**Univerșitatea „Tranșilvania” din Brasov**

**Facultatea de Inginerie Electrica și Știinta Calculatoarelor**

**P r o i e c t A.S.C.N.**

Îndrumator: Nume:Constantin Adrian

Prof.Florin Moldoveanu Specializarea :Automatica

Grupa:4482B

**2010**

**Cuprins :**

* **Introducere………………………………………………………….pag 3.**
* **Tema proiectului……………………………………………………pag 4.**
* **Exprimarea funcțiilor logice asociate circuitului combinațional cu FCD**

**FCC,tabel de adevar și diagrame Karnaugh..............................pag 5.**

* **Formele minime disjunctive și conjunctive ale funcțiilor............pag 8.**
* **Implementarea fiecarei funcți,independent, numai cu porți logice**

**ȘI-NU (porțile logice sunt realizate în tehnologia TTL).............pag15.**

* **Implementarea ansămblului funcțiilor logice numai cu porți logice**

**ȘI-NU (porțile logice sunt realizate în tehnologiaTTL)..............pag19.**

* **Implementarea ansămblului funcțiilor logice în urmatoarea varianta:**

**primele doua funcții logice de ieșire cu porți logice ȘI-NU, realizate**

**în tehnologia TTL, iar urmatoarele doua cu porți logice SĂU-NU,**

**realizate în tehnologia CMOS..................................................pag20.**

* **Implementarea ansămblului funcțiilor logice cu MUX-uri de 8**

**respectiv 16 căi (circuitele sunt realizate în tehnologia TTL).....pag25.**

* **Implementarea ansămblului funcțiilor logice cu DMUX-uri de 8**

**respectiv 16 căi și porți logice ȘI-NU în prima varianta, respectiv**

**ȘI în a doua varianta (tehnologia CMOS).................................pag31.**

* **Timpii de propagare „intrare-ieșire”, pentru toate schemele logice**

**obtinute..................................................................................pag39.**

* C**alculeze puterile dișipate pentru schemele logice obținute......pag40**
* **Compararea soluțiilor de implementare obținute.....................pag41.**
* **Bibliografie.............................................................................pag41.**

**Introducere**

Un circuit logic combinațional este un circuit de comutare combinaționala ce se caracterizeaza prin aceea că la un moment dat starea ieșirilor circuitului depinde doar de starea intrarilor săle. Legatura dintre starea ieșirilor și starea intrarilor este data și în acest caz de funcția de transfer a circuitului.

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

Schema bloc generala (modelul general) al unui CLC reprezentat printr-o rețea de comutare său schema logică este urmatorul:

Reţea de comutare

X1

X2

Xn

Z1

Z2

zm

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

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

Relații generale între aceste valori:

Z1 = f(X1, …,Xn) fi – functii logice

……………. Xi – variabile logice

Zm = f(X1, …,Xn)

Analiza circuitelor logice combinaționale

Prin analiza unui CLC se înțelege obținerea expreșiilor marimilor de ieșire cunoscandu-se setul variabilelor de intrare  și configurația rețelei. Înțelegem numarul și tipul de elemente logice care intra în componență,modul de conectare, punctul în care se aplică variabilele de intrare, numarul de nivele logice etc.

Analiza rețelelor CLC realizate cu elemente logice de tip inversor (ȘI-NU, SĂU-NU): numarul maxim de elemente logice aflate între intrarea și ieșirea rețelei determina numarul de nivele logice ale acestuia. Numerotarea lor se face de la ieșire catre intrare. Într-o rețea de comutare realizate cu elemente ȘI-NU respectiv SĂU-NU s-a observat că o variabila de intrare apare negata în expreșia ieșirii în cazul în care a parcurs un numar impar de elemente de inversăre și necomplementata daca a parcurs un numar par de astfel de elemente.

