|  |  |
| --- | --- |
| univ | UNIVERSITATEA TRANSILVANIA BRAŞOV Facultatea de Inginerie Electrica şi Ştiinţa Calculatoarelor  Specializare:Automatica şi Informatică Aplicată |

PROIECT

ANALIZA ŞI SINTEZA CIRCUITELOR NUMERICE

Îndrumător : Student:

Prof. dr. ing. Moldoveanu Florin Ilinca Elena Victoriţa

Anul II Grupa: 4491

Braşov 2010-2011

**Cuprins**

Noţiuni introductive………………………………………………………………………………….. 3

Enunţul proiectului ........................................................................................ ..................................... 4

Tabelul de adevăr ................................................................................................................................ 6

Formele canonice disjunctive şi conjunctive ale funcţiilor...................................................................7

Formele minime disjunctive şi conjunctive prin metoda diagramelor Karnaugh................................ 9

Minimizarea funcţiilor *c*  şi *e* prin metoda Quine-McCluskey ...................................................... . ... 16

Implementarea funcţiilor logice individuale cu porţi ŞI-NU TTL ..................................................... 21

Implementarea ansamblului cu porţi SI-NU TTL .......... .................................................................. 28

Implementarea ansamblului cu porţi SAU-NU CMOS si ŞI-NU TTL............................................. 29

Implementare cu MUX-uri în tehnologie CMOS ............................................................................. 31

Implementare cu DMUX-uri în tehnologie TTL ............................................................................... 34

Calculul timpilor de propagare .......................................................................................................... 40

Calcului puterilor disipate ................................................................................................................. 41

Încapsularea circuitelor integrate………………………………………………………………….42

Implementarea cu CI a ansamblului funcţiilor TTL……………………………………………….43

Implementarea cu CI a ansamblului funcţiilor MUX/DMUX…………………………………….44

Concluzii finale ................................................................................................................................. 45

Bibliografie......................................................................................................................................... 46

**Noţiuni introductive**

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)

Decodificatoarele (DCD-urile) reprezintă o clasă de circuite logice combinaţionale care, în cazul general, au *n* intrări, , ,..., , pe care se aplică cei *n* biţi ai cuvântului deşi *m* ieşiri (*m* ≤ 2*n* ) , , ,…, ; pentru fiecare cuvânt de cod aplicat la intrare se obţine semnal logic doar pe una dintre ieşiri sau, se poate spune că decodificatorul serveşte la identificarea unui cod de intrare prin activarea unei singure linii de ieşire, corespunzătoare acestui cod [2,15,22,23,26÷30,33]. Decodificarea este necesară în numeroase aplicaţii cum sunt: adresarea memoriilor, afişarea numerică, multiplexarea datelor, etc.

Decodificatorul *BCD-*7 *segmente* este un circuit de decodificare utilizat la comanda sistemelor de afişare numerică realizate din şapte segmente luminoase: becuri, diode electroluminiscente sau cristale lichide. Decodificatorul studiat este un CLC cu patru intrări, notate cu , , *,‍* şi şapte ieşiri notate *a*, *b*, *c*, *d*, *e*, *f*, *g.*

Se consideră două mulţimi de elemente A şi B . A codifica elementele mulţimii A prin elementele mulţimii B,înseamnă a face să corespundă fiecărui element aA o secvenţă de elemente b.

Dacă se notează cu I mulţimea şirurilor de elemente din B,atunci aplicaţia care asociază fiecarui element din A un element din I se numeşte cod şi se notează cu: f : A → I.

Codul 5421 este un cod ponderat şi are caracteristic faptul că cifrele zecimale de la 5 pâna la 9 se deosebesc de cifrele de la 0 la 4 numai prin primul bit

**Enunţul proiectului**

Să se proiecteze un decodificator BCD 5421/ 7 segmente (logică combinaţională).Se va studia cazul în care elementele tubului de afişare cu 7 segmente sunt aprinse iniţial cât şi cazul în care elementele tubului sunt stinse iniţial. Proiectarea se va referi la o singură decadă.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 diagrame Karnaugh.

b) Să se obţină formele minime disjunctive şi conjunctive pentru funcţiile logice asociate decodificatorului BCD 5421/ 7 segmente (utilizând combinaţiile indife-rente) prin metoda diagramelor Karnaugh; se vor obţine, de asemenea, formele minime disjunctive pentru două dintre funcţiile logice de ieşire 3,5 şi prin metoda Quine-McCluskey.

c) Să se implementeze fiecare funcţie logică, 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 4 funcţii logice de ieşire cu porţi logice ŞI-NU, realizate în tehnologia TTL, iar următoarele 3 cu porţi logice SAU-NU, realizate în tehnologia CMOS.

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.

Tabelul de adevăr

Acest mod de reprezentare corespunde reprezentării tabelare a funcţiilor bo oleene. Tabelele denumite *de adevăr* sau *combinaţionale* conţin în partea stângă un număr delinii egal cu numărul combinaţiilor posibile ale valorilor argumentelor, iar în partea dreaptă valorile funcţiei pentru fiecare combinaţie de valori ale argumentelor.

