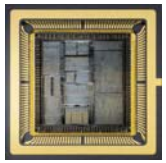
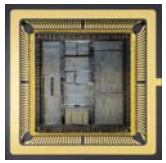


# Lecture 3: CMOS Logic

## CSCI 5330
## Digital CMOS VLSI Design

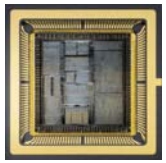
**Instructor**: Saraju P. Mohanty, Ph. D.

**NOTE**: The figures, text etc included in slides are borrowed from various books, websites, authors pages, and other sources for academic purpose only. The instructor does not claim any originality.



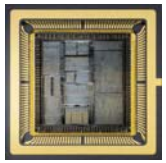
# Lecture Outline

- MOS Transistors
- CMOS Logic



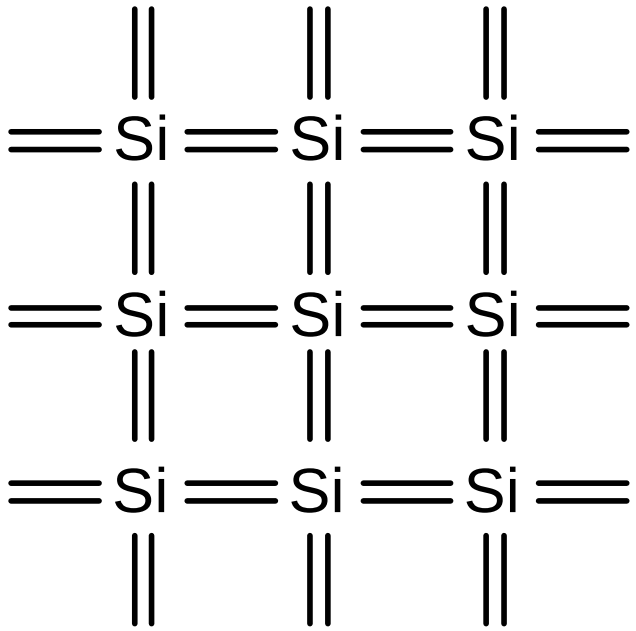
# Introduction

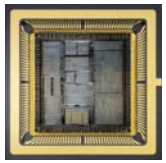
- Integrated circuits: many transistors on one chip.
- *Very Large Scale Integration* (VLSI): very many
- *Complementary Metal Oxide Semiconductor*
  - Fast, cheap, low power transistors
- How to build your own simple CMOS chip
  - CMOS transistors
  - Building logic gates from transistors
  - Transistor layout and fabrication



# Silicon Lattice

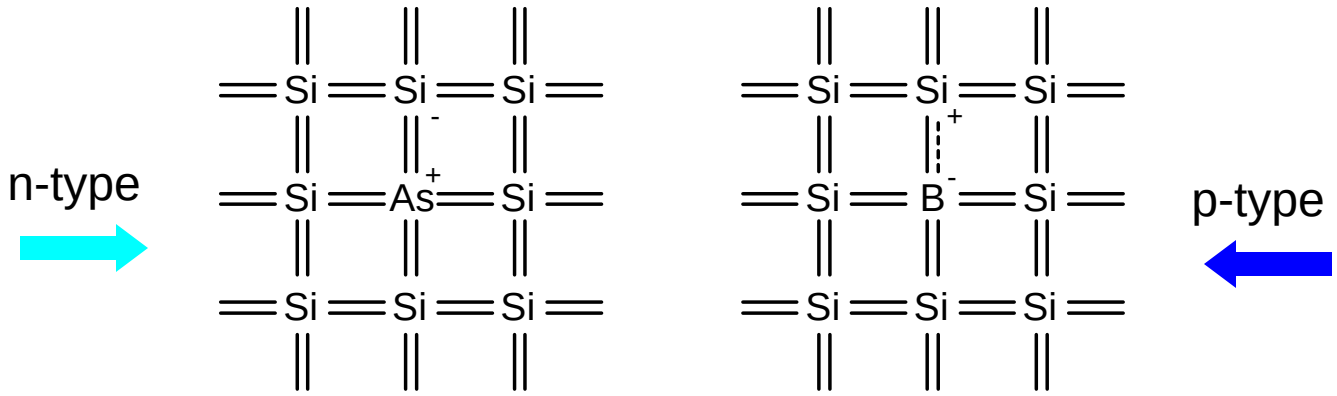
- Transistors are built on a silicon substrate
- Silicon is a Group IV material
- Forms crystal lattice with bonds to four neighbors

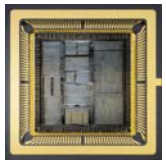




# Dopants

- Silicon is a semiconductor
- Pure silicon has no free carriers and conducts poorly
- Adding dopants increases the conductivity
- Group V: extra electron (n-type)
- Group III: missing electron, called hole (p-type)



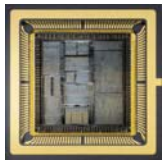


# Why Silicon?

Could any other group IV metal serve purpose ?

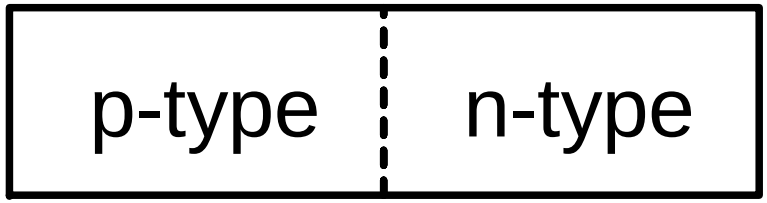
Key Advantages:

- Available abundantly in nature.
- Has excellent physical and electrical properties.
- Matured chemistry for fabrication

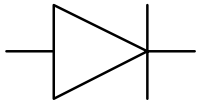


# p-n Junctions

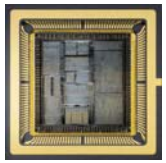
- A junction between p-type and n-type semiconductor forms a diode.
- Current flows only in one direction




anode cathode

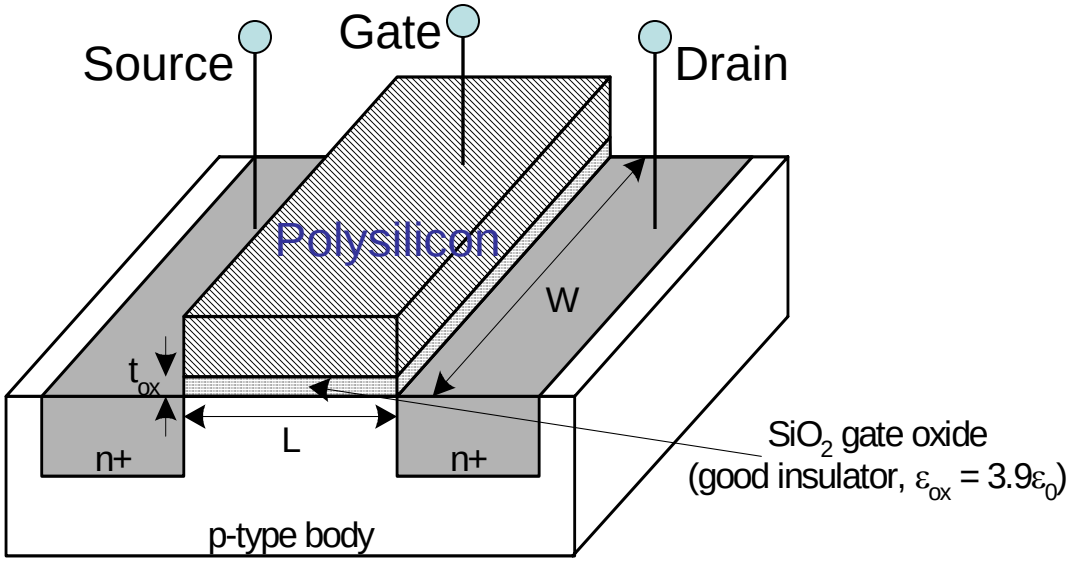


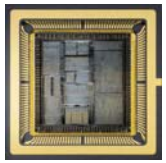
symbol



# MOS Transistor

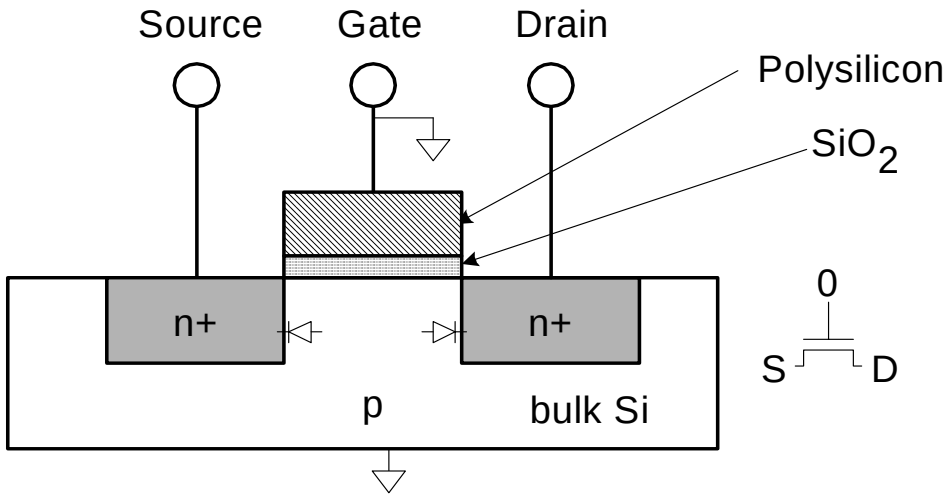
- Four terminals: gate, source, drain, body (bulk, or substrate)
- Gate – oxide – body stack looks like a capacitor
  - Gate and body are conductors
  - $SiO_2$ (oxide) is a very good insulator
  - Called metal – oxide – semiconductor (MOS) capacitor
  - Even though gate is no longer made of metal

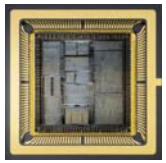




# nMOS Operation

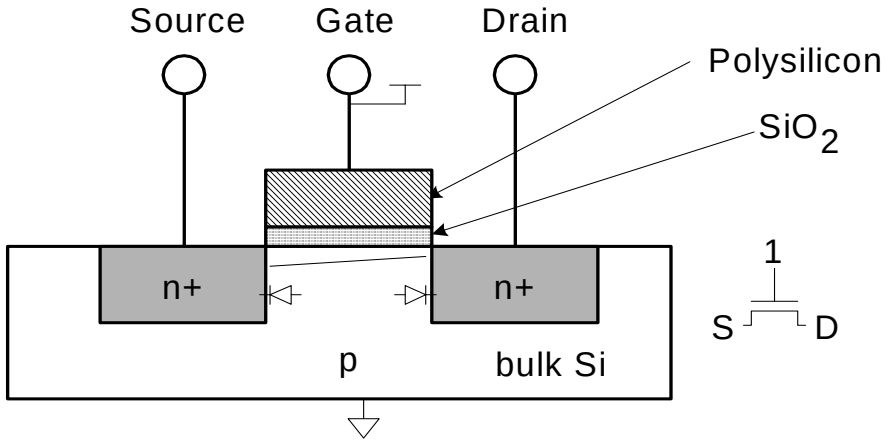
- Body is commonly tied to ground (0 V)
- When the gate is at a low voltage:
  - P-type body is at low voltage
  - Source-body and drain-body diodes are OFF
  - No current flows, transistor is OFF

