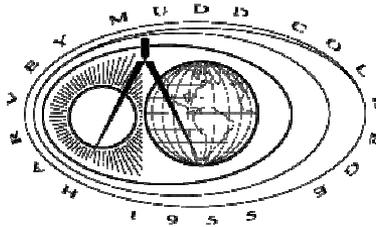# Introduction to CMOS VLSI Design

Lecture 1: Circuits & Layout

David Harris

Harvey Mudd College

Spring 2004

Outline

- ❑ A Brief History
- ❑ CMOS Gate Design
- ❑ Pass Transistors
- ❑ CMOS Latches & Flip-Flops
- ❑ Standard Cell Layouts
- ❑ Stick Diagrams

A Brief History

❑ 1958: First integrated circuit
  – Flip-flop using two transistors
  – Built by Jack Kilby at Texas Instruments
❑ 2003
  – Intel Pentium 4 μprocessor (55 million transistors)
  – 512 Mbit DRAM (> 0.5 billion transistors)
❑ 53% compound annual growth rate over 45 years
  – No other technology has grown so fast so long
❑ Driven by miniaturization of transistors
  – Smaller is cheaper, faster, lower in power!
  – Revolutionary effects on society

Annual Sales

❑ 10^{18} transistors manufactured in 2003
  – 100 million for every human on the planet
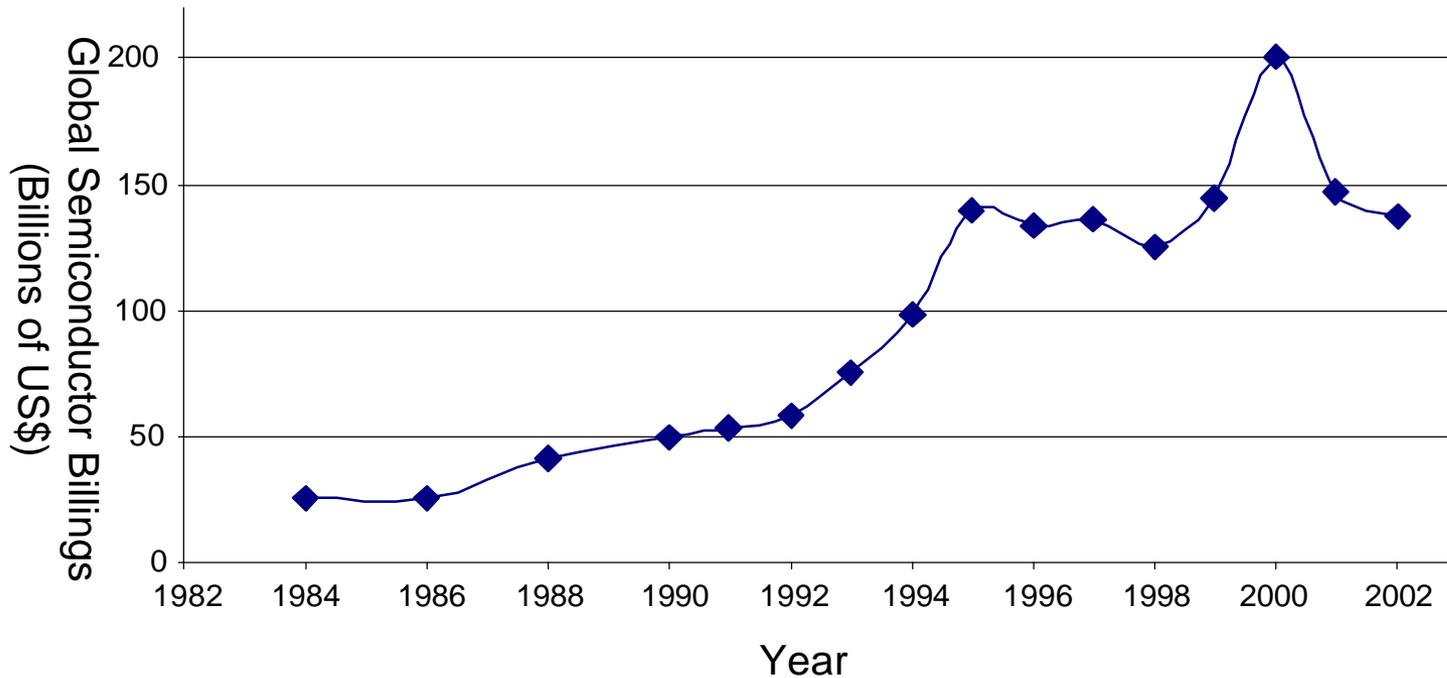
Invention of the Transistor

- Vacuum tubes ruled in first half of 20th century Large, expensive, power-hungry, unreliable
- 1947: first point contact transistor
  - John Bardeen and Walter Brattain at Bell Labs
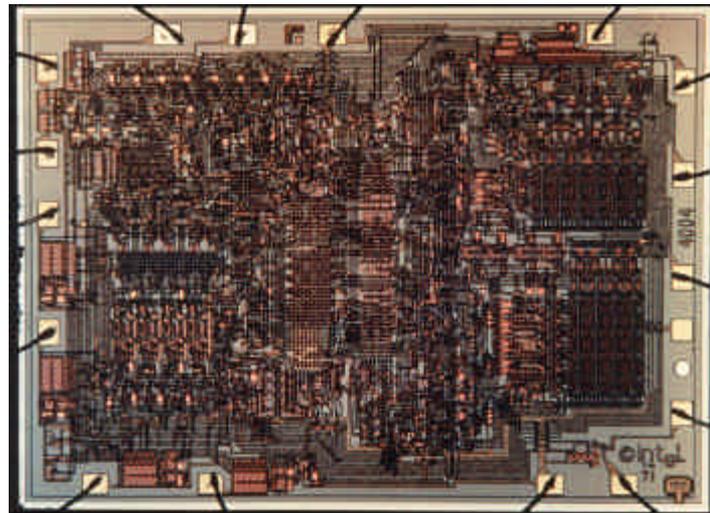 - Read Crystal Fire
    By Riordan, Hoddeson

Transistor Types

❑ Bipolar transistors

- npn or pnp silicon structure
- Small current into very thin base layer controls large currents between emitter and collector
- Base currents limit integration density

❑ Metal Oxide Semiconductor Field Effect Transistors

- nMOS and pMOS MOSFETS
- Voltage applied to insulated gate controls current between source and drain
- Low power allows very high integration

MOS Integrated Circuits

❑ 1970's processes usually had only nMOS transistors

– Inexpensive, but consume power while idle
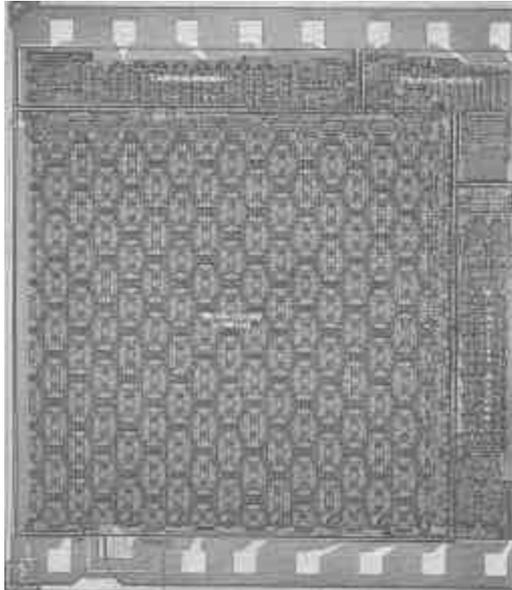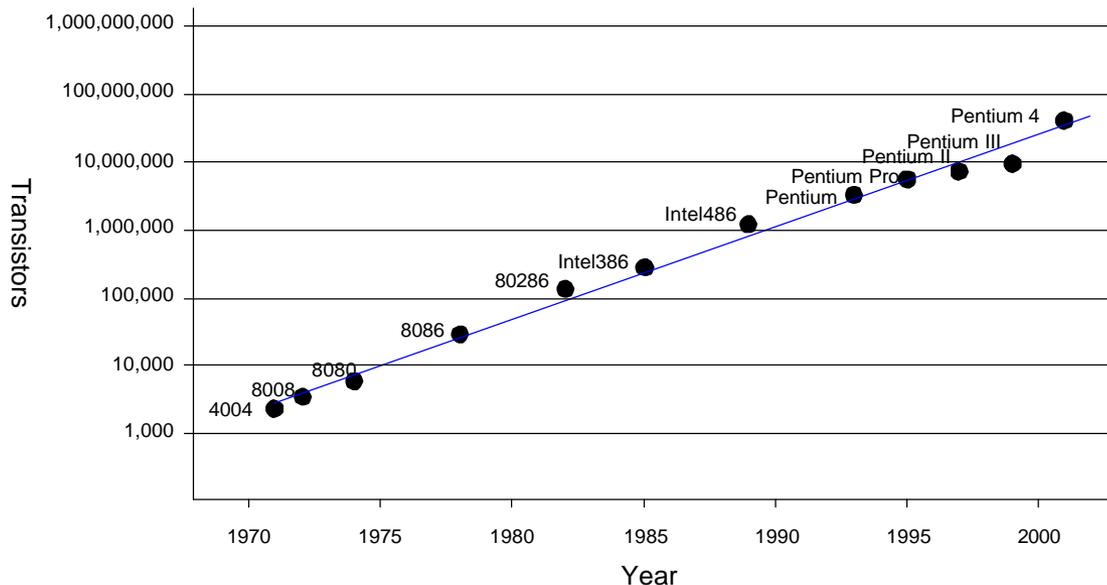

