

Logică digitală

-Curs 6-
Circuite logice
combinaționale

Agenda discutie

- ☐ Quiz – interactive video
 - Video interactive: MUX, DEC
 - ☐ Circuite combinationalale, clasificare
 - ☐ Sumatoare: RCA, CLA
 - ☐ ALU
 - ☐ MUX, DEC
-

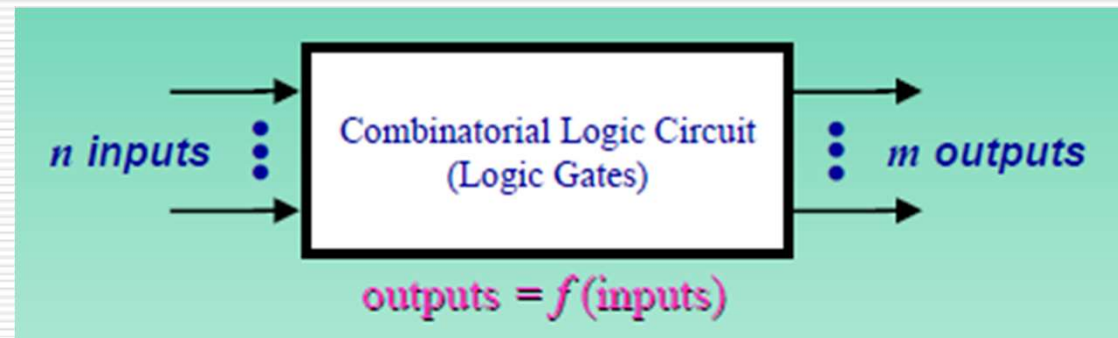
Circuite logice combinaționale

- ☐ Circuite de procesare
 - ☐ Circuite de conversie
 - ☐ Circuite de interconectare
 - ☐ Componente universale
-

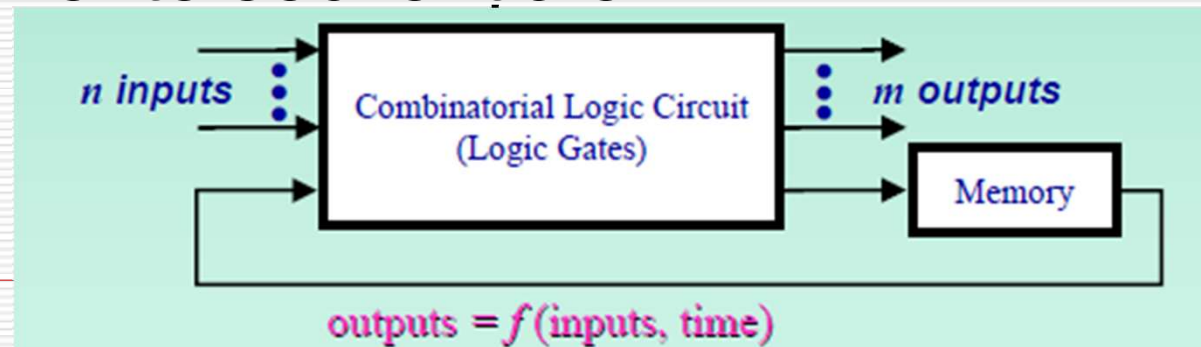
Clasificare componente digitale

□ Componente combinaționale

- Ușor de analizat, partiționat, verificat



□ Componente secvențiale



Clasificare circuite combinaționale (I)

□ Procesare

- Operații aritmetice (Adunare, Scădere, Înmulțire, Împărțire)
 - Operații logice (ȘI, SAU-Exclusiv, Negare, etc.)
 - Comparare
 - Operații de manipulare la nivel de bit (shift-are, rotație, ...).
-

Clasificare circuite combinaționale (II)

- Conversie date
 - Codificatoare
 - Decodificatoare
 - Interconnect-uri
 - Selecția sursei/destinației
 - Magistrale și interfete magistrală
 - Alte componente (blocuri din UC)
 - ROM
 - PLA
-

Cuvinte cheie design digital

☐ Încapsulare

- Definirea unor componente/blocuri simple

☐ Iterare

- Replicarea/Instanțierea componentelor în design

☐ Ierarhie

- Realizarea unor blocuri mai mari din blocuri mai mici
-

Exemplu – Sumatorul cu propagare serială a transportului

x_i	y_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth Table

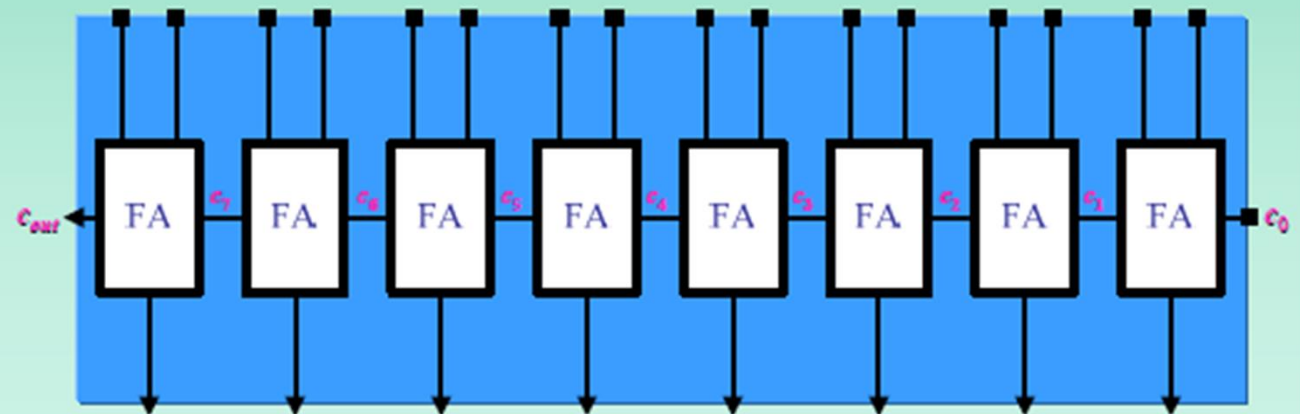
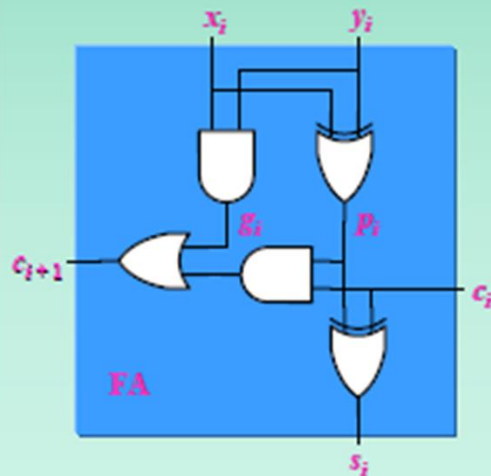
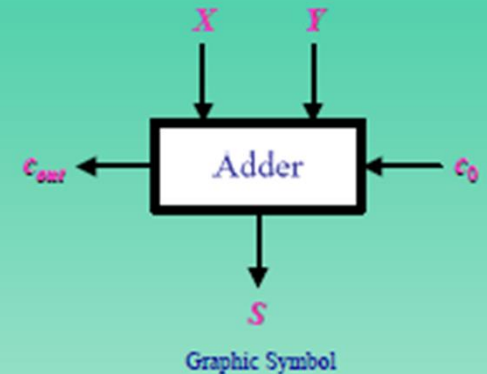
$x_i y_i$	00	01	11	10
c_i				
0		1		1
1	1		1	

$$s_i = x_i \oplus y_i \oplus c_i$$

Map Representation

$x_i y_i$	00	01	11	10
c_i				
0			1	
1		1	1	1

$$c_{i+1} = x_i y_i + c_i (x_i \oplus y_i)$$



CLA – Sumatorul cu calculul anticipat al transportului

- ☐ Pentru creșterea performanței sumatoarelor se încearcă calculul transportului în mod anticipat
 - ☐ transportul de pe un rang să nu depindă de transportul de pe rangul anterior
-