TEMA PROIECT Nr.73

Să se proiecteze un convertor de cod de 4 biti pentru converșia codului binar zecimal ponderat 4221 în codul binar zecimal ponderat 5421 (logica combinaționala).Proiectul va cuprinde urmatoarele puncte:

a) Să se exprime funcțiile logice asociate circuitului combinațional cu FCD (forma canonica disjunctiva), FCC (forma canonica conjunctiva) tabel de adevar și diagrame Karnaugh.

b) Să se obtină formele minime disjunctive și conjunctive pentru funcțiile logice asociate convertorului de cod, (utilizand combinațiile indiferente) prin metoda diagramelor Karnaugh; de asemenea se vor obtine formele minime disjunctive pentru **primele** două functii logice de ieșire și prin metoda Quine-McCluskey.

c) Să se implementeze fiecare funcție, independent, numai cu porți logice ȘI-NU (porțile logice sunt realizate în tehnologia TTL).

d) Să se implementeze ansămblul funcțiilor logice numai cu porți logice ȘI-NU (porțile logice sunt realizate în tehnologiaTTL).

e) Să se implementeze ansămblul funcțiilor logice în urmatoarea varianta: primele doua funcții logice de ieșire cu porți logice **ȘI-NU**, realizate în tehnologia **TTL**, iar urmatoarele doua cu porți logice **SĂU-NU**, realizate în tehnologia **CMOS**.

f) Să se implementeze ansămblul funcțiilor logice cu MUX-uri de 8 respectiv 16 căi (circuitele sunt realizate în tehnologia **TTL**).

g) Să se implementeze ansămblul funcțiilor logice cu DMUX-uri de 8 respectiv 16 căi și porți logice ȘI-NU în prima varianta, respectiv ȘI în a doua varianta (toate circuitele sunt realizate în tehnologia **CMOS**).

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

i) Să se calculeze puterile dișipate pentru toate schemele logice obținute.

j) Să se compare soluțiile de implementare obținute.

k) Se va face analiza, prin șimulare, a tuturor schemelor logice obținute utilizandu-se pachetul de programe OrCAD.

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

a) Să se exprime funcțiile logice asociate circuitului combinațional cu FCD (forma canonica disjunctiva), FCC (forma canonica conjunctiva) tabel de adevar și diagrame Karnaugh.

Convertorul de cod - generalitați

Sunt circuite logice combinaționale care in cazul general au n intrari și n ieșiri și servesc la transcrierea informației dintr-un cod binar in alt cod binar.

La baza proiectarii unui convertor de cod se află tabelul de corespondentă dintre cuvintele de cod din/in care se face converșia. Notand fiecare din pozițiile binare a acestor cuvinte cu cate o variabilă, tabelul de corespondență se transformă intr-un tabel de adevăr.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Codul 4221 | | | | Codul 5432 | | | |
| X1 | X2 | X3 | X4 | A | B | C | D |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 8 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

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

Funcțiile logice cu FCD și FCC și diagramele Karnaugh

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

*forma produsului de sume = forma canonică conjuctiva.*

*forma sumei de produse = forma canonică conjuctiva.*

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

Diagramele Karnaugh

Diagrama Karnaugh reprezintă un ansămblu de 2n locații în care se notează cu n numarul variabilelor funcției boolene ce urmează a se minimiza, dispuse într-o anumită ordine fiecare locație corespunzand unui termen P său S. Diagrama Karnaugh este constituită astfel încît să nu existe termeni canonici ai funcției care să nu apară în diagrama și de asemenea să nu existe două său mai multe compartimente care să contina același termen canonic ( I principiu).

Totodata diagramele Karnaugh sunt astfel organizate încat două compartimente vecine pe linie său pe coloană să contină două combinații binare care au proprietatea de adiacența ( al-II-lea principiu). Se conșideră vecine și au proprietatea de adiacență și combinațiile plasăte în locații șituate la extremitațile liniilor și coloanelor diagramei.

Diagramele de mai mult de 4 variabile se pot obține prin alăturarea pe linie său pe coloană a unor diagrame Karnaugh de 4 variabile conșiderate ca diagrame elementare; în diagramele de mai mult de 4 variabile se conșideră vecine și au proprietatea de adiacența și locațiile său grupuri de locații care ocupă poziții șimetrice în două diagrame elementare.

1) 





Diagrama Karnaugh pentru funcția A

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**0**

**\***

**1**

**1**

**0**

**\***

**1**

**\***

**0**

**0**

**1**

**\***

2) 

Diagrama Karnaugh pentru funcția B

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**0**

**0**

**0**

**\***

**1**

**\***

**0**

**1**

**0**

**\***

3) 

Diagrama Karnaugh pentru funcția C

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**1**

**0**

**1**

**\***

**0**

**\***

**1**

**0**

**1**

**\***

4)  Diagrama Karnaugh pentru funcția D

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**1**

**\***

**0**

**0**

**1**

**\***

**0**

**\***

**0**

**0**

**1**

**\***

b) Să se obtinaâă formele minime disjunctive și conjunctive pentru funcțiile logice asociate convertorului de cod, (utilizand combinațiile indiferente) prin metoda diagramelor Karnaugh; de asemenea se vor obtine formele minime disjunctive pentru **primele** două funcții logice de ieșire și prin metoda Quine-McCluskey.

