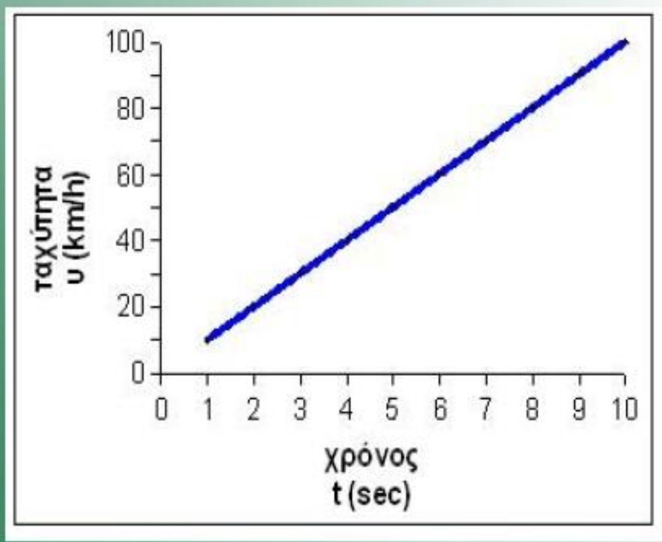
Αναλογικά & ψηφιακά ηλεκτρονικά (Θεωρία)

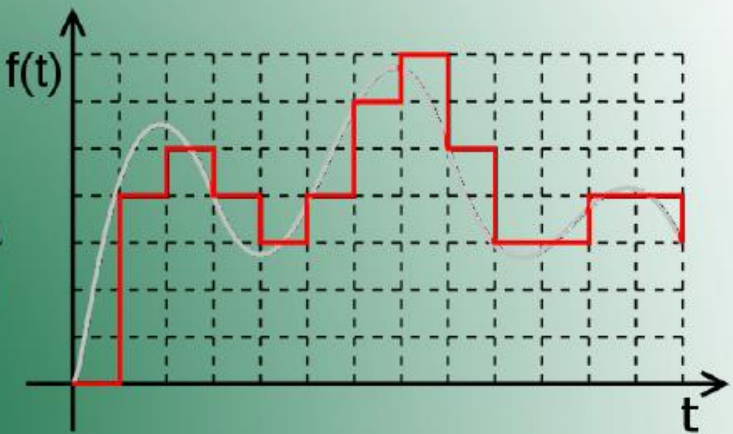
Αναλογικά μεγέθη

Αναλογικό μέγεθος ονομάζεται εκείνο που μπορεί να πάρει <u>οποιαδήποτε τιμή</u> σε μια περιοχή τιμών, όπως η ταχύτητα, το βάρος, η θερμοκρασία, το ύψος, κτλ.



Ψηφιακά μεγέθη

Ψηφιακό μέγεθος ονομάζεται εκείνο που μπορεί να πάρει <u>συγκεκριμένες (διακριτές) τιμές</u> σε μια περιοχή τιμών, όπως το πλήθος των φάουλ σε ένα αγώνα, τα δευτερόλεπτα, κτλ.



Δυαδικά μεγέθη

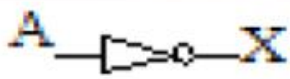
Δυαδικό μέγεθος ονομάζεται το ψηφιακό μέγεθος που μπορεί να πάρει μόνο δύο διακριτές τιμές, όπως είναι η κατάσταση μιας λάμπας (σβηστή-αναμμένη), η κατάσταση ενός διακόπτη (ανοικτός-κλειστός), κτλ.

Βασικές λογικές πράξεις και πύλες (NOT, AND, OR)

Λογική πράξη NOT – πύλη NOT

Το αποτέλεσμα της πράξης NOT είναι το συμπλήρωμα της αρχικής μεταβλητής.

ΣΗΜ: Το συμπλήρωμα του 0 είναι το 1, ενώ το συμπλήρωμα του 1 είναι το 0.



$$X = \bar{A}$$

A	X
0	1
1	0

Λογική πράξη AND – πύλη AND

Το αποτέλεσμα της πράξης AND είναι “1”, αν και οι δύο μεταβλητές είναι “1”.

$X = A \cdot B$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Λογική πράξη OR – πύλη OR

Το αποτέλεσμα της πράξης OR είναι “1”, αν τουλάχιστον μια από τις δύο μεταβλητές είναι “1”.

$X = A + B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	$\bar{A}$
0	1
1	0

AND

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Αξιώματα Huntington

Αξιώματα Huntington

- **Ουδέτερα στοιχεία**

  $x \cdot 1 = 1 \cdot x = x$ $\quad$ $x + 0 = 0 + x = x$

- **Αντιμεταθετική ιδιότητα**

  $x \bullet y = y \bullet x$ $\quad$ $x + y = y + x$

- **Επιμεριστική ιδιότητα**

  $x \bullet (y+z) = (x \bullet y) + (x \bullet z)$ $\quad$ $x + (y \bullet z) = (x+y) \bullet (x+z)$

- **Συμπλήρωμα**

  $\bar{A} \bullet A = 0$ $\quad$ $A + \bar{A} = 1$

Άλγεβρα Boole

Θεωρήματα άλγεβρας Boole

$x \bullet x = x$ $\qquad$ $x + x = x$

$x \bullet 0 = 0$ $\qquad$ $x + 1 = x$

$x = \overline{\overline{x}}$

$x \bullet y \bullet z = x \bullet (y \bullet z) = (x \bullet y) \bullet z$ $\qquad$ $x + y + z = x + (y + z) = (x + y) + z$

$x + x \bullet y = x$ $\qquad$ $x \bullet (x + y) = x$

- Θεώρημα Morgan

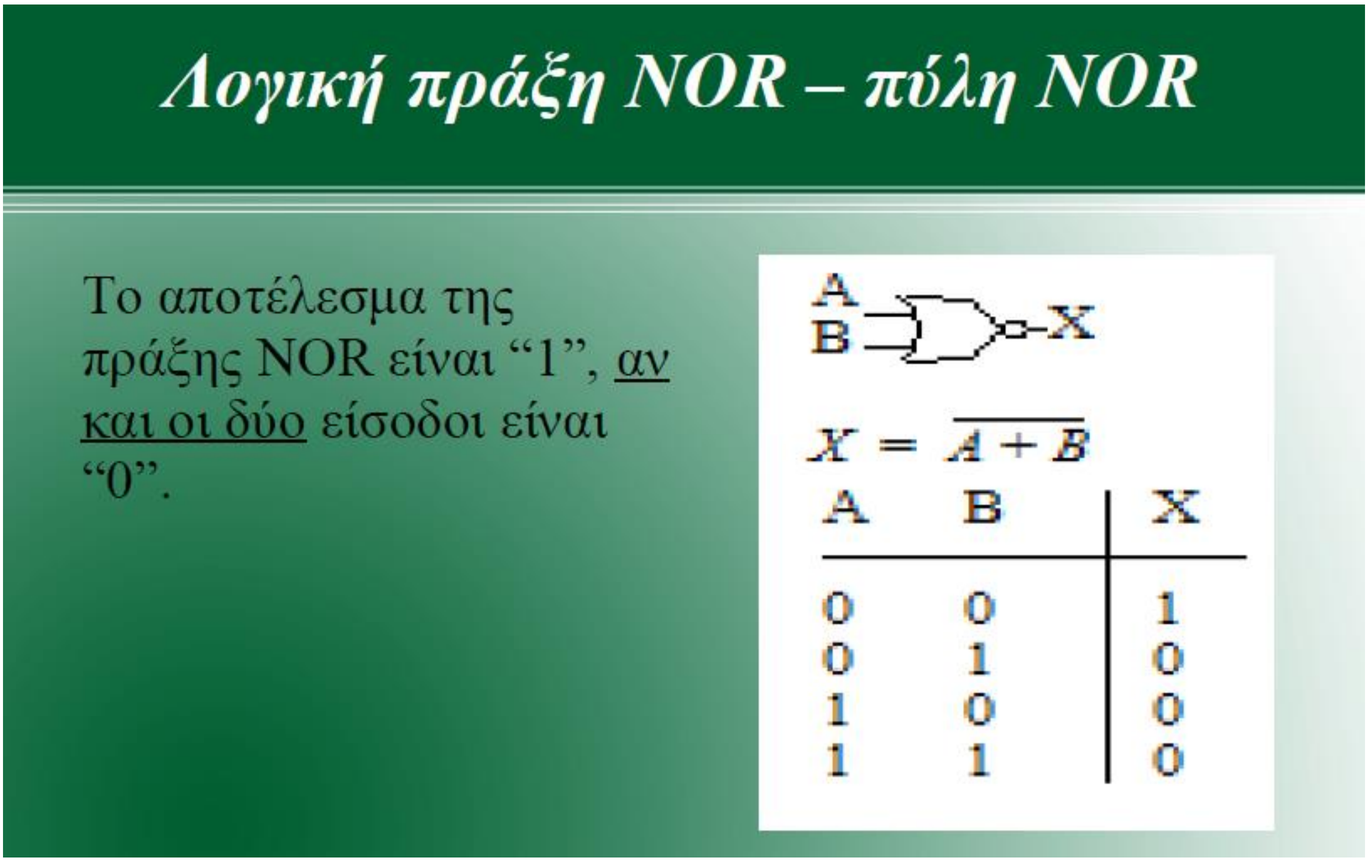
$\overline{x \bullet y} = \overline{x} + \overline{y}$ $\qquad$ $\overline{x + y} = \overline{x} \bullet \overline{y}$

