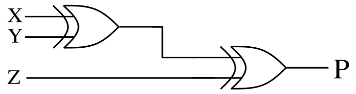
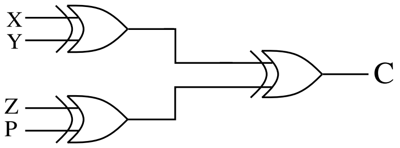
Parity Generators/Checkers

Three–Bit Message			Parity Bit
X	Y	Z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- We can use a parity tree to generate parity.
  - A 3-bit even parity generator.



- We can use a parity tree to check the code word for correct parity.
  - A 4-bit tree: $C = 0$ if the even parity is correct; $C = 1$ if an error occurs.



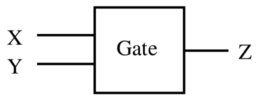
Logic Levels

- Two-valued Boolean logic are assigned to two voltage levels, **high** (H) and **low** (L).
- **Positive logic system:** 0 is assigned to L, and 1 to H.
- **Negative logic system:** 0 is assigned to H, and 1 to L.
- Modern IC designs predominately use positive logic.

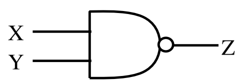
X	Y	Z
L	L	H
L	H	H
H	L	H
H	H	L

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

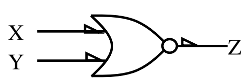
X	Y	Z
1	1	0
1	0	0
0	1	0
0	0	1



Gate behavior



Positive logic NAND



Negative logic NOR

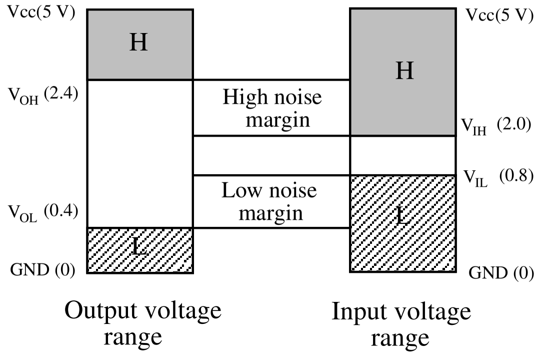
Noise Margin

- **Noise:** All undesirable voltage variations that are superimposed on normal operating voltage levels.

- **Noise Margin:** The maximum noise voltage level that a logic signal can tolerate without errors.

- High-level noise margin: $V_{OH} - V_{IH}$.

- Low-level noise margin: $V_{IL} - V_{OL}$.



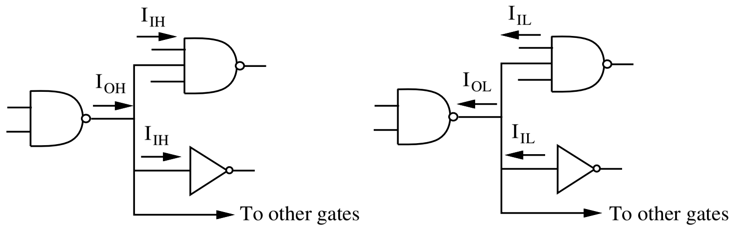
Fan-in/-out

- **Fan-in:** Number of inputs available on a gate.

- **Fan-out (standard load):** Number of inputs that each output can drive.

- Fan-out = $\min\{\frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}}\}$.

- Exp: $I_{OH} = 600\mu A, I_{IH} = 40\mu A, I_{OL} = 16\mu A, I_{IL} = 1.6\mu A \Longrightarrow \min\{\frac{600}{40}, \frac{16}{1.6}\} = 10$.



Gate as a current source

Gate as a current sink

Power Dissipation

- **Power supply (V_{CC}):** Voltage required to allow a circuit to operate.
- **Power dissipation:** The amount of power a circuit consumes.
- Each gate can be in 3 states: **high**, **transition**, **low** states.
- Respective currents: I_{CCH} (high); I_{CCT} (transition); I_{CCL} (low).
- In modern technologies (CMOS, etc): $I_{CCT} \gg I_{CCH}$ and $I_{CCT} \gg I_{CCL} \Longrightarrow$ Average power dissipation:

$$P_{avg} = V_{CC} \times I_{CCT}.$$

- In older logic families (TTL, etc): $I_{CCH} \gg I_{CCT}$ and $I_{CCL} \gg I_{CCT} \Longrightarrow$ Average power dissipation:

$$P_{avg} = V_{CC} \times (I_{CCH} + I_{CCL})/2.$$

- The power dissipation is an increasingly important issue in modern IC designs.
  - The power dissipation of a circuit defines the battery life.
  - It is proportional to the heat generated by a chip.

