



Universitatea  
Transilvania  
din Brașov

UNIVERSITATEA TRANSILVANIA BRAȘOV

Facultatea de Inginerie Electrică și Știința  
Calculatoarelor

Tudor Georgian Iulian

# Proiect ASCN

Îndrumător

Prof. dr. ing. MOLDOVEANU Florin

2020

## Cuprins

1. Fișa proiectului de disciplină .....	3
2. Reprezentare funcțiilor booleene.....	4
2.1 TABELUL DE ADEVĂR.....	4
2.2 Forma canonică disjunctive .....	5
2.3 Forma canonică conjunctiva.....	5
2.4 Diarame Karnaugh .....	5
3. Formele minime disjunctive și conjunctive .....	8
3.1 Minimizarea funcțiilor prin metoda Quine – McCluskey.....	8
4. Reprezentarea funcțiilor booleene prin scheme logice.....	10
5. Implementare cu circuite multiplexoare .....	16
6. Implementare cu circuite demultiplexoare.....	22
7. Calculul timpilor de propagare și al puterilor disipate .....	26
8. Simulări Pspice.....	28
9. Anexa A .....	29
Circuite integrate.....	29

# 1. Fișa proiectului de disciplina

## Disciplina:

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

## Tema de proiect nr. 30:

Să se proiecteze un convertor de cod de 4 biți pentru conversia codului binar-zecimal ponderat 7421 în codul binar-zecimal neponderat Gray (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 indifferente) prin metoda diagramelor Veitch-Karnaugh; de asemenea, se vor obține formele minime disjunctive pentru ultimele două funcții logice de ieșire și prin metoda Quine-McCluskey.
- Să se implementeze fiecare funcție logică, independent, numai cu porți logice SAU-NU (se vor utiliza circuite integrate realizate în tehnologia CMOS).
- Să se implementeze ansamblul funcțiilor logice numai cu porți logice SAU-NU (se vor utiliza circuite integrate realizate în tehnologia CMOS).
- Să se implementeze ansamblul funcțiilor logice în următoarea variantă: primele două funcții logice de ieșire cu porți logice SAU-NU (circuite integrate CMOS), iar următoarele două cu porți logice ȘI-NU (circuite integrate TTL).
- 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.

## 2. Reprezentare funcțiilor booleene

### 2.1 TABELUL DE ADEVĂR

Tabelul de adevăr constituie reprezentarea tabelară a funcțiilor booleene și constă într-un tabel în care pe linii în partea stângă se plasează toate combinațiile posibile ale valorilor argumentelor funcțiilor, iar în partea dreaptă, valorile pentru combinațiile respective.

Tabelul de adevăr reprezintă cea mai completă modalitate de reprezentare a unei funcții booleene deoarece pentru fiecare combinație posibilă a valorilor argumentelor funcției se cunoaște valoarea acesteia.

Nr. crt	Ez	Ponderat 7421				Neponderat Gray			
		$x_1$	$x_2$	$x_3$	$x_4$	$f_1$	$f_2$	$f_3$	$f_4$
0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	1
2	2	0	0	1	0	0	0	1	1
3	3	0	0	1	1	0	0	1	0
4	4	0	1	0	0	0	1	1	0
5	5	0	1	0	1	0	1	1	1
6	6	0	1	1	0	0	1	0	1
7	7	0	1	1	1	0	1	0	0
8	9	1	0	0	1	1	1	0	0
9	10	1	0	1	0	1	1	0	1

Fig 2.1 tabel de adevăr

Combinatii indifferente :  $P_8 = P_{11} = P_{12} = P_{13} = P_{14} = P_{15} = x$

## 2.2 Forma canonică disjunctivă

Forma canonică disjunctivă este o modalitate de reprezentare analitică a funcțiilor booleene în care orice funcție data prin table de adevăr poate fi scrisă astfel:

$$f(x_1, x_2, x_3, \dots, x_m) = P_{i_1} + P_{i_2} + \dots + P_{i_n} = \sum_{ij \in M_1} P_{ij} \quad (2.1)$$

Unde  $M_1$  este mulțimea numerelor combinațiilor valorilor argumentelor pentru care funcția ia valoarea "1".

$$f_1^{FCD}(x_1, x_2, x_3, x_4) = x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = P_9 + P_{10} \sum(9,10) \quad (2.2)$$

$$f_2^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = P_4 + P_5 + P_6 + P_7 + P_9 + P_{10} = \sum(4,5,6,7,9,10) \quad (2.3)$$

$$f_3^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 = P_2 + P_3 + P_4 + P_5 = \sum(2,3,4,5) \quad (2.4)$$

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = P_1 + P_2 + P_4 + P_5 + P_6 + P_{10} = \sum(1,2,5,6,10) \quad (2.5)$$

## 2.3 Forma canonică conjunctivă

Forma canonică conjunctivă este o modalitate de reprezentare analitică a funcțiilor booleene în care orice funcție data prin table de adevăr poate fi scrisă astfel:

$$f(x_1, x_2, x_3, \dots, x_m) = S_{i_1} * S_{i_2} * \dots * S_{i_n} = \prod_{ij \in M_0} S_{ij} \quad (2.6)$$

Unde  $M_0$  este mulțimea numerelor combinațiilor valorilor argumentelor pentru care funcția ia valoarea "0".

$$f_1^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + x_2 + \bar{x}_3 + x_4)(x_1 + x_2 + \bar{x}_3 + \bar{x}_4)(x_1 + \bar{x}_2 + x_3 + x_4)(x_1 + \bar{x}_2 + x_3 + \bar{x}_4)(x_1 + \bar{x}_2 + \bar{x}_3 + x_4)(x_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) = S_0 * S_1 * S_2 * S_3 * S_4 * S_5 * S_6 * S_7 = \prod(0,1,2,3,4,5,6,7) \quad (2.7)$$

$$f_2^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + x_2 + \bar{x}_3 + x_4)(x_1 + x_2 + \bar{x}_3 + \bar{x}_4) = S_0 * S_1 * S_2 * S_3 = \prod(0,1,2,3) \quad (2.8)$$

$$f_3^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + x_3 + \bar{x}_4)(x_1 + \bar{x}_2 + \bar{x}_3 + x_4)(x_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4)(\bar{x}_1 + x_2 + x_3 + \bar{x}_4)(\bar{x}_1 + x_2 + \bar{x}_3 + \bar{x}_4) = S_0 * S_1 * S_6 * S_7 * S_9 * S_{10} = \prod(0,1,6,7,9,10) \quad (2.9)$$

$$f_4^{FCC}(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4)(x_1 + x_2 + \bar{x}_3 + \bar{x}_4)(x_1 + \bar{x}_2 + x_3 + x_4)(x_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4)(\bar{x}_1 + x_2 + \bar{x}_3 + \bar{x}_4) = S_0 * S_3 * S_4 * S_7 * S_{10} = \prod(0,3,4,7,10) \quad (2.10)$$

## 2.4 Diarame Karnaugh

Diagrama Karnaugh este tot o reprezentare tabelară dar, în raport cu tabelul de adevăr, este mai compactă datorită dispunerii bidirecționale a valorilor argumentelor

În cazul general al unei funcții booleene de  $n$  argumente, diagrama Karnaugh conține  $2^p$  linii și  $2^q$  coloane, astfel că  $p + q = n$ . Dacă  $n$  este par, în mod obișnuit  $p = q$ , iar dacă  $n$  este impar,  $q = p + 1$ . Rezultă o diagramă cu  $2^p * 2^q = 2^n$  câmpuri (compartimente sau locații) în care se trec

valorile funcției pentru combinațiile corespunzătoare ale valorilor argumentelor. Valorile argumentelor se indică la capetele liniilor și coloanelor diagramei.

Orice funcție booleană dată prin tabel de adevăr poate fi scrisă sub următoarea formă analitică:

$$f(x_1, x_2, \dots, x_n) = P_{i1} + P_{i2} + \dots + P_{ik} = \sum_{i_j \in M_1} P_{i_j} \quad (2.11)$$

Unde  $M_1$  este mulțimea numerelor combinațiilor valorilor argumentelor pentru care funcția ia valoarea 1.

$$f_1^{FCD}(x_1, x_2, x_3, x_4) = x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = P_9 + P_{10} \sum(9,10) \quad (2.12)$$

$$f_1^{FMD}(x_1, x_2, x_3, x_4) = x_1 \quad (2.13)$$

$$\bar{f}_1^{FMD}(x_1, x_2, x_3, x_4) = f_1^{FMC}(x_1, x_2, x_3, x_4) \quad (2.14)$$

$$f_1^{FMC}(x_1, x_2, x_3, x_4) = x_1 \quad (2.15)$$

	$x_1 x_2$			
$x_3 x_4$	00	01	11	10
00	0	0	*	*
01	0	0	*	1
11	0	0	*	*
10	0	0	*	1

Fig 2.2 Diagrama Karnaugh pentru funcția  $f_1$

$$f_2^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = P_4 + P_5 + P_6 + P_7 + P_9 + P_{10} = \sum(4,5,6,7,9,10) \quad (2.15)$$

$$f_2^{FMD}(x_1, x_2, x_3, x_4) = x_1 + x_2 \quad (2.16)$$

$$\bar{f}_2^{FMD}(x_1, x_2, x_3, x_4) = f_2^{FMC}(x_1, x_2, x_3, x_4) \quad (2.17)$$

$$f_2^{FMC}(x_1, x_2, x_3, x_4) = x_1 + x_2 \quad (2.18)$$

	$x_1 x_2$			
$x_3 x_4$	00	01	11	10
00	0	1	*	*
01	0	1	*	1
11	0	1	*	*
10	0	1	*	1

Fig 2.3 Diagrama Karnaugh pentru funcția  $f_2$

$$f_3^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 = P_2 + P_3 + P_4 + P_5 = \sum(2,3,4,5) \quad (2.19)$$

$$f_3^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2x_3 + x_2\bar{x}_3 \quad (2.20)$$

$$\bar{f}_3^{FMD}(x_1, x_2, x_3, x_4) = f_3^{FMC}(x_1, x_2, x_3, x_4) \quad (2.21)$$

$$f_3^{FMC}(x_1, x_2, x_3, x_4) = \bar{x}_1 * (\bar{x}_2 + \bar{x}_3) + (x_2 * x_3) \quad (2.22)$$

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

Fig 2.4 Diagrama Karnaugh pentru funcția  $f_3$

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 = P_1 + P_2 + P_4 + P_5 + P_6 + P_{10} = \sum(1,2,5,6,10) \quad (2.23)$$

$$f_4^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2x_4 + x_3\bar{x}_4 \quad (2.24)$$

$$\bar{f}_4^{FMD}(x_1, x_2, x_3, x_4) = f_4^{FMC}(x_1, x_2, x_3, x_4) \quad (2.25)$$

$$f_4^{FMC}(x_1, x_2, x_3, x_4) = (x_3 + x_4) * (\bar{x}_3 + \bar{x}_4) * (\bar{x}_1 + \bar{x}_3) \quad (2.26)$$

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

Fig 2.5 Diagrama Karnaugh pentru funcția  $f_4$

## 3. Formele minime disjunctive și conjunctive

### 3.1 Minimizarea funcțiilor prin metoda Quine – McCluskey

Metoda Quine-McCluskey este o metodă algebrică de minimizare a funcțiilor booleene folosită în cazul funcțiilor cu număr mare de variabile, pentru care metodele grafice de minimizare (de exemplu, metoda diagramelor Karnaugh) devin greu de utilizat. Metoda se bazează pe o teoremă datorată lui W. V. Quine, care are o mare importanță pentru abordarea sistematică a minimizării funcțiilor booleene și care se enunță în felul următor:

Teorema lui Quine. Dacă în forma canonică disjunctivă a unei funcții booleene se fac toate operațiile de alipire parțială și apoi toate operațiile de absorbție, se obține disjuncția implicantilor primi.

