standard TTL
- HTTL (High TTL) continind circuite cu viteza mai mare de lucru decit cele din seria standard
- STTL (Schottky TTL) - circuite cu viteza sporita fata de cele HTTL, dar si cu consum de putere mai mare
- TLLP (TTL Low Power) - circuite cu consum redus de putere
- TTLSLP (TTL Schottky Low Power) - circuite cu viteza mare dar si cu consum redus de putere.
- FAST TTL, contine circuite cu viteza mare de raspuns.

Schema interna a portii SI NU TTL

Schema interna a acestor CI este prezentata in figura 3.35. Tranzistorul de intrare T_1 este tranzistor multiemitor. Cu ajutorul lui, prin adaugarea de emitoare (B) se realizeaza functia logica.

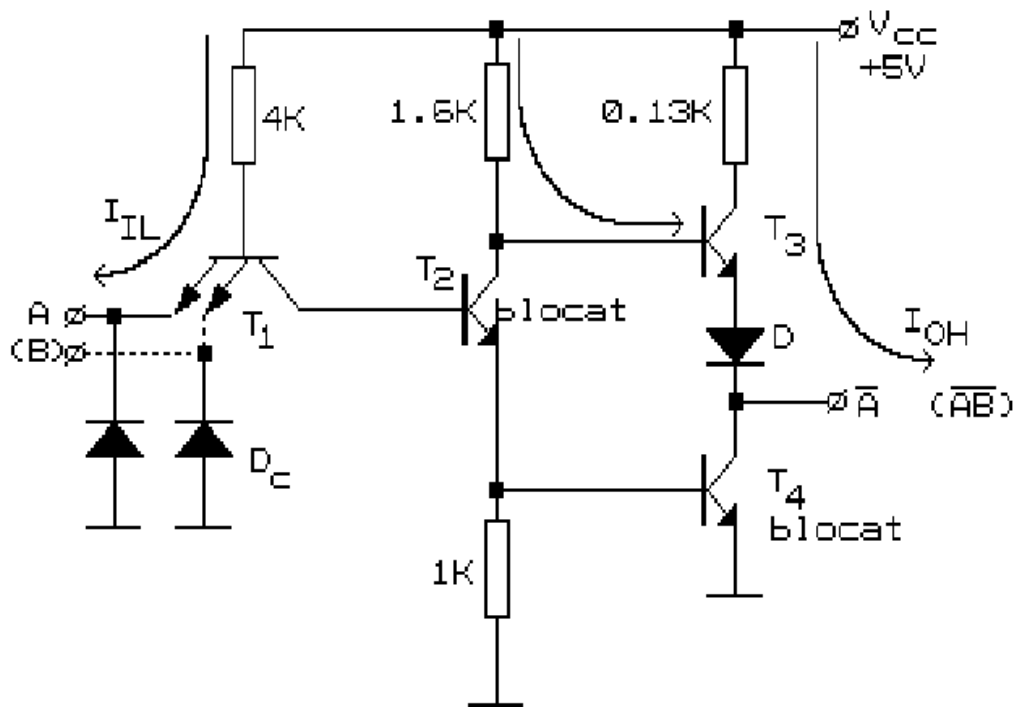


Fig.3.35 Inversorul TTL standard

Poarta SI NU se obtine prin adaugarea de intrari la tranzistorul multiemitor T_1 . S-a considerat situatia $A = 0$ logic = U_{IL} (tensiunea de intrare la 0 logic).

Tranzistoarele T_3 si T_4 formeaza etajul de iesire TOTEM POLE. T_2 comanda in contratimp inversorul T_4 si repetorul T_3 . D are rolul de a ridica potentialul emitorului lui T_3 , pentru a asigura blocarea acestuia (vezi figura 3.36.). Diodele de intrare D_c au rolul de a taia variatiile negative ale tensiunii de intrare (datorate de obicei reflexiilor pe liniile de interconectare dintre circuite).

In figura 3.35 este analizata schema portii inversoare, considerind ca la intrarea A se aplica 0 logic adica tensiune $U_{IL}=0,8V$ (IL - Input LOW -Intrare jos). Datorita diferentei mari de tensiune intre V_{cc} si intrarea A , se deschide jonctiunea baza-emitor a lui T_1 , prin care va circula un curent $I_{IL}=1,6$ mA (IL - Input LOW). Jonctiunea baza colector a lui T_1 fiind blocata, T_2 si T_4 vor fi blocate deoarece nu primesc curent in baza. T_3 va fi deschis datorita curentului injectat in baza sa prin rezistenta de 1,2 k. Curentul de iesire I_{OH} este injectat in rezistenta de sarcina. Tensiunea la iesire va avea nivel logic 1, $U_{OH}=2,4$ V, in functie de incarcarea circuitului.

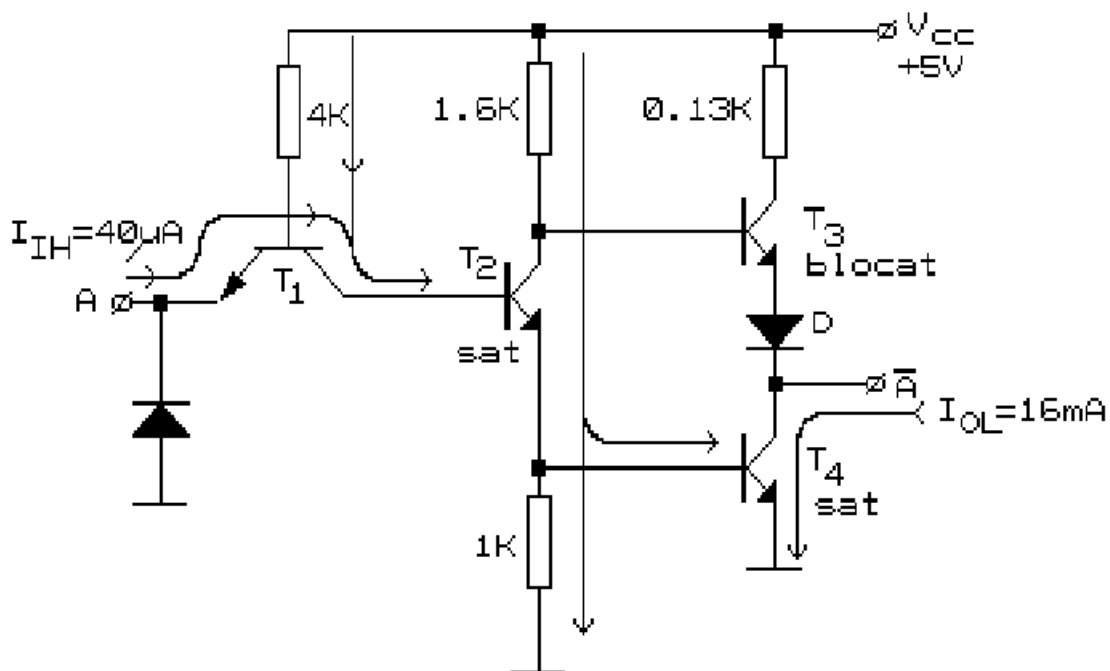


Fig.3.36 Functionarea inversorului pentru "1" la intrare

In figura 3.36 este prezentata situatia tranzistoarelor si circulatia curentilor pentru 1 logic, U_{IH} , aplicat la intrare. Fata de cazul anterior (fig.3.35), la cresterea tensiunii de intrare de la 0,8V la 2V, jonctiunea BE a lui T_1 se blocheaza. Prin ea va circula curentul, suma de curenti reziduali, $I_{IH}=40\mu A$. Curentul de la V_{cc} va circula prin jonctiunea BC a lui T_1 care se deschide si va intra in baza lui T_2 , de asemenea deschis. T_4 primeste curent de baza si intra in saturare. T_3 este blocat. Curentul de iesire I_{OL} este absorbit din sarcina.

Observatie: Daca lasam in gol intrarile unui CI, comportarea acestuia este identica cu cea de la aplicarea lui "1". In schemele complexe, cu numeroase CI, nu este indicata lasarea intrarilor neutilizate in gol, deoarece, avind impedanta mare, culeg zgomote si fluctueaza nivelul logic.

De asemenea, nu trebuie neglijate intrarile neutilizate ale unui circuit integrat deoarece, ele ar putea fi active pe 1 logic (exemplu: intrarea CLR a 74193) ceea ce ar stinjeni functionarea normala a circuitului.

Din aceste doua aspecte prezentate rezulta urmatoarea regula: toate intrarile neutilizate ale unui CI trebuie conectate la niveluri logice inactive, stabile "0" (masa) sau "1", prin rezistenta la V_{cc} . Rezistenta limiteaza curentul in cazul strapungerii intre emitoare. Trebuie mentionat ca se utilizeaza o singura rezistenta pentru un grup de intrari ce se conecteaza la 1. Calculul valorii R se face astfel incit sa se asigure curentii reziduali I_{IH} ai intrarilor (fig.3.37).

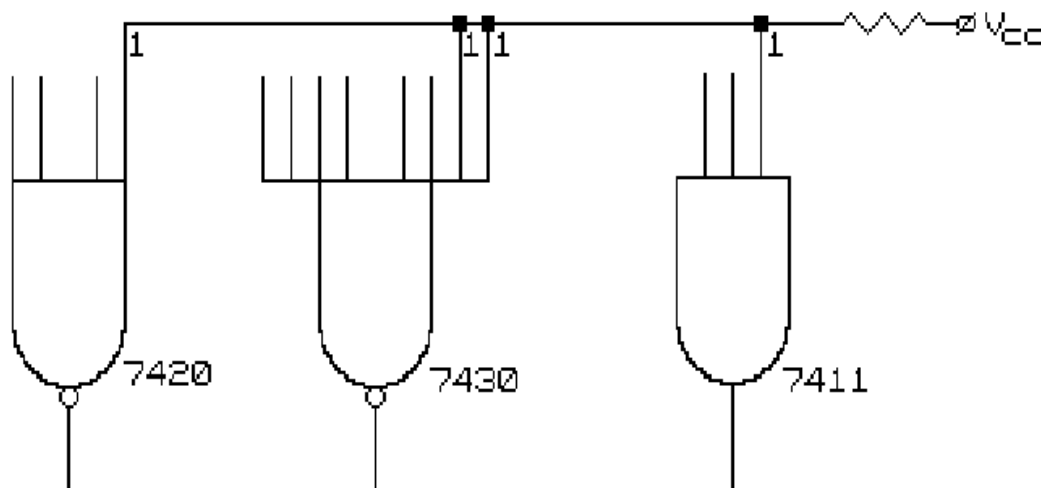


Fig.3.37 "1" logic la intrarile CI inseamna conectarea acestora prin rezistenta (R) la sursa (Vcc).

Parametrii circuitelor integrate

1. Tensiunile si curentii pe intrari si iesiri

U_{IL} - tensiunea de intrare (Input) asociata nivelului logic 0 (LOW)
 $U_{IL} = 0 \dots 0,8 \text{ V}$

U_{OL} - tensiunea de iesire (Output) asociata nivelului logic 0 (LOW)
 $U_{OL} = 0 \dots 0,4 \text{ V}$

U_{IH} - tensiunea de intrare asociata nivelului logic 1 (IH - Input High)
 $U_{IH} = 2 \dots 5 \text{ V}$

U_{OH} - tensiunea de iesire asociata nivelului logic 1 (OH - Output High)
 $U_{OH} = 2,4 \dots 5 \text{ V}$

I_{ILmax} - valoarea maxima a curentului de intrare in starea logica 0 la intrare (IL - Input LOW - Intrare Jos)
 $I_{ILmax} = 1,6 \text{ mA}$

I_{OLmax} - valoarea maxima a curentului absorbit de iesirea unei porti in starea logica 0
 $I_{OLmax} = 16 \text{ mA}$

I_{IH} - valoarea maxima a curentului rezidual absorbit de intrarea blocata, in starea logica 1.
 $I_{IHmax} = 40 \text{ } \mu\text{A}$

I_{OHmin} - valoarea minima a curentului ce poate fi injectat in sarcina de iesirea CI in starea logica 1.
 $I_{OHmin} = 0,8 \text{ mA}$

In figura 3.38 sint prezentate sensurile tensiunilor si ale curentilor ce caracterizeaza cele 2 stari distincte ale unui inversor integrat.

Valorile specificate sint valabile pentru toate CI din seria TTL standard.

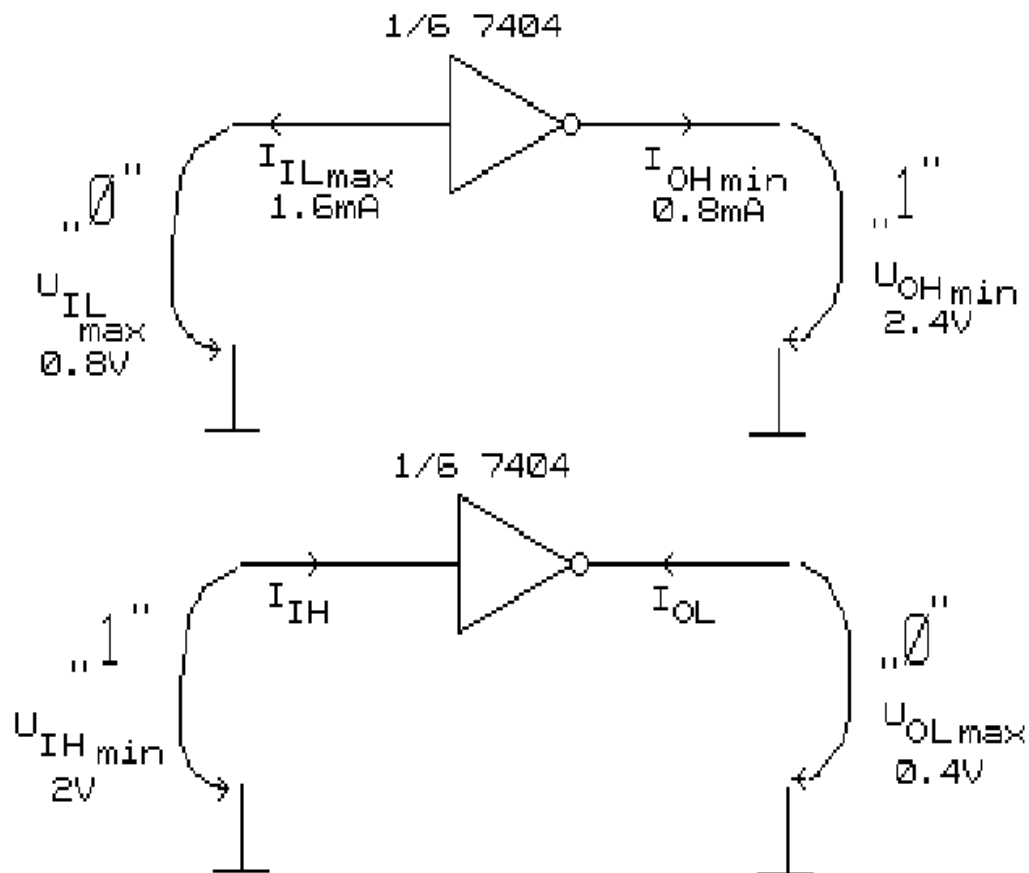


Fig.3.38 Sensurile si valorile tensiunilor si curentilor la intrarile si iesirile unei porti integrate.

2. Marginea de zgomot, M.

- M_L este marginea de zgomot la 0 logic (fig.3.39) si este diferenta intre tensiunea maxima acceptata la intrarea unui CI pentru 0 si cea maxima furnizata la iesire, pentru acelasi nivel logic.

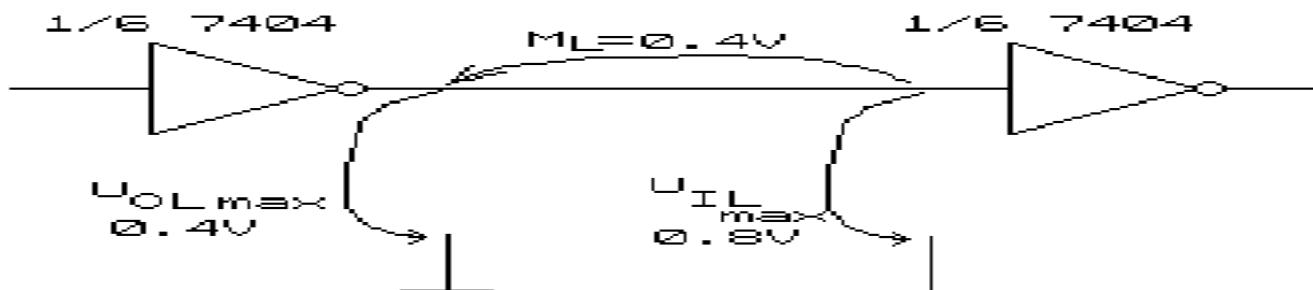


Fig.3.39. Marginea de zgomot la nivel logic 0, M_L

- M_H este marginea de zgomot la nivel logic 1 (figura 3.40). E definita ca diferenta dintre tensiunea minima furnizata la iesirea 1 a unui CI si tensiunea minima acceptata la intrare, corespunzator aceluasi nivel logic.

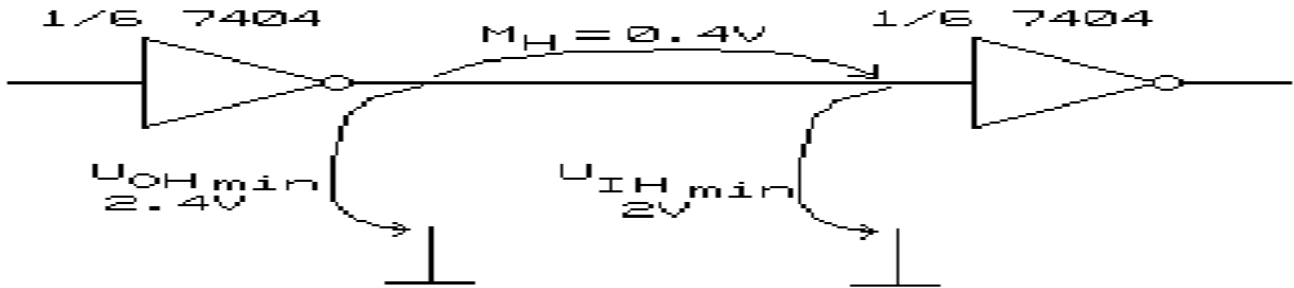


Fig.3.40 Marginea de zgomot la nivel logic 1, M_H

Aceste margini de zgomot, asigurate din proiectarea CI, garanteaza buna functionare a acestora chiar in conditiile aparitiei pe liniile de interconectare a zgomotelor electrice (datorate reflexiilor, diafoniilor etc.) manifestate prin fluctuatii ale tensiunii.

Observatie. Exista anumite scheme cu CI la care se conecteaza o rezistenta intre iesirea si intrarea unor circuite (fig.3.41 si fig.3.42).

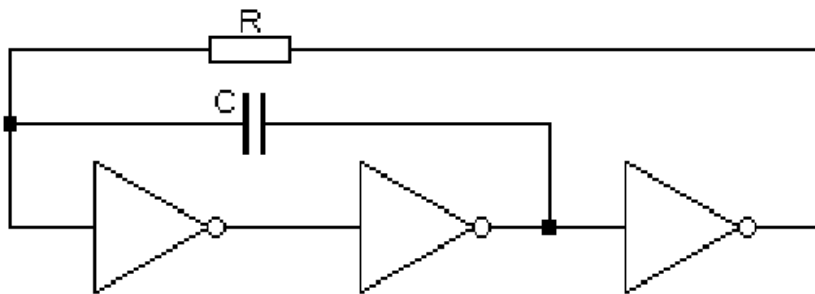


Fig.3.41 Oscilator cu inversoare (nr.impar)

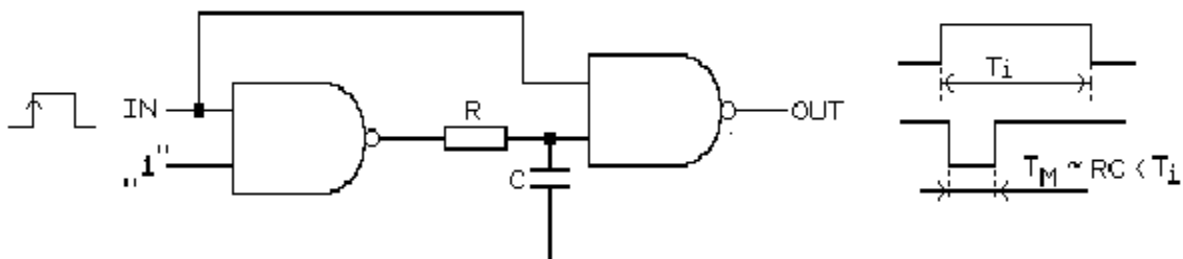


Fig.3.42 Monostabil cu porti

Circuitul genereaza cite un impuls de durata T_M (proporzionala cu $R \cdot C$) declansat de fiecare front pozitiv al impulsurilor de comanda. Conditia de buna functionare este $T_M < T_i$.

Aceasta rezistenta trebuie sa permita, la caderea de tensiune pe ea de $M_L = M_H = 0,4 \text{ V}$, inchiderea curentului maxim la intrare $I_{IL} = 1,6 \text{ mA}$. Deci R

nu poate depasi valoarea de 250Ω . Prin conectarea ei se distruge marginea de zgomot la nivel logic 0. Rezulta ca cele doua CI unite prin R trebuie cablate foarte apropiat, cu fire scurte, ecranate, eventual cu plan de masa, pentru a se elimina de tot zgomotele.

3. Caracteristica de transfer

Variatia tensiunii de iesire in functie de cea de intrare este prezentata in figura 3.43. Oscilatiile care apar daca fixam la intrarea portii o tensiune $U = 0,8 - 2 V$ sau daca parcurgem aceasta zona cu tensiune lent variabila (rezultata de exemplu prin integrarea pe condensator), sint de frecvente de ordinul MHz-ilor si se datoresc reactiei pozitive care apare in etajul de iesire TOTEM POLE, in care toti tranzistorii sint in RAN.

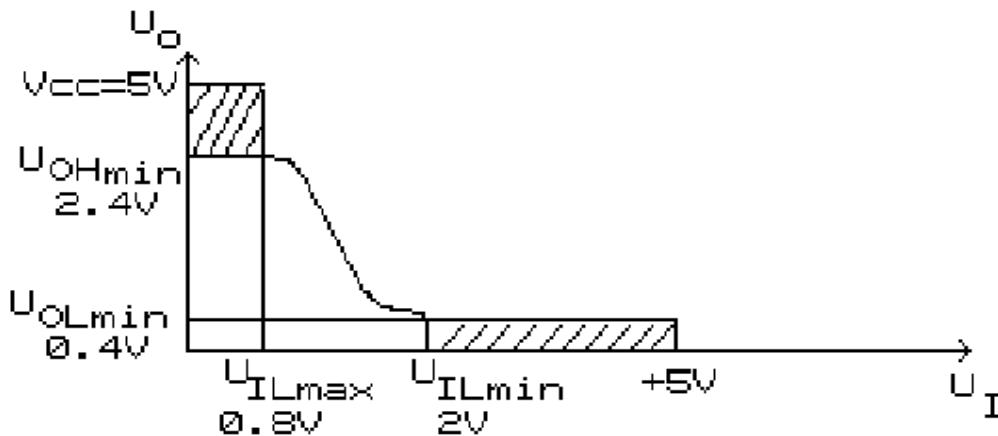


Fig.3.43. Caracteristica de transfer a unei porti TTL standard

Mentinerea unui nivel de intrare constant intre 0,8 - 2 V sau trecerea lenta prin aceasta zona a caracteristicii produc OSCILATII ale CI (AUTOOSCILATII).

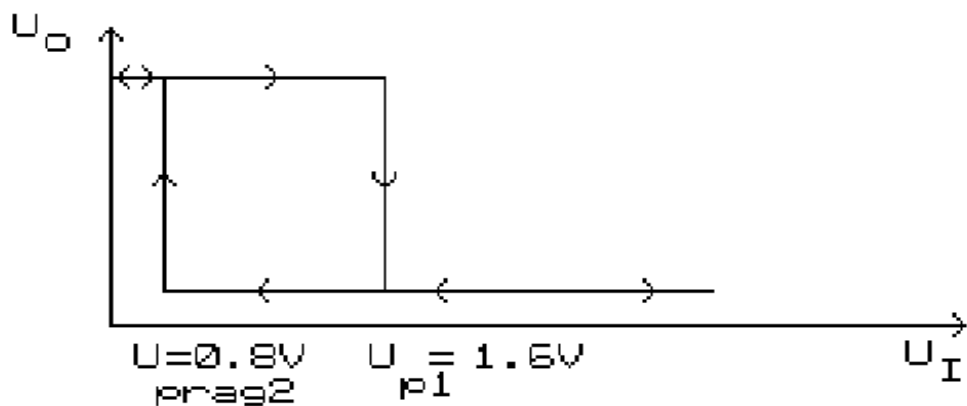


Fig.3.44 Caracteristica de transfer cu histerezis

In situatii speciale in care nu se poate evita variatia lenta a tensiunii

de la intrarea CI (exemple: la conversia analog digitala a unor variabile de proces, la schemele din fig.3.41 si 3.42 s.a.) se utilizeaza circuite speciale, cu caracteristica de transfer cu histerezis, numite trigger Schmitt (ex.7413) - fig. 3.44. La cresterea U_I , comutarea apare cind se depaseste tensiunea de prag U_{p1} . Comutarea inversa nu se mai produce la U_{p1} , ci la o tensiune de prag mai scazuta U_{p2} . In figura 3.45 este prezentata modalitatea de transformare a unui semnal lent variabil in semnal numeric.

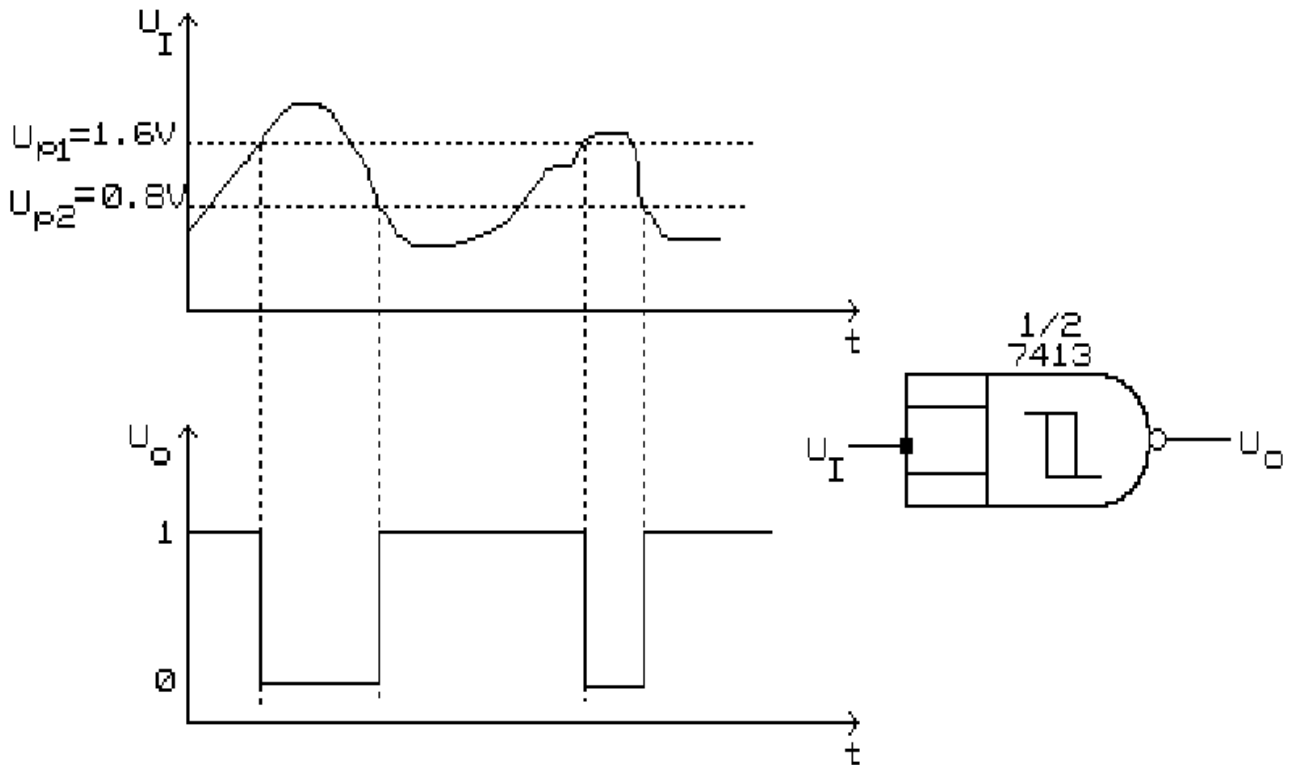


Fig.3.45 Comutarea portii inversoare cu histerezis

O alta aplicatie a acestui tip de poarta este aceea de a asigura buna functionare in cazul existentei la intrarea CI a zgomotelor de nivel mare (exemplu: circuitele receptoare de BUS).

4. FAN OUT

(avantaj de iesire). Este o marime care arata cite alte CI de acelasi tip pot fi comandate cu o poarta logica.

Se defineste FAN OUT-ul la nivel logic 0, F_L , ca raport al curentului de iesire si al celui de intrare in stare 0: $F_L = \frac{I_{OLmin}}{I_{ILmax}} = \frac{16mA}{1,6mA} = 10$ Acest raport, calculat cu datele ce specifica portile TTL standard este 10. Deci, o poarta in 0 la iesire poate comanda alte 10 porti TTL standard.

Se defineste FAN OUT-ul la nivel logic 1, F_H , ca raportul dintre curentii de iesire si de intrare in starea 1 (High = Inalt):

$F_H = \frac{I_{OHmin}}{I_{IHmax}} = \frac{0.8mA}{40\mu A} = 20$ Rezulta ca o poarta TTL standard poate comanda in starea 1 alte 20 de porti similare.

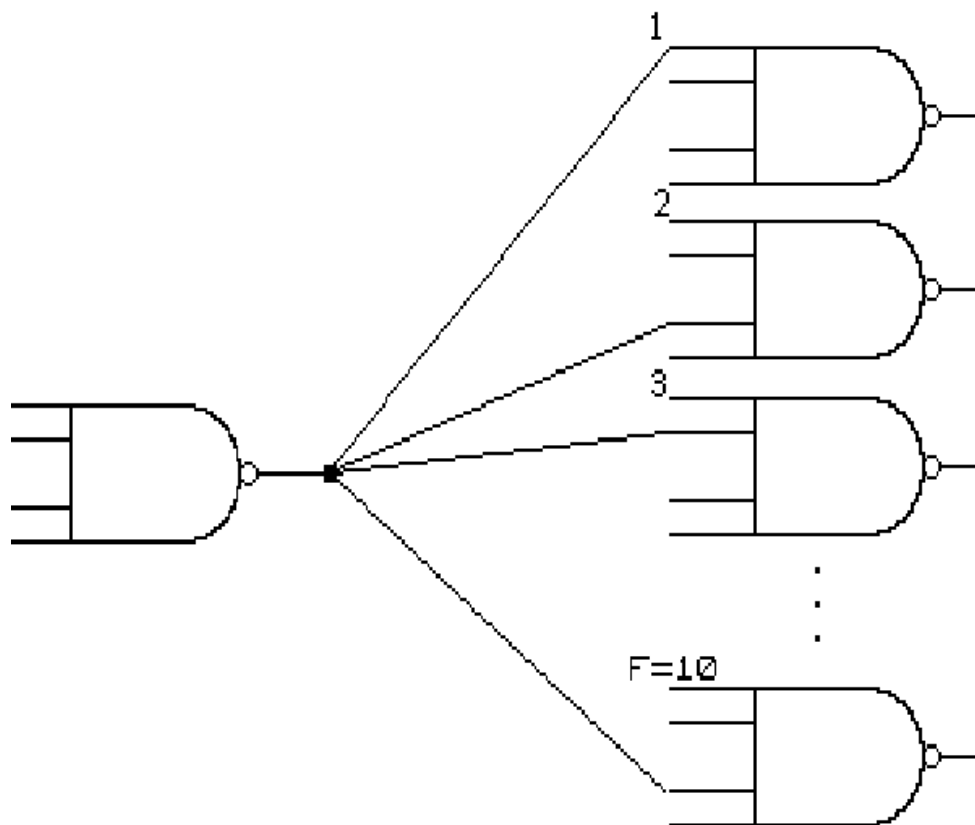


Fig.3.46 FAN OUT (evantai de iesire) - numarul de intrari ce pot fi comandate de iesirea unui CI.

Dar, cum o poarta logica este utilizata in ambele situatii (si 0 si 1 la iesire) rezulta ca incarcarea maxima este de 10 alte porti:

$$F = \min(F_L, F_H) = 10$$

In practica, in cazul proiectelor industriale, pentru a asigura o buna fiabilitate a schemelor, functionarea corecta a acestora in conditiile variatiilor de temperatura a dispersiei CI, se lucreaza cu FAN OUT maximum 8.

Denumirea foarte sugestiva a acestei marimi - FAN OUT (evantai de iesire) - este sugerata de figura 3.46.

5. FAN IN

(evantai de intrare) - reprezinta numarul de unitati de sarcina (US) cu care incarca o iesire de CI, fiecare dintre intrarile unui circuit.

Prin unitate de sarcina US se intelege un curent $I_{IL1} = -1,6 \text{ mA}$ sau $I_{IH} = 40 \mu\text{A}$ (in cazul seriei TTL standard).

Pentru circuitele integrate elementare (porti) FAN OUT-ul FI este egal cu 1US. Dar la CI mai complexe FI poate fi si mai mare de 1US. De exemplu, intrarea MC (Mode Control) a registrului 7495 si intrarile T si \bar{R} ale bistabilului D 7474 au FI = 2US. Intrarea E (Enable) a circuitului 7475, bistabil latch, are FI=4US.

6. Fronturile

sint duratele necesare unui impuls pentru a varia intre 0,1 A si 0,9 A (A - amplitudinea) ca in figura 3.47.

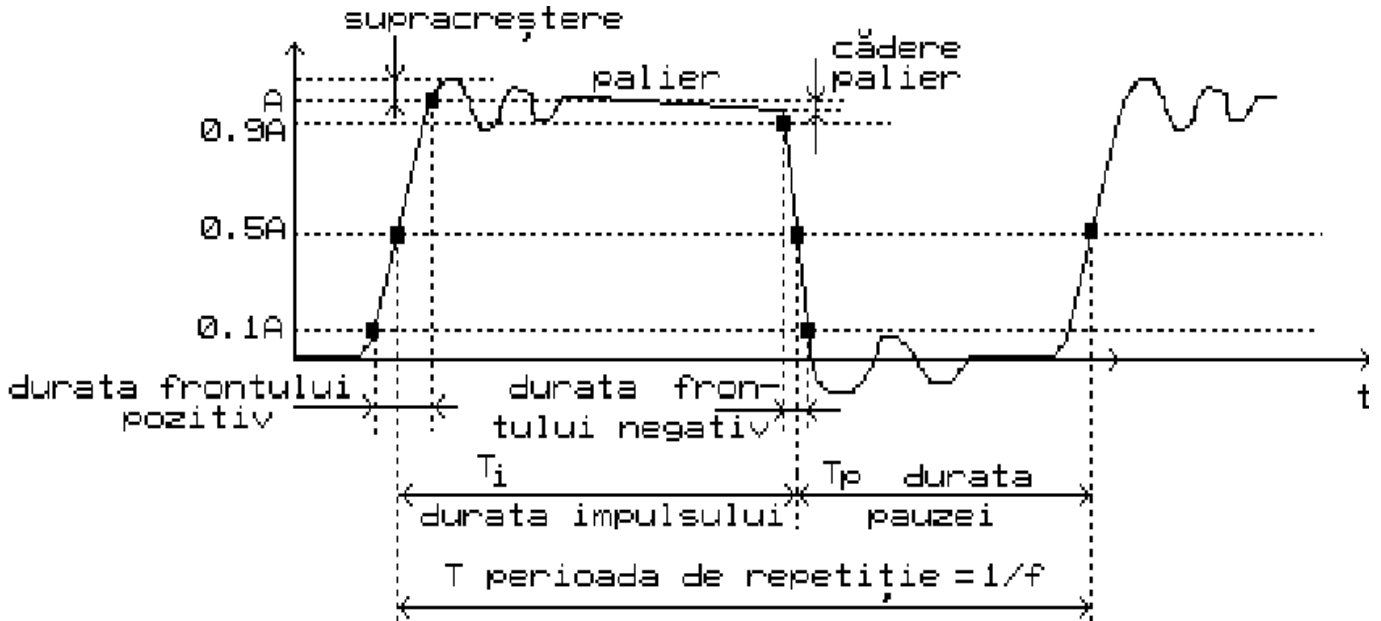


Fig.3.47 Parametrii impulsurilor reale si modul de masurare al acestora.

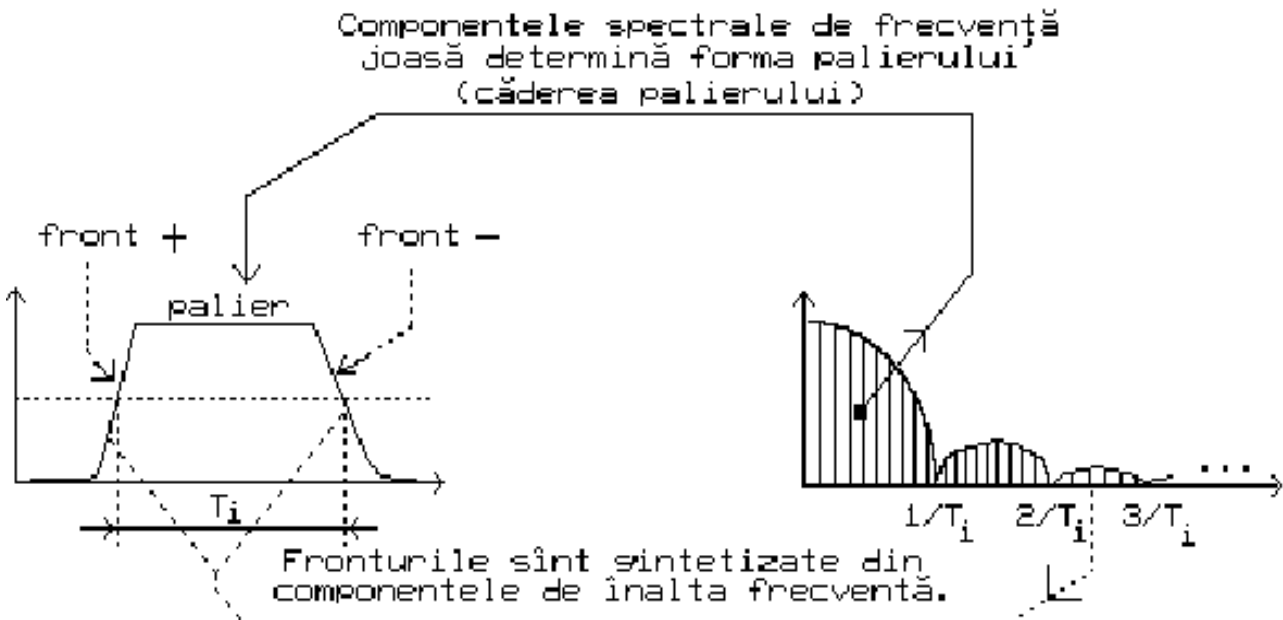


Fig.3.48 Relația între parametrii principali ai impulsului real (palier, fronturi) și spectrul acestuia.

La ieșirile CI din seria TTL standard fronturile sînt bune, de cca 7ns, aceasta datorîndu-se imbinării calitatilor inversorului T , și

repetorului T_4 din etajul final. Masurarea acestor factori se poate face comandind circuitele la frecventa mare (cu impulsuri cu T_i si T_p de acelasi ordin de marime cu fronturile) si reglind baza de timp si amplificarea astfel incit impulsul sa poata fi vizualizat pe intreg ecranul. Unele osciloscopiae au chiar marcaje la 0,1 A si 0,9 A. Pentru masuratori corecte este necesar ca osciloscopul sa aiba o banda de cel putin 50 MHz, pentru a nu taia frecventele de sus din spectrul impulsului (fig.3.48). Din figura 3.48 rezulta ca fronturile impulsului real sint mai mari ca zero datorita limitarii spectrului la o frecventa de sus, f_s , la trecerea prin circuite.

Cu ajutorul asociatiilor dintre forma impulsului si spectrul acestuia putem estima raspunsul diverselor circuite la un semnal dreptunghiular ideal aplicat la intrare (fig. 3.49).

Circuit de derivare

- C taie componentele spectrale de frecvente joase
- rezulta ca se distruge palierul impulsului

Circuit de integrare

- C taie componentele spectrale de frecvente inalte
- rezulta ca se strica fronturile impulsului

Divizorul de tensiune are: -comportare derivativa pt. $R_1C_1 > R_2C_2$
 -comportare integrativa pt. $R_1C_1 < R_2C_2$
 -comportare compensata pt. $R_1C_1 = R_2C_2$

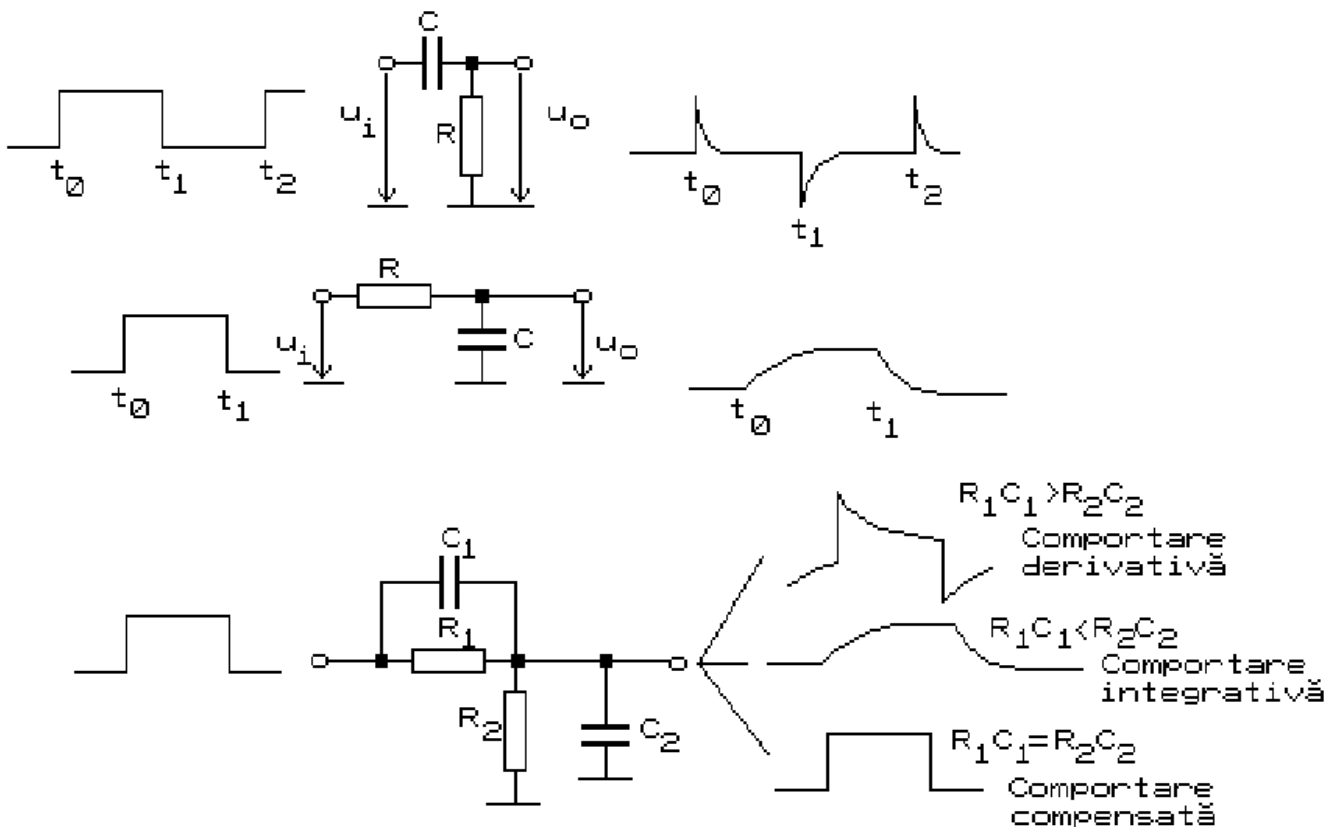


Fig.3.49 Raspunsul circuitelor RC la semnalul dreptunghiular

7. Timpii de propagare

sint intirzierile cu care se obtin raspunsurile la iesire fata de intrare. Aceste decalaje apar in special datorita timpilor de stocare si se definesc raportat la forma de unda de iesire (fig.3.50).