Intel 1101 256-bit SRAM   Intel 4004 4-bit μProc

❑ 1980s-present: CMOS processes for low idle power

Moore's Law

❑ 1965: Gordon Moore plotted transistor on each chip

– Fit straight line on semilog scale

– Transistor counts have doubled every 26 months
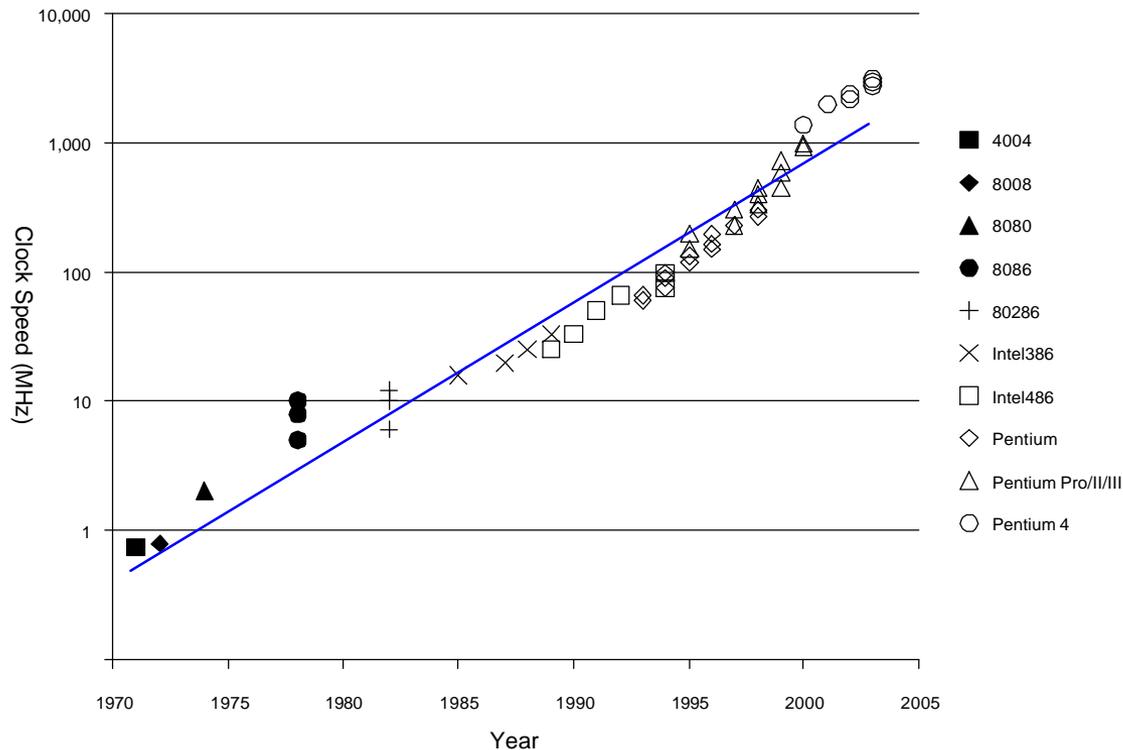

Integration Levels

SSI: 10 gates

MSI: 1000 gates

LSI: 10,000 gates

VLSI: > 10k gates

Corollaries

❑ Many other factors grow exponentially
  – Ex: clock frequency, processor performance

CMOS Gate Design
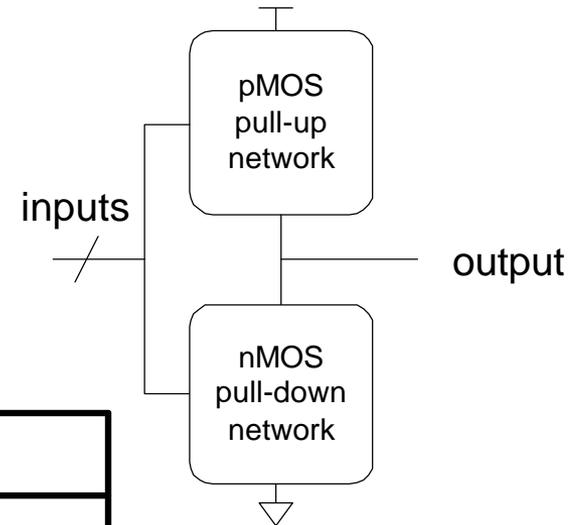
❑ Activity:
   – Sketch a 4-input CMOS NAND gate

CMOS Gate Design

❑ Activity:

– Sketch a 4-input CMOS NOR gate
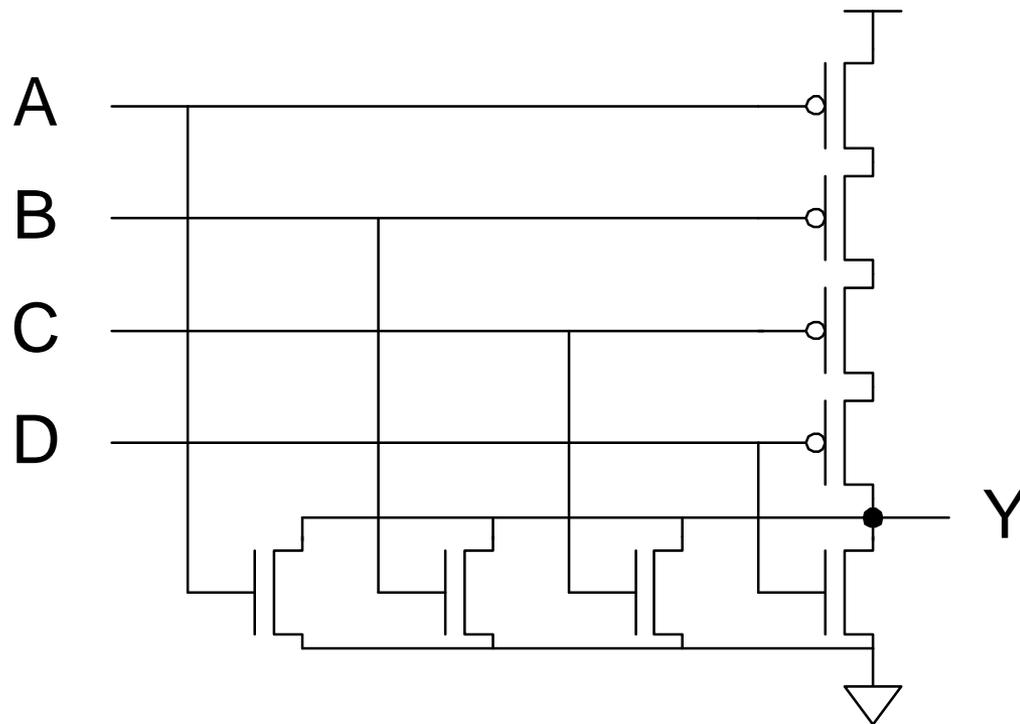
Complementary CMOS

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

pMOS
pull-up
network

inputs

output

nMOS
pull-down
network

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

Series and Parallel

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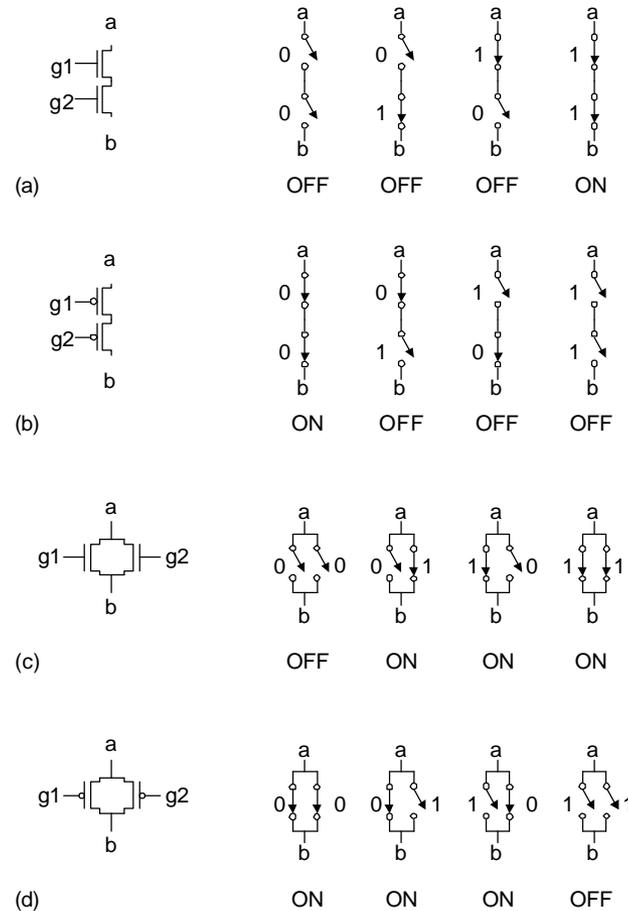
Conduction Complement