CLA

$$c_1 = a_0b_0 + c_0(a_0 + b_0) = g_0 + c_0p_0$$

$$c_2 = a_1b_1 + c_1(a_1 + b_1) = g_1 + c_1p_1 = g_1 + (g_0 + c_0p_0)p_1 =$$

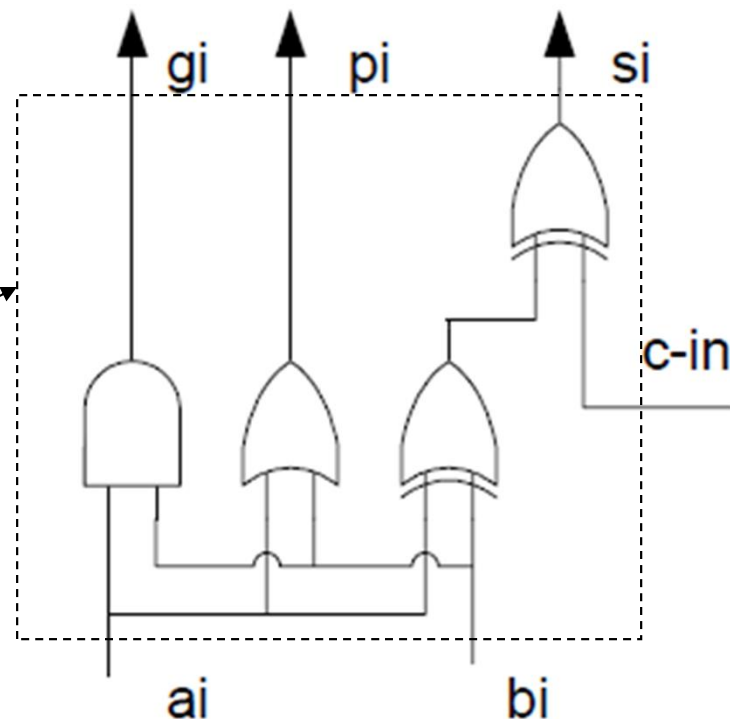
$$= g_1 + g_0p_1 + c_0p_0p_1$$

$$c_3 = g_2 + g_1p_2 + g_0p_1p_2 + c_0p_0p_1p_2$$

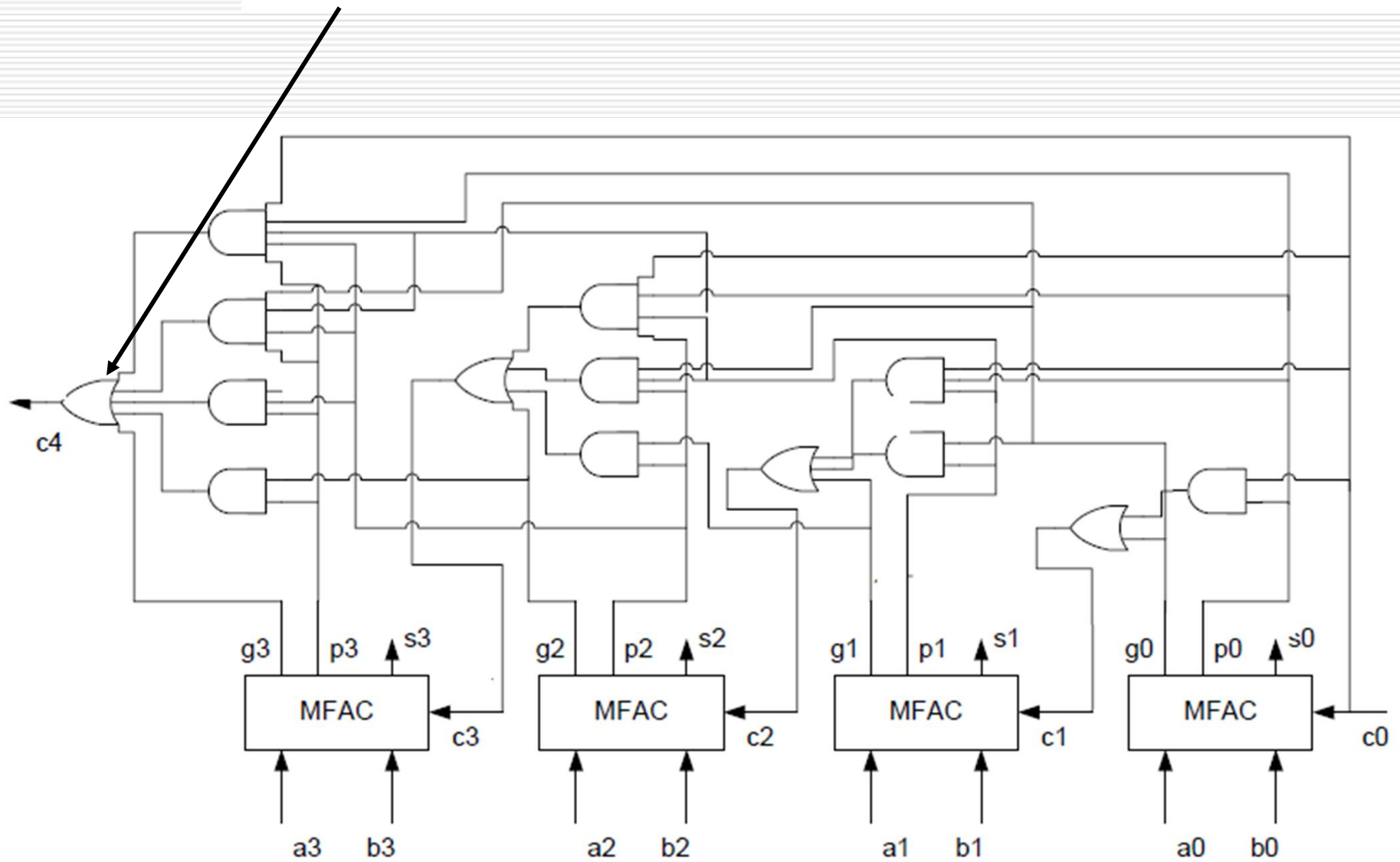
$$c_4 = g_3 + g_2p_3 + g_1p_2p_3 + g_0p_1p_2p_3 + c_0p_0p_1p_2p_3$$