Tabelul de adevăr este cea mai completă reprezentare a unei funcţii booleene deoarece pentru fiecare combinaţie posibilă a valorilor argumentelor se indică valoarea funcţiei.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Intrări BCD 5421 | | | | Ieşiri 7 segmente | | | | | | |
| Cifra | x1 | x2 | x3 | x4 | a | B | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

digit

Există situaţii când pentru unele combinaţii ale valorilor argumentelor, o funcţie booleană să nu aibă valoarea determinată,aceste combinaţii se numesc combinaţii indiferente. În mod obişnuit în tablele de definiţie ale funcţiilor, valorile nedeterminate sunt indicate cu “\*” sau “*d”* ( Simbolul „*d*” provine de la prescurtarea expresiei englezeşti *„don’t care”* ,nu ţine cont).

Combinaţiile indiferente sunt: *d = \* ======*

*P₄==0100*

*P₅=x₂x₄=0101*

*P₇=x₂x₃x₄=0111*

*P₁₄=x₁x₂x₃=1110*

*P₆=x₂x₃=0110*

*P₁₃=x₁x₂x₄=1101*

*P₁₅=x₁x₂x₃x₄=1111*

Forma canonică disjunctivă(FCD). Forma canonică conjunctivă(FCC)

Se iau toate n-uplele pe care funcţia le aplică în 1 respectiv în 0, obţinându-se tremenii canonici conjunctivi(TCC) respectivi termenii canonici disjunctivi(TCD).Termenii obţinuţi se reunesc sub operaţia de disjuncţie pentru forma canonică disjuctivă FCD respectiv conjuncţie pentru canonică conjunctivă FCC.

*( = + + + + + + + = + + ++ + + + = )*

*( = (+++) (+++) =∙=*

*( + + + + + + = + + + + + + = )*

*((+++) ∙ (+++) =∙=*

*( + + + + + + + = + + + + + + + = )*

*( (+++) = =*

*+ + + + + + = + + + + + + = )*

*(+++) (+++) ∙ (+++ =∙ ∙ =*

*+ + + = + + +=*

*(+++) ∙ (+++) ∙ (+++ ∙ (+++) ∙ (+++ ∙ (+++) = ∙ ∙ ∙ ∙ =*

*+ + + + + = + + + + + = )*

*(+++) ∙ (+++) ∙ (+++) ∙ (+++ = ∙ ∙ =*

*+ + + + + + = + + + + + + = )*

*( +++ ∙ (+++) ∙ (+++ = ∙ =*

Obţinerea formelor minime(FMD,FMC) cu metoda diagramei 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).Se completează cu 1 logic locaţiile corespunzătoare termenilor canonici conjunctivi pentru care funcţia ia valoarea 1 şi cu 0 în rest iar termenii pentru care funcţia nu este definită(combinaţiile indiferente) se marchează cu \* sau d.

Pentru obţinerea formelor minime (FMD şi FMC) se formează subcuburi de dimensiuni cât mai mari şi se scriu termenii corespunzatori subcuburilor obţinute eliminându-se variabilele redundante.

Termenii minimi conjunctivi(TMC) obţinuţi se reunesc sub operaţia de disjuncţie rezultând astfel forma minimă disjunctivă(FMD) iar pentru forma minimă conjunctivă(FMC) se reunesc sub operaţia de conjuncţie termenii minimi disjunctivi(TMD).

Diagrama Krnaugh pentru funcţia **a**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

1

0

1

\*

1

1

1

1

\*

\*

*() = + +*

Diagrama Krnaugh pentru funcţia **b**

00

01

11

10

00

01

11

10

1

1

1

\*

\*

\*

1

1

1

\*

1

0

0

1

\*

\*

*() = + +*

Diagrama Krnaugh pentru funcţia **c**

00

01

11

10

00

01

11

10

1

1

0

\*

\*

\*

1

1

1

\*

1

1

1

1

\*

\*

*() = + +*

Diagrama Krnaugh pentru funcţia **d**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

0

0

1

\*

1

1

1

1

\*

\*

*() = + ++*

Diagrama Krnaugh pentru funcţia **e**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

0

0

0

\*

1

1

0

0

\*

\*

*() =+*

Diagrama Krnaugh pentru funcţia **f**

00

01

11

10

00

01

11

10

1

0

0

\*

\*

\*

0

1

0

\*

1

1

1

1

\*

\*

*() =+*

Diagrama Krnaugh pentru funcţia **g**

00

01

11

10

00

01

11

10

0

0

1

\*

\*

\*

0

1

1

\*

1

1

1

1

\*

\*

*() = + +*

Diagrama Krnaugh pentru funcţia **a**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

1

0

1

\*

1

1

1

1

\*

\*

*() =( + ) ∙ ( + + )*

Diagrama Krnaugh pentru funcţia **b**

00

01

11

10

00

01

11

10

1

1

1

\*

\*

\*

1

1

1

\*

1

0

0

1

\*

\*

*() = + + )*

Diagrama Krnaugh pentru funcţia **c**

00

01

11

10

00

01

11

10

1

1

0

\*

\*

\*

1

1

1

\*

1

1

1

1

\*

\*

*() =( + + )*

Diagrama Krnaugh pentru funcţia **d**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