Pentru deducerea FMD (formei minime disjuctive) a funcției se aleg combinațiile cele mai avantajoase de termeni adiacenți astfel încît să se reducă dupa regulă  și să rezulte cît mai puțini termeni (implicați primi) cu cît mai puține variabile.

Pentru obținerea FMC (formei minime conjunctive), se obține mai întîi forma minima conjunctiva a funcției negate. Apoi se neaga această formă minimă, și folosind formulele lui DeMorgan, se obtine forma minima conjunctiva a funcției date.

Formele minime disjunctive cu diagrama Karnaugh

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**0**

**\***

**1**

**1**

**0**

**\***

**1**

**\***

**0**

**0**

**1**

**\***





X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**0**

**0**

**0**

**\***

**1**

**\***

**0**

**1**

**0**

**\***

**

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**1**

**0**

**1**

**\***

**0**

**\***

**1**

**0**

**1**

**\***



X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**1**

**\***

**0**

**0**

**1**

**\***

**0**

**\***

**0**

**0**

**1**

**\***

Formele minime conjunctive cu diagrame Karnaugh

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**0**

**\***

**1**

**1**

**0**

**\***

**1**

**\***

**0**

**0**

**1**

**\***





X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**0**

**0**

**0**

**\***

**1**

**\***

**0**

**1**

**0**

**\***

**

X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**0**

**\***

**0**

**\***

**1**

**0**

**1**

**\***

**0**

**\***

**1**

**0**

**1**

**\***



X1X2

X3X4

00 01 11 10

00

01

11

10

X1

X4

X2

X3

**0**

**\***

**1**

**\***

**1**

**\***

**0**

**0**

**1**

**\***

**0**

**\***

**0**

**0**

**1**

**\***

Formele minime disjunctive pentru **primele** două funcții logice de ieșire și prin metoda

Quine-McCluskey.

Pentru funcția A





Combinații indiferente



|  |  |  |
| --- | --- | --- |
| Grupa | Indicii | X1X2X3X4 |
| 1 | \*4 | 0 1 0 0 |
| \*8 | 1 0 0 0 |
| 2 | \*5 | 0 1 0 1 |
| 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 |

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii | X1X2X3X4 |
| 1 | 4,12 | - 1 0 0 |
| 8,9 | 1 0 0 - |
| 8,12 | 1 - 0 0 |
| 2 | 5,13 | - 1 0 1 |
| 9,11 | 1 0 - 1 |
| 9,13 | 1 - 0 1 |
| 10,14 | 1 - 1 0 |
| 12,13 | 1 1 0 - |
| 12,14 | 1 1 - 0 |
| 3 | 7,15 | - 1 1 1 |
| 11,15 | 1 - 1 1 |
| 13,15 | 1 1 - 1 |
| 14,15 | 1 1 1 - |

|  |  |  |
| --- | --- | --- |
| Gr | Indice | X1X2X3X4 |
| 1 | 4,5,12,13 | - 1 0 - |
| 8,9,12,13 | 1 - 0 - |
| 8,10,12,14 | 1 - - 1 |
| 2 | 5,7,13,15 | - 1 - 1 |
| 9,1113,15 | 1 - - 1 |
|  | 10,11,14,15 | 1 - 1 - |
|  | 12,13,14,15 | 1 1 - - |

|  |  |  |
| --- | --- | --- |
| Gr | Indice | X1X2X3X4 |
| 1 | 8,9,10,11,12,13,14,15 | 1 - - - |

Dupa ultimul ciclu de comparare (atunci cînd nu mai avem termeni de comparat) vom obține o formă minima, pe care nu o putem conșidera ultima formă minimă a funcției.

Pentru aceasta trebuie să alegem dintre implicanții primi obtinuți la sfârțitul primei etape numarul minim care să acopere său să includa toți termenii canonici ai funcției canonice. Pentru aceasta se face tabelul acoperirilor :

La intersecția unei linii cu o coloană se va plasă un \* care va indica faptul ca implicantul prim de pe linia respectivă va include termenul canonic de pe coloana corespunzatoare.

