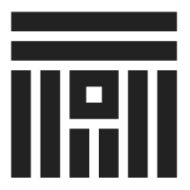
**Universitatea *Transilvania* din Brașov**

**Facultatea de Inginerie Electrică și Știința Calculatoarelor**

**Departamentul Automatică și Informatică**  

PROIECT

ANALIZA ȘI SINTEZA CIRCUITELOR NUMERICE I

Student:

Grabovenco Bogdan–Iulian

Grupa:

AIA4LF412

Îndrumător:

Prof. Dr. Ing. Moldoveanu Florin Dumitru

Brașov 2023

Cuprins

1. Tema de proiect4
2. Noțiuni introductive5
   1. Circuite logice combinaționale5
   2. Analiza circuitelor logice combinaționale5
   3. Convertoare de cod6
   4. Codul 84216
   5. Codul “2 din 5” 6
   6. Interfața TTL-CMOS 7
3. Reprezentare funcțiilor logice8
   1. Tabelul de adevăr8
   2. Forma canonică disjunctivă8
   3. Forma canonică conjunctivă9
4. Metoda diagramelor Veitch-Karnaugh10
   1. Obținerea FMD și FMC pentru f110
   2. Obținerea FMD și FMC pentru f211
   3. Obținerea FMD și FMC pentru f311
   4. Obținerea FMD și FMC pentru f412
   5. Obținerea FMD și FMC pentru f513
5. Metoda Quine-McCluskey14
   1. Obținerea FMD pentru f114
   2. Obținerea FMD pentru f316
6. Implementarea funcțiilor folosind porți logice ȘI-NU18
   1. Implementarea funcției f1 cu porți logice ȘI-NU (CI TTL)18
   2. Implementarea funcției f2 cu porți logice ȘI-NU (CI TTL)20
   3. Implementarea funcției f3 cu porți logice ȘI-NU (CI TTL)22
   4. Implementarea funcției f4 cu porți logice ȘI-NU (CI TTL)23
   5. Implementarea funcției f5 cu porți logice ȘI-NU (CI TTL)24
   6. Implementarea ansamblului de funcții cu porți logice ȘI-NU (CI TTL)25
   7. Implementarea ansamblului de funcții cu porți logice ȘI-NU (CI TTL) pentru f1 și f2 și cu porți logice SAU-NU (CI CMOS) pentru f3, f4 și f526
7. Implementarea funcției f1 utilizând MUX-uri de 8, respectiv 16 căi (CI TTL)28
   1. Circuite multiplexoare28
   2. Implementarea funcției cu MUX de 8 căi (CI TTL) 28
   3. Implementarea funcției cu MUX de 16 căi (CI TTL) 31
   4. Implementarea funcției cu 2 MUX-uri de 8 căi legate în paralel (CI TTL) 33
8. Implementarea functiei f1 utilizând DMUX-uri de 8, respectiv 16 căi (CI TTL)38
   1. Circuite demultiplexoare38
   2. Implementarea funcției cu DMUX de 8 căi utilizând porți ȘI-NU (CI TTL) 38
   3. Implementarea funcției cu DMUX de 16 căi utilizând porți ȘI-NU (CI TTL) 40
   4. Implementarea funcției cu DMUX de 8 căi legate în pararel cu porți ȘI-NU (CI TTL) 42
   5. Implementarea funcției cu DMUX de 16 căi utilizând porți ȘI (CI CMOS) 44
9. Calcularea timpilor de propagare47
10. Calcularea puterilor disipate48
11. Compararea soluțiilor de implementare50
12. Analiza prin simulare, a tuturor schemelor logice obținute utilizându-se pachetul de programe OrCAD51
13. Bibliografie57
14. Tema de proiect

**Tema de proiect nr. 24** Să se proiecteze un convertor de cod de 4 biți pentru conversia codului binar-zecimal ponderat 8421 în codul binar-zecimal neponderat „2 din 5” (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 şi diagrame Veitch-Karnaugh.
* Să se obțină formele minime disjunctive şi 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 două dintre funcțiile logice de ieşire (f1 și f3) şi prin metoda Quine-McCluskey.
* Să se implementeze fiecare funcție logică, independent, numai cu porți logice ŞI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).
* Să se implementeze ansamblul funcțiilor logice numai cu porți logice ŞI-NU (se vor utiliza circuite integrate realizate în tehnologia TTL).
* Să se implementeze ansamblul funcțiilor logice în următoarea variantă: primele două funcții logice de ieşire cu porți logice ȘI-NU (circuite integrate TTL), iar următoarele trei 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 şi porți logice ŞI-NU în prima variantă, respectiv ŞI în a doua variantă (se vor utiliza circuite integrate realizate în tehnologia CMOS).
* Să se calculeze timpii de propagare „intrare-ieşire”, 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.

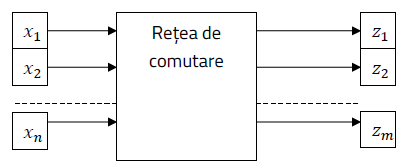
Pe schemele logice obținute se vor specifica tipul şi gradul de utilizare al fiecărui circuit integrat.

1. Noțiuni introductive
   1. Circuite logice combinaționale

Un circuit logic combinațional sau, pe scurt, CLC este un circuit de comutare care se caracterizează prin aceea că starea ieşirilor sale la un moment dat depinde numai de starea intrărilor sale la momentul considerat. Legătura între starea intrărilor şi starea ieşirilor circuitului este dată de funcțiile de transfer ale acestuia, denumite în acest caz funcții de comutare.

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.

Schema bloc generală (modelul general) al unui CLC reprezentat printr-o rețea de comutare sau schemă logică este următorul:



x1, x2, ..., xn – mărimi de intrare, iar z1,z2, ..., zm - mărimi de ieșire.

Relațiile generale dintre mărimile de intrare și cele de ieșire:

z1 = ff(x1, x2, ..., xn), z2 = f2(x1, x2, ..., xn), ..., zm= fm (x1, x2, ..., xn), iar f1, f2, ..., fm se numesc funcții proprii.

* 1. Analiza circuitelor logice combinaționale

Prin analiza unui CLC se înțelege obținerea expresiilor mărimilor de ieșire cunoscându-se setul variabilelor de intrare x1, 𝑥2, ..., xn și configurația rețelei. Înțelegem numărul și tipul de elemente logice care intră în componentă ,modul de conectare, punctul în care se aplică variabilele de intrare, numărul de nivele logice etc.

Analiza rețelelor CLC realizate cu elemente logice de tip inversor (ŞI-NU, SAU-NU): numărul maxim de elemente logice aflate între intrarea și ieșirea rețelei determină numărul de nivele

logice ale acestuia. Numerotarea lor se face de la ieșire către intrare. Într-o rețea de comutare realizată cu elemente ŞI-NU, respectiv SAU-NU, s-a observat că o variabilă de intrare apare negată în expresia ieșirii în cazul în care a parcurs un număr impar de elemente de inversare și necomplementată dacă a parcurs un număr par de astfel de elemente.

* 1. Convertoare de cod

Convertoarele de cod sunt circuite logice combinaționale având, în cazul general, n intrări şi m ieşiri, care servesc la transcrierea informației dintr-un cod binar în alt cod binar.

La baza proiectării unui convertor de cod se află tabelul de corespondențe dintre cuvintele binare ale celor două coduri. Fiecare poziție din cuvântul binar al codului din care se face conversia se notează cu o variabilă; totalitatea acestor variabile reprezintă variabilele de intrare pentru circuitul combinațional. De asemenea, se notează cu o anumită variabilă fiecare poziție din codul în care se face conversia; totalitatea acestor variabile reprezintă ieşirile circuitului combinațional.

Cu aceste notații suplimentare, tabelul de corespondențe, între cuvintele de cod, se transformă în tabel de adevăr pentru funcțiile realizate de circuit şi exprimă dependența variabilelor de ieşire de cele de intrare.

* 1. Codul 8421

Codul binar 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.

* 1. Codul “2 din 5”

Codul “2 din 5” este un cod pseudoponderat deoarece pentru cifrele zecimale 1, 2, …, 9 se pot asocial biților din secvența ponderile 74210, dar, în schimb, secvența asociată cifrei zero nu mai respectă această regulă a ponderilor. Caracteristica principală a codului “2 din 5” constă în faptul că toate secvențele binare associate cifrelor zecimale au același număr de biți semnificative (1), anume câte 2 (din cele 32 de combinații binare posibile care se pot forma cu 5 biți, numai 10 satisfac această condiție). Această proprietate oferă un criteriu de depistare a erorilor sau, altfel spus, creează posibilitatea controlului asupra transmisiei informației codificate în acest mod.

* 1. Interfața TTL-CMOS

Datorită nivelul de tensiune diferit dintre circuitele TTL (Transistor-Transistor Logic) și circuitele CMOS (Complementary Metal-Oxide-Semiconductor) este necesară utilizarea unei interfețe care să permită adaptarea nivelul de tensiune.

La circuitele TTL “1” logic este reprezentat de o tensiune care se situează între 2,4V și 5V, iar “0” logic este reprezentat de o tensiune ce cade în intervalul 0V – 0,8V. La circuitele CMOS “1” logic este reprezentat de un nivel de tensiune apropiat tensiunii de alimentare (3,3V – 5V), iar “0” logic este oferit printr-o tensiune apropiată de 0V; diferența de tensiune este reprezentată în figura 2.5.1, iar trecerea de la circuitele TTL la CMOS se face cu ajutor interfeței din figura 2.5.2.



fig.2.5.1

fig.2.5.2

1. Reprezentarea funcțiilor logice
   1. Tabelul de adevăr

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cifra zecimala | Codul 8421 | | | |  | Codul “2 din 5” | | | | |
|  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |

* 1. Forma canonică disjunctivă

Pentru FCD (forma canonică disjunctiva) se iau in considerare combinatiile pentru care functia are valoarea 1.

* 1. Forma canonică conjunctivă

Pentru a obține din tabelul de adevar FCC (forma canonica conjunctiva) se iau in considerare combinațiile pentru care o funcția are valoarea 0.

1. Metoda diagramelor Veitch-Karnaugh

Metoda diagramelor „K” este o metodă grafo-analitica foarte utilă la minimizarea funcțiilor cu un număr relativ mic de variabile. Metoda pornește de la formele canonice ale functiei. Diagramele Karnaugh se prezintă sub forma unui pătrat (sau dreptunghi) cu 2n locații, în cazul nostru cu 16 locații. În fiecare locație va aparea un termen canonic al funcției.

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

* 1. Obținerea FMD și FMC pentru funcția f1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | \* | 1 |
| 01 | 0 | 0 | \* | 1 |
| 11 | 0 | 1 | \* | \* |
| 10 | 0 | 1 | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | \* | 1 |
| 01 | 0 | 0 | \* | 1 |
| 11 | 0 | 1 | \* | \* |
| 10 | 0 | 1 | \* | \* |

fig.4.1.2

fig.4.1.1

()

* 1. Obținerea FMD și FMC pentru funcția f2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | \* | 0 |
| 01 | 0 | 1 | \* | 1 |
| 11 | 1 | 0 | \* | \* |
| 10 | 0 | 0 | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | \* | 0 |
| 01 | 0 | 1 | \* | 1 |
| 11 | 1 | 0 | \* | \* |
| 10 | 0 | 0 | \* | \* |

fig.4.2.2

fig.4.2.1

( *)( )*

* 1. Obținerea FMD și FMC pentru funcția f3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | \* | 1 |
| 01 | 1 | 1 | \* | 0 |
| 11 | 0 | 0 | \* | \* |
| 10 | 1 | 0 | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | \* | 1 |
| 01 | 1 | 1 | \* | 0 |
| 11 | 0 | 0 | \* | \* |
| 10 | 1 | 0 | \* | \* |

fig.4.3.2

fig.4.3.1

* 1. Obținerea FMD și FMC pentru funcția f4

fig.4.4.2

fig.4.4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | \* | 0 |
| 01 | 0 | 0 | \* | 0 |
| 11 | 0 | 1 | \* | \* |
| 10 | 1 | 0 | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | \* | 0 |
| 01 | 0 | 0 | \* | 0 |
| 11 | 0 | 1 | \* | \* |
| 10 | 1 | 0 | \* | \* |

* 1. Obținerea FMD și FMC pentru funcția f5

fig.4.5.2

fig.4.5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | \* | 0 |
| 01 | 1 | 0 | \* | 0 |
| 11 | 1 | 0 | \* | \* |
| 10 | 0 | 1 | \* | \* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | \* | 0 |
| 01 | 1 | 0 | \* | 0 |
| 11 | 1 | 0 | \* | \* |
| 10 | 0 | 1 | \* | \* |

1. 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 Veitch-Karnaugh) devin greu de utilizat.

* 1. Obținerea FMD pentru f1

**Tab. 2.1 Prima etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 8 | 1000🗸 |  |
| 2 | 6 | 0110🗸 |  |
| 9 | 1001🗸 |  |
| 10\* | 1010🗸 |  |
| 12\* | 1100🗸 |  |
| 3 | 7 | 0111🗸 |  |
| 11\* | 1011🗸 |  |
| 13\* | 1101🗸 |  |
| 14\* | 1110🗸 |  |
| 4 | 15\* | 1111🗸 |  |

**Tab. 2.2 A doua etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 8,9 | 100-🗸 |  |
| 8,10 | 10-0🗸 |  |
| 8,12 | 1-00🗸 |  |
| 2 | 6,7 | 011-🗸 |  |
| 6,14 | -110🗸 |  |
| 9,11 | 10-1🗸 |  |
| 9,13 | 1-01🗸 |  |
| 10,11 | 101-🗸 |  |
| 10,14 | 1-10🗸 |  |
| 12,13 | 110-🗸 |  |
| 12,14 | 11-0🗸 |  |
| 3 | 7,15 | -111🗸 |  |
| 11,15 | 1-11🗸 |  |
| 13,15 | 11-1🗸 |  |
| 14,15 | 111-🗸 |  |

**Tab. 2.3 A treia etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 8,9,10,11 | 10--🗸 |  |
| 8,9,12,13 | 1-0-🗸 |  |
| 8,10,12,14 | 1--0🗸 |  |
| 2 | 6,7,14,15 | -11- | a |
| 9,11,13,15 | 1--1🗸 |  |
| 10,11,14,15 | 1-1-🗸 |  |
| 12,13,14,15 | 11--🗸 |  |

**Tab. 2.4 A patra etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 8,9,10,11,12,13,14,15 | 1--- | b |

**Tab. 2.5 Tabelul acoperirilor pentru functia f1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Termeni canonici  Implicanți primi | 6 | 7 | 8 | 9 | IPE |
| a – 6,7,14,15 | \* | \* |  |  | 🗸 |
| b – 8,9,10,11,12,13,14,15 |  |  | \* | \* | 🗸 |

* 1. Obținera FMD pentru f3

**Tab. 2.1 Prima etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 1 | 1000🗸 |  |
| 2 | 0010🗸 |  |
| 8 | 1000🗸 |  |
| 2 | 5 | 0101🗸 |  |
| 10\* | 1010🗸 |  |
| 12\* | 1100🗸 |  |
| 3 | 11\* | 1011🗸 |  |
| 13\* | 1101🗸 |  |
| 14\* | 1110🗸 |  |
| 4 | 15\* | 1111🗸 |  |

**Tab. 2.2 A doua etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 1,5 | 0-01 | a |
| 2,10 | -010 | b |
| 8,10 | 10-0🗸 |  |
| 8,12 | 1-00 | c |
| 2 | 5,13 | -1-01 |  |
| 10,11 | 101-🗸 |  |
| 10,14 | 1-10🗸 |  |
| 12,13 | 110-🗸 |  |
| 12,14 | 11-0🗸 |  |
| 3 | 11,15 | 1-11🗸 |  |
| 13,15 | 11-1🗸 |  |
| 14,15 | 111-🗸 |  |

