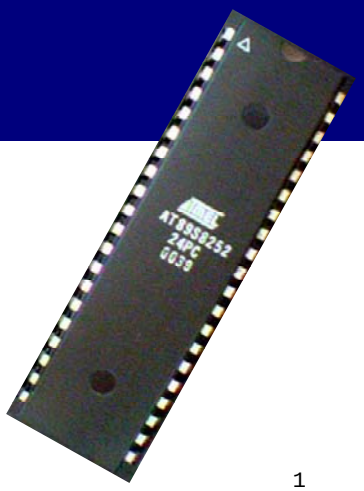
Digital Integrated Circuits & Microcontrollers

Chapter 3. Logic gates



Classification of Digital Circuits

- **Combinational logic circuits.**
  - Output depends only on present input.
- **Sequential circuits.**
  - Output depends on present input and present state of the circuit.

Logic Families

- **Transistor Transistor Logic (TTL)** is one of the most popular and widespread of all logic families.
  - Very high number of SSI and MSI devices available in the market.
  - Several number of sub-families that provide a wide range of speed and power consumption.

Logic Families

- **Sub families:**
  - **74*xx*** : The original TTL family.
    - These devices had a propagation delay of 10ns and a power consumption of 10mW, and they were introduced in the early 60's.
  - **74H*xx*** : High speed.
    - Speed was improved by reducing the internal resistors. Note that this improvement caused an increase in the power consumption.
  - **74L*xx*** : Low power.
    - Power consumption was improved by increasing the internal resistances, and the speed decreased.

Logic Families

- **74S*xx*** : Schottky.
  - The use of Schottky transistors improved the speed. The power dissipation is less than the 74H*xx* sub-family.
- **74LS*xx*** : Low power Schottky.
  - Uses Schottky transistors to improve speed. High internal resistances improves power consumption.
- **74AS*xx*** : Advanced Schottky.
  - Twice as fast as 74S*xx* with approximately the same power dissipation.

Logic Families

- **74ALS*xx*** : Advanced Low power Schottky.
  - Lower power consumption and higher speed than 74LS*xx* .
- **74F*xx*** : Fast.
  - Performance is between 74AS*xx* and 74ALS*xx*.

- Most TTL sub-families have a corresponding 54-series (military) version, and these series operate in a wider temperature and voltage ranges.

Logic Families

- **Complementary Metal Oxide Semiconductor (CMOS)** replaced TTL devices in the 90's due to advances in the design of MOS circuits made in mid 80's.
- **Advantages:**
  - Operate with a wider range of voltages that any other logic family.
  - Has high noise immunity.
  - Dissipates very low power at low frequencies.
  - It requires an extremely low driving current.
  - High fanout.

Logic Families

- **Disadvantages:**
  - Power consumption increases with frequency.
  - Susceptible to ESD - electro-static discharges.
- **Sub-families:**
  - ***40xx*** **:** Original CMOS family.
    - Fairly slow, but it has a low power dissipation.
  - **74HC*xx* :** High speed CMOS.
    - Better current sinking and sourcing than 40*xx*. It uses voltage supply between 2 and 6 volts.
    - Higher voltage →higher speed.
    - Lower voltage →lower power consumption.

Logic Families

- **74HCT*xx*** : High speed CMOS, TTL compatible.
  - Better current sinking and sourcing than 40*xx*. It uses voltage supply of 5V. Compatible with TTL family.
- **74AC*xx*** : Advanced CMOS.
  - Very fast. It can source and sink high currents. Not TTL compatible.
- **74ACT*xx*** : Advanced CMOS, TTL compatible.
  - Same as 74AC*xx*, but it is compatible with TTL family.
- **74FCT*xx* :** Fast CMOS, TTL compatible.
  - It is faster and has lower power dissipation than the 74AC*xx* and 74ACT*xx* sub-families. Compatible with TTL family.

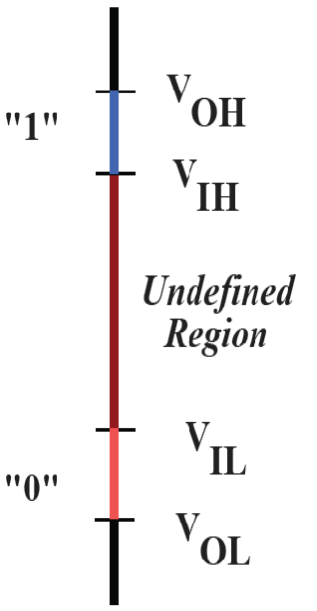
Logic Families

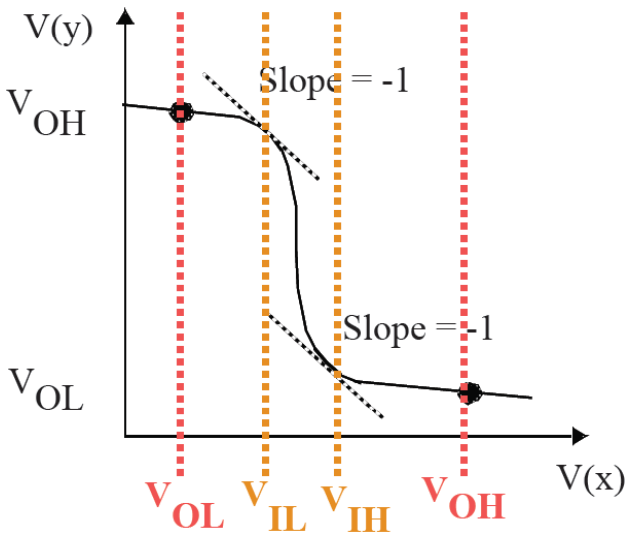
- Prefixes, to identify the manufacturer.
  - SN : Texas Instrument.
  - MN : Motorola.
  - DM : National
  - N : Signetics
  - P : Intel
  - H : Harris
  - AMD : Advanced Micro Devices
- Suffixes, identifies the packaging.
  - N : Plastic DIP (dual in-line package)
  - P : Plastic DIP
  - J : Ceramic DIP
  - W : Ceramic flat package.
  - D : Plastic ‘small outline’ package

