 UNIVERSITATEA “TRANSILVANIA” BRAŞOV

Facultatea de Inginerie Electrică si Știința Calculatoarelor

Catedra de Automatică Și Informatică Aplicată

**PROIECT ASCN**

***Îndrumător: Student:***

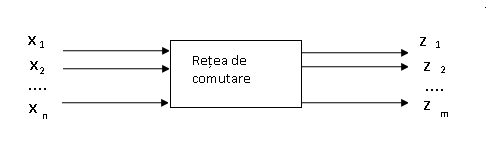
***Prof. dr. ing. Moldoveanu Florin*** Ciobotaru Cristina

Grupa 4492

**Introducere**

Un circuit logic combinaţional (CLC) este un circuit de comutare care se caracterizează prin aceea că starea ieşirilor sale la un moment dat depinde numai de starea intrărilor sale la momentul considerat. Legătura între starea intrărilor şi starea ieşirilor circuitului este dată de funcţiile de transfer ale acestuia.

Schema bloc generală al unui CLC reprezentată printr-o rețea de comutare sau schema logica este următoarea:



x1,x2,...,xn -mărimi de intrare

z1,z2,...,zm -mărimi de iesire

Relaţii generale între aceste valori:

f(xi, …, xn) fi – functii logice

z1 = ……………. xi– variabile logice

zm = f(xi, …, xm)

Prin analiza unui CLC se înțelege determinarea expresiilor mărimilor de ieșire z1,z2,...,zm în funcție de mărimile de intrare x1,x2,...,xn si cunoscând configurația (numărul si tipul de elemente logice care intră în componență, modul de conectare, punctul în care se aplică variabilele de intrare, numărul de nivele logice etc.) si componența schemei logice.

Analiza rețelelor CLC realizate cu elemente logice de tip inversor (ŞI-NU, SAU-NU): numărul maxim de elemente logice aflate între intrarea si ieşirea rețelei determină numărul de nivele logice ale acestuia. Numerotarea lor se face de la ieșire către intrare. Un element logic ȘI-NU respectiv SAU-NU realizează operația SAU respectiv ȘI asupra variabilelor de intrare complementate dacă se află pe un nivel de inversare impar si operația ȘI respectiv SAU asupra variabilelor de intrare necomplementate dacă se află pe un nivel de inversoare impar.

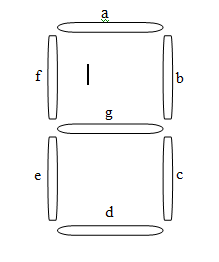
Majoritatea sistemelor numerice necesită decodificarea informațiilor reprezentate prin cod mașină.Multe aplicații (de exemplu: multiplexarea datelor, multiplicarea proporțională, afișarea numerică, convertorul numeric analogic, adresarea memoriei) necesită decodificarea informației.

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 de cod şi *m* ieşiri (*m* ≤ 2n). 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.

Codul *BCD* (Binary Coded Decimal) asociază fiecarei cifre zecimale o secvență de cod binar cu *4* respectiv *5* cifre.Pentru reprezentarea numerelor in cod BCD se concatenează secvențele binare corespunzătoare cifrelor din codurile BCD respective.

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.

Dispunerea segmentelor unui decodificator BCD-7 *segmente* este următoarea:



Codul „*2 din 5*” se utilizează pentru reprezentarea numerelor zecimale printr-un grup de 5 biţi.Caracteristica principală constă in faptul că toate secvențele binare asociate cifrelor zecimale au *2*  biți semnificativi.Utilizarea unui astfel de cod creează posibilitatea controlului asupra transmisiei informației codificate in acest cod.

# TEMĂ PROIECT Nr.16

Să se proiecteze un decodificator BCD neponderat „2 din 5” / 7 segmente (logică com-binaţională).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 di-agrame Karnaugh.

b) Să se obţină formele minime disjunctive şi conjunctive pentru funcţiile logice asociate decodificatorului BCD, „2 din 5” / 7 segmente (utilizând combinaţiile indiferente) prin metoda diagramelor Karnaugh; de asemenea se vor obţine formele minime disjunctive pentru două funcţii logice de ieşire f4 si 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 3 funcţii logice de ieşire cu porţi logice ȘI-NU, realizate în tehnologia TTL, iar următoarele 4 cu porţi logice SAU-NU, realizate în tehnologia CMOS.

f) Să se implementeze ansamblul funcţiilor logice cu MUX-uri de 4 respectiv 16 căi (circuitele sunt realizate în tehnologia TTL).

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 CMOS).

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.

**Tabel de adevăr:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Echiv. Zec.** | **Nr.**  **Crt** | **Cod „*2 din 5”*** | | | | | **Ieșiri 7 *segmente*** | | | | | | |
| **x1** | **x2** | **x3** | **x4** | **x5** | **fa** | **fb** | **fc** | **fd** | **fe** | **ff** | **fg** |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 6 | 2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 9 | 3 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 10 | 4 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 12 | 5 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 17 | 6 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 18 | 7 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 20 | 8 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 24 | 9 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Combinatii indiferente: P0=P1=P2=P4=P7=P8=P11=P13=P14=P15=P16=P19=P21=P22=P23=P25=P26=P27=P28=P29=P30=P31=\*

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

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

* forma canonică conjunctivă (FCC), unde termenii funcției sunt legți între ei prin ŞI, iar variabilele din cadrul unui termen sunt legate prin SAU;
* forma canonică disjunctivă (FCD), unde termenii funcției sunt legați între ei prin SAU, iar variabilele din cadrul unui termen sunt legate prin ŞI.