**Tab. 2.3 A treia etapă pentru determinarea implicanților primi**

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indici |  | Implicanți primi |
| 1 | 8,10,12,14 | 1--0 | d |
| 2 | 10,11,14,15 | 1-1- |  |
| 12,13,14,15 | 11-- | e |

**Tab. 2.5 Tabelul acoperirilor pentru functia f3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Termeni canonici  Implicanți primi | 1 | 2 | 5 | 8 | IPE |
| a – 1,5 | \* |  | \* |  | 🗸 |
| b – 2,10 |  | \* |  |  | 🗸 |
| c – 8,12 |  |  |  | \* | 🗸 |
| d – 8,10,12,14 |  |  |  | \* |  |
| e – 12,13,14,15 |  |  |  |  |  |

1. Implementarea funcțiilor folosind porți logice ȘI-NU
   1. Implementarea funcției f1 cu porți logice ȘI-NU (CI TTL)

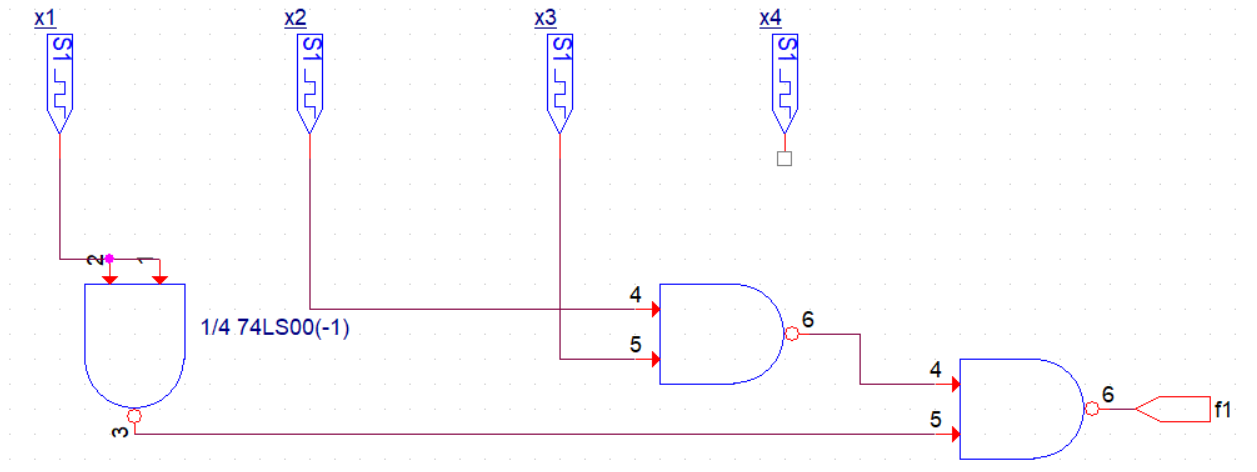


fig.6.1.1

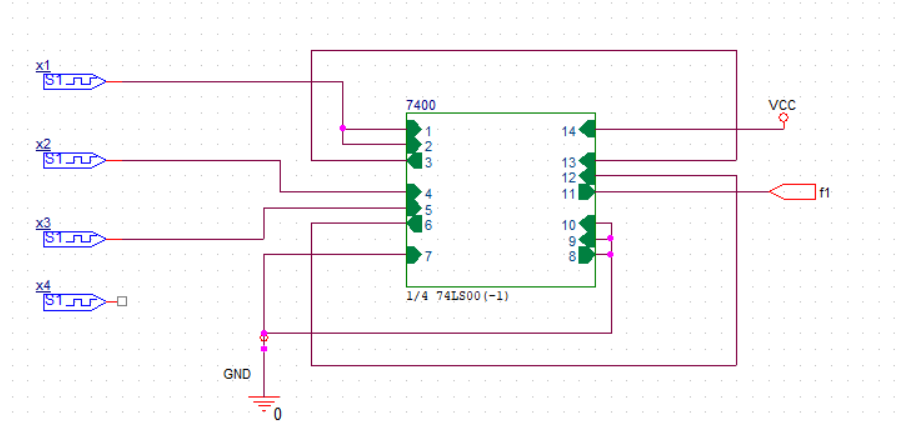


fig.6.1.2

Pentru implementare cu circuite integrate s-a folosit un CI 74LS00, descris de următoarele figuri, iar porțile nefolosite, atât intrările, cât și ieșirile sunt legate la GND cum se întâmplă și în fig. 6.1.2, unde intrările 9, 10 și ieșirea 8 sunt legate la GND.

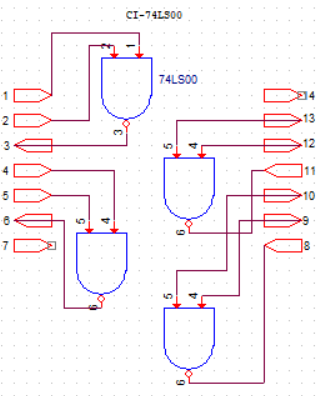


fig.6.1.3

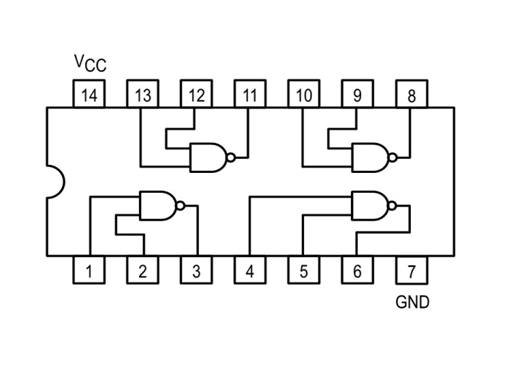


fig.6.1.4

* 1. Implementarea funcției f2 cu porți logice ȘI-NU (CI TTL)

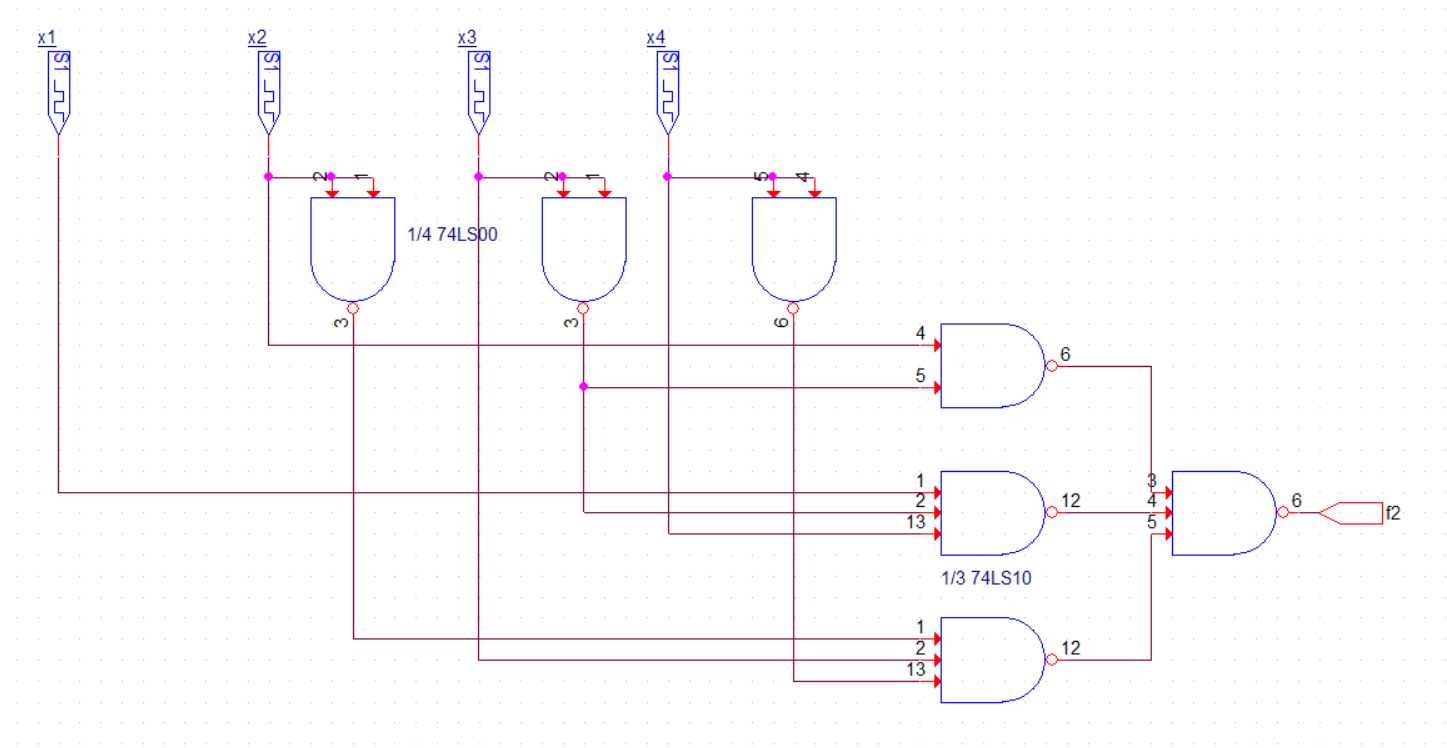


fig.6.2.1

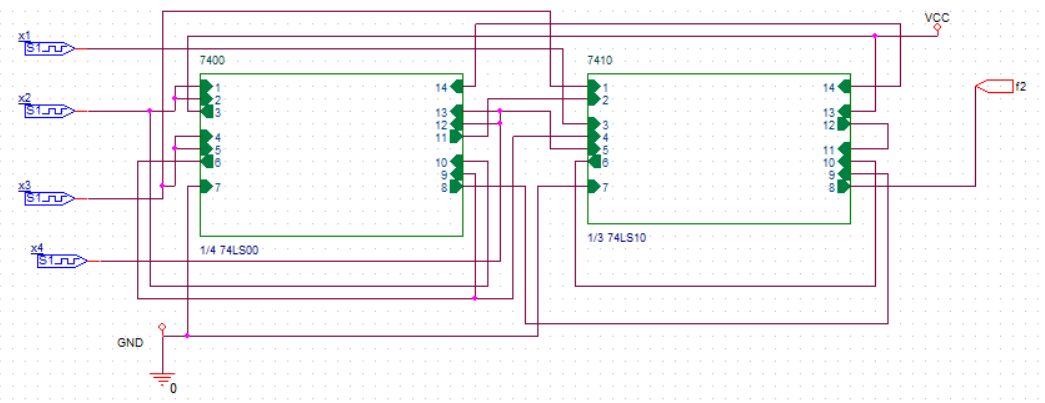


fig.6.2.2

Pentru implementare cu circuite integrate s-a folosit adițional un CI 74LS10, descris de următoarele figuri:

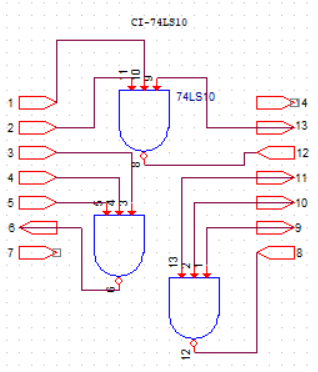


fig.6.2.3

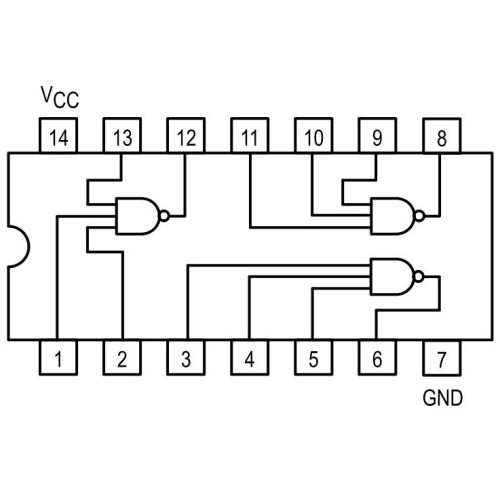


fig.6.2.4

* 1. Implementarea funcției f3 cu porți logice ȘI-NU (CI TTL)

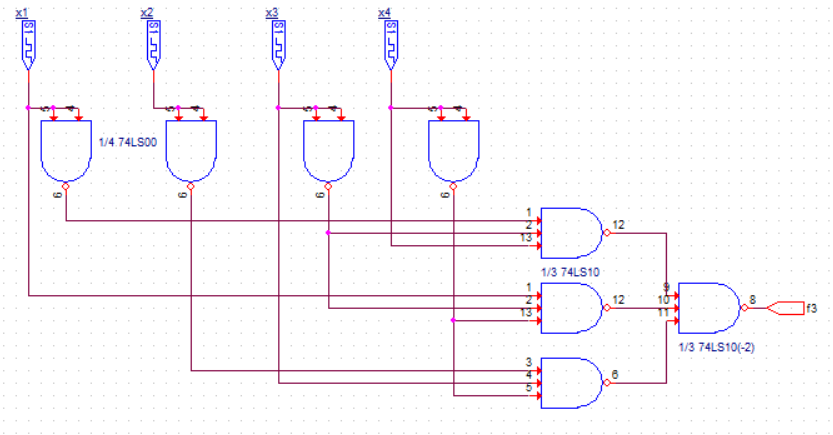


fig.6.3.1

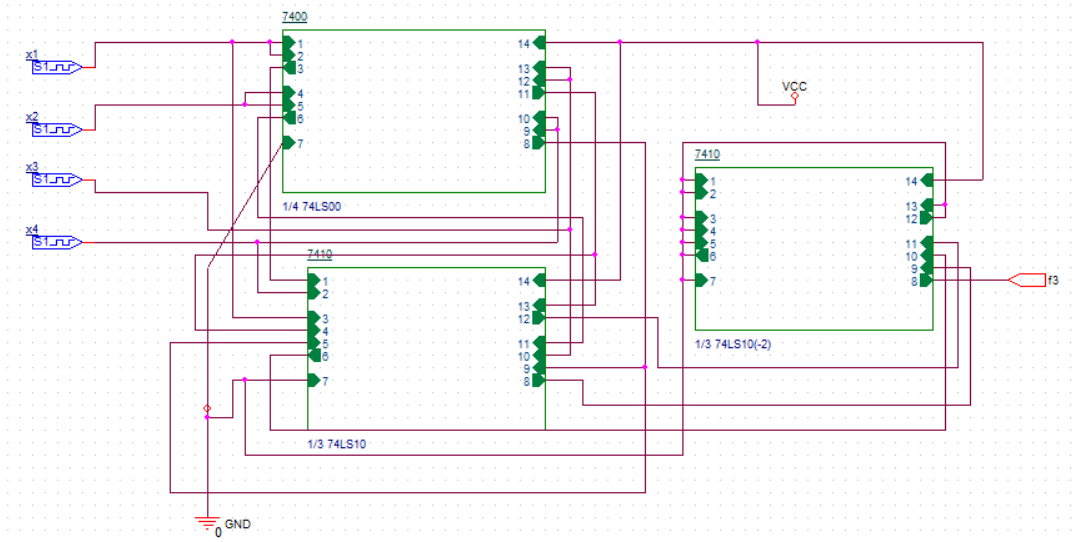


fig.6.3.2

* 1. Implementarea funcției f4 cu porți logice ȘI-NU (CI TTL)