Logic Circuits

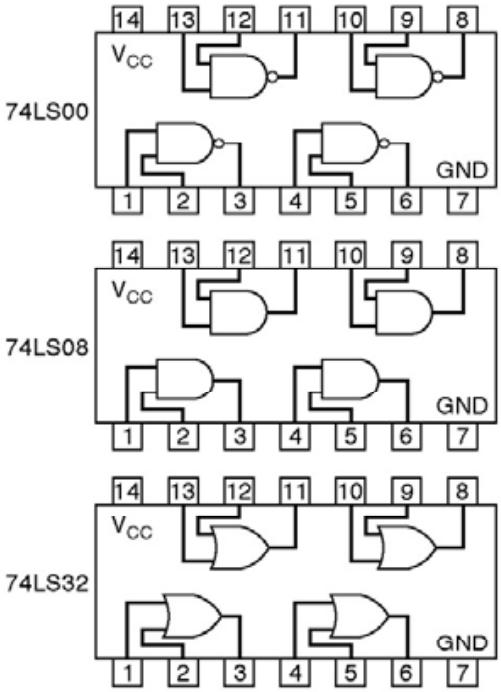
- Several voltage and current levels are of interest when working with logic gates, including:
  - V_{OL} = output voltage when the gate is LOW
  - V_{OH} = output voltage when the gate is HIGH
  - V_{IL} = input voltage when the gate is LOW
  - V_{IH} = input voltage when the gate is HIGH
  - I_{OL} = output current when the gate is LOW
  - I_{OH} = output current when the gate is HIGH
  - I_{IL} = input current when the gate is LOW
  - I_{IH} = input current when the gate is HIGH

Logic Circuits





Example



74LL Data Sheet

SN5400, SN54LS00, SN54S00,
SN7400, SN74LS00, SN74S00
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

DECEMBER 1983—REVISED MARCH 1988

- **Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs**
- **Dependable Texas Instruments Quality and Reliability**

description

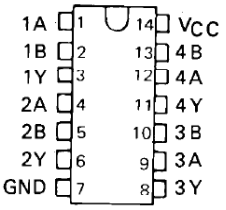
These devices contain four independent 2-input-NAND gates.

The SN5400, SN54LS00, and SN54S00 are characterized for operation over the full military temperature range of −55°C to 125°C. The SN7400, SN74LS00, and SN74S00 are characterized for operation from 0°C to 70°C.

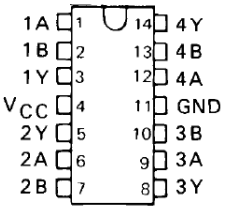
FUNCTION TABLE (each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

SN5400 . . . J PACKAGE
SN54LS00, SN54S00 . . . J OR W PACKAGE
SN7400 . . . N PACKAGE
SN74LS00, SN74S00 . . . D OR N PACKAGE
(TOP VIEW)



SN5400 . . . W PACKAGE
(TOP VIEW)



74LL Data Sheet

		SN5400			SN7400			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High-level input voltage	2			2			V
V_{IL}	Low-level input voltage			0.8			0.8	V
I_{OH}	High-level output current			−0.4			−0.4	mA
I_{OL}	Low-level output current			16			16	mA
T_A	Operating free-air temperature	−55		125	0		70	°C

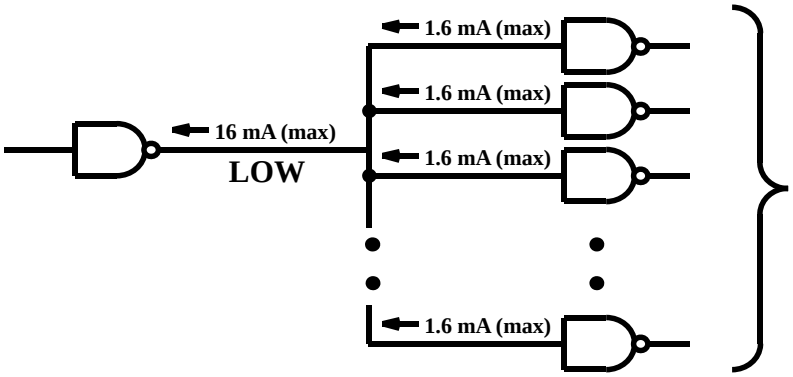
electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN5400			SN7400			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IK}	V_{CC} = MIN, I_I = −12 mA			−1.5			−1.5	V
V_{OH}	V_{CC} = MIN, V_{IL} = 0.8 V, I_{OH} = −0.4 mA	2.4	3.4		2.4	3.4		V
V_{OL}	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = 16 mA		0.2	0.4		0.2	0.4	V
I_I	V_{CC} = MAX, V_I = 5.5 V			1			1	mA
I_{IH}	V_{CC} = MAX, V_I = 2.4 V			40			40	μA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V			−1.6			−1.6	mA
I_{OS}§	V_{CC} = MAX	−20		−55	−18		−55	mA
I_{CCH}	V_{CC} = MAX, V_I = 0 V		4	8		4	8	mA
I_{CCL}	V_{CC} = MAX, V_I = 4.5 V		12	22		12	22	mA

Fanout

$$\text{fanout} = \left|\frac{I_{OL}(\max)}{I_{IL}(\max)}\right| \quad \text{or} \quad \text{fanout} = \left|\frac{I_{OH}(\max)}{I_{IH}(\max)}\right|$$

- The number of ***standard loads*** that the output can drive**.**
  - the number of standard loads is limited by the amount of input current each load requires as compared to the current that the driving gate can deliver.



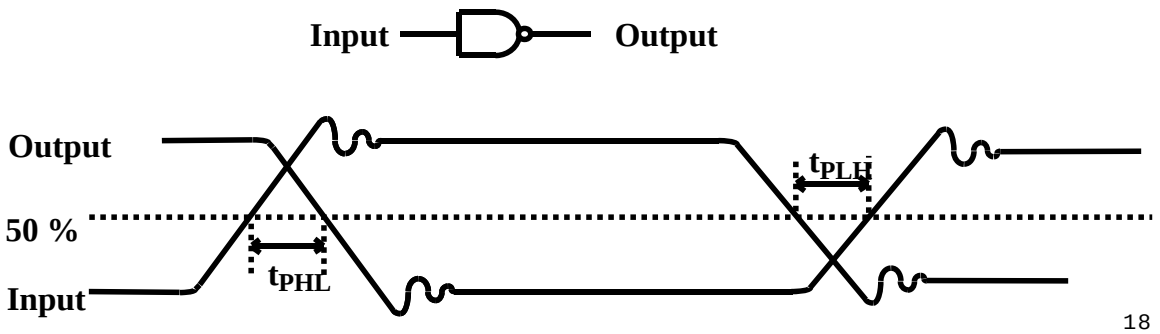
Fanout

$$\text{fanout} = \left|\frac{I_{OL}(\text{max})}{I_{IL}(\text{max})}\right| \quad \text{or} \quad \text{fanout} = \left|\frac{I_{OH}(\text{max})}{I_{IH}(\text{max})}\right|$$

- Fanout is much higher for CMOS devices than for TTL devices.
- I_{IL} and I_{IH} are extremely small for CMOS devices (< 1 uA).
- However, the input capacitance of CMOS gates affects propagation delay, so increased fanout results in increased delay.