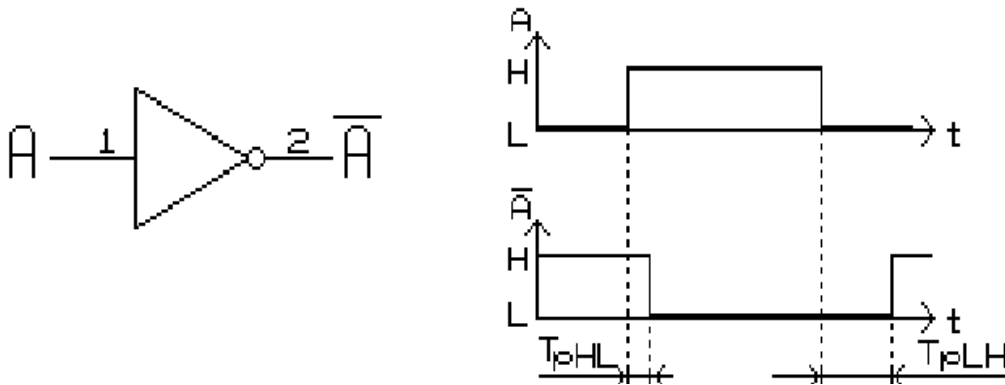


Fig.3.50 Definirea timpilor de propagare T_{pHL} si T_{pLH} pentru o poarta inversoare

Se observa, consultind cataloagele, ca la seria TTL standard circuitele elementare au $T_{pLH}=12$ ns de cca. 2 ori mai mare decit $T_{pHL}=7$ ns. Aceasta se datoreste in principal faptului ca, la tranzitia din L in H comuta succesiv din saturare doua tranzistoare (T_2 si T_4 din figura 3.36), in timp ce, la tranzitia din H in L intervine timpul de comutare al unui singur tranzistor (T_3 din 3.35). Aceasta diferenta intre T_{pLH} si T_{pHL} la seria TTL standard constituie unul din principalele motive ale utilizarii preferentiale in implementare a portilor inversoare fata de portile neinversoare (fig.3.51 si fig.3.52).

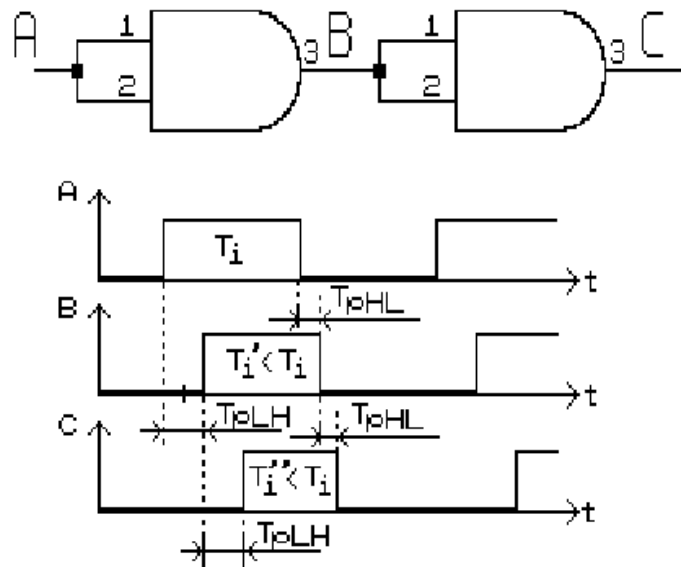
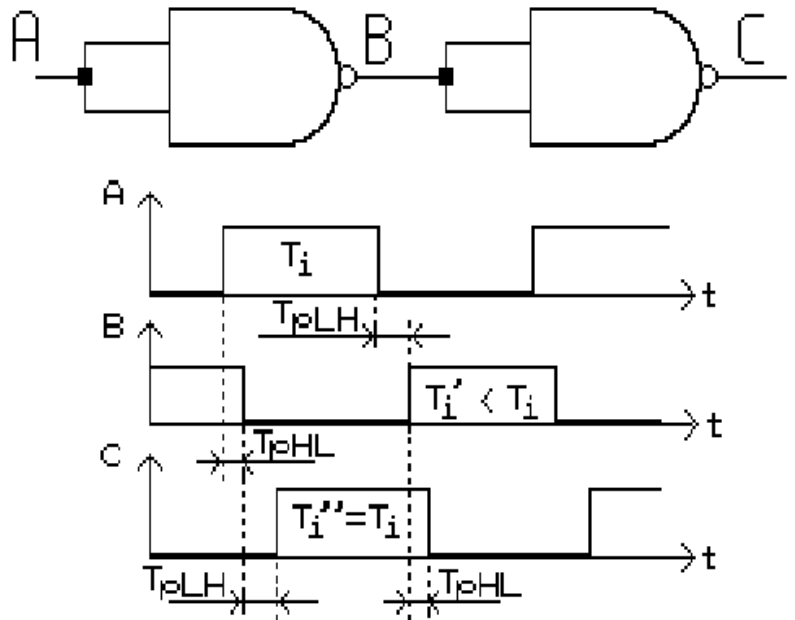


Fig.3.51 Prin inserierea mai multor porti neinversoare, durata impulsului T_i scade cu cca 5 ns pe fiecare poarta.

$$T_i' = T_i - (T_{pLH} - T_{pHL})$$

$$T_i'' = T_i - 2(T_{pLH} - T_{pHL})$$

Fig.3.52 Dupa un numar par de etaje inversoare se reface durata impulsului de la intrare.



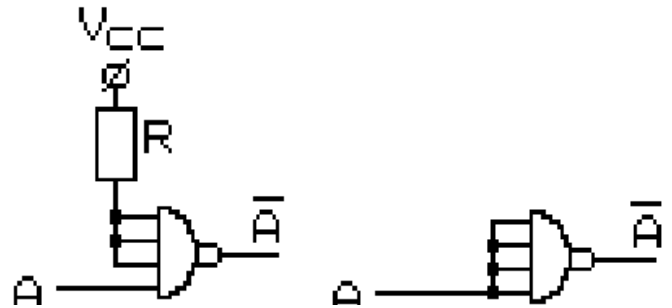
Este important de observat ca in implementarile cu porti in tehnologie TTL standard se utilizeaza SI NU-uri in majoritatea situatiilor datorita urmatoarelor avantaje:

- exista o mare diversitate de porti SI NU integrate (cu 2,3,4,8,12 intrari) din cauza extinderii simple a numarului de intrari doar prin adaugarea de emitoare la T_1 (fig.3.35).
- poarta SI NU in sine este mai rapida decit alte tipuri de porti (SI, SAU EXCLUSIV) care se obtin din aceasta prin adaugarea de tranzistoare. De exemplu, pentru obtinerea functiei SI se completeaza schema din figura 3.35 cu inca un tranzistor inversor, plasat inaintea etajului de iesire. Deci cresc timpii de propagare t_{PLH} si t_{PHL} prin aceasta poarta.
- dupa un numar par de etaje de SI NU-uri se reface durata impulsului de intrare. Deci schemele cu SI NU-uri pot functiona la frecvente mai mari decit cele cu porti neinversoare (SI-uri, SAU-uri).
- teoremele lui De Morgan permit implementarea cu SI NU-uri a oricarei functii logice. Sint necesare 3 niveluri de porti:
 - I - inversoare (sau SI NU-uri conectate ca inversoare) pentru negarea variabilelor de intrare ale functiei
 - II - primul nivel de SI NU-uri pentru realizarea termenilor functiei (negati)
 - III - al 2-lea nivel de SI NU-uri pentru reunirea termenilor constituenti ai functiei.

Trebuie observat ca, la utilizarea SI NU-urilor ca inversoare (fig.3.54b), nu creste FAN IN-ul de la "0" prin conectarea in paralel a intrarilor aceleesi porti, deoarece curentul este limitat de rezistenta de baza si U_{BE} a tranzistorului multiemitor T_1 (fig.3.35).

Va creste, proportional cu numarul de intrari doar FAN IN-ul de la nivel logic 1, deoarece trebuie asigurati curentii reziduali ai jonctiunilor BE.

Fig.3.54 Conectarea ca inversor a unei porti SI NU.

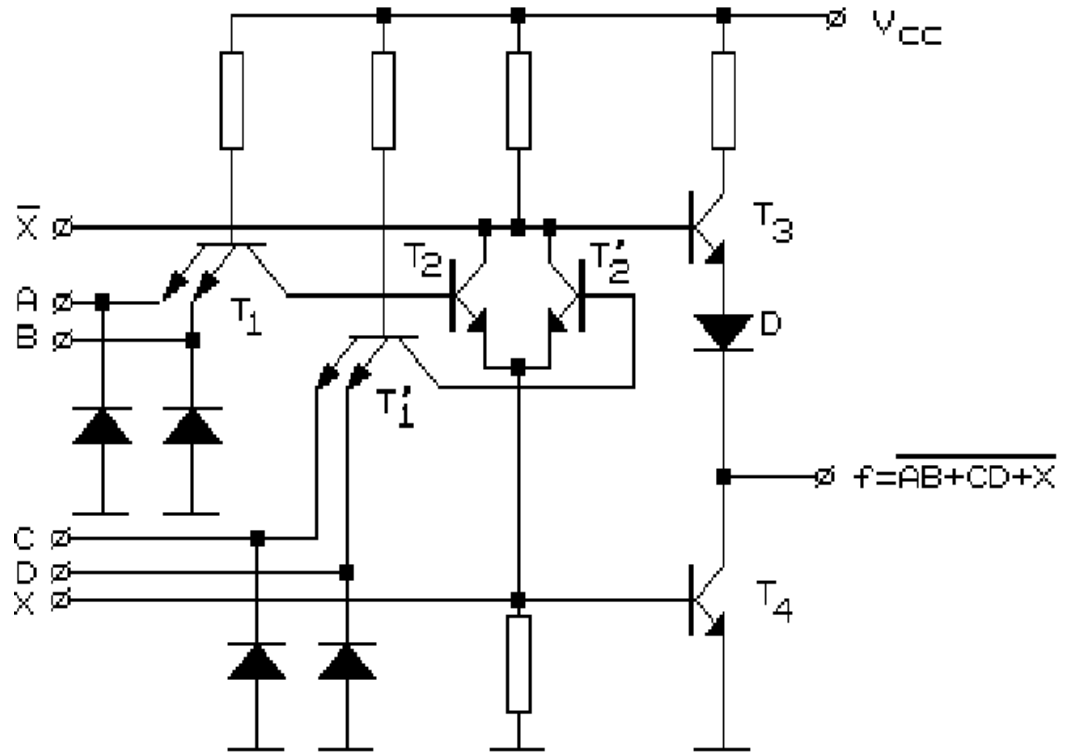


Se fac abateri de la regula utilizarii SI NU-urilor in implementare numai in situatii bine justificate, in care prin aceste abateri se obtine viteza mai mare de lucru sau reducerea numarului de CI. De exemplu, porti XOR se pot folosi la implementarea sumatorului binar complet, reducind-o ca numar de CI.

Poarta SI SAU NU (AND OR INVERT) cu expandare

Functia SI se obtine cu tranzistoarele multiemitor, iar functia SAU NU rezulta prin conectarea in paralel a tranzistoarelor T_2 si T_2' (fig.3.55)

Fig.3.55 Poarta SI SAU NU cu expandare (1/2 7450).



Functia SAU NU se poate extinde prin conectarea la X si X' a altor tranzistoare, de exemplu din circuitul expandor 7460 care are un SI cu 4 intrari (T" multi-emitor) ce comanda pe T" cu C si E accesibile exterior.

Etaje de iesire din circuitele TTL

1. Etajul de iesire TOTEM POLE :

este format din tranzistoarele T_3 , T_4

Aceste etaje de iesire nu pot fi conectate in paralel: daca unul dintre etaje ar avea T_4 saturat, acesta ar scurtcircuita practic la masa celelalte iesiri conectate in paralel; s-ar depasi curentul maxim admisibil al iesirilor, conducind la distrugerea tranzistoarelor

2. Etajul de iesire OPEN COLLECTOR (cu colector in gol)

permite conectarea iesirilor in paralel, prin eliminarea din schema a tranzistorului T_3 si a diodei D din etajul TOTEM POLE (fig.3.56).

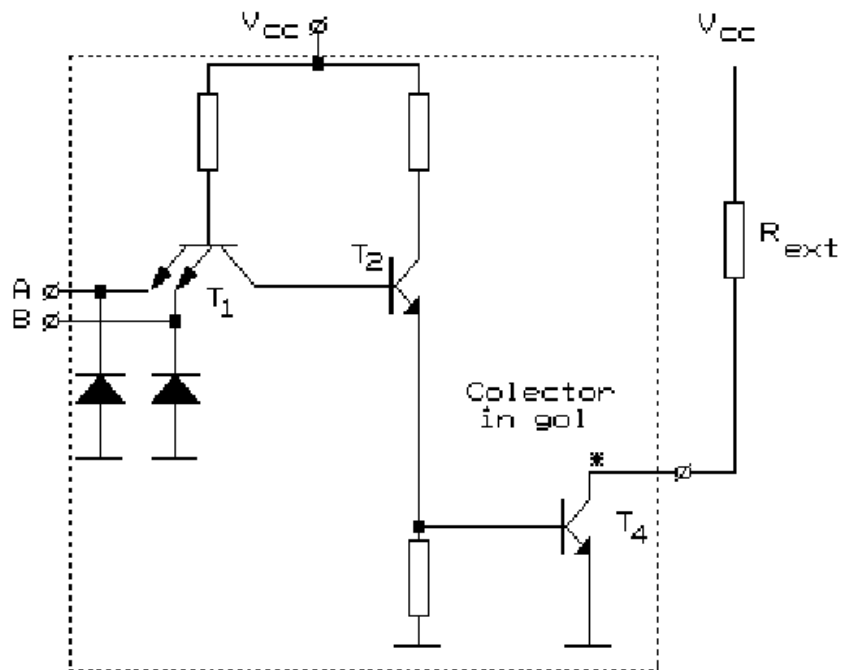
Fig.3.56 Poarta SI NU cu colectorul in gol 7401

Prin conectarea iesirilor in paralel se realizeaza functia SI cablat.

Tranzistoarele T_4 conectate in paralel sint alimentate de la V_{cc} printr-o rezistenta exterioara a circuitului integrat (fig.3.56).

Dimensionarea R_{ext} se face in functie de numarul portilor OC conectate in paralel, de circuitele de sarcina, si de cele doua stari "0" si "1". R_{ext} trebuie sa aiba ori valoare cit mai mica

pentru a nu inrautati mult frontul de blocare, ori cit mai mare, pentru reducerea consumului. Datorita inlaturarii repetorului din etajul de iesire TOTEM POLE, circuitele OC au aceleasi dezavantaje cu inversorul cu tranzistor.



3. Etajul de iesire THREE STATE (CU TREI STARI)

TS, inlatura dezavantajul anterior. Se pastreaza etajele de iesire TOTEM POLE, conectarea acestora in paralel fiind permisa datorita starii a 3-a, in care ambele tranzistoare finale sint blocate (fig.3.57). Cind $\bar{I}=0=0$, ($I=1$), inversorul N este blocate si elementele adaugate in schema, emitorul E_2 al lui T_1 si dioda D_T nu influenteaza functionarea portii care poate avea la iesire 2 stari logice 0 si 1 ($A \rightarrow \bar{A}$). Daca $\bar{I}_1=1$, inversorul N se deschide punind practic la masa emitorul E_2 si catodul D_T . Deoarece $E_2=I=0$, se blocheaza tranzistoarele T_2 si T_4 . Dioda D_T fixeaza tensiunea in baza lui T_3 la un nivel sub cel necesar deschiderii. Avind ambele tranzistoare finale blocate (T_3 si T_4) poarta se gaseste in starea a 3-a, stare de inalta impedanta (HiZ). Integratele cu iesire TS se pot conecta deci in paralel, dar numai cu conditia de a fi dezinhitate selectiv, doar cite unul odata, restul ramainind in TS. Altfel, se ajunge la conectarea nepermisa a unor etaje TOTEM POLE in paralel.

In cele ce urmeaza se vor prezenta doua aplicatii ale CI TS.

Aplicatia 1. Multiplexare cu CI TS

Se va considera exemplul unui MUX 4:1 (fig.3.58). Sint necesare 4 inversoare TS conectate in paralel la iesire pentru concentrarea celor 4 cai de date D_0-D_3 pe o singura cale W. Selectia MUX se va face prin comanda demultiplexata a iesirii din TS.

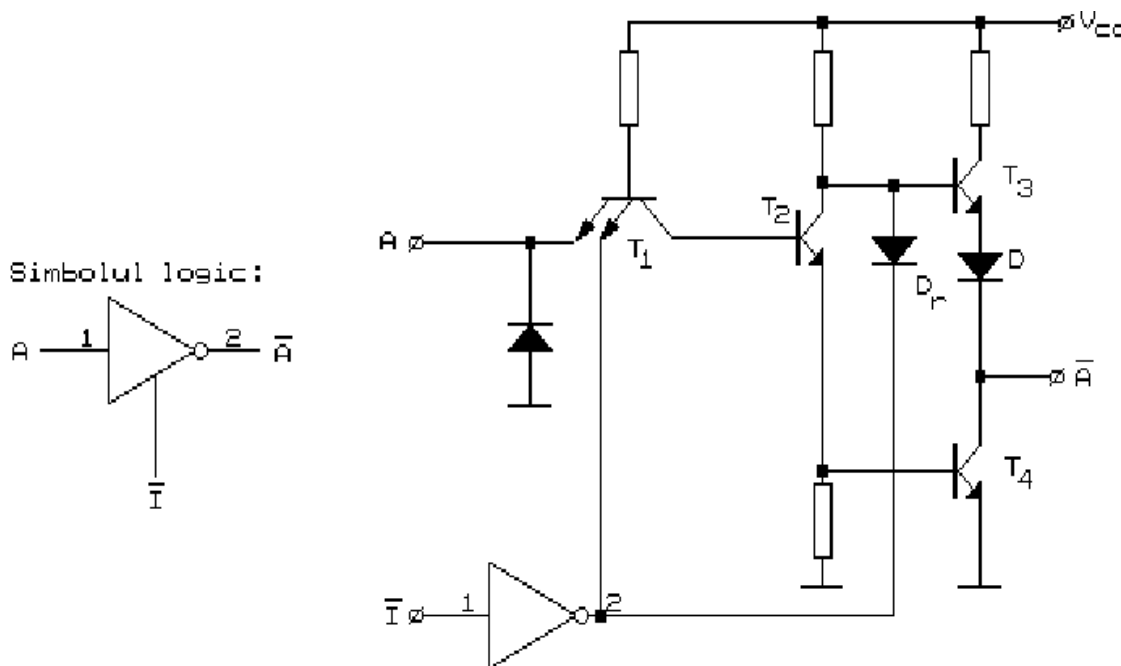


Fig.3.57. Inversorul TS

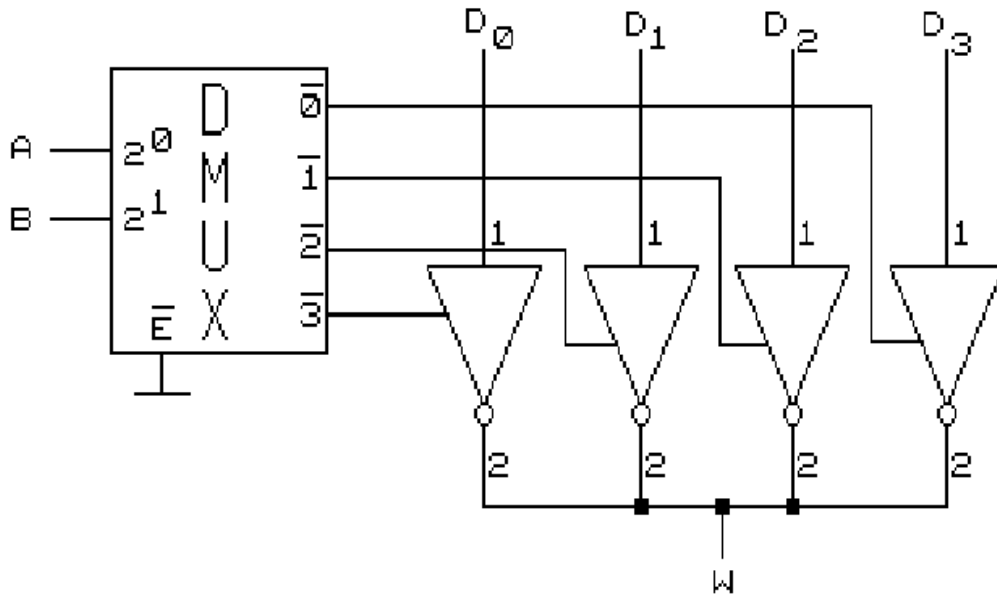


Fig.3.58. MUX 4:1 realizat cu 4 inversoare TS a caror dezinhibare e comandata demultiplexat

Aplicatia 2. Realizarea magistralelor bidirectionale: BUS-uri

BUS-ul este o modalitate de interconectare intre sisteme, care optimizeaza numarul de legaturi prin ideea transmiterii bitilor pe cai comune bidirectionale. Consideram cazul interconectarii a 4 sisteme (fig.3.59).

Daca fiecare comunica in ambele sensuri si ca emitator si ca receptor cu toate celelalte sint necesare 12 conexiuni. Acestea se pot reduce la un singur fir bidirectional prin optimizarea numarului de conexiuni ca in figura 3.60. Daca unul din sisteme, de exemplu S₁ este emitator, oricare dintre celelalte fiind receptoare, sensul de transfer pe BUS este de sus in jos.

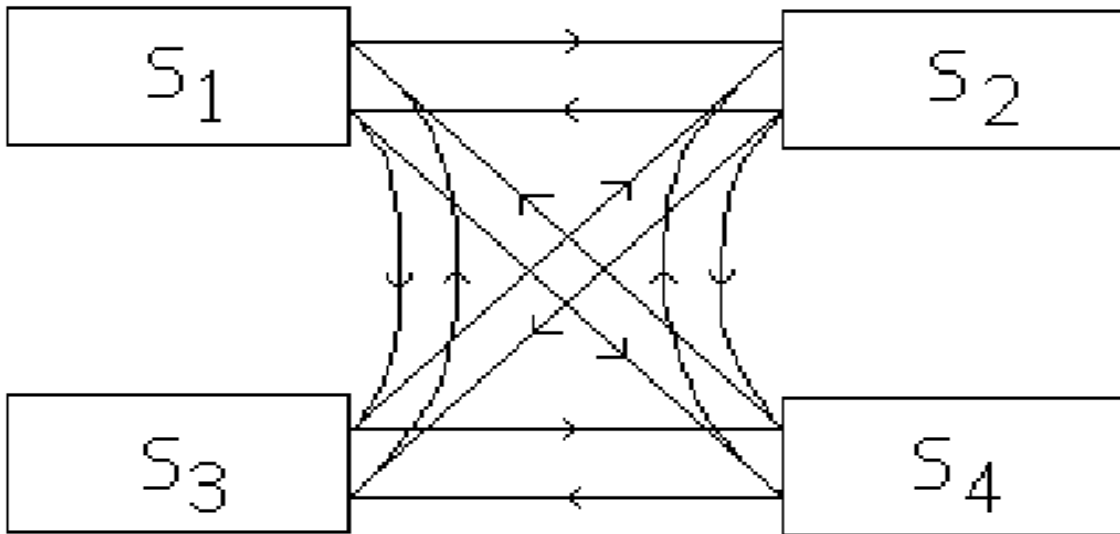


Fig.3.59 Interconectarea pe 1 bit a 4 sisteme presupune 12 legaturi

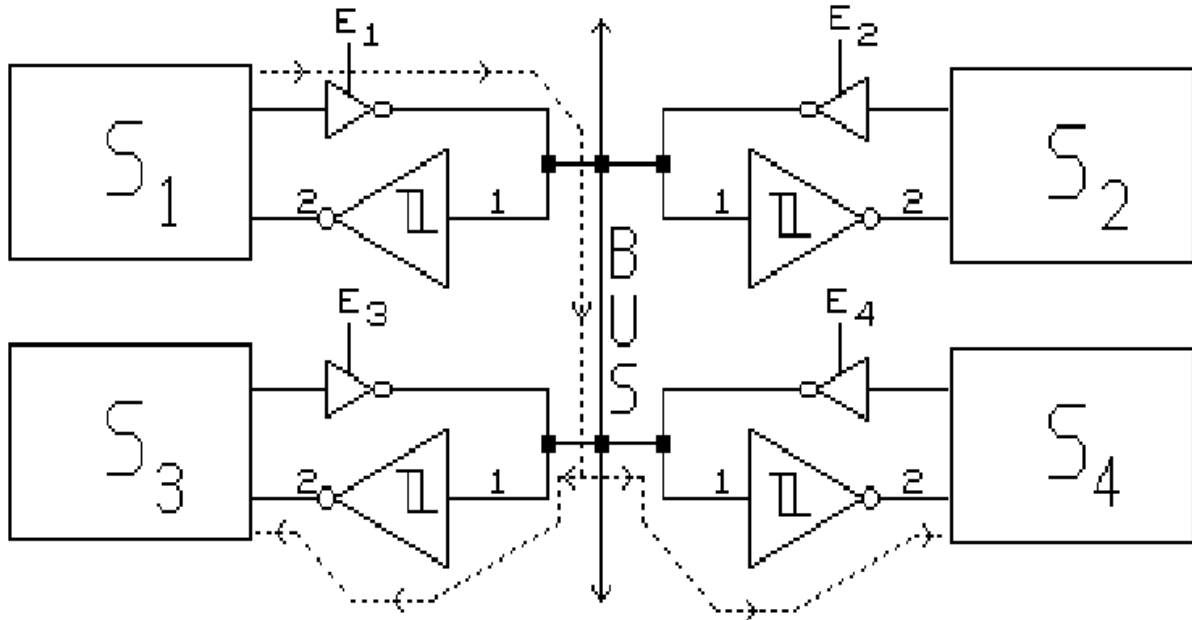


Fig.3.60 Interconectarea pe BUS a 4 sisteme presupune o singura conexiune bidirectionala.

Daca S_4 este emitor ($\overline{E_4}=0$) si S_1 sau/si S_2 sint receptoare, sensul de transfer pe BUS se inverseaza.

Generalizind pentru n sisteme, rezulta ca numarul de conexiuni necesare pentru fiecare bit se reduce de la $A_n^2=n(n-1)$ la un singur fir bidirectional prin utilizarea BUS-ului.

Conditile de functionare corecta a BUS-ului sint:

- In fiecare moment sa fie deschis un singur emitor. Toate celelalte sisteme pot fi receptoare. Notatiile cu bare a semnalelor de comanda a emisiei E se refera la faptul ca acestea sint active pe nivel logic 0.
- Emitatoarele BUS-ului nu trebuie sa fie neaparat circuite TS; se pot utiliza si porti cu colectorul in gol (OC).
- Receptoarele de BUS trebuie sa fie inversoare sau porti SAU NU cu intrari trigger Schmitt pentru a avea o imunitate mare la zgomote. Nu este indicata utilizarea portilor SI sau SI NU deoarece acestea induc zgomote pe BUS: in tranzistorul multiemitor de la intrare (fig.3.35), curentul emitorului conectat la BUS este influentat de comutarile care au loc pe celalalt emitor.
- Linile de BUS se comporta ca linii lungi care trebuie adaptate pentru a nu apare reflexii periculoase pentru interpretarea nivelurilor logice. Impedanta caracteristica a acestor linii este de $75\Omega - 150\Omega$, dependent de cablajul realizat, de existenta planului de masa. Adaptarea se obtine plasind la extremitatile BUS-ului divizoare rezistive R_1, R_2 , avind valori de ordinul sutelor de ohmi (fig.3.61). In aceste conditii, adaptarea perfecta se obtine doar la capetele BUS-ului. Receptoarele conectate pe parcurs nu lucreaza bine adaptate. Din acest motiv apar zgomote datorate reflexiilor. Acestea nu vor influenta receptia daca circuitele au imunitate crescuta la zgomote, adica au intrare de trigger Schmitt.

Adaptarea s-ar fi putut realiza si numai cu rezistentele R_1 care ar fi

trebuie sa aiba valori mici, egale cu impedanta liniei. In acest caz, consumul de curent al BUS-ului este mare. Poate fi redus prin adaptarea cu divizor rezistiv: R_1, R_2 trebuie sa egaleze rezistenta caracteristica, fiecare dintre ele avind deci valoare mai mare. In starea logica 0, emitatoarele de bus trebuie sa suporte curenti importanti (50-100 mA) asigurand in principal curentii prin cele 2 rezistente R_1 .

De exemplu pentru un BUS cu impedanta caracteristica $Z_0=150\Omega$ si 16 receptoare, rezulta $R_1=250\Omega, R_2=375\Omega$ si curentul in "0" de cca 50 mA.

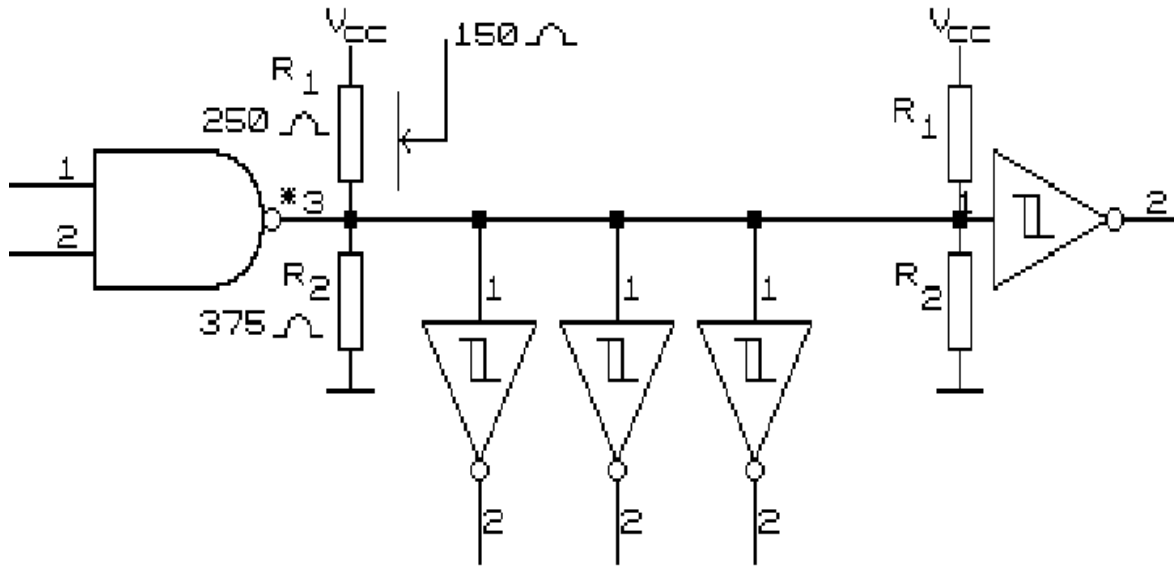


Fig.3.61 Adaptarea BUS-ului cu divizoare rezistive R_1, R_2 .

Familii de CI TTL

In cele prezentate anterior s-au facut in principal referiri la seria TTL standard. Fata de aceasta au aparut diverse perferctionari in vederea atingerii urmatoarelor scopuri:

- . cresterea vitezei de lucru
- . scaderea consumului de putere

Pentru reducerea timpilor de propagare s-au modificat schemele interne. In etajul final, tranzistorului T_3 (fig.3.35) i s-a adaugat T'_3 in montaj Darlington (fig.3.62) in cazul CI HTTL.

La familia TTL Schottky (STTL) s-a pastrat aceasta idee si viteza a fost sporita in plus prin utilizarea diodelor Schottky si adaugarea acestora in paralel cu jonctiunea BC a tuturor tranzistoarelor in scopul eliminarii rapide a sarcinii stocate in baza (fig.3.63).

Se obtine astfel o echilibrare a timpilor de propagare in paralel cu reducerea lor: $T_{PHL} = T_{PLH} = 3ns$.

Scaderea consumului de putere s-a realizat la seriile TTL LP in principal prin marirea valorilor rezistentelor din schema interna a porturilor.

In concurenta cu tehnologia CMOS care devine tot mai rapida la consum redus s-au dezvoltat noi serii de CI TTL: ASTTL (Advanced Schottky TTL) si FTTL (Fast TTL).

In tabelul urmatoare se vor prezenta comparativ diferitele familii de CI TTL si CMOS.

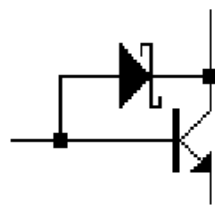


Fig.3.63 Se reduce timpul de stocare prin conectarea diodei Schottky in paralel pe jonctiunea BC.

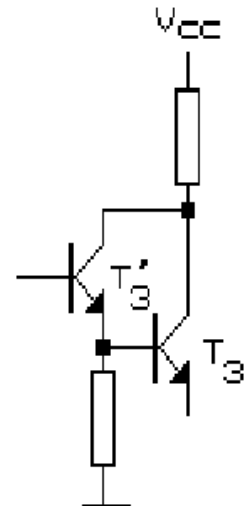


Fig.3.62 Adaugarea tranzistorului T'_3 in montaj Darlington la circuitele HTTL.

Familii de CI	TTL standard	Schottky TTL	LPSTTL	ASTTL	FTTL	CMOS (MMC)	HCMOS
Parametru							
Disiparea de putere pe poarta [mW]	10	19	2	8.5	5.5	0.001	$2.5 \cdot 10^{-6}$
Timp de propagare tipic [ns]	10	3	10	1.5	3	40	7
Frecventa maxima de lucru [MHz]	25	100	33	160	125	12	55

Se observa ca in timp ce la STTL cistigul de viteza este platit in consum mare de putere, la LPSTTL, puterea disipata e mica dar timpul de propagare este acelasi cu cel de la seria TTL standard, la noile serii ASTTL si FTTL, raportul intirziere/putere disipata este deosebit de favorabil.

Famiiliile de CI care trebuie luate in considerare in proiectare sint evident:

- . ASTTL si FTTL cind se doreste viteza mare,
- . HCMOS pentru consum redus.

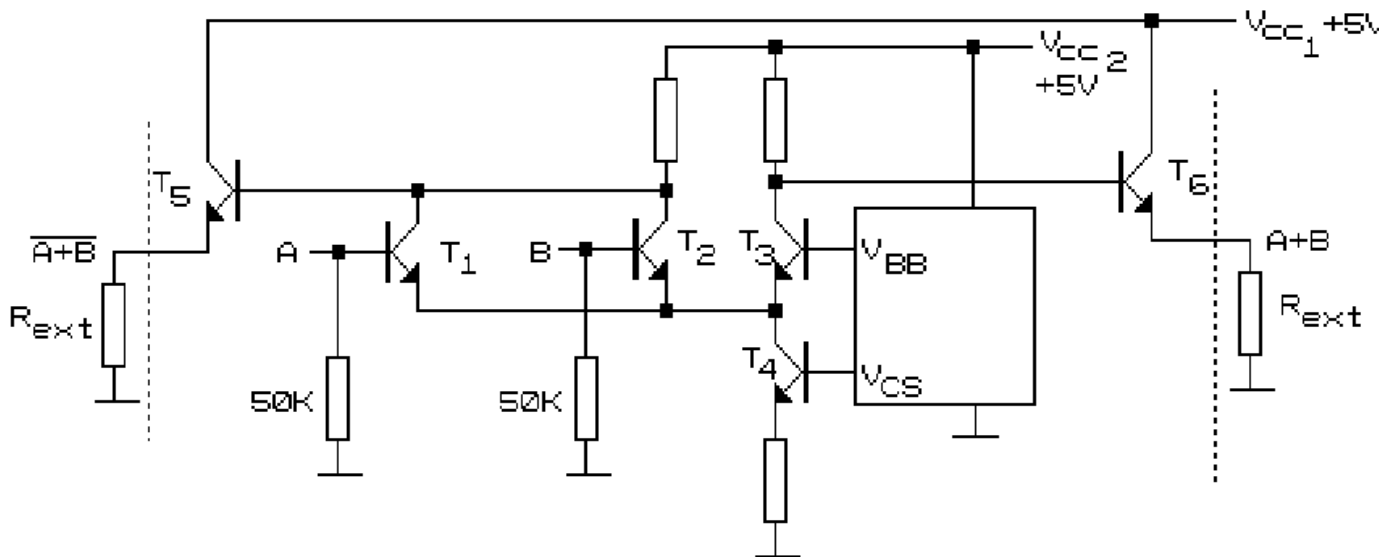


Fig. 3.64. Schema interna a unei porti ECL.

3.2.4.4. Circuite integrate ECL

Prin tehnologia ECL (logica cuplata prin emitor) se obtin CI de mare viteza, cu timpi de propagare de $1ns$ si, drept urmare, cu consum crescut de putere, $25mW/poarta$. In aceasta tehnologie se realizeaza simplu porti SAU NU si SAU (fig.3.64).

Functionarea schemei este urmatoarea: daca cel puțin una dintre intrarile A si B se afla la nivel logic 1 ($-1V$), tranzistoarele T_1, T_2 se deschid, curentul lor este suportat de T_4 . In acest timp, T_3 este blocat. In baza lui T_6 este injectat curent si tranzistorul se deschide, furnizind la iesire 1 logic ($-1V$). T_5 , neprimind curent de baza se blocheaza trecind in 0 logic ($-1,6V$).

Cind $A=B=0$, T_1, T_2 se blocheaza, curentul ce trecea prin ele este injectat in baza lui T_5 care se deschide (linia punctata din figura 3.64.) asigurand in emitor tensiunea de $-1,6V$ (0 logic). Curentul lui T_4 este obtinut acum prin deschiderea lui T_3 . T_6 se blocheaza din lipsa curentului de baza. In acest caz, iesirea $A + B = 0$ ($-1,6V$).

Comutarea rapida, datorata utilizarii repetoarelor, a surselor de curent constant, impune precautii speciale de separare a alimentarii schemei logice, V_{CC2} , de cea a etajului final, V_{CC1} . Astfel, eventualele zgomote aparute pe V_{CC1} din cauza variatiilor mari de curent prin T_5 si T_6 , nu pot influenta functionarea logica a circuitului. Repetoarele T_5 si T_6 asigura impedanta mica de iesire, in medie de cca 7Ω . Prin utilizarea generatoarelor de curent constant se obtin efecte importante privind marirea impedantei de intrare si a vitezei de comutare.

O caracteristica importanta a acestui tip de circuite este aceea ca rezistentele exterioare R_{ext} au valori variabile, calculate in functie de contextul in care functioneaza poarta (pozitia ei pe cablaj, impedanta caracteristica a liniei care o leaga de circuitele pe care le comanda). Trebuie specificat faptul ca este obligatorie realizarea cablajului pentru CI ECL numai cu plan de masa.

Valorile uzuale pentru rezistentele din emitor sint de ordinul sutelor de ohmi.

3.2.4.5. Circuite integrate I²L (Integrated Injection Logic)

Circuitele obtinute in aceasta tehnologie sint caracterizate prin faptul ca nivelurilor logice 0 si 1 le corespund valori distincte de curent, respectiv:

0 inseamna existenta unui curent de conductie (tranzistor blocat)
 1 inseamna lipsa de curent

In figura 3.63 este prezentata schema unei celule inversoare I²L. Tranzistorul Tg este generator de curent.

Cind curentul acestuia este absorbit de intrarea A (I_i), deci cind A=0, tranzistorul T, neprimind curent pe baza, este blocat. Rezulta I₂=0, adica A'=1.

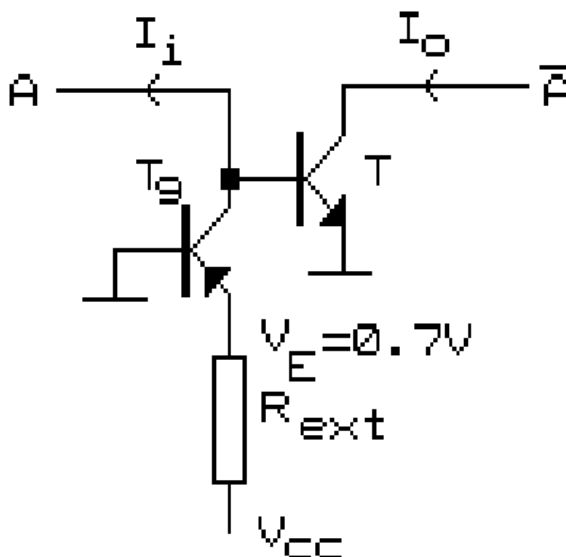


Fig.3.63. Inversor I²L. (III).

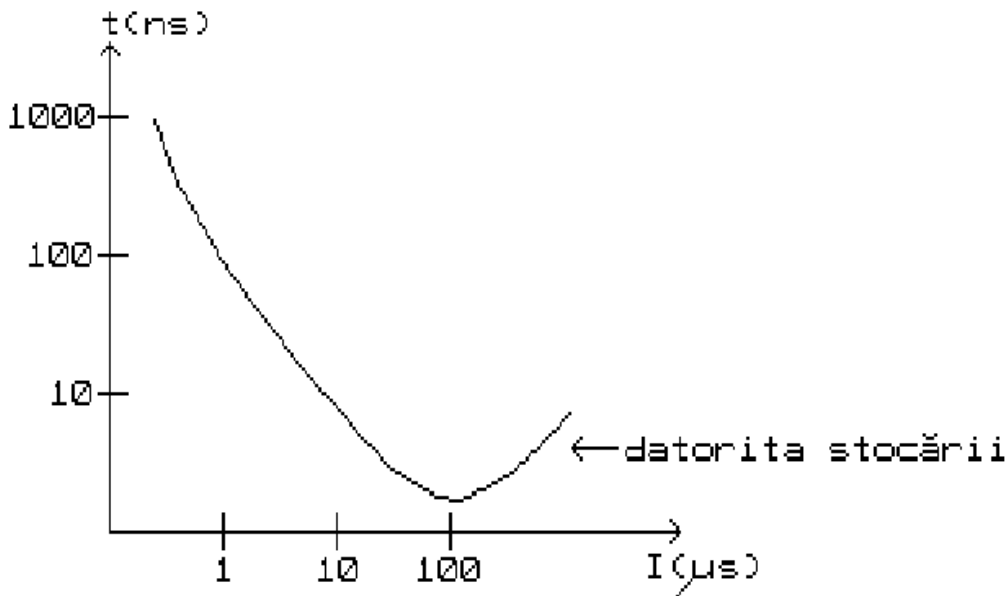


Fig.3.64 Timpul de propagare al portilor I²L descreste prin marirea curentului injectat in generatoare.

Invers, cind A = 1 si nu se absoarbe curent pe intrare (I_i = 0), curentul generatorului Tg este injectat in baza lui T care va conduce in saturatie. In acest caz A'=0.

Prin rezistenta exterioara R_{ext} se injecteaza curent din sursa V_{cc}. R_{ext} este comuna pentru toate emitoarele generatoarelor de curent din scheme complexe I²L. Prin reglarea valorii acestei rezistente se stabileste curentul injectat de generatoare. De acesta depinde timpul de comutare al portilor (fig.3.64): cu cit se injecteaza mai mult curent, cu atit creste viteza de lucru a portilor I²L.

Acest fenomen este inregistrat pina la curenti de cca 100 μA,

intervenind apoi procesele de stocare ce maresc valorile timpilor de propagare.

Curentii din scheme fiind curenti de baza au valori foarte mici: uzual zeci de microamperi. Puterea disipata pe poarta este de zeci de microwati. Astfel, pentru circuite cu cca 1000 de porti I²L se disipa puteri de zeci de miliwati. Rezistenta exterioara se regleaza pentru a putea asigura injectia de curent necesara circuitelor.

Cu ajutorul structurilor de inversor conectate in paralel se obtin simplu porti logice SAU NU si SAU (fig.3.65).

Cascadarea inversoarelor pentru a realiza SI NU-uri nu este posibila deoarece nu asigura scheme uniforme pentru generatoarele de curent T_g.

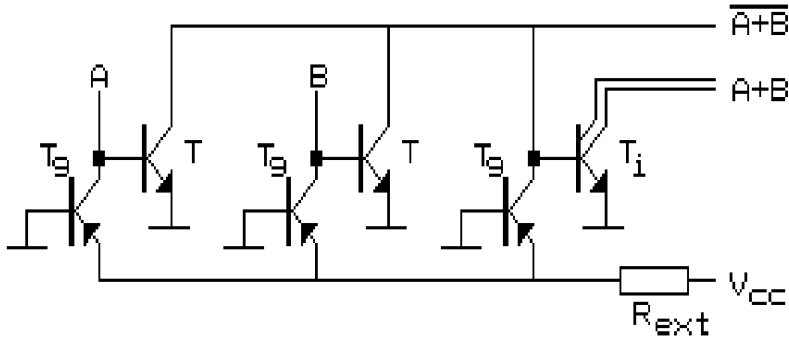


Fig.3.65 Poarta SAU NU si SAU in tehnologie I²L.

