UNIVERSITATEA “*TRANSILVANIA*” DIN BRAŞOV

Facultatea de *Inginerie Electrică şi Ştiinţa Calculatoarelor*

Catedra de *Automatică Şi Informatică Aplicată*

###### **Proiect ASCN**

*(temă proiect nr.* ***81****)*

#### Îndrumător: Student:

***Prof. dr. ing. MOLDOVEANU Florin*** ***Drăgan Călin***

Grupa: ***4421***

2014

**Cuprins**

Enunţul proiectului ...............................................................................................................................3

Introducere în circuitele logice combinaţionale....................................................................................4

Tabelul de adevăr ................................................................................................................................ .5

Formele canonice disjunctive şi conjunctive ale funcţiilor...................................................................6

Formele minime disjunctive şi conjunctive prin metoda diagramelor Karnaugh................................ 8

Minimizarea funcţiilor *F1* şi *F2* prin metoda Quine-McCluskey ......................................................11

Implementarea funcţiilor logice individuale cu porţi ŞI-NU TTL .....................................................14

Implementarea F1 cu porţi logice ŞI-NU.................................................................................15

Implementarea F2 cu porţi logice ŞI-NU.................................................................................16

Implementarea F3 cu porţi logice ŞI-NU.................................................................................17

Implementarea F4 cu porţi logice ŞI-NU.................................................................................18

Implementarea ansamblului cu porţi SI-NU TTL .......... .................................................................. 19

Implementarea ansamblului cu porţi SAU-NU CMOS si ŞI-NU TTL............................................. 20

Implementare cu MUX-uri în tehnologie CMOS ............................................................................. 21

Implementarea funcţiei F1 cu 2xMUX de 8 căi conectate în paralel.......................................21

Implementarea funcţiei F2 cu MUX de 8 căi şi o variabilă aplicată pe intrările de date.........23

Implementarea funcţiei F3 cu MUX de 16 căi.........................................................................24

Implementare cu DMUX-uri în tehnologie TTL ............................................................................... 25

Implementarea lui F1 cu DMUX-uri de 8 căi conectate în paralel...........................................26

Implementarea funcţiei F2 cu DMUX 8 căi şi reţea de porţi logice.........................................28

Implementarea funcţiei logice F3 cu DMUX de 16 căi şi porti ŞI..........................................29

Date de catalog ale circuitelor integrate folosite la implementări ..................................................... 31

Calculul timpilor de propagare .......................................................................................................... 32

Calcului puterilor disipate ................................................................................................................. 33

Concluzii finale ................................................................................................................................. 34

Bibliografie......................................................................................................................................... 35

**Enunţul proiectului nr. 81**

Să se proiecteze un convertor de cod de 4 biţi pentru conversia codului binar zecimal ponderat 5421 în codul binar zecimal neponderat Gray (logică combinaţională).Proiectul va cuprinde următoarele puncte:

a) Să se exprime funcţiile logice asociate circuitului combinaţional cu FCD (forma canonică disjunctivă), FCC (forma canonică conjunctivă) tabel de adevăr şi di-agrame Karnaugh.

b) Să se obţină formele minime disjunctive şi conjunctive pentru funcţiile logice asociate convertorului de cod, (utilizând combinaţiile indiferente) prin metoda diagramelor Karnaugh; de asemenea se vor obţine formele minime disjunctive pentru **primele** două funcţii logice de ieşire şi prin metoda Quine-McCluskey.

c) Să se implementeze fiecare funcţie, independent, numai cu porţi logice ŞI-NU (porţile logice sunt realizate în tehnologia TTL).

d) Să se implementeze ansamblul funcţiilor logice numai cu porţi logice ŞI-NU (porţile logice sunt realizate în tehnologia TTL).

e) Să se implementeze ansamblul funcţiilor logice în următoarea variantă: primele două funcţii logice de ieşire cu porţi logice **SAU-NU**, realizate în tehnologia **CMOS**, iar următoarele două cu porţi logice **ȘI-NU**, realizate în tehnologia **TTL**.

f) Să se implementeze ansamblul funcţiilor logice cu MUX-uri de 8 respectiv 16 căi (circuitele sunt realizate în tehnologia **CMOS**).

g) Să se implementeze ansamblul funcţiilor logice cu DMUX-uri de 8 respectiv 16 căi şi porţi logice ŞI-NU în prima variantă, respectiv ŞI în a doua variantă (toate circuitele sunt realizate în tehnologia **TTL**).

h) Să se calculeze timpii de propagare „intrare-ieşire”, pentru toate schemele logice obţinute.

i) Să se calculeze puterile disipate pentru toate schemele logice obţinute.

j) Să se compare soluţiile de implementare obţinute.

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

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

Introducere

Un circuit logic combinaţional este un circuit de comutare combinaţională ce se caracterizează prin aceea că la un moment dat starea ieşirilor circuitului depinde doar de starea intrărilor sale. Legătura dintre starea ieşirilor şi starea intrărilor este dată şi în acest caz de funcţia de transfer a circuitului.

Suportul fizic utilizat în CLC este destul de variat. Studiul CLC şi al celor secvenţiale se face folosind un model al acestora numit reţea de comutare sau schemă logică. În reţeaua de comutare sau schema logică se face abstracţie de caracteristicile constructive ale elementelor ce concep reţeaua sau schema logică avându-se în vedere doar proprietăţile lor funcţionale. Din acest motiv reţeaua are un înalt grad de generalitate, permiţând studiul unor clase largi de circuite logice.