Unde $g_i = a_i b_i$ și $p_i = a_i + b_i$

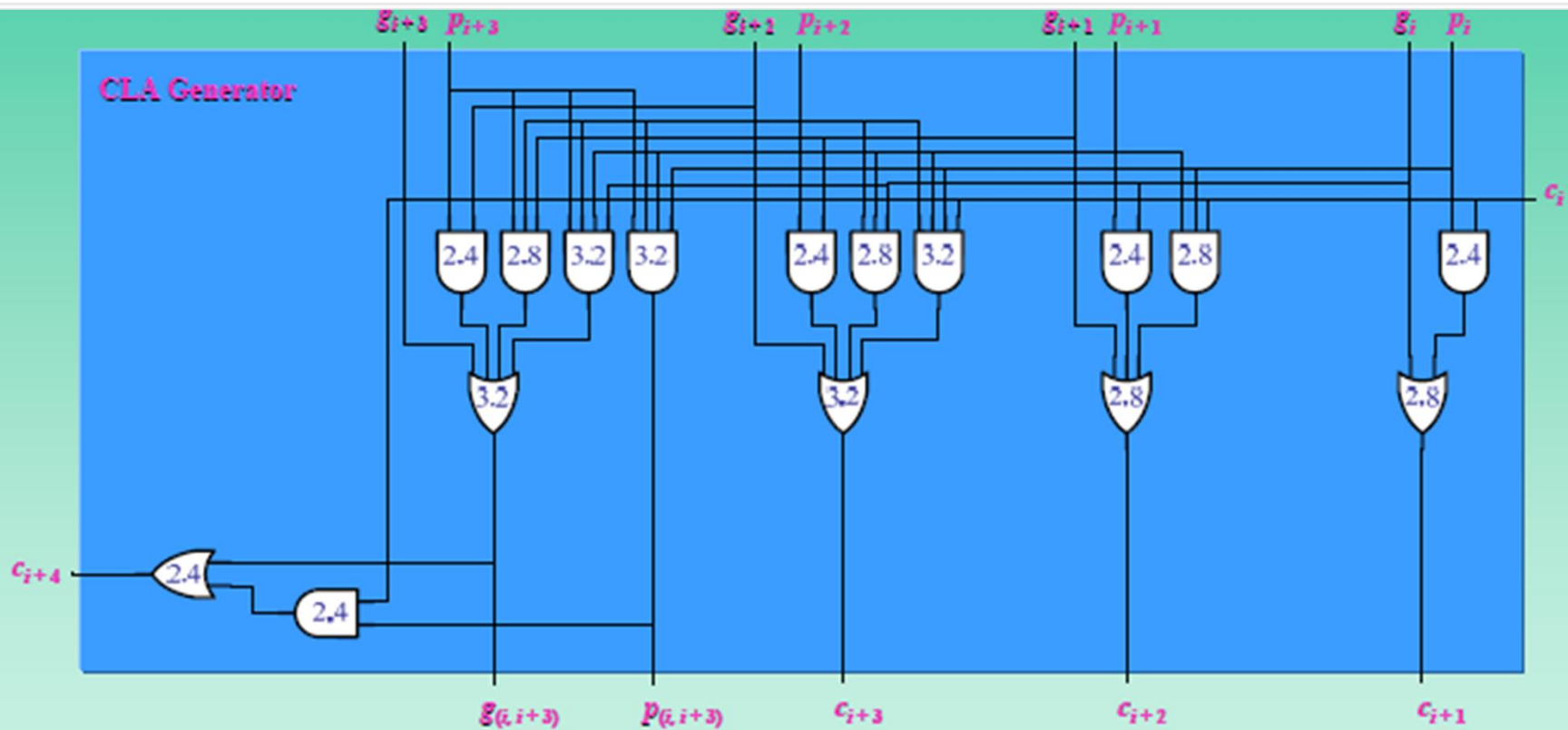
**Celula de
însuare
modificata**



$$c_4 = g_3 + g_2p_3 + g_1p_2p_3 + g_0p_1p_2p_3 + c_0p_0p_1p_2p_3$$



CLA Generator



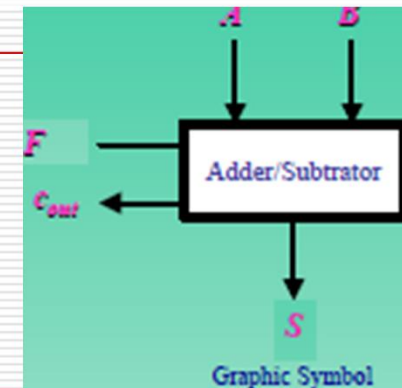
Observații:

- ❑ numărul de intrări a anumitor porți crește foarte mult, consecințe:
 - întârzierea porților crește cu număr de intrări;
 - bibliotecile de porți pot să nu aibă porți cu atâtea intrări
 - ❑ Fanout (numărul de porți comandate de ieșirea unei porți) crește considerabil pentru anumite porți
 - ❑ Calculul transporturilor se face în paralel
-

Unitate sumator/scăzător C₂

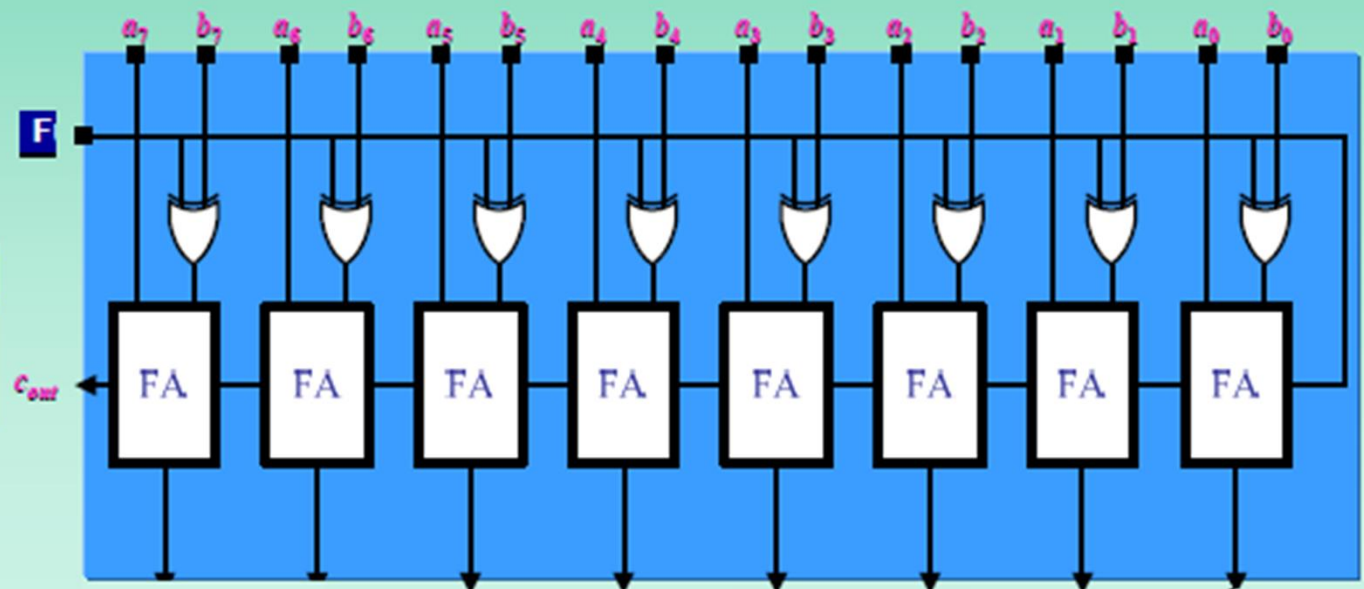
□ Scădere

$$A - B = A + B' + 1$$



<i>F</i>	Function	Comment
0	$A + B$	Addition
1	$A + B' + 1$	Subtraction

Truth Table

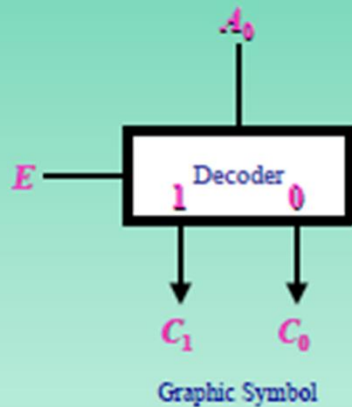


8-bit Adder/Subtractor Unit Schematic

Decodificator

- circuite logice combinaționale ce prezintă un anumit n intrări și până la 2^n ieșiri, care activează **ieșirea (UNA SINGURĂ)** corespunzătoare valorii combinației vectorului de intrare
 - Pot avea intrări de activare, astfel încât ieșirea selectată nu pot fi activată decât dacă intrările de activare sunt active.
 - Pt. n intrări și cu m ieșiri \rightarrow decodificator n -la- m .
 - Uzual sunt folosite pt. activarea (EN) componentelor
-

Decodificatorul 1-la-2



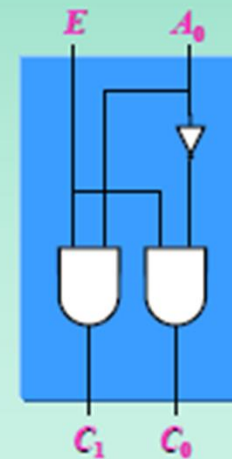
$$C_0 = EA'_0$$

$$C_1 = EA_0$$

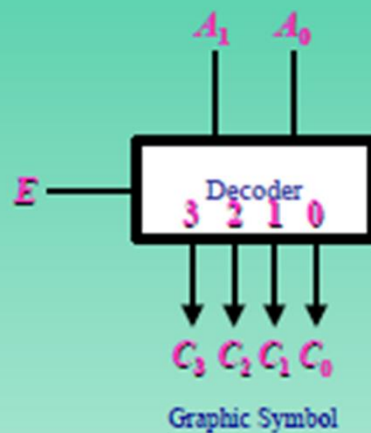
Boolean Expression

E	A_0	C_1	C_0
1	0	0	1
1	1	1	0
0	X	0	0

Truth Table



Decodificatorul 2-la-4

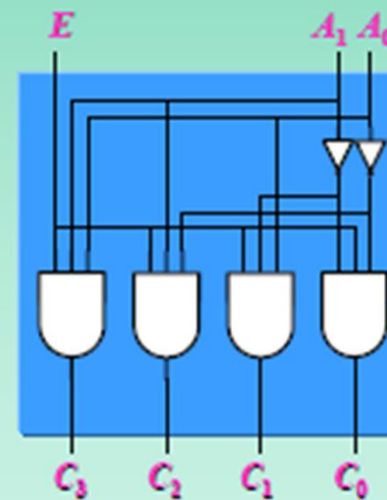


$$\begin{aligned}
 C_0 &= E_0 A'_1 A'_0 \\
 C_1 &= E_0 A'_1 A_0 \\
 C_2 &= E_0 A_1 A'_0 \\
 C_3 &= E_0 A_1 A_0
 \end{aligned}$$

Boolean Expression

E	A_1	A_0	C_3	C_2	C_1	C_0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	X	X	0	0	0	0