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

**Formele canonice disjunctive**

Pentru obținea formelor canonice disjunctive din tabelul de adevăr se iau in considerare combinațiile pentru care funcția are valoarea 1.





**Formele canonice conjunctive**

Pentru obținerea formelor canonice conjunctive din tabelul de adevăr se iau in considerare combinațiile pentru care functia are valoarea 0.







**b) Să se obţină formele minime disjunctive şi conjunctive pentru funcţiile logice asociate decodificatorului BCD, „2 din 5” / 7 segmente (utilizând combinaţiile indiferente) prin metoda diagramelor Karnaugh; de asemenea se vor obţine formele minime disjunctive pentru două funcţii logice de ieşire f4 si prin metoda Quine-McCluskey.**

**Diagrame Karnaugh**

Metoda diagramelor Karnaugh este o metodă grafo-analitică care pornește de la o formă canonică a funcției.Diagrama se prezinta sub forma unui pătrat sau dreptunghi având 2n locații.

În fiecare locație se va plasa un termen canonic așa incât in diagramă se vor găsi toți termenii canonici posibili.

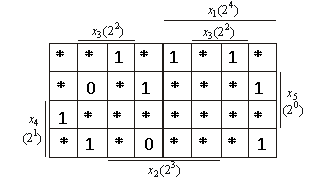
Diagrama K este construită astfel incât între două compartimente vecine pe linie sau pe coloană conțin termeni canonici care au proprietatea de adiacență.De asemenea se consideră vecine si au aceeași proprietate si locațiile situate la extremitățile liniilor si coloanelor.

Diagramele K pentru mai mult de 4 variabile se obțin prin alăturarea pe linii respectiv coloane a unei variabile de 4 coloane considerate diagrame elementare.

În cazul nostru diagrama K are forma unui dreptunghi cu 32 locații.



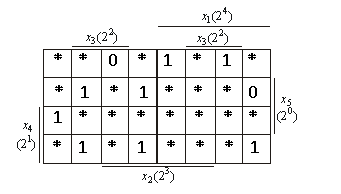
Fa







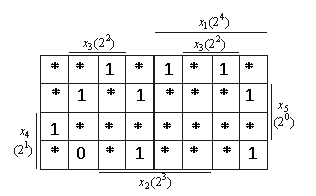
Fb







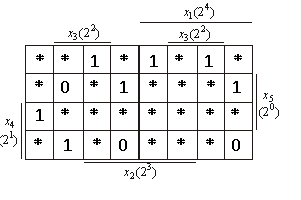
Fc







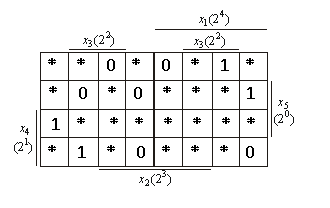
Fd







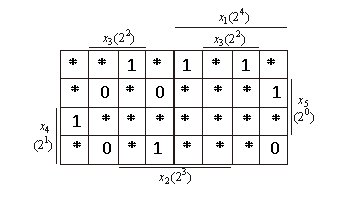
Fe







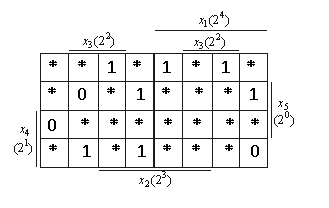
Ff







Fg







**Forma minimă disjunctivă prin metoda Quine-McCluskey**

Aceasta metodă pornește de la forma canonică a funcției de minimizat.

Metoda are două etape:

* + se determină implicanții primi;
  + se selectează dintre implicanții primi obținuți doar aceia care acoperă toți termenii canonici ai funcției date si asigură realizarea acesteia la un cost minim.

Termenii canonici se compară în felul următor:

* + se compară fiecare termen canonic cu toți ceilalți;
  + când se găsesc doi termeni care au proprietatea de adiacență, variabila redundantă se elimina, obținându-se un termen elementar;
  + primul ciclu de comparații se consideră încheiat în momentul în care s-au comparat între ei toți termenii canonici, obținându-se toți implicanții primi posibili;
  + se compară între ei pe același criteriu termenii elementari obținuți;
  + se vor face atatea cicluri de comparație câte sunt necesare, pentru a nu mai exista termeni elementari cu proprietatea de adiacență.

Combinații indiferente se vor compara cu celelalte, dar nu se vor compara între ele.

Termenii elementari rămași necomparaţi în diverse etape sunt implicanți primi.