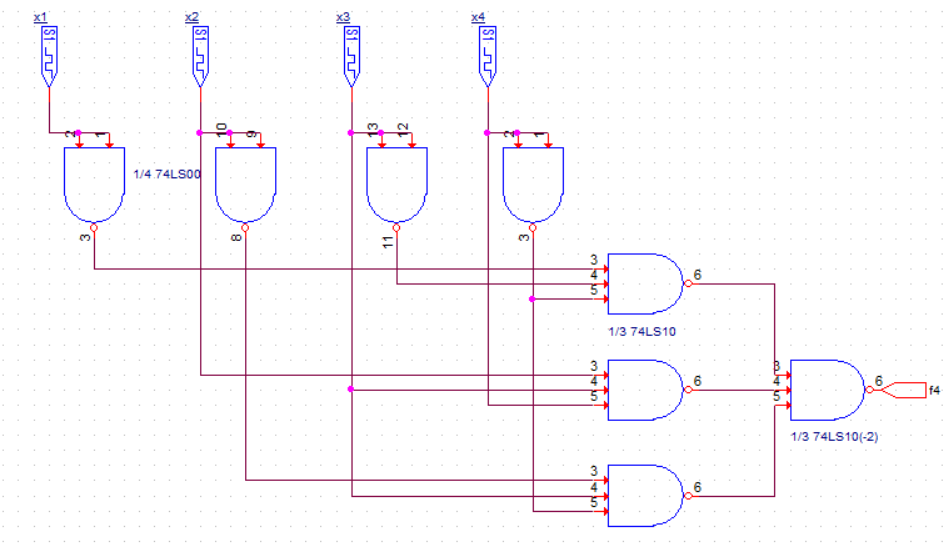


fig.6.4.1

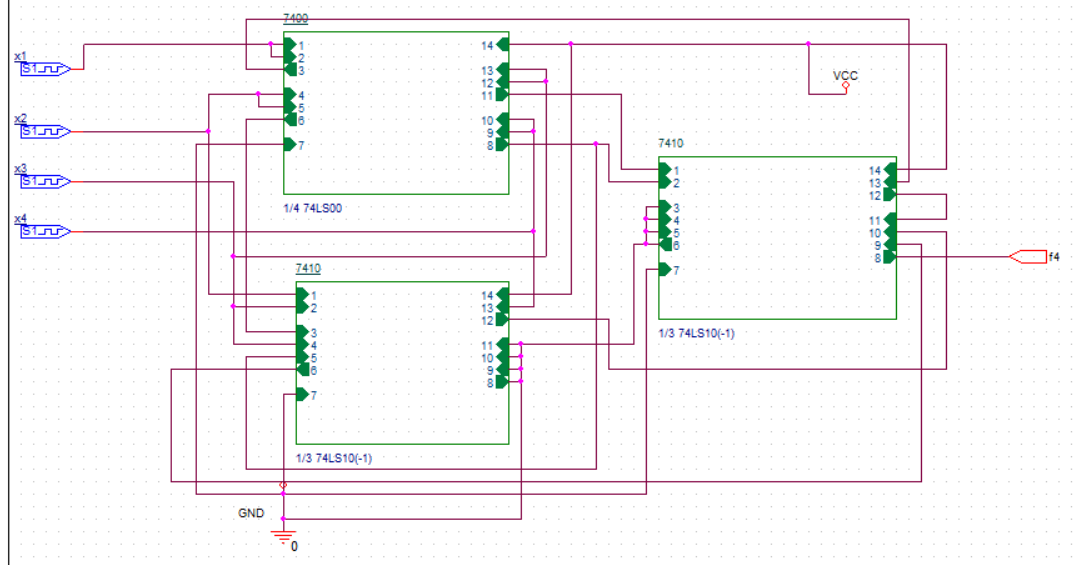


fig.6.4.2

* 1. Implementarea funcției f5 cu porți logice ȘI-NU (CI TTL)

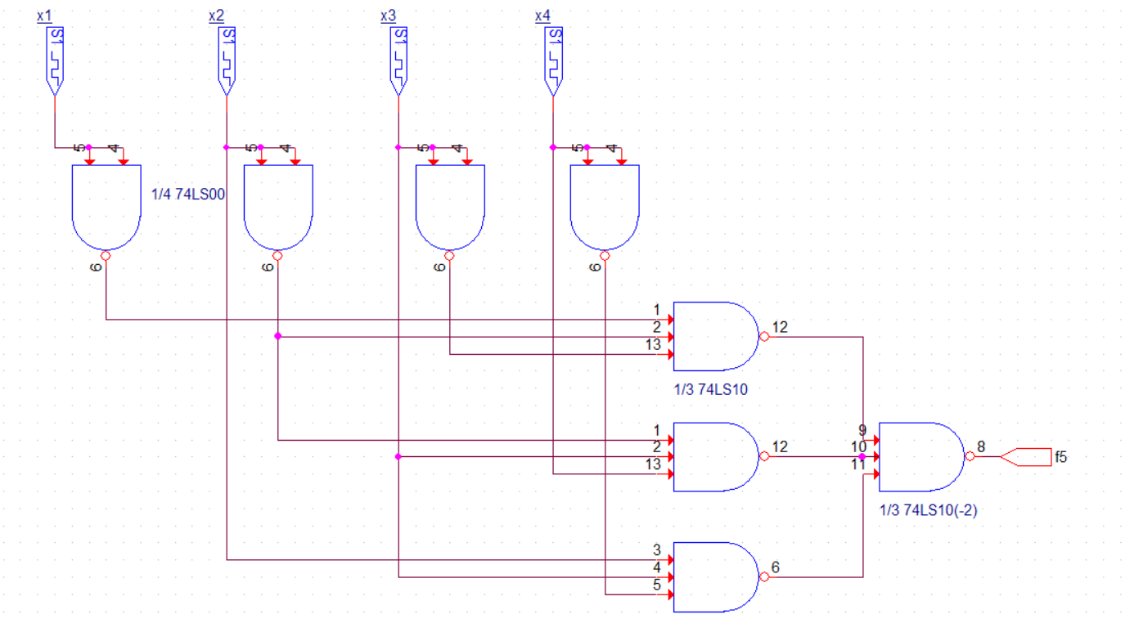


fig.6.5.1

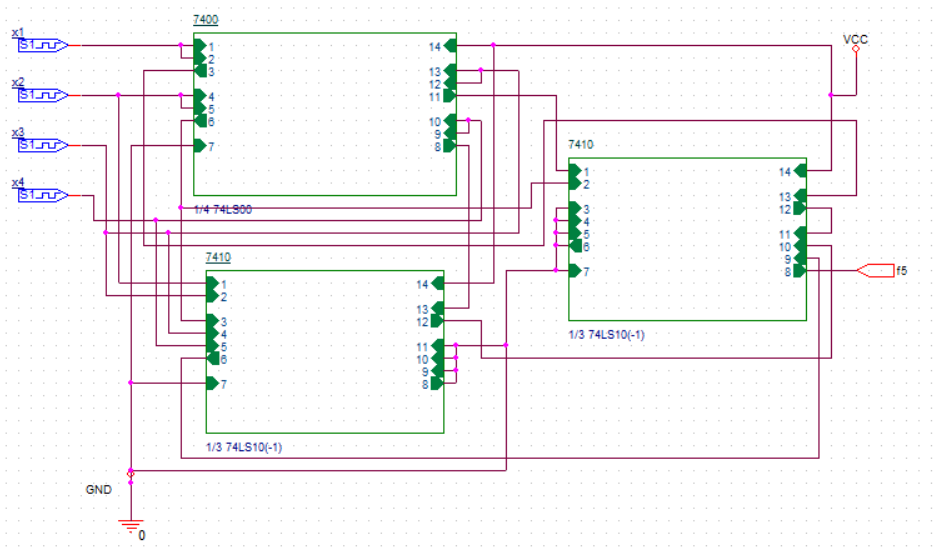


fig.6.5.2

* 1. Implementarea ansamblului de funcții cu porți logice ȘI-NU (CI TTL)

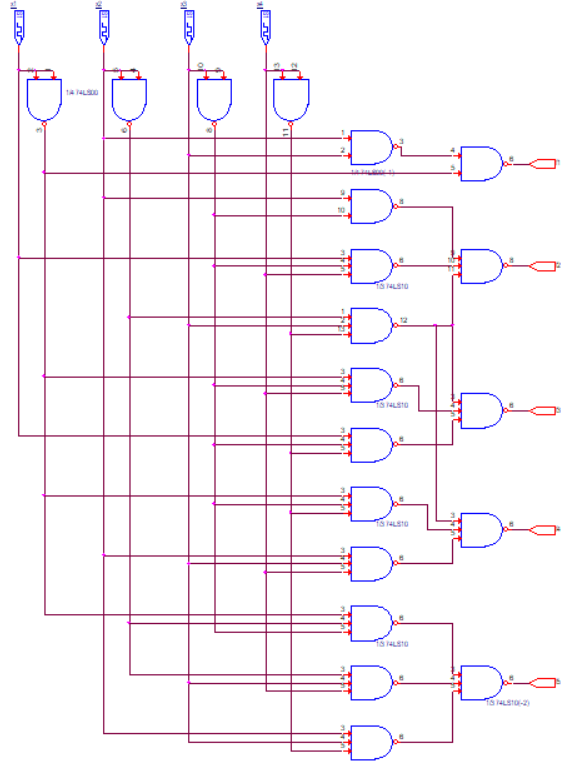


fig.6.5.1

2x74LS00(-1)

e

121e

5x74LS10(-2)

e

121e

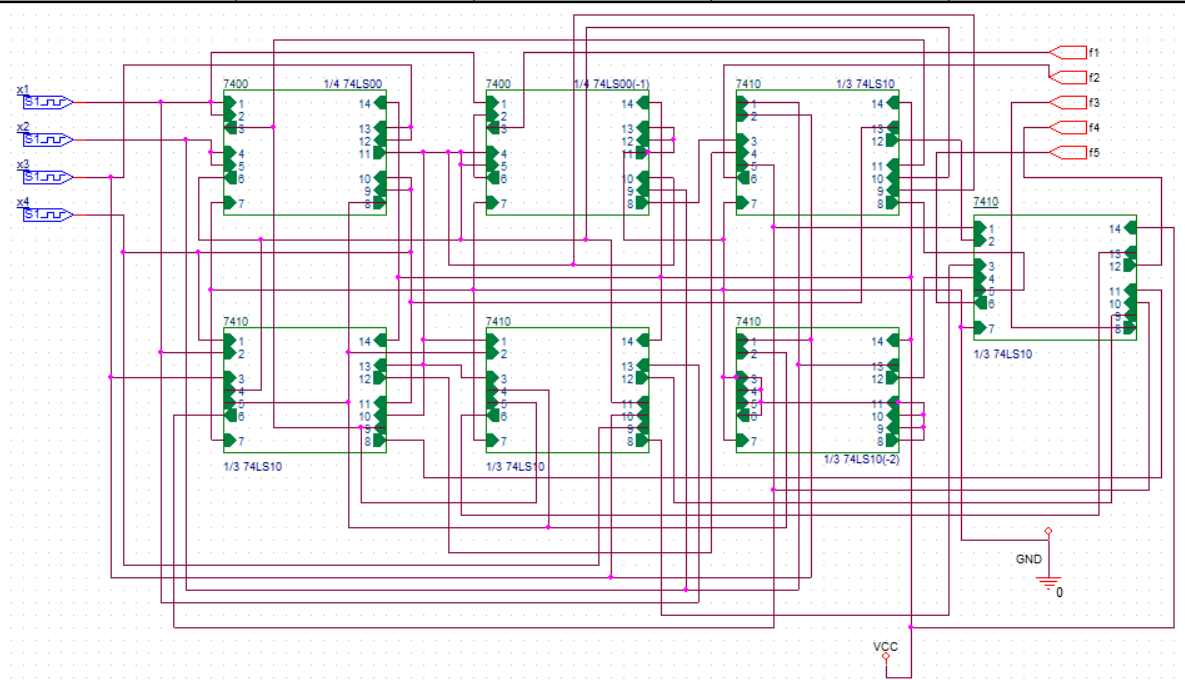


fig.6.5.2

2x74lS00(-1)

e

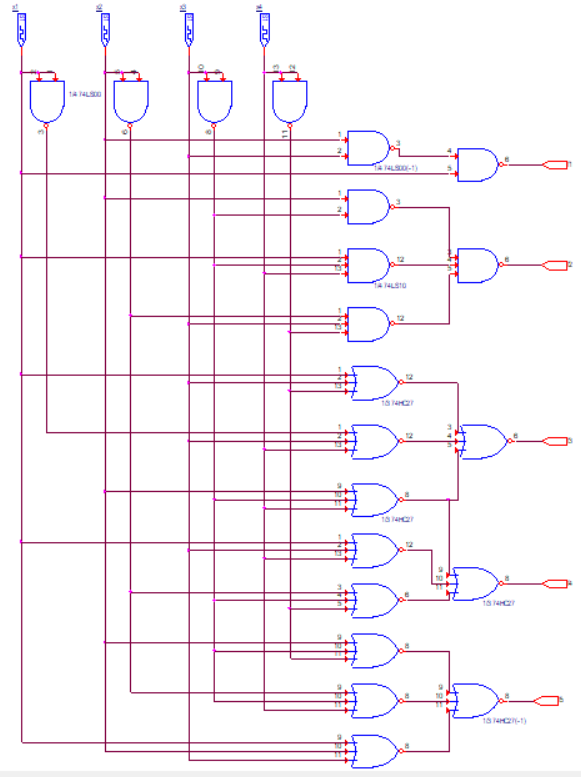
121e

5x74LS10(-2)

e

121e

* 1. Implementarea ansamblului de funcții cu porți logice ȘI-NU (CI TTL) pentru f1 și f2 și cu porți logice SAU-NU (CI CMOS) pentru f3, f4 și f5