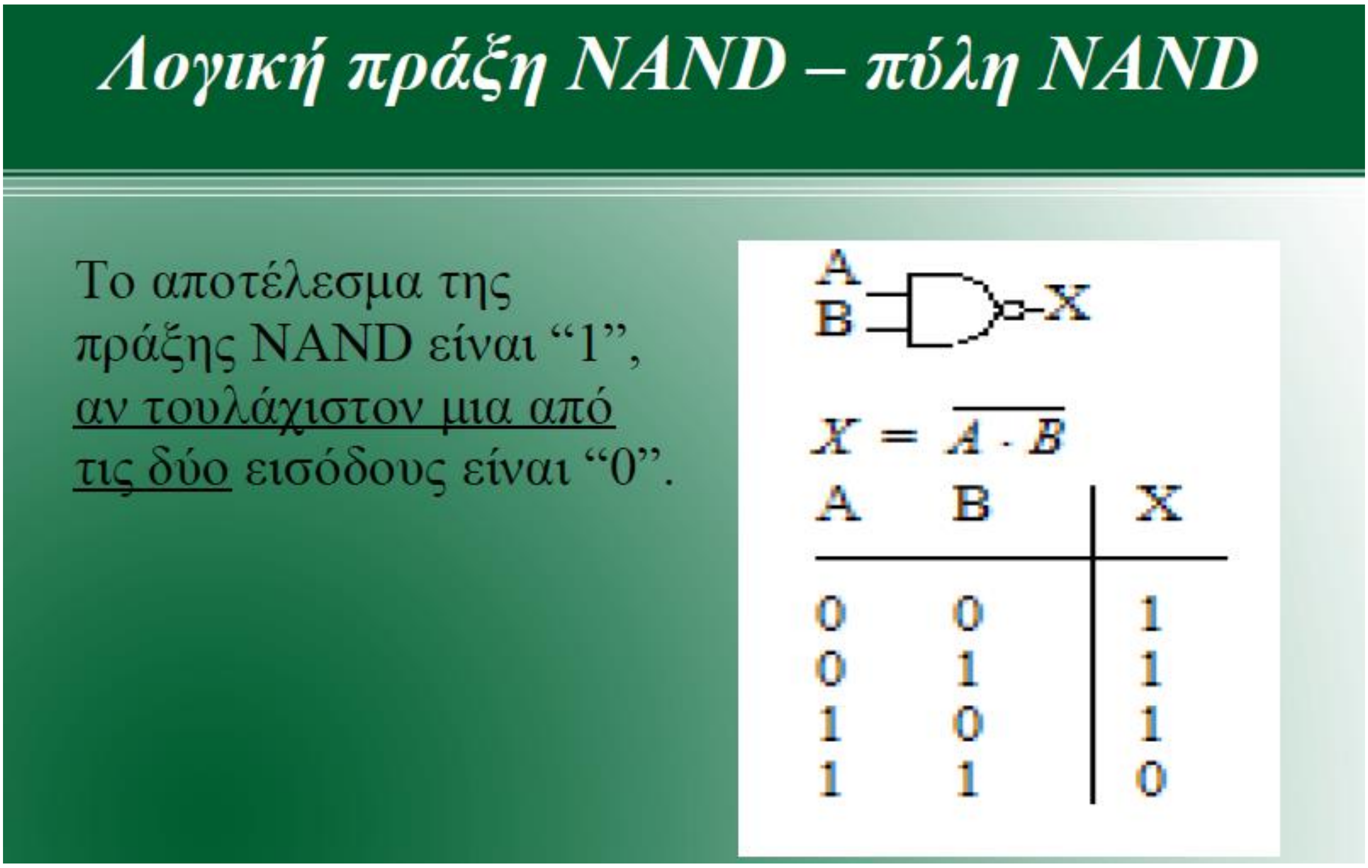
Προτεραιότητα πράξεων

Σε μια έκφραση της άλγεβρας Boole εκτελούνται πρώτα οι πράξεις μέσα στις παρενθέσεις, μετά υπολογίζονται τα συμπληρώματα, στη συνέχεια εκτελούνται οι πράξεις AND και τέλος εκτελούνται οι πράξεις OR.

Προτεραιότητα	Πράξη
1	( )
2	NOT
3	AND
4	OR

Λογικές πύλες NOR, NAND

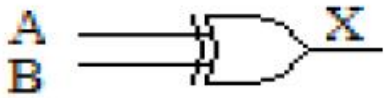




Λογικές πύλες XOR, XNOR

Λογική πράξη XOR – πύλη XOR

Το αποτέλεσμα της πράξης XOR είναι “1”, <u>αν οι δύο είσοδοι είναι διαφορετικές</u> μεταξύ τους.



$$X = A \oplus B$$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Λογική πράξη XNOR – πύλη XNOR

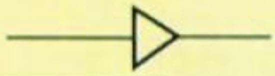
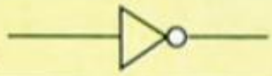
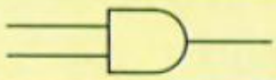
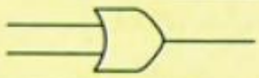
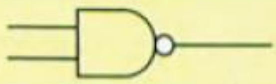
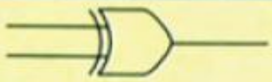
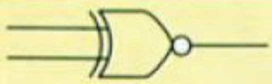
Το αποτέλεσμα της πράξης XNOR είναι “1”, <u>αν οι δύο είσοδοι είναι ίσες</u> μεταξύ τους.

$$X = A \odot B$$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Λογικές πύλες (συνοπτικοί πίνακες)

Λογική Πύλη	Είσοδοι	Έξοδος	Συνάρτηση
Απομονωτής Buffer	A	Y	$Y=A$
Αντιστροφέας NOT	A	Y	$Y=\overline{A}$
AND	A,B	Y	$Y=A\cdot B$
OR	A,B	Y	$Y=A+B$
NAND	A,B	Y	$Y=\overline{A\cdot B}$
NOR	A,B	Y	$Y=\overline{A+B}$
XOR	A,B	Y	$Y=A\cdot\overline{B}+\overline{A}\cdot B=A\oplus B$
XNOR	A,B	Y	$Y=A\cdot B+\overline{A}\cdot\overline{B}=\overline{A\oplus B}=A\odot B$

Λογική Πύλη	Λογικό Διάγραμμα
Απομονωτής Buffer	A  $Y=A$
Αντιστροφέας NOT	A  $Y=\overline{A}$
AND	A B  $Y=A\cdot B$
OR	A B  $Y=A+B$
NAND	A B  $Y=\overline{A\cdot B}$
NOR	A B  $Y=\overline{A+B}$
XOR	A B  $Y=A\oplus B$
XNOR	A B  $Y=A\odot B$

BUFFER

A	$Y=A$
0	0
1	1

NOT

A	$Y=\overline{A}$
0	1
1	0

AND

A	B	$Y=A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

NAND

A	B	$Y=\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	$Y=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

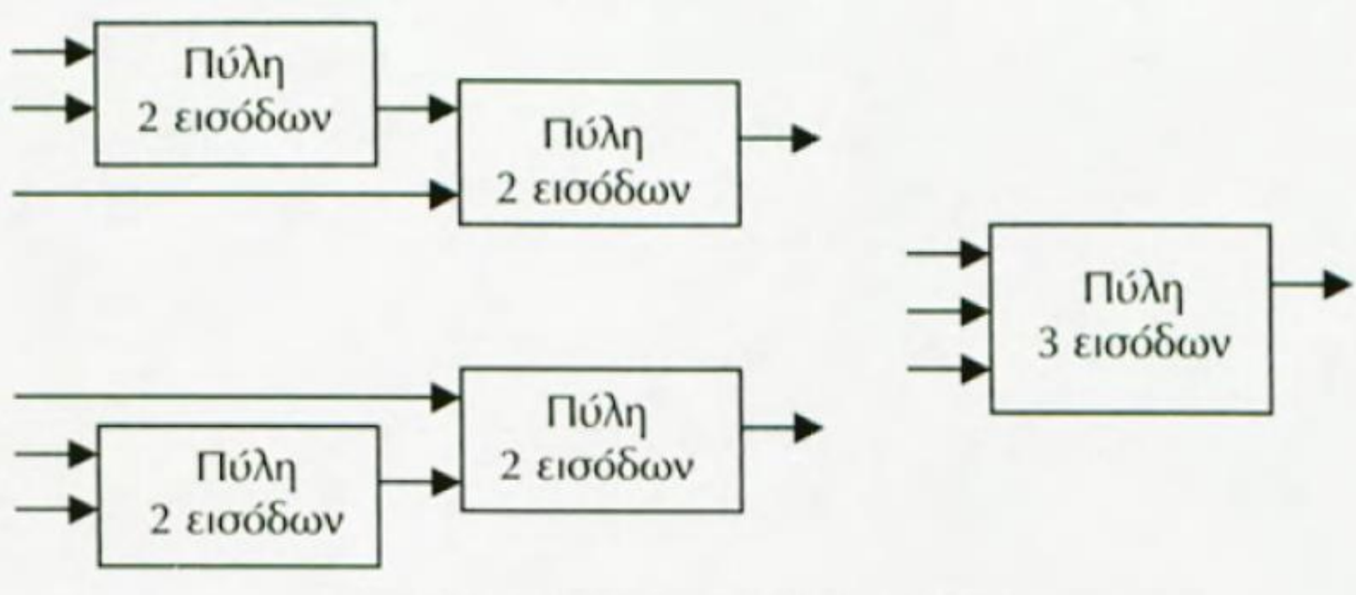
XOR

A	B	$Y=A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

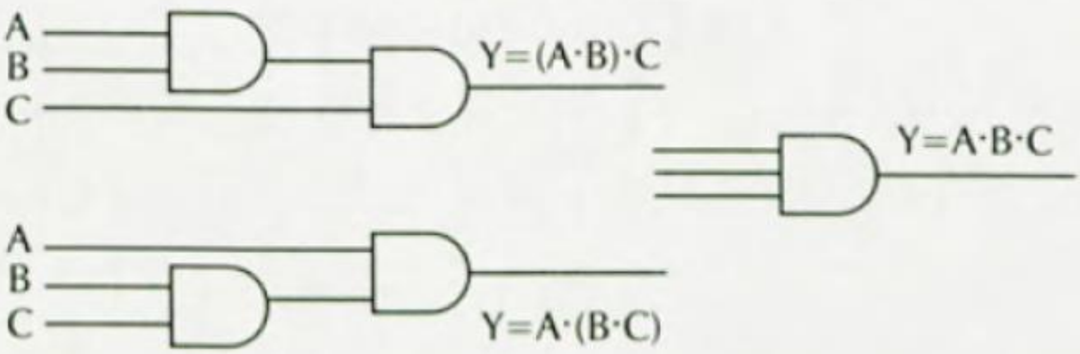
A	B	$Y=A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Λογικές πύλες πολλαπλών εισόδων

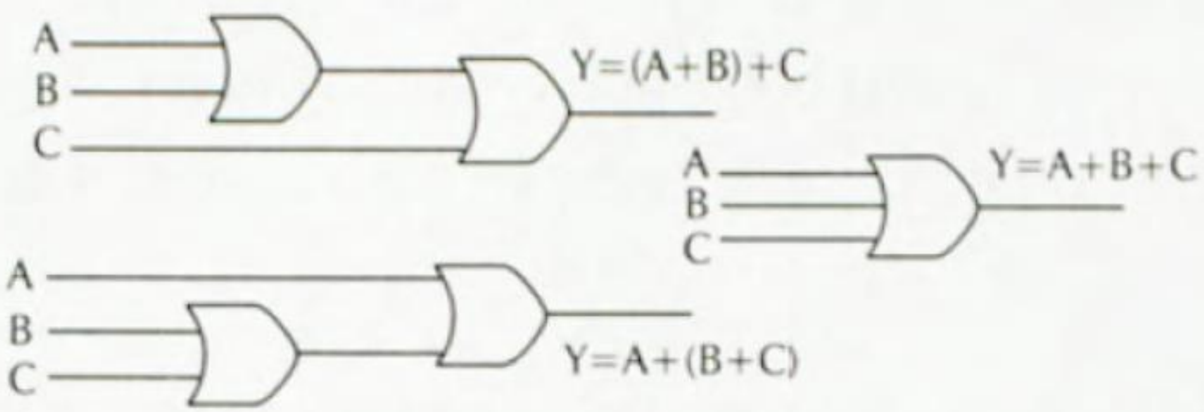


Τεχνική επέκτασης εισόδων πυλών

$$Y=A\cdot B\cdot C=(A\cdot B)\cdot C=A\cdot(B\cdot C)$$



Υλοποίηση πύλης AND τριών εισόδων με πύλες AND δύο εισόδων



Υλοποίηση πύλης OR τριών εισόδων με πύλες OR δυο εισόδων

Φύλλα δεδομένων (Datasheet)

Φύλλα δεδομένων (Datasheet)