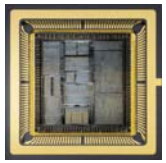




# nMOS Operation …..

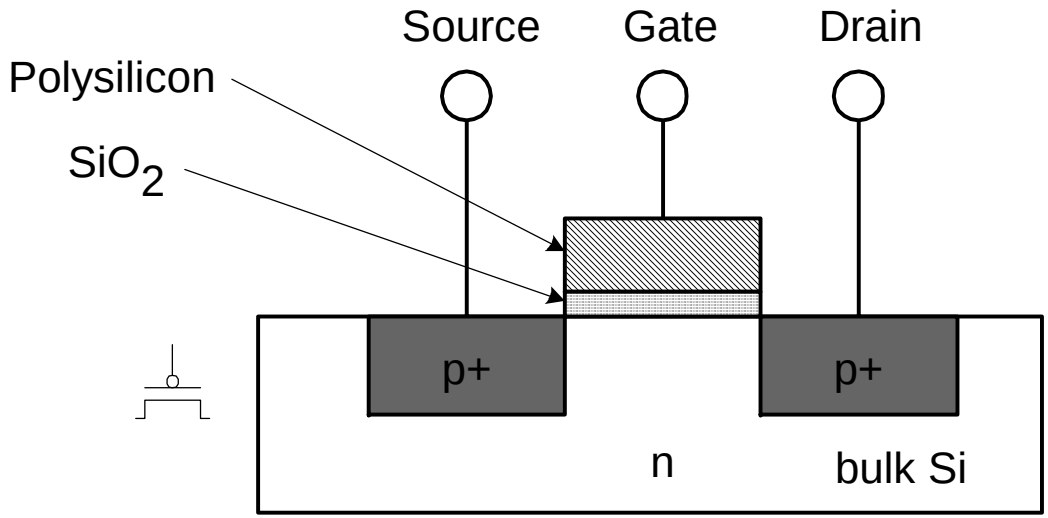
- When the gate is at a high voltage:
  - Positive charge on gate of MOS capacitor
  - Negative charge attracted to body
  - Inverts a channel under gate to n-type
  - Now current can flow through n-type silicon from source through channel to drain, transistor is ON

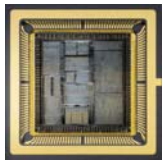




# pMOS Transistor

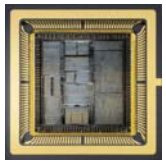
- Similar, but doping and voltages reversed
  - Body tied to high voltage ($V_{DD}$)
  - Gate low: transistor ON
  - Gate high: transistor OFF
  - Bubble indicates inverted behavior





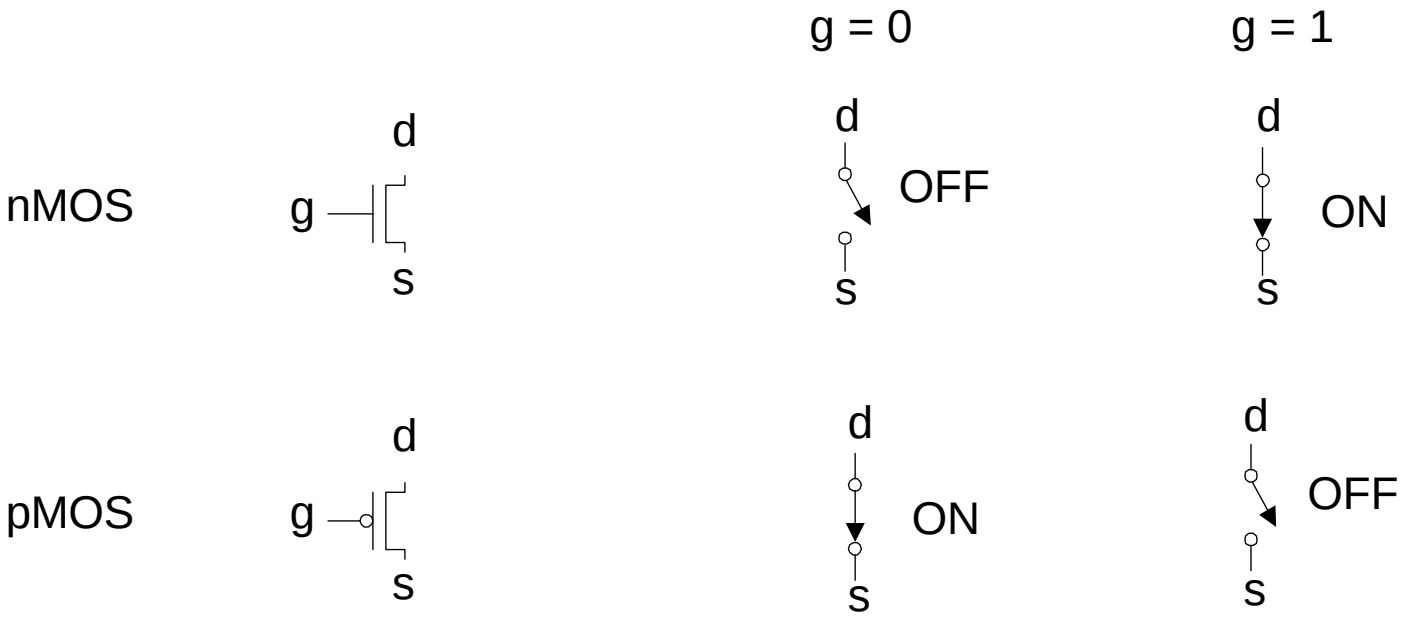
# Power Supply Voltage

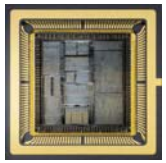
- GND = 0 V
- In 1980's, $V_{DD}$ = 5V
- $V_{DD}$ has decreased in modern processes
  - High $V_{DD}$ would damage modern tiny transistors
  - Lower $V_{DD}$ saves power
- $V_{DD}$ = 3.3, 2.5, 1.8, 1.5, 1.2, 1.0, …
- For 65nm or 45nm it is approximately 0.7V.



# Transistors as Switches: Ideal

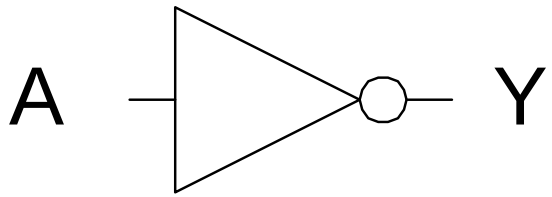
- We can view MOS transistors as electrically controlled switches
- Voltage at gate controls path from source to drain

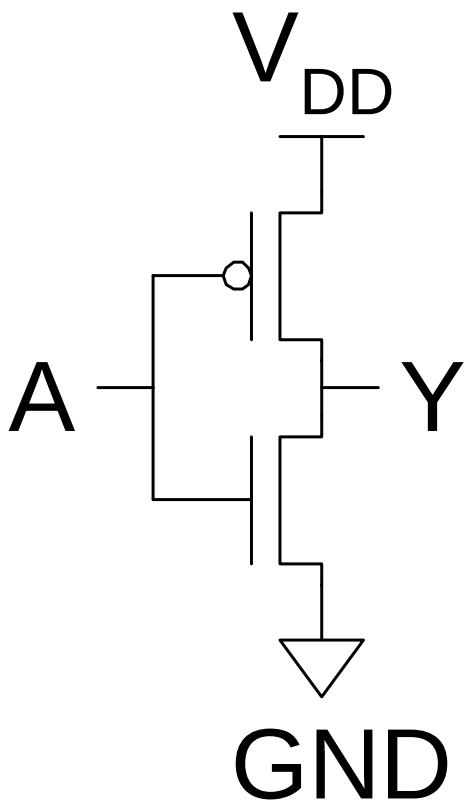


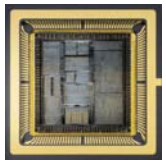


# CMOS Inverter

| A | Y |
|---|---|
| 0 | |
| 1 | |

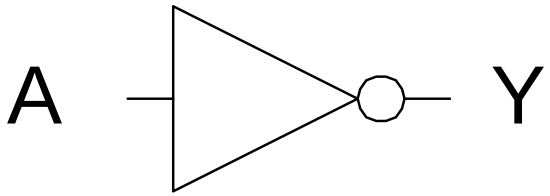


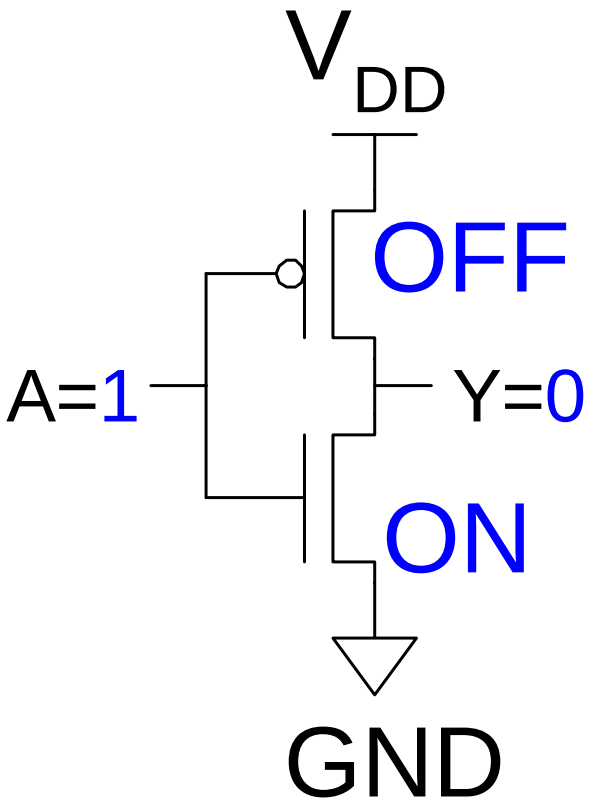



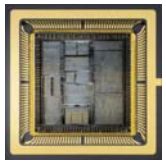


# CMOS Inverter

| A | Y |
|---|---|
| 0 | |
| **1** | **0** |

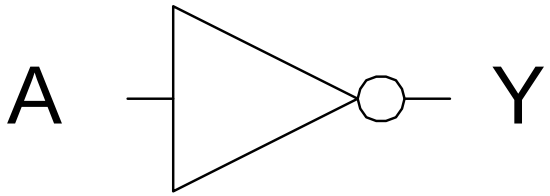


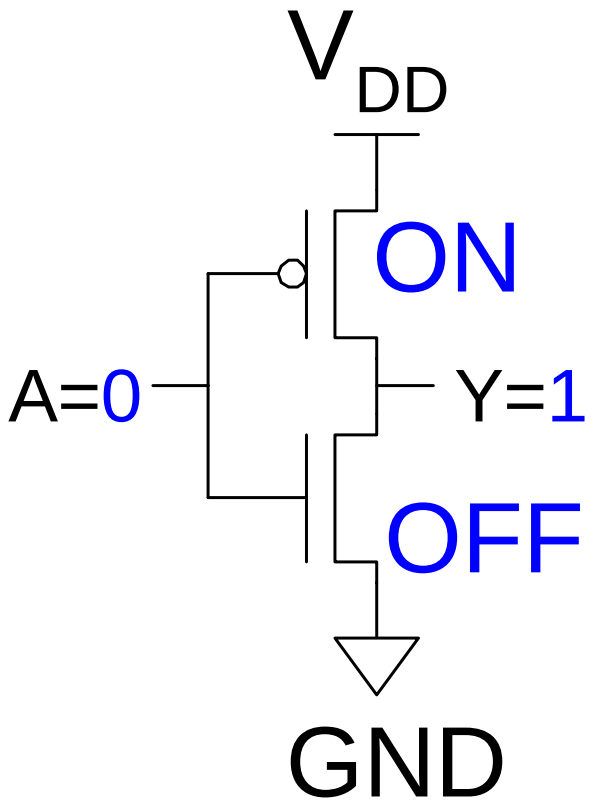





# CMOS Inverter

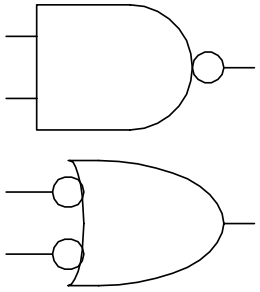
| A | Y |
|---|---|
| **0** | **1** |
| 1 | 0 |