1x74LS10

e

121e

4x74HC27(-1)

e

121e

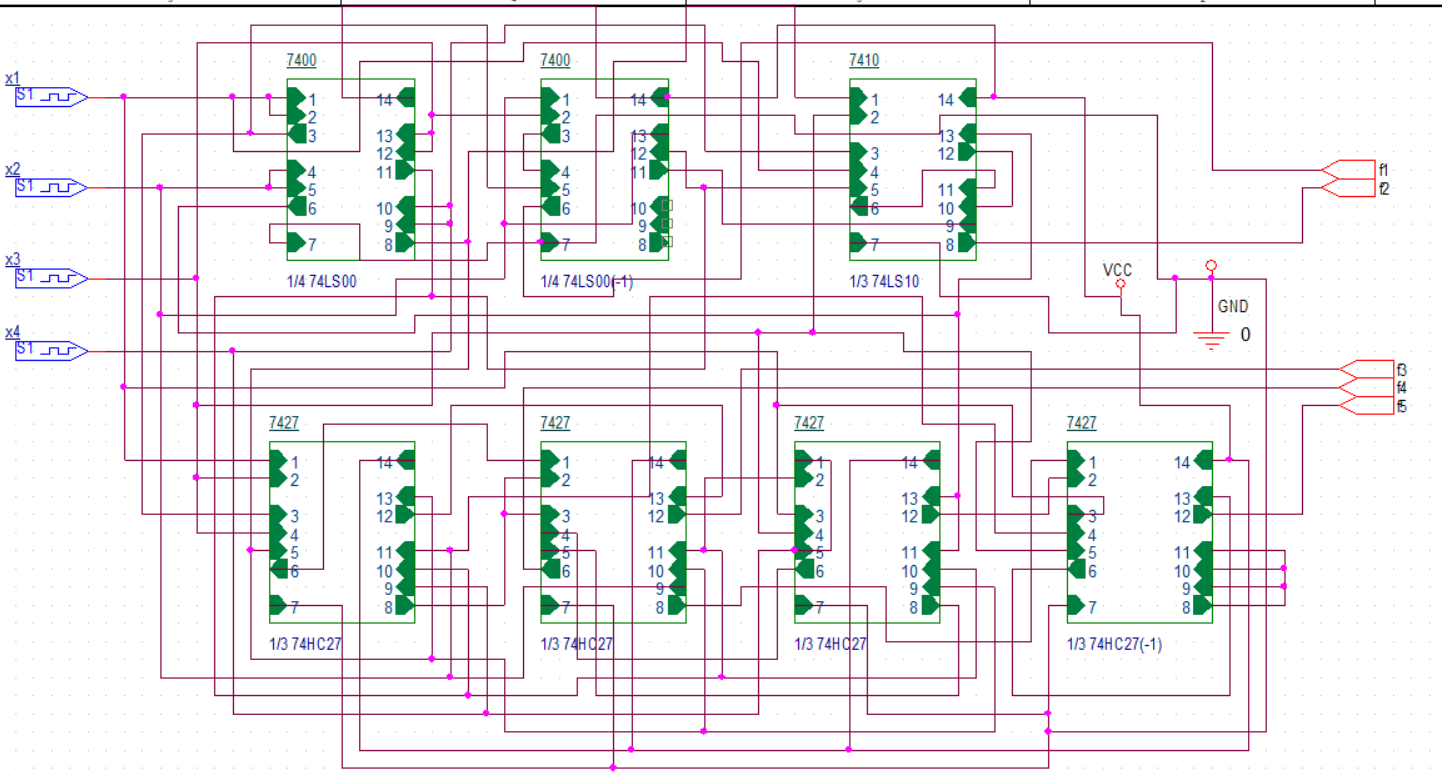
2x74LS00(-1)

e

121e

Interfață TTL-CMOS

fig.6.7.1



Interfață TTL-CMOS

fig.6.7.2

2x7400(-1)

e

121e

1x7410

e

121e

4x7427(-1)

e

121e

1. Implementarea funcției f1 utilizând MUX-uri de 8, respectiv 16 căi (CI TTL)
   1. Circuite multiplexoare

Un circuit de multiplexare este un circuit logic combinațional (CLC) care, în cazul general, are 2n intrări de date (I2n-1 … I2 I1 I0), n intrări de selecție (S0 S1 … Sn+1) şi o ieşire (Z). Expresia ieşirii Z la un moment dat este dată de intrare Ik, k=0, … ,2n-1 unde k reprezintă echivalentul zecimal al numărului binar dat de stările 1 şi 0 ale intrărilor de selecție: k=Sn-1, Sn-2, … , S1, S0.

* 1. Implementarea funcției cu MUX de 8 căi (CI TTL)

La implementarea unei funcții cu un multiplexor de 8 căi şi 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. Pentru a obține însă termenii 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echiv.  zec. | Variabilele de intrare | | | | Funcția |
|  |  |  |  |  |
| 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 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | \* |
| 11 | 1 | 0 | 1 | 1 | \* |
| 12 | 1 | 1 | 0 | 0 | \* |
| 13 | 1 | 1 | 0 | 1 | \* |
| 14 | 1 | 1 | 1 | 0 | \* |
| 15 | 1 | 1 | 1 | 1 | \* |

Astfel pentru :

=000 : , oricare ar fi =>D0=0

=001 : , oricare ar fi =>D1=0

=010 : , oricare ar fi =>D2=0

=011 : , oricare ar fi =>D3=1

=100 : , oricare ar fi =>D4=1

=101 : , oricare ar fi =>D5=0

=110 : , oricare ar fi =>D6=0

=111 : , oricare ar fi =>D7=0

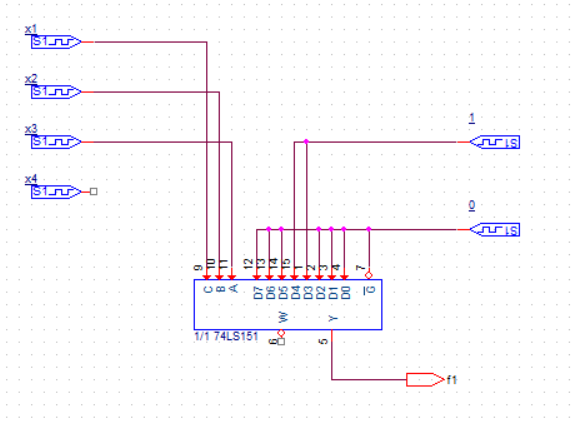


fig.7.1.1

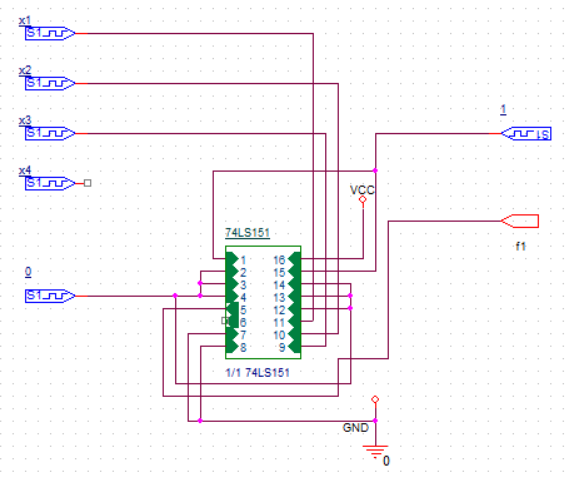


fig.7.1.2

* 1. Implementarea funcției cu MUX de 16 căi (CI TTL)

La implementarea cu multiplexoare cu 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 termeni canonici care nu sunt prezenți în expresia funcției, iar termenii canonici prezenți în expresia funcției se vor lega la “0” logic.

iar

sunt combinații indiferente.

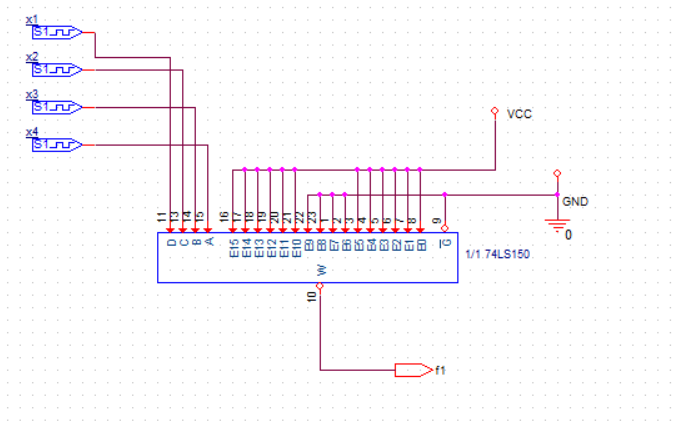


fig.7.3.1

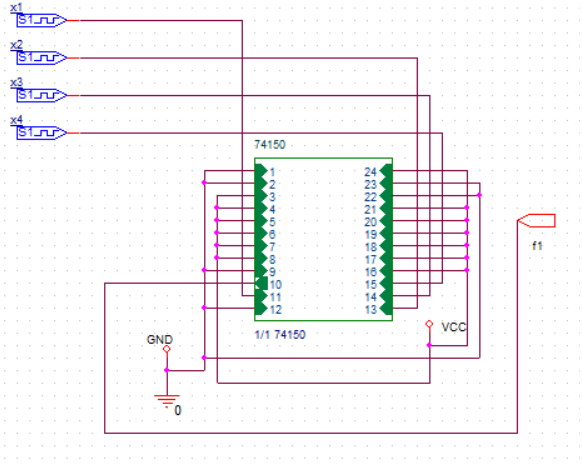


fig.7.3.2

Pentru implementare cu circuite integrate s-a folosit un CI74LS150, după cum urmează în figura de mai jos.

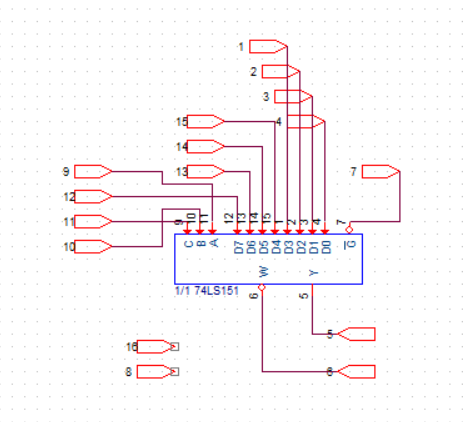


fig.7.3.3

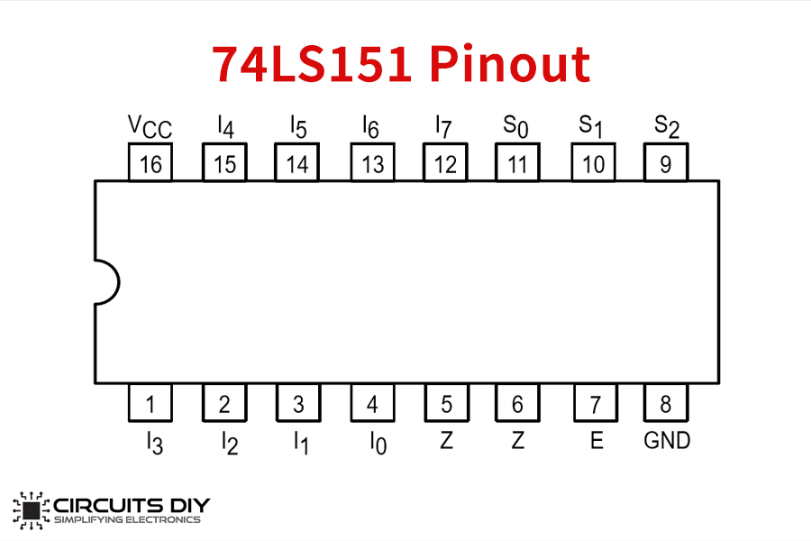


fig.7.3.4

* 1. Implementarea funcției cu 2 MUX-uri de 8 căi, legate în paralel (CI TTL)

La implementarea unei funcții logice cu 2 multiplexoare de 8 căi legate în paralel variabilele funcției de implementate se aplică astfel : prima variabilă (cea mai semnificativă) se aplică pe intrarea de strobare la primul multiplexor aşa cum este ea, iar la al doilea multiplexor negată, celelalte variabile aplicându-se în ordinea semnificativității lor pe intrările de date.

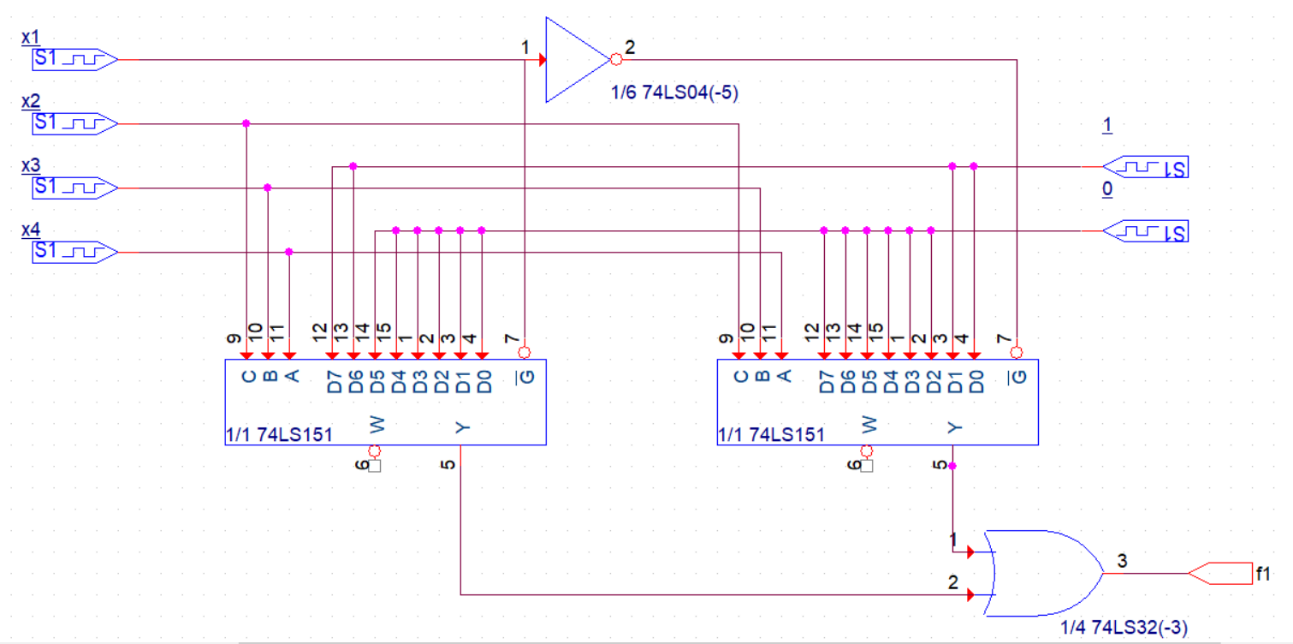


fig.7.4.1

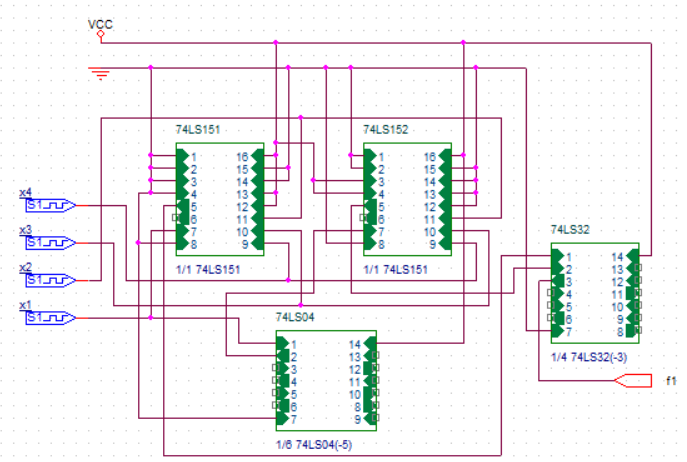


fig.7.4.2

Pentru implementare cu circuite integrate s-a folosit un CI74LS04 , un CI74LS32 și 2 CI74LS151, circuite ce se regăsesc în figurile de mai jos. CI74LS151 s-a folosit, de asemenea, în cazul implementării funcției f1 cu doar un MUX de 8 căi (fig. 7.4.7, fig. 7.4.8).