Daca pe o coloană apare un șingur \*, implicantul prim de pe acea linie se numeste *implicant prim esențial* și va apare în forma minima disjunctiva in mod obligatoriu.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Implicanți | | | | Indici | Termeni canonici | | | | | | | | | | | | | | | | | |
| X X2 X3X4 | | | |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| - | 1 | 0 | - | 4,5,12,13 |  |  |  |  | \* | \* |  |  |  |  |  |  | \* | \* |  |  |  |
| - | 1 | - | 1 | 5,7,13,15 |  |  |  |  |  | \* |  | \* |  |  |  |  |  | \* |  | \* |  |
| 1 | - | - | - | 8,9,10,11,12,13,14,15 |  |  |  |  |  |  |  |  | \* | \* | \* | \* | \* | \* | \* | \* | IPE |



Pentru funcția B





Combinații indiferente



|  |  |  |
| --- | --- | --- |
| Grupa | Indicii | X1X2X3X4 |
| 1 | \*4 | 0 1 0 0 |
| \*8 | 1 0 0 0 |
| 2 | \*5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| \*10 | 1 0 1 0 |
| 3 | \*7 | 0 1 1 1 |
| \*11 | 1 0 1 1 |
| 4 | 15 | 1 1 1 1 |

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii | X1X2X3X4 |
| 1 | 4,6 | 0 1 - 0 |
| 2 | 6,7 | 0 1 1 - |
| 3 | 7,15 | - 1 1 1 |
| 11,15 | 1 - 1 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Implicanți | | | | Indici | Termeni canonici | | | | | | | | | | | | | | | | | |
| X X2 X3X4 | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| 0 | 1 | - | 0 | 4,6 |  |  |  |  | \* |  | \* |  |  |  |  |  |  |  |  |  | IPE |
| 0 | 1 | 1 | - | 6,7 |  |  |  |  |  |  | \* | \* |  |  |  |  |  |  |  |  |  |
| - | 1 | 1 | 1 | 7,15 |  |  |  |  |  |  |  | \* |  |  |  |  |  |  |  | \* |  |
| 1 | - | 1 | 1 | 11,15 |  |  |  |  |  |  |  |  |  |  |  | \* |  |  |  | \* | IPE |



c) Să se implementeze fiecare funcție, independent, numai cu porți logice ȘI-NU (porțile logice sunt realizate în tehnologia TTL).

Implementarea funcțiilor reprezintă punerea în practica a informatiilor și a rezultatelor obtinute pâna acum. Pentru ca implementarea să fie cât mai avantajoasă se tine cont de unii factori astfel încât să obtinem circuite cât mai fiabile la preturi de cost cât mai scazute.

Metodele matematice prezentate anterior sunt insuficiente pentru proiectarea cu circuite integrate. Unul din scopurile importante ale acesteia este obținerea unor scheme cu numar minim de circuite integrate cu un cost cât mai scazut .

Conform cerintelor vom implementa functiile cu porți logice ȘI-NUîn tehnica integrata TTL. Pentru a putea realiza schema cu porți logice ȘI-NU plecam de la una din formele minime (în acest caz de la FMD), prelucrata astfel încât operatiile logice să fie compatibile cu tipul de poarta logica utilizata în circuit ȘI-NU, și anume se va nega de doua ori a functia și se vor folosi formulele lui De Morgan. Se vor folosi circuite integrate din productia Texas Instruments din seria 74LS.

Încapsularea porților in circuitele integrate care se vor folosi :

** **

Implementarea funcției A.





S-a folosit un circuit integrat 74LS00.

Implementarea funcției B.



X3

X1

U6A

74LS10

1

2

13

12

U5A

74LS00

1

2

3

U4A

74LS00

1

2

3

B

X2

U3A

74LS00

1

2

3

X4

2

12

X2

3

9

14

4

13

7

12

10

11

5

10

X1

1

74LS10

11

9

14

8

B

13

X3

4

VCC

3

1

8

5

VCC

6

6

74LS00

7

2

X4

S-a folosit un circuit integrat 74LS00 și unu 74LS10.

Implementarea funcției C.

**

U8A

74LS10

1

12

2

13

U6A

74LS00

1

2

3

C

X1

U5A

74LS00

1

2

3

U9A

74LS10

1

12

2

13

U7A

74LS10

1

12

2

13

X2

U3A

74LS00

1

2

3

X3

U4A

74LS00

1

2

3

X4



S-au folosit doua circuite integrate,unu 74LS00 și unu 74LS10.