Fie un sistem de impicanți primi  $\varphi_k$  ai unei funcții de  $n$  argumente. Conform teoremei are loc relația:

$$f(x_1, x_2, x_3, x_4) = \sum_k \varphi_k \quad (3.1)$$

Relația de mai sus trebuie să fie adevărată atât pentru  $f = 0$  cât și pentru  $f = 1$ . Atunci când  $f = 0$ , așa cum s-a arătat mai sus, toți impicanții primi ai funcției sunt 0, deci și  $\sum \varphi_k$ . Când  $f = 1$ , va exista cel puțin un implicant prim  $\varphi_j = 1$ , astfel că întreaga disjuncție din partea dreaptă a relației date va avea valoarea 1

$$f_3^{FCD}(x_1, x_2, x_3, x_4) = \sum (2,3,4,5) \quad (3.2)$$

0	2	0010
	4	0100
	8	1000
1	3	0011
	5	0101
	12	1100
2	11	1011
	13	1101
	14	1110
3	15	1111

Gr	Indicii	x1x2x3x4
0	2,3	001-
	4,5	010-
	4,12	-100
	8,12	1-00
1	3,11	-011
	5,13	-101
	12,13	110-
	12,14	11-0
2	11,15	1-11
	13,15	11-1
	14,15	111-



4,5,12,13	-10-
12,13,14,15	11--

4,5,12,13	-10-	$\overline{x_2 x_3}$
2,3	001-	$\overline{x_1 x_2 x_3}$

	2	3	4	5
4,5,12,13			x	x
2,3	x	x		
3,11		x		

$$f_3^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_3 + x_2 \bar{x}_3 \quad (3.3)$$

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = \sum (1, 2, 5, 6, 10) \quad (3.4)$$

0	1	0001
	2	0010
	8	1000
1	5	0101
	6	0110
	10	1010
	12	1100
2	11	1011
	13	1101
	14	1110
3	15	1111

Gr	Indicii	x1x2x3x4
0	1,5	0-01
	2,6	0-10
	2,10	-010
	8,10	11-0
	8,12	1-00
1	5,13	-101
	6,14	-101
	10,11	101-
	10,14	1-10
	12,13	110-
	12,14	11-0
2	11,15	1-11
	13,15	11-1
	14,15	111-

2,5,10,14	--10
8,10,12,14	1—0
10,11,14,15	1-1-
12,13,14,15	11--

2,6,10,14	--10	$\overline{x_3 x_4}$
1,5	0-01	$\overline{x_1 x_3 x_4}$

	1	2	5	6	10
2,6,10,14		x		x	x
8,10,12,14					x
10,11,14,15					x
1,5	x		x		
5,13			x		

$$f_4^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_4 + x_3 \bar{x}_4 \quad (3.5)$$

## 4. Reprezentarea funcțiilor booleene prin scheme logice

Schema logică (logigrama, rețeaua de comutare sau, mai rar, schema de operatori) este o reprezentare grafică a funcției booleene obținută prin adoptarea unor semne (simboluri) convenționale pentru operatorii logici. În general, numim operator n-ăr o funcție booleană de n variabile  $y = f(x_1, x_2, \dots, x_n)$ . Mulțimea operatorilor utilizați în practică este determinată de posibilitatea realizării acestora prin elemente fizice. Schema logică indică de fapt topologia unui circuit logic care materializează o funcție booleană. Ca urmare, simbolurile grafice adoptate pentru operatorii logici constituie o reprezentare a circuitelor logice care materializează funcțiile logice elementare. Una dintre posibilitățile de realizare fizică a acestora o reprezintă porțile logice (v. §2.3.2). În tabelul 1.8 sunt indicate cele mai utilizate simboluri grafice pentru principalele funcții elementare de două argumente, în conformitate cu trei sisteme de norme:

- normele Comisiei Electrotehnice Internaționale (CEI)2, care sunt recunoscute internațional;
- normele americane MIL-STD-806 B (MIL)3,4, care sunt foarte frecvent utilizate în practică (adoptate și în literatura de specialitate românească);
- normele germane DIN 40700, ediția 1963. Folosind aceste simboluri grafice expresiile algebrice ale funcțiilor booleene pot fi reprezentate sub formă de scheme logice.

$$f_1^{FMD}(x_1, x_2, x_3, x_4) = x_1 \quad (4.1)$$

$$\overline{x_1 + x_1} = \overline{x_1} * \overline{x_1} = x_1 \quad (4.2)$$



Fig 4.1 implementarea funcției  $f_1$

$$f_2^{FMD}(x_1, x_2, x_3, x_4) = x_1 + x_2 \quad (4.3)$$

$$\overline{x_1 + x_2} = \overline{x_1} * \overline{x_2} = x_1 + x_2 \quad (4.4)$$

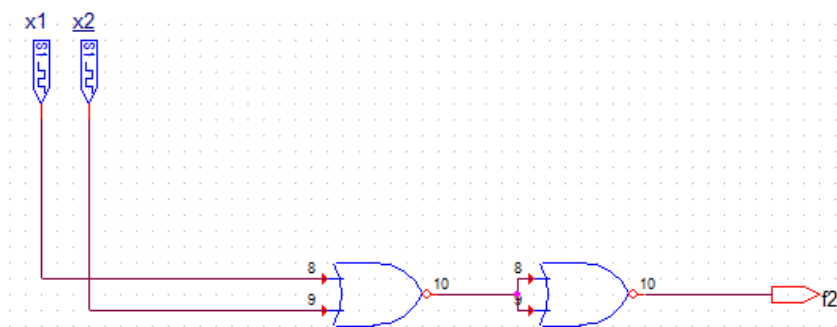


Fig 4.2 implementarea funcției  $f_2$  cu porți logice SAU-NU

Se vor utiliza următoarele circuite integrate: 1 x 4001

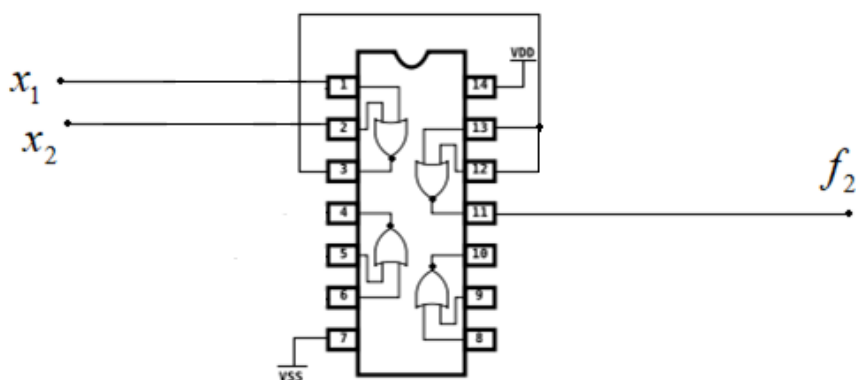


Fig 4.3 implementarea funcției  $f_2$  cu circuite integrate

$$f_3^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_3 + x_2 \bar{x}_3 \quad (4.5)$$

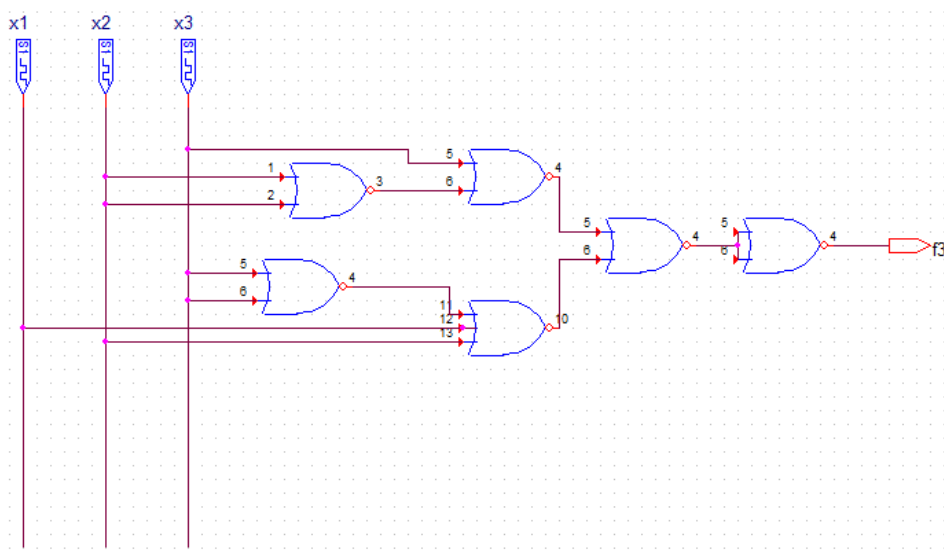


Fig 4.4 implementarea funcției  $f_3$  cu porți logice SAU-NU

Se vor utiliza următoarele circuite integrate: 2 x 4001

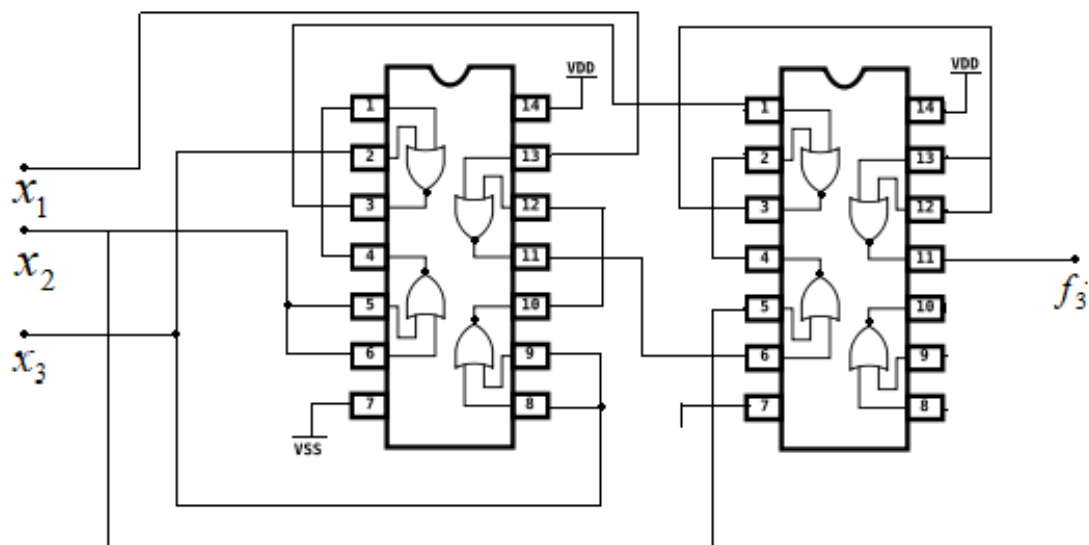


Fig 4.5 implementarea funcției  $f_3$  cu circuite integrate

$$f_4^{FMD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_4 + x_3 \bar{x}_4 \quad (4.6)$$