Pentru cresterea fan out-ului se utilizeaza tranzistoare multicolector (T_i in figura 3.65). Acestea trebuie sa aiba β suficient de mare pentru a permite cresterea curentului in sarcina fara a scoate tranzistorul din saturatie.

Tehnologia I²L are avantajul implementarii unor functii logice complexe cu

circuite relativ simple. Cu toate acestea, in prezent se utilizeaza in special la interfata digital analogic.

3.2.4.6. Circuite integrate CMOS

Tehnologia unipolara s-a impus in domeniul CI digitale in primul rind datorita consumului redus de putere pe poarta: cel putin trei ordine de marime comparativ cu portile bipolare. Primele circuite realizate cu tranzistoare pMOS (MOS cu canal p) au fost dezavantajate de existenta mai multor surse de alimentare, incompatibilitatea cu CITTTL, viteza redusa de raspuns. Prin trecerea la utilizarea tehnologiei nMOS (cu canal n) s-au inlaturat primele doua inconveniente, ramainind insa acela al frecventei joase de lucru.

In 1968, firma RCA din SUA a produs comercial primele circuite CMOS (Complementary MOS), folosind tranzistoare complementare: cu canal n si p.

Favorizate de parametri ca:

- . imunitate mare la zgomot
- . caracteristica de transfer aproape ideala
- . consum de putere redus

Circuitele CMOS au avut de invins handicapul vitezei lente de comutare. Prin perfectionari tehnologice s-a obtinut reducerea consumului de putere, concomitent cu scaderea spectaculoasa a timpilor de propagare. S-a realizat astfel familia HCMOS care reprezinta azi un concurent serios pentru circuitele LPSTTL, asa cum rezulta din tabelul comparativ din capitolul 3.2.4.3. Competitia CMOS-TTL a generat preocupari de crestere a vitezei, concretizate in aparitia familiilor ASTTL si FTTL. Totusi se pare ca viitorul apartine circuitelor CMOS.

In figura 3.66. este prezentata schema inversorului CMOS. Ambele

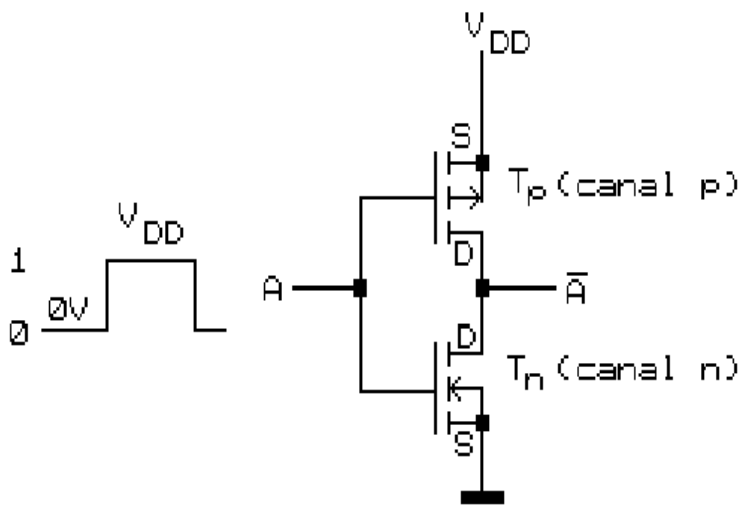


Fig.3.66 Inversor CMOS

tranzistoare au substratul conectat la sursa. Dreptele, legate impreuna, constituie iesirea circuitului A. Cind $A=0$, T_n este blocat iar T_p lucreaza in regiunea nesaturata (liniara). Cind tensiunea de intrare depaseste tensiunea de prag a tranzistorului n, V_n , acesta va lucra in saturare. Crescind in continuare tensiunea de intrare, ambele tranzistoare sint saturate: curentul absorbit din sursa de alimentare V_{DD} atinge acum valoarea maxima. Tranzistorul T_p se blocheaza cind se ajunge la $V_{DD}-V_p$, unde V_p este tensiunea de prag a acestuia.

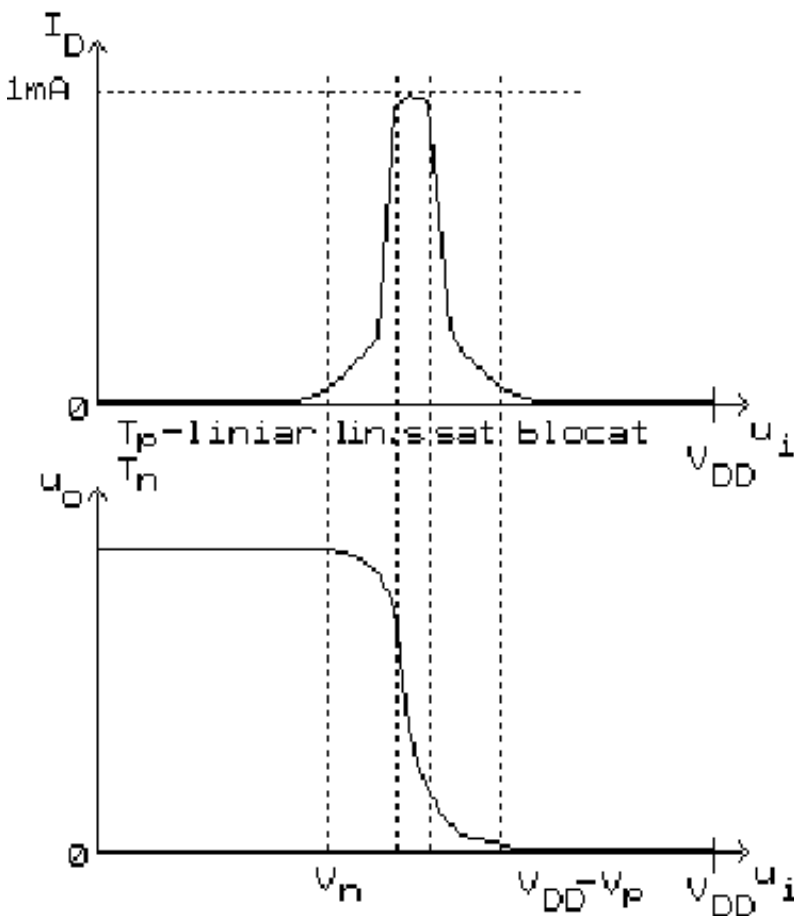


Fig.3.67 Variatia curentului de drena in functie de tensiunea de intrare si caracteristica de transfer a inversorului CMOS

absorbita de la sursa V_{DD} si I_{med} depind de frecventa si durata impulsurilor (fig.3.68).

Poarta NAND CMOS se obtine din conectarea mai multor inversoare: T_n cascade si T_p in paralel, ca in figura 3.67., unde tranzistoarele au fost desenate cu simboluri mai simple. Daca $A=B=0$, T_{n1} si T_{n2} sint blocate, iar T_{p1} si T_{p2} functioneaza in regiunea liniara; rezulta $\overline{AB}=1$.

tranzistoare au substratul conectat la sursa.

Dreptele, legate impreuna, constituie iesirea circuitului A. Cind $A=0$, T_n este blocat iar T_p lucreaza in regiunea nesaturata (liniara). Cind tensiunea de intrare depaseste tensiunea de prag a tranzistorului n, V_n , acesta va lucra in saturare. Crescind in continuare tensiunea de

In figura 3.67 este ilustrata variatia curentului de drena in functie de tensiunea de intrare, modalitatea de functionare a tranzistoarelor, precum si caracteristica de transfer a inversorului. Se observa ca apare curent de drena doar pe durata comutarii ceea ce explica reducerea insemnata a consumului in tehnologia CMOS.

Caracteristica de transfer e foarte apropiata de cea ideala, cu o tranzitie brusca in jurul valorii de 2,5V. Din acest motiv, marginea de zgomot este de circa 2 ori mai mare decit la circuitele TTL.

In tehnologia CMOS s-au realizat, pin la pin compatibile, toate CI TTL din familia 74XXX. Cele CMOS au indicatorul 74CXXX.

Deoarece CI MOS consuma curent numai pe frontul semnalului de comanda, puterea medie

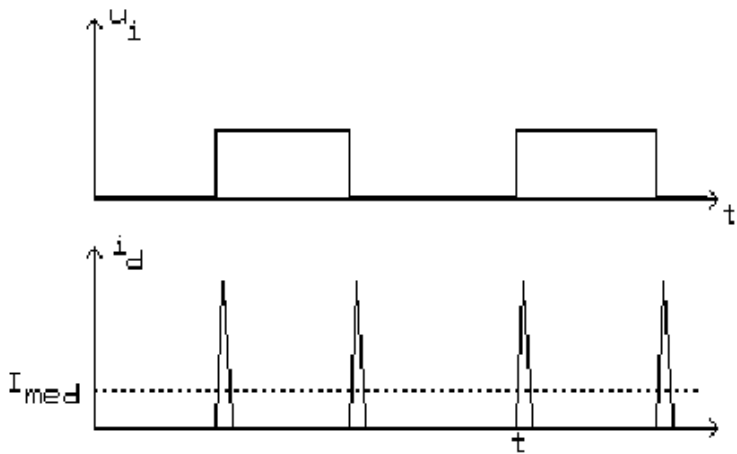


Fig.3.68. CI CMOS absorb curent din sursa numai pe fronturile impulsurilor. Curentul mediu I_{med} depinde de factorul de umplere.

La fel, daca $A=1, B=0$ sau $A=0, B=1$, unul din tranzistoarele T_{n1} sau T_{n2} este blocat, perechea lui de tip p fiind in liniara, si se tot

Numai pentru $A=B=1$, T_{n1} si T_{n2} vor fi deschise amindoua si \overline{AB} va fi 0.

Poarta SAU NU (NOR) se realizeaza prin conectarea in paralel a tranzistoarelor n si in serie a celor de tip p (fig.3.70).

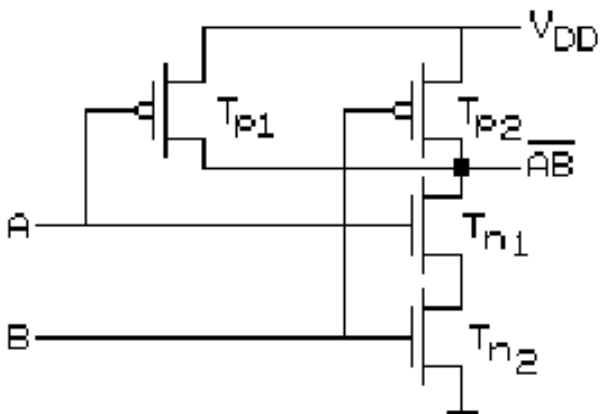


Fig.3.69 Poarta SI NU (NAND) cu tranzistoare CMOS

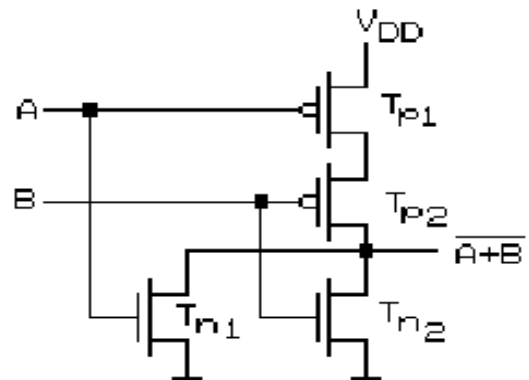
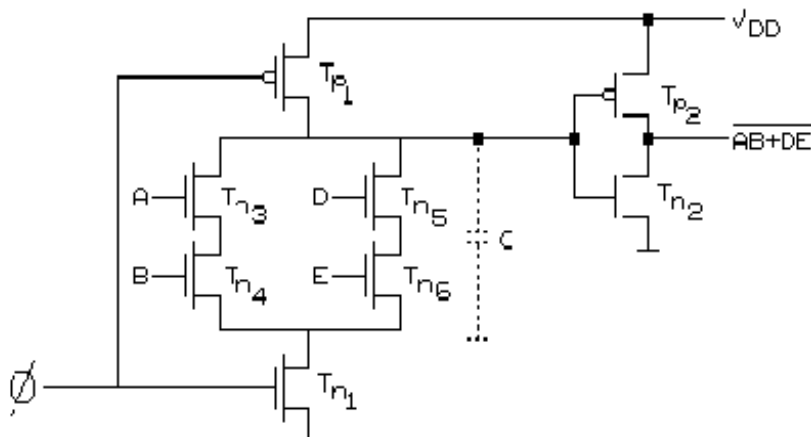


Fig.3.70 Poarta SAU NU CMOS

In portile prezentate, cu ajutorul carora se pot obtine circuite integrate complexe, intervine un numar egal de tranzistoare cu canal p si n. Pentru a permite curenti de comanda egali in cele 2 tranzistoare, aria ocupata de tranzistoarele pMOS trebuie sa fie de 2 ori mai mare decit cea a tranzistoarelor nMOS in scopul de a compensa mobilitatea redusa a golurilor in pMOS.

Se creaza astfel dezavantajul unei capacitati de intrare in T_p de 2 ori mai mare, drept consecinta a dublarii ariei.

Pentru reducerea ariei ocupate de CI MOS si a capacitatilor de intrare, a aparut in 1981, la firma Bell, ideea scaderii numarului de tranzistoare cu canal p. Circuitele realizate intr-o astfel de tehnologie, numita CMOS dinamic functioneaza conditionat de un semnal de tact (Φ in figura 3.71).



In schema portii SI SAU NU (AND OR INVERT) intervin doar 2 tranzistoare cu canal p fata de 7 cu canal n. T_{p1} si T_{p1} controleaza curentul unitatii logice formate din $T_{n3} \div T_{n6}$. Cind $\Phi=0$, capacitatea se incarca prin T_{p1} . Cind $\Phi=1$, T_{p1} este blocat, T_{p1} este deschis, iar tranzistoarele din unitatea logica decid ce se intimpla cu capacitatea C.

In aceasta tehnologie se

obtin circuitele de cea mai mare complexitate deoarece este asigurata reducerea la minimum a ariei ocupate de tranzistoarele p.

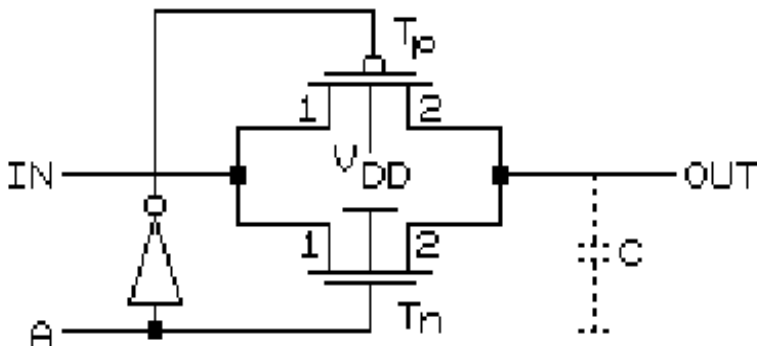


Fig.3.72 Poarta de transmisie: comutator bidirectional comandat de variabila A.

Poarta de transmisie este o alta structura de interes: se comporta ca un comutator bidirectional comandat (fig.3.72.). A este variabila de comanda, care se aplica inversat pe grila lui T_p .

Daca $A=0$, poarta este blocata.

Daca $A=1$, poarta conduce, curentul avind sensul de la IN la OUT sau invers, in functie de

tensiunea de pe condensatorul C si tensiunea de intrare. Astfel, pentru $IN=1(5V)$ si $OUT=0$, terminalul 1 al T_n actioneaza ca drena, iar terminalul 2 al T_n ca sursa. Curentul circula de la IN la OUT si C se incarca pina la $U_c = 4V$, cind - atingindu-se tensiunea de prag a lui T_n ($= 1V$) - acesta se blocheaza. Incarcarea in continuare a C pina la 5V se realizeaza prin T_p , care este deschis.

Invers, cind $IN=0$ si $OUT=1$, terminalul 2 al T_n actioneaza ca drena si terminalul 2 al T_p ca sursa. Condensatorul se descarca si curentul schimba sensul; va circula de la OUT la IN.

In situatia in care $A=0$, tensiunea pe grila lui T_n nu va fi pozitiva fata de a celorlalte terminale, iar T_n este blocat. De asemenea T_p este blocat, deoarece tensiunea pe grila sa nu va fi negativa fata de a celorlalte terminale.

Poarta de transmisiune este utilizata in toate schemele in care este necesara transmisiia bidirectionala a curentilor.

3.4. Cuplaje paralel ale SL_0 . Decodificatorul. Demultiplexorul

DCD. Structura interna. Proprietati

Decodificatorul este un circuit cu n intrari si 2 iesiri, care semnalizeaza aparitia configuratiilor de intrare prin 0 logic pe iesirea corespunzatoare.

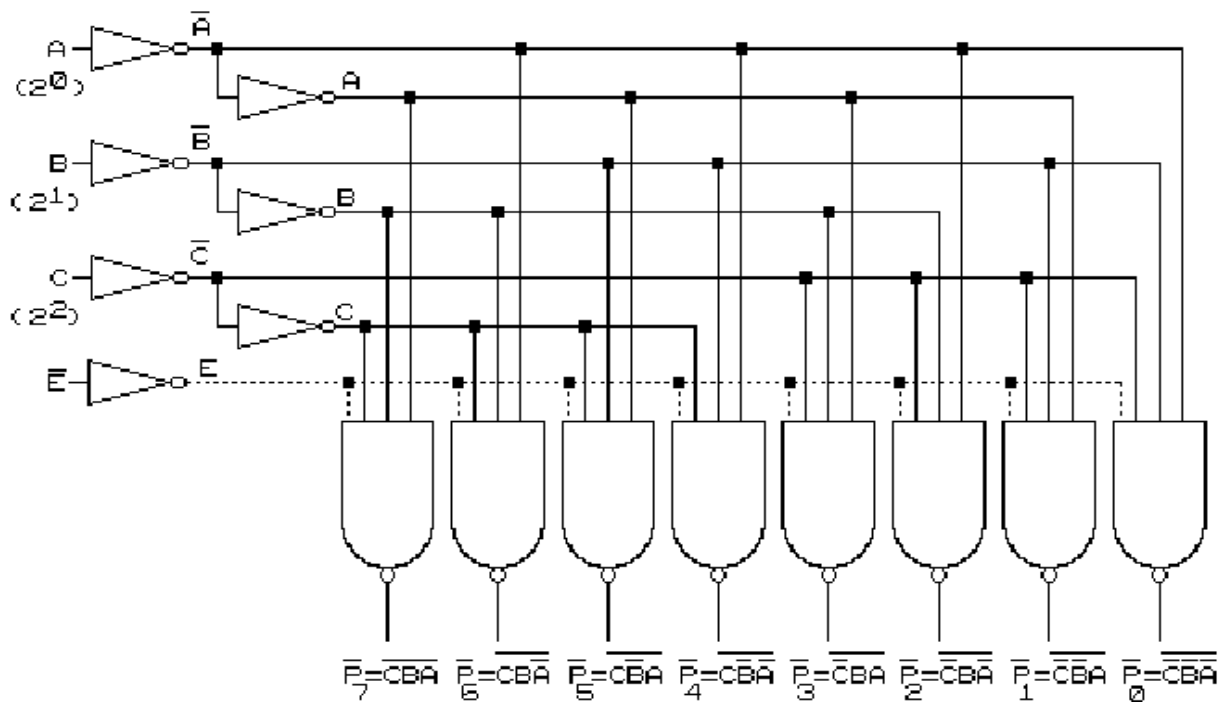


Fig.3.73 Structura DMUX (cu E) si DCD.

In figura 3.73. este prezentata schema unui decodificator - DCD - cu 3 intrari si 8 iesiri. Se observa ca iesirile reprezinta termenii canonici negati, \overline{P}_k , iar conexiunile reflecta regula de scriere a numerelor in binar: pentru rangul 2^0 alterneaza 2^0 de 0 cu 2^0 de 1, pentru rangul 2^1 alterneaza 2^1 de 0 cu 2^1 de 1, etc.

O alta proprietate a circuitului este aceea ca DCD realizeaza functia de exponentiere in baza 2: pentru numarul N exprimat binar, prezent la intrare, se calculeaza la iesire valoarea functiei 2^N , de asemenea in binar ($\overline{P}_7\overline{P}_6\dots\overline{P}_0=2^N$) - figura 3.74.

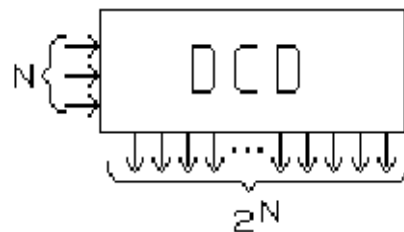


Fig.3.74 DCD calculeaza functia 2^N

Demultiplexorul este un DCD cu functionare conditionata de semnalul E (ENABLE - a permite). Interventia acestuia in schema este descrisa prin liniile punctate in figura 3.73. Pentru $\overline{E}=0$, decodificarea este permisa deoarece semnalul inversat $E=1$ predeschide portile NAND. Cind $\overline{E}=1$, decodificatorul este inhibat: toate iesirile vor fi pe 1 logic ($P_7=\dots=P_0=1$), indiferent de codurile de la

intrarile de selectie.

Functionarea DMUX este prezentata complet in urmatorul tabel de adevar:

\bar{E}	C	B	A	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0
1	X	X	X	1	1	1	1	1	1	1	1

Notatiile cu bara ($\bar{E}, \bar{7}, \bar{6}...$) atrag atentia asupra faptului ca semnalele sint active pe 0 logic. X este o variabila numita "don't care" (indiferenta) ce poate lua arbitrar orice valoare: fie 0, fie 1. Utilizarea ei a facut posibila o exprimare compactata a tabelului de adevar. Metode similare se folosesc pentru comprimarea unor tabele de adevar cu variabile de intrare numeroase.

Pe linga faptul ca este un DCD cu functionare conditionata de \bar{E} , DMUX este caracterizat de proprietatea de a distribui datele de pe o cale (\bar{E}), pe mai multe cai ($\bar{0}, \bar{1}.. \bar{7}$) alese prin codul de selectie (C,B,A). Intr-adevar, daca in schema din figura 3.73. fixam, de exemplu, CBA=011, poarta $\bar{P}_3 = \bar{3}$ este predeschisa, iar \bar{E} se va transmite la iesirea acesteia prin doua niveluri de inversare, deci cu valoare adevarata. DMUX se comporta ca si cum s-ar realiza un scurtcircuit intre \bar{E} si iesirea \bar{P}_3 .

Acest scurtcircuit poate fi mutat pe oricare alta iesire, in functie de codul CBA. Proprietatea de distribuire a datelor de pe o cale, pe mai multe, numita demultiplexare de date, a impus numele circuitului si este ilustrata in figura 3.75.

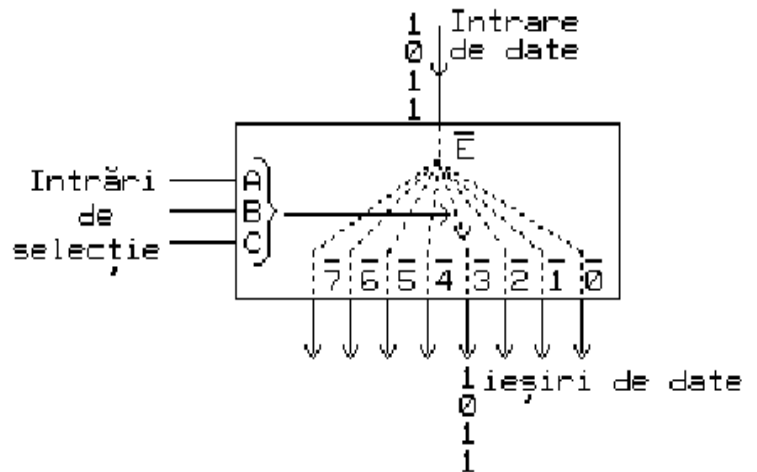


Fig.3.75 DMUX 1:8, cu distribuire a datelor de pe o cale \bar{E} pe 8 cai ($\bar{0}, \bar{1}, \dots, \bar{7}$) comandat de codul de selectie CBA.

Circuitele integrate de acest tip existente sint:

- DMUX 1:16 (74154)
- DMUX 1:4 (74155 - 74156)

Simbolul logic prin care reprezentam in scheme aceste circuite este infatisat in figura 3.76. Notatiile intrarilor si iesirilor se specifica in interior deoarece reflecta proprietatile intrinseci ale structurii, ordinea sa interna. Notatiile exterioare indica modul de utilizare a DMUX : in figura este transformat in DMUX 1:8 selectat prin codul CBA.

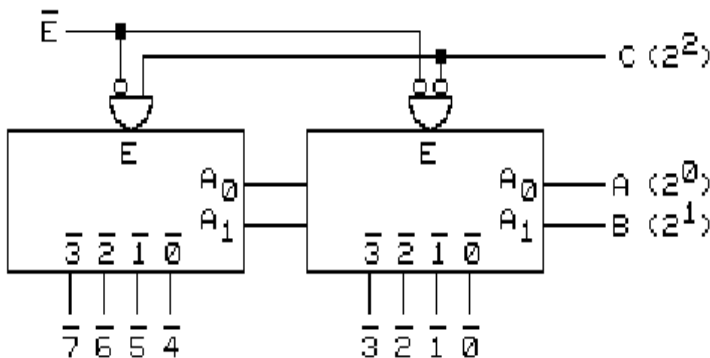


Fig.3.76 Simbolul logic al DMUX 74155 (2x DMUX 1:4) si utilizarea acestuia ca DMUX 1:8

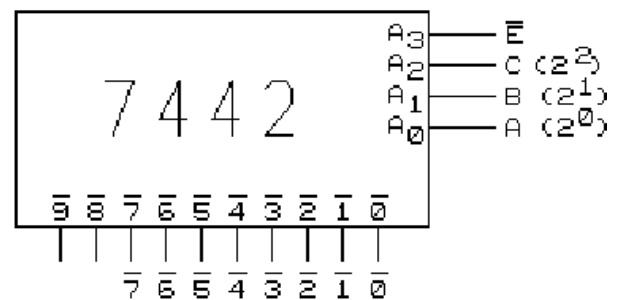


Fig.3.77 Decodificatorul BCD - zecimal 7442 utilizat ca DMUX cu 8 cai.

Exista si decodificatoare integrate. De exemplu 7442, DCD din BCD (Binary Coded Decimal) in zecimal. Acesta semnalizeaza la iesiri prin 0 logic, aparitia pe intrari a numerelor 0-9, codificate in binar pe 4 biti. In figura 3.77 este reprezentat simbolul logic al circuitului, conectat astfel incit sa realizeze functia unui DMUX 1:8.

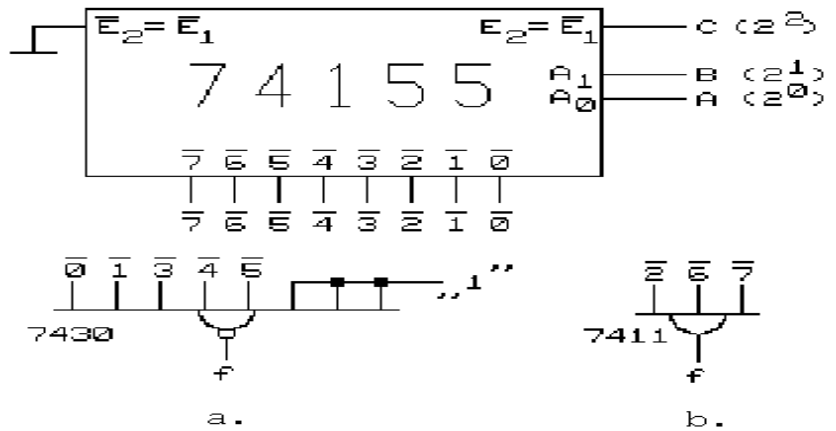
Conventia privind notatiile interioare si exterioare sint aceleasi ca in cazul anterior si vor fi mentinute pe tot parcursul lucrarii.

Utilizarea circuitelor DCD si DMUX este in strinsa legatura cu proprietatile lor. Astfel, datorita faptului ca emite la iesire termenii canonici negati, DCD este folosit in implementari de functii logice.

Consideram o funcție de 3 variabile:

$$f(C, B, A) = P_0 + P_1 + P_3 + P_4 + P_5 = \overline{P_0} \cdot \overline{P_1} \cdot \overline{P_3} \cdot \overline{P_4} \cdot \overline{P_5}$$

Pentru a obține funcția trebuie unite prin NAND ieșirile DCD $\overline{0}, \overline{1}, \overline{3}, \overline{4}, \overline{5}$ (fig.3.78.a).



Negata funcției f : **Fig.3.78** a) Implementarea f cu DMUX + NAND
b) Implementarea f cu DMUX + AND

$$\overline{f}(C, B, A) = P_2 + P_6 + P_7 = \overline{P_2} \cdot \overline{P_6} \cdot \overline{P_7} \quad \underline{\underline{f}}(C, B, A) = \overline{P_2} \cdot \overline{P_6} \cdot \overline{P_7}$$

și aceasta se realizează cu ajutorul unui AND conectat la ieșirile $\overline{2}, \overline{6}, \overline{7}$ ale DCD (fig.3.78.b).

Generalizând acest exemplu, rezulta că metoda de implementare cu DCD și porți constă în:

- . exprimarea funcției în forma canonică cu mintermeni $f = \sum_{k=1}^n a_k P_k$
- . conectarea variabilelor funcției, cu respectarea ordinii, la intrările unui DCD (sau DMUX utilizat ca DCD, $\overline{E}=0$). prin această conectare, la ieșirile decodificatorului se obțin termenii canonici negați (P_k) ai funcției;
- . conectarea la un SI NU a acelor ieșiri ale DCD corespunzătoare termenilor P_k marcați prin $a_k=1$, sau
- . conectarea la un SI a ieșirilor DCD ai caror P_k au $a_k=0$ în expresia funcției.

Este indicată aplicarea acestei metode în cazul implementării sistemelor de funcții logice deoarece:

- . este necesar un singur circuit MSI, decodificatorul, care furnizează termenii canonici pentru toate funcțiile sistemului;
- . pentru fiecare funcție se adaugă o singură poartă.

Pentru simplificarea desenului se notează intrările portilor ca în figura 3.78.

Pentru a fi implementate cu această metodă, funcțiile exprimate necanonic se aduc la forma canonică prin SI cu 1 exprimat prin SAU între variabilele ce lipsesc din termenii respectivi și negatele acestora ($1=B+B$) și prin desfacerea parantezelor. Astfel funcția minimizată:

$$f = \overline{C} \cdot \overline{A} + B$$

poate fi adusa la forma canonica prin urmatoarele prelucrari:
 $f = \overline{C} \cdot \overline{A} \cdot (B + \overline{B}) + B \cdot (A + \overline{A}) \cdot (\overline{C} + C) = \overline{C} \cdot B \cdot \overline{A} + \overline{C} \cdot \overline{B} \cdot \overline{A} + C \cdot B \cdot A + \overline{C} \cdot B \cdot \overline{A} = 1 P_2 + P_0 + P_7 + P_3$

Cealalta aplicatie importanta a DMUX, aceea de distribuire a datelor pe mai multe cai, este limitata de circuitele existente. Realizarea demultiplexarii pe mai mult de 16 canale necesita proiectarea unor scheme alcatuite din mai multe DMUX. Sa consideram problema obtinerii unui DMUX 1:64. Solutionarea acesteia sistematica presupune 3 etape:

Etapa I. Scheme bloc a DMUX ce trebuie realizat (figura 3.79) la care notatiile se fac in exterior deoarece se refera la un sistem cu mai multe CI, nu la un singur CI.

Etapa II. Stabilirea simbolurilor logice ale CI utilizate. Vom folosi pentru realizarea celor 64 de canale, 4 demultiplexoare de 16 cai fiecare (sau 8 DMUX de 1:8).

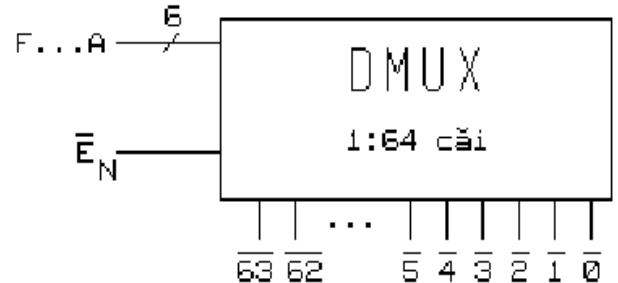


Fig.3.79 Schema bloc a DMUX 1:64

Etapa III presupune proiectarea schemei propriuzise si consta din:

- .reprezentarea CI prin simbolurile lor logice care sint identice si au aceleasi notatii interioare;
- .interconectarea CI intre ele in vederea obtinerii functionarii dorite;
- .realizarea notatiilor exterioare ale schemei bloc cu specificarea tuturor intrarilor si iesirilor.

Schema DMUX 1:64 avind canalele obtinute cu DMUX 1:16(74154) este prezentata in figura 3.80. Cele 4 DMUX 1:16 sint selectate in paralel de bitii cei mai putin semnificativi (CMPS=DCBA), dar DMUX 1:4 conectat pe \bar{E} permite functionarea doar a unui 74154 pentru un cod de intrare stabilit.

Daca selectia 74155 este realizata cu bitii cei mai semnificativi (CMS=FE), care variaza cu frecventa cea mai mica, ordinea de desemnare a canalelor este cea din figura. Evident, daca vom modifica ordinea de conectare a intrarilor de selectie, aceasta va produce si schimbarea numerelor canalelor de date.

Aceiasi schema se poate reproiecta si cu demultiplexoare 1:8. Sint necesare 9 x DMUX 1:8.

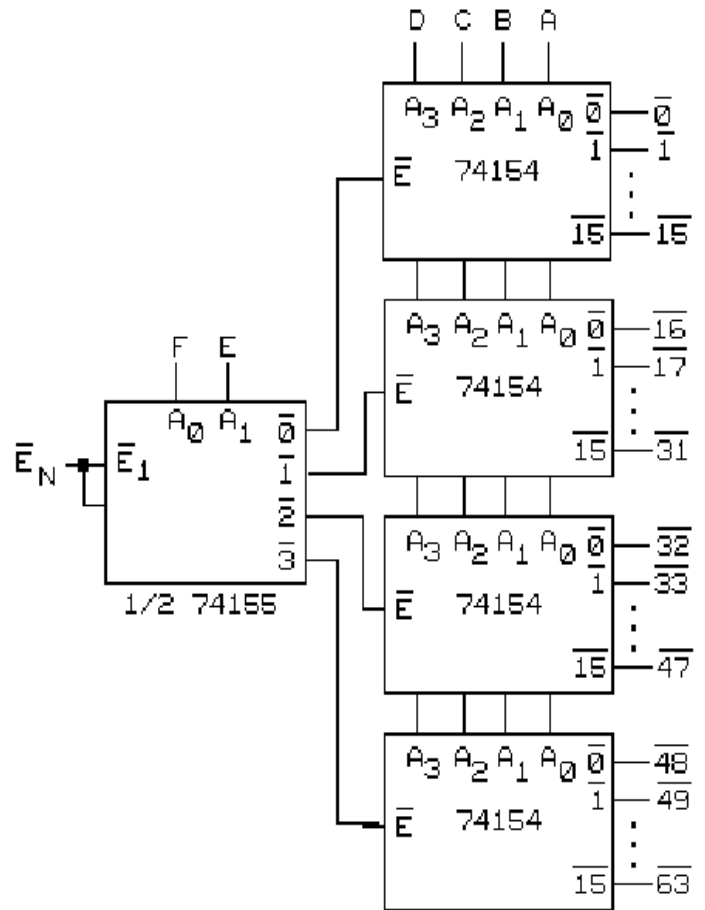


Fig.3.80 Schema de extindere a demultiplexarii de la 16 la 64 de canale.

3.5. Extensii serie ale sistemelor logice de ordinul 0. Multiplexorul

Multiplexorul este un circuit a carui structura implementeaza functii cu 2, 3 si 4 intrari, in forma de exprimare canonica generalizata. Astfel, pentru functii cu 3 variabile, regasim in interiorul MUX, AND-uri care realizeaza termenii canonici, validati de coeficientii a_k , ale caror iesiri sint adunate in SAU, corespunzator

$$f(C,B,A) = \sum_{k=0}^7 a_k P_k = \sum_{k=0}^7 D_k P_k$$

In figura 3.43 este prezentata structura unui MUX integrat cu 3 cai de selectie A_2, A_1, A_0 .

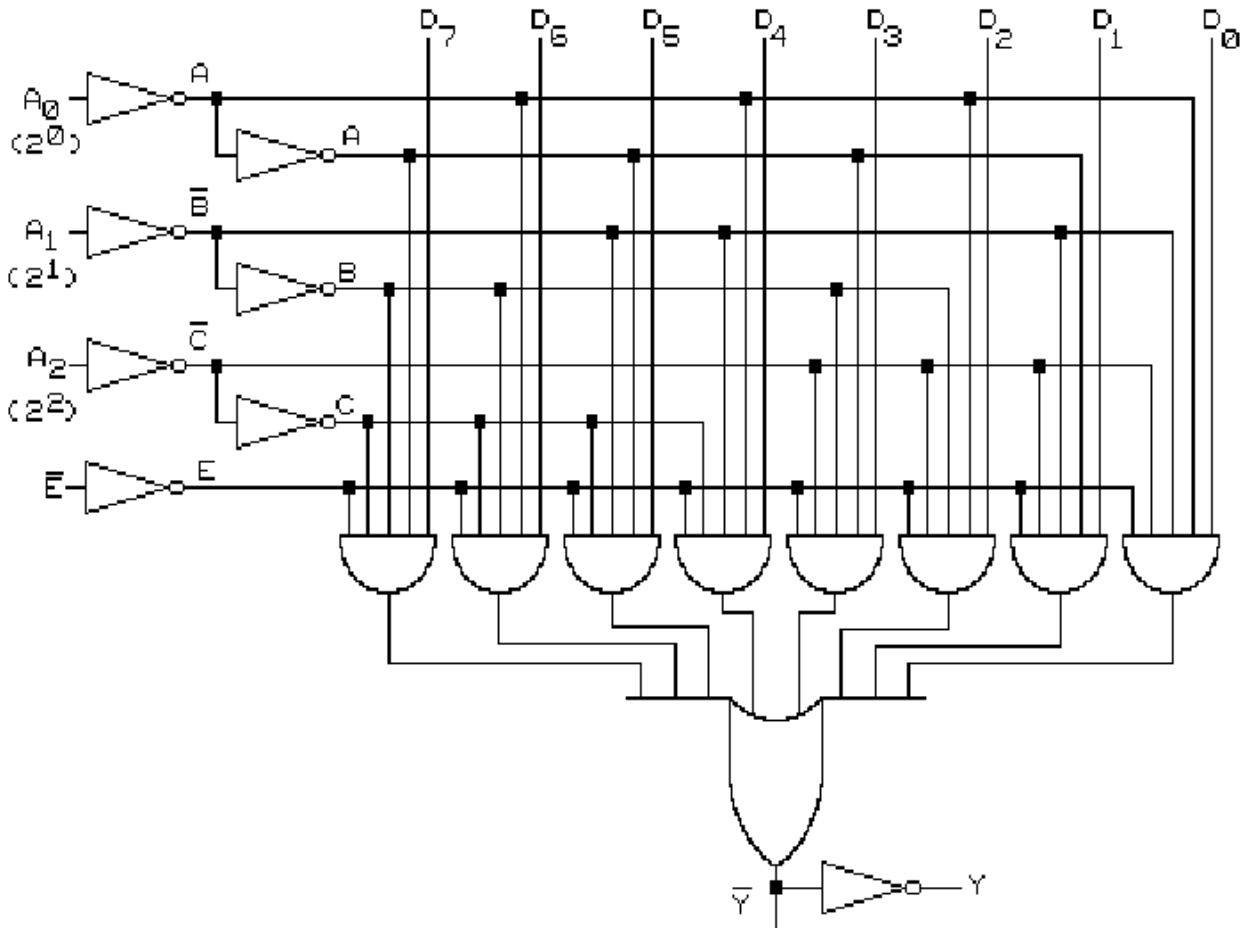


Fig.3.43 Structura interna a MUX cu 3 cai de selectie.

Prin particularizarea valorilor vectorului functiei (D_0, D_1, \dots, D_7) se pot implementa cu acest MUX, functii logice de 3 sau mai multe variabile. Astfel, functia de vot majoritar cu 3 variabile A, B, C se realizeaza cu schema din figura 3.44. Pornind de la forma canonica a acestei functii, prezentata in subcapitolul 3.2.2.1, $f_{VM} = P_3 + P_5 + P_6 + P_7$ se stabileste vectorul $(D_7=1, D_6=1, D_5=1, D_4=0, D_3=1, D_2=0, D_1=0, D_0=0)$ de conectare a intrarilor D_k ale DMUX.

Cu un MUX cu 3 cai de selectie si cel mult inca un inversor se poate implementa orice functie de 4 variabile (prin introducerea unei variabile in tabelul de adevar sau in diagramele de reprezentare). Pentru exemplificare se considera functia descrisa de tabelul urmator (a).

D	C	B	A	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

a.

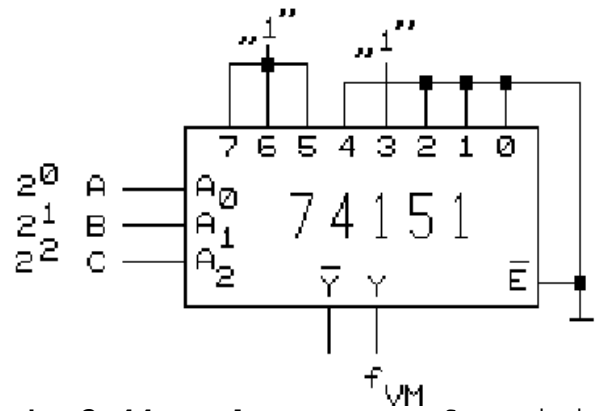


Fig.3.44 Implementarea funcției de vot majoritar de 3 variabile $f_{VM}(A,B,C)$ cu MUX 74151.

D	C	B	f_A
0	0	0	0
0	0	1	A
0	1	0	A
0	1	1	1
1	0	0	0
1	0	1	A'
1	1	0	A
1	1	1	1

b.

D	B	A	f_C
0	0	0	0
0	0	1	C
0	1	0	C
0	1	1	1
1	0	0	0
1	0	1	C
1	1	0	1
1	1	1	C

c.

C	B	A	f_D
0	0	0	0
0	0	1	0
0	1	0	D
0	1	1	D'
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

d.

D	A	f_{CB}
0	0	CB
0	1	C+B
1	0	B
1	1	C

e.

D	B	f_{CA}
0	0	CA
0	1	C+A
1	0	CA
1	1	C+A'

f.

D	C	f_{BA}
0	0	BA
0	1	B+A
1	0	BA'
1	1	B+A

g.

D	B	f_{DA}
0	0	0
0	1	A+D
1	0	A
1	1	1

h.

C	A	f_{DB}
0	0	0
0	1	D'B
1	0	B
1	1	1

i.

B	A	f_{DC}
0	0	0
0	1	C
1	0	C+D
1	1	D'+C

j.

Implementarea functiei prelucrata ca in tabelul (b) necesita, pe langa MUX si un inversor al variabilei A. Schema este prezentata in figura 3.45. Variabilele D,C,B din stanga tabelului de adevar sint conectate la intrarile de selectie ale MUX, in ordinea corespunzatoare: $D=2^2$, $C=2^1$, $B=2^0$. Astfel, termenii canonici ce se obtin in interiorul MUX nu mai sint cei ai functiei f_A (cu vectorul dependent de A).

Prin introducerea in tabel a variabilei D, f_D (tabelul d) are o implementare similara. Numai pentru f_C se poate elimina inversorul, obtinind o structura minima, cu un singur CI (fig.3.46).

Trebuie remarcat ca nu orice functie permite o astfel de implementare minima, fara inversor. In cazul general, vectorul functiei este dependent de 0, 1, de variabila si de variabila negata. Este necesara inspectarea tuturor posibilitatilor pentru a se minimiza implementarea cu MUX.