0

0

1

\*

1

1

1

1

\*

\*

*() = ++ ) ∙ ()*

Diagrama Krnaugh pentru funcţia **e**

00

01

11

10

00

01

11

10

1

0

1

\*

\*

\*

0

0

0

\*

1

1

0

0

\*

\*

*() =*

Diagrama Krnaugh pentru funcţia **f**

00

01

11

10

00

01

11

10

1

0

0

\*

\*

\*

0

1

0

\*

1

1

1

1

\*

\*

*() =*

Diagrama Krnaugh pentru funcţia **g**

00

01

11

10

00

01

11

10

0

0

1

\*

\*

\*

0

1

1

\*

1

1

1

1

\*

\*

*() = (+ + )*

Minimizarea funcţiilor **e** şi **c** utilizând metoda Quine-McCluskey

Această metodă porneşte de la forma canonică a funcţiei de minimizat şi este deasemenea o metodă analitică. Metoda are două etape:

- se determină implicanţii primi;

- se selectează dintre implicanţii primi obţinuţi doar aceia care acoperă total termenii canonici ai funcţiei date şi asigură realizarea acesteia la un preţ de 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. *( + + + + + + + = + + + + + + + = )*

→ → 0000 → → 1001

→ → 0001 → → 1010

→ → 0011 → → 1011

→ → 0100 → → 1100

\*→ → 0101 \*→ → 1101

\*→ → 0110 \*→ → 1110

\*→ → 0111 \*→ → 1111

→ → 1000

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 4 | 0100 |
| 8 | 1000 |
| 2 | 3 | 0011 |
| \*5 | 0101 |
| \*6 | 0110 |
| 9 | 1001 |
| 10 | 1010 |
| 12 | 1100 |
| 3 | \*7 | 0111 |
| 11 | 1011 |
| \*13 | 1101 |
| \*14 | 1110 |
| 4 | \*15 | 1111 |

\*Combinaţiile indiferente (5, 6, 7, 13, 14 şi 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. Se trece la al doilea ciclu de comparaţii.

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0,1 | 000- |
| 0,4 | 0-00 |
| 0.8 | -000 |
| 1 | 1,3 | 00-1 |
| 1,5 | 0-01 |
| 1,9 | -001 |
| 4,5 | 010- |
| 4,6 | 01-0 |
| 4,12 | 100- |
| 8,9 | 10-0 |
| 8,10 | 10-0 |
| 8,12 | 1-00 |
| 2 | 3,11 | -011 |
| 3,7 | 0-11 |
| 9,11 | 10-1 |
| 9,13 | 1-01 |
| 10,11 | 101- |
| 10,14 | 1-10 |
| 12,13 | 110- |
| 12,14 | 11-0 |
| 3 | 11,15 | 1-11 |

Al treilea ciclu de comparaţii:

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0,1cu 4,5 | 0-0- |
| 0,1 cu 8,9 | -00- |
| 0,4 cu 8,12 | --00 |
| 1 | 1,3 cu 9,11 | -0-1 |
| 1,3 cu 5,7 | 0--1 |
| 1,5 cu 9,13 | --01 |
| 4,5 cu 12,13 | -10- |
| 4,6 cu 12,14 | -1-0 |
| 8,9 cu 10,11 | 10-- |
| 8,9 cu 12,13 | 1--0 |
| 8,10 cu 12,14 | 1--0 |
| 2 | 3,7 cu11,15 | --11 |
| 9,11 cu 13,15 | 1--1 |
| 10,11 cu14,15 | 1-1- |

Al patrulea ciclu de comparaţii

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

Nu se mai pot face alte comparaţii.

Etapa a II-a este necesară deoarece nimeni nu ne poate asigura dacă forma obţinută nu mai poate fi minimizată. Pentru aceasta se face tabelul implicanţilor primi:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\IP |  |  |  |  |  |
|  |  | \* |  |  |  |
|  |  | \* | \* | \* |  |
|  |  |  | \* | \* |  |
|  |  | \* |  |  |  |
|  |  | \* |  |  | \* |
|  | \* | \* | \* | \* | \* |
|  | \* |  |  |  | \* |
|  | \* |  | \* | \* | \* |
|  | \* | \* |  |  |  |

Tabelul implicanţilor primi neesenţiali

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC\IP |  |  |  |  |
|  |  | \* | \* |  |
|  | \* |  |  | \* |
|  | \* | \* | \* | \* |

*() = + +*

1. *+ + + = + + +=*

→ → 0000 → → 1001

→ → 0010 → → 1011

\*→ → 0101 \*→ → 1101

\*→ → 0110 \*→ → 1110

\*→ → 0111 \*→ → 1111

Primul ciclu de comparaţii

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0 | 0000 |
| 1 | 2 | 0010 |
| 2 | 5\* | 0101 |
| 6\* | 0110 |
| 9 | 1001 |
| 3 | 11 | 1011 |
|  | 7\* | 0111 |
|  | 13\* | 1101 |
|  | 14\* | 1110 |
| 4 | 15\* | 1111 |