Selectarea implicanților primi care intră în FMD se face în felul următor:

* + la intersecția unei linii cu o coloană se va plasa un \* care va indica faptul că implicantul prim de pe linia respectivă va include termenul canonic de pe linia corespunzătoare.
  + dacă pe o linie apare un singur \*, implicantul prim de pe acea linie se numește implicant prim esential si va apărea în forma minimă disjunctivă în mod obligatoriu.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Grupa | Indici | X1 | X2 | X3 | X4 | X5 |  |
| 0 | 0\* | 0 | 0 | 0 | 0 | 0 | ✓ |
| 1 | 1\* | 0 | 0 | 0 | 0 | 1 | ✓ |
| 2\* | 0 | 0 | 0 | 1 | 0 | ✓ |
| 4\* | 0 | 0 | 1 | 0 | 0 | ✓ |
| 8\* | 0 | 1 | 0 | 0 | 0 | ✓ |
| 16\* | 1 | 0 | 0 | 0 | 0 | ✓ |
| 2 | 3 | 0 | 0 | 0 | 1 | 1 | ✓ |
| 6 | 0 | 0 | 1 | 1 | 0 | ✓ |
| 9 | 0 | 1 | 0 | 0 | 1 | ✓ |
| 12 | 0 | 1 | 1 | 0 | 0 | ✓ |
| 17 | 1 | 0 | 0 | 0 | 1 | ✓ |
| 20 | 1 | 0 | 1 | 0 | 0 | ✓ |
| 24 | 1 | 1 | 0 | 0 | 0 | ✓ |
| 3 | 7\* | 0 | 0 | 1 | 1 | 1 | ✓ |
| 11\* | 0 | 1 | 0 | 1 | 1 | ✓ |
| 13\* | 0 | 1 | 1 | 0 | 1 | ✓ |
| 14\* | 0 | 1 | 1 | 1 | 0 | ✓ |
| 19\* | 1 | 0 | 0 | 1 | 1 | ✓ |
| 21\* | 1 | 0 | 1 | 0 | 1 | ✓ |
| 22\* | 1 | 0 | 1 | 1 | 0 | ✓ |
| 25\* | 1 | 1 | 0 | 0 | 1 | ✓ |
| 26\* | 1 | 1 | 0 | 1 | 0 | ✓ |
| 28\* | 1 | 1 | 1 | 0 | 0 | ✓ |
| 4 | 15\* | 0 | 1 | 1 | 1 | 1 | ✓ |
| 23\* | 1 | 0 | 1 | 1 | 1 | ✓ |
| 27\* | 1 | 1 | 0 | 1 | 1 | ✓ |
| 29\* | 1 | 1 | 1 | 0 | 1 | ✓ |
| 30\* | 1 | 1 | 1 | 1 | 0 | ✓ |
| 5 | 31\* | 1 | 1 | 1 | 1 | 1 | ✓ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Grupa | Indici | X1 | X2 | X3 | X4 | X5 |  |
| 1 | 1\*,3 | 0 | 0 | 0 | - | 1 | ✓ |
| 1\*,9 | 0 | - | 0 | 0 | 1 | ✓ |
| 1\*,17 | - | 0 | 0 | 0 | 1 | ✓ |
| 2\*,3 | 0 | 0 | 0 | 1 | - | ✓ |
| 2\*,6 | 0 | 0 | - | 1 | 0 | ✓ |
| 4\*,6 | 0 | 0 | 1 | - | 0 | ✓ |
| 4\*,12 | 0 | - | 1 | 0 | 0 | ✓ |
| 4\*,20 | - | 0 | 1 | 0 | 0 | ✓ |
| 8\*,9 | 0 | 1 | 0 | 0 | - | ✓ |
| 8\*,12 | 0 | 1 | - | 0 | 0 | ✓ |
| 8\*,24 | - | 1 | 0 | 0 | 0 | ✓ |
| 16\*,17 | 1 | 0 | 0 | 0 | - | ✓ |
| 16\*,20 | 1 | 0 | - | 0 | 0 | ✓ |
| 16\*,24 | 1 | - | 0 | 0 | 0 | ✓ |
| 2 | 3,7\* | 0 | 0 | - | 1 | 1 | ✓ |
| 3,11\* | 0 | - | 0 | 1 | 1 | ✓ |
| 3,19\* | - | 0 | 0 | 1 | 1 | ✓ |
| 6,7\* | 0 | 0 | 1 | 1 | - | ✓ |
| 6,14\* | 0 | - | 1 | 1 | 0 | ✓ |
| 6,22\* | - - | 0 | 1 | 1 | 0 | ✓ |
| 9,11\* | 000d 0 | 1 | 0 | - | 1 1 | ✓ ✓ |
| 9,13\* | 0 | 1 | - | 0 | 1 1 | ✓ ✓ |
| 9,25\* | - | 1 | 0 | 0 | 1 | ✓ |
| 12,13\* | 0 | 1 | 1 | 0 | - | ✓ |
| 12,14\* | 0 | 1 | 1 | - | 0 | ✓ |
| 12,28\* | - | 1 | 1 | 0 | 0 | ✓ |
| 17,19\* | 1 | 0 | 0 | - | 1 | ✓ |
| 17,21\* | 1 | 0 | - | 0 | 1 | ✓ |
| 17,25\* | 1 | - | 0 | 0 | 1 | ✓ |
| 20,21\* | 1 | 0 | 1 | 0 | - | ✓ |
| 20,22\* | 1 | 0 | 1 | - | 0 | ✓ |
| 20,28\* | 1 | - | 1 | 0 | 0 | ✓ |
| 24,25\* | 1 | 1 | 0 | 0 | - | ✓ |
| 24,26\* | 1 | 1 | 0 | - | 0 | IPM |
| 24,28\* | 1 | 1 | - | 0 | 0 | ✓ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Grupa | Indici | X1 | X2 | X3 | X4 | X5 |  |
| 1 | 1\*,3,9,11\* | 0 | - | 0 | - | 1 | ✓ |
| 1\*,3,17,19\* | - | 0 | 0 | - | 1 | ✓ |
| 1\*,9,17,25\* | - | - | 0 | 0 | 1 | ✓ |
| 2\*,3,6,7\* | 0 | 0 | - | 1 | - | ✓ |
| 4\*,6,12,14\* | 0 | - | 1 | - | 0 | ✓ |
| 4\*,6,20,22\* | - | 0 | 1 | - | 0 | ✓ |
| 4\*,12,20,28\* | - | - | 1 | 0 | 0 | ✓ |
| 8\*,9,12,13\* | 0 | 1 | - | 0 | - | ✓ |
| 8\*,9,24,25\* | - | 1 | 0 | 0 | - | ✓ |
| 8\*,12,24,28\* | - | 1 | - | 0 | 0 | ✓ |
| 16\*,17,20,20\* | 1 | 0 | - | 0 | - | ✓ |
| 16\*,17,24,25\* | 1 | - | 0 | 0 | - | ✓ |
| 16\*,20,24,26\* | 1 | - | - | 0 | 0 | ✓ |