In situatiile in care se introduc doua dintre variabilele functiei in tabel (e - j) se urmareste de asemenea varianta minima, cu cel mai simplu vector. Este cazul functiei f_{AD} care se realizeaza cu MUX cu 2 cai de selectie si o poarta (fig.3.47). Solutia este neconvenabila fata de cea din figura 3.46 deoarece introduce poarta SAU.

Exista functii de 4 variabile care permit implementare minima cu MUX de 2 cai: cele care au vectorul dependent de cite una din variabile si nu de combinatii ale lor.

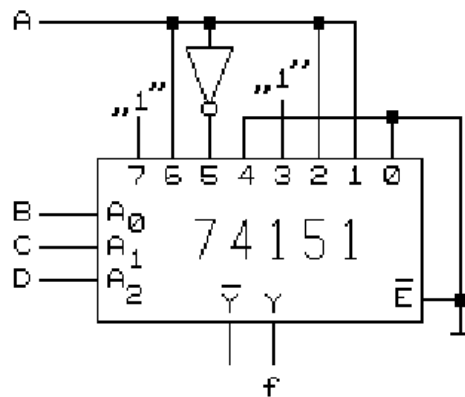


Fig.3.45. Implementarea f cu vectorul dependent de A.

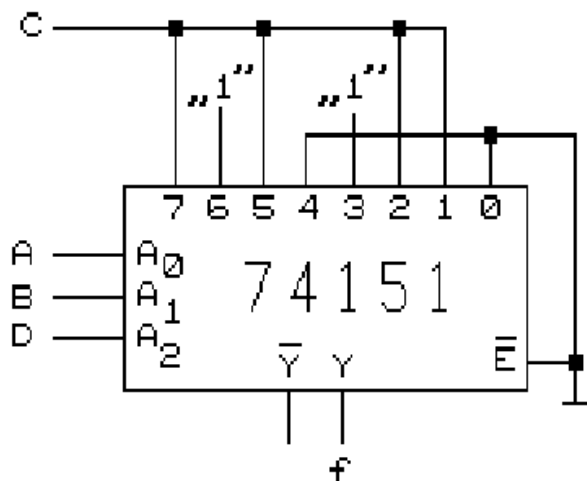


Fig.3.46 Implementarea f cu vectorul dependent de C. Este o realizare minima.

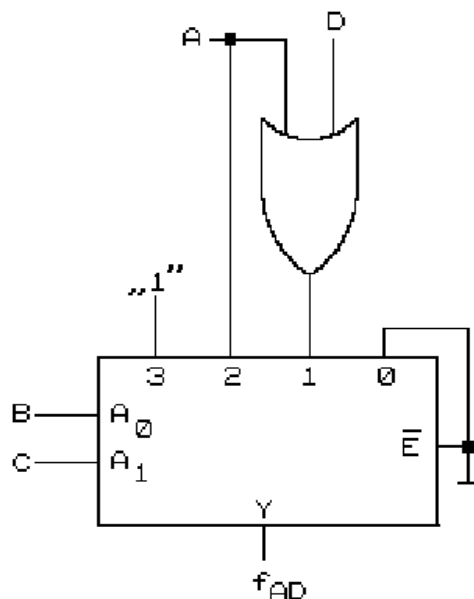


Fig.3.47 Implementarea minima a $f(D,C,B,A)$ cu 2 variabile (D,A) introduse in vectorul functiei (f_{AD}).

Astfel:

$$f = P_6 + P_7 + P_8 + P_9 + P_{11} + P_{12} + P_{13} + P_{15} = \bar{D} \cdot C \cdot B \cdot \bar{A} + \bar{D} \cdot C \cdot B \cdot A + D \cdot \bar{C} \cdot \bar{B} \cdot \bar{A} + D \cdot \bar{C} \cdot \bar{B} \cdot A + D \cdot \bar{C} \cdot B \cdot A + D \cdot C \cdot \bar{B} \cdot \bar{A} + D \cdot C \cdot \bar{B} \cdot A + D \cdot C \cdot B \cdot A$$

va fi prelucrata cu ajutorul algebrei Boole in vederea implementarii minime cu MUX cu 2 cai de selectie. Trebuie balcate toate cele 6 posibilitati de selectie cu grupe de cite 2 variabile pentru a gasi vectorul functiei in forma cea mai redusa. Acesta va fi $D_0=0, D_1=C, D_2=1, D_3=A$ si se obtine cu selectie prin variabilele D si B ($D=A_1, B=A_0$) in modul urmator:

$$f = \bar{D} \cdot B \cdot (C \cdot \bar{A} + C \cdot A) + D \cdot \bar{B} \cdot (\bar{C} \cdot \bar{A} + \bar{C} \cdot A + C \cdot \bar{A} + C \cdot A) + D \cdot B \cdot (\bar{C} \cdot A + C \cdot A) = 0 \cdot (\bar{D} \cdot \bar{B}) + C \cdot (\bar{D} \cdot B) + 1 \cdot (D \cdot \bar{B}) + A \cdot (D \cdot B)$$

O alta proprietate a MUX, des utilizata in aplicatii, este aceea de a concentra datele de pe mai multe canale (D_0, D_1, \dots, D_7) pe unul singur (Y). Comutarea canalelor se realizeaza cu ajutorul codurilor la intrarile de selectie. In figura 3.48 se arata ca, datorita acestei proprietati, MUX poate fi asemuit cu un comutator care realizeaza scurtcircuitarea intre iesire si oricare din caile de date, comandat de intrarile de selectie. Considerind schema din figura 3.43 si fixind unul din codurile de selectie ($A_2 A_1 A_0 = 110$) se observa ca

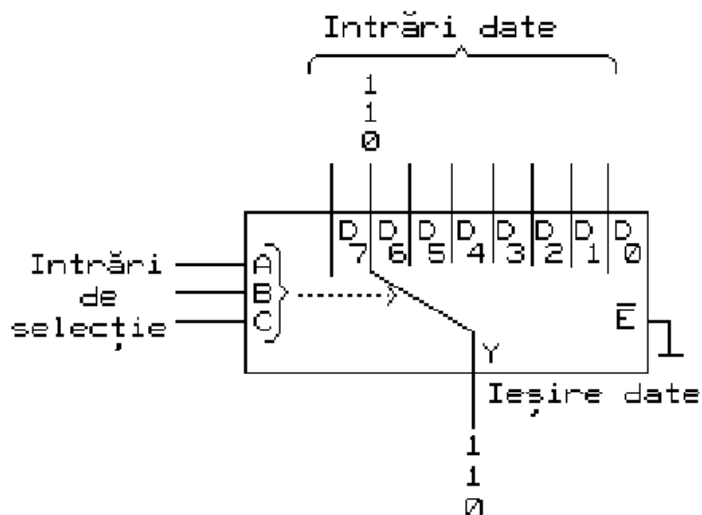


Fig. 3.48. MUX 8:1 cu concentrarea datelor de pe 8 cai pe una singura.

poarta continind P_6 este predeschisa si datele de la D_6 sint transmise la iesirea Y. Aceasta proprietate de concentrare a datelor se numeste multiplexare si a determinat numele circuitului.

MUX integrate existente sint:

MUX 2:1 cu iesire inversoare Y	-	74158
MUX 2:1 cu iesire neinversoare Y	-	74157
MUX 4:1 (iesire neinversoare)	-	74153
MUX 8:1 (cu Y si $Y\bar{Y}$)	-	74151
MUX 16:1 (doar cu Y)	-	74150

Numai prin extinderea MUX se pot obtine concentrari ale unui numar mai mare de 16 canale. Problema se rezolva intr-o maniera similara cu cea a extinderii DMUX. In figura 3.50 este prezentata una din solutiile de realizare a unui MUX cu 64 canale, a carei schema de ansamblu este cea din figura 3.49.

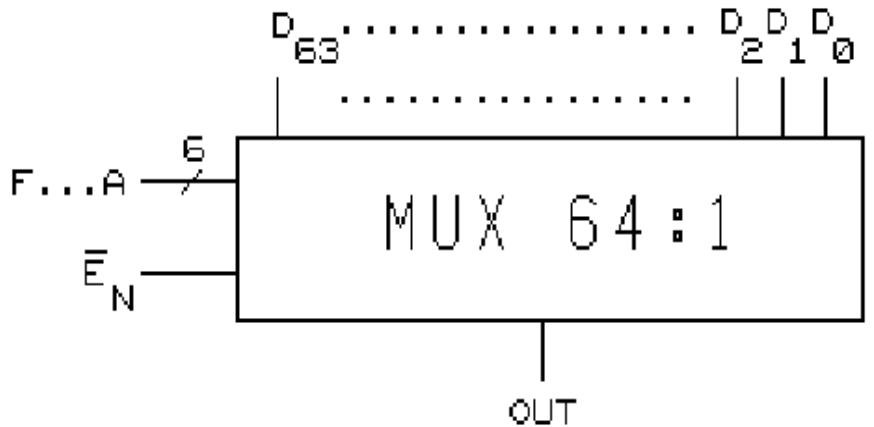


Fig.3.49. Schema bloc a unui MUX 64:1

In schema de extindere a MUX, 8 circuite care asigura obtinerea numarului de canale functioneaza in paralel, selectate de cei mai putin semnificativi biti. Al noualea MUX transmite la OUT, conform codului FED iesirea unuia din MUX-urile de pe primul nivel.

Intrarile \bar{E}_{19} ale $MUX_1 \div MUX_8$ pot fi conectate la masa, conditionarea functionarii schemei se realizeaza doar la MUX-ul final.

O alta varianta de implementare a schemei este cea presupunind utilizarea a $4 \times MUX_{16:1}$, cu iesiri multiplexate 4:1 selectate cu bitii cei mai semnificativi FE.

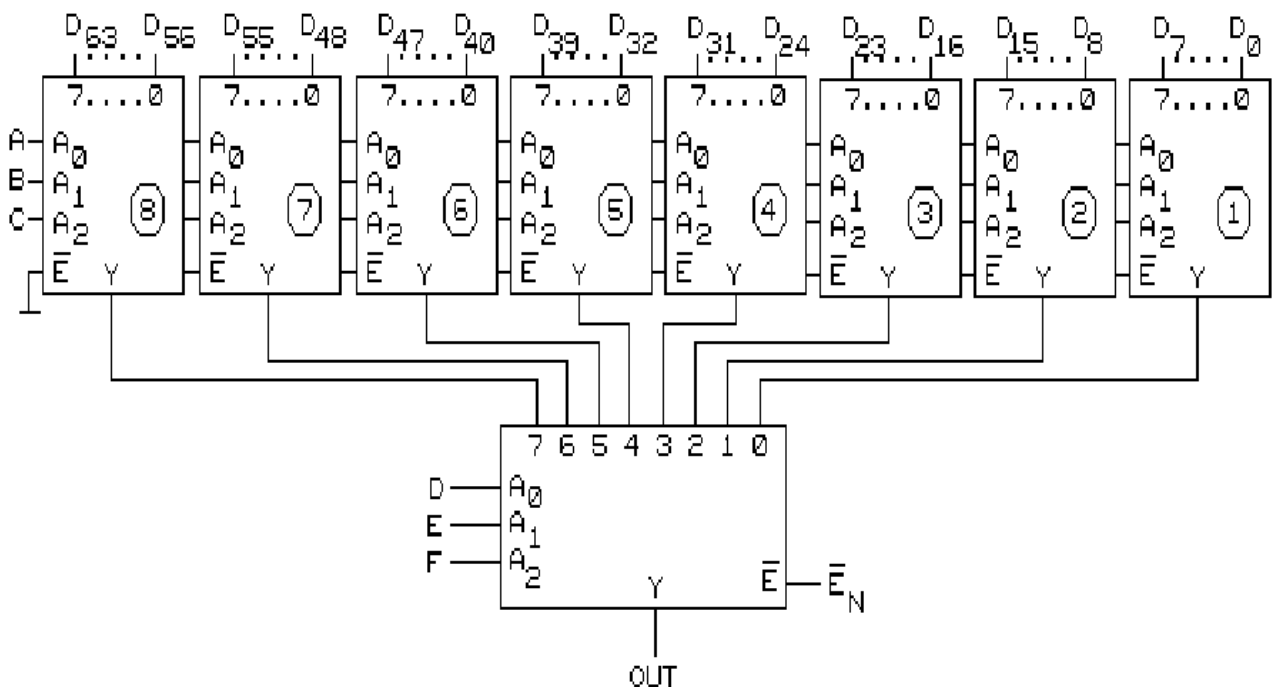


Fig.3.50 Extinderea MUX 8:1 la MUX 64:1

3.6. Cuplaje serie paralel ale sistemelor logice de ordinul 0.

3.6.1. ROM (Read Only Memory - Memorii numai cu citire)

Aceste circuite au o structura permitind generalizarea implementarii cu DCD si porti pentru sisteme complexe de functii booleene, dependente de un numar mare (n) de variabile. DCD din interiorul ROM-ului are un numar corespunzator de iesiri 2^{n-1} . Acestea pot fi conectate (pentru $a_k = 1$) sau neconectate (pentru $a_k = 0$) la intrarile portilor SI NU. Se obtine astfel oricare din cele 2^{2^n} functii cu n variabile. Portile SI NU au fiecare 2^n intrari, potential conectabile la liniile de iesire ale DCD. In figura 3.89 este sugerata structura logica interna a unui ROM ce poate implementa functii de 5 variabile. Specificarea a 4 din cele 2^{2^5} functii se realizeaza prin efectuarea unora din legaturile posibile. Consideram, de exemplu, SI NU-ul cu iesirea notata cu O_3 . Conectarea intrarilor lui la liniile $\bar{2}, \bar{3}, \dots, \bar{31}$ ale DCD este echivalenta cu introducerea in expresia functiei O_3 a termenilor canonici P_2, P_3, \dots, P_{31} . Prin programarea matricii de

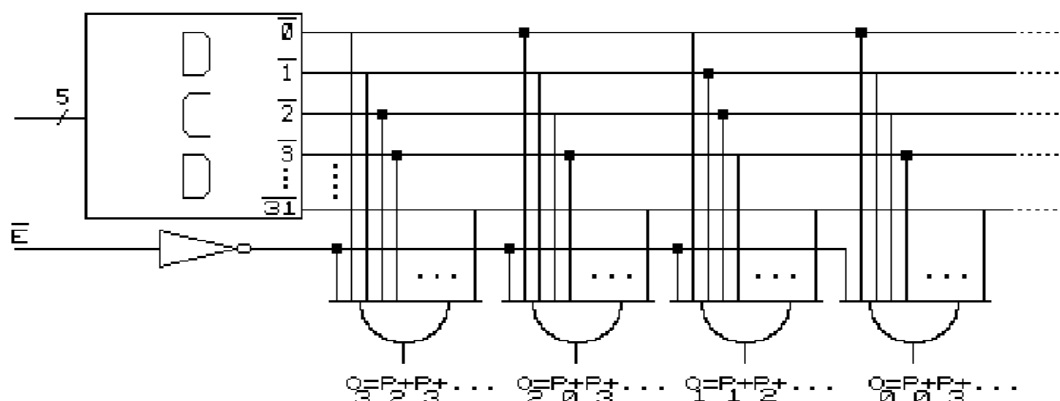


Fig.3.89 ROM ce implementeaza 4 functii de 5 variabile.

legaturi se pot obtine oricare alte functii de 5 variabile. Este evident ca pentru implementarea cu ROM functiile trebuie aduse la forma canonica cu mintermeni.

In figura 3.90, portile SI NU sunt reprezentate simplificat prin linii verticale pe care sunt marcate conexiunile. Cu ajutorul acestei scheme se poate explica foarte sugestiv proprietatea de memorare a ROM. La selectarea prin ADRESE a uneia din liniile de iesire a DCD (ex.1'), apare la $O_3 \div O_0$ un cuvint binar de 4 biti (0010) determinat de conexiunile liniei respective cu SI NU-urile.

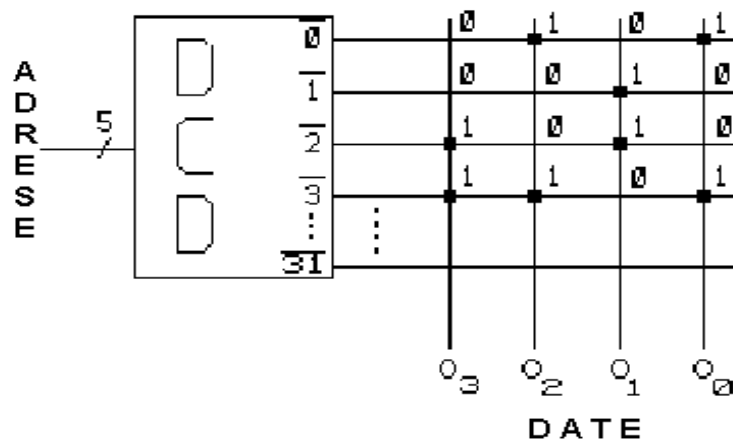


Fig.3.90 Proprietatea de memorare a ROM.

Putem asimila fiecare legatura potentiala cu o celula de memorare binara: pentru 0 logic, cind nu exista conexiune, pentru 1 logic, cind este efectuata conectarea.

Practic, nu gresim considerind matricea de celule (de legaturi) drept un tabel binar, memorat in ROM, ce poate fi citit linie cu linie. Scrierea acestui tabel se realizeaza prin procedee tehnologice sau electrice independente de sistemul in care este utilizat ROM-ul. De aceea memoriile sint "numai cu citire", proprietate sugerata de insasi denumirea de ROM (Read Only Memory).

Definitii:Intrarile DMUX, adica bitii prin care se selecteaza rindurile memorate in ROM se numesc **biti (cai) de adresa** .

Iesirile DMUX , egale cu numarul rindurilor tabelului memorat in ROM se numesc **adrese** de memorie.

Ansamblul bitilor memorati la fiecare adresa se numeste **cuvint** de memorie. Cuvintele reprezinta de fapt continutul adreselor de memorie

Se observa ca numarul adreselor este egal cu numarul cuvintelor memorate.

Citirea memoriei consta in aparitia cuvintelor la iesirile ROM, pe **caile de date**, ca urmare a selectarii adreselor cu ajutorul bitilor de adresa

Datorita proprietatii DMUX intre numarul de cuvinte(adrese)ale ROM si numarul de biti de adresa (cai de date) exista relatia:

$$\text{Numarul de cuvinte(adrese)} = 2^{\text{numarul de biti de adresa}}$$

In cataloagele de CI, memoriile ROM sint specificate prin **organizarea** bitilor memorati, adica prin:

$$\text{numarul de cuvinte} \times \text{numarul de biti pe cuvint}$$

Simbolul logic al ROM

Circuitele se reprezinta grafic prin simbolul logic: un dreptunghi pe care sint figurate toate intrarile (cai de adrese si Enable) si iesirile (cai de date), cu semnificatiile lor notate in interiorul dreptunghiului. Astfel in figura 3.91 este prezentat simbolul logic al unui ROM cu organizare 2^m cuvinte x n biti/cuvint.

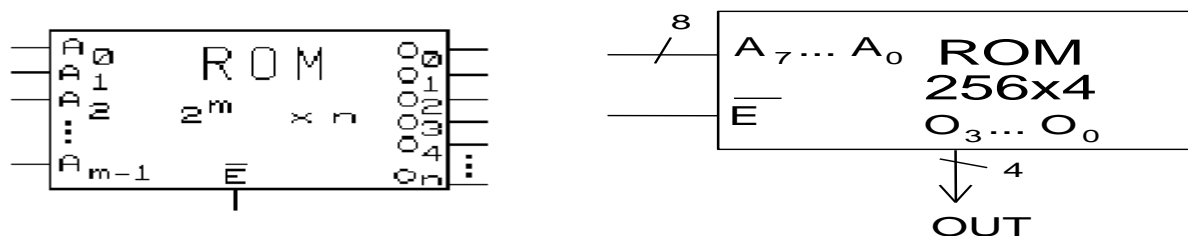


Fig.3.91. Simboluri logice ale memoriilor ROM

Din organizare deducem numarul de intrari si iesiri ale ROM.

- daca numarul de cuvinte este 2^m ,rezulta ca numarul de intrari este m. Se deseneaza intrarile si se noteaza, in interior cu A_0, A_1, \dots, A_{m-1} ;
- daca numarul de biti pe cuvint este n, acesta este

egal cu numarul de iesiri. Se deseneaza iesirile si se noteaza in interior cu O_0, O_1, \dots, O_{n-1}

- ROM-ul, ca orice circuit complex, are si intrare de Enable, E' , care permite sau inhiba functionarea circuitului

Cu aceeasi metoda, in figura 3.91 s-a stabilit si simbolul logic al unui ROM de 256 cuvinte x 4 biti /cuvint. Trebuie remarcata simplificarea notatiilor: toate liniile de adresa si de date se reprezinta prin cite o singura linie taiata, iar in dreptul taieturii se noteaza numarul de fire din manunchi

Unitatile de masura utilizate pentru a exprima organizarea memoriilor sint: bitul, byte-ul (1 byte = 8 biti), kilobitul (1 KB = 1024 biti = 2^{10} biti), megabitul (1Mb = 2^{20} biti), gigabitul (1Gb = 2^{30} biti).

Realizarea tehnologica a ROM

In functie de modalitatile de programare a celulelor de memorie, ROM-urile sint de mai multe tipuri:

Mask ROM sint programate de fabricant pe baza tabelelor de adevar trimise de catre utilizator. Structurile se realizeaza uniform, cu procese tehnologice identice, cu exceptia ultimei masti prin care se programeaza legaturile portilor SI NU de iesire cu liniile demultipleorului, individualizindu-se ROM-urile prin continut. Este indicata folosirea Mask ROM pentru productii de serii mari, singurele care permit amortizarea costurilor mastii de conexiuni.

PROM (Programmable ROM) sint circuite programabile de catre utilizator prin selectarea fiecărei celule de memorie si aplicarea pe iesirile corespunzatoare ale unor curenti ce disting un fuzibil in tehnologia bipolară sau incarca o capacitate MOS in cazul ROM-urilor unipolare.

In figura 3.92 sint reprezentate doua celule din memoria bipolară. Daca fuzibilul este "ars", pe linia de adrese selectata, k' , inversorul are 1 la intrare, deci 0 la iesire. Constatam ca am memorat 0 logic in celula cu fuzibilul ars. Daca fuzibilul este lasat intact, memoram 1 logic in celula corespunzatoare deoarece, 0-ul liniei O' a DMUX este transmis portii SI NU. Deci cele 2 valori logice ale iesirii, 0 si 1, sint memorate prin absenta si, respectiv, prezenta fuzibilului.

Fig. 3.92

Fuzibilii nu sint sirme, ci sint proiectati in pastila de siliciu. Uneori, se pot produce defectari in timp ale PROM, prin refacerea unor legaturi. Initial, "arderea" fuzibililor se face cu un aparat special, numit Programator de PROM, care injecteaza in iesirile portilor SI NU curenti de valoare specificata in catalog. Se selecteaza fiecare celula in care trebuie memorat 0, cu adresarea corespunzatoare a DMUX si a portii SI NU de injectie, programindu-se succesiv toate zerourile. Uzual, Programatorul de PROM este conectat la calculator, care, pe baza tabelului binar ce trebuie memorat in PROM, comanda automat programarea si verifica in final continutul PROM. Intrucit memoriile PROM au parametri de programare diferiti, Programatoarele de PROM sint dedicate fiecarui tip de memorie.

Circuitele ROM cu aceeasi organizare se deosebesc intre

ele doar prin continutul programat, fapt ce este marcat prin lipirea pe fiecare circuit a unei etichete cu o notatie de identificare (numere de ordine, s.a).

Din metoda tehnologica prezentata, rezulta ca PROM-urile bipolare au dezavantajul major ca nu pot fi sterse si rescrise. PROM-urile cu continut eronat, fie datorita greselilor de proiectare a tabelului de adevar, fie datorita functionarii deficitare a Programatorului de PROM, nu mai pot fi utilizate. Greselile sint foarte costisitoare in situatia utilizarii PROM-urilor bipolare. Pe de alta parte, acestea sint indispensabile, datorita vitezei mari de raspuns, in cazul aplicatiilor ce prelucreaza rapide.

Tot programare unica au si PROM-urile unipolare care memoreaza prin incarcarea sau neincarcarea capacitatilor tranzistoarelor MOS, plasate in celulele de memorie.

REEPROM (Reprogrammable PROM) sint memorii unipolare ce pot fi sterse si programate din nou. Stergerea se efectueaza prin descarcarea capacitatilor MOS care este provocata de iradierea cu ultraviolete a structurii circuitului printr-o fereastră de cuarț plasata deasupra pastilei de siliciu (fig.3.93). Fereastră de cuarț este transparenta pentru radiatiile ultraviolete, deci le permite trecerea catre celulele de ROM. Dupa stergere, REEPROM-ul se poate rescrie din nou cu programatorul.

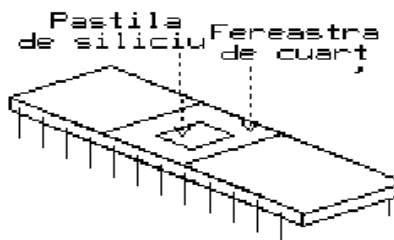


Fig.3.93. Continutul PROM-urilor MOS se sterge cu radiatii ultraviolete pentru care fereastră de cuarț ce acopera pastila de siliciu este transparenta.

Pentru a preveni stergerile accidentale, nedorite, dupa programare trebuie acoperita complet fereastră de cuarț cu eticheta de identificare a REEPROM-ului.

EEPROM (Electrically Erasable PROM) sint PROM-uri cu stergere electrica. Spre deosebire de REEPROM-urile cu continut alterabil cu lumina ultravioleta ce actioneaza asupra intregii structuri, EEPROM-urile au avantajul ca fiecare celula poate fi stearsa si reprogramata fara a influenta celelalte celule.

Tipurile de memorii ROM in functie capacitate si tehnologia de realizare sint prezentate in tabelul urmator. Se observa ca la capacitati mici se utilizeaza numai tehnologie bipolară, in timp ce tehnologia MOS este suverana la capacitati de peste 8 Kbiti.

Capacitate		Tehnologie
32 x 8	512 x 4 (0,5K x 4)	Bipolară
256 x 4	512 x 8 (0,5K x 8)	
256 x 8	1024 x 4 (1K x 4)	
1K x 8	32K x 8	MOS
2K x 8	64K x 8	
4K x 8	512K x 8 (0,5 M x 8)	
8K x 8	1024K x 8 (1M x 8)	
16K x 8	1M x 16, 2M x 8	

Pentru a ilustra cit de mare este cantitatea de

informatie inmagazinata in ROM, vom considera cazul unei memorii de capacitate relativ mica : $2K \times 8 = 16K$ biti

Numarul combinatiilor binare din aceasta memorie este $2^{16K} = 10^{5000}$. Pentru a ne da seama cit de mare este acest numar este suficient sa il comparam cu numarul electronilor dintr-o sfera cu dimensiunea pamintului: care este doar 10^{63} .

Complexitatea unui sistem numeric este caracterizata prin doua marimi: SIZE (dimensiune) si DEPTH (adincime)

Dimensiunea (SIZE) este definita in doua moduri prin:

- numarul de porti de acelasi tip din structura
- nr.total de intrari al tuturor portilor sistemului.

Adincimea (DEPTH) definita prin numarul de niveluri de porti si da informatii despre viteza de raspuns a sistemului. Cu cit sint mai putine niveluri de porti, cu atit mai rapid este raspunsul sistemului.

Aceste doua marimi definite anterior sint utilizate pentru proiectarea pe siliciu a CI complexe.

Schema interna a ROM de mare capacitate

Rezultate deosebit de interesante se pot evidentia calculind dimensiunea (SIZE) definita ca numarul de intrari in porti pentru o schema de tipul celei din figura 3.89, avind organizarea 2^n cuvinte cu m biti pe cuvint:

$$\text{SIZE ROM } (2^n \times m) = 2n + 2^n n + m 2^n = 2^n (m \times n)$$

unde $2^n n$ sint intrarile in cele 2^n porti ale DCD, iar $m 2^n$ sint intrarile in cele m porti de iesire. Formula finala s-a obtinut prin neglijarea intrarilor in inversoarele DCD, $2n$ fiind nesemnificativ fata de variatia eponentiala a celorlalti termeni.

Observatie importanta Formula demonstreaza dependenta exponentiala a complexitatii unui sistem integrat de numarul sau de intrari.Reducerea unei singure intrari din sistem are drept consecinta injumatatirea complexitatii acestuia.

Aceasta crestere exponentiala a dimensiunii ROM-ului in functie de numarul de intrari a determinat o regindire structurala a ROM, care sa permita realizabilitatea tehnologica a memoriilor de capacitate mare.

La cresterea capacitatii ROM, schema interna prezentata in fig. 3.89 nu mai poate fi utilizata deoarece conduce la DMUX extrem de mari. Chiar in cazul unui ROM de 4K, decodorul ar avea 4096 de iesiri, ceea ce este extrem de mult. In asemenea situatii(vezi fig.3.94), bitii de adresa se impart in doua jumatati. Prima jumatate selecteaza DMUX de adrese, a doua jumatate selecteaza MUX de iesire. Matricea de celule de memorie este organizata astfel incit se citesc simultan 8 biti pentru fiecare rang binar ($2^0, 2^1, \dots, 2^7$). MUX-ul rangului respectiv selecteaza doar unul din cei 8 biti si il transmite la iesire, dependent de adresa primita. MUX-urile se realizeaza cu circuite cu trei stari (TS), de asemenea pentru reducerea dimensiunilor.

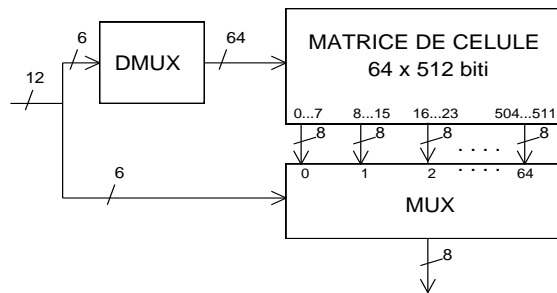


Fig. 3.94

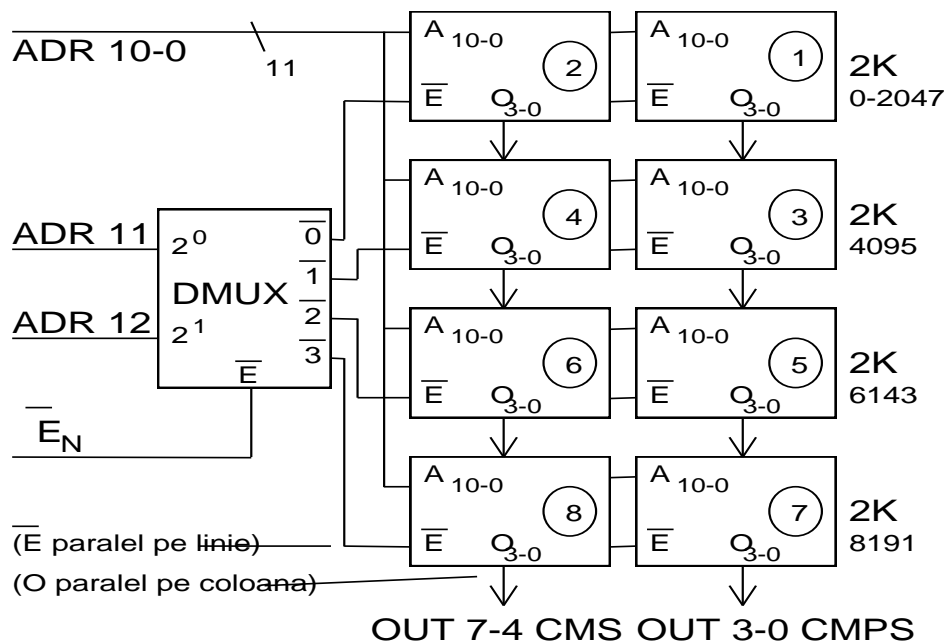
Prin aceasta ingenioasa idee structurala se inlocuieste uriasul DMUX cu 4096 cai cu citeva circuite de dimensiuni reduse :un DMUX cu 64 cai si 8 MUX-uri cu 64 cai. Matricea de memorie are evident acelasi numar de celule in ambele de iesire. Concret, pentru un ROM cu n intrari si m iesiri, cu aceasta structura, dimensiunea (SIZE) va fi:

$$SIZE = \frac{n}{2} \cdot 2^{\frac{n}{2}} + m \cdot \frac{n}{2} \cdot 2^{\frac{n}{2}} = (m+n) \cdot 2^{\frac{n}{2}}$$

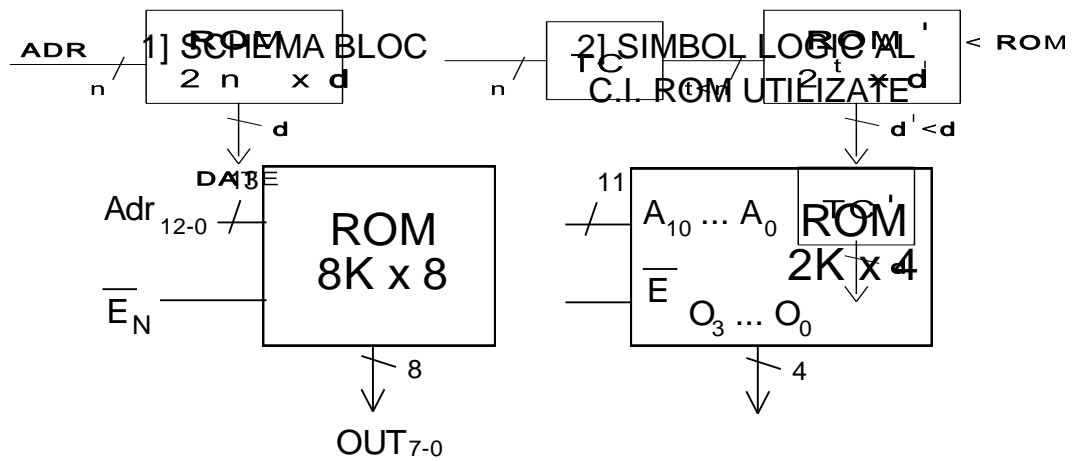
3] PROIECTAREA SCHEMEI CALCULUL NUMARULUI DE C.I. ROM

$$\frac{(8K \times 8)}{(2K \times 4)} = 4 \text{ LINII} \times 2 \text{ COLOANE}$$

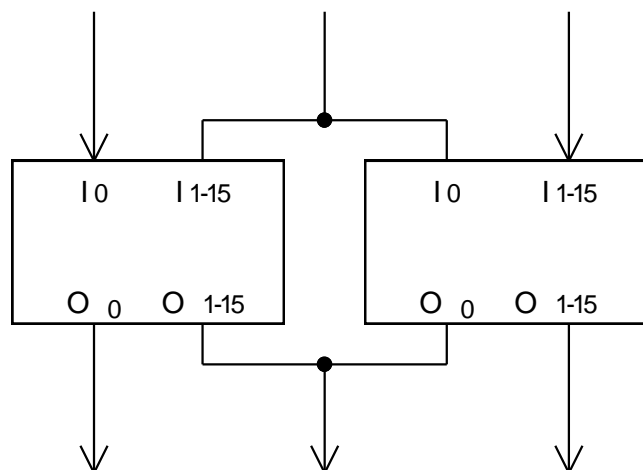
$$= 4 = 2$$



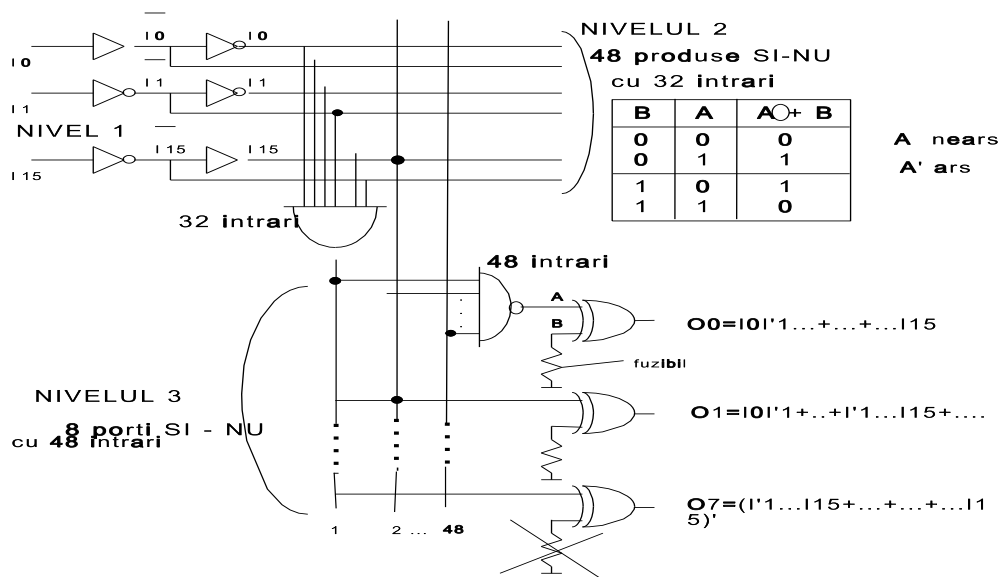
In concluzie, complexitatea circuitului nu mai variaza exponential cu 2^n ci cu $2^{n/2}$.



REDUCEREA DIMENSIUNII ROM PLA (Programable Logic Array)



EXTINDEREA



3.7. Sinteza privind metodele de implementare a functiilor logice

Toate categoriile de sisteme prezentate pina acum se pot utiliza in implementarea functiilor booleene. In fiecare situatie in parte, proiectarea presupune sau nu minimizarea functiilor, asa cum este prezentat in tabelul urmator. De asemenea, pentru fiecare metoda de implementare aleasa, trebuie utilizate reprezentarile adecvate ale functiilor (diagrame VK, tabel de adevar s.a.), specificate in coloana din mijloc a tabelului.

Metode de implementare	Minimizare algebrica (Reprezentarea functiilor)	Aplicatia
a) cu porti (SSI)	DA (Diagrame VK)	1 - 6 functii logice cu maxim 6 variabile
b) cu MUX (MSI)	NU (Tabel de adevar) pt. ca MUX contine in interior termeni canonici *se implementeaza prin introducerea variabilor in vectorul functiei	functii cu variabile numeroase de intrare
c) cu DCD (DMUX cu E' la masa) (MSI)	NU (Tabel de adevar) pentru ca DCD are in interior termeni canonici	6 ÷ 8 functii cu 4 ÷ 5 variabile
d) cu ROM (MSI - LSI)	NU (Tabel de adevar) metoda c) generalizata	4 ÷ 16 functii cu 5 ÷ 16 variabile de intrare
e) cu PLA (LSI)	DA (Prelucrari Boole) metoda a) generalizata	numar mare de functii cu mai mult de 12 intr.
f) cu circuite specializate	NU	presupune studierea catalogului

Ultima metoda, f), se refera la proiectarea cu circuite deja existente, dedicate unei aplicatii. Asemenea circuite exista, nu trebuie decit studiate din cataloage si utilizate conform specificatiilor fiecaruia. Citeva din cele mai uzuale circuite dedicate vor fi prezentate in continuare.

Sumatorul de 4 biti, 7483.

Circuitul realizeaza adunarea binara 2 numere de 4 biti, tinind cont de transportul (C- Carry) de la rangul anterior (C-) si generind transport catre rangul urmator (.C+). Circuitul are la baza sumatorul de 1 bit (Fig.3.101), descrisa de urmatorul tabel de adevar:

C	B	A	S	C ₊
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

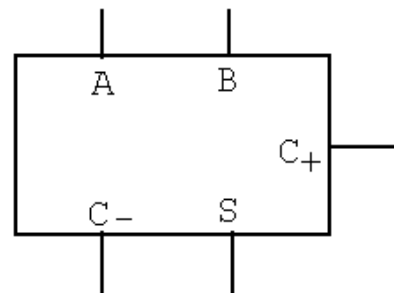
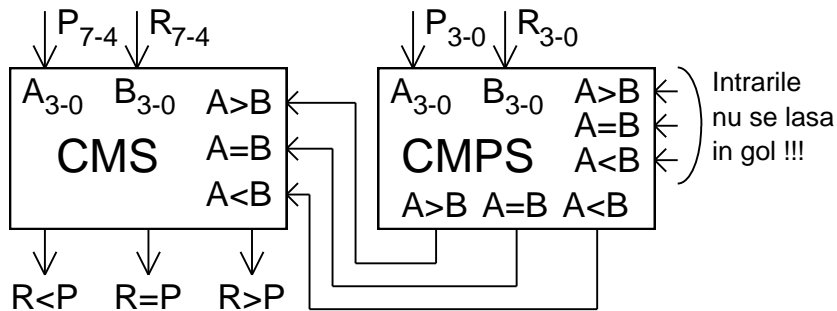


Fig. 3.101 Sumator complet de un bit

Circuitele 7483 contin 4 asemenea sumatoare si permit extinderea numarului de biti prelucrati, prin conectarea mai multor CI , cu transportul propagat de la rangul CMPS, catre rangul CMS.

Comparatorul de 4 biti, 7485

Fig. 3.102 Comparator de 8 biti



Contine are 8 intrari, cite 4 pentru fiecare din numerele de comparat , A_{3-0} si B_{3-0} . La iesiri apar rezultatele compararii, $A<B$, $A=B$, $A>B$ si se tine cont si de rezultatul compararii prin intrarile, $A<B$, $A=B$, $A>B$.

Extinderea numarului de biti de comparat se face prin utilizarea mai multor CI conectate in cascada, ca in fig.3.102. in care este proiectat un comparator de 8 biti. Intrarile de comparare ale CI CMPS sint conecteaza ca pentru $A=B$. Daca la comparaea bitilor CMPS rezulta ca P_{7-4} este mai mare sau mai mic decit R_{7-4} , acesta este rezultatul compararii, pentru ca nu mai conteaza bitii CMPS. Dar daca $P_{7-4} = R_{7-4}$, rezultatul compararii este dat de bitii CMPS si se propaga la iesiri prin comparatorul CMS

Unitatea Logico Aritmetica (Arithmetic Logic Unit - ALU) ,74181

Circuitul lucreaza cu doua numere binare de 4 biti fiecare, realizind 16 operatii logice (SI, SAU s.a) si 16 aritmetice (adunare, scadere). Cele 16 operatii se aleg cu bitii de comanda F_{3-0} . Bitul de selectie al operatiilor arirmetice si logice este M. Semnalele de transport de intrare si iesire sint utile numai la operatiile aritmetice. Semnalele de iesire P' si G' furnizeaza informatii despre transport si sint

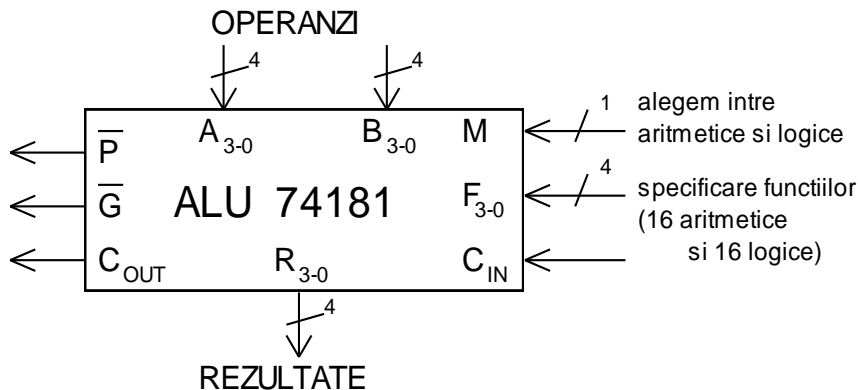


Fig.3.103. Unitate Logico- Aritmetica

utilizate la conectarea cu circuitul CLA 74182 ce se va prezenta ulterior. Pentru extinderea numarului de biti operabili cu ALU se utilizeaza mai multe CI conectate ca in fig. 3.104. Fiecare din cele 4 ALU prelucreaza anumite ranguri binare, transporturile fiind conectate $C_{OUT} - C_{IN}$, incepind de la circuitul de rang CMS pina la CMS.(liniile punctate din Fig. 3.104. In figura se neglijeaza deocamdata circuitul 74182 si conexiunile sale). Rezultatele prelucrarii unor numere de 16 biti se obtin pe 17 biti, datorita transportului final, C_p care apare in Autor SANDA MAICAN

cazul operatiilor de adunare. Trebuie remarcat ca rezultatul se obtine corect dupa 4 timpi de propagare ai ALU. In tot acest interval la iesiri apare o succesiune de raspunsuri incorecte datorate propagarii treptate a transportului. Reamintim ca acest fenomen se numeste hazard combinational

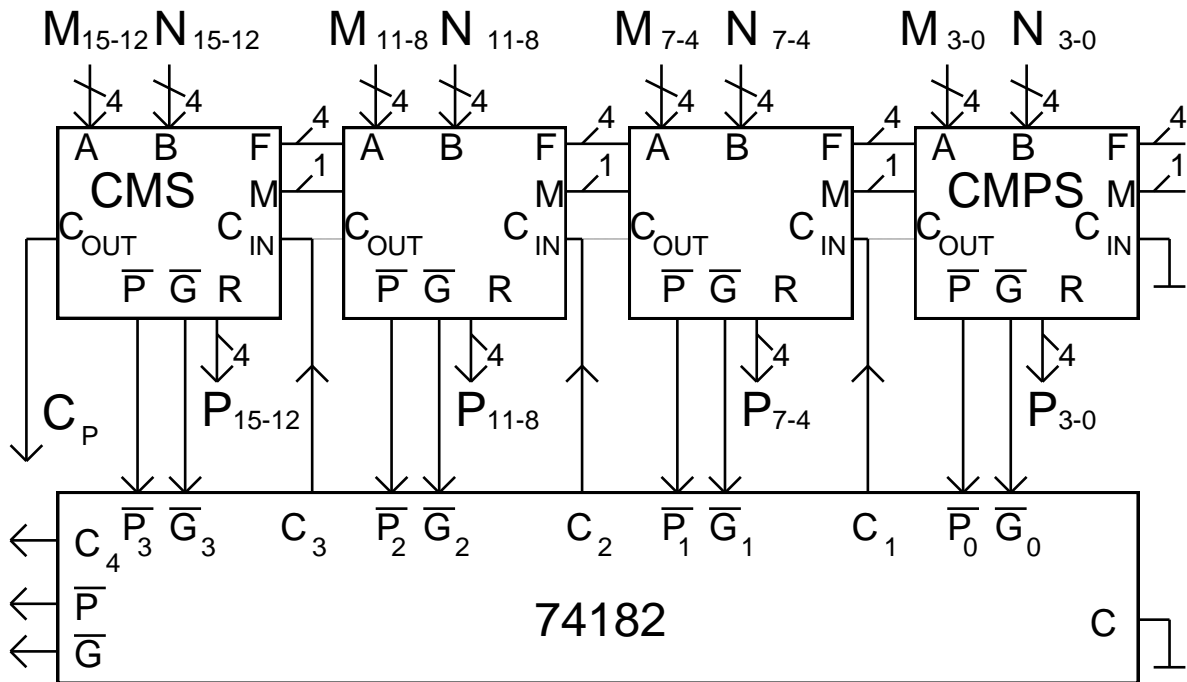
Circuit de propagare rapida a transportului, Carry Look Ahead

Circuitul CLA 74182, primeste semnalele P' si G' de la cele 4 ALU (fig.3.104) si genereaza transportul final C_4 intr-un singur timp de propagare al ALU, reducind perioada de hazard

Fig. 3.104. ALU de 16 biti si CLA

Combinational. In acest caz ALU nu mai au conectate C_{OUT} cu C_{IN} , pentru ca CLA genereaza in paralel transporturile de intrare pentru toate cele 4 ALU.