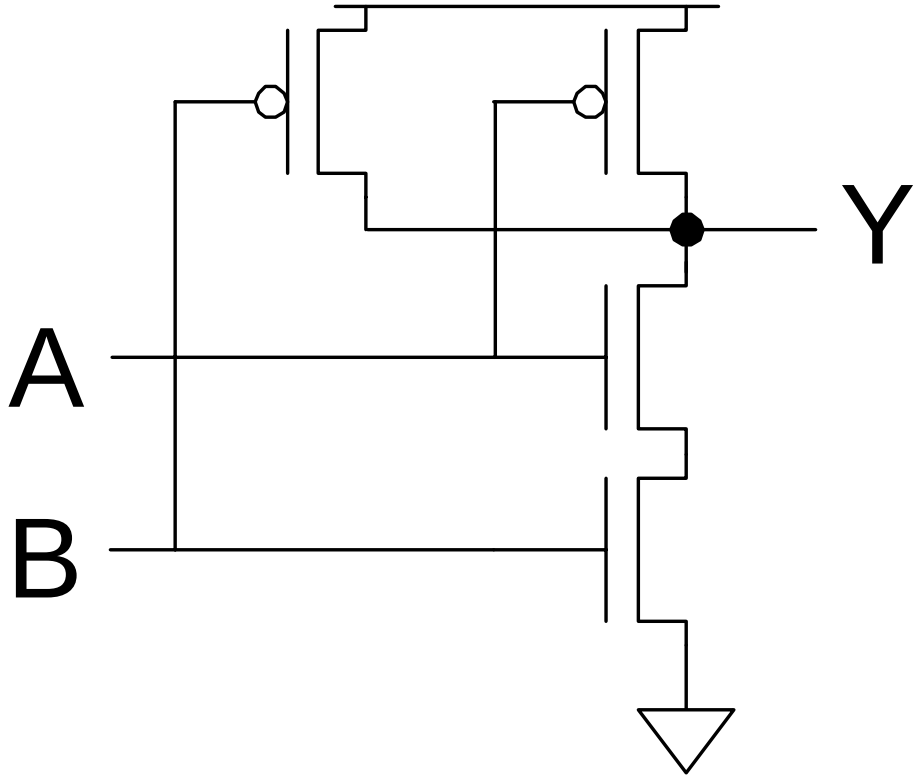


# CMOS NAND Gate

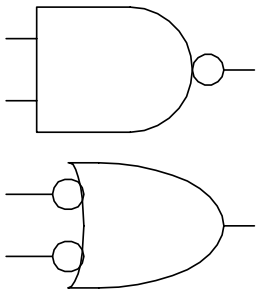
| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

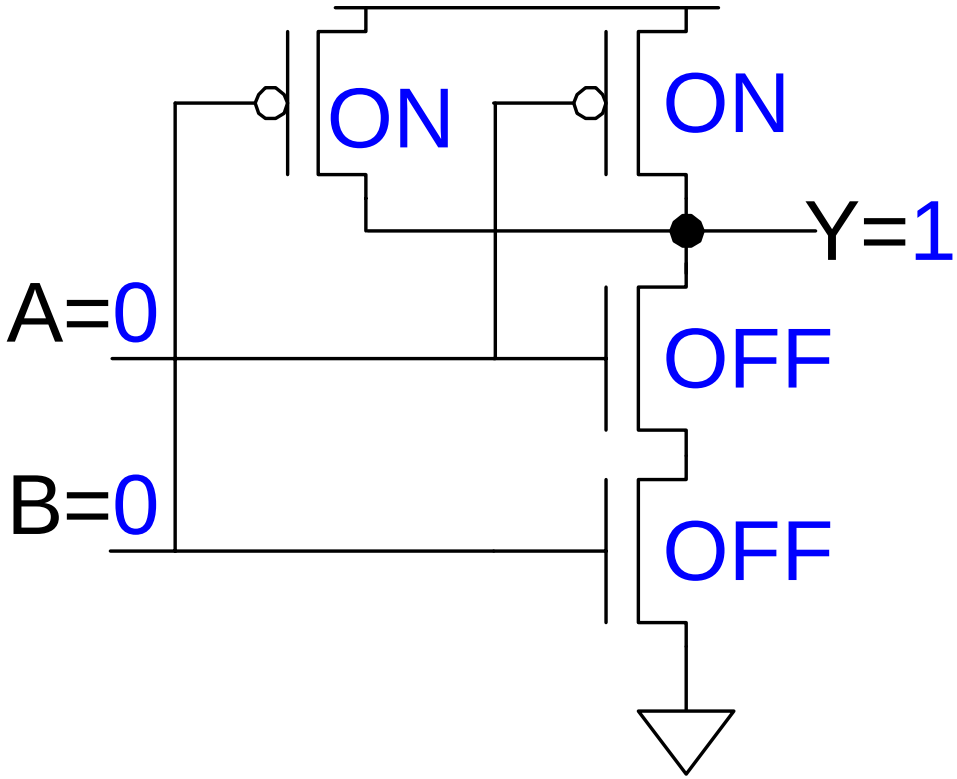




# CMOS NAND Gate

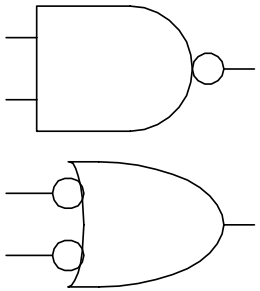
| A | B | Y |
|---|---|---|
| **0** | **0** | **1** |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

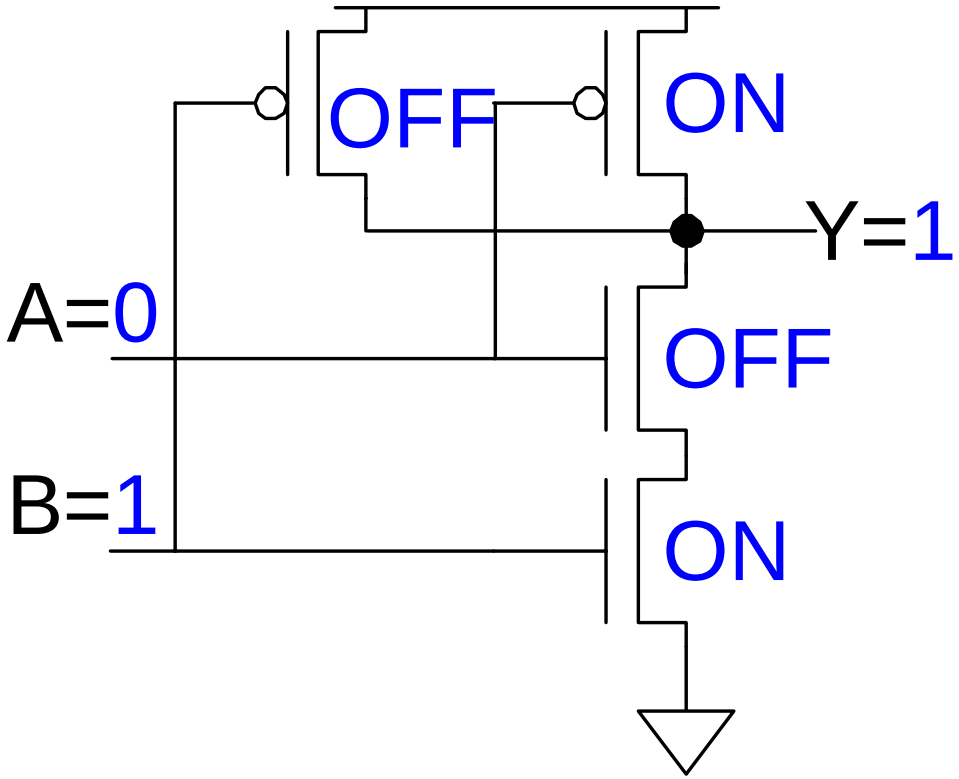




# CMOS NAND Gate

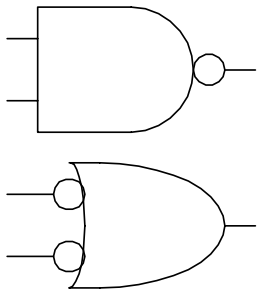
| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| **0** | **1** | **1** |
| 1 | 0 | |
| 1 | 1 | |

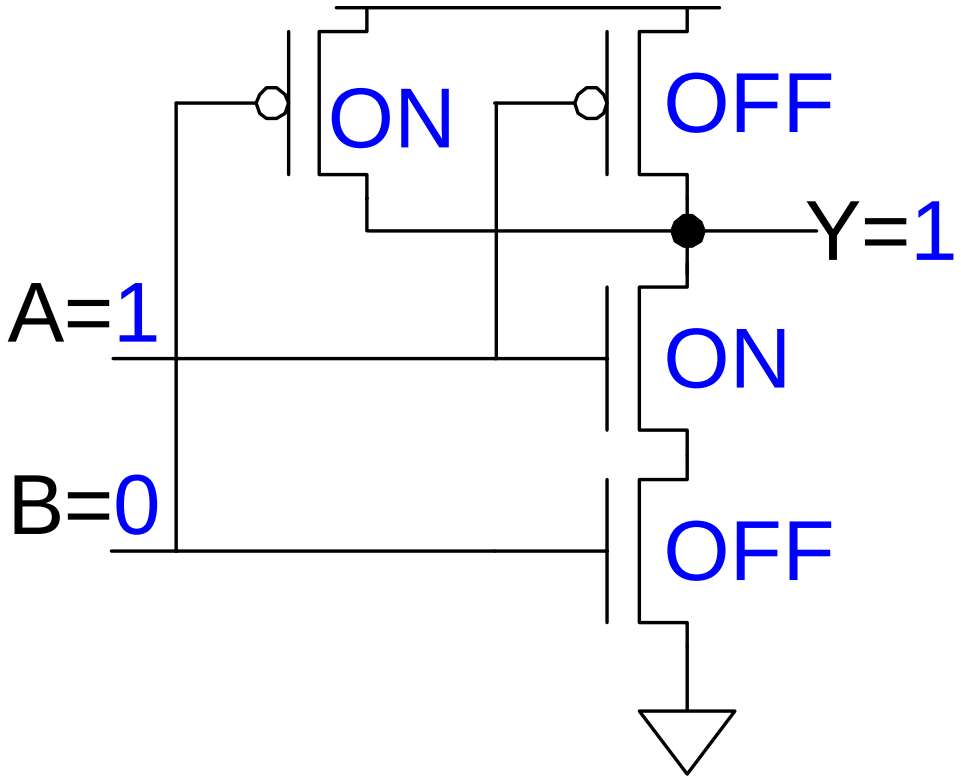




# CMOS NAND Gate

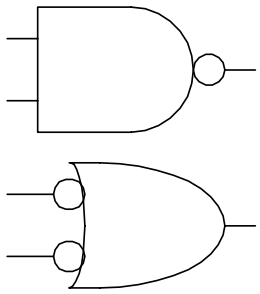
| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| **1** | **0** | **1** |
| 1 | 1 | |