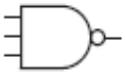
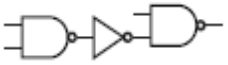
Propagation Delay

- The time that it takes a gate to switch logic levels
  - $\mathbf{t_{PLH}}$ = propagation delay when the OUTPUT switches from LOW to HIGH
  - $\mathbf{t_{PHL}}$ = propagation delay when the OUTPUT switches from HIGH to LOW
  - $\mathbf{t_D}$ = the maximum of $\mathbf{t_{PHL}}$ and $\mathbf{t_{PLH}}$

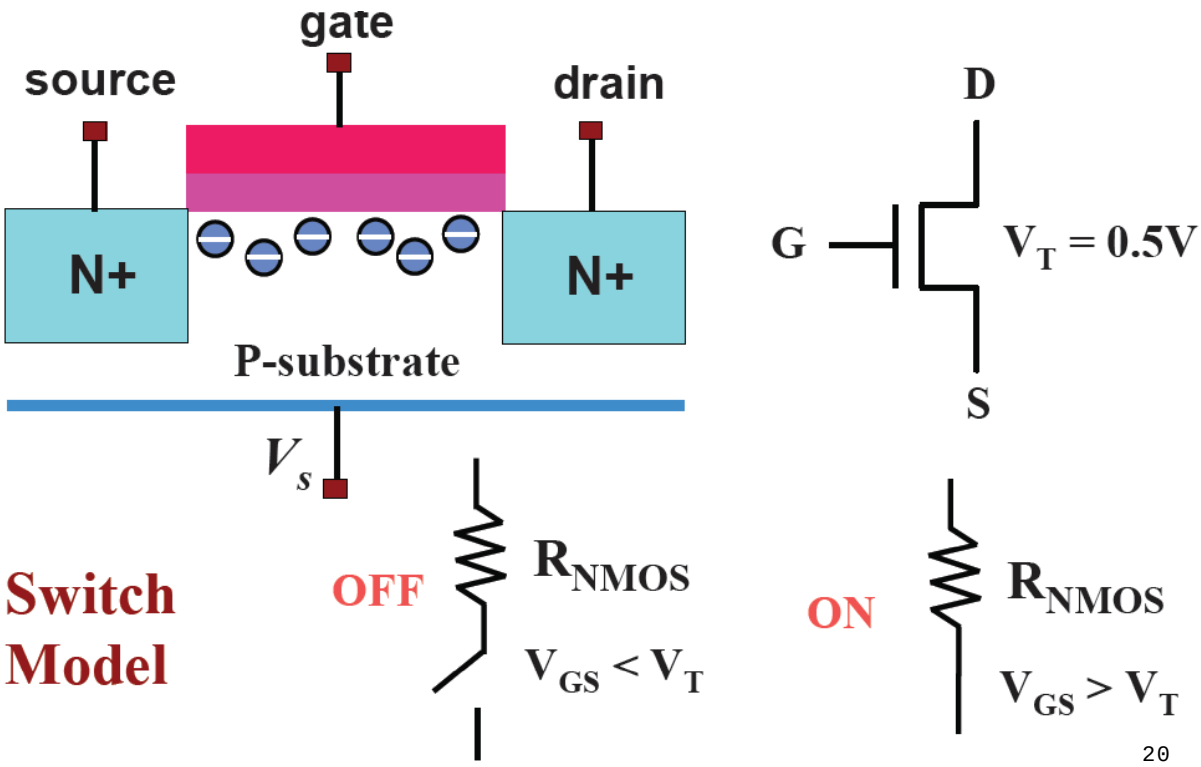


Propagation Delay

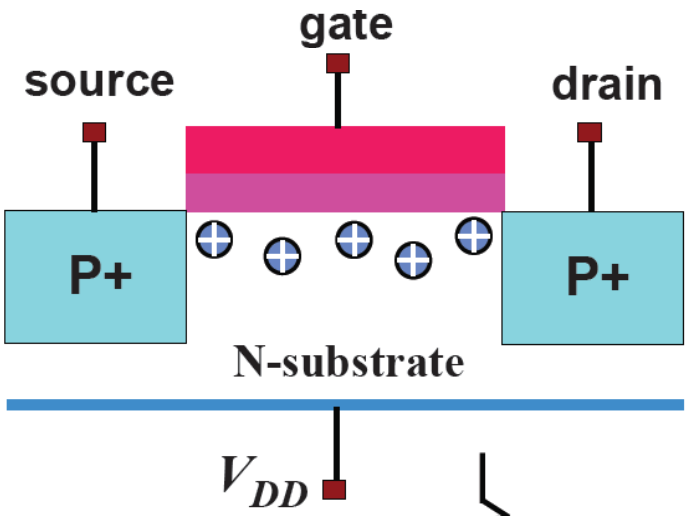
- Fanout affects propagation delay for CMOS gates.
  - the delay (in ns) for a 2-input NAND is
    $t_D = 0.05 + 0.014*SL$
    where SL is the number of ***standard loads***.

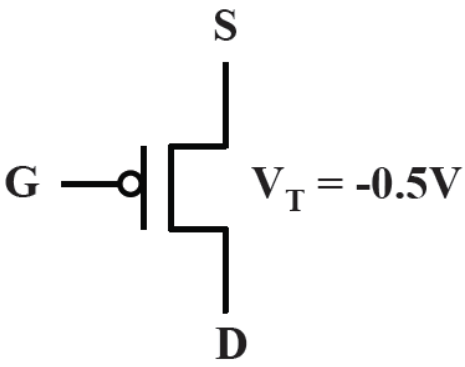
Cell Name	Cell Schematic	Normalized Area	Typical Input Load	Typical Input-to-Output Delay	Basic Function Templates
Inverter		1.00	1.00	0.04 + 0.012 × SL	
2NAND		1.25	1.00	0.05 + 0.014 × SL	
3NAND		1.50	1.00	0.06 + 0.017 × SL	