Tabelul acoperirilor

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC  IP | 3 | 6 | 9 | 12 | 17 | 20 | 24 |
|  |  |  |  |  |  |  | \* |
|  | \* |  | \* |  |  |  |  |
|  | \* |  |  |  | \* |  |  |
|  |  |  | \* |  | \* |  |  |
|  | \* | \* |  |  |  |  |  |
|  |  | \* |  | \* |  |  |  |
|  |  | \* |  |  |  | \* |  |
|  |  |  |  | \* |  | \* |  |
|  |  |  | \* | \* |  |  |  |
|  |  |  | \* |  |  |  | \* |
|  |  |  |  | \* |  |  | \* |
|  |  |  |  |  | \* | \* |  |
|  |  |  |  |  | \* |  | \* |
|  |  |  |  |  |  | \* | \* |



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

Implemantarea cu porți logice pornește de la una dintre formele minime ale funcției (FMD sau FMC).

Prin implementarea unei funcții booleene sau a unui sistem de funcții booleene se ințelege realizarea acesteia cu circuite integrete adică proiectarea si desenarea schemei logice utilizându-se simbolurile grafice asociate circuitelor integrate si calculul numărului necesar de circuite integrate.

În cazul nostru se vor utiliza circuitele integrate 74LS00 si 74LS10 :

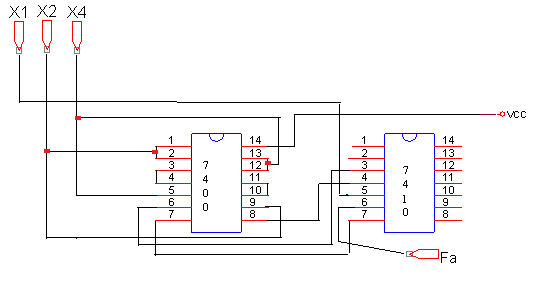
**** ****

Implementarea funcției Fa :





Implementarea funcției Fa cu circuite integrate:



1\*74LS00(-)

1\*74LS10(-1)

