



Universitatea
Transilvania
din Brașov



Universitatea
Transilvania
din Brașov
FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

PROIECT ASCN1

Conducător: Profesor dr. ing. Moldoveanu Florin-Dumitru

Student: Borș Dorin

Grupa: 4LF494

2021

FIȘA PROIECTULUI DE DISCIPLINĂ

Disciplina:

Analiza și sinteza circuitelor numerice I, an II/sem. 4.

Tema de proiect nr. 31:

Să se proiecteze un convertor de cod de 4 biți pentru conversia codului binar-zecimal ponderat 4221 în codul binar-zecimal ponderat 8421 (logică combinațională).

Cerințe de proiectare:

În rezolvarea temei proiectului se vor trata următoarele puncte:

- Să se reprezinte funcțiile logice asociate circuitului combinațional prin forma canonică disjunctivă (FCD), forma canonică conjunctivă (FCC), tabel de adevăr și diagrame Veitch-Karnaugh.
- Să se obțină formele minime disjunctive și conjunctive pentru funcțiile logice asociate convertorului de cod, (utilizându-se combinațiile indiferente) prin metoda diagramelor Veitch-Karnaugh; de asemenea, se vor obține formele minime disjunctive pentru primele două dintre funcțiile logice de ieșire și prin metoda Quine-McCluskey.
- Să se implementeze fiecare funcție logică, independent, numai cu porți logice ȘI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).
- Să se implementeze ansamblul funcțiilor logice numai cu porți logice ȘI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).
- Să se implementeze ansamblul funcțiilor logice în următoarea variantă: primele două funcții logice de ieșire cu porți logice ȘI-NU (circuite integrate TTL), iar următoarele două cu porți logice SAU-NU (circuite integrate CMOS).
- Să se implementeze ansamblul funcțiilor logice cu MUX-uri de 8, respectiv 16 căi (se vor utiliza circuite integrate realizate în tehnologia TTL).
- Să se implementeze ansamblul funcțiilor logice cu DMUX-uri de 8, respectiv 16 căi și porți logice ȘI-NU în prima variantă, respectiv ȘI în a doua variantă (se vor utiliza circuite integrate realizate în tehnologia CMOS).
- Să se calculeze timpii de propagare „intrare-ieșire”, pentru toate schemele logice obținute.
- Să se calculeze puterile disipate pentru toate schemele logice obținute.
- Să se compare soluțiile de implementare obținute.
- Se va face analiza, prin simulare, a tuturor schemelor logice obținute utilizându-se pachetul de programe OrCAD.

Pe schemele logice obținute se vor specifica tipul și gradul de utilizare al fiecărui circuit integrat.

Bibliografia recomandată:

- [1] Maican, S. – *Sisteme numerice cu circuite integrate. Culegere de probleme*, Ed. Tehnică, București, 1980.
- [2] Ștefan, Gh. M., Bistriceanu, V. – *Circuite integrate digitale. Probleme. Proiectare*, Ed. Albastră, Cluj-Napoca, 2000.
- [3] Wakerly, J.F. – *Circuite digitale*, Ed. Teora, București, 2002.
- [4] Wilkinson, B. – *Electronică digitală. Bazele proiectării*, Ed. Teora, București, 2002.
- [5] Moldoveanu, F., Floroian, D. – *Circuite logice și comenzi secvențiale. Circuite logice combinaționale*, Ed. Universității Transilvania din Brașov, 2003.
- [6] Toacșe, Gh., Nicula, D. – *Electronică digitală*, Ed. Tehnică, București, 2005.

Condiții de redactare:

Pentru redactare se va folosi template-ul recomandat de cadrul didactic îndrumător.

Evaluări pe parcurs:

S-au stabilit două vize pentru evaluarea pe parcurs a proiectului la următoarele date:

- viza I-a: 14.04.2021;
- viza a II-a: 19.05.2021.

Termenul de predare și susținere:

Proiectul se va preda și susține în ultima săptămână a semestrului.

Notarea proiectului:

Forma finală a proiectului trebuie să conțină rezolvări pentru toate punctele cerute prin temă iar, din punct de vedere al redactării, să aibă forma solicitată. În cursul susținerii, studentul trebuie să dovedească cunoașterea metodelor specifice de rezolvare pentru problemele date, utilizarea corectă și fluentă a termenilor specifici și interpretarea corectă a rezultatelor. Notarea va porni de la nota 10, dacă studentul a primit ambele vize de evaluare, de la nota 8 dacă studentul a primit o singură viză și de la nota 6 în cazul în care studentul nu are nicio viză.

Martie 2021

Titular activități de proiect,
Prof. dr. ing. Florin Dumitru MOLDOVEANU

Cuprins

1. Exercițiul 1	7
1.1 Enunțul problemei.....	7
1.2 Considerații teoretice.....	7
1.2.1 Circuit logic combinațional.....	7
1.2.2 Forma canonică conjunctivă și disjunctivă ale funcției.....	7
1.2.3 Tabel de adevăr.....	7
1.2.4 Diagramă Veitch-Karnaugh.....	8
1.3 Rezolvare.....	8
1.3.1 Tabel de adevăr.....	9
1.3.2 Formele canonice disjunctive și conjunctive ale funcțiilor.....	9
1.3.3 Diagramele Veitch-Karnaugh ale funcțiilor.....	10
2. Exercițiul 2	12
2.1 Enunțul problemei.....	12
2.2 Considerații teoretice.....	12
2.2.1 Formele minime disjunctive și conjunctive ale funcțiilor (metoda diagramelor Karnaugh)	12
2.2.2 Metoda Quine-McCluskey.....	13
2.3 Rezolvare.....	14
2.3.1 Formele minime disjunctive ale funcțiilor.....	14
2.3.2 Formele minime conjunctive ale funcțiilor.....	16
2.3.3 Metoda Quine-McCluskey pentru primele două funcții logice de ieșire.....	18
3. Exercițiul 3	21
3.1 Enunțul problemei.....	21
3.2 Considerații teoretice.....	21
3.2.1 Implementarea funcțiilor folosind porți logice ȘI-NU în tehnologie TTL.....	21
3.3 Rezolvare.....	22
4. Exercițiul 4	24
4.1 Enunțul problemei.....	24
4.2 Rezolvare.....	24
5. Exercițiul 5	25
5.1 Enunțul problemei.....	25

5.2 Considerații teoretice.....	25
5.2.1 Implementarea funcției SAU-NU și funcției ȘI-NU.....	25
5.3 Rezolvare.....	26
6. Exercițiul 6	26
6.1 Enunțul problemei.....	26
6.2 Considerații teoretice.....	27
6.2.1 Implementarea schemei cu MUX-uri în tehnologie TTL	27
6.2.2 Implementarea funcției a cu 2xMUX-uri de 8 căi conectate în paralel.....	27
6.2.3 Implementarea funcției b cu MUX de 8 căi și o variabilă aplicată pe intrările de date	27
6.2.4 Implementarea funcțiilor c și d cu MUX de 16 căi	27
6.3 Rezolvare.....	28
6.3.1 Implementarea funcției a cu 2xMUX-uri de 8 căi conectate în paralel.....	28
6.3.2 Implementarea funcției b cu MUX de 8 căi și o variabilă aplicată pe intrările de date	29
6.3.3 Implementarea funcției c cu MUX de 16 căi	31
6.3.4 Implementarea funcției d cu MUX de 16 căi	32
7. Exercițiul 7.....	34
7.1 Enunțul problemei.....	34
7.2 Considerații teoretice.....	35
7.3 Rezolvare.....	35
7.3.1 Implementarea funcției a cu DMUX de 8 căi și rețea de porți logice ȘI-NU.....	35
7.3.2 Implementare funcției b cu DMUX-uri de 8 căi conectate în paralel	37
7.3.3 Implementare funcției c cu DMUX de 16 căi și porți logice ȘI-NU.....	38
7.3.4 Implementarea funcției d cu DMUX de 16 căi și porți logice ȘI.....	39
8. Exercițiul 8	40
8.1 Enunțul problemei.....	40
8.2 Considerații teoretice.....	40
8.3 Rezolvare.....	41
9. Exercițiul 9	43
4.1 Enunțul problemei.....	43
4.2 Rezolvare.....	43
10. Exercițiul 10.....	46
10.1 Enunțul problemei.....	46

10.2 Rezolvare	46
11. ANEXA 1	48
12. Bibliografie.....	52

1. Exercițiul 1

1.1 Enunțul problemei

Să se reprezinte funcțiile logice asociate circuitului combinațional prin forma canonică disjunctivă (FCD), forma canonică conjunctivă (FCC), tabel de adevăr și diagrame Veitch-Karnaugh.

1.2 Considerații teoretice

1.2.1. Circuit logic combinațional

Un circuit logic combinațional este un circuit de comutare combinațională ce se caracterizează prin aceea că la un moment dat starea ieșirilor circuitului depinde doar de starea intrărilor sale. Legătura dintre starea ieșirilor și starea intrărilor este dată și în acest caz de funcția de transfer a circuitului.

Suportul fizic utilizat în CLC este destul de variat. Studiul CLC și al celor secvențiale se face folosind un model al acestora numit rețea de comutare sau schemă logică. În rețeaua de comutare sau schema logică se face abstracție de caracteristicile constructive ale elementelor ce concep rețeaua sau schema logică avându-se în vedere doar proprietățile lor funcționale. Din acest motiv rețeaua are un înalt grad de generalitate, permițând studiul unor clase largi de circuite logice.

1.2.2 Forma canonică conjunctivă și disjunctivă ale funcției

Pentru a obține din tabelul de adevăr forma canonică conjunctivă se iau în considerare combinațiile pentru care funcția are valoarea 0, iar pentru forma canonică disjunctivă se iau în considerare combinațiile pentru care funcția are valoarea 1.

1.2.3 Tabel de adevăr

Acest mod de reprezentare corespunde reprezentării tabelare a funcțiilor booleene și de care s-a uzitat până aici. Tabelele denumite de adevăr sau combinaționale conțin în partea stângă un număr de linii egal cu numărul combinațiilor posibile ale valorilor argumentelor, iar în partea dreaptă valorile funcției pentru fiecare combinație de valori ale argumentelor. Tabelul de adevăr este cea mai practică reprezentare a unei funcții booleene deoarece pentru fiecare combinație posibilă a valorilor argumentelor se indică valoarea funcției.

1.2.4 Diagramă Veitch-Karnaugh

Diagrama Karnaugh se construiește punând valorile unor variabile pe linii și al celorlalte variabile pe coloane. Distribuția valorilor trebuie să respecte codului Gray, astfel că între două căsuțe vecine există o diferență de un bit. Valoarea asociată unei căsuțe este valoarea funcției pentru combinația respectivă de valori de intrare. Valorile posibile sunt 1, 0 sau X (nu contează).

După ce am construit diagrama Karnaugh, trebuie să aflăm expresiile minimale care vor fi utilizate în expresia finală. Acest lucru va fi făcut prin încadrarea valorilor de 1, astfel încât figura reprezentată de acestea să aibă următoarele proprietăți:

- are formă dreptunghiulară
- conține 2^n elemente, unde n este un număr natural (1, 2, 4, 8, 16 etc. elemente)
- este maximală, adică nu poate fi inclusă într-o altă figură cu aceste proprietăți, dar cu n mai mare. Dacă un mintermen are un vecin un alt mintermen de aceeași dimensiune, aceștia se porțuni, rezultând un mintermen cu mai puține variabile în componență.
- conține elemente X doar dacă zona rezultată este mai mare decât cea care putea fi obținută fără elementul respectiv (adică dacă se obține o reducere a termenului algebric echivalent).

Grupurile generate sunt convertite într-o expresie booleană prin:

- determinarea valorilor posibile ale variabilelor în grupul respectiv. Dacă figura se află pe o coloană unde $A=1$, se ia în considerare valoarea A . Dacă figura se află pe o coloană unde $A=0$, se ia în considerare valoarea \bar{A} .
- se elimină variabilele care sunt prezente atât cu valoarea posibilă 1, cât și cea 0.

Fiecare grup generează astfel un produs, iar valoarea expresiei booleene este suma acestor produse. Funcția inversă se poate obține grupând valorile de 0, în locul celor de 1.

1.3 Rezolvare

Codul 8421 se mai numește și *codul binar-zecimal natural (NBCD – Natural Binary Coded Decimal)*, deoarece având ca ponderi puterile lui 2, fiecare tetradă de biți reprezintă de fapt exprimarea cifrei zecimale respective în sistemul de numerație binar.

Codul 4221 are aceleași proprietăți ca și codul 2421 (Aiken), utilizează ponderea 2 în două poziții distincte ale tetradei, iar tetradele care reprezintă cifre zecimale a căror sumă este egală cu 9 se completează reciproc; primele cinci cifre zecimale au pe prima poziție 0, iar ultimele cinci 1.

1.3.1 Tabel de adevăr

Nr. ctr	4221				8421			
	x_1	x_2	x_3	x_4	a	b	c	d
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	1	0	0	1	0	0
5	1	0	0	1	0	1	0	1
6	1	1	0	0	0	1	1	0
7	1	1	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0
9	1	1	1	1	1	0	0	1

Tab. 1.1 Reprezentarea funcției prin table de adevăr

Combinății indiferente: 4,5,7,8,10,11.

1.3.2 Formele canonice disjunctive și conjunctive ale funcțiilor

$$a^{FCD}(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 = P_{14} + P_{15} = \Sigma(14, 15) \quad (1.1)$$

$$b^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 = P_6 + P_9 + P_{12} + P_{13} = \Sigma(6, 9, 12, 13) \quad (1.2)$$

$$c^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 = P_2 + P_3 + P_{12} + P_{13} = \Sigma(2, 3, 12, 13) \quad (1.3)$$

$$d^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 x_3 x_4 = P_1 + P_3 + P_9 + P_{13} + P_{15} = \Sigma(1, 3, 9, 13, 15) \quad (1.4)$$

$$a^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + x_2 + \bar{x}_3 + x_4)(x_1 + x_2 + \bar{x}_3 + \bar{x}_4) \cdot$$

$$(x_1 + \bar{x}_2 + \bar{x}_3 + x_4)(\bar{x}_1 + x_2 + x_3 + \bar{x}_4)(\bar{x}_1 + \bar{x}_2 + x_3 + x_4)(\bar{x}_1 + \bar{x}_2 + x_3 + \bar{x}_4) = S_0 \cdot S_1 \cdot S_2 \cdot S_3 \cdot S_6 \cdot S_9 \cdot$$

$$S_{12} \cdot S_{13} = \Pi(0,1,2,3,6,9,12,13) \quad (1.5)$$

$$b^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + x_2 + \bar{x}_3 + x_4)(x_1 + x_2 + \bar{x}_3 + \bar{x}_4) \cdot$$

$$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + x_4)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) = S_0 \cdot S_1 \cdot S_2 \cdot S_3 \cdot S_{14} \cdot S_{15} = \Pi(0,1,2,3,14,15) \quad (1.6)$$

$$c^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + \bar{x}_2 + \bar{x}_3 + x_4)(\bar{x}_1 + x_2 + x_3 + \bar{x}_4) \cdot$$

$$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + x_4)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) = S_0 \cdot S_1 \cdot S_6 \cdot S_9 \cdot S_{14} \cdot S_{15} = \Pi(0,1,6,9,14,15) \quad (1.7)$$

$$d^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + \bar{x}_3 + x_4)(x_1 + \bar{x}_2 + \bar{x}_3 + x_4)(\bar{x}_1 + \bar{x}_2 + x_3 + x_4) \cdot$$

$$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + x_4) = S_0 \cdot S_2 \cdot S_6 \cdot S_{12} \cdot S_{14} = \Pi(0,2,6,12,14) \quad (1.8)$$

1.3.3 Diagramele Veitch-Karnaugh ale funcțiilor

		x_1x_2			
		00	01	11	10
x_3x_4	00	0 0	* 4	0 12	* 8
	01	0 1	* 5	0 13	0 9
	11	0 3	* 7	1 15	* 11
	10	0 2	0 6	1 14	* 10

Fig. 2.1 Diagrama Veitch-Karnaugh pentru funcția a

$X_3X_4 \backslash X_1X_2$		X_1X_2			
		00	01	11	10
00	0 0	*	1 12	*	8
01	0 1	*	1 13	1 9	
11	0 3	*	0 15	*	11
10	0 2	1 6	0 14	*	10

Fig. 2.2 Diagrama Veitch-Karnaugh pentru funcția b

$X_3X_4 \backslash X_1X_2$		X_1X_2			
		00	01	11	10
00	0 0	*	1 12	*	8
01	0 1	*	1 13	0 9	
11	1 3	*	0 15	*	11
10	1 2	0 6	0 14	*	10

Fig. 2.3 Diagrama Veitch-Karnaugh pentru funcția c

$X_3X_4 \backslash X_1X_2$		X_1X_2			
		00	01	11	10
00	0	0 0	* 4	0 12	* 8
		1 1	* 5	1 13	1 9
11	3	1 3	* 7	1 15	* 11
		0 2	0 6	0 14	* 10

Fig. 2.4 Diagrama Veitch-Karnaugh pentru funcția d

2. Exercițiul 2

2.1 Enunțul problemei

Să se obțină formele minime disjunctive și conjunctive pentru funcțiile logice asociate convertorului de cod, (utilizându-se combinațiile indifferente) prin metoda diagramelor Veitch-Karnaugh; de asemenea, se vor obține formele minime disjunctive pentru primele două dintre funcțiile logice de ieșire și prin metoda Quine-McCluskey.

