

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΟΜΕΑΣ ΥΛΙΚΟΥ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Εργαστήριο Ηλεκτρονικών & Λογικού Σχεδιασμού

**ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ
ΛΟΓΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ**

Γ.Φ. ΑΛΕΞΙΟΥ - Ε.Ζ. ΨΑΡΑΚΗΣ

Πάτρα 2018

Ακαδ. Έτος : 2020-21

Εαρινό Εξάμηνο : 2020-21

Όνοματεπώνυμο:Κατσαρός Ανδρέας

A.M. : 1084522

Τμήμα :

Άσκηση:__ ΑΣΚΗΣΗ 1 __

Αναλυτική Βαθμολογία Πρακτικού Μέρους:

	Βαθμός Αριθ.	Αξιολογητή Ολογραφ.	Υπογραφή Αξιολογητή
<i>Άσκηση 1</i>			
<i>Άσκηση 3</i>			
<i>Άσκηση 5</i>			
<i>Άσκηση 6</i>			
<i>Άσκηση 7</i>			
M.O.			

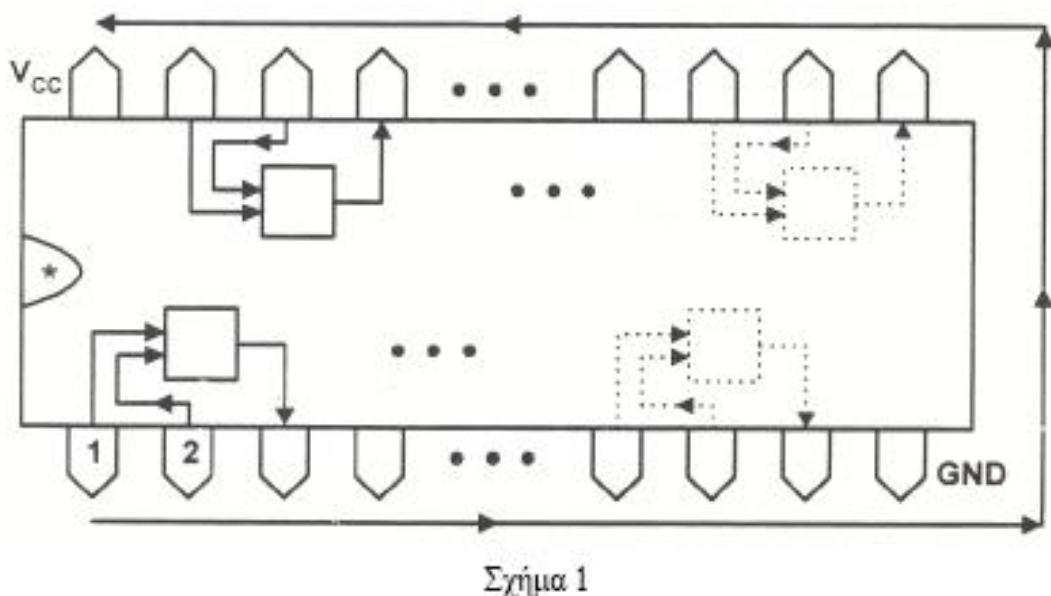
ΑΣΚΗΣΗ 1

ΘΕΜΑ: ΛΟΓΙΚΕΣ ΠΥΛΕΣ - ΣΥΝΑΡΤΗΣΕΙΣ BOOLE

(Morris Mano, Κεφ. 1- 4).

ΘΕΩΡΙΑ:

Γενικά χαρακτηριστικά ολοκληρωμένων κυκλωμάτων (**Integrated Circuits –ICs**).



1. Όταν η εγκοπή (* στο σχήμα 1) είναι αριστερά, η αρίθμηση για τους ακροδέκτες του chip γίνεται κατά τη φορά του βέλους (αντίθετα από τη φορά του ρολογιού).
2. Στα περισσότερα chips ο πάνω αριστερός ακροδέκτης είναι η τροφοδοσία (V_{CC}) και ο κάτω δεξιός είναι η γείωση (GND).
3. Οι είσοδοι - έξοδοι των πυλών έχουν την τοπολογία του σχήματος 1, εκτός από τη NOR (LS7402) που είναι ανάποδα (**δείτε το ΠΑΡΑΤΗΜΑ**).

ΔΙΑΔΙΚΑΣΙΑ

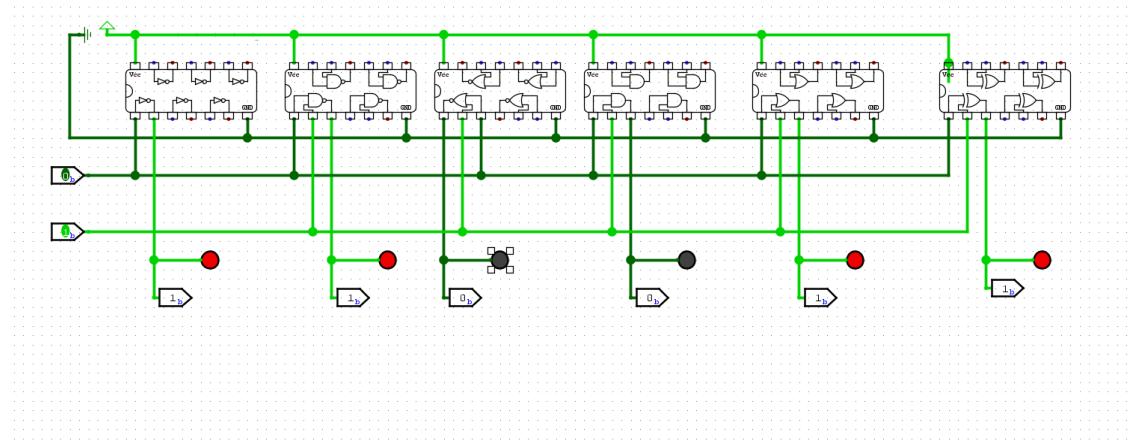
- Γράψτε παρακάτω και επαληθεύστε στον simulator, τους πίνακες αλήθειας των πυλών: NOT, AND, OR, NAND, NOR, XOR τοποθετώντας και ενδεικτικές φωτογραφίες (screenshots) από τα αντίστοιχα κυκλώματα.

ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ:

A	B	NOT (A)	NOT(B)	AND	OR	NAND	NOR	XOR
0	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0	1
1	0	0	1	0	1	1	0	1
1	1	0	0	1	1	0	0	0

ΚΥΚΛΩΜΑΤΑ (screenshots):

Κύκλωμα με είσοδο b1= 0 και b2= 1 :



2. Απλοποιείστε και υλοποιείστε στον simulator, τοποθετώντας και ενδεικτικές φωτογραφίες (screenshot), κάθε μία από τις παρακάτω συναρτήσεις :

1^η Συνάρτηση

$F(A, B, C, D) = \Sigma (1, 4, 5, 9, 10)$, που έχει τους ελαχιστόρους αδιάφορου τιμής:
 $d(A, B, C, D) = \Sigma (0, 2, 6, 8)$

με τον ελάχιστο αριθμό πύλες NOR. Χρησιμοποιείστε MONO τα ICs 74LS02, 74LS27

ΑΠΛΟΠΟΙΗΜΕΝΗ ΣΥΝΑΡΤΗΣΗ BOOLE:

Φτιάχνουμε τον παρακάτω Χάρτη Karnaugh και έχουμε:

<u>AB</u>	<u>CD</u>	--	--	--
--	<u>00</u>	<u>01</u>	<u>11</u>	<u>10</u>
<u>00</u>	X	1		X
<u>01</u>	1	1		X
<u>11</u>				
<u>10</u>	X	1		1

Προκύπτει η παρακάτω απλοποιημένη μορφή:

$$F(A, B, C, D) = A' C' + B' C' + B' D' = (A+C)' + (B+C)' + (B+D)'$$

Μπορούμε να γράψουμε την παραπάνω συνάρτηση με τον εξής τρόπο:

$$F(A, B, C, D) = ((A+C)' + (B+C)' + (B+D)')''$$

Μετά την έξοδο του 74LS27 έχουμε:

$$((A+C)' + (B+C)' + (B+D)')''$$

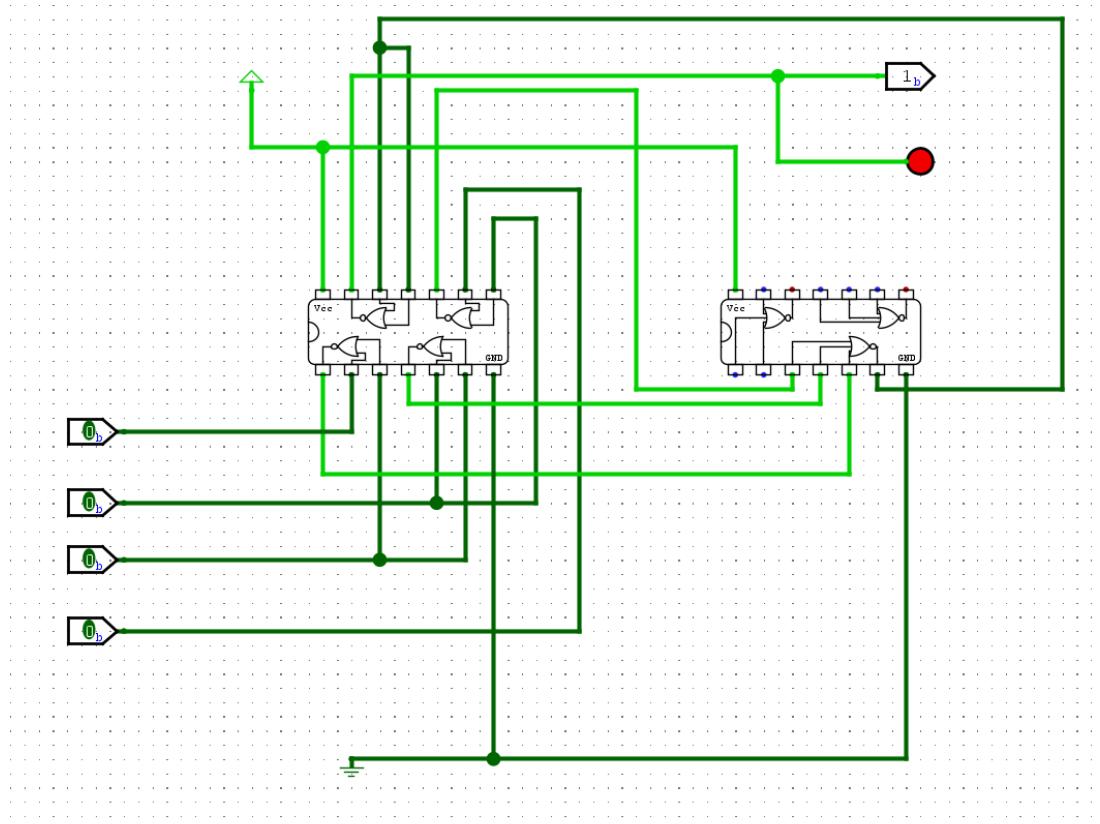
Η πύλη NOR λειτουργεί και ως αντιστροφέας, αλλά και σαν οποιαδήποτε πύλη.