Al doilea ciclu de comparaţii

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0,2 | 00-0 |
| 1 | 2,6 | 0-10 |
| 2 | 9,11 | 10-1 |
| 9,13 | 1-10 |
| 3 | 11,15 | 1-11 |

Al treilea ciclu de comparaţii

|  |  |  |
| --- | --- | --- |
| Grupa | Indicii termenilor |  |
| 0 | 0,2 | 00-0 |
| 1 | 2,6 | 0-10 |
| 2 | 9,11 | 10-1 |
| 9,11,13,15 | 1—1 |

Tabelul acoperirilor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC\IP |  |  |  |  |
|  | \* |  |  |  |
|  | \* | \* |  |  |
|  |  |  | \* | \* |
|  |  |  | \* | \* |

Tabelul implicanţilor primi neesentiali

|  |  |  |
| --- | --- | --- |
| TC\IP |  |  |
|  | \* | \* |
|  | \* | \* |

*() =+*

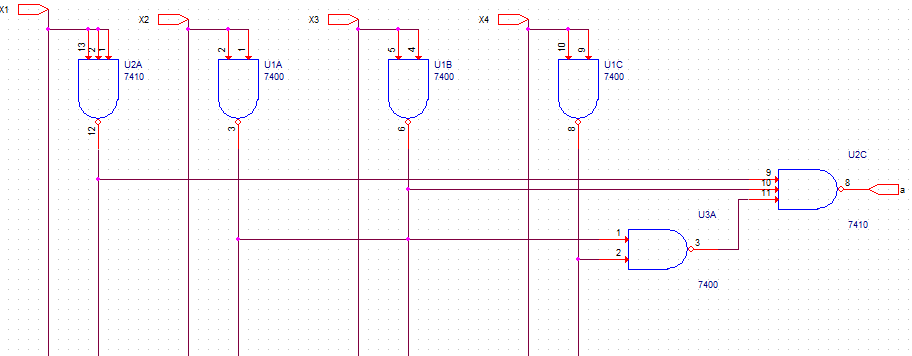
Implementare funcţiilor cu porţi ŞI-NU în tehnologia TTL

Pentru a implementa funcţiile logice cu porţi logice ŞI-NU acestea trebuiesc aduse la o formă adecvată,una dintre formele minime ale funcţiei,(FMD) . Procedeul care se va folosi este negarea de două ori a funcţiei şi folosirea formulelor lui De Morgan. Se vor folosi circuite integrate din producţia Texas Instruments din seria 74LS.

Implementarea funcţiei a

*() = + +*

*() = = + + = ∙ ∙*



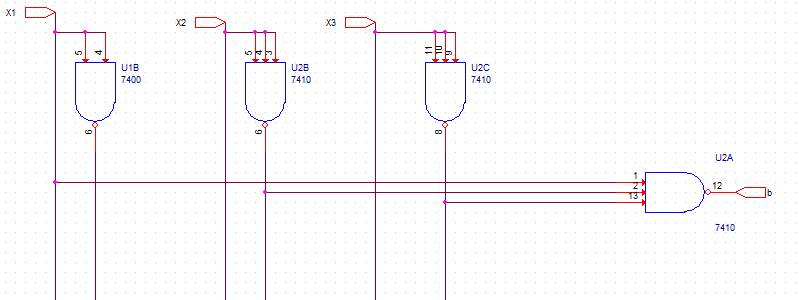
CI folosite : 1×7400(-)

1×7410(-)

Implementarea funcţiei b

*() = + +*

*() = = + + = ∙ ∙*

**

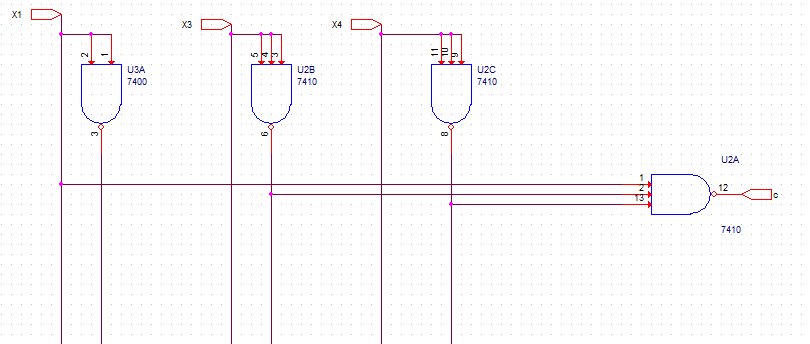
CI folosite : 1×7400(- 3)

1×7410(-)