3.8. Hazardul combinational



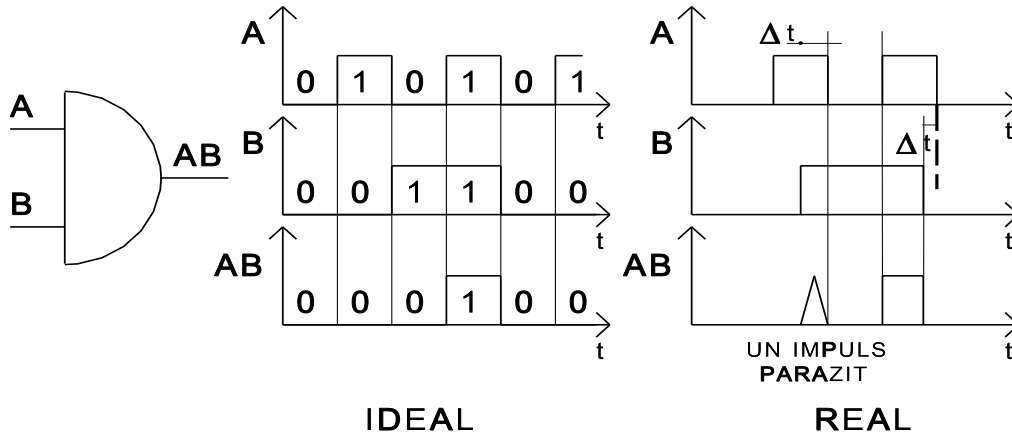
Asa cum am vazut la inceputul acestui capitol, hazardul combinational este una din proprietatile fundamentale al SLo.

Este un fenomen nedorit pentru ca raspunsurile incorecte ale sistemelor in intervalul de propagare, T_P , afecteaza circuitele comandate de SLo. Din acest motiv trebuie sa studiem cauzele de aparitie ale acestui fenomen pentru a sti apoi cum se pot inlatura efectele lui neplacute.

HAZARD COMBINATIONAL

1) CAUZE APARITIE

1.1] COMUTAREA NESINCRONA A INTRARILOR



IMPULSUL PARAZIT APARE LA COMUTAREA A DOUA SAU MAI MULTE INTRARI

Fig. 3.105 Raspunsul circuitului SI

3.8.1. Cauze de aparitie

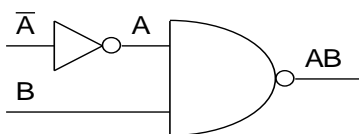
1. Comutarea nesincrona a intrarilor

Se considera cazul simplu al unei porti AND cu 2 intrari, careia i se aplica configuratiile binare in ordinea numararii. In cazul ideal, considerind semnalele sincrone (comutind simultan), raspunsul apare ca in fig.3.105. Pentru $A=B=1$, poarta AND raspunde cu 1. In realitate nu exista sincronism si semnalele A, B pot comuta decalat. Presupunem situatia defavorabila cind A soseste intirziat fata de B. Acest decalaj produce pe durata sa un raspuns suplimentar al AND, numit impuls parazit. Un astfel de raspuns apare numai la comutarea a doua sau mai multe semnale de intrare. S-ar evita impulsurile parazite daca intrarile ar varia in coduri GRAY (care schimba doar o singura variabila de la o configuratie la alta).

2. Propagarea pe cai de intirziere diferita..

In fig. 3.106 se observa ca semnalul din A' ajunge la iesire pe o cale mai lunga (un inversor) decit semnalul B. Aceasta intirziere prin inversor a lui A' produce aceleasi efecte ca in cazul anterior, adica aparitia unui impuls parazit. Formele de unda sint aceleasi ca in fig. 3.105.

1.2] PROPAGAREA IN-OUT PE CAI DE INTIRZIERE DIFERITA



Metode de evidentiere a hazardului combinational.

EVIDENTIAREA HAZARDULUI COMBINATIONAL:

- IN SCHEMA : ANALIZA INTIRZIERILOR,
- IN EXPRESII LOGICE : $f = AB + C\bar{B}$,
- IN DIAGrame VK : SALTURI INTRE SUPRAFETE.

Acestea depind de modul de reprezentare ale functiilor logice combinational:

- la reprezentarea prin expresii logice, HC este prezent in situatiile in

Fig. 3.106

care o variabila apare in anumiti termeni adevarata, iar in altii negata De exemplu la $f = AB + CB'$,

- B este variabila care produce hazard:
- la reprezentarea prin diagrame VK., hazardul este prezent daca apar salturi intre suprafete de acoperire alaturate (fig.3.107);
- in cazul schemelor se determina HC, punind in evidenta cai de propagare cu intirzieri diferite ale semnalelor de intrare.

3.8.2. Metoda de inlaturare a hazardului combinational.

2] METODA DE INLATURARE A HC: COD GRAY + INTRODUCERE DE PORTI SUPLIMENTARE

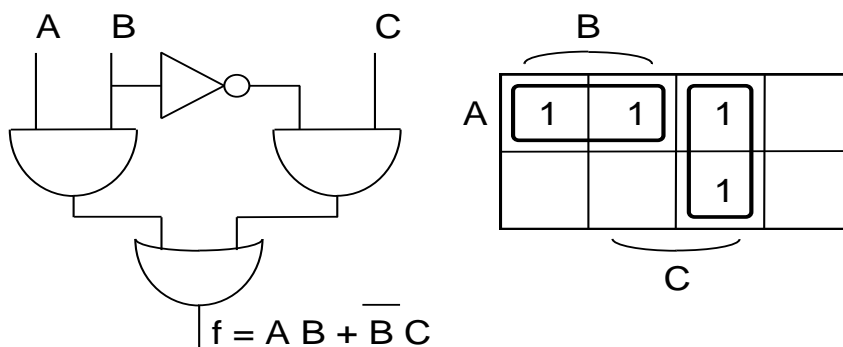


Fig. 3.107

Trebuie subliniat de la inceput ca metoda ce se va prezenta are o aplicabilitate limitata si nu constituie o solutie a majoritatii cazurilor. Ea consta in indeplinirea simultana a doua conditii

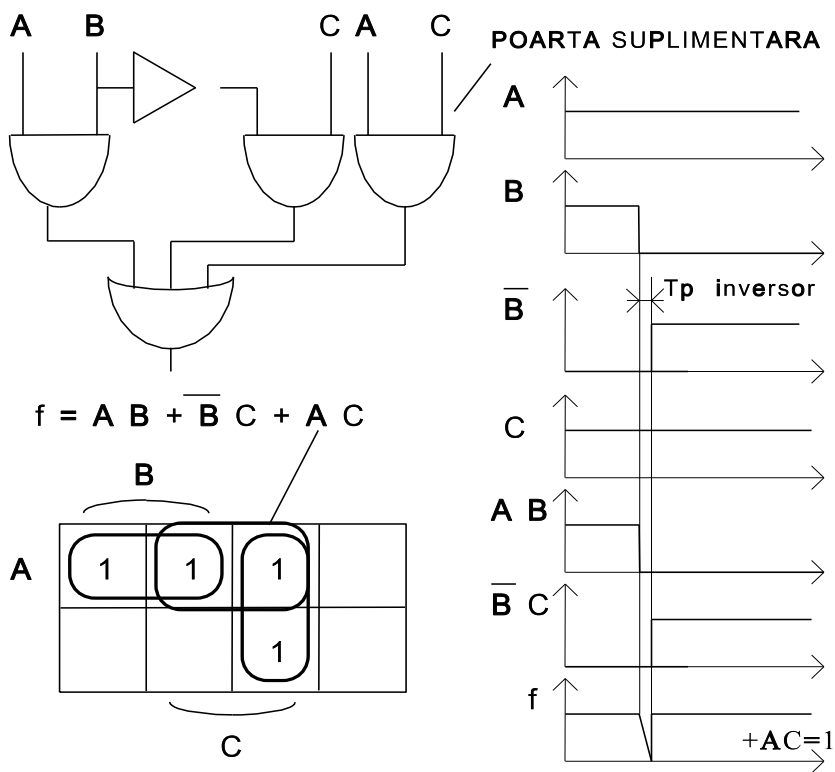
- comutarea semnalelor de intrare in cod GRAY (fapt neposibil in orice situatie;
- completarea schemelor cu porti suplimentare, corespunzatoare acoperirii salturilor intre suprafete din diagramele VK. In fig.3.108 se reprezinta modul de eliminare a HC pentru functia din fig. 3. 107 si formele de unda care demonstreaza anihilarea impulsului parazit.

Metoda are aplicabilitate limitata pentru ca nu intotdeauna putem avea coduri GRAY la intrarile circuitelor si nu toate functiile permit acoperiri de salturi intre suprafete (Ex. Functia Sau exclusiv $f = AB' + A'B$ va avea sigur HC).

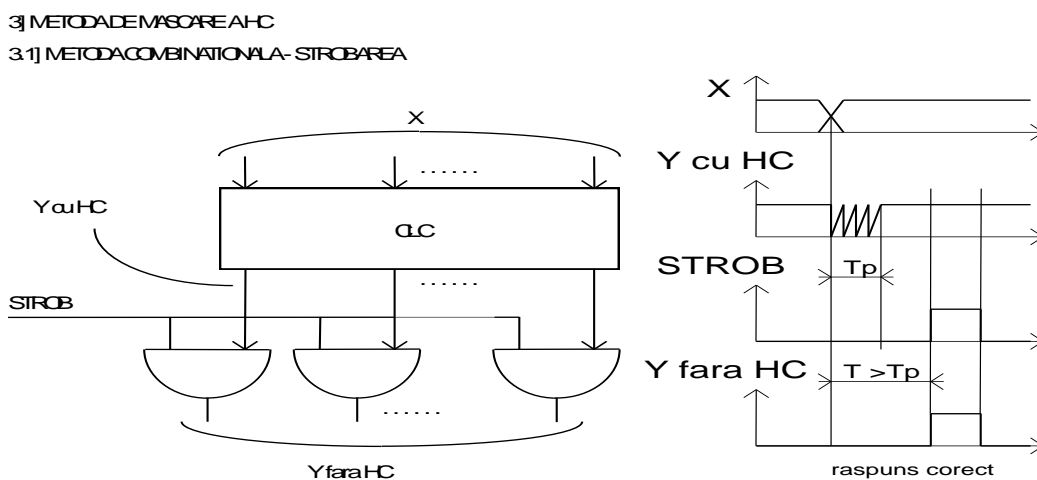
3.8.3. Metode de mascare a HC.

Aceste metode, spre deosebire de cea anterioara, sint general valabile si pot fi aplicate in toate situatiile. Se bazeaza pe urmatorul principiu: hazardul este lasat sa apara, dar este impiedicata transmiterea lui catre celelalte circuite.

3.8.3.1. Metoda combinationala de mascare a HC. STROBAREA.



Circuitele integrate sint proiectate cu strobare interna pentru a nu mai fi necesara adaugarea de porti suplimentare. Enable semnalul care inhiba functionarea CI complexe se foloseste si pentru strobare. In fig.3.110 se prezinta modul de eliminare a hazardului la DMUX.



3.8.3.2. Metoda secventiala de mascare a HC

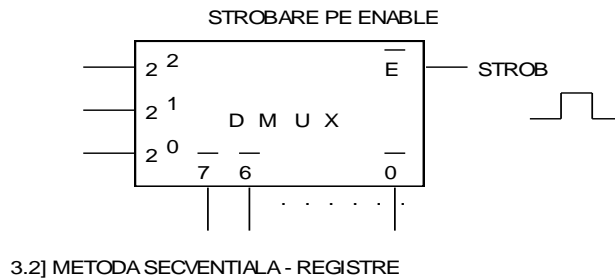


Fig. 3.110.

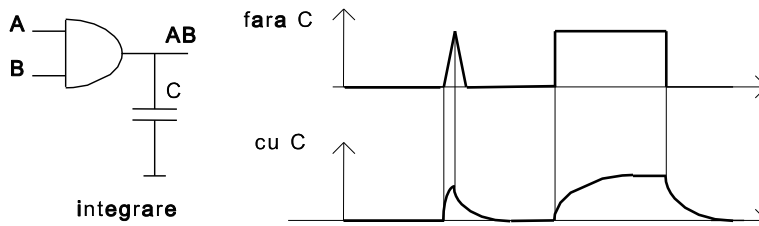
Aceasta se va studia la capitolul 4.4.2., la aplicatii ale registrelor.

3.8.3.3. Metode incorecta de mascare a HC

In fig. 3.111, condensatorul conectat la iesirea portii realizeaza o integrare a semnalului, atenuind impulsurile parazite pentru a nu mai actiona circuitele urmatoare. Metoda este incorecta pentru ca nu asigura o protectie stabila la semnalele parazite:

- datorita dispersiei valorii condensatorului si varierii in timp si cu temperatura a duratei impulsului parazit, nu se poate asigura disparitia totala a acestuia;
- daca se pune un condensator de valoare mare , care printr-o integrare puternica , ar elimina riscurile anterioare, acest strica si fornturile impulsului util, ceea ce poate produce oscilatia circuituluicomandat de poarta (vezi in subcapitolul 3.4.2.3 paragraful Caracteristica de transfer).

4] METODE INCORECTE DE ELIMINARE A HC



text

4. Sisteme logice de ordinul 1 Sisteme de memorare

Proprietati fundamentale:

- se obtin din sistemele de ordin 0 conectate in reactie.
- realizeaza memorarea semnalelor de intrare.
- datele dispar la decuplarea surselor de alimentare.

4.1 Bistabilul RS

Se numeste bistabil, un circuit care are 2 stari stabile.

La cuplarea alimentarii circuitul poate sa treaca in oricare din cele 2 stari. Comutarea intr-una din acestea este determinata de semnalele de intrare.

O varianta este prezentata in fig. 4.1.1. Bistabilul este realizat cu doua porti NAND legate in reactie.

Stari stabile: $QQ'=01$; $QQ'=10$ (Q si Q' sint in antifaza).

Intrarea R' (Reset), activa in 0, produce trecerea bistabilului in starea $QQ'=01$, cu $S'=1$ (inactiva).

Intrarea S' (set) activa, in starea $QQ'=10$, cind $R'=1$ (inactiv).

Daca $R'=S'=1$ se produce memorarea, bistabilul nu schimba starea pe care o avea.

Daca $R'=S'=0$, atunci $Q=Q'=1$, iar la trecerea $R'=S'=1$ nu putem cunoaste starea bistabilului. Ea poate fi sau $QQ'=01$ sau $QQ'=10$.

Deci $R'=S'=0$ trebuie evitat.

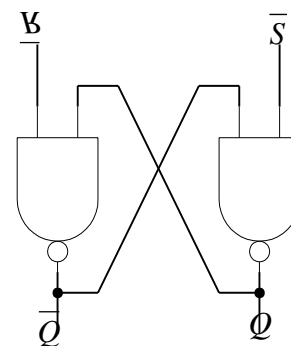


Fig. 4.1.1. Bistabil RS

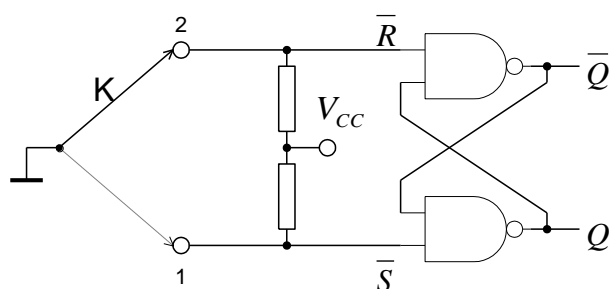


Fig. 4.1.2. Comutator care elimina impulsurile parazite.

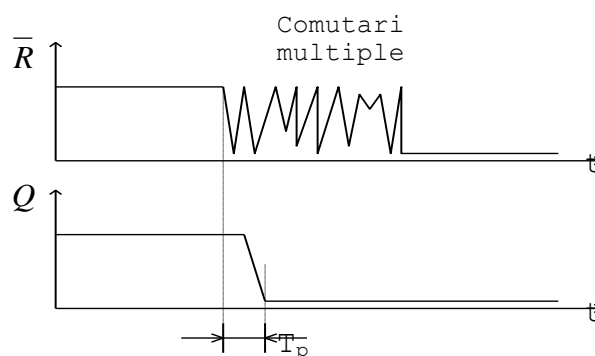


Fig. 4.1.3. Formele de unda ale comutatorului mecanic cu bistabil de tip RS.

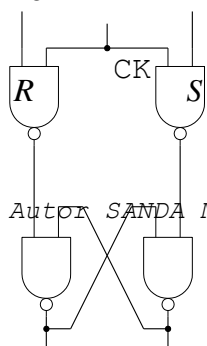
Aplicatii

Tampon

pot unei vor

pentru comutatoare bipolare (fig.4.1.2; 4.1.3.).

Comutatoarele mecanice dau comutari multiple. Acestea pot schimba starea (de ex. la numararea comutarilor unui buton se contorizeaza mai multe decat in realitate).



4.2. Bistabilul RS cu ceas (CLOCK) - prescurtat CK (fig. 4.2.1.)

Comutarea nu are loc decit pe palierul 1 al semnalului CK, datorita portilor NAND care permit trecerea spre bistabilul RS a semnalelor R si S numai cu CK=1.

Cu CK=0, se produce memorarea (bistabilul nu mai comuta).

4.3. Bistabilul de tip LATCH

Daca introducem un inversor intre intrarile R si S ale circuitului din paragraful anterior obtinem schema din fig. 4.3. care memoreaza data de la intrare. Cind CK=1 iesirea Q urmareste starea intrarii: Q=D, deci este transparent. Cind CK=0 intrarea este inhibata si Q nu se mai modifica, deci memoreaza starea anterioara.

Deci CK este o comanda de scriere in LATCH.

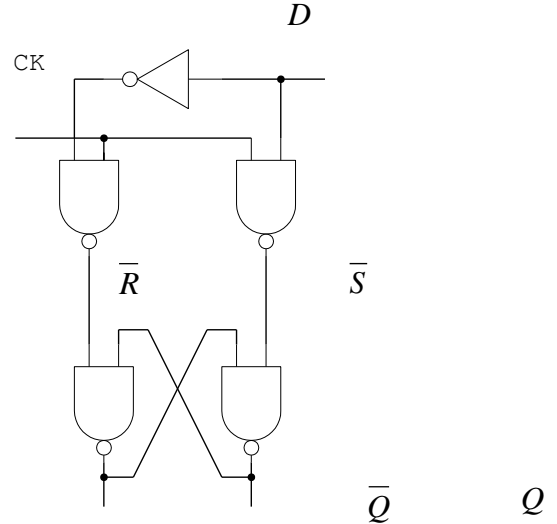


Fig. 4.3. Bistabilul de tip LATCH

4.4. Extensii paralele ale sistemelor logice de ordinul 1

LATCH-ul adresabil

(fig. 4.4.1;4.4.2) este un circuit format din mai multe celule de memorie (de tip LATCH).

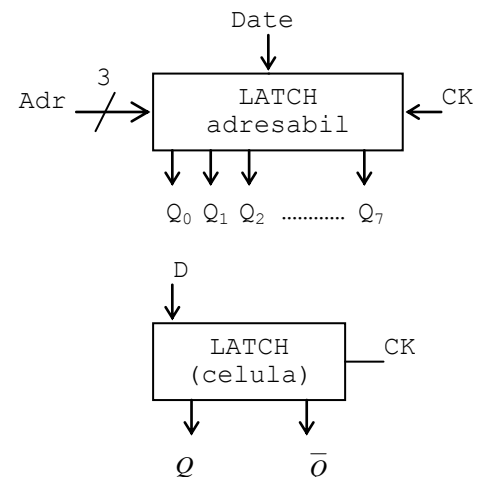


Fig. 4.4.1. Latch-ul adresabil si simbolul celulei din care este compus

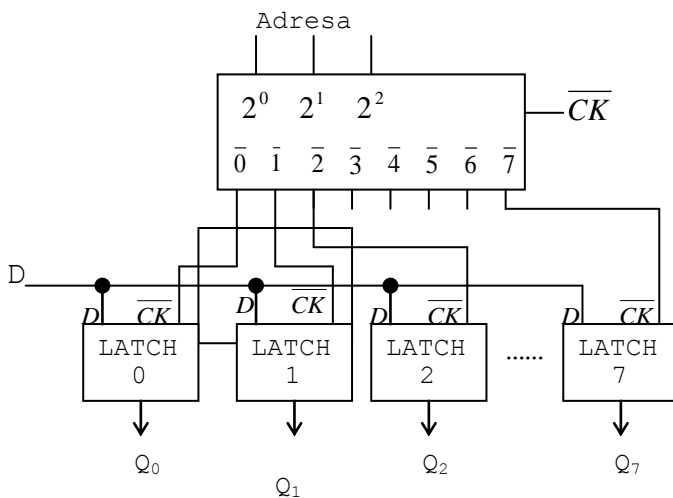


Fig. 4.4.2. Structura unui LATCH ADRESABIL

Putem scrie doar in Latch-ul selectat cu ajutorul unei adrese. Pentru scriere trebuie specificata adresa Latch-ului in care vrem sa scriem.

Celelalte nu vor fi modificate. Datele din toate Latch-urile apar simultan la iesiri, indiferent de adresa de la intrarea de adrese.

4.5. Memoria RAM (Random Acces Memory=memorie cu acces aleator) este o memorie in care se poate scrie si din care se poate citi.

4.5.1. Memoria RAM statica (**SRAM**=Static RAM) (fig. 4.5.1.1.)

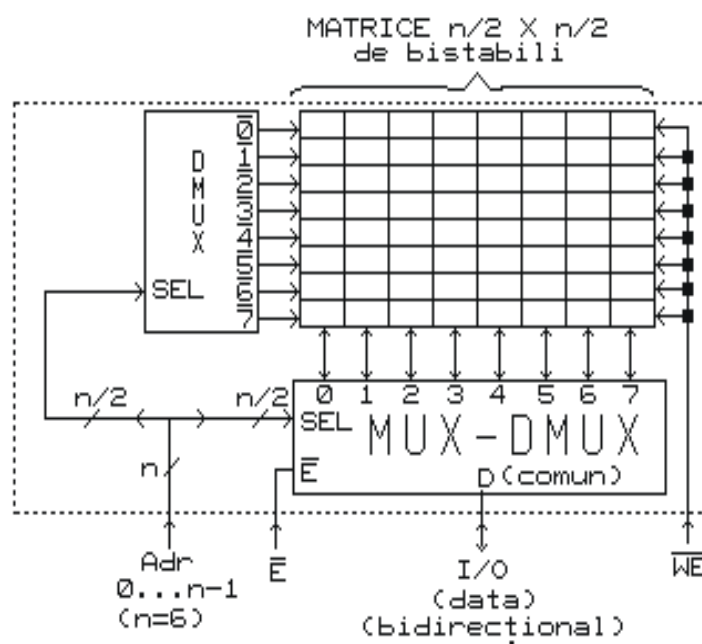
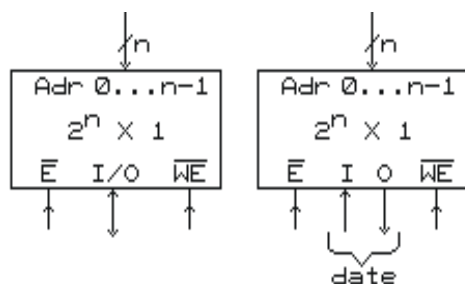
Sint extensii serie-paralel ale sistemelor de ordinul 1.
Memoria are doua moduri de lucru:

- $\overline{WE}=0$ impune scrierea datei de la pinul I/O in celula selectata de adresa. (WE=Write Enable=autorizare scriere)

- $\overline{WE}=1$ =regim de citire, data din celula selectata de adresa apare la pinul I/O (este deschis sensul multiplexarii).

Iesirile unei memorii RAM sint OC (Open Colector) sau TS, pentru a permite conectarea in paralel a mai multor circuite, pentru extinderea capacitatii de memorie. Daca circuitul nu este selectat ($E'=1$), atunci iesirea este in starea de mare impedanta (sau 1 logic la OC) si nu se poate scrie. Unele circuite au intrarea de date diferita de iesire (pini diferiti - fig.4.5.1.2.).

Capacitatea unui singur circuit integrat poate fi de exemplu 256x1 (eventual in tehnologie bipolară)...1Mx4 (de



tip MOS).

MUX-DMUX se realizeaza cu porti de transmisie (uzual CMOS).

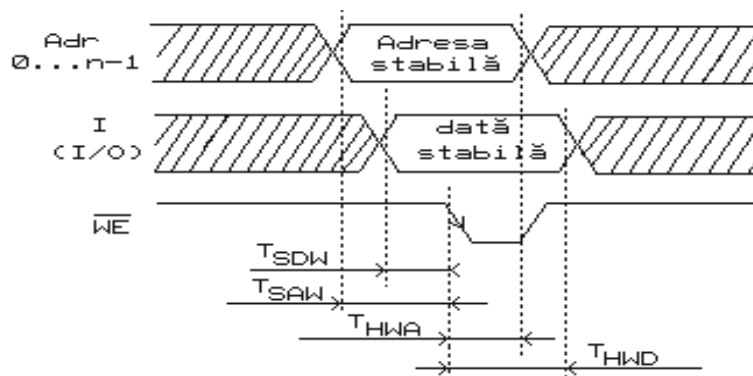
SECVETE DE COMANDA ALE MEMORIILOR RAM STATICE (fig. 4.5.1.3.)

a) Secventa de comanda la scriere in SRAM

Pentru scriere trebuie respectata succesiunea:

- fixarea adresei si a datelor la intrarile corespunzatoare.

- emiterea comenzii de scriere (scrierea are loc la tranzitia din 1 in 0 a semnalului WE').

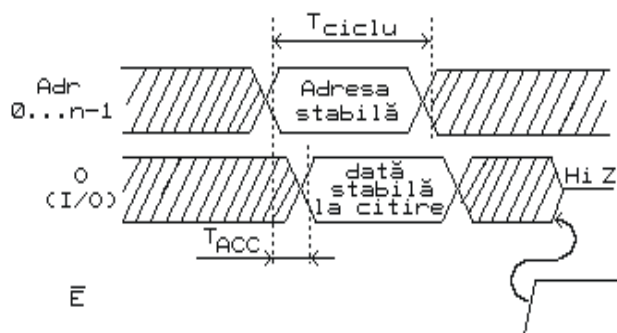


Dupa ce adresa este stabila la intrari, se va astepta un timp TSAW (T Set- up Adress to Write) mai mare ca cel specificat in catalog, apoi se poate genera front negativ la intrarea de scriere. Similar, dupa ce datele sint stabile la intrarea de date, frontul de scriere se va aplica dupa un timp TSDW (T Set-up Data to Write), mai mare ca cel specificat in catalog (datorita propagarii prin schema interna a circuitului). Datele si adresele vor fi stabile la intrari (nu se vor modifica) si dupa frontul de scriere un timp "de mentinere"(hold): THWD (T Hold Write to Data), respectiv THWA (valorile minime se dau in catalog).

Producatorul de circuite specifica un timp de cilu T_{ciclu} =durata unui ciclu=intervalul de timp de la inceputul ciclului (de scriere in acest caz) pina la inceputul urmatorului ciclu.

Mai apar si alti timp*i* ce trebuie respectati (de exemplu durata palierului 0 la intrarea WE, etc.) indicati in cataloage.

b) Secventa de comanda pentru citire (fig. 4.5.1.4.)

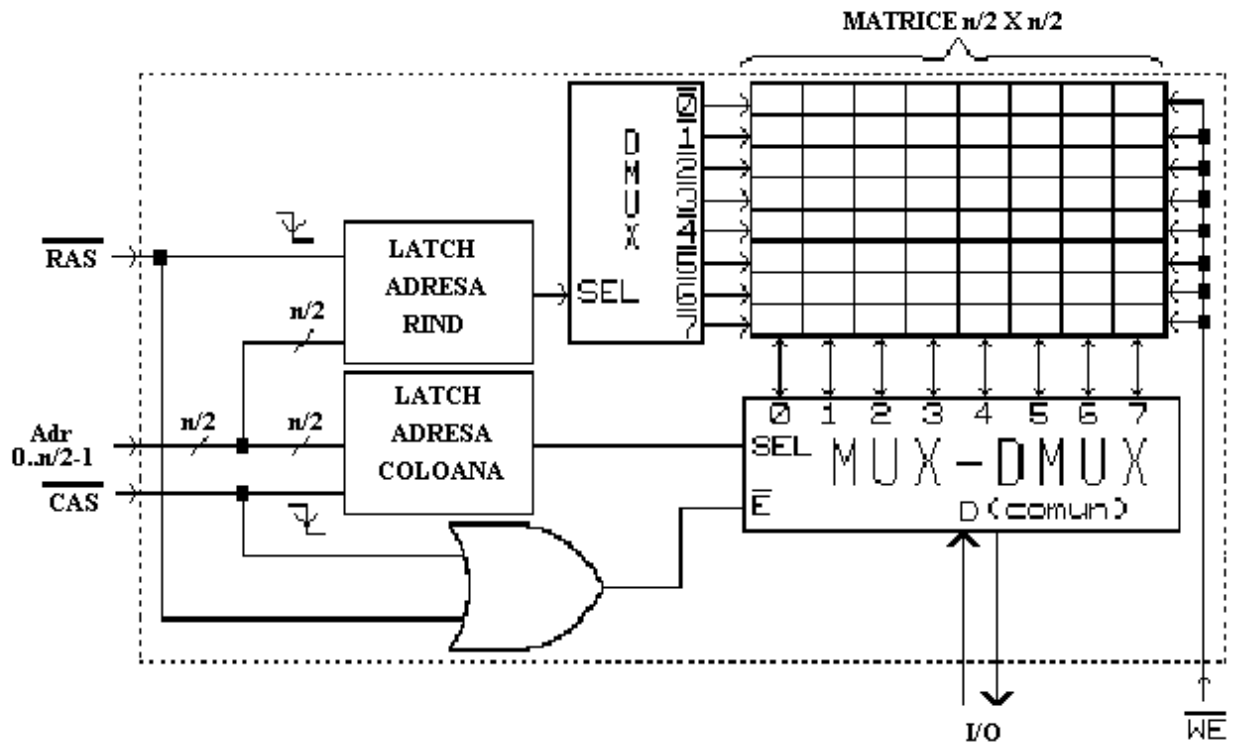


Se stabileste adresa de unde vom citi si $\overline{WE}=1$.

Dupa un timp TACC (timp de acces la date) data apare stabila la iesire. Valoarea maxima este specificata in catalog (reprezinta practic un timp de propagare de la adresa la iesire). Ex.: 30ns pentru HCMOS si bipolar; 100ns la MOS.

4.5.2. Memoriile RAM dinamice

DRAM



Cresterea capacitatii de memorie pentru o tehnologie ce nu permite cresterea densitatii de integrare se poate realiza doar prin reducerea complexitatii unei singure celule si a numarului de pini de interconectare (care consuma o arie mare de siliciu).

Reducerea complexitatii unei celule se realizeaza prin reducerea numarului de tranzistoare ale fiecarei celule. Nu se mai folosesc LATCH-uri ci condensatoare (realizate folosind capacitatea de poarta a tranzistorului MOS). Acesta introduce o problema suplimentara: in timp se descarca, deci pierde informatia. Din acest motiv apare necesitatea reimprospatarii (**REFRESH**) care reface sarcina din condensator.

Adresa nu se mai transmite cu toti bitii simultan, ci succesiv in doua etape: jumatate in prima etapa, jumatate in a doua etapa. Se reduce numarul de conexiuni cu exteriorul.

Transmiterea se face cu doua semnale de STROBE: RAS si CAS

RAS = Row Adress Strobe -memoreaza adresa de rind (ce rind selectez din matricea de celule)

CAS = Column Adress Strobe -memoreaza adresa de coloana (ce coloana transfera bitul de date)=selectia MUX-DMUX din calea de date.

Schema interna este prezentata in fig. 4.5.2.1.)

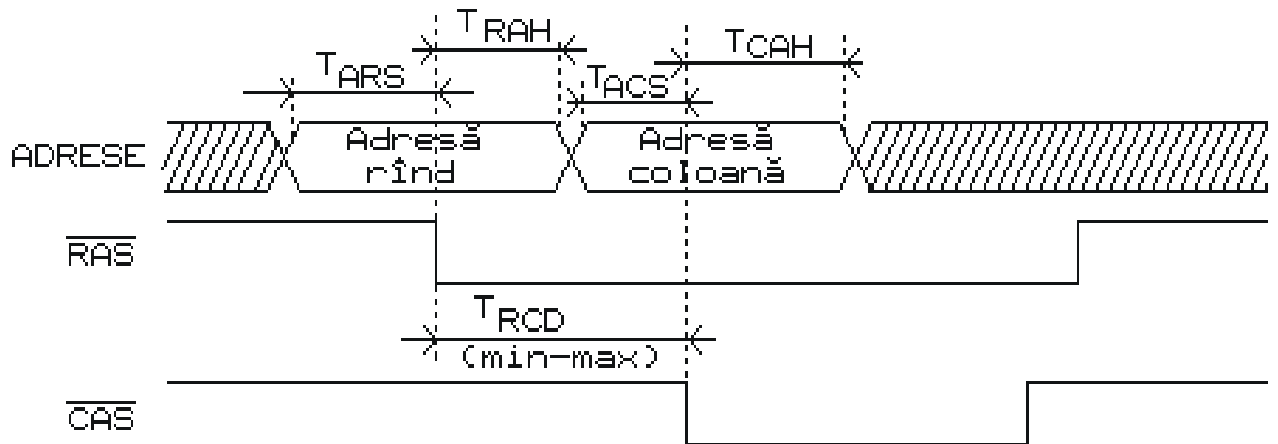
COMANDA MEMORIILOR DRAM

ADRESAREA MEMORIEI DRAM SE REALIZEAZA IN PATRU SECVENTE:

- Transmiterea adresei de rind (linie) la circuit (adresa stabila);
- Validarea acesteia (preluarea adresei de linie) cu strobul RAS;
- Transmiterea adresei de coloana la circuit;
- validarea acesteia cu strobul CAS;

Exista trei moduri de adresare a memoriilor dinamice:

- 1) Normala;
- 2) Paginata=transfer date din celulele de pe aceeași linie (rînd), deci nu mai dau mereu adresa de rînd (economie de timp), ci doar adresa de coloana (dau o singură adresă de rînd pentru fiecare pagină).
- 3) **NIBBLE**=specific o adresă inițială, apoi transfer patru date plasate la adrese consecutive, incrementînd un numărator intern de doi biți.



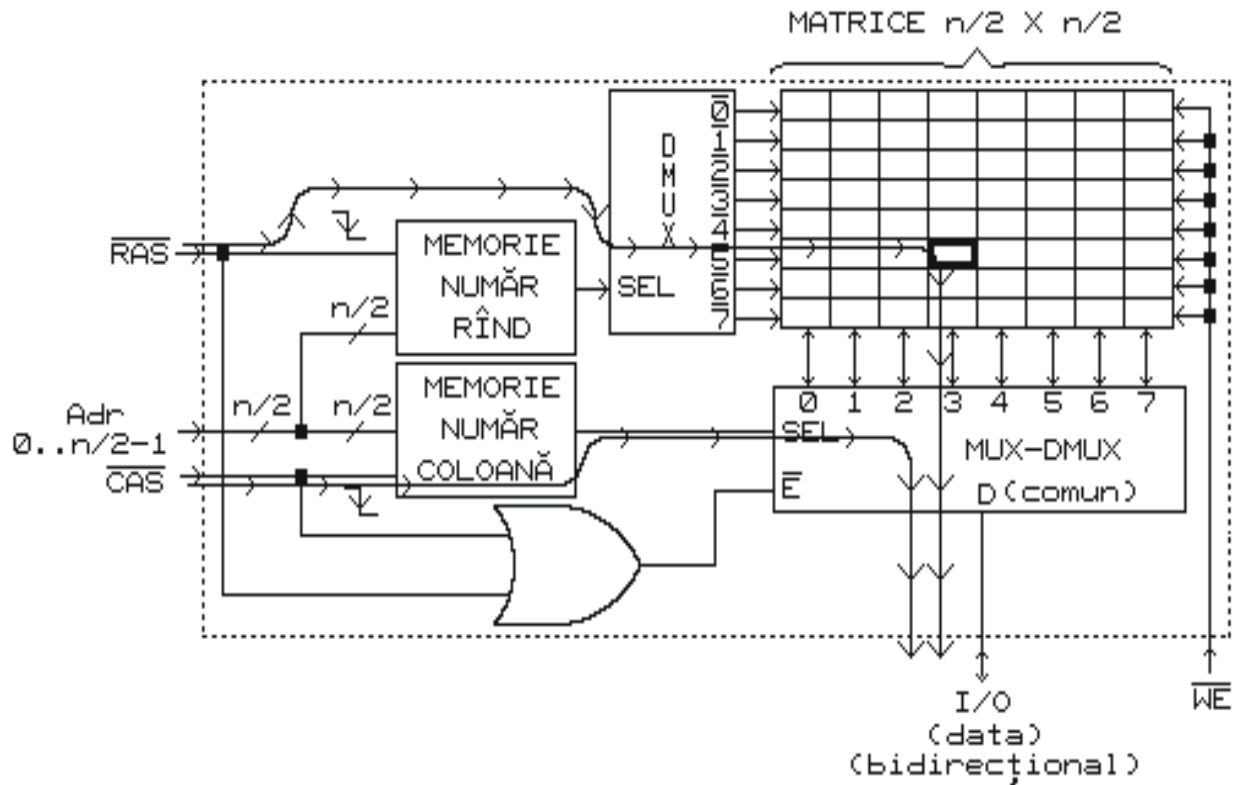
1) ADRESAREA NORMALA (fig. 4.5.2.2.)

- Stabilim adresa de linie;
- După TARS (Address to RAS Set-up) se generează front negativ pe RAS;
- După TRAH (Row Address Hold) schimbă adresa cu cea de coloana;
- După ce adresa de coloana este stabilă, se așteaptă un timp TACS (Address to CAS Set-up), se emite un front negativ la strob-ul CAS' ;
- Adresa de coloana va mai fi stabilă un timp de hold TCAH (Column Address Strobe);

Dacă dorim scriere, semnalul WE' va fi 0 (stabil) anterior frontului negativ al CAS' (acest front produce scrierea) cu un timp de set-up. Acest front al CAS' nu trebuie să apară după cel negativ al RAS' mai devreme ca TRCDmin (RAS to CAS Delay). Valoarea maximă a acestui timp nu are la scriere nici o importanță.

Dacă facem citire, WE' va fi stabil 1 înaintea CAS' cu un timp de set-up.

Timpul de propagare de la strobarea adresei de linie (cu RAS) pînă la ieșire este: TRAC (Row Acces time=timp de acces de la RAS)= Tp de la RAS la ieșirea registrului de memorare a adresei de linie + Tp celula de memorie+TpMUXdate. Timpul de propagare de la strobarea adresei de coloana (cu CAS) la ieșire TCAC este format doar din Tp de la CAS la ieșirea registrului de memorare a adresei de coloana + TpMUX. Acest timp este mai mic decît primul (fig. 4.5.2.3.; fig. 4.5.2.4.a,b).

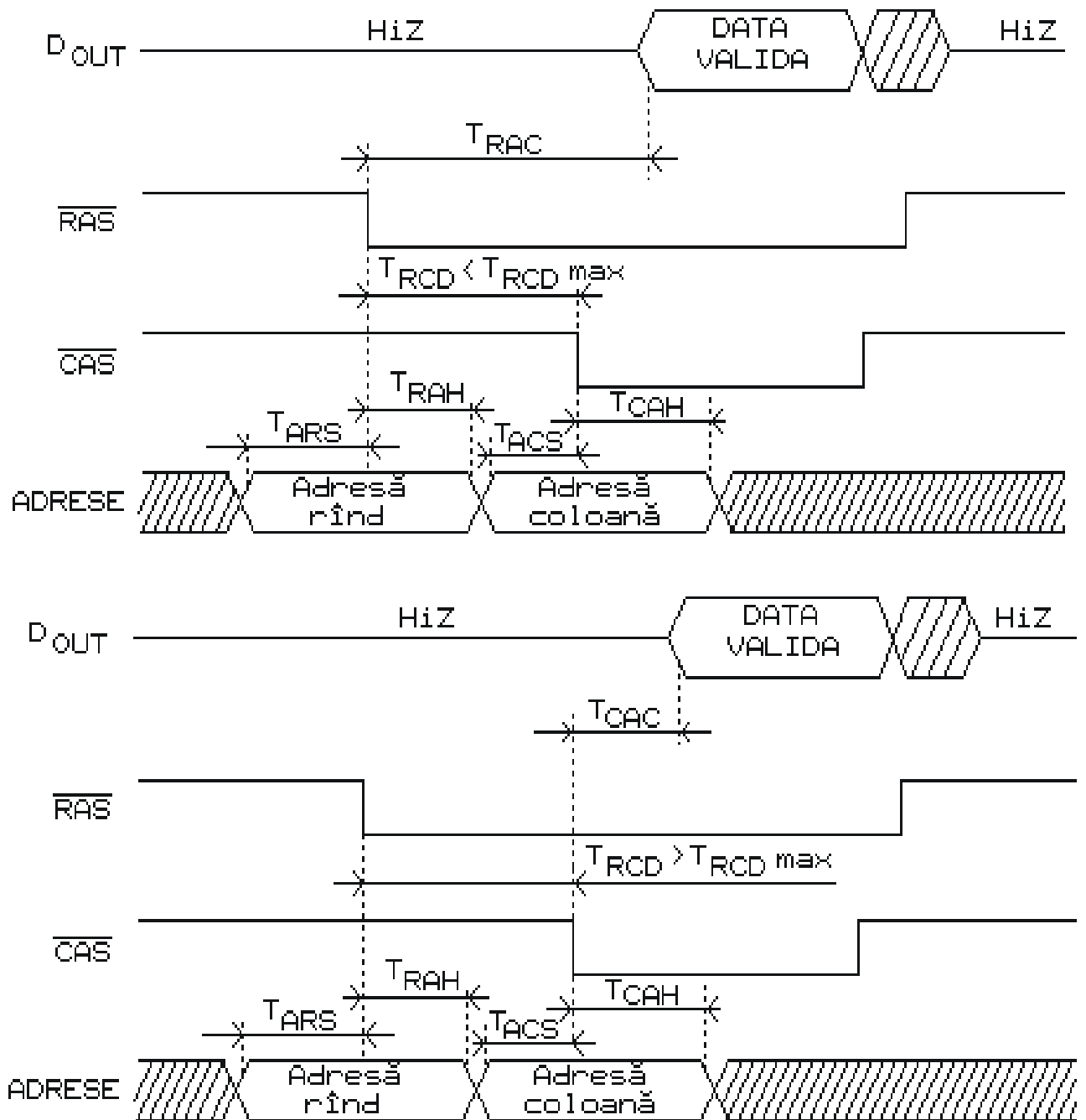


Daca strobul \overline{CAS} apare suficient de devreme dupa \overline{RAS} (intre $TRCD_{min}$ si $TRCD_{max}$), desi MUX-ul de date a selectat deja coloana, nu a avut loc propagarea de la RAS la MUX-ul de date (la iesire nu apar date stabile!). Deci datele la iesire apar dupa $TRAC$, independent de momentul cind a fost dat CAS (depind de momentul cind a fost dat RAS).