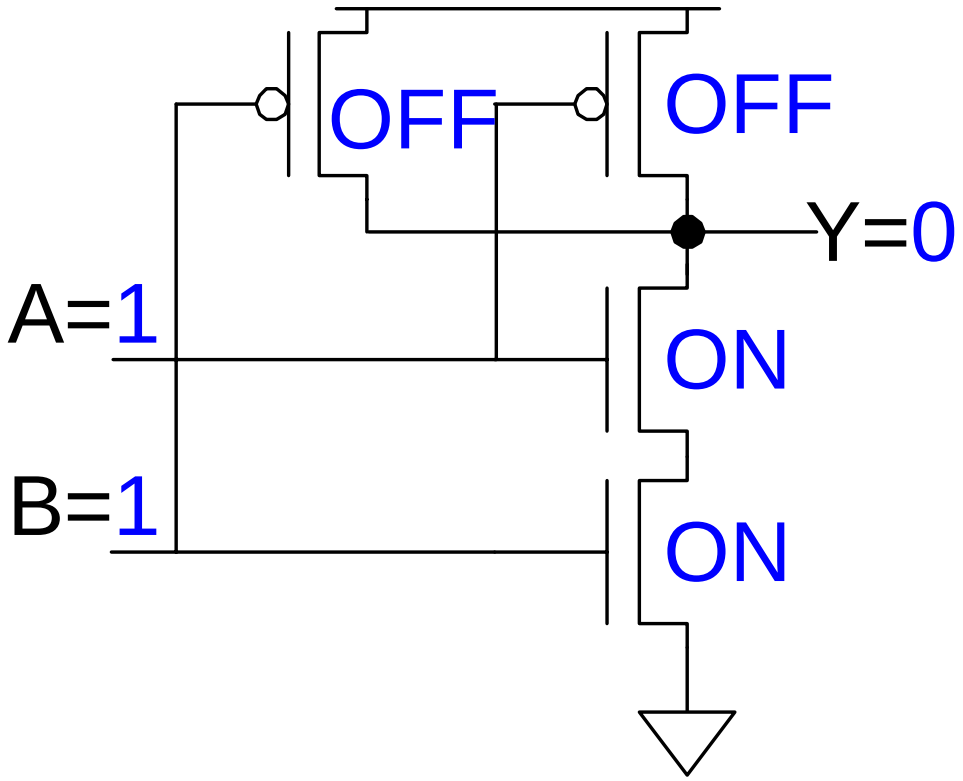




# CMOS NAND Gate

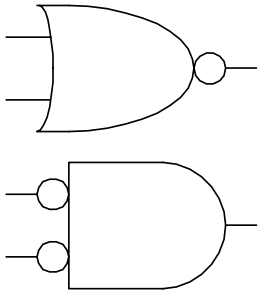
| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| **1** | **1** | **0** |