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













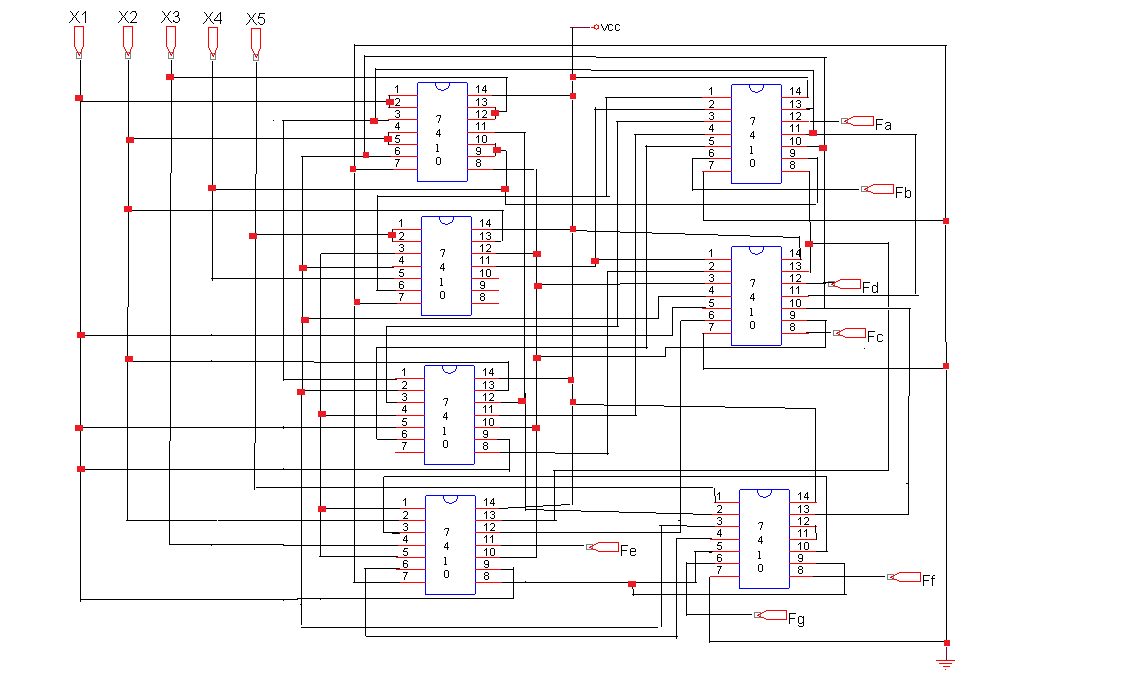


****

4\*74LS00(-)

4\*74LS10(-1)

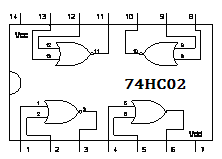
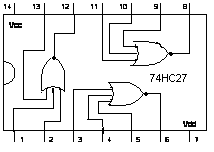
Implementarea ansamblului cu circuite integrate:



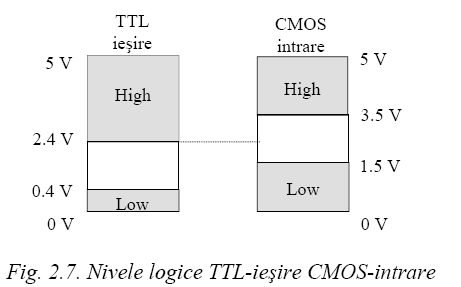
**e) Să se implementeze ansamblul funcţiilor logice în următoarea variantă: primele 3 funcţii logice de ieşire cu porţi logice ȘI-NU, realizate în tehnologia TTL, iar următoarele 4 cu porţi logice SAU-NU, realizate în tehnologia CMOS.**

Se vor folosi circuitele integrate 74LS00 si 74LS10 in tehnologia TTL si 74HC02 si 74HC27 in tehnologia CMOS.

Când circuitul TTL trebuie sa comande circuitele CMOS alimentate dintr-o singură sursă de tensiune de 5V, nivelul minim de ieșire in stare High garantat de TTL(2,4V), este mai mic decat nivelul minim de intrare acceptat de CMOS in stare High(3,5V).



Nivele logice TTL-ieșire CMOS-intare

În acest caz se utilizează o rezistență conectată între ieșire si Vcc pentru a crește nivelul de ieșire in stare High al circuitului TTL.Valoarea minimă a rezistenței este determinată de curentul maxim absorbit pe la ieșirea TTL(16mA pentru TTL standard), iar valoarea maximă este determinată de curentul tranzistorului de iesire in stare blocată. Valorile cele mai indicate pentru aceasta rezistentă se situează in gama de valori 1,5kohmi...4,7kohmi pentru toate familiile TTL.













3\*74LS00(-2)

1\*74LS(-)

3\*74HC02(-1)

3\*74HC27(-2)

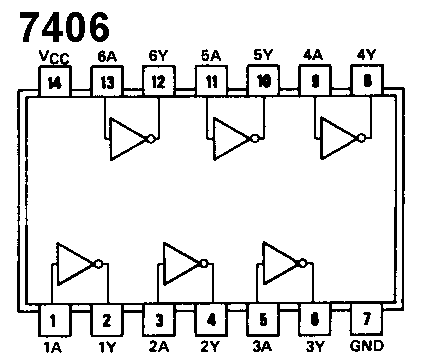
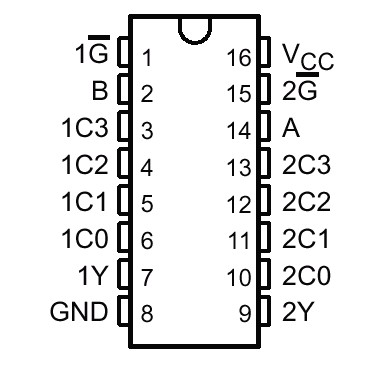
**f) Să se implementeze ansamblul funcţiilor logice cu MUX-uri de 4 respectiv 16 căi (circuitele sunt realizate în tehnologia TTL).**

Un circuit de multiplexare este un circuit logic combinational care, in cazul general, are 2n intrari de date (I2n-1 … I2 I1 I0), n intrari de selectie (S0 S1 … Sn+1) si o iesire (Z). Expresia iesirii Z la un moment dat este data de intrare Ik, k=0, … ,2n-1 unde k reprezinta echivalentul zecimal al numarului binar dat de starile 1 si 0 ale intrarilor de selectie: k=Sn-1, Sn-2, … , S1, S0.