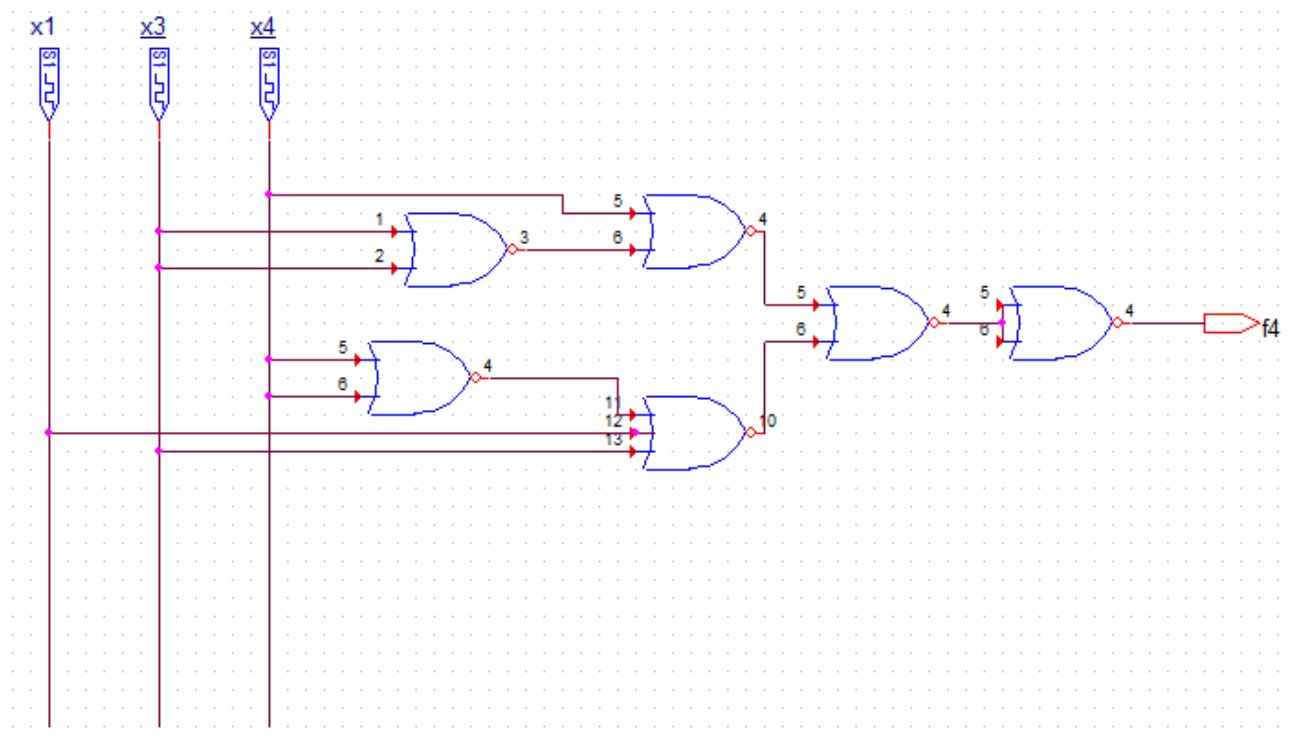


Fig 4.6 implementarea funcției  $f_4$  cu porți logice SAU-NU

Se vor utiliza următoarele circuite integrate: 2 x 4001

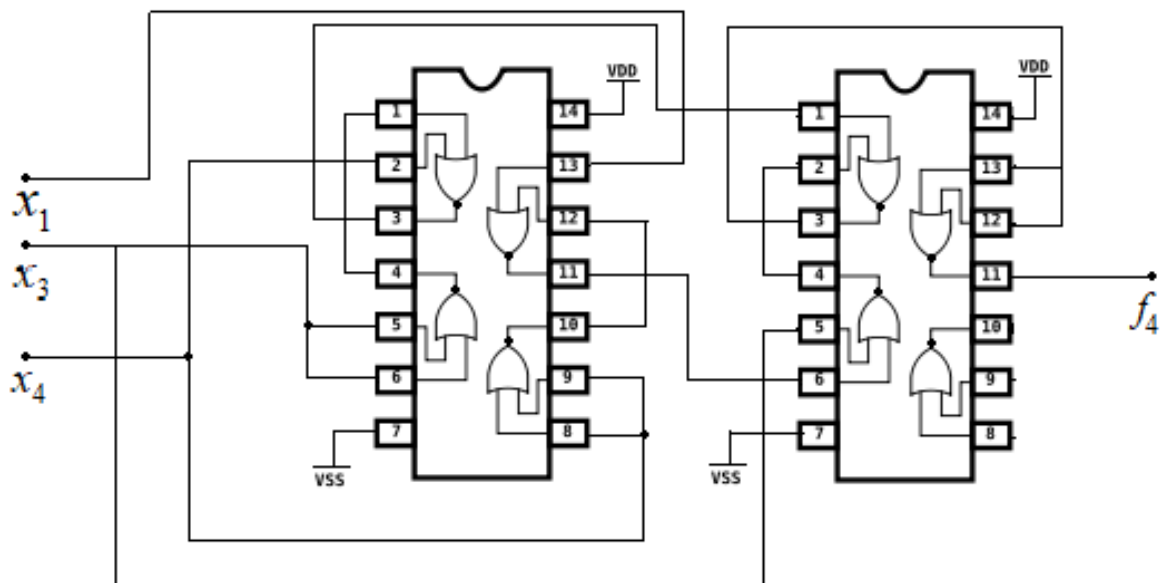


Fig 4.7 implementarea funcției  $f_4$  cu circuite integrate

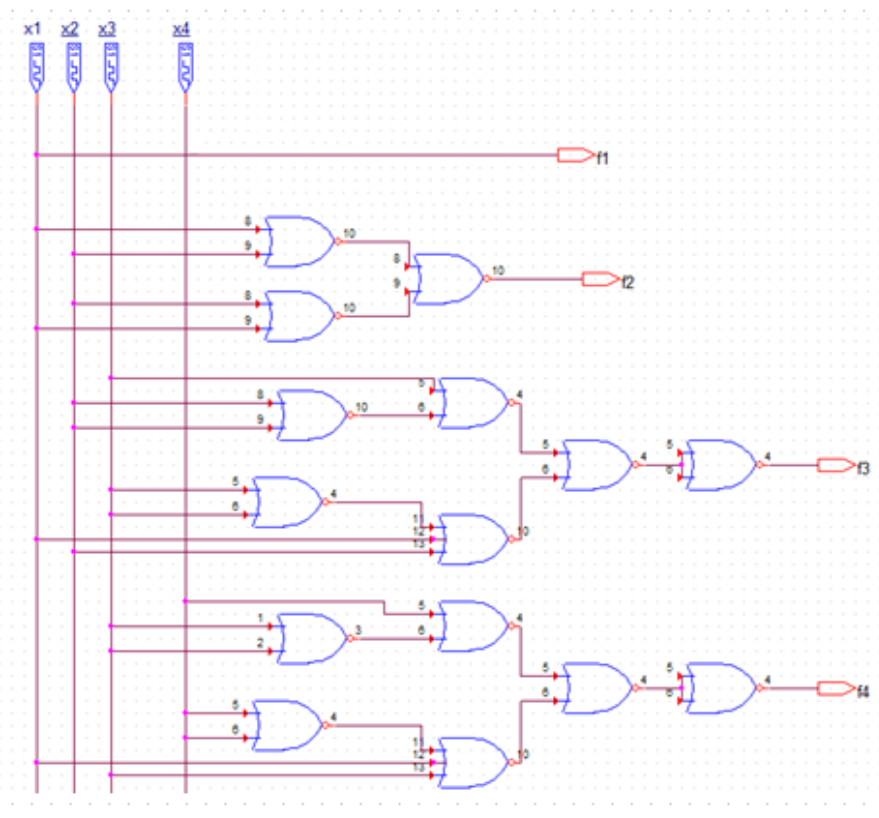


Fig 4.8 implementarea ansamblului de funcții cu porți logice SAU-NU

Se vor utiliza următoarele circuite integrate: 5 x 4001

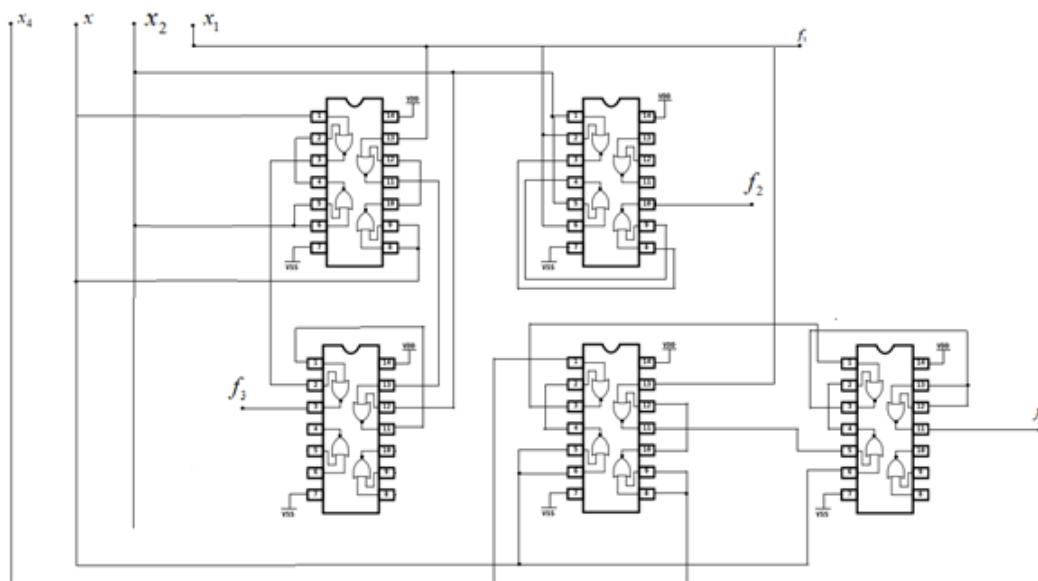


Fig 4.9 implementarea ansamblului de funcții SAU-NU cu circuite integrate

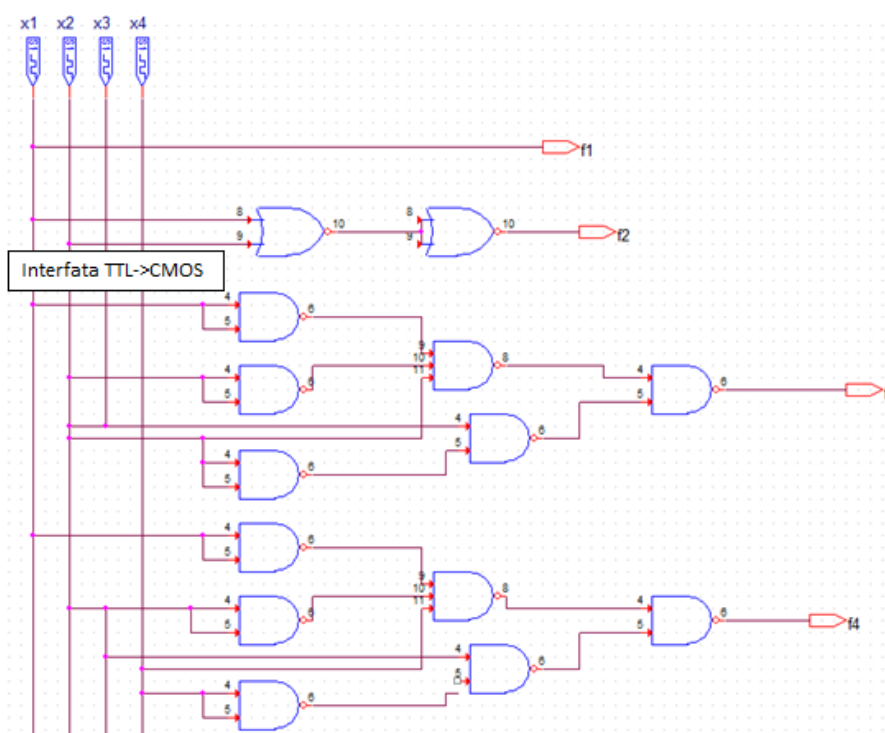


Fig 4.10 implementarea ansamblului de funcții cu porți logice primele 2 cu SAU-NU iar următoarele 2 cu SI-NU

Se vor utiliza următoarele circuite integrate: 1 x 4001; 3 x 7400

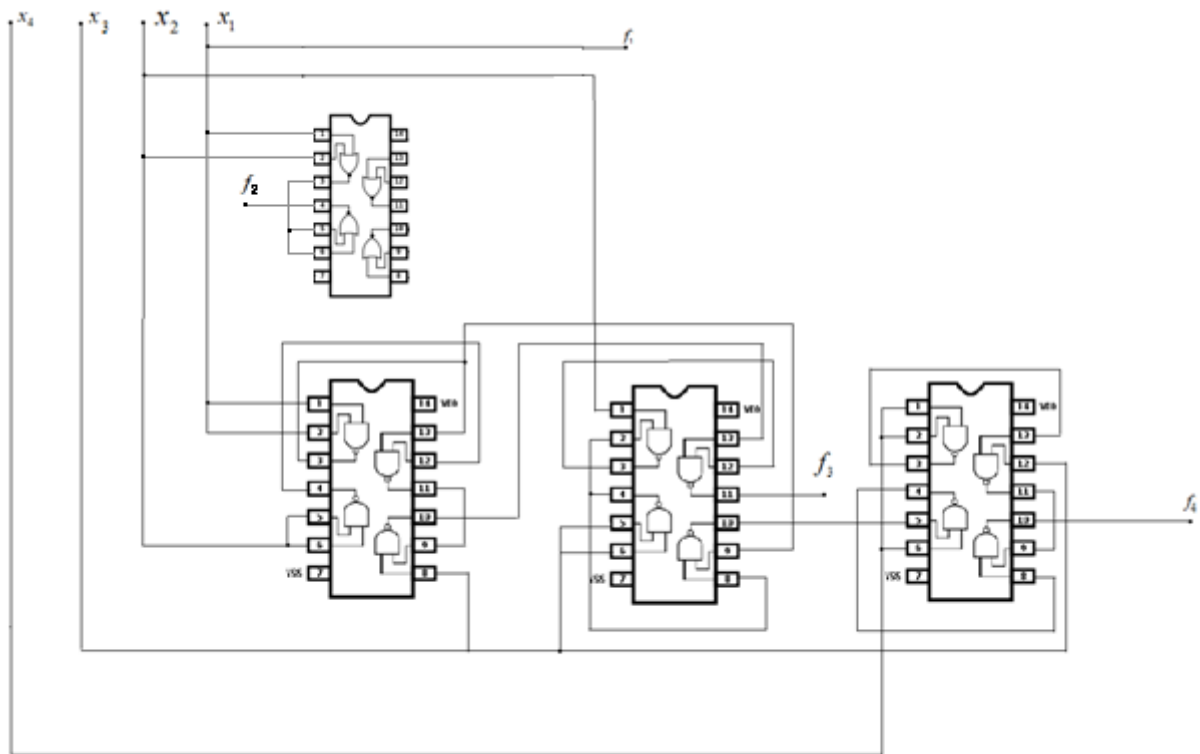


Fig 4.11 implementarea ansamblului de funcții primele 2 cu SAU-NU și următoarele 2 cu SI-NU cu circuite integrate

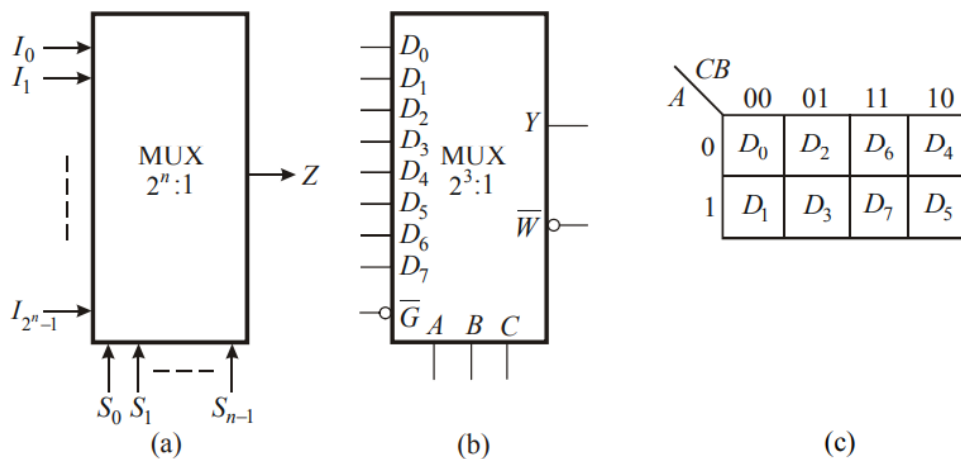
## 5. Implementare cu circuite multiplexoare

Multiplexorul (MUX-ul) este unul dintre cele mai importante circuite MSI. Este denumit uneori și „selector” deoarece este utilizat și ca un comutator de selectare a anumitor căi [2,15,22,23,26÷30,33].

Multiplexorul/selectorul este un circuit logic combinațional care are, în cazul general,  $2^n$  intrări de date,  $I_0, I_1, \dots, I_{2^n-1}$ ,  $n$  intrări de selecție (adresă),  $S_0, S_1, \dots, S_{n-1}$  și o ieșire  $Z$  (fig a). Starea ieșirii circuitului la un moment dat este aceeași cu starea intrării  $I_k$ , unde indicele  $k$  este echivalentul zecimal al numărului binar reprezentat de stările 0 și 1 ale intrărilor de selecție:

$$k = (S_{n-1}S_{n-2} \dots S_1S_0) \quad (5.1)$$

Pentru ca la ieșire să apară întotdeauna numai intrarea selectată trebuie ca selecția să se facă după stabilizarea intrărilor de adresă. Din acest motiv, multiplexoarele sunt prevăzute cu o intrare suplimentară, intrarea de autorizare/validare sau strobare,  $\bar{G}$ , care condiționează selecția fiecărei intrări. Ea are de fapt rolul