

## **LUCRAREA NR. 5**

### **CIRCUITE LOGICE COMBINAȚIONALE MSI**

#### **1. Scopul lucrării**

Se studiază și se verifică funcționarea unor componente integrate pe scară medie (MSI – **M**edium **S**cale **I**ntegration): demultiplexor, multiplexor, decodificator. Se studiază și se verifică funcționarea convertorului de cod din binar natural în cod Gray (binar reflectat). Se studiază comportarea unor circuite combinaționale MSI conform descrierii funcționale din catalog: detector/generator de paritate, codificator prioritar.

#### **2. Considerații teoretice**

Blocurile constitutive ale dispozitivelor numerice sunt unități mai mari decât porțile logice obișnuite. În tehnologia MSI se încadrează circuitele integrate pe scară medie, și anume cele care cuprind 50-500 de tranzistori integrați.

Circuitele integrate MSI oferă utilizatorului structuri logice mai complexe, disponibile ca module standard. Din acest motiv, sinteza cu circuite integrate SSI (**S**mall **S**cale **I**ntegration) se utilizează în prezent numai acolo unde nu pot fi folosite circuitele cu înalt grad de integrare. În mod obișnuit circuitele logice elementare sunt necesare pentru a realiza adaptări sau interfațări ale circuitelor integrate MSI și LSI (**L**arge **S**cale **I**ntegration) standardizate, care nu satisfac întotdeauna cu exactitate toate cerințele proiectului.

Anumite combinații ale unui număr relativ mic de porți logice reprezintă funcții care se întâlnesc foarte des și constituie ceea ce am putea numi un al doilea nivel de circuite elementare - MSI. Întotdeauna forma ecuațiilor logice care dorim să fie implementate cu circuite MSI trebuie corelată cu circuitele integrate MSI disponibile în cataloage. Din acest motiv, un sistem de proiectat trebuie definit mai întâi sub formă de blocuri MSI și LSI, iar momentul în care se trece la scrierea ecuațiilor logice trebuie amânat cât mai mult.

## 2.1 Demultiplexorul (DMUX)

Demultiplexarea este operația de distribuire a unui semnal sursă  $x$  la mai multe destinații  $y_i$  în funcție de valoarea unor *semnale de selecție*  $s$ . În mod evident, semnalul  $s$  denotă un index, un număr. Pentru a deriva un circuit dintr-un demultiplexor, este necesar să stabilim o codificare a întregilor în termenii semnalelor numerice. Codificarea standard este cea *binară*, care se bazează pe ipoteza de pornire că valoarea binară a unui semnal ("0" sau "1") este luată drept valoare numerică și că fiecare componentă a semnalului ( $s_0, s_1, \dots$ ) este un termen ponderat în suma  $s$ , și anume:

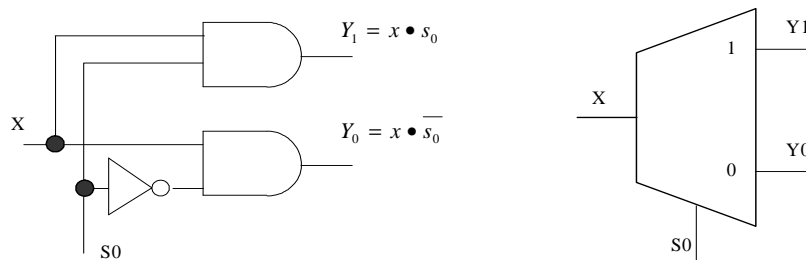
$$s = s_0 \cdot 2^0 + s_1 \cdot 2^1 + s_2 \cdot 2^2 + \dots s_i \cdot 2^i + \dots \quad (5.1)$$

Funcția demultiplexorului o exprimăm ca:

$$y_i = (\text{if } i = s \text{ then } x \text{ else } 0)$$

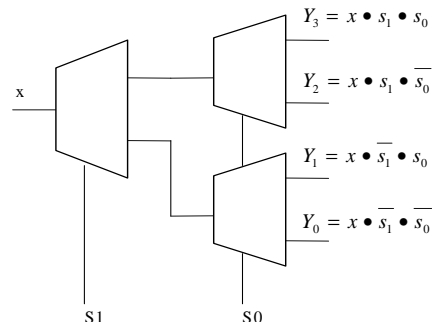
$$y_i = x \cdot (i = s)$$

Circuitul rezultat este prezentat în figura 5.1, împreună cu simbolul utilizat pentru demultiplexor.