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

Reţea de comutare

X1

X2

Xn

Z1

Z2

zm

X1, X2, …,Xn – setul valorilor de intrare

Z1, Z2, …, Zm – setul valorilor de iesire

Relaţii generale între aceste valori:

f(X1, …,Xn) fi – funcţii logice

Z1 = ……………. Xi – variabile logice

Zm = f(X1, …,Xn)

Analiza circuitelor logice combinaţionale

Prin analiza unui CLC se înţelege obţinerea expresiilor mărimilor de ieşire cunoscându-se setul variabilelor de intrare  si configuraţia reţelei. Înţelegem numărul şi 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 reţelelor CLC realizate cu elemente logice de tip inversor (ŞI-NU, SAU-NU): numărul maxim de elemente logice aflate între intrarea şi ieşirea reţelei determinã numărul de nivele logice ale acestuia. Numerotarea lor se face de la ieşire către intrare. Într-o reţea de comutare realizate cu elemente ŞI-NU respectiv SAU-NU s-a observat cã o variabilã de intrare apare negată în expresia ieşirii în cazul în care a parcurs un număr impar de elemente de inversare si necomplementată dacă a parcurs un număr par de astfel de elemente.

**Tabelul de adevăr**

Codul 5421 are caracteristica următoare: cifrele de la 5 la 9 se deosebesc de cifrele de la 0 la 4 numai prin primul bit ( 0 pentru intervalul 0-4 respectiv 1 pentru intervalul 5-9).

Codul Gray se caracterizează prin aceea că trecerea de la o cifră zecimală la următoarea se face prin modificarea unui singur rang binar al tetradei.

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. crt. | Echiv.  zec. | **5 4 2 1** | **G R A Y** |
| **X1 X2 X3 X4** | **F1 F2 F3 F4** |
| 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 | **8** | 1 0 0 0 | 0 1 1 1 |
| 6 | **9** | 1 0 0 1 | 0 1 0 1 |
| 7 | **10** | 1 0 1 0 | 0 1 0 0 |
| 8 | **11** | 1 0 1 1 | 1 1 0 0 |
| 9 | **12** | 1 1 0 0 | 1 1 0 1 |

Se observă că pentru echivalenţii zecimali ai cifrelor 5,6,7,13,14,15 avem combinaţii indiferente.

Pentru reprezentarea unei funcţii booleene se folosesc două forme de bază, numite forme canonice:

* **forma produsului de sume,** denumită şi **forma maxterm** (forma canonică conjunctivă - FCC), unde variabilele sau complementarele lor în cadrul unui maxterm sunt legate prin SAU, iar maxtermii între ei sunt legaţi prin ŞI;
* **forma sumei de produse,** denumită şi **forma** **minterm** (forma canonică disjunctivă - FCD), unde variabilele sau complementerele lor în cadrul unui minterm sunt legate prin ŞI, iar mintermii intre ei sunt legaţi prin SAU.

Formele canonice disjunctive şi conjunctive ale funcţiilor

Pentru a obţine din tabelul de adevăr forma canonică conjunctivă se iau în considerare combinaţiile pentru care funcţia are valoarea 0, iar pentru forma canonică disjunctivă se iau în considerare combinaţiile pentru care funcţia are valoarea 1.





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **0** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **0** |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **1** | **1** |
| 01 | **0** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **1** |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **0** | **1** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **1** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **1** |
| 01 | **1** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |

Formele minime ale funcţiilor:

- cu metoda diagramelor Karnaugh -

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

Diagrama este astfel organizată, încât două compartimente vecine pe linie sau pe coloană, să difere printr-o aceeaşi variabilă, variabilă care într-o combinaţie să apară negată, şi în alta adevărată. (proprietatea de adiacenţă).

Pentru obţinerea formei minime conjunctive, se obţine mai întâi forma minimă conjunctivă a funcţiei negate. Apoi se neagă această formă minimă, şi folosind formulele lui DeMorgan, se obţine forma minimă conjunctivă a funcţiei date.

**Formele minime disjunctive ale funcţiilor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **0** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **0** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **1** | **1** |
| 01 | **0** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **1** |

Tabelul 1 : F1 – FMD



Tabelul 2 : F2 – FMD



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **0** | **1** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **1** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |

Tabelul 3: F3 – FMD



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **1** |
| 01 | **1** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |

Tabelul 4: F4 – FMD



**Formele minime conjunctive ale funcţiilor** se obţin de la forma minimă disjunctivă a funcţiei negate ( luând in considerare 0-urile) apoi aplicânduse formulele lui De Morgan, se obţine forma minimă conjunctivă a funcţiei date.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **0** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **0** |

**Tabelul 1: F1 – FMC**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **0** | **1** |
| 01 | **0** | **🞾** | **🞾** | **0** |
| 11 | **1** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |

**Tabelul 2: F2 – FMC**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **1** | **1** | **1** |
| 01 | **0** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **1** |
| 10 | **0** | **🞾** | **🞾** | **1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1X2  X3  X4 | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **1** |
| 01 | **1** | **🞾** | **🞾** | **1** |
| 11 | **0** | **🞾** | **🞾** | **0** |
| 10 | **1** | **🞾** | **🞾** | **0** |

**Tabelul 3: F3 – FMC**





**Tabelul 4: F4 – FMC**





**Formele minime disjunctive pentru primele două funcţii:**

**- cu metoda Quine-McCluskey -**

Metoda Quine-McCluskey este o metodă algebrică de minimizare a funcţiilor booleene folosită în cazul funcţiilor cu număr mare de variabile, pentru care metodele grafice de minimizare (de exemplu, metoda diagramelor Karnaugh) devin greu de utilizat.

Metoda are două etape:

* + se determină implicanţii primi;
  + se selectează dintre implicanţii primi obţinuţi doar aceia care acoperă toţi termenii canonici ai funcţiei date şi asigură realizarea acesteia la un cost minim.

Termenii canonici se compară în felul următor:

- se compară fiecare termen canonic cu toţi ceilalţi;

- când se găsesc doi termeni care au proprietate de adicenţă, variabila redundantă se elimină, obţinându-se un termen elementar;

- primul ciclu de comparaţii se consideră încheiat în momentul în care s-au comparat între ei toţi termenii canonici, obţinându-se toţi implicanţii primi posibili;

- se compară între ei pe acelaşi criteriu termenii elementari obţinuţi;

- se vor face atâtea cicluri de comparaţie câte sunt necesare, pentru a nu mai exista termeni elementari cu proprietatea de adiacenţă.



