EECS 598 VLSI for Wireless Communication and Machine Learning

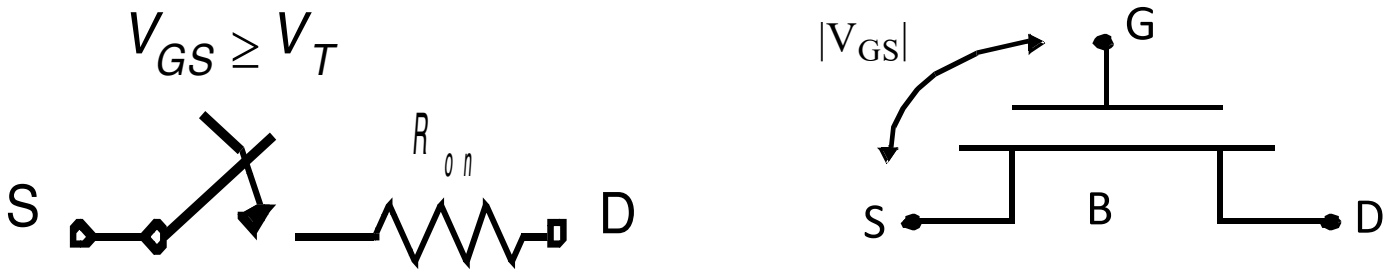
Digital Integrated Circuit Overview

Prof. Hun-Seok Kim

hunseok@umich.edu

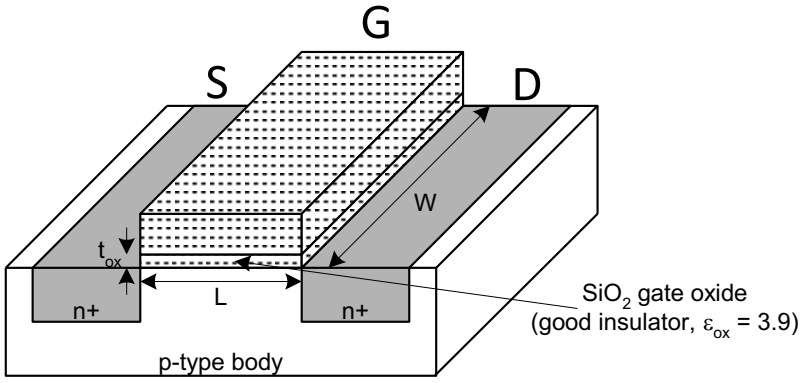
What is a Transistor?

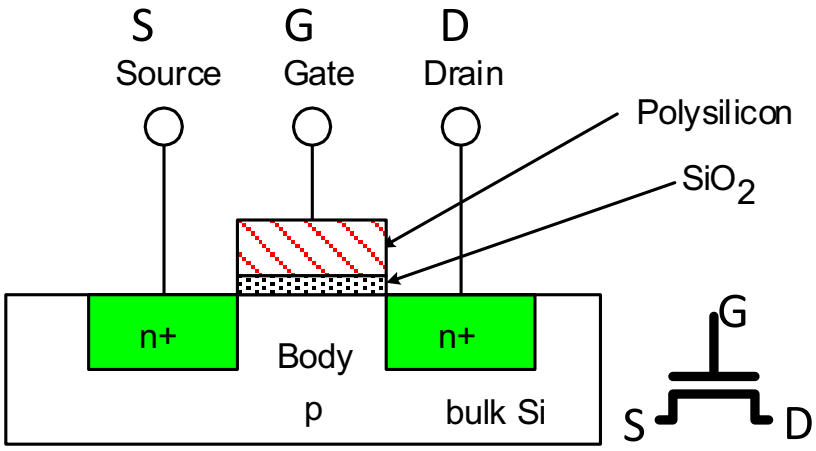
A Switch! ⟷ An MOS Transistor



MOS Transistor

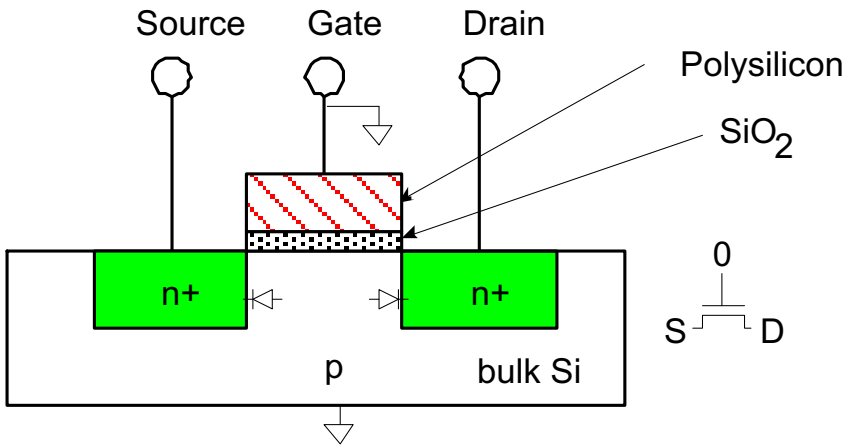
- Four terminals: gate, source, drain, body
- Gate – oxide – body stack looks like a capacitor
  - Gate and body are conductors
  - SiO_2 (oxide) is a very good insulator
  - Called metal – oxide – semiconductor (MOS) capacitor
  - Even though gate is often not made of metal





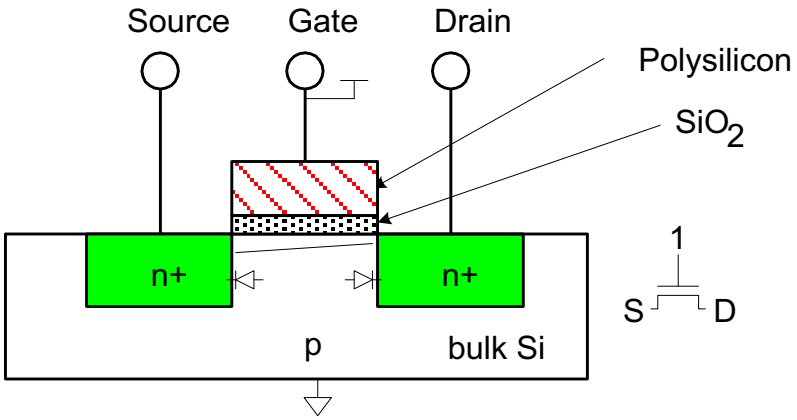
NMOS Operation

- Body is usually 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 cont.

- 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

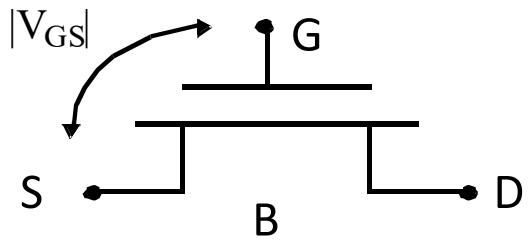


MOS FET IV characteristics (conductivity / resistance)

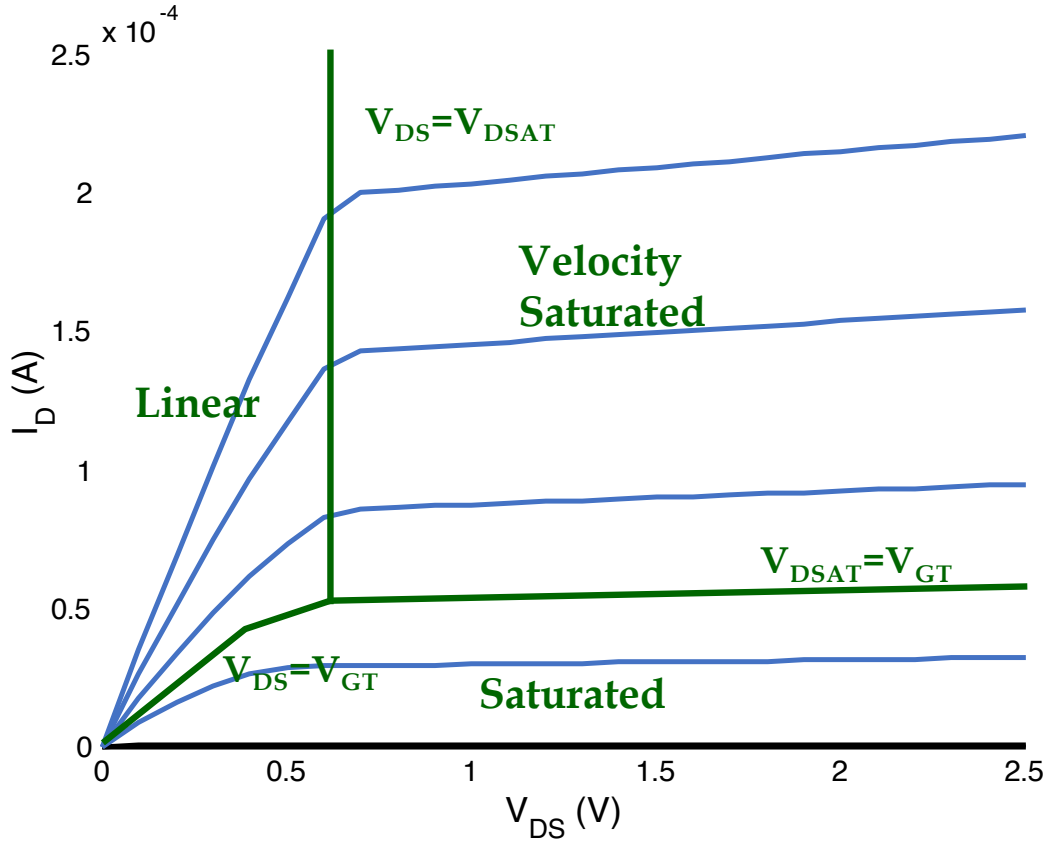
- Current I_D is a function of V_{GS}, V_{DS}, and V_T
- Transistor conductivity / resistance is controlled by terminal voltages

- Let
  - $V_{GT} = V_{GS} - V_T$
  - $V_{min} = \min(V_{GT}, V_{DS}, V_{DSAT})$
  - $V_T = V_{T0} + \gamma\left(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|}\right)$
- Then
  - $$I_{DS} = \begin{cases} 0 & V_{GT} \le 0 \\ k'\left(\frac{W}{L}\right)\left(V_{GT}V_{min} - \frac{V_{min}^2}{2}\right)(1 + \lambda V_{DS}) & V_{GT} > 0 \end{cases}$$

An NMOS Transistor

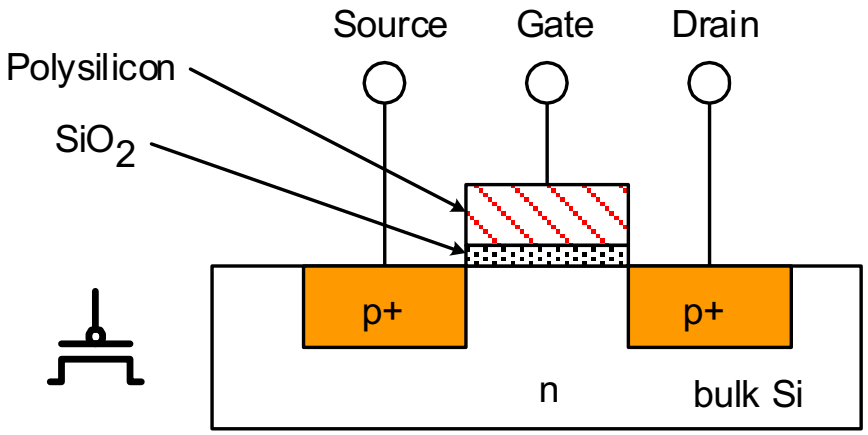


MOS FET IV characteristics (conductivity / resistance)



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



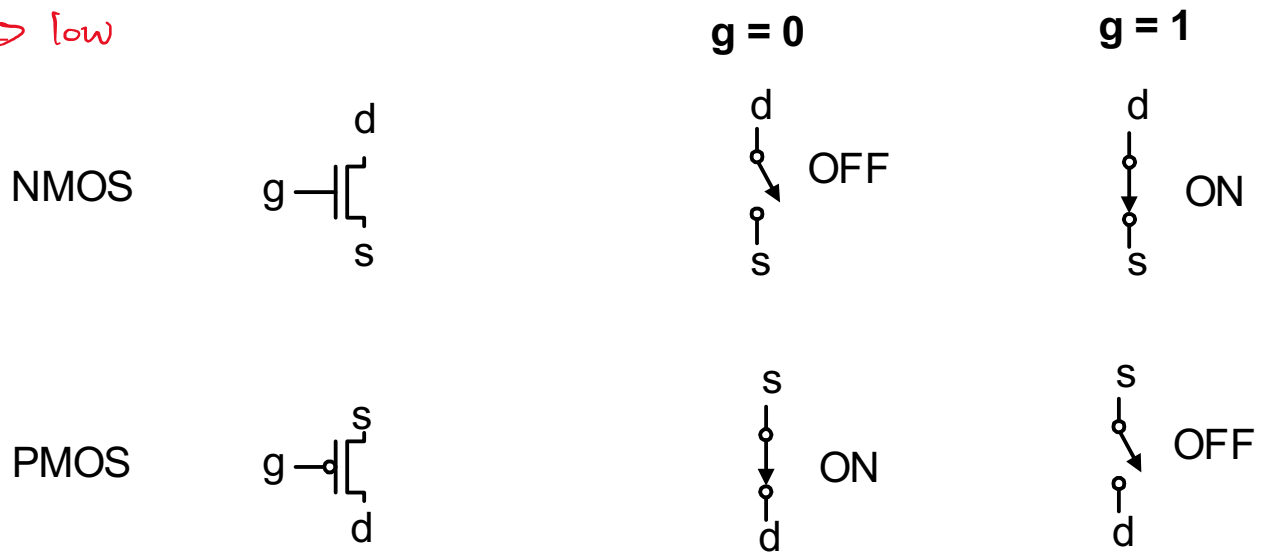
Transistors as Switches

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

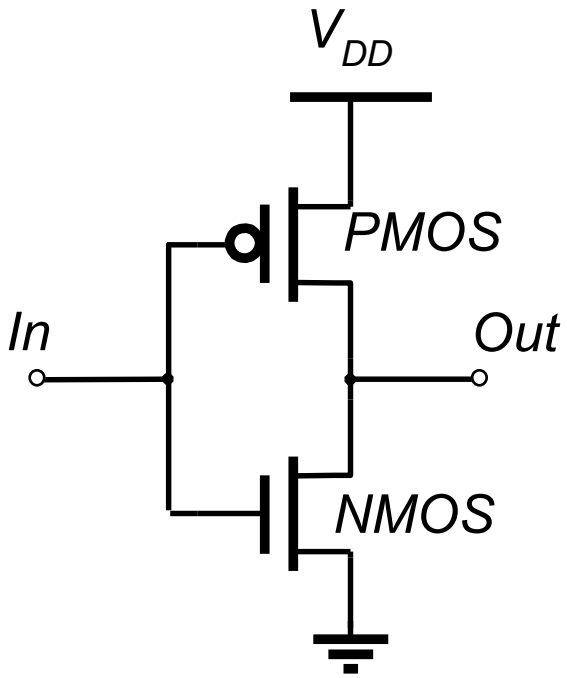
notation

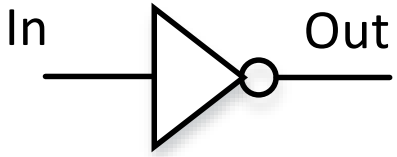
1 ⟺ V_{DD} ⟺ high

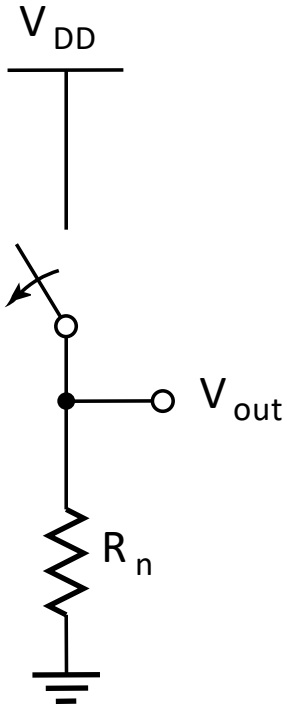
0 ⟺ GND ⟺ low



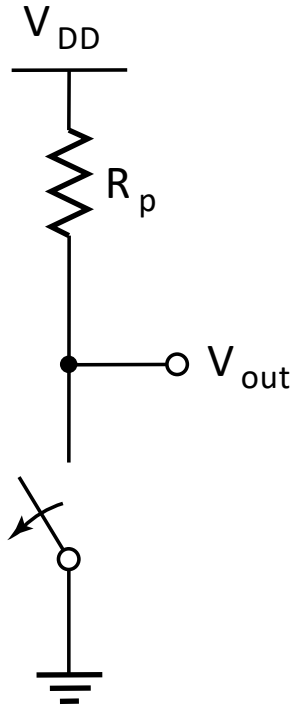
CMOS Inverter





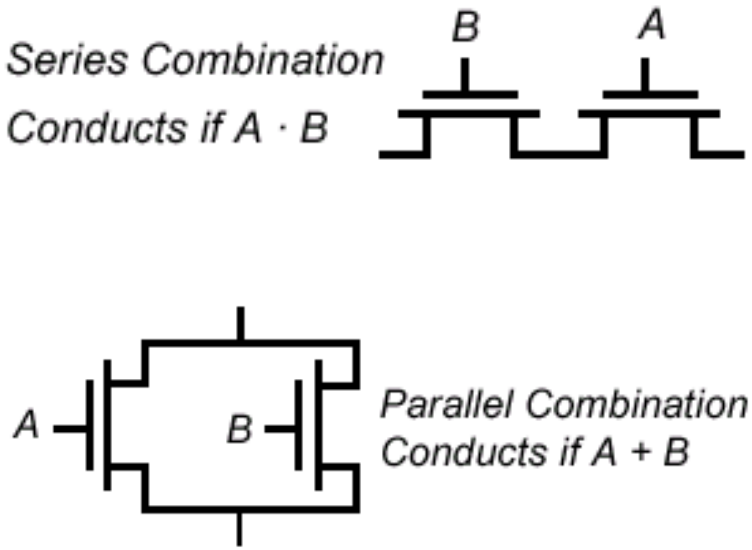


$V_{in} = V_{DD}$



$V_{in} = 0$

Two major functions: AND, OR

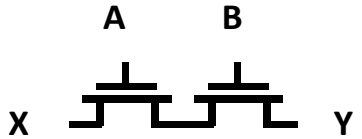


CMOS combinational gates are based on these two combinations of transistors (series & parallel)