2.2 Considerații teoretice

2.2.1 Formele minime disjunctive și conjunctive ale funcțiilor (metoda diagramelor Karnaugh)

Metoda diagramelor Karnaugh este o metodă grafo-analitică, foarte utilă la minimizarea funcțiilor cu un număr relativ mic de variabile. Metoda pornește de la una din formele canonice ale funcției. Diagrama Karnaugh se prezintă sub forma unui pătrat sau dreptunghi cu 2^n locații, în cazul nostru 16 locații. În fiecare locație va apărea un termen canonic al funcției.

Diagrama este astfel organizată, încât două compartimente vecine pe linie sau pe coloană, să difere printr-o aceeași variabilă, variabilă care într-o combinație să apară negată, și în alta adevărată. (proprietatea de adiacență).

Pentru obținerea formei minime conjunctive, se obține mai întâi forma minimă conjunctivă a funcției negate. Apoi se neagă această formă minimă, și folosind formulele lui DeMorgan, se obține forma minimă conjunctivă a funcției date.

2.2.2 Metoda Quine-McCluskey

Această metodă pornește de la forma canonică a funcției de minimizat. Metoda are două etape:

- se determină implicantii primi;
- se selectează dintre implicantii primi obținuți doar aceia care acoperă toți termenii canonici ai funcției date și asigură realizarea acestora la un cost minim.

Termenii canonici se compară în felul următor:

- se compară fiecare termen canonic cu toți ceilalți;
- când se găsesc doi termeni care au proprietate de adiacență, variabila redundantă se elimină, obținându-se un termen elementar;
- primul ciclu de comparații se consideră încheiat în momentul în care s-au comparat între ei toți termenii canonici, obținându-se toți implicantii primi posibili;
- se compară între ei pe același criteriu termenii elementari obținuți;
- se vor face atâtea cicluri de comparație câte sunt necesare, pentru a nu mai exista termeni elementari cu proprietatea de adiacență.

Combinațiile indiferente se vor compara cu celelalte, dar nu se vor compara între ele. Dacă nu se compară o combinație, ea ar putea fi conținută de forma minimă a funcției. În primul ciclu de comparații s-au comparat toate combinațiile posibile. Se trece la al doilea ciclu de comparații.

2.3 Rezolvare

2.3.1 Formele minime disjunctive ale funcțiilor

		x_1x_2			
		00	01	11	10
x_3x_4	00	0 0	* 4	0 12	* 8
	01	0 1	* 5	0 13	0 9
	11	0 3	* 7	1 15	* 11
	10	0 2	0 6	1 14	* 10

Fig. 2.5 Metoda diagramei Veitch-Karnaugh pentru a^{FMD}

$$a^{FMD}(x_1, x_2, x_3, x_4) = x_1x_3 \quad (2.1)$$

		x_1x_2			
		00	01	11	10
x_3x_4	00	0 0	* 4	1 12	* 8
	01	0 1	* 5	1 13	1 9
	11	0 3	* 7	0 15	* 11
	10	0 2	1 6	0 14	* 10

Fig. 2.6 Metoda diagramei Veitch-Karnaugh pentru b^{FMD}

$$b^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1x_2 + x_1\bar{x}_3 \quad (2.2)$$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0 0	* 4	1 12	* 8
01	0 1	* 5	1 13	0 9
11	1 3	* 7	0 15	* 11
10	1 2	0 6	0 14	* 10

Fig. 2.7 Metoda diagramei Veitch-Karnaugh pentru c^{FMD}

$$c^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_2x_3 + x_2\bar{x}_3 \quad (2.3)$$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0 0	* 4	0 12	* 8
01	1 1	* 5	1 13	1 9
11	1 3	* 7	1 15	* 11
10	0 2	0 6	0 14	* 10

Fig. 2.8 Metoda diagramei Veitch-Karnaugh pentru d^{FMD}

$$d^{FMD}(x_1, x_2, x_3, x_4) = x_4 \quad (2.4)$$

2.3.2 Formele minime conjunctive ale funcțiilor

		x_1x_2			
		00	01	11	10
x_3x_4	00	0 0	* 4	0 12	* 8
	01	0 1	* 5	0 13	0 9
	11	0 3	* 7	1 15	* 11
	10	0 2	0 6	1 14	* 10

Fig. 2.9 Metoda diagramei Veitch-Karnaugh pentru a^{-FMD}

$$a^{-FMD} = \bar{x}_1 + \bar{x}_3 \quad (2.5)$$

$$a^{FMC} = \overline{a^{-FMD}} = \overline{\bar{x}_1 + \bar{x}_3} = x_1x_3 \quad (2.6)$$

		x_1x_2			
		00	01	11	10
x_3x_4	00	0 0	* 4	1 12	* 8
	01	0 1	* 5	1 13	1 9
	11	0 3	* 7	0 15	* 11
	10	0 2	1 6	0 14	* 10

Fig. 2.10 Metoda diagramei Veitch-Karnaugh pentru b^{-FMD}

$$b^{-FMD} = x_1x_3 + \bar{x}_1\bar{x}_2 \quad (2.7)$$

$$b^{FMC} = \overline{b^{-FMD}} = \overline{x_1x_3 + \bar{x}_1\bar{x}_2} = (\bar{x}_1 + \bar{x}_3)(x_1 + x_2) \quad (2.8)$$

X_1X_2 X_3X_4		00	01	11	10
00	0 0	*	1 12	*	8
01	0 1	*	1 13	0 9	
11	1 3	*	0 15	*	11
10	1 2	0 6	0 14	*	10

Fig. 2.11 Metoda diagramei Veitch-Karnaugh pentru c^{-FMD}

$$c^{-FMD} = \bar{x}_2\bar{x}_3 + x_2x_3 \quad (2.9)$$

$$c^{FMC} = \overline{c^{-FMD}} = \overline{x_2x_3 + \bar{x}_2\bar{x}_3} = (\bar{x}_2 + \bar{x}_3)(x_2 + x_3) \quad (2.10)$$

X_1X_2 X_3X_4		00	01	11	10
00	0 0	*	4	0 12	* 8
01	1 1	*	5	1 13	1 9
11	1 3	*	7	1 15	* 11
10	0 2	0 6	0 14	* 10	

Fig. 2.12 Metoda diagramei Veitch-Karnaugh pentru d^{-FMD}

$$d^{-FMD} = \bar{x}_4 \quad (2.11)$$

$$d^{FMC} = \overline{d^{-FMD}} = \overline{\bar{x}_4} = x_4 \quad (2.12)$$

2.3.3 Metoda Quine-McCluskey pentru primele două funcții logice de ieșire

$$\bullet \quad a^{FCD}(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 = P_{14} + P_{15} = \Sigma(14, 15) \quad (2.13)$$

Combinatii indifferente :

$$P_{14} \rightarrow x_1 x_2 x_3 \bar{x}_4 \rightarrow 1110$$

$$*P_4 \rightarrow \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \rightarrow 0100$$

$$P_{15} \rightarrow x_1 x_2 x_3 x_4 \rightarrow 1111$$

$$*P_5 \rightarrow \bar{x}_1 x_2 \bar{x}_3 x_4 \rightarrow 0101$$

$$*P_7 \rightarrow \bar{x}_1 x_2 x_3 x_4 \rightarrow 0111$$

$$*P_8 \rightarrow x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \rightarrow 1000$$

$$*P_{10} \rightarrow x_1 \bar{x}_2 x_3 \bar{x}_4 \rightarrow 1010$$

$$*P_{11} \rightarrow x_1 \bar{x}_2 x_3 x_4 \rightarrow 1011$$

Grupa	Index	x_1	x_2	x_3	x_4	
1	4*	0	1	0	0	✓
	8*	1	0	0	0	✓
2	5*	0	1	0	1	✓
	10*	1	0	1	0	✓
3	7*	0	1	1	1	✓
	11*	1	0	1	1	✓
	14	1	1	1	0	✓
4	15	1	1	1	1	✓

Tab. 2.1 Cuburi 0-dimensionale

Grupa	Index	x_1	x_2	x_3	x_4	
2	10*,14	1	-	1	0	✓
3	7*,15	-	1	1	1	✓
	11*,15	1	-	1	1	✓
	14,15	1	1	1	-	✓

Tab. 2.2 Cuburi 1-dimensionale

Grupa	Index	x_1	x_2	x_3	x_4	
2	10*,14,11*,15	1	-	1	-	x_1x_3

Tab. 2.4 Cuburi 2-dimensionale

Termeni canonici	$x_1x_2x_3\bar{x}_4$	$x_1x_2x_3x_4$	
x_1x_3	*	*	IPE

Tab. 2.5 Tabelul acoperirilor

$$a^{FMD}(x_1, x_2, x_3, x_4) = x_1x_3 \quad (2.14)$$

$$\bullet \quad b^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1x_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 = P_6 + P_9 + P_{12} + P_{13} = \sum(6,9,12,13) \quad (2.15)$$

Combinatii indifferente :

$$P_6 \rightarrow \bar{x}_1x_2x_3\bar{x}_4 \rightarrow 0110$$

$$P_9 \rightarrow x_1\bar{x}_2\bar{x}_3x_4 \rightarrow 1001$$

$$P_{12} \rightarrow x_1x_2\bar{x}_3\bar{x}_4 \rightarrow 1100$$

$$P_{13} \rightarrow x_1x_2\bar{x}_3x_4 \rightarrow 1101$$

$$*P_4 \rightarrow \bar{x}_1x_2\bar{x}_3\bar{x}_4 \rightarrow 0100$$

$$*P_5 \rightarrow \bar{x}_1x_2\bar{x}_3x_4 \rightarrow 0101$$

$$*P_7 \rightarrow \bar{x}_1x_2x_3x_4 \rightarrow 0111$$

$$*P_8 \rightarrow x_1\bar{x}_2\bar{x}_3\bar{x}_4 \rightarrow 1000$$

$$*P_{10} \rightarrow x_1\bar{x}_2x_3\bar{x}_4 \rightarrow 1010$$

$$*P_{11} \rightarrow x_1\bar{x}_2x_3x_4 \rightarrow 1011$$

Grupa	Index	x_1	x_2	x_3	x_4	
1	4*	0	1	0	0	✓
	8*	1	0	0	0	✓
2	5*	0	1	0	1	✓
	6	0	1	1	0	✓
	9	1	0	0	1	✓
	10*	1	0	1	0	✓
	12	1	1	0	0	✓
3	7*	0	1	1	1	✓
	11*	1	0	1	1	✓
	13	1	1	0	1	✓

Tab. 2.6 Cuburi 0-dimensionale

Grupa	Index	x_1	x_2	x_3	x_4	
1	4*,5*	0	1	0	-	✓
	4*,6	0	1	-	0	✓
	4*,12	-	1	0	0	✓
	8*,9	1	0	0	-	✓
	8*,10*	1	0	-	0	✓
	8*,12	1	-	0	0	✓
2	5*,7*	0	1	-	1	✓
	5*,13	-	1	0	1	✓
	6,7	0	1	1	-	✓
	9,11*	1	0	-	1	✓
	9,13	1	-	0	1	✓
	10*,11*	1	0	1	-	✓
	12,13	1	1	0	-	✓

Tab. 2.7 Cuburi 1-dimensionale

Grupa	Index	x_1	x_2	x_3	x_4	
1	$4^*, 6, 5^*, 7^*$	0	1	-	-	$\bar{x}_1 x_2$
	$4^*, 5^*, 6, 7^*$	0	1	-	-	
	$4^*, 12, 5^*, 13$	-	1	0	-	$x_2 \bar{x}_3$
	$4^*, 5^*, 12, 13$	-	1	0	-	
	$8^*, 10^*, 9, 11^*$	1	0	-	-	$x_1 \bar{x}_2$
	$8^*, 9, 10^*, 11^*$	1	0	-	-	
	$8^*, 9, 12, 13$	1	-	0	-	$x_1 \bar{x}_3$
	$8^*, 12, 9, 13$	1	-	0	-	

Tab. 2.8 Cuburi 2-dimensionale

Term. canonici Implicanți primi	$\bar{x}_1 x_2 x_3 \bar{x}_4$	$x_1 \bar{x}_2 \bar{x}_3 x_4$	$x_1 x_2 \bar{x}_3 \bar{x}_4$	$x_1 x_2 \bar{x}_3 x_4$	
$\bar{x}_1 x_2$	*				IPE
$x_2 \bar{x}_3$			*	*	
$x_1 \bar{x}_2$			*		
$x_1 \bar{x}_3$		*	*	*	IPE

Tab. 2.9 Tabelul acoperirilor

$$b^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 + x_1 \bar{x}_3 \quad (2.16)$$

3. Exercițiul 3

3.1 Enunțul problemei

Să se implementeze fiecare funcție logică, independent, numai cu porți logice ȘI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).

3.2 Considerații teoretice

3.2.1 Implementarea funcțiilor folosind porți logice ȘI-NU în tehnologie TTL

Pentru a implementa funcțiile logice cu porți logice ȘI-NU acestea trebuie aduse la o formă adecvată. Procedeu care se va folosi este negarea de două ori a funcției și folosirea formulelor lui De Morgan. Se vor folosi circuite integrate din producția Texas Instruments din seria 74LS(vezi Anexa 1).