1)

Combinaţii indiferente (5, 6, 7, 13, 14, 15) se vor compara cu celelalte, dar nu se vor compara între ele.

Dacă nu se compară o combinaţie, ea ar putea fi conţinută de forma minimă a funcţiei. În primul ciclu de comparaţii s-au comparat toate combinaţiile posibile. Se trece la al doilea ciclu de comparaţii

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 1 | 5\* | 0 1 0 1 |  |
| 6\* | 0 1 1 0 |  |
| 12 | 1 1 0 0 | \* |
| 2 | 7\* | 0 1 1 1 |  |
| 11 | 1 0 1 1 | \* |
| 13\* | 1 1 0 1 | \* |
| 14\* | 1 1 1 0 | \* |
| 3 | 15\* | 1 1 1 1 | \* |

***5\*,6\*,7\*-Necomparate nu intră in tabelul***

***acoperirilor pentru că sunt combinaţii***

***indiferente***

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 1,2 | 12,13 | 1 1 0 - |  |
| 12,14 | 1 1 – 0 |  |
| 2,3 | 11,15 | 1 - 1 1 |  |

***12,13;12,14;11,15– Implicaţi Primi (IP)***

***Tabelul acoperirilor pentru F1:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Echiv. zec. | TC\IP |  |  |  |
| 11 |  |  |  | \* |
| 12 |  | \* | \* |  |

***La intersecţia unei linii cu o coloană se plasează un semn, de exemplu un asterisc, dacă implicantul prim de pe linia respectivă include termenul canonic de pe coloana respectivă***



2) ****

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 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 | \* |
| 14\* | 1 1 1 0 | \* |
| 4 | 15\* | 1 1 1 1 | \* |

***7\*-Necomparata nu intră in tabelul***

***acoperirilor pentru că este combinaţie***

***indiferentă***

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 1,2 | 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 | \* |
| 2,3 | 9,11 | 1 0 – 1 | \* |
| 9,13 | 1 – 0 1 |  |
| 10,11 | 1 0 1 - | \* |
| 10,14 | 1 - 1 0 | \* |
| 12,13 | 1 1 0 - | \* |
| 12,14 | 1 1 – 0 | \* |
| 3,4 | 11,15 | 1 – 1 1 | \* |

***4,5;4,6;4,12– Implicaţi Primi (IP)***

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 1,2,3 | 8,9,10,11 | - 1 0 0 | \* |
| 8,9,12,13 | 1 0 - 0 | \* |
| 2,3,4 | 8,10,12,14 | 1 0 1 - | \* |
| 9,11,13,15 | 1 - 1 0 | \* |
|  | 10,11,14,15 | 1 -1 - | \* |

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa | Indicii termenilor | **X1 X2 X3 X4** | Comp |
| 1,2,3,4 | 8,9,10,11,12,13,14,15 | 1 - - - |  |
|  | | | |

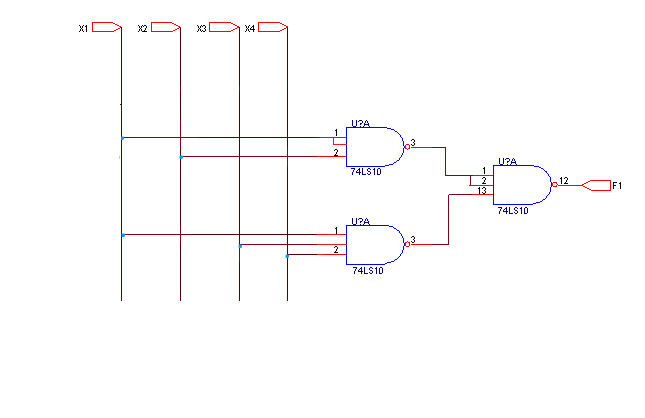
Tabelul de acoperire pentru F2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echiv. zec. | TC\IP |  |  |  |  |
| 4 |  |  | \* | \* | \* |
| 8 |  | \* |  |  |  |
| 9 |  | \* |  |  |  |
| 10 |  | \* |  |  |  |
| 11 |  | \* |  |  |  |
| 12 |  | \* |  |  |  |

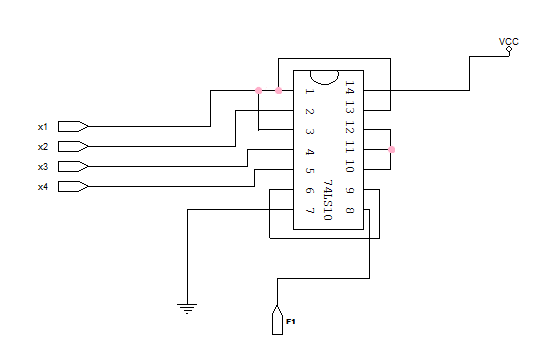


c) Să se implementeze fiecare funcţie, independent, numai cu porţi logice ŞI-NU (porţile logice sunt realizate în tehnologia TTL).

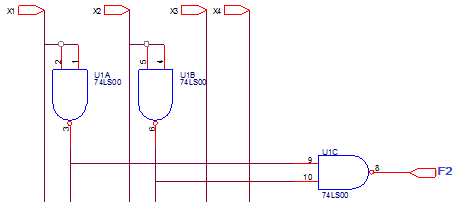




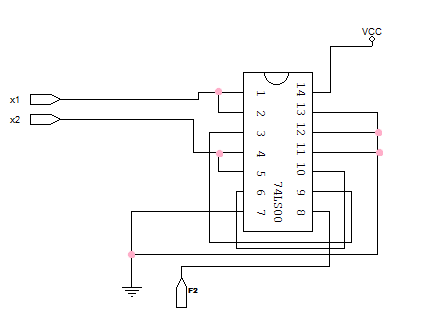
Încapsulare F1 în tehnologia TTL :