de a comanda inhibarea respectiv dezinhibarea funcționării circuitului. Această intrare suplimentară poate fi folosită și la extinderea numărului de intrări, prin legarea mai multor circuite de multiplexare. Intrarea s-a notat cu  $\bar{G}$  deoarece semnalul de autorizare permite selectarea numai atunci când are valoarea logică 0.



Implementarea funcției  $f_4$  cu MUX de 8 căi.

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 = \\ (\bar{x}_1 \bar{x}_2 \bar{x}_3)x_4 + (\bar{x}_1 \bar{x}_2 x_3)\bar{x}_4 + (\bar{x}_1 x_2 \bar{x}_3)x_4 + (\bar{x}_1 x_2 x_3)\bar{x}_4 + x_2 x_3 \bar{x}_4 = P'_0 * x_4 + P'_2 * \bar{x}_4 + \\ P'_5 * x_4 + P'_7 * \bar{x}_4 + P'_{10} * \bar{x}_4 \quad (5.2)$$



Nr crt	$x_1$	$x_2$	$x_3$	$x_4$	$f_4$	
0	0	0	0	0	0	$x_4$
1	0	0	0	1	1	
2	0	0	1	0	1	$\overline{x_4}$
3	0	0	1	1	0	
4	0	1	0	0	0	$x_4$
5	0	1	0	1	1	
6	0	1	1	0	1	$\overline{x_4}$
7	0	1	1	1	0	
8	1	0	0	0	$x$	0
9	1	0	0	1	0	
10	1	0	1	0	1	$\overline{x_4}$
11	1	0	1	1	$x$	
12	1	1	0	0	$x$	0
13	1	1	0	1	$x$	
14	1	1	1	0	$x$	0
15	1	1	1	1	$x$	

Fig 5.1 Tabelul de adevar pentru functi  $f_4$

Combinatii indifferente :  $P_8 = P_{11} = P_{12} = P_{13} = P_{14} = P_{15} = x$

$$\begin{aligned}
 f_4^{FCD}(x_1, x_2, x_3, x_4) &= \overline{x_1} \overline{x_2} \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 \overline{x_4} + x_1 \overline{x_2} x_3 \overline{x_4} = \\
 &= (\overline{x_1} \overline{x_2} \overline{x_3})x_4 + (\overline{x_1} \overline{x_2} x_3)\overline{x_4} + (\overline{x_1} x_2 \overline{x_3})x_4 + (\overline{x_1} x_2 x_3)\overline{x_4} + \overline{x_2} x_3 \overline{x_4} = \\
 &= P'_0 * x_4 + P'_2 * \overline{x_4} + P'_5 * x_4 + P'_7 * \overline{x_4} + P'_{10} * \overline{x_4}
 \end{aligned}
 \tag{5.3}$$

Din primele doua linii ale tabelului,  $x_1 x_2 x_3 = 000$ ; se observa ca  $f = 1$ , pentru  $x_4 = 1$ , deci la intrare de date  $D_0$  se va aplica  $x_4$

Din urmatoarele doua linii ale tabelului,  $x_1 x_2 x_3 = 001$ ; se observa ca  $f = 1$ , pentru  $x_4 = 0$ , deci la intrare de date  $D_1$  se va aplica  $\overline{x_4}$