NMOS Transistors in Series/Parallel Connection

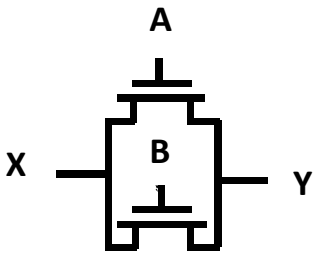
Transistors can be thought as a switch controlled by its gate signal

NMOS switch closes when switch control input is high



Y = X if A and B

AND



Y = X if A OR B

OR

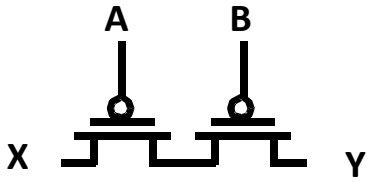
PMOS Transistors in Series/Parallel Connection

PMOS switch closes when switch control input is low

De Morgan

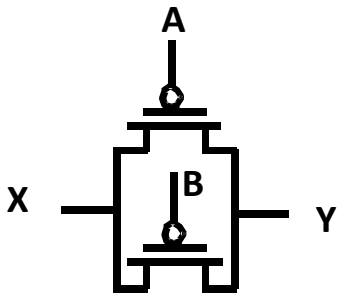
$\overline{A+B} = \bar{A}\,\bar{B}$

$\overline{AB} = \bar{A} + \bar{B}$



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

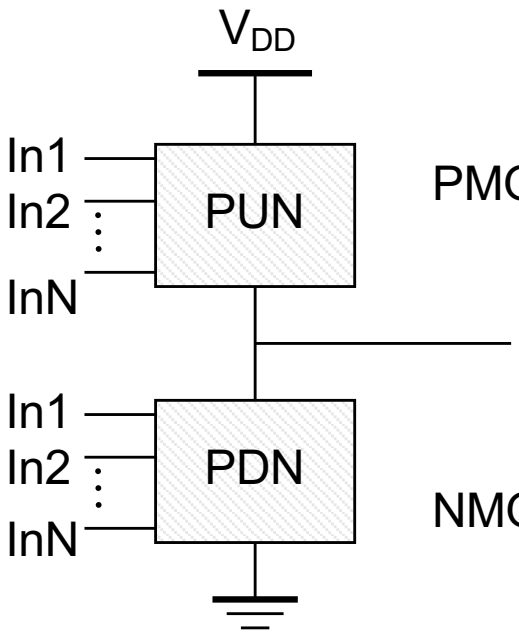
NOR



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

NAND

Static CMOS General Form



PMOS only

Pull-up network: make a connection from V_{dd} to F when F(In1,In2…)=1

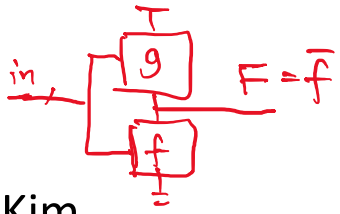
F(In1,In2,…InN)

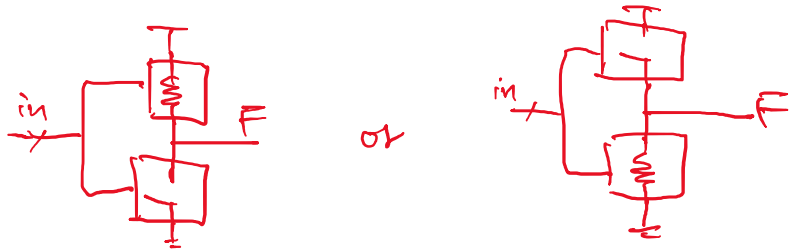
NMOS only

Pull-down network: make a connection from Ground to F when F(In1,In2…)=0

PUN and PDN are *dual* networks

$F(A,B,\cdots) = \overline{f(A,B,\cdots)} = g(\bar{A},\bar{B},\cdots)$





Complementary CMOS Logic Style Construction

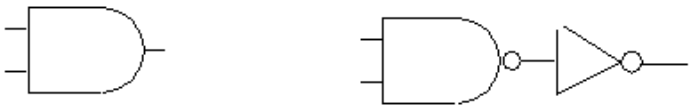
- PUN is the **dual** of the PDN

  Can be shown using DeMorgan's Theorem

$$\overline{A+B} = \bar{A}\bar{B}$$

$$\overline{AB} = \bar{A}+\bar{B}$$

- **The complementary gate is inverting**

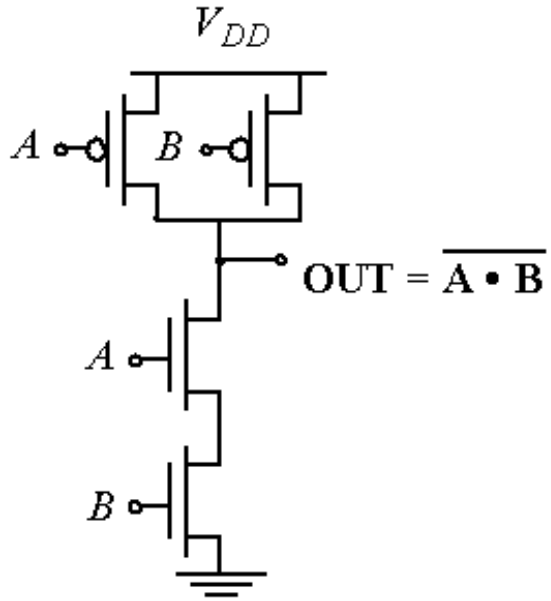


AND = NAND + INV

Example Gate: NAND (most common gate in digital circuits)

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

Truth Table of a 2 input NAND gate



PDN: Connects OUT to ground when A●B=1

PUN: Connects OUT to V_{dd} when $\overline{A} + \overline{B} = 1$

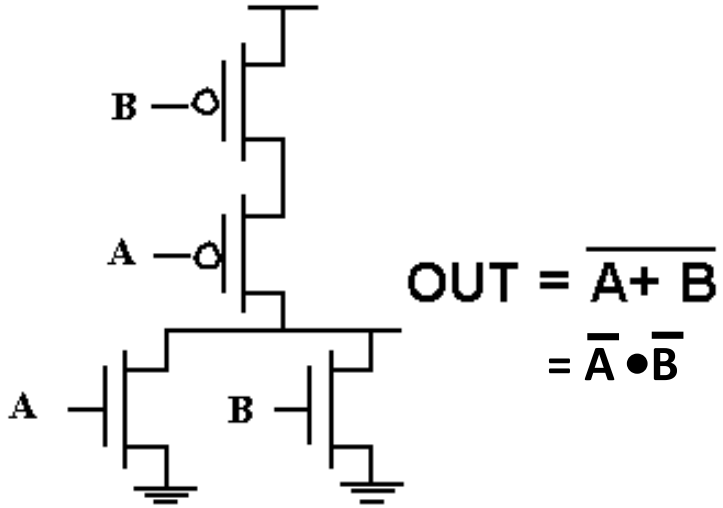
So OUT = Complement of PDN function

Also OUT = PUN function with each input inverted

Example Gate: NOR

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

Truth Table of a 2 input NOR gate

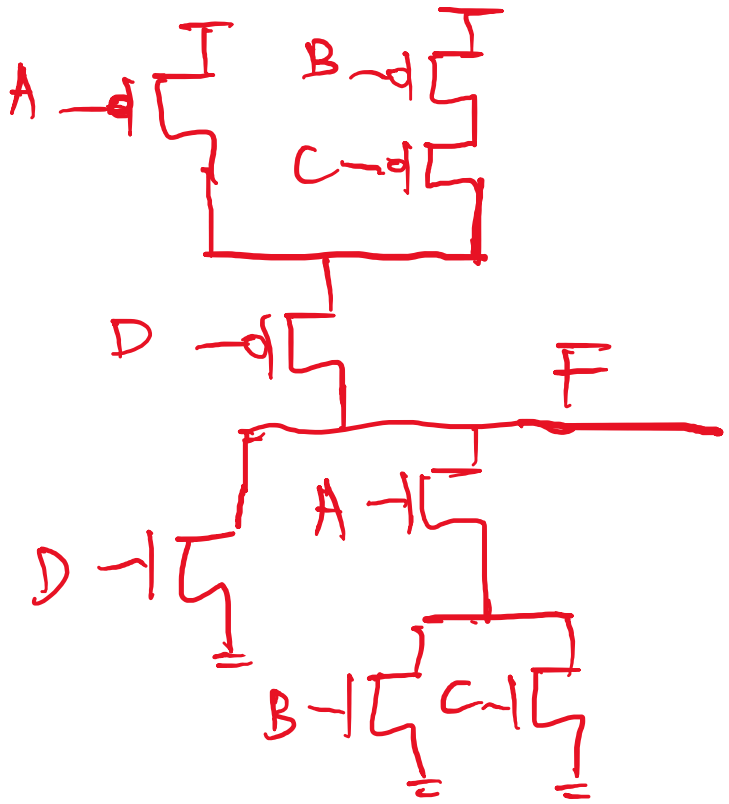


N-input gates require 2N inputs for static CMOS

Building Complex Gates

$F = \overline{D + A(B+C)}$ $\quad f = D + A(B+C)$

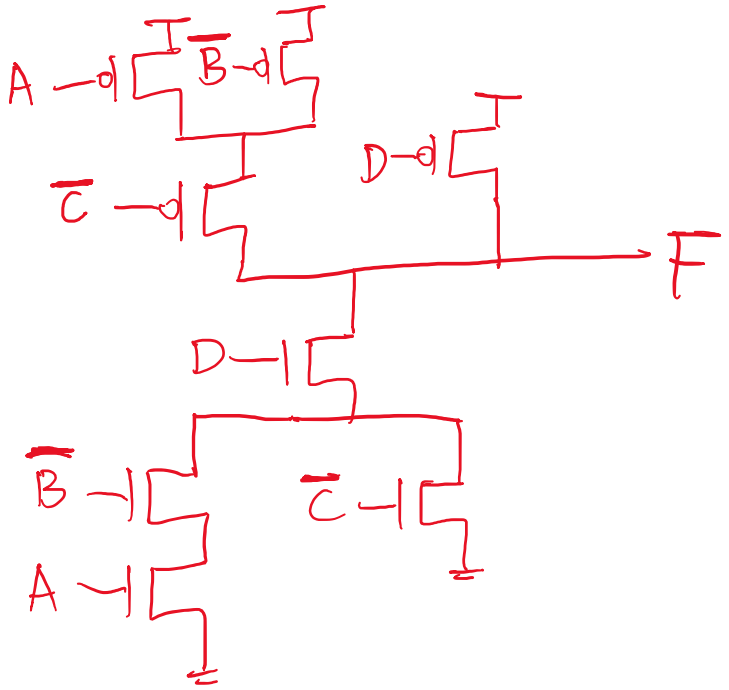
$g = \bar{D}(\bar{A} + \bar{B}\bar{C})$



$$F = (\bar{A}+B)C+\bar{D} \quad = \overline{\overline{(\bar{A}+B)C+\bar{D}}} \quad = \overline{(A\cdot\bar{B}+\bar{C})D} \quad = (\bar{A}+B)C+\bar{D}$$

$$f = (A\cdot\bar{B}+\bar{C})D$$

$$g = (\bar{A}+B)C+\bar{D}$$



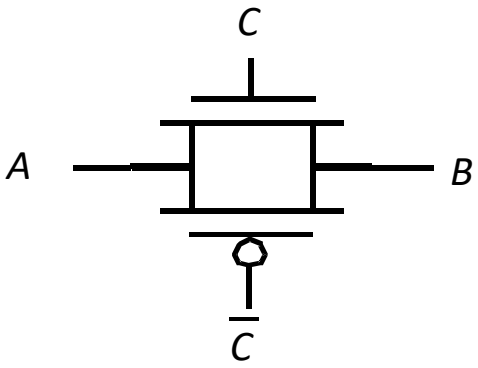
note: inputs are $A, \bar{B}, \bar{C}, D$ for both PDN & PUN

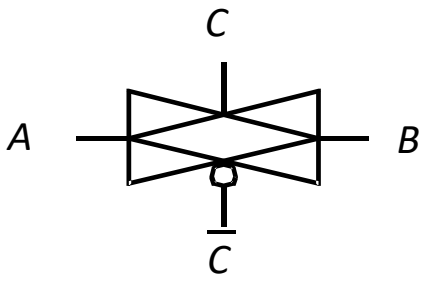
inputs must be the same for PDN, PUN of a CMOS gate

Static Properties of CMOS Gates (same as inverters)

- Full rail-to-rail swing (high noise margins)
- Always a connection to V_{dd} or ground in steady state (low output impedance)
- No direct path between V_{dd} and ground → No static power dissipation (ignoring leakage currents)
- Gates are inverting → if we want an AND gate, we need to add an inverter after a NAND gate

Transmission Gate



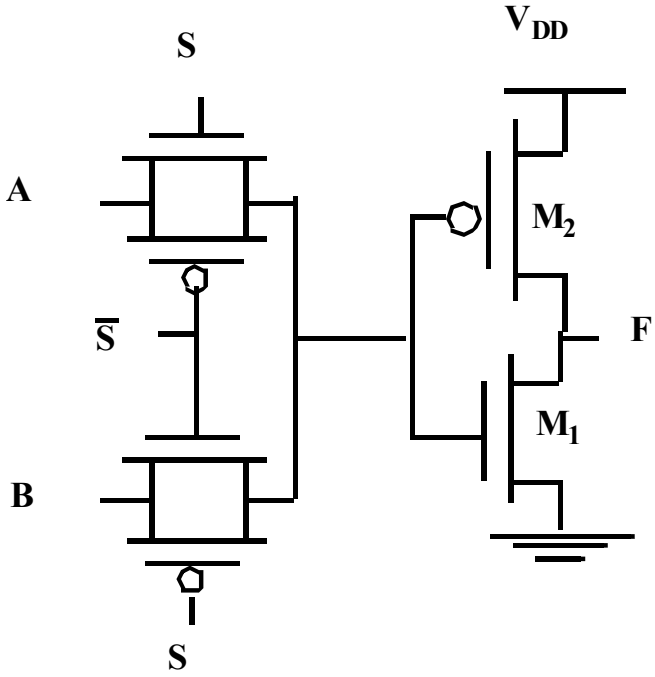


Multiplexers are commonly used in digital design

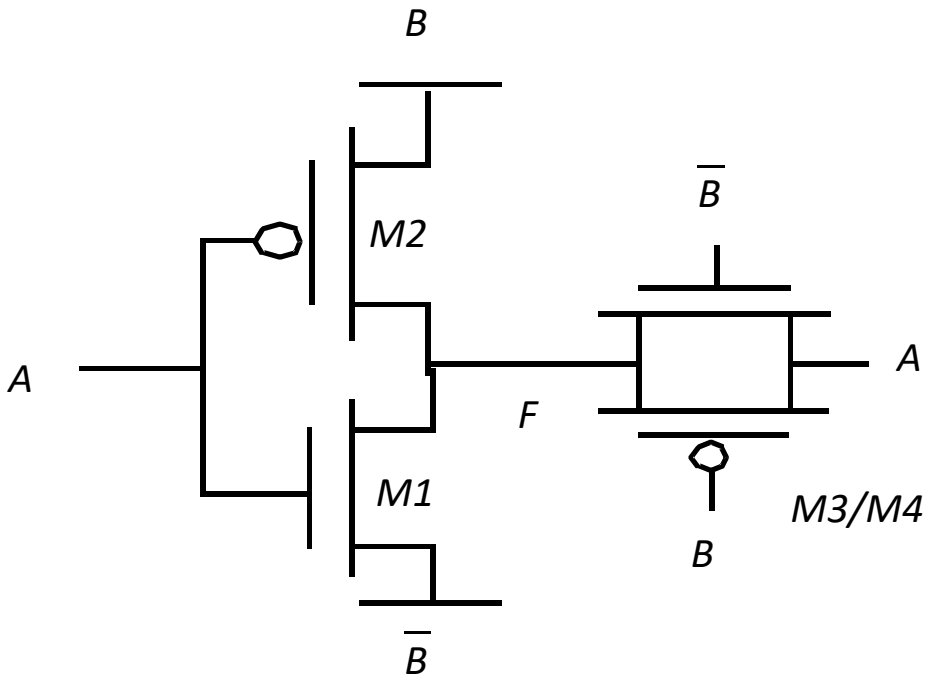
Function: Select one of N inputs based on a controlling signal (S)