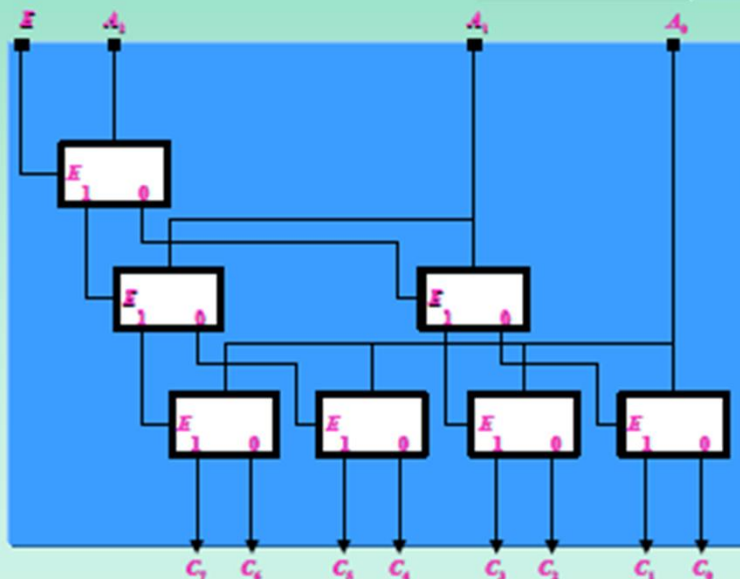
Truth Table



Decodificatorul 3-la-8

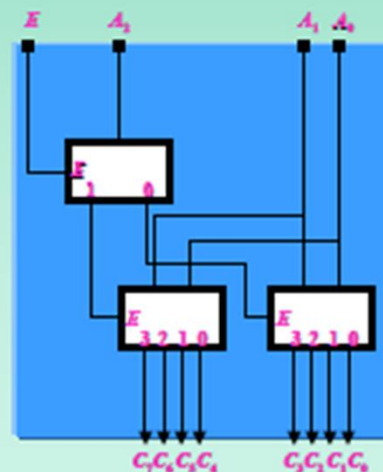
E	A_2	A_1	A_0	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
0	X	X	X	0	0	0	0	0	0	0	0

Truth Table

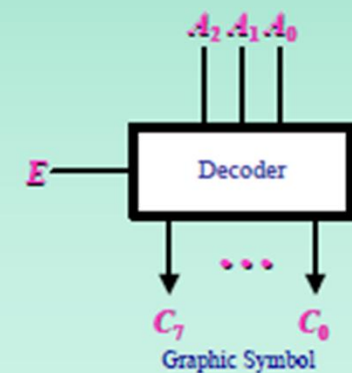


Implementation with 1-to-2 Decoders

Copyright © 2004-2005 by Daniel D. Gajski



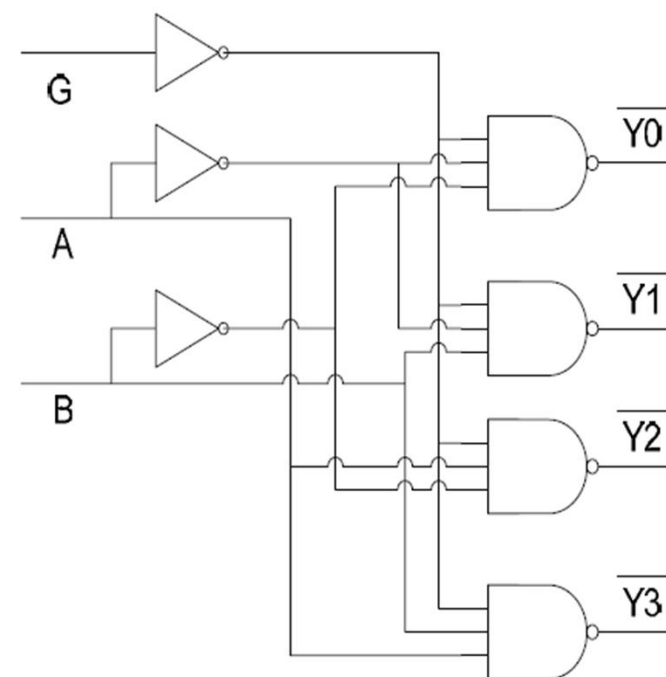
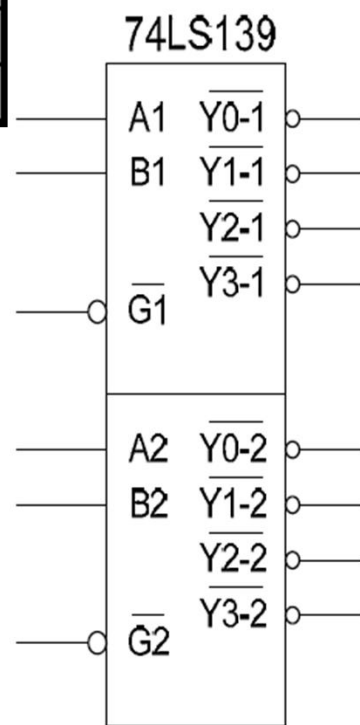
Implementation with 2-to-4 Decoders



Graphic Symbol

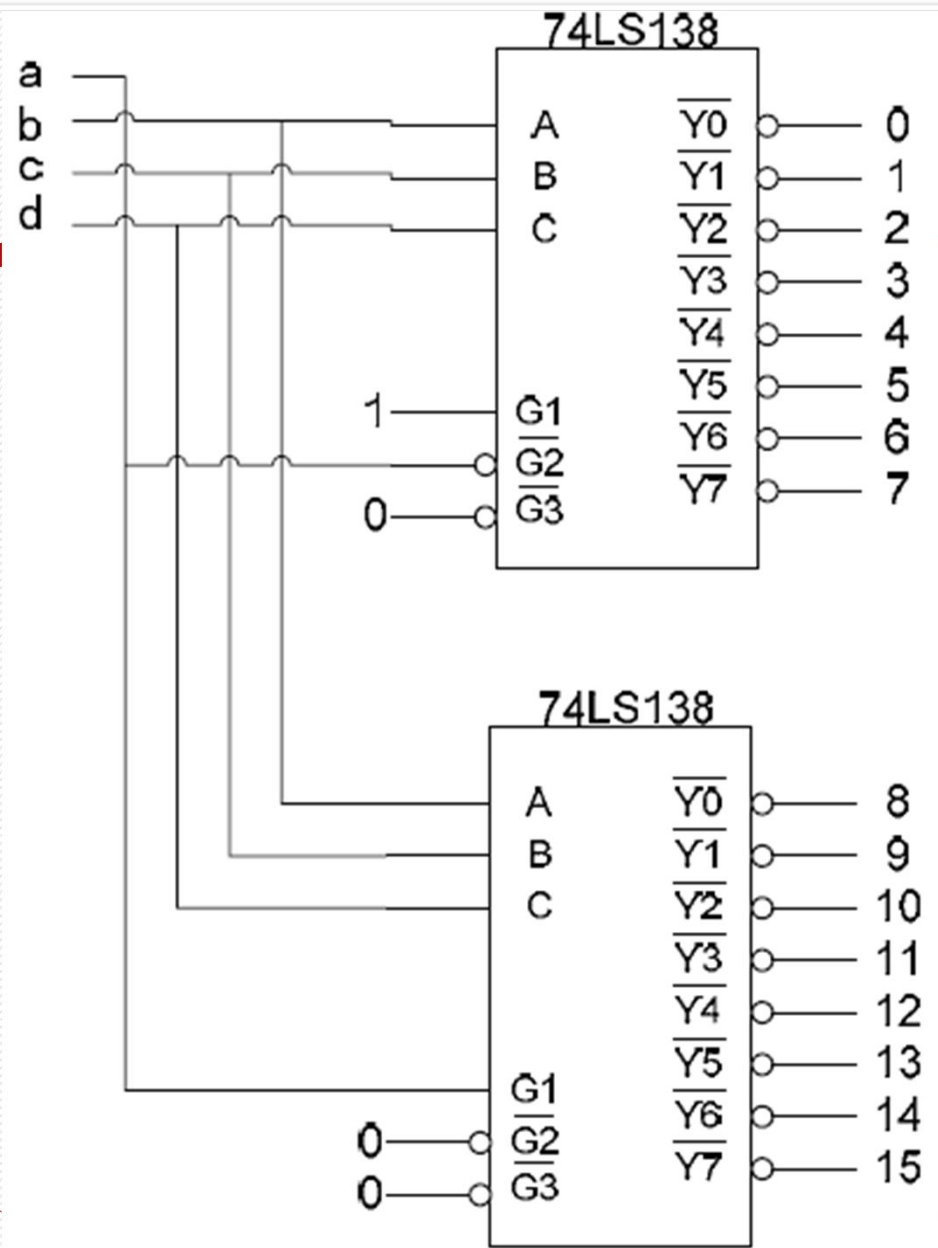
Circuite integrate pe scară medie ce îndeplinesc funcția de decodificator