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 010$ ; se observa ca  $f = 1$ , pentru  $x_4 = 1$ , deci la intrare de date  $D_2$  se va aplica  $x_4$

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 011$ ; se observa ca  $f = 1$ , pentru  $x_4 = 0$ , deci la intrare de date  $D_3$  se va aplica  $\overline{x_4}$

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 100$ ; se observa ca  $f = 0$ , pentru orice combinație, deci la intrarea de date  $D_4$  se va aplica 0

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 101$ ; se observa ca  $f = 1$ , pentru  $x_4 = 0$ , deci la intrare de date  $D_5$  se va aplica  $\overline{x_4}$

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 110$ ; se observa ca  $f = 0$ , pentru orice combinație, deci la intrarea de date  $D_6$  se va aplica 0

Din următoarele două linii ale tabelului,  $x_1x_2x_3 = 111$ ; se observa ca  $f = 0$ , pentru orice combinație, deci la intrarea de date  $D_7$  se va aplica 0

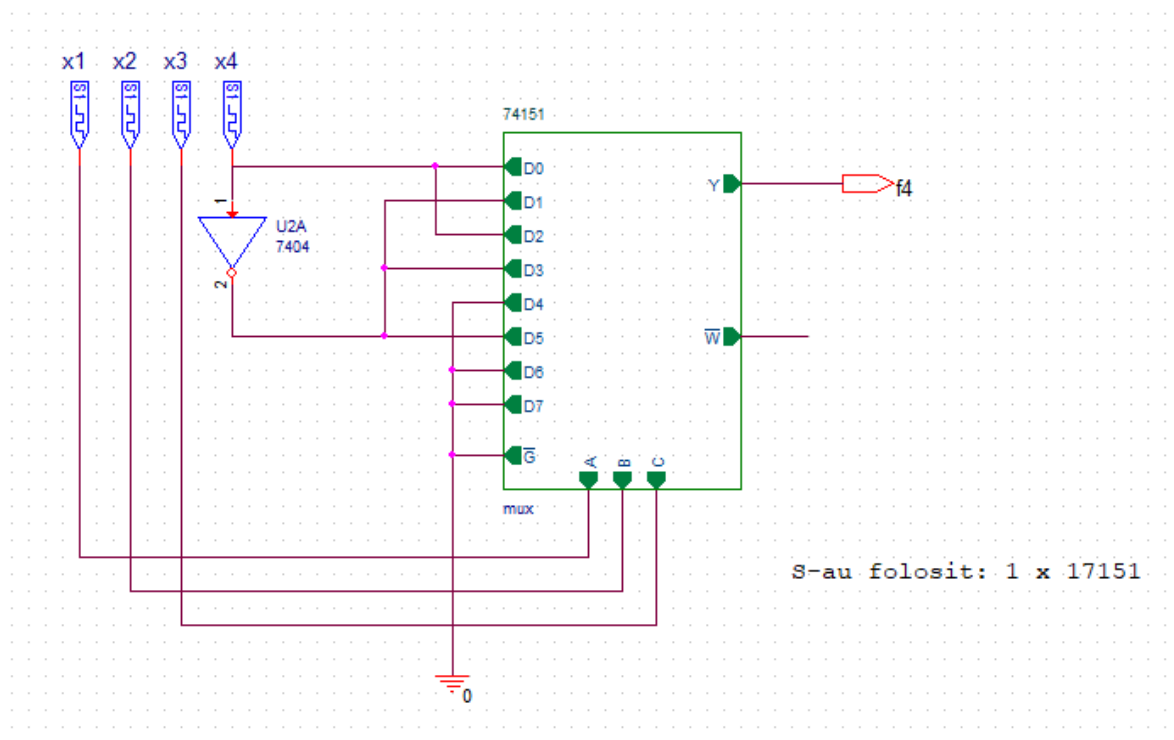


Fig 5.1 implementarea funcției  $f_4$  cu multiplexor de 8 căi

Pentru implementarea multiplexorului de 8 ci se vor utiliza următoarele circuite integrate: 1 x 17151

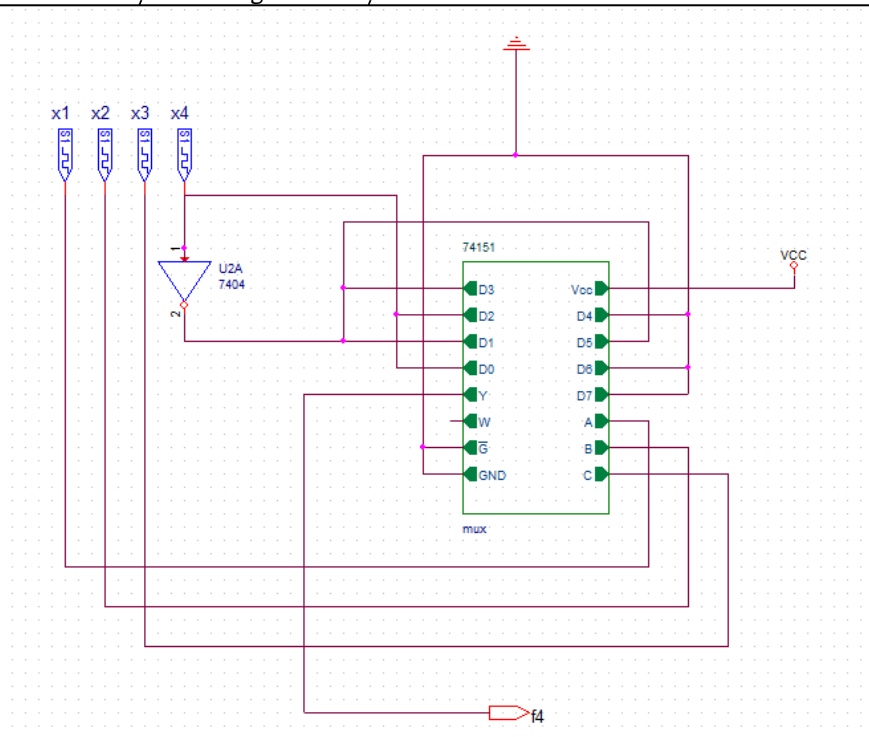


Fig 5.2 implementarea multiplexorului de 8 căi cu circuite integrate

Implementare funcției  $f_4$  cu multiplexoare de 8 căi legate în paralel

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_2} \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 \overline{x_4} + x_1 \overline{x_2} x_3 \overline{x_4} =$$

$$= P_1 + P_2 + P_5 + P_6 + P_{10} = \Sigma(1,2,5,6,10) \quad (5.4)$$

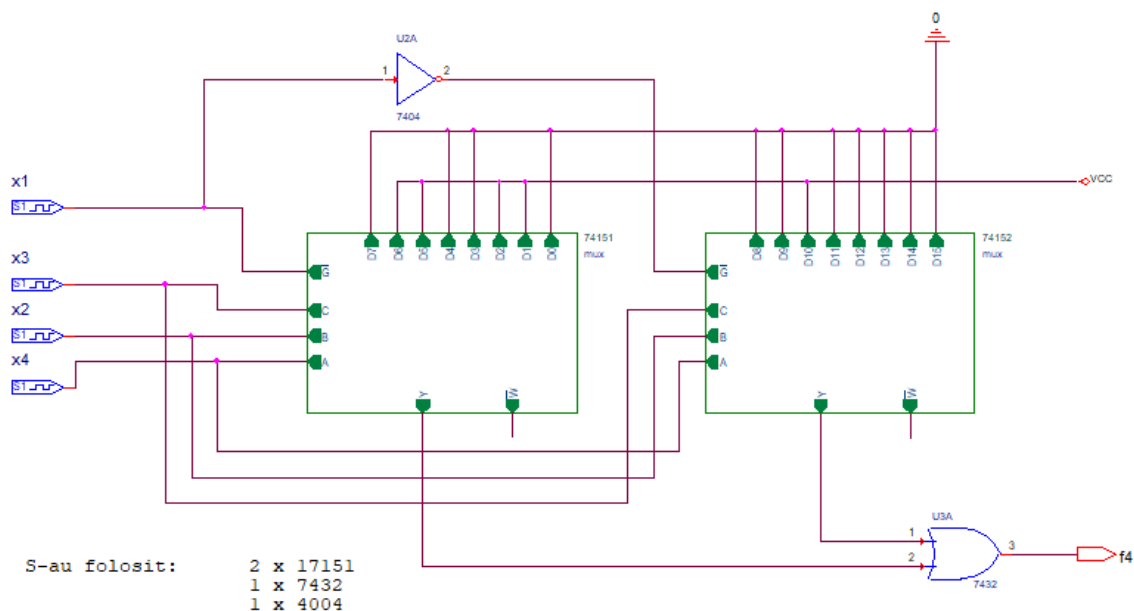


Fig 5.3 implementarea funcției  $f_4$  cu multiplexor de 8 căi în paralel

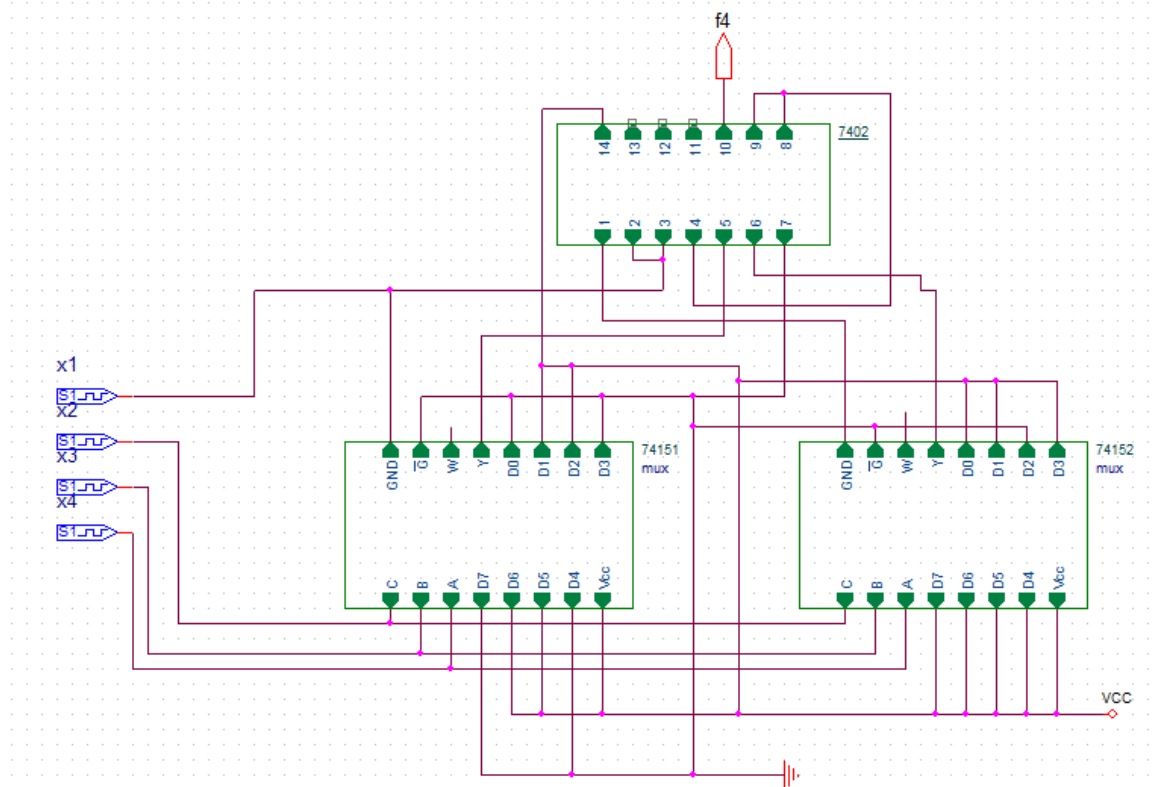


Fig 5.4 implementarea multiplexorului de 8 căi în paralel cu circuite integrate

Implementare funcției  $f_4$  cu multiplexoar de 16 căi

$$\bar{f}_4^{FCD}(x_1, x_2, x_3, x_4) = \overline{P_0 + P_3 + P_4 + P_7 + P_8 + P_{15}} \quad (5.5)$$