Implementarea funcției Fa cu MUX de 4 căi

Circuite integrate folosite:

74LS06 74LS153



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Echiv.zec. | X1 | X2 | X3 | X4 | X5 | Fa |
| 0 | 0 | 0 | 0 | 0 | 0 | 0\* |
| 1 | 0 | 0 | 0 | 0 | 1 | 1\* |
| 2 | 0 | 0 | 0 | 1 | 0 | 0\* |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0\* |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 0 | 1 | 1 | 1 | 0\* |
| 8 | 0 | 1 | 0 | 0 | 0 | 0\* |
| 9 | 0 | 1 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1\* |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 |
| 13 | 0 | 1 | 1 | 0 | 1 | 0\* |
| 14 | 0 | 1 | 1 | 1 | 0 | 0\* |
| 15 | 0 | 1 | 1 | 1 | 1 | 0\* |
| 16 | 1 | 0 | 0 | 0 | 0 | 0\* |
| 17 | 1 | 0 | 0 | 0 | 1 | 1 |
| 18 | 1 | 0 | 0 | 1 | 0 | 1 |
| 19 | 1 | 0 | 0 | 1 | 1 | 0\* |
| 20 | 1 | 0 | 1 | 0 | 0 | 1 |
| 21 | 1 | 0 | 1 | 0 | 1 | 0\* |
| 22 | 1 | 0 | 1 | 1 | 0 | 0\* |
| 23 | 1 | 0 | 1 | 1 | 1 | 0\* |
| 24 | 1 | 1 | 0 | 0 | 0 | 1 |
| 25 | 1 | 1 | 0 | 0 | 1 | 0\* |
| 26 | 1 | 1 | 0 | 1 | 0 | 0\* |
| 27 | 1 | 1 | 0 | 1 | 1 | 0\* |
| 28 | 1 | 1 | 1 | 0 | 0 | 0\* |
| 29 | 1 | 1 | 1 | 0 | 1 | 0\* |
| 30 | 1 | 1 | 1 | 1 | 0 | 0\* |
| 31 | 1 | 1 | 1 | 1 | 1 | 0\* |

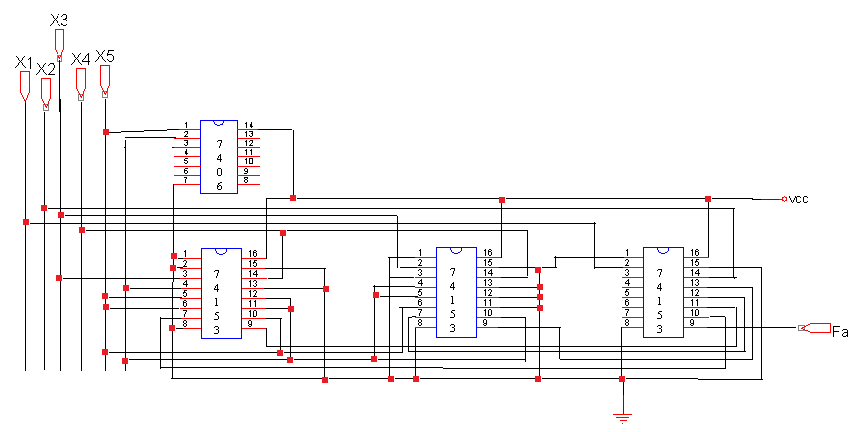




1\*74LS06(-5)

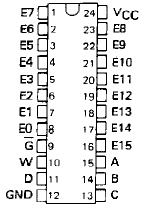
3\*74LS153(-1)

Implementare MUX de 4 căi cu circuite integrate:

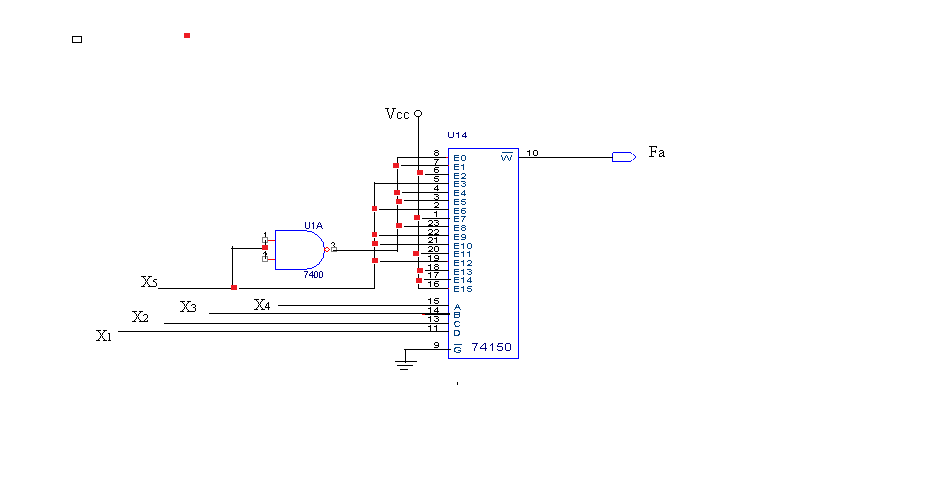


Implementarea funcției Fa cu MUX de 16 căi

Circuitele integrate folosite :