3.3 Rezolvare

$$a^{FMD} = x_1 x_3 \quad (3.1)$$

$$\overline{a^{FMD}} = \overline{x_1 x_3} \quad (3.2)$$

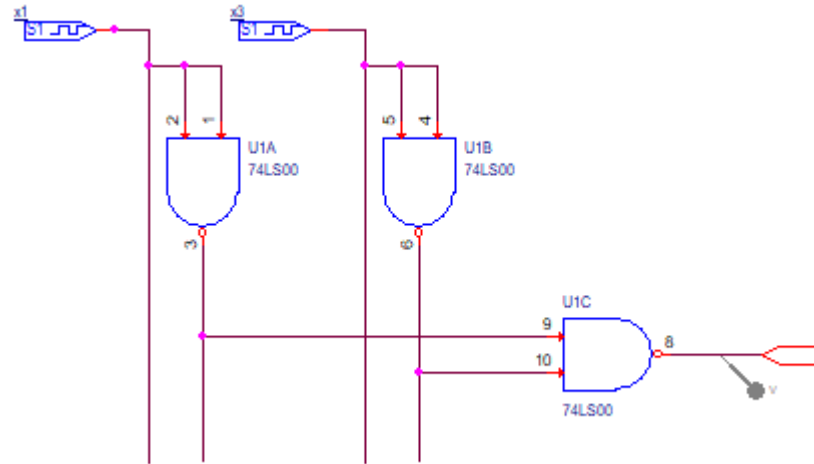


Fig. 3.1 Reprezentarea funcției $\overline{a^{FMD}}$ cu porți logice ȘI-NU

S-a folosit următorul circuit integrat: 1x74LS00(-1)

$$b^{FMD} = \bar{x}_1 x_2 + x_1 \bar{x}_3 \quad (3.3)$$

$$\overline{b^{FMD}} = \overline{\bar{x}_1 x_2 + x_1 \bar{x}_3} = \overline{\bar{x}_1 x_2} \cdot \overline{x_1 \bar{x}_3} \quad (3.4)$$

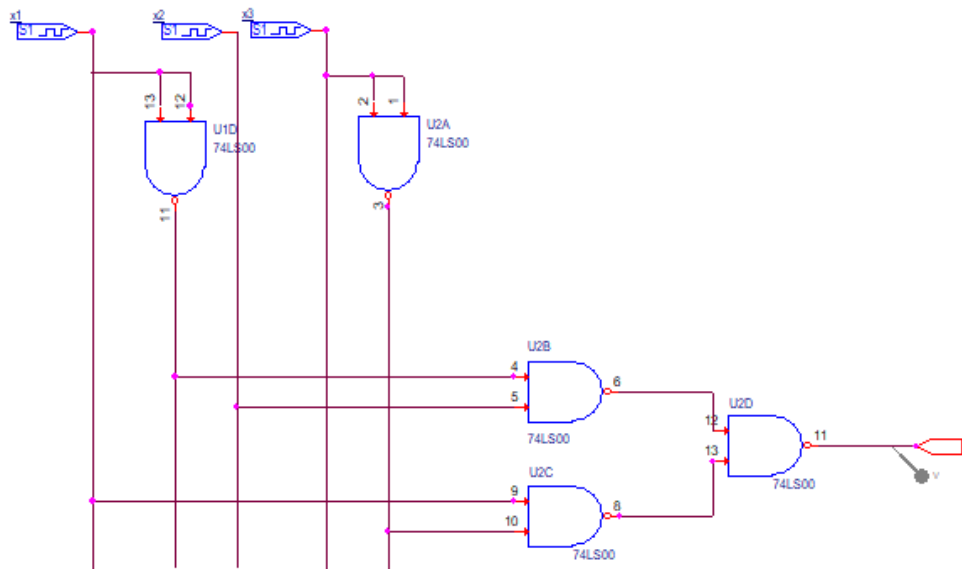


Fig. 3.2 Reprezentarea funcției $\overline{b^{FMD}}$ cu porți logice ȘI-NU

S-a folosit următorul circuit integrat: 2x74LS00(-3)

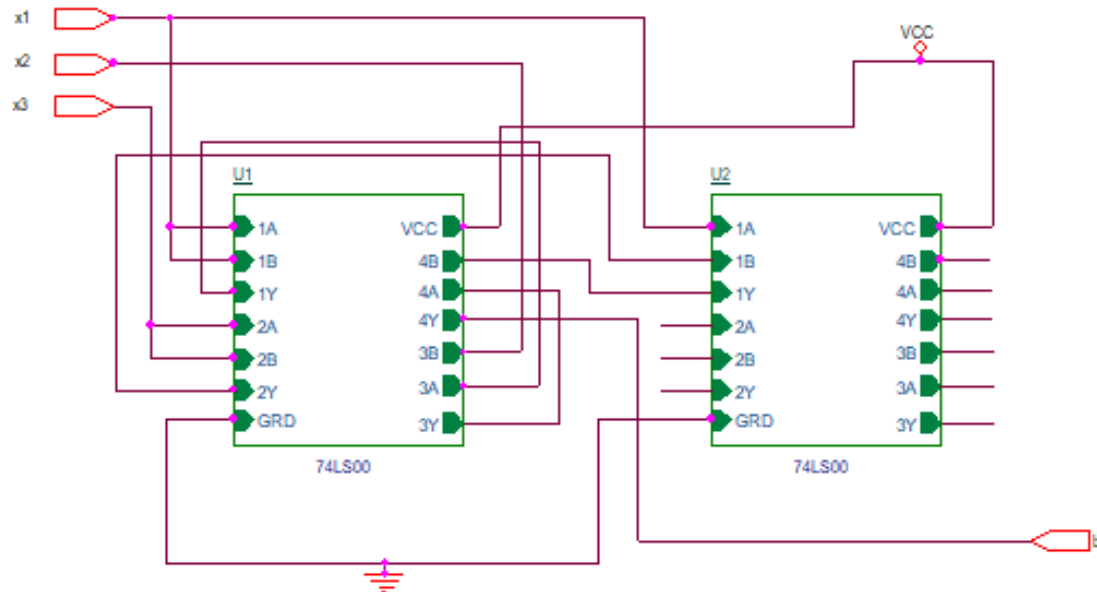


Fig. 3.3 Reprezentarea funcției $\overline{\overline{b^{FMD}}}$ cu circuite integrate 74LS00, tehnologia TTL

$$c^{FMD} = \bar{x}_2 x_3 + x_2 \bar{x}_3 \quad (3.5)$$

$$\overline{\overline{c^{FMD}}} = \overline{\overline{\bar{x}_2 x_3 + x_2 \bar{x}_3}} = \overline{\bar{x}_2 x_3} \cdot \overline{x_2 \bar{x}_3} \quad (3.6)$$

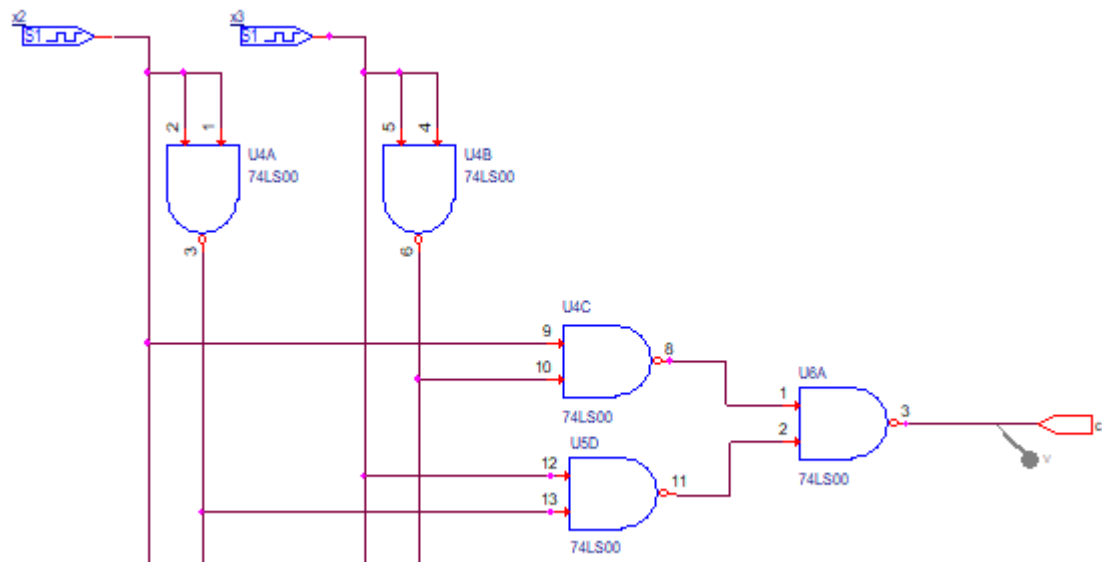


Fig. 3.4 Reprezentarea funcției $\overline{\overline{c^{FMD}}}$ cu porți logice ȘI-NU

S-a folosit următorul circuit integrat: 2x74LS00(-3)

$$d^{FMD} = x_4 \quad (3.7)$$

$$\overline{\overline{d^{FMD}}} = \overline{x_4} \quad (3.8)$$



Fig. 3.5 Reprezentarea funcției $\overline{\overline{d^{FMD}}}$ cu porți logice ȘI-NU

4. Exercițiul 4

4.1 Enunțul problemei

Să se implementeze ansamblul funcțiilor logice numai cu porți logice ȘI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).

4.2 Rezolvare

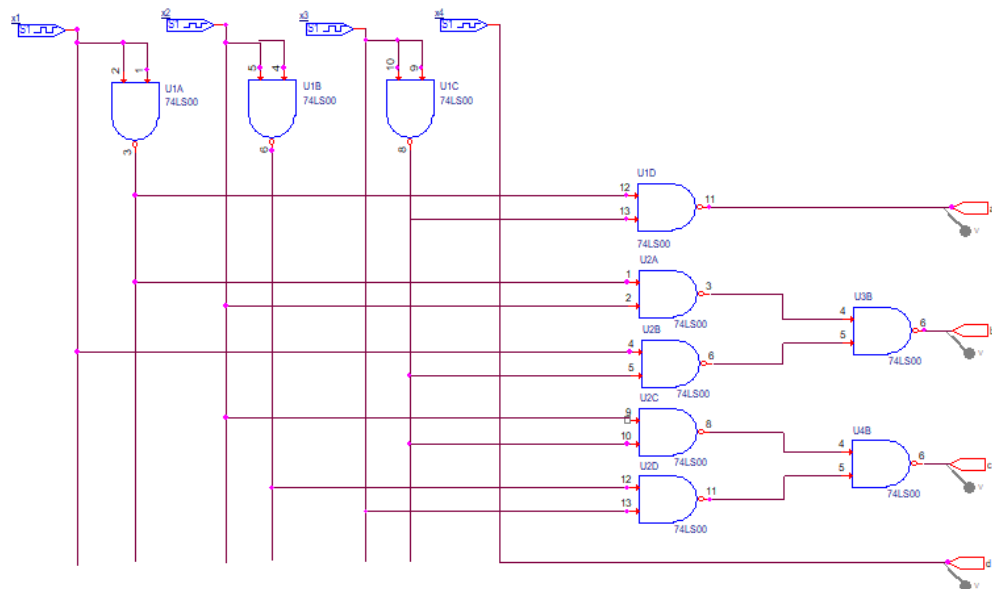


Fig. 4.1 Reprezentarea ansamblului funcțiilor logice cu porți ȘI-NU

S-a folosit următorul circuit integrat: 3x74LS00(-2)

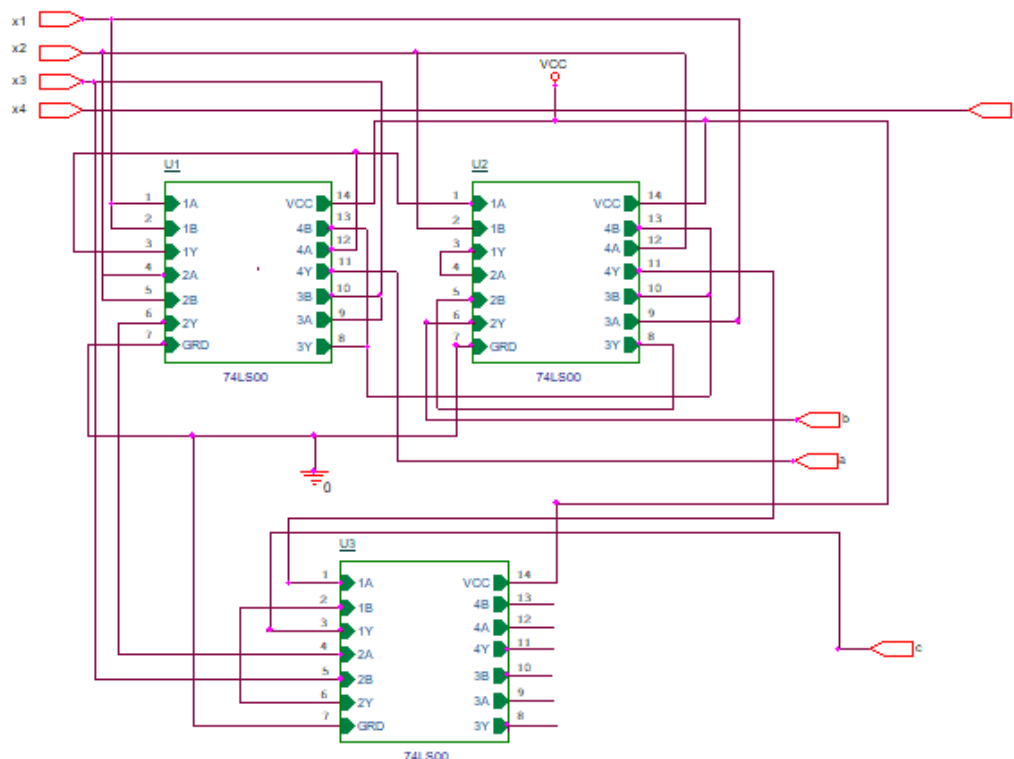


Fig. 4.2 Reprezentarea funcțiilor cu circuite integrate 74LS00, tehnologia TTL

5. Exercițiul 5

5.1 Enunțul problemei

Să se implementeze ansamblul funcțiilor logice în următoarea variantă: primele două funcții logice de ieșire cu porți logice ȘI-NU (circuite integrate TTL), iar următoarele două cu porți logice SAU-NU (circuite integrate CMOS).

5.2 Considerații teoretice

5.2.1 Implementarea funcției SAU-NU și funcției ȘI-NU

Pentru implementare cu funcții SAU-NU se pleacă de la forma minimă obținută pentru realizarea implementării cu funcții ȘI-NU aplicând formulele lui De-Morgan.