**Figura 5.1** Demultiplexorul 1:2

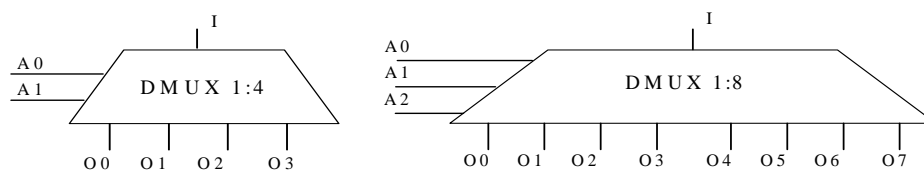
Pentru a obține un demultiplexor cu mai multe ieșiri, se folosește tehnica de *cascadare*: un demultiplexor cu  $2^n$  ieșiri poate fi într-adevăr obținut prin cascada a  $n$  nivele de demultiplexoare cu 2 ieșiri (cum se poate vedea în figura 5.2 pentru  $n=2$ ). Se vorbește despre un demultiplexor  $1$ -la- $2^n$ .



**Figura 5.2** Cascadarea demultiplexoarelor

Dacă intrarea  $x$  este menținută constantă la valoarea 1 logic, atunci  $y_s = 1$  și toate celelalte ieșiri sunt 0. În acest caz, demultiplexorul acționează ca un *decodificator* al cuvântului de cod binar  $s$  (iar  $x$  este numit *semnal de validare (enable)*). Așadar, termenii de demultiplexor și decodificator sunt adeseori folosiți ca echivalenți în practică.

Există mai multe tipuri de demultiplexoare disponibile: 1-la-2 (prezentat anterior), 1-la-4, 1-la-8, 1-la-16 etc. (figura 5.3):



**Figura 5.3** Demultiplexoare 1:4 și 1:8

## 2.2 Multiplexorul (MUX)

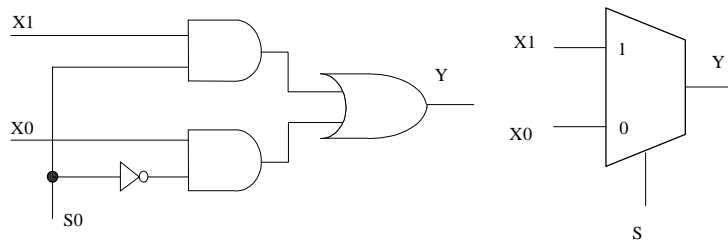
Circuitele de multiplexare sunt circuite logice combinaționale care permit trecerea datelor de la una din intrări spre o ieșire unică. Un multiplexor este inversul demultiplexorului. Funcția sa constă în a uni mai multe surse  $x_i$  către o destinație unică  $y$  în funcție de semnalul de selecție  $s$ . Din acest motiv el mai este numit și selector.

Funcția multiplexorului poate fi exprimată astfel:

$$y = (\text{if } s \text{ then } x_1 \text{ else } x_0)$$

$$y = x_0 \bullet \bar{s} + x_1 \bullet s$$

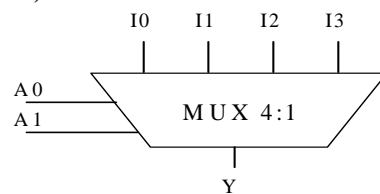
Circuitul rezultat are două intrări și este prezentat în figura 5.4, împreună cu simbolul utilizat pentru multiplexor.



**Figura 5.4** *Multiplexor 2:1*

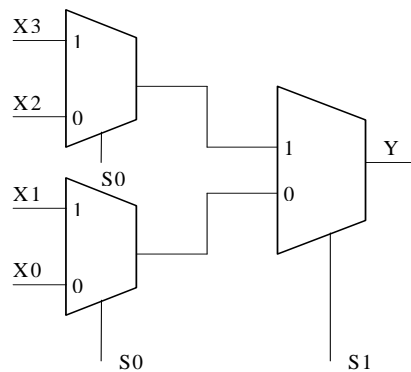
Cascadarea se face în mod analog cu cea a demultiplexoarelor.