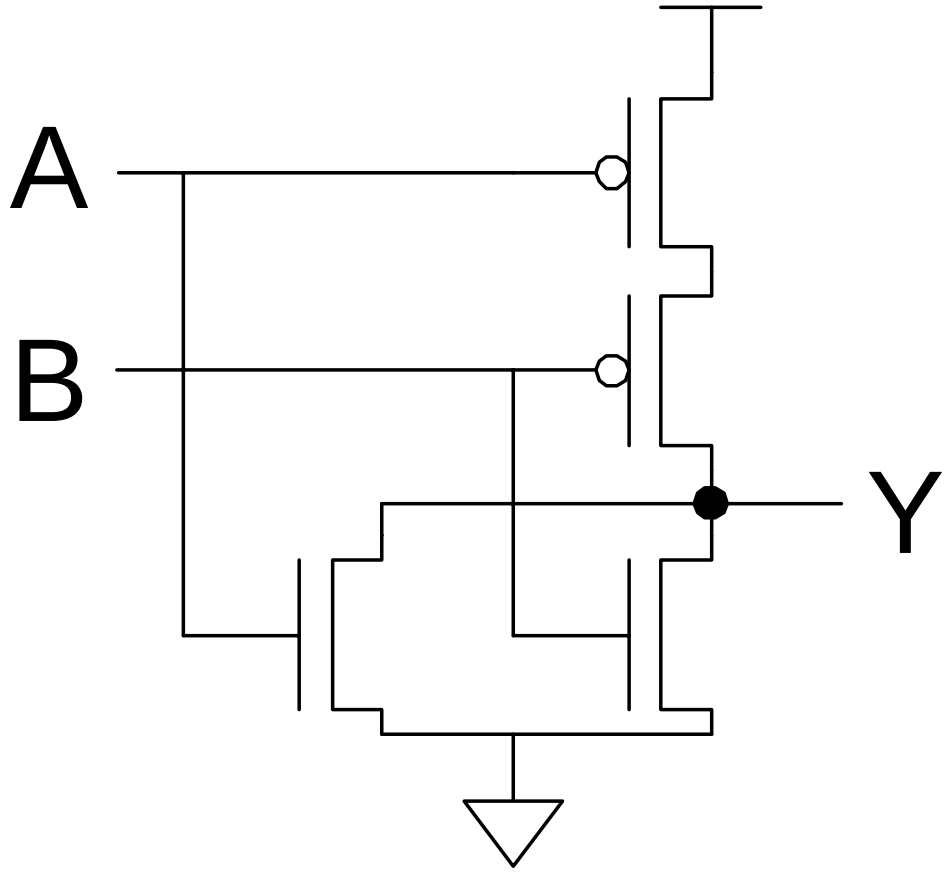


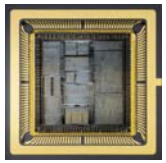


# CMOS NOR Gate (Dual of NAND)

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

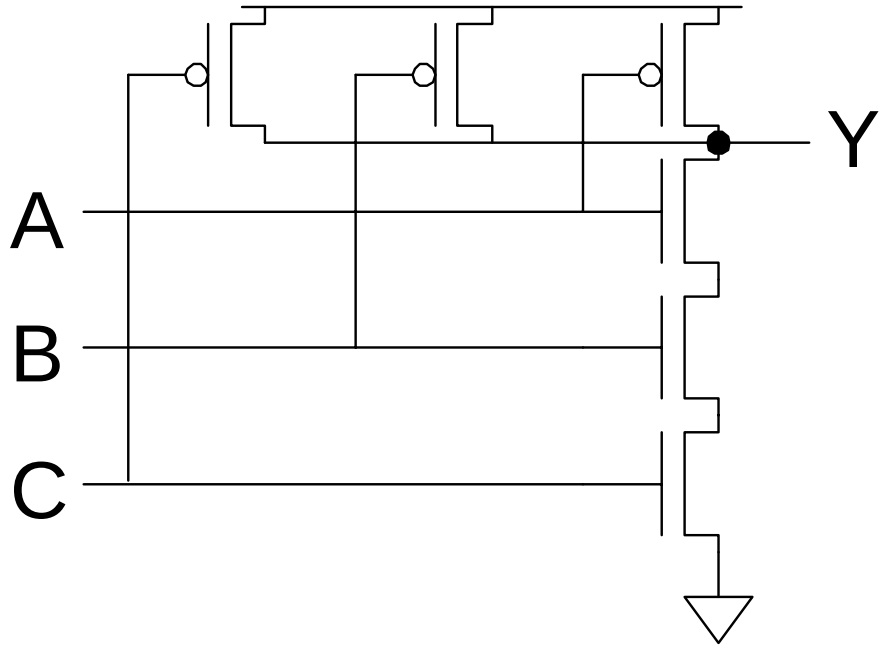


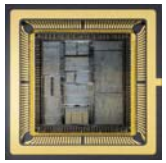




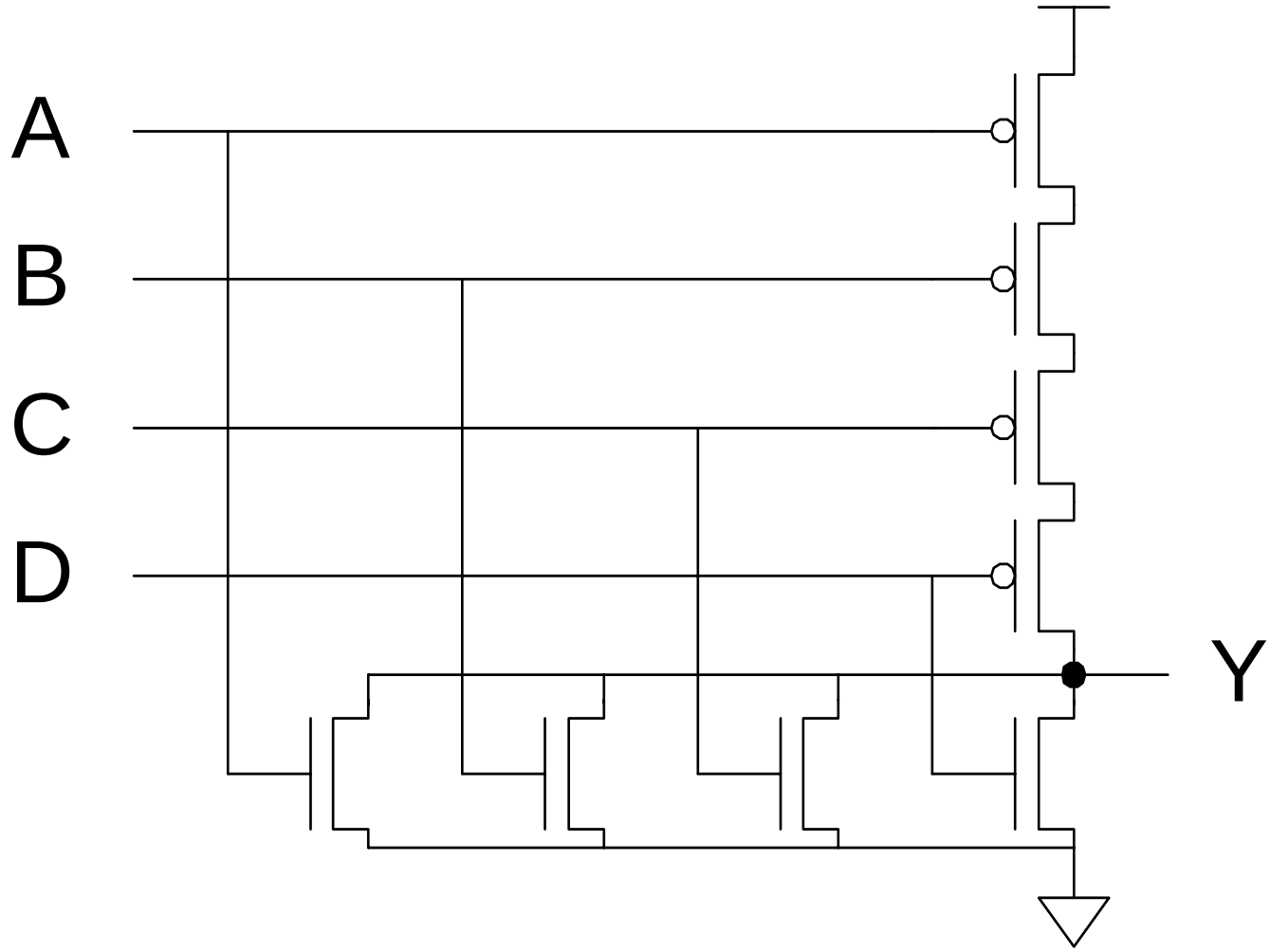
# 3-input NAND Gate

- Y pulls low if ALL inputs are 1
- Y pulls high if ANY input is 0



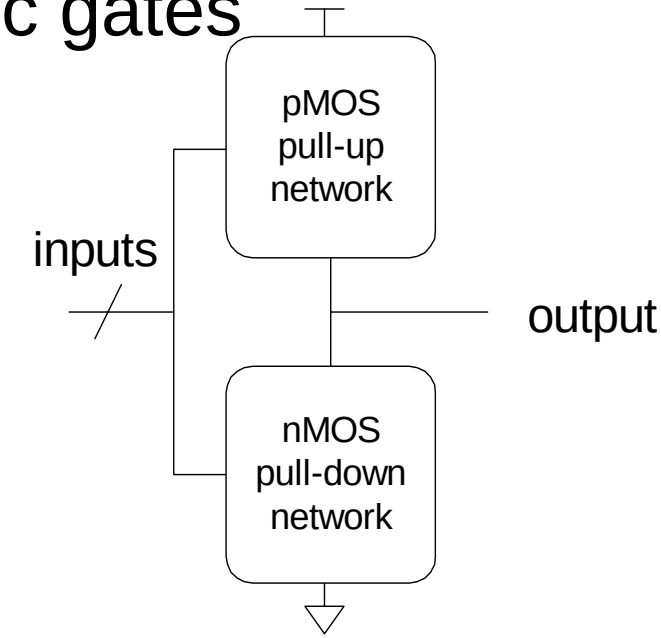


# 4-input NOR Gate



# Complementary CMOS : General

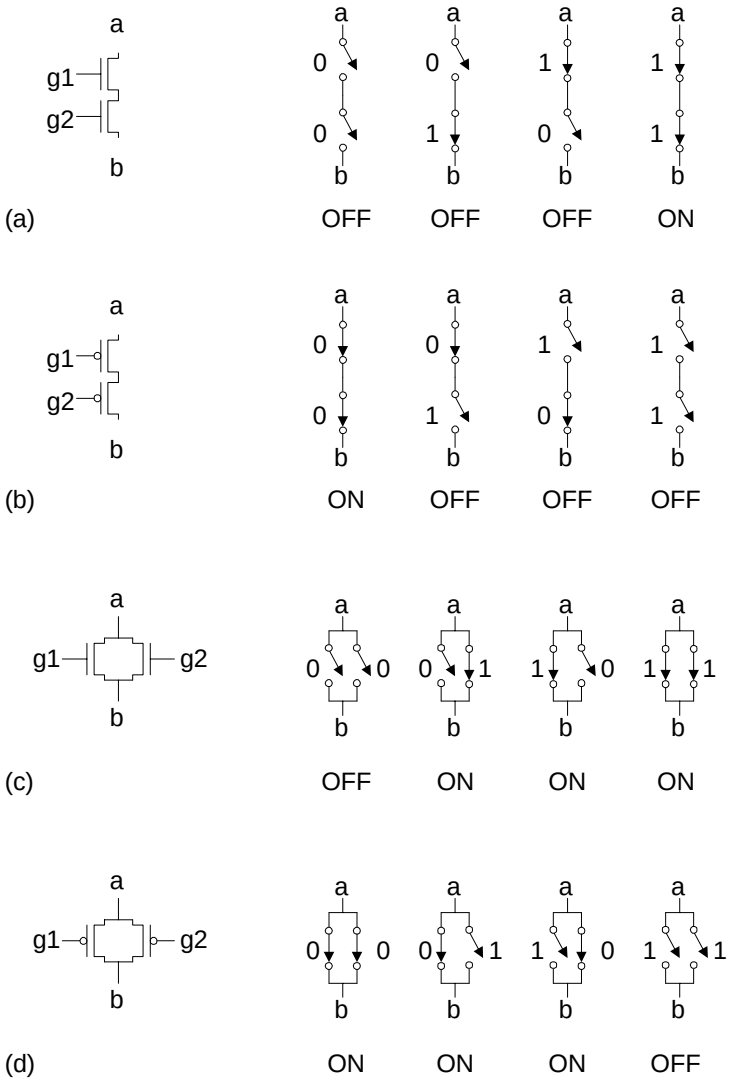
- Complementary CMOS logic gates
  - nMOS *pull-down network*
  - pMOS *pull-up network*
  - a.k.a. static CMOS

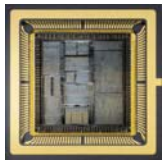



| | Pull-up OFF | Pull-up ON |
|---|---|---|
| Pull-down OFF | Z (float) | 1 |
| Pull-down ON | 0 | X (crowbar) |

# Connection and Behavior of Transistors

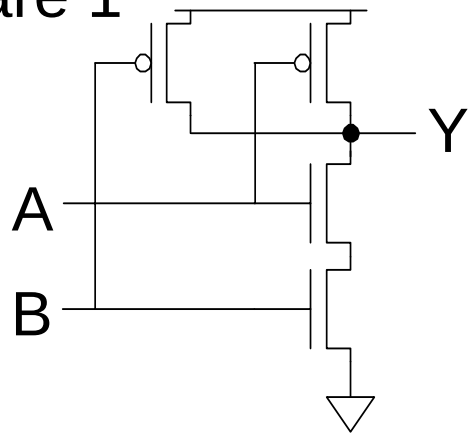
- nMOS: 1 = ON
- pMOS: 0 = ON
- *Series*: both must be ON
- *Parallel*: either can be ON



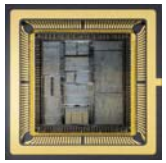


# Conduction Complement

- Complementary CMOS gates always produce 0 or 1
- Ex: NAND gate
  - Series nMOS: Y=0 when both inputs are 1
  - Thus Y=1 when either input is 0
  - Requires parallel pMOS

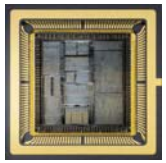



- Rule of *Conduction Complements*
  - Pull-up network is complement of pull-down
  - Parallel -> series, series -> parallel



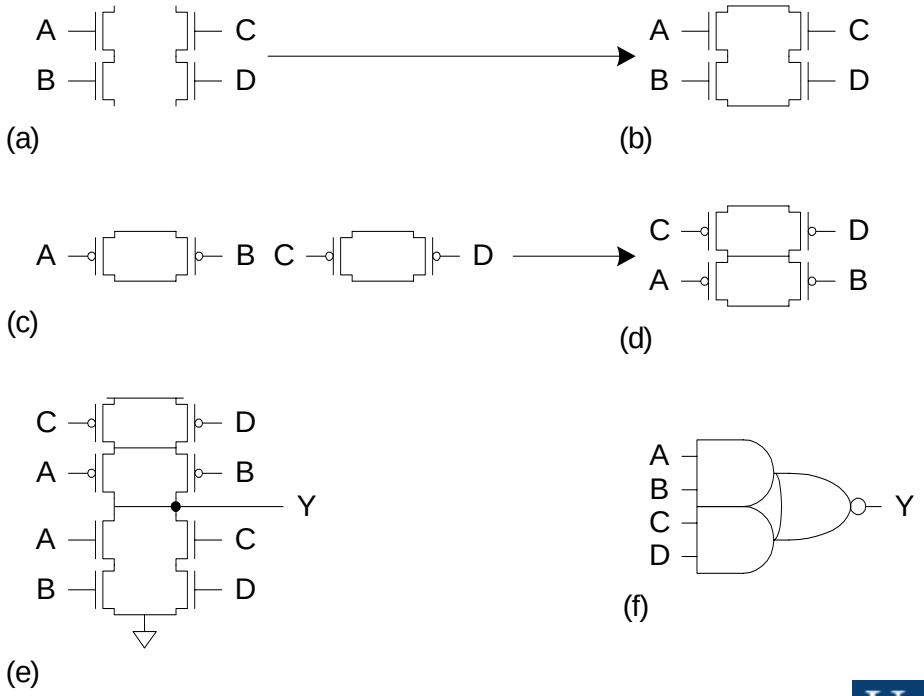
# Signal Strength

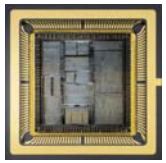
- *Strength* of signal
  - How close it approximates ideal voltage source
- $V_{DD}$ and GND rails are strongest 1 and 0
- nMOS pass strong 0
  - But degraded or weak 1
- pMOS pass strong 1
  - But degraded or weak 0
- Thus, nMOS are best for pull-down network
- And, pMOS are best for pull-up network



# Compound Gates

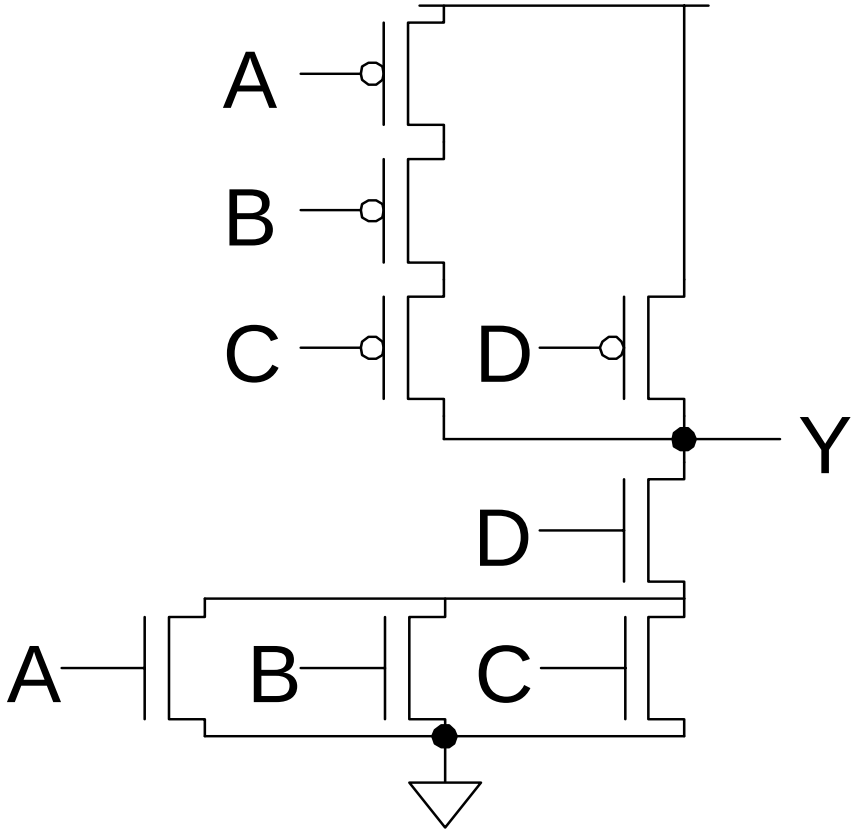
- Formed using combination of series and parallel switch structures.
- *Compound gates* can do any inverting function
- Example: $Y = \overline{A \cdot B + C \cdot D}$ (AND-AND-OR-INVERT )

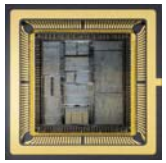




# Compound Gates …

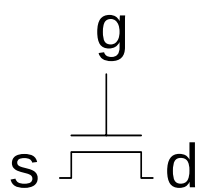
Example: $Y = \overline{(A + B + C) \cdot D}$