Implementarea funcției D.







S-au folosit doua circuite integrate 74LS00.

d) Să se implementeze ansămblul funcțiilor logice numai cu porți logice ȘI-NU (porțile logice sunt realizate în tehnologiaTTL).





S-au folosit trei circuite integrate 74LS00 și doua 74LS10.

e) Să se implementeze ansămblul funcțiilor logice în urmatoarea varianta: primele doua functii logice de ieșire cu porți logice **ȘI-NU**, realizate în tehnologia **TTL**, iar urmatoarele doua cu porți logice **SĂU-NU**, realizate în tehnologia **CMOS**.

Circuitul ce comanda poate fi TTL și circuitul comandat poate fi CMOS, său invers,circuitul de comanda poate fi CMOS și circuitul comandat TTL. Vom conșidera ambele șituatii. De asemenea, deoarece circuitele CMOS permit alimentare și la tenșiuni mai mari de 5V, vom conșidera și un exemplu în care tenșiunile de alimentare ale celor doua circuite difera.

Când circuitele TTL trebuie să comande circuite CMOS alimentate dintr-o șingura sursă de tenșiune de 5V, nivelul minim de ieșire în stare High garantat de TTL (2.4V) este mai mic decât nivelul minim de intrare acceptat de CMOS pentru stare High (3.5V). Se poate observa aceasta șituatie în Fig. 2.8. care alatura cele doua restrictii de nivele de tenșiune.



Se poate creste nivelul de ieșire în stare High al circuitului TTL utilizând o rezistenta conectata între ieșire și VCC ca în Fig. 2.9.



Valoarea minima a rezistentei este determinata de curentul maxim absorbit pe la ieșirea TTL (16mA pentru TTL standard), iar valoarea maxima este determinata de curentul tranzistorului de ieșire în stare blocat. Valorile cele mai indicate pentru aceasta rezistenta se șitueaza în gama 1.5 kΩ ... 4.7 kΩ pentru toate familiile TTL. În Tabelul 2. 5. se gasesc valori recomandate pentru rezistenta suplimentara pentru familiile HTTL și LTTL.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rx** | **TTL** | **standard HTTL** | **LTTL** |
| Rxmin (Ω) | 390 | 270 | 1500 |
| Rxmax (kΩ) | 4.7 | 4.7 | 27 |

În cazul interfetei CMOS-TTL, cerinta de baza este ca ieșirea CMOS să poata absorbi un curent suficient în stare Low, la o tenșiune maxima de ieșire de 0.4V. Majoritatea circuitelor din seria MMC 4xxx au aceeași capabilitate de curent la ieșire, putând să comande o sărcina LSTTL (Low power Schottky) în conditiile cele mai defavorabile (0.36mA la VO = 0.4V și VDD = 5V). Pentru circuite de puteri mai mari trebuie utilizate numai circuite buffer său circuite open-drain. Circuitele buffer MMC se alimenteaza din sursă TTL de 5V și au avantajul ca pot fi comandate la intrare cu tenșiuni variind între 5V și 15 V. Interfatarea se mai poate face și cu translatoare de nivel. Schema de legatura este șimpla, este redata în Fig. 2.10.



În șistemele care utilizeaza alimentarea circuitelor CMOS din surse cu tenșiuni mai mari de 5V, viteza și imunitatea la zgomot se pot îmbunatati prin utilizarea circuitelor TTL opencollector. Solutia este ilustrata în Fig. 2.11.



Valoarea rezistentei Rx depinde de valoarea tenșiunii sursei de alimentare VDD (la VDD=10V, se recomanda utilizarea unei rezistente de 39 kΩ).

****  






S-au folosit un circuit 74LS00,unu 74LS10,doua 74HC02,unu 74HC27.

f) Să se implementeze ansămblul funcțiilor logice cu MUX-uri de 8 respectiv 16 cai (circuitele sunt realizate în tehnologia **TTL**).

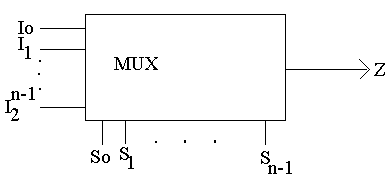
**Implementarea cu multiplexoare:**

Un circuit de multiplexare este un circuit logic combinational care, în cazul general, are 2n intrari de date (I2n-1 … I2 I1 I0), n intrari de selectie (S0 S1 … Sn+1) și o ieșire (Z).