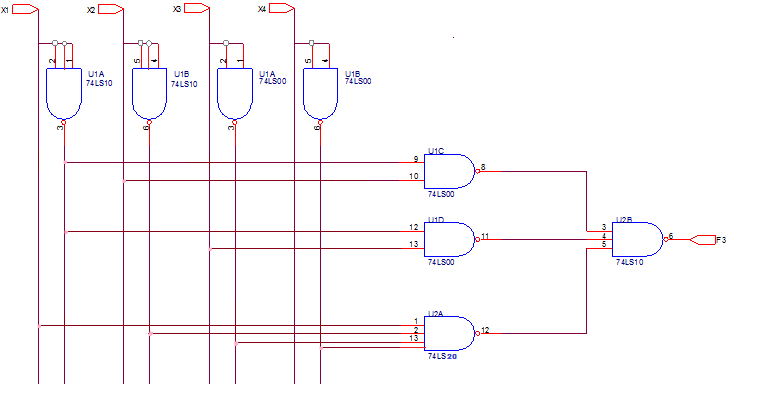




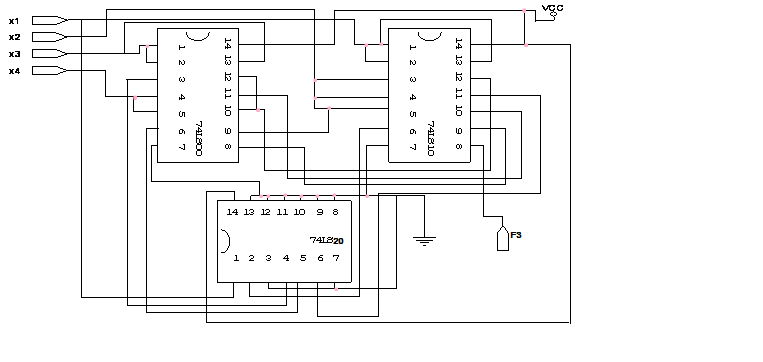


Încapsulare F2 în tehnologia TTL :

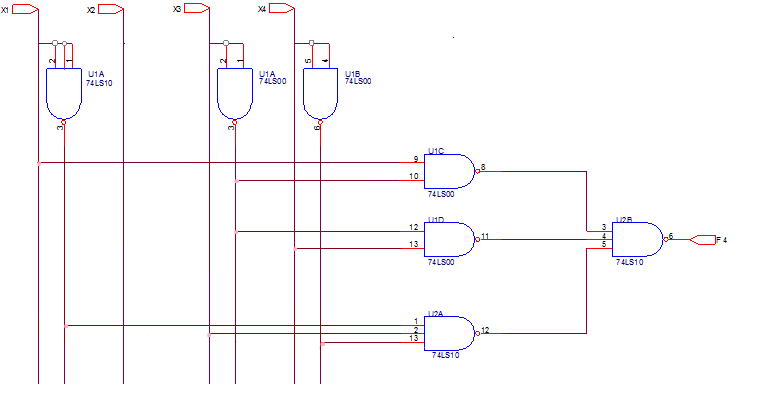




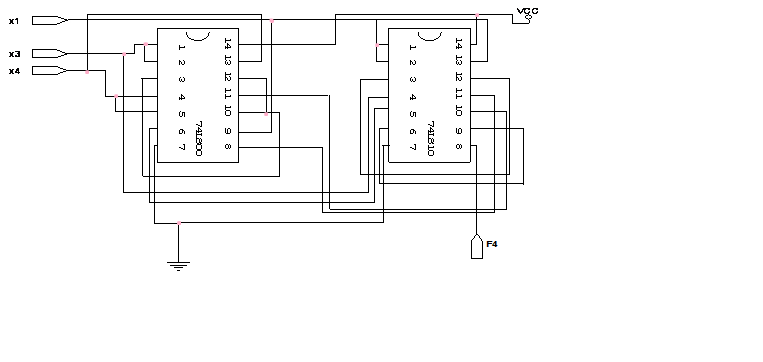
Încapsulare F3 în tehnologia TTL :

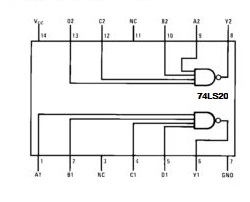




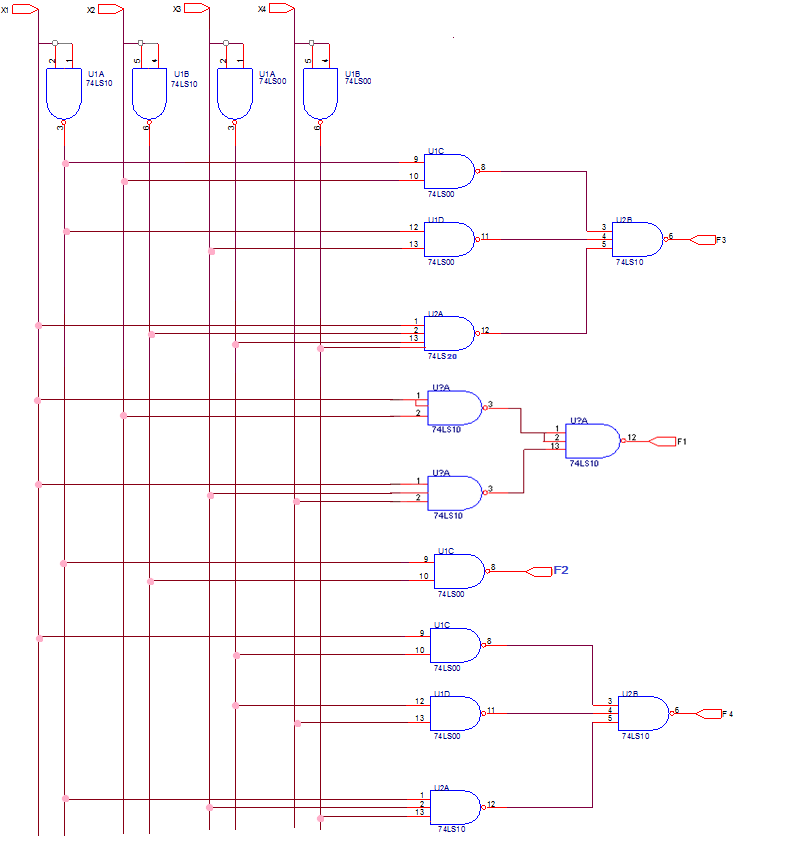


Încapsulare F4 în tehnologia TTL :