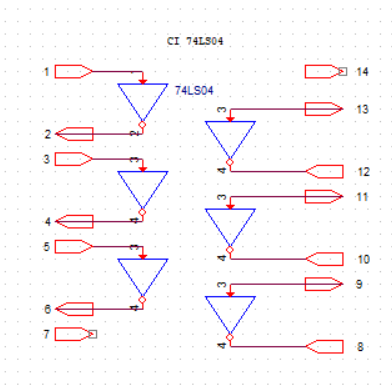


fig.7.4.3

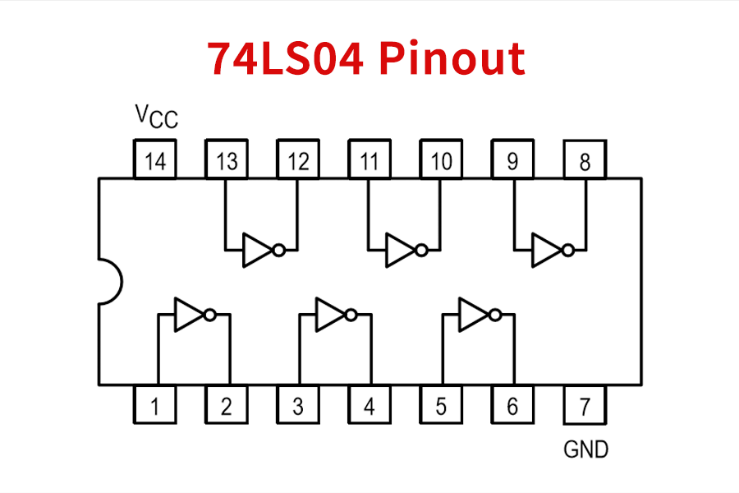


fig.7.4.4

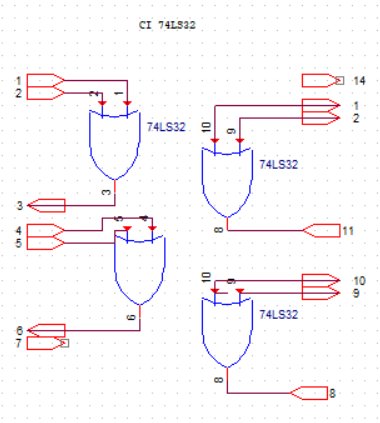


fig.7.4.5

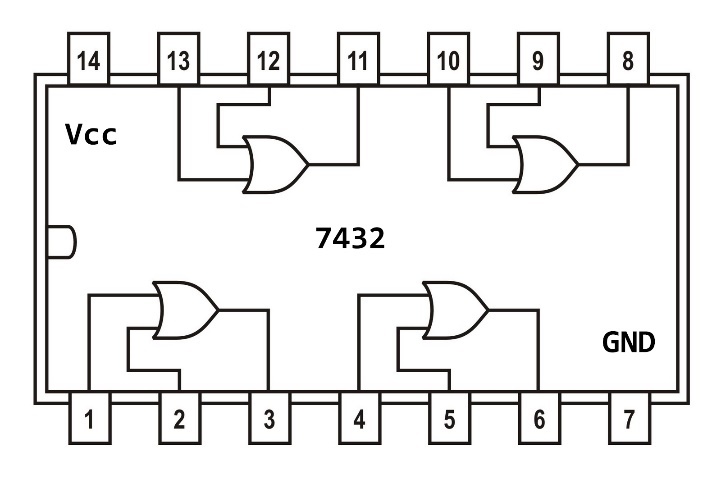


fig.7.4.6

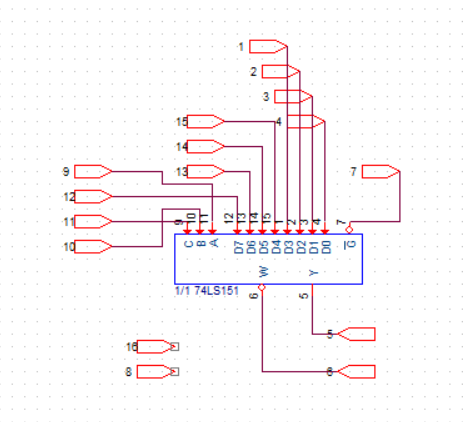


fig.7.4.7

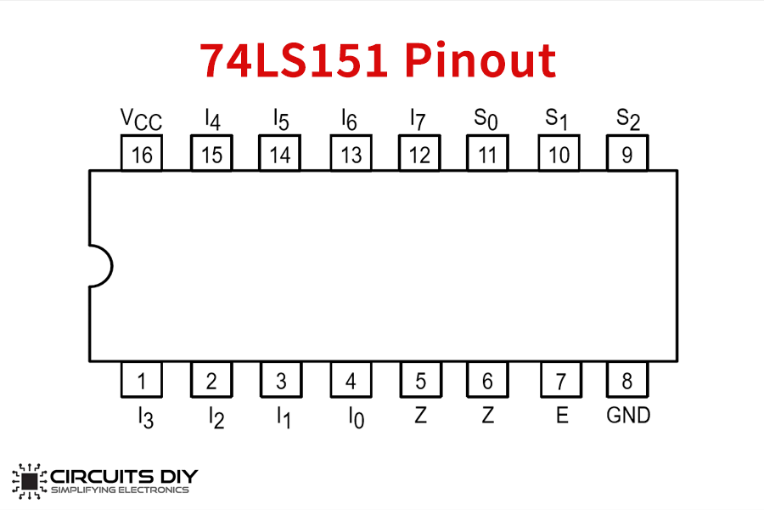


fig.7.4.8

1. Implementarea funcției f1 utilizând DMUX-uri de 8, respectiv 16 căi (CI TTL)
   1. Circuite demultiplexoare

Demultiplexoarele sunt circuite combinaționale care, în cazul general au o intrare de date I, n intrări de selecție S0, S1, …, Sn-1 si 2ieşiri Z0, Z1, …, Z2.

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

* 1. Implementarea funcției cu DMUX de 8 căi utilizând porți ȘI-NU (CI TTL)

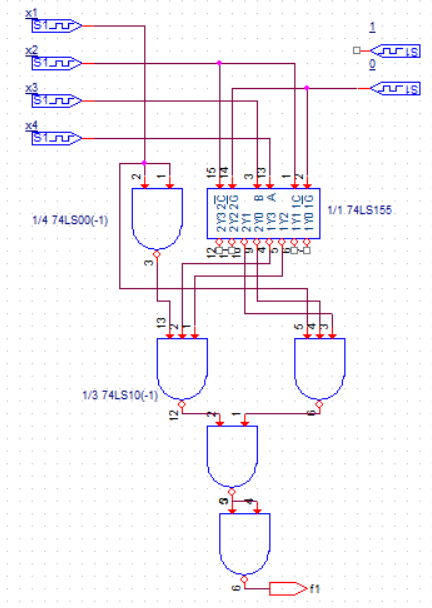


fig.8.2.1

Unde 2Y0= P’0, 2Y1= P’1, 2Y2= P’2, 2Y3= P’3,1Y0= P’4,1Y1= P’5, 1Y2= P’6, 1Y3 =P’7

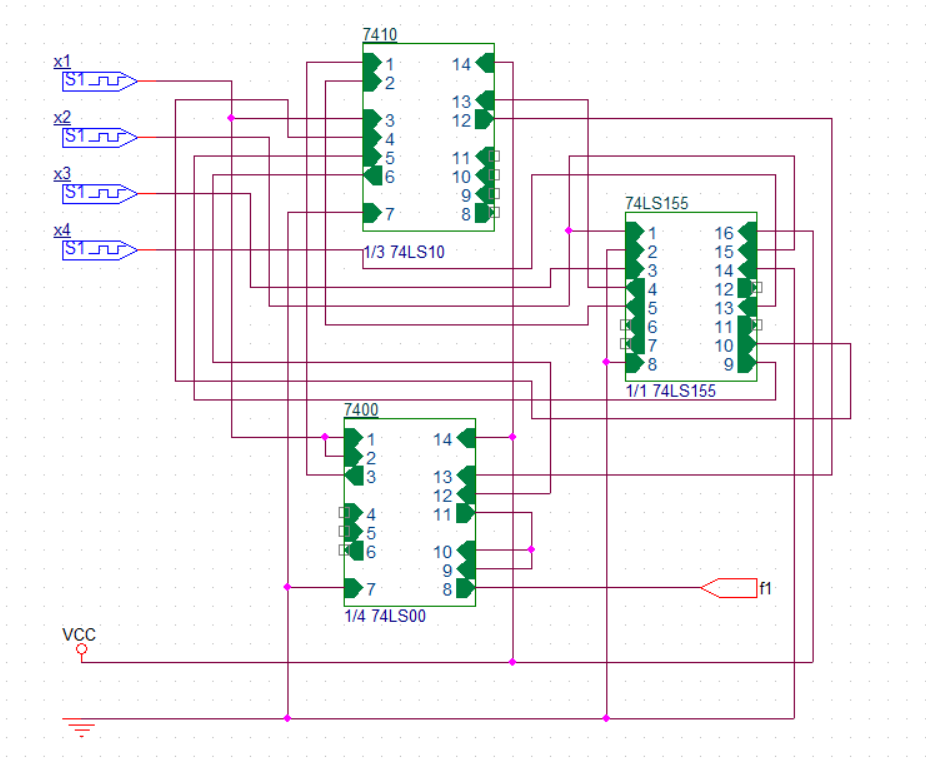


fig.8.2.2

Pentru implementarea folosind circuite integrate am folosit adițional circuitelor folosite la implementări anterioare CI74LS155, circuit reprezentat în figurele de mai jos.

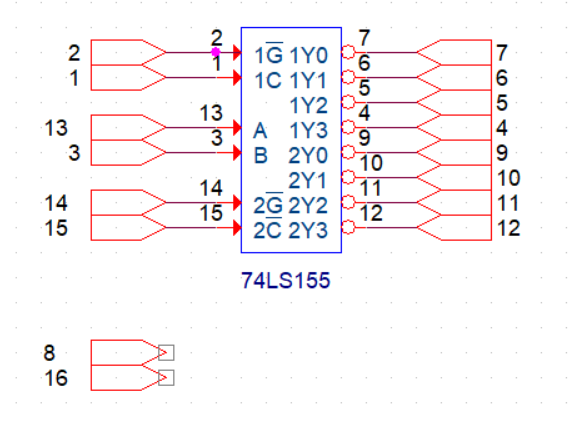


fig.8.2.3

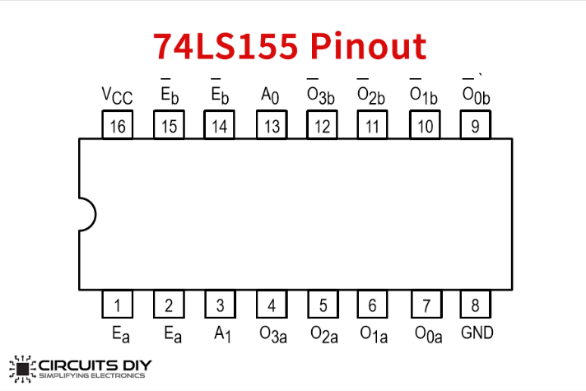


fig.8.2.4

* 1. Implementarea funcției cu DMUX de 16 căi utilizând porți ȘI-NU (CI TTL)

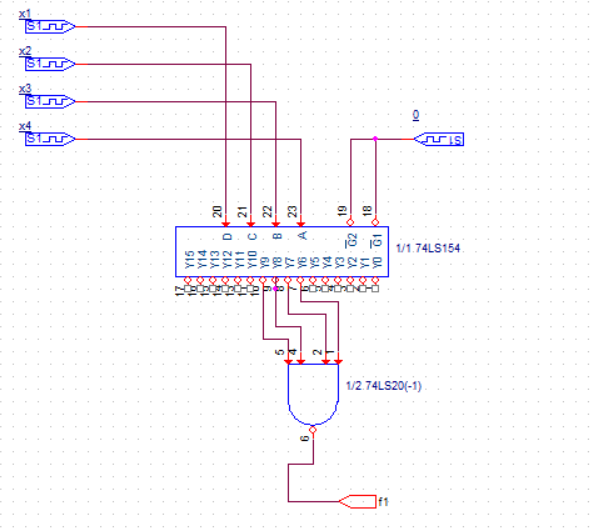


fig.8.3.1

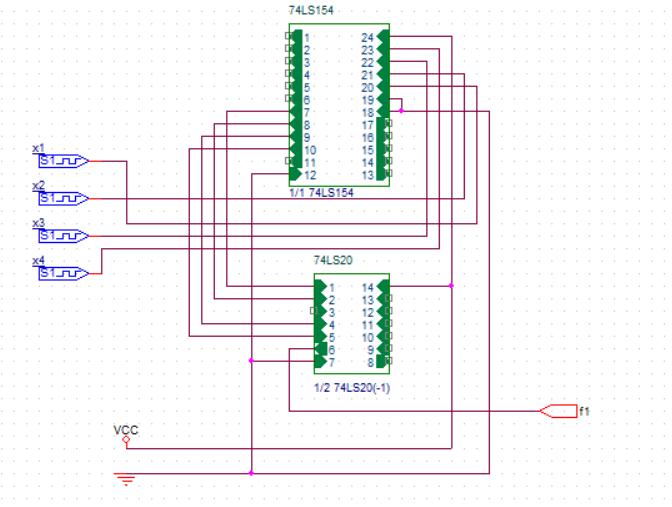


fig.8.3.2

Pentru implementarea cu circuite integrate s-a folosit CI74LS20 și CI74LS154, după cum urmează în figurile următoare.

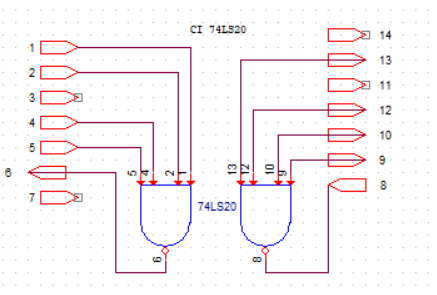


fig.8.3.3

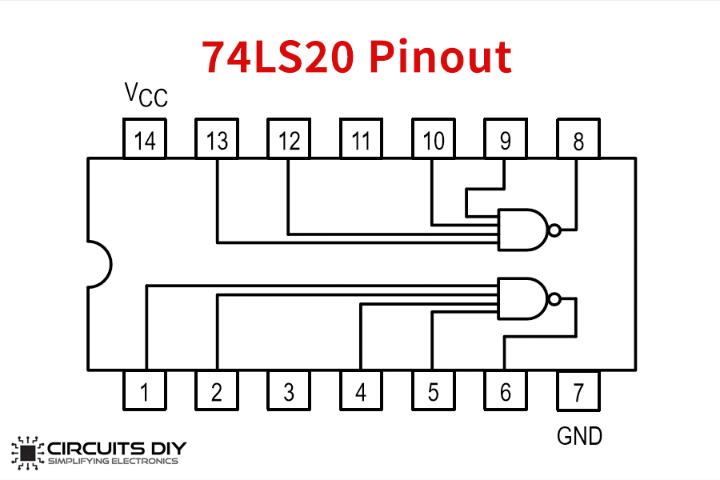


fig.8.3.4

* 1. Implementarea funcției cu DMUX de 8 căi legate în paralel cu porți ȘI-NU (CI TTL)

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 aşa cum este ea si pentru al doilea negată.