Επομένως περνάμε ξανά στο 74LS02 τη παραπάνω συνάρτηση. Έτσι έχουμε:

(Επόμενη σελίδα)

ΚΥΚΛΩΜΑ (screenshot):

ΓΙΑ A=0 B=0 C=0 D=0:



2^η Συνάρτηση

$$F(A, B, C, D) = (A + B)(A' + D')(A' + B')$$

με όχι παραπάνω από τρεις πύλες NAND. Χρησιμοποιείστε MONO τα ICs **74LS00, 74LS10** και **74LS04**

ΑΠΛΟΠΟΙΗΜΕΝΗ ΣΥΝΑΡΤΗΣΗ BOOLE:

Μπορούμε να γράψουμε την παραπάνω συνάρτηση με τον εξής τρόπο:

$$F(A, B, C, D) = ((A'B')'(AD)'(AB)')' = ((A'B')'(AB)'(AD)')''$$

Μετά την έξοδο του **74LS10** έχουμε:

$$F(A, B, C, D) = ((AB)'(AD)'(A'B')')'$$

Η πύλη NAND λειτουργεί και ως αντιστροφέας, αλλά και σαν οποιαδήποτε πύλη.

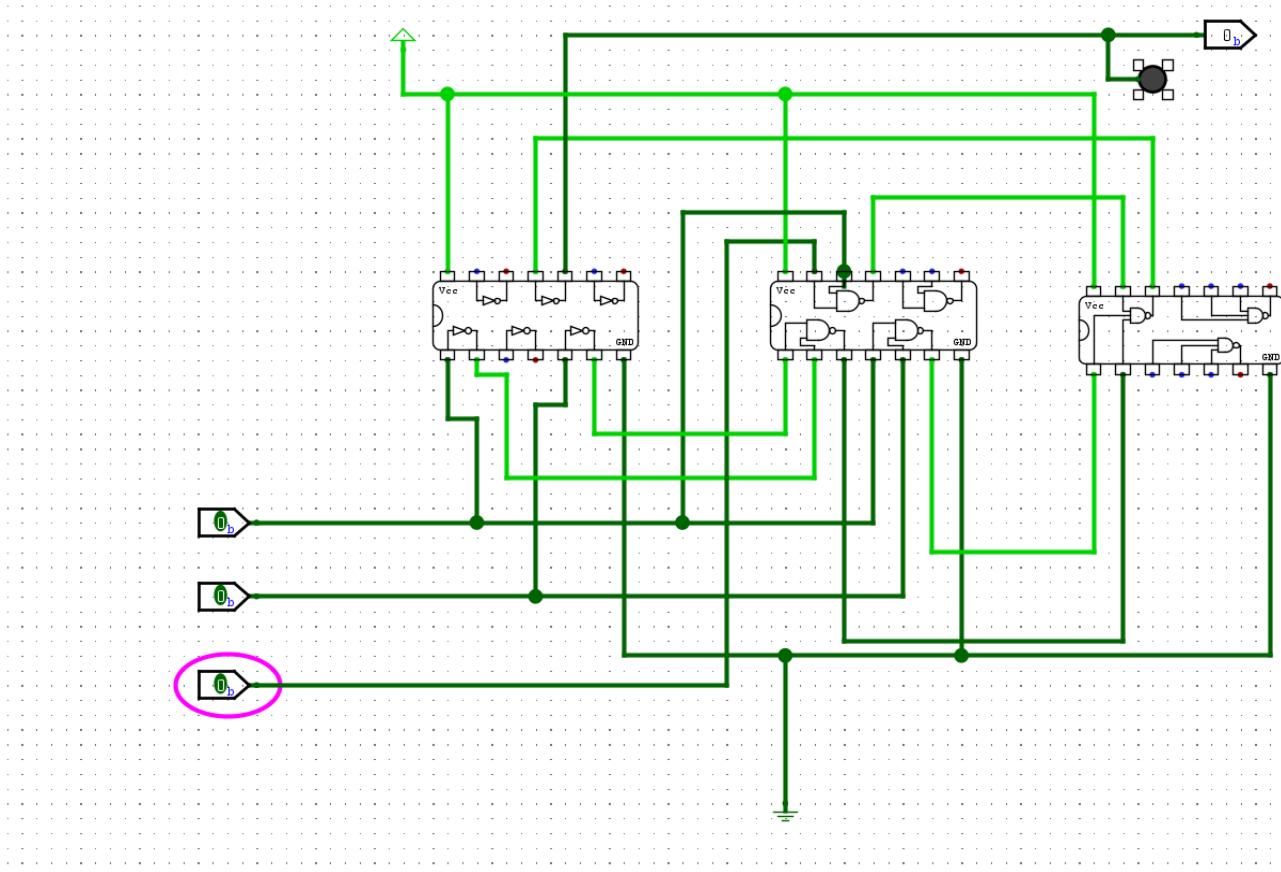
Μετά την έξοδο του **74LS04** έχουμε:

$$F(A, B, C, D) = ((A'B')'(AB)'(AD)')'' = (A'B')'(AD)'(AB)'$$

(Επόμενη σελίδα)

KΥΚΛΩΜΑ (screenshot):

ΓΙΑ Α=0 Β=0 Δ=0 Σ=Χ ΕΧΟΥΜΕ:



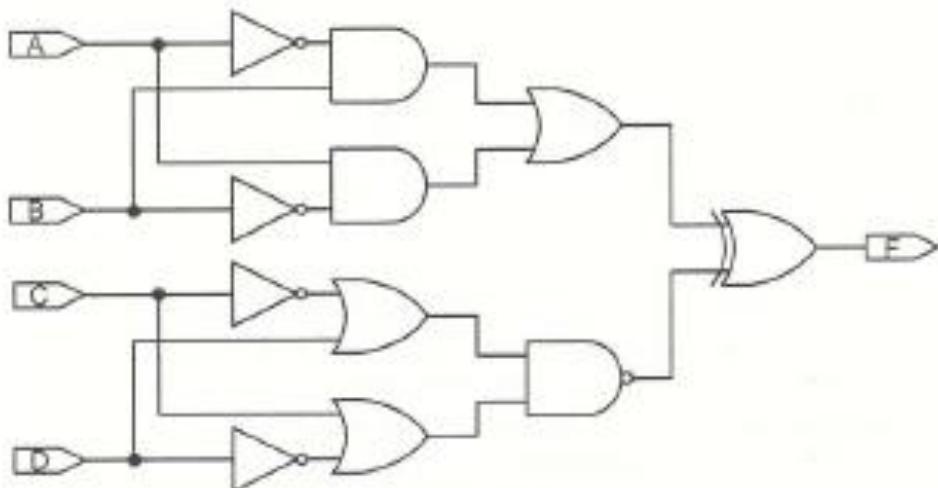
3. Γιατί στις υλοποιήσεις συνδυαστικών κυκλωμάτων υπάρχει η συνήθεια να γίνεται η υλοποίηση με ένα μόνο είδος πύλης;

ΑΠΑΝΤΗΣΗ:

Στις υλοποιήσεις συνδυαστικών κυκλωμάτων, υπάρχει η συνήθεια να γίνεται η υλοποίηση με ένα μόνο είδος πύλης διότι με αυτό τον τρόπο, μειώνεται το κόστος. Φτιάχνοντας το κύκλωμά μας με ένα μόνο είδος πύλης, εάν το παραγγείλουμε σε πολύ μεγάλες ποσότητες θα μειώσουμε την τιμή και ενδεχομένως και τον αριθμό των chips.

Ιδιαίτερα, επιλέγω για την υλοποίηση μου πύλες NAND ή NOR με τις οποίες μπορώ να φτιάξω οποιαδήποτε πύλη διότι έχουν την ιδιότητα της καθολικότητας. Τα πλεονεκτήματα αυτών των πυλών είναι ότι είναι οι πιο απλές στην αγορά (απαιτούν λιγότερα transistor), άρα είναι και πιο φθηνές.

4. Αναλύστε και υλοποιήστε το κύκλωμα του σχήματος 2. Τι κάνει το κύκλωμα αυτό ;



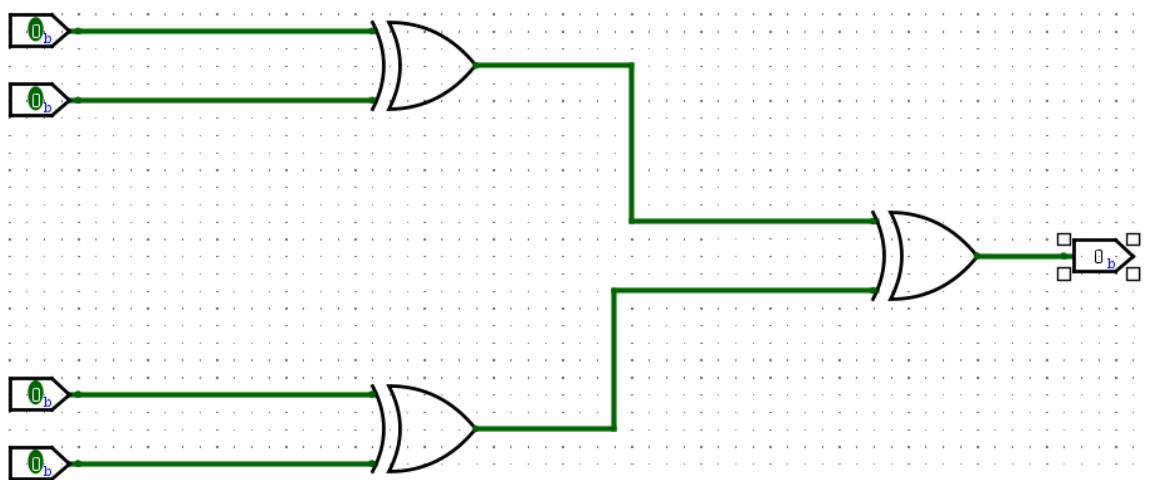
Σχήμα 2

ΑΠΑΝΤΗΣΗ:

Η δουλειά του παραπάνω κυκλώματος είναι να υπολογίζει το PARITY BIT.

Αυτό το καταλαβαίνουμε, υλοποιώντας το παραπάνω κύκλωμα με τον παρακάτω τρόπο:

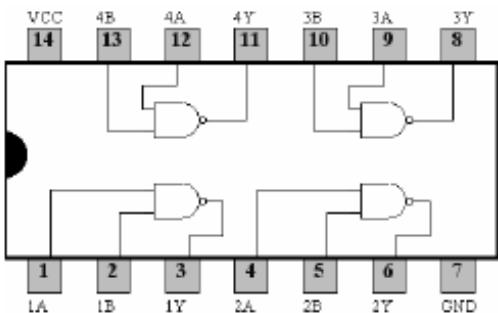
Όλες οι είσοδοι έχουν την τιμή 0:



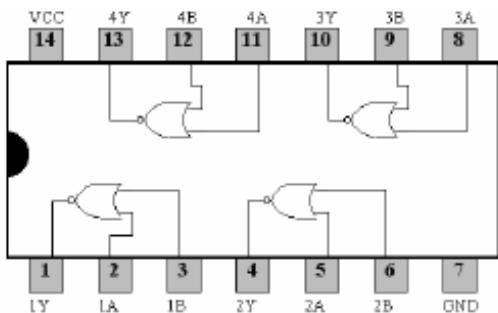
ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