Implementarea funcţiei c

*() = + +*

*() = = + + = ∙ ∙*

**

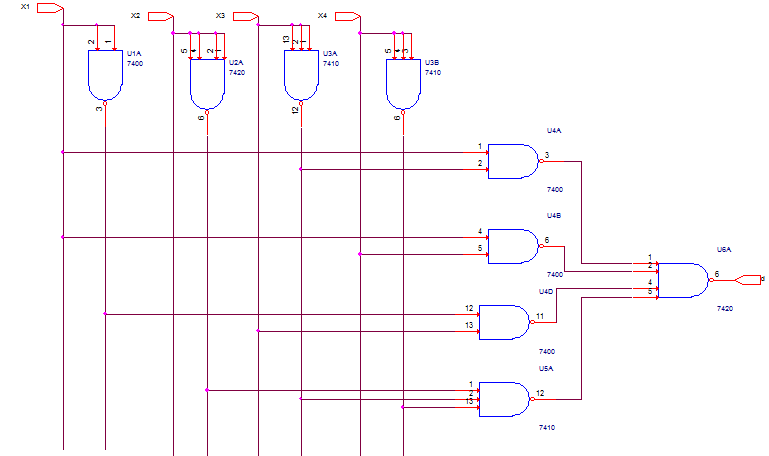
CI folosite : 1×7400(- 3)

1×7410(-)

Implementarea funcţiei d

*() = + ++*

*() = = + ++= ∙ ∙ ∙*



CI folosite : 1×7400(- 3)

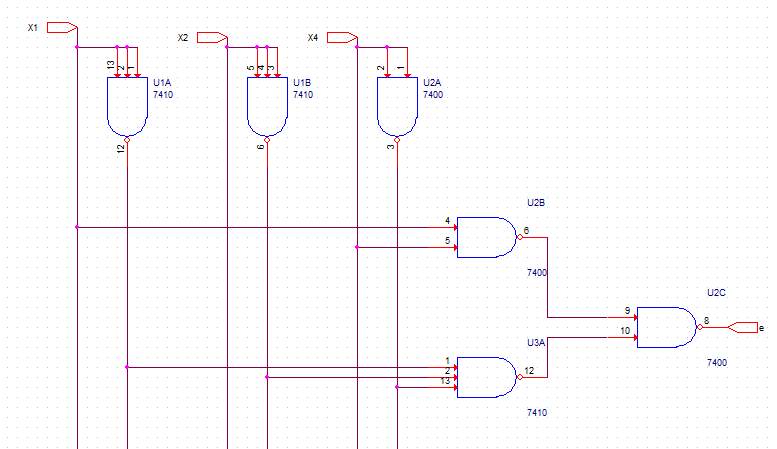
1×7410(-)

1×7420(-)

Implementarea funcţiei e

*() =+*

*() =+ = ∙*

**

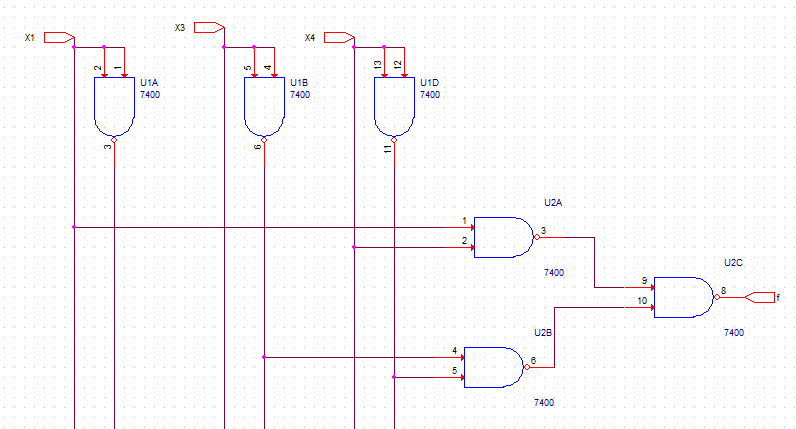
CI folosite : 2×7400(- )

1×7410(-1)

Implementarea funcţiei f

*() =+*

*() = =+ =∙*

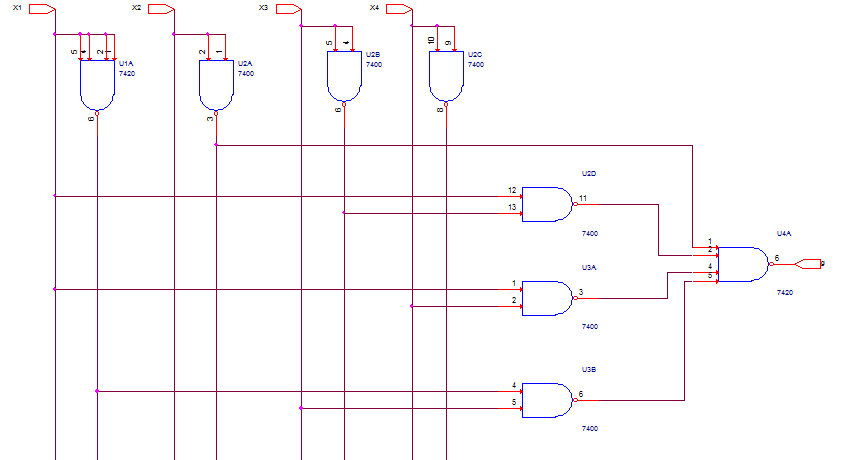


CI folosite : 2×7400(- )

Implementarea funcţiei g

*() = + +*

*() = + + = ∙ ∙*



CI folosite : 2×7400(- )

1×7420(-)

Implementarea ansamblului funcţiilor cu porţi ŞI-NU în tehnologie TTL

Pentru implementarea ansamblului funcţiilor in tehnologia TTL s-au folosit şapte CI, dintre care: patru de tipul 74LS00 (utilizat în proporţie de 100%), două de tipul 74LS10 (o poartă neutilizată) şi unul de tipul 74LS20 (utilizat în proporţie de 100%).