5.3 Rezolvare

$$\overline{a^{FMD}} = \overline{x_1 x_3} \quad (5.1)$$

$$\overline{b^{FMD}} = \overline{\overline{x_1 x_2} + x_1 \overline{x_3}} = \overline{\overline{x_1 x_2}} \cdot \overline{x_1 \overline{x_3}} \quad (5.2)$$

$$\overline{c^{FMD}} = \overline{\overline{x_2 x_3} + x_2 \overline{x_3}} = \overline{\overline{x_2 x_3}} \cdot \overline{x_2 \overline{x_3}} = \overline{(x_2 + \overline{x_3})} \cdot \overline{(x_2 + x_3)} = \overline{(x_2 + \overline{x_3})} + \overline{(x_2 + x_3)} \quad (5.3)$$

$$\overline{d^{FMD}} = \overline{x_4} \quad (5.4)$$

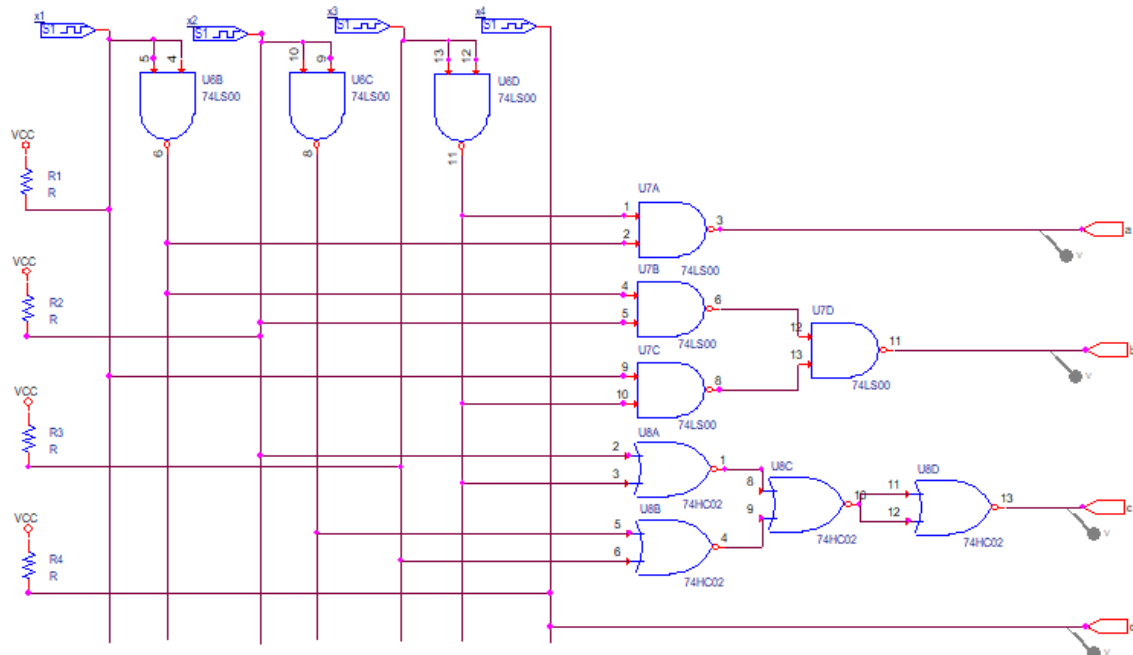


Fig. 5.1 Reprezentarea funcțiilor cu circuite integrate 74LS00 și 74HC02 (tehnologia TTL și CMOS)

S-au folosit următoarele circuite integrate: 2x74LS00(-1); 1x74HC02(-)

6. Exercițiul 6

6.1 Enunțul problemei

Să se implementeze ansamblul funcțiilor logice cu MUX-uri de 8, respectiv 16 căi (se vor utiliza circuite integrate realizate în tehnologia TTL).

6.2 Considerații teoretice

6.2.1 Implementarea schemei cu MUX-uri în tehnologie TTL

Un circuit de multiplexare este un circuit logic combinațional care, în cazul general, are 2^n intrări de date ($I_{2^n-1} \dots I_2 I_1 I_0$), n intrări de selecție ($S_0 S_1 \dots S_{n-1}$) și o ieșire (Z). Expresia ieșirii Z la un moment dat este dată de intrare I_k , $k=0, \dots, 2^n-1$ unde k reprezintă echivalentul zecimal al numărului binar dat de stările 1 și 0 ale intrărilor de selecție: $k=S_{n-1}, S_{n-2}, \dots, S_1, S_0$.

6.2.2 Implementarea funcției *a* cu 2xMUX-uri de 8 căi conectate în paralel

La implementarea unei funcții logice cu două multiplexoare de 8 căi legate în paralel variabilele funcției de implementat se aplică astfel : prima variabilă (cea mai semnificativă) se aplică pe intrarea de strobare la primul multiplexor așa cum este ea iar la al doilea multiplexor negată, celelalte variabile aplicându-se în ordinea semnificativității lor pe intrările de date. Pe intrările de selecție se vor aplica semnale corespunzătoare echivalențelor zecimale prezente în expresia funcției urmând ca ieșirile celor două multiplexoare să fie legate într-o poartă SAU.

6.2.3 Implementarea funcției *b* cu MUX de 8 căi și o variabilă aplicată pe intrările de date

La implementarea unei funcții cu un multiplexor de 8 căi și o variabilă aplicată pe intrările de date se pleacă de la tabelul de adevăr al funcției de implementat separându-se variabila cea mai puțin semnificativă. Astfel din cele n variabile se separă $n-1$ având ponderile cele mai mari și se aplică pe intrările de selecție ale multiplexorului. Astfel se obțin termeni canonici de $n-1$ variabile. Pentru a obține însă termeni canonici de n variabile prezenți în expresia funcției de implementat trebuie adăugată variabila separată. Acest lucru se realizează prin aplicarea variabilei pe intrările de date ale MUX-ului.

6.2.4 Implementarea funcțiilor *c* și *d* cu MUX de 16 căi

La implementarea cu multiplexoare de 16 căi variabilele funcției se vor aplica pe intrările de date. Deoarece circuitul are ieșiri active în 0 logic rezultă că pe intrările de selecție se vor lega la 1 logic termenii canonici care nu sunt prezenți în expresia funcției iar termenii canonici prezenți în expresia funcției se vor lega la 0 logic.

6.3 Rezolvare

- $$a^{FCD}(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 = P_{14} + P_{15} = \Sigma(14, 15) \quad (6.1)$$

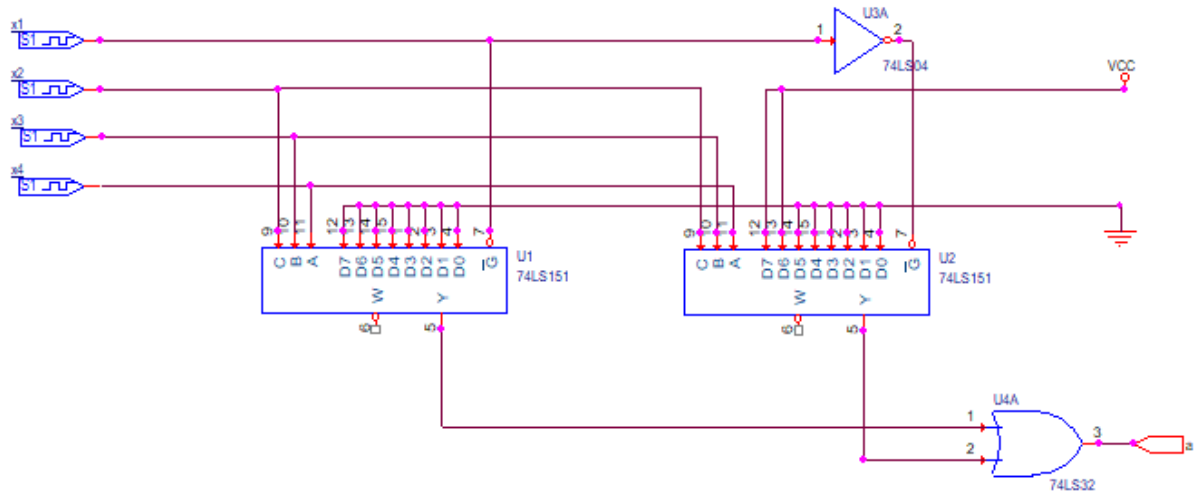


Fig 6.1 Implementarea funcției a cu 2xMUX-uri de 8 căi conectate în paralel

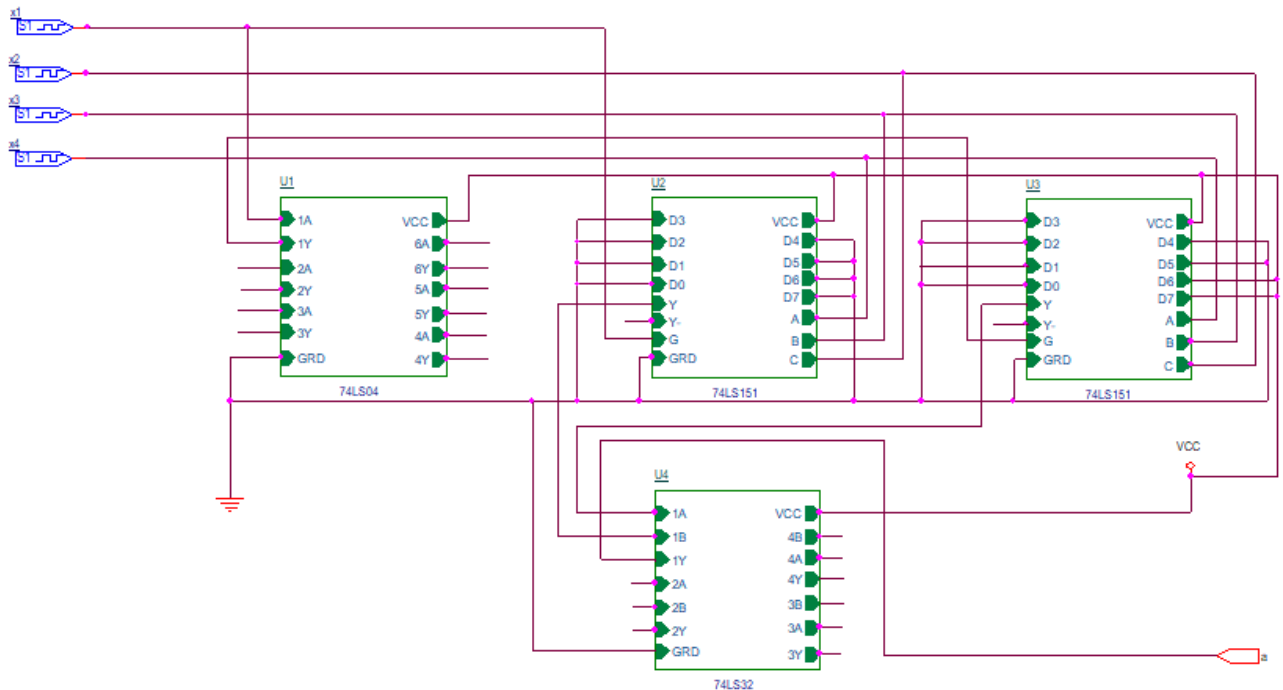


Fig. 6.2 Reprezentarea funcției a cu circuite integrate (tehnologia TTL)

$$\bullet \quad b^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 x_2 \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 = P_6 + P_9 + P_{12} + P_{13} = \sum(6, 9, 12, 13) \quad (6.2)$$

Nr ctr	x_1	x_2	x_3	x_4	b
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4*	0	1	0	0	*
5*	0	1	0	1	*
6	0	1	1	0	1
7*	0	1	1	1	*
8*	1	0	0	0	*
9	1	0	0	1	1
10*	1	0	1	0	*
11*	1	0	1	1	*
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Tab. 6.1 Reprezentarea funcției b prin tabel de adevăr

Pentru: $x_1 x_2 x_3 = 000$, $b=0$ oricare ar fi $x_4 \rightarrow D_0 = 0$
 $x_1 x_2 x_3 = 001$, $b=0$ oricare ar fi $x_4 \rightarrow D_1 = 0$
 $x_1 x_2 x_3 = 011$, $b=1$ pentru $x_4 = 0 \rightarrow D_2 = \bar{x}_4$
 $x_1 x_2 x_3 = 100$, $b=1$ pentru $x_4 = 1 \rightarrow D_3 = x_4$
 $x_1 x_2 x_3 = 110$, $b=1$ oricare ar fi $x_4 \rightarrow D_4 = 1$
 $x_1 x_2 x_3 = 111$, $b=0$ oricare ar fi $x_4 \rightarrow D_5 = 0$

$$b^{FCD}(x_1, x_2, x_3, x_4) = P'_0 \cdot 0 + P'_1 \cdot 0 + P'_3 \cdot \bar{x}_4 + P'_4 \cdot x_4 + P'_6 \cdot 1 + P'_7 \cdot 0 \quad (6.3)$$

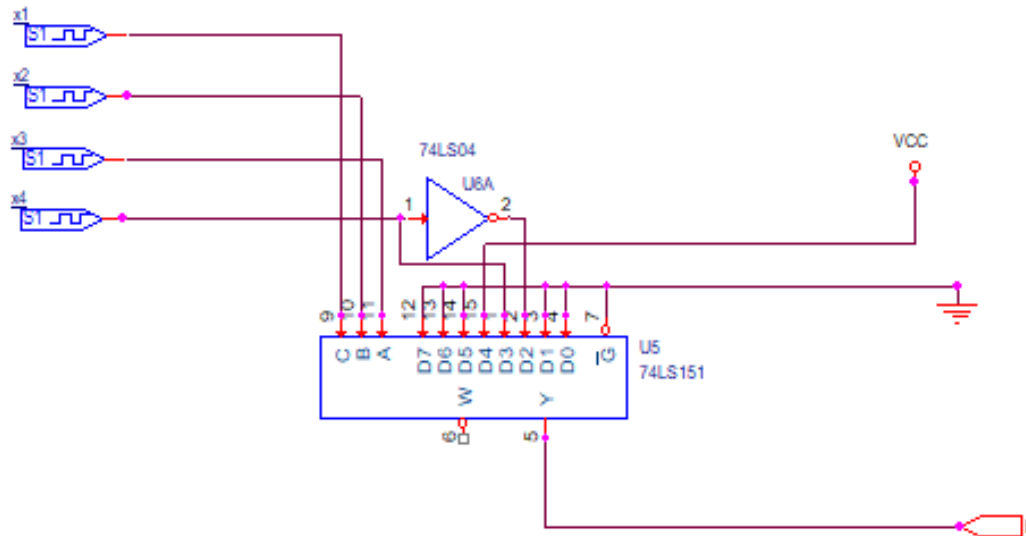


Fig. 6.3 Implementarea funcției **b** cu MUX de 8 căi și o variabilă aplicată pe intrările de date

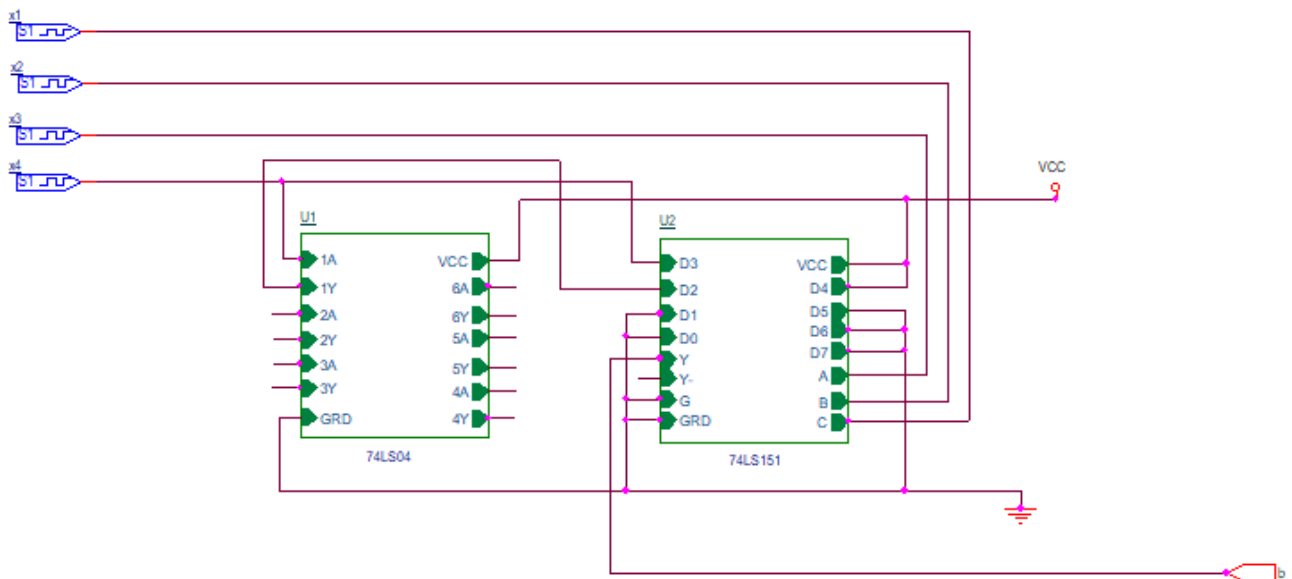


Fig. 6.4 Reprezentarea funcției **b** cu circuite integrate (tehnologia TTL)