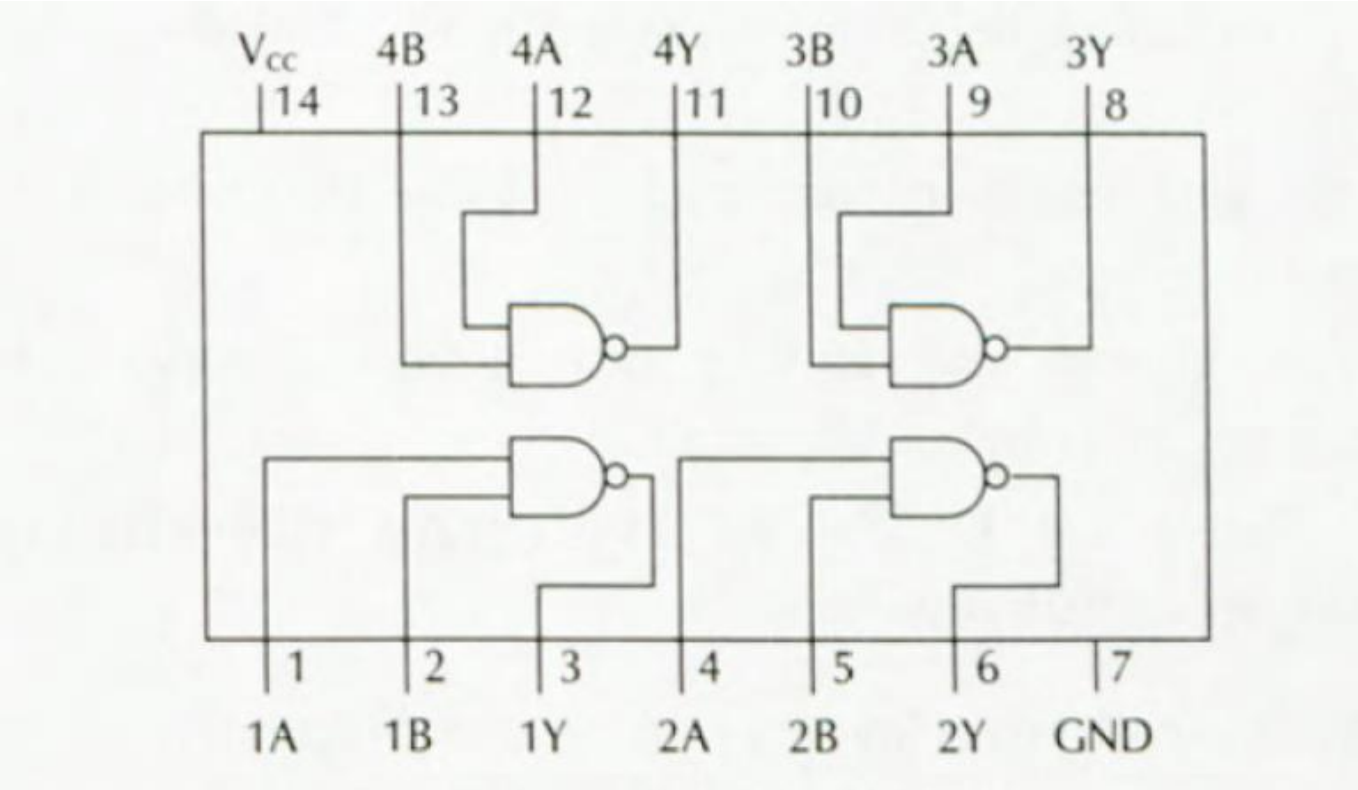
Δίνονται από τις εταιρείες κατασκευής και περιέχουν:

- Πληροφορίες για το περιεχόμενο του ολοκληρωμένου
- Λογική συνάρτηση και πίνακα αληθείας
- Μέγιστες απόλυτες τιμές
- Συνιστώμενες συνθήκες λειτουργίας
- Ηλεκτρικά χαρακτηριστικά
- Χαρακτηριστικά μεταγωγής

Ολοκληρωμένα κυκλώματα με Λογικές Πύλες

74xx ολοκληρωμένα κυκλώματα

7400	4 πύλες NAND 2 εισόδων
7402	4 πύλες NOR 2 εισόδων
7404	6 πύλες NOT
7408	4 πύλες AND 2 εισόδων
7410	3 πύλες NAND 3 εισόδων
7421	2 πύλες AND 4 εισόδων
7427	3 πύλες NOR 3 εισόδων
7430	1 πύλη NAND 8 εισόδων
7432	4 πύλες OR 2 εισόδων

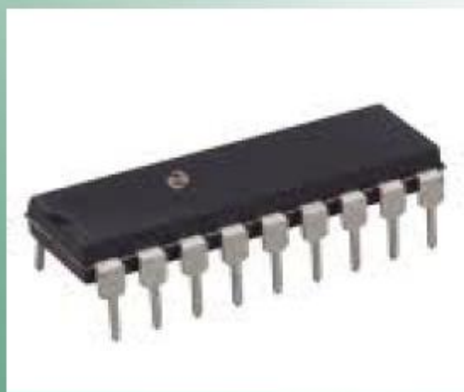


Το ολοκληρωμένο κύκλωμα 74LS00

Τεχνολογίες Ο.Κ.

Κλίμακες ολοκλήρωσης ολοκληρωμένων κυκλωμάτων

- **SSI** (λιγότερα από 10 πύλες)
- **MSI** (10-100 πύλες)
- **LSI** (100-1.000 πύλες)
- **VLSI** (1.000-100.000 πύλες)
- **ULSI** (100.000 πύλες και άνω)



Τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων

- **TTL** (η πιο διαδεδομένη λογική οικογένεια, βασίζεται σε διπολικά τρανζίστορ)
- **CMOS** (σε συστήματα χαμηλής κατανάλωσης ισχύος)
- **MOS** (υψηλή συγκέντρωση πυλών σε μικρό χώρο)
- **ECL** (για συστήματα πολύ υψηλής ταχύτητας, π.χ. Υπερυπολογιστές, επεξεργαστές σήματος)

Ανάλυση ονομασίας ολοκληρωμένων κυκλωμάτων (TTL)

DM 74 LS 00

Εταιρεία κατασκευής
π.χ.
DM=National Semiconductors
uA=Fairchild
HD=Hitachi Semiconductor
IN=Integral Semiconductor
SL=System Logic
SN=Texas Instruments
Philips
Motorola

Σειρά Λογικής Οικογένειας
π.χ.
(κενό)=Standard TTL
H=Υψηλής ταχύτητας TTL
L=Χαμηλής ισχύος
S=Schottky TTL
LS=χαμηλής ισχύος Schottky TTL
AS=προηγμένα Schottky TTL
ALS=προηγμένα χαμηλής ισχύος Schottky TTL

Ανάλυση ονομασίας ολοκληρωμένων κυκλωμάτων (CMOS)

<u>DM</u> <u>74 HC</u> <u>00</u>

<u>Εταιρεία κατασκευής</u>
π.χ.
DM=National Semiconductors
uA=Fairchild
HD=Hitachi Semiconductor
IN=Integral Semiconductor
SL=System Logic
SN=Texas Instruments
Philips
Motorola

<u>Σειρά Λογικής Οικογένειας</u>
π.χ.
74C=Συμβατοί αρκοδέκτες με TTL
74HC=Υψηλής ταχύτητας και συμβατοί ακροδέκτες με TTL
74HCT=Υψηλής ταχύτητας και ηλεκτρικά συμβατά με TTL
40=Κλασσικά CMOS

Λογικές τιμές και περιοχές τάσεων

Λογικές τιμές και περιοχές τάσης

Τα ψηφιακά ολοκληρωμένα κυκλώματα αναγνωρίζουν:

- στις **εισόδους** τους

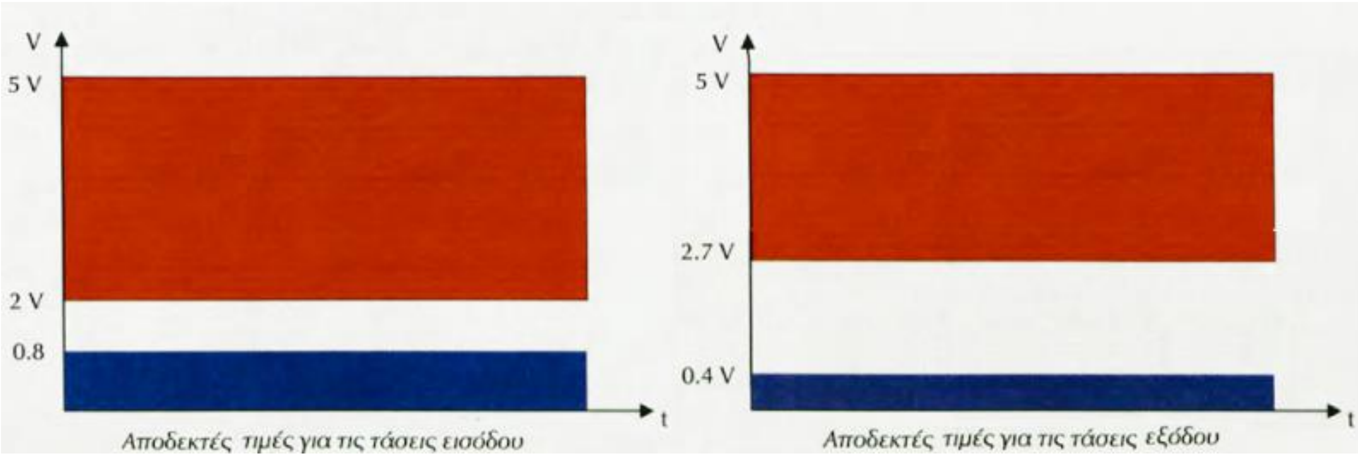
  α) ως λογικό “0” την περιοχή τάσεων 0~0,8V και

  β) ως λογικό “1” την περιοχή τάσεων 2~5V.

- στις **εξόδους** τους

  α) ως λογικό “0” την περιοχή τάσεων 0~0,4V και

  β) ως λογικό “1” την περιοχή τάσεων 2,7~5V.



Αποδεκτές τιμές για τις τάσεις εισόδου

Αποδεκτές τιμές για τις τάσεις εξόδου

Δεκαδικό σύστημα

Δεκαδικό Σύστημα

Στο δεκαδικό σύστημα χρησιμοποιούμε τα δέκα ψηφία (από το 0 έως το 9) για την απεικόνιση των δεκαδικών αριθμών. Η βάση είναι το 10 και η αξία των ψηφίων εξαρτάται από τις θέσεις τους.

π.χ. $(\mathbf{5832})_{10} = \mathbf{5}*10^{3} + \mathbf{8}*10^{2} + \mathbf{3}*10^{1} + \mathbf{2}*10^{0}$

- **MSD** (Most Significant Digit)
- **LSD** (Least Significant Digit)

	MSD			LSD
Ψηφία	5	8	3	2
Θέση	3	2	1	0
Βάρος	10^3	10^2	10^1	10^0
Αξία	$5x10^3=5000$	$8x10^2=800$	$3x10^1=30$	$2x10^0=2$

Δυαδικό σύστημα

Δυαδικό Σύστημα

Στο δυαδικό σύστημα χρησιμοποιούμε δύο ψηφία (το 0 και το 1) για την απεικόνιση των δυαδικών αριθμών. Η βάση είναι το 2 και η αξία των ψηφίων εξαρτάται από τις θέσεις τους.