Unde 2Y0= P0, 2Y1= P1, 2Y2= P2, 2Y3= P3,1Y0= P4,1Y1= P5, 1Y2= P6, 1Y3 =P7 pentru primul DMUX, iar pentru al doilea 2Y0= P8, 2Y1= P9, 2Y2= P10, 2Y3= P11,1Y0= P12,1Y1= P13, 1Y2= P14, 1Y3 =P15

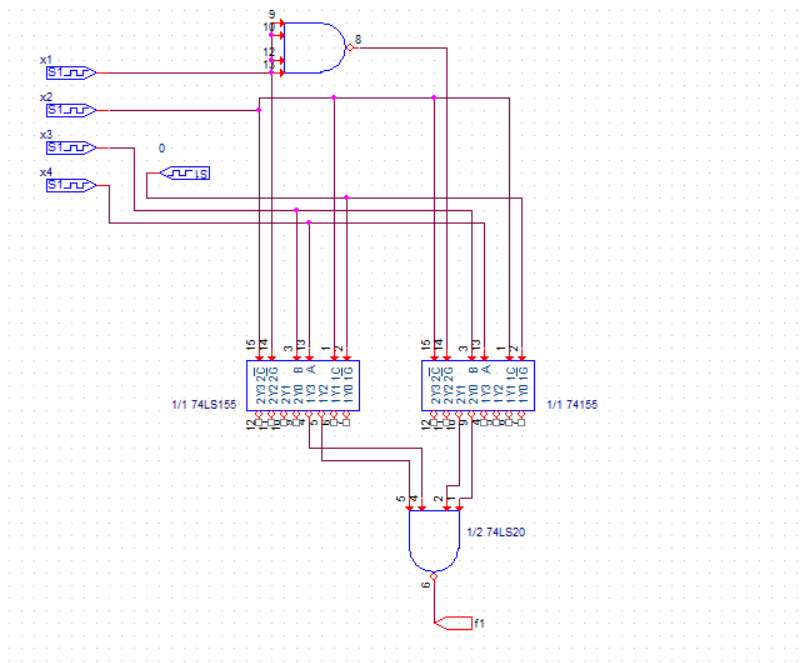


fig.8.4.1

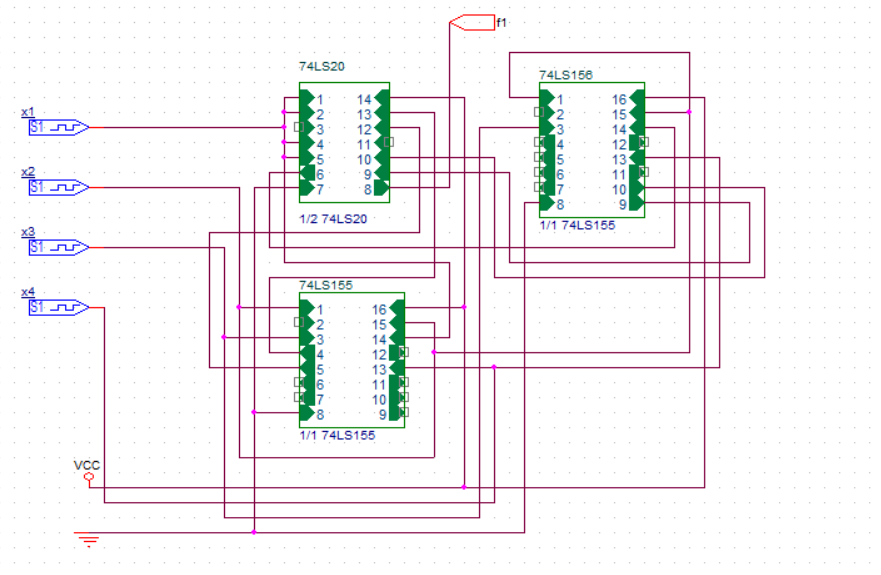


fig.8.4.2

* 1. Implementarea funcției cu DMUX de 16 căi utilizând porți ȘI (CI CMOS)

iar sunt combinații indiferente.

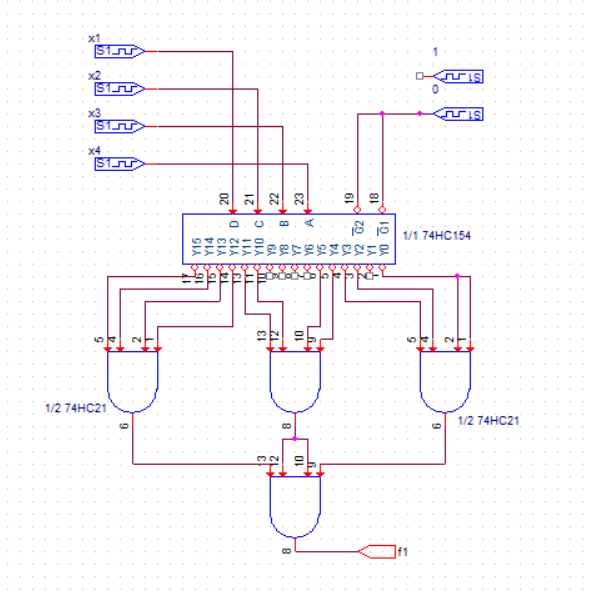


fig.8.5.1

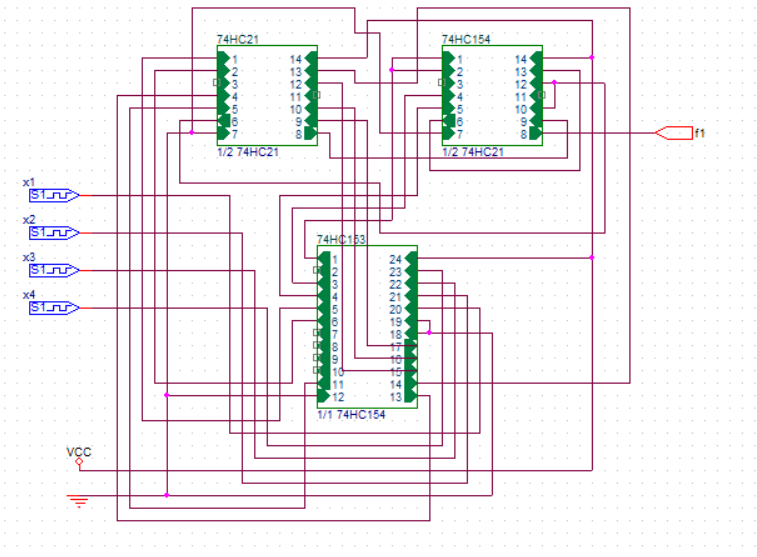


fig.8.5.2

Pentru implementarea cu circuite integrate s-a utilizat 74HC21 și 74HC154, care au intrările și ieșirile porților logice legate la aceeași pini ca la variantele TTL. Sunt illustrate în următoarele figuri.

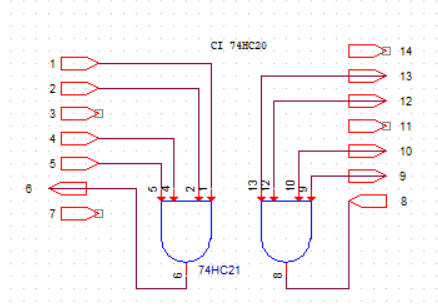


fig.8.5.3

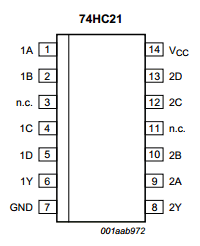


fig.8.5.4

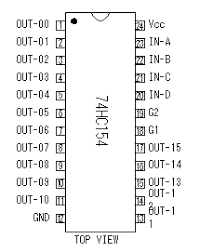
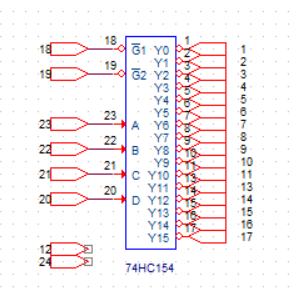


fig.8.5.6

fig.8.5.5

1. Calcularea timpilor de propagare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C.I. | Tehnologie | tPLH[ns] | tPHL[ns] | tp = ½\*( tPLH+ tPHL) [ns] |
| 74LS00 | TTL | 10 | 10 | 10 |
| 74LS04 | 15 | 15 | 15 |
| 74LS10 | 15 | 15 | 15 |
| 74LS20 | 15 | 15 | 15 |
| 74LS32 | 22 | 22 | 22 |
| 74LS150 | 35 | 33 | 34 |
| 74LS151 | 43 | 30 | 36.5 |
| 74LS154 | 30 | 30 | 30 |
| 74LS155 | 15 | 30 | 22.5 |
| 74HC21 | CMOS | 10 | 12 | 11 |
| 74HC27 | 8 | 10 | 9 |
| 74HC154 | 11 | 13 | 12 |

Parametri sunt măsurați la temperatura de 25o C, tensiunea de alimentare Vcc=5[V] și CL=15[pF].

ȘI-NU (f1): tp = 10\*3/4 = 7.5[ns]

ȘI-NU (f2): tp = 10+15 = 25[ns]

ȘI-NU (f3): tp = 10+15+15\*1/3 = 25+5 = 30[ns]

ȘI-NU (f4): tp = 10+15+15\*1/3 = 30[ns]

ȘI-NU (f5): tp = 10+15+15\*1/3 =30[ns]

ȘI-NU (f1, f2, f3, f4, f5) Ansamblu: tp = 10+10\*3/4+4\*15+15\*2/3 = 10+10+60+7.5 = 87.5 [ns]

ȘI-NU (f1, f2) și SAU-NU (f3, f4, f5) Ansamblu: tp = 15+9\*3+9\*1/3+10+10\*2/3 = 15+27+3+10+6.6 = 61.6 [ns]

MUX de 8 căi cu variabilă aplicată pe intrările de date (f1): tp = 36.5[ns]

MUX de 16 căi (f1): tp = 34[ns]

MUX de 8 căi în paralel (f1): tp = 2\*36.5+15\*1/6+22\*1/4 = 70+2.5+5.5 = 78[ns]

DMUX de 8 căi și rețea de porți logice (f1): tp = 22.5+10\*3/4+15\*2/3 = 22.5+7.5+10 = 40[ns]

DMUX de 16 căi și porți ȘI-NU (f1): tp = 30+15\*1/2 = 30+7.5 = 37.5[ns]

DMUX de 8 căi în paralel (f1): tp = 2\*22.5+15 = 60[ns]

DMUX (CMOS) de 16 căi și porți ȘI (f1): tp = 12+2\*11 = 12+22 = 34[ns]

1. Calcularea puterilor disipate

TTL:

|  |  |  |  |
| --- | --- | --- | --- |
| C.I. | ICCH[mA] | ICCL[mA] | PD[mW] |
| 74LS00 | 1.6 | 4.4 | 8 |
| 74LS04 | 2.4 | 6.6 | 9.5 |
| 74LS10 | 1.2 | 3.3 | 7.85 |
| 74LS20 | 0.8 | 2.2 | 6.5 |
| 74LS32 | 6.2 | 9.8 | 13 |
| 74LS150 | 40 | 40 | 45 |
| 74LS151 | 10 | 10 | 15 |
| 74LS154 | 14 | 14 | 19 |
| 74LS155 | 10 | 10 | 15 |

PD = VCC+(ICCL+ICCH)/2, unde VCC = 5[V]

ȘI-NU (f1): PD = 9.5\*3/4 = 7.1[mW]

ȘI-NU (f2): PD = 8+7.85 = 15.9[mW]

ȘI-NU (f3): PD = 8+7.85+7.85\*1/3 = 18.5[mW]

ȘI-NU (f4): PD = 8+7.85+7.85\*1/3 = 18.5[mW]

ȘI-NU (f5): PD = 8+7.85+7.85\*1/3 = 18.5[mW]

ȘI-NU (f1, f2, f3, f4, f5) Ansamblu: PD = 8+8\*3/4+4\*7.85+7.85\*2/3 = 8+6+31.4+5.2 = 50.6[mW]

MUX de 8 căi cu variabilă aplicată pe intrările de date (f1): PD = 15[mW]

MUX de 16 căi (f1): PD = 45[mW]

MUX de 8 căi în paralel (f1): PD = 2\*15+9.5\*1/6+13\*1/4 = 30+1.6+3.2 = 33.8[mW]

DMUX de 8 căi și rețea de porți logice (f1): PD = 15+8\*1/4+7.85\*1/3 = 15+2+2.6 = 19.6 [mW]

DMUX de 16 căi și porți ȘI-NU (f1): PD = 19+6.5\*1/2 = 19+3.25 = 22.3[mW]

DMUX de 8 căi în paralel (f1): PD = 2\*15+6.5 = 30+6.5 = 36.5 [mW]

CMOS:

|  |  |  |  |
| --- | --- | --- | --- |
| C.I | CP[pF] | CL[pF] | PD[mW] |
| 74HC21 | 25 | 50 | 0.151 |
| 74HC27 | 25 | 50 | 0.151 |
| 74HC154 | 88 | 50 | 0.263 |

PD = VCC2 \*fi\*(CPD+CPL)

74HC21: PD = 52∗105∗10-12∗(25+50) = 0.151[𝑚𝑊]

74HC27: PD =52∗105∗10-12∗(25+50) = 0.151[𝑚𝑊]

74HC154: PD = 52∗105∗10-12∗(88+50) = 0.263[𝑚𝑊]

ȘI-NU (f1, f2) și SAU-NU (f3, f4, f5) Ansamblu: PD = 7.85+8\*3+8\*1/3+0.151\*4+0.151\*2/3 = 35.2[mW]

DMUX (CMOS) de 16 căi și porți ȘI (f1): PD = 0.263+2\*0.151 = 0.263+0.302 = 0.565[mW]

1. Compararea soluțiilor de implementare

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. crt. | Implementare | Nr. CI | Nr. porți neutilizate | tP[n] | PD[mW] |
| 1. | f1 cu ȘI-NU | 1 | 1 | 7.5 | 7.1 |
| 2. | f2 cu ȘI-NU | 2 | - | 25 | 15.9 |
| 3. | f3 cu ȘI-NU | 3 | 2 | 30 | 18.5 |
| 4. | f4 cu ȘI-NU | 3 | 2 | 30 | 18.5 |
| 5. | f5 cu ȘI-NU | 3 | 2 | 30 | 18.5 |
| 6. | Ansamblu cu ȘI-NU | 7 | 3 | 87.5 | 50.6 |
| 7. | Ansamblu cu ȘI-NU și SAU-NU | 7 | 2 | 61.6 | 32.5 |
| 8. | f1 cu MUX de 8 și var. aplicate pe int. de date | 1 | - | 36.5 | 15 |
| 9. | f1 cu MUX de 16 | 1 | - | 34 | 45 |
| 10. | f1 cu MUX-uri de 8 în paralel | 4 | 8 | 78 | 33.8 |
| 11. | f1 cu DMUX de 8 și rețea de porți logice | 3 | 2 | 40 | 19.6 |
| 12. | f1 cu DMUX de 16 | 2 | 1 | 37.5 | 22.3 |
| 13. | f1 cu DMUX de 8 în paralel | 3 | - | 60 | 36.5 |
| 14. | f1 cu DMUX de 16(CMOS) | 3 | - | 34 | 0.565 |