- $\bar{c}^{FCD}(x_1, x_2, x_3, x_4) = P_0 + P_1 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9 + P_{10} + P_{11} + P_{14} + P_{15}$
 $= \sum(0, 1, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15)$ (6.4)

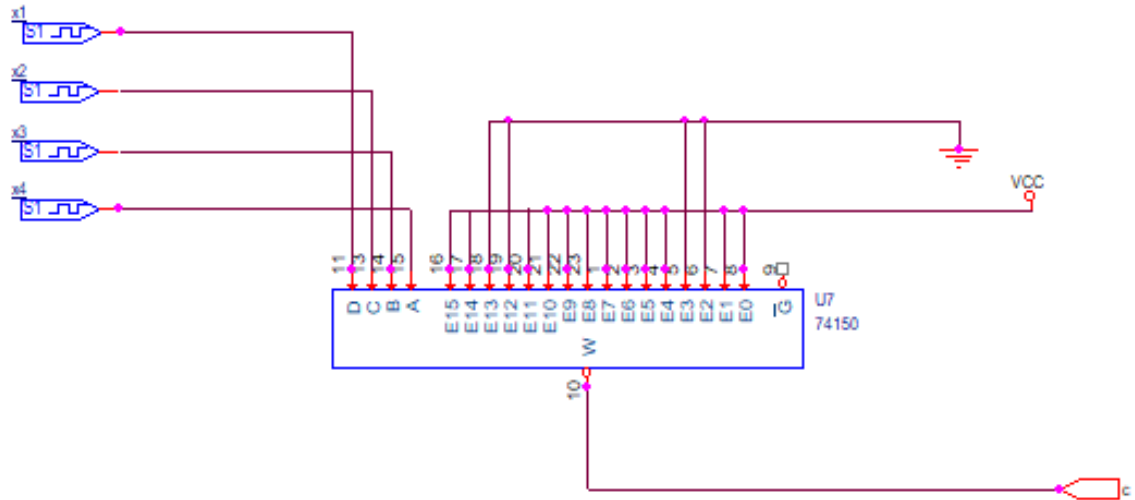


Fig 6.5 Implementarea funcției c cu MUX de 16 căi

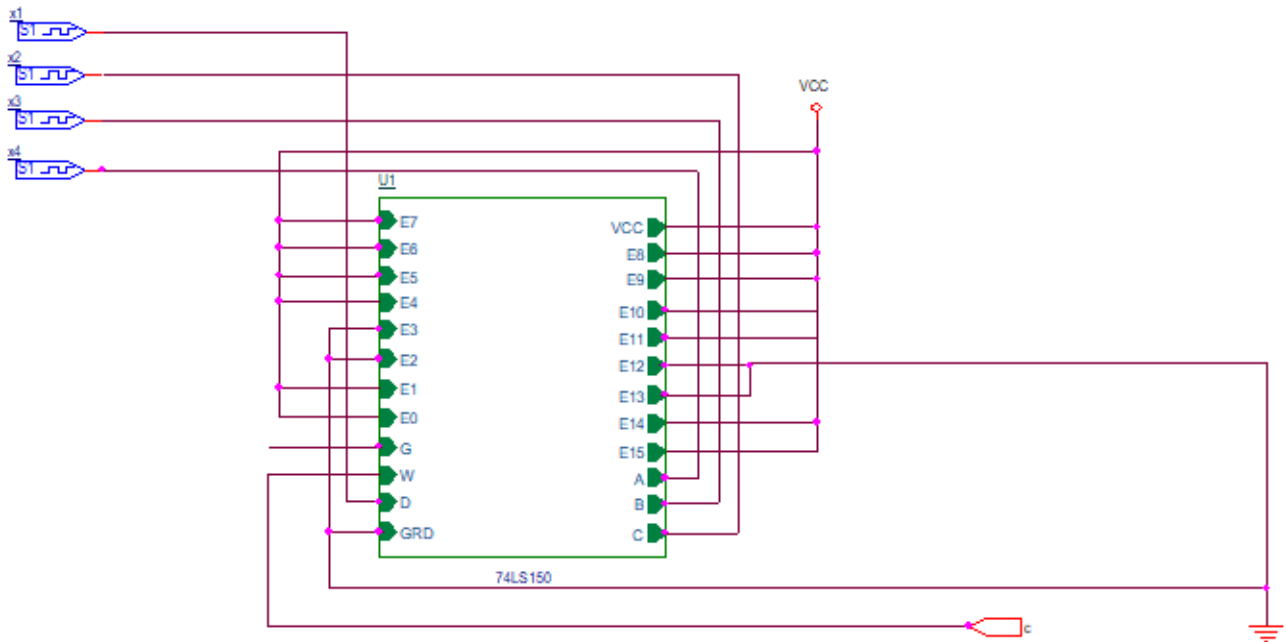


Fig. 6.6 Reprezentarea funcției c cu circuite integrate (tehnologia TTL)

- $\bar{d}^{FCD}(x_1, x_2, x_3, x_4) = P_0 + P_2 + P_4 + P_5 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{14} = \Sigma(0,2,4,5,6,7,8,10,11,12,14)$ (6.5)

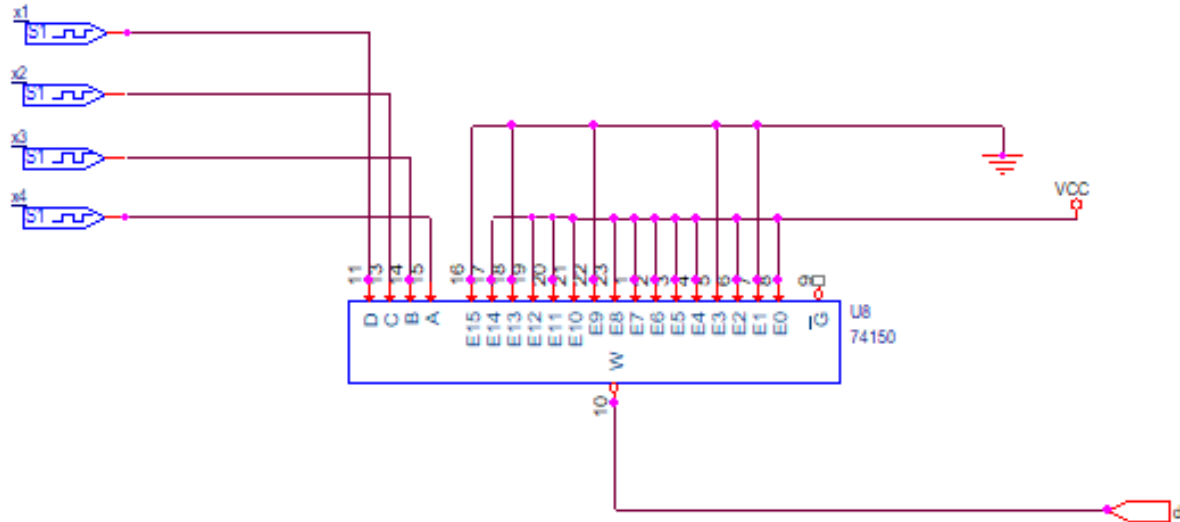


Fig 6.7 Implementarea funcției d cu MUX de 16 căi

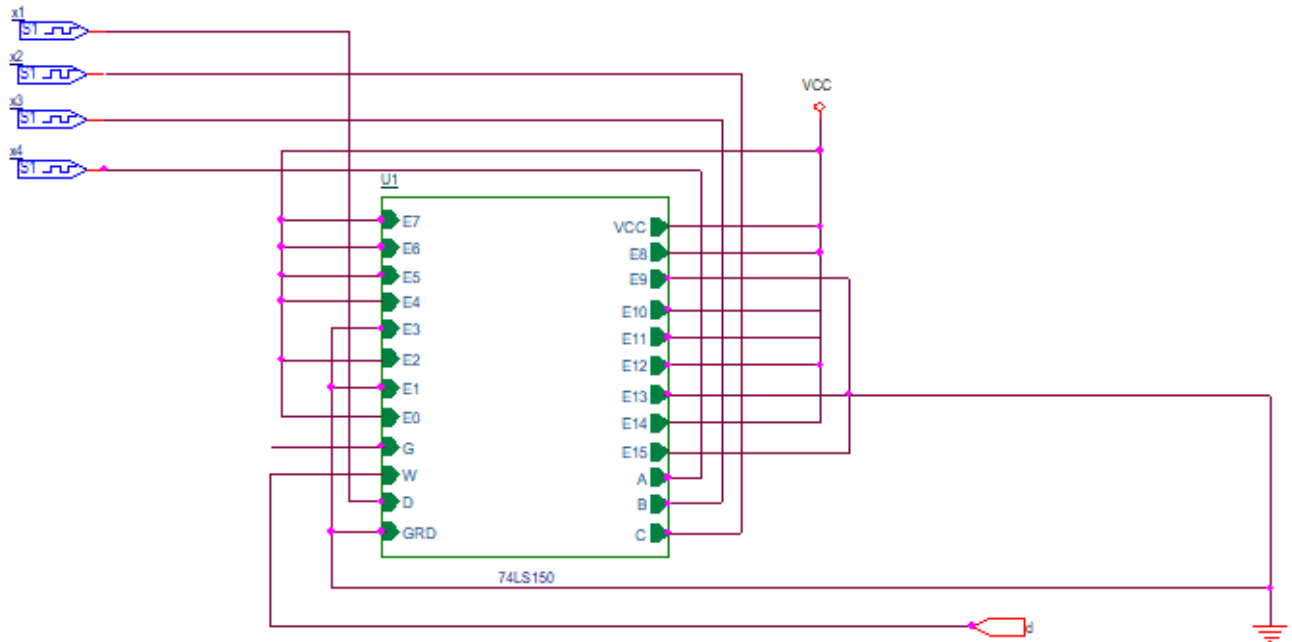


Fig. 6.8 Reprezentarea funcției d cu circuite integrate (tehnologia TTL)

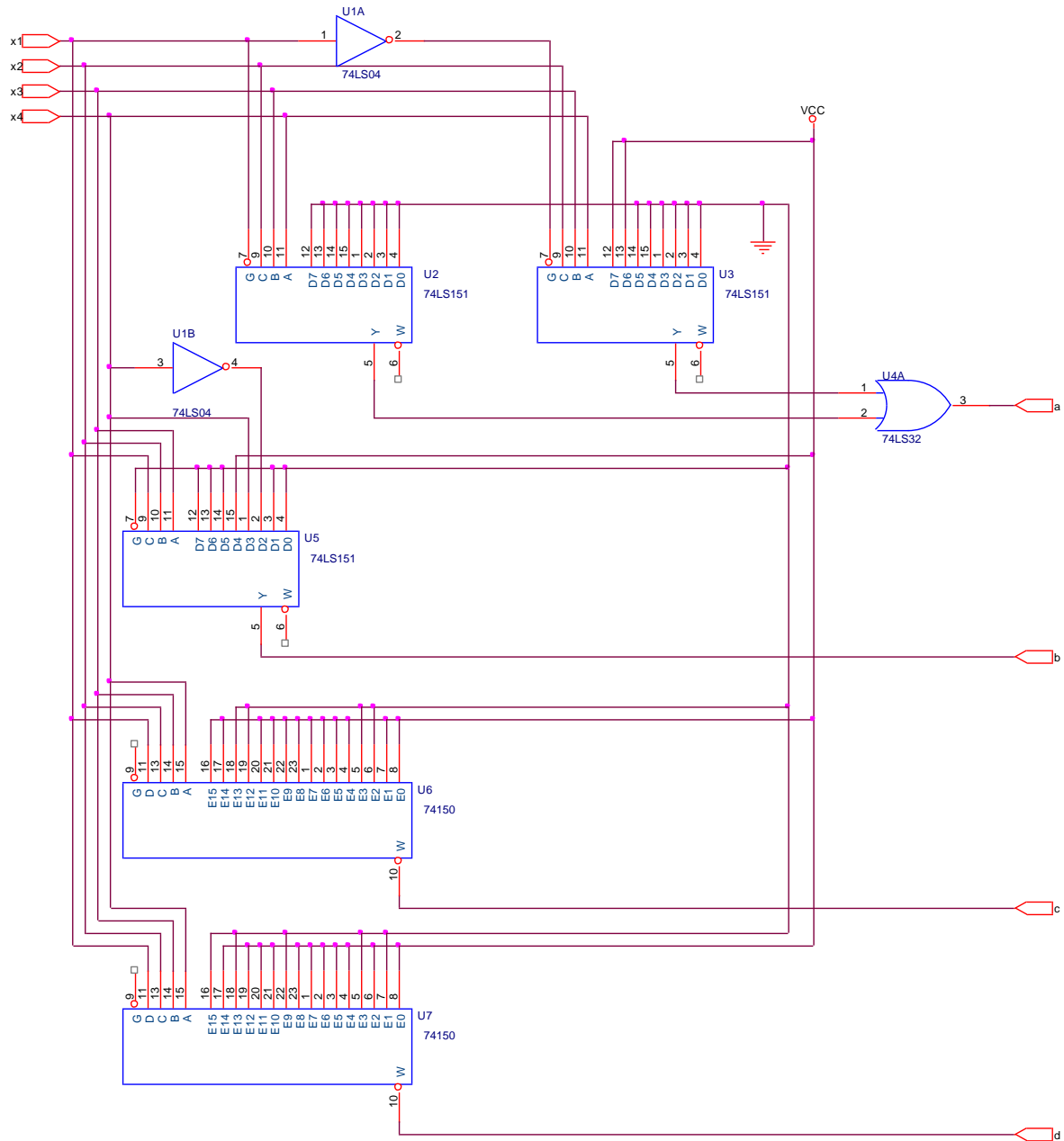


Fig. 6.9 Reprezentarea ansamblului de funcții cu MUX de 8 și 16 căi

S-au folosit următoarele circuite integrate: 1x74LS04(-4); 1x74LS32(-3); 2x74LS150; 3x74LS151