π.χ. $(\mathbf{1011})_2 = 1*2^3 + 0*2^2 + 1*2^1 + 1*1^0 = (11)_{10}$

- **MSB** (Most Significant Bit)
- **LSB** (Least Significant Bit)

	MSB			LSB
Ψηφία	1	0	1	1
Θέση	3	2	1	0
Βάρος	2^3	2^2	2^1	2^0

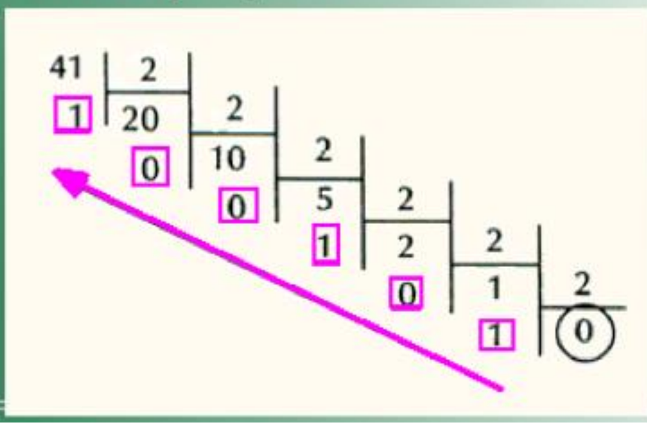
Αρίθμηση στο Δυαδικό Σύστημα

00	0000	**07**	0111	**14**	1110
01	0001	**08**	1000	**15**	1111
02	0010	**09**	1001	**16**	10000
03	0011	**10**	1010	**17**	10001
04	0100	**11**	1011	**18**	10010
05	0101	**12**	1100	**19**	10011
06	0110	**13**	1101	**20**	10100

Μετατροπή δεκαδικού αριθμού σε Δυαδικό αριθμό

Ο δεκαδικός αριθμός **διαιρείται συνεχώς με το 2**, μέχρι να προκύψει **πηλίκο ίσο με μηδέν**. Τα υπόλοιπα των διαιρέσεων (0 και 1) αποτελούν τα bit του αντίστοιχου δυαδικού αριθμού.

Δηλαδή:
$(\mathbf{41})_{10} = (\mathbf{101001})_2$



Δεκαεξαδικό σύστημα

Δεκαεξαδικό σύστημα (Hex)

- Στο δεκαεξαδικό σύστημα χρησιμοποιούμε δεκαέξι ψηφία (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Η βάση είναι το 16 και η αξία των ψηφίων εξαρτάται από τις θέσεις τους. Τα γράμματα A έως F αντιστοιχούν στους δεκαδικούς αριθμούς 10 έως 15.

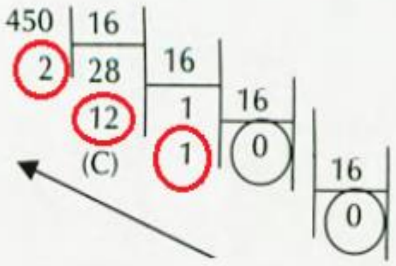
- $(FF)_{16} = 15*16^1 + 15*16^0 = 240+15 = (255)_{10}$
- $(3A9)_{16} = 3*16^2 + 10*16^1 + 9*16^0 = (937)_{10}$
- $(1D5F)_{16} = 1*16^3+13*16^2+5*16^1+15*16^0 = (7519)_{10}$

Δεκαδικό Βάση 10	Δεκαεξαδικό Βάση 16
00	0
01	1
02	2
03	3
04	4
05	5
06	6
07	7
08	8
09	9
10	A

Δεκαδικό Βάση 10	Δεκαεξαδικό Βάση 16
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
...	...
31	1F
32	20

Μετατροπή δεκαδικού σε δεκαεξαδικό

$(450)_{10}=(1C2)_{16}$



Μετατροπή δεκαεξαδικού σε δυαδικό

7	6	5	4	3	2	1	0
0111	0110	0101	0100	0011	0110	0001	0000
F	E	D	C	B	A	9	8
1111	1110	1101	1100	1011	1010	1001	1000

$(9E64)_{16}=(1001111001100100)_2$

9	E	6	4
1001	1110	0110	0100

Μετατροπή δυαδικού σε δεκαεξαδικό

$(0010111101010001)_2=(2F51)_{16}$

0010	1111	0101	0001
2	F	5	1

Πρόσθεση δυαδικών αριθμών

Πρόσθεση στο Δυαδικό Σύστημα

Οι μνημονικοί κανόνες της δυαδικής πρόσθεσης είναι:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10^{*}$$

* άθροισμα 0 και 1 κρατούμενο

Παράδειγμα πρόσθεσης στο Δυαδικό Σύστημα

Η πρόσθεση ξεκινάει από τα LSB προς τα MSB των προσθετέων. Κάθε bit του αθροίσματος είναι 0 ή 1. Το κρατούμενο κάθε θέσης προστίθεται στα bit των προσθετέων της επόμενης θέσης.

$$(1001)_2 + (1100)_2 = (10101)_2$$

Κρατούμενο		1			
		1	0	0	1
	+	1	1	0	0
	1	0	1	0	1

Αφαίρεση δυαδικών αριθμών

Αφαίρεση στο Δυαδικό Σύστημα

Οι μνημονικοί κανόνες της δυαδικής αφαίρεσης είναι:

$$0 - 0 = 0$$
$$0 - 1 = 11^{*}$$
$$1 - 0 = 1$$
$$1 - 1 = 0$$

*** 1 το αποτέλεσμα και 1 δανεικό**

Αφαίρεση στο Δυαδικό Σύστημα

Η αφαίρεση ξεκινάει από τα LSB προς τα MSB του μειωτέου και του αφαιρετέου. Το κρατούμενο κάθε θέσης προστίθεται στα bit των αφαιρετέων της επόμενης θέσης.

$$(1001)_2-(0100)_2=(0101)_2$$

Δανεικό	1			
	1	0	0	1
–	0	1	0	0
	0	1	0	1

Κώδικες

Δυαδικοί κώδικες

Κώδικας ονομάζεται ο συστηματικός τρόπος παράστασης πληροφοριών.

- Κώδικας **BCD** (δυαδικός κώδικας)
- Κώδικας **GRAY** (κυκλικός κώδικας)
- Κώδικας **ASCII** (αλφαριθμητικός κώδικας)

Κώδικας BCD

Δυαδικός κώδικας BCD

Χρησιμοποιεί τα βάρη του δυαδικού συστήματος αρίθμησης για την κωδικοποίηση των δεκαδικών ψηφίων. Κάθε ψηφίο του δεκαδικού συστήματος κωδικοποιείται ξεχωριστά σύμφωνα με τον διπλανό πίνακα.

π.χ. 923 = 1001 0010 0011

και 1000 0111 0100 0000 = 9740

Δεκαδικό ψηφίο	Δυαδικός Κώδικας BCD 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Κώδικας Gray

Κυκλικός κώδικας GRAY

Στον κώδικα Gray οι κωδικοποιημένοι αριθμοί που αντιστοιχούν σε δύο διαδοχικούς δεκαδικούς αριθμούς διαφέρουν μόνο κατά ένα bit.

Δεκαδικό ψηφίο	Κώδικας Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101

Κώδικας ASCII

Αλφαριθμητικός κώδικας ASCII

Με τον όρο **χαρακτήρα** εννοούμε αριθμούς, γράμματα, σημεία στίξης και ειδικούς χαρακτήρες.

Ο κώδικας **ASCII** περιλαμβάνει 128 χαρακτήρες:
- 94 εκτυπώσιμους (π.χ. A, B, 4, f, >, @, κτλ) και
- 34 μη εκτυπώσιμους χαρακτήρες ελέγχου (enter, κενό, αλλαγή γραμμής κειμένου, κτλ).

Σημ: Το πρότυπο **ΕΛΟΤ-928** είναι η επέκταση του ASCII, καθώς περιέχει και **ελληνικούς χαρακτήρες**.

$b_7b_6b_5$ / $b_4b_3b_2b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	'	<	L	\	l	¦
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	Æ	n	~
1111	SI	US	/	?	O	-	o	DEL

Συνδυαστικά κυκλώματα

Συνδυαστικά κυκλώματα

Τα ψηφιακά κυκλώματα διακρίνονται σε **συνδυαστικά** και **ακολουθιακά** κυκλώματα.

Ένα συνδυαστικό κύκλωμα αποτελείται από:

Α) **εισόδους**

Β) **λογικές πύλες**

Γ) **εξόδους**

Για κάθε δυνατό συνδυασμό εισόδων υπάρχει **μόνο ένας** δυνατός συνδυασμός εξόδων. Κάθε χρονική στιγμή οι έξοδοι εξαρτώνται από τις τιμές των εισόδων.

Πίνακας αληθείας

Πίνακας αληθείας

Στον πίνακα αληθείας καταγράφονται οι τιμές των εξόδων για κάθε δυνατό συνδυασμό των εισόδων.

Αν έχουμε **n** εισόδους, το πλήθος των γραμμών του πίνακα θα είναι **2^n** (όλοι οι δυνατοί συνδυασμοί).

Παράδειγμα πίνακα αληθείας

Το κύκλωμα για την πρόσθεση δύο bit (είσοδοι **X** και **Y**) θα έχει δύο εξόδους: το **άθροισμα S** και το **κρατούμενο C**.

Είσοδοι		Έξοδοι	
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Λογική συνάρτηση

Λογική συνάρτηση εξόδου από τον πίνακα αληθείας

Για κάθε έξοδο θα προκύψει **μια** λογική συνάρτηση εξόδου με τη βοήθεια του πίνακα αληθείας.

Μεθοδολογία: Επιλέγουμε τις σειρές εισόδου όπου η αντίστοιχη έξοδος είναι «1». Όταν η τιμή του Χ ή του Υ είναι «1», τότε μπαίνει ως έχει στη συνάρτηση, αλλιώς μπαίνει το συμπλήρωμα της εισόδου Χ ή Υ. Επομένως:

$$S = \overline{x} \cdot y + x \cdot \overline{y} = x \oplus y$$

$$C = x \cdot y$$

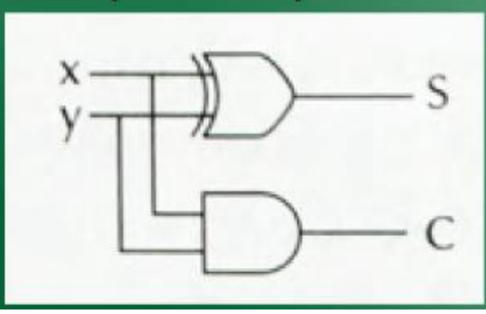
Είσοδοι		Έξοδοι	
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Συνδυαστικό κύκλωμα

Λογικό (συνδυαστικό) κύκλωμα από τη λογική συνάρτηση εξόδου

Από τον πίνακα αληθείας προέκυψαν οι δύο συναρτήσεις εξόδου Χ και Υ, οι οποίες μπορούν να χρησιμοποιηθούν για την σχεδίαση του λογικού κυκλώματος.

Είσοδοι		Έξοδοι	
x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S = \overline{x} \cdot y + x \cdot \overline{y} = x \oplus y$$

$$C = x \cdot y$$

Μέθοδοι απλοπλοίησης λογικών συναρτήσεων

Απλοποίηση λογικών συναρτήσεων

Η απλοποίηση των συναρτήσεων εξόδου ενός συνδυαστικού κυκλώματος οδηγεί σε **απλούστερο** και επομένως **οικονομικότερο** κύκλωμα.

Μέθοδοι απλοποίησης λογικών συναρτήσεων:

- με χρήση της **άλγεβρας Boole**
- με χρήση των **χαρτών Karnaugh**

Απλοποίηση με άλγεβρα Boole

Απλοποίηση με χρήση άλγεβρας Boole

Η μέθοδος απλοποίησης λογικών συναρτήσεων με χρήση της άλγεβρας **Boole** βασίζεται στη χρήση των **Αξιωμάτων** και των **Θεωρημάτων** της άλγεβρας Boole.