Here N = 2

# of controlling signals = $\log_2 N$



Transmission Gate XOR



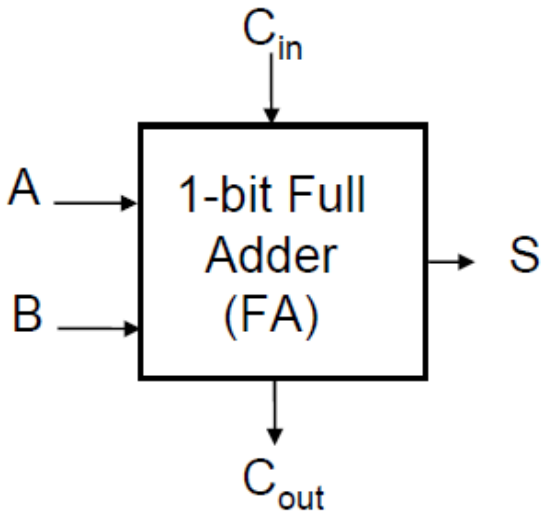
XOR → exclusive OR

Same inputs: F = 0
Different inputs: F = 1

Fairly compact compared to static CMOS XOR gates (8 transistors)

Full Adder

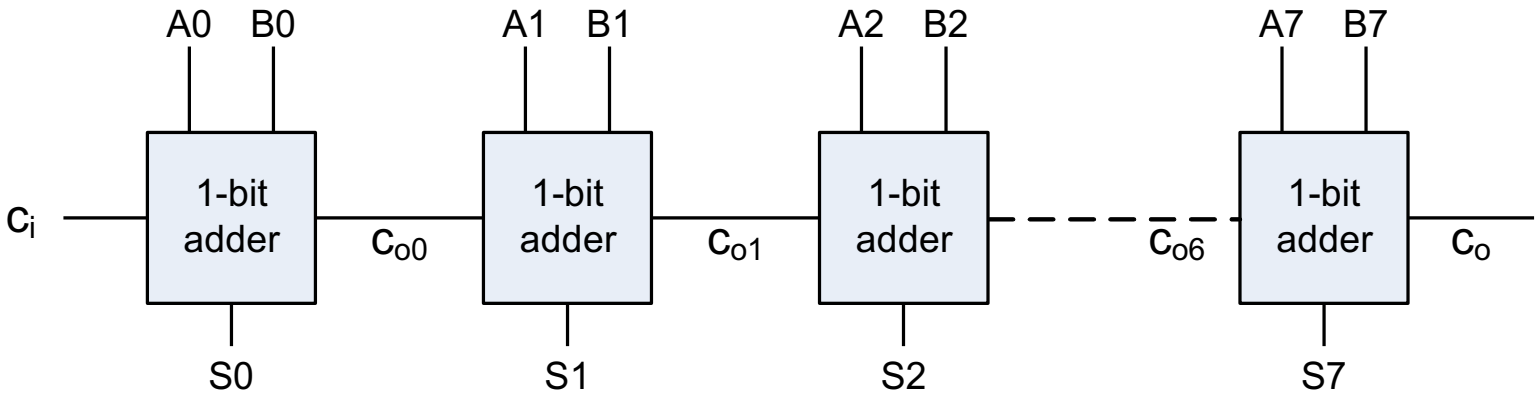
- A[n-1:0] + B[n-1:0] = S[n-1:0]



A	B	C_{in}	C_{out}	S	carry status
0	0	0	0	0	kill
0	0	1	0	1	kill
0	1	0	0	1	propagate
0	1	1	1	0	propagate
1	0	0	0	1	propagate
1	0	1	1	0	propagate
1	1	0	1	0	generate
1	1	1	1	1	generate

- $C_o = AB + BC_i + AC_i = AB + (A + B)C_i$
- $S = A \oplus B \oplus C_i = ABC_i + \overline{C_o}(A + B + C_i)$
- $G = AB,\ K = \overline{A}\,\overline{B},\ P = A \oplus B$

Ripple Carry Adder



Worst case delay linear with the number of bits

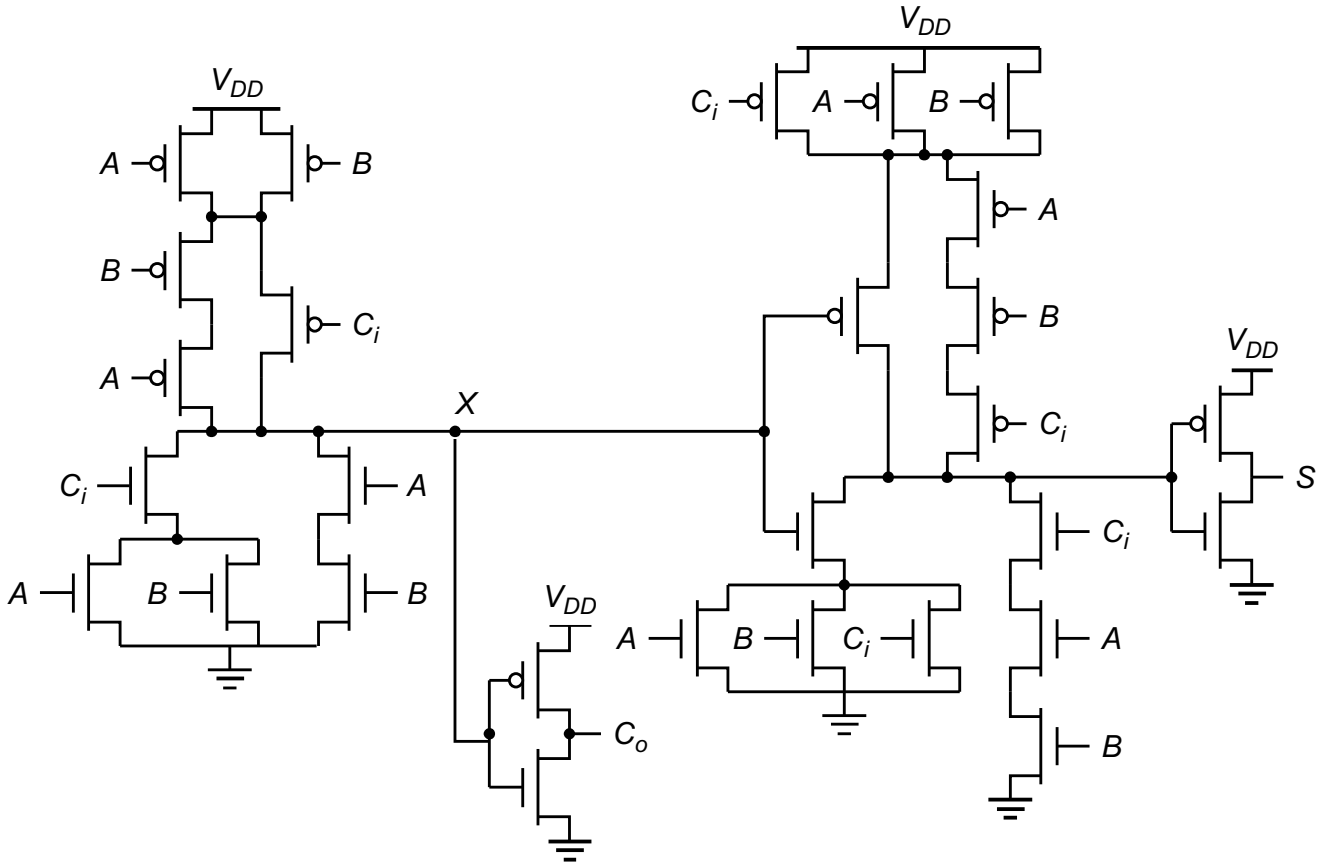
$$t_d = O(N)$$

$$t_{adder} = (N\text{-}1)t_{carry} + t_{sum}$$

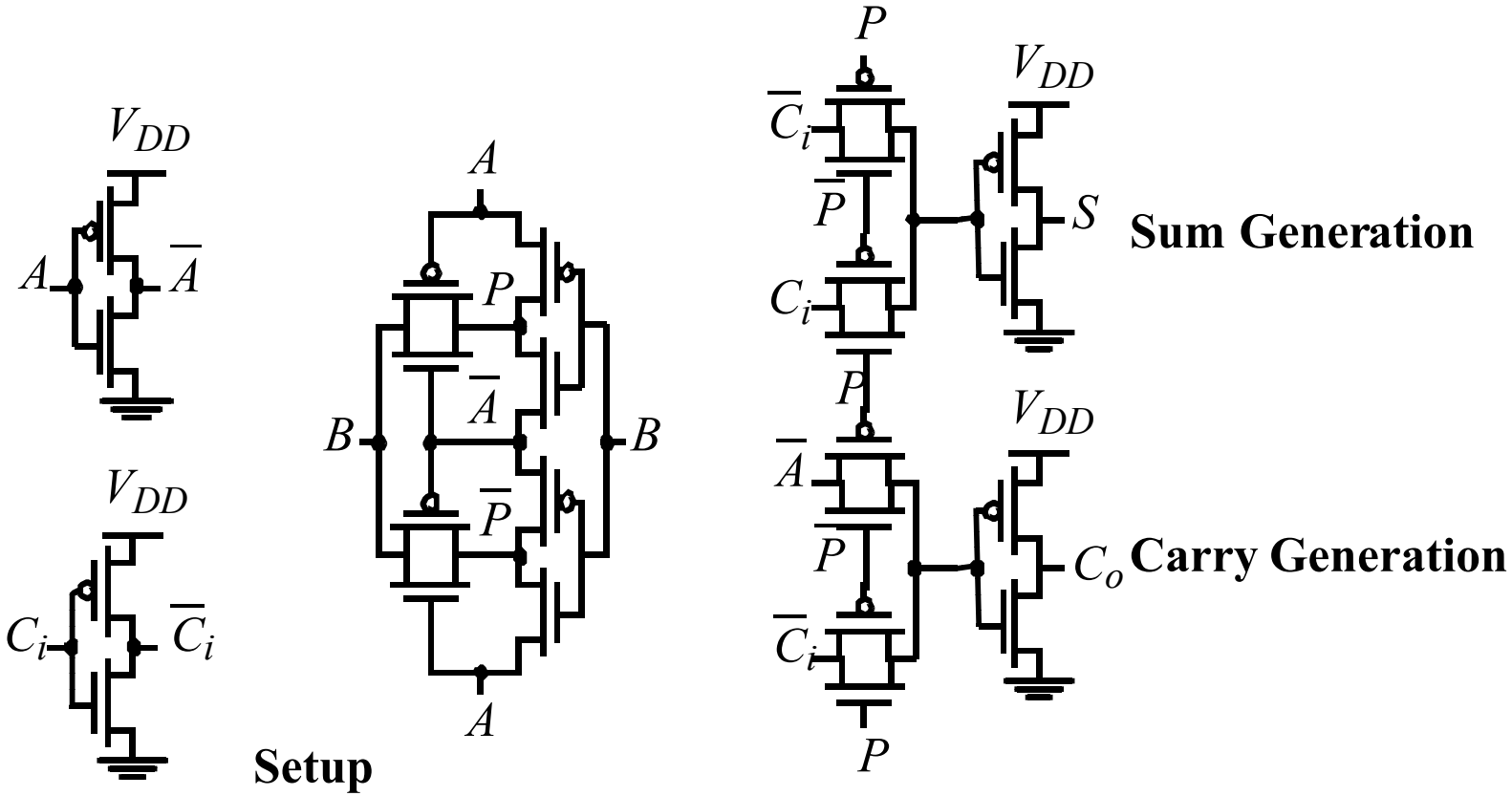
Goal: Make the fastest possible carry path circuit

Static CMOS Full Adder

- $C_o = AB + BC_i + AC_i = AB + (A + B)C_i$
- $S = A \oplus B \oplus C_i = ABC_i + \overline{C_o}(A + B + C_i)$
- 28 transistors

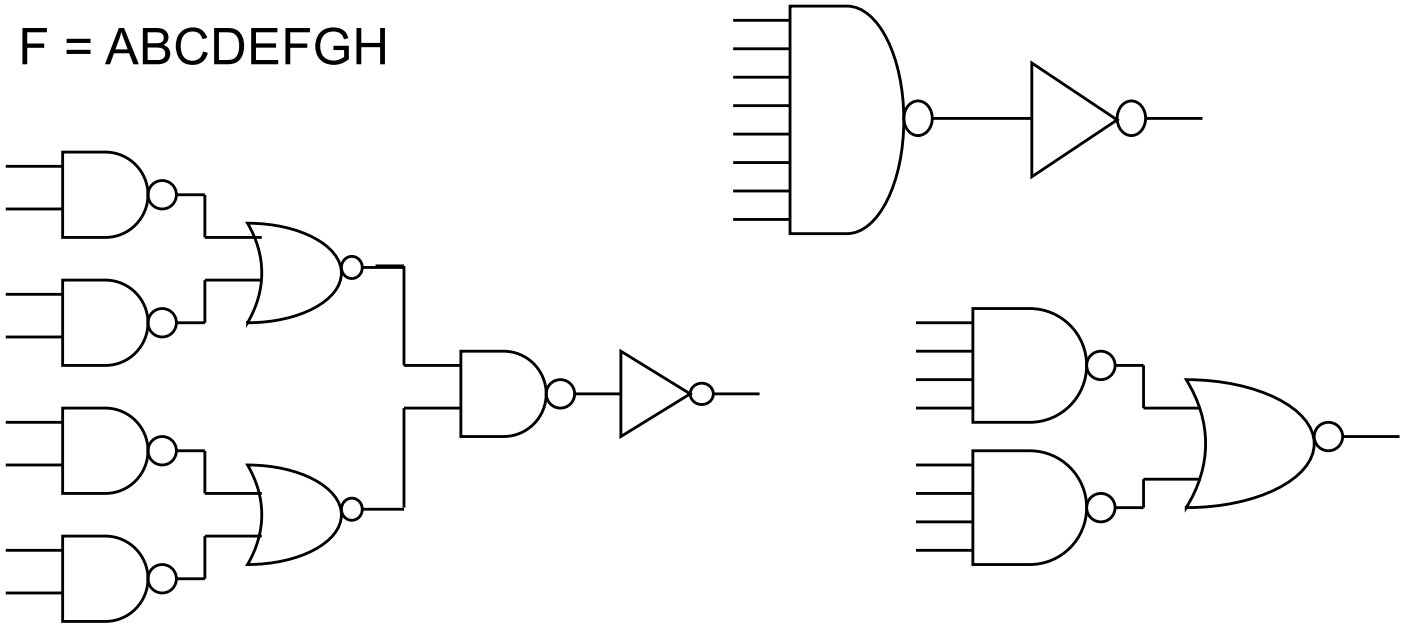


Transmission Gate Full Adder



Synthesis using cell library

- Hierarchical logic structures



Synthesis using standard ‘cell library’

Key VLSI design goals

What are key design metrics for VLSI for wireless communication or machine learning or any other applications?

- Functionality
- Performance (speed, delay)
- Power / Energy
- Area

What factors determine the performance (speed, delay), power(energy), and area of a circuit?

How are these metrics related?