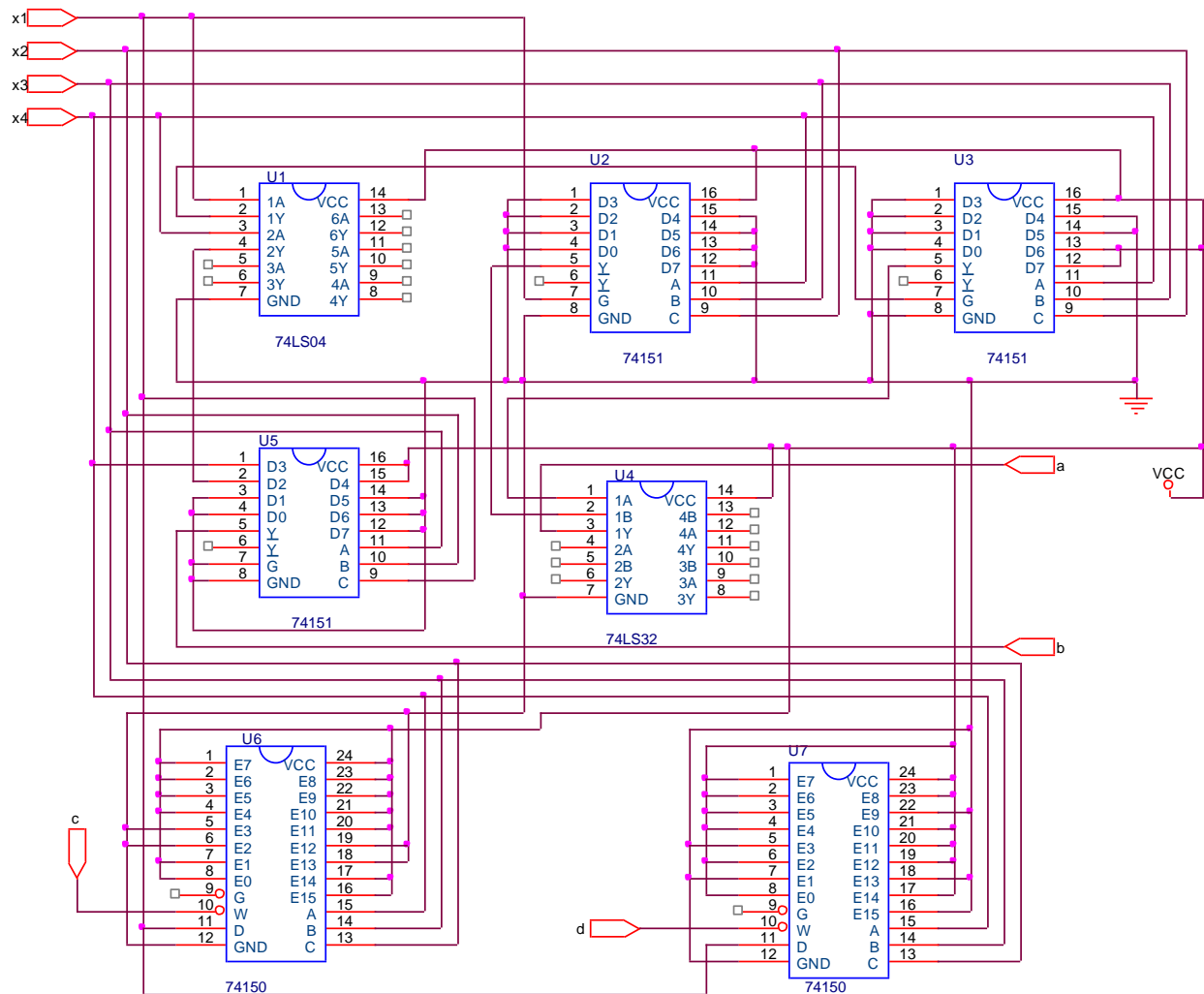


Fig. 6.10 Reprezentarea ansamblului de funcții cu circuite integrate (tehnologia TTL)

7. Exercițiul 7

7.1 Enunțul problemei

Să se implementeze ansamblul funcțiilor logice cu DMUX-uri de 8, respectiv 16 căi și porți logice ȘI-NU în prima variantă, respectiv ȘI în a doua variantă (se vor utiliza circuite integrate realizate în tehnologia CMOS).

7.2 Considerații teoretice

Demultiplexoarele sunt circuite combinaționale care, în cazul general au o intrare de date I, n intrări de selecție S_0, S_1, \dots, S_{n-1} și 2^n ieșiri $Z_0, Z_1, \dots, Z_{2^n-1}$.

Pentru implementarea funcțiilor cu DMUX variabilele funcției se vor aplica pe intrările de selecție în raport cu ponderile acestora.

Pentru realizarea nivelului logic SAU se pot folosi porți:

1. ȘI-NU, în acest caz se leagă la intrările porților ȘI-NU ieșirile DMUX corespunzătoare termenilor canonici prezentați în expresia funcției de implementat.
2. ȘI, în acest caz se leagă la intrările porților ȘI ieșirile DMUX corespunzătoare termenilor canonici care nu apar în expresia funcției (se vor considera funcțiile negate).

În cazul în care funcția are, în caz general, n variabile și se impune să se implementeze un DMUX $1:2^{n-1}$ se va separa variabila cu ponderea cea mai mare, iar cele $n-1$ variabile de stare se vor aplica pe intrările DMUX în raport cu ponderile lor. Deoarece la ieșirile acestor circuite se obțin termeni canonici de $n-1$ variabile, iar în forma în care a fost funcția de implementat sunt termeni canonici de n variabile, este necesar să se adauge și variabila lipsă. Acest lucru se realizează prin intermediul unei rețele cu porți logice. La intrările unei porți ȘI se vor aplica ieșirile DMUX-ului corespunzătoare termenilor canonici care nu apar în expresia funcției (se ia în considerare negata funcției).

7.3 Rezolvare

7.3.1 Implementarea funcției a cu DMUX de 8 căi și rețea de porți logice ȘI-NU

La implementarea unei funcții cu DMUX-uri de 8 căi și rețea de porți logice dintre cele n variabile ale funcției se separă $n-1$ având ponderile cele mai mici și se aplică pe intrările DMUX-ului. Astfel la ieșirile acestuia se obțin termeni canonici de $n-1$ variabile. Pentru a obține termeni canonici de n variabile trebuie adăugată variabila separată. Acest lucru se face în exteriorul DMUX-ului de obicei printr-o rețea de porți logice. Deoarece ieșirile DMUX-ului sunt active în 0 logic rezultă că pentru realizarea nivelului SAU se folosesc porți logice ȘI-NU ale funcției negate.

$$\begin{aligned} a^{FCD}(x_1, x_2, x_3, x_4) &= P_{14} + P_{15} = x_1 x_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 = x_1 (x_2 x_3 \bar{x}_4) + x_1 (x_2 x_3 x_4) = \overline{x_1 P'_6} + \overline{x_1 P'_7} \\ &= \overline{x_1 P'_6} \cdot \overline{x_1 P'_7} = (\bar{x}_1 + \bar{P}'_6) \cdot (\bar{x}_1 + \bar{P}'_7) = \bar{x}_1 + \bar{x}_1 \bar{P}'_7 + \bar{x}_1 \bar{P}'_6 + \bar{P}'_6 \bar{P}'_7 = x_1 \cdot \bar{x}_1 \bar{P}'_7 \cdot \bar{x}_1 \bar{P}'_6 \cdot \bar{P}'_6 \bar{P}'_7 \end{aligned} \quad (7.1)$$

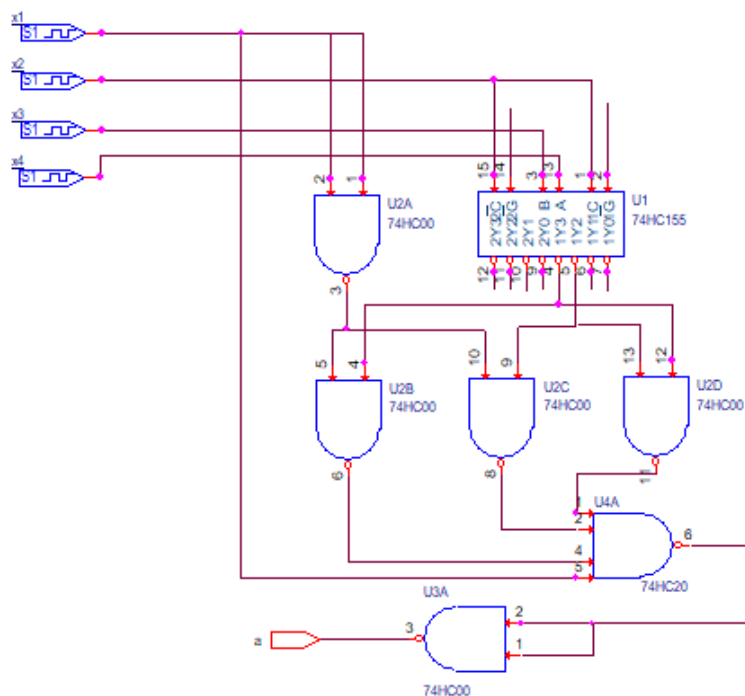


Fig. 7.1 Reprezentarea funcției a cu DMUX de 8 căi și porți logice ȘI-NU

S-au folosit următoarele circuite integrate: 2x74HC00(-3); 1x74HC20(-1); 1x74HC155

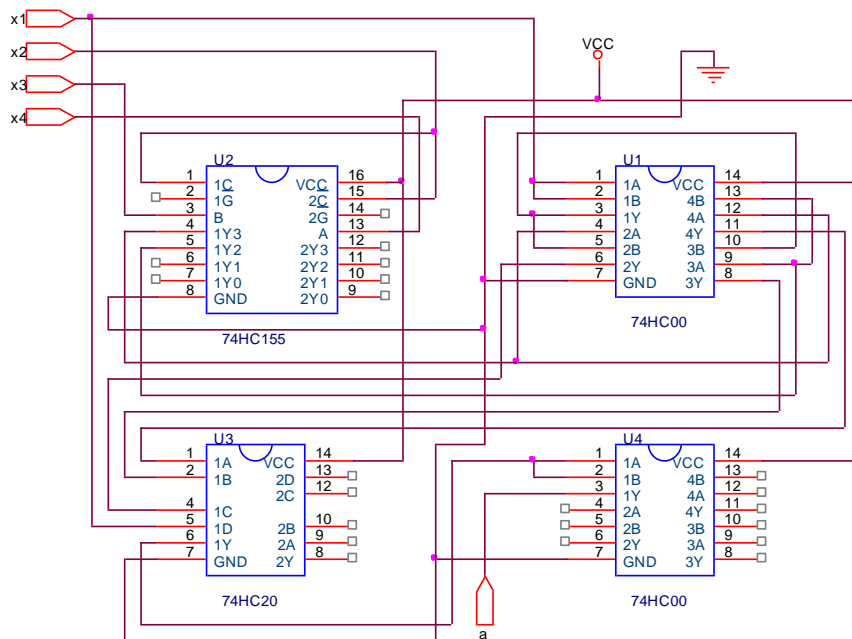


Fig. 7.2 Reprezentarea funcției a cu circuite integrate (tehnologia CMOS)

7.3.2 Implementare funcției b cu DMUX-uri de 8 căi conectate în paralel

La implementarea cu demultiplexoare de 8 căi cele două demultiplexoare se conectează în paralel pe intrările de selecție respectiv de strobare cu variabila cea mai semnificativă aplicată pe intrările de strobare pentru primul demultiplexor așa cum este ea și pentru al doilea negată. Deoarece demultiplexorul are ieșiri active în 0 logic rezultă că pentru a realiza însumarea termenilor canonici prezenți în expresia funcției se pleacă de la negata acestora aplicată prin porți ȘI-NU, conform regulilor lui De-Morgan.

$$b^{FCD}(x_1, x_2, x_3, x_4) = P_6 + P_9 + P_{12} + P_{13} = \sum(6, 9, 12, 13) \quad (7.2)$$

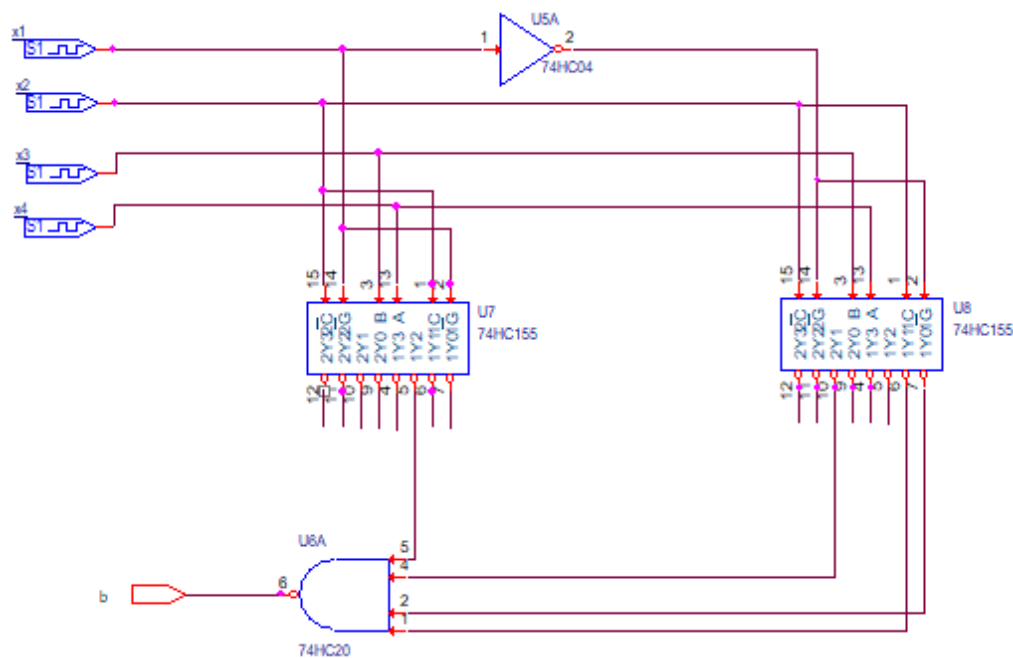


Fig. 7.3 Reprezentarea funcției b cu DMUX de 8 căi și porți logice ȘI-NU

S-au folosit următoarele circuite integrate: 1x74HC04(-3); 1x74HC20(-1); 2x74HC155

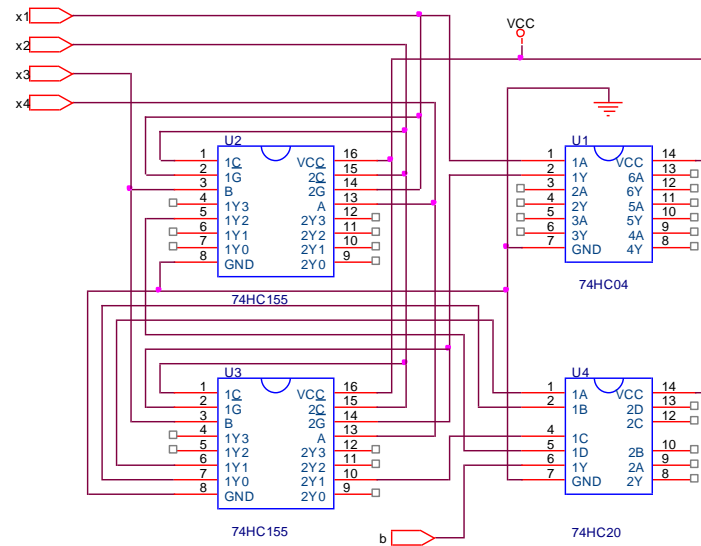


Fig. 7.4 Reprezentarea funcției b cu circuite integrate (tehnologia CMOS)

7.3.3 Implementare funcției c cu DMUX de 16 căi și porți logice ȘI-NU

$$c^{FCD}(x_1, x_2, x_3, x_4) = P_2 + P_3 + P_{12} + P_{13} = \sum(2, 3, 12, 13) \quad (7.3)$$