NMOS - *ON when Switch Input is High*



PMOS - *ON when Switch Input is Low*





Switch Model

OFF

R_{PMOS}

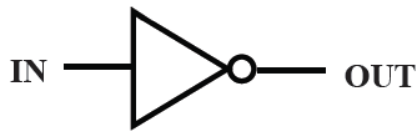
$V_{GS} > V_T$

ON

R_{PMOS}

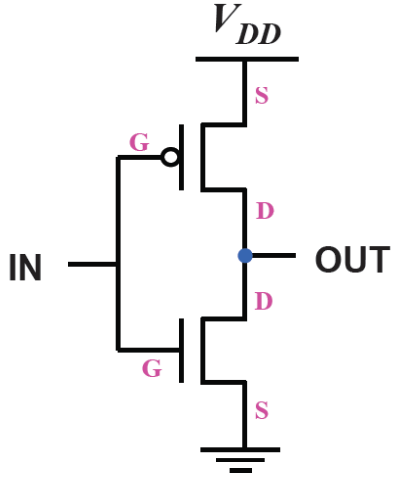
$V_{GS} < V_T$

The CMOS Inverter

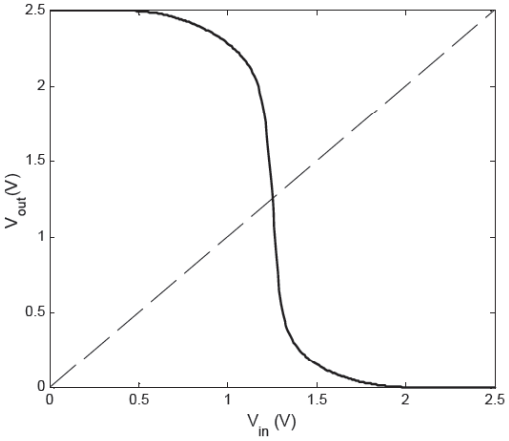


Truth Table

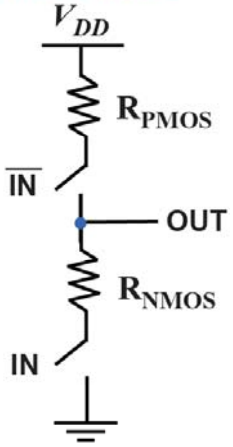
IN	OUT
0	1
1	0



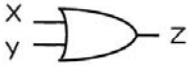
Rail-to-rail swing (LV to VDD)



Switch Model



Common Logic Gates

Gate	Symbol	Truth-Table	Expression
NAND		X Y Z 0 0 1 0 1 1 1 0 1 1 1 0	$Z = \overline{X \cdot Y}$
AND		X Y Z 0 0 0 0 1 0 1 0 0 1 1 1	$Z = X \cdot Y$
NOR		X Y Z 0 0 1 0 1 0 1 0 0 1 1 0	$Z = \overline{X + Y}$
OR		X Y Z 0 0 0 0 1 1 1 0 1 1 1 1	$Z = X + Y$

Common Logic Gates

XOR
$(X \oplus Y)$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

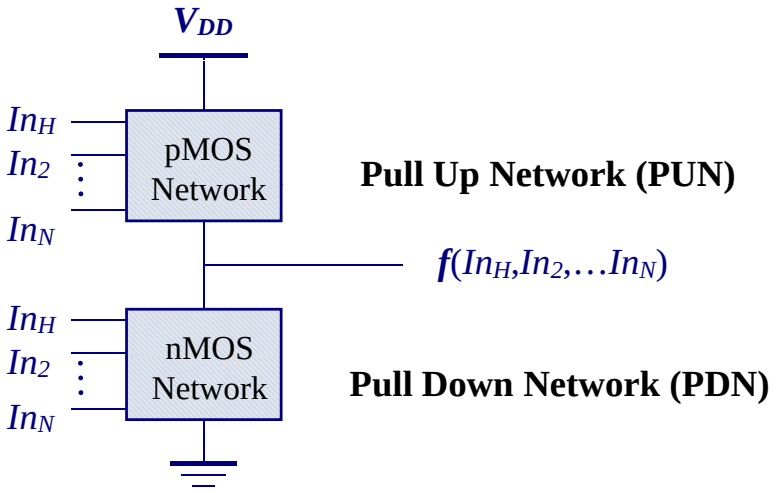
$Z = X\overline{Y} + \overline{X}Y$
X or Y but not both
("inequality", "difference")

XNOR
$\overline{(X \oplus Y)}$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

$Z = \overline{X}\,\overline{Y} + XY$
X and Y the same
("equality")

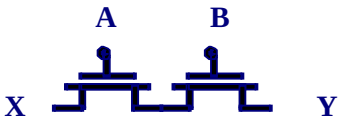
Static CMOS



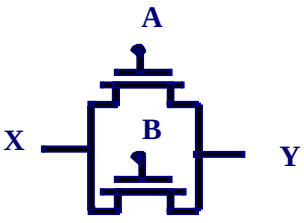
- **The complementary operation of a CMOS gate**
  - The nMOS network (PDN) is on and the pMOS network (PUN) is off
  - The pMOS network is on and the nMOS network is off.

NMOS Series/Parallel Connection

- Transistors can be thought as a switch controlled by its gate signal
- NMOS switch closes when switch control input is HIGH



Y = X if A AND B = AB

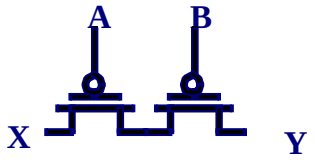


Y = X if A OR B = A+B

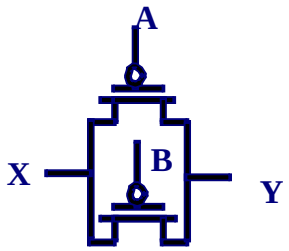
NMOS Transistors pass a "strong" L but a "weak" H

PMOS Series/Parallel Connection

- PMOS switch closes when switch control input is LOW



$Y = X$ if $\bar{A}$ AND $\bar{B} = \overline{A+B}$



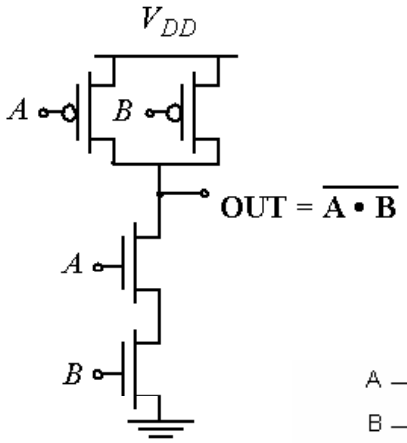
$Y = X$ if $\bar{A}$ OR $\bar{B} = \overline{AB}$

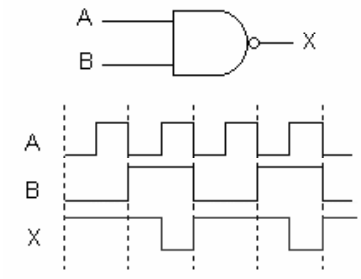
PMOS Transistors pass a "strong" H but a "weak" L

The NAND Gate

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

Truth Table of a 2 input NAND gate





PDN: G = A B $\Rightarrow$ Conduction to GND

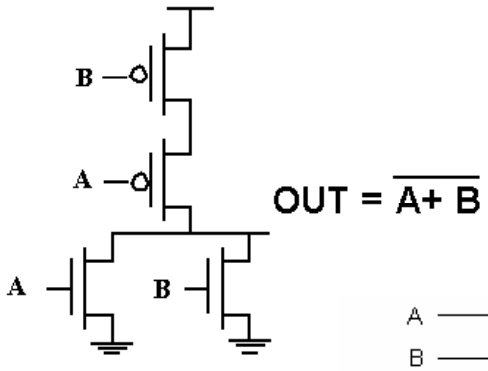
PUN: F= $\bar{A} + \bar{B} = \overline{AB} \Rightarrow$ Conduction to V_{DD}