Circuite integrate folosite :

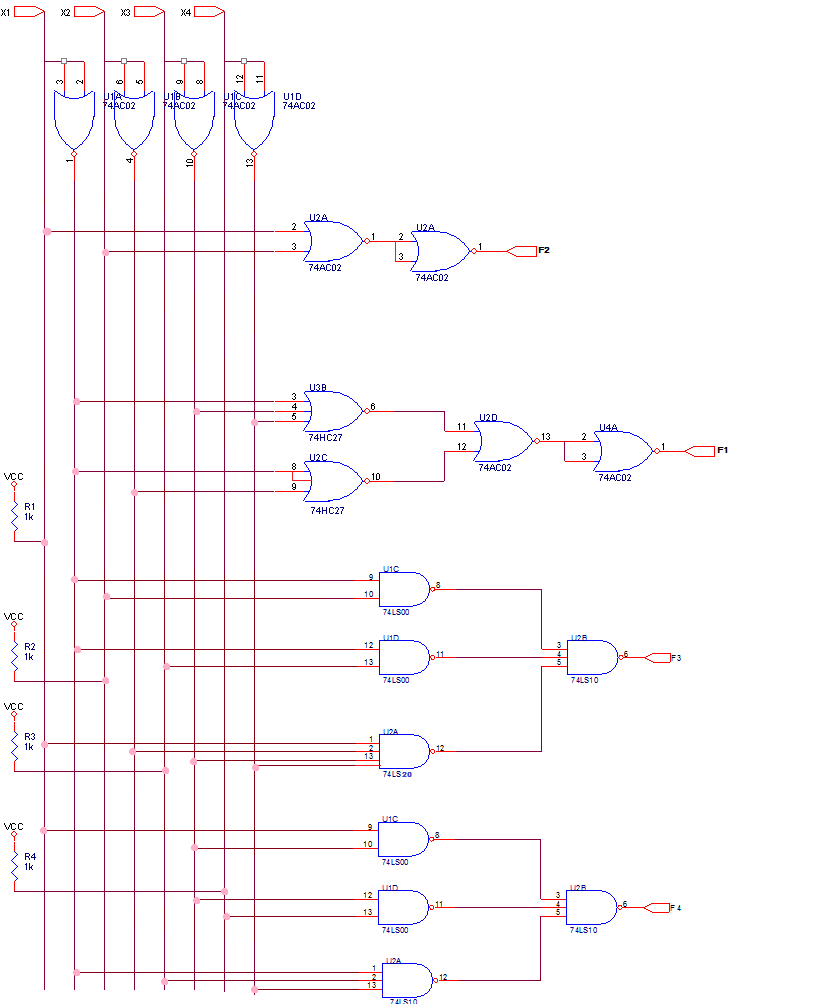


**Implementarea ansamblului folosind porţi ŞI-NU:**

****

S-au folosit 2x74LS00 (-),2x74LS10 (-),1x74LS20 (-).

e) Să se implementeze ansamblul funcţiilor logice în următoarea variantă: primele două funcţii logice de ieşire cu porţi logice **SAU-NU**, realizate în tehnologia **CMOS**, iar următoarele două cu porţi logice **ȘI-NU**, realizate în tehnologia **TTL**.



**Implementarea schemei cu MUX-uri în tehnologie CMOS**

Un circuit de multiplexare este un circuit logic combinaţional care, în cazul general, are 2n intrări de date (I2n-1 … I2 I1 I0), n intrări de selecţie (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 selecţie: k=Sn-1, Sn-2, … , S1, S0.

Circuite integrate folosite :

 74HC151 74HC04 74HC32

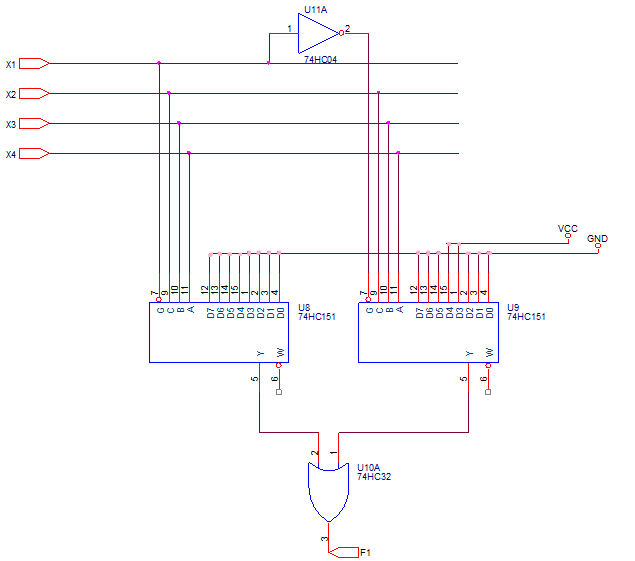


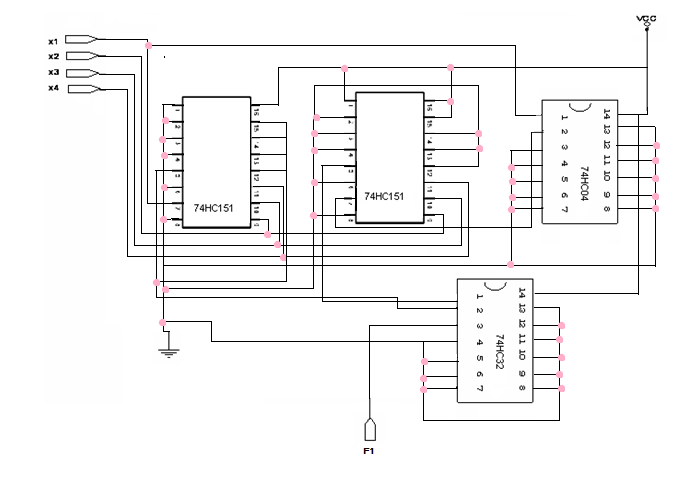
**Implementarea funcţiei F1 cu 2xMUX de 8 căi conectate în paralel**

La implementarea unei funcţii logice cu 2 multiplexoare de 8 căi legate în paralel variabilele funcţiei de implementat 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-seîn ordinea semnificativitaţii lor pe intrările de date. Pe intrările de selecţie se vor aplica semnale corespunzătoare echivalenţilor zecimali prezenţi în expresia funcţiei urmând ca ieşirile celor două multiplexoare să fie legate într-o poartă SAU.