G	A	B	Y0	Y1	Y2	Y3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

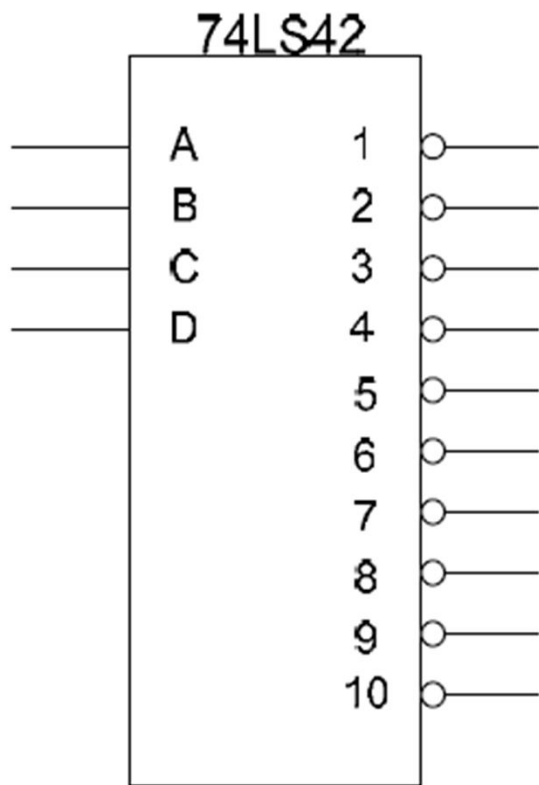


[illegible]

DEC 4-la-16

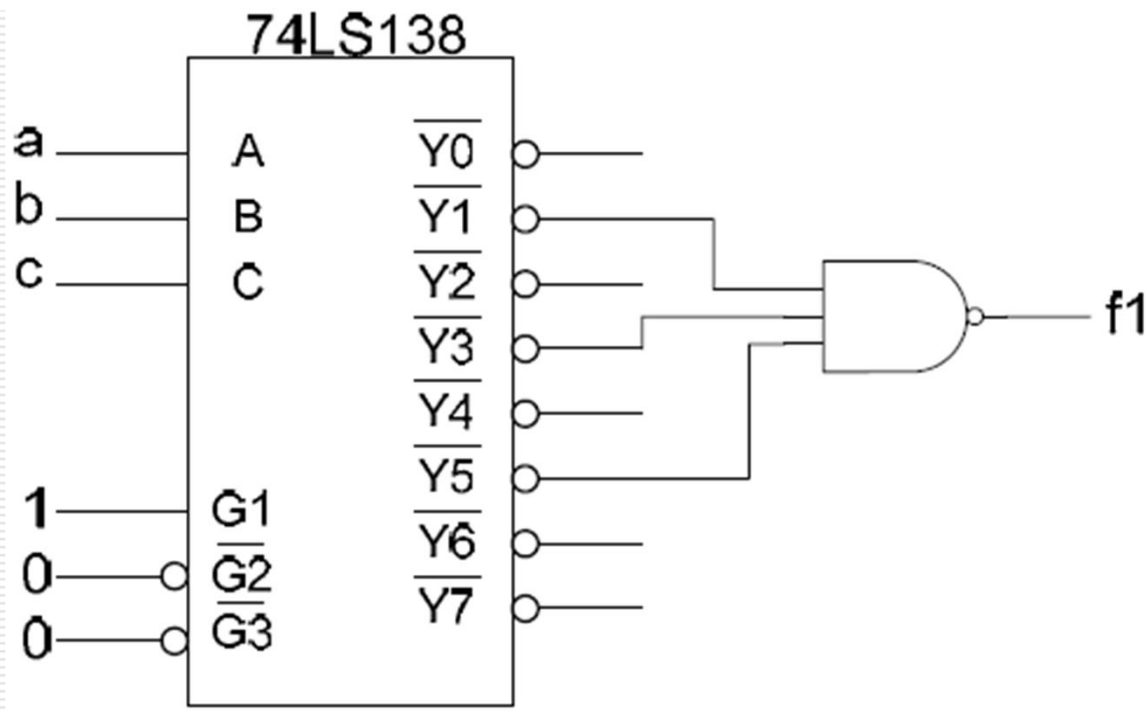


14LS42: Decodificator 4-la-10

[illegible]

Sinteza funcțiilor logice folosind decodificatoare

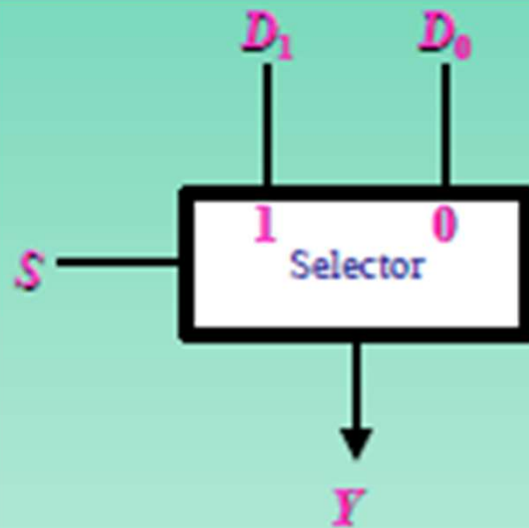
Să se implementeze cu ajutorul unui decodificator 74LS138 funcția logică
 $f1(a,b,c) = \sum(1,3,5)$



Multiplexor(Selector)

- ❑ Multiplexorul este un circuit logic combinațional ce conectează ieșirea acestuia la una din cele n intrări.
 - ❑ Selecția uneia din cele n intrări se face cu ajutorul a $\log_2 n$ intrări de selecție.
 - ❑ Poate fi privit ca un comutator digital.
 - ❑ Este folosit pt. selecția unei singure surse de date din mai multe.
-