Multiplexoarele genereaza termenii canonici în interior, iar poșibilitatea utilizarii multiplexoarelor MUX 2n:1pentru implementarea funcțiilor logice de n variabile sub forma FCD decurge din faptul ca prin structura să se obtin toti termenii canonici de “n“‚ variabile (nivelul logic ȘI) cât și suma logica complexa între acesti termeni (nivelul logic SĂU).

De asemenea, spre deosebire de metodele clașice de implementare cu porți logice, la care în prealabil functia său functiile logice sunt aduse la formele minime, în cazul implementarii cu multiplexoare, se porneste de la FCD.

Multiplexoarele aduna termenii canonici pe care îi genereaza în interior, având intrari de validare ce permit numai transmiterea acelora ce intervin în expreșia funcției.

Schema bloc generala:

Expreșia ieșirii Z la un moment dat este data de intrare Ik, k=0, … ,2n-1 unde k reprezintă echivalentul zecimal al numarului binar dat de starile 1 și 0 ale intrarilor de selectie: k=Sn-1, Sn-2, … , S1, S0.

Încapsularea circuitelor:



Implementarea funcției C cu multiplexoare cu 8 cai cu o variabila pe intrarile de date.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echiv.  zec. | Variabilele de intrare | | | | Functia |
| **X1** | **X2** | **X3** | **X4** | **C** |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 1 | 0 | 1 |
| **3** | 0 | 0 | 1 | 1 | 1 |
| **6** | 0 | 1 | 1 | 0 | 0 |
| **9** | 1 | 0 | 0 | 1 | 0 |
| **12** | 1 | 1 | 0 | 0 | 0 |
| **13** | 1 | 1 | 0 | 1 | 1 |
| **14** | 1 | 1 | 1 | 0 | 1 |
| **15** | 1 | 1 | 1 | 1 | 0 |

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

X1X2X3=001 b=0 oricare ar fi X4 =>D1=1

X1X2X3=011 b=0 pentru X4 =0 =>D3= 0

X1X2X3=100 b=0 pentru X4 =1=>D4= 0

X1X2X3=110 b=1 oricare ar fi X4 =>D6=X4

X1X2X3=111 b=1 oricare ar fi X4 =>D7= 





14

VCC

X3

MUX

0

1

2

3

4

5

6

7

EN

0

2

G

0

7

U1

74151

4

3

2

1

15

14

13

12

11

10

9

6

5

7

11

6

X2

VCC

2

9

10

X1

13

3

12

74LS00

5

C

4

8

7

X4

1

S-a folosit un circuit 74LS00 și unu 74151.

Implementarea funcției D cu multiplexoare cu 8 cai cu o variabila pe intrarile de date.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Codul 4221 | | | | Codul 5432 |
| X1 | X2 | X3 | X4 | Functia D |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 |





S-au folosit doua circuite 74151,unu 74LS00,unu 74HC02.

Implementarea cu mux cu 16 cai a funcției A.



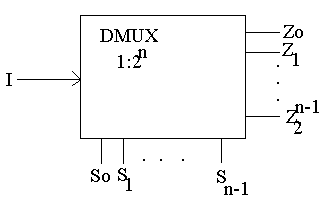
S-a folosit un circuit 74150.

g) Să se implementeze ansămblul funcțiilor logice cu DMUX-uri de 8 respectiv 16 cai și porți logice ȘI-NU în prima varianta, respectiv ȘI în a doua varianta (toate circuitele sunt realizate în tehnologia **CMOS**).

**Implementarea cu demultiplexoare:**

Demultiplexoarele sunt circuite combinationale care, în cazul general au o intrare de date I, n intrari de selectie S0, S1, …, Sn-1 și 2ieșiri Z0, Z1, …, Z2.

Schema bloc:



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

Pentru realizarea nivelului logic SĂU se pot folosi porți:

1. ȘI-NU, în acest caz se leaga la intrarile porților ȘI-NU ieșirile DMUX corespunzatoare termenilor canonici prezentati în expreșia funcției de implementat.
2. ȘI, în acest caz se leaga la intrarile porților ȘI ieșirile DMUX corespunzatoare termenilor canonici care nu apar în expreșia funcției (se vor conșidera functiile negate).