Propagation Delay

- **Rise time:** The delay for a signal to switch from 10% to 90% of its normal value.
- **Fall time:** The delay for a signal to switch from 90% to 10% of its normal value.
- **H-to-L propagation delay (t_{PHL}):** The time for the **output** signal to reach 50% of its normal value on the H-to-L transition after the input signal reached 50% of its normal signal. t_{PLH} is defined similarly.
- In modern technologies, the **propagation delay** t_{pd} is measured by

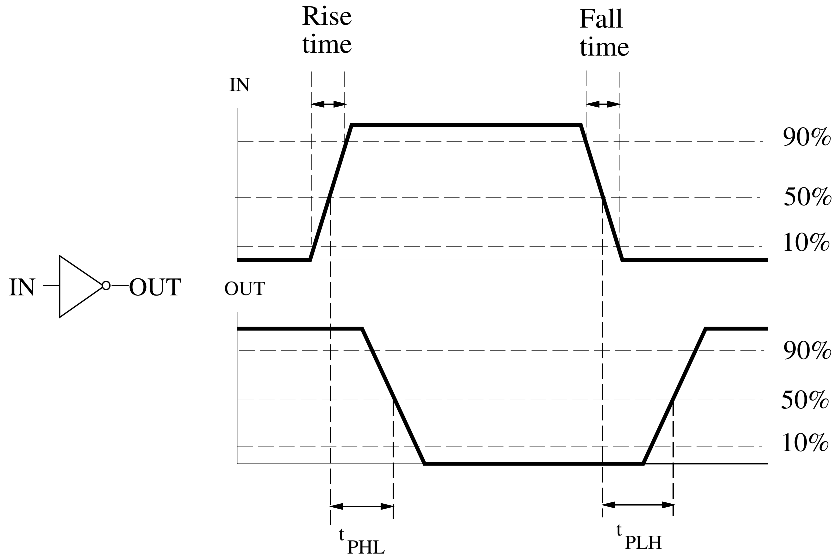
$$t_{pd} = \frac{(t_{PHL} + t_{PLH})}{2}.$$

- In older technologies, the **propagation delay** t_{pd} is measured by

$$t_{pd} = \max\{t_{PHL}, t_{PLH}\}.$$

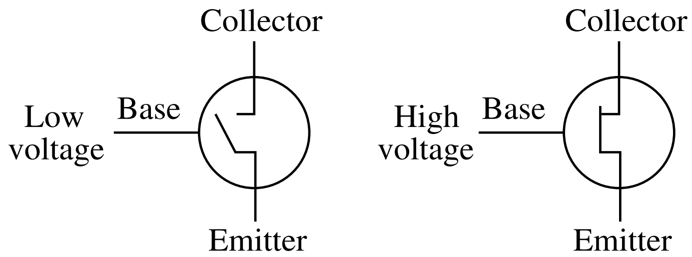
Propagation Delay for an Inverter

- High-to-low (t_{PHL}) and low-to-high (t_{PLH}) propagation delays are measured on **output transitions**.



Digital Logic Families: Bipolar Transistors

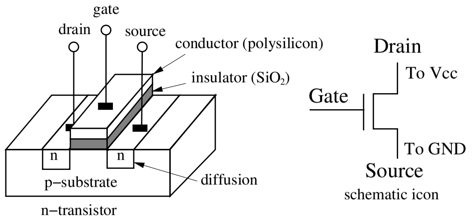
- Gates could be constructed using two transitor technologies: **bipolar** and **metal-oxide-semiconductor (MOS)**; based on physical characteristics, they are organized into **families**.
- **Bipolar technologies:** using resistors, diodes, transistors.
  - RTL (Resistor-Transistor Logic) and DTL (Diode-Transistor Logic): used in 1960s.
  - TTL (Transistor-Trnasistor Logic): dominated digital designs in 1970s and 1980s.
  - ECL (Emitter-Coupled Logic): mainly for high-speed applications, e.g., supercomputers.



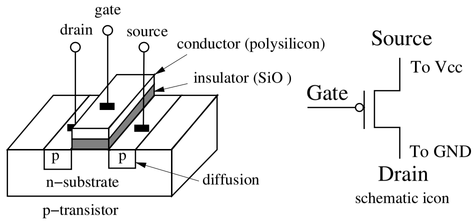
Bipolar transistor model

Digital Logic Families: MOS Transistors

- **MOS technologies:** a layer of mental above a layer of semi-conductor with a layer of silicon dioxide serving as an insulation between them.
  - MOS, CMOS (Complementary MOS), BiCMOS (Bipolar CMOS) dominate modern digital designs.
  - CMOS dominates nMOS and pMOS, due to CMOS's lower power dissipation, high fan-out, simple process, and high regularity.