ΠΑΡΑΡΤΗΜΑ

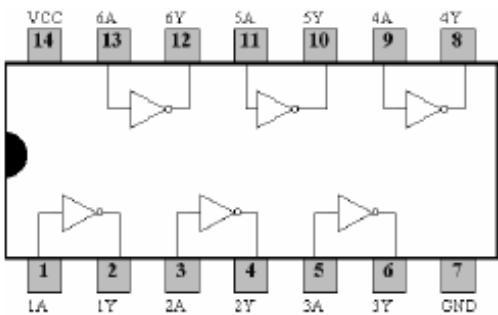
Διαγράμματα Ολοκληρωμένων



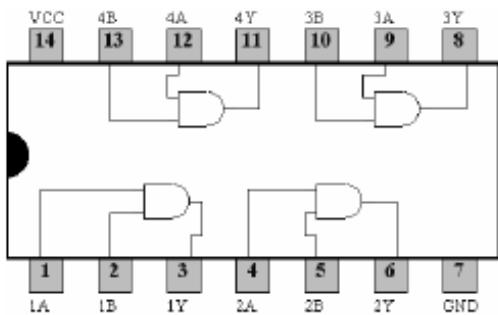
7400 Quad 2-Input NAND Gate



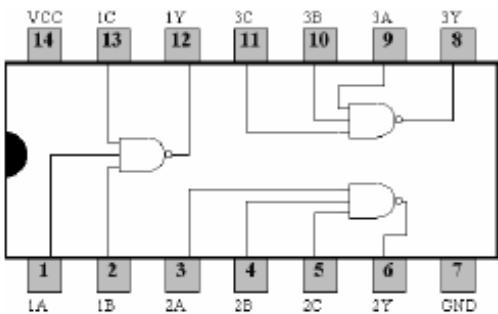
7402 Quad 2-Input NOR Gate



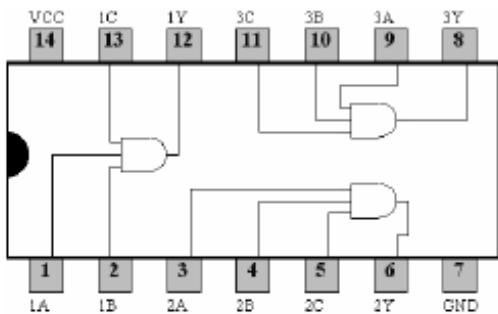
7404 Hex Inverter



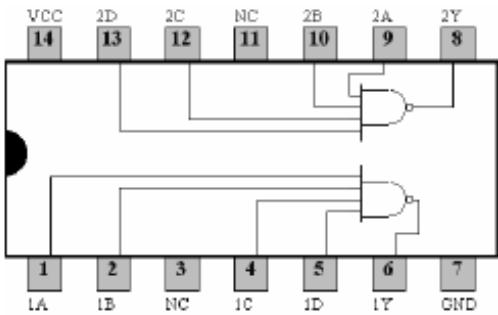
7408 Quad 2-Input AND Gate



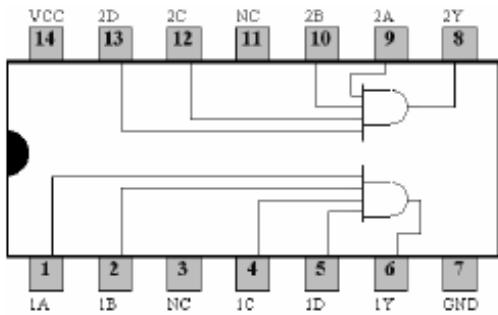
7410 Triple 3-Input NAND Gate



7411 Triple 3-Input AND Gate

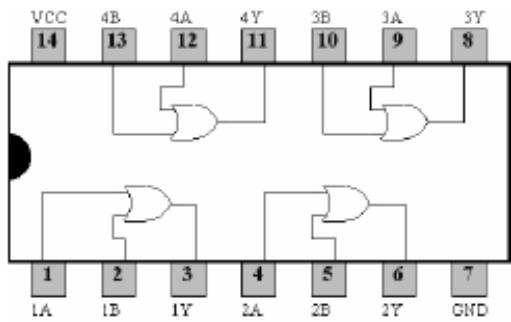


7420 Dual 4-Input NAND Gate

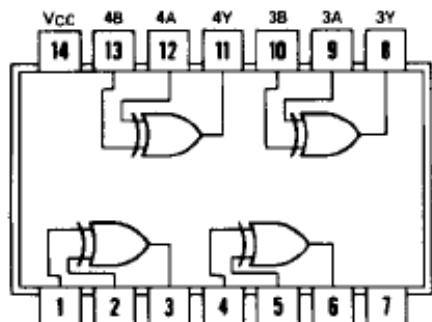


7421 Dual 4-Input AND Gate

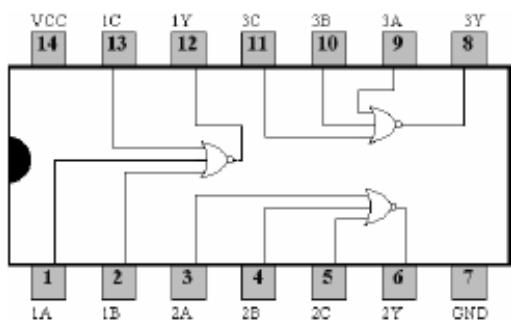
Σελίδα 11



7432 Quad 2-Input OR Gate



7486 Quad 2-Input XOR Gate

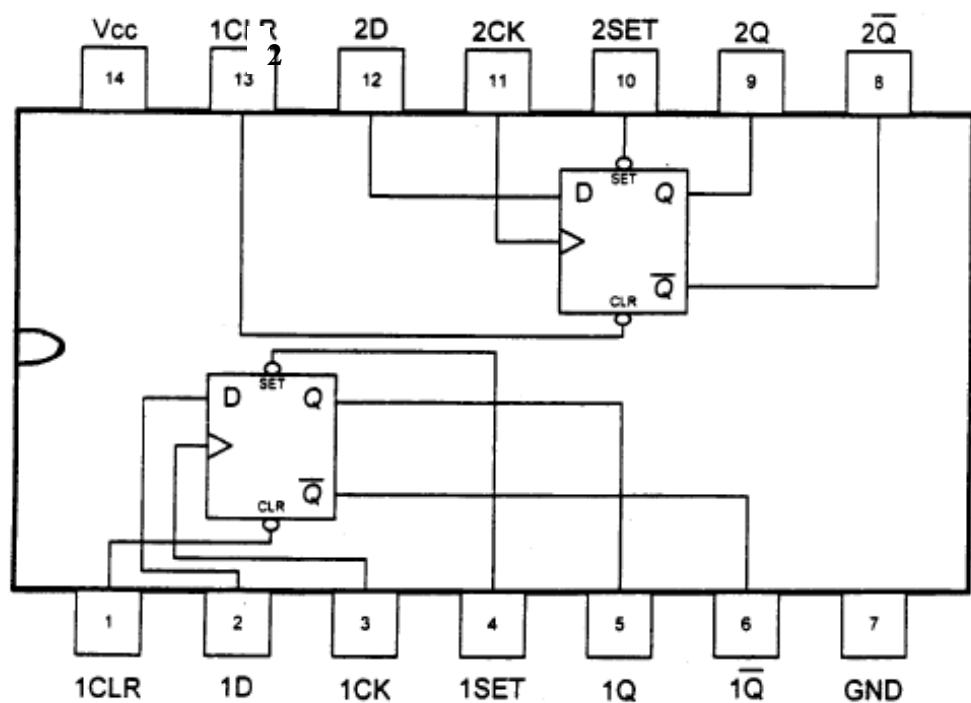


7427 Triple 3-Input NOR Gate

**DUAL D-TYPE POSITIVE EDGE TRIGGERED
FLIP-FLOPS WITH PRESET AND CLEAR**

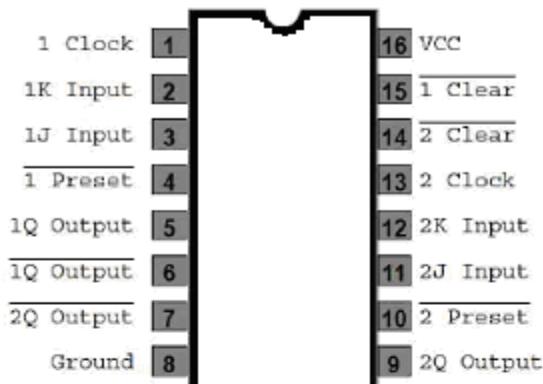
FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	Q'
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q ₀	Q' ₀



DUAL JK FLIP_FLOP with CLEAR/PRESET

74112