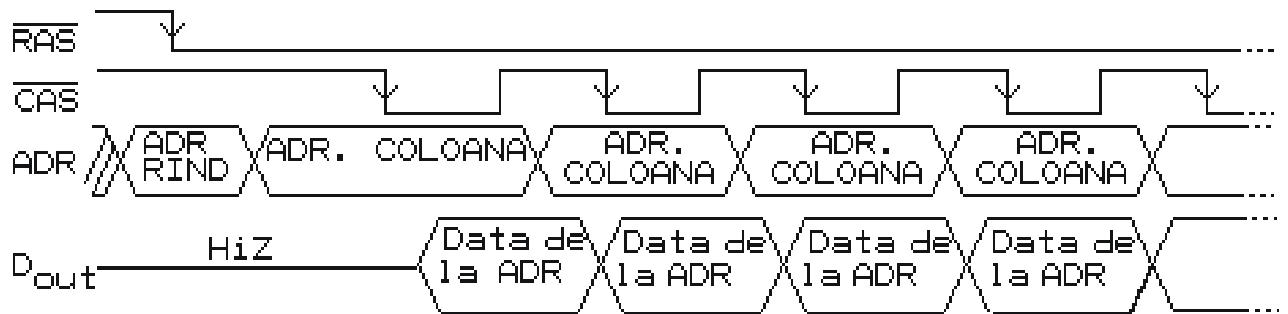
Daca CAS apare mult mai tirziu ca RAS (peste $TRCD_{max}$), datele sint deja prezente la intrarea MUX-ului, dar acesta nu a selectat coloana. Datele vor fi stabile la iesirea circuitului functie de CAS (dupa $TCAC$).

Deci $TRCD$ poate fi mai mare ca $TRCD_{max}$ (uzual $TRCD > TRCD_{max}$!) si nu afecteaza functionarea schemei ci specifica doar timpul dupa care datele apar stabile la iesire.

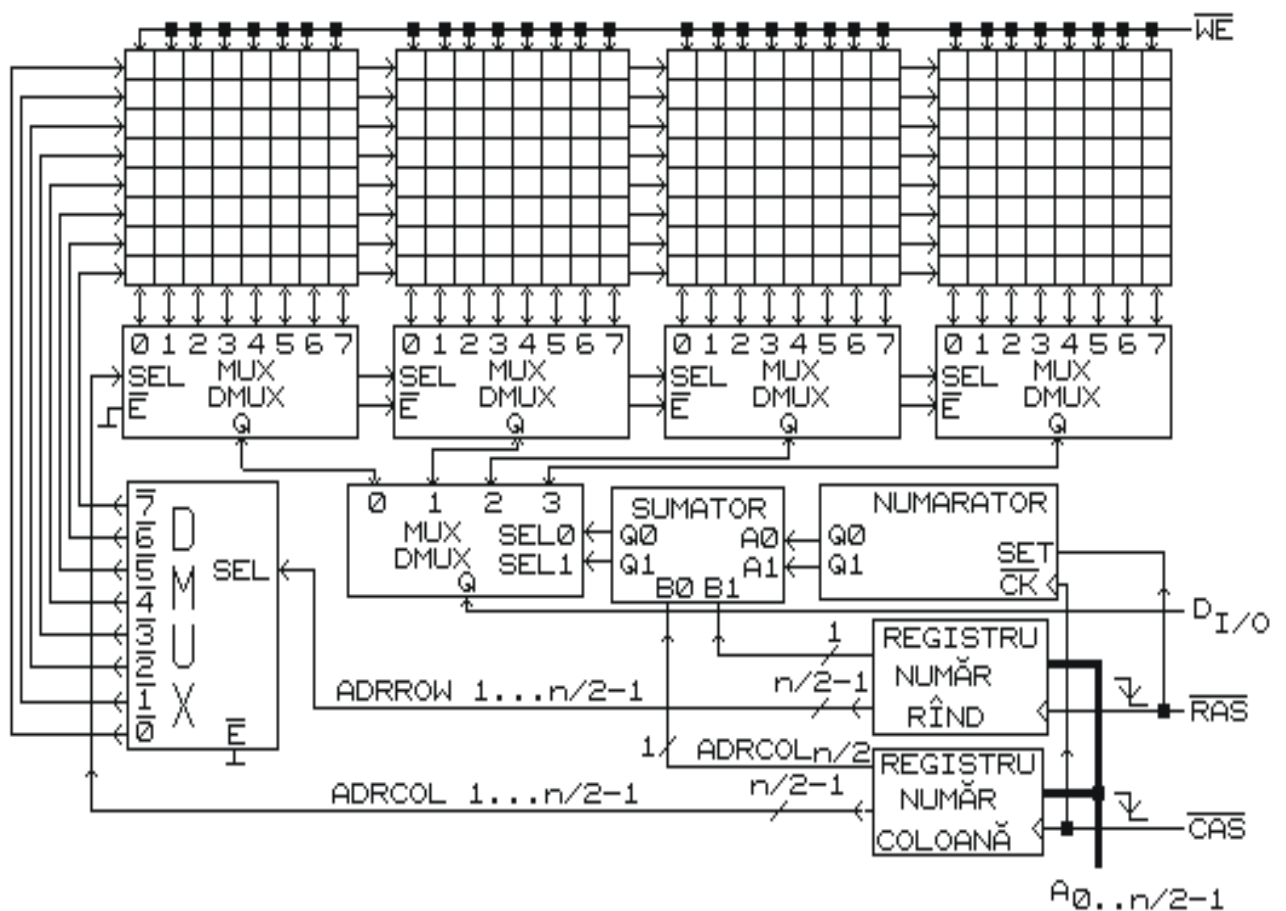
Se mai observa ca iesirea de date este in stare de inalta impedanta daca oricare dintre semnalele RAS si CAS este 1 (nu la toate circuitele).

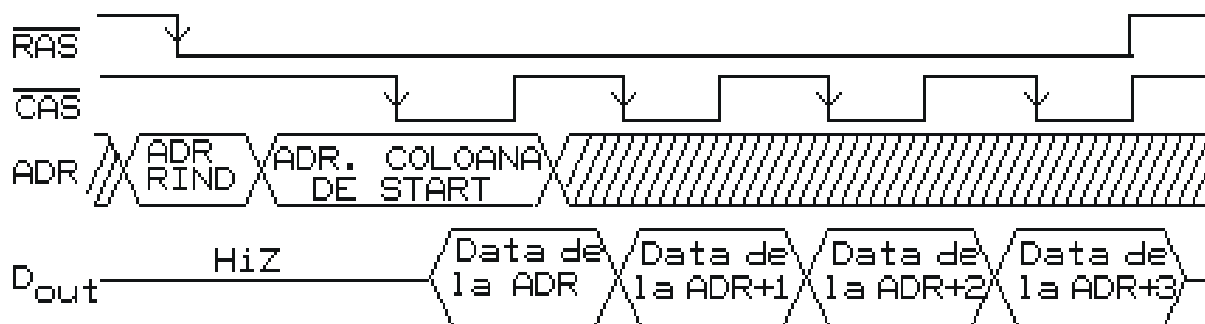


2) Adresarea paginata (fig. 4.5.2.5.)



Acest mod creste viteza de lucru. Se poate aplica la schemele care transfera date de la adrese succesive: se adreseaza rindul din matricea de celule de memorare, apoi generam doar adresele de coloana ale celulelor pe care le utilizam (deci vor fi dintr-un singur rind!).





3) Adresarea NIBBLE (fig. 4.5.2.6.; fig. 4.5.2.7.)

Creste si mai mult viteza de lucru.

Circuitele care lucreaza in acest mod au 4 matrice de memorie (fig. 4.5.2.6.). Dintre acestea numai una este cuplata la pinul de date. Selectarea se face cu ajutorul unui numarator de 2 biti. Adresa de coloana este suma dintre numarator si adresa memorata de stroburile CAS' si RAS' (numaratorul contine 00 dupa primul CAS, deci prima adresa de coloana este chiar cea strobata). Atentie! Adresele succesive care se obtin nu difera decit prin 2 biti. Deci, daca adresa de start este: linia **1000** si coloana **0000**, urmatoarele adrese se obtin prin incrementarea numarului 10 (format din bitii mai semnificativi ai adreselor de linie si coloana -scrise mai sus cu caractere groase-. Adresele vor fi: **1000 1000**; **0000 0000**; **0000 1000**.

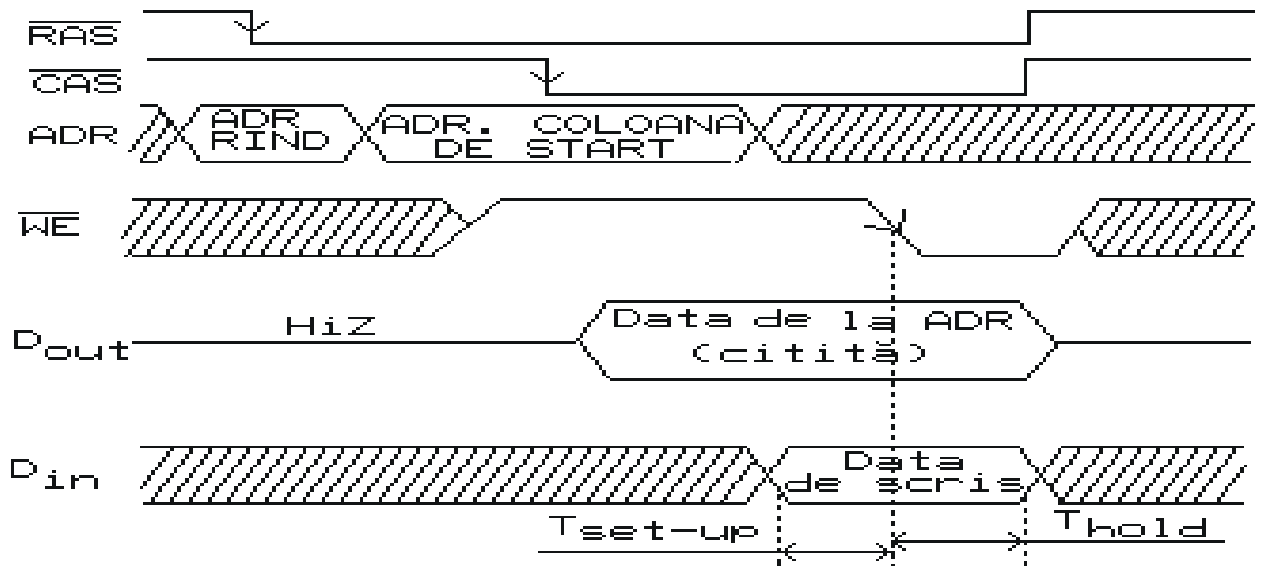
Din bitii de adresa de linie unul se duce la sumator, iar ceilalti selecteaza liniile din cele 4 matrice (simultan in toate 4). Similar, unul din bitii din adresa de coloana se sumeaza cu continutul numaratorului, iar ceilalti selecteaza coloanele din cele 4 matrice.

Sumatorul va selecta una din cele patru matrice de memorie ce va fi conectata la pinul de date. Urmatoarele fronturi active ale semnalului CAS vor incrementa numaratorul, care va selecta una dupa alta cele patru matrice.

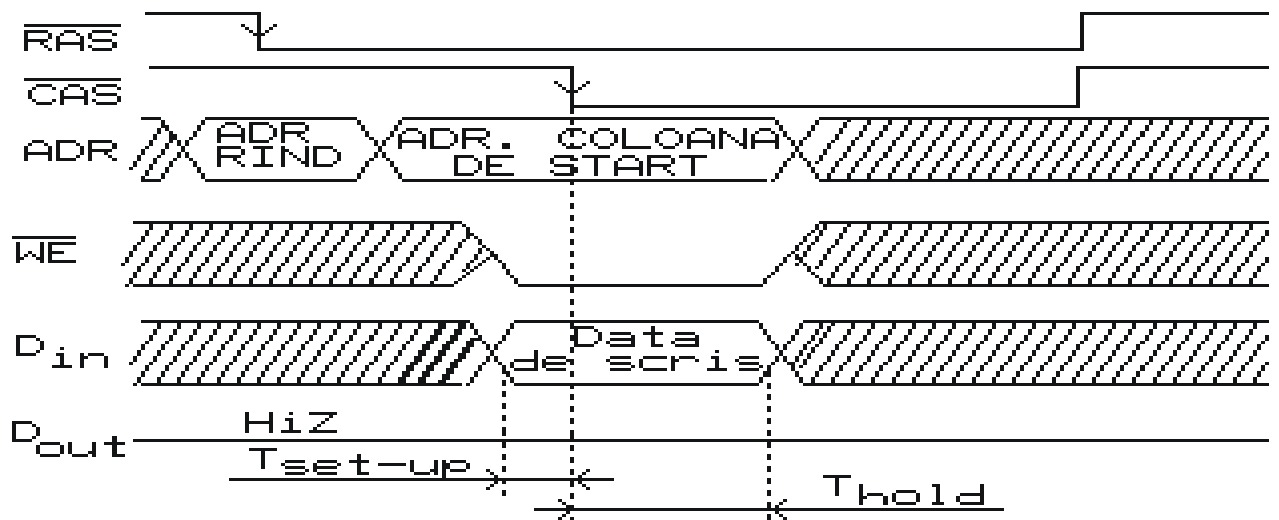
Cicluri de scriere in memoria DRAM

Toate modurile de adresare permit toate tipurile de cicluri de scriere!

1) READ MODIFY WRITE (citire modificare scriere; fig. 4.5.2.8.). Uneori este necesara citirea unei date, prelucrarea ei, apoi scrierea datei prelucrate la aceeaasi adresa. Pentru economie de timp nu se genereaza un ciclu de scriere separat. Dupa ce data a fost citita intr-un ciclu de citire, se pastreaza stob-urile (RAS', CAS') pe nivel 0 si, dupa ce data citita a fost prelucrata de circuitul exterior, aceasta este rescrisa la aceeaasi adresa prin simpla trecere a semnalului WE' pe nivel 0. Frontul negativ al acestui semnal comanda scrierea.



2) EARLY WRITE (fig. 4.5.2.9.) este un ciclu simplu de scriere, (doar scriere nu si citire) in care semnalul WE'=0 apare inaintea frontului negativ al CAS. Din acest motiv iesirea memoriei nu iese din starea de



inalta impedanta.

REFRESH-ul memoriilor dinamice

La memoriile dinamice celulele de memorare nu sint latch-uri ci tranzistoare MOS la care este folosita capacitatea de poarta (aceasta poate fi incarcata cu sarcina sau nu, determinind astfel conductia sau blocarea tranzistorului). Acest condensator se descarca in timp, fiind necesara reincarcarea lui. Procesul de refacere a sarcinii pe condensator este numit **REFRESH** (REIMPROSPATARE). Aceasta se realizeaza prin simpla adresare a celulei de memorie. Cum celulele de memorie dintr-o linie sint selectate tote simultan si adresa de coloana nu face decit sa "lege" iesirea de date a celulelor dintr-o coloana la pinul de date (deci nu participa la activarea celulelor), REFRESH-ul se realizeaza pentru toate celulele dintr-o linie simultan, deci este

suficient sa adresam linia respectiva.

Pentru a nu se altera informatia memorata este necesar ca fiecare linie de memorie sa fie selectata (cu adresa de rind) la un interval de timp mai mic decit TRFHSmax (de ex. 2...8ms). Daca memoria respectiva este folosita intr-o schema care citeste sau scrie din toate liniile suficient de repede (de exemplu o memorie video de unde se citeste permanent pentru a avea o imagine pe ecranul unui monitor si se scrie doar cind apar modificari in acea imagine), refresh-ul este asigurat fara un ciclu special. Daca nu sintem siguri ca citirile si scrierile acopera complet toate liniile matricei de memorie in timpul TRFSHmax, trebuie introduse cicluri de REFRESH care consta doar din generarea si strobarea cu RAS a adresei de rind, dupa care RAS revine in 1. La acestea se va specifica doar adresa de linie, deoarece REFRESH-ul se realizeaza simultan pentru toate celulele dintr-o linie.

Nu are importanta daca aceste cicluri sint grupate sau nu, important este doar ca fiecare linie sa fie adresata intr-un mod oarecare intr-un timp mai mic decit TRFSHmax.

Extinderea memoriilor DRAM

1. Datorita transiterii adresei de linie succesiv in doua jumatati, este necesar un MUX pentru bitii de adresa. Acelasi MUX va fi folosit si pentru adresa de REFRESH daca este necesar. Adresele pot fi multiplexate in orice mod, important este doar ca adresa unde se scrie o informatie sa fie aceeasi cu cea de unde o citim.

2. Semnalele RAS, CAS vor fi utilizate (unul din ele) pentru selectia circuitului activat. Deoarece preferam sa facem REFRESH simultan in tote CHIP-urile de memorie, vom avea semnalul RAS comun pentru toate circuitele, deci CAS va fi cel ce va face selectia.

Proiectarea matricelor de memorie respecta aceleasi etape ca la ROM-uri.

EXEMPLU: Priectati un RAM de 4Mx8biti cu circuite integrate DRAM de 1Mx1bit

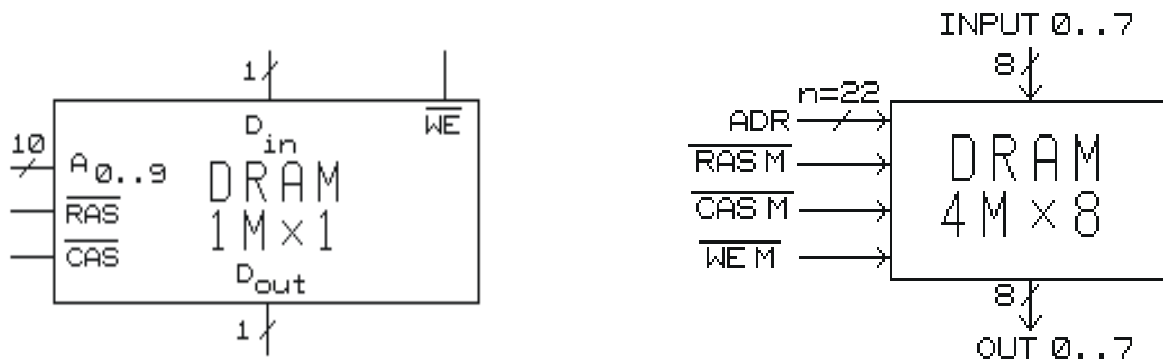


Fig. 4.5.2.10.

- 1) Schema bloc a memoriei mari (fig. 4.5.2.10.a.)
- 2) Schema bloc a circuitului integrat (fig. 4.5.2.10.b.)
- 3) Capacitatea: $(4M \times 8) / (1M \times 1) = 4 \text{ linii} \times 8 \text{ coloane}$.

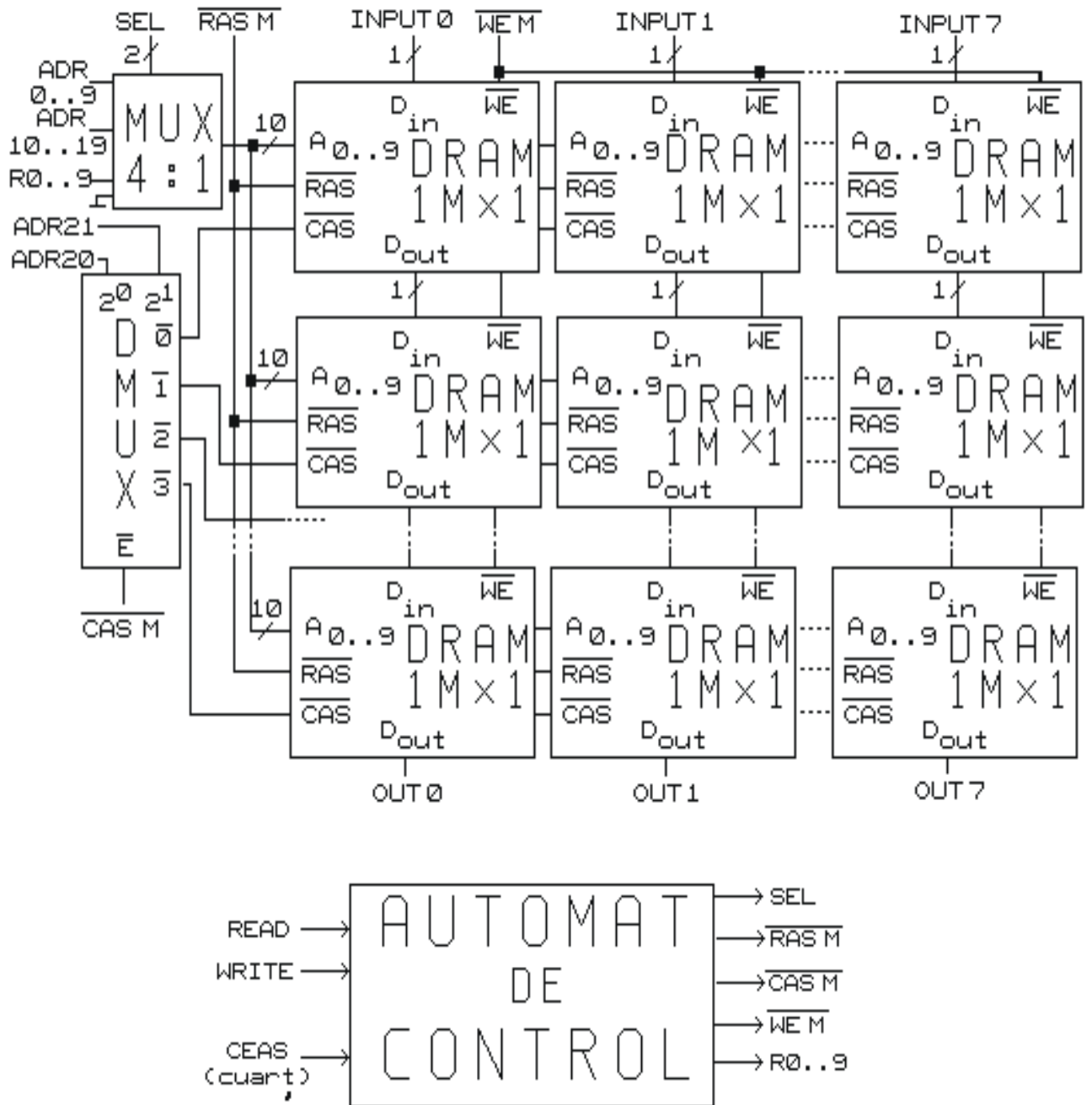
In fig. 4.5.2.11. este prezentata schema memoriei.

Automatul care comanda schema (fig. 4.5.2.12) trebuie sa asigure succesiunile necesare fiecarui tip de ciclu folosit, inclusiv REFRESH daca este necesar.

Ex.: Pentru generarea adresei automatul va comanda SEL (se propaga prin MUX adresa de rind), apoi RAS (adresa de rind este strobata), apoi schimba SEL (se propaga adresa de coloana), apoi va genera CAS.

OBS.

Fiecare intrare RAS sau adresa are o capacitate proprie de ordinul a 5pF. Deci capacitatea vazuta de automatul de control este: $4 \times 8 \times 5 \text{pF} = 160 \text{pF}$!, deci ar deteriora fronturile. Din acest motiv este necesara comandarea circuitelor prin buffere.



OBS.

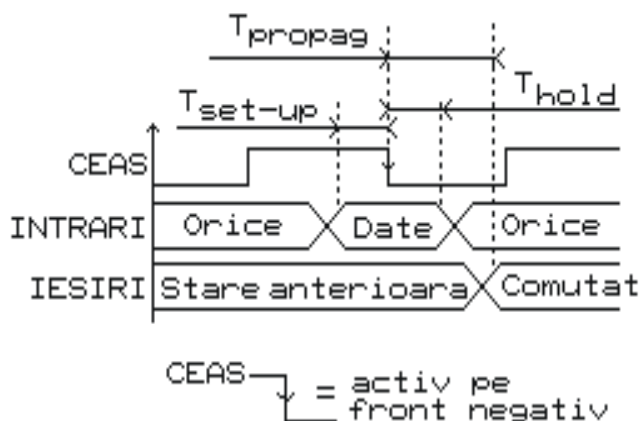
Nu se admite utilizarea intirzierilor prin porti sau a monostabilelor pentru comanda SEL sau RAS deoarece nu sint stabile in timp si nu sint reproductibile (fiecare poarta are un timp de propagare intre 0 si T_{pMAX} , care depinde puternic de temperatura; nu se defineste un timp minim de propagare!; similar constanta de timp a monostabililor este instabila).

4.6. BISTABILI CU COMUTARE PE FRONT. PRINCIPIUL **MASTER-SLAVE**

Bistabilii cu comutare pe front schimba starea iesirilor in momentul aparitiei frontului unui semnal numit **CEAS**, in functie de starea intrarilor la acelasi front al ceasului (fig.4.6.1).

Aceasta impune ca datele sa fie stabile doar in zona frontului (putin inainte si putin dupa), in comparatie cu bistabilii cu comutare pe palier, care impun ca datele sa fie stabile pe toata perioada palierului.

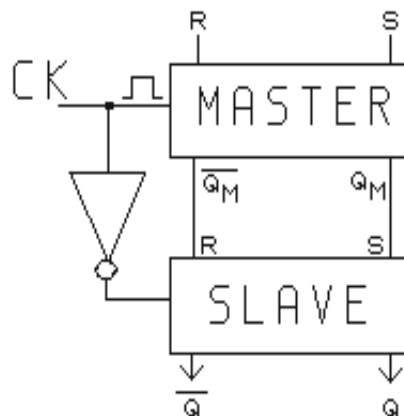
In functie de circuit, comutarea se poate produce pe frontul pozitiv sau negativ al ceasului, acesta se numeste front activ.



La structurile PIPE-LINE care folosesc circuite cu comutare pe palier este necesar ca latch-urile consecutive sa fie comandate cu ceas in antifaza.

O structura similara este folosita si la bistabilii cu comutare pe front.

In fig.4.6.2. este prezentat un bistabil RS cu comutare pe front. Acesta este alcatuit din doua latch-uri conectate in cascada, adica unul dupa altul, dar comandate de ceas in antifaza.



Cind semnalul CK (CEAS, Clock) este in "1" primul latch, numit **MASTER** (stapin), va comuta in functie de semnalele R si S. Al doilea latch, numit **SLAVE** (sclav) este inchis, deci la iesire nu se vad modificarile din MASTER. Cind semnalul CK trece in "0" latch-ul MASTER se inchide si va pastra starea impusa de intrari la momentul inchiderii, deci la momentul frontului negativ al ceasului. Aceasta stare va fi prezenta si la iesiri deoarece latchul SLAVE este acum deschis (transparent). La trecerea in "1" a ceasului SLAVE se inchide (nu modifica iesirea indiferent de intrari), iar MASTER se deschide, deci ciclul se repeta.

Deci la frontul negativ al ceasului, la iesire apare starea impusa de situatia intrarilor RS din acel moment, indiferent de comutarile anterioare. Starea iesirilor se mentine pina la un nou front activ (cel negativ) al CK. Acesta este un circuit de tip MASTER-SLAVE

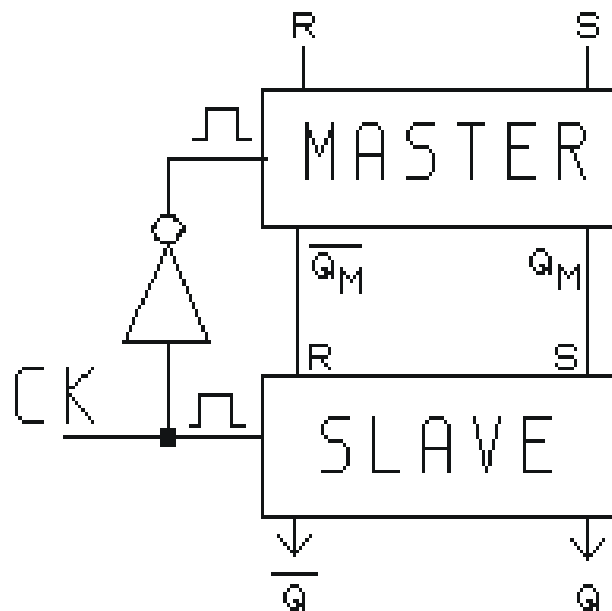
(sclavul copiaza starea stapinului).

In fig.4.6.3 este prezentata o varianta care lucreaza pe front pozitiv.

ATENTIE !!

Pentru ca datele sa fie corect preluate de circuitele cu comutare pe front este necesar ca aceste date sa fie stabile cu un interval de timp inaintea frontului activ al ceasului, timp dat in catalog sub denumirea de Tset-up.

Mai trebuie ca datele sa fie stabile si dupa front un timp numit timp de mentinere, dat in cataloage ca Thold.

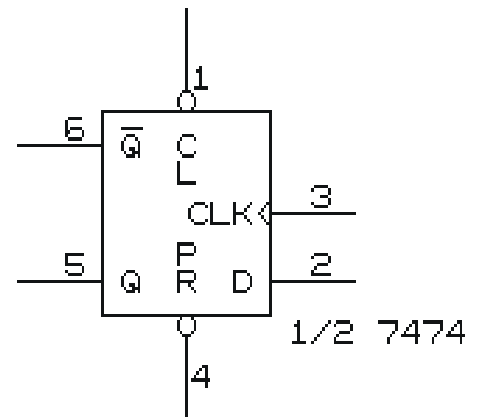
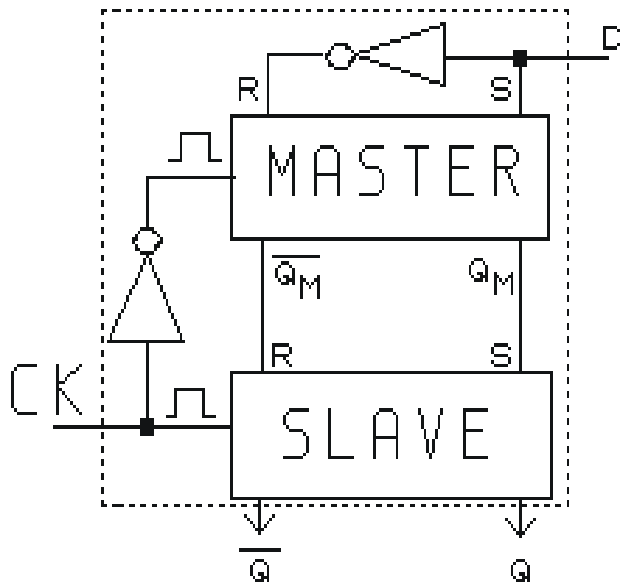


4.7. BISTABILUL DE TIP 'D'

Un mod de realizare a acestui bistabil este prezentat in fig.4.7.1. El se obtine dintr-un bistabil RS (prezentat in subcapitolul anterior) prin introducerea unui inversor. Daca 'D'=0 atunci 'S'=0 si 'R'=1, deci iesirea 'Q' va trece in '0' pe frontul pozitiv al ceasului. Daca 'D'=1 atunci 'S'=1(activ) si 'R'=0 (inactiv), deci iesirea 'Q' va trece in '1'. Deci 'Q' copiaza starea intrarii 'D' la frontul activ al ceasului.

Bistabilii de tip D cu comutare pe front exista realizati ca circuite integrate. In aceasta structura ele mai pot avea doua intrari pe palier R' si S' (active pe "0") care trec in 0 respectiv 1 iesirea, fiind prioritare fata de "D" (cind una din ele este activa "D" nu mai are efect). Intrarea CEAS poate fi activa pe front pozitiv si negativ.

Simbolul logic este prezentat in fig.4.7.2.



TABELUL DE TRANZITII PENTRU CIRCUIITUL TIP 'D':

D	Q	\bar{Q}
0	0	1
1	1	0

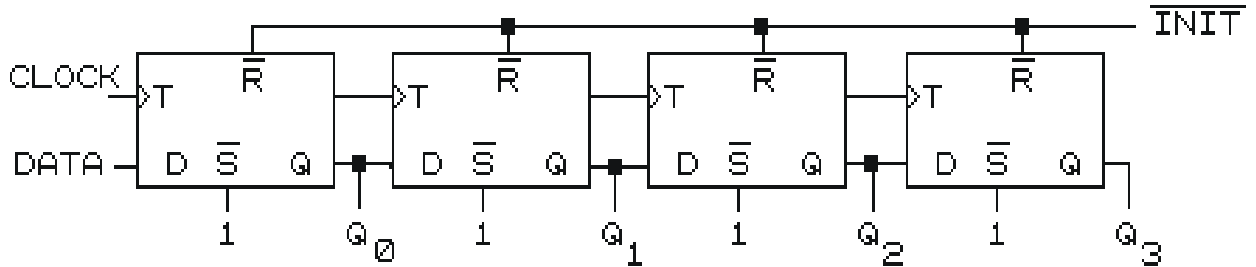
OBS.: In tabelul de tranzitii partea stinga reprezinta starea intrarii la momentul t (inaintea frontului activ al ceasului), iar partea dreapta semnifica starea iesirilor la momentul t+1 (dupa frontul activ, pina la urmatorul front activ, cind va fi incarcata o noua data de la intrarea "D").

4.8. EXTENSII ALE BISTABILILOR CU COMUTARE PE FRONT-REGISTRE

Un numar arbitrar de bistabili cu comutare pe front conectati impreuna intr-un circuit integrat formeaza un REGISTRU.

CLASIFICARE:

- 1-REGISTRE DE DEPLASARE SERIE
 - 2-REGISTRE CU MEMORARE (INCARCARE PARALEL)
 - 3-REGISTRE CU FUNCTIONARE MIXTA (SERIE+PARALEL)
- 1- REGISTRE DE DEPLASARE SERIE** sint bistabili tip 'D'cu ceas comun



care au intrarea 'D' conectata la iesirea bistabilului precedent, ca in figura 4.8.1. Deci la fiecare front activ datele se deplaseaza dintr-un bistabil in urmatorul. In primul bistabil se incarca la fiecare front date din exterior. Rezulta astfel o deplasare spre dreapta a datelor. Datele din ultimul bistabil se pierd. Unele circuite integrate cu functia de 'registru de deplasare' de capacitate mare (numar mare de bistabili legati in 'serie') nu au disponibile la pini (picioruse) iesirile tuturor bistabililor datorita numarului limitat de pini, dar sigur ultimul bit este conectat.

TABELUL DE TRANZITII daca la intrare se aplica succesiv: 1;0;0;0;1;0;1;1;1;0.

D	Q0	Q1	Q2	Q3
1	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
1	1	0	0	0
0	0	1	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1
0	0	1	1	1

Exista circuite care pot face si deplasare spre stanga si spre dreapta, numindu-se REGISTRE DE DEPLASARE SERIE BIDIRECTIONALE.

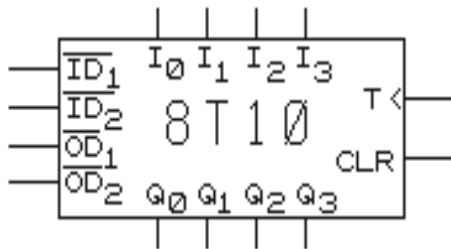
Se observa ca data plasata la intrare (D) este preluata de primul bistabil (Q0) la primul front activ al ceasului. La fiecare front activ ea este transferata spre dreapta cu o pozitie. Se mai poate vedea din tabel ca registrul de deplasare a convertit sirul serial de 4 biti(1000) aplicat la intrare in sir paralel (la care avem acces simultan la toti bitii) de 4 biti dupa 4 cicluri de tact=lungimea registrului din exemplu. Deci registrul de deplasare poate face o conversie a unui sir de date, cu lungime egala cu a registrului, din forma seriala in forma paralela.

2-REGISTRE DE MEMORARE (REGISTRE CU INCARCARE PARALEL)

Sint circuite formate din bistabili ce au comun doar semnalul de de ceas, dar au intrari si iesiri independente.

Aceste circuite mai pot avea si o intrare de stergere CLR comuna tuturor bistabililor constitutivi. Daca iesirile sint Three-state, atunci mai exista o intrare comuna ce trece iesirile in starea de mare impedanta simultan.

Exemplu: Registrul 8T10 de la firma Signetics este de tip paralel de 4 biti.



I0;I1;I2;I3=intrari
Q0;Q1;Q2;Q3=iesiri
ID=ID1+ID2=Input disable
CLR=CLear
T=ceas activ pe front pozitiv
OD=OD1+OD2=Output Disable

TABELUL DE TRANZITII:

I0...3	$\overline{ID} = \overline{ID1} + \overline{ID2}$	$\overline{OD} = \overline{OD1} + \overline{OD2}$		Q0...3
0	0	0		0 memorare
1	0	0		1 memorare
X	1	0		Q anterior (t)
X	X	1		HiZ

Trecerea in starea de inalta impedanta (HiZ) a iesirilor este asincrona si nu inhiba functionarea interna a registrului. Acesta poate memora noi date chiar daca nu apar la iesire.

APLICATII ALE REGISTELOR DE MEMORARE:

RALU=Registre+ALU

RALU se realizeaza folosind registre de memorare pentru unul din operanzi. Rezultatul operatiei este depus tot intr-un astfel de registru si poate constitui operand pentru urmatoarea operatie. Selectia registrului destinatie (unde va fi stocat rezultatul) este facuta de un multiplexor iar registrul sursa este selectat de alt multiplexor.

DMUX-S selecteaza drept sursa pentru operandul A al ALU unul din registrele de memorare 0..3. Operandul A este prelucrat de ALU cu operandul B venit din exterior, conform indicatiilor date de F (Function) si A/L (Arithmetical/Logical). DMUX-D specifica registrul destinatie al rezultatului operatiei efectuate.

La programare se va specifica registrul sursa, registrul destinatie,

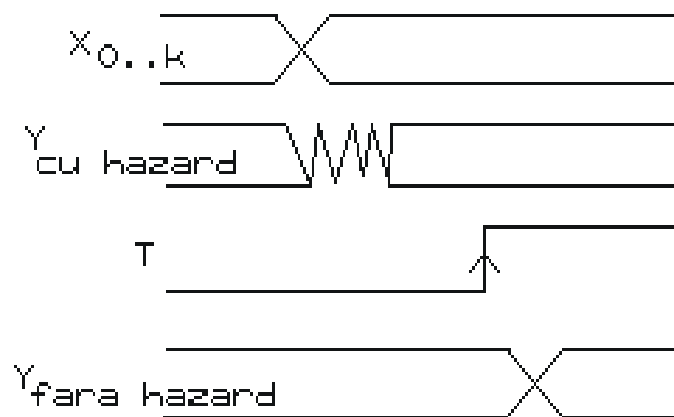
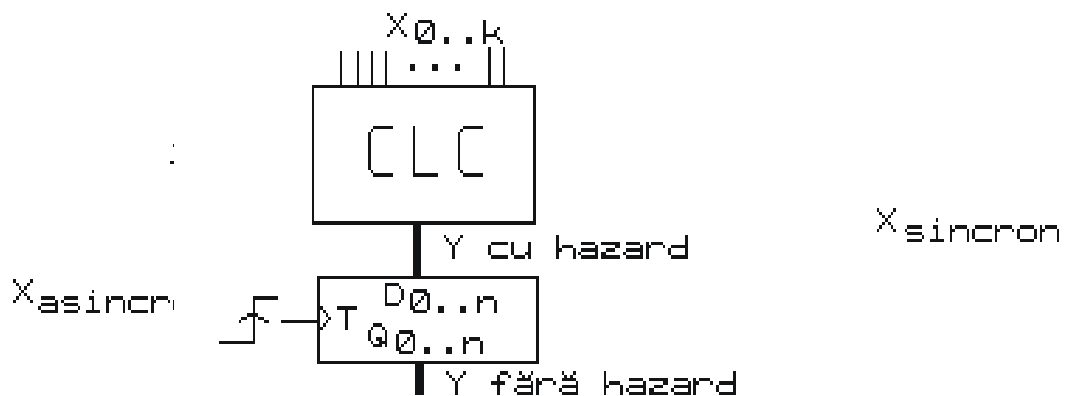
functia, tipul operatiei(A/L), operandul B:

SURSA | DESTINATIE | F | A/L | OPERAND B

ELIMINAREA HAZARDULUI COMBINATIONAL

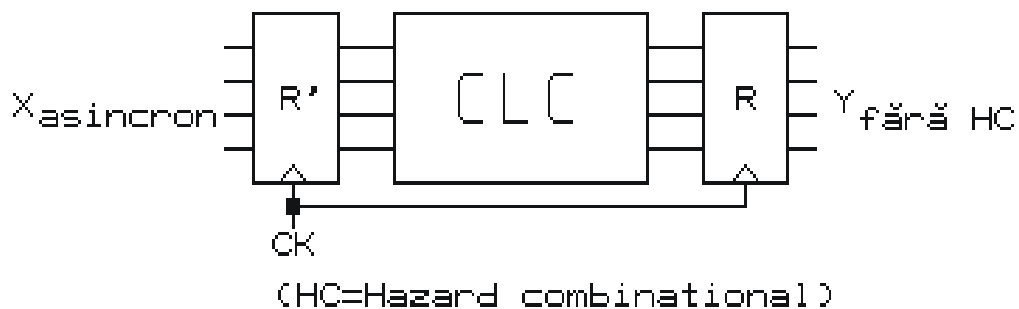
-Presupunem ca toate intrarile in CLC (FIG.4.8.3.) se modifica simultan la momentul t0. Datorita timpilor de propagare diferiti, la iesire apare hazard combinational. Acesta poate fi redus foarte mult daca la iesirea CLC se cupleaza un registru de memorare. Dupa ce datele s-au stabilizat la intrarea registrului, se aplica un front activ la intrarea de ceas si datele sint preluate de registru. Datele apar la iesire aproape simultan, diferentele (hazardul) se reduc la un timp de propagare "de la CLOCK la IESIRE " al registrului de memorare.

-A doua situatie: intrarile sint asincrone (vin de la alt sistem).



Este necesara sincronizarea intrarilor ! Se realizeaza cu o schema cu doi bistabili.

-Schema din fig.4.8.3. se completeaza cu un registru R' pentru sincronizarea variabilelor de intrare si se obtine schema din fig.4.8.5.



SCHEME PIPE-LINE (PENTRU MARIREA FRECVENTEI DE LUCRU A SISTEMULUI)

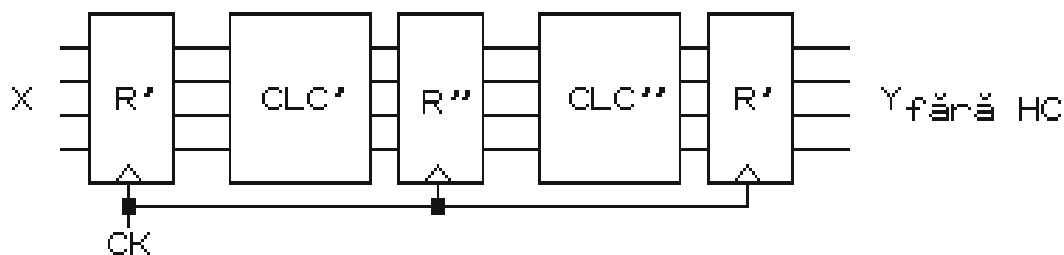
La schema anterioara cantitatea de informatie ce poate fi prelucrata in unitatea de timp este limitata de timpul de propagare prin CLC. Intervalul de timp necesar pentru prelucrarea unui singur cuvint de date este egal cu perioada ceasului: $T_{ck} = T_{pR'} + T_{pCLC} + T_{suR}$ unde:

T_{ck} =perioada ceasului

$T_{pR'}$ =Tpropagare prin registru (de la frontul activ al CK la iesire).