❑ 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
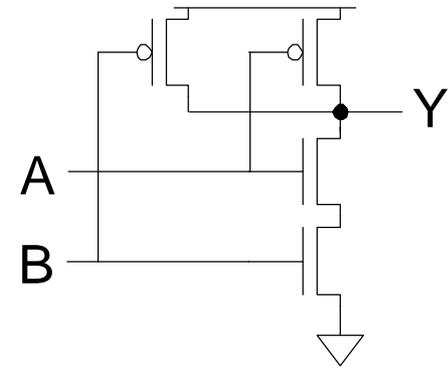
Compound Gates

❑ *Compound gates* can do any inverting function

❑ Ex: $Y = \overline{A \cdot B + C \cdot D}$ (AND-AND-OR-INVERT, AOI22)

(a)

(b)

(c)

(d)

(e)

(f)
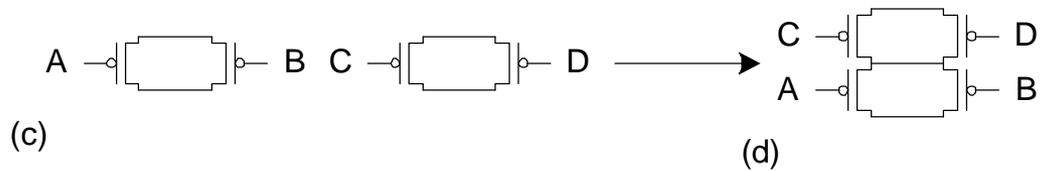
Example: O3AI

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

Example: O3AI

❑ $Y = \overline{(A + B + C) \cdot D}$
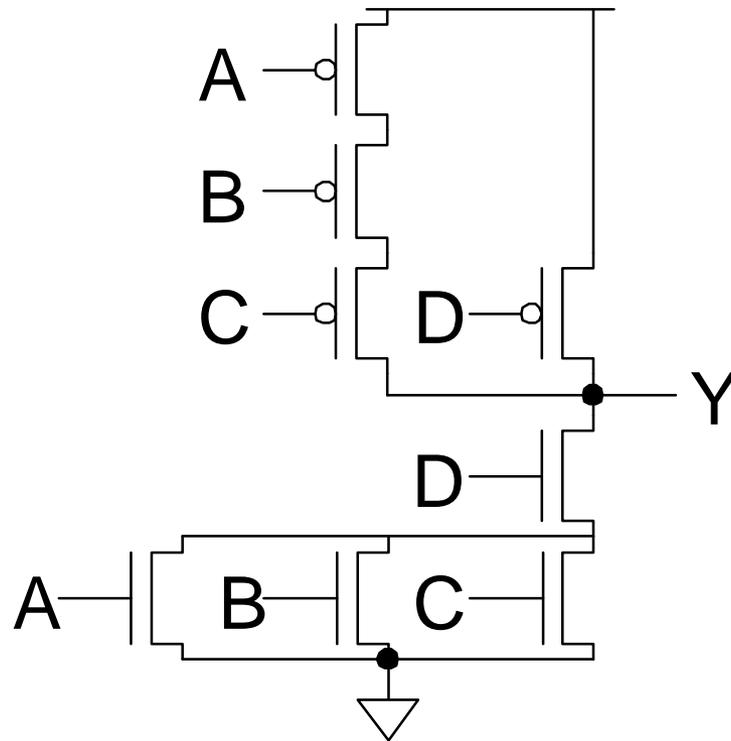
Signal Strength

❑ *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

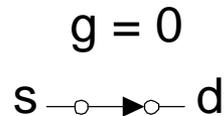– But degraded or weak 0

❑ Thus nMOS are best for pull-down network

Pass Transistors
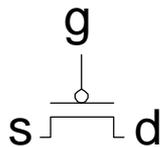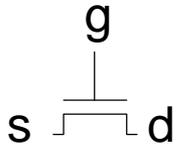
❑   Transistors can be used as switches

g
s ⊣⊢ d

g
s ⊣∘⊢ d

Pass Transistors

❑ Transistors can be used as switches

g

s ⌐ d

g = 0

s —○ ▸—○— d

g = 1

s —○—▸—○— d

Input   Output
       g = 1
0 —○—▸○— strong 0

       g = 1
1 —○—▸○— degraded 1

g

s ⌐ d

g = 0

s —○—▸—○— d

g = 1

s —○ ▸—○— d

Input   Output
       g = 0
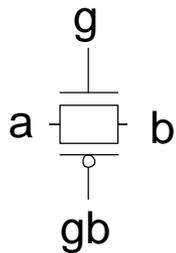0 —○—▸○— degraded 0
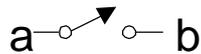
       g = 0
—○—▸○— strong 1
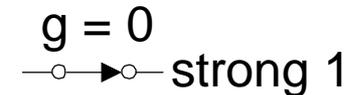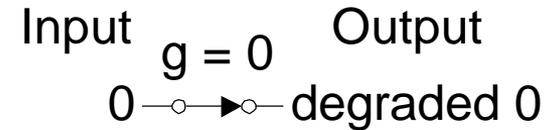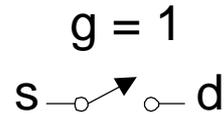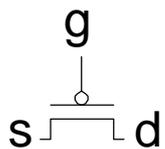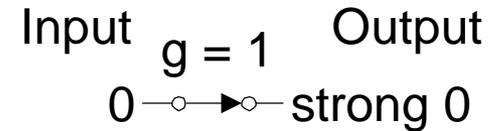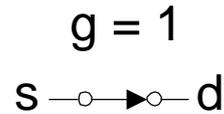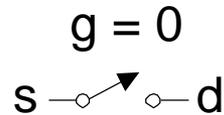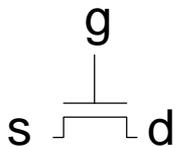
Transmission Gates

❑ Pass transistors produce degraded outputs

❑ *Transmission gates* pass both 0 and 1 well

Transmission Gates

❑ Pass transistors produce degraded outputs

❑ *Transmission gates* pass both 0 and 1 well
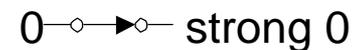
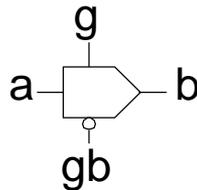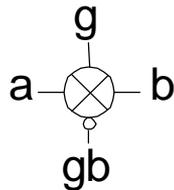Input   Output

g = 0, gb = 1   g = 1, gb = 0
a —○—▶—○— b   0 —○—▶○— strong 0

g = 1, gb = 0   g = 1, gb = 0
a —○—▶○— b   1 —○—▶○— strong 1
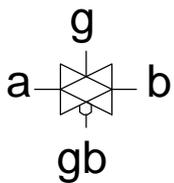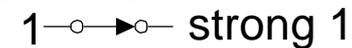
Tristates

❑ *Tristate buffer* produces Z when not enabled

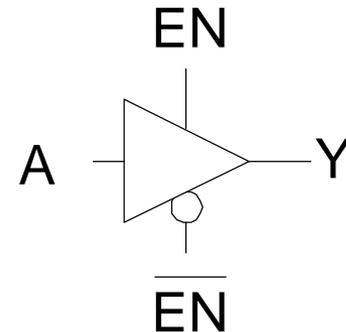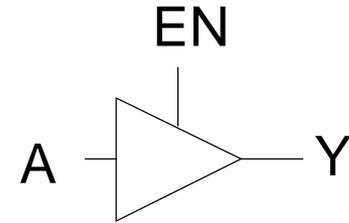| EN	A	Y
0	0	
0	1	
1	0	
1	1	
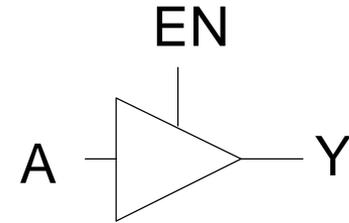
Tristates

❑ *Tristate buffer* produces Z when not enabled

| EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1
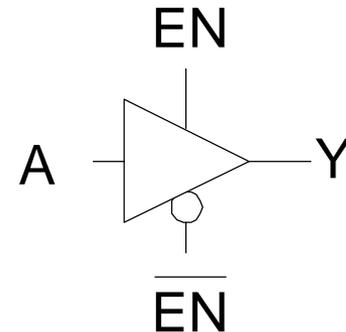
Nonrestoring Tristate