MOS FET IV characteristics (conductivity / resistance)

- **Current I_D is a function of V_{GS}, V_{DS}, and V_T**
- **Transistor conductivity / resistance is controlled by terminal voltages and transistor size**

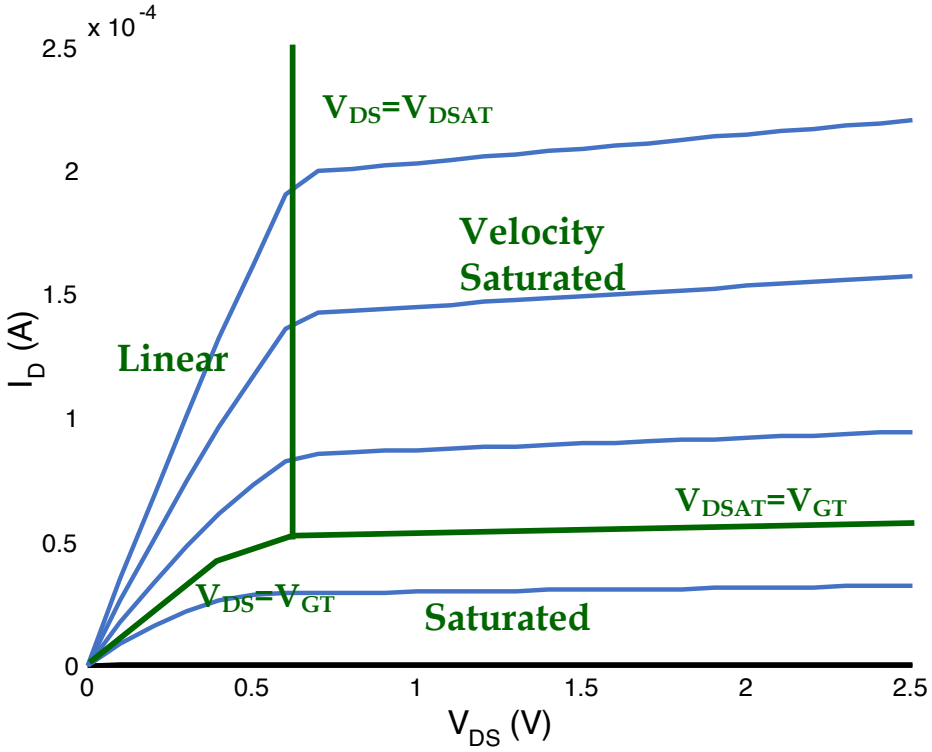
Let

- $V_{GT} = V_{GS} - V_T$
- $V_{min} = \min(V_{GT}, V_{DS}, V_{DSAT})$
- $V_T = V_{T0} + \gamma\left(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|}\right)$

Then

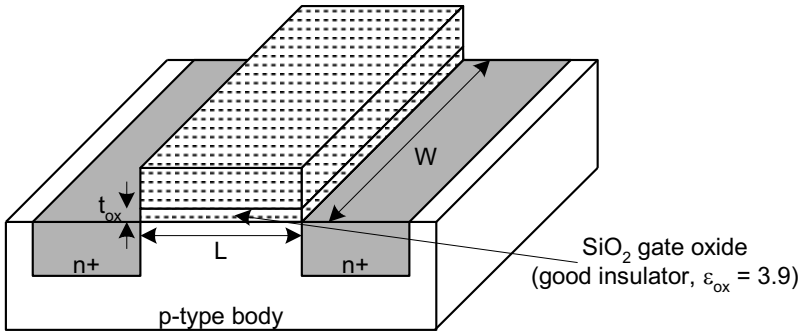
- $I_{DS} =$

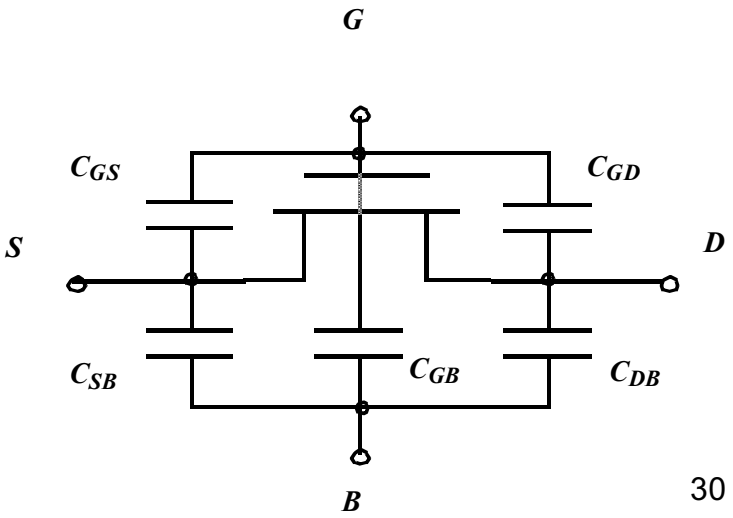
$$\begin{cases} 0 & V_{GT} \le 0 \\ k'\left(\frac{W}{L}\right)\left(V_{GT}V_{min} - \frac{V_{min}^2}{2}\right)(1 + \lambda V_{DS}) & V_{GT} > 0 \end{cases}$$



Capacitance

- Any two conductors separated by an insulator have capacitance
- MOS transistor gate to channel capacitor is very important
  - Creates channel charge necessary for operation
- Source and drain have capacitance to body
  - Across reverse-biased diodes
  - Called junction or diffusion capacitance; associated with source/drain diffusion and p-n junction
- MOS transistor cap is proportional to the transistor width (W) given length (L): C $\propto$ WL

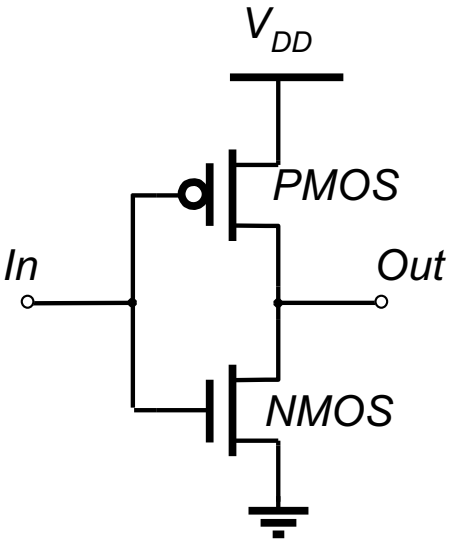


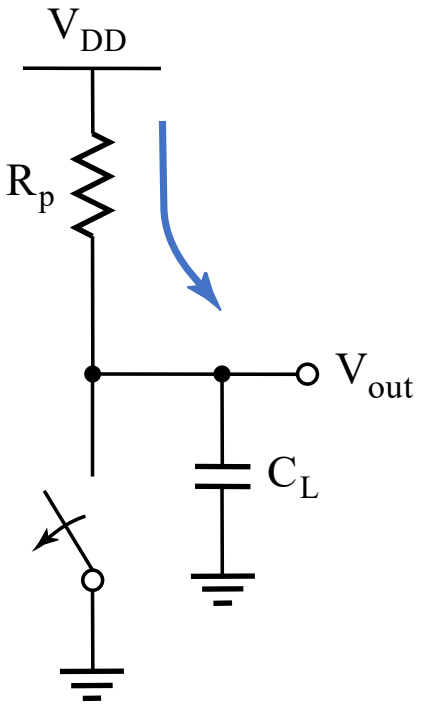


CMOS Inverter: Transient Response

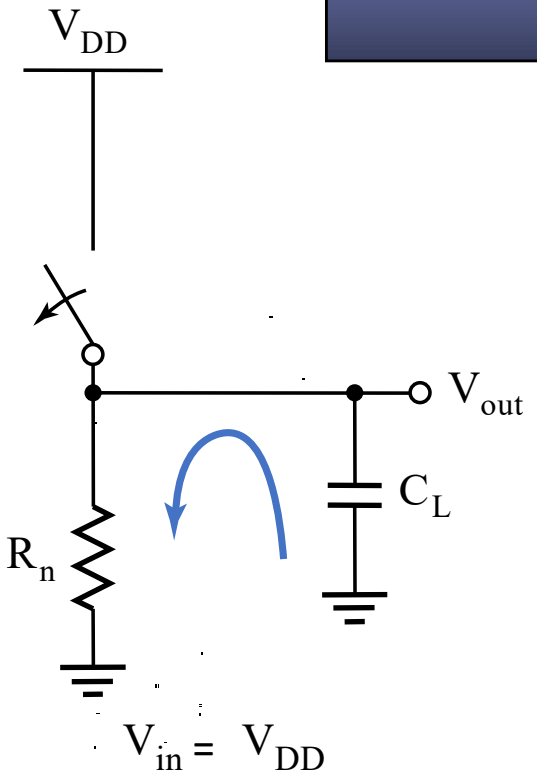
$t_p = f(R_{on}.C_L)$

$= 0.69\ R_{on} C_L$





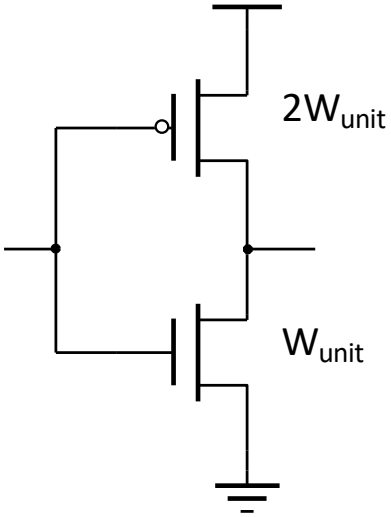
(a) Low-to-high



(b) High-to-low

A ‘unit’ size inverter

- A ‘unit’ size inverter for 250nm technology
  - $L_N=L_P=L_{unit}=0.25\mu m$
  - $W_N=L_{unit}$, $W_P=2L_{unit}$
  - NMOS internal cap $C_{dn}=C_{unit}$
  - PMOS internal cap $C_{dp}=2C_{unit}$
  - Internal Cap $C_{int}= C_{dn}+C_{dp}=3C_{unit}$
  - Equivalent R: $R_n=R_p=R_{unit}$
    - Same pull-up and pull-down currents

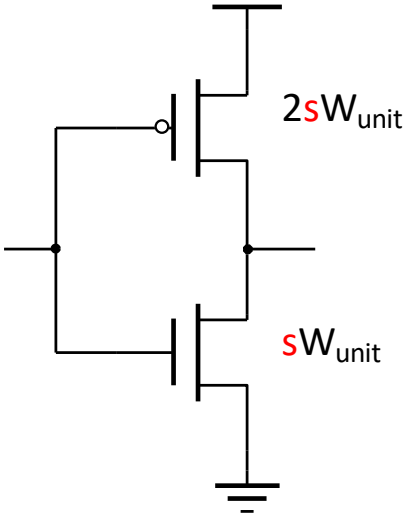


$$t_{pHL} = 0.69R_{unit}(C_{int}+C_L) = 0.69R_{unit}(3C_{unit}+C_L)$$

$$t_{pLH} = 0.69R_{unit}(C_{int}+C_L) = 0.69R_{unit}(3C_{unit}+C_L)$$

Inverter Scaling

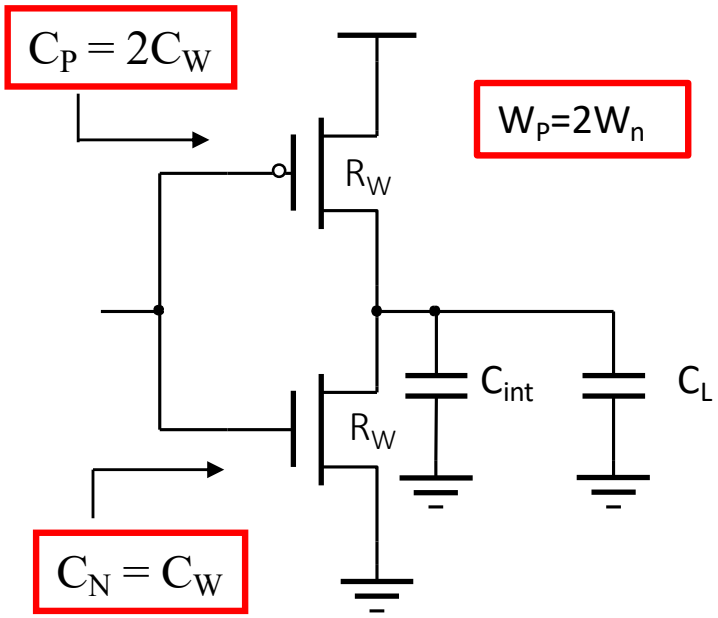
- Scaling factor s
  - $L_N=L_P=L_{unit}=0.25\mu m$
  - $W_N=sL_{unit}$, $W_P=2sL_{unit}$
  - NMOS internal cap $C_{dn}=sC_{unit}$
  - PMOS internal cap $C_{dp}=2sC_{unit}$
  - Internal cap $C_{int}= C_{dn}+C_{dp}=3sC_{unit}$
  - Equivalent R: $R_n=R_p=R_{unit}/s$

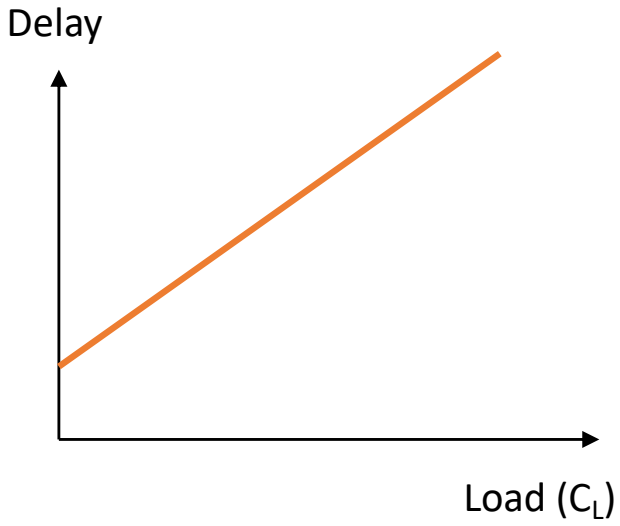


$$t_{pHL} = 0.69R_{unit}/s\ (sC_{int}+C_L) = 0.69R_{unit}/s\ (3sC_{unit}+C_L)$$

$$t_{pLH} = 0.69R_{unit}/s\ (sC_{int}+C_L) = 0.69R_{unit}/s\ (3sC_{unit}+C_L)$$

Inverter with Load

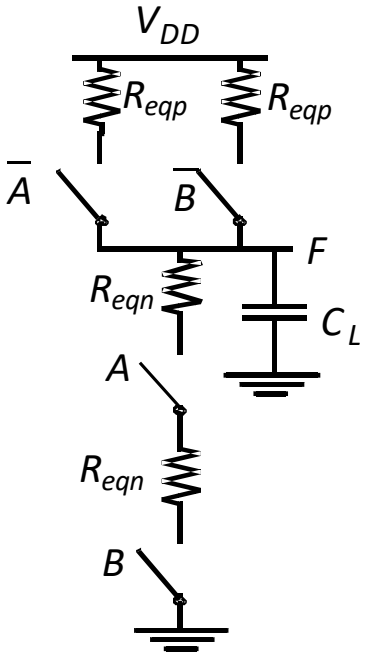




Delay = $0.69R_W(C_{int} + C_L) = 0.69R_WC_{int}+0.69R_WC_L = 0.69R_WC_{int}(1+ C_L /C_{int})$
= Delay (Internal) + Delay (Load)

What's the desired size of this inverter to minimize the delay given C_L?
What about the size of the previous stage that drives the input of this inverter?

Analysis of Propagation Delay



2-input NAND

1. Determine "Worst Case Input" transition (Delay depends on input values)

2. Example: t_{pLH} for 2-input NAND

- **Worst case when only ONE PMOS Pulls up the output node**
- **For 2 PMOS devices in parallel, the resistance is lower**

$$t_{pLH} = 0.69*0.5*R_{eqp}C_L$$

3. Example: t_{pHL} for 2-input NAND

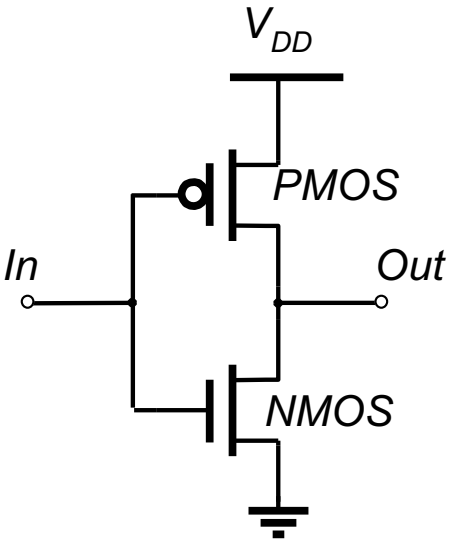
- **Worst case : TWO NMOS in series**

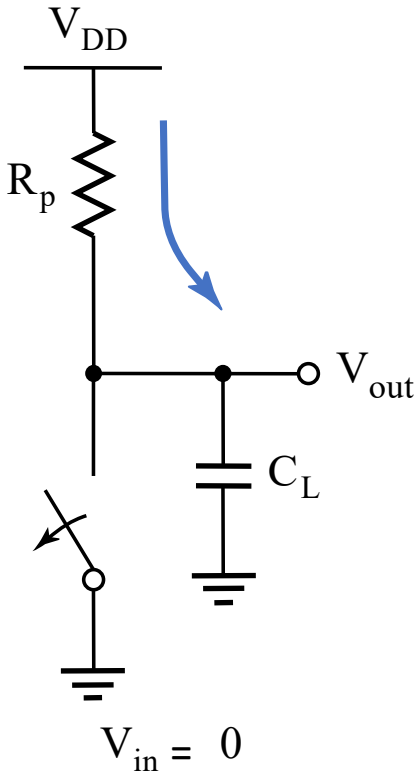
$$t_{pHL} = 0.69(2R_{eqn})C_L$$

Power Components in CMOS

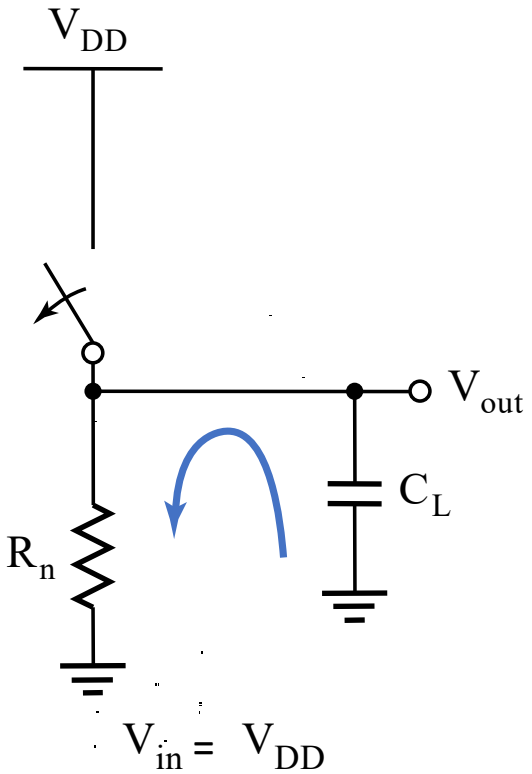
- Dynamic power consumption (dominant)
  - Charging and discharging capacitances through devices
- Short-circuit power (manageable)
  - Direct path between V_{dd} and ground during switching transients
- Static power (exponentially increasing)
  - Leakage currents exist and consume power when not switching

