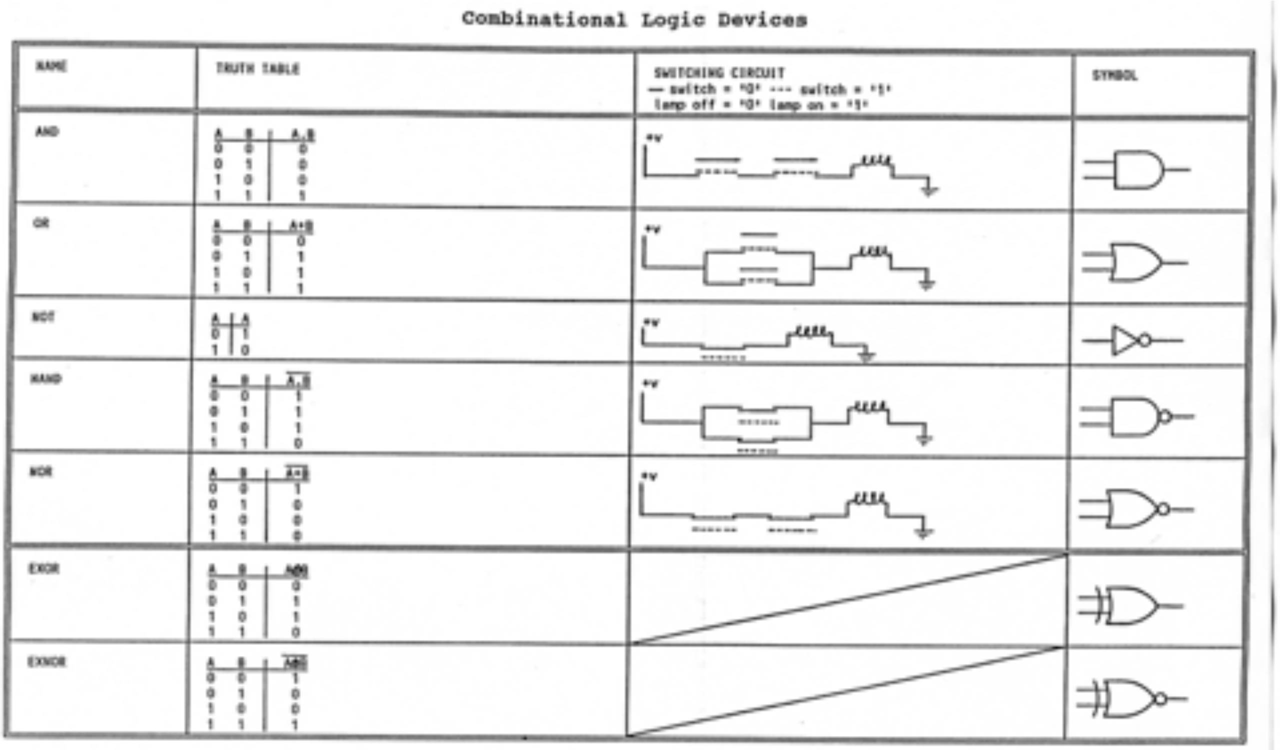
Implementation of Logic

pp14-18



Combinational Logic Devices

NAME	TRUTH TABLE	SWITCHING CIRCUIT — switch = '0' --- switch = '1' lamp off = '0' lamp on = '1'	SYMBOL
AND	A B | A.B: 0 0 | 0; 0 1 | 0; 1 0 | 0; 1 1 | 1	+V	
OR	A B | A+B: 0 0 | 0; 0 1 | 1; 1 0 | 1; 1 1 | 1	+V	
NOT	A | $\overline{A}$: 0 | 1; 1 | 0	+V	
NAND	A B | $\overline{A.B}$: 0 0 | 1; 0 1 | 1; 1 0 | 1; 1 1 | 0	+V	
NOR	A B | $\overline{A+B}$: 0 0 | 1; 0 1 | 0; 1 0 | 0; 1 1 | 0	+V	
EXOR	A B | $A \oplus B$: 0 0 | 0; 0 1 | 1; 1 0 | 1; 1 1 | 0		
EXNOR	A B | $\overline{A \oplus B}$: 0 0 | 1; 0 1 | 0; 1 0 | 0; 1 1 | 1		

a. The applications in digital circuits
b. Represent the validity using volts.
    a. TRUE = '1' = 5V
    b. FASLE = '0' = 0V
c. These logic functions are implemented using logic gates.

AND
Connecting two switches in serial gives you an AND gate. This means that only when both switches are down does current flow.

OR
Connecting two switches in parallel gives you an OR gate. This means that when switches A or B are closed the entire circuit is closed and current may flow.

NOT
Not is an inverter, it is simply a switch that is normally on, and disconnected when on.

d. The gates can be combined…

NOT(AND) = NAND
NOT(OR) = NOR

HW: Make a NOR into a NOT

NOR TABLE

A	B	A.B	!(A+B)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NAND TABLE

A	B	A.B	!(A.B)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

AND Table

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

OR Table

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

NOT Table

A	!A
0	1
1	0

Truth Tables and Logic Functions

pp 18 - 20, 23 - 24

- Gates can be larger - they can have as many inputs as you need, for example a 3 input AND gate. To represent this we need larger truth tables.

A	B	C	A.B.C	A+A+C
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- Complete set of logic gates - 2 inputs give 4 combinations, outputs have 2 values so there are 2^4 sets of outputs.

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Case 6: Exclusive OR (EXOR): True when the inputs differ. Notated as $A \oplus B$

$A \oplus B = A \cdot B + A \cdot B$

Case 9: Exclusive NOR (EXNOR): True when inputs are the same.

0	FALSE
1	AND
2	
3	A
4	
5	B
6	EXOR
7	OR
8	NOR
9	EXNOR
10	NOT(B)
11	
12	NOT(A)
13	
14	NOT(A.B)
15	TRUE