❑ Transmission gate acts as tristate buffer

– Only two transistors

– But *nonrestoring*

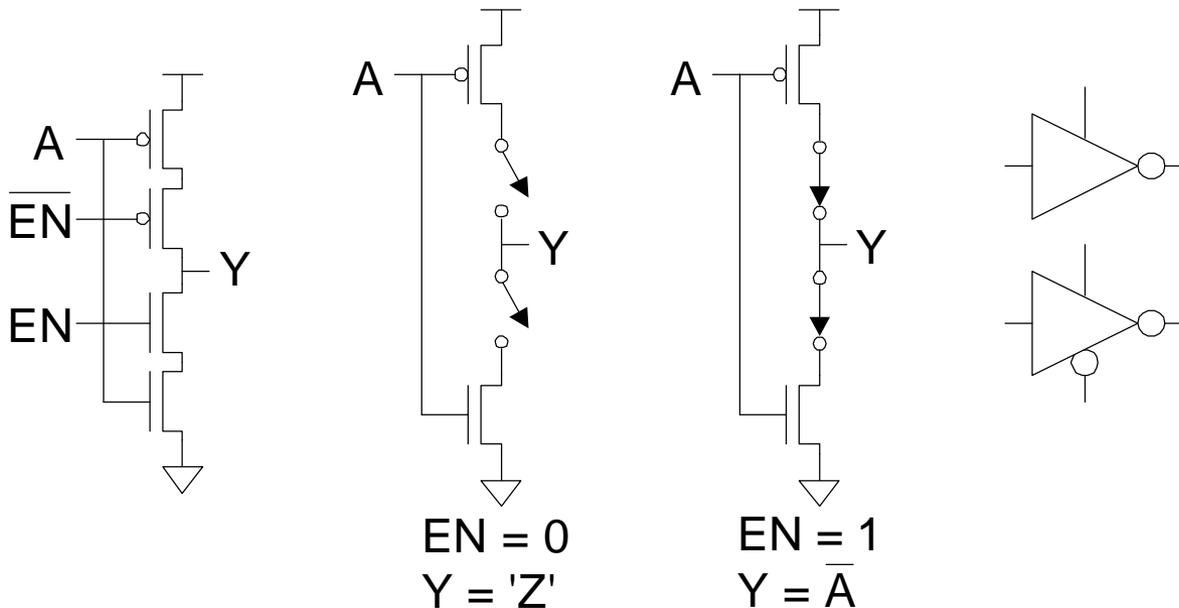• Noise on A is passed on to Y
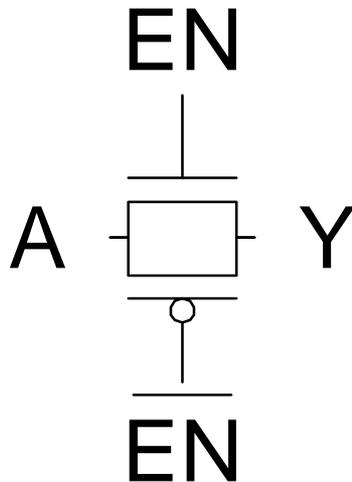
EN

A ⊣□⊢ Y

EN

Tristate Inverter

❑ Tristate inverter produces restored output

– Violates conduction complement rule
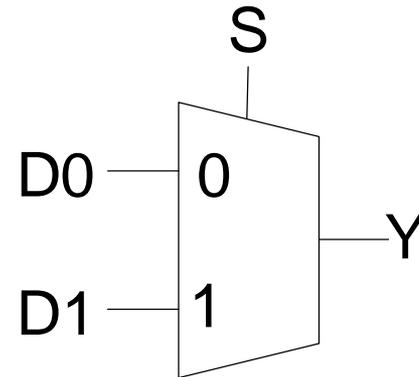
– Because we want a Z output

Tristate Inverter

❑ Tristate inverter produces restored output

– Violates conduction complement rule

– Because we want a Z output

EN = 0
Y = 'Z'

EN = 1
Y = $\overline{A}$
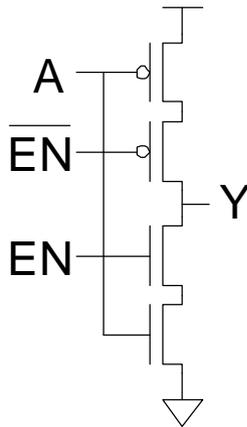
Multiplexers

❑ 2:1 *multiplexer* chooses between two inputs

S	D1	D0	Y
0	X	0	
0	X	1	
1	0	X	
1	1	X	

Multiplexers

❑ 2:1 multiplexer chooses between two inputs

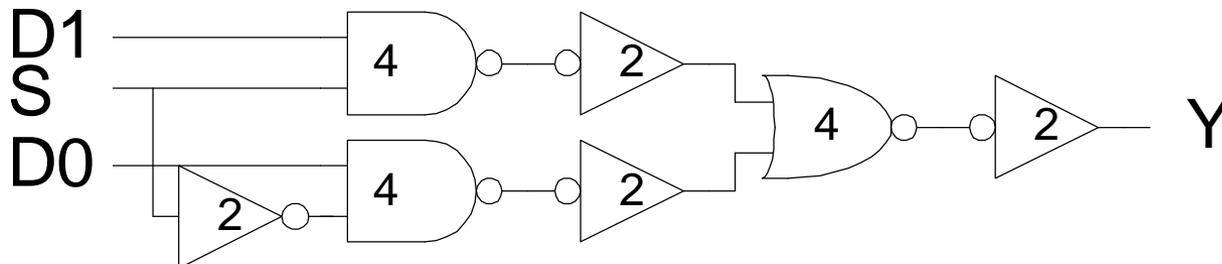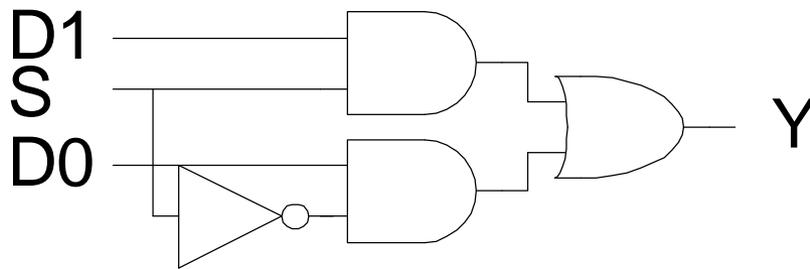| S	D1	D0	Y
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1
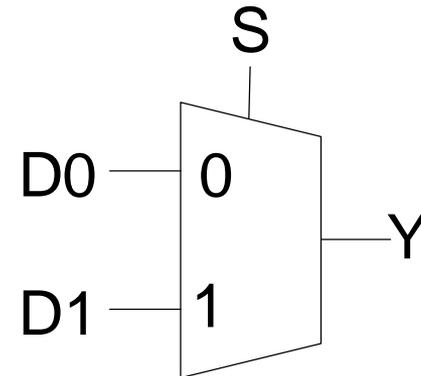
Gate-Level Mux Design

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

Gate-Level Mux Design

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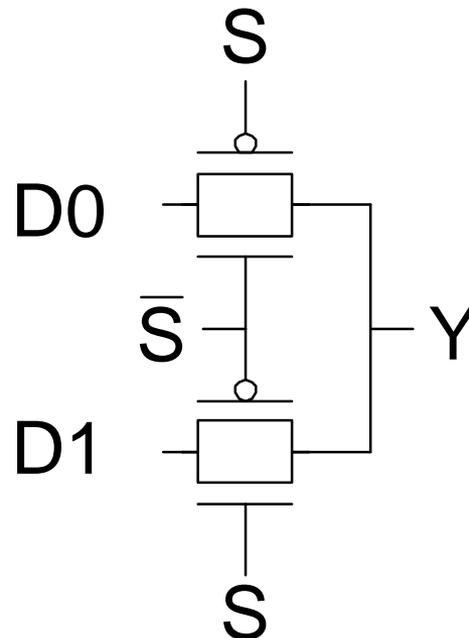
❑ How many transistors are needed? 20

Transmission Gate Mux
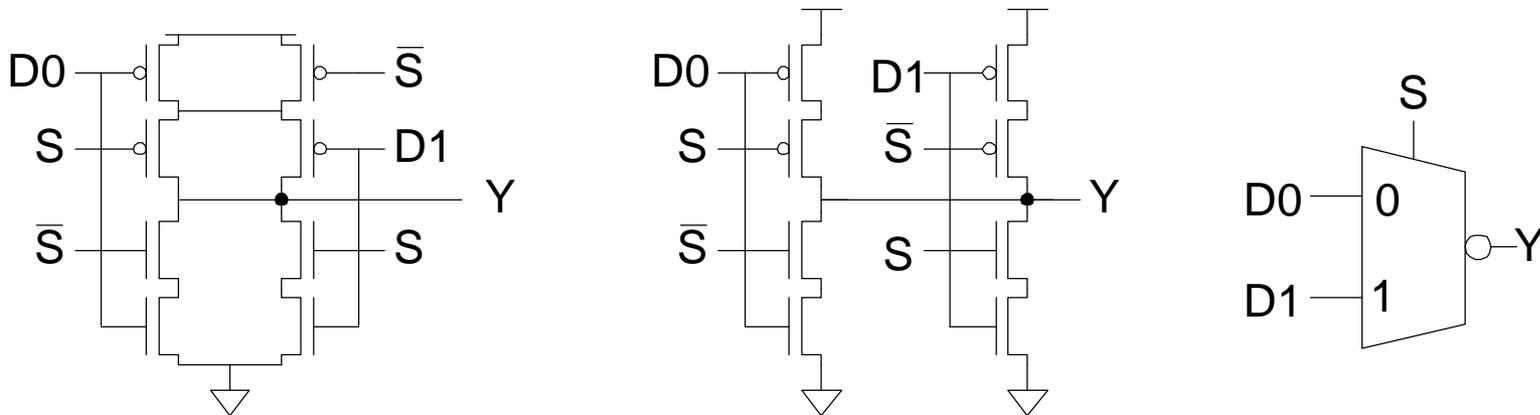
❑ Nonrestoring mux uses two transmission gates

Transmission Gate Mux

❑ Nonrestoring mux uses two transmission gates

    – Only 4 transistors

Inverting Mux

❑ Inverting multiplexer
  – Use compound AOI22
  – Or pair of tristate inverters
  – Essentially the same thing