Implementarea schemei cu primele patru funcţii cu porţi ŞI-NU în tehnologie TTL iar următoarele trei funcţii cu porţi SAU-NU în tehnologie CMOS

*() = = + + = ∙ ∙*

*() = = + + = ∙ ∙*

*() = = + + = ∙ ∙*

*() = = + ++= ∙ ∙ ∙*

*() =+ = + + ++*

*() = =+ = + + +*

*() = + + + ++*



Implementarea cu multiplexoare MUX

Multiplexorul (MUX-ul) este unul dintre cele mai importante circuite MSI. Este denumit

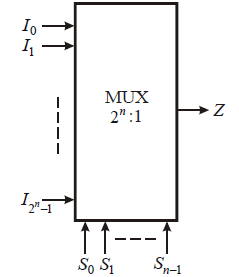
uneori şi „selector“ deoarece este utilizat şi ca un comutator de selectare a anumitor căi.

Multiplexorul/selectorul este un circuit logic combinaţional care are, în cazul general,

2*n* intrări de date, ……*I n*− 1, *n* intrări de selecţie (adresă), *S*0 , *S*1…*Sn*−1 şi o

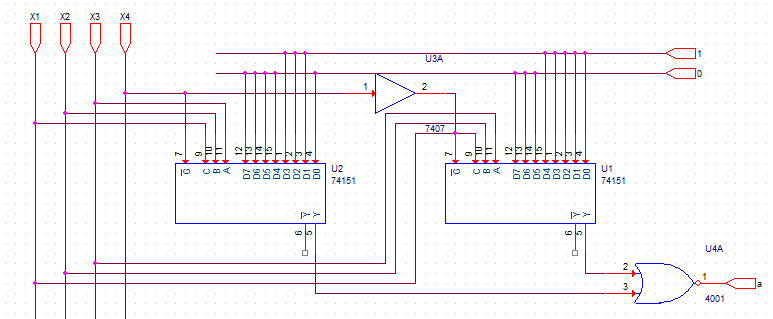
ieşire *Z* . Starea ieşirii circuitului la un moment dat este aceeaşi cu starea intrării *Ik* , unde indicele *k* este echivalentul zecimal al numărului binar reprezentat de stările 0 şi 1 ale intrărilor de selecţie: *k* = (*Sn*−1*Sn*−2….*S*1*S*0 ) .

Ieşirea selectează deci intrarea indicată de variabilele de adresă.

**

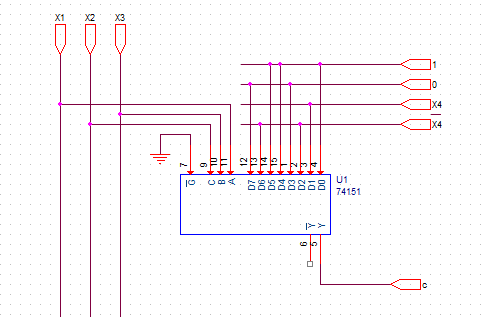
Implementarea cu MUX de 3 biţi sau 8 căi a funcţiei **a**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echivalent zecimal | x₁ | x₂ | x₃ | x₄ | a |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | \* |
| 6 | 0 | 1 | 1 | 0 | \* |
| 7 | 0 | 1 | 1 | 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 |
| 13 | 1 | 1 | 0 | 1 | \* |
| 14 | 1 | 1 | 1 | 0 | \* |
| 15 | 1 | 1 | 1 | 1 | \* |



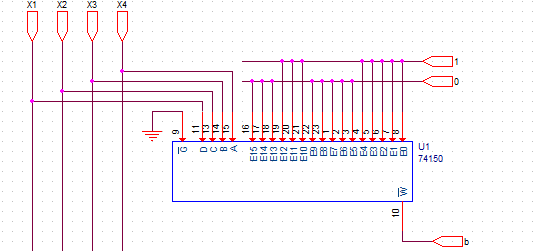
Implementarea cu MUX de 3 biţi sau 8 căi a funcţiei **c**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echivalent zecimal | x₁ | x₂ | x₃ | x₄ | c |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | \* |
| 6 | 0 | 1 | 1 | 0 | \* |
| 7 | 0 | 1 | 1 | 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 |
| 13 | 1 | 1 | 0 | 1 | \* |
| 14 | 1 | 1 | 1 | 0 | \* |
| 15 | 1 | 1 | 1 | 1 | \* |



Implementarea cu MUX de 4 biţi sau 16 căi a funcţiei **b**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echivalent zecimal | x₁ | x₂ | x₃ | x₄ | b |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | \* |
| 6 | 0 | 1 | 1 | 0 | \* |
| 7 | 0 | 1 | 1 | 1 | \* |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | \* |
| 14 | 1 | 1 | 1 | 0 | \* |
| 15 | 1 | 1 | 1 | 1 | \* |



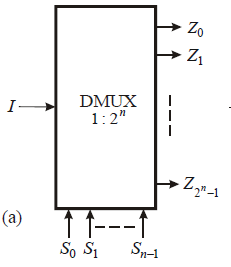
Implementarea cu demultiplexoare DMUX

Circuitele demultiplexoare (DMUX-urile) realizează operaţia inversă multiplexării şi

anume distribuie un semnal (0 sau 1) de pe o cale, pe mai multe căi, în funcţie de adresa

acelor căi, motiv pentru care se mai numesc şi circuite distribuitoare .