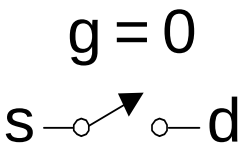


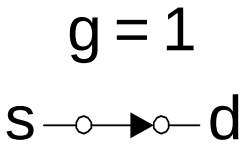


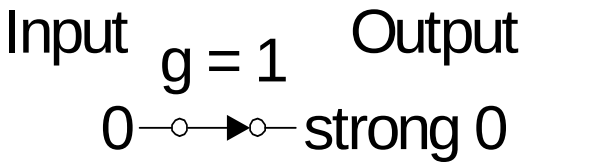
# Pass Transistors

- Transistors can be used as switches

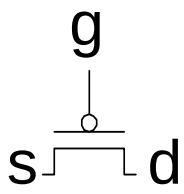


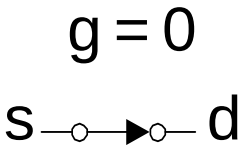



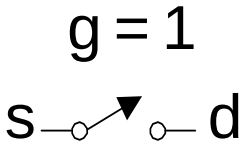



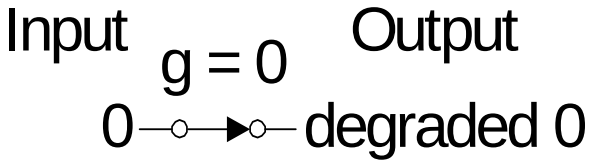



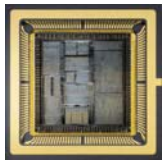




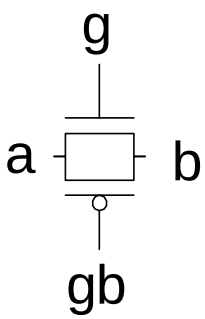



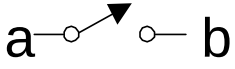



# Transmission Gates

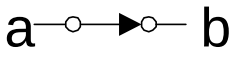
- Pass transistors produce degraded outputs
- *Transmission gates* pass both 0 and 1 well




g = 0, gb = 1

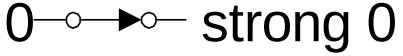



g = 1, gb = 0



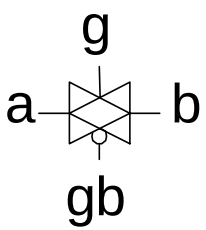

Input Output

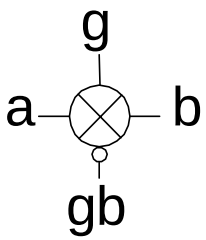
g = 1, gb = 0

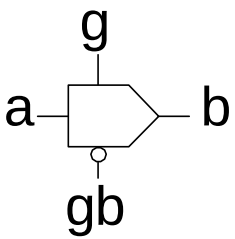


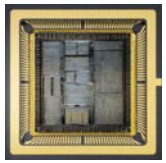

g = 1, gb = 0



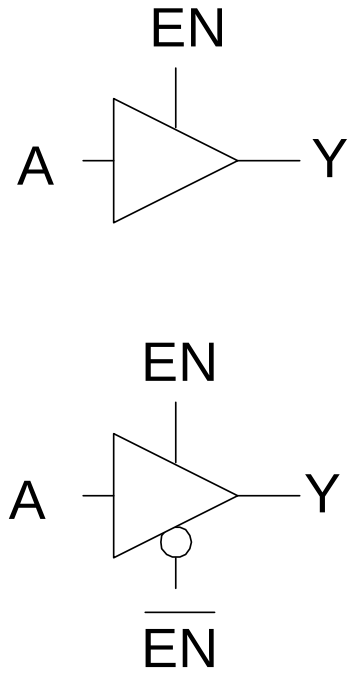



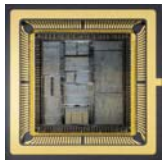


# Tristate Buffer

- *Tristate buffer* produces Z when not enabled

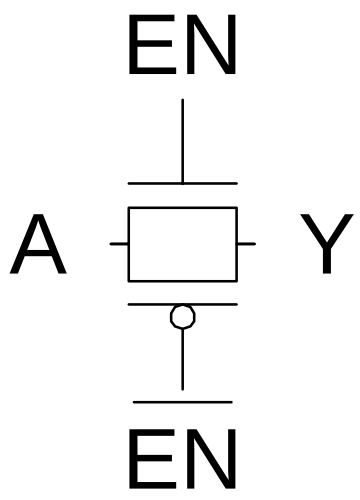
| EN | A | Y |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

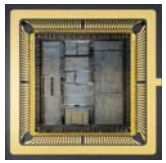




# Nonrestoring Tristate Buffer

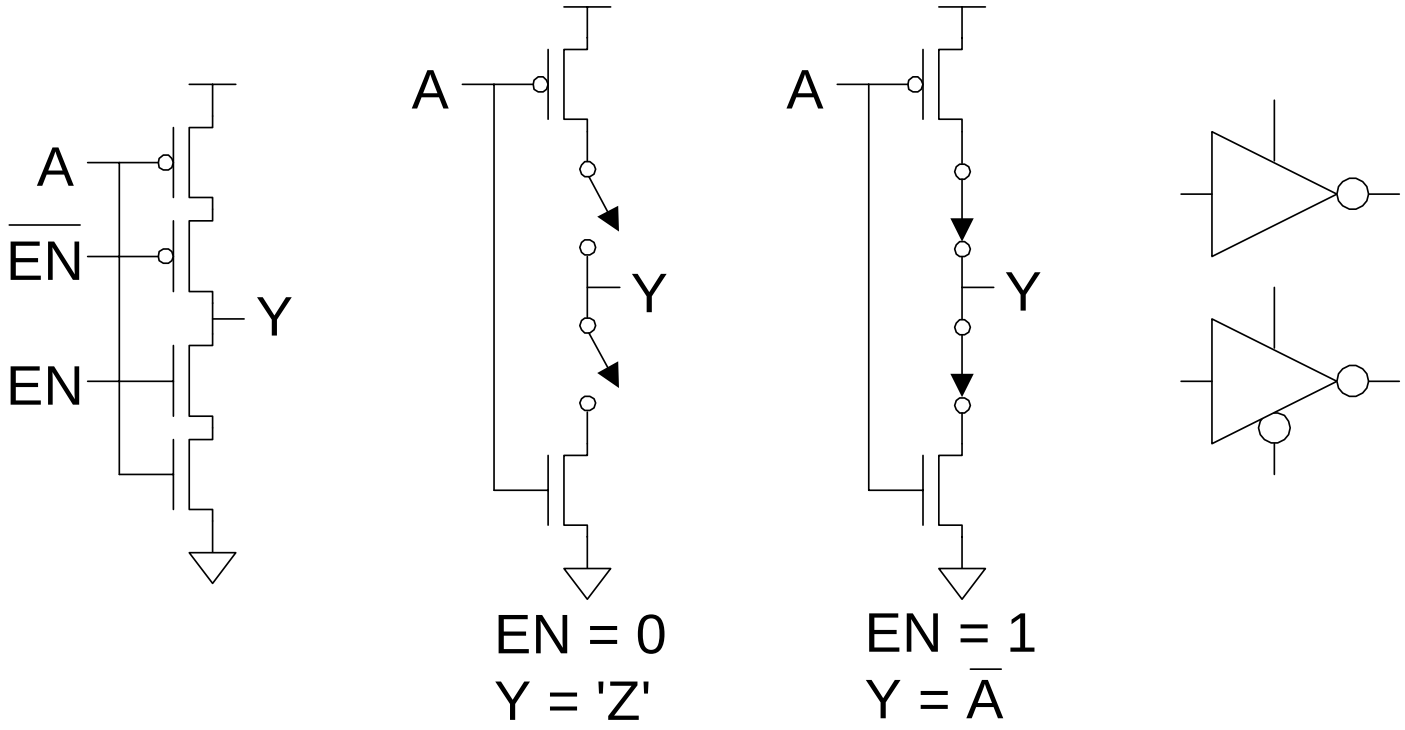
- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y

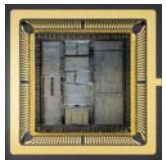




# Tristate Inverter

- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output

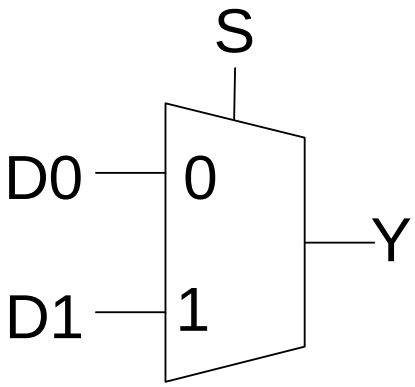




# Multiplexers

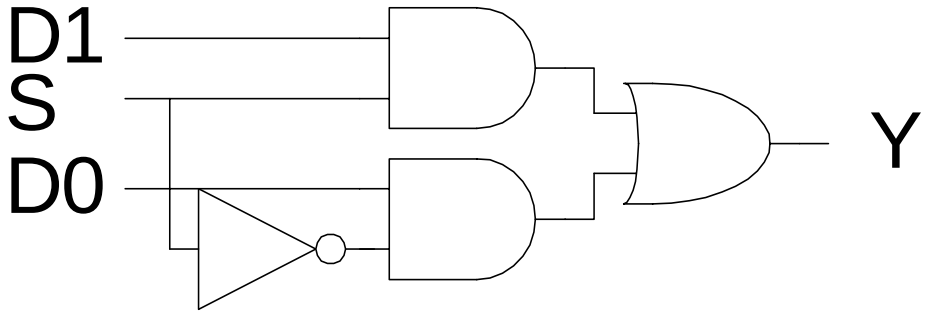
- 2:1 multiplexer chooses between two inputs

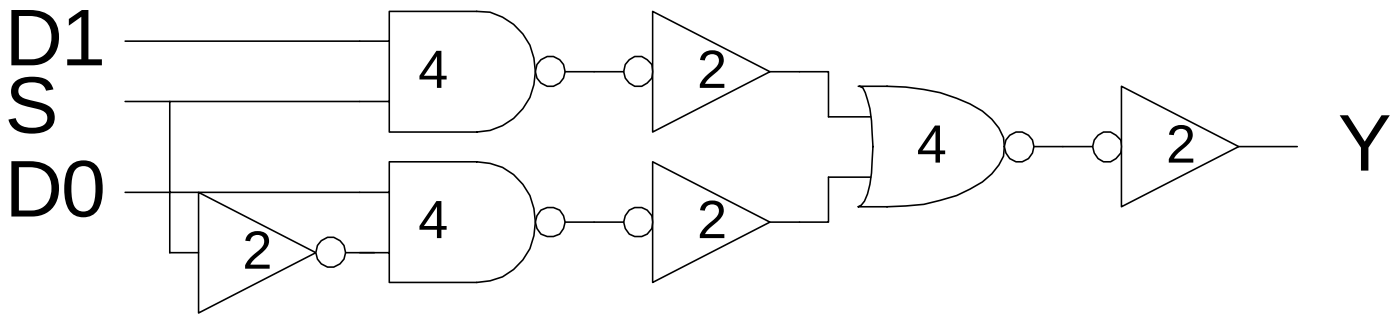
| S | D1 | D0 | Y |
|---|---|---|---|
| 0 | X | 0 | 0 |
| 0 | X | 1 | 1 |
| 1 | 0 | X | 0 |
| 1 | 1 | X | 1 |



# Multiplexers Design : One Approach

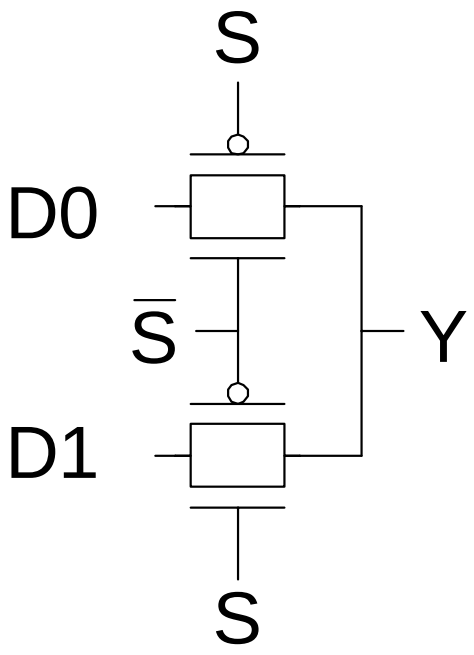
- $Y = SD_1 + \overline{S}D_0$ (too many transistors)
- How many transistors are needed? 20