MUX 2-la-1



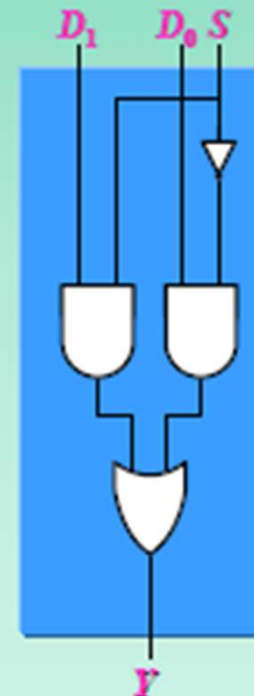
Graphic Symbol

$$Y = S'D_0 + SD_1$$

Boolean Expression

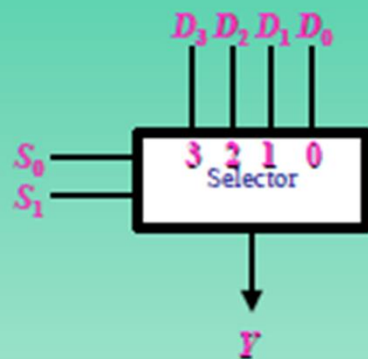
S	Y
0	D_0
1	D_1

Truth Table



Logic Schematic

MUX 4-la-1



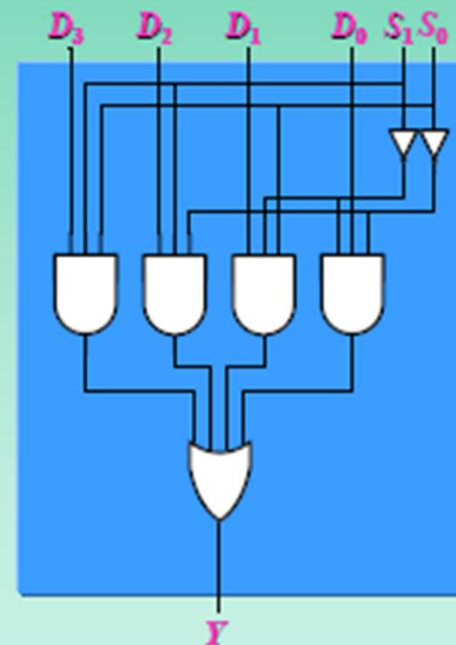
Graphic Symbol

$$Y = S'_1 S'_0 D_0 + S'_1 S_0 D_1 + S_1 S'_0 D_2 + S_1 S_0 D_3$$

Boolean Expression

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Truth Table

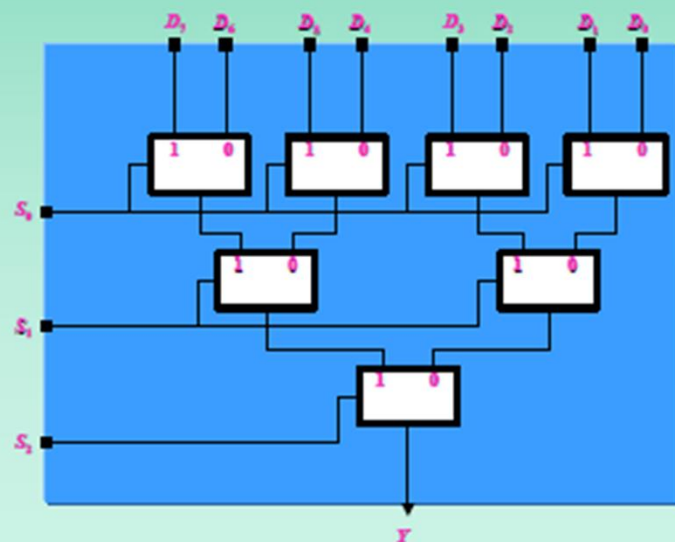


Logic Schematic

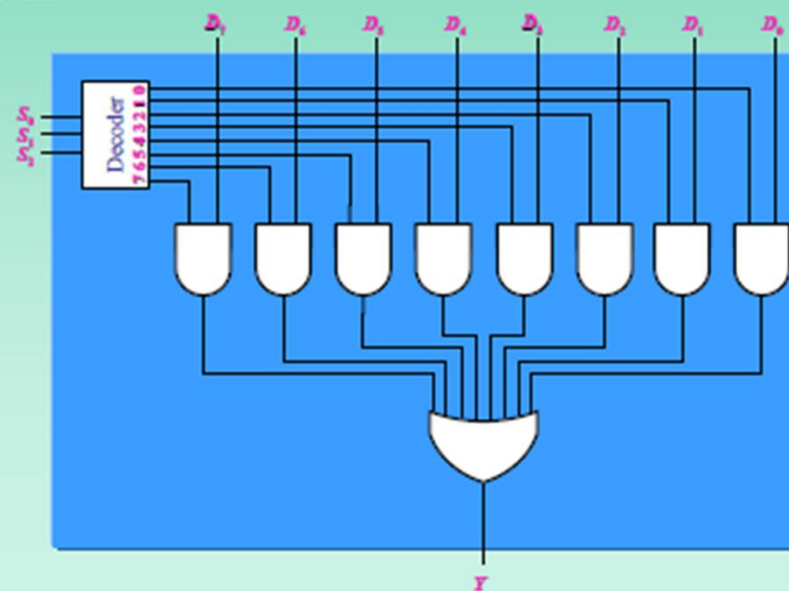
MUX 8-la-1

S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Truth Table

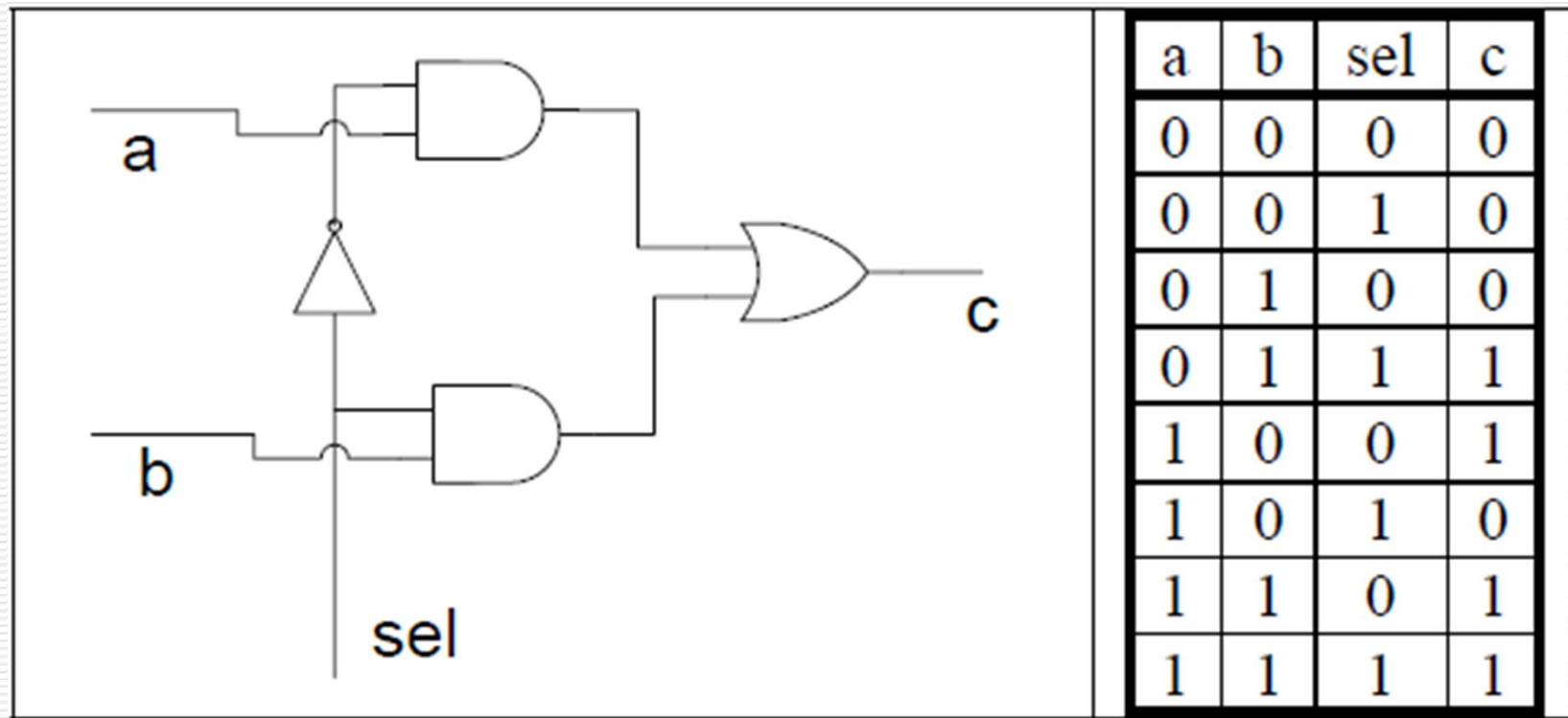


Implementation with 2-to-1 Selectors

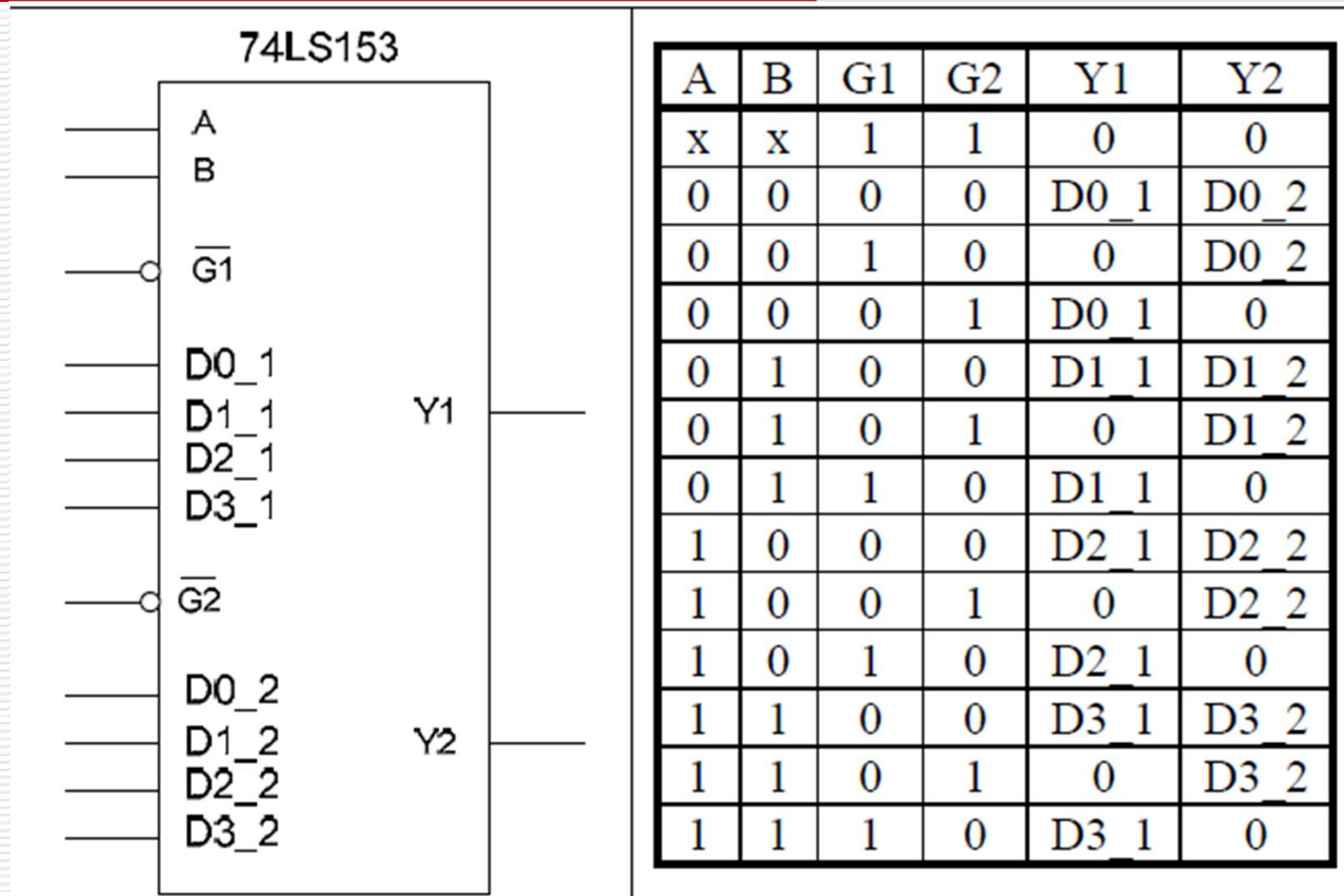