Demultiplexorul este un circuit logic combinaţional care are, în cazul general, o intrare de date *I*, *n* intrări de selecţie (adresă), *S*0 , *S*1,.., *Sn*−1 şi ieşiri,….



Orice funcţie logică de *n* variabile, dată sub FCD, se poate implementa cu ajutorul unui

DCD cu *n* intrări sau a unui DMUX 1: 2*n* , dacă la ieşirea acestor circuite termenii

canonici (nivelul logic ŞI), existenţi în tabelul de adevăr al funcţiei sunt introduşi într-o

poartă logică pentru realizarea nivelului logic SAU.

Implemrntarea cu DMUX de 8 căi şi porţi logice ŞI−NU pentru funcţia **e**

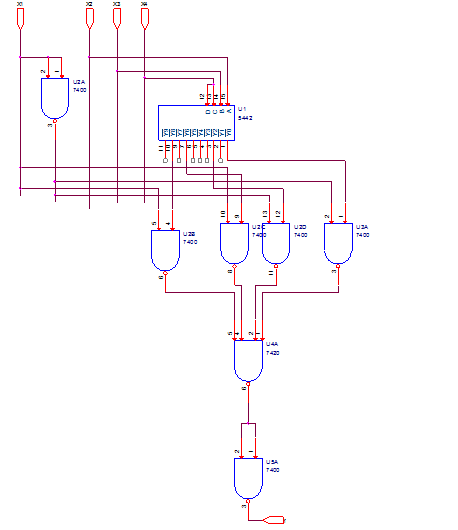
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Echivalent zecimal | x₁ | x₂ | x₃ | x₄ | e |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | \* |
| 6 | 0 | 1 | 1 | 0 | \* |
| 7 | 0 | 1 | 1 | 1 | \* |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | \* |
| 14 | 1 | 1 | 1 | 0 | \* |
| 15 | 1 | 1 | 1 | 1 | \* |

|  |
| --- |
|  |

*+ + + = + + +=*

*)+)+ )+) = + ++=*

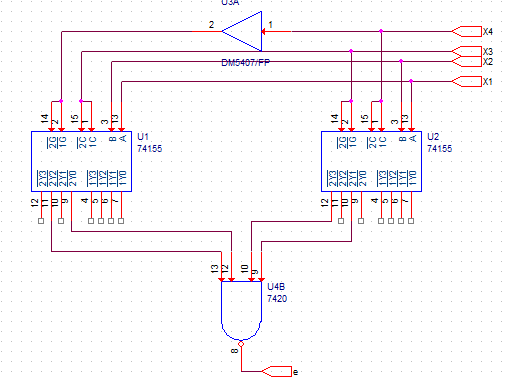
*+ ++ = ∙ ∙ ∙*

**

Implementarea cu DMUX de 8 căi şi porţi logice ŞI−NU a funcţiei **e**

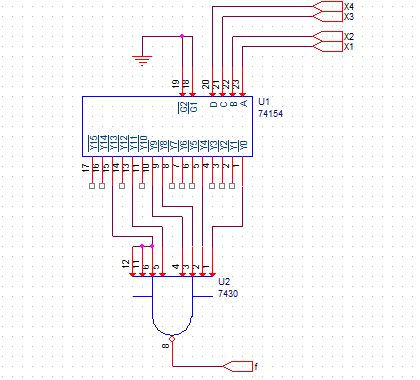
*+ + + = + + +=*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x₁ | x₂ | x₃ | x₄ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |



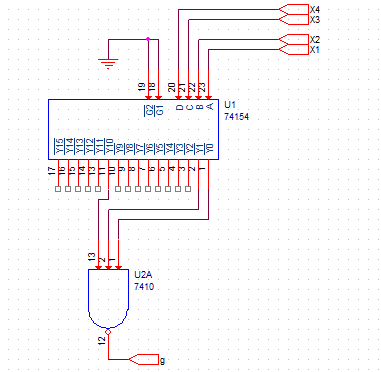
Implementarea cu DMUX de 16 căi şi porţi logice ŞI−NU a funcţiei **f**