Aceste circuite se găsesc și în formă integrată, în capsule MSI. Prezentăm în continuare un multiplexor cu 4 intrări: selectarea intrărilor se face cu 2 biți (figura 5.5):



**Figura 5.5** *Multiplexor 4:1*

Implementarea din 3 MUX 2:1 (prin cascadata) duce la realizarea unui MUX 4:1 folosind schema bloc (figura 5.6):

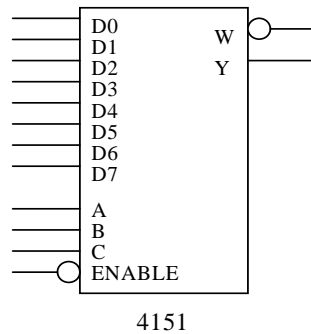


**Figura 5.6** *Multiplexor 4:1 obținut prin cascadata a trei multiplexoare 2:1*

Circuitul poate fi implementat cu porți TTL sau folosind 3 circuite integrate specializate MUX 2:1.

Prezentăm simbolul circuitului multiplexor integrat MSI 4151 cu trei intrări de selecție (adresă) care realizează funcția (figura 5.7):

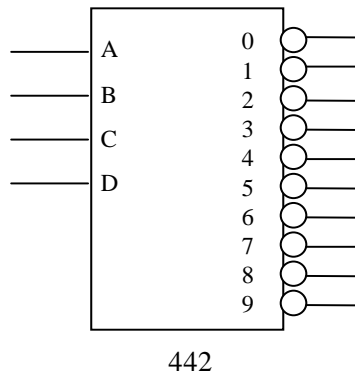
$$Y = \overline{CBA} \cdot D_0 + \overline{CBA} \cdot D_1 + \overline{CBA} \cdot D_2 + \overline{CBA} \cdot D_3 + \overline{CBA} \cdot D_4 + \overline{CBA} \cdot D_5 + \overline{CBA} \cdot D_6 + \overline{CBA} \cdot D_7$$
 când  $ENABLE = 0$ . Pentru  $ENABLE = 1$  ieșirea circuitului este  $Y = 0$ .



**Figura 5.7** Multiplexor MSI 8:1 (4151)

### 2.3 Decodificatorul

Decodificatorul este un circuit logic combinațional cu funcția de a asigura o singură ieșire a circuitului activă la un moment dat, pentru o combinație (cuvânt de cod binar) corespunzătoare a variabilelor de intrare. În general ieșirile decodificatoarelor sunt active pe “0” logic, deoarece implementarea lor este realizată cu porți de tip ȘI-NU. Vom studia ca exemplu decodificatorul BCD-zecimal, care decodifică intrările din cod BCD în zecimal. Circuitul MSI corespunzător este 442. În figura 5.8 este prezentat simbolul circuitului 442, iar în tabelul 5.1 tabelul lui de adevăr.



**Figura 5.8** Decodificator MSI BCD-zecimal (442)

**Tabelul 5.1** Tabelul de adevăr al decodificatorului BCD-zecimal

Nr	Intrări - BCD				Ieșiri - zecimal									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0
INVALID	1	0	1	0	1	1	1	1	1	1	1	1	1	1
	1	0	1	1	1	1	1	1	1	1	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	1	1	1	1	1	1
	1	1	1	0	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1

#### 2.4 Convertor de cod din binar natural în binar reflectat (Gray)

Un convertor de cod este un circuit combinațional cu ieșiri multiple care realizează conversia unui cuvânt dintr-un cod în alt cod.

Se consideră că la intrare se aplică un cuvânt în cod binar natural pe 4 biți  $B_3B_2B_1B_0$ , iar la ieșire se obține cuvântul în cod Gray (binar reflectat), tot pe 4 biți,  $G_3G_2G_1G_0$ .

În tabelul 5.2 se prezintă funcționarea convertorului de cod din binar natural în binar reflectat (Gray).