În cazul în care functia are, în caz general, n variabile și 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 ieșirile acestor circuite se obtin termeni canonici de n-1 variabile, iar în forma în care a fost functia de implementat sunt termeni canonici de n variabile, este necesăr să se adauge și variabila lipsă. Acest lucru se realizeaza prin intermediul unei retele cu porți logice.



Implementarea funcției D cu demultiplexor cu8 cai cu o retea de porți.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Codul 4221 | | | | Codul 5432 |
| X1 | X2 | X3 | X4 | Functia D |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 |





S-au folosit doua circuite 74LS00,și unu 74HC155.

Implementarea funcției D cu doua demultiplexoare cu 8 cai legate in paralel.





S-au folosit doua circuite integrate 74LS00,și doua 74HC155.

Implementarea funcției B cu demultiplexor cu 2 cai porți Și-Nu.

X1

X2

U3A

74LS00

1

2

3

U2

74HC154

1

2

3

4

5

6

7

8

9

10

11

13

14

15

16

17

23

22

21

20

18

19

Y0

Y1

Y2

Y3

Y4

Y5

Y6

Y7

Y8

Y9

Y10

Y11

Y12

Y13

Y14

Y15

A

B

C

D

G1

G2

X4

X3

B

10

8

X4

74HC02

X3

VCC

U1

74HC154

1

2

3

4

5

6

7

8

9

10

11

13

14

15

16

17

23

22

21

20

18

19

Y0

Y1

Y2

Y3

Y4

Y5

Y6

Y7

Y8

Y9

Y10

Y11

Y12

Y13

Y14

Y15

A

B

C

D

G1

G2

X1

1

3

2

11

6

X2

13

7

12

14

5

9

4

B

S-a folosit un circuit 74HC154și unu 74HC02.

Implementarea funcției B cu demultiplexor cu 16 cai porți Și.



**74LS06**



6

13

4

10

8

X4

74LS02

X3

VCC

U1

74HC154

1

2

3

4

5

6

7

8

9

10

11

13

14

15

16

17

23

22

21

20

18

19

Y0

Y1

Y2

Y3

Y4

Y5

Y6

Y7

Y8

Y9

Y10

Y11

Y12

Y13

Y14

Y15

A

B

C

D

G1

G2

X1

1

1

3

2

12

2

10

14

11

6

X2

74LS08

13

7

12

11

14

5

8

5

7

9

4

9

3

B

VCC

S-a folosit un circuit 74LS00,unu 74LS08,unu 74HC154.

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

**Date de catalog ale circuitelor integrate folosite**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **C.I.** | **Tehnologie** | **tPLH**  **[ns]** | **tPHL**  **[ns]** | **ICCH(TTL)**  **IOH(CMOS)**  **[mA]** | **ICCL(TTL)**  **IOL(CMOS)**  **[mA]** | **tp**  **[ns]** | **Pd**  **[mW]** |
| 74LS00 | TTL | 10 | 10 | 0.8 | 2.4 | 10 | 8 |
| 74LS06 | TTL | 10 | 15 | 23 | 21 | 12,5 |  |
| 74LS10 | TTL | 10 | 10 | 0.6 | 1.8 | 10 | 6 |
| 74HC02 | CMOS | 40 | 40 |  |  | 90 | 500 |
| 74HC27 | CMOS | 27 | 27 |  |  | 27 | 500 |
| 74HC155 | CMOS | 160 | 32 |  |  | 96 | 185 |
| 74HC154 | CMOS | 190 | 38 |  |  | 114 | 185 |
| 74151 | CMOS | 38 | 30 |  |  | 34 | 290 |
| 74150 | CMOS | 35 | 33 |  |  | 34 | 200 |

**\***Pentru aflarea puterii dișipate a integratelor in tehnologie CMOS am folosit formula:

, unde Vcc=5 V, fi=105 , CL=50 pF, iar Cp reprezintă puterea dișipata și se gaseste in datele de catalog ale integratelor.