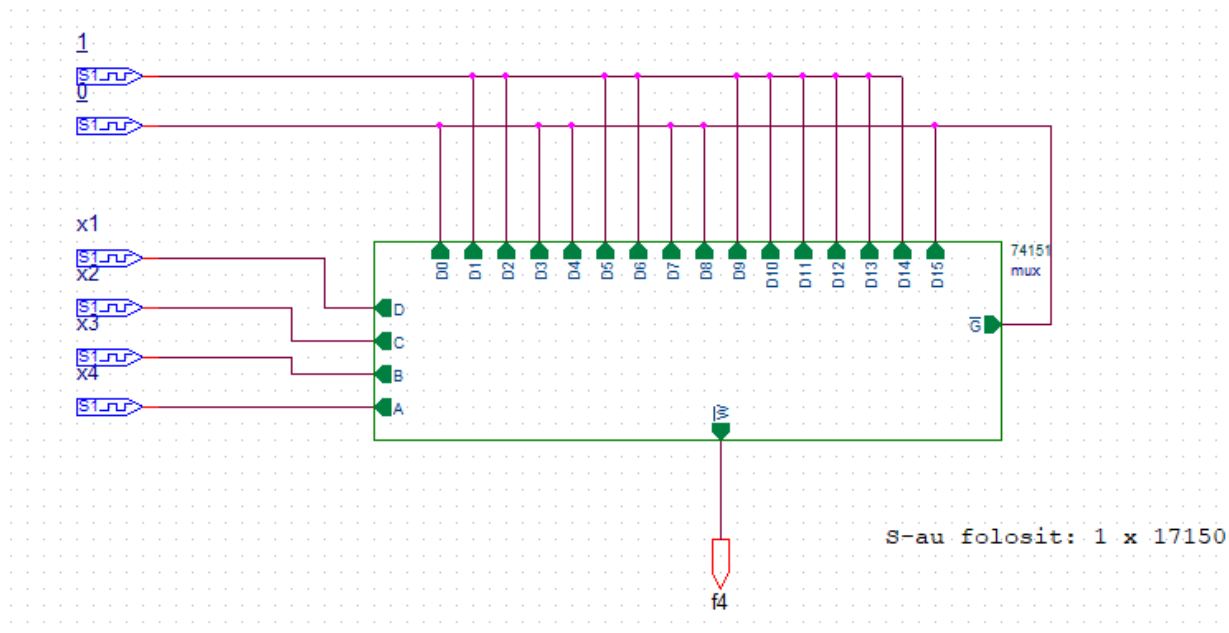


Fig 5.5 implementarea funcției  $f_4$  cu multiplexor de 16 căi

Fig x.x.x implementarea funcției  $f_4$  cu multiplexor de 16 căi

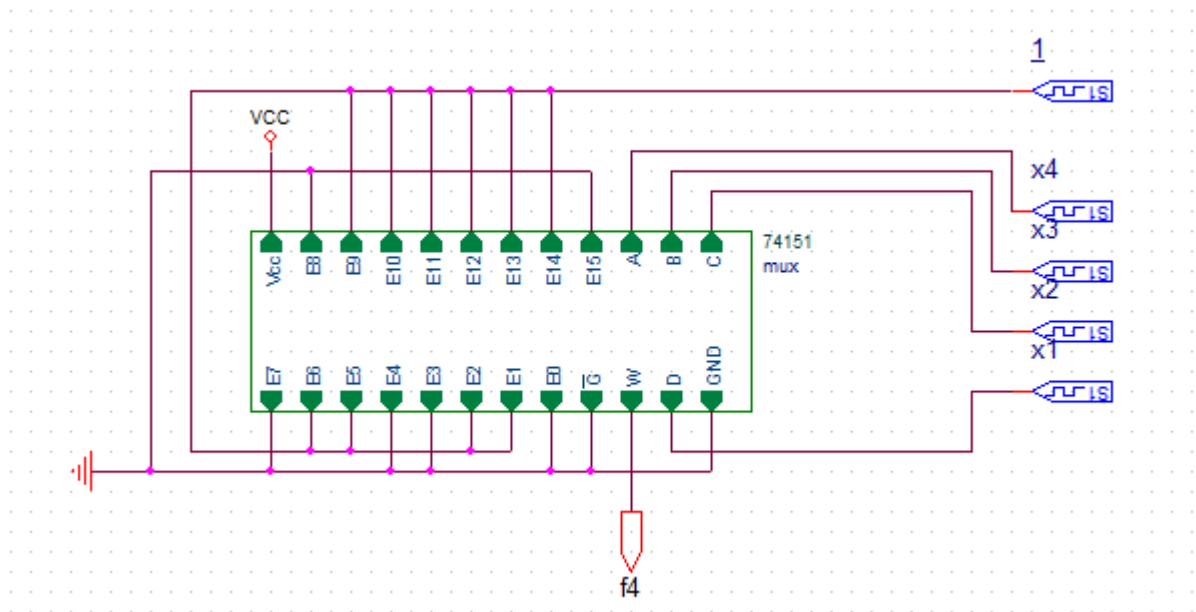


Fig 5.6 implementarea funcției  $f_4$  cu multiplexor de 16 căi cu circuite integrate

## 6. Implementare cu circuite demultiplexoare

Implementarea funcției  $f_4$  cu demultiplexor de 8 căi

$$\begin{aligned}
 f_4^{FCD}(x_1, x_2, x_3, x_4) &= \overline{x_1} \overline{x_2} \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 \overline{x_4} + x_1 \overline{x_2} x_3 \overline{x_4} = \\
 &= \overline{x_1}(\overline{x_2} \overline{x_3} x_4) + \overline{x_1}(\overline{x_2} x_3 \overline{x_4}) + \overline{x_1}(x_2 \overline{x_3} x_4) + \overline{x_1}(x_2 x_3 \overline{x_4}) + x_1(\overline{x_2} x_3 \overline{x_4}) = \\
 &= \overline{\overline{x_1} P'_1 + P'_2 + \overline{x_1} P'_5 + \overline{x_1} P'_6} \\
 &= \overline{\overline{x_1} P'_1 * P'_2 * \overline{x_1} P'_5 * \overline{x_1} P'_6}
 \end{aligned} \tag{6.1}$$

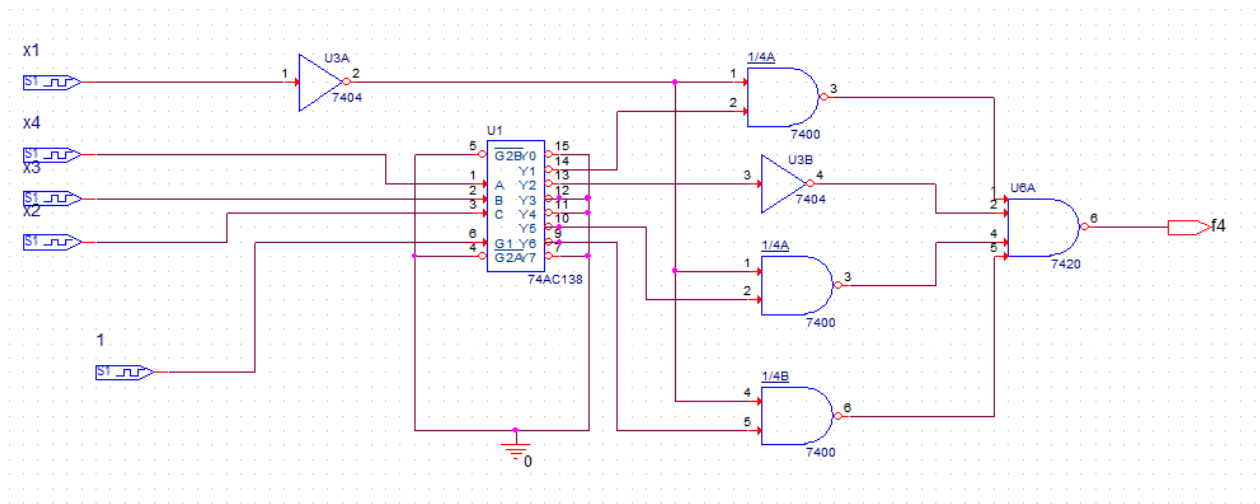


Fig 6.1 implementarea funcției  $f_4$  cu demultiplexor de 8 căi

Implementare funcției  $f_4$  cu demultiplexoare de 8 căi legate în paralel

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = P_1 + P_2 + P_5 + P_6 + P_{10} \tag{6.2}$$

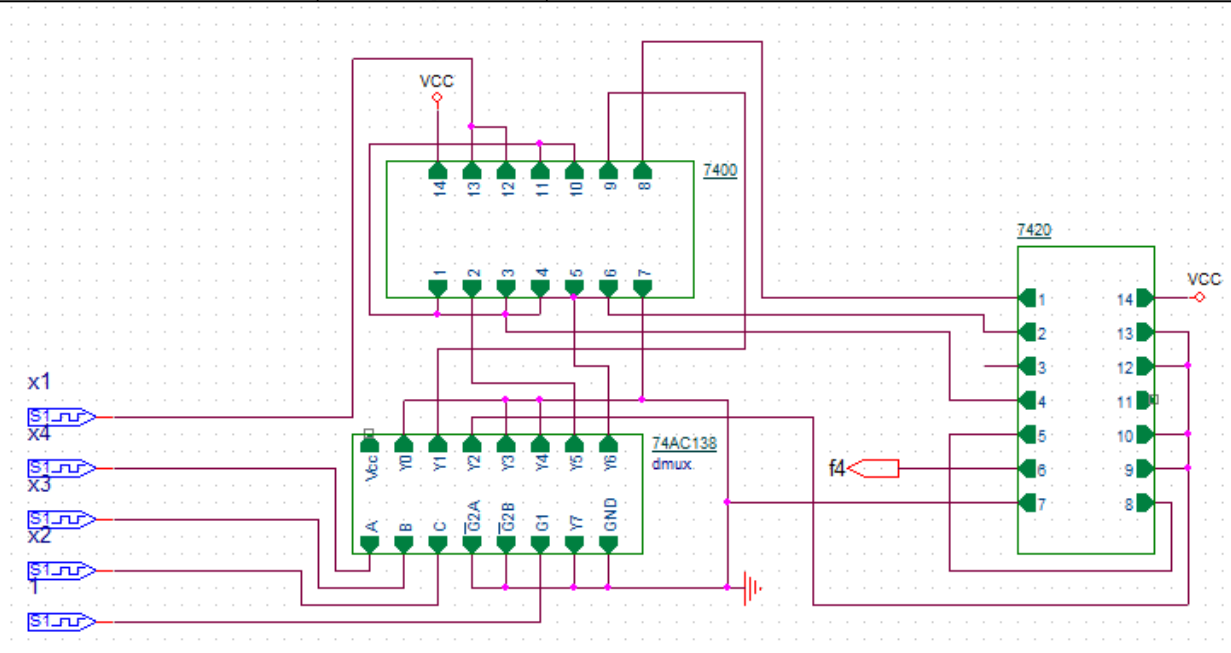


Fig 6.2implementarea funcției  $f_4$  cu demultiplexor de 8 căi cu circuite integrate