T_{pCLC} =Tpropagare prin CLC

T_{suR} =Tset-up registru



b.Schema obtinuta prin fragmentarea CLC in nivele cu timpi de propagare apropiati. $T_{pCLC'} \approx T_{pCLC} \approx T_{pCLC''}$ (din

fig.4.8.5.)/2

$T_{ck} = \max(T_{pR}' + T_{pCLC}' + T_{suR}'', T_{pR}'' + T_{pCLC}'' + T_{pR}) =$
=aproximativ $\max(T_{pCLC}', T_{pCLC}'')$ deoarece T_{pCLC}' este mult mai mare decit T_{pR}', T_{suR}' in cele mai multe cazuri.

La schemele PIPE-LINE putem mari frecventa ceasului de un numar de ori egal cu numarul de tronsoane in care am spart CLC initial.

Circuitele PIPE-LINE cresc viteza de prelucrare a sistemului prin lucrul in paralel al tronsoanelor (pe durata propagarii prin al doilea tronson, date noi se propaga prin primul tronson etc.) ceea ce face ca sistemul sa primeasca noi date de prelucrat mult mai des (la un interval egal cu noul T_{ck} care este, in exemplul nostru, aprox. 1/2 din T_{pCLC} al schemei care nu este PIPE-LINE). Evident, durata de la introducerea unei date pina cind ea apare la iesire este cam aceeaasi cu cea a schemei fara structura PIPE-LINE (chiar mai mare).

Deci schemele PIPE-LINE prelucreaza mai multe date in acelasi interval de timp, dar nu reduc durata de prelucrare a unei date !

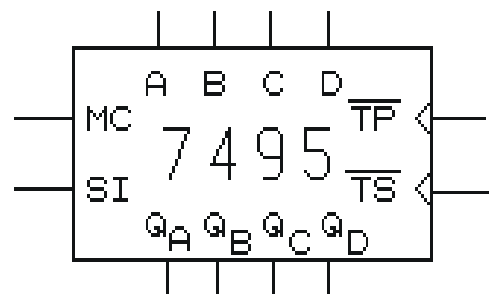
3-REGISTRE CU FUNCTIONARE MIXTA

Acestea sint registre ce pot face si memorare (functionare paralel) si deplasare (functionare serie). Exeplu circuitul 7495. Acest circuit este de 4 biti (are 4 bistabili) si are o intrare serie 'SI', 4 intrari paralel: A; B; C; D, 4 iesiri: Q0; Q1; Q2; Q3, MC care selecteaza unul din modurile: memorare/deplasare dreapta, TP=ceas in modul paralel, TS=ceas in modul serial.

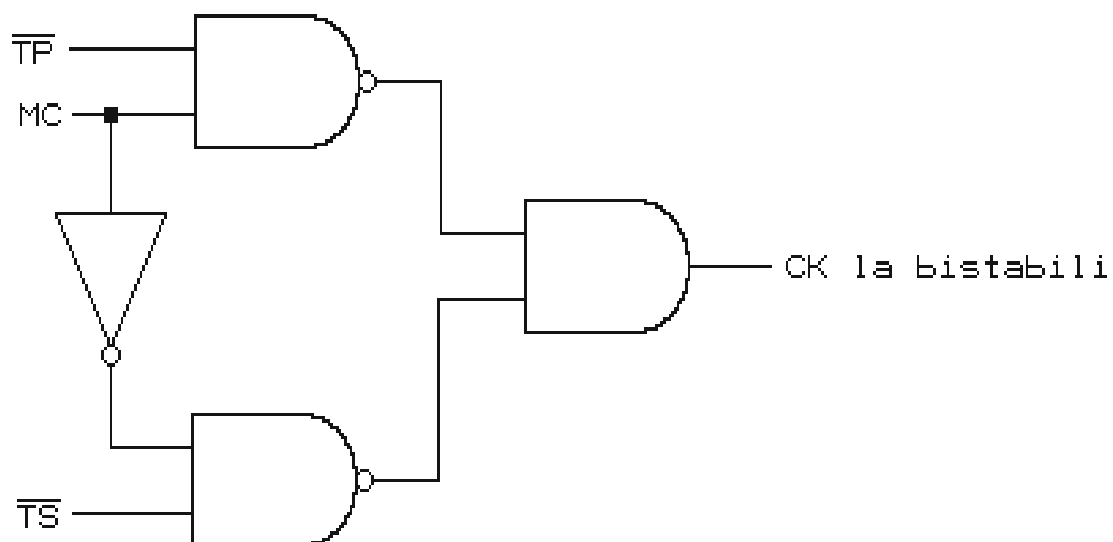
Functionarea circuitului 7495:

Cind **MC=1**: circuitul lucreaza in modul paralel cu ceasul TP activ pe front negativ (TS nu are nici o influenta).

Cind **MC=0**: circuitul lucreaza in modul serie cu ceasul TS activ pe front negativ (TP nu are nici o influenta). Datele se preiau de la intrarea SI.

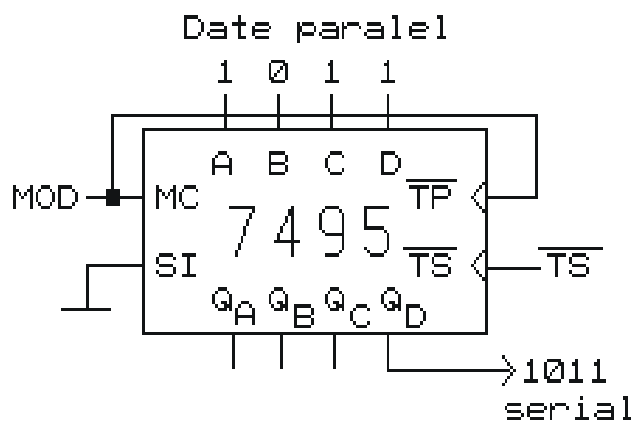


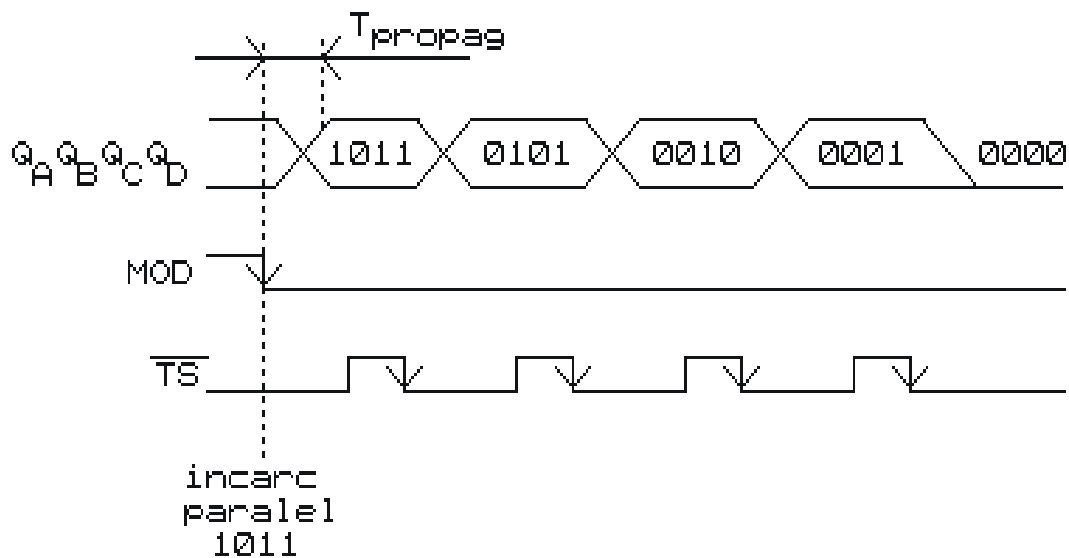
Subtilitati de utilizare, folosind schema interna a circuitului:



Daca $\overline{TP}=1$, $\overline{TS}=0$ si comut MC din 1 in 0 apare puls de ceas. In situatia inversa iar apare puls de ceas la bistabili.

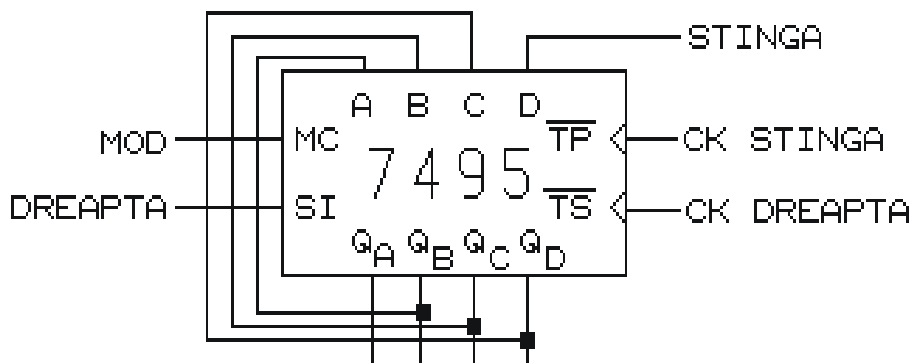
Folosind aceste caracteristici se poate obtine o schema de conversie PARALEL-SERIE a datelor (fig. 4.8.9.).





Circuitul poate fi conectat astfel incit sa realizeze deplasare bidirectionala, prin legarea iesirii Qj la intrarea bistabilului j-1, iar intrarea bistabilului 3 (D) constituie intrare pentru deplasarea stinga.

Alt exemplu este 74194 care poate incarca paralel, poate deplasa stinga si dreapta.

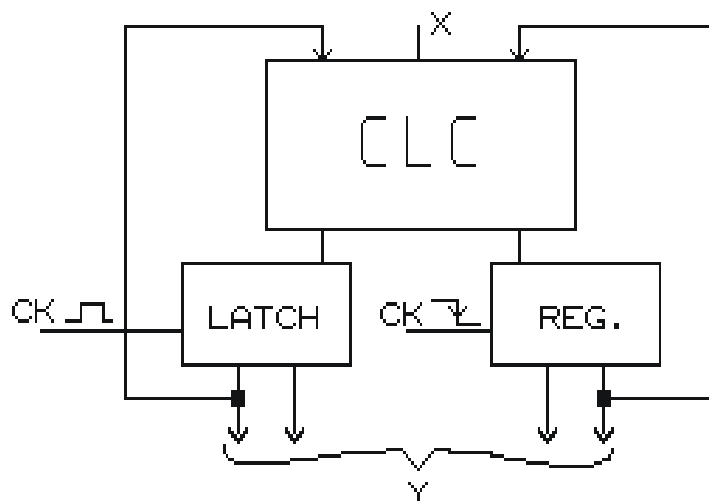


CAP.5 SISTEME DE ORDINUL II (SISTEME SECVENTIALE)

Se obtin din sisteme de ordin inferior din care cel putin unul este de ordinul 1 (de memorare) prin conectare in reactie. Trebuie ca in bucla de reactie sa existe o prelucrare efectiva.

La sisteme logice de ordin 2 bucla de reactie este controlata de ceas, functionarea se face in secvente (functionare secventiala).

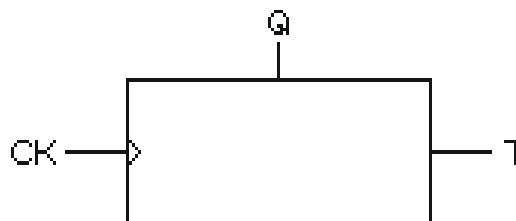
La aceste sisteme iesirile depind atat de intrari cit si de starea anterioara a sistemului (de istoria lui).



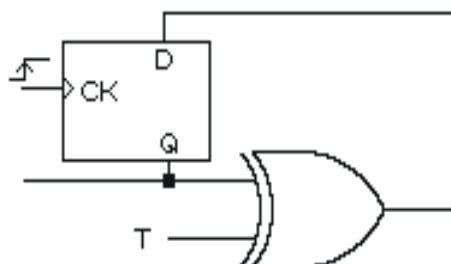
5.1. BISTABILUL T

Daca $T=0$ nu se modifica starea iesirii.

Daca $T=1$ la fiecare impuls de ceas iesirea Q se inverseaza.



T comanda transcrierea lui Q; CLC este XOR. Circuitul de ordinul 1 este bistabilul D.



5.2. BISTABILUL J-K "MASTER-SLAVE" (JKMS)

Acesta este cu comutare pe front si adauga structurii MASTER-SLAVE reactia si porti de conditionare pe intrari.

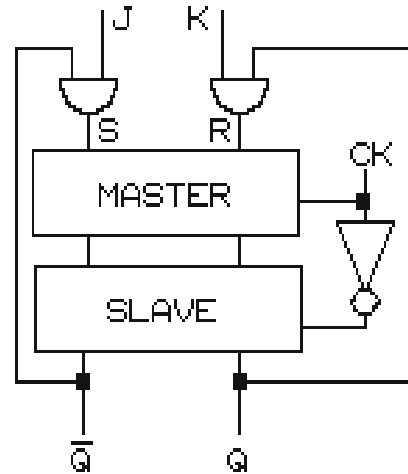
FUNCTIONARE

TABELUL DE TRANZITII

Daca $J=K=0$, atunci nu se va modifica starea iesirii.

Daca $J=0;K=1$, atunci, dupa frontul activ, Q trece in 0

Daca $J=1;K=0$, atunci, dupa frontul activ al ceasului, iesirea se inverseaza (daca a fost 0 trece in 1, daca a fost 1 trece in 0.



J	K	Q^+
0	0	Q (la t)
0	1	0
1	0	1
1	1	\bar{Q} (la t)

mom. t			t+1
Q	J	K	Q+
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Prin compactare rezulta:

Q	J	K	Q+
0	0	X	0
0	1	X	1
1	X	1	0
1	X	0	1

Compactam acest tabel si rezulta:

Q	J	K	Q+
0	A	X	A
1	X	\bar{A}	A

O forma interesanta si mai simplu de retinut este:

Q-->Q+	J	K
0--> A	A	X
0--> A	X	\bar{A}

Acest tabel indica valorile necesare la intrarile J si K pentru o anumita tranzitie (functie de starea anterioara).

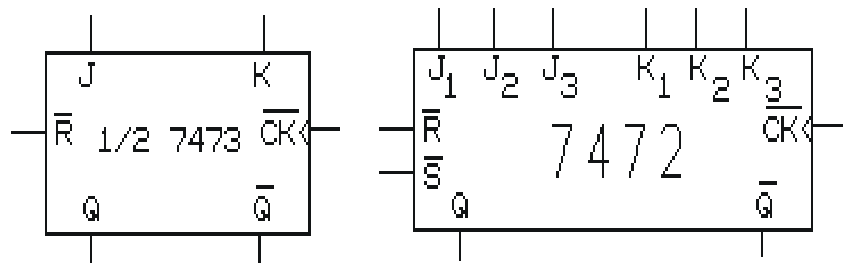
Bistabili JK integrati:

CIRCUITUL 7473 (2 bistabili J-K intr-o capsula) comuta pe front negativ, are intrare de RESET activain 0, are o intrare J si una K. Nu are intrare de SET.

Circuitul 7472 (un singur bistabil intr-o capsula) are cite trei intrari J si trei intrari K, o intrare RESET activa in 0 si o intrare SET activa tot in 0.

Notind: $J=J_1*J_2*J_3$

si $K=K_1*K_2*K_3$, functionarea este identica cu a bistabilului J-K.



5.3. NUMARATOARE

Numaratoarele sint circuite la care iesirile depind de valoarea anterioara printr-o regula de incrementare sau decrementare intr-un cod specific circuitului. Codul in care numara este un sir ordonat de numere de 'k' biti.

A incrementa=a face ca iesirea sa treaca in starea urmatoare din sir.
A decremanta=a trece iesirea in starea anterioara din sir.

Sirul de numere este specific fiecarui cod. Astfel pot fi: Numaratoare in cod binar natural, numaratoare zecimale, numaratoare in cod GRAY etc.

La incrementare, cind numaratorul a ajuns la ultimul numar din sir, urmatoarea stare va fi primul numar din sir (si ciclul se repeta). Exista numaratoare care doar incrementeaza (de ex. 7493) si altele care pot si decremanta (de ex. 74193). Exista circuite care pot prelua din exterior starea de pornire la fel ca registrele de memorare sau latchurile.

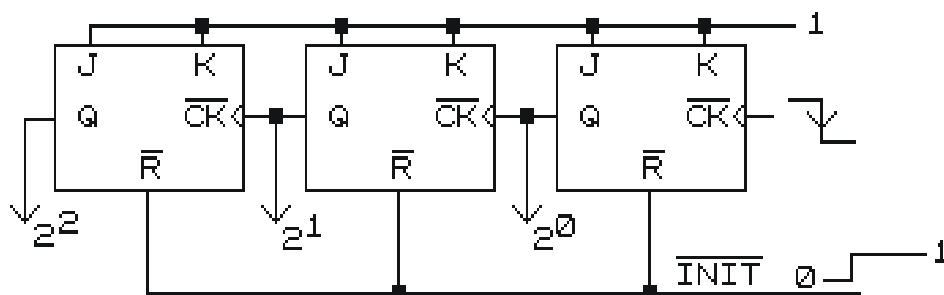
Numaratoarele au intrare de RESET (aducere la zero). Mai pot avea evident si alte intrari.

Clasificare dupa modul de lucru al bistabililor care le compun: **ASINCRONE** si **SINCRONE**.

5.3.1. NUMARATOARE ASINCRONE

Sint formate din bistabili T conectati in cascada. Primul bistabil are intrarea CK legata la un pin unde se vor aplica impulsurile ce vor fi numarate. Restul bistabililor au CK legat la Q al bistabilului anterior ca in figura 5.3.1.1.

In fig.5.3.1.2. apar formele de unda asociate.



Dezavantajul acestor numaratoare este viteza redusa de lucru, deoarece este necesar un timp de propagare prin toti bistabilii.

La numaratoarele asincrone apare un timp de propagare mare:

Semnalul de intrare comuta primul bistabil dupa un timp egal cu timpul de propagare de la CK la iesire, notat T_{pCK-Q0} (pentru bistabilul0). Dupa alt T_{pCK-Q1} va comuta si bistabilul1, dupa alt T_{pCK-Q2} bistabilul2 etc.

Mai mult, pe durata propagarii numaratorul prezinta la iesiri alte numere decit cele succesive. Ex.: La trecerea din 1 1 1 1 in 0 0 0 0 apar la iesiri numerele: 1 1 1 1; 1 1 1 0; 1 1 0 0; 1 0 0 0; 0 0 0 0. Se observa propagarea de la bistabilul 0 la 3. (fig. 5.3.1.3.) Acest fenomen se numeste HAZARD SECVENTIAL.

Cel mai mare timp de propagare apare cind comuta toti bistabilii (cind se atinge capacitatea maxima de numarare). Acesta este cel care va impune durata hazardului.

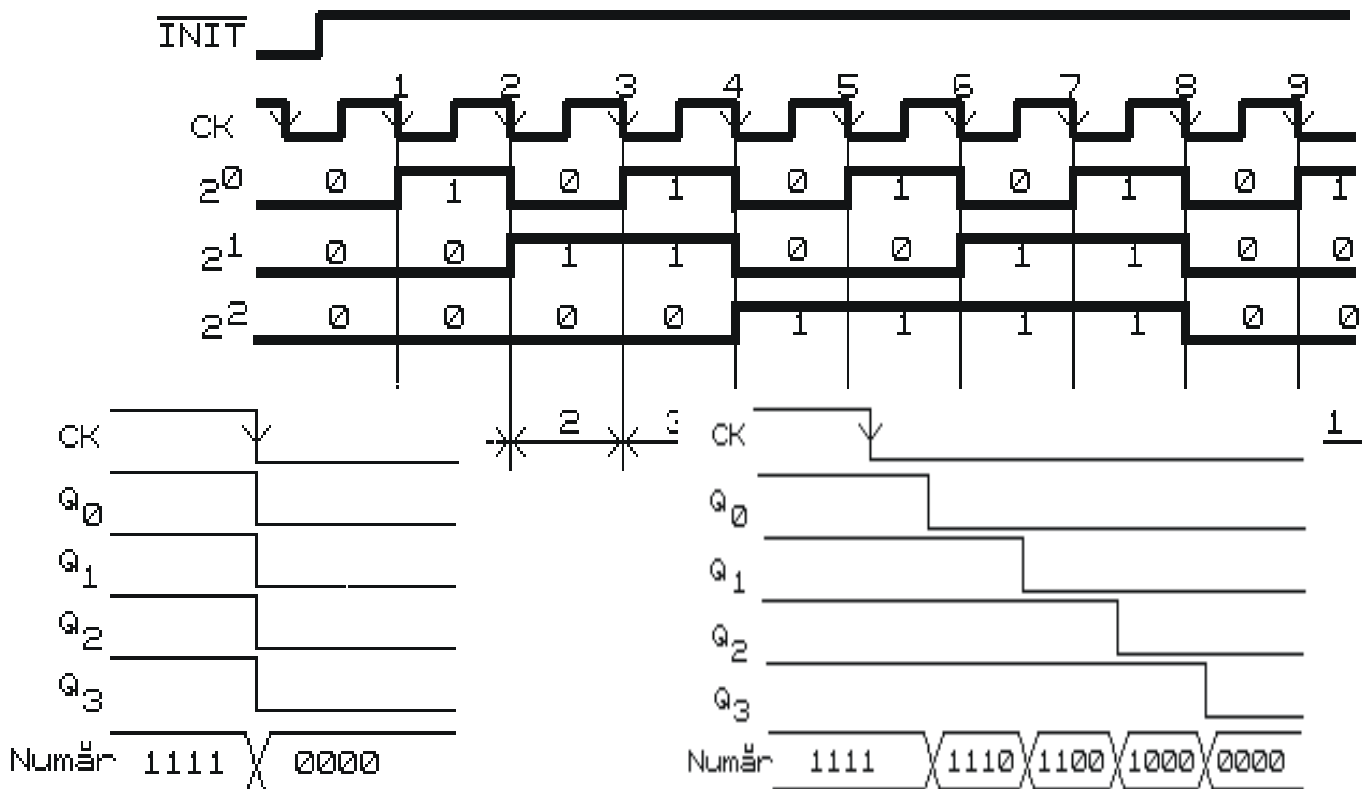


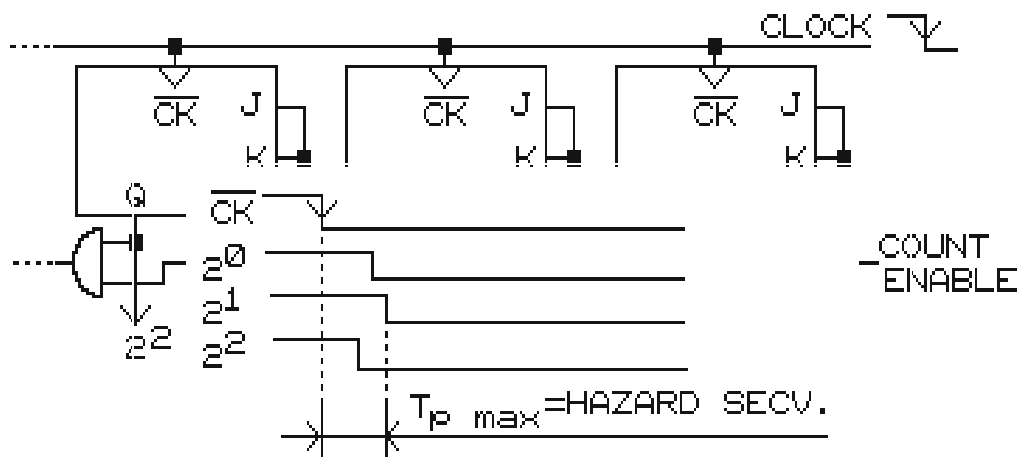
Fig. 5.3.1.3. HAZARDUL SECVENTIAL.

5.3.2. NUMARATOARE SINCRONE

Hazardul secvential poate fi redus foarte mult folosind numaratoare sincrone. Acestea sint circuite formate din bistabili cu ceas comun si au, uneori o intrare de validare a incrementarii. Cind este activa, la fiecare impuls de ceas numatorul incrementeaza iesirea (numara impulsurile de ceas).

Hazardul se reduce la timpul de propagare printr-o singura celula, deoarece bistabilii nu au timpi de propagare identici, chiar daca au ceas comun (fig.5.3.2.1.; fig.5.3.2.2.). Durata hazardului este independenta de capacitatea numatorului, in comparatie cu cele asincrone unde creste proportional cu aceasta. La schema din fig.5.3.2.1. perioada ceasului trebuie sa fie mai mare ca durata de propagare prin toate portile AND, deci creste proportional cu capacitatea de numarare. Daca s-ar folosi porti AND cu peste 2 intrari, reducind astfel adincimea CLC-ului realizat, se poate creste frecventa ceasului. Adincimea se poate reduce pina la un singur nivel.

In acest caz se obtine viteza maxima: $T_{ck} = T_p \text{ bistab} + T_p \text{ AND} + T_{set-up}$.



NUMARATOARE SINCRONE INTEGRATE

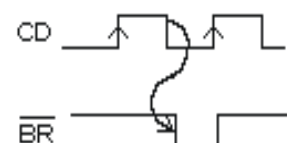
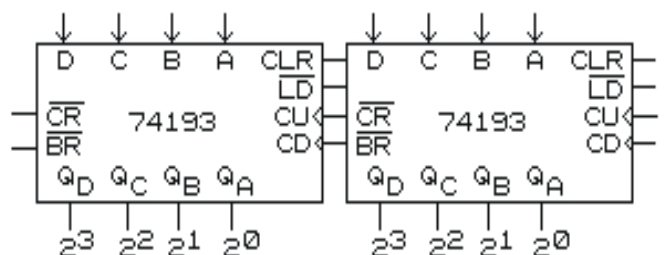
(Au si functia de registru=asigura pornirea de la un numar fixat; pot avea intrare de RESET)

- Pot fi: -cu incarcare pe palier si numarare pe front. ex.:74193
- cu incarcare si numarare pe frontul ceasului. ex.:74169

CIRCUITUL 74193 -numarator binar reversibil
de 4 biti (numara intre 0..15)

- CLR - CLEAR =aducere la 0;activ pe palier 1 logic. Este 'cel mai' prioritar.
- LD' - LOAD =incarca numaratorul cu un numar prestabilit la intrarile A B C D. Este mai putin prioritar ca CLR (nu incarca daca CLR=1) si este mai prioritar ca numararea. Este activ pe palierul 0.
- CU - Count Up (numara inainte)=ceas activ pe front pozitiv; incrementeaza numaratorul.
- CD - Count Down (numara in jos;decrementeaza)=ceas activ pe front pozitiv care reduce cu 1 continutul numaratorului la fiecare front pozitiv.
- CR' - Carry (transport la numararea in sus)=cind numaratorul trece din 1 1 1 1 in 0 0 0 0 apare un puls ca in figura 5.3.2.3. (unde este reprezentat pulsul pentru iesirea BO).
- BR' - BoRrow (imprumut la numararea in jos) semnaleaza similar trecerea din 0 0 0 0 in 1 1 1 1.

Daca extindem numararea pe mai multe CIP-uri ca in fig.5.3.2.3., numaratorul nu mai este sincron. Mai mult, la borrow impulsul are durata unei semiperioade a ceasului si se strica (se reduce) prin propagarea de la un CIP la altul. Deci este necesar ca perioada ceasului sa fie mai mare. Se poate realiza si o expandare pentru viteza mare folosind porti AND.



CIRCUITUL 74169 este numarator sincron si la numarare si la LOAD.

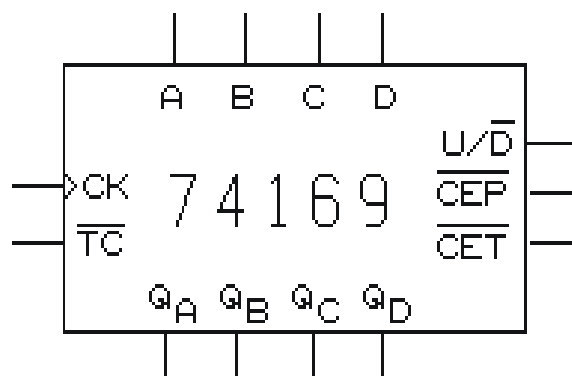
\overline{LD} =LOAD=incarcare sincrona (activ pe 0).

U/\overline{D} =Up/ \overline{Down} =daca este 1, atunci va numara in sus la fiecare front al ceasului; daca este 0 va numara in jos la fiecare front activ al ceasului.

Nu are CLEAR.

\overline{CEP} ; \overline{CET} =Intrari ce valideaza numararea, daca ambele sint 0.

\overline{TC} =Terminal Count=transport la CIP-ul urmator. Este activ (0) daca iesirile sint 1111 si \overline{CET} este activa.



AUTOMATE FINITE ELEMENTARE

5.4.1. Definitie matematica si modele ale automatelor finite

1. Definitia prin cvintuplu:

Este un ansamblu de multimi si functii:

$$A = \{X, Y, Q, f, g\} ,$$

unde

1.1. X este multimea de intrare (multimea vectorilor de intrare)

$X = \{x_0, x_1, \dots, x_i, \dots, x_n\}$, x_i , $i=0..n$ -sunt vectori (configuratii) de intrare
(combinatii ale variabililor de intrare)

$x_i = x_t x_{t-1} \dots x_p \dots x_1 x_0$ (o configuratie ce se poate aplica pe intrari)

$$x_p = \{1, 0\} \quad p=0..t$$

X este o multime completa: pentru un numar $t+1$ de intrari, are 2^{t+1} elemente.

1.2 $Y = \{y_0, y_1, \dots, y_k, \dots, y_m\}$, $k=0..m$ este multimea configuratiilor (vectorilor) de iesire

$y_k = y_u y_{u-1} \dots y_r \dots y_0$ -este o configuratie de iesire

$y_r = \{0, 1\}$ $y_u..y_0$ -sunt variabile binare de iesire
 $r=0..u$

Y nu este o multime completa: nr. combinatiilor de iesire, $m+1$ este mai mic decat cel posibil de realizat cu $u+1$ variabile:
 $m+1 <= 2^{u+1}$

1.3 $Q = \{q_0, q_1, \dots, q_j, \dots, q_1\}$ -multimea de stare (multimea starilor, multimea configuratiilor de stari) (vectorul de stare)

$q_j = q_v q_{v-1} \dots q_s \dots q_0$ -stari (vectori de stare)
 $j=0..1$

$q_v..q_0$ -variabile de stare $q_s = \{0, 1\}$

Multimea de sta"re este o multime completa, $l+1 = 2^{v+1}$

Definitia 1: Evolutia automatului este un sir de stari inlantuite:

$$q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow \dots q_k \dots$$

Definitia 2: Starea initiala:este starea de inceput a evolutiei automatului (q_0)

Definitia 3: Multimea starilor initiale e o submultime a lui Q care contine totalitatea starilor din care poate porni sistemul.

Clasificarea dupa nr. de elemente ale Q_i

- automate neinitiale (fara stari initiale)
- automate cu o stare initiala
- automate cu mai multe stari initiale

Definitia 4: Stari accesibile: sint stari apartinand unei evolutii posibile;

Stari inaccesibile: (stari de blocare) nu apartin unei evolutii.

Definitia 5: Stari echivalente: sint stari care genereaza siruri de iesiri identice: $q_i \sim q_j$.

1.4. Functia f : este functia de tranzitie a starilor:

$$f: X \times Q^t \rightarrow Q^{t+1}$$

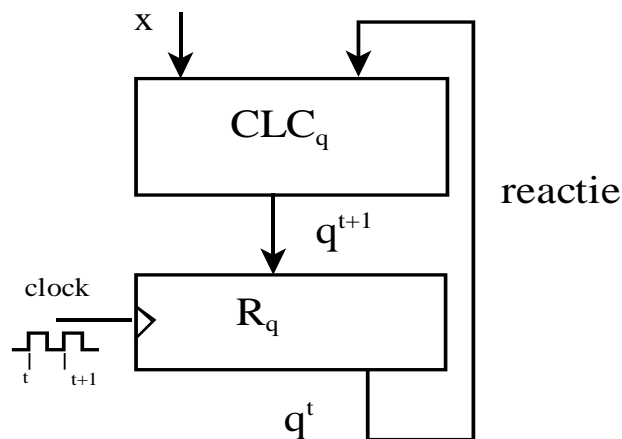
Starea sistemului depinde de starea anterioara si de intrari.

Semiautomatul: $SA = \{X, Q, f\}$:descrie evolutia sistemului in spatiul starilor:

CLC_q-realizeaza prelucrarea (functia f)

R_q- memoreaza data pe unul din fronturile ceasului.

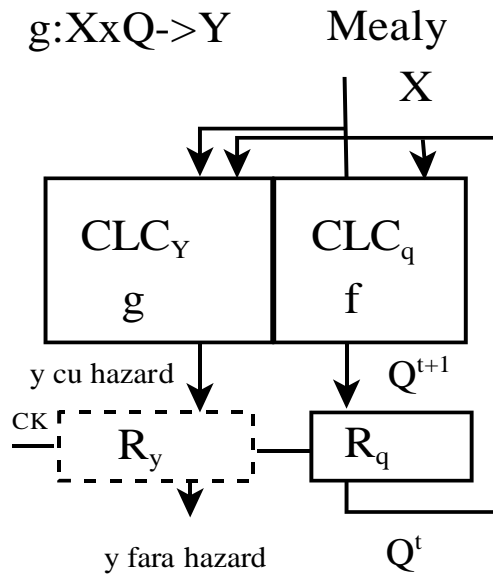
Multimea $T = \{0, 1, 2, \dots, N\}$ (multimea momentelor), marcheaza impulsurile de tact. Apare in evolutia multimii Q



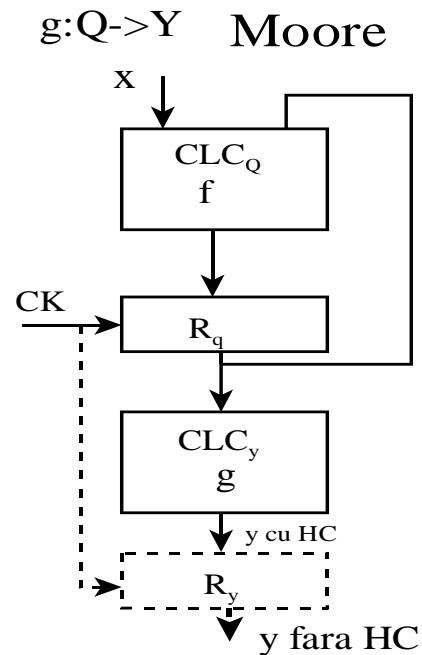
Fig

1.5 **g- este functia de tranzitie a iesirilor.**

Unui semiautomat ii pot corespunde mai multe automate, dependent de modalitatea de definire a functiei g:



Fig



Fig

Functia g se realizeaza tot cu un CLC - CLC y

Tipuri de automate	Intirziera iesirilor fata de intrari
Mealy imediat	0 tacturi
Mealy cu intirziere (cu R _y)	1 tact
Moore imediat	1 tact
Moore cu intarziere (cu R _y)	2 tacturi

Automate echivalente (cu aceeasi intirziere)

-Mealy cu intirziere si Moore imediat (1 tact)

2). Definirea prin TRIPLET:

$A=\{X,Y,M\}$, unde M este functia de transfer: $M:X \rightarrow Y$
 Aceasta definire este accesibila experimental; este incompleta.
 Se fixeaza intrarile si se emit impulsuri de tact. Se noteaza evolutia iesirilor.

Se poate reprojeta continutul circuitului, dar nu avem o solutie unica.

Modalitati de descriere a functiilor f si g

1. Descriere prin tabelul de tranzitii

Semiautomat					Mealy			
Q+X	x0	x1	...	xn	x0	x1	...	xn
q0	q3	q7	...	q20	y10	y0	...	y15
...
ql

Moore	
Q	Y
q0	y12
q1	y4
...	...
q10	y30

2. Descrierea prin graf: (fig. 4, fig. 5):

NODURI	STARI (Mealy) STARI/IESIRI (Moore)
LATURI	Orientate dupa evolutia in timp (sageata)
PE LATURI	IN/OUT (MEALY)
	IN (MOORE)

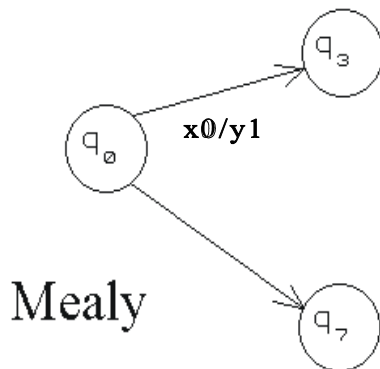


fig. 4

Fig

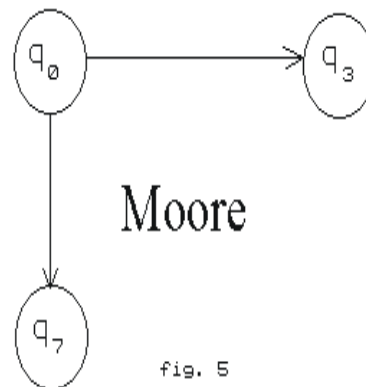
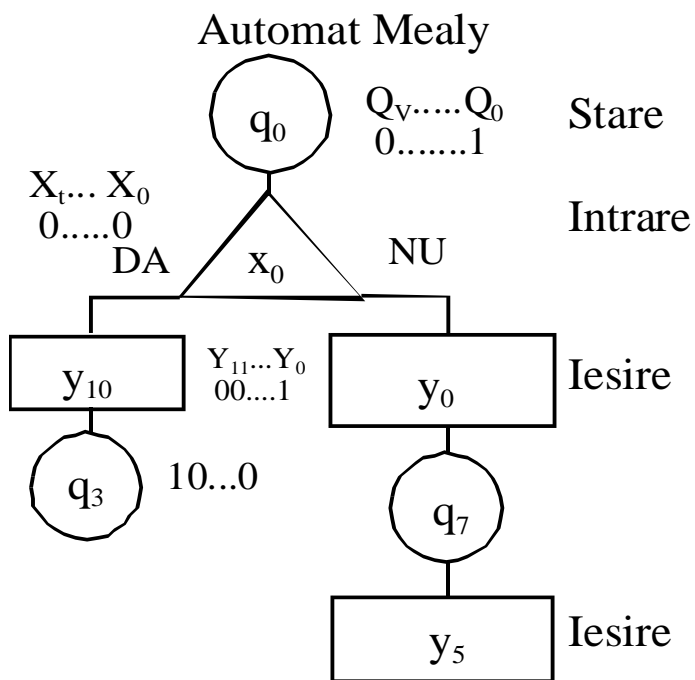


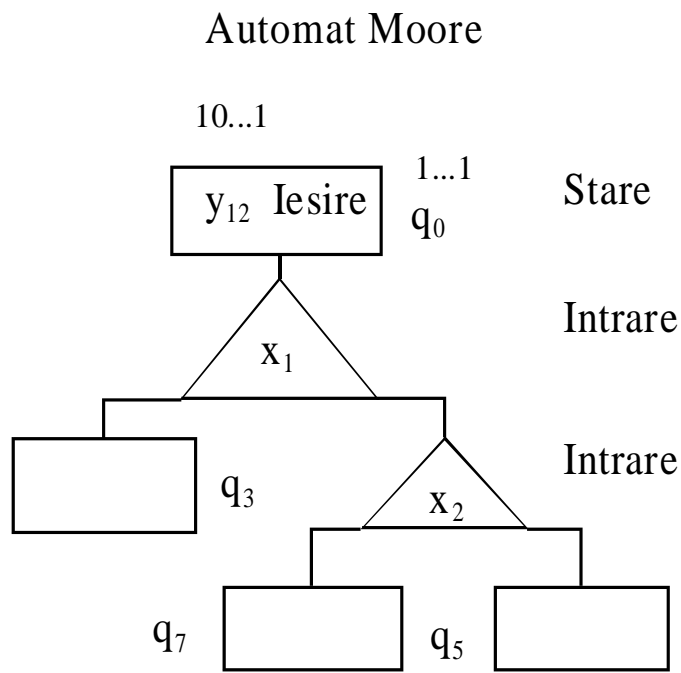
fig. 5

Fig

3) Descrierea prin organigrama:



Fig



Starea se scrie linga iesire

Fig

4) descrierea prin graf in forma binara: se exprima cu variabile (coduri) binare vectorii q , x , y din graf

5) Descrierea prin tabel de tranzitii binare:

Semiautomat		Mealy	
X	Q ^t	Q ^{t+1}	Y
X _t ...X ₀	Q _v ...Q ₀	Q _v ...Q ₀	Y _u ...Y ₀
0...0	0...0	0...0	10...1
0...0	0...1	10...0	00...1
(x ₀)	(q ₀)	(q ₃)	(y ₁₀)
·	·	·	·
·	·	·	·
·	·	·	·
1...1	1...1	01...1	11...1

Moore	
Q ^t	Y
Q _v ...Q ₀	Y _u ...Y ₀
0...0	10..01
·	·
·	·
·	·
·	·
1...1	00...1
(q ₀)	(y ₁₂)

6) Descrierea prin forme de unda:(fig. 7)

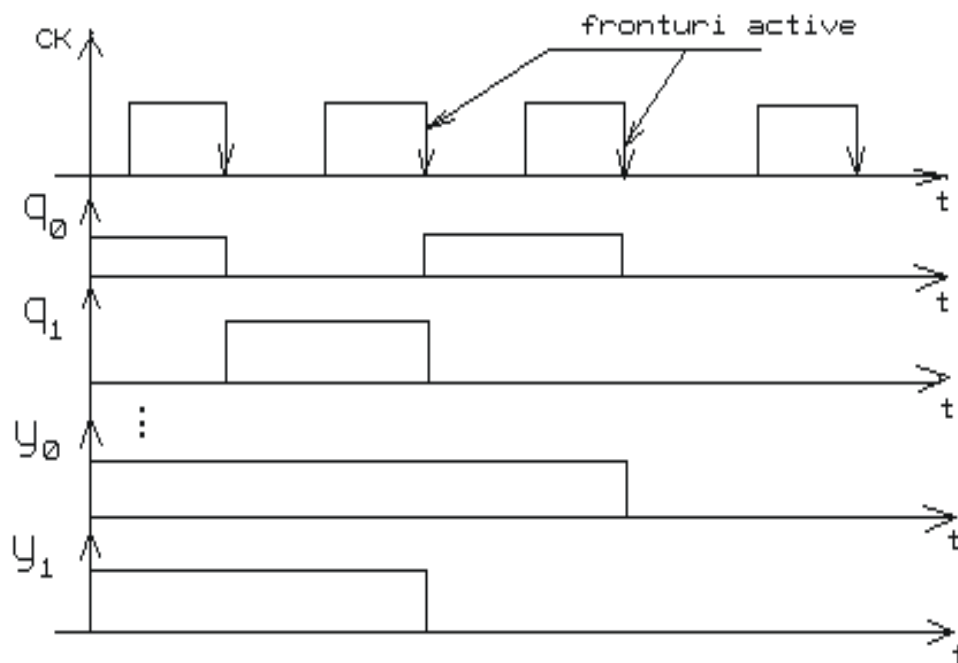
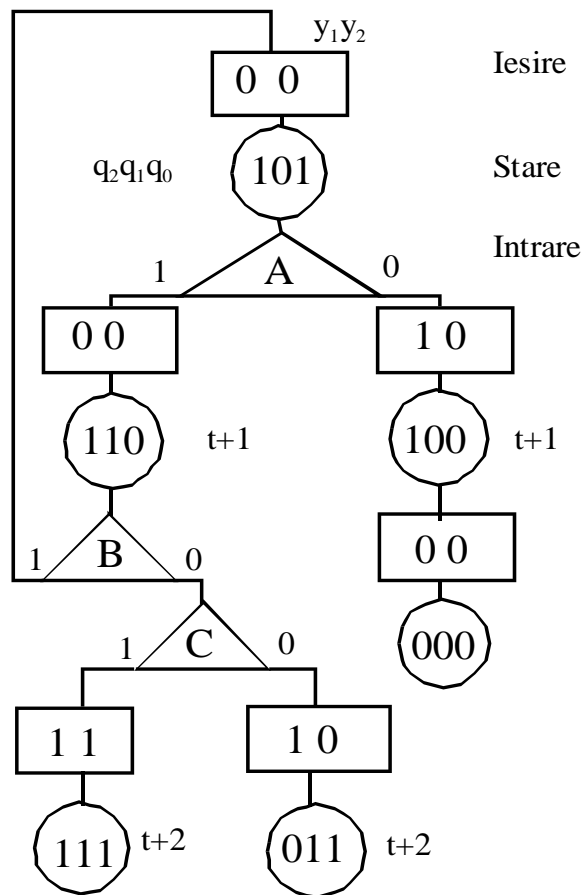


Fig. 7. Descrierea prin forme de unda a automatelor

7) Descrierea prin diagrame V-K:



Fig

	Q ₁			
Q ₂	110	111	101	100
	010	011	001	000
	Q ₀			

	Q ₁			
Q ₂	B+C, B', 1		1, A, 0	0, 0, 0
	Q ₀			

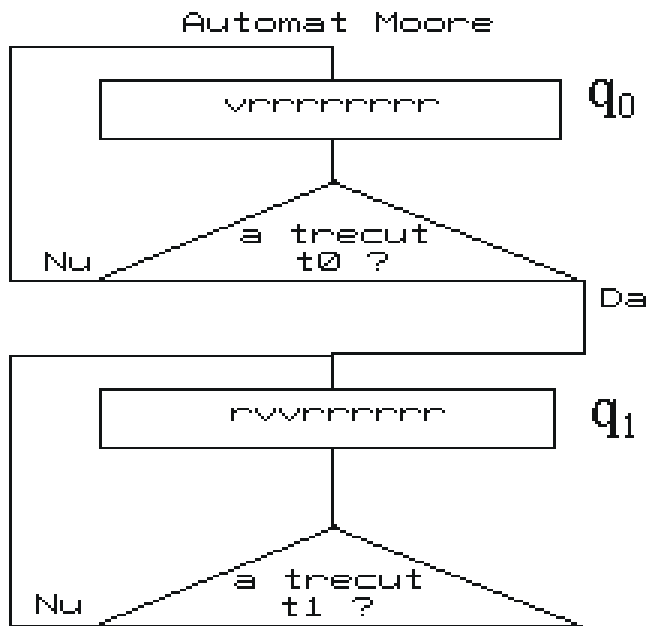
$Q_2^+ Q_1^+ Q_0^+$

Succesorii starii 110

Q₂⁺=1 pt:
 $B+B'C=B+C$
 (semiabsorbție)
 Q₁⁺=1 pt:
 $B'C'+B'C=B'$

	Q ₁			
Q ₂	B', B+C,		A', 0	0, 0
	Q _n			

8) Descrierea in limbaj natural:



Fig

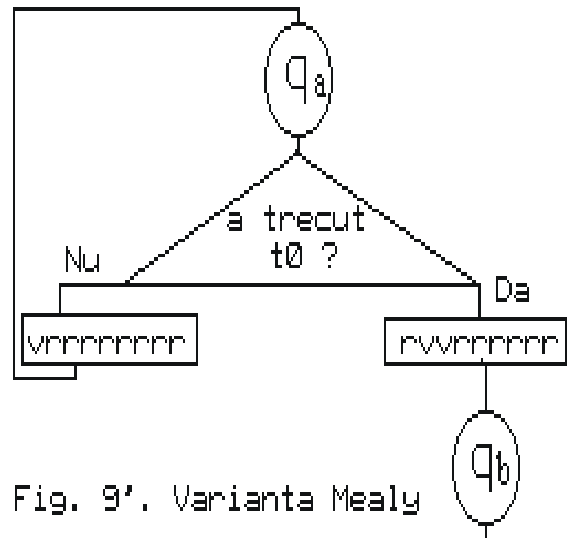
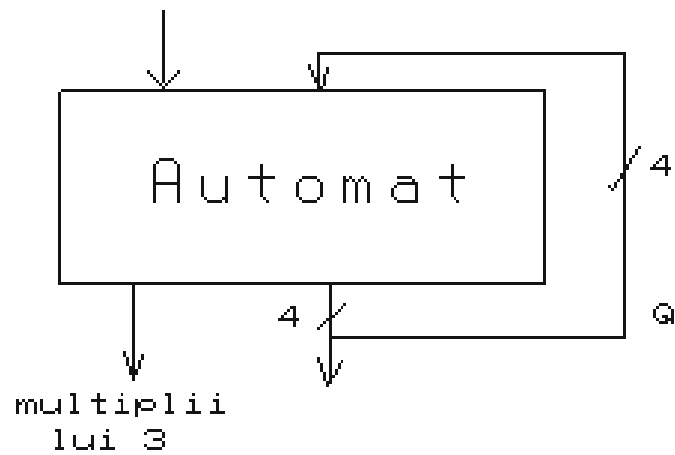


Fig. 9'. Varianta Mealy

Fig



Fig

Clasificarea pe niveluri de complexitate a modalitatii de descriere a functiilor de tranzitie f si g:

I. Limbajul natural;

II. Descrieri matematice : organigrama, graful, tabel de tranzitii;

III.Descrieri matematice binare: organigrama binara, graf binar, forme de unda;

IV.Nivelul de implementare: tabelul de tranzitii binare, diagrame V-K.