The nMOS switch passes "0" well.



The pMOS switch passes "1" well.

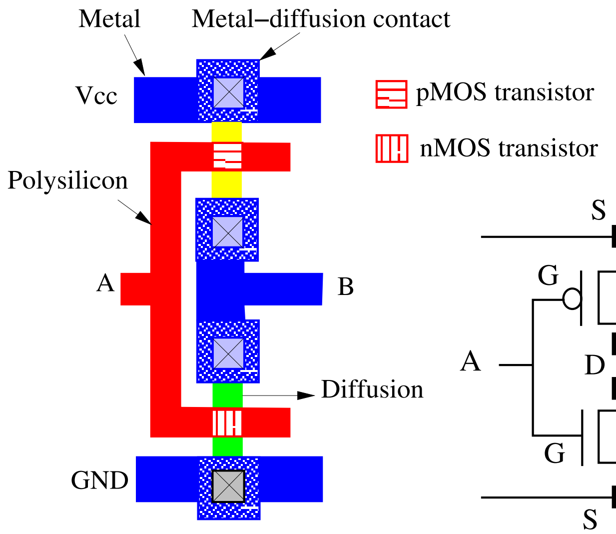
Characteristics of 2-input NAND

Logic family	# of resistors	# of diodes	# of transistors	Noise margin (V)	Fan out	Power dissipation (mW)	Delay (ns)
RTL	3	-	2	0.4	5	20	25
DTL	4	3	2	1.0	8	12	30
TTL	4	1	5	0.4	10	10	9
Schottky TTL	5	2	7	0.4	10	19	3
ECL	8	2	6	0.15	25	25	2
CMOS	-	-	4	0.5	30–50	0.1–0.3	1

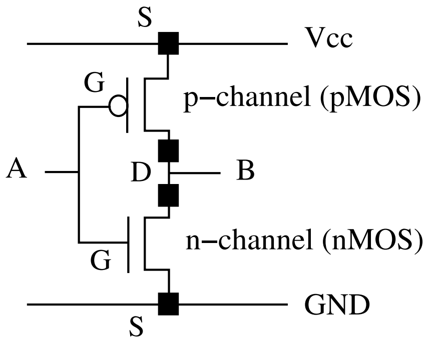
Level of Integration

- The number of gates put in a single chip differentiates integrated circuits (ICs).
- **Small-scale integration (SSI):** # of gates ≤ 10.
  - Simple logic functions, basic storage elements, etc.
- **Medium-scale integration (MSI):** 10–100 gates.
  - 4-bit adders, 8-to-1 multiplexers, counters, etc.
- **Large-scale integration (LSI):** 100 to a few 1000 gates.
  - Programmable logic array (PLA), PROM, small processors, etc.
- **Very large-scale integration (VLSI):** several thousand to millions of gates.
  - Processors, large memory array, etc.
- **Ultra large-scale integration (ULSI):** # of gates $>$ 10 million.
  - Advanced processors, system chips, etc.