Function Table

Inputs					Outputs	
PR	CLR	CLK	J	K	Q	\bar{Q}
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H (Note 1)	H (Note 1)
H	H	↓	L	L	Q_0	\bar{Q}_0
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	Toggle	
H	H	H	X	X	Q_0	\bar{Q}_0

H = HIGH Logic Level

L = LOW Logic Level

X = Either LOW or HIGH Logic Level

↓ = Negative Going Edge of Pulse

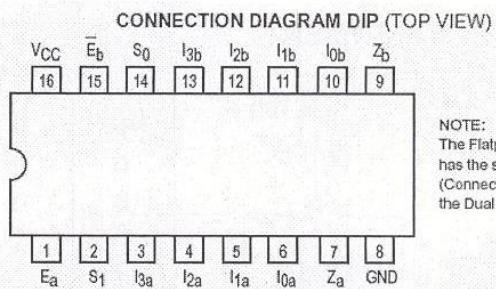
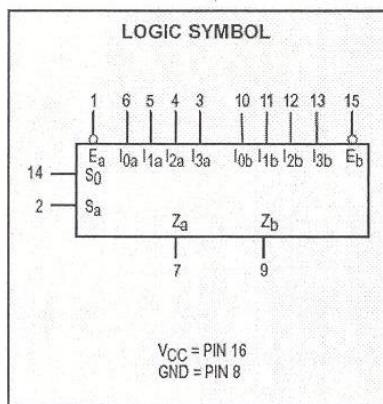
Q_0 = The output logic level before the indicated input conditions were established.

Toggle = Each output changes to the complement of its previous level on each falling edge of the clock pulse.

Note 1: This configuration is nonstable; that is, it will not persist when preset and/or clear inputs return to their inactive (HIGH) level.

74153

DUAL 4-INPUT MULTIPLEXER



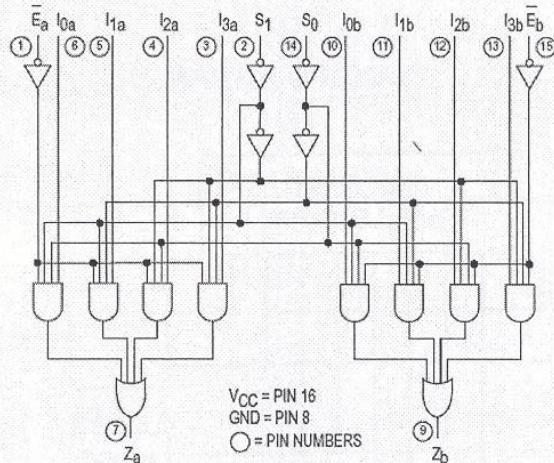
PIN NAMES

S ₀	Common Select Input
E	Enable (Active LOW) Input
I _{0, I₁}	Multiplexer Inputs
Z	Multiplexer Output (Note b)

LOADING (Note a)

HIGH	LOW
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.