CMOS Inverter: Energy Consumption



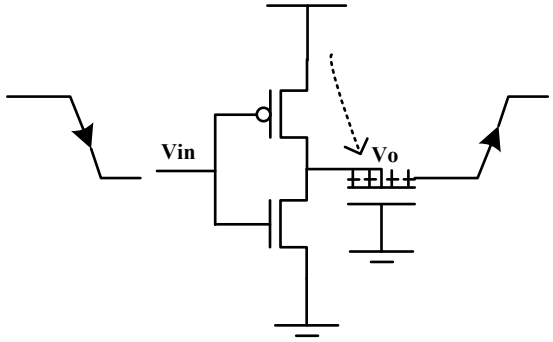


(a) Low-to-high



(b) High-to-low

Dynamic Power Consumption (1/2)



- Input 1→0
  - –Energy drawn from power supply:

$$E_{supply} = \int_0^\infty P(t)\cdot dt = \int_0^\infty V_{dd} i(t)\cdot dt = \int_0^{V_{dd}} V_{dd} C\cdot \frac{dV_O}{dt}\cdot dt = V_{dd} C\cdot \int_0^{V_{dd}} dV_O = CV_{dd}^{\ 2}$$

  - – Energy consumed by PMOS:

$$E_{PMOS} = \int_0^\infty P(t)\cdot dt = \int_0^\infty (V_{dd} - V_O)\cdot i_p(t)\cdot dt = C\cdot \int_0^{V_{dd}} (V_{dd} - V_O)\cdot dV_O = \frac{1}{2} CV_{dd}^{\ 2}$$

  - – Energy in the capacitor:

$$E_{CAP} = \int_0^\infty V_O \cdot i_C(t)\cdot dt = C\cdot \int_0^\infty V_O \cdot \frac{dV_O}{dt}\cdot dt = C\cdot \int_0^{V_{dd}} V_O \cdot dV_O = \frac{1}{2} CV_{dd}^{\ 2}$$

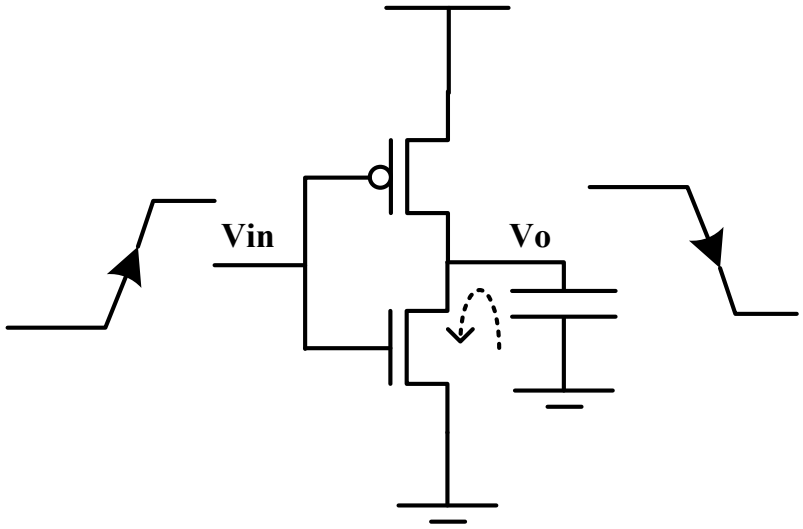
  - – Power:

$$\boxed{P = f\cdot E_{PMOS} = \frac{1}{2} fCV_{dd}^{\ 2}}$$

Dynamic Power Consumption (2/2)

- Input 0→1
  - Energy drawn from supply: 0
  - Energy consumed by NMOS equals the energy stored on capacitance:

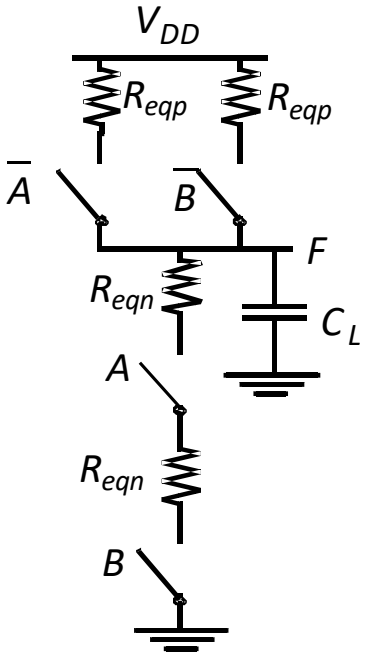
$$E_{NMOS} = \int V_O i(t) \cdot dt = \frac{1}{2} C V_{dd}^{\ 2}$$



  - Power:

$$P = f \cdot E_{NMOS} = \frac{1}{2} C V_{dd}^{\ 2} f$$

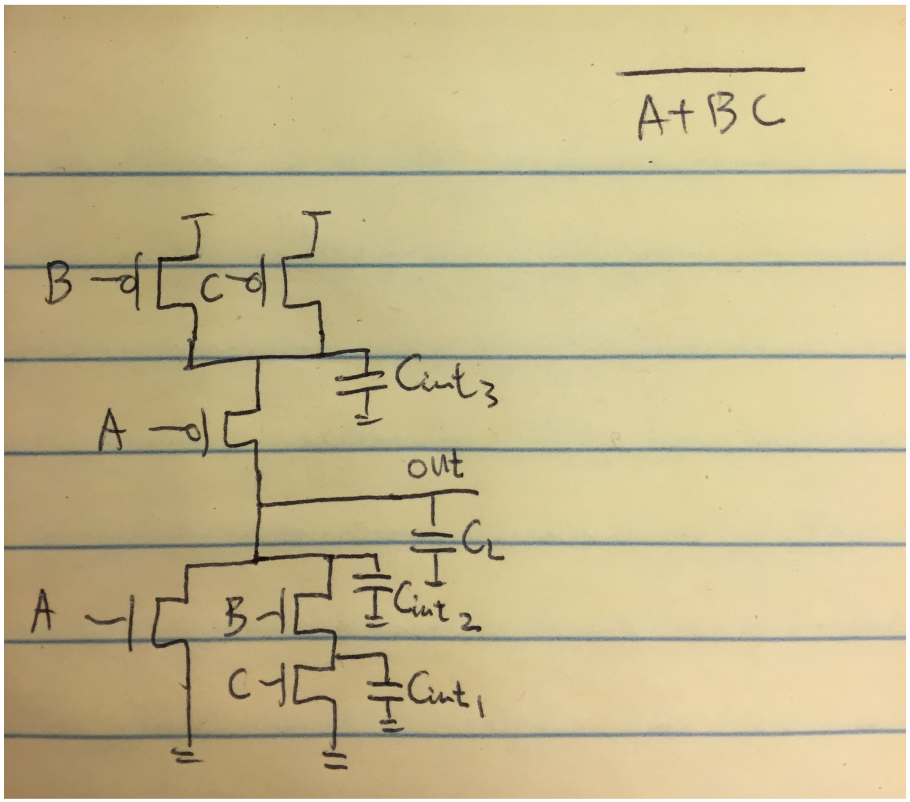
CMOS Gate Energy / Power



2-input NAND

- Energy consumption for output 0 → 1
- Energy consumption for output 1 → 0

CMOS Gate Energy Consumption



Energy vs. Power

Each rising transition on C_L requires $C_L * V_{dd}^2$ of energy BUT 1/2 of this energy is lost (to heat) while the other half is *stored* on the capacitor

For every transition, $C * V_{dd}^2/2$ of energy is **consumed**

For every period (both a L→H and H→L transition on C_L), $C * V_{dd}^2$ of energy is consumed from the supply

Then, power = rate of energy consumption so:

Energy = $P_{dyn}\, T_{sw} = C_L * V_{dd}^2$

→ $P_{dyn} = C_L * V_{dd}^2 * f_{sw}$

Where f_{sw} is the frequency with which C_L switches (L→H + H→L)

Switching Activity, α_{sw}

Let $f_{sw} = \alpha_{sw} * f_{clock}$ since we usually know clock frequency of a design (e.g. 3.6 GHz Core i7)

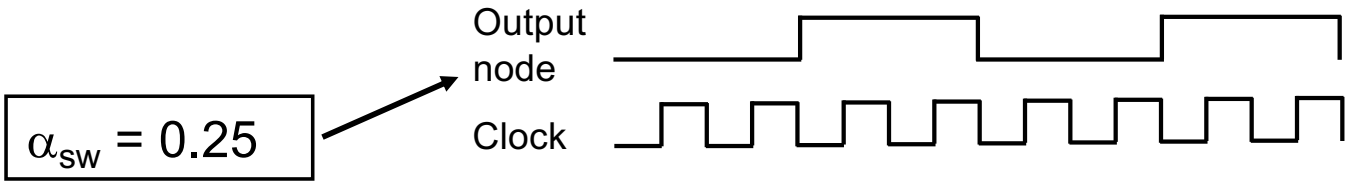
$$\mathbf{0 < \alpha_{sw} < 1}$$

For α_{sw} = 0, the circuit never switches so no dynamic power is consumed

For α_{sw} = 1, the node switches as often as the clock (the circuit cannot switch more often than this) so $f_{sw} = f_{clock}$

Most cases → somewhere in between

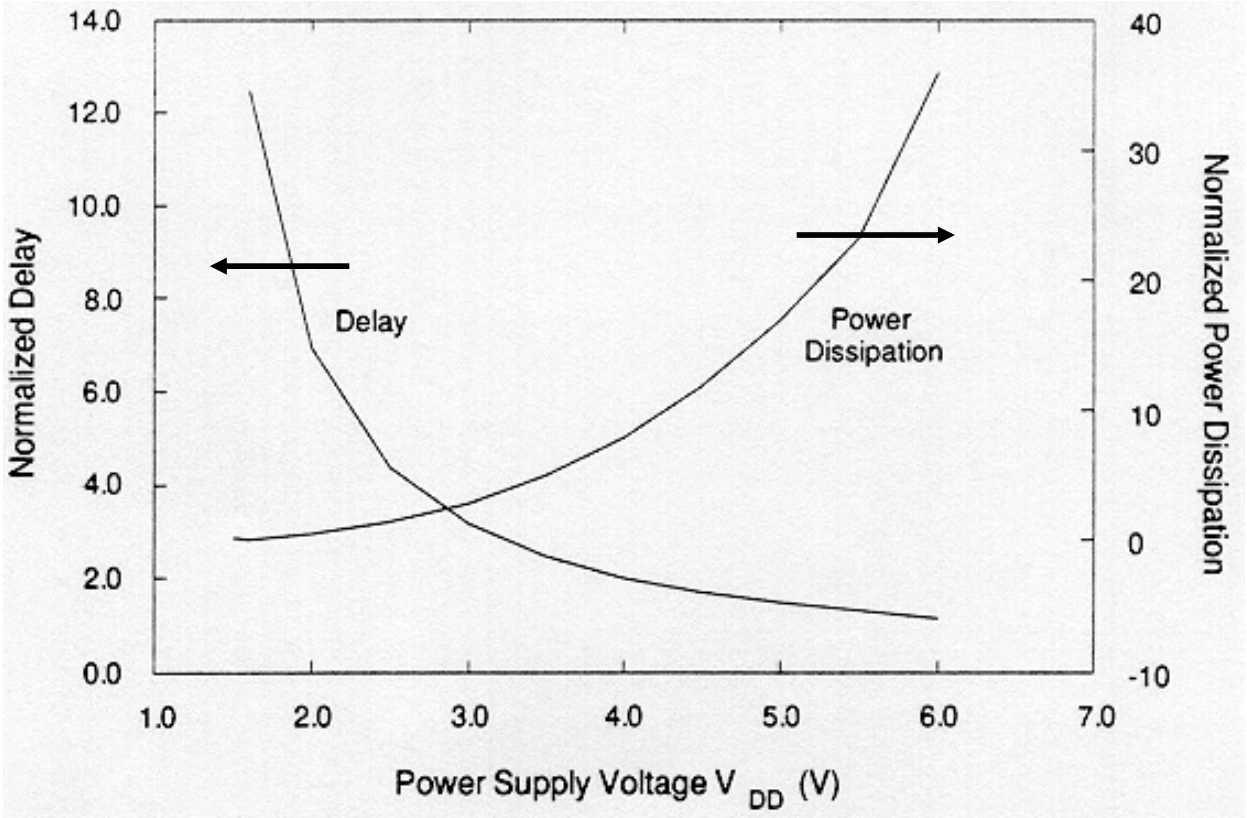
Lower α_{sw} = lower power



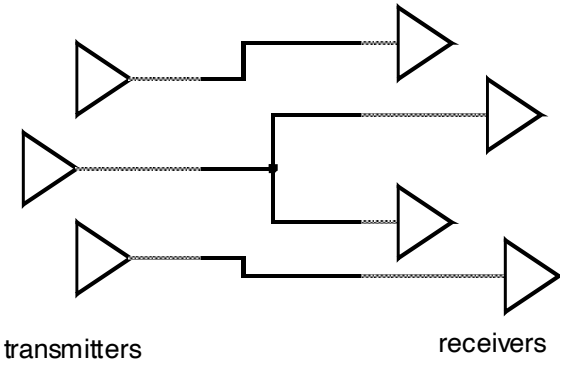
$$P_{dyn} = \alpha_{sw} * C_L * V_{dd}^2 * f_{clock}$$

Fundamental Tradeoff: Power vs. Delay

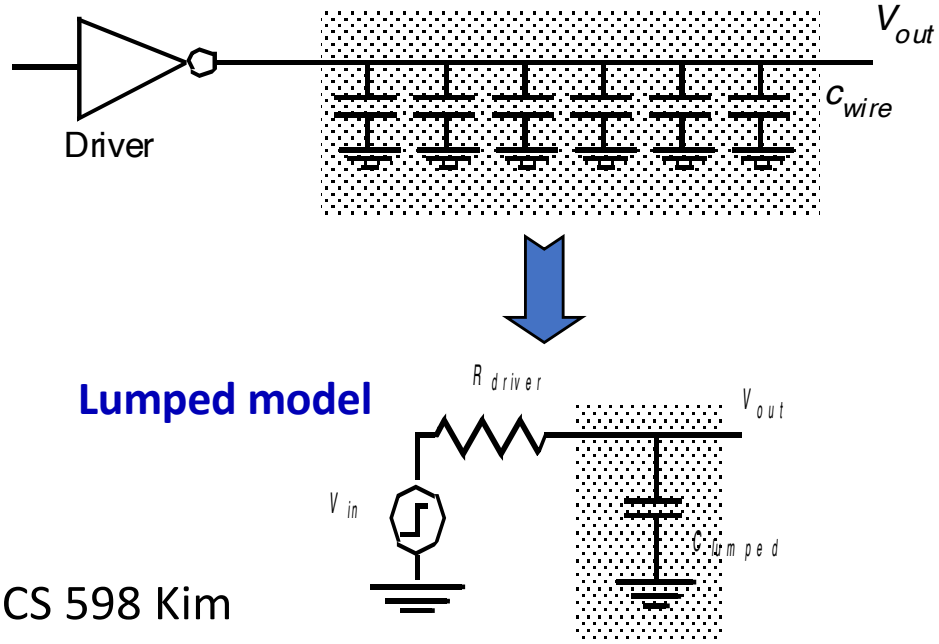
Power and delay tradeoff via voltage scaling