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

Timpii de propagare se vor calcula dupa formula:

Calculul timpilor de propagare pentru fiecare din schemele urmatoare:

Implementarea funcției logice **A** cu porți logice ȘI-NU:

tp=0.5(10+10)+0.5(10+10)=**20ns**

Implementarea funcției logice **C** cu porți logice ȘI-NU:

tp=0.5(10+10)\*2+0.5(10+10)=**30ns**

Implementarea funcției logice **C** cu porți logice ȘI-NU:

tp=0.5(10+10)\*2+0.5(10+10)=**30ns**

Implementarea funcției logice **D** cu porți logice ȘI-NU:

tp=0.5(10+10)\*3=**30ns**

Implementarea **ansămblului** folosind porți ȘI-NU:

tp=0.5(10+10)\*2+0.5(10+10)=**30ns**

Implementarea **ansămblului** folosind atât porți SĂU-NU cât și porți ȘI-NU, vom lua timpul cel mai lung prin circuit:

tp=0.5(10+10)\*2+0.5(10+10)=**30ns**

Implementarea lui **D** cu un 2xMUX de 8 cai conectate în paralel:

tp=0.5(10+10)+0.5(38+30)+0.5(40+40)=10+34+40=**84**

Implementarea lui **C** cu MUX de 8 cai:

tp=0.5(10+10)+0.5(38+30)=**44ns**

Implementarea lui **A** cu un MUX de 16 cai:

tp=0.5(35+33)=**34ns**

Implementarea lui **D** cu 2xDMUX-uri de 8 cai conectate în paralel:

tp=0.5(10+10)\*3+0.5(160+32)=**126ns**

Implementarea lui **D** cu un DMUX de 8 cai și retea de porți logice:

tp=0.5(160+32)+0.5(10+10)\*3=**126ns**

Implementarea lui **C** cu DMUX 16 cai și porți logice ȘI-NU:

tp=0.5(14+14)+0.5(190+38)=**128ns**

Implementarea lui **D** cu DMUX 16 cai și porți logice ȘI:

tp=0.5(14+14)+0.5(190+38)=**128ns**

**Calculul puterilor dișipate**

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

Puterile dișipate la:

Implementarea funcției logice **C** cu porți logice ȘI-NU:

Pd=P74LS00+P74LS10=8+2.5(0.6+1.8)=**14mW**

Implementarea funcției logice **D** cu porți logice ȘI-NU:

Pd=P74LS00\*=**24mW**

Implementarea **ansămblului** folosind porți ȘI-NU:

Pd=P74LS00\*2+P74LS10=16+2.5(0.6+1.8)=**22mW**

Implementarea **ansămblului** folosind atât porți SĂU-NU cât și porți ȘI-NU:

Pd= P74HC02 \*2+ P74LS00+P74LS10=500+8+6=**514mW**

Implementarea lui **D** cu un 2xMUX de 8 cai conectate în paralel:

Pd=P74LS151\*2+P74LS00+P74HC02=290+10+500=**800mW**

Implementarea lui **C** cu MUX de 8 cai:

Pd=P74LS150= 145+10=**155mW**

Implementarea lui A cu un MUX de 16 cai:

Pd=P74LS150=**200mW**

Implementarea lui **D** cu 2xDMUX-uri de 8 cai conectate în paralel:

Pd=P74HC155+ P74HC02+ P74HC27=185+500+500=**1185mW**

Implementarea lui **D** cu un DMUX de 8 cai și retea de porți logice:

Pd=P74HC155+ P74HC02=185+500=**685mW**

Implementarea lui **B** cu DMUX 16 cai și porți logice ȘI-NU:

Pd=P74HC154+ P74LSOO=185+8=**193mW**

Implementarea lui **D** cu DMUX 16 cai și porți logice ȘI:

Pd=P74HC154+ P74HC27=185+500=**685mW**

**Concluzii finale**

Putem compara implementarile obtinute din mai multe puncte de vedere.

Din punctul de vedere al complexitatii implementarii schemei logice cea mai putin complexa este schema cu un MUX de 16 cai în tehnologie TTL său schema cu un DMUX de 16 cai și probabil și cele mai ieftine. La capitolul timp de propagare, schema cu MUX-uri de 16 cai TTL este cea mai rapida.Din punctul de vedere al consumul de energie, schema cu 2xDMUX-uri de 8 cai este cea mai mare consumatoare de putere.

**Bibliografie**

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

șite utilizat pentru obținerea caracteristicilor circuitelor integrate folosite în cadrul prezentului Proiect.

* <http://www.google.ro> – motor de cautare international
* <http://www.datasheetcatalog.com/datasheets_pdf/> -pagina de cautare a diferitelor date de catalog ale produselor diferitelor firme producatoare de circute integrate