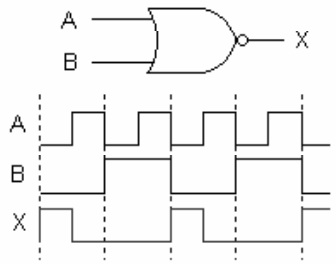
$$\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

The NOR Gate

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

Truth Table of a 2 input NOR gate

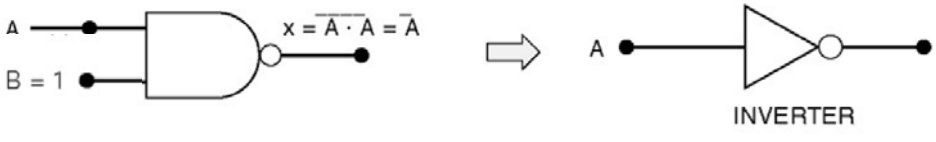




Inverter with more inputs gates

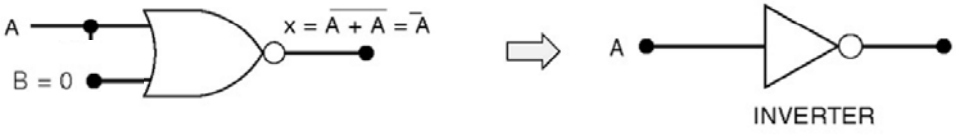
- NAND gate

A	B	X
0	1	1
1	1	0



- NOR gate

A	B	X
0	0	1
1	0	0

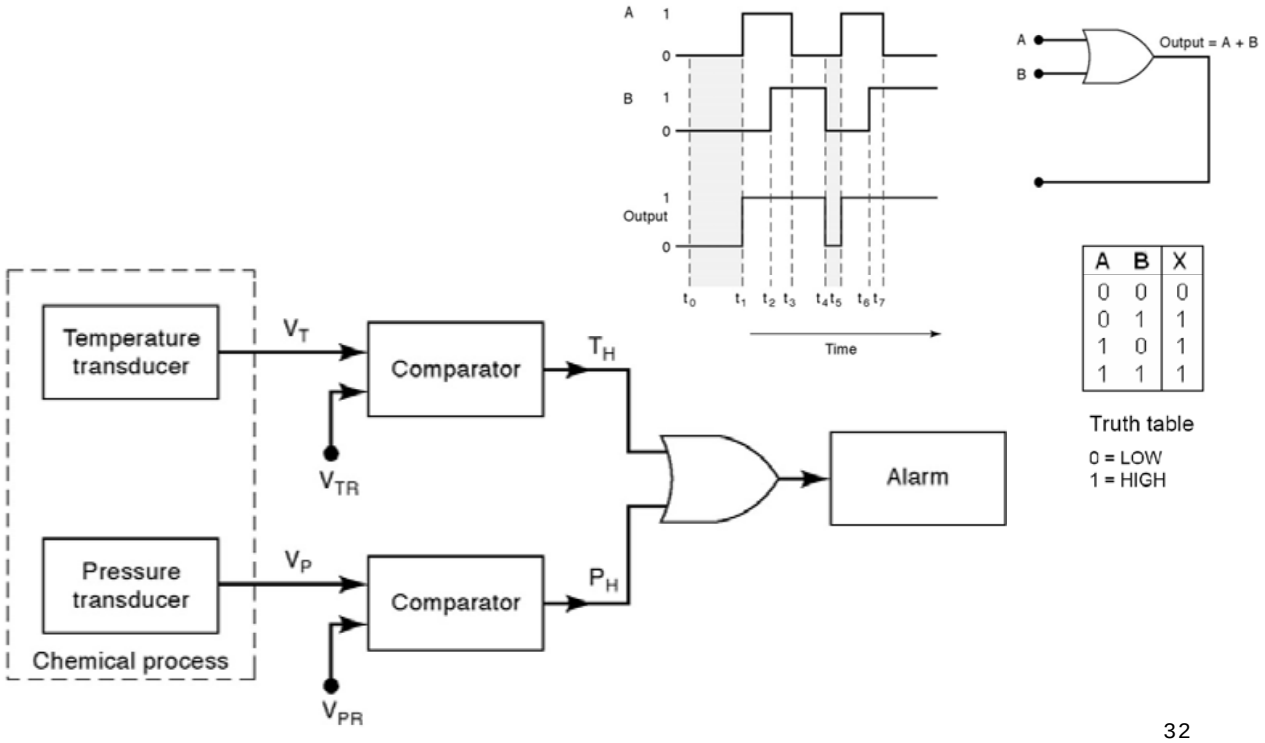


Enable/Disable Circuits

- AND gate

- NAND gate

Example using the OR gate



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

Truth table

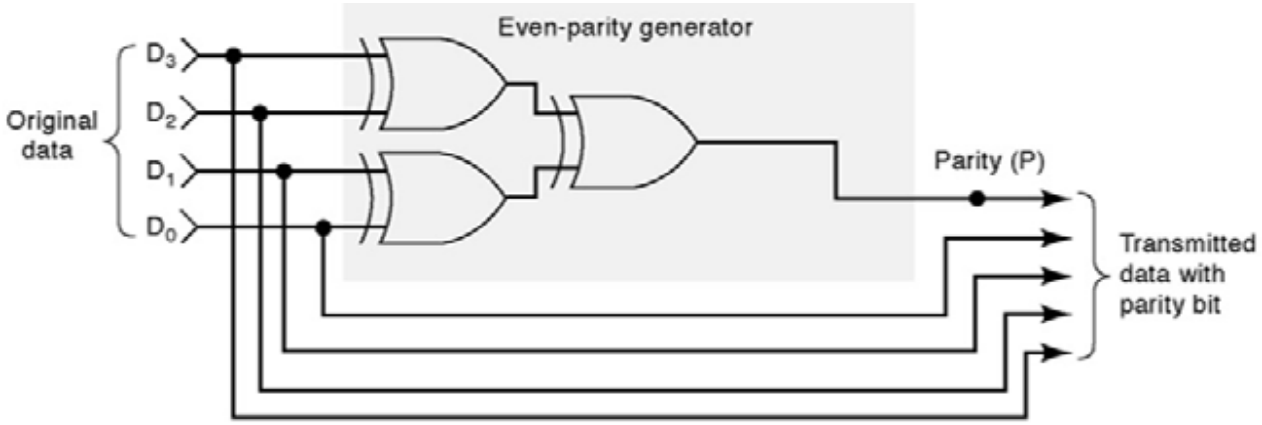
0 = LOW
1 = HIGH

Examples using the XOR gate

- Parity generator

XOR
(X ⊕ Y)