Παράδειγμα:

$$Y = \overline{A} \cdot B \cdot C + \overline{A} \cdot B \cdot \overline{C} =$$

$$= \overline{A} \cdot B \cdot (C + \overline{C}) =$$

$$= \overline{A} \cdot B$$

Ελάχιστοι όροι

Ελάχιστοι όροι μιας συνάρτησης

Ελάχιστοι όροι μιας συνάρτησης ονομάζονται τα γινόμενα όλων των όρων της συνάρτησης, όπου ο κάθε όρος εμφανίζεται στην κανονική ή συμπληρωματική του μορφή. **Κάθε συνάρτηση μπορεί να εκφρασθεί ως άθροισμα ελάχιστων όρων.**

A	B	C	Ελάχιστοι όροι
0	0	0	$m0 = \overline{A} \cdot \overline{B} \cdot \overline{C}$
0	0	1	$m1 = \overline{A} \cdot \overline{B} \cdot C$
0	1	0	$m2 = \overline{A} \cdot B \cdot \overline{C}$
0	1	1	$m3 = \overline{A} \cdot B \cdot C$
1	0	0	$m4 = A \cdot \overline{B} \cdot \overline{C}$
1	0	1	$m5 = A \cdot \overline{B} \cdot C$
1	1	0	$m6 = A \cdot B \cdot \overline{C}$
1	1	1	$m7 = A \cdot B \cdot C$

Έκφραση λογικής συνάρτησης ως άθροισμα ελάχιστων όρων

Κάθε συνάρτηση μπορεί να εκφρασθεί ως άθροισμα ελάχιστων όρων ακολουθώντας την παρακάτω μέθοδο:

- Βρίσκουμε όλες τις γραμμές του πίνακα αληθείας όπου η έξοδος Y=1.
- Γράφουμε τη συνάρτηση Y ως άθροισμα των ελάχιστων όρων που αντιστοιχούν σε αυτές τις γραμμές.

$$Y = \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C$$

A B C	Y
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

Μετατροπή λογικής συνάρτησης σε άθροισμα ελάχιστων όρων

Κάθε συνάρτηση μπορεί να μετατραπεί σε άθροισμα ελάχιστων όρων ως εξής: Αν σε κάποιο γινόμενο δεν υπάρχουν όλοι οι όροι, τότε πολλαπλασιάζουμε το γινόμενο αυτό με το άθροισμα της μεταβλητής (που λείπει) με το συμπλήρωμά της.

$$
\begin{aligned}
Y &= A\cdot B + \overline{A}\cdot C = \\
&= A\cdot B\cdot (C+\overline{C}) + \overline{A}\cdot(B+\overline{B})\cdot C = \\
&= A\cdot B\cdot C + A\cdot B\cdot \overline{C} + \overline{A}\cdot B\cdot C + \overline{A}\cdot \overline{B}\cdot C
\end{aligned}
$$

Χάρτες Karnaugh

Χάρτες Karnaugh

Οι χάρτες Karnaugh είναι ένας τρόπος αναπαράστασης των λογικών συναρτήσεων. Αποτελούνται από τετράγωνα που αντιστοιχούν στους ελάχιστους όρους της λογικής συνάρτησης.

	$\overline{B}$	B
$\overline{A}$	m0	m1
A	m2	m3

Χάρτης Karnaugh δύο μεταβλητών

	$\overline{B}\cdot\overline{C}$	$\overline{B}\cdot C$	$B\cdot C$	$B\cdot\overline{C}$
$\overline{A}$	m0	m1	m3	m2
A	m4	m5	m7	m6

Χάρτης Karnaugh τριών μεταβλητών

Αναπαράσταση λογικών συναρτήσεων με χάρτες Karnaugh

Η αναπαράσταση της λογικής συνάρτησης στο χάρτη γίνεται θέτοντας «1» σε κάθε τετράγωνο του χάρτη που αντιστοιχεί σε ελάχιστο όρο που υπάρχει στη συνάρτηση (δηλαδή η έξοδος Υ είναι «1»). Στα υπόλοιπα τετράγωνα βάζουμε «0» (ή μένει κενό).

	$\overline{B}$	B
$\overline{A}$	m0	m1
A	m2	m3

	$\overline{B}$	B
$\overline{A}$		1
A	1	

$$Y(A, B) = A \cdot \overline{B} + \overline{A} \cdot B$$

Χάρτης Karnaugh 4x4

Χάρτης Karnaugh 4x4

Χρησιμοποιείται για την αναπαράσταση λογικών συναρτήσεων τεσσάρων μεταβλητών.

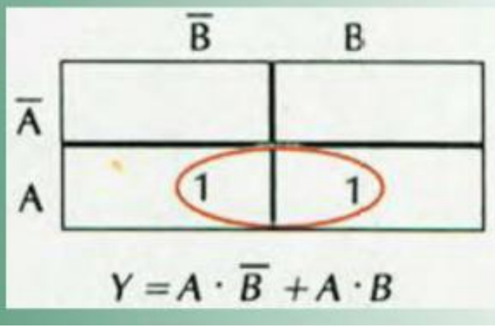
	$\overline{C} \cdot \overline{D}$	$\overline{C} \cdot D$	$C \cdot D$	$C \cdot \overline{D}$
$\overline{A} \cdot \overline{B}$	m0	m1	m3	m2
$\overline{A} \cdot B$	m4	m5	m7	m6
$A \cdot B$	m12	m13	m15	m14
$A \cdot \overline{B}$	m8	m9	m11	m10

Χάρτης Karnaugh τεσσάρων μεταβλητών

Απλοποίηση λογικών συναρτήσεων με χάρτη Karnaugh

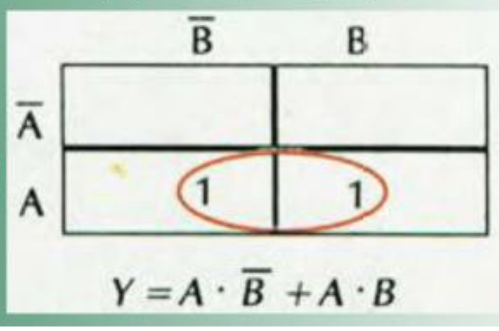
Απλοποίηση λογικών συναρτήσεων με χάρτη Karnaugh

- **Γειτονικά** ονομάζονται τα τετράγωνα που είναι σε **συνεχόμενες οριζόντιες ή κάθετες** θέσεις (αλλά όχι σε διαγώνιες).
- Δημιουργούμε τις <u>μεγαλύτερες δυνατές ομάδες</u> γειτονικών τετραγώνων που περιέχουν «1». Το πλήθος των «1» κάθε ομάδας πρέπει να είναι **δύναμη του 2** (δηλαδή 2, 4, 8 ή 16).



Απλοποίηση λογικών συναρτήσεων με χάρτη Karnaugh

- Οι μεταβλητές που μέσα σε μια ομάδα εμφανίζονται μαζί με το συμπλήρωμά τους, μπορούν να απαλειφθούν από τη συνάρτηση.
- Στο παράδειγμά μας υπάρχει μόνο μια ομάδα δυο γειτονικών τετραγώνων, στην οποία η μεταβλητή B εμφανίζεται ταυτόχρονα με το συμπλήρωμά της. Οπότε μπορεί να διαγραφεί η μεταβλητή B και η συνάρτηση να γίνει **Y=A**.



Σχεδίαση συνδυαστικού κυκλώματος

Σχεδίαση συνδυαστικών κυκλωμάτων

Από την **περιγραφή λειτουργίας** ενός συνδυαστικού κυκλώματος (ΣΚ)μπορούμε να σχεδιάσουμε το κύκλωμα, ακολουθώντας τα εξής βήματα:

- Κατασκευάζουμε τον **πίνακα αληθείας** του ΣΚ
- Βρίσκουμε τις **συναρτήσεις εξόδου** του ΣΚ
- **Απλοποιούμε** τις συναρτήσεις εξόδου του ΣΚ
- **Σχεδιάζουμε** το λογικό κύκλωμα του ΣΚ

Ανάλυση συνδυαστικού κυκλώματος

Ανάλυση συνδυαστικών κυκλωμάτων

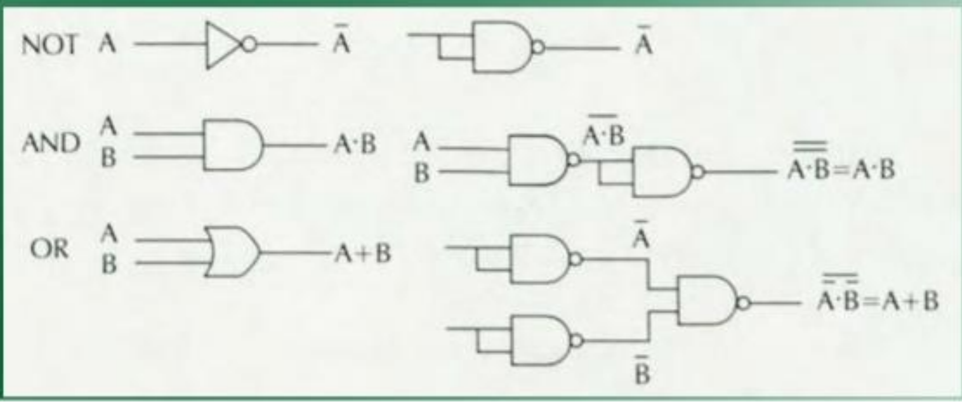
Το συνδυαστικό κύκλωμα (ΣΚ) μπορούμε να το αναλύσουμε και να περιγράψουμε τη λειτουργία του, ακολουθώντας τα εξής βήματα:

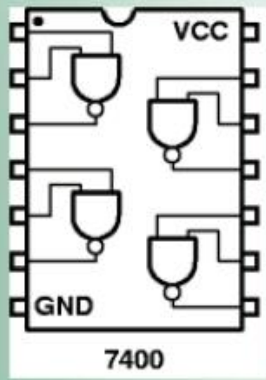
- Βρίσκουμε τις **συναρτήσεις εξόδου** του ΣΚ
- Κατασκευάζουμε τον **πίνακα αληθείας** του ΣΚ
- Περιγράφουμε τη **λειτουργία** του ΣΚ

Οικουμενικές πύλες

Οικουμενικές πύλες (universal gates)

Οι πύλες **NAND** και **NOR** ονομάζονται **οικουμενικές** πύλες, γιατί μπορούν να αντικαταστήσουν τις πύλες AND, NOT και OR, με αποτέλεσμα να προκύπτουν πιο οικονομικές κατασκευές (χρήση λιγότερων ΟΚ).





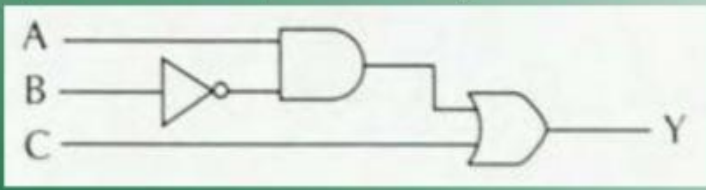
Κατασκευή κυκλώματος μόνο με πύλες NAND

Μέθοδος σχεδίασης ΣΚ με οικουμενικές πύλες

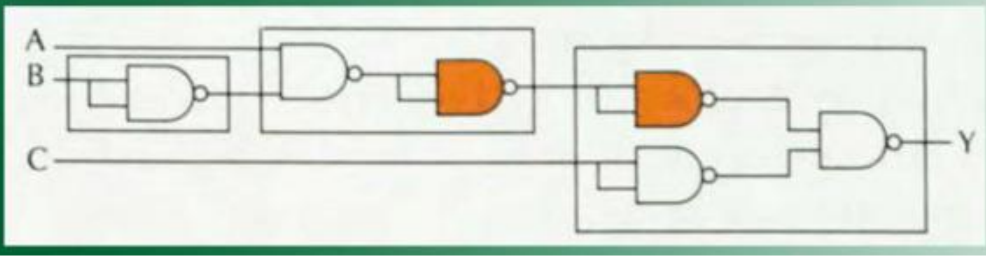
- Σχεδιάζουμε το ΣΚ με πύλες AND, NOT και OR.
- Αντικαθιστούμε όλες τις πύλες με αντίστοιχους συνδυασμούς από πύλες NAND
- Διαγράφουμε δύο συνεχόμενες πύλες NAND που λειτουργούν ως πύλες NOT (με βραχυκυκλωμένες εισόδους), καθώς η δεύτερη αναιρεί τον λόγο ύπαρξης της πρώτης.