.



****

****S-au folosit 2x74HC151(-),1x74HC04(-5),1x74HC32(-3).

**Implementarea funcţiei F2 cu MUX de 8 căi şi o variabilă aplicată pe intrările de date**

La implementarea unei functii cu un multiplexor de 8 căi şi o variabilă aplicată pe intrările de date se pleaca de la tabelul de adevăr al funcţiei de implementat separându-se variabila cea mai puţin semnificativă. Astfel din cele *n* variabile se separă *n-1* având ponderile cele mai mari si se plică pe intrările de selecţie ale multiplexorului. Astfel se obţin termeni canonici de *n-1* variabile. Pentru a obţine însă termenii canonici de *n* variabile prezenţi în expresia funcţiei 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 | | | | Funcţia |
| **X1** | **X2** | **X3** | **X4** | **F2** |
| **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 | 1 |
| **8** | 1 | 0 | 0 | 0 | 1 |
| **9** | 1 | 0 | 0 | 1 | 1 |
| **10** | 1 | 0 | 1 | 0 | 1 |
| **11** | 1 | 0 | 1 | 1 | 1 |
| **12** | 1 | 1 | 0 | 0 | 1 |

Astfel pentru : X1X2X3=000 F2 =0 oricare ar fi X4 =>D0=0

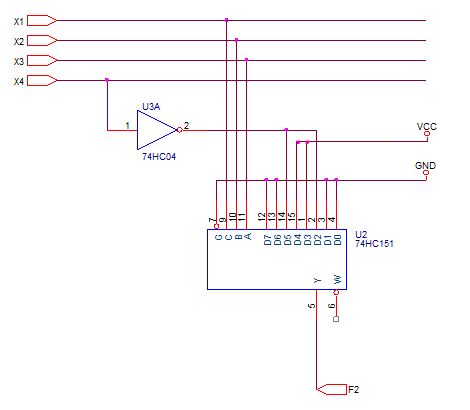
X1X2X3=001 F2 =0 oricare ar fi X4 =>D1=0

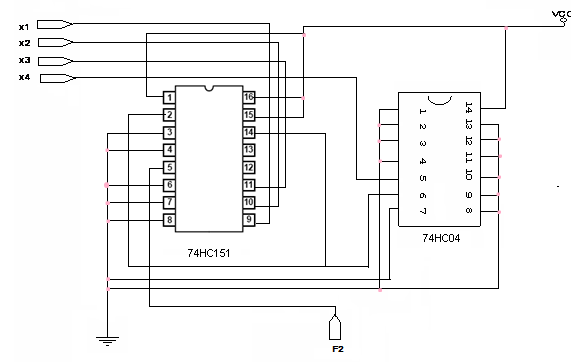
X1X2X3=010 F2 =1 pentru X4 =0 =>D2= 

X1X2X3=100 F2 =1 oricare ar fi X4 =>D3=1

X1X2X3=101 F2 =0 oricare ar fi X4 =>D4=1

X1X2X3=110 F2 =1 pentru X4 =0 =>D5=





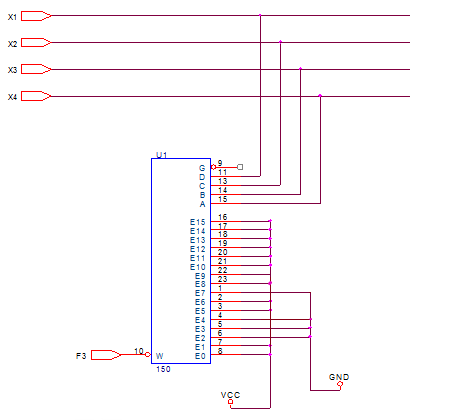
S-au folosit următoarele circuite integrate : 1x74HC151(-),1x74HC04(-5).

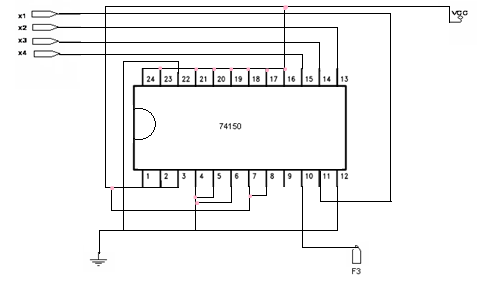
**Implementarea funcţiei F3 cu MUX de 16 căi**

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

****

****





S-au folosit următoarele circuite integrate: 1x74150 (-).

**Implementarea schemei cu DMUX-uri în tehnologie TTL**

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

Pentru implementarea funcţiilor cu DMUX variabilele funcţiei se vor aplica pe intrările de selecţie în raport cu ponderile acestora.

Pentru realizarea nivelului logic SAU se pot folosi porti:

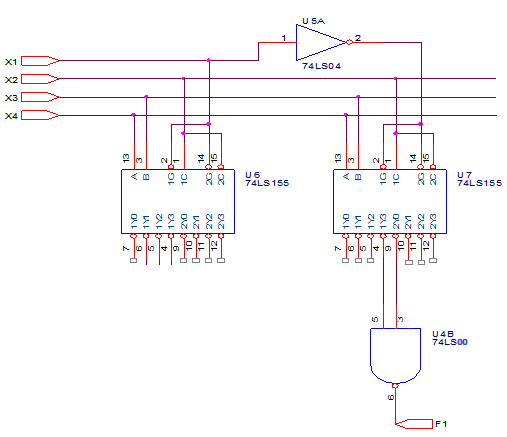
1. ŞI-NU, în acest caz se leagă la intrările porţilor ŞI-NU ieşirile DMUX corespunzătoare termenilor canonici prezentaţi în expresia funcţiei de implementat.
2. ŞI, în acest caz se leagă la intrările porţilor ŞI ieşirile DMUX corespunzătoare termenilor canonici care nu apar în expresia funcţiei (se vor considera funcţiile negate).