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

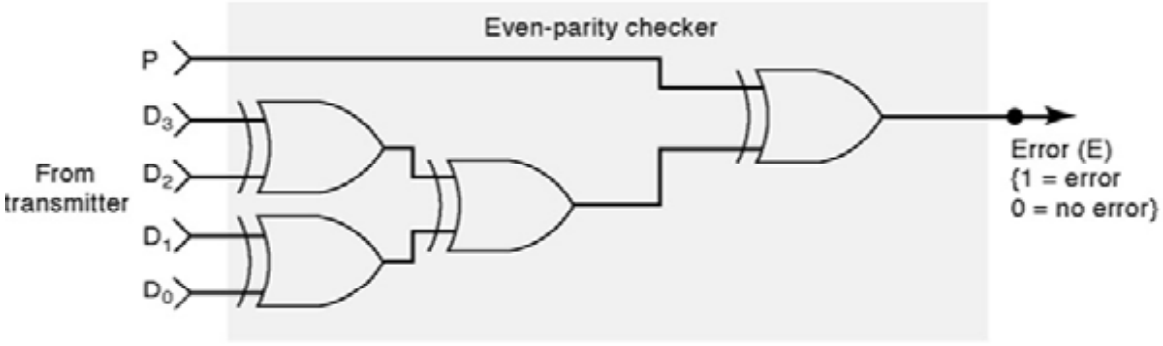


Examples using the XOR gate

- Parity checker

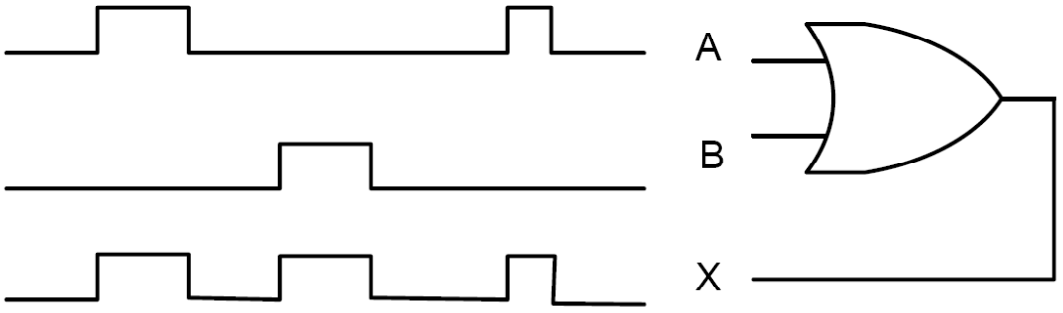
XOR
(X ⊕ Y)

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0



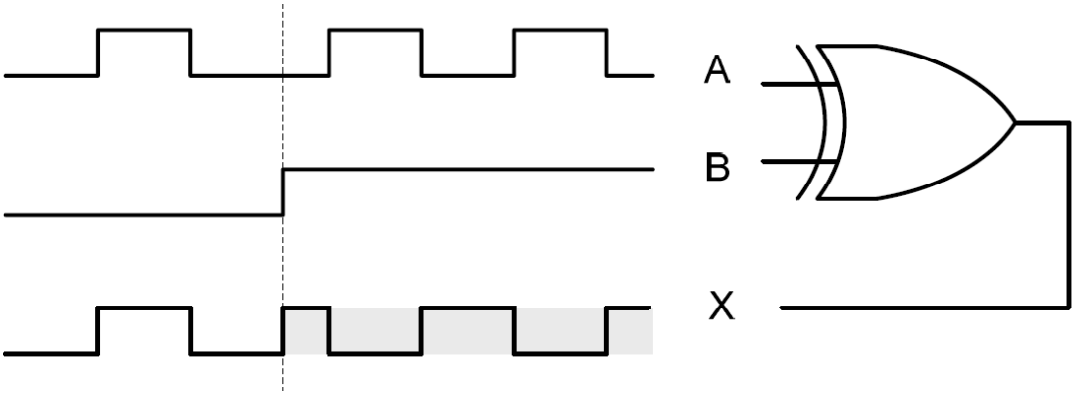
Merging & Inversion Circuits

- OR gate performs signal merging function



Merging & Inversion Circuits

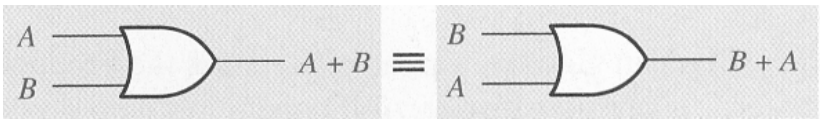
- XOR gate performs selectable inversion function



