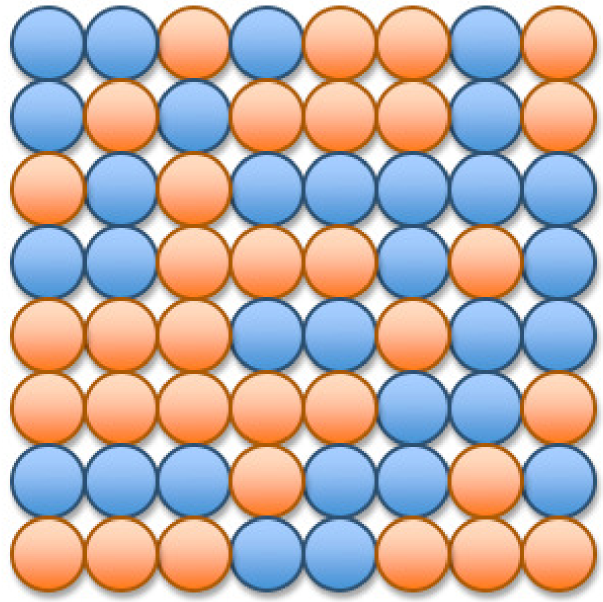


UNIT 8A

Computer Organization:
Boolean Logic, Computational Circuits

Boolean Logic & Truth Tables

- Computer circuitry works based on Boolean logic: operations on true (1) and false (0) values.

A	B	A ∧ B (A AND B) (conjunction)	A ∨ B (A OR B) (disjunction)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

A	¬A (NOT A) (negation)
0	1
1	0

Exercise

- Write the truth table for

$E = \boxed{A \land B \land C}$ ✓

A	B	C	E
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
			0
1	1	1	1

$\neg A \lor B$

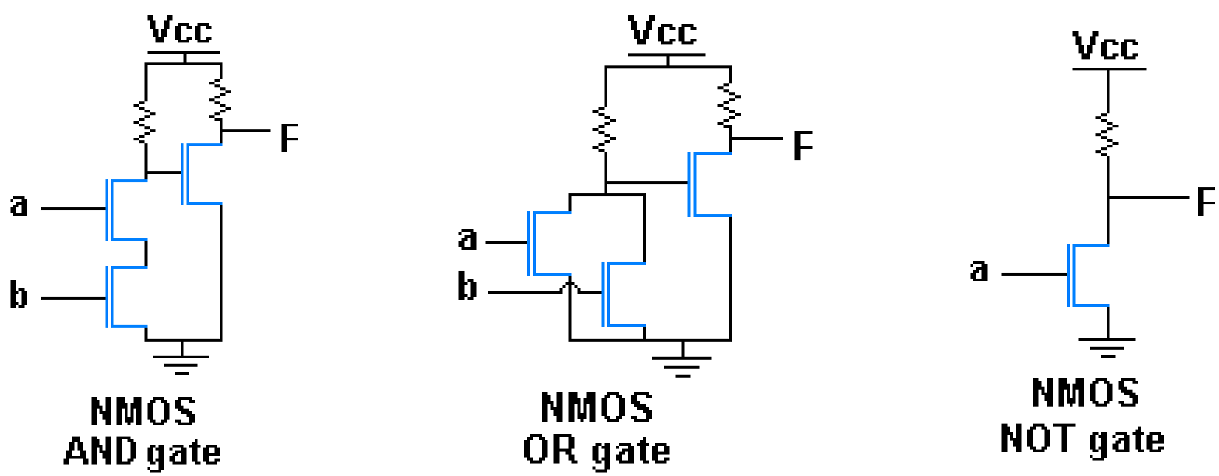
A	B	
0	0	1
0	1	1
1	0	0
1	1	1

$\neg A \lor B \land A$

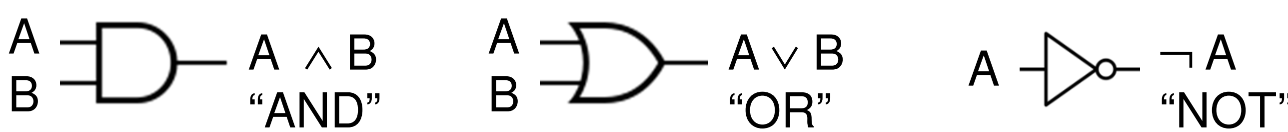
do this

Gates

- A gate is an electronic device that implements a logical function. (Images from Wikipedia)
- Circuit diagram:



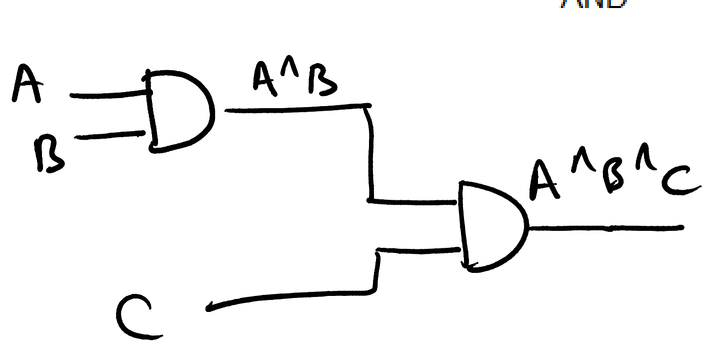
- Abstract Diagram:



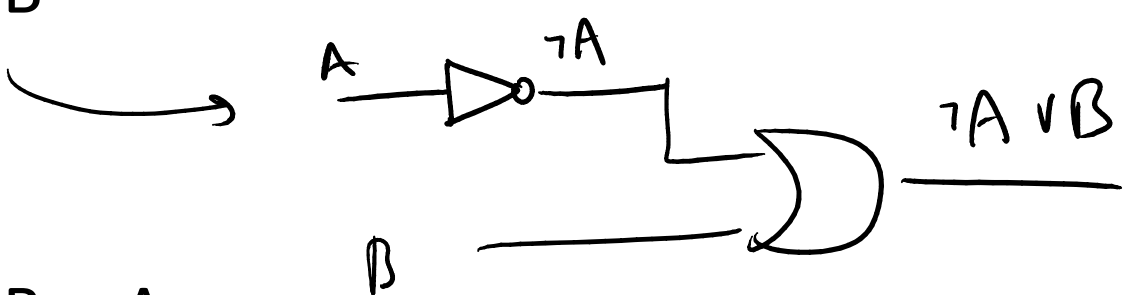
Exercise

- Draw the following using ➔ 

$(A \wedge B) \wedge C$



$\neg A \vee B$



$\neg A \vee B \wedge A$

Properties (Similar to × and +)

- Commutative: $a \wedge b = b \wedge a$ $\qquad$ $a \vee b = b \vee a$
- Associative: $a \wedge b \wedge c = (a \wedge b) \wedge c = a \wedge (b \wedge c)$
  $a \vee b \vee c = (a \vee b) \vee c = a \vee (b \vee c)$
- Distributive: $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
  $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ ← Not true for + and ×
- Identity: $a \wedge 1 = a$ $\qquad$ $a \vee 0 = a$
- Dominance: $a \wedge 0 = 0$ $\qquad$ $a \vee 1 = 1$
- Idempotence: $a \wedge a = a$ $\qquad$ $a \vee a = a$
- Complementation: $a \wedge \neg a = 0$ $\qquad$ $a \vee \neg a = 1$
- Double Negation: $\neg \neg a = a$

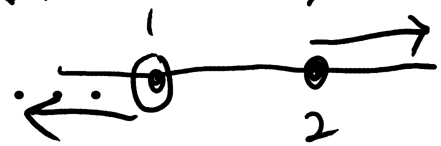
DeMorgan's Law

Nand: $\neg(A \land B) = \neg A \lor \neg B$ OR

not(x ≥ 1 $\boxed{\text{and}}$ x < 2)

⟺ x < 1 or x ≥ 2

`if not (x > 15 and x < 110) then ...`



is logically equivalent to

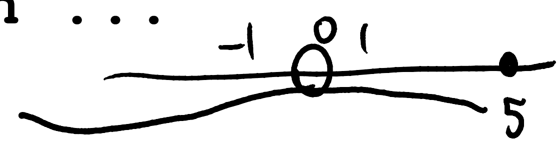
`if (not x > 15) or (not x < 110) then ...`

Nor: $\neg(A \lor B) = \neg A \land \neg B$

not (x == 0 or x > 5)