Παράδειγμα σχεδίασης ΣΚ με οικουμενικές πύλες

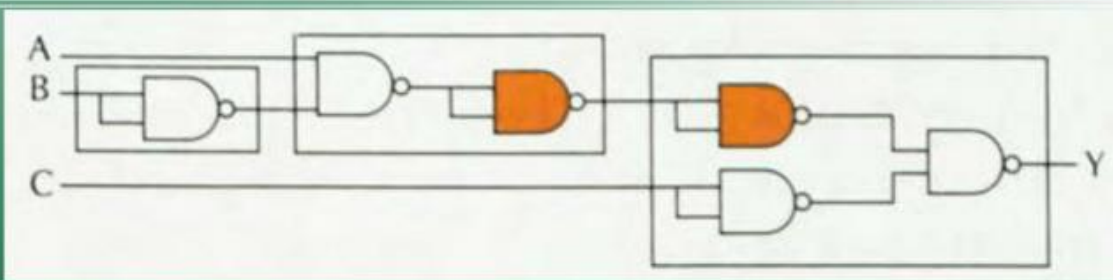
- Σχεδιάζουμε το ΣΚ με πύλες AND, NOT και OR.



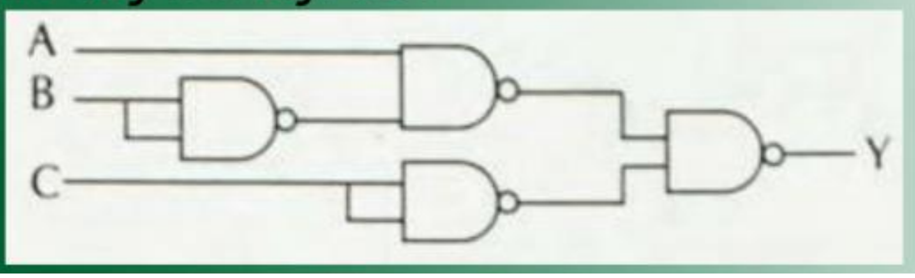
- Αντικαθιστούμε όλες τις πύλες με αντίστοιχους συνδυασμούς από πύλες NAND



Παράδειγμα σχεδίασης ΣΚ με οικουμενικές πύλες



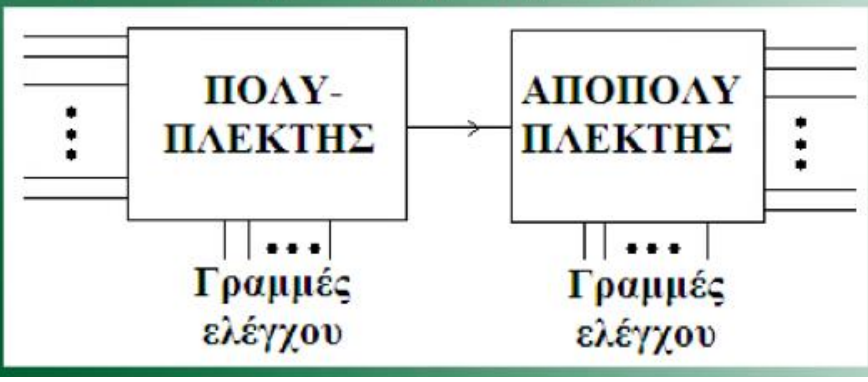
- Διαγράφουμε δύο συνεχόμενες πύλες NAND που λειτουργούν ως πύλες NOT



Πολύπλεξη – Αποπολύπλεξη

Πολύπλεξη (multiplexing) & Αποπολύπλεξη (demultiplexing)

- **Πολύπλεξη** είναι η επιλογή μιας γραμμής εισόδου και η οδήγησή της στην μοναδική γραμμή εξόδου.
- **Αποπολύπλεξη** είναι η οδήγηση της γραμμής εισόδου στην επιλεγμένη γραμμή εξόδου.



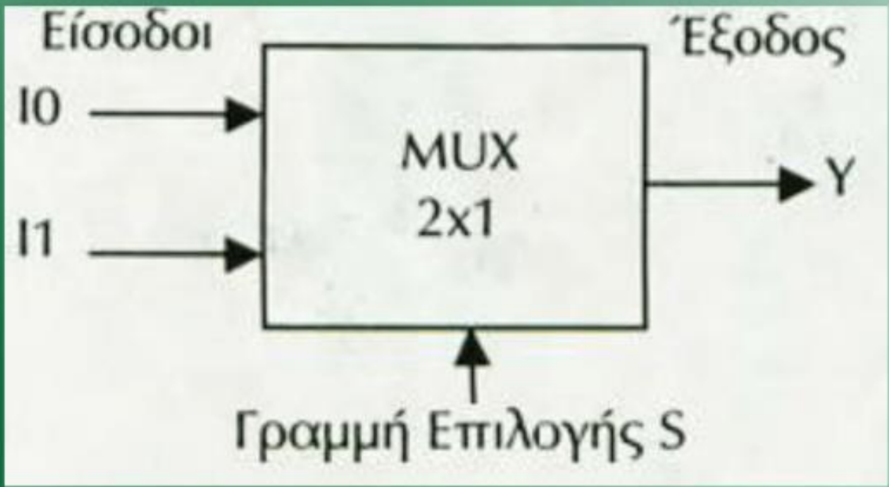
Πολυπλέκτες

Πολυπλέκτης

Ο Πολυπλέκτης **2^n εισόδων** είναι ένα ΣΚ που έχει **n γραμμές επιλογής** (ελέγχου) και **μια γραμμή εξόδου**, στην οποία κατευθύνονται τα δεδομένα από την γραμμή εισόδου που υποδεικνύουν οι γραμμές επιλογής.

Πολυπλέκτης 2 εισόδων (MUX 2x1)

Ανάλογα με την τιμή της γραμμής επιλογής **S**, μια από τις εισόδους $\mathbf{I_0}$ και $\mathbf{I_1}$ μεταβιβάζεται στην έξοδο **Y**.

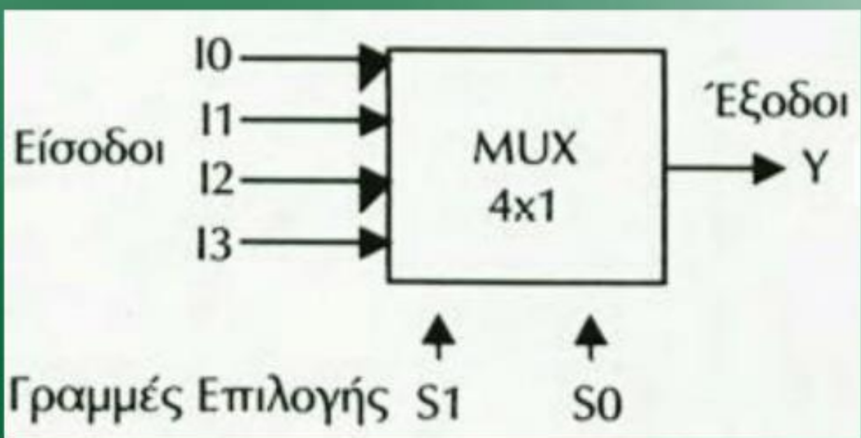


S	Y
0	I0
1	I1

Πολυπλέκτες 4x1 και 8x1

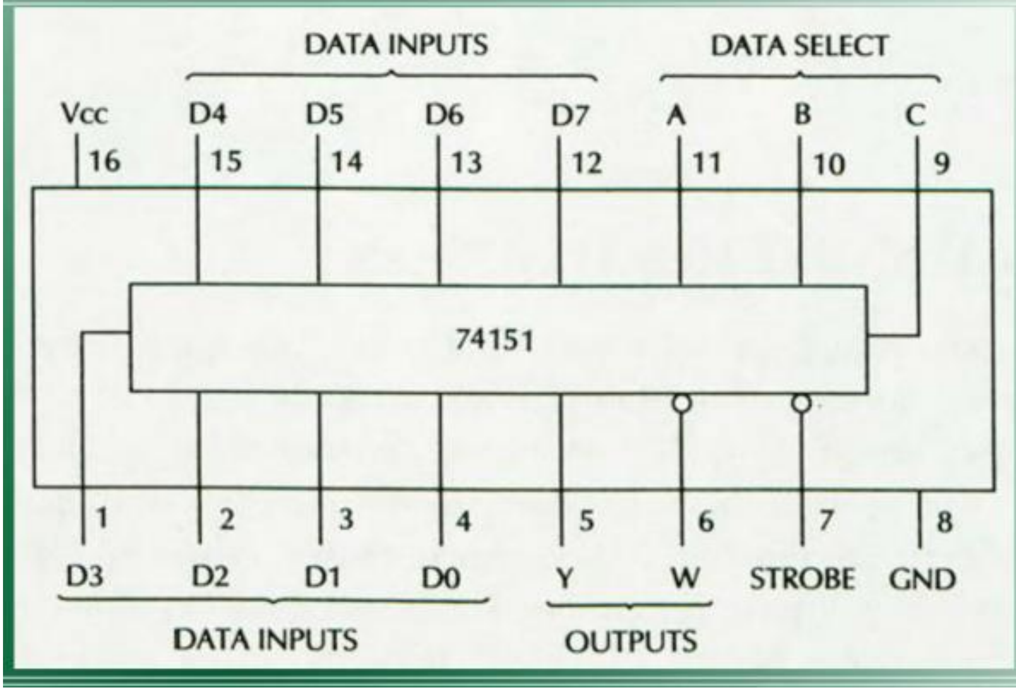
Πολυπλέκτης 4 εισόδων (MUX 4x1)

Ανάλογα με τις τιμές των γραμμών επιλογής $\mathbf{S_0}$, $\mathbf{S_1}$ μια από τις εισόδους $\mathbf{I_0}$, $\mathbf{I_1}$, $\mathbf{I_2}$ ή $\mathbf{I_3}$ μεταβιβάζεται στην έξοδο **Y**.



S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

Πολυπλέκτης 8 εισόδων (Ο.Κ. 74151 - MUX 8x1)



STR	C	B	A	Y
1	X	X	X	0
0	0	0	0	I0
0	0	0	1	I1
0	0	1	0	I2
0	0	1	1	I3
0	1	0	0	I4
0	1	0	1	I5
0	1	1	0	I6
0	1	1	1	I7

Εφαρμογή πολυπλεκτών (Ι)

Εφαρμογή πολυπλεκτών (Ι)

Κάθε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί με ένα πολυπλέκτη 2^n εισόδων.

- Οι μεταβλητές της λογικής συνάρτησης αποτελούν τις **γραμμές επιλογής** του πολυπλέκτη.
- Οι **είσοδοι** του πολυπλέκτη αντιστοιχίζονται με τη στήλη της εξόδου Υ του πίνακα αληθείας.
- Η **έξοδος** του πολυπλέκτη μας δίνει τη λογική συνάρτηση Υ που θέλουμε να υλοποιήσουμε.

Εφαρμογή πολυπλεκτών (ΙΙ)

Εφαρμογή πολυπλεκτών (ΙΙ)

Κάθε ΣΚ n εισόδων και m εξόδων μπορεί να υλοποιηθεί με m πολυπλέκτες 2^n εισόδων.