LOGIC DIAGRAM



TRUTH TABLE

SELECT INPUTS		INPUTS (a or b)				OUTPUT	
S ₀	S ₁	E	I ₀	I ₁	I ₂	I ₃	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	X	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level

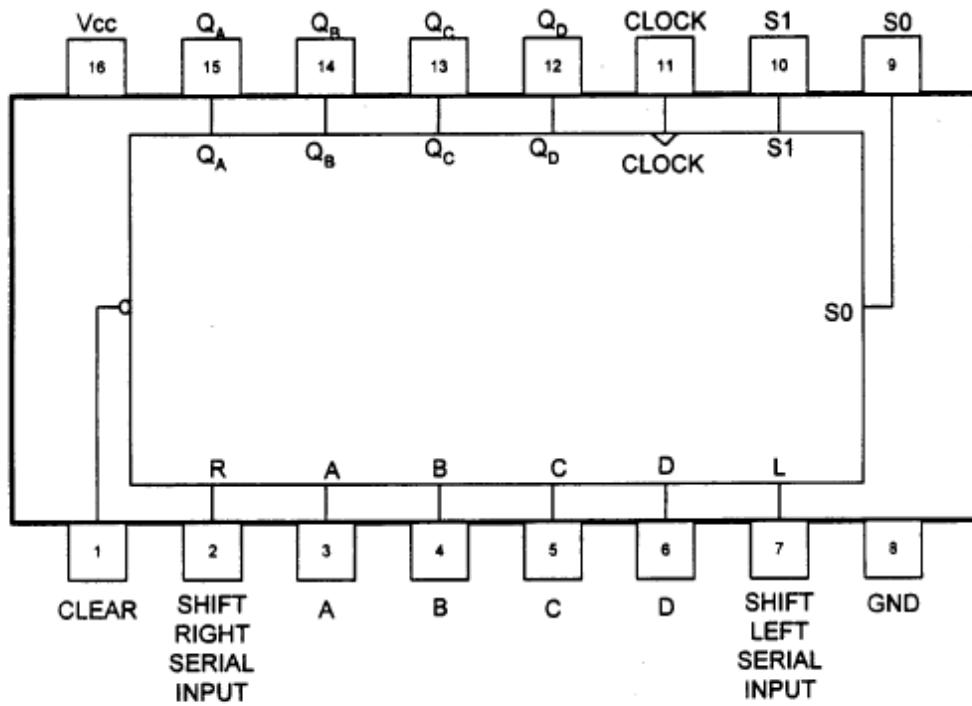
L = LOW Voltage Level

X = Don't Care

4 - BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER

FUNCTION TABLE

CLEAR	MODE		CLOCK	SERIAL		PARALLEL				OUTPUTS			
	S ₁	S ₀		LEFT	RIGHT	A	B	C	D	Q _A	Q _B	Q _C	Q _D
L	X	X	X	X	X	X	\bar{E}_a	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}
H	H	H	\uparrow	X	X	a	b	c	d	a	b	c	d
H	L	H	\uparrow	X	\bar{H}	X	X	X	X	H	Q _{A1}	Q _{B1}	Q _{C1}
H	L	H	\uparrow	X	L	X	X	X	X	L	Q _{A1}	Q _{B1}	Q _{C1}
H	H	L	\uparrow	H	X	X	X	X	X	Q _{B1}	Q _{C1}	Q _{D1}	H
H	H	L	\uparrow	L	X	X	X	X	X	Q _{B1}	Q _{C1}	Q _{D1}	L
H	L	L	X	X	X	X	X	X	X	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}



4008 4-BIT FULL ADDER

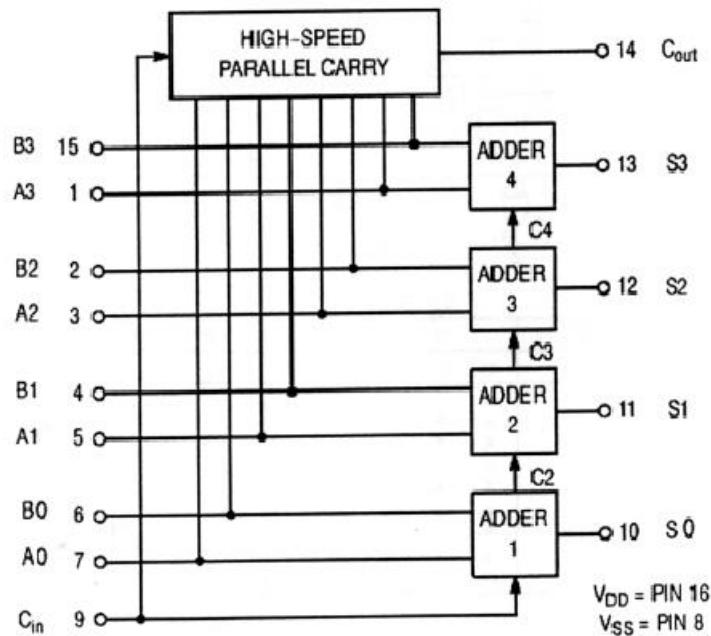
PIN ASSIGNMENT

A3	1	16	V _{DD}
B2	2	15	B3
A2	3	14	C _{out}
B1	4	13	S3
A1	5	12	S2
B0	6	11	S1
A0	7	10	S0
V _{SS}	8	9	C _{in}