⟺ x != 0 and x ≤ 5

`if not (x < 15 or x > 110) then ...`



is logically equivalent to

`if (not x < 15) and (not x > 110) then ...`

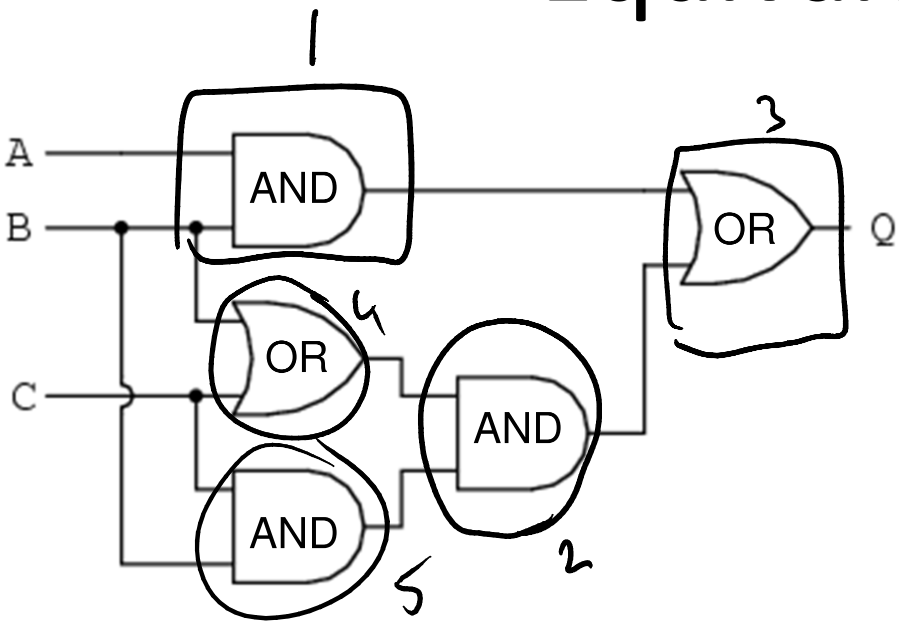
Simplify/rewrite

$A \wedge A = A$

- $(A \wedge A) \wedge A = A$ $\longrightarrow (A) \wedge A \rightarrow A$
- $A \wedge (B \wedge \neg B) \Rightarrow A \wedge 0 \Rightarrow 0$
- $\neg((A \vee B) \wedge C)$ $\Rightarrow \neg(A \vee B) \vee \neg C$

  $\Rightarrow \neg A \wedge \neg B \vee \neg C$

Equivalence



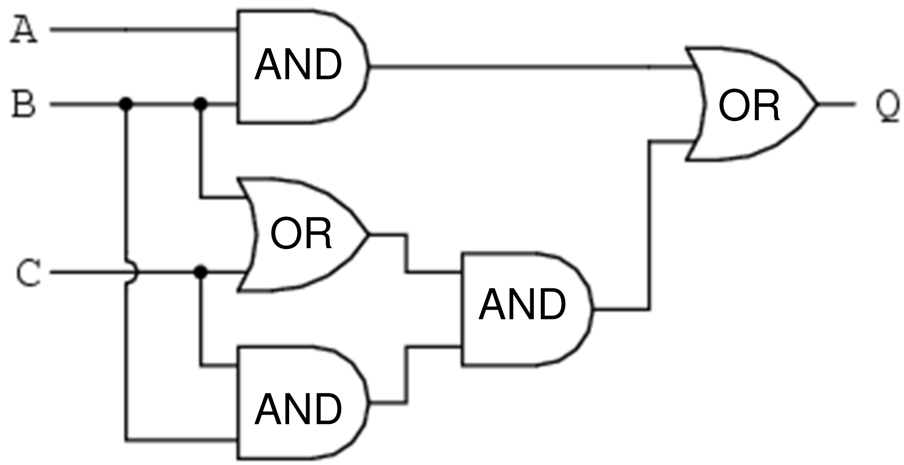
$$Q = (A \wedge B) \boxed{\vee} ((B \vee C) \wedge (C \wedge B))$$