74LS00 74LS153



1\*74LS00(-3)

1\*74LS150(-)

**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 CMOS).**

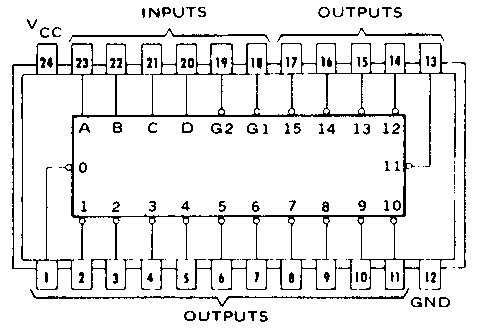
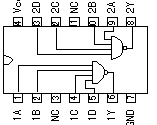
Demultiplexoarele sunt circuite combinationale care, în cazul general au o intrare de date I, n intrari de selectie S0, S1, …, Sn-1 si 2iesiri Z0, Z1, …, Z2.

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

În cazul în care functia are, în caz general, n variabile si se impune sa 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 iesirile 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 necesar sa se adauge si variabila lipsa. Acest lucru se realizeaza prin intermediul unei retele cu porti logice. La intrarile unei porti SI se vor aplica iesirile DMUX-ului corespunzatoare termenilor canonici care nu apar în expresia functiei (se ia în considerare negata functiei).

Implementarea functiei Fe cu DMUX de 16 cai cu porti ȘI-NU

Circuite folosite:



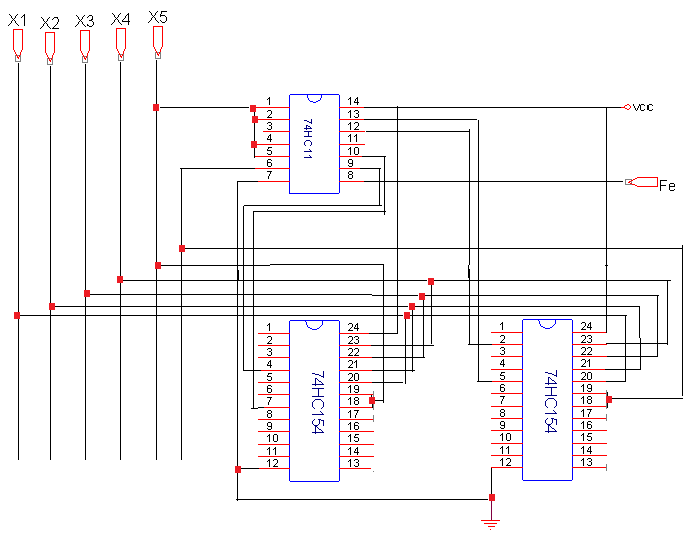
74HC154 74HC20



1\*74HC20(-)

2\*74HC154(-)

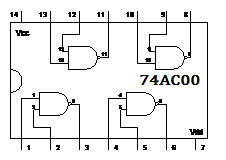
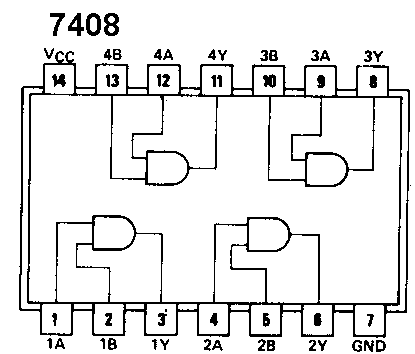
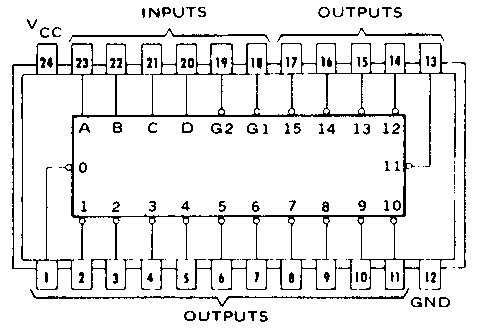
Implementarea DMUX de 16 căi cu porţi ŞI+NU cu circuite integrate



Implementarea funcției Fb cu porți Și

Circuite folosite:

74HC08 74HC154 74AC00



1\*74AC00(-3)

1\*74HC08(-3)

2\*74HC154(-)

Implementarea funcției Fe cu DMUX de 16 căi si rețea de porți ȘI-NU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | Fe |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |

=   

= =

= =

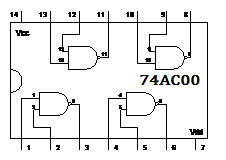
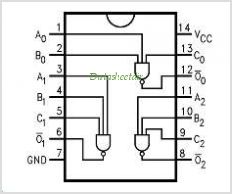
= =

= =

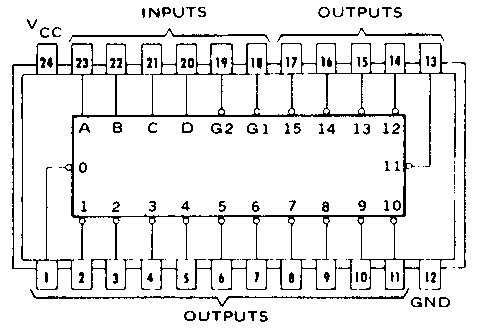


Circuite integrate folosite:

74AC00 74HC10

74HC154





1\*74AC00(-1)

1\*74HC10(-1)

2\*74HC154(-)

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

Timpii de propagare se vor calcula dupa formula: 

Date de catalog pentru circuitele folosite la T=25̊̊̊̊ C si Vcc=5V

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C.I. | Tehnologia | tPLH[ns] | tPHL[ns] | ICCH(TTL)/  IOH(CMOS)  [mA] | ICCL(TTL)/  IOL(CMOS)  [mA] | Pd [mW] |
| 74LS00 | TTL | 15 | 15 | 0.8 | 2.4 | 8 |
| 74LS04 | TTL | 15 | 15 | 15 | 30 | 112.5 |
| 74LS10 | TTL | 15 | 15 | 0.6 | 1.8 | 6 |
| 74LS150 | TTL | 35 | 33 | 40 | 40 | 200 |
| 74LS153 | TTL | 13 | 10 | 29 | 29 | 145 |
| 74HC154 | CMOS | 38 | 38 | - | - | 0,1 |
| 74AC00 | CMOS | 8.5 | 7 | -24 | 24 | 0.1 |
| 74HC02 | CMOS | 7 | 7 | -4 | 4 | 0.1 |
| 74HC10 | CMOS | 6 | 7 | -25 | 25 | 0.1 |
| 74HC27 | CMOS | 23 | 23 | -25 | 25 | 0.1 |

Calculul timpilor de propagare pentru:

* funcția Fa 
* funcția Fb 
* funcția Fc 
* funcția Fd 
* funcția Fe 
* funcția Ff 
* funcția Fg 
* ansamblu cu porți ȘI-NU



* ansamblul cu porţi ŞI-NU şi SAU-NU



* implementarea cu MUX de 4 căi



* implementare cu MUX de 16 căi



* implementare cu DMUX de 16 căi cu porti ȘI-NU



* implementare cu DMUX de 16 căi cu porți ȘI



* implementare cu DMUX de 16 căi cu rețea de porți ȘI-NU 

1. **Să se calculeze puterile disipate pentru toate schemele logice obţinute.**

Puterile disipate pe fiecare circuit integrat se vor calcula cu formula:



Calculul puterilor disipate pentru:

* Funcția Fa 
* Funcția Fb 
* Funcția Fc 
* Funcția Fd 
* Funcția Fe 
* Funcția Ff 
* Funcția Fg 
* ansamblu cu porti ȘI-NU



* ansamblu cu porti ȘI-NU si SAU-NU



* implementarea cu MUX de 4 căi



* implementarea MUX de 16 căi



* implementarea cu DMUX de 16 căi cu porți ȘI-NU



* implementarea cu DMUX de 16 căi cu porți SI



* implementarea cu DMUX de 16 căi cu rețea de porți ȘI-NU



**j) Să se compare soluţiile de implementare obţinute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tehnologia | Implementarea | C.I | tp [ns] | Pd [mW] |
| **TTL** | Fa cu porţi ŞI-NU | 1\*7400  1\*7410 | 30 | 16 |
| Ansamblul cu porţi ŞI-NU | 4\*7400  4\*7410 | 60 | 56 |
| MUX 4 căi | 3\*74LS153 | 34,5 | 435 |
| MUX 16 căi | 1\*74LS150 | 34 | 200 |
| **CMOS** | DMUX 16 căi+ŞI-NU | 2\*74HC154  1\*74HC20 | 104 | 0,3 |
| DMUX 16 căi+ŞI | 2\*74HC154  1\*74HC08  1\*74AC00 | 89,25 | 0,4 |
| DMUX cu reţea ŞI+NU | 2\*74HC154  1\*74AC00  1\*74HC10 | 90,25 | 0,4 |
| **TTL şi CMOS** | Ansamblul cu porţi ŞI-NU şi SAU-NU | 3\*74LS00  3\*74LS10  3\*74HC02  3\*74HC27 | 152,5 | 30,6 |

Cea mai puţin complexă schemă este: in tehnologia TTL implementarea cu MUX de 16 căi iar in tehnologia CMOS implementarea cu DMUX de 16 căi si porţi ŞI-NU, cea mai complexă fiind implementarea ansamblului cu porţi ŞI-NU (TTL) si SAU-NU (CMOS).

Din punct de vedere al timpilor de propagare se observă că implementările realizate in tehnologia TTL sunt mult mai rapide decât cele realizate in tehnologia CMOS. În schimb din punct de vedere al consumului de putere, implementările realizate în tehnologia CMOS sunt cele mai mici consumatoare.Implementările cu MUX sunt cele mai mari consumatoare de putere.

**Bibliografie**

1. „CURS Circuite Logice si Comenzi Secventiale”

Lector: Conf. dr. ing. Florin MOLDOVEANU

2. <http://www.ti.com>

1. <http://www.google.ro>
2. <http://www.datasheetcatalog.com>