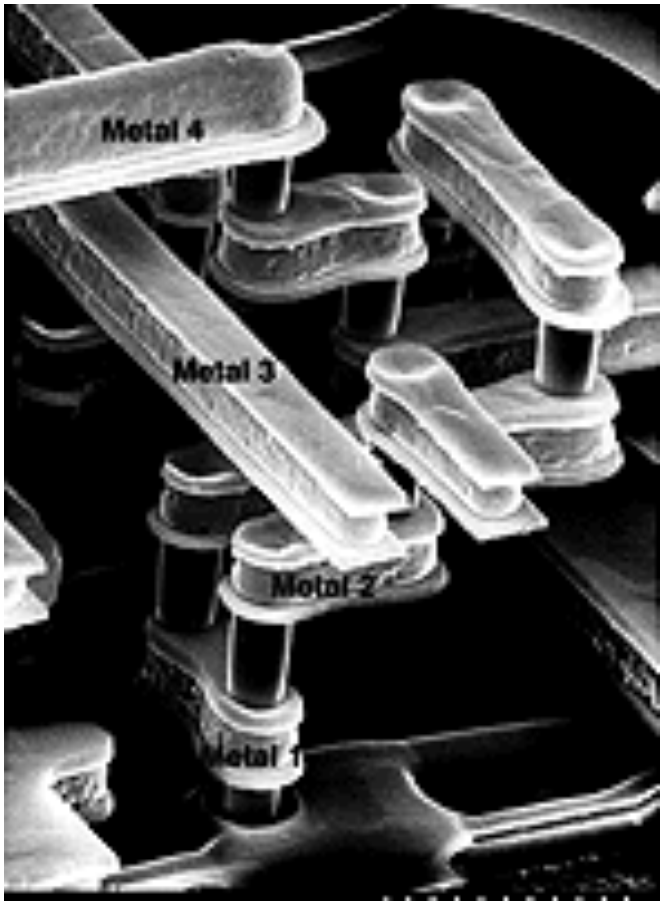
The Wire



schematic



Lumped model

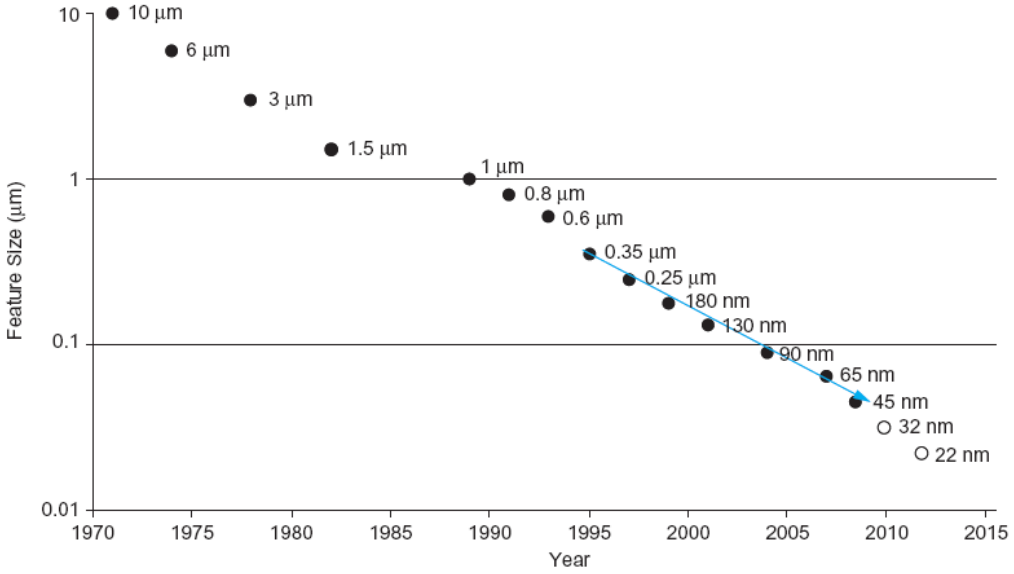


physical

Capacitance and resistance are proportional to wire length

Process Technology Scaling

- The only constant in VLSI is constant change
- Feature size shrinks by 30% every 2-3 years
  - Transistors become cheaper
  - Transistors become faster and lower power
  - Wires do not improve (and may get worse)
- Scale factor S
  - Typically $S = \sqrt{2}$
  - Technology nodes



Process Technology Scaling

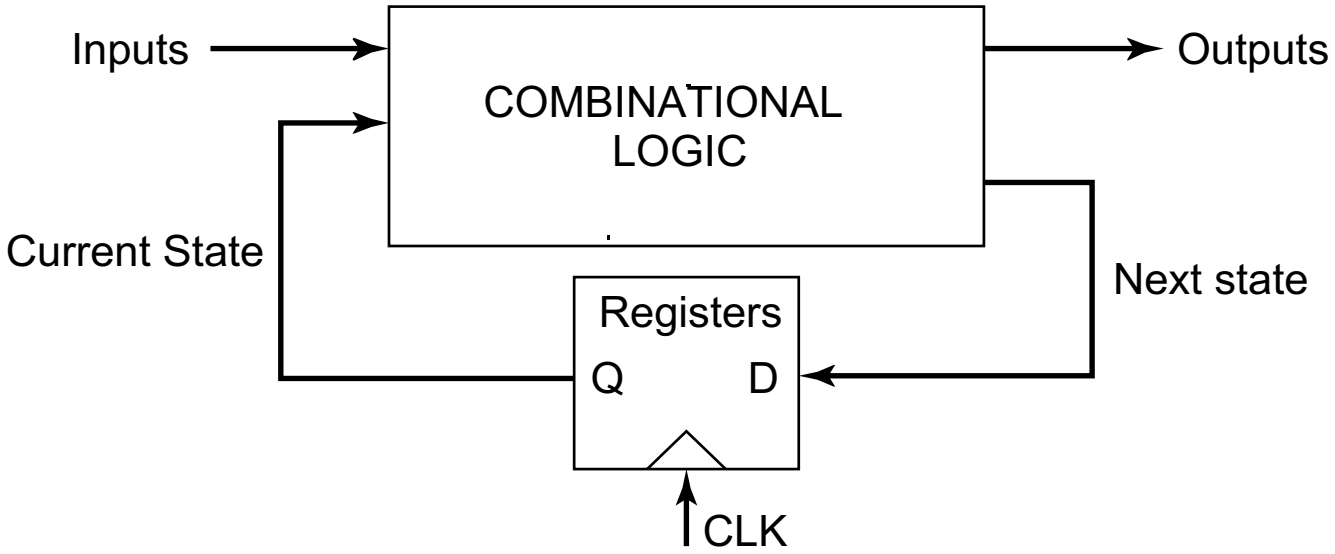
newer technology

S, U ≥ 1

S: feature size scaling, U: voltage scaling

Parameter	Sensitivity	Dennard Scaling (U=S)	General Scaling (U,S)	Fixed Voltage Scaling (U = 1)
L: Length		1/S	1/S	1/S
W: Width		1/S	1/S	1/S
t_{ox}: gate oxide thickness		1/S	1/S	1/S
V_{DD}: supply voltage		1/S	1/U	1
V_t: threshold voltage		1/S	1/U	1
NA: substrate doping		S	S^2/U	S^2
β	$W/(Lt_{ox})$	S	S	S
I_{on}: ON current	$\beta(V_{DD}-V_t)V_{min}$	1/S	1/U	1
R: effective resistance	V_{DD}/I_{on}	1	1	1
C: gate capacitance	WL/t_{ox}	1/S	1/S	1/S
τ: gate delay	RC	1/S	1/S	1/S
f: clock frequency	$1/\tau$	S	S	S
E: switching energy / gate	CV_{DD}^2	$1/S^3$	$1/(SU^2)$	1/S
P: switching power / gate	Ef	$1/S^2$	$1/U^2$	1
A: area per gate	WL	$1/S^2$	$1/S^2$	$1/S^2$
Switching power density	P/A	1	S^2/U^2	S^2
Switching current density	I_{on}/A	S	S^2/U	S^2

Sequential Logic



- *Combinational logic*
  - output depends on current inputs
- *Sequential logic*
  - output depends on current and previous inputs
  - Requires separating previous, current, future data

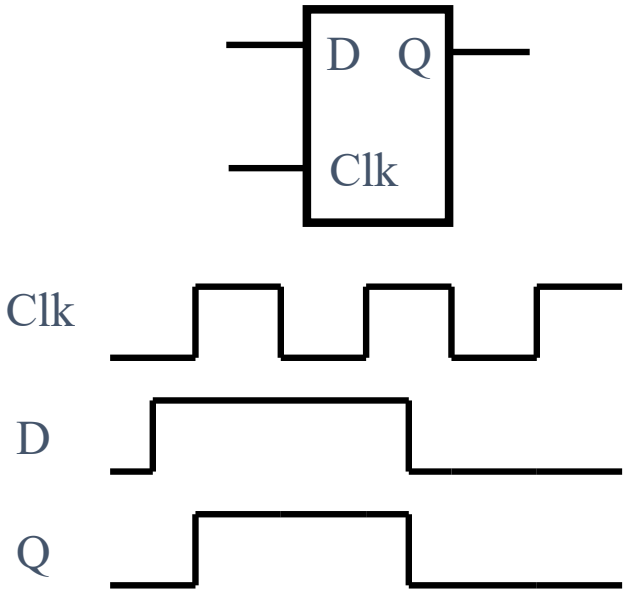
Naming Conventions

- In our text:
  - a latch is level sensitive
  - a register is edge-triggered
- There are many different naming conventions
  - For instance, many books call edge-triggered elements flip-flops
  - Registers and flip-flop terms are often used interchangeably therefore

Latch versus Register

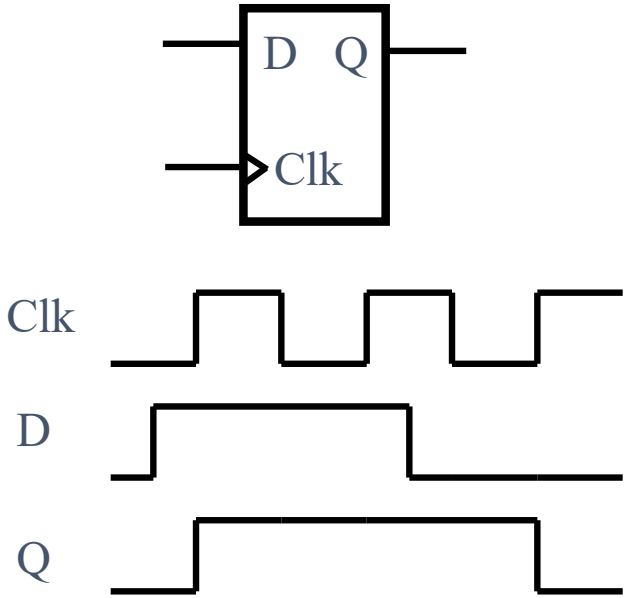
- Latch

  stores data when clock is low

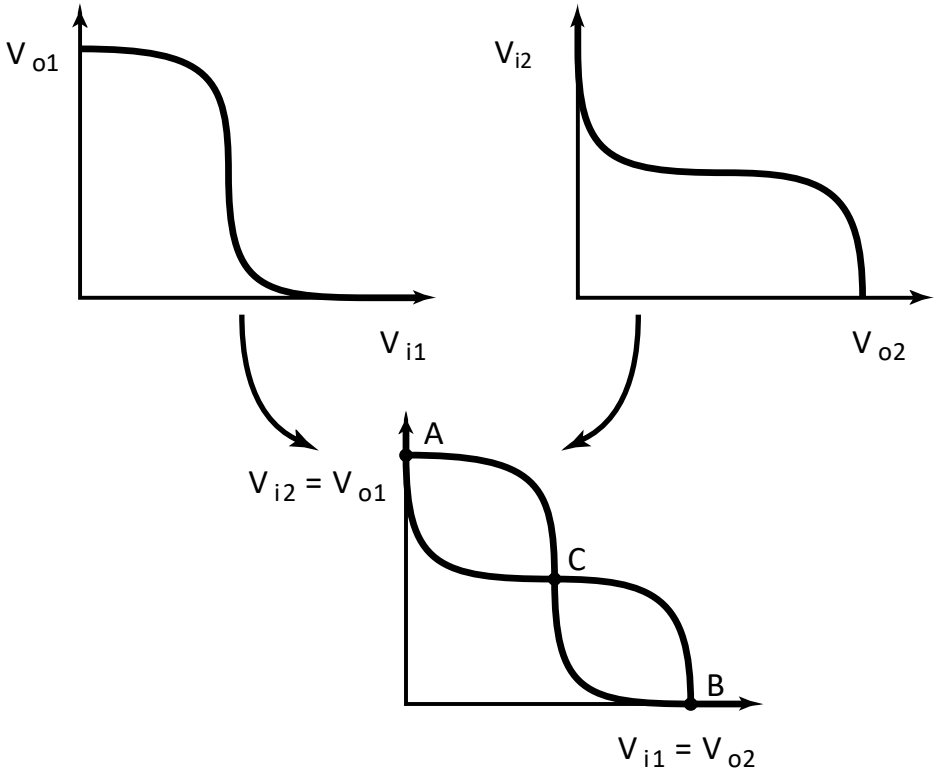


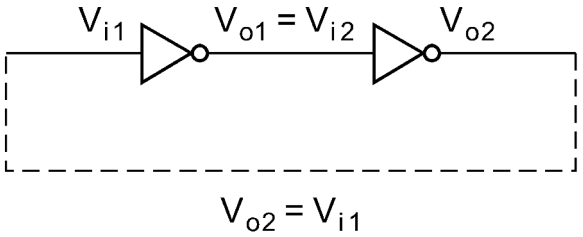
- Register

  stores data when clock rises



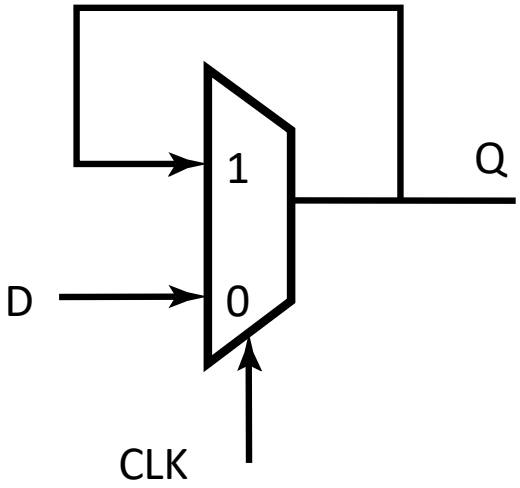
Bit storage mechanism: Bi-Stability





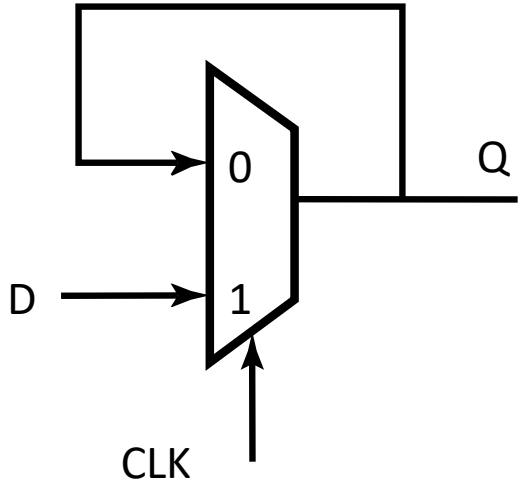
Mux-Based Latches

Negative latch
(transparent when CLK= 0)



$$Q = \overline{Clk} \cdot D + Clk \cdot Q$$

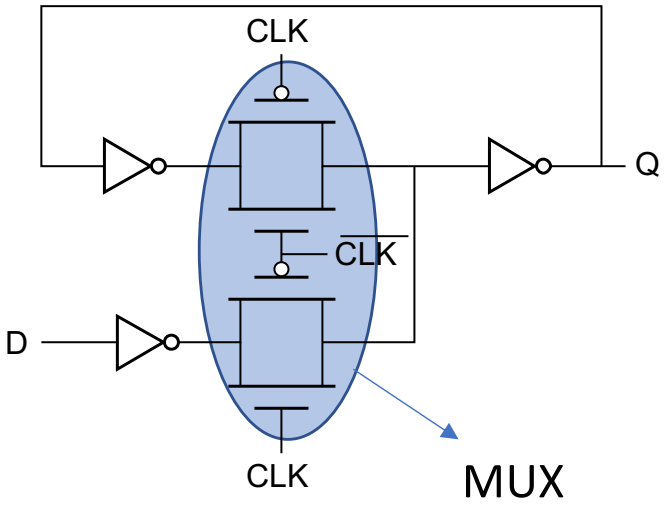
Positive latch
(transparent when CLK= 1)



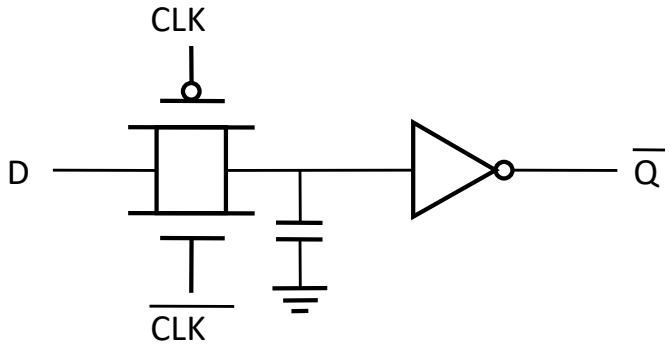
$$Q = Clk \cdot D + \overline{Clk} \cdot Q$$