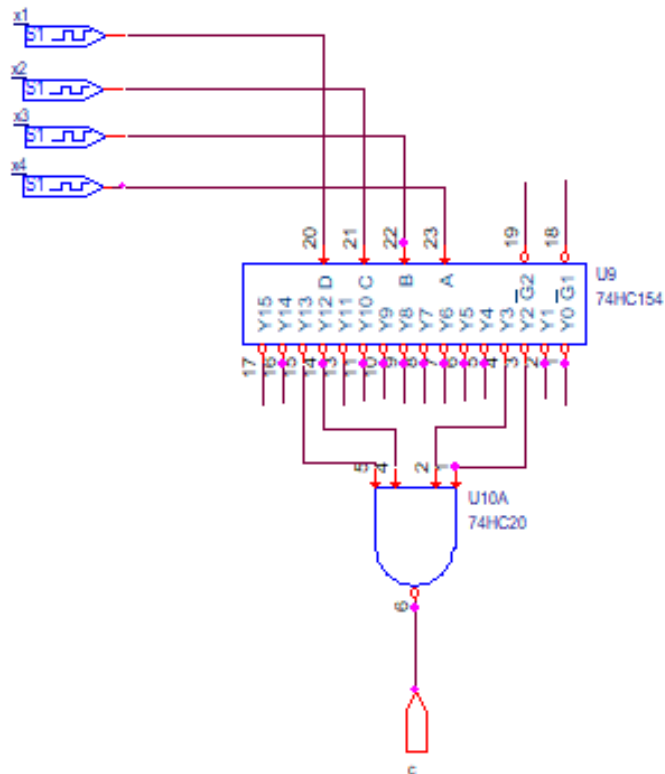


Fig. 7.5 Reprezentarea funcției c cu DMUX de 16 căi și porți logice ȘI-NU

S-au folosit urmatoarele circuite integrate: 1x74HC20(-1); 1x74HC154

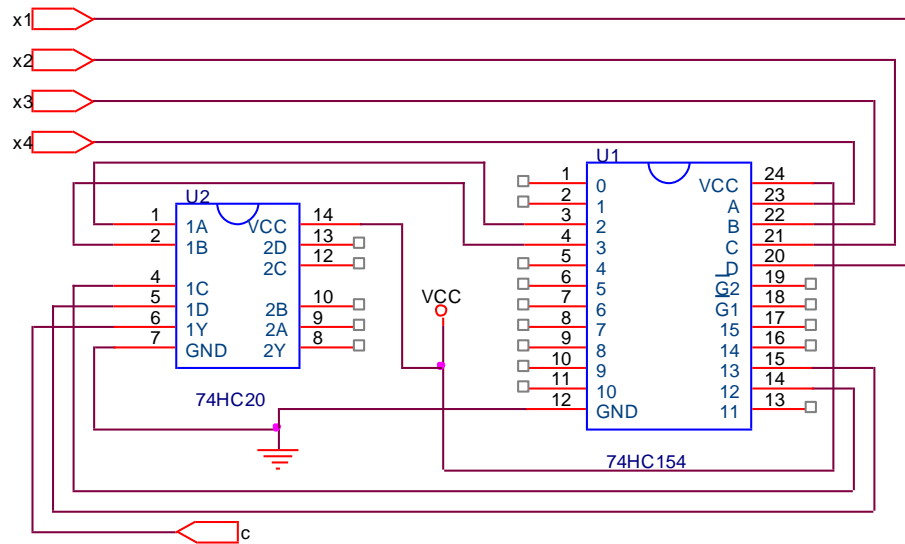


Fig. 7.6 Reprezentarea funcției c cu circuite integrate (tehnologia CMOS)

7.3.4 Implementarea funcției d cu DMUX de 16 căi și porți logice ȘI

$$\bar{d}^{FCD}(x_1, x_2, x_3, x_4) = P_0 + P_2 + P_4 + P_5 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{14} = \sum(0, 2, 4, 5, 6, 7, 8, 10, 11, 12, 14) \quad (7.4)$$

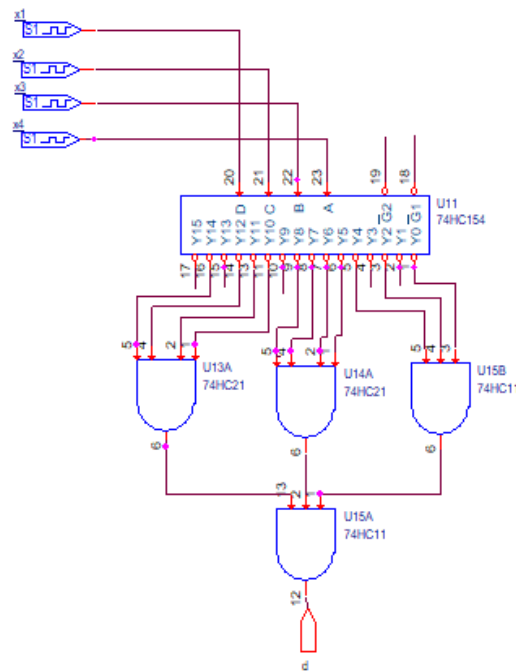


Fig. 7.7 Reprezentarea funcției d cu DMUX de 16 căi și porți logice ȘI

S-au folosit urmatoarele circuite integrate: 1x74HC21(-); 1x74HC154; 1x74HC11(-1)

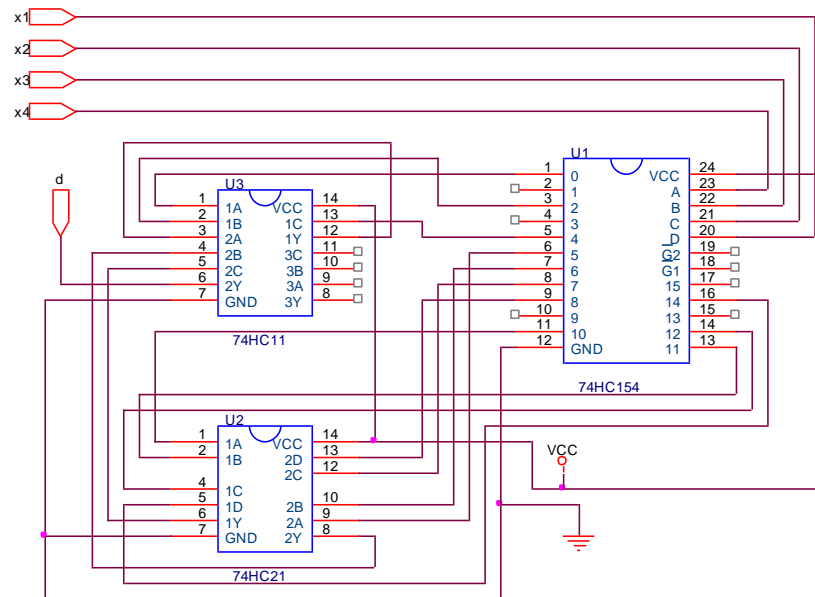


Fig. 7.8 Reprezentarea funcției d cu circuite integrate (tehnologia CMOS)

8. Exercițiul 8

8.1 Enunțul problemei

Să se calculeze timpii de propagare „intrare-ieșire”, pentru toate schemele logice obținute.

8.2 Considerații teoretice

Notat cu t_p și măsurat în ns, reprezintă timpul scurs în intervalul în care se modifică o variabilă la intrare și răspunsul este furnizat la ieșire unei porți logice. Forma de calcul pentru timpul de propagare este:

$$t_p = \frac{1}{2} (t_{pLH} + t_{pHL}) \quad (8.1)$$

8.3 Rezolvare

C.I.	Tehnologia	t_{pLH} [ns]	t_{pHL} [ns]	$I_{CCH}(TTL)$ $I_{OH}(CMOS)$ [mA]	$I_{CCL}(TTL)$ $I_{OL}(CMOS)$ [mA]	P_d [mW]
74LS00	TTL	9	10	0.8	2.4	
74LS04	TTL	9	10	1.2	3.6	
74LS32	TTL	14	14	3.1	4.9	
74LS150	TTL	23	22			200
74LS151	TTL	27	18			30
74HC00	CMOS	9	9			
74HC02	CMOS	9	9			
74HC04	CMOS	9	9			
74HC11	CMOS	10	10			
74HC20	CMOS	14	14			
74HC21	CMOS	14	14			
74HC154	CMOS	19	19			
74HC155	CMOS	15	15			

Tab. 8.1 Date de catalog ale circuitelor integrate folosite

Pentru 1/4 74LS00: $t_p = \frac{9+10}{2} = 9,5ns$

Pentru 1/6 74LS04: $t_p = \frac{9+10}{2} = 9,5ns$

Pentru 1/4 74LS32: $t_p = \frac{14+14}{2} = 14ns$

Pentru 74LS150: $t_p = \frac{23+22}{2} = 22,5ns$

Pentru 74LS151: $t_p = \frac{27+18}{2} = 22,5ns$

Pentru 1/4 74HC00: $t_p = \frac{9+9}{2} = 9ns$

Pentru 1/4 74HC02: $t_p = \frac{9+9}{2} = 9ns$

Pentru 1/6 74HC04: $t_p = \frac{9+9}{2} = 9ns$

Pentru 1/3 74HC11: $t_p = \frac{10+10}{2} = 10ns$

Pentru 1/2 74HC20: $t_p = \frac{14+14}{2} = 14ns$

Pentru 1/2 74HC21: $t_p = \frac{14+14}{2} = 14ns$

Pentru 74HC154: $t_p = \frac{19+19}{2} = 19ns$

Pentru 74HC155: $t_p = \frac{15+15}{2} = 15ns$

Implementarea funcției logice **a**: $t_p = 1 \cdot 9.5 = 9.5ns$

Implementarea funcțiilor logice **b** și **c**: $t_p = 2 \cdot 9.5 = 19ns$

Implementarea funcției logice **d**: $t_p = 0ns$

Implementarea ansamblului funcțiilor logice cu porți SI-NU: $t_p = 3 \cdot 9.5 = 28.5ns$

Implementarea ansamblului funcțiilor logice cu porți SI-NU și porți SAU-NU: $t_p = 2 \cdot 9.5 + 9 = 28ns$

Implementarea funcției **a** cu 2xMUX-uri de 8 căi conectate în paralel: $t_p = 9.5 + 2 \cdot 22.5 + 14 = 68.5ns$

Implementarea funcției **b** cu MUX de 8 căi și o variabilă aplicată pe intrările de date:

$$t_p = 22.5 + 9.5 = 32ns$$

Implementarea funcției **c** și **d** cu MUX de 16 căi: $t_p = 22.5ns$

Implementarea funcției **a** cu DMUX-uri de 8 căi și porți logice ȘI-NU:

$$t_p = 15 + 2 \cdot 9 + 14 = 47ns$$

Implementarea funcției **b** cu 2xDMUX-uri de 8 căi conectate în paralel: $t_p = 2 \cdot 15 + 9 + 14 = 53ns$

Implementarea funcției **c** cu DMUX de 16 căi și porți logice ȘI-NU: $t_p = 19 + 14 = 33ns$

Implementarea funcției **d** cu DMUX de 16 căi și porți logice ȘI: $t_p = 10 + 19 + 14 = 43ns$

9. Exercițiul 9

9.1 Enunțul problemei

Să se calculeze puterile disipate pentru toate schemele logice obținute.

9.2 Considerații teoretice

Se vor calcula puterile disipate pe fiecare circuit integrat cu formula:

$$P_{dcc} = V_{cc} \frac{I_{CCL} + I_{CCH}}{2} \text{ (pentru TTL)} \quad (9.1)$$

Pentru aflarea puterii disipate a integratelor în tehnologie CMOS am folosit formula:

$$P_d = V_i^2 \cdot f_i (C_p + C_L), \quad (9.2)$$

unde $V_{cc} = 5 \text{ V}$, $f_i = 10^5$, $C_L = 50 \text{ pF}$, iar C_p reprezintă puterea disipată.

9.3 Rezolvare

C.I.	Tehnologia	I_{CCH}, mA	I_{CCL}, mA	P_d, mW
74LS00	TTL	1.6	4.4	15
74LS04	TTL	2.4	6.6	22.5
74LS32	TTL	6.2	9.8	40
74LS150	TTL	14	14	70
74LS151	TTL	10	10	50

Tab. 9.1 Valorile curenților TTL

$$\text{Pentru 74LS00: } P_d = 5 \cdot \frac{1.6+4.4}{2} \cdot 10^{-3} = 15 \text{ mW}$$

$$\text{Pentru 74LS04: } P_d = 5 \cdot \frac{2.4+6.6}{2} \cdot 10^{-3} = 22.5 \text{ mW}$$

$$\text{Pentru 74LS32: } P_d = 5 \cdot \frac{6.2+9.8}{2} \cdot 10^{-3} = 40 \text{ mW}$$

$$\text{Pentru 74LS150: } P_d = 5 \cdot \frac{14+14}{2} \cdot 10^{-3} = 70 \text{ mW}$$

Pentru 74LS151: $P_d = 5 \cdot \frac{10+10}{2} \cdot 10^{-3} = 50mW$

C.I.	Tehnologia	C_P, pF	C_L, pF	P_d, mW
74HC00	CMOS	20	50	0.175
74HC02	CMOS	22	50	0.18
74HC04	CMOS	20	50	0.175
74HC11	CMOS	25	50	0.187
74HC20	CMOS	25	50	0.187
74HC21	CMOS	25	50	0.187
74HC154	CMOS	57	50	0.267
74HC155	CMOS	53	50	0.257

Tab. 9.2 Valorile curenților CMOS

Pentru 74HC00: $P_d = 25 \cdot 10^5 \cdot (20 + 50) \cdot 10^{-12} = 1.75 \cdot 10^{-4}W = 0.175 mW$

Pentru 74HC02: $P_d = 25 \cdot 10^5 \cdot (22 + 50) \cdot 10^{-12} = 1.8 \cdot 10^{-4}W = 0.18 mW$

Pentru 74HC04: $P_d = 25 \cdot 10^5 \cdot (22 + 50) \cdot 10^{-12} = 1.75 \cdot 10^{-4}W = 0.175 mW$

Pentru 74HC11: $P_d = 25 \cdot 10^5 \cdot (25 + 50) \cdot 10^{-12} = 1.875 \cdot 10^{-4}W = 0.187 mW$

Pentru 74HC20: $P_d = 25 \cdot 10^5 \cdot (25 + 50) \cdot 10^{-12} = 1.875 \cdot 10^{-4}W = 0.187 mW$

Pentru 74HC21: $P_d = 25 \cdot 10^5 \cdot (25 + 50) \cdot 10^{-12} = 1.875 \cdot 10^{-4}W = 0.187 mW$

Pentru 74HC154: $P_d = 25 \cdot 10^5 \cdot (57 + 50) \cdot 10^{-12} = 2.675 \cdot 10^{-4}W = 0.267 mW$

Pentru 74HC155: $P_d = 25 \cdot 10^5 \cdot (53 + 50) \cdot 10^{-12} = 2.575 \cdot 10^{-4}W = 0.257 mW$

Implementarea funcției logice **a**: $P_d = 15 mW$

Implementarea funcțiilor logice **b** și **c**: $P_d = 2 \cdot 15 = 30 mW$

Implementarea funcției logice d : $P_d = 0 \text{ mW}$

Implementarea ansamblului funcțiilor logice cu porți ȘI-NU: $P_d = 3 \cdot 15 = 45 \text{ mW}$

Implementarea ansamblului funcțiilor logice cu porți ȘI-NU și porți SAU-NU:

$$P_d = 2 \cdot 15 + 0.18 = 30.18 \text{ mW}$$

Implementarea funcției a cu 2xMUX-uri de 8 căi conectate în paralel :

$$P_d = 2 \cdot 50 + 22.5 + 40 = 162.5 \text{ mW}$$

Implementarea funcției b cu MUX de 8 căi și o variabilă aplicată pe intrările de date:

$$P_d = 50 + 22.5 = 72.5 \text{ mW}$$

Implementarea funcției c și d cu MUX de 16 căi: $P_d = 70 \text{ mW}$

Implementarea funcției a cu DMUX-uri de 8 căi și porți logice ȘI-NU:

$$P_d = 2 \cdot 0.175 + 0.187 + 0.257 = 0.794 \text{ mW}$$

Implementarea funcției b cu 2xDMUX de 8 căi conectate în paralel:

$$P_d = 0.175 + 0.187 + 2 \cdot 0.257 = 0.876 \text{ mW}$$

Implementarea funcției c cu DMUX de 16 căi și porți logice ȘI-NU:

$$P_d = 0.187 + 0.267 = 0.454 \text{ mW}$$

Implementarea funcției d cu DMUX de 16 căi și porți logice ȘI:

$$P_d = 0.187 + 0.187 + 0.267 = 0.641 \text{ mW}$$

- Să se compare soluțiile de implementare obținute:

Putem compara implementările obținute din mai multe puncte de vedere.