1. Analiza prin simulare, a tuturor schemelor logice obținute utilizându-se pachetul de programe OrCAD

Funcția f1 cu porți ȘI-NU

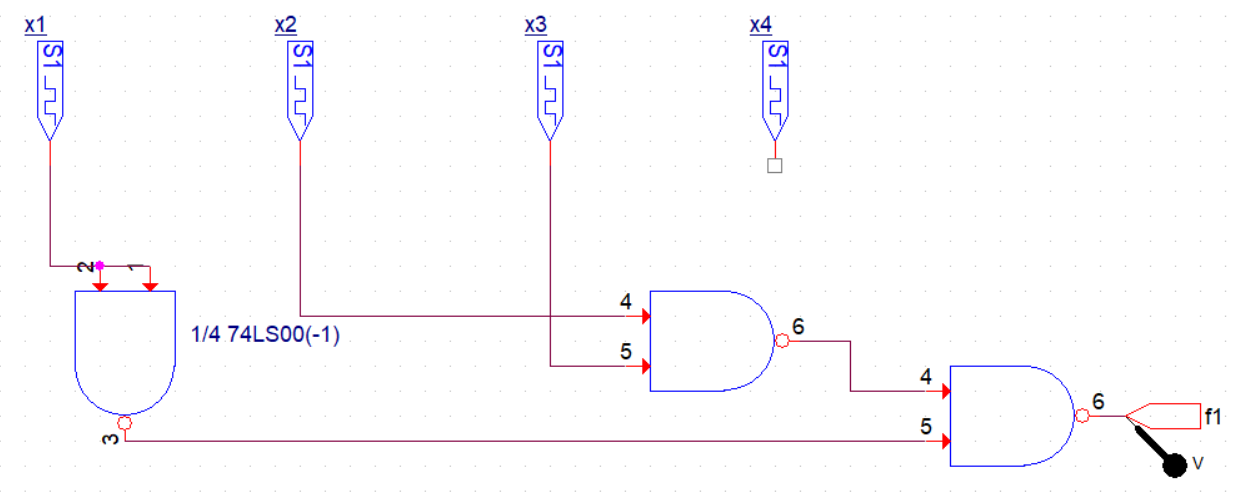


fig.12.1

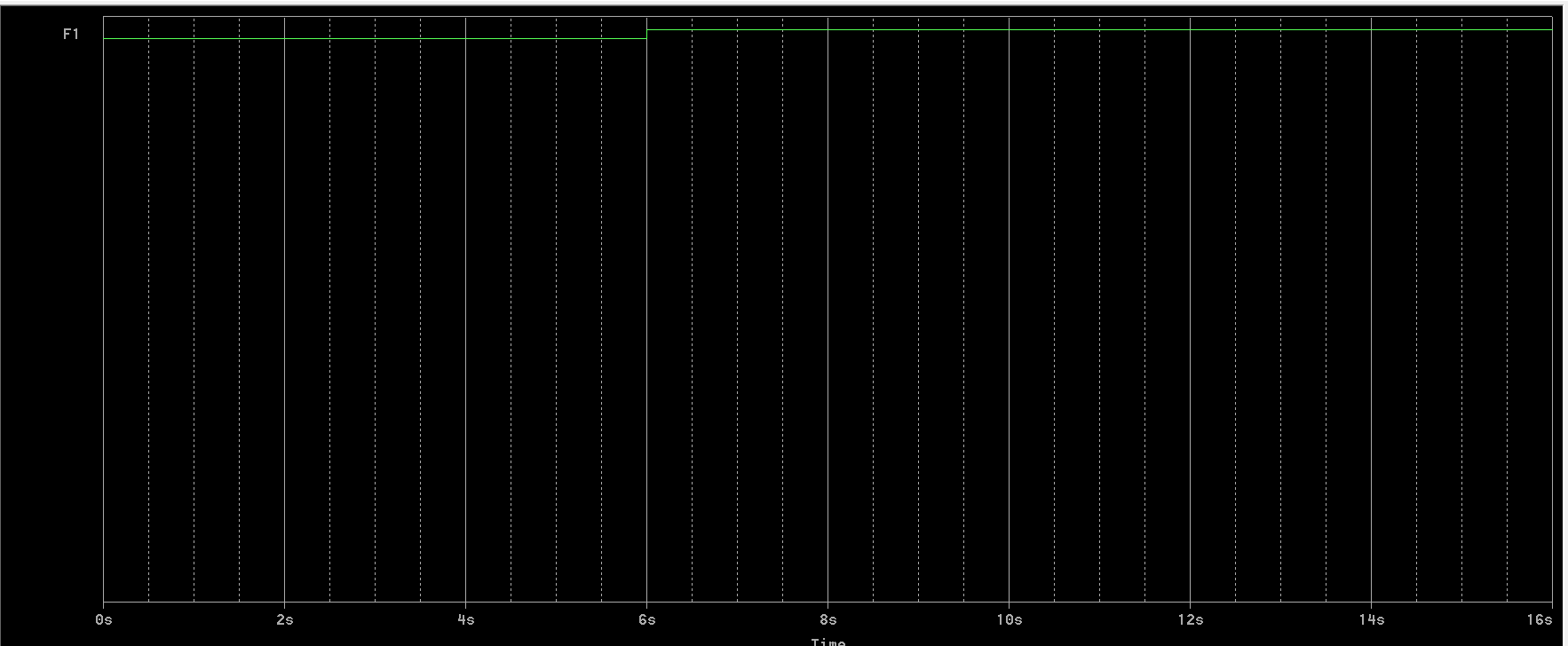


fig.12.2

Ansamblul funcțiilor

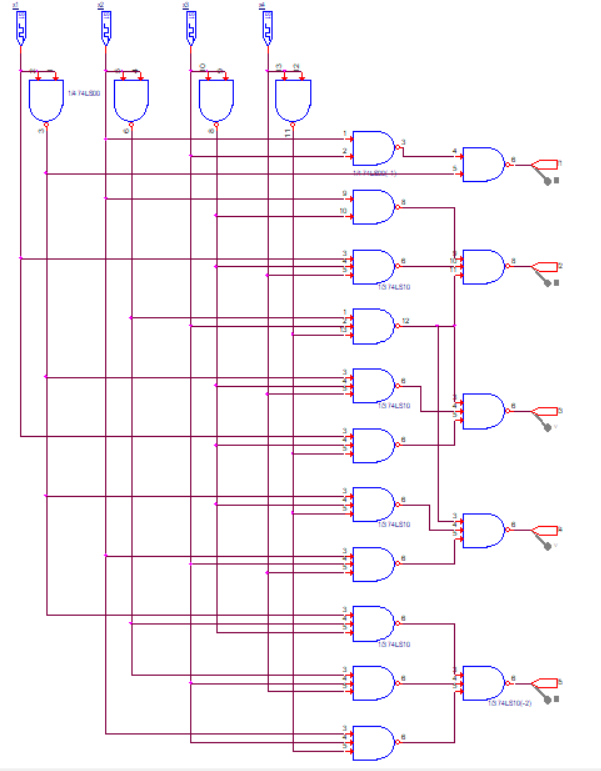


fig.12.3

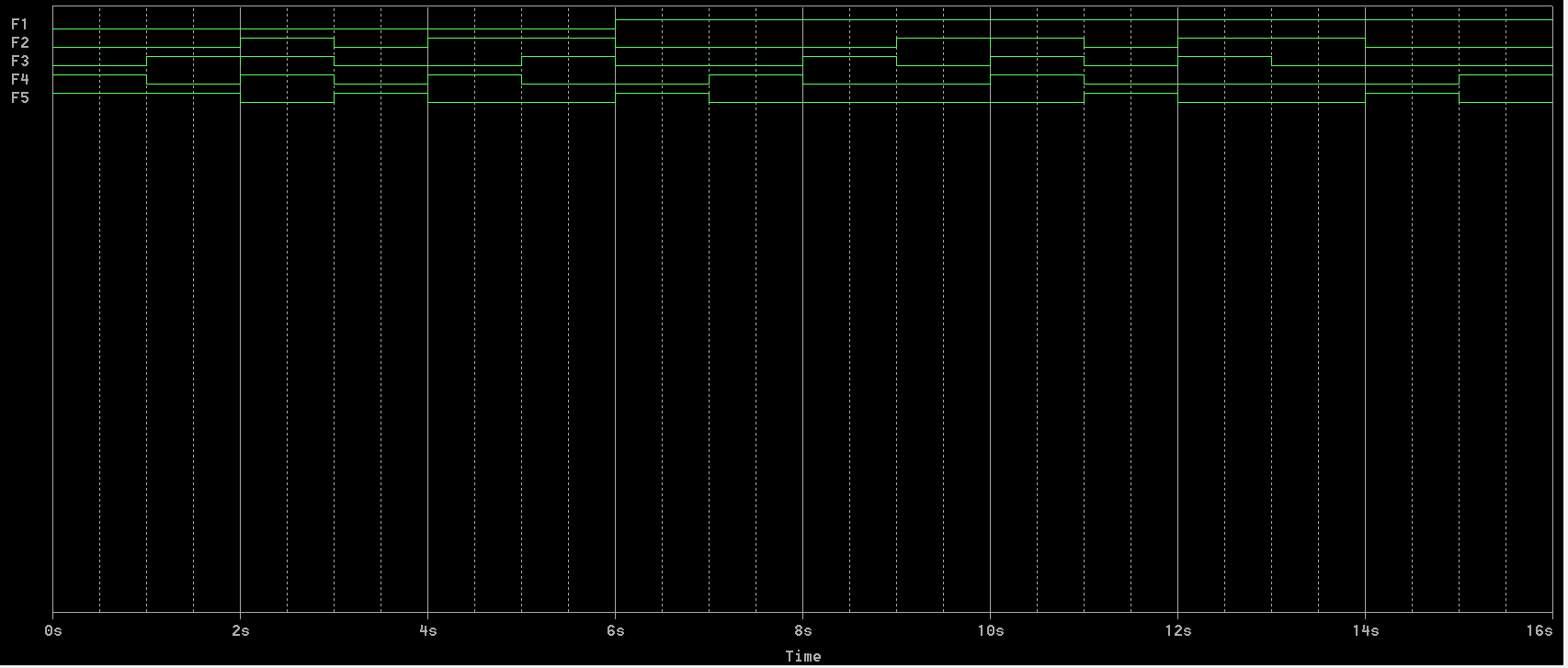


fig.12.4

Funcția f1 cu MUX de 8 căi

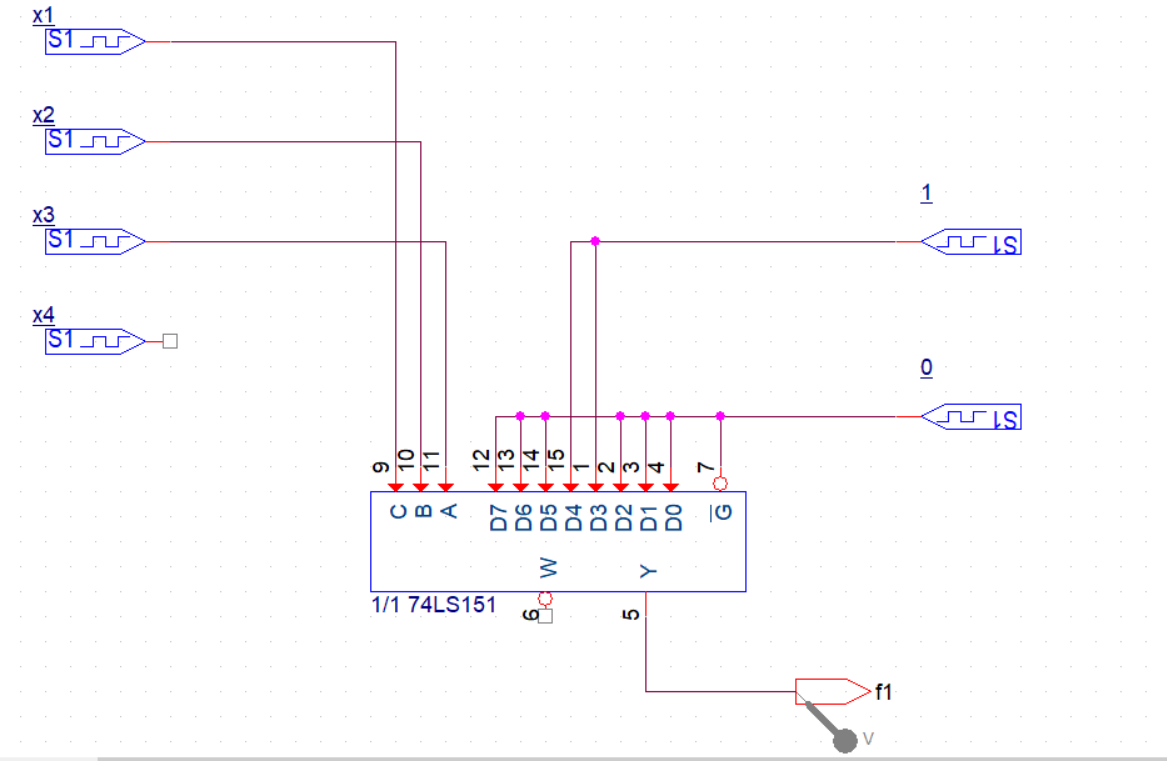


fig.12.5

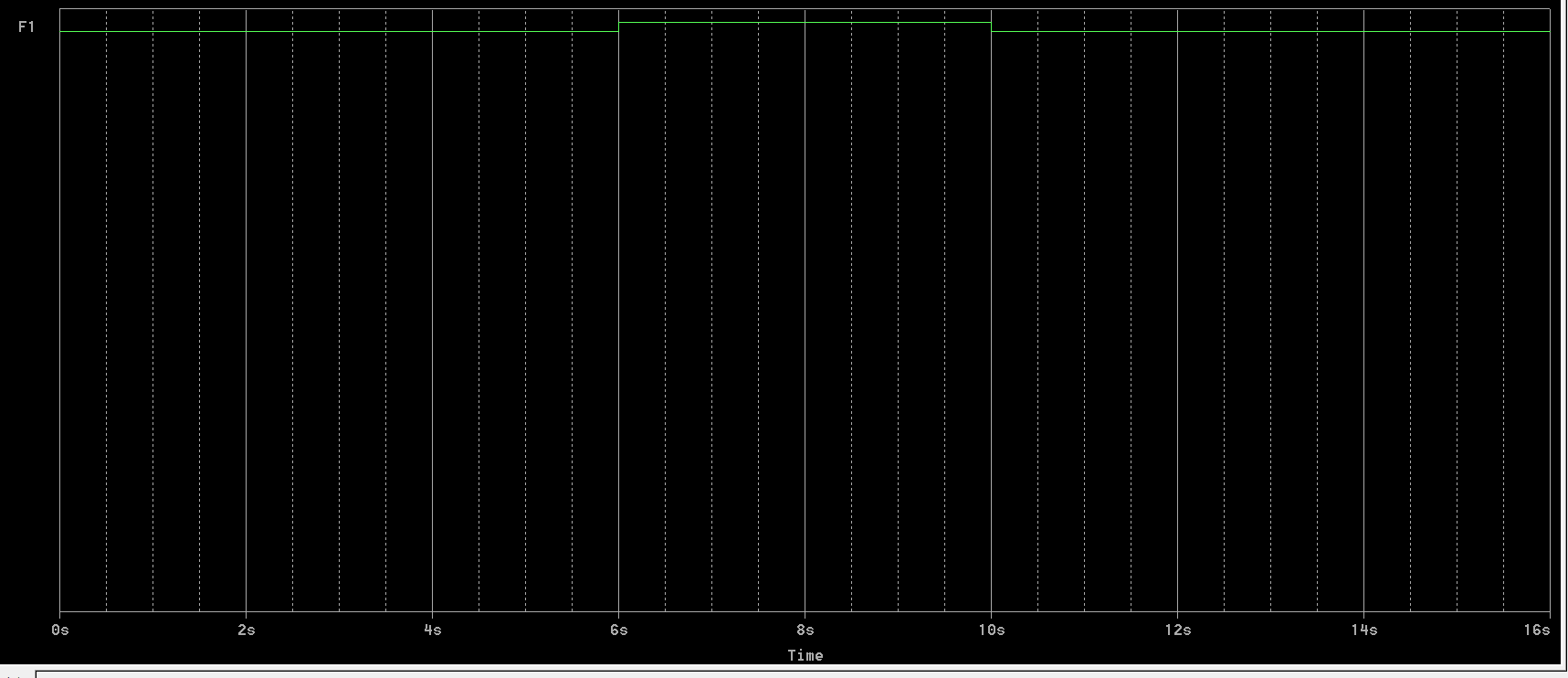


fig.12.6

Funcția f1 cu MUX-uri de 8 căi puse în paralel

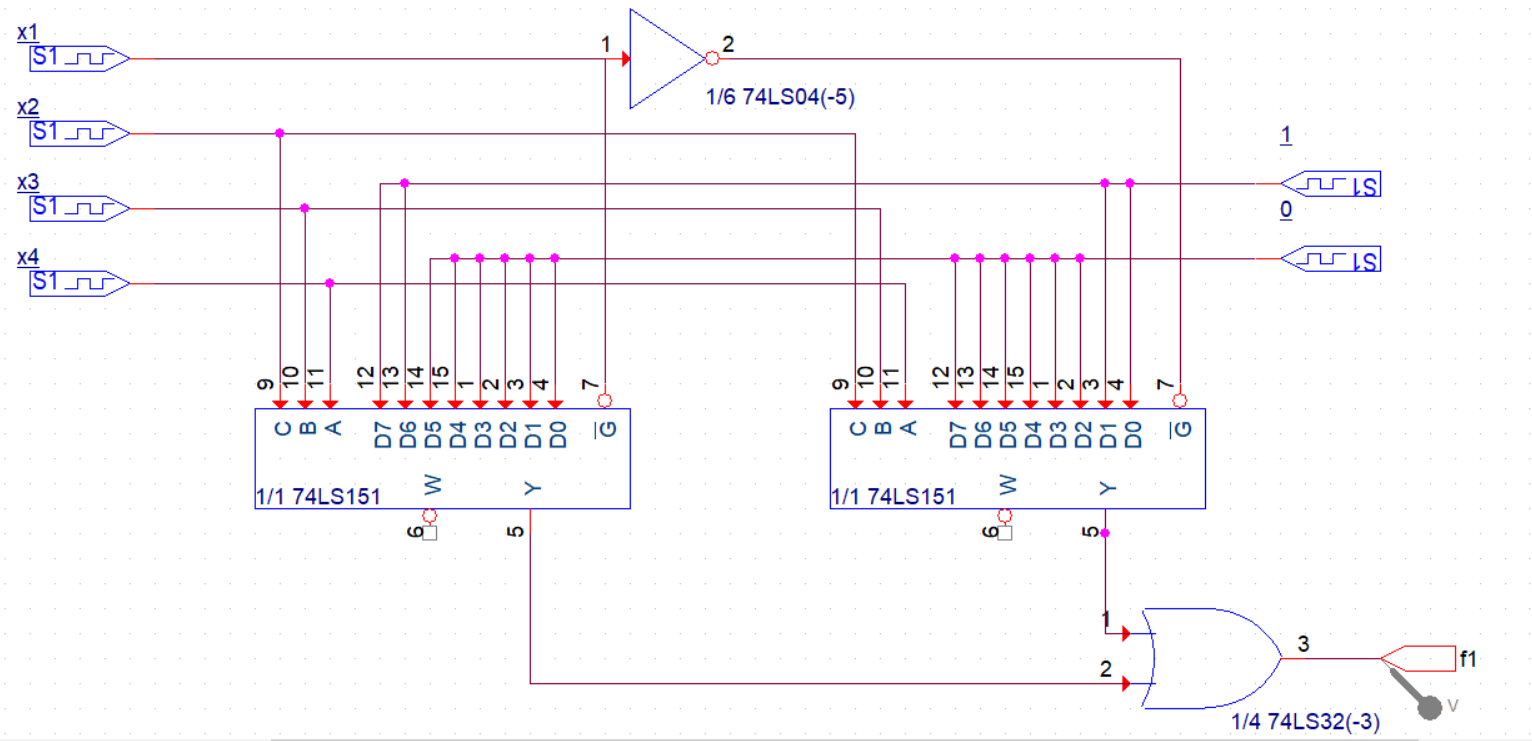


fig.12.7

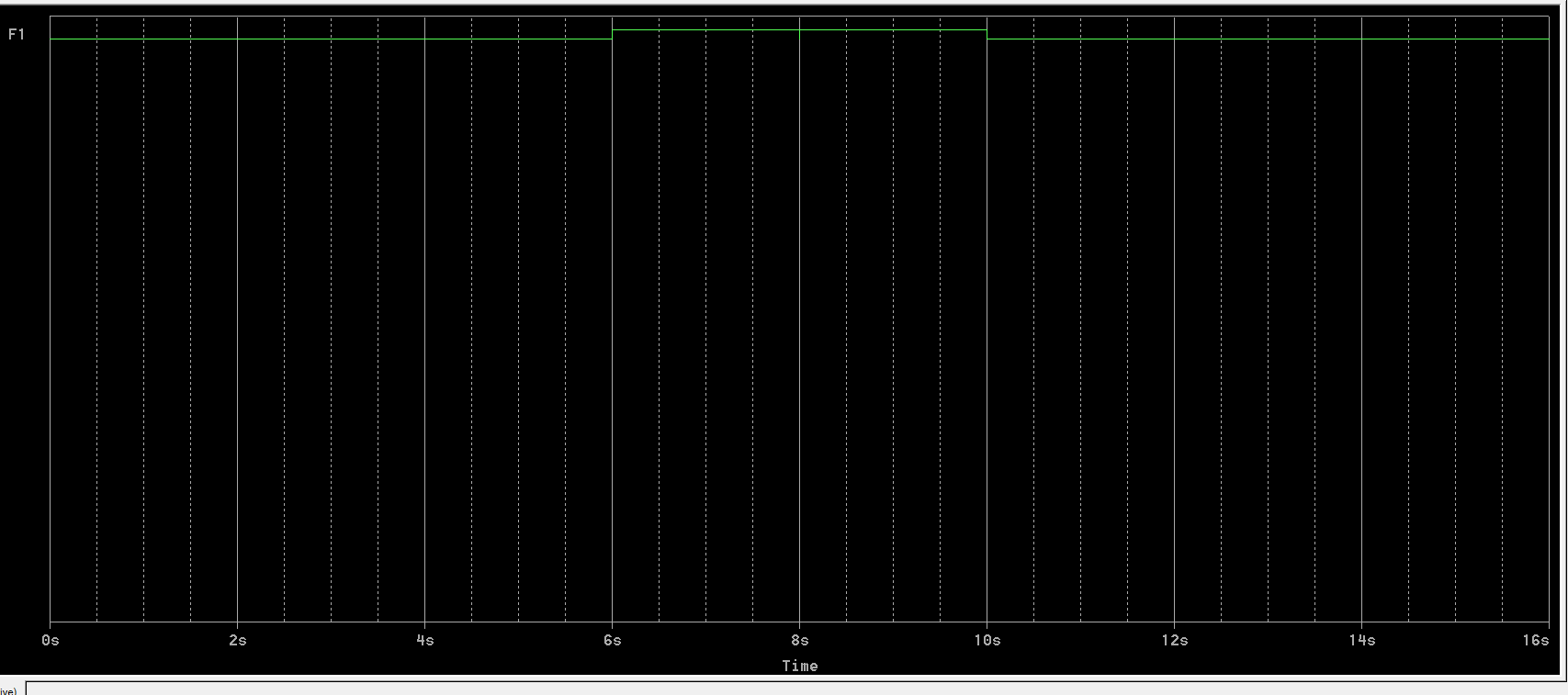


fig.12.8

Funcția f1 cu DEMUX de 8 căi

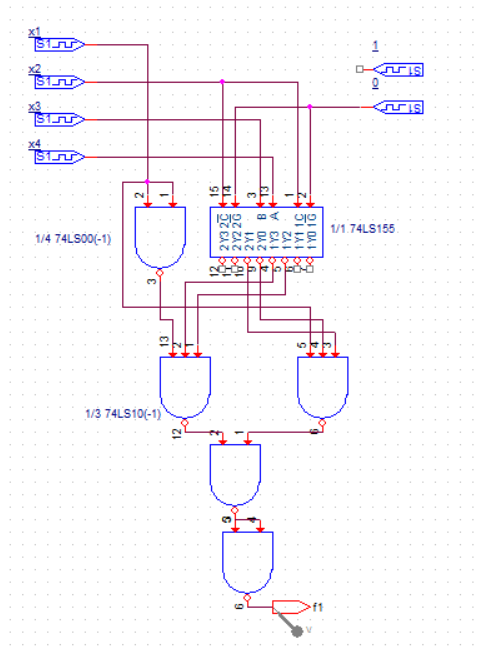


fig.12.9

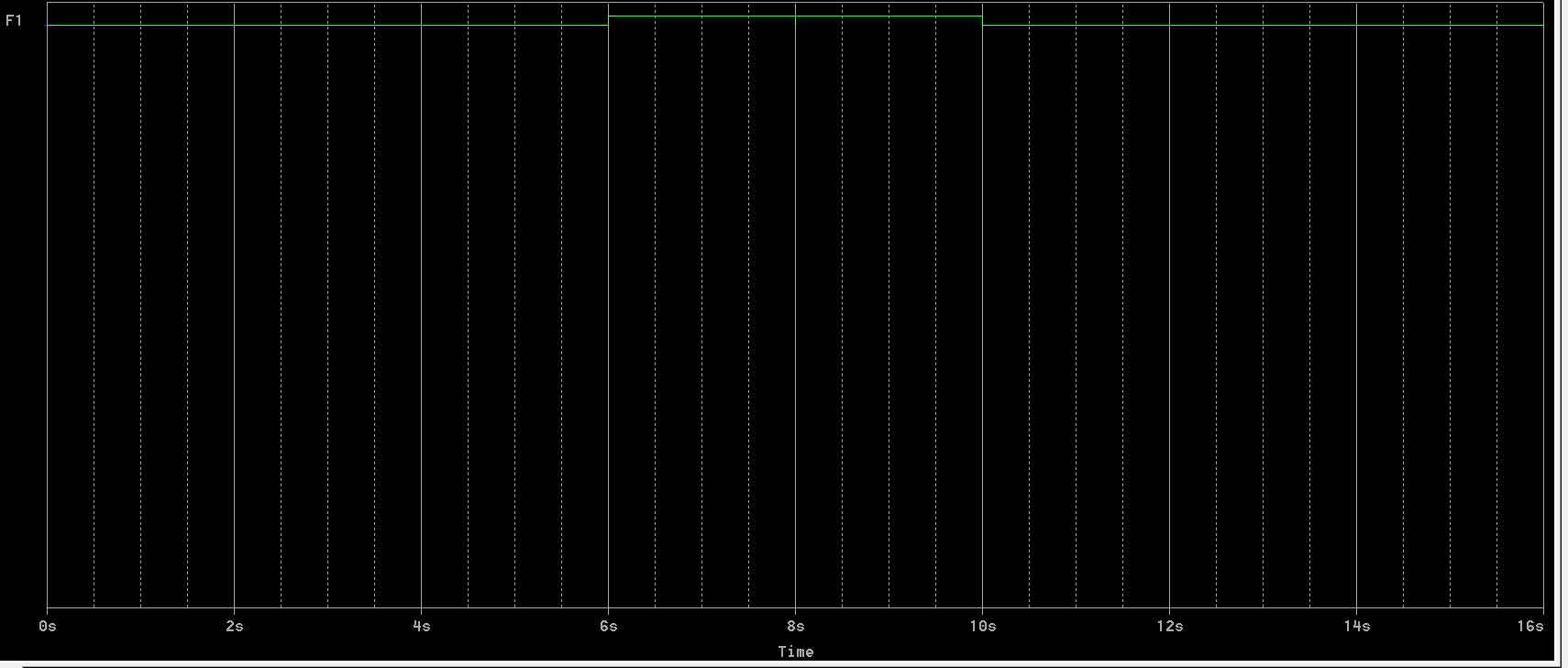


fig.12.10

Funcția f1 cu DEMUX de 16 căi

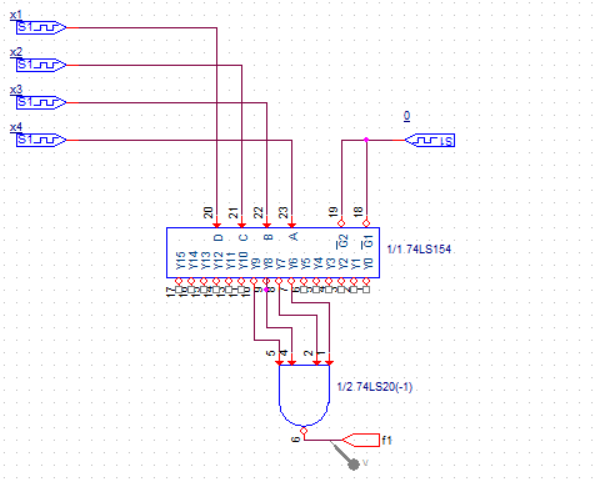


fig.12.11

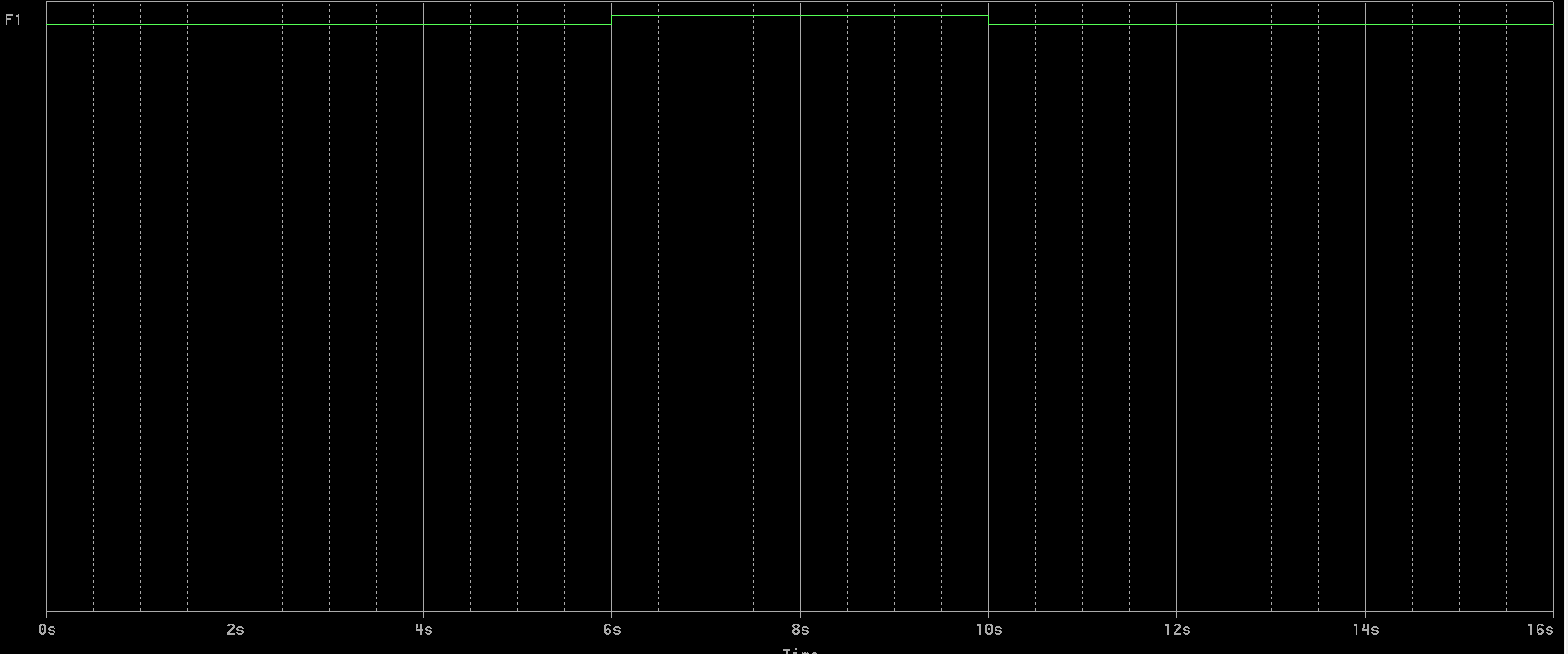


fig.12.12

1. Bibliografie

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

Datasheet-uri:

* <https://www.alldatasheet.com/datasheet-pdf/pdf/15541/PHILIPS/74HC154.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/125043/ETC1/74S00.html>
* <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/74S04.PDF>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/15577/PHILIPS/74HC27.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/15563/PHILIPS/74HC21.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/5666/MOTOROLA/74LS155.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/51038/FAIRCHILD/74LS154.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/5661/MOTOROLA/74LS151.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/5707/MOTOROLA/74LS32.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/5681/MOTOROLA/74LS20.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/12605/ONSEMI/74LS10.html>