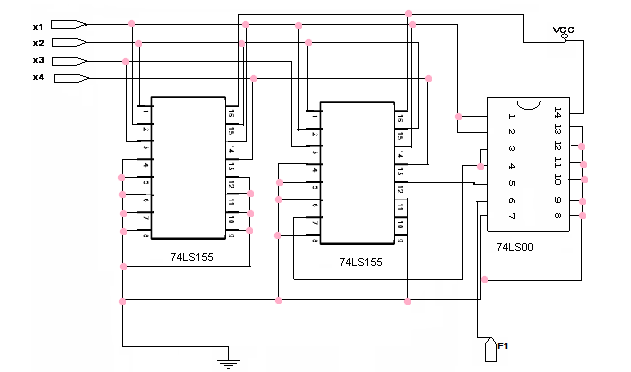
În cazul în care funcţia are, în caz general, n variabile şi se impune să se implementeze un DMUX 1:2 se va separa variabila cu ponderea cea mai mare, iar cele n-1 variabile de stare se vor aplica pe intrarile DMUX în raport cu ponderile lor. Deoarece la ieşirile acestor circuite se obţin termeni canonici de n-1 variabile, iar în forma în care a fost funcţia de implementat sunt termeni canonici de n variabile, este necesar sa se adauge si variabila lipsă. Acest lucru se realizeaza prin intermediul unei reţele cu porţi logice. La intrările unei porţi ŞI se vor aplica ieşirile DMUX-ului corespunzătoare termenilor canonici care nu apar în expresia funcţiei (se ia în considerare negata funcţiei).

**Implementarea lui F1 cu DMUX-uri de 8 căi conectate în paralel:**

La implementarea du demultipelxoare de 8 căi cele două demultiplexoare se conectează în paralel pe intrările de selecţie 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. Deoarece demultiplexorul are ieşiri active în 0 logic rezultă că pentru a realiza însumarea termenilor canonici prezenţi în expresia funcţiei se pleacă de la negata acestora aplicată prin porţi ŞI-NU, conform regulilor lui De-Morgan.







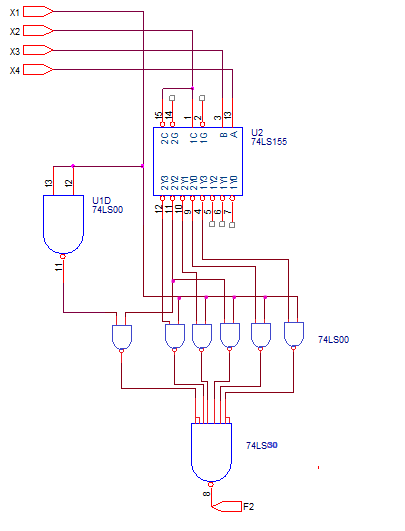
S-au folosit următoarele circuite integrate: 2x74155 (-),1x7400(-2).

**Implementarea funcţiei F2 cu DMUX 8 căi şi reţea de porţi logice**

La implementarea unei funcţii cu DMUX-uri de 8 căi şi reţea de porţi logice dintre cele *n* variabile ale funcţiei se separă *n-1* având ponderile cele mai mici şi se aplică pe intrările DMUX-ului. Astfel la ieşirile acestuia se obţin termeni canonici de *n-1* variabile. Pentru a obţine termeni canonici de *n* variabile trebuie adaugată variabila separată . Acest lucru se face în exteriorul DMUX-ului de obicei printr-o reţea de porţi logice. Deoarece ieşirile DMUX-ului sunt active în 0 logic rezultă că pentru realizarea nivelului SAU se folosesc porţi logice ŞI-NU ale funcţiei negate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echiv.  zec. | Variabilele de intrare | | | | Funcţia |
| **X1** | **X2** | **X3** | **X4** | **F2** |
| **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 | 1 |
| **8** | 1 | 0 | 0 | 0 | 1 |
| **9** | 1 | 0 | 0 | 1 | 1 |
| **10** | 1 | 0 | 1 | 0 | 1 |
| **11** | 1 | 0 | 1 | 1 | 1 |
| **12** | 1 | 1 | 0 | 0 | 1 |

****



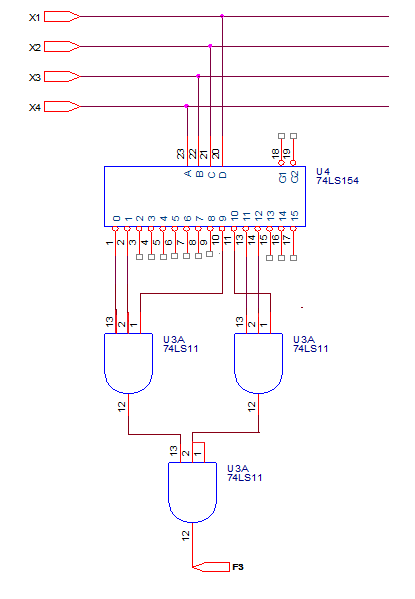
**Implementarea funcţiei logice F3 cu DMUX de 16 căi şi porti ŞI**

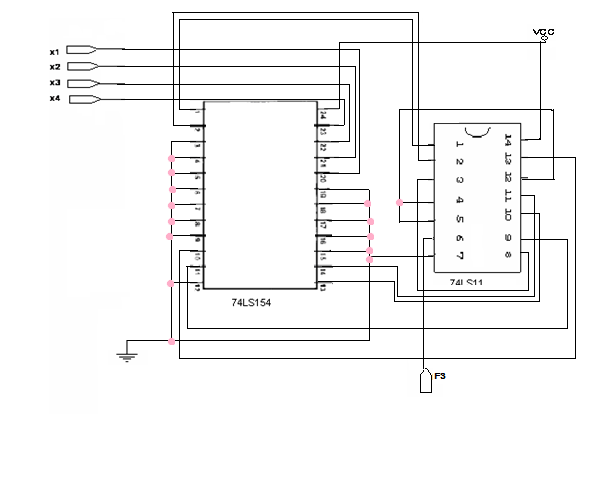
****

****

**74LS154 74LS11**

****





S-au folosit următoarele circuite integrate : 1x74LS154(-),1x74LS11(-1).