Storage Mechanisms

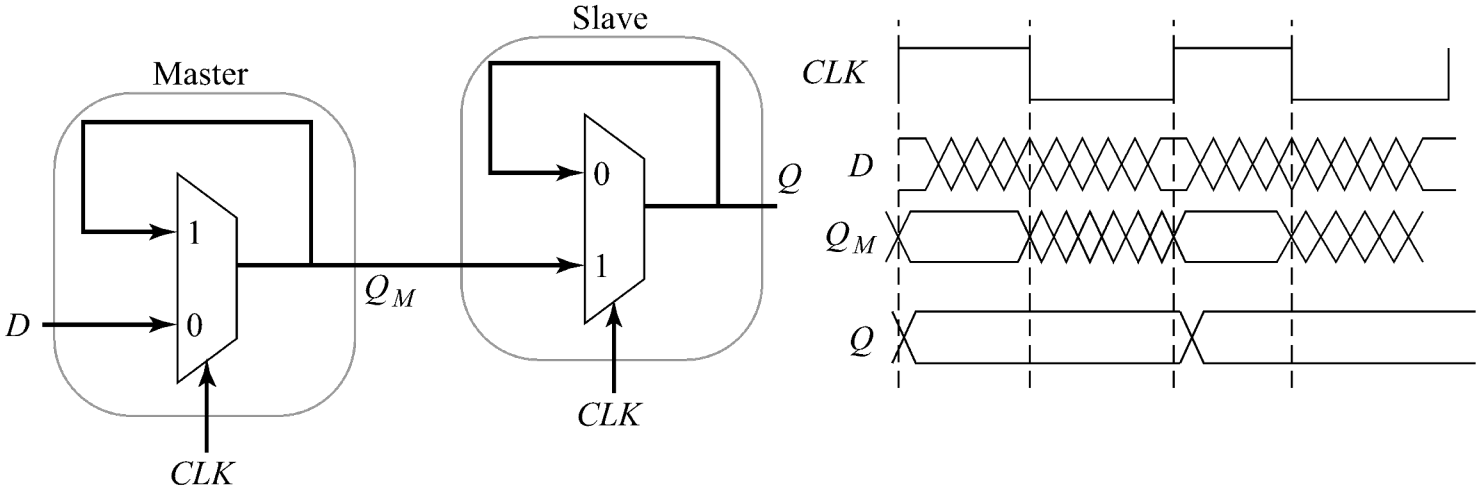
Static



Dynamic (charge-based)



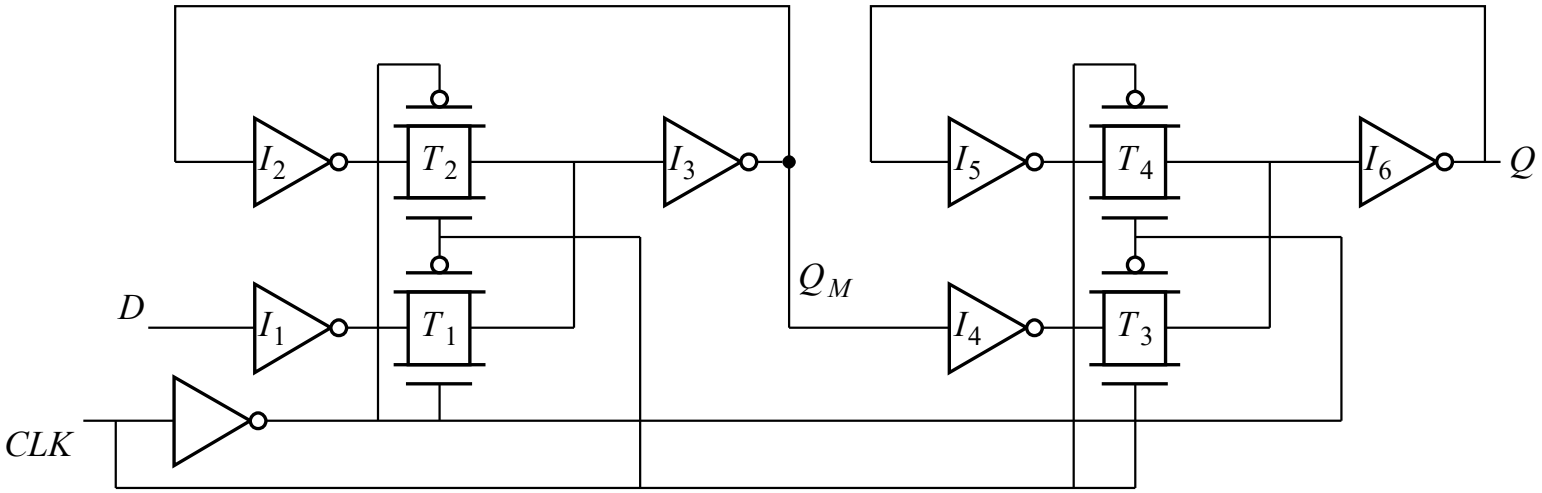
Master-Slave (Edge-Triggered) Register



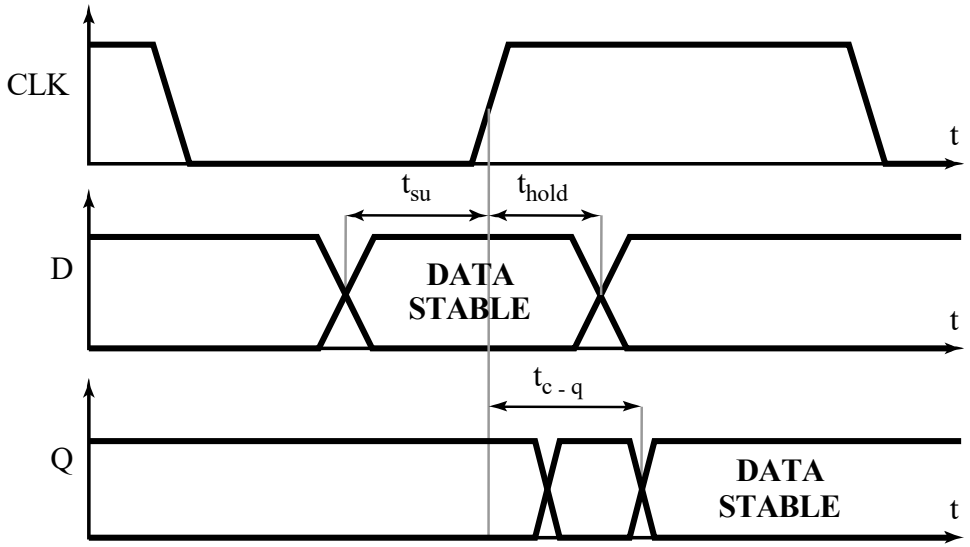
Two opposite latches trigger on edge
Also called master-slave latch pair

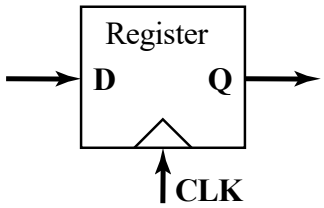
Master-Slave Register

Multiplexer-based latch pair

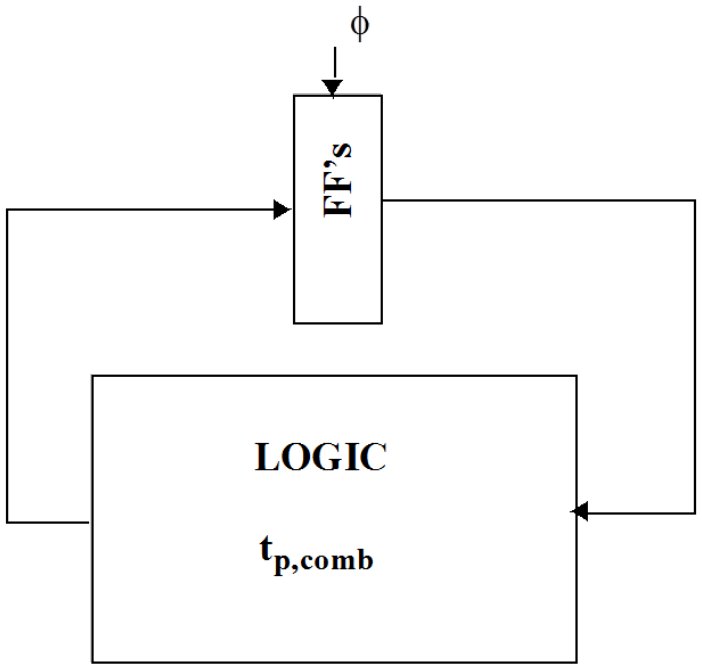


Register Timing Definitions



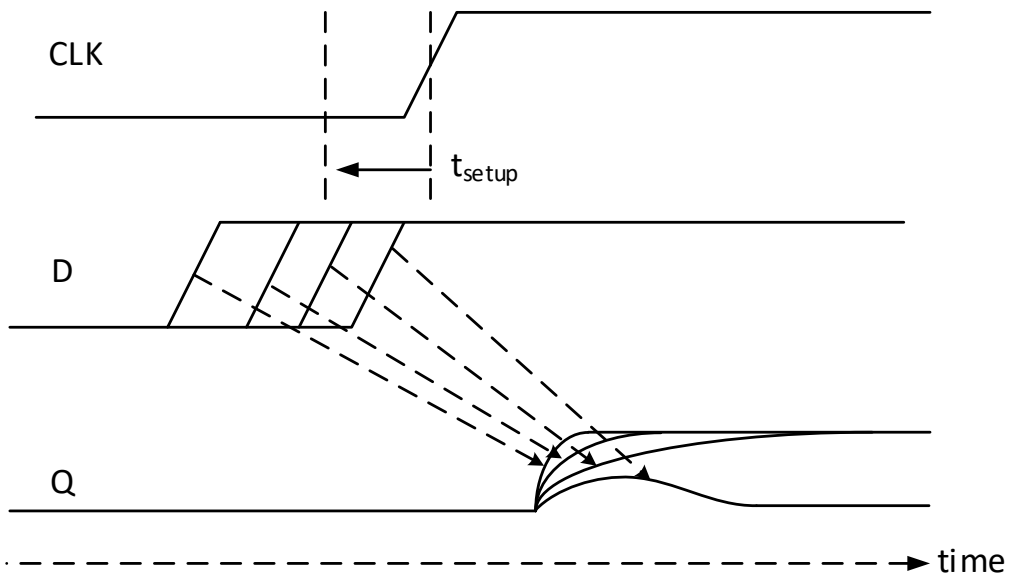


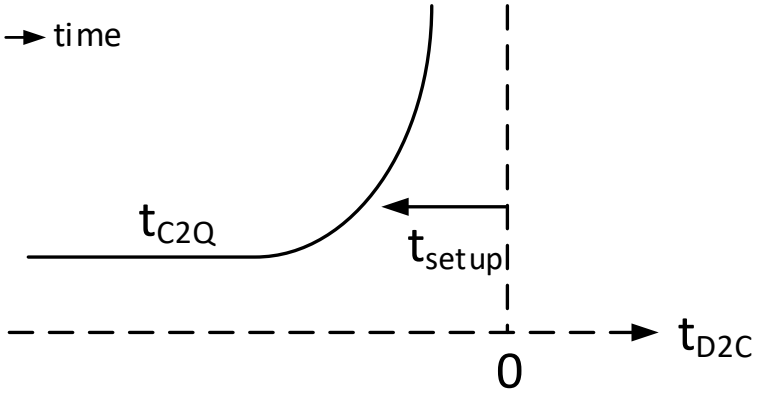
Maximum Clock Frequency



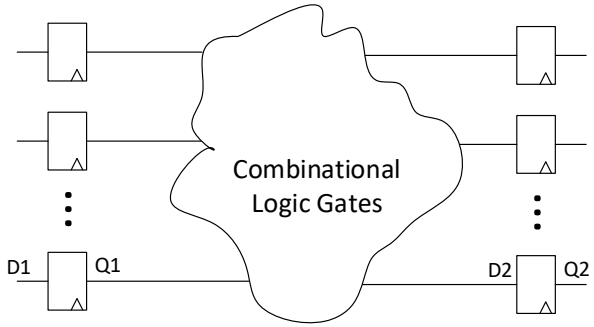
$$t_{clk\text{-}Q} + t_{p,comb} + t_{setup} = T$$

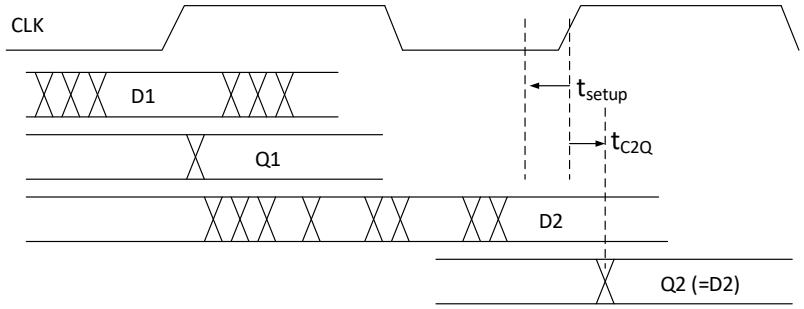
Setup Time



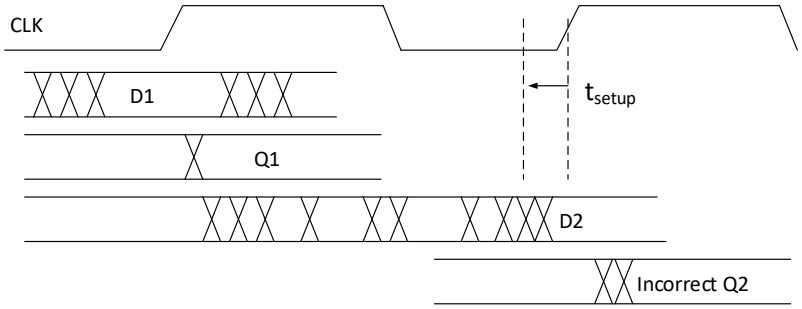


Setup Time Constraint



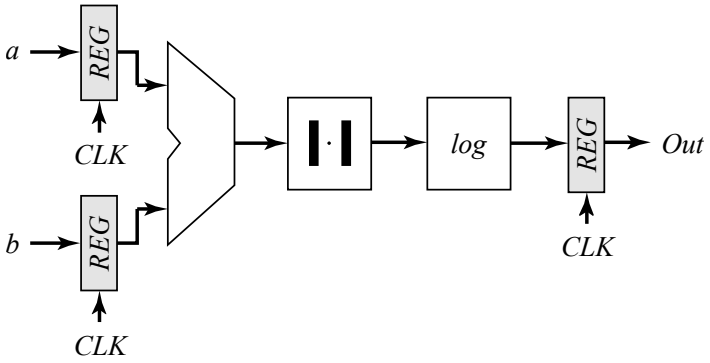


Setup time constraint satisfied

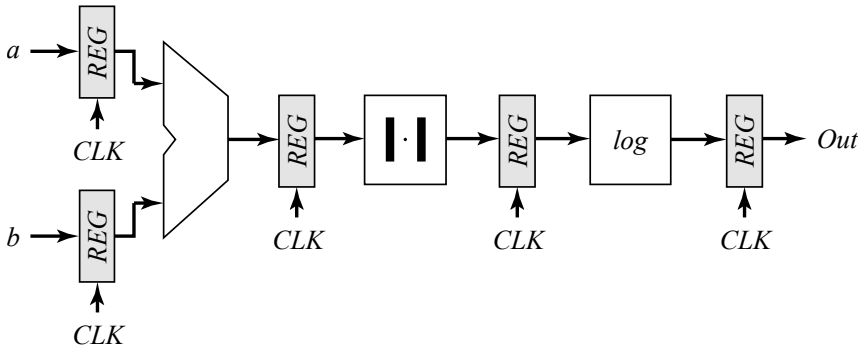


Setup time constraint violated

Pipelining



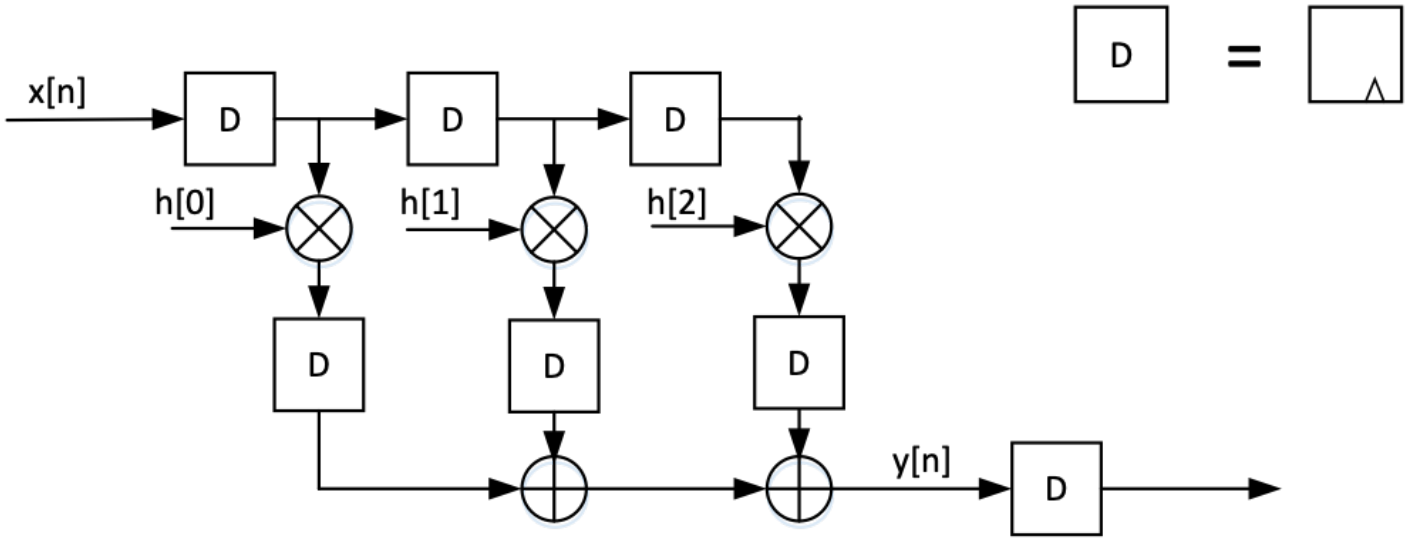
Reference



Pipelined

Clock Period	Adder	Absolute Value	Logarithm
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\lvert a_1 + b_1 \rvert$	
3	$a_3 + b_3$	$\lvert a_2 + b_2 \rvert$	$\log(\lvert a_1 + b_1 \rvert)$
4	$a_4 + b_4$	$\lvert a_3 + b_3 \rvert$	$\log(\lvert a_2 + b_2 \rvert)$
5	$a_5 + b_5$	$\lvert a_4 + b_4 \rvert$	$\log(\lvert a_3 + b_3 \rvert)$

Pipeline Example



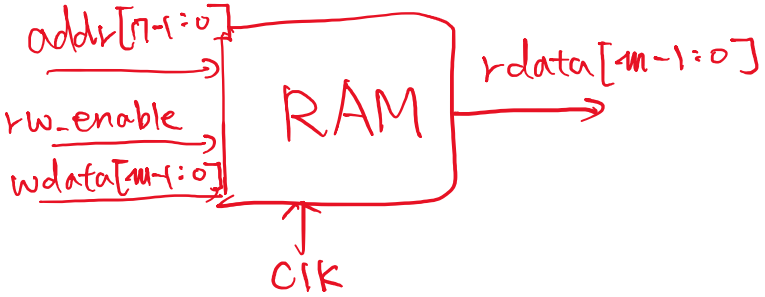
$$y[n] = h[2]x[n-2] + h[1]x[n-1] + h[0]x[n]$$

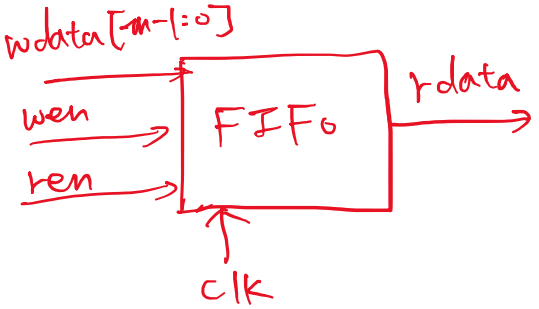
Semiconductor Memory Classification

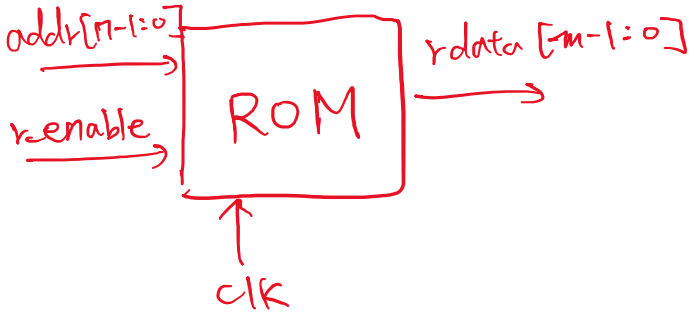
Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	Non-Random Access	EPROM E^2PROM FLASH	Mask-Programmed Read Only Memory (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM		

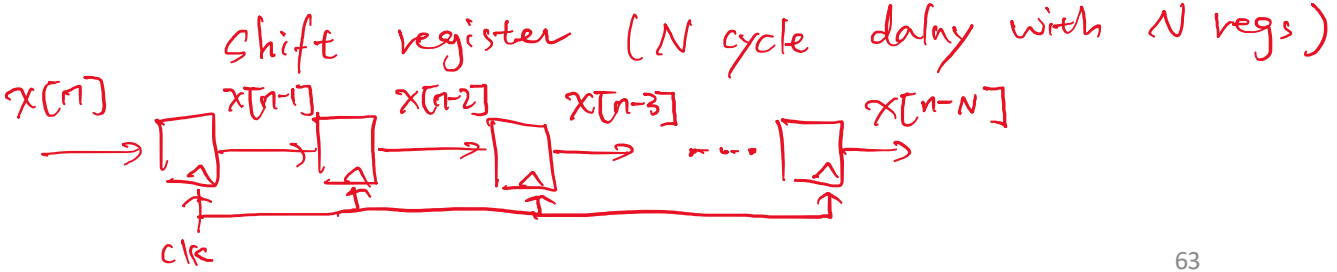
Key design metric: AREA

Memory Interface

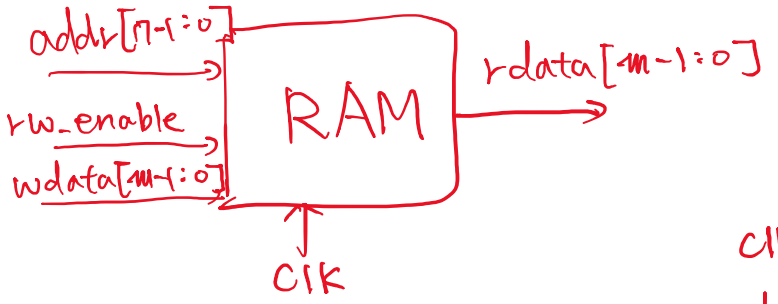




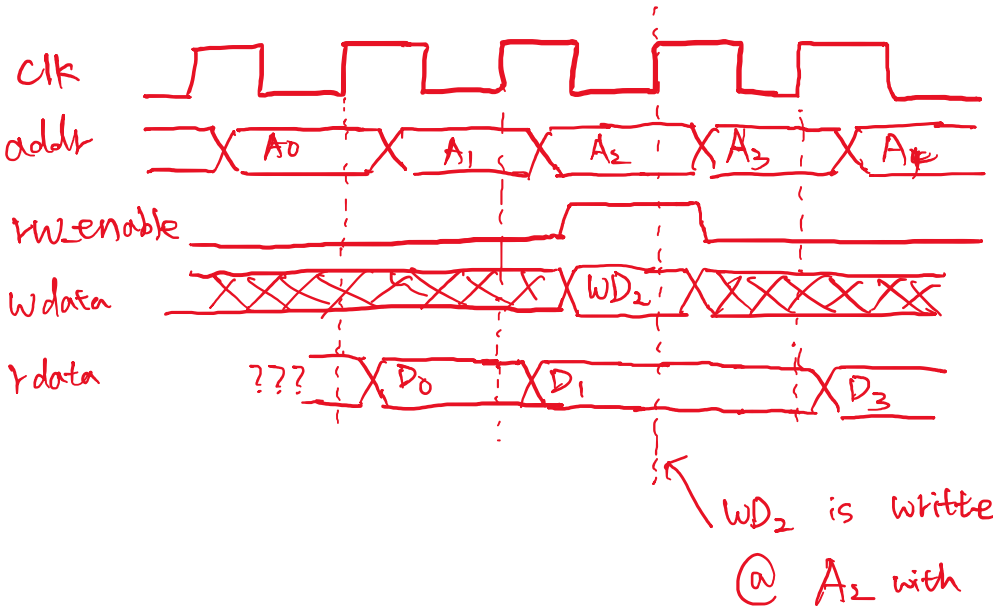




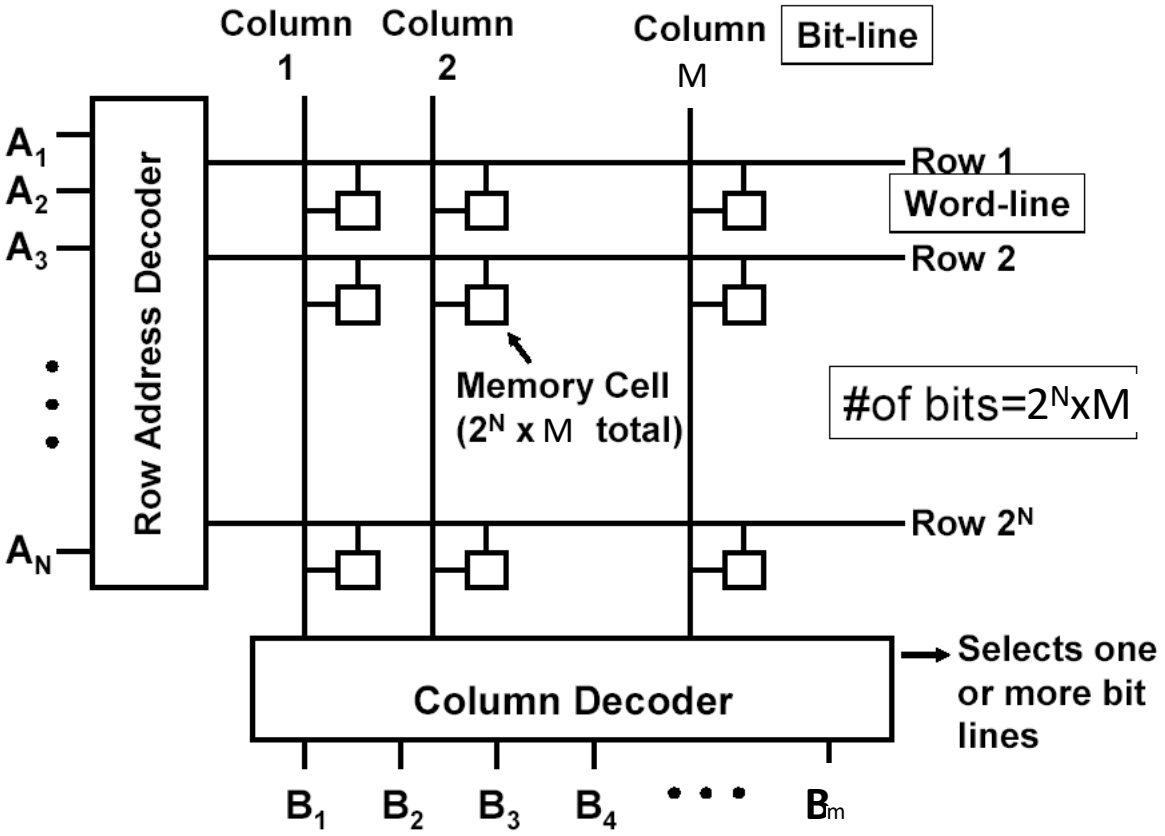
Memory Timing: Definitions



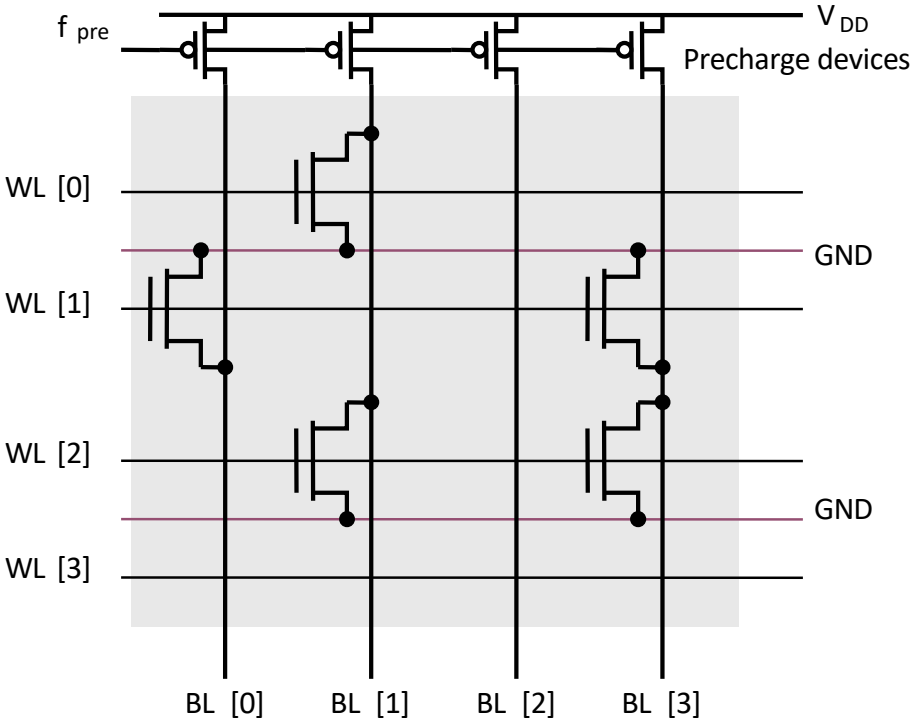
$$rw\_enable = \begin{cases} 0 & \text{when read} \\ 1 & \text{when write} \end{cases}$$



General Memory Structure



Precharged MOS NOR ROM



Address Decoders

M:2^M decoder:
Collection of 2^M M-input complex logic gates
Organized in regular and dense fashion

here
M is # of bits for address
2^M is # of rows

(N)AND Decoder

$$WL_0 = \overline{A_0}\,\overline{A_1}\,\overline{A_2}\,\overline{A_3}\,\overline{A_4}\,\overline{A_5}\,\overline{A_6}\,\overline{A_7}\,\overline{A_8}\,\overline{A_9}$$

MSB LSB

$WL_0 = 1$ when Addr = 000…0

$$WL_{511} = \overline{A_0}A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

MSB LSB

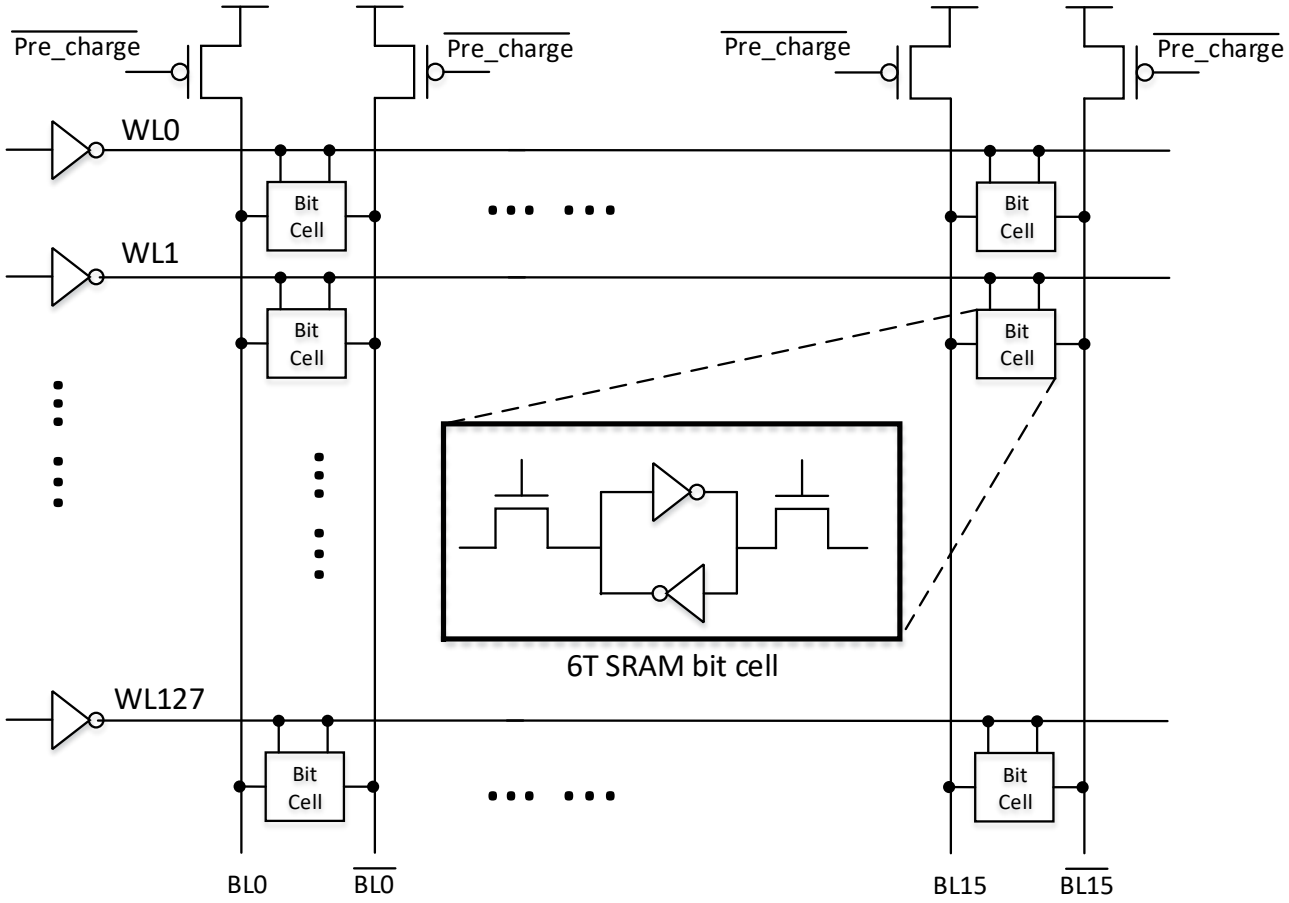
$W_{511} = 1$ when Addr = 0111…1

NOR Decoder