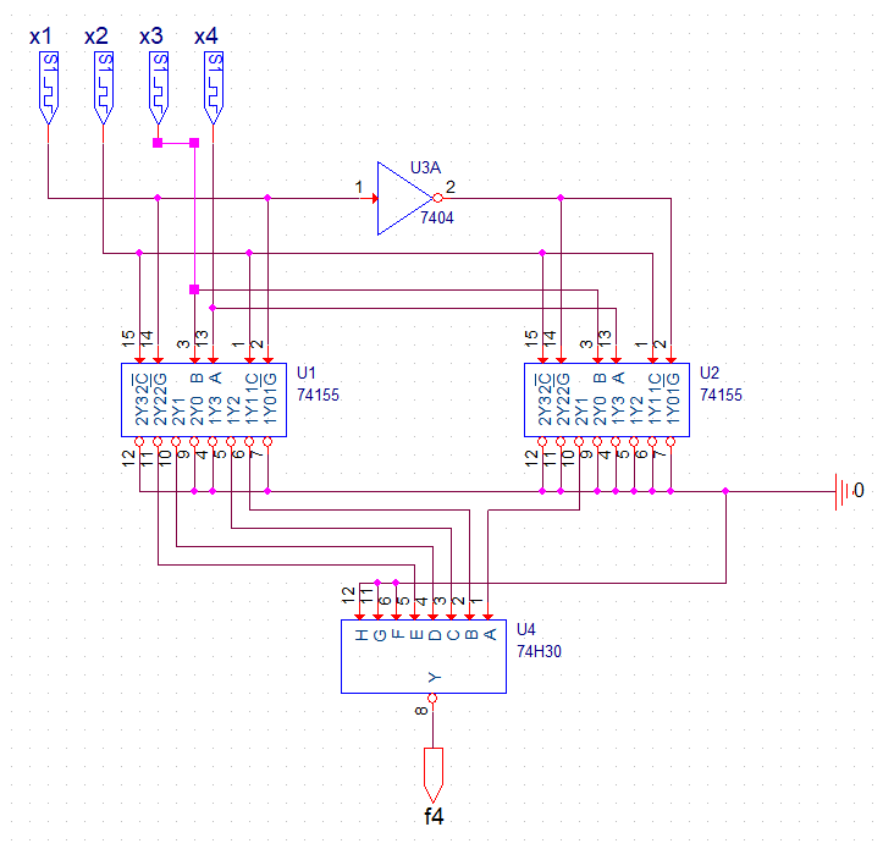


Fig 6.3implementarea funcției  $f_4$  cu doua demultiplexoare de 8 căi în paralel

Implementare funcției  $f_4$  cu demultiplexoar de 16 căi

$$f_4^{FCD}(x_1, x_2, x_3, x_4) = P_1 + P_2 + P_5 + P_6 + P_{10} \quad (6.3)$$

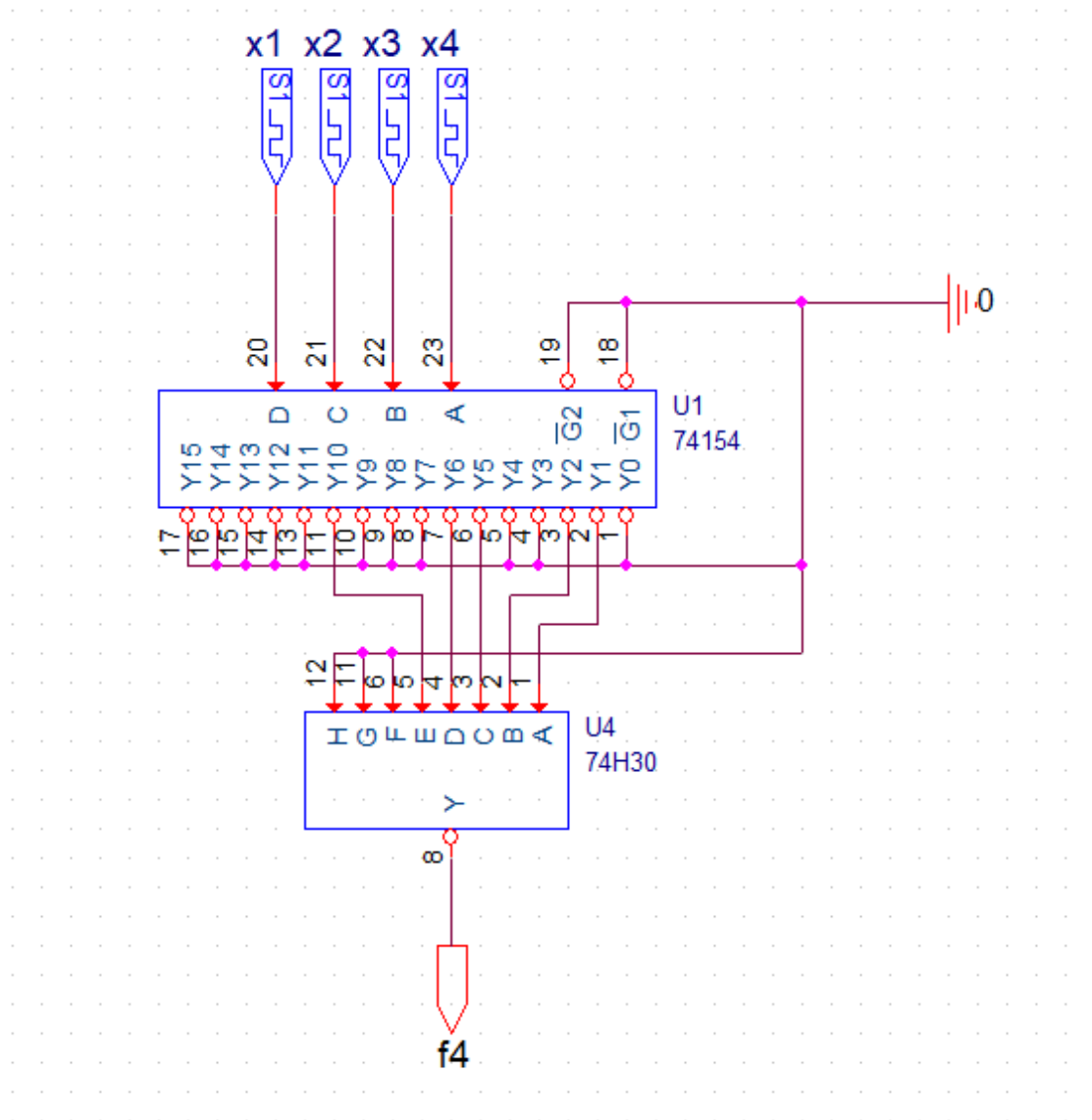


Fig 6.4 implementarea funcției  $f_4$  cu demultiplexoar de 16 căi și rețea de porți logice ȘI-NU



Implementare funcției  $f_4$  cu demultiplexor de 16 căi și rețea de porți logice ȘI

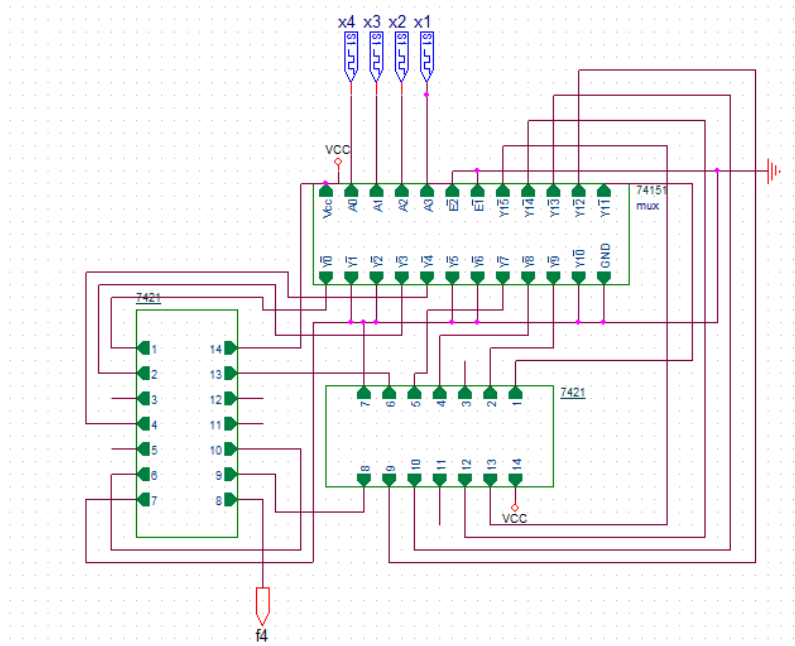


Fig 6.5 implementarea funcției  $f_4$  cu demultiplexor de 16 căi și rețea de porți logice ȘI-NU cu circuite integrate

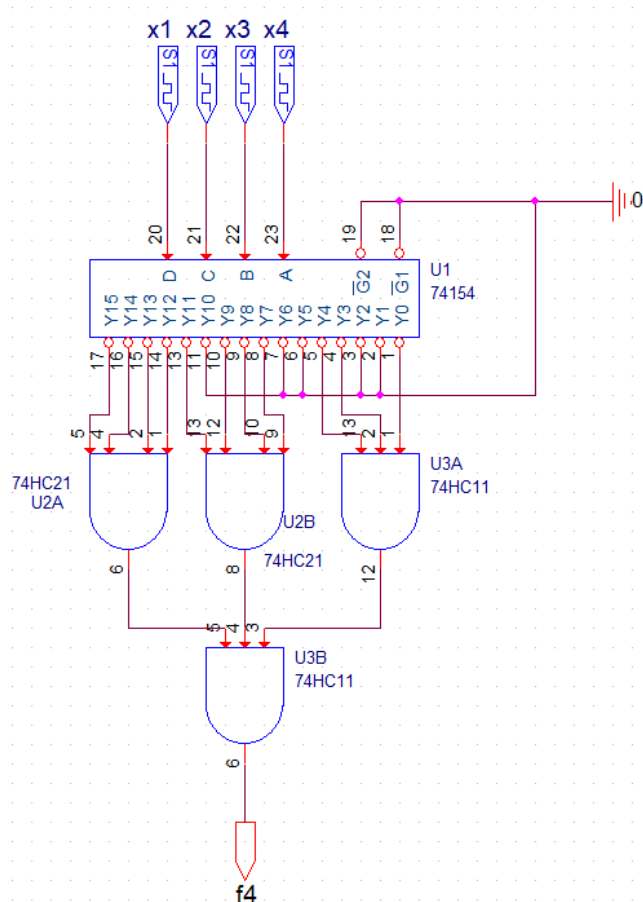


Fig 6.6 implementarea funcției  $f_4$  cu demultiplexor de 16 căi și rețea de porți logice ȘI