❑ Noninverting multiplexer adds an inverter
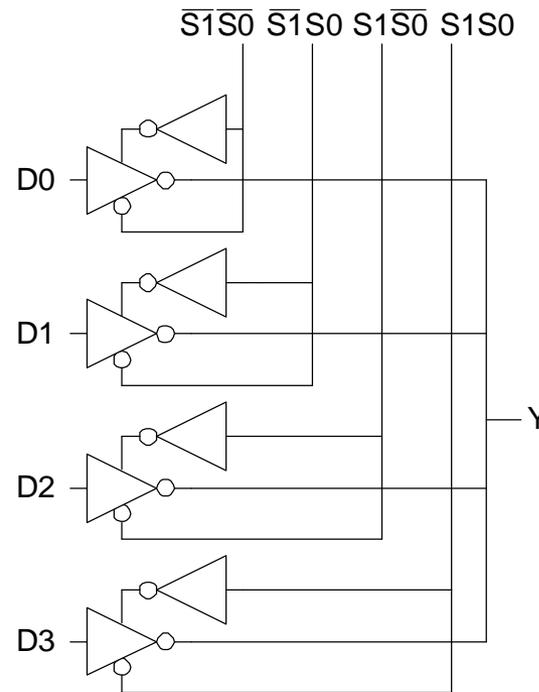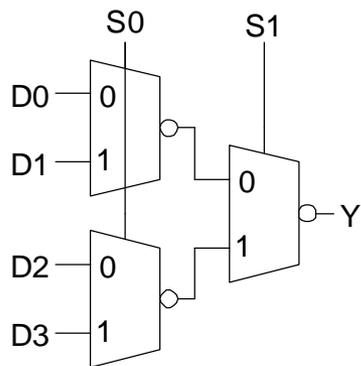
4:1 Multiplexer

❑ 4:1 mux chooses one of 4 inputs using two selects

4:1 Multiplexer

❑ 4:1 mux chooses one of 4 inputs using two selects

- – Two levels of 2:1 muxes
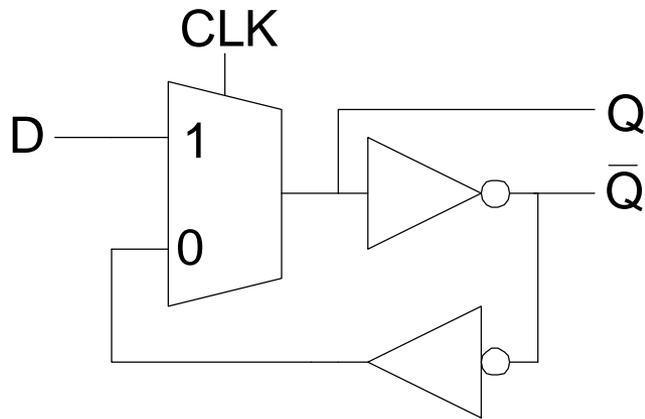- – Or four tristates

$\overline{S1}\,\overline{S0}$ $\overline{S1}\,S0$ $S1\,\overline{S0}$ $S1\,S0$

D0

D1

Y

D2

D3

S0   S1

D0 ── 0

D1 ── 1
        0
         Y
        1
D2 ── 0

D3 ── 1

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*

CLK
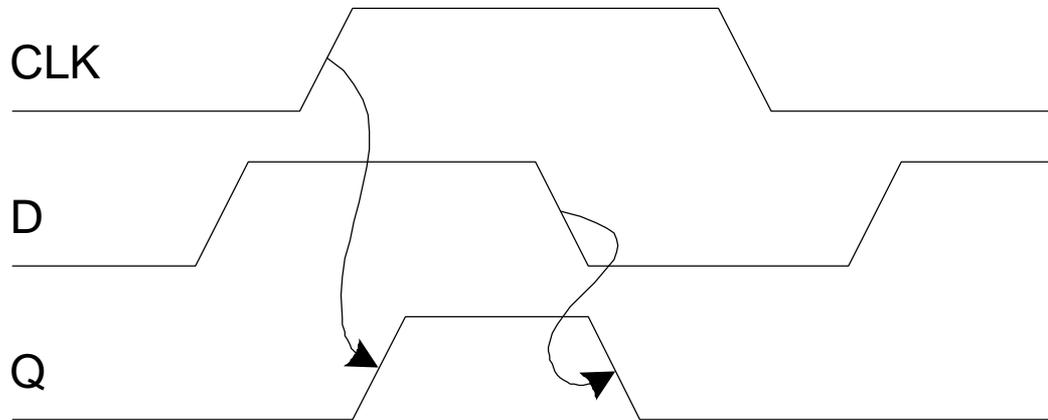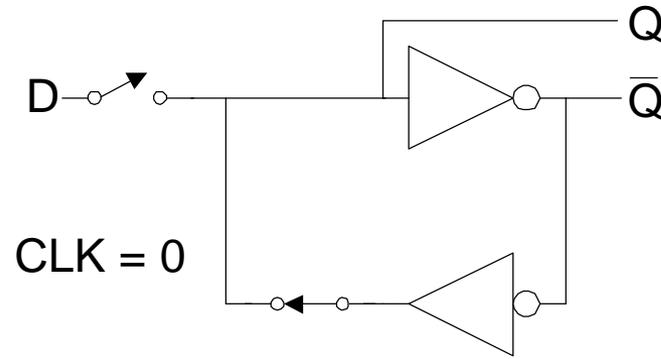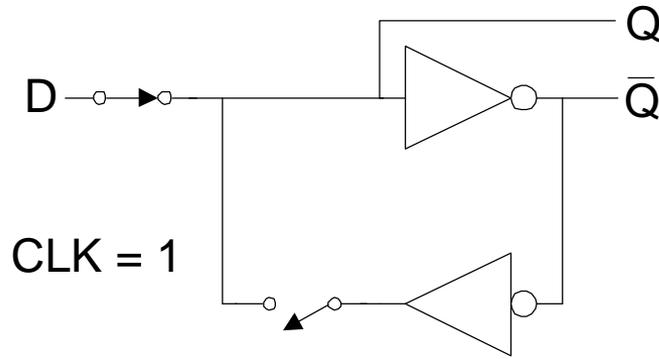
D — Latch — Q