Din punctul de vedere al complexității implementării schemei logice, cea mai puțin complexă este schema cu un DMUX de 16 căi și porți logice ȘI-NU sau schema cu un MUX de 16 căi și probabil și cele mai ieftină. La capitolul timp de propagare, schema cu MUX de 16 căi în tehnologia TTL este cea mai rapidă. Din punctul de vedere al consumului de energie, schema cu 2xMUX-uri de 8 căi conectate în paralel este cea mai mare consumatoare de putere.

10.Exercițiul 10

10.1 Enunțul problemei

Se va face analiza, prin simulare, a tuturor schemelor logice obținute utilizându-se pachetul de programe OrCAD.

10.2 Rezolvare

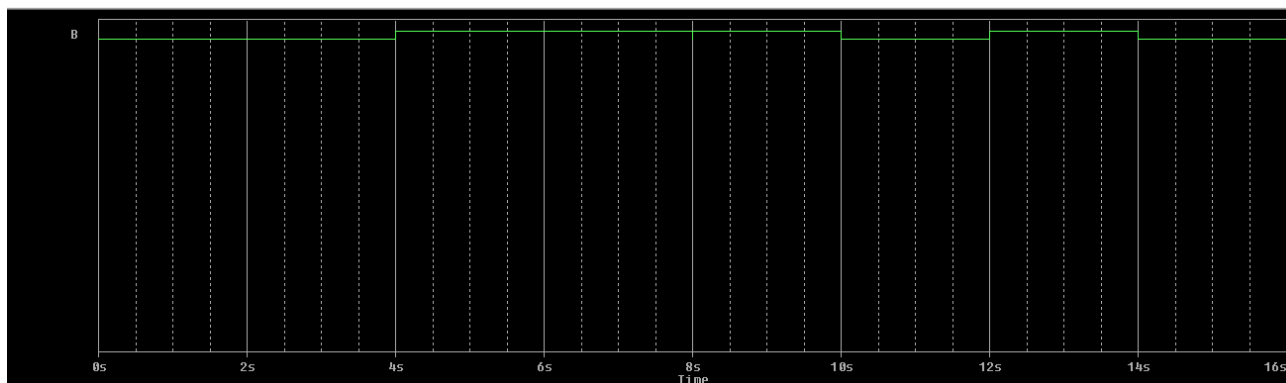


Fig 10.1 Simularea funcției logice b

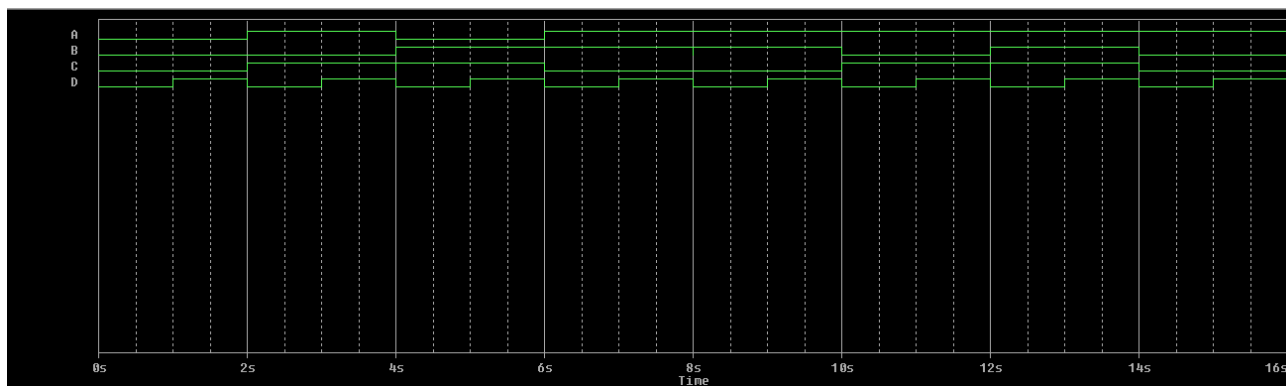


Fig 10.2 Simularea ansamblului funcțiilor logice cu porți ȘI-NU

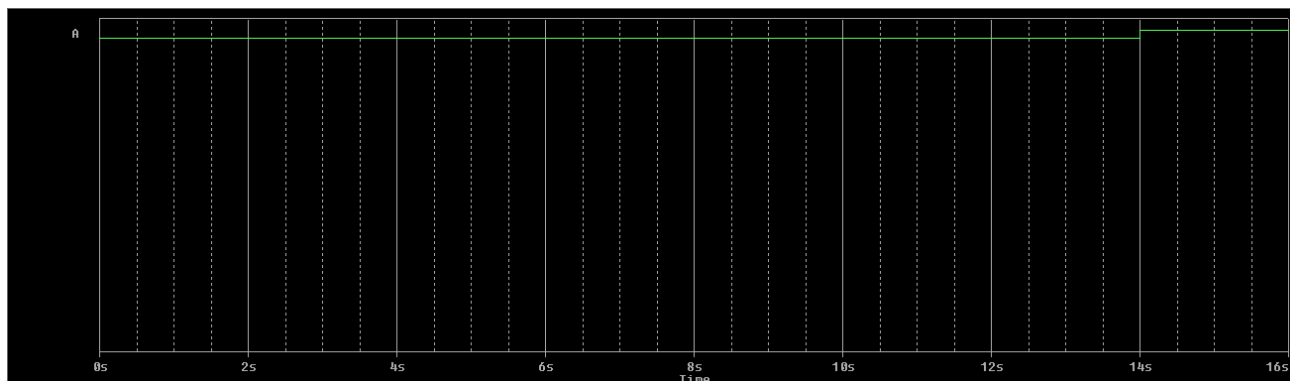


Fig 10.3 Simularea funcției a cu 2xMUX-uri de 8 căi conectate în paralel

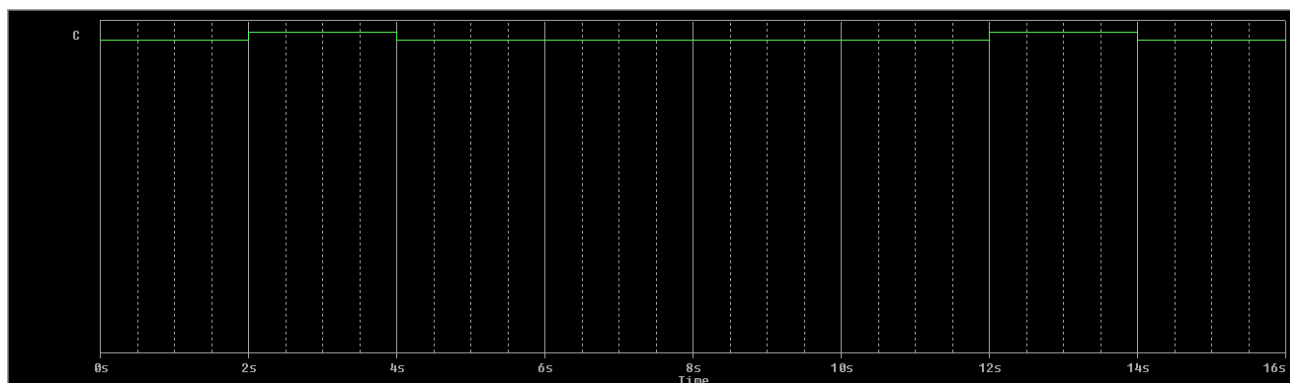


Fig 10.4 Simularea funcției c cu DMUX de 16 căi și porți logice ȘI-NU

Anexa 1

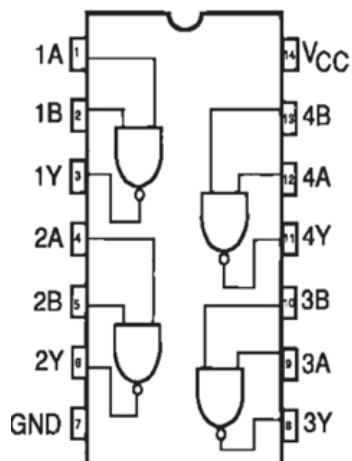


Fig. 1 Circuitul integrat 74LS00

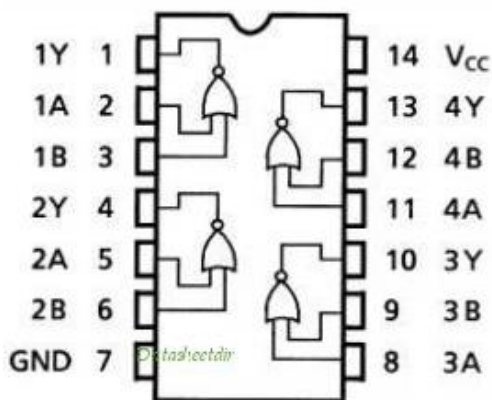


Fig. 2 Circuitul integrat 74HC02

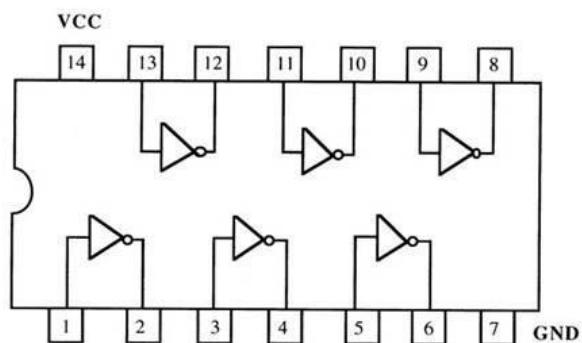


Fig. 3 Circuitul integrat 74LS04

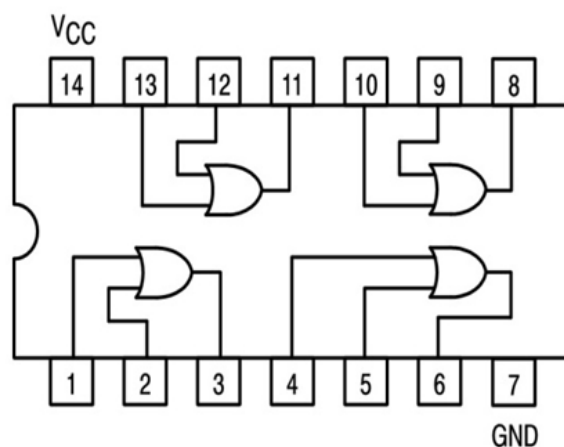


Fig. 4 Circuitul integrat 74LS32

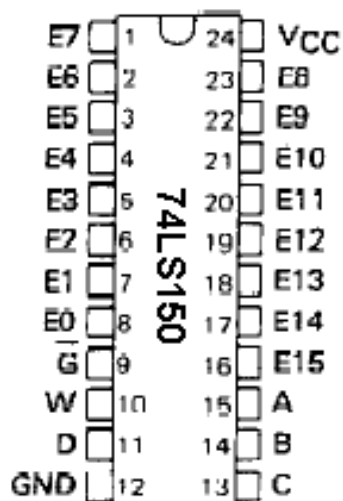


Fig. 5 Circuitul integrat 74LS150

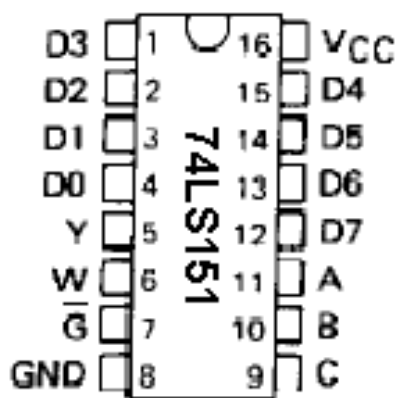


Fig. 5 Circuitul integrat 74LS151

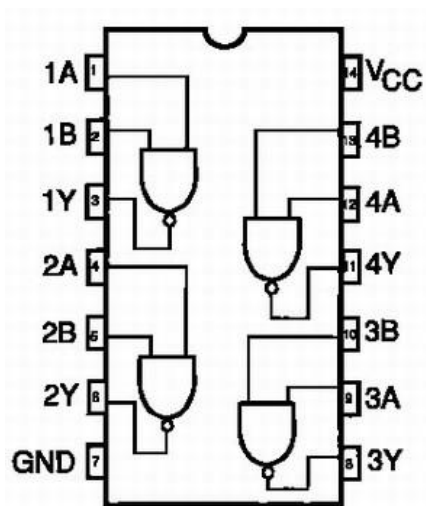


Fig. 6 Circuitul integrat 74HC00

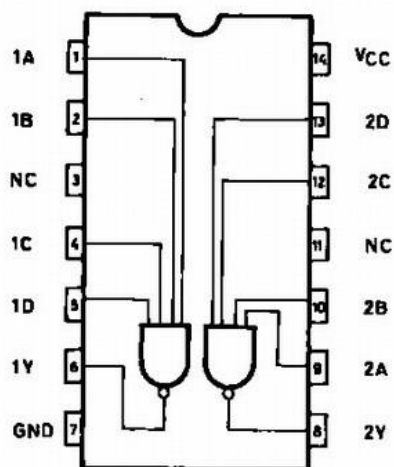


Fig. 7 Circuitul integrat 74HC20



Fig. 8.a Circuitul integrat 74HC155

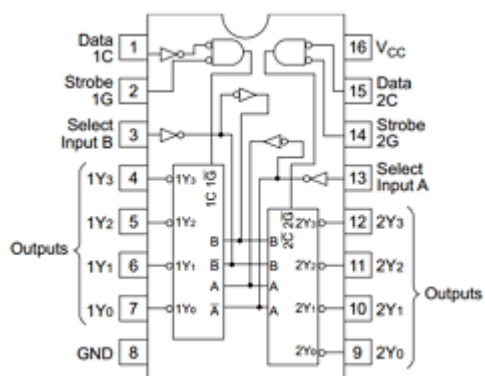


Fig. 8.b Circuitul integrat 74HC155

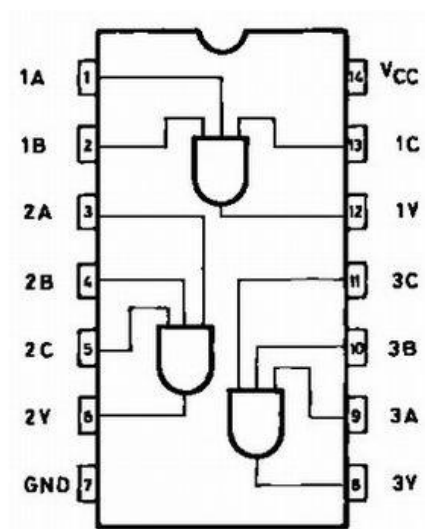


Fig. 9 Circuitul integrat 74HC11

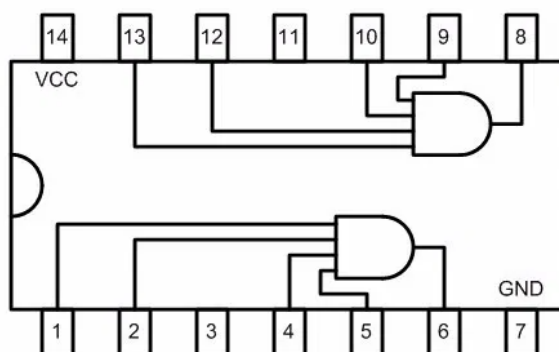


Fig. 10 Circuitul integrat 74HC21

Bibliografie

1. Moldoveanu, F., Floroian, D. – *Circuite logice și comenzi secvențiale. Circuite logice combinaționale*, Ed. Universității Transilvania din Brașov, 2003.
2. <https://datasheetspdf.com/> - Date de catalog
3. <https://ampere-electronics.com/> - Informații circuite integrate