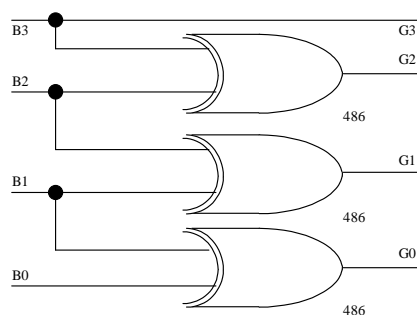
**Tabelul 5.2** Tabelul de adevăr al unui convertor de cod din binar natural în binar reflectat

B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Expresiile funcțiilor de ieșire, după minimizare, sunt (5.2):

$$\begin{aligned}
 G_0 &= B_1 \bullet \overline{B_0} + B_0 \bullet \overline{B_1} = B_1 \oplus B_0 \\
 G_1 &= B_2 \bullet \overline{B_1} + B_1 \bullet \overline{B_2} = B_2 \oplus B_1 \\
 G_2 &= B_3 \bullet \overline{B_2} + B_2 \bullet \overline{B_3} = B_3 \oplus B_2 \\
 G_3 &= B_3
 \end{aligned} \quad (5.2)$$

Implementarea funcțiilor obținute pentru ieșirile convertorului de cod se poate face cel mai simplu utilizând porți logice SAU-EXCLUSIV, ca în figura 5.9.



**Figura 5.9** Convertor de cod din binar în binar reflectat (Gray)

## 2.5 Detector / generator de paritate

Circuitul integrat 4180 din figura 5.10 reprezintă un detector / generator de paritate pe 9 biți (8 biți în cuvântul de cod transmis + bitul de paritate), a cărui comportare este descrisă în tabelul 5.3. Acest gen de circuite se folosește mai ales în transmisii de date, pentru verificarea corectitudinii transmisiei.

**Tabelul 5.3** Tabelul de adevăr al unui detector / generator de paritate

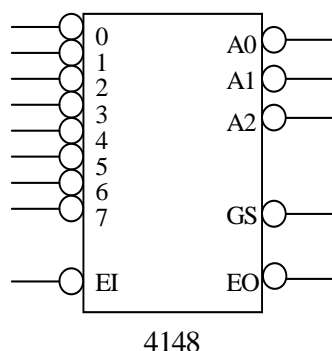
Intrări			Ieșiri	
$\Sigma$ de valori de 1 de la A la H	EI	OI	$\Sigma$ par	$\Sigma$ impar
Par	1	0	1	0
Impar	1	0	0	1
Par	0	1	0	1
Impar	0	1	1	0
X	1	1	0	0
X	0	0	1	1

În funcție de operația efectuată, generare sau detectare de paritate, intrările EI (pentru paritate pară) și OI (pentru paritate impară) sunt folosite ca și bit de paritate (bitul 9).

Lungimea cuvintelor a căror paritate se generează / verifică poate fi extinsă prin cascada circuitelor 4180.







**Figura 5.11** Codificatorul prioritar integrat MSI 4148

### 3. Desfășurarea lucrării

1. Se realizează circuitul din figura 5.1 și i se verifică funcționarea conform ecuațiilor.
2. Se realizează circuitele din figurile 5.2, 5.3 și 5.4 și se verifică funcționarea lor.
3. Verificați comportarea circuitului multiplexor 8:1, 4151 (figura 5.7) și a circuitului decodificator BCD – zecimal 442 (figura 5.8).
4. Verificați comportarea circuitului detector / generator de paritate 4180 (figura 5.10) și extindeți domeniul la 16 biți.
5. Verificați comportarea circuitului codificator prioritar 4148 (figura 5.11).
6. Implementați funcția:

$$f(A, B, C, D, E) = A + \overline{C} \cdot D + B \cdot \overline{D} + \overline{B} \cdot D + \overline{B} \cdot C \cdot E$$

folosind *numai* un multiplexor. Sunt disponibile semnalele “0”, “1” și variabilele numai în forma directă, nu și negate. Verificați practic corectitudinea implementării.

7. Verificați practic comportarea convertorului de cod din codul binar natural în codul Gray (binar reflectat) (figura 5.9), conform tabelului de adevăr și ecuațiilor.