CLK

D

Q
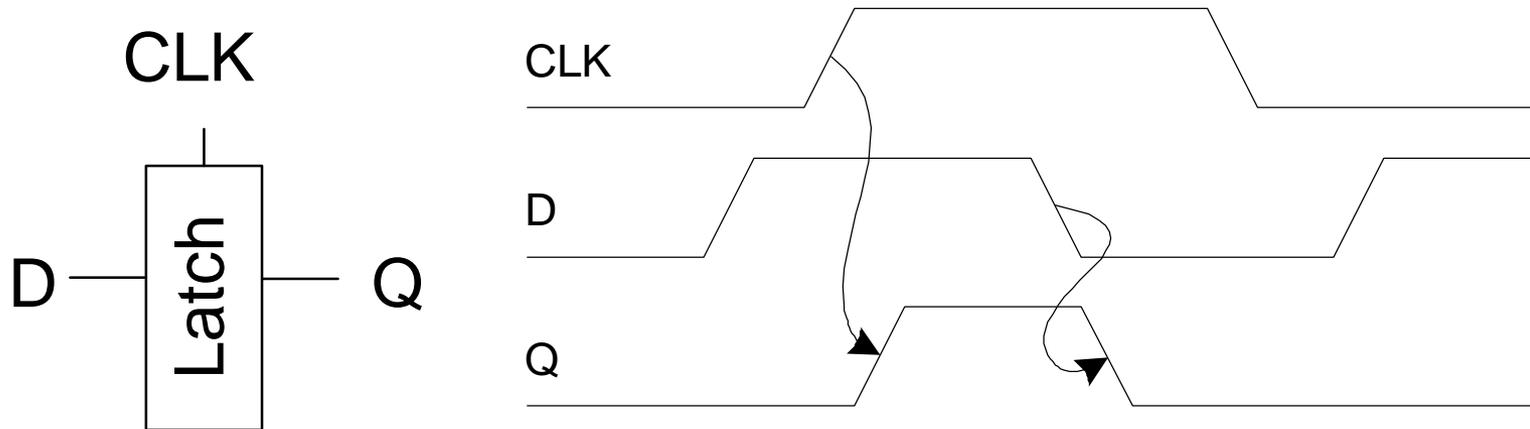
D Latch Design

❑ Multiplexer chooses D or old Q

D Latch Operation

CLK = 1

CLK = 0

D   Q / Q̄

D   Q / Q̄

CLK
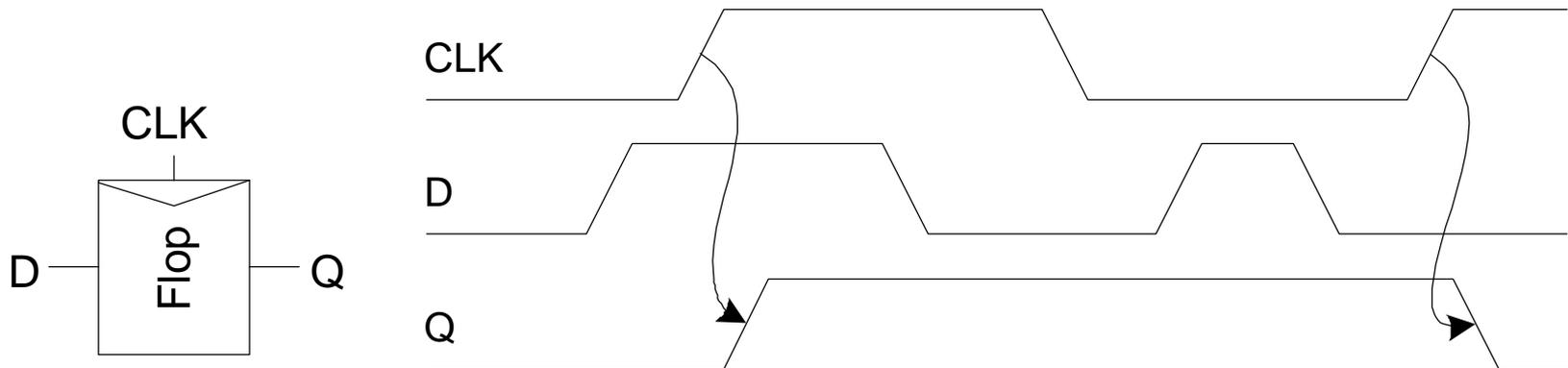
D

Q
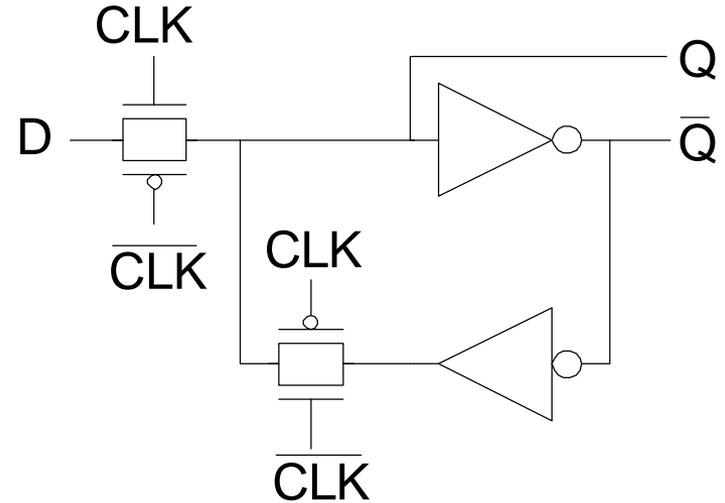
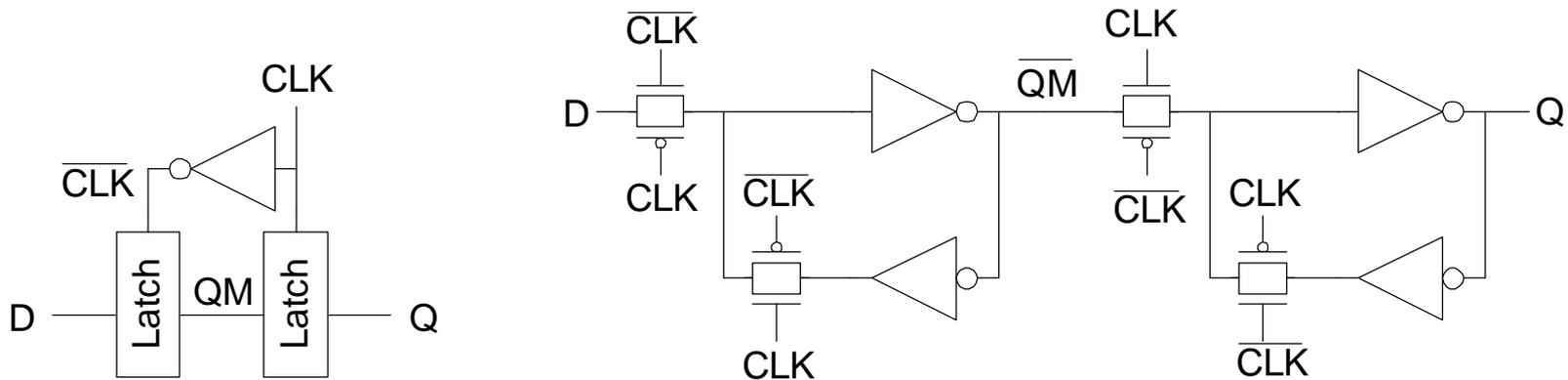# D Flip-flop

❑ When CLK rises, D is copied to Q

❑ At all other times, Q holds its value

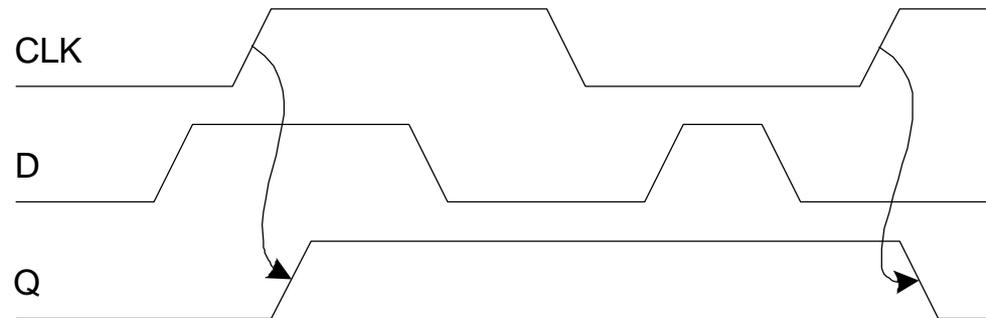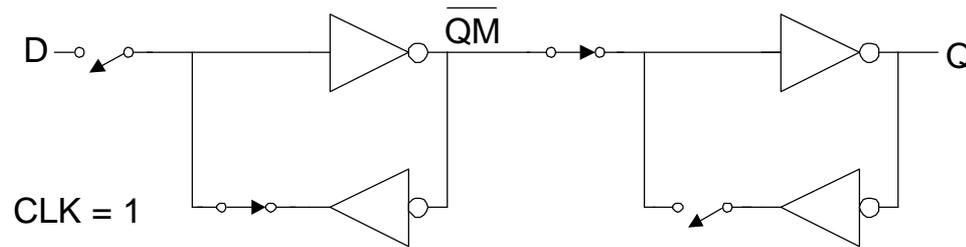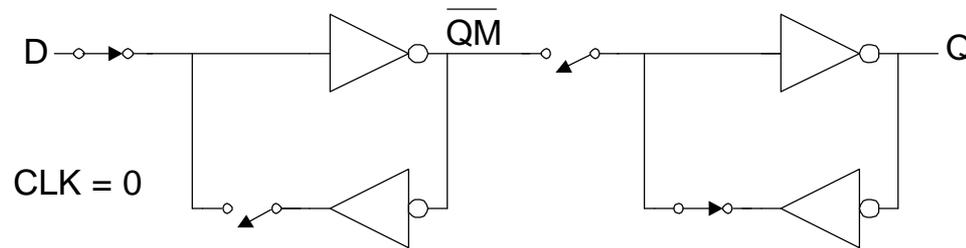❑ a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

D Flip-flop Design
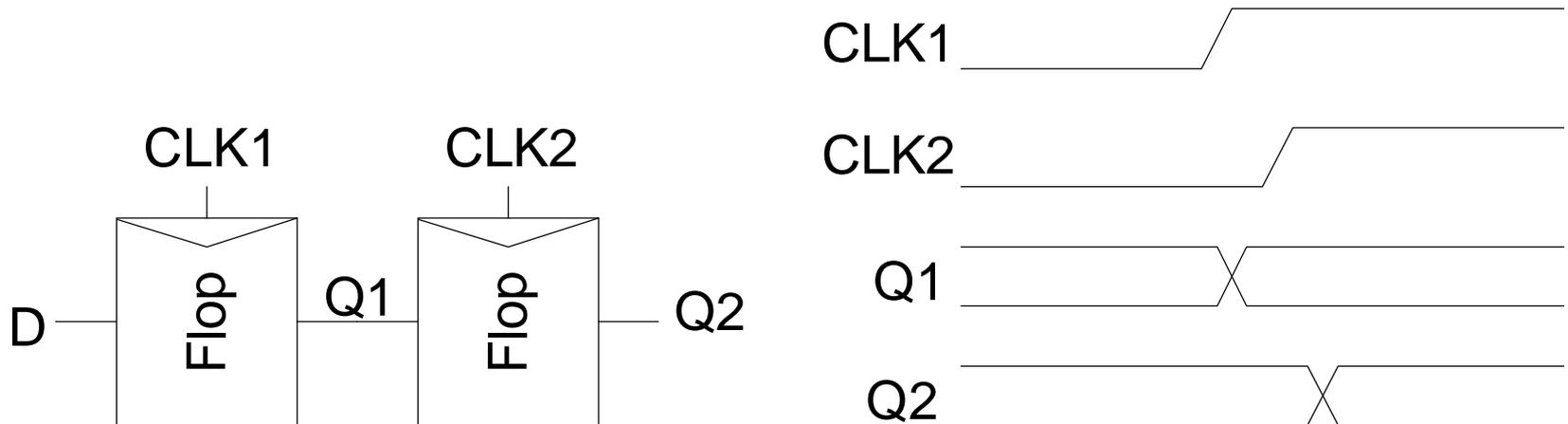
❑ Built from master and slave D latches