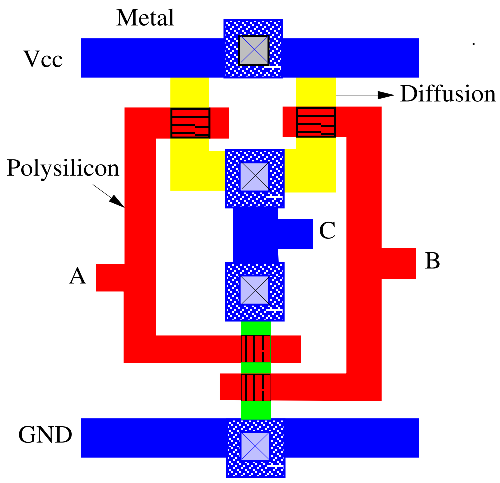
CMOS Inverter Gate



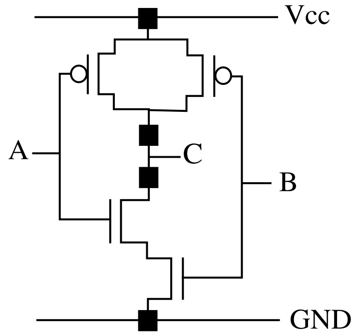
A	B
1	0
0	1



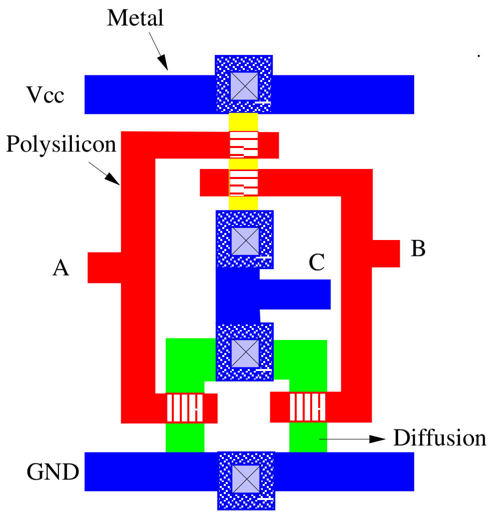
CMOS NAND Gate



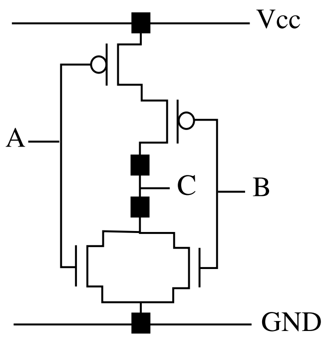
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0



CMOS NOR Gate

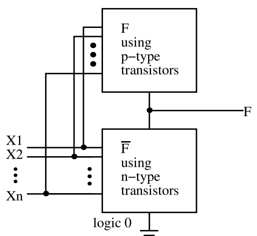


A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

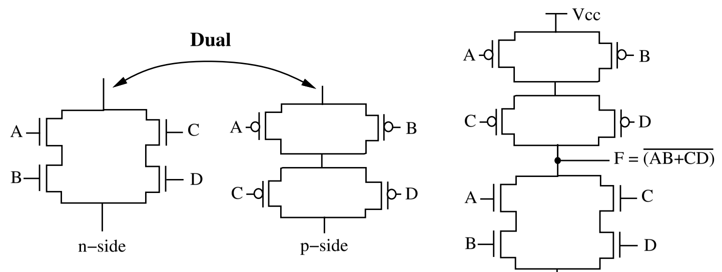


Construction of Compound Gates

- Example: $F = \overline{A \cdot B + C \cdot D}$.
- Step 1: Invert F to derive n-network ($\overline{F} = A \cdot B + C \cdot D$).
- Step 2: Make connections of transistors:
  - AND $\Longleftrightarrow$ Series connection; OR $\Longleftrightarrow$ Parallel connection.
- Step 3: Expand F to derive p-network ($F = \overline{AB + CD} = \overline{AB} \cdot \overline{CD} = (\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D})$); each input is inverted.
- Step 4: Make connections of transistors (same as Step 2).
- Step 5: Connect the n-network to GND (typically, 0V) and the p-network to V_{CC} (5V, 3.3V, or 2.5V).

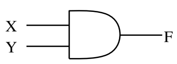
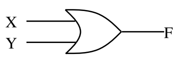
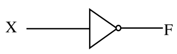
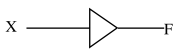
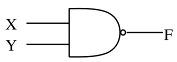
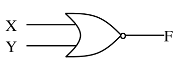
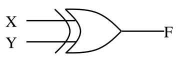
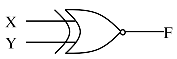


General Structure

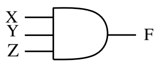
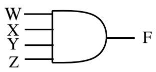
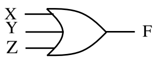
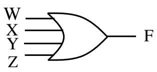
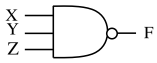
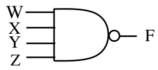
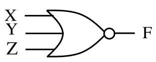
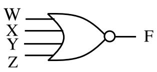


CMOS compound gate for $F = \overline{(AB + CD)}$

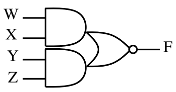
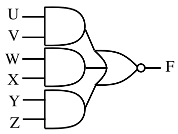
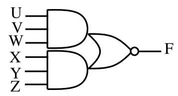
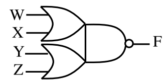
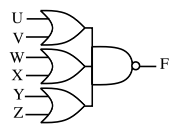
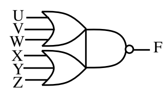
Basic CMOS Logic Library