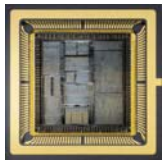


# Multiplexers Design : 2nd Approach

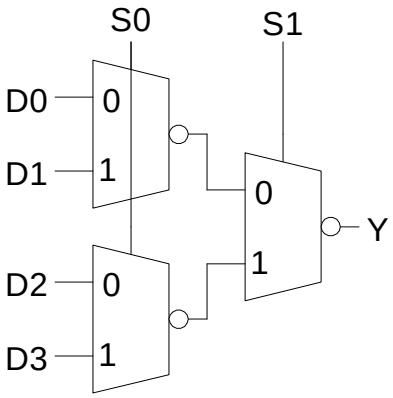
- Nonrestoring mux uses two transmission gates
  - Only 4 transistors

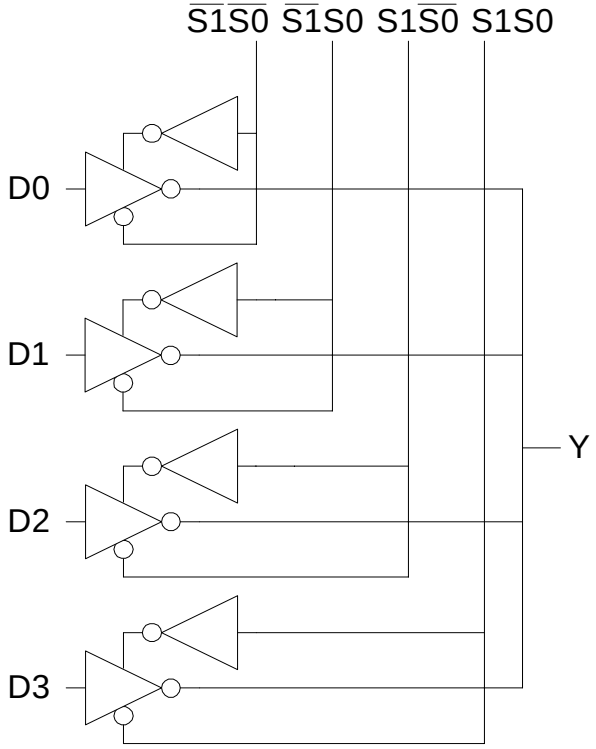


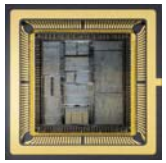


# 4:1 Multiplexer

- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 2:1 muxes
  - Or four tristates

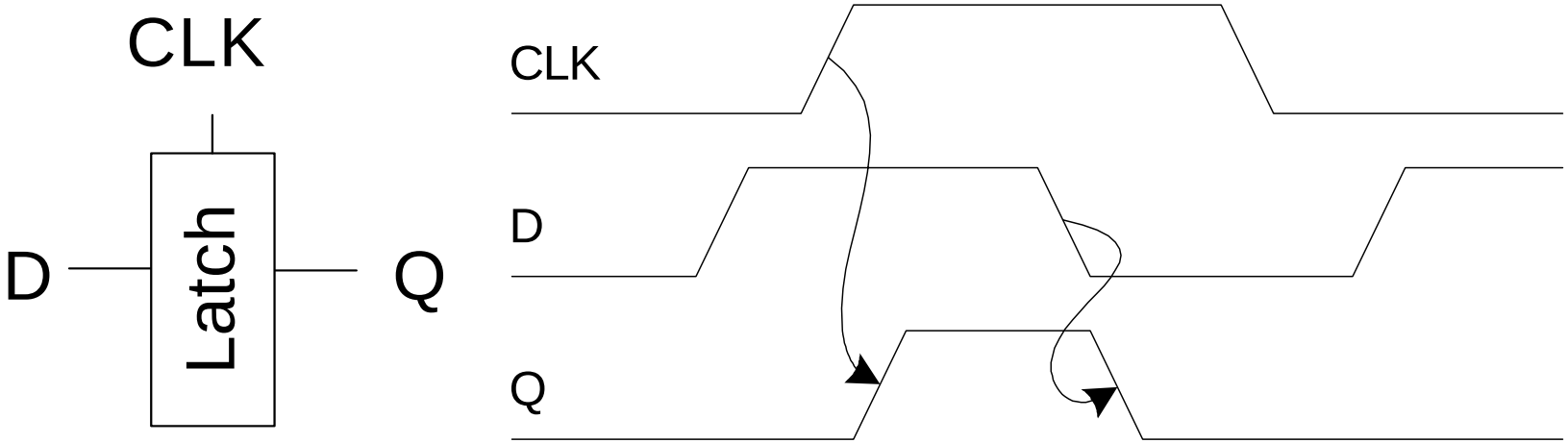


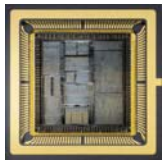





# D Latch

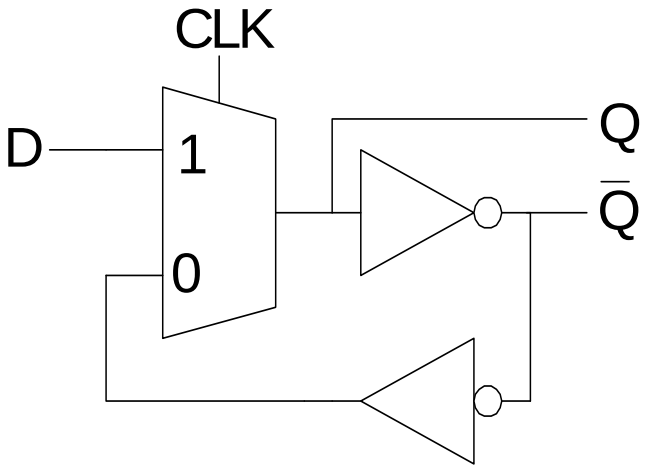
- When CLK = 1, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = 0, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