## 7. Calculul timpilor de propagare și al puterilor disipate

Pentru calculul timpului de propagare al fiecărui element folosit se va utiliza formula:

$$t_p = \frac{1}{2}(t_{PLH} + t_{PHL})$$

$$74LS00: t_p = 0,5 * (9 + 10) = 9,5ns$$

$$74LS10: t_p = 0,5 * (11 + 7) = 9ns$$

$$74LS20: t_p = 0,5 * (12 + 8) = 10ns$$

$$74LS00: t_p = 0,5 * (9 + 10) = 9,5ns$$

$$74LS32: t_p = 0,5 * (14 + 14) = 14ns$$

$$74LS150: t_p = 0,5 * (23 + 22) = 22,5ns$$

$$74LS151: t_p = 0,5 * (17 + 19) = 13ns$$

$$w: t_p = 0,5 * (14 + 14) = 14ns$$

$$74HC155: t_p = 0,5 * (13 + 18) = 15,5ns$$

$$74H30: t_p = 5,5ns$$

$$CD4001B: t_p = 0,5 * (110 + 120) = 115ns$$

Timpul de propagare pentru funcția  $f_4$  implementată cu porți logice SAU-NU:  $t_p = 4 * 115 = 460ns$

Timpul de propagare pentru ansamblu  $f_4$  implementată cu porți logice SAU-NU:  $t_p = 4 * 115 = 460ns$

Timpul de propagare pentru funcția  $f_4$  implementată cu MUX de 8 căi:  $t_p = 13ns$

Timpul de propagare pentru funcția  $f_4$  implementată cu DMUX de 16 căi cu porți logice SI-NU:

$$t_p = 14 + 5,5 = 19,5ns$$

Pentru calculul puterilor disipate al fiecărui element se va utiliza una din formulele:

$$\text{Pentru tehnologia TTL: } P_d = V_{cc}^2 \frac{I_{CCL} + I_{CCL}}{2}$$

$$\text{Pentru tehnologia CMOS: } P_d = V_{cc}^2 * f_i * (C_p + C_i)$$

$$74LS00: P_d = 2,5 * (0,8 + 2,4) = 8mW$$

$$74HC154: P_d = 25 * 10^5 * 138 * 10^{-12} = 0,345mW$$

$$74HC155: P_d = 25 * 10^5 * 103 * 10^{-12} = 0,2575mW$$

$$CD4001B: P_d = 25 * 10^5 * 72 * 10^{-12} = 0,18mW$$

$$74LS151: P_d = 25 * \frac{20 - 0,4}{2} = 100mW$$

$$74H30: P_d = 25 * \frac{4,2 + 10}{2} = 177,5mW$$

Puterea disipată pentru funcția  $f_4$  implementată cu porți logice SAU-NU:  $P_d = 4 * 0,18 = 0,72mW$

Puterea disipată pentru ansamblu  $f_4$  implementată cu porți logice SAU-NU:  $P_d = 4 * 0,18 = 0,72mW$

Puterea disipată pentru funcția  $f_4$  implementată cu MUX de 8 căi:  $P_d = 100mW$

Puterea disipată pentru funcția  $f_4$  implementată cu DMUX de 16 căi cu porți logice SI-NU:

$$P_d = 0,345 + 177,5 = 177,845mW$$

Nr. Crt	Implementarea	Nr. Niveluri logice	$t_p$ [Ns]	$P_d$ [mW]	Nr. CI	Nr. Porti neutilizate
1	Ansamblu porți	3	460	0,72	5	3
2	Mux 8 căi	1	13	100	1	0
3	Mux 16 căi	2	19,5	177,845	3	0

Fig 7.1 Tabel de comparare a implementarilor

Prin compararea soluțiilor de implementare obținute, ajungem la concluzia ca implementarea cu multiplexor de 8 căi este optimă.

## 8. Simulari Pspice

Programul PSpice este un program de simulare a circuitelor analogice. Programul are o parte de desenare a schemei, numită Capture CIS și o parte de analiză a circuitului (motorul de calcul - engine). Legătura dintre cele două părți este asigurată de un "parser" (analizor de lexica), care transformă desenul în formă matricială a ecuațiilor de circuit bazate pe teorema tensiunilor de noduri (TTN). Înainte de a începe simularea propriu-zisă, trebuie configurate opțiunile de simulare, numite în PSpice "Simulation Profile", ce conține analizele de circuit propuse și opțiunile pentru aceste analize. Partea finală a programului este cea de afișare grafică sau tabelară a rezultatelor.

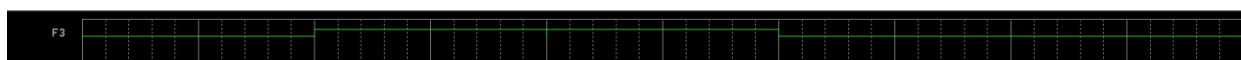


Fig 8.1 simulare funcție individuală  $f_3$

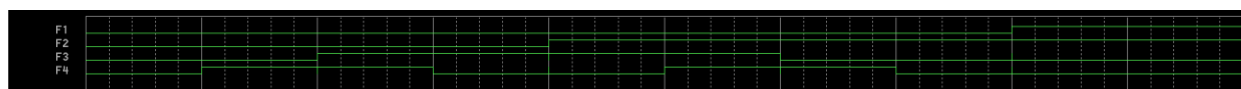


Fig 8.2 simulare ansamblu porți logice Și-NU

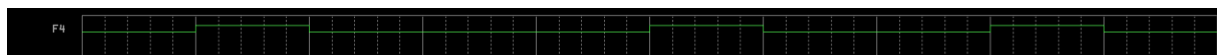


Fig 8.3 simulare mux 8 căi

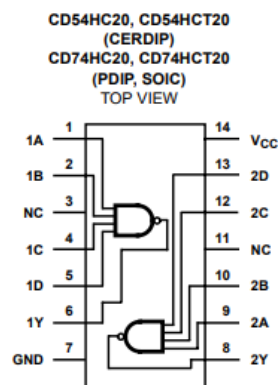
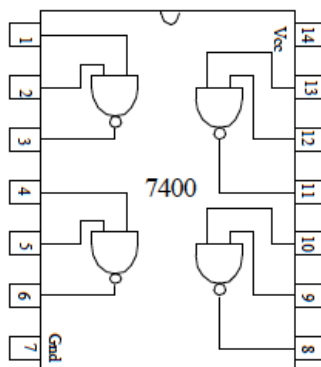
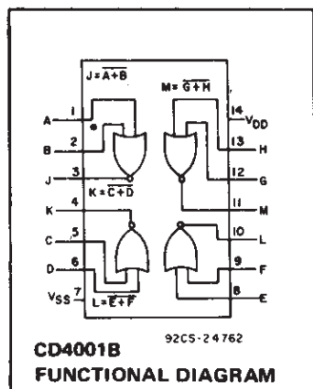


Fig 8.4 simulare dmux 16 căi Și

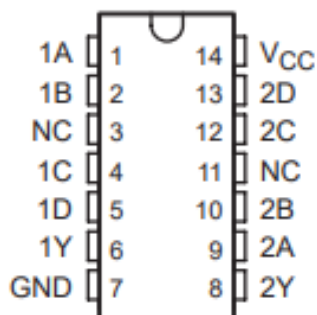
## 9. Anexa A

### Circuite integrate

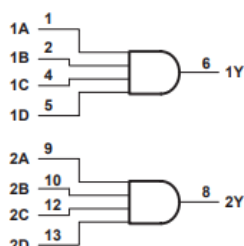
Circuitele integrate CD4001B, LS7400 și CD74HC20:



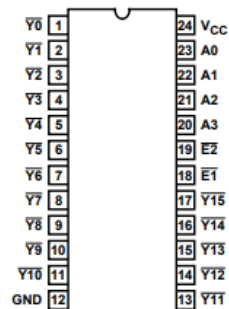
Circuitele integrate SN74AS21 cu schema logică și dmux de 16 căi CD74HC154:



logic diagram (positive logic)

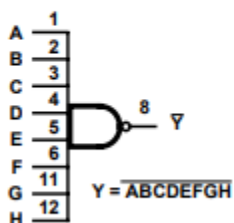
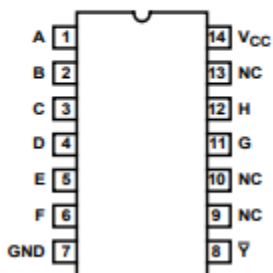


**CD54HC154, CD54HCT154 (CERDIP)**  
**CD74HC154, CD74HCT154 (PDIP, SOIC)**  
**TOP VIEW**

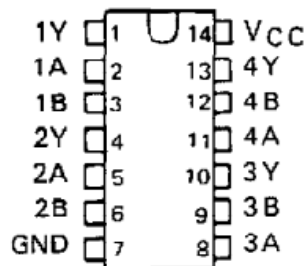


Circuitele integrate SN74HC30 cu schema logică și SN74LS02:

**CD54HC30, CD54HCT30 (CERDIP)**  
**CD74HC30 (PDIP, SOIC, SOP, TSSOP)**  
**CD74HCT30 (PDIP, SOIC)**  
**TOP VIEW**

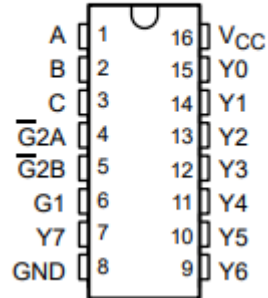


**SN7402 . . . N PACKAGE**  
**SN74LS02, SN74S02 . . . D OR N PACKAGE**  
**(TOP VIEW)**



Circuitele integrate pentru demultiplexorul de 8 căi CD74AC138 și SN74155 pentru demultiplexoarele de 8 căi în paralel:

**CD54AC138 . . . F PACKAGE  
CD74AC138 . . . E OR M PACKAGE  
(TOP VIEW)**



**SN54155, SN54156, SN54LS155A,  
SN54LS156 . . . J OR W PACKAGE  
SN74155, SN74156 . . . N PACKAGE  
SN74LS155A, SN74LS156 . . . D OR N PACKAGE  
(TOP VIEW)**