Metoda de proiectare a automatelor

Etapa 1. Trecera de la nivelul I (limbaj natural) la nivelul II (descrierea matematica)

- stabilirea multimilor X, Y, Q, si a relatiilor dintre ele (f si g)
- alegerea modelelor (Mealy sau Moore, cu sau fara intirziere)

Etapa 2. trecerea de la nivelul II la III (descrieri binare)

- codificarea starilor (criterii in cursul urmator)
- codificarea sau stabilirea variabilelor binare pentru IN si OUT (variabile codificate, partial codificate sau independente)

Etapa 3- alegarea schemei bloc a automatului

Alternative pentru semiautomat

R_Q	CLC_Q	Obs.
Bistabili D	porti - minimizare, diagrama VK	SL2
	MUX	
	DMUX + porti	
	ROM, PLA	
Bistabili JK	porti, minimizare	SL3 ceva mai simplu: Jk, realizeaza prelucrare
	PLA	
Numaratoare	porti + minimizare	SL3

Etapa 4. Definire CLC_Q si CLC_Y prin:

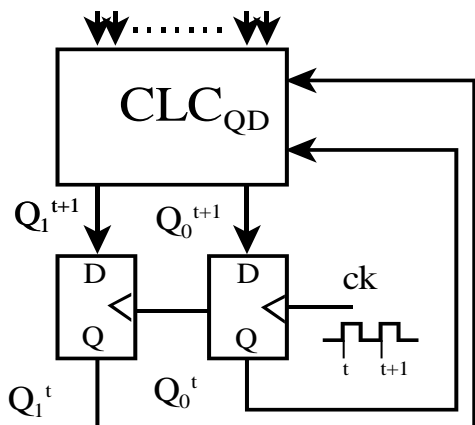
- tabel de adevar - pentru implementarea cu MUX, DMUX, ROM
- diagrame VK - pentru minimizare

- functii boolene - pentru implementare cu PLA

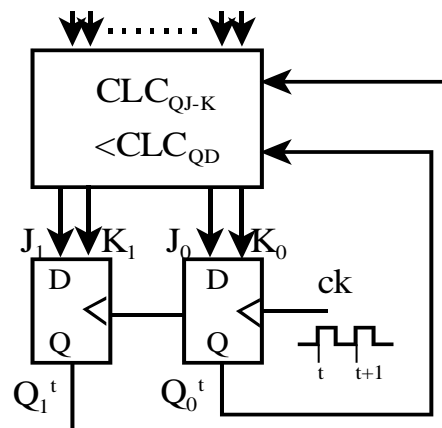
Etapa 5. Implementarea schemei - desenul final

Etapa 6. Realizarea si validarea schemei

SCHEME BLOC DE SEMIAUTOMATE

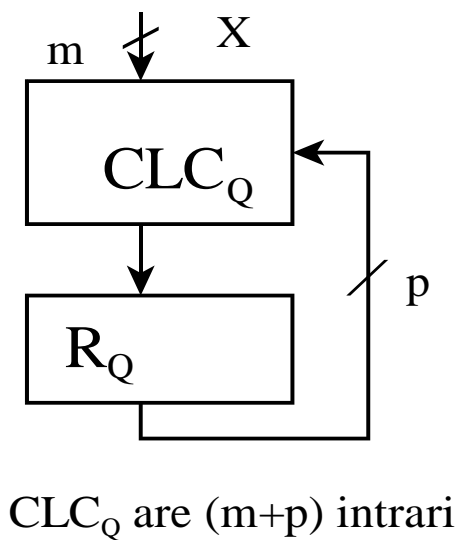


Fig

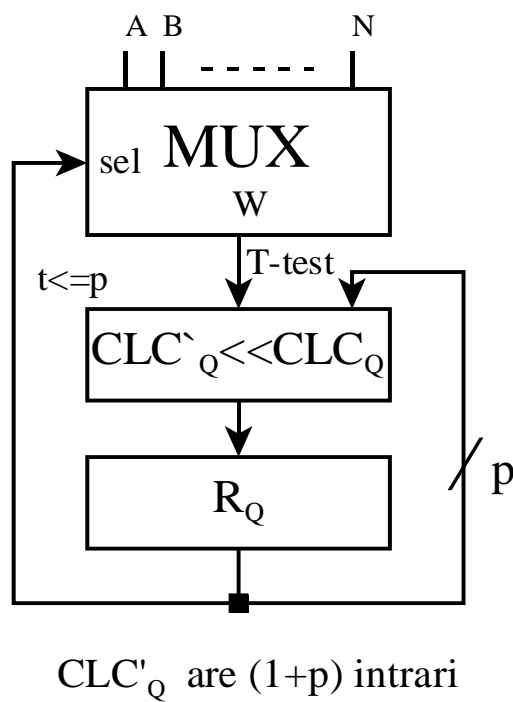


Fig

3) Reducerea CLC_Q prin multiplexarea intrarilor testate succesiv, independent

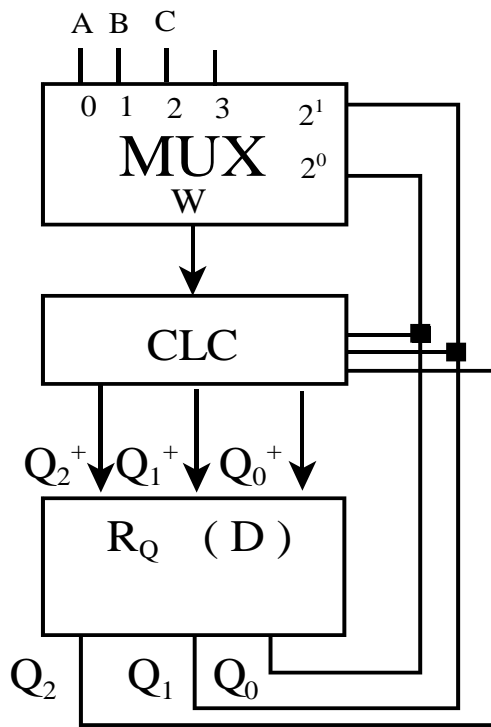


Fig

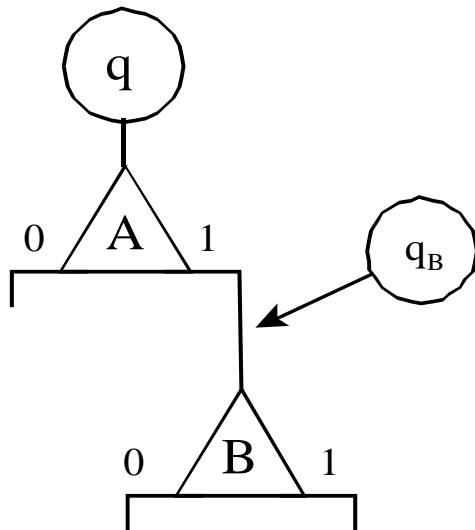


Fig

Pentru 3 variabile de intrare (A, B, C) doar 2 din cei 3 biti de stare (Q_2, Q_1, Q_0) sunt necesari la selectia MUX

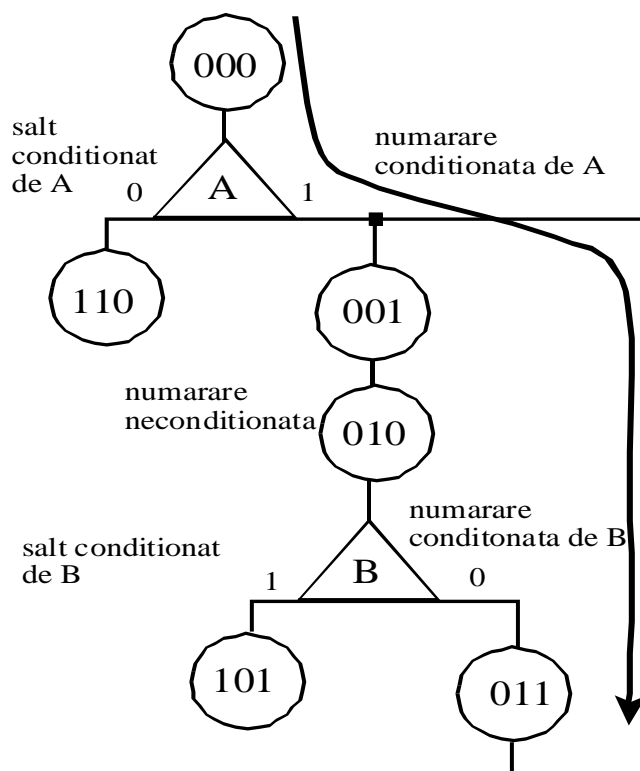


Fig

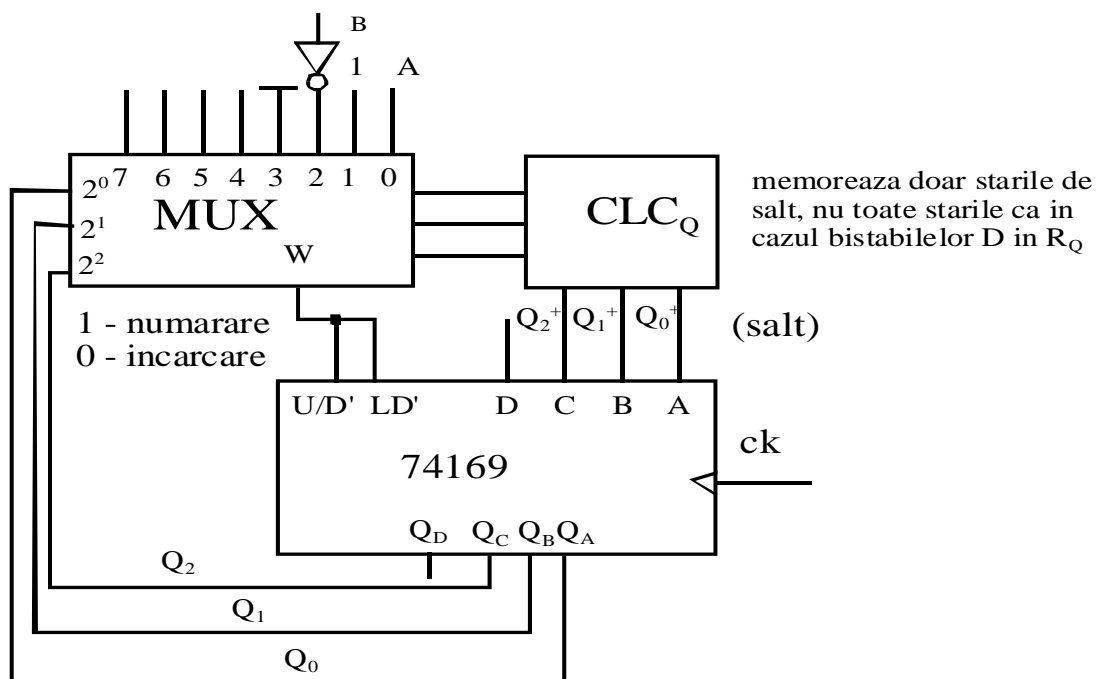


Fig

4) Scema bloc in cazul R_Q cu numarator

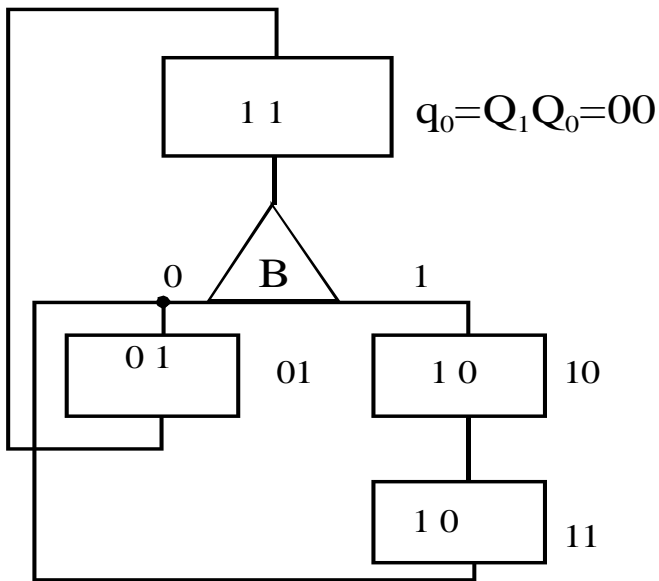


Fig

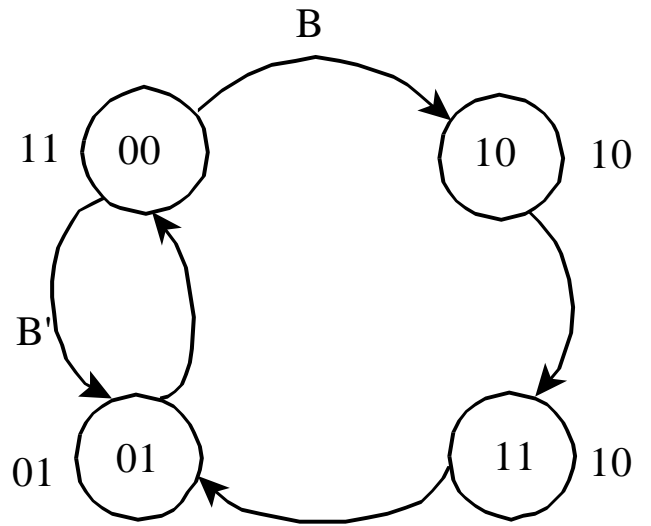


Fig

Proiectarea unui semiautomat cu bistabili D si J-K



Fig



Fig

	Q_1	
Q_0	11	01
	10	00

	Q_1	
Q_0	01	00
	11	$B B'$

	Q_1	
Q_0	10	01
	10	11

$Q \rightarrow Q^+$	JK
0 \rightarrow A	A X
1 \rightarrow A	X A'

	Q_1	
Q_0	X1, X0	X0, X1
	X0, X1	BX, B'X

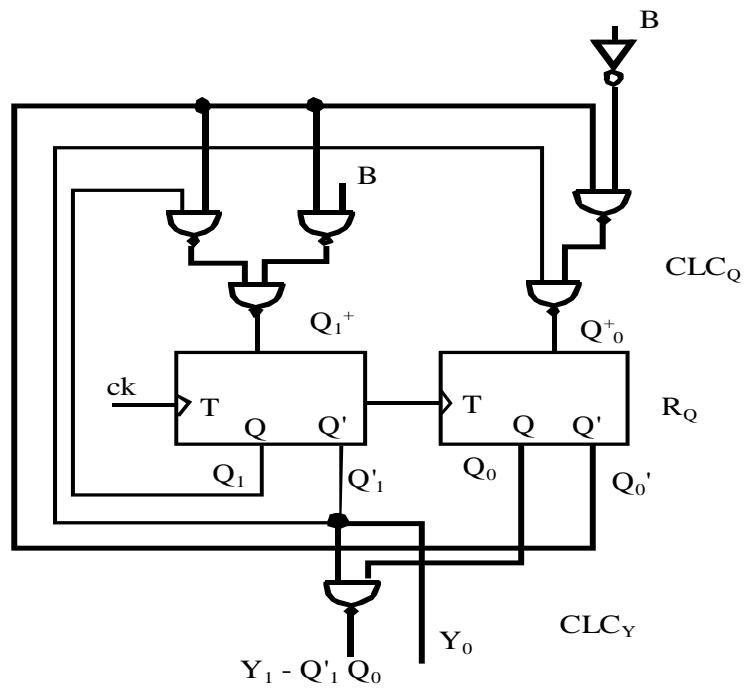
Bistabili JK: functiile $J_1 K_1, J_0 K_0$

	Q_1	
Q_0		
	1	B

	\bar{Q}_1	
Q_0	1	
	1	B'

	Q_1	
Q_0	1	
	1	1

	Q_1	
Q_0		1
		1



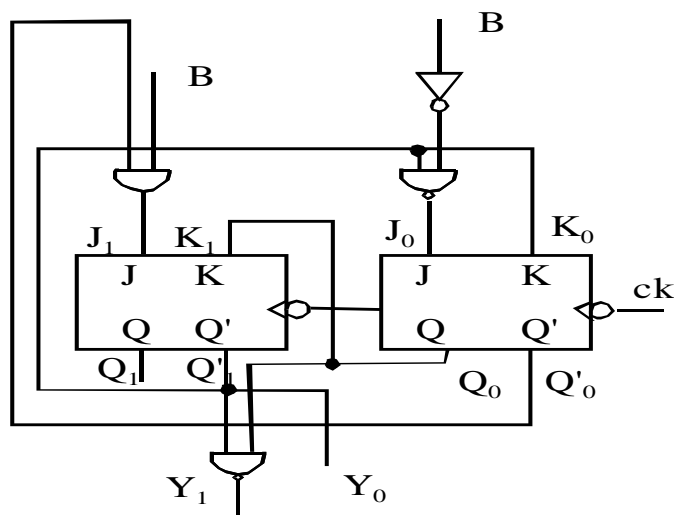
Fig

	Q_1	
Q_0	X	
	X	B

	Q_1	
Q_0	1	X
		X

	Q_1	
Q_0	X	X
	1	B'

	Q_1	
Q_0		1
	X	X



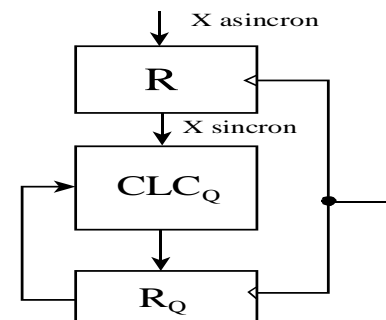
Fig

Criteria de codificare a starilor

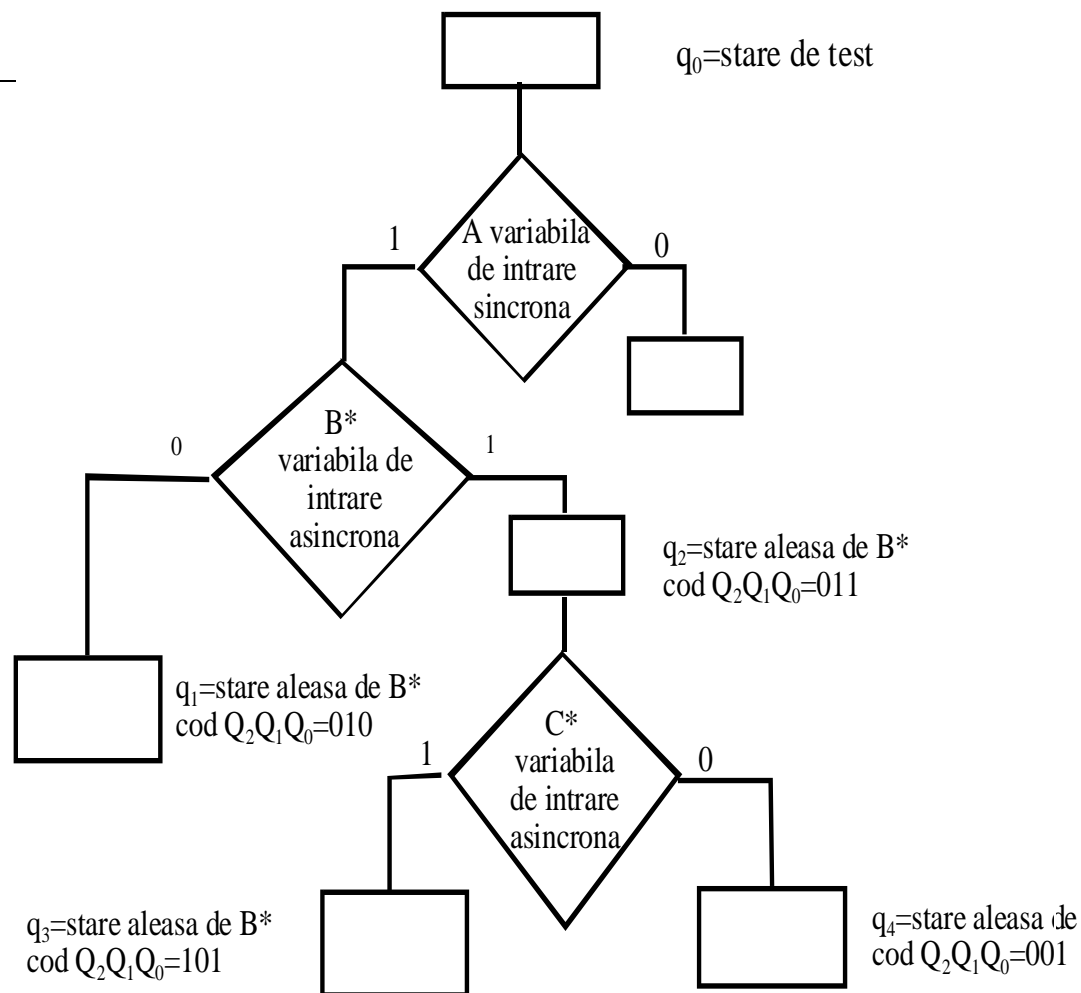
a) Criteriul dependentei reduse

- asigura implementare minima
- asigura buna functionare in cazul in care o variabila de intrare este asincrona.

Starile alese de variabila asincrona sa difere doar printr-un bit



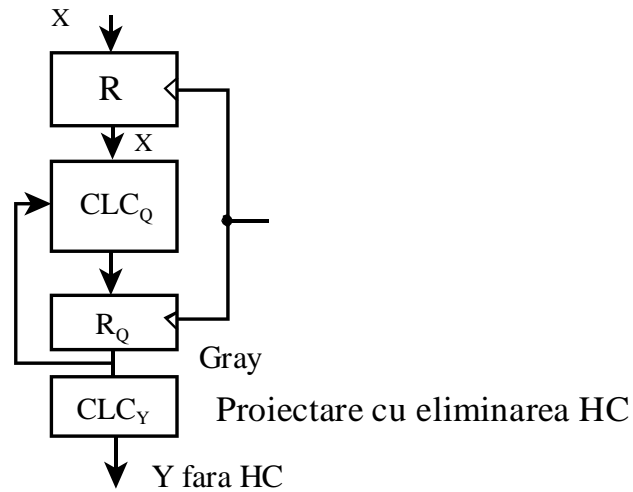
Fig



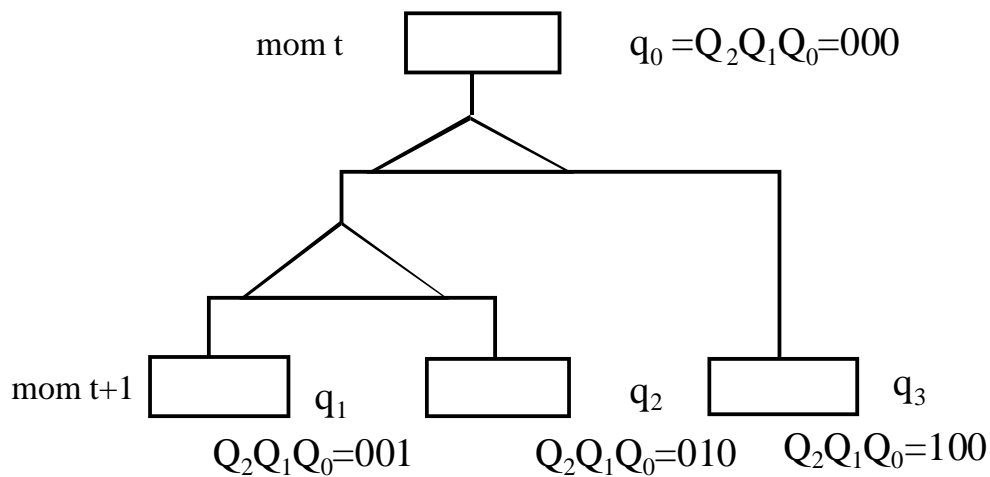
Fig

2.Codificarea cu variatie minima

-asigura succedarea starilor in coduri Gray, deci eliminarea hazardului combinational la CLC_y datorate nesincronismului pt MOORE imediat



Fig



Fig

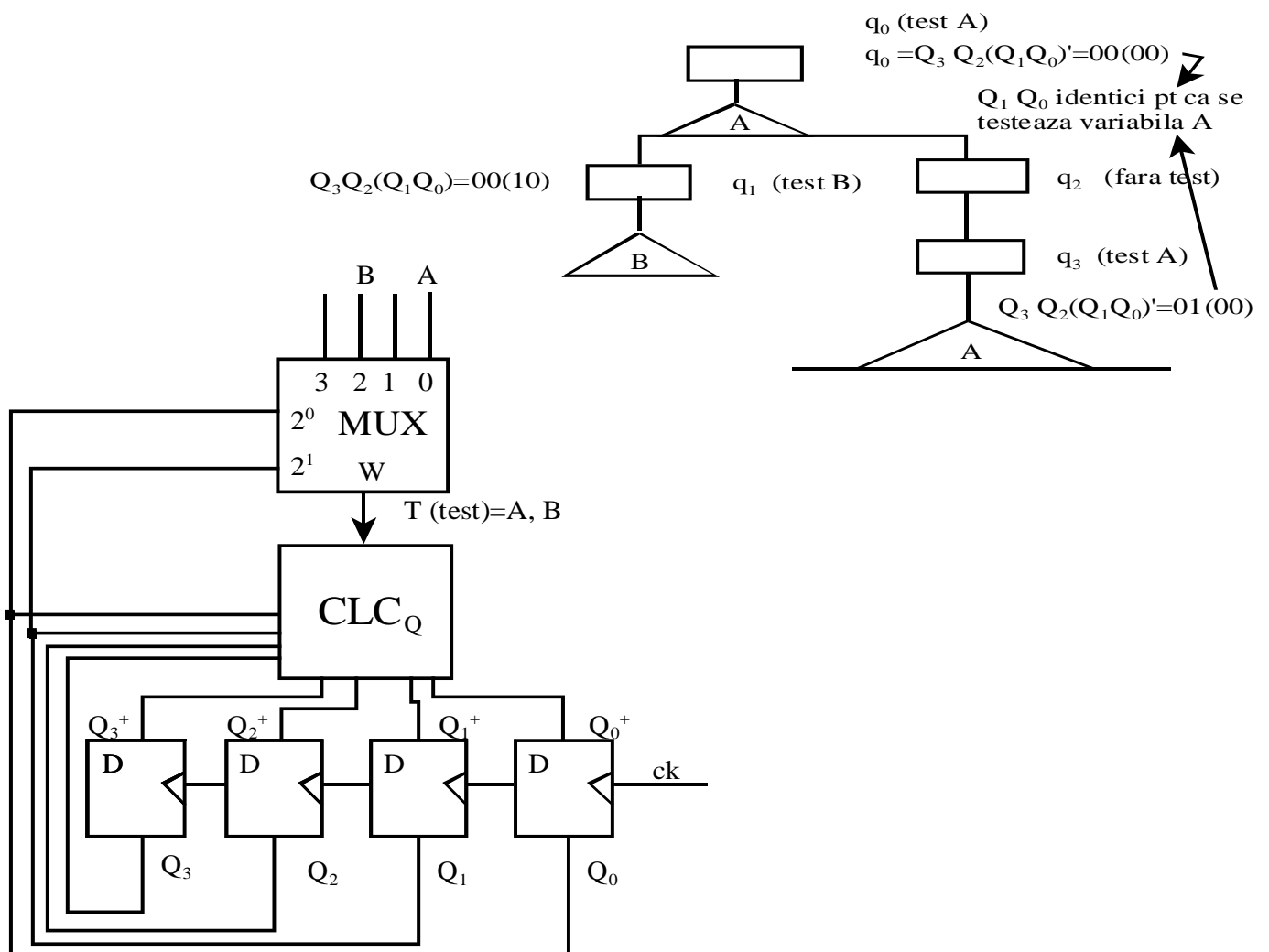
3. Codificarea starilor in cazul in care intrarile sunt testate independent (in scheme cu MUX de testare a intrarilor)

Obs: numarul conditiilor de test (al intrarilor) este uzual mai mic decat cel al starilor; in aceasta situatie nu toti bitii de stare sunt utilizati pt selectia MUX. Metoda consta in:

- alegerea bitilor de stare pt selectia MUX

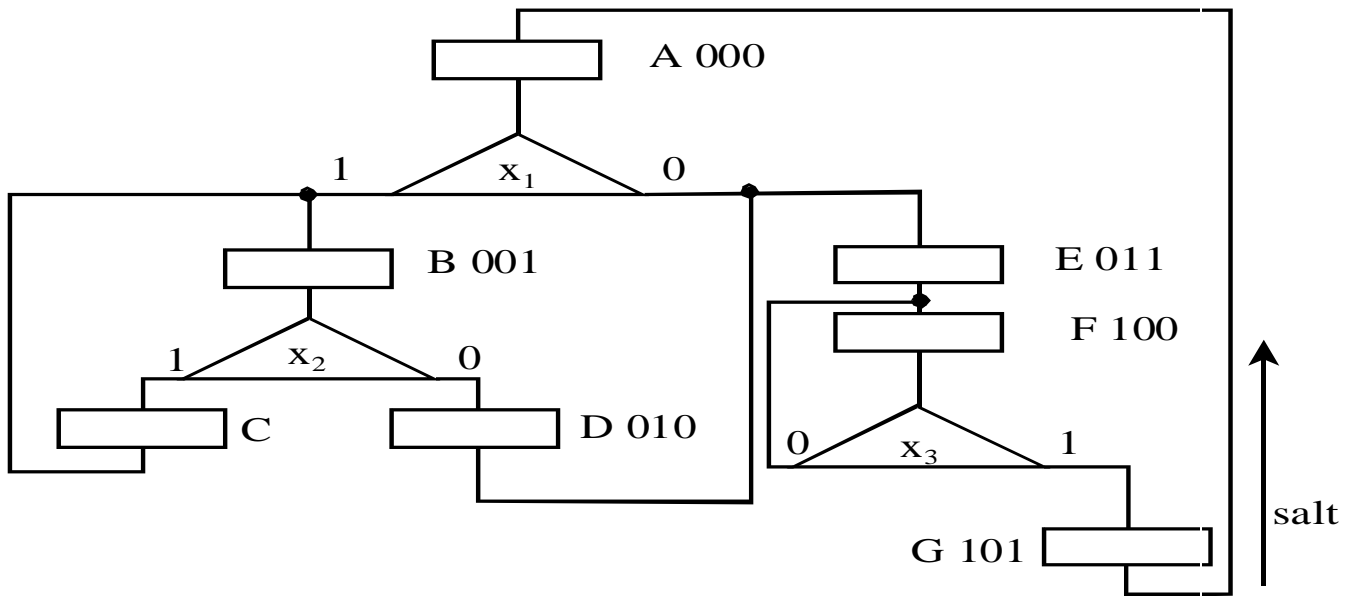
Ex: pt automatul cu 16 stari si 4 variabile de intrare independente sunt necesari 4 bitii de stare $Q_3 Q_2 Q_1 Q_0$ si 2 bitii pentru selectarea MUX (fie $Q_1 Q_0$)

- codificarea identica a bitilor pentru selectia MUX a acelorasi varibile si diferita pentru variabile distincte

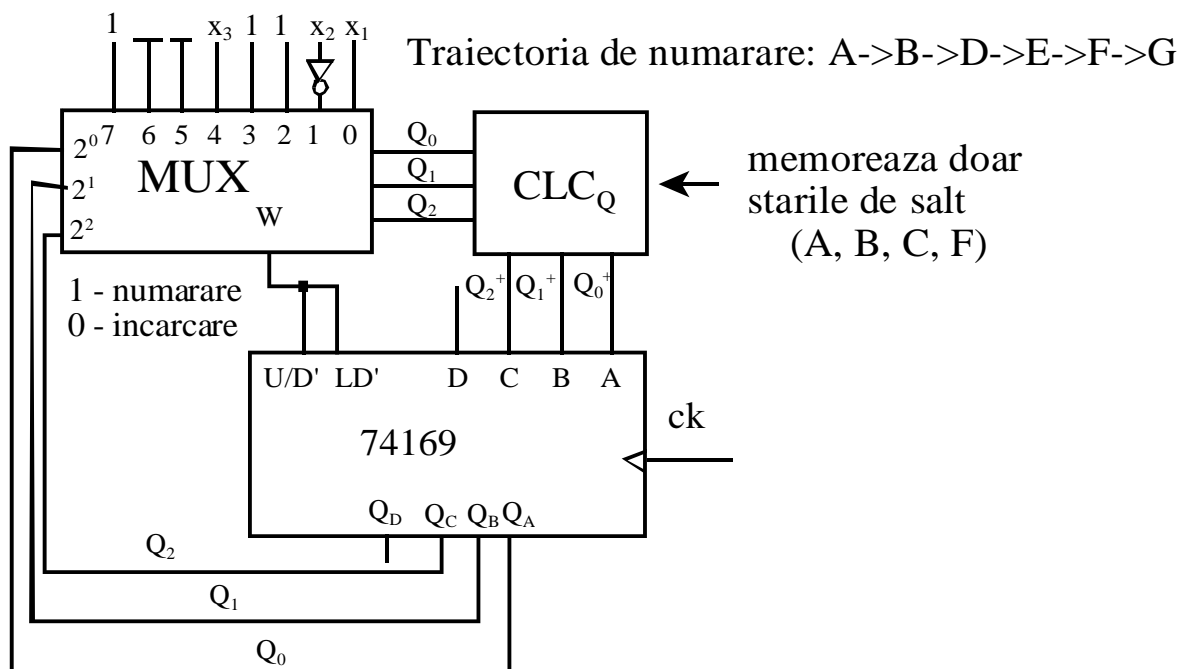


Fig

4. Codificarea prin numarare pe traiectoria cea mai lunga:
 - utilizata pentru automate cu numarator pe spatiul starilor: se alege traiectoria cea mai lunga si se codifica cu coduri succesive



Fig



Fig

Proiectarea CLC_Q

- memoreaza doar starile de salt A, B, C, F

	Q ₁			
Q ₂	110	111	101	100
	010	011	001	000
	Q ₀			

	Q ₁			
Q ₂	001 B	xxx	000 A	100 F
	xxx	xxx	110 C	001 B
	Q ₀			

	Q ₁			
Q ₂		X		1
	X	X	1	
	Q ₀			

$$Q_2^+ Q_1^+ Q_0^+$$

	Q ₁			
Q ₂		X		
	X	X	1	
	Q ₀			

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

Solutia II nu e buna pentru ca introduce un inversor suplimentar pentru Q'₀

$$Q_1^+ = Q_2' Q_0$$

	Q ₁			
Q ₂		X		
	X	X		1
	Q ₀			

$$Q_0^+ = Q_2' Q_0' + Q_2 Q_1 \quad \text{I}$$

$$+ Q_1 Q_0' \quad \text{II}$$

Automate cu spatiul starilor incomplet specificat

$Q_u = \{0, 1, 2, \dots, 5, 6\}$
 $Q_n = \{7\}_H$

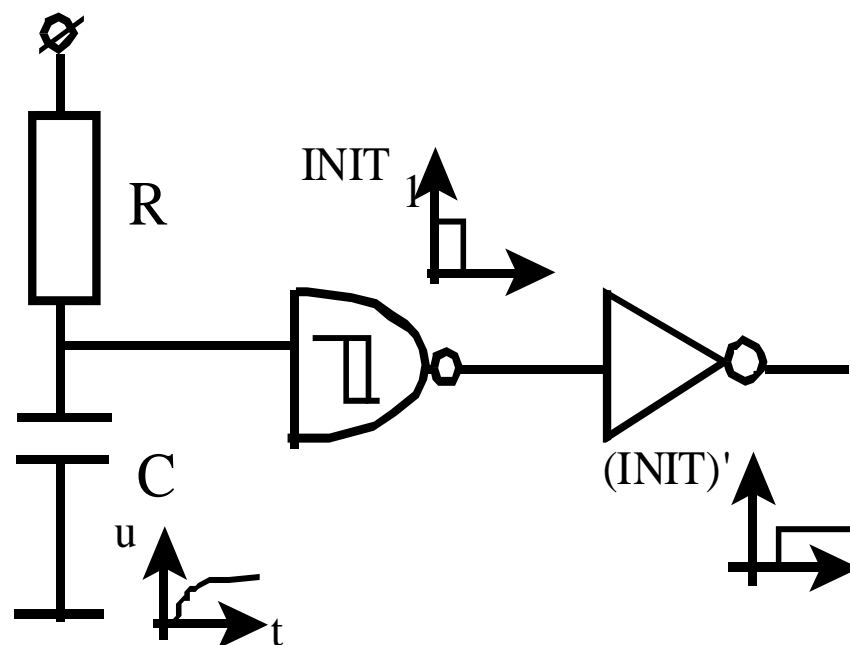
u - util
n - neutil

Trecerea in starile neutile poate avea loc:

- in momentul initial
- in timpul functionarii

Remediu:

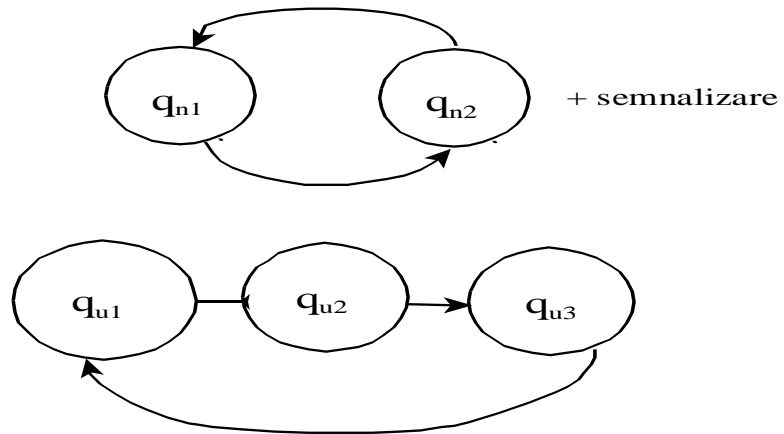
- prin autoinitializarea automatului



Fig

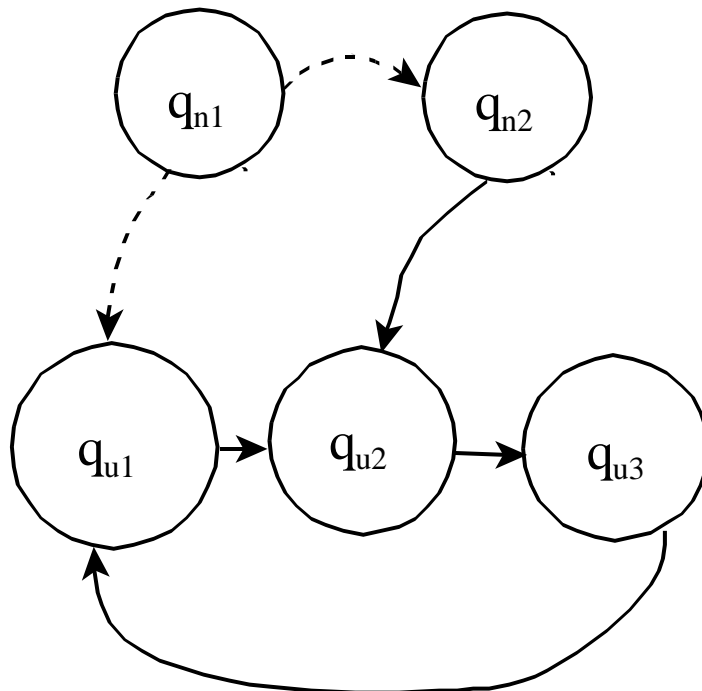
b) In proiectare se pot urmari 2 scopuri:

I. Detectia erorilor si semnalizarea lor prin blocarea automatului in starile neutilizate



Fig

II. Autocorectia erorilor prin tranzitia din starile neutile in cele utile



Fig