*+ + + + + = + + + + + = )*

**

Implementarea cu DMUX de 16 căi şi porţi logice ŞI a funcţiei **g**

*+ + + + + + = + + + + + + = )*

****

Date de catalog ale circuitelor integrate folosite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C.I. | Tehnologie | [ns] | [ns] | [mA] | [mA] | [mW] |
| 74LS00 | TTL | 11 | 7 | 4 | 12 |  |
| 74LS10 | TTL | 11 | 7 | 3 | 9 |  |
| 74LS20 | TTL | 12 | 8 | 2 | 6 |  |
| 74LS30 | TTL | 12 | 15 | 3,5 | 6 |  |
| 74LS150 | TTL | 23 | 22 |  |  | 200 |
| 74LS151 | TTL | 17 | 19 |  |  | 145 |
| 74HC154 | CMOS | 38 | 38 |  |  | 750 |
| CD4001 | CMOS | 110 | 120 | 0,88 | -0,7 |  |
| CD4002 | CMOS | 125 | 125 | 1 | -0,8 |  |
| CD4025 | CMOS | 120 | 130 | 0,88 | -0,7 |  |

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

Timpii de propagare se vor calcula după formula:



Timpul de propagare pentru schema cu porţi ŞI-NU în tehnologie TTL



Dintre timpii de propagare al funcţiilor îl alegem pe cel mai mare, 37ns. Deci, 

Analog se procedează şi pentru restul schemelor. Vom preciza în continuare doar funcţia cu latenţa cea mai mare.

Timpul de propagare pentru schema cu porţi TTL şi CMOS



Timpul de propagare pentru schema cu MUX-uri de 8 căi TTL



Timpul de propagare pentru schema cu MUX-uri de 16 căi TTL



Calculul puterilor disipate

Se vor calcula puterile disipate pentru un circuit integrat cu formula: 

Puterea disipară pentru schema cu porţi ŞI-NU în tehnologie TTL



Puterea disipată pentru schema cu porţi TTL şi CMOS



Puterea disipată pentru schema cu MUX-uri de 16 căi



Puterea disipată pentru schema cu DMUX-uri de 8 căi



Încapsularea porţilor în circuitele integrate folosite

** **

** **





Implementarea ansamblului funcţiilor cu circuite integrate TTL

3

14

8

9

2

12

13

2

a

x2

74LS20

4

2

9

2

U5

12

8

74LS00

4

2

14

b

9

g

9

11

9

8

74LS20

14

4

10

7

7

U8

U7

1

1

2

1

74LS30

U4

e

Vcc

8

13

11

13

5

4

5

Vcc

3

10

Vcc

3

8

U1

11

3

1

U2

6

11

8

5

Vcc

12

13

Vcc

7

5

1

14

74LS10

Vcc

2

6

d

10

7

7

12

U3

74LS00

x4

11

11

4

74LS00

5

1

4

8

10

7

5

12

6

6

11

3

9

11

3

12

9

74LS00

10

Vcc

6

14

3

5

1

4

1

2

Vcc

5

8

U6

12

14

f

10

10

6

7

14

13

x3

4

13

13

6

3

14

12

10

13

c

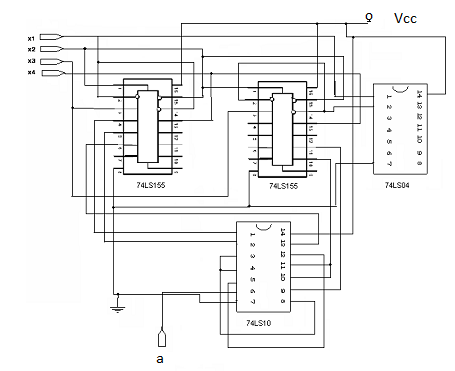
6

9

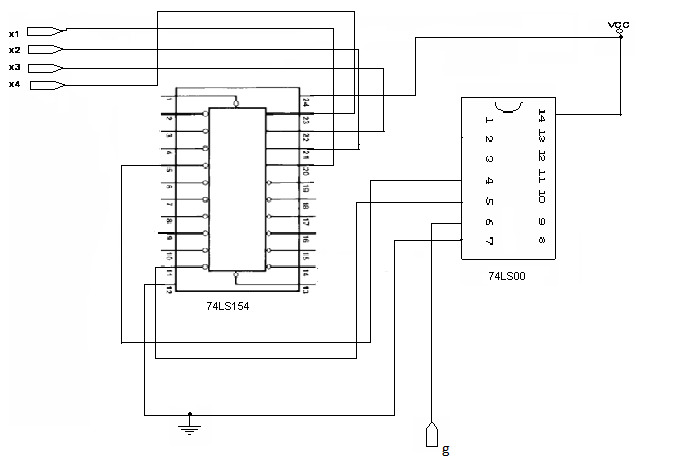
x1

7

Implementarea cu CI a funcţiei cu MUX de 8 căi



Implementarea cu CI a funcţiei cu DMUX de 16 căi



Concluzii finale

Putem compara implementările obţinute din mai multe puncte de vedere.

Din punctul de vedere al complexităţii implementării cea mai puţin complexă este schema cu DMUX-uri de 16 căi şi porţi ŞI în tehnologie CMOS şi probabil şi cea mai ieftină. La capitolul timp de propagare, schema cu MUX-uri de 16 căi TTL este cea mai rapidă.

În schimb din punctul de vedere al consumul de energie, schemele cu MUX-uri sunt mari consumatoare de putere. Cea mai mică consumatoare este implementarea cu circuite 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 date

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