

ΕΡΓΑΣΤΗΡΙΟ 3

ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ



Περιεχόμενα

... συνέχεια των αριθμητικών κυκλωμάτων

- Πολλαπλασιαστής 2×2 bit
- Πολλαπλασιαστής 4×2 bit

Συγκριτής - Αποκωδικοποιητής - Πολυπλέκτης *Comparator - Decoder (DEC) - Multiplexer (MUX)*

- Συγκριτής 1 bit
- Συγκριτής 2 bit
- Αποκωδικοποιητής 2×4
- Αποκωδικοποιητής 3×8
- Πολυπλέκτης 2 σε 1
- Πολυπλέκτης 4 σε 1

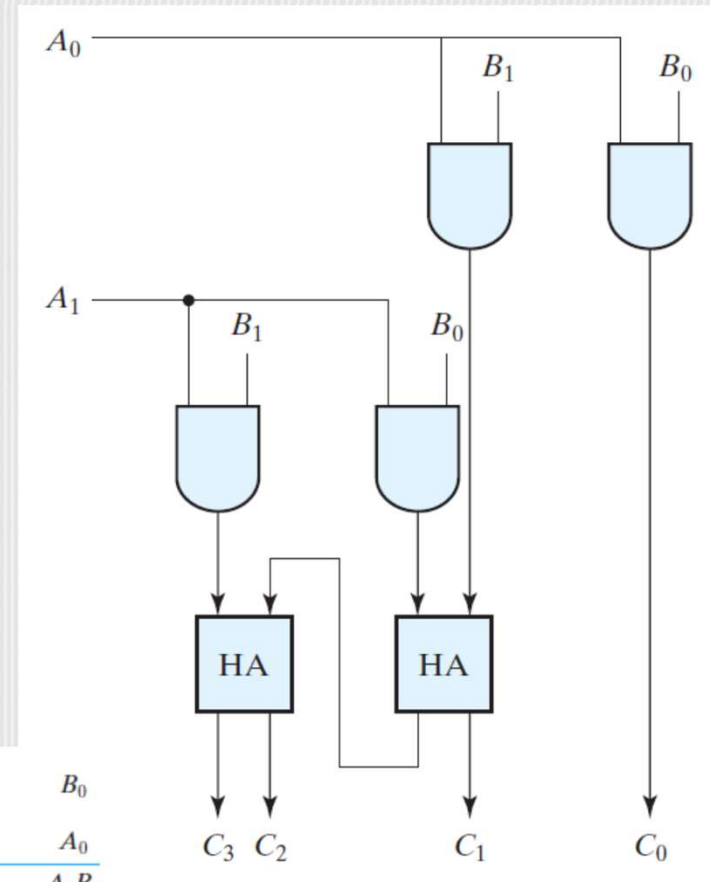
Δυαδικός Πολλαπλασιαστής 2x2

- Να σχεδιαστεί το κύκλωμα του δυαδικού πολλαπλασιαστή 2x2.
- Ποια στρατηγική/λογική ακολουθούμε;

		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0

Δυναδικός Πολλαπλασιαστής 2×2

- Ποιος ο ρόλος των πυλών AND;
- Πως επιτυγχάνεται η μετατόπιση;
- Οι έξοδοι έχουν την έννοια του κρατούμενου;
- Θα μπορούσαμε και με 2 Full-adders;



	B ₁	B ₀
A ₁	A ₁ B ₁	A ₁ B ₀
A ₀	A ₀ B ₁	A ₀ B ₀
	C ₃ C ₂	C ₁ C ₀

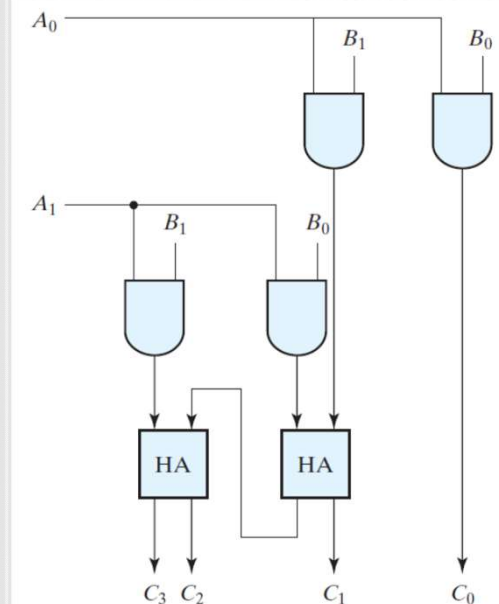
Δυαδικός Πολλαπλασιαστής 4x2

- Να σχεδιαστή το κύκλωμα του δυαδικού πολλαπλασιαστή 4x2
- Πόσες εξόδους θα έχουμε;
- Τι πύλες ή υποκυκλώματα θα χρησιμοποιήσουμε;

Βοηθητικά παραθέτω το κύκλωμα του 2x2 Multiplier

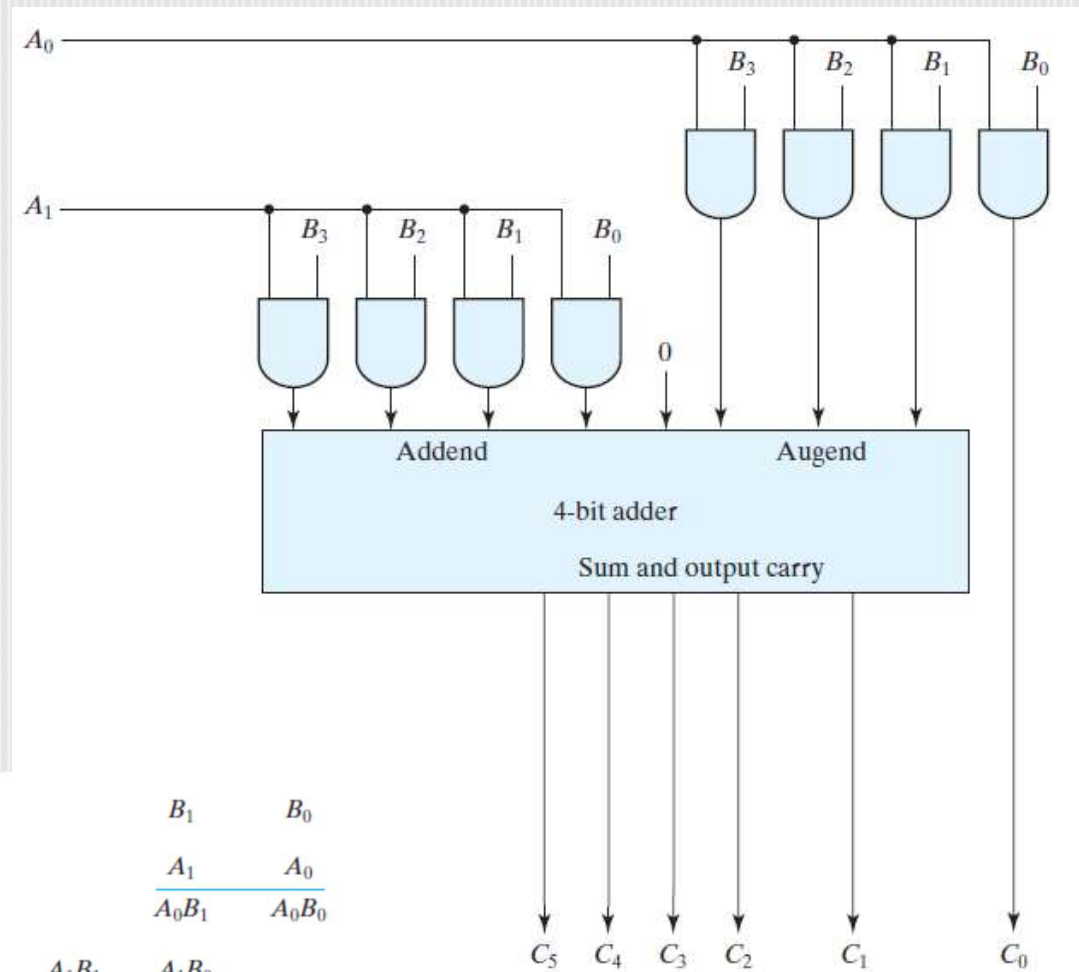


	B_1	B_0
A_1	A_1B_1	A_1B_0
A_0	A_0B_1	A_0B_0
	C_3	C_2
	C_1	C_0



Δυαδικός Πολλαπλασιαστής 4×2

- Πως επιτυγχάνεται η μετατόπιση πριν την πρόσθεση;
- Για ποιο λόγο υπάρχει μια είσοδος μηδέν στον 4-bit adder;



Περιεχόμενα

Συγκριτής - Αποκωδικοποιητής - Πολυπλέκτης *Comparator - Decoder (DEC) - Multiplexer (MUX)*

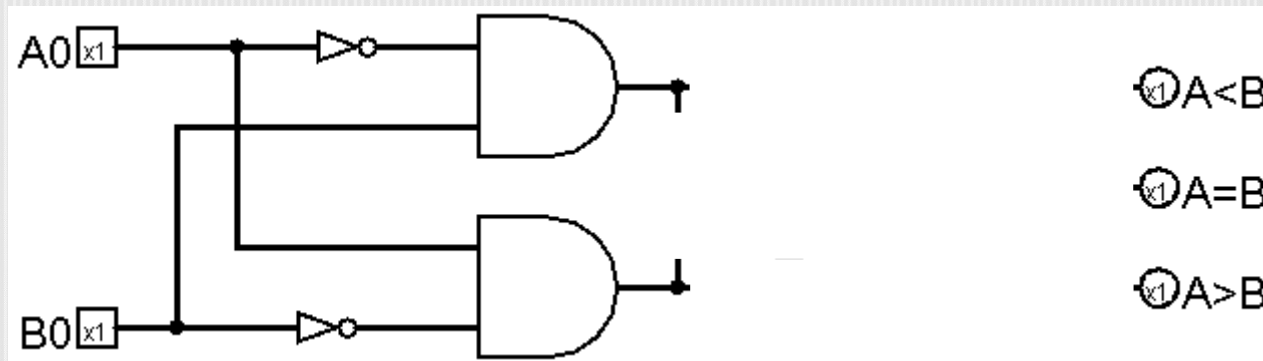
- Πολλαπλασιαστής 2×2 bit
- Πολλαπλασιαστής 4×2 bit
- Συγκριτής 1 bit
- Συγκριτής 2 bit
- Αποκωδικοποιητής 2×4
- Αποκωδικοποιητής 3×8
- Πολυπλέκτης 2 σε 1
- Πολυπλέκτης 4 σε 1

Συγκριτής 1 bit

- Να σχεδιάσετε το κύκλωμα του 1-bit συγκριτή.
- Πόσες εισόδους και εξόδους θα έχουμε;
- Ποιες από τις βασικές πύλες να χρησιμοποιήσουμε;
- Η αποκλειστική διάζευξη θα μπορούσε να χρησιμεύσει; Αν ναι σε τι;

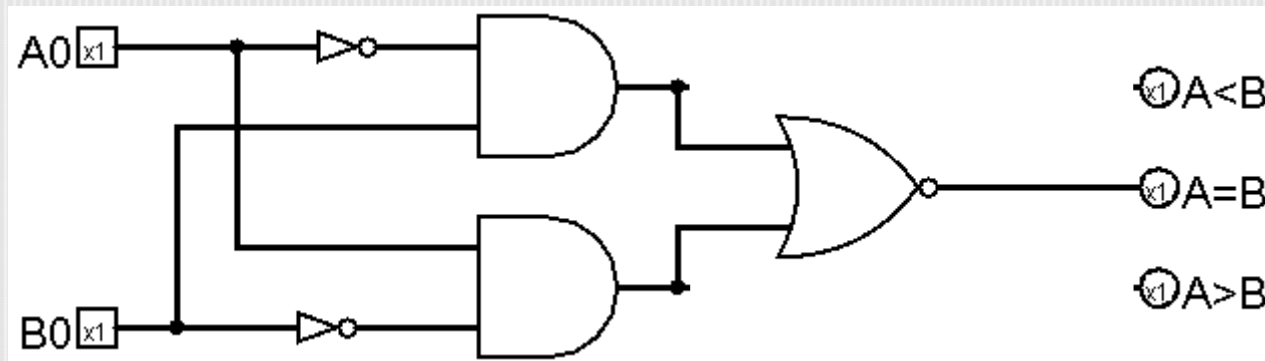
Συγκριτής 1 bit

1. Προσπαθείς να υλοποιήσεις την αποκλειστική διάζευξη ώστε να διαχωρίσεις τις ισότητες από τις ανισότητες.
2. Έπειτα προσπαθείς να διαχωρίσεις τις ανισότητες μεταξύ τους.



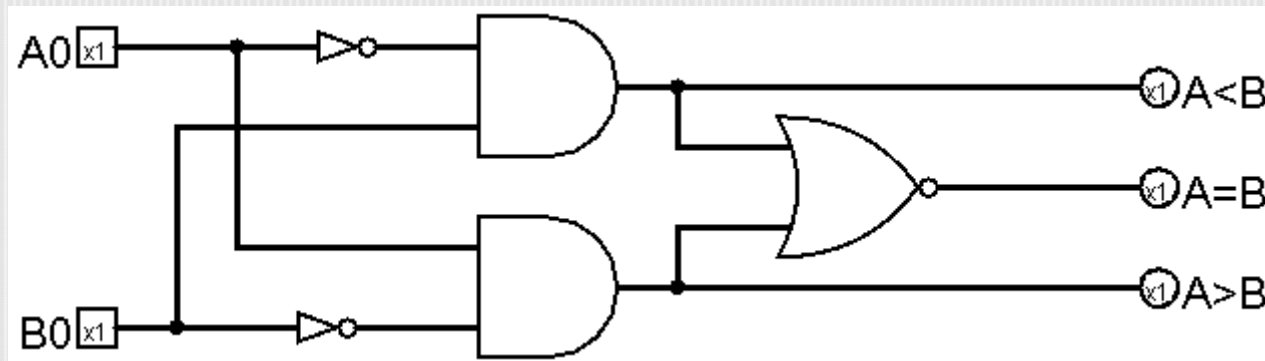
Συγκριτής 1 bit

1. Προσπαθείς να υλοποιήσεις την αποκλειστική διάζευξη ώστε να διαχωρίσεις τις ισότητες από τις ανισότητες.
2. Έπειτα προσπαθείς να διαχωρίσεις τις ανισότητες μεταξύ τους.



Συγκριτής 1 bit

1. Προσπαθείς να υλοποιήσεις την αποκλειστική διάζευξη ώστε να διαχωρίσεις τις ισότητες από τις ανισότητες.
2. Έπειτα προσπαθείς να διαχωρίσεις τις ανισότητες μεταξύ τους.



Συγκριτής 1 bit

1.

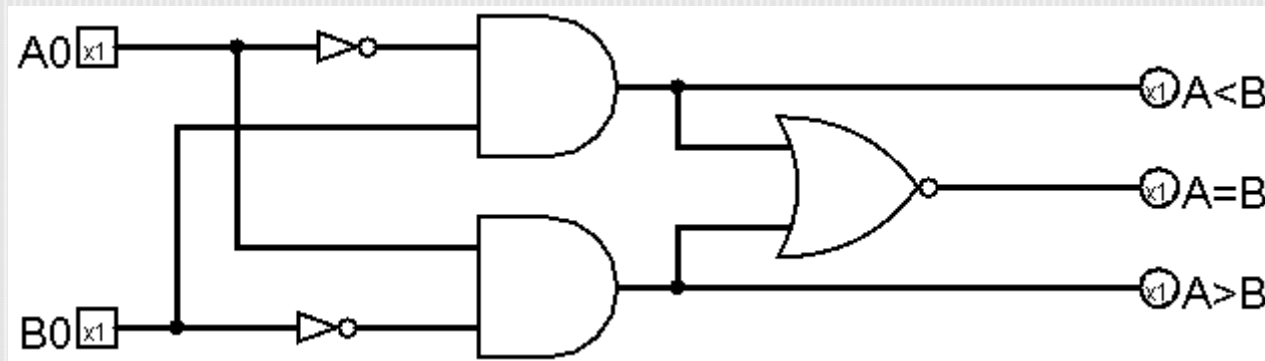
AND		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

2.

AND		
A'	B	A.B
1	0	0
1	1	1
0	0	0
0	1	0

AND		
A	B'	A.B
0	1	0
0	0	0
1	1	1
1	0	0

αντιστοιχείς μεταξύ τους.



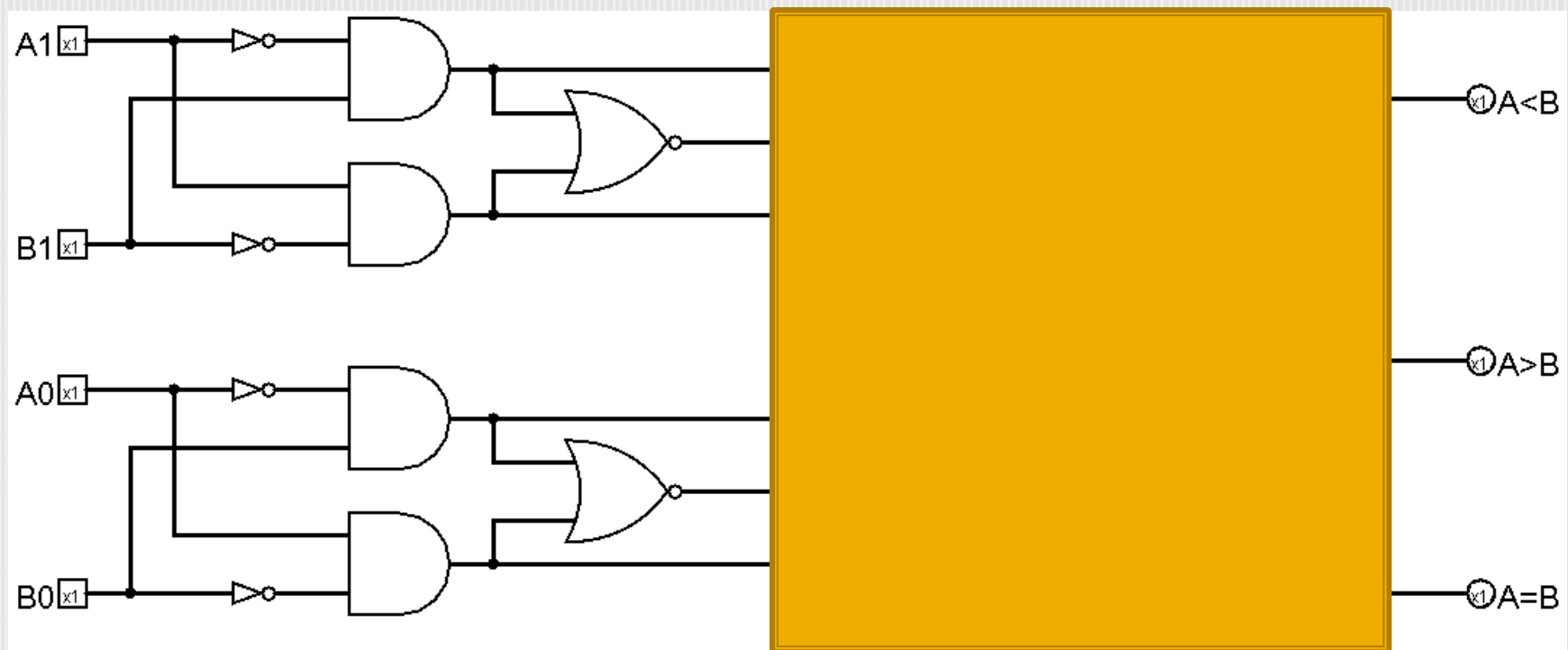
Συγκριτής 2 bit

- Να σχεδιαστεί το κύκλωμα του 2-bit συγκριτή.
- Χρησιμοποιήστε την ίδια λογική που χρησιμοποιούμε και στους δεκαδικούς...



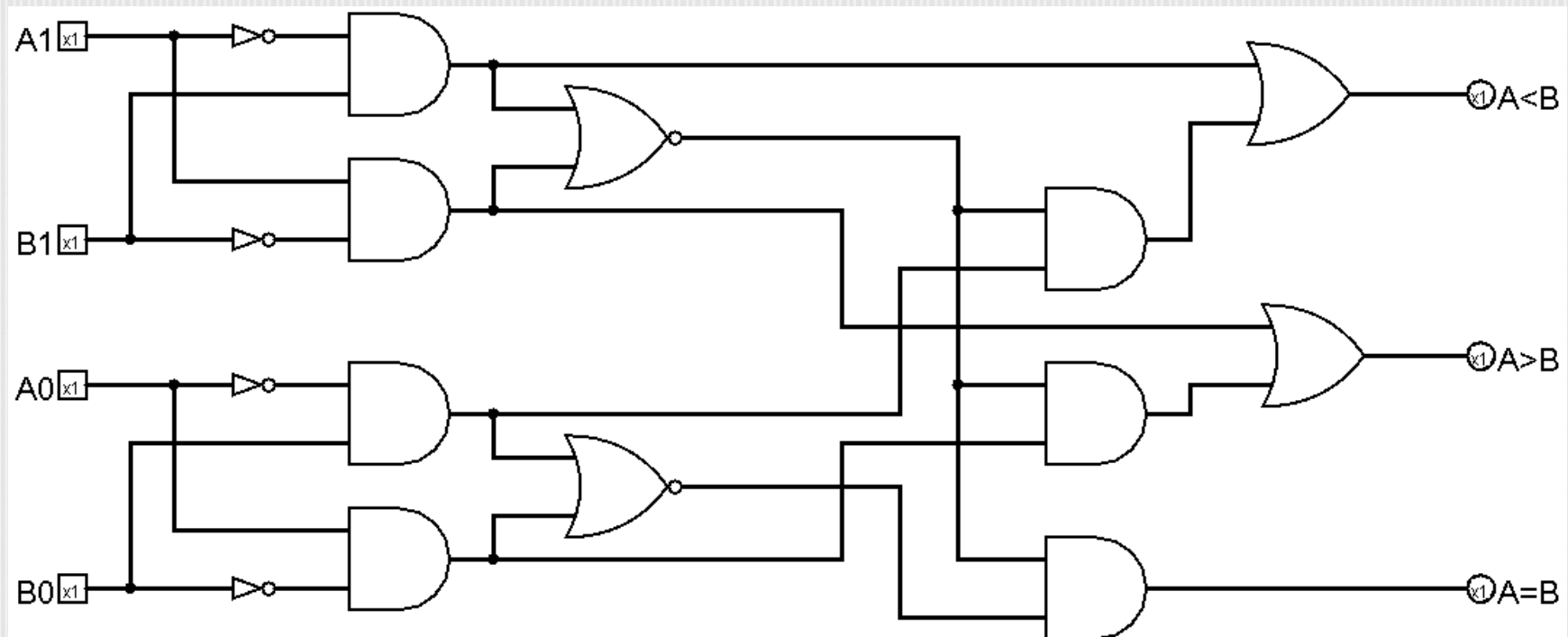
Συγκριτής 2 bit

- Να σχεδιαστεί το κύκλωμα του 2-bit συγκριτή.
- Χρησιμοποιήστε την ίδια λογική που χρησιμοποιούμε και στους δεκαδικούς...



Συγκριτής 2 bit

- Να σχεδιαστεί το κύκλωμα του 2-bit συγκριτή.
- Χρησιμοποιήστε την ίδια λογική που χρησιμοποιούμε και στους δεκαδικούς...

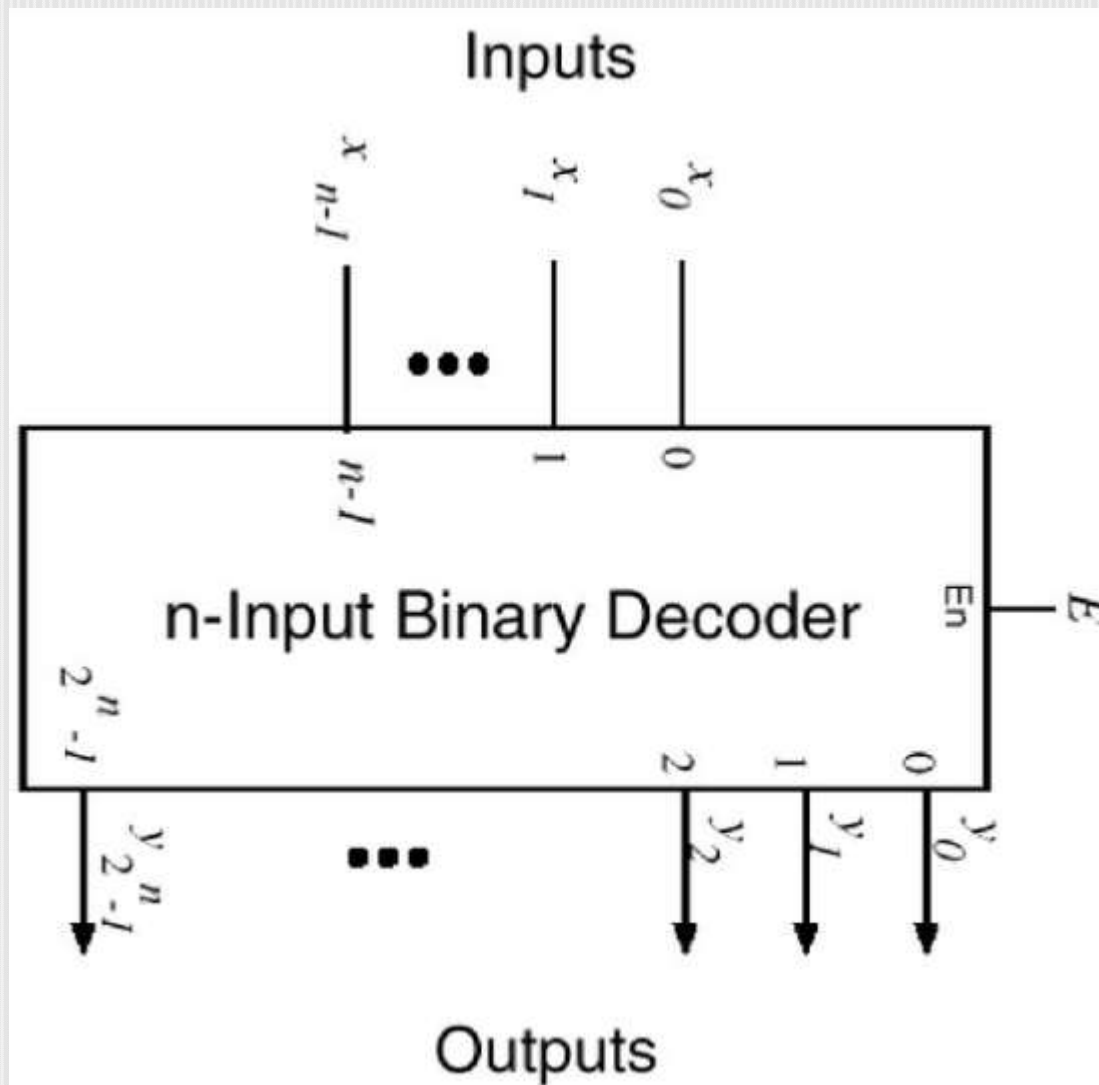


Περιεχόμενα

Συγκριτής - Αποκωδικοποιητής - Πολυπλέκτης *Comparator - Decoder (DEC) - Multiplexer (MUX)*

- Πολλαπλασιαστής 2×2 bit
- Πολλαπλασιαστής 4×2 bit
- Συγκριτής 1 bit
- Συγκριτής 2 bit
- Αποκωδικοποιητής 2×4
- Αποκωδικοποιητής 3×8
- Πολυπλέκτης 2 σε 1
- Πολυπλέκτης 4 σε 1

Κύκλωμα αποκωδικοποιητή

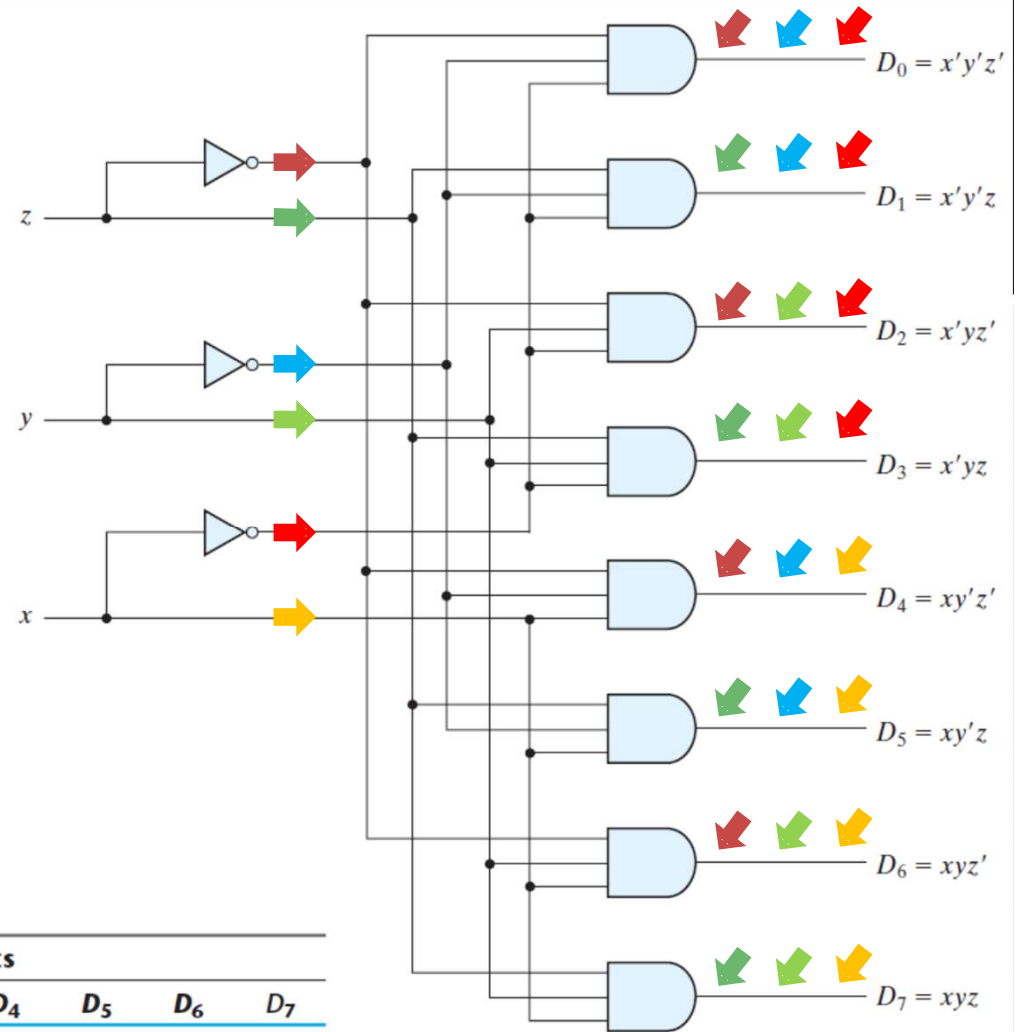


Αποκωδικοποιητής 3×8

Να σχεδιάσετε το κύκλωμα αποκωδικοποιητή 3 σε 8 με είσοδο επίτρεψης

- Να βρεθούν οι λογικές συναρτήσεις του κυκλώματος και να σχεδιαστή το κύκλωμα
- Να υλοποιηθεί το κύκλωμα στον προσομοιωτή και να επαληθευτεί η λειτουργία του

Αποκωδικοποιητής $3 \times 8 (=2^3)$



Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

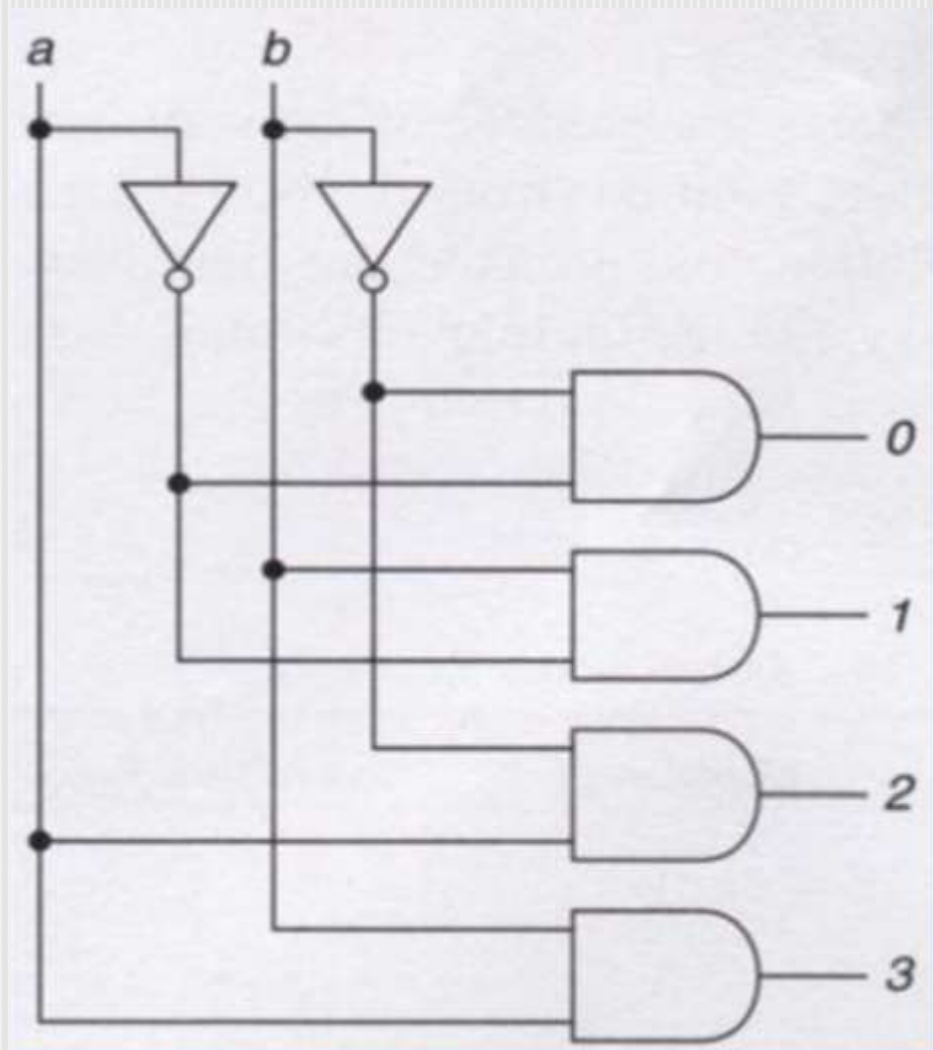
Αποκωδικοποιητής 2×4

Να σχεδιάσετε το κύκλωμα αποκωδικοποιητή 2 σε 4 με είσοδο επίτρεψης

- ❑ Να βρεθούν οι λογικές συναρτήσεις του κυκλώματος και να σχεδιαστή το κύκλωμα
- ❑ Να σχεδιαστή το κύκλωμα χρησιμοποιώντας πύλες NOR και NAND
- ❑ Να υλοποιηθεί το κύκλωμα στον προσομοιωτή και να επαληθευτεί η λειτουργία του
- ❑ Να υλοποιηθεί το κύκλωμα ως υποκύκλωμα

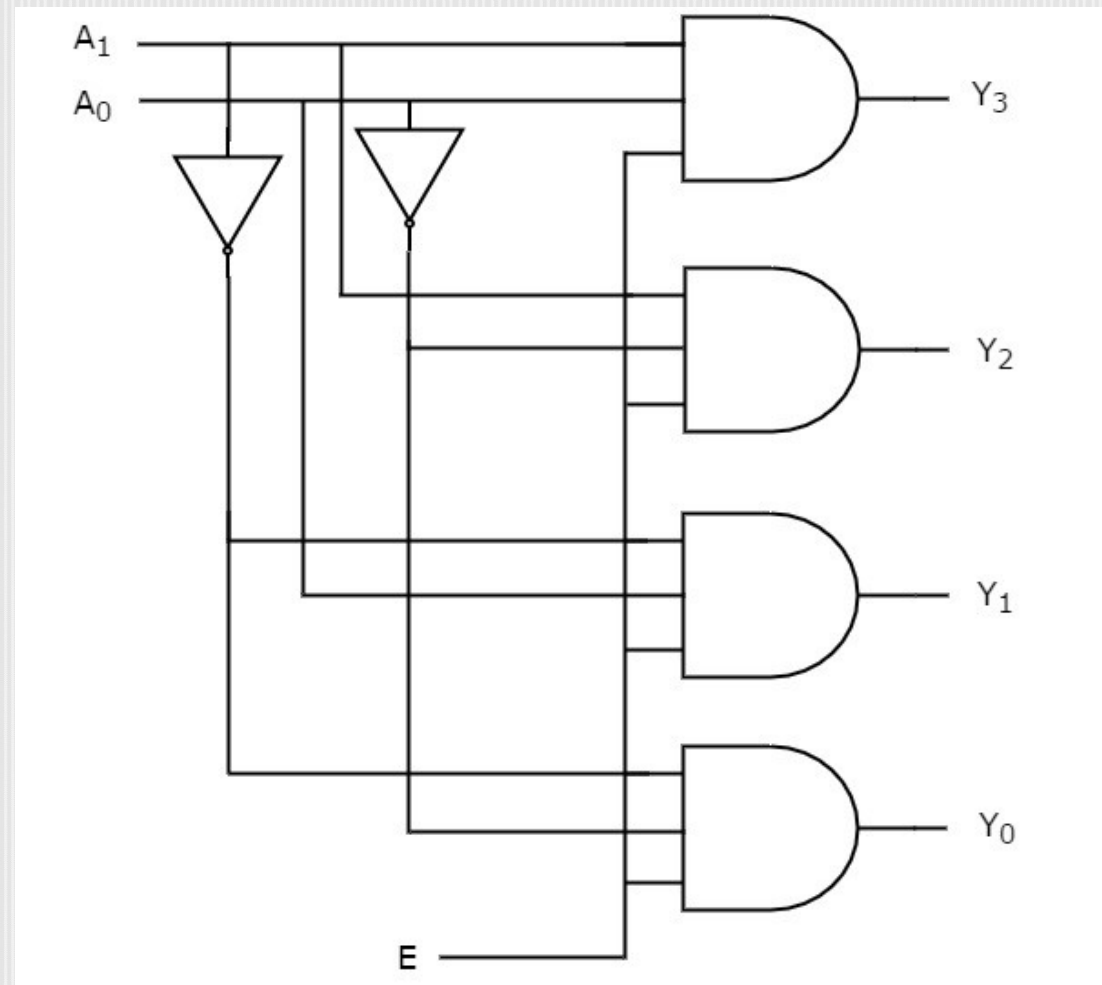
Αποκωδικοποιητής 2×4

<i>a</i>	<i>b</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



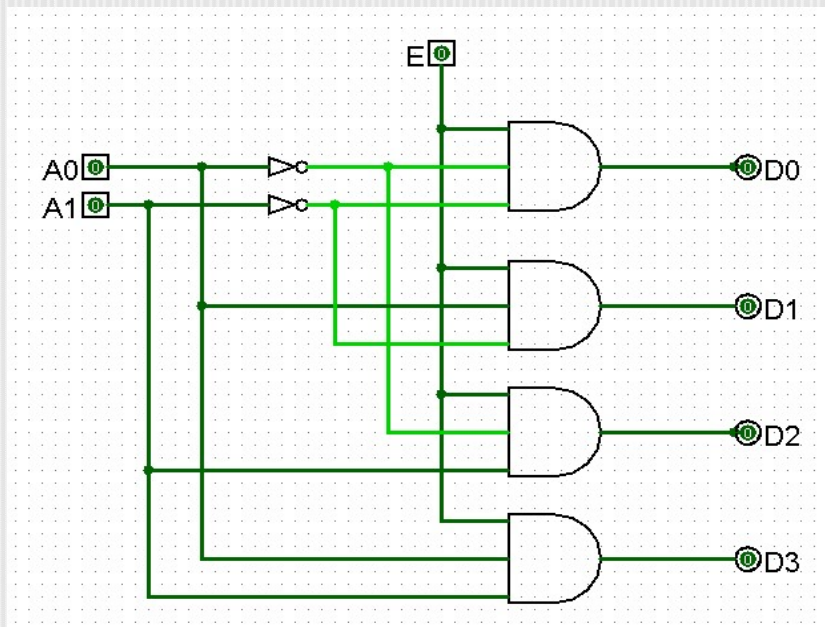
Αποκωδικοποιητής 2×4 με είσοδο επέτρσης (enable line)

Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	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



Αποκωδικοποιητής 2×4

Κανονικός vs Συμπληρωματικός με είσοδο επίτρεψης

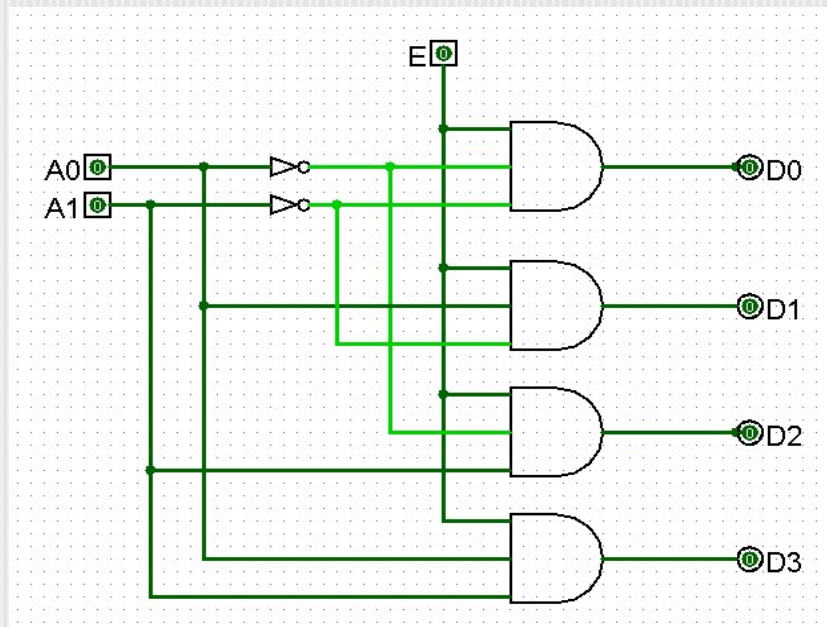


Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

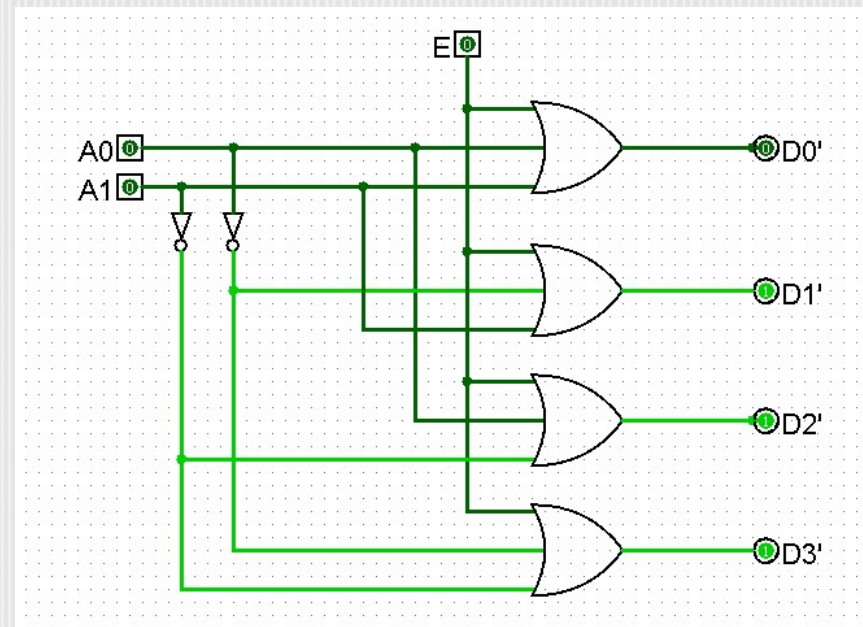
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1

Αποκωδικοποιητής 2×4

Κανονικός vs Συμπληρωματικός με είσοδο επίτρησης



Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	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

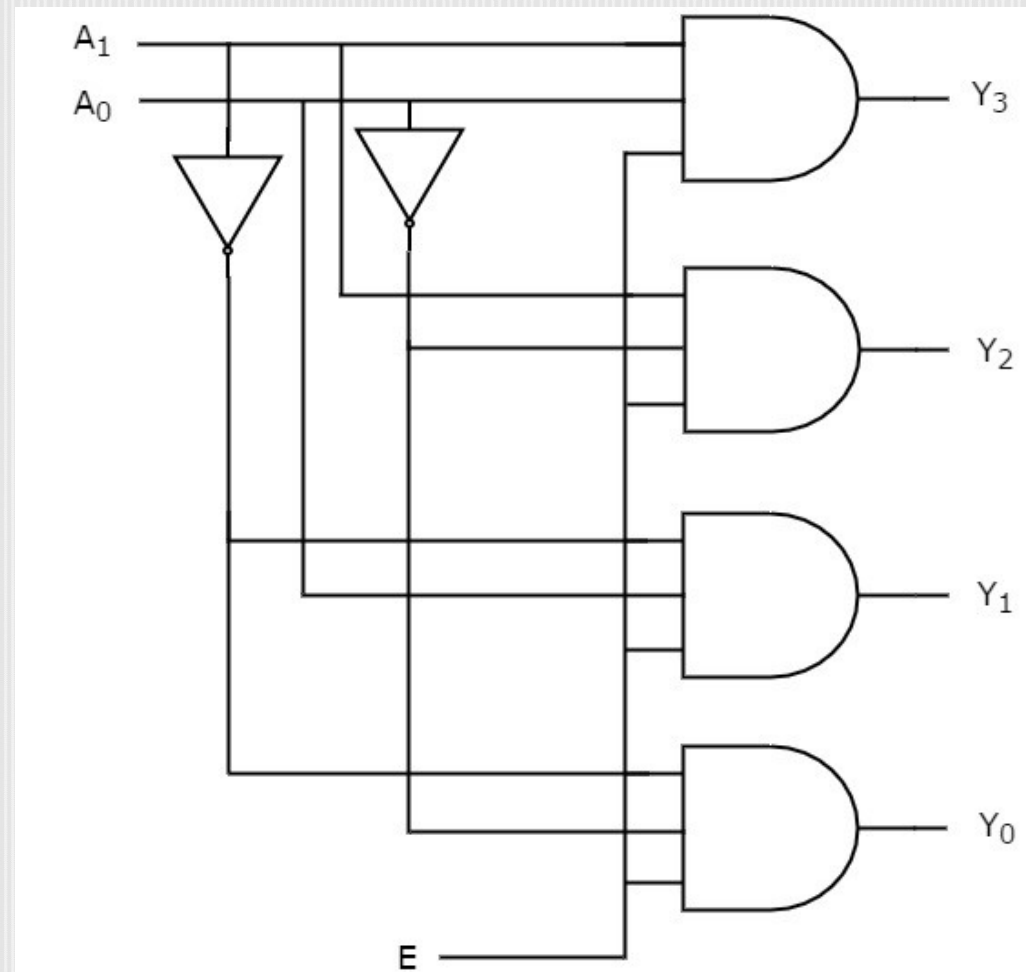
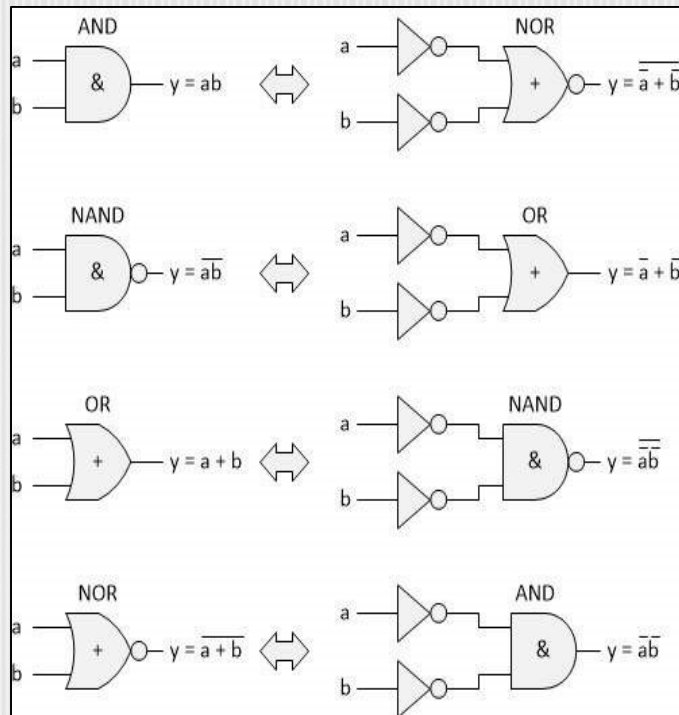


E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1

Αποκωδικοποιητής 2×4

Βρες την υλοποίηση με NAND και NOR

Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	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



Πολυπλέκτης 2 σε 1

Χρήσιμες ισοδυναμίες για μετατροπή
σε υλοποιήσεις με NAND και NOR

In [electrical](#) and [computer engineering](#), De Morgan's laws are commonly written as:

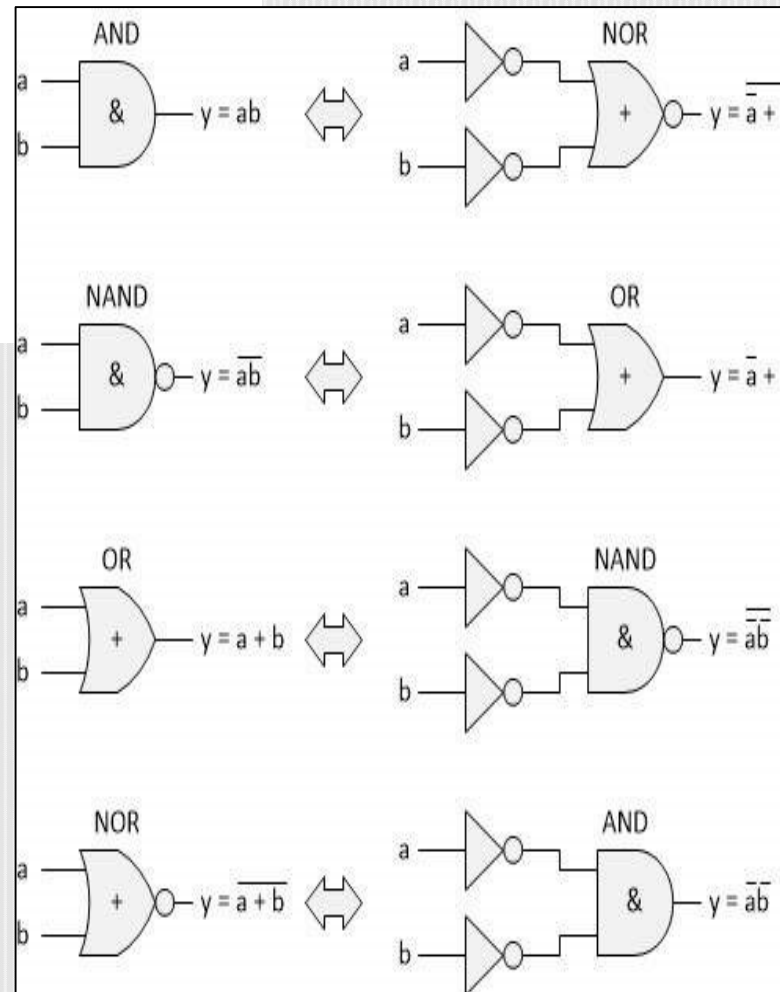
$$\overline{A \cdot B} \equiv \overline{A} + \overline{B}$$

and

$$\overline{A + B} \equiv \overline{A} \cdot \overline{B},$$

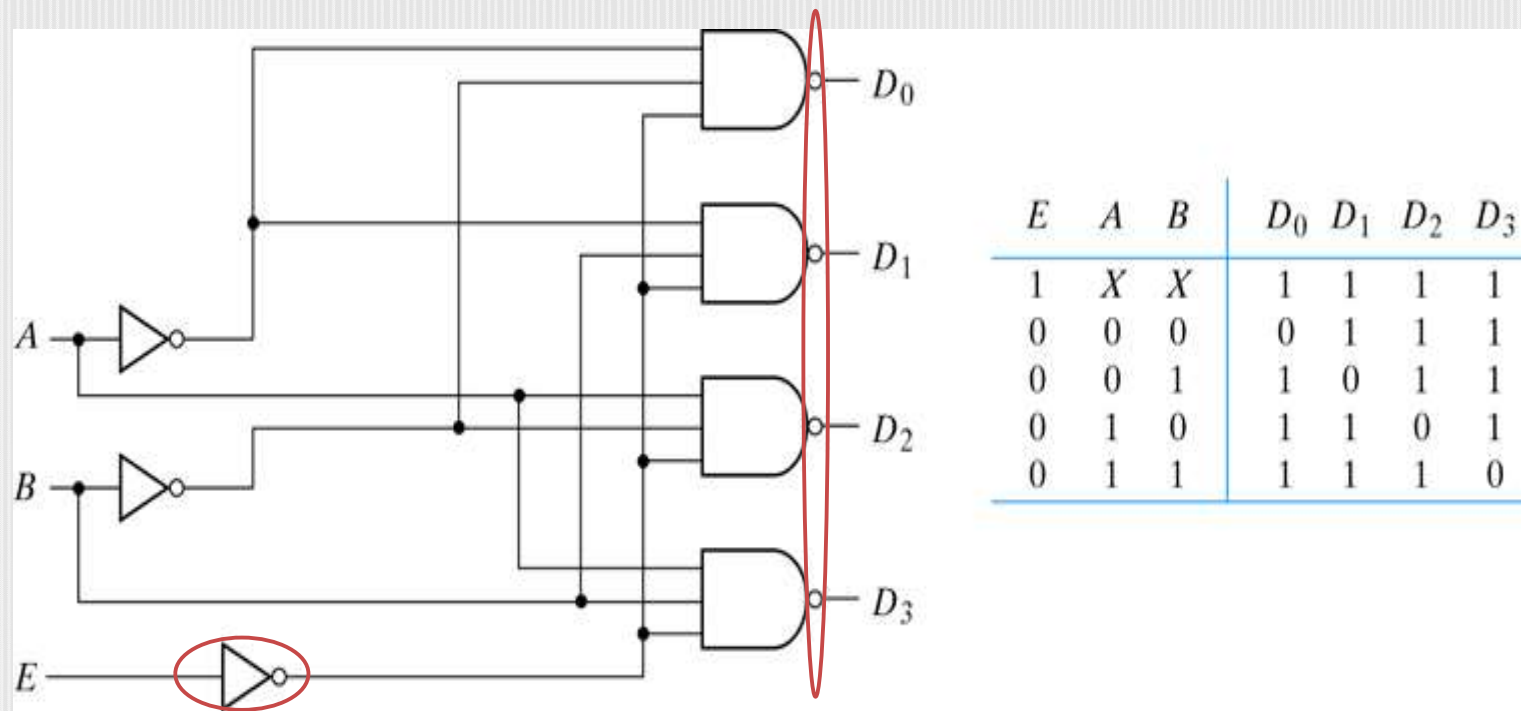
where:

- \cdot is the logical AND,
- $+$ is the logical OR,



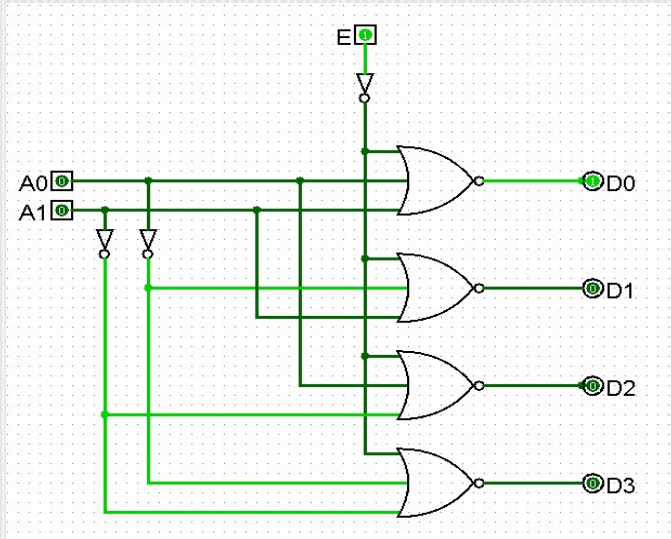
Αποκωδικοποιητής 2×4

Υλοποιημένος με NAND (συμπληρωματικός)

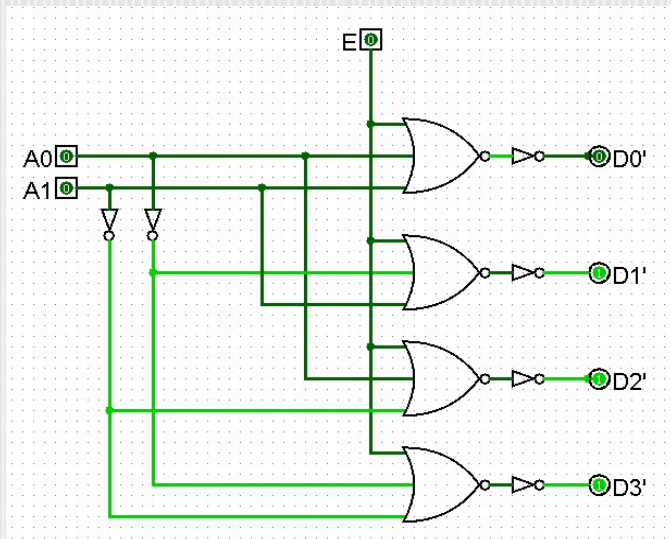


Αποκωδικοποιητής 2×4

Υλοποιημένος με NOR



Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	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



Συμπληρωματικός

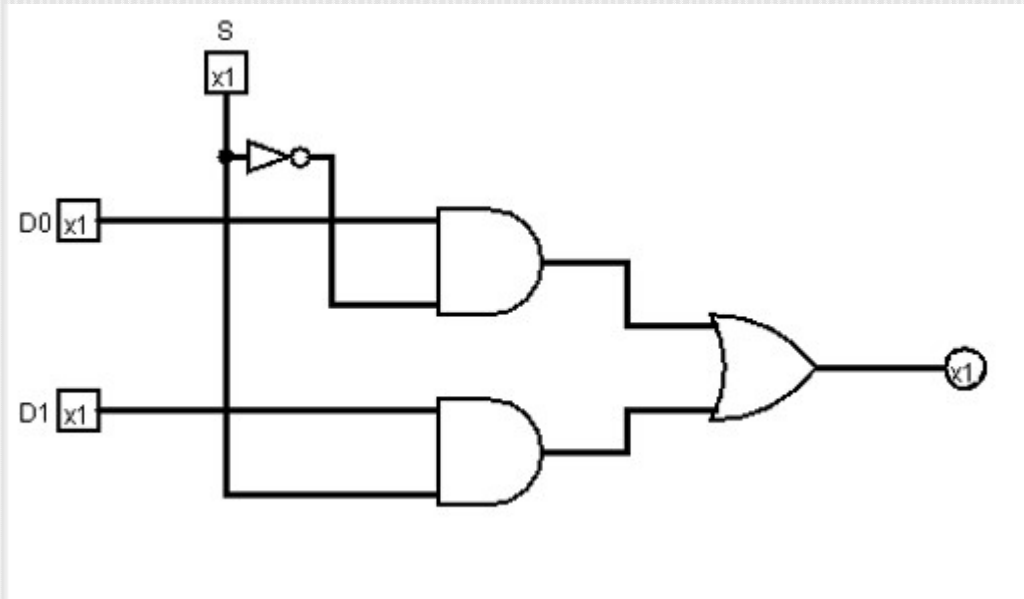
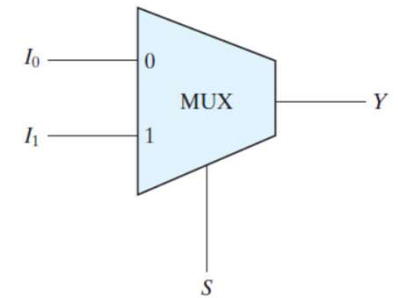
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Περιεχόμενα

Συγκριτής - Αποκωδικοποιητής - Πολυπλέκτης *Comparator - Decoder (DEC) - Multiplexer (MUX)*

- Πολλαπλασιαστής 2×2 bit
- Πολλαπλασιαστής 4×2 bit
- Συγκριτής 1 bit
- Συγκριτής 2 bit
- Αποκωδικοποιητής 2×4
- Αποκωδικοποιητής 3×8
- Πολυπλέκτης 2 σε 1
- Πολυπλέκτης 4 σε 1

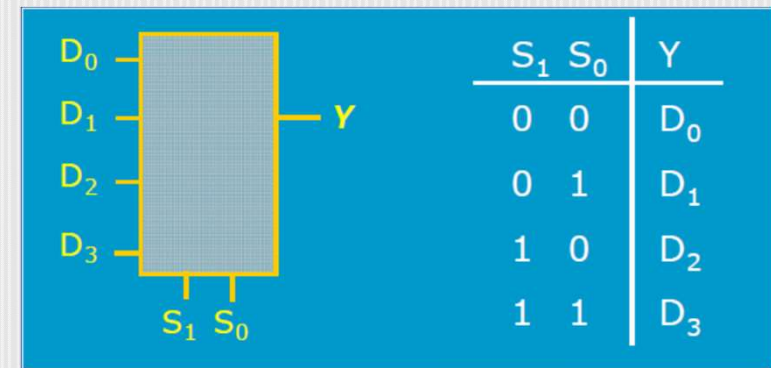
Πολυπλέκτης 2 σε 1



Πολυπλέκτης 4 σε 1

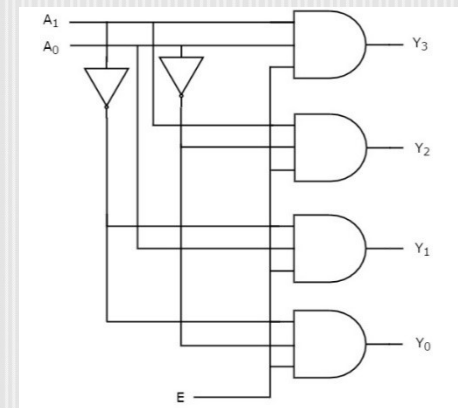
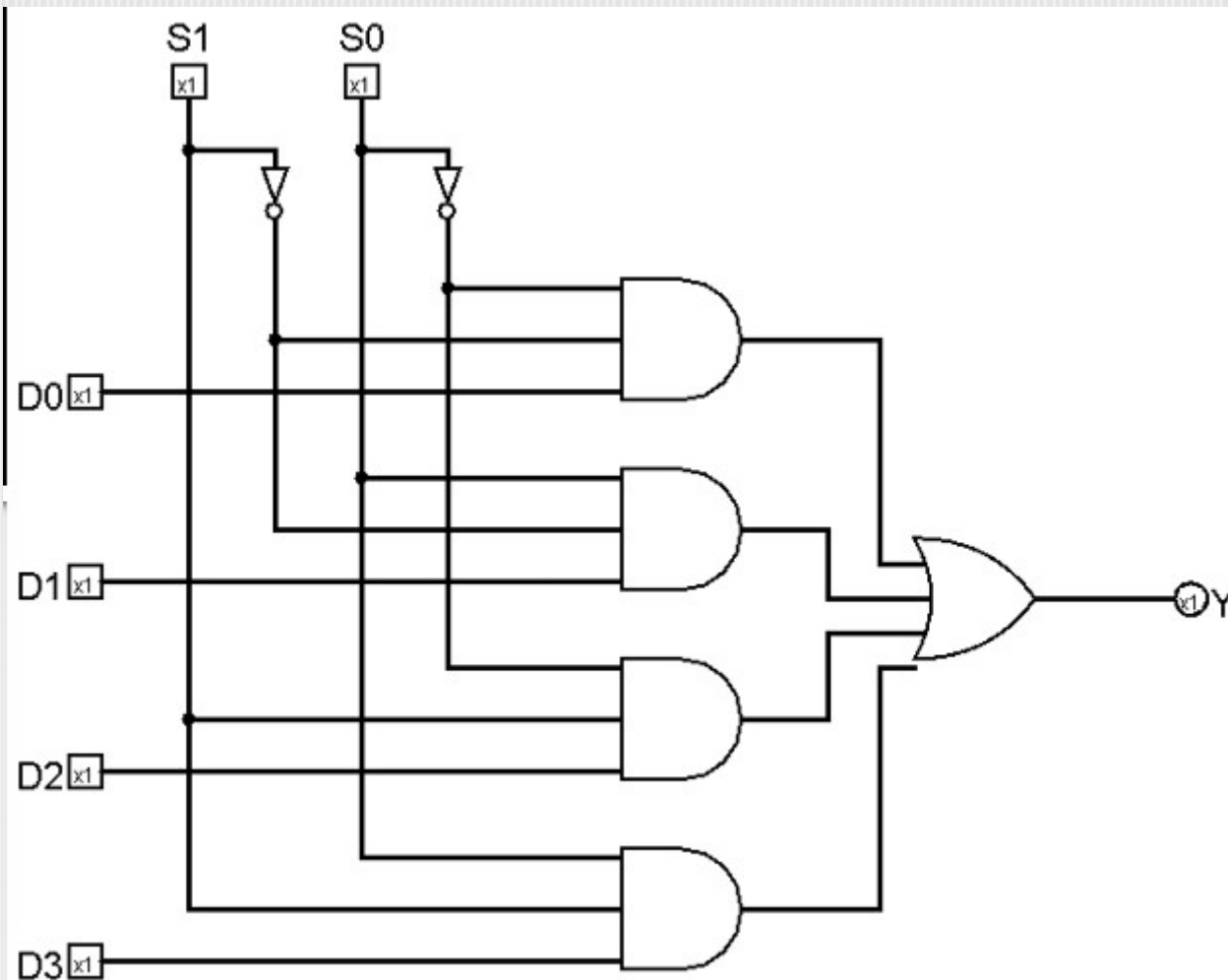
Να υλοποιηθεί το κύκλωμα πολυπλέκτη 4 σε 1

- Να βρεθεί η λογική συνάρτηση του κυκλώματος και να σχεδιαστεί το κύκλωμα
- Να σχεδιασθή το κύκλωμα χρησιμοποιώντας πύλες NOR.
- Να υλοποιηθεί το κύκλωμα στον προσομοιωτή και να επαληθευτεί η λειτουργία του
- Να υλοποιηθεί το κύκλωμα ως υποκύκλωμα (chip)



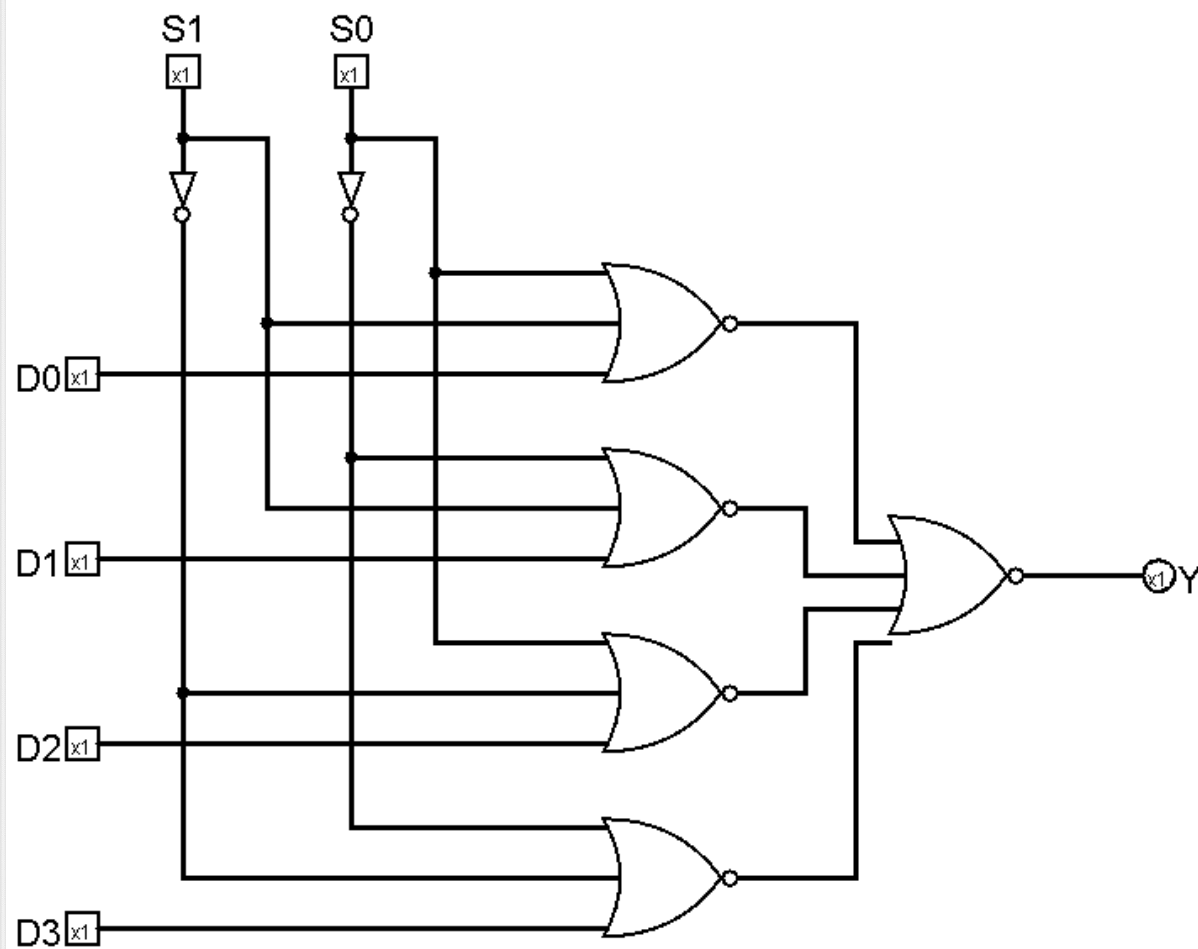
Πολυπλέκτης 4 σε 1

Αντιπαράθεση με αποκωδικοποιητή 2×4



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

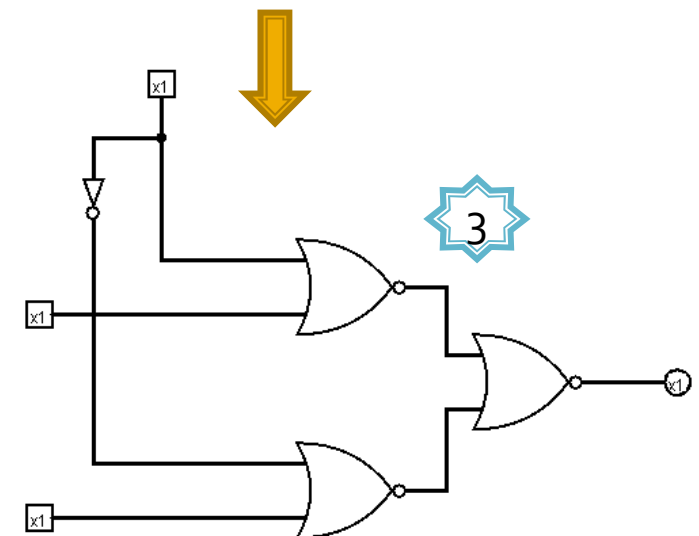
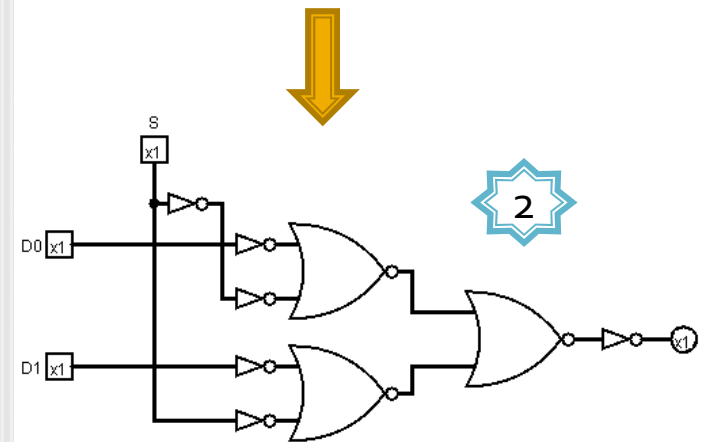
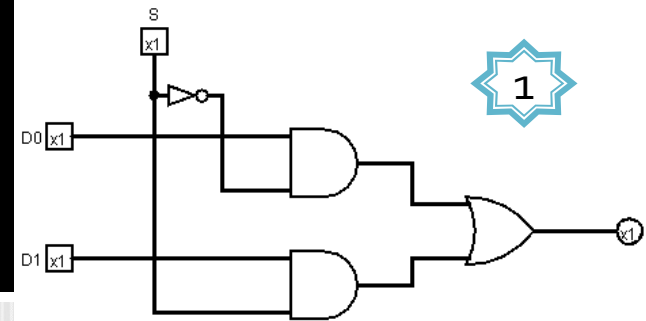
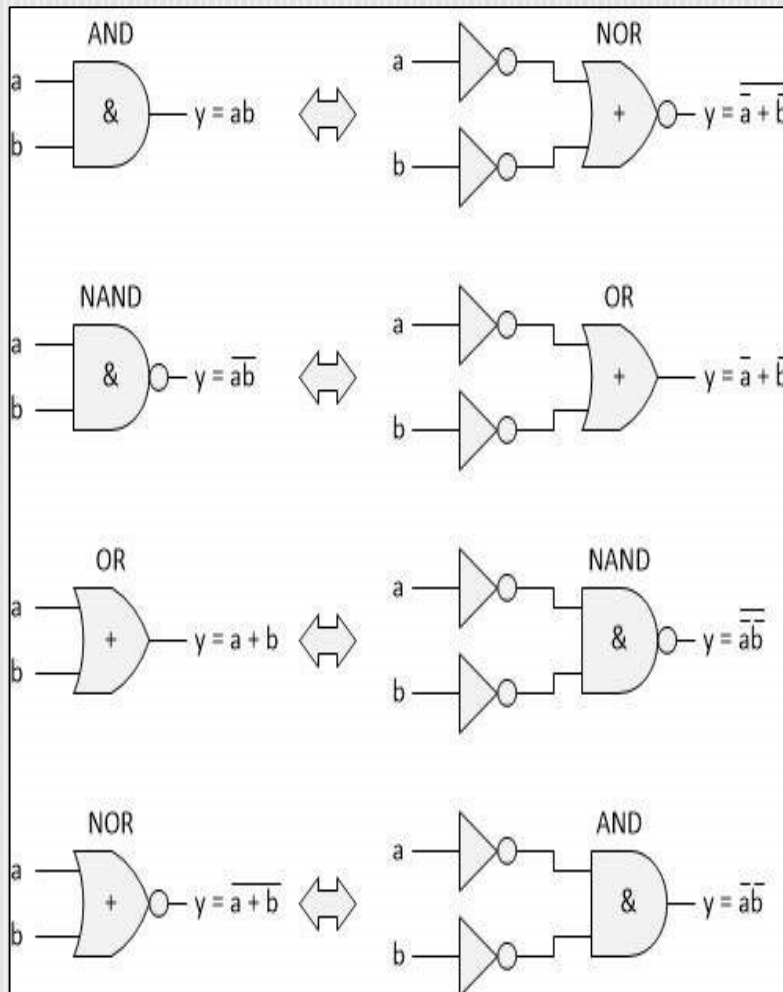
Πολυπλέκτης 4 σε 1 με NOR



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Πολυπλέκτης 2 σε 1

Χρήσιμες ισοδυναμίες για μετατροπή
σε υλοποιήσεις με NAND και NOR



Ασκήσεις για υποβολή

3^ο εργαστήριο

Παραδοτέα 2 αρχεία:

1. αρχείο κειμένου με σχήματα (κατά προτίμηση .pdf)
2. αρχείο .circ

Να σχεδιάσετε και να υλοποιήσετε στον εξομοιωτή τα παρακάτω κυκλώματα:

- Να σχεδιαστή κύκλωμα δυαδικού πολλαπλασιαστή 4×4 .
- Να σχεδιαστεί κύκλωμα δυαδικού συγκριτής 4×4 .
- Να σχεδιάσετε και να υλοποιήσετε κύκλωμα πολυπλέκτη 2 σε 1 με είσοδο επίτρεψης, όπου κάθε μια από τις εισόδους/εξόδους θα είναι των 8 bit .
*(σε κανονική λειτουργία το σύστημα είναι απενεργοποιημένο οπότε για $E=1$, όλες οι εξοδοι 0)
- Να σχεδιάσετε και να υλοποιήσετε στον προσομοιωτή έναν 4×16 αποκωδικοποιητή χρησιμοποιώντας δυο 3×8 αποκωδικοποιητές.

Υποχρεωτική υποβολή με αξιολόγηση

ΤΕΛΟΣ

3^{ου} εργαστηρίου

Ερωτήσεις;