- Οι μεταβλητές των λογικών συναρτήσεων αποτελούν τις **γραμμές επιλογής** των πολυπλεκτών
- Οι **είσοδοι** των πολυπλεκτών αντιστοιχίζονται με τις στήλες εξόδων του πίνακα αληθείας.
- Οι **έξοδοι** των πολυπλεκτών μας δίνουν τις λογικές συναρτήσεις που θέλουμε να υλοποιήσουμε.

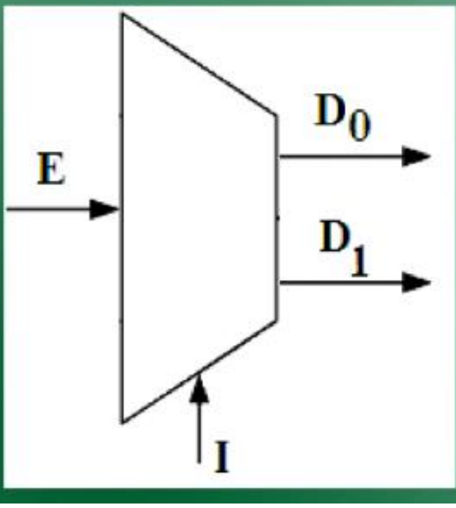
Αποπολυπλέκτες

Αποπολυπλέκτης $1x2^n$ (DeMux)

Είναι ένα συνδυαστικό κύκλωμα που έχει μια είσοδο, **n** γραμμές επιλογής και **2^n** γραμμές εξόδου, στις οποίες μεταβιβάζει τα δεδομένα της γραμμής εισόδου, ανάλογα με την τιμή των γραμμών επιλογής.

Αποπολυπλέκτης 1x2

Έχει μια είσοδο **E**, μια γραμμή επιλογής **I** και δύο γραμμές εξόδου **D_0, D_1**.

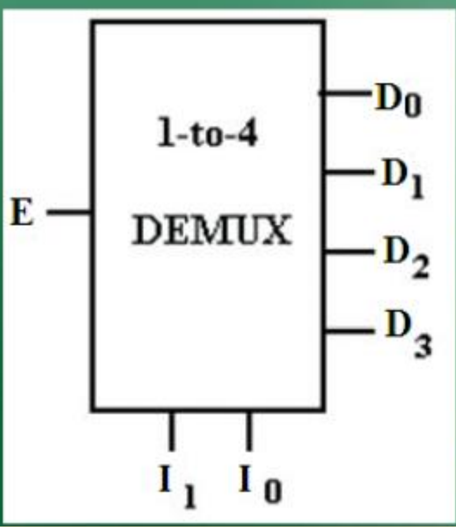


E	I0	D0	D1
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

Αποπολυπλέκτης 1x4

Αποπολυπλέκτης 1x4

Έχει μια είσοδο **E**, δύο γραμμές επιλογής **I_0, I_1** και τέσσερις γραμμές εξόδου **D_0, D_1, D_2, D_3**.

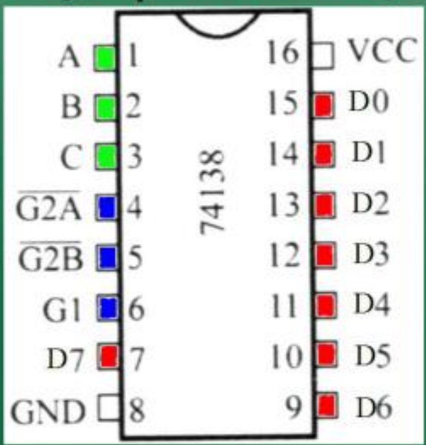


E	I1	I0	D0	D1	D2	D3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Αποπολυπλέκτης 1x8 (Ο.Κ.74138)

Αποπολυπλέκτης 1x8 (Ο.Κ. 74138)

Οι **A**, **B**, **C** είναι οι γραμμές επιλογής, οι είσοδοι ενεργοποίησης είναι $\mathbf{G_1}$=1 και $\mathbf{G_{2A}}$=$\mathbf{G_{2B}}$=0, ενώ οι D_0 έως D_7 είναι οι έξοδοι του αποπολυπλέκτη.

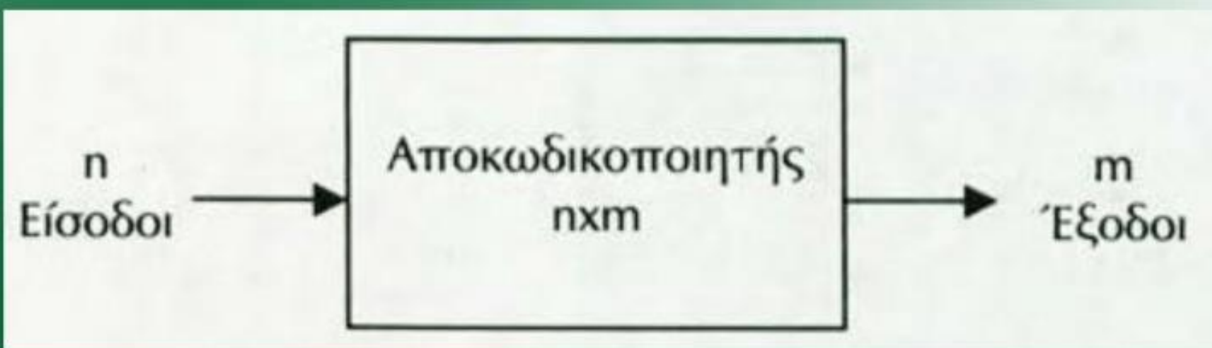


C	B	A	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Αποκωδικοποιητές

Αποκωδικοποιητής NxM (Decoder)

Ο αποκωδικοποιητής από **n** σε **m** (**nxm**) είναι ένα συνδυαστικό κύκλωμα με **n γραμμές εισόδου** και **m γραμμές εξόδου**. Για κάθε συνδυασμό εισόδου, μόνο μια από τις εξόδους του είναι ενεργή.



Αποκωδικοποιητής 3x8

Ο αποκωδικοποιητής 3 σε 8 (3x8) έχει τρεις εισόδους (C, B, A) και οκτώ εξόδους (D0 έως D7).

C	B	A	D0	D1	D2	D3	D4	D5	D6	D7
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 4x10

Ο αποκωδικοποιητής 4 σε 10 (4x10) έχει 4 εισόδους (D, C, B, A) και 10 εξόδους (D0 έως D9).

D	C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Υλοποίηση συνδυαστικών κυκλωμάτων με αποκωδικοποιητές

Υλοποίηση συνδυαστικών κυκλωμάτων με αποκωδικοποιητές

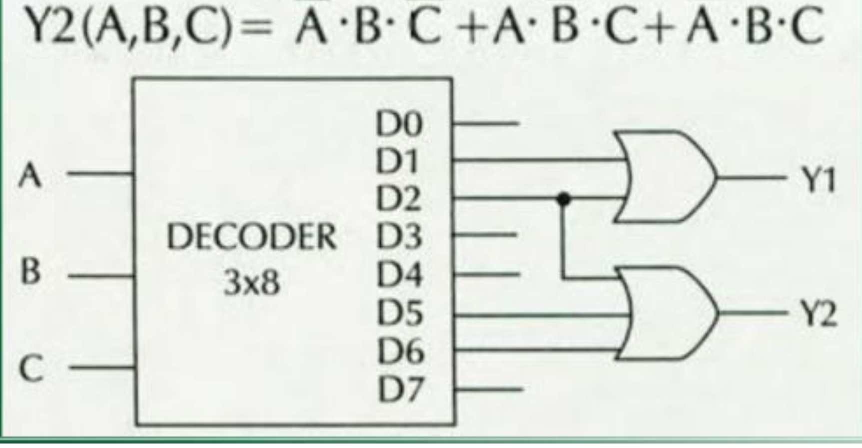
Κάθε λογική συνάρτηση μπορεί να εκφρασθεί ως άθροισμα ελαχίστων όρων.
Επομένως:
Κάθε συνδυαστικό κύκλωμα **n** εισόδων και **m** εξόδων μπορεί να υλοποιηθεί με ένα **αποκωδικοποιητή nx2^n** και **m πύλες OR** που θα «αθροίζουν» τις εξόδους του αποκωδικοποιητή που αντιστοιχούν στους ελάχιστους όρους για τους οποίους η συνάρτηση δίνει «1».

Υλοποίηση συνδυαστικών κυκλωμάτων με αποκωδικοποιητές

Να υλοποιηθεί με ένα αποκωδικοποιητή και πύλες OR το Σ.Κ με τις ακόλουθες συναρτήσεις εξόδου:

$$Y1(A,B,C)=A\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot B\cdot\overline{C}$$

$$Y2(A,B,C)=\overline{A}\cdot B\cdot\overline{C}+A\cdot\overline{B}\cdot C+\overline{A}\cdot B\cdot C$$



Ολοκληρωμένα κυκλώματα αποκωδικοποιητών

Ολοκληρωμένα κυκλώματα αποκωδικοποιητών

Ο.Κ απωδικοποιητών 2x4: 74139, 74155, 74156

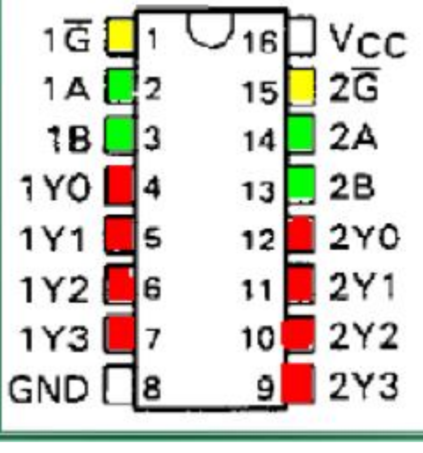
Ο.Κ απωδικοποιητών 3x8: 74138

Ο.Κ απωδικοποιητών 4x16: 74154, 74159

Ο.Κ απωδικοποιητών 4x10: 7442, 7443, 7444, 7445, 74141, 74145

Ο.Κ.74139
Διπλός αποκωδικοποιητής 2x4

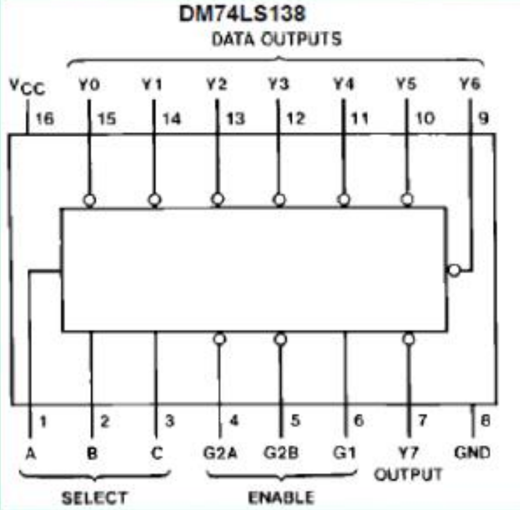
Το Ο.Κ.74139 περιέχει 2 αποκωδικοποιητές 2x4. Οι είσοδοι 1G και 2G ενεργοποιούν ή απενεργοποιούν όποιον από τους δυο αποκωδικοποιητές θέλουμε.



INPUTS			OUTPUTS			
ENABLE	SELECT					
$\overline{G}$	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

Ο.Κ.74138
Αποκωδικοποιητής 3x8

Λειτουργεί ως αποκωδικοποιητής όταν G_1=1, G_{2A}=0 και G_{2B}=0. Η ενεργοποιημένη έξοδος είναι «0» (Low).