TRUTH TABLE
(One Stage)

C _{in}	B	A	C _{out}	S
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

BLOCK DIAGRAM



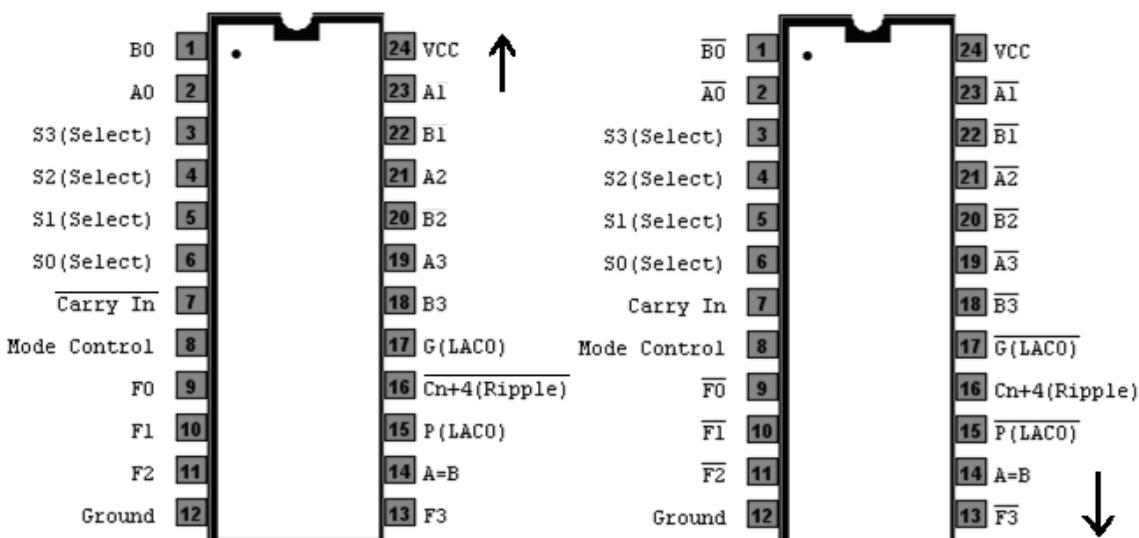
V_{DD} = PIN 16
V_{SS} = PIN 8

74LS181 – 4-bit ARITHMETIC LOGIC UNIT

TABLE 2

SELECTION				ACTIVE-HIGH DATA		
S3	S2	S1	S0	M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS	
					$\overline{C}_n = H$ (no carry)	$\overline{C}_n = L$ (with carry)
L	L	L	L	$F = \overline{A}$	$F = A$	$F = A \text{ PLUS } 1$
L	L	L	H	$F = \overline{A + B}$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L	L	H	L	$F = \overline{AB}$	$F = A + \overline{B}$	$F = (A + \overline{B}) \text{ PLUS } 1$
L	L	H	H	$F = 0$	$F = \text{MINUS } 1 \text{ (2's COMPL)}$	$F = \text{ZERO}$
L	H	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
L	H	L	H	$F = \overline{B}$	$F = (A + B) \text{ PLUS } \overline{AB}$	$F = (A + B) \text{ PLUS } \overline{AB} \text{ PLUS } 1$
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L	H	H	H	$F = \overline{AB}$	$F = AB \text{ MINUS } 1$	$F = \overline{AB}$
H	L	L	L	$F = \overline{A + B}$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H	L	L	H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H	L	H	L	$F = B$	$F = (A + \overline{B}) \text{ PLUS } AB$	$F = (A + \overline{B}) \text{ PLUS } AB \text{ PLUS } 1$
H	L	H	H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A^\dagger$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H	H	L	H	$F = \overline{A + B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	L	$F = A + B$	$F = (A + \overline{B}) \text{ PLUS } A$	$F = (A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

[†] Each bit is shifted to the next more significant position.



Σελίδα 18

TABLE 1

SELECTION				ACTIVE-LOW DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	Cn = L (no carry)	Cn = H (with carry)
L	L	L	L	F = \bar{A}	F = A MINUS 1	F = A
L	L	L	H	F = \bar{AB}	F = AB MINUS 1	F = AB
L	L	H	L	F = $\bar{A} + B$	F = \bar{AB} MINUS 1	F = \bar{AB}
L	L	H	H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L	H	L	L	F = $\bar{A} + B$	F = A PLUS ($A + \bar{B}$)	F = A PLUS ($A + \bar{B}$) PLUS 1
L	H	L	H	F = \bar{B}	F = AB PLUS ($A + \bar{B}$)	F = AB PLUS ($A + \bar{B}$) PLUS 1
L	H	H	L	F = $A \oplus B$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = $A + \bar{B}$	F = $A + \bar{B}$	F = $(A + \bar{B})$ PLUS 1
H	L	L	L	F = \bar{AB}	F = A PLUS ($A + B$)	F = A PLUS ($A + B$) PLUS 1
H	L	L	H	F = $A \oplus B$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = \bar{AB} PLUS ($A + B$)	F = \bar{AB} PLUS ($A + B$) PLUS 1
H	L	H	H	F = $A + B$	F = $(A + B)$	F = $(A + B)$ PLUS 1
H	H	L	L	F = 0	F = A PLUS A^{\dagger}	F = A PLUS A PLUS 1
H	H	L	H	F = \bar{AB}	F = AB PLUS A	F = AB PLUS A PLUS 1
H	H	H	L	F = AB	F = \bar{AB} PLUS A	F = \bar{AB} PLUS A PLUS 1
H	H	H	H	F = A	F = A	F = A PLUS 1

[†]Each bit is shifted to the next more significant position.