Implementation with 3-to-8 Decoder

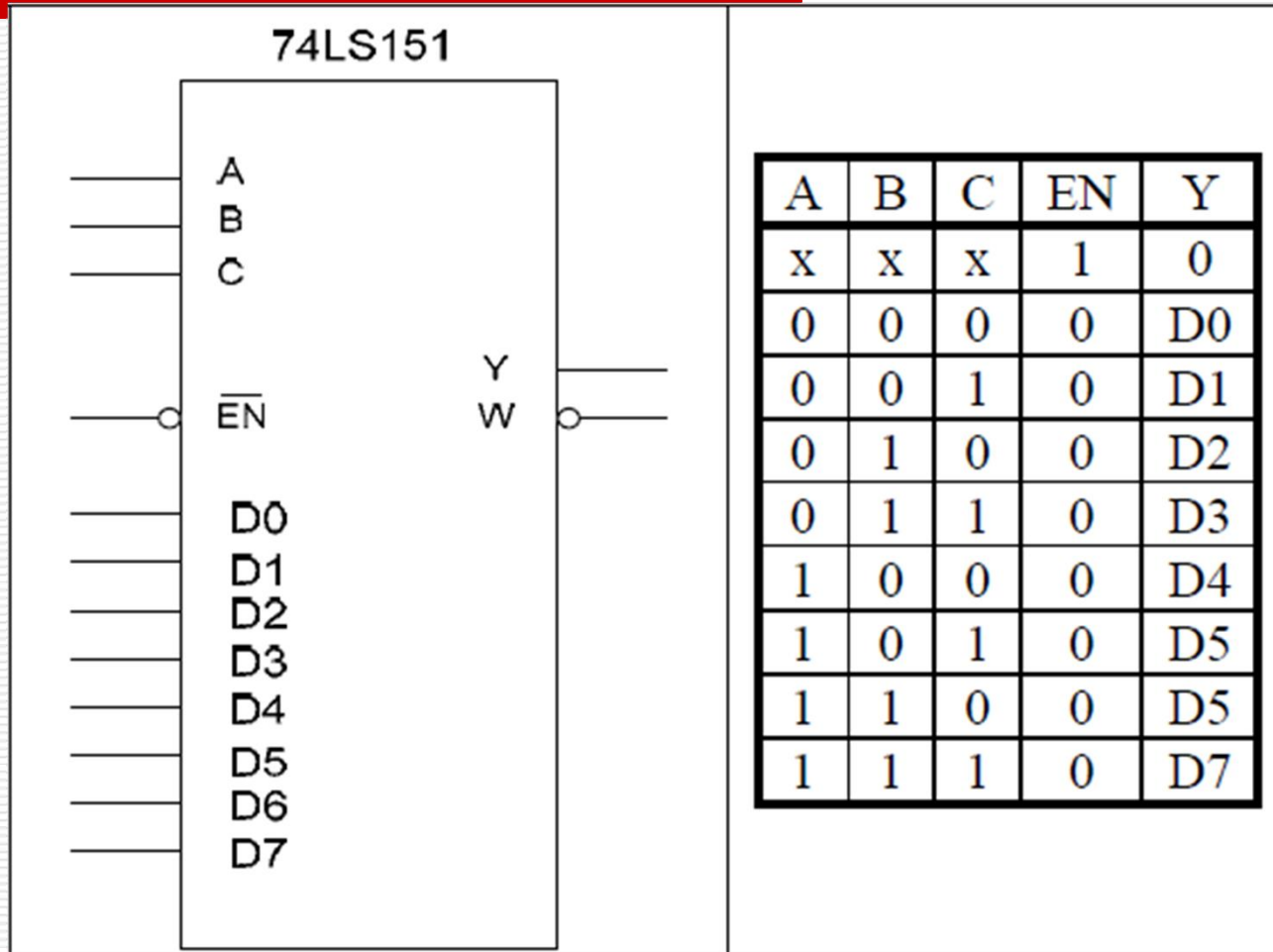
MUX 2-la-1



74LS153: MUX 4-la-1



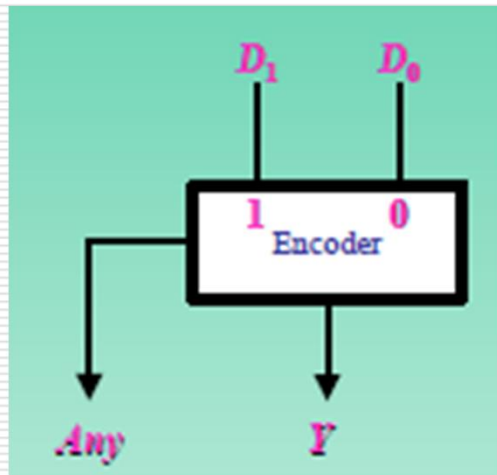
74LS151: MUX 8-la-1



Codificatoare

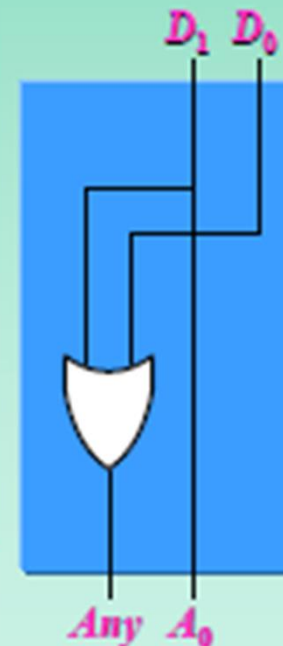
- ❑ Circuite combinaționale care realizează într-un sens funcția inversă decodificatoarelor
- ❑ Dacă conectăm un decodicator la ieșirea unui codificator nu obținem identitatea!
- ❑ Au până la 2^n intrări și un număr de n ieșiri
- ❑ În cele mai multe cazuri valoarea ieșirii unui codificator este dată de indexul celui mai semnificativ bit de intrare activ
- ❑ exemplu de aplicație: arbitrarea accesului la o resursă (ex. magistrală, controler întreruperi) → **codificator de prioritate**