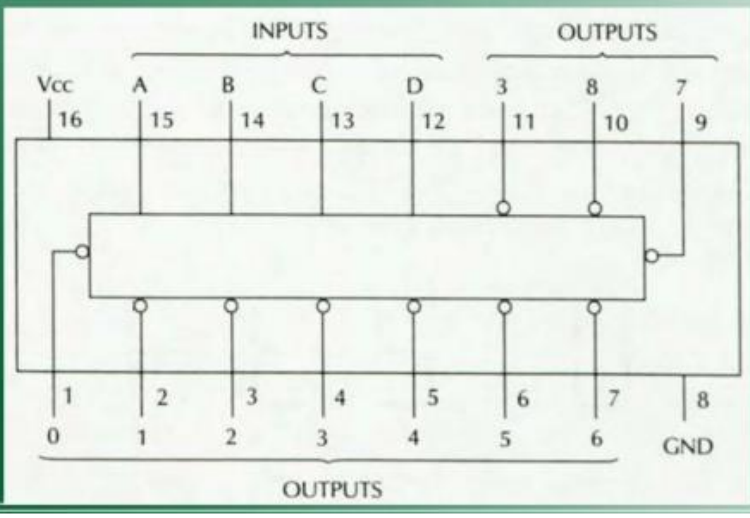


DM74LS139

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Ο.Κ.7442: Αποκωδικοποιητής 4x10 (BCD σε δεκαδικό)

Έχει 4 εισόδους (A, B, C, D) και 10 εξόδους στις οποίες εμφανίζει "0" όταν ενεργοποιούνται.



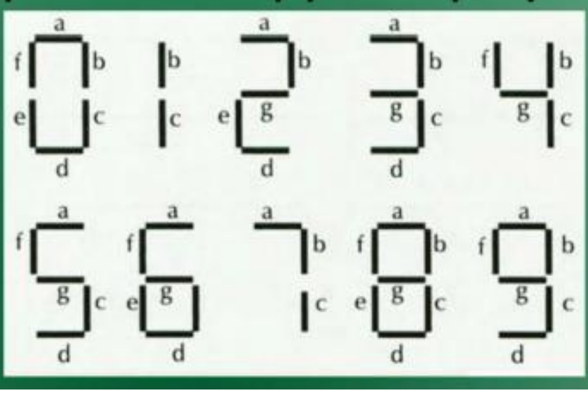
Ο.Κ.7442: αποκωδικοποιητής 4x10 (BDC σε δεκαδικό)

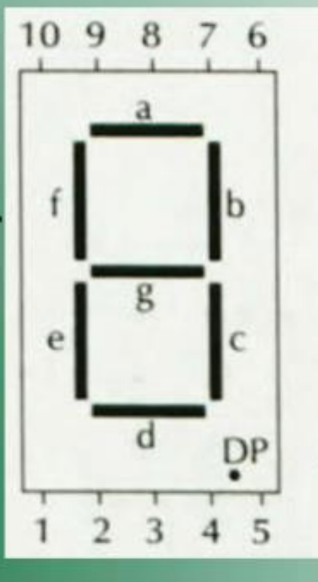
D	C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Αποκωδικοποιητές οδηγοί

Ενδείκτες 7 τμημάτων (7 segment displays)

Αναπαριστούν τους δεκαδικούς αριθμούς 0-9 χρησιμοποιώντας LED για κάθε κομμάτι αριθμού.

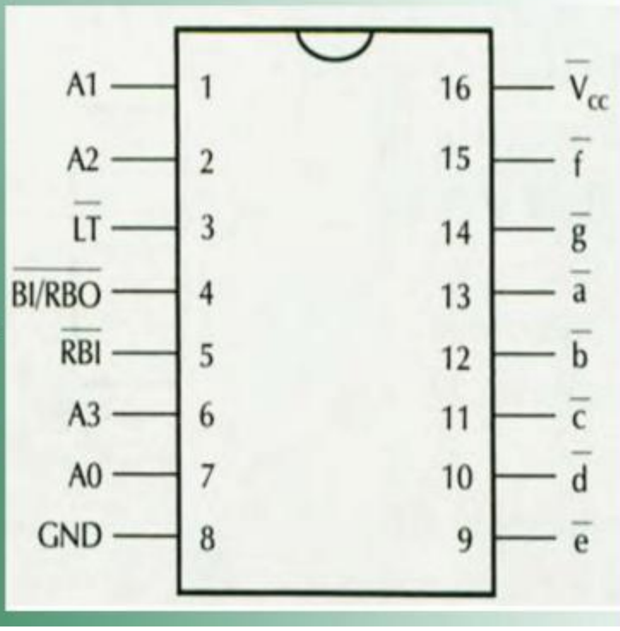




1. e
2. d
3. Άνοδος ή κάθοδος
4. c
5. DP
6. b
7. a
8. Άνοδος ή κάθοδος
9. f
10. g

Αποκωδικοποιητές 7447-7448
(BCD to 7 segments decoder)

Χρησιμοποιείται αποκλειστικά για την οδήγηση ενδείκτη 7 τμημάτων (αποκωδικοποιητής οδηγός). Ο **7447** οδηγεί display **κοινής ανόδου**, ενώ ο **7448** οδηγεί display **κοινής καθόδου.**

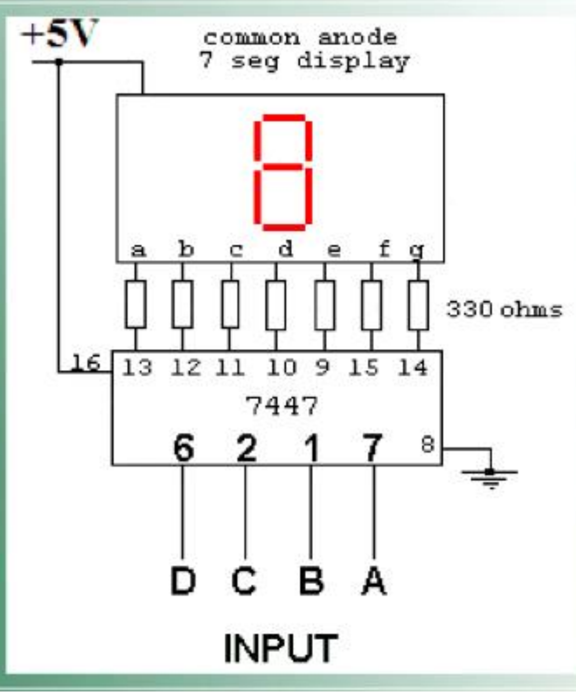


D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0

Πίνακας Αληθείας του ολοκληρωμένου κυκλώματος 7447

Σύνδεση του αποκωδικοποιητή 7447 με τον ενδείκτη 7 τμημάτων

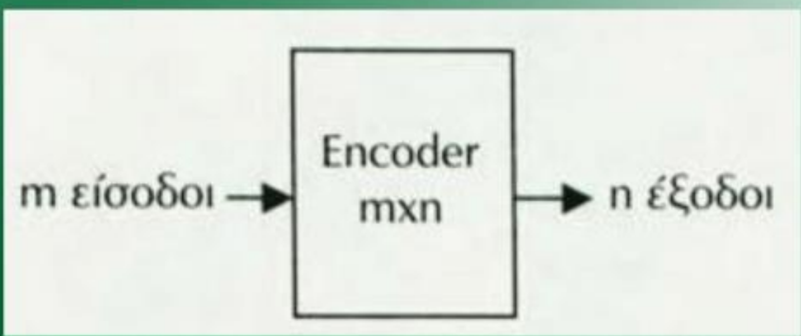
Στο display κοινής ανόδου, όλες οι άνοδοι των LED που περιέχει συνδέονται μαζί στην τάση +5V. Οπότε οι κάθοδοι των LED (a έως g) οδηγούνται από τις εξόδους του αποκωδικοποιητή 7447, έτσι ώστε να εμφανιστούν οι δεκαδικοί αριθμοί 0-9 στο display.



Κωδικοποιητές

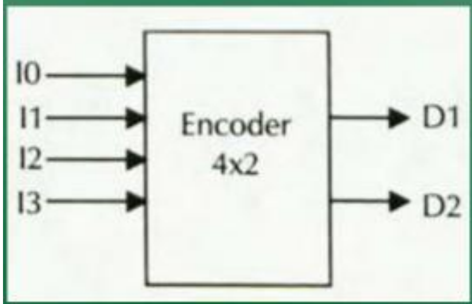
Κωδικοποιητές NxM (Encoders)

Είναι συνδυαστικά κυκλώματα με **m γραμμές εισόδου** και **n γραμμές εξόδου**. Όταν μια από τις γραμμές εισόδου ενεργοποιηθεί, τότε στην έξοδο παράγεται ένας n-bit κωδικός που αντιστοιχεί στην είσοδο.



Κωδικοποιητής 4x2

Έχει **4 γραμμές εισόδου** και **2 γραμμές εξόδου**.

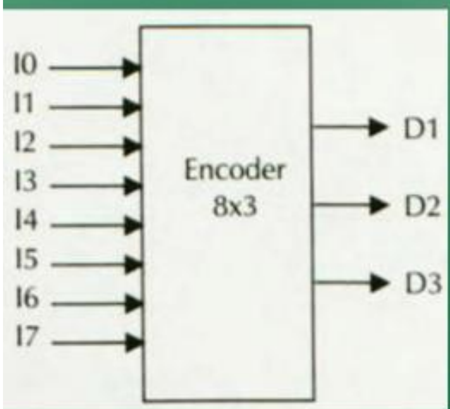


I0	I1	I2	I3	D2	D1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Κωδικοποιητής 8x3

Κωδικοποιητής 8x3

Έχει **8 γραμμές εισόδου** και **3 γραμμές εξόδου**.



I0	I1	I2	I3	I4	I5	I6	I7	D3	D2	D1
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Κωδικοποιητές προτεραιότητας

Ο.Κ. κωδικοποιητών προτεραιότητας (Priority encoders)

Ο κωδικοποιητής προτεραιότητας διαθέτει **καθορισμένη προτεραιότητα στις εισόδους του**, με αποτέλεσμα όταν ενεργοποιηθούν κάποιες από τις εισόδους του κωδικοποιητή, τότε **η έξοδος θα καθοριστεί από την είσοδο με τη μεγαλύτερη προτεραιότητα.**

- Το Ο.Κ.**74148** είναι ένας κωδικοποιητής προτεραιότητας 8x3.
- Το Ο.Κ.**74147** είναι ένας κωδικοποιητής προτεραιότητας από δεκαδικό σε BCD.

Ο.Κ. 74148: κωδικοποιητής προτεραιότητας 8x3

Στην έξοδο $A_2A_1A_0$ παράγεται ο αριθμός (με ανάστροφη λογική) που αντιστοιχεί στην είσοδο με την μεγαλύτερη προτεραιότητα που είχε «0» (Low).

0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	H	H	H	H	H	H	H	H	H	H	H	L
X	X	X	X	X	X	X	L	L	L	L	L	H
X	X	X	X	X	X	L	H	L	L	H	L	H
X	X	X	X	X	L	H	H	L	H	L	L	H
X	X	X	X	L	H	H	H	L	H	H	L	H
X	X	X	L	H	H	H	H	H	L	L	L	H
X	X	L	H	H	H	H	H	H	L	H	L	H
X	L	H	H	H	H	H	H	H	H	L	L	H
L	H	H	H	H	H	H	H	H	H	H	L	H

logic level, L = low logic level, X = irrelevant

Ο.Κ. 74147: κωδικοποιητής προτεραιότητας από δεκαδικό σε BCD

Στην έξοδο DCBA παράγεται ο BCD κωδικός (με ανάστροφη λογική) που αντιστοιχεί στον μεγαλύτερο δεκαδικό αριθμό που ενεργοποιήθηκε στην είσοδο.

I1	I2	I3	I4	I5	I6	I7	I8	I9	D	C	B	A
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0