1 3 4 2 5

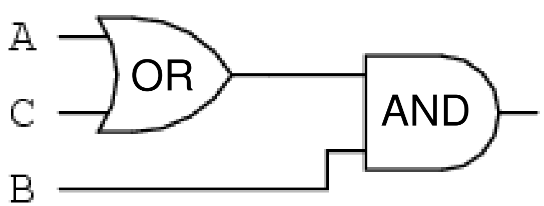
A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	✓
0	1	1	✓
1	0	0	✓
1	0	1	
1	1	0	
1	1	1	

http://www.allaboutcircuits.com/vol_4/chpt_7/6.html

Equivalence



$Q = (A \wedge B) \vee ((B \vee C) \wedge (C \wedge B))$



$Q = B \wedge (A \vee C)$

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

http://www.allaboutcircuits.com/vol_4/chpt_7/6.html

Show equivalence

- $Q = \underbrace{(A \wedge B)}_{P} \vee (\underbrace{(B \vee C)}_{Q} \wedge \underbrace{(C \wedge B)}_{R})$
- $Q = B \wedge (A \vee C)$

$P \vee Q \wedge P \vee R$

$(A \wedge B) \vee Q$

$A \vee Q \wedge B \vee Q$

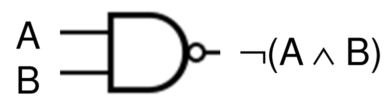
$A \vee (B \vee C) \wedge B \vee (B \vee C)$

A

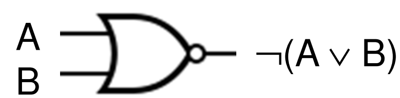
More gates

A	B	A nand B	A nor B	A xor B
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0

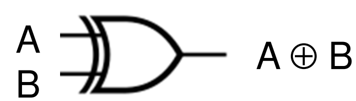
- nand (“not and”): A nand B = not (A and B)



- nor (“not or”): A nor B = not (A or B)



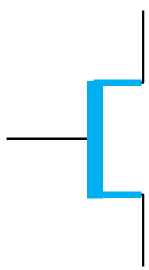
- xor (“exclusive or”):
  A xor B = (A and not B) or (B and not A)



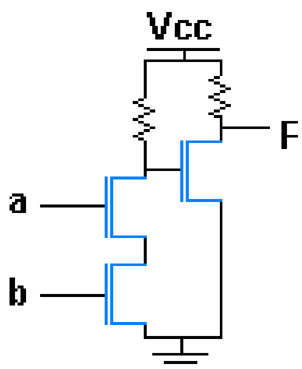
Exercise

- $(\neg A \wedge \neg B)$ using a NAND gate

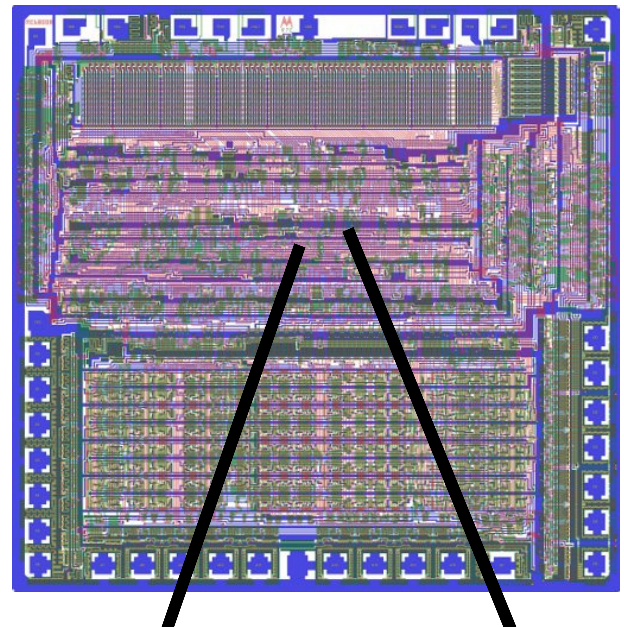
Gates Are Built From Transistors

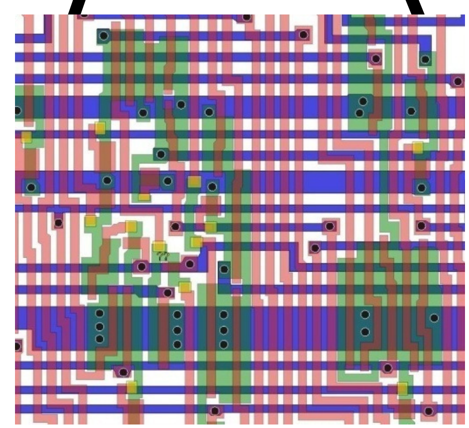


Transistor (switch)



NMOS
AND gate





Every spot where a red line crosses a green line is a transistor.