Codificator 2-la-1

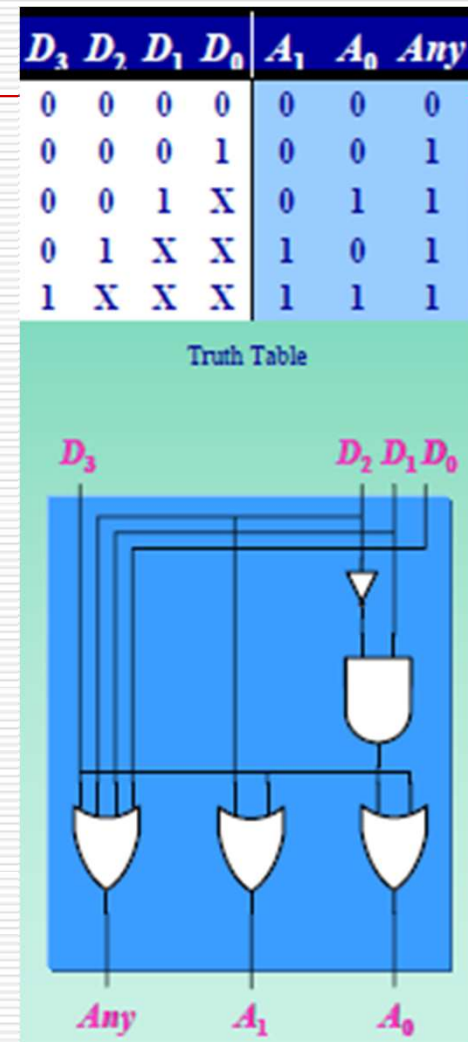
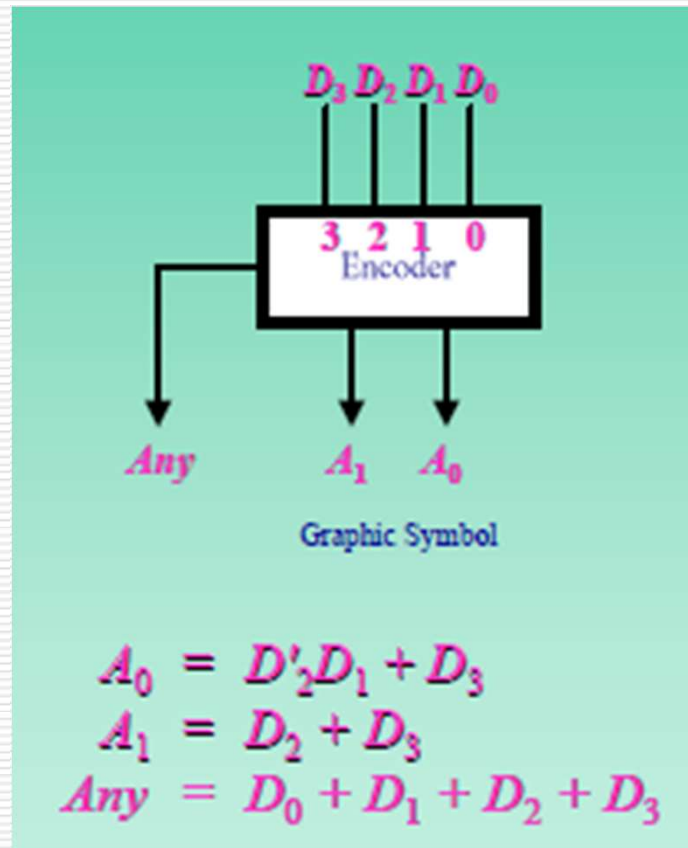


D_1	D_0	A_0	Any
0	0	0	0
0	1	0	1
1	X	1	1

Truth Table

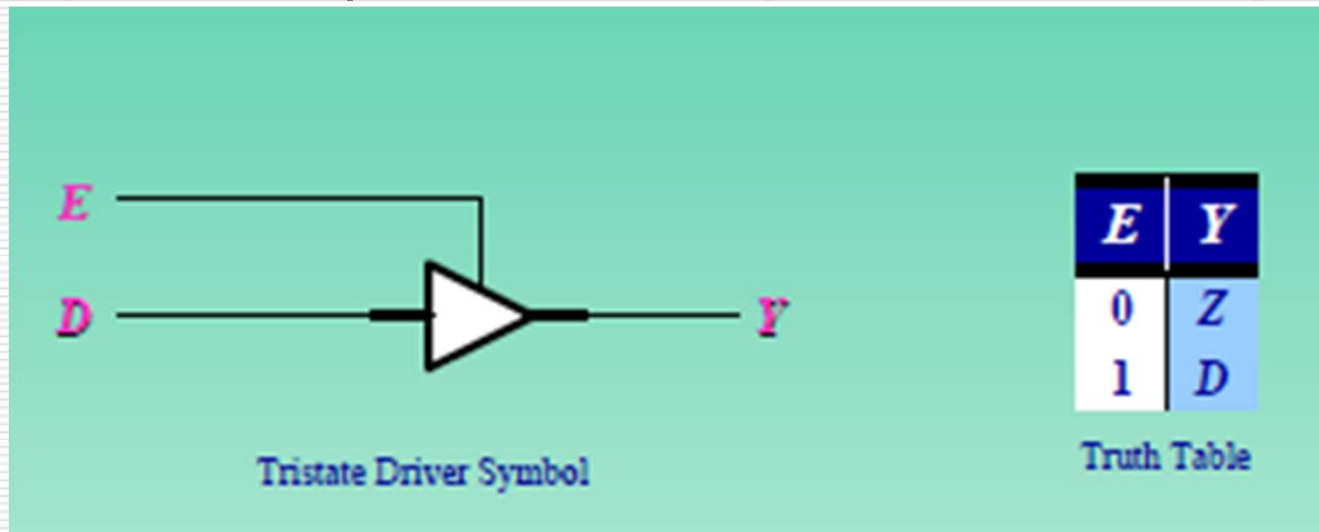


Codificator 4-la-2

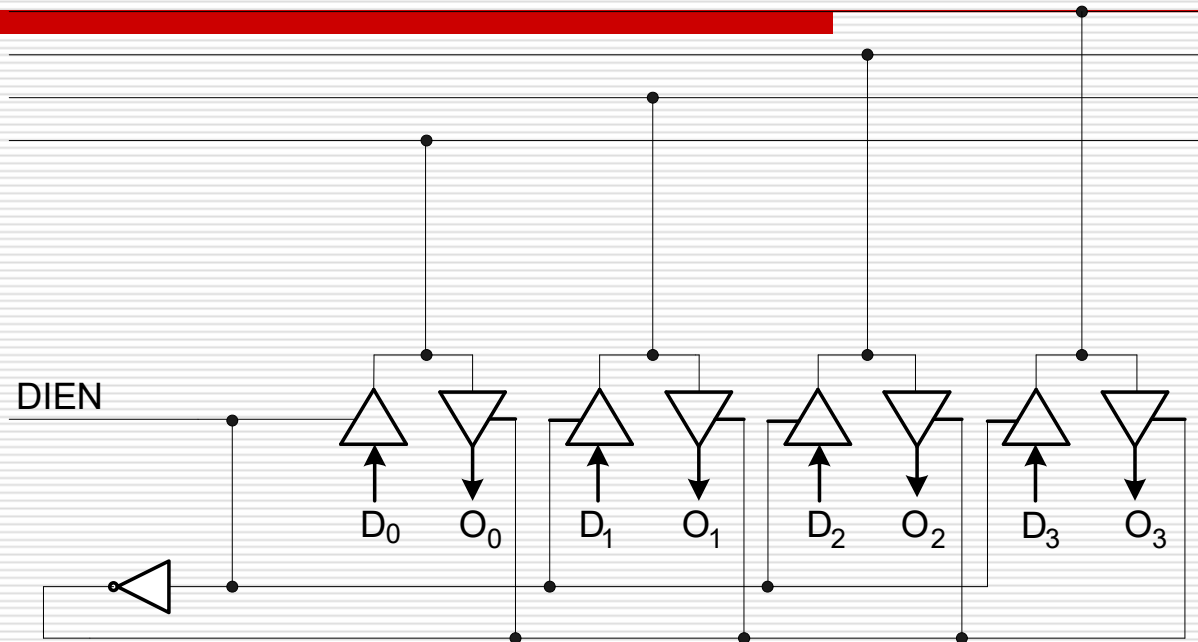


Magistrale

- Poartă cu trei stări (tristate driver) are trei valori de ieșire: 0, 1, Z - impedanță ridicată (\equiv disconnect)



Magistrale: Utilizarea circuitelor poartă cu trei stări pentru conectarea la magistrala date



D_i – date transmise pe magistrală

O_i – date recepționate de pe magistrală

DIEN- comandă de transmisie/recepție date

Întrebări?

**Enough Talking Let's Get To It
!!Brace Yourselves!!**