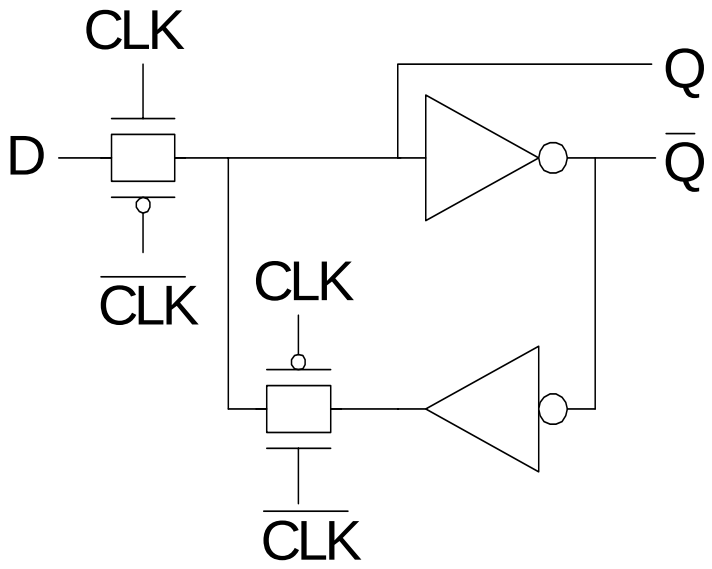




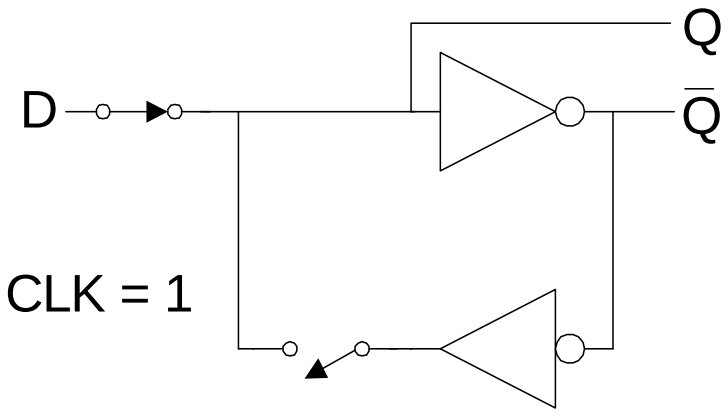
# D Latch : Design

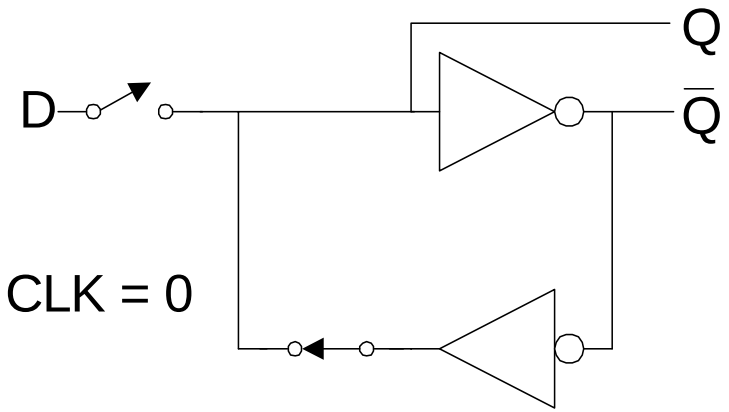
- Multiplexer chooses D or old Q

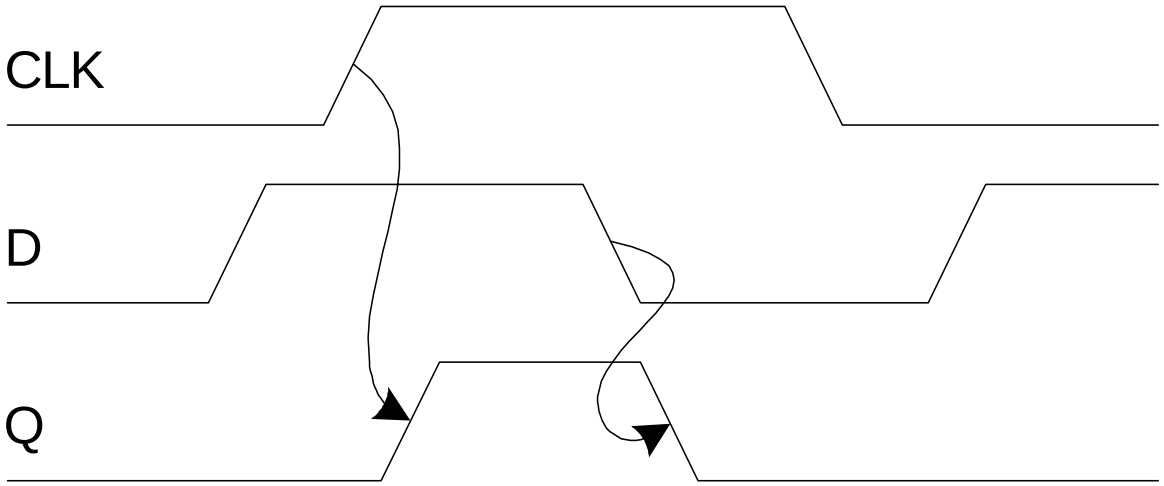


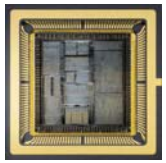



# D Latch : Operation



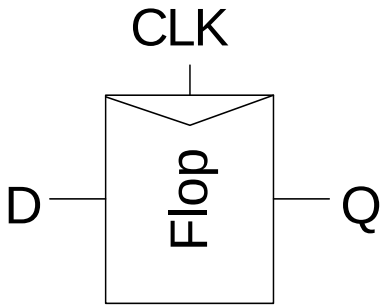



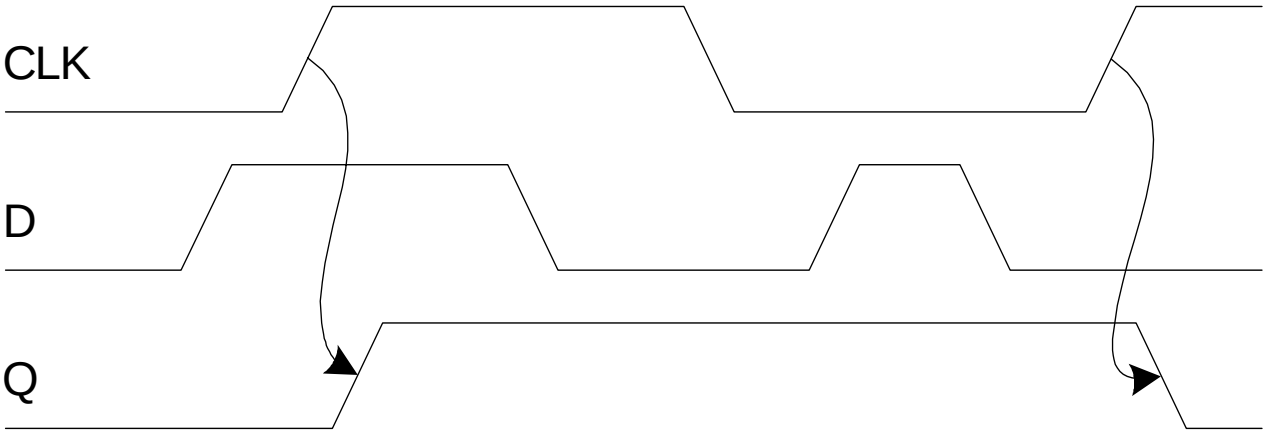



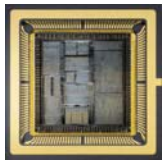


# D Flip-flop

- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

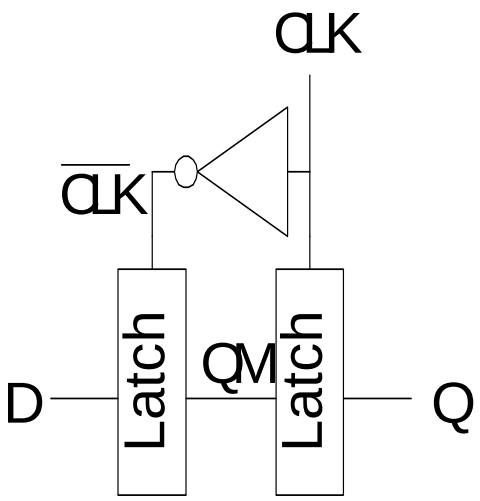


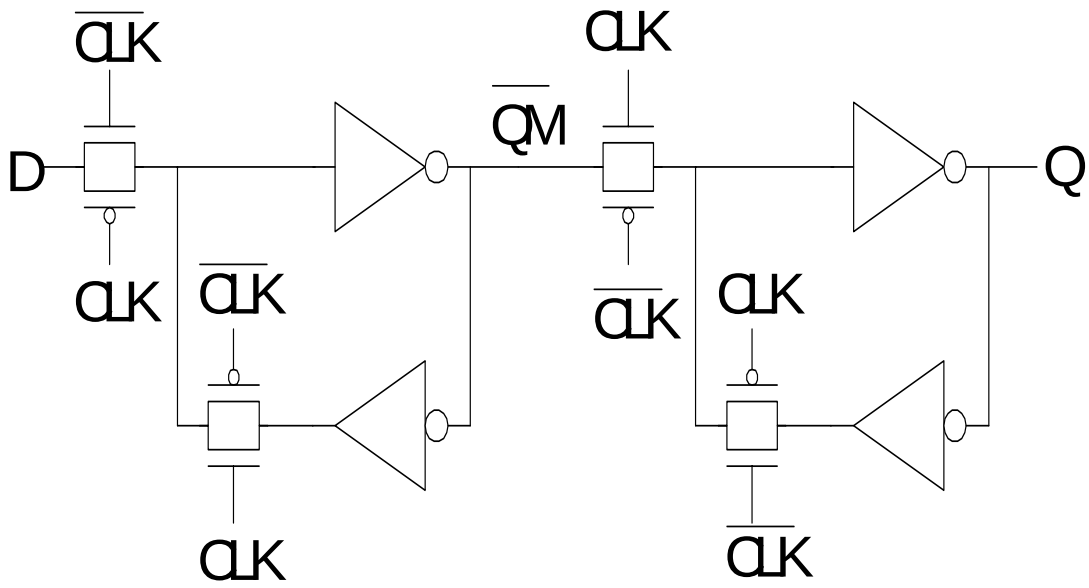



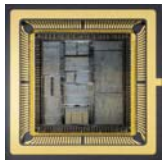


# D Flip-flop : Design

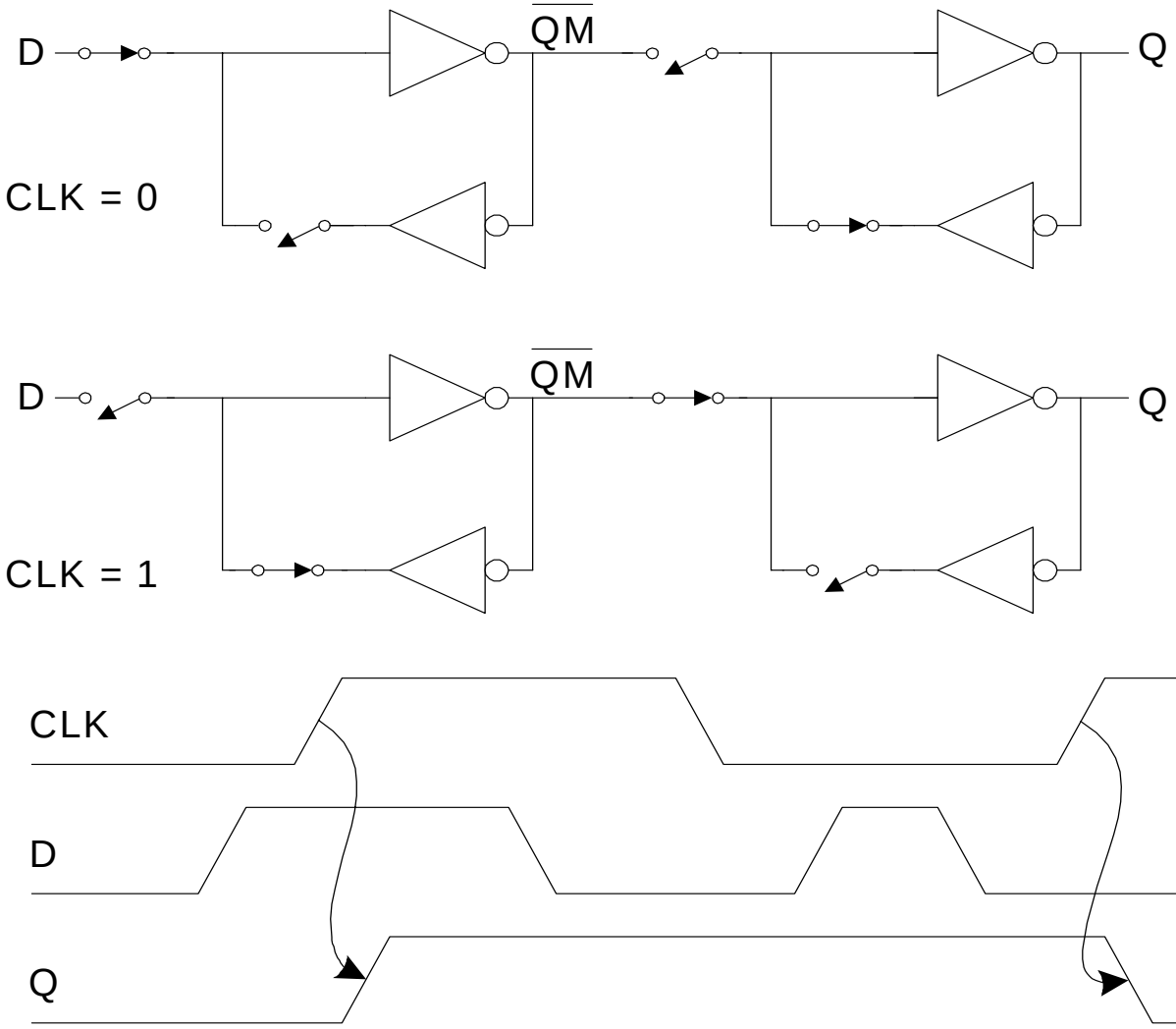
- Built from master and slave D latches

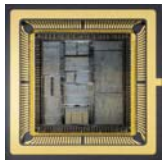




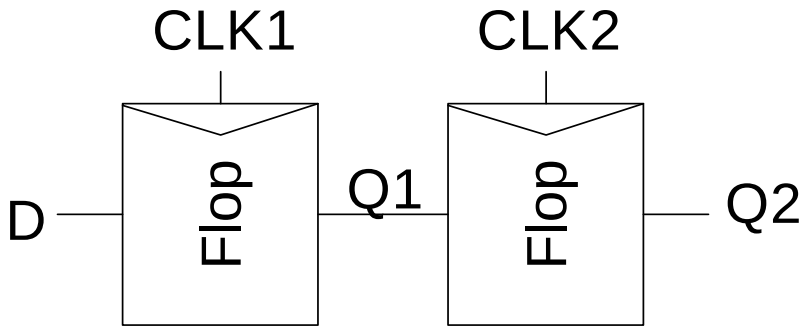
# D Flip-flop : Operation

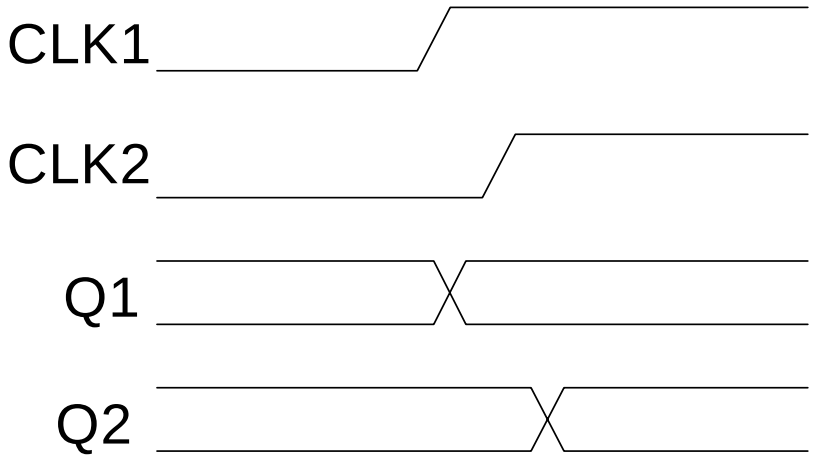


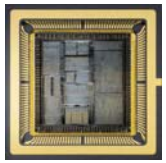


# D-FF : Race Condition

- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*





# D-FF: Nonoverlapping Clocks

- Nonoverlapping clocks can prevent races
  - As long as nonoverlap exceeds clock skew
- We will use them in this class for safe design
  - Industry manages skew more carefully instead