D Flip-flop Operation

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*

CLK1

CLK2

D — Flop — Q1 — Flop — Q2

CLK1

CLK2

Q1

Q2

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
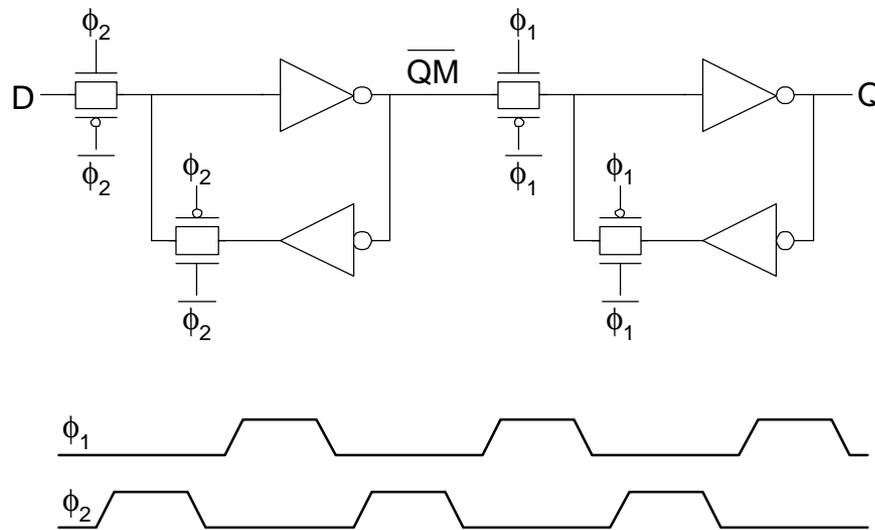
Gate Layout

❑ Layout can be very time consuming

   – Design gates to fit together nicely

   – Build a library of standard cells

❑ Standard cell design methodology

   – V_{DD} and GND should abut (standard height)

   – Adjacent gates should satisfy design rules

   – nMOS at bottom and pMOS at top

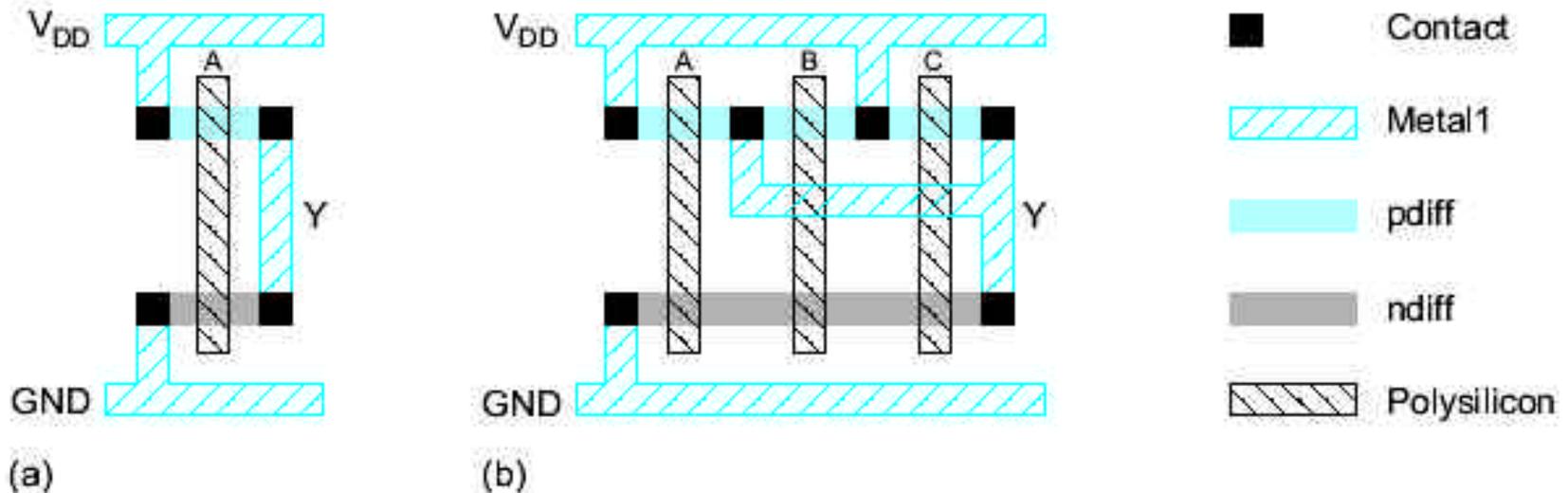   – All gates include well and substrate contacts

Example: Inverter
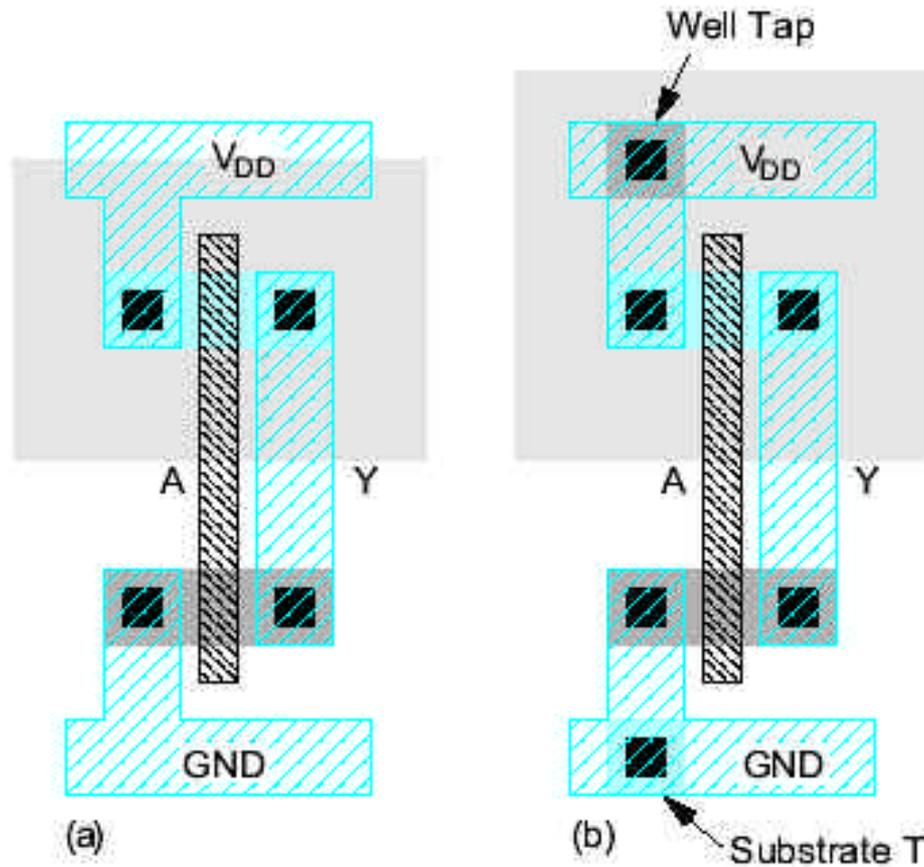
Example: NAND3

- ❑ Horizontal N-diffusion and p-diffusion strips
- ❑ Vertical polysilicon gates
- ❑ Metal1 V_{DD} rail at top
- ❑ Metal1 GND rail at bottom
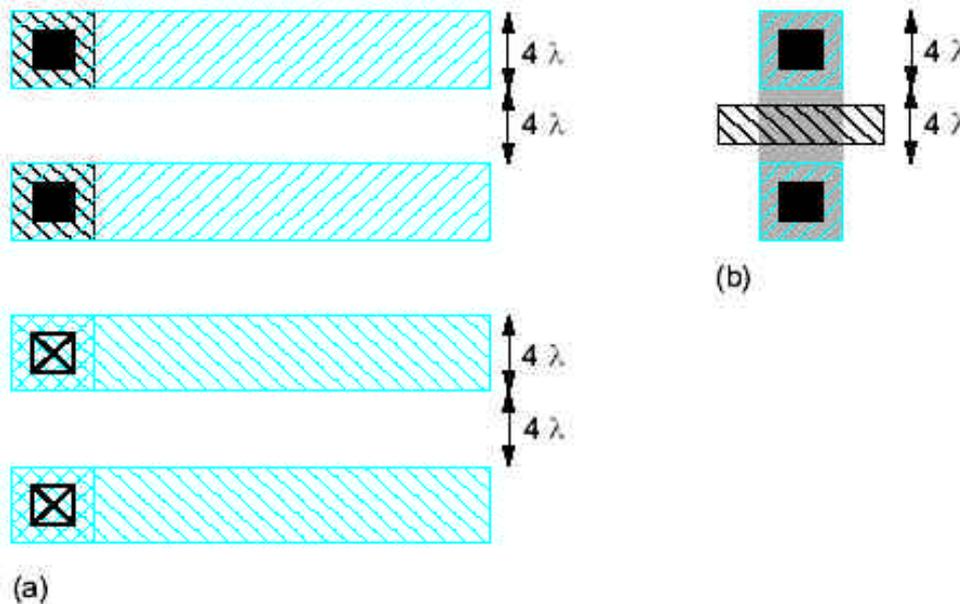- ❑ 32 λ by 40 λ
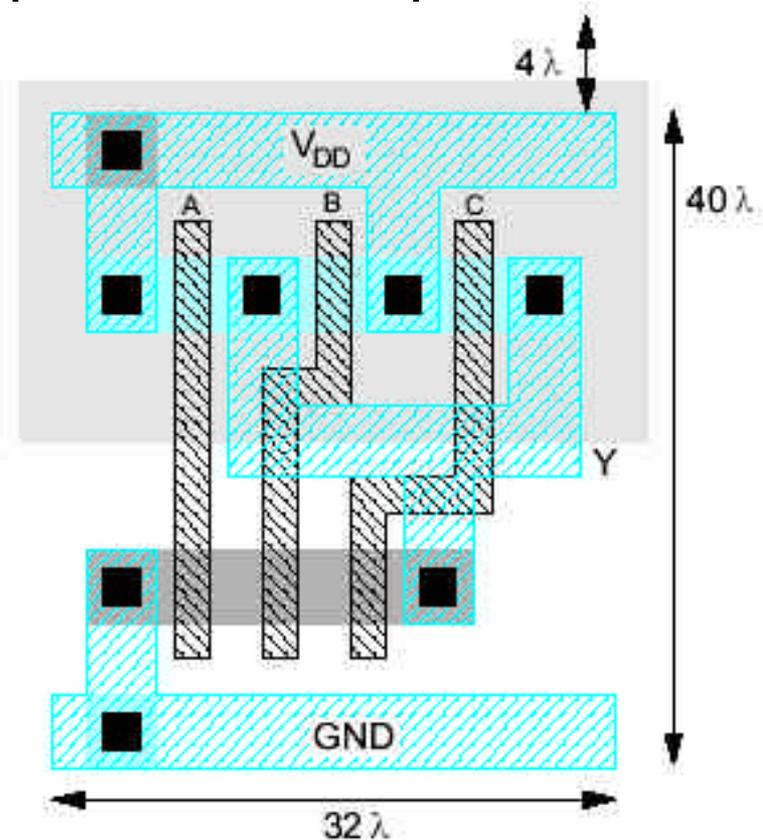
Stick Diagrams

❑ *Stick diagrams* help plan layout quickly
- – Need not be to scale
- – Draw with color pencils or dry-erase markers

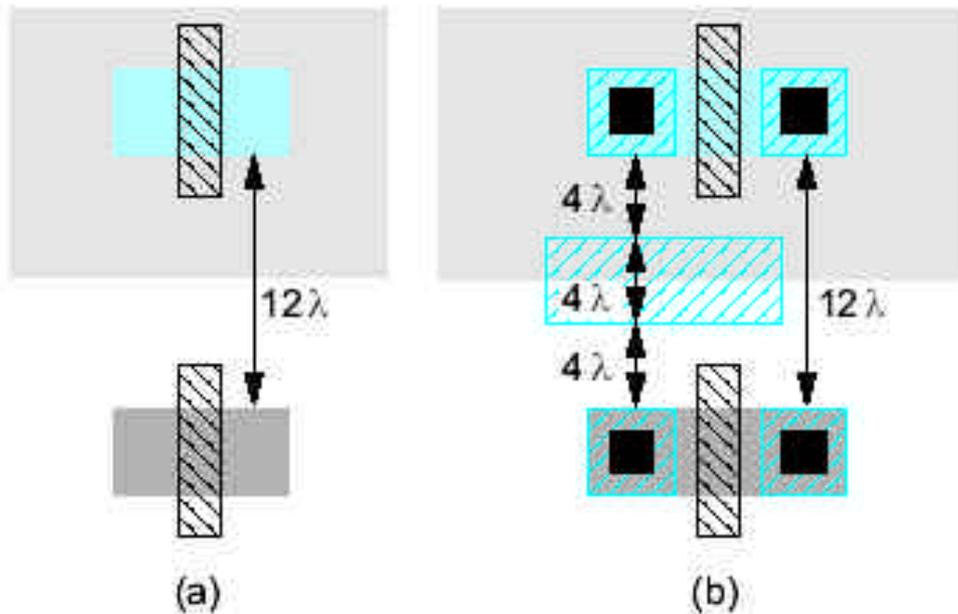# Wiring Tracks

- ❑ A *wiring track* is the space required for a wire
    - – 4 λ width, 4 λ spacing from neighbor = 8 λ pitch
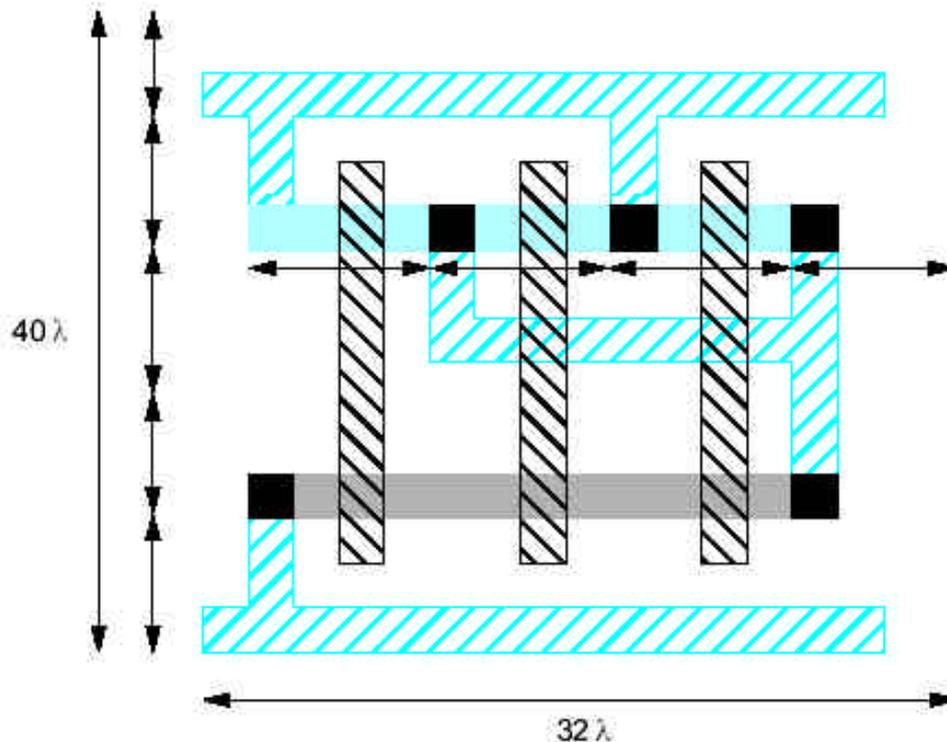- ❑ Transistors also consume one wiring track

Well spacing

❑ Wells must surround transistors by 6 λ

– Implies 12 λ between opposite transistor flavors

– Leaves room for one wire track

(a)   (b)

Area Estimation

❑ Estimate area by counting wiring tracks

– Multiply by 8 to express in λ

Example: O3AI

❑ Sketch a stick diagram for O3AI and estimate area

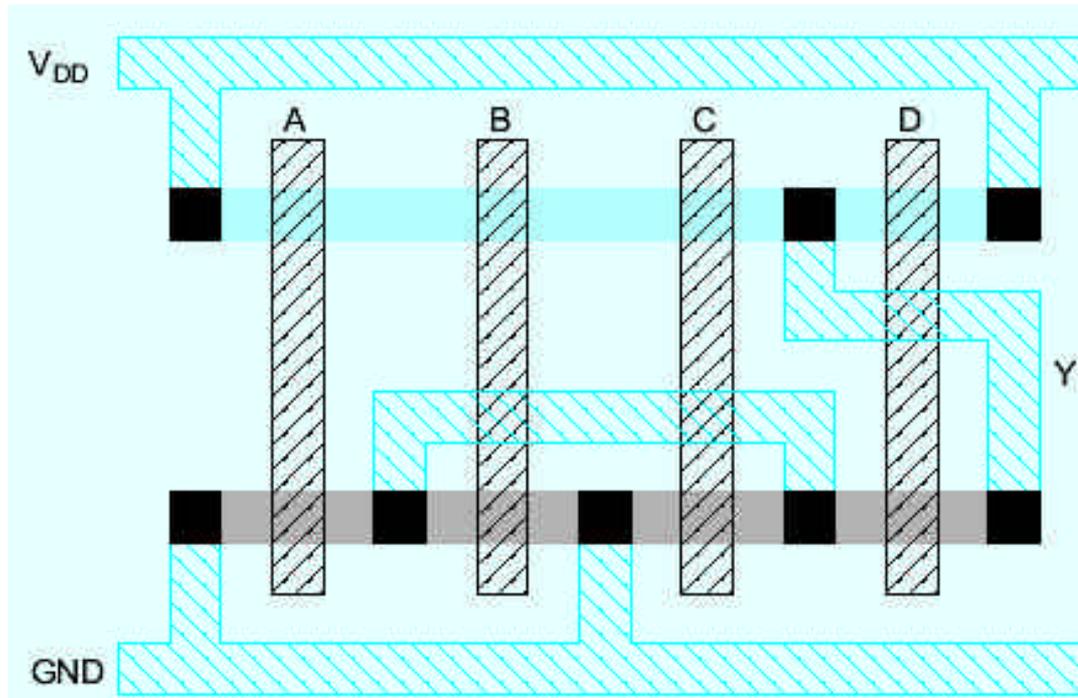- $Y = \overline{\left(A + B + C \right) \cdot D}$

Example: O3AI

❑ Sketch a stick diagram for O3AI and estimate area

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

Example: O3AI

❑ Sketch a stick diagram for O3AI and estimate area

$$- \quad Y = \overline{(A + B + C) \cdot D}$$