**Date de catalog ale circuitelor integrate folosite**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C.I. | Tehnologie | [ns] | [ns] | [mA] | [mA] | [mW] |
| 74LS00 | TTL | 22 | 15 | 8 | 22 |  |
| 74LS10 | TTL | 22 | 15 | 6 | 16.5 |  |
| 74LS20 | TTL | 22 | 15 | 0.8 | 2.2 |  |
| 74AC02 | CMOS | 131 | 131 | -24 | 24 |  |
| 74HC27 | CMOS | 24 | 20 | -4 | 4 |  |
| 74HC32 | CMOS | 25 | 19 | -4 | 4 |  |
| 74HC04 | CMOS | 24 | 19 | -4 | 4 |  |
| 74HC151 | CMOS | 63 | 49 | -6 | 6 |  |
| 74150 | TTL | 35 | 33 | 10 | 10 | 200 |
| 74LS155 | TTL | 20 | 27 | 40 | 40 | 125 |
| 74LS154 | TTL | 35 | 35 | 14 | 14 | 45 |
| 74LS04 | TTL | 22 | 15 | 2.4 | 6.6 |  |
| 74LS30 | TTL | 22 | 15 | 2 | 6 |  |
| 74LS11 | TTL | 15 | 20 | 3.6 | 4.6 |  |
| 74LS21 | TTL | 15 | 20 | 2.4 | 4.4 |  |

Toţi parametrii sunt măsuraţi la temperatura de 25 de grade C si tensiunea de alimentare de Vcc=5V.

**Calcularea timpilor de propagare „intrare-ieşire”**

Timpii de propagare se vor calcula după formula:



Calculul timpilor de propagare pentru fiecare din schemele următoare:

Implementarea primei funcţii logice F1 cu porţi logice ŞI-NU:



Implementarea primei funcţii logice F2 cu porţi logice ŞI-NU:



Implementarea primei funcţii logice F3 cu porţi logice ŞI-NU:



Implementarea primei funcţii logice F4 cu porţi logice ŞI-NU:



Implementarea ansamblului folosind porţi ŞI-NU:



Implementarea ansamblului folosind atât porţi SAU-NU cât şi porţi ŞI-NU:



Implementarea lui F1 cu un 2xMUX de 8 căi conectate în paralel



Implementarea lui F2 cu MUX de 8 căi şi o variabilă aplicată pe intrările de date



Implementarea lui F3 cu un MUX de 16 căi



Implementarea lui F1 cu două DMUX-uri de 8 căi conectate în paralel



Implementarea lui F2 cu un DMUX de 8 căi şi reţea de porţi logice



Implementarea primei funcţii logice F3 cu DMUX de 16 căi şi porti ŞI



**Calculul puterilor disipate:**

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



Puterile disipate la:

Implementarea primei funcţii logice F1 cu porţi logice ŞI-NU:



Implementarea funcţiei logice F2 cu porţi logice ŞI-NU:



Implementarea funcţiei logice F3 cu porţi logice ŞI-NU:



Implementarea funcţiei logice F4 cu porţi logice ŞI-NU:



Implementarea ansamblului folosind porţi ŞI-NU



Implementarea ansamblului folosind atât porţi SAU-NU cât şi porţi ŞI-NU:



Implementarea lui F1 cu un 2xMUX de 8 căi conectate în paralel



Implementarea lui F2 cu MUX de 8 căi şi o variabilă aplicată pe intrările de date



Implementarea lui F3 cu un MUX de 16 căi



Implementarea lui F1 cu două DMUX-uri de 8 căi conectate în paralel



Implementarea lui F2 cu un DMUX de 8 căi şi reţea de porţi logice



Implementarea primei funcţii logice F3 cu DMUX de 16 căi şi porti ŞI



**Concluzii finale**

Putem compara implementările obţinute din mai multe puncte de vedere.

Din punctul de vedere al complexităţii implementării schemei logice cea mai puţin complexă este schema cu un DMUX de 16 căi şi porţi ŞI-NU în tehnologie TTL (pag. 29) sau sau schema cu un MUX de 16 căi (pag. 24) şi probabil şi cele mai ieftină. La capitolul timp de propagare, schema cu MUX-uri de 16 căi CMOS este cea mai rapidă.(pag. 24) În schimb din punctul de vedere al consumul de energie, schemele cu MUX-uri sunt mari consumatoare de putere. Cea mai mică consumatoare, ţinând cont ca la unele subpuncte ale proiectului nu s-a implementat ansamblul funcţiilor ci numai una maxim două funcţii, este implementarea cu circuite CMOS (pag. 21 , MUX-urile în tehnologie CMOS ).

**Bibliografie**

1. „CURS Circuite Logice şi Comenzi Secvenţiale”

Lector: Conf. dr. ing. Florin MOLDOVEANU

1. <http://www.ti.com> – pagina oficiala a Texas Instruments, Dallas SUA;

site utilizat pentru obţinerea caracteristicilor circuitelor integrate folosite în cadrul prezentului Proiect.

1. <http://www.google.ro> – motor de căutare internaţional

4. <http://www.datasheetcatalog.com/datasheets_pdf/> -pagină de căutare a diferitelor date de catalog ale produselor diferitelor firme producătoare de circute integrate