Name	Distinctive shape	Algebraic equation	Cost (# of transistors)	Gate delay (ns)
AND		F=XY	**6**	**2.4**
OR		F=X+Y	**6**	**2.4**
NOT (Inverter)		$F=\overline{X}$	**2**	**1.0**
Buffer (Driver)		F=X	**4**	**2.0**
NAND		$F=\overline{XY}$	**4**	**1.4**
NOR		$F=\overline{X+Y}$	**4**	**1.4**
Exclusive–OR (XOR)		$F=X\overline{Y}+\overline{X}Y = X \oplus Y$	**14**	**4.2**
Exclusive–NOR (XNOR)		$F=XY+\overline{X}\,\overline{Y} = \overline{X \oplus Y}$	**12**	**3.2**

Multiple-input CMOS Logic Library

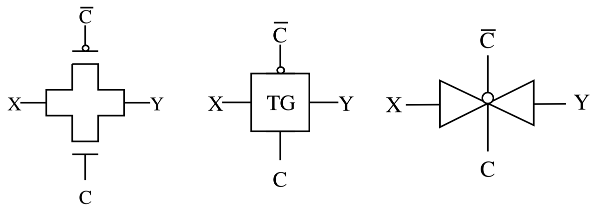
Name	Graphic symbol	Algebraic equation	Cost (# of transistors)	Gate delay (ns)
3–input AND		$F = XYZ$	**8**	**2.8**
4–input AND		$F = WXYZ$	**10**	**3.2**
3–input OR		$F = X + Y + Z$	**8**	**2.8**
4–input OR		$F = W + X + Y + Z$	**10**	**3.2**
3–input NAND		$F = \overline{XYZ}$	**8**	**1.8**
4–input NAND		$F = \overline{WXYZ}$	**10**	**2.2**
3–input NOR		$F = \overline{X + Y + Z}$	**8**	**1.8**
4–input NOR		$F = \overline{W + X + Y + Z}$	**10**	**2.2**

Multiple-operator CMOS Logic Library

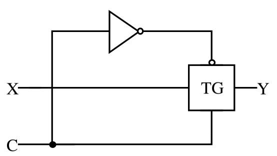
Name	Graphic symbol	Algebraic equation	Cost (# of transistors)	Gate delay (ns)
2–wide, 2–input AOI		$F = \overline{WX+YZ}$	**8**	**2.0**
3–wide, 2–input AOI		$F = \overline{UV + WX + YZ}$	**12**	**2.4**
2–wide, 3–input AOI		$F = \overline{UVW + XYZ}$	**12**	**2.2**
2–wide, 2–input OAI		$F = \overline{(W + X)(Y + Z)}$	**8**	**2.0**
3–wide, 2–input OAI		$F = \overline{(U + V)(W + X)\,(Y + Z)}$	**12**	**2.2**
2–wide, 3–input OAI		$F = \overline{(U + V + W)(X + Y + Z)}$	**12**	**2.4**

CMOS Transmission Gate (TG)

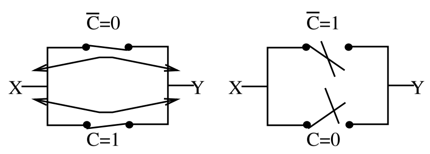
- A parallel n-channel and p-channel device driven by **opposite** control signals has two states:
  - **OFF (open)** when the drive signal is “0”
  - **ON (closed)** when the drive signal is “1”
- Current can flow either way through the TG.



Three types of notation.



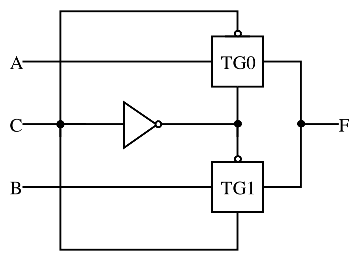
Control signals are mutually complemented.



C = 1: X and Y are connected;
C = 0: X and Y are disconnected.

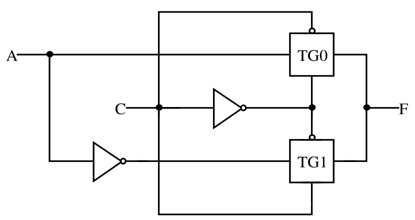
CMOS Logic with TGs

- Transmission gate-based multiplexer: ($C = 1 \Longrightarrow$ select B; $C = 0 \Longrightarrow$ select A)



F(A, B, C) = $A\overline{C} + BC$

- Transmission gate-based XOR logic:



A	C	TG1	TG0	F
0	0	No path	Path	0
0	1	Path	No path	1
1	0	No path	Path	1
1	1	Path	No path	0

F(A, C) = $A \oplus C$