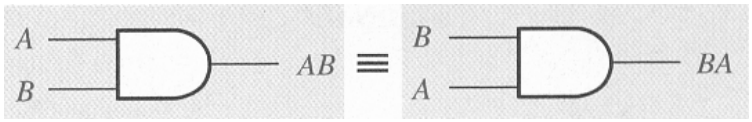
Boolean Algebra

- Commutative Laws

$$A + B = B + A$$



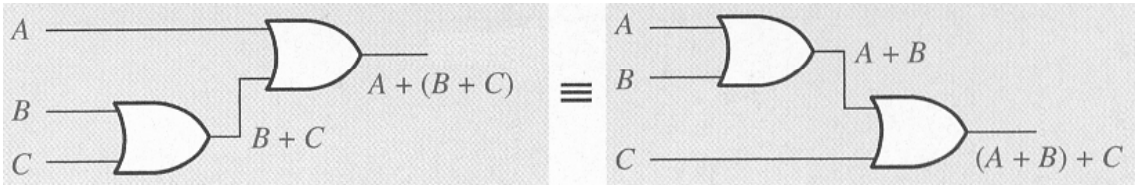
$$A \cdot B = B \cdot A$$



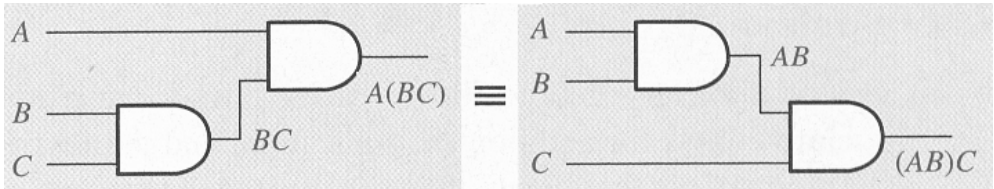
Boolean Algebra

- Associative Laws

$$A + (B + C) = (A + B) + C$$



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

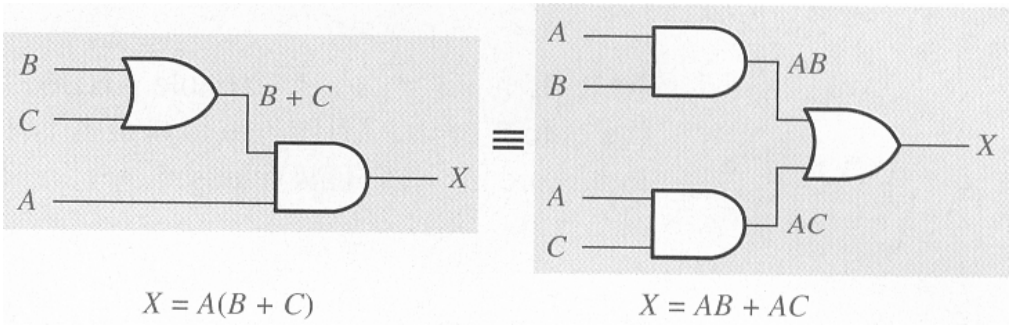


Boolean Algebra

- Distributive Laws

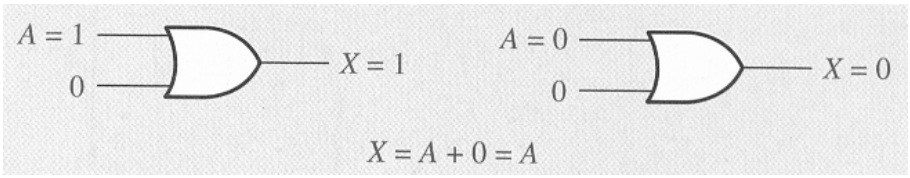
$$A \bullet (B + C) = A \bullet B + A \bullet C$$

$$A\,(B + C) = A\,B + A\,C$$



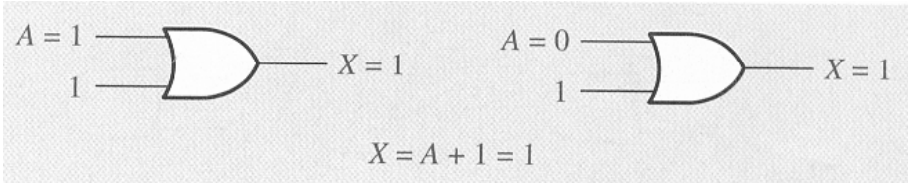
Rules of Boolean Algebra

- **A + 0 = A**



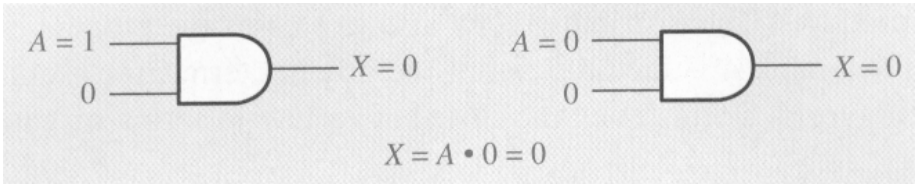
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

- **A + 1 = 1**



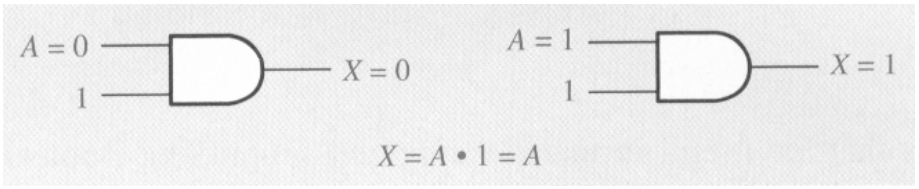
Rules of Boolean Algebra

- **A • 0 = 0**



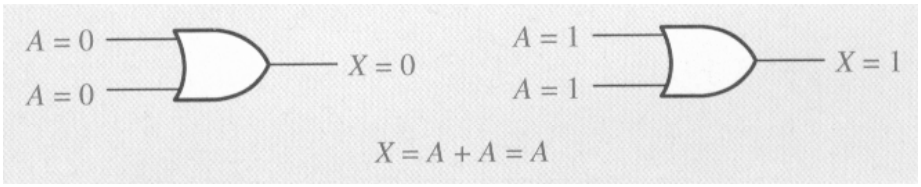
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

- **A • 1 = A**



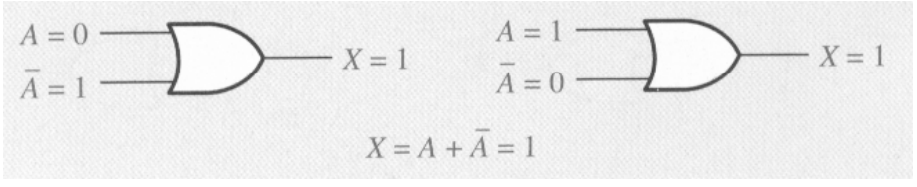
Rules of Boolean Algebra

- **A + A = A**



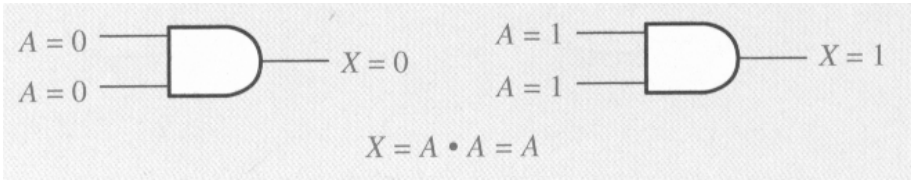
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

- **A + $\overline{A}$ = 1**



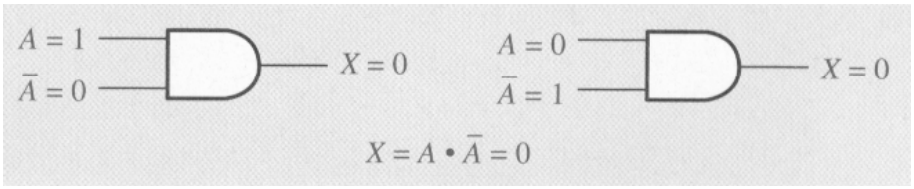
Rules of Boolean Algebra

- **A • A = A**



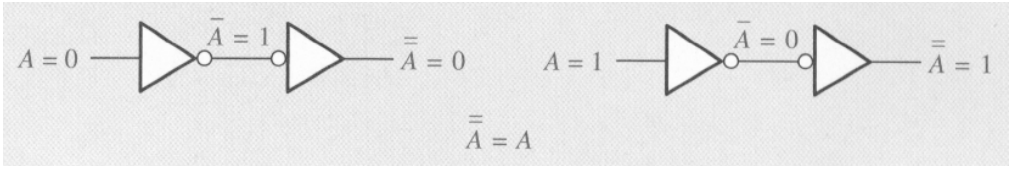
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

- **A • $\overline{A}$ = 0**



Rules of Boolean Algebra

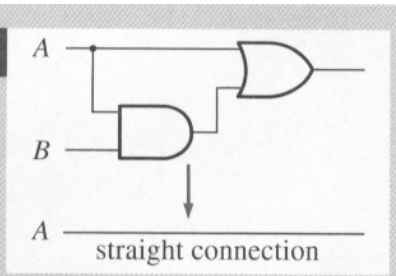
- **$\overline{\overline{A}}$ = A**



- **A + AB = A**

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

equal

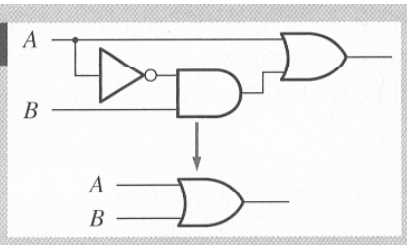


Rules of Boolean Algebra

- **$A + \overline{A}B = A + B$**

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

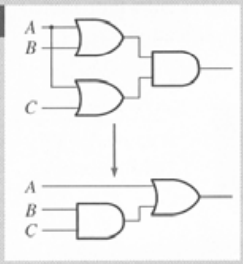
equal



- **$(A + B)(A + C) = A + BC$**

A	B	C	A + B	A + C	(A + B)(A + C)	BC	A + BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



Examples

- $y = A\overline{B}D + A\overline{B}\overline{D}$

  $y = A\overline{B}$

- $z = (\overline{A} + B)(A + B)$

  $z = B$

- $x = ACD + \overline{A}BCD$

  $x = ACD + BCD$

DeMorgan's Theorems

- Theorem 1
  - $\overline{(x + y)} = \overline{x} \cdot \overline{y}$
- Theorem 2
  - $\overline{(x \cdot y)} = \overline{x} + \overline{y}$

Remember:
"Break the bar,
change the operator"

- DeMorgan's theorem is very useful in digital circuit design
  - It allows **ANDs to be exchanged with ORs by using invertors**

DeMorgan's Theorems

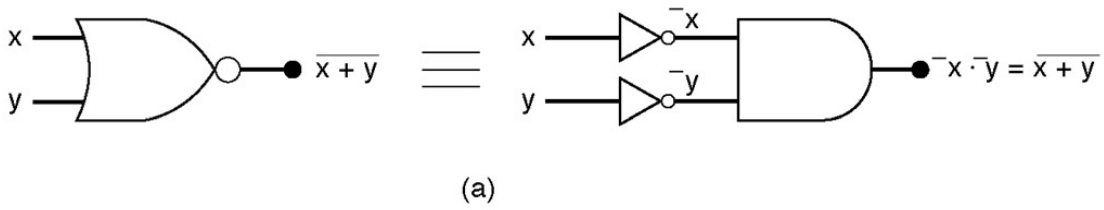
- Examples:

- $F = \overline{X \cdot Y} + \overline{P \cdot Q}$ — 2 NAND plus 1 OR
  $= \overline{X} + \overline{Y} + \overline{P} + \overline{Q}$ — 1 OR with some input invertors

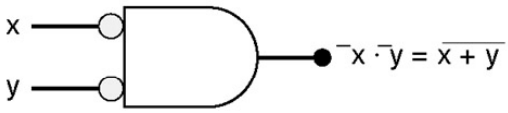
- $Z = \overline{(\overline{A}+C)\cdot(B+\overline{D})}$
  $= A\overline{C} + \overline{B}\,D$

DeMorgan's Theorems

- Examples:



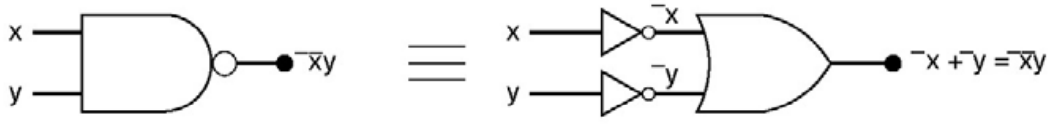
(a)



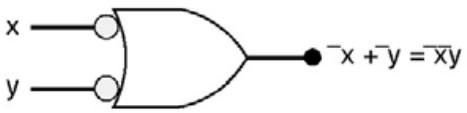
(b)

DeMorgan's Theorems

- Examples:



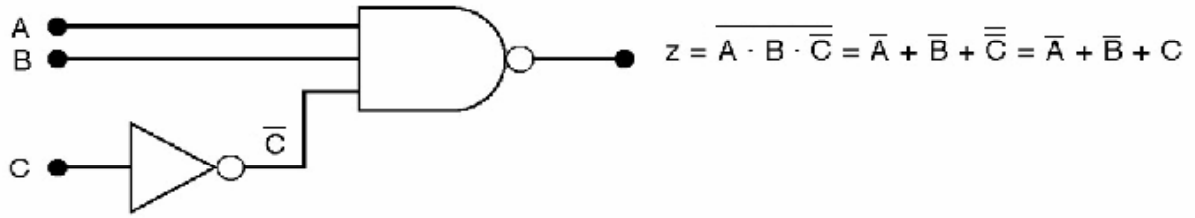
(a)



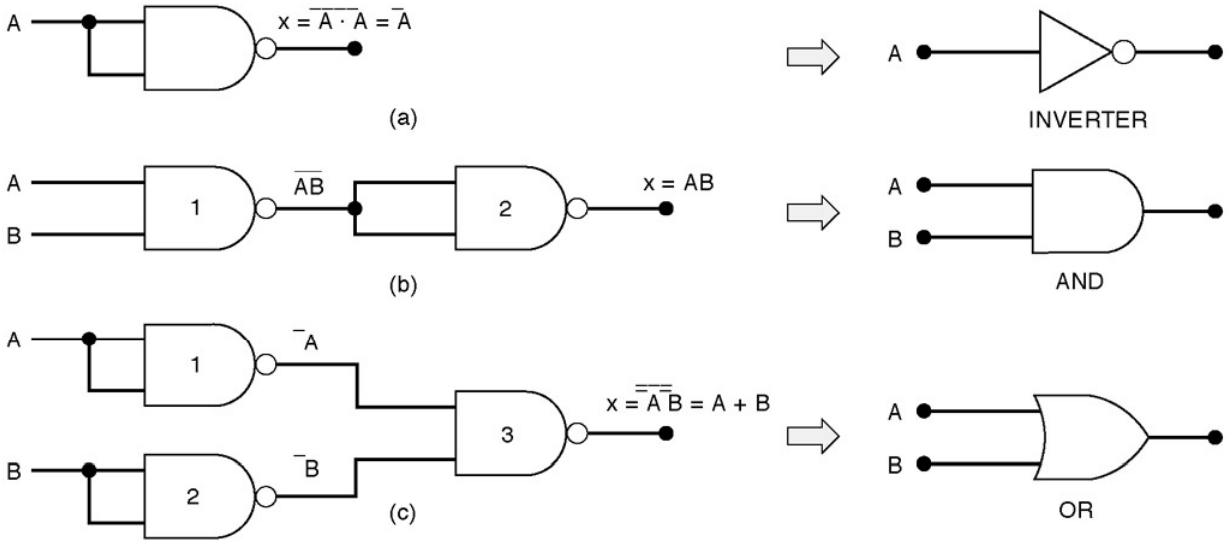
(b)

DeMorgan's Theorems

- Examples:



Universality of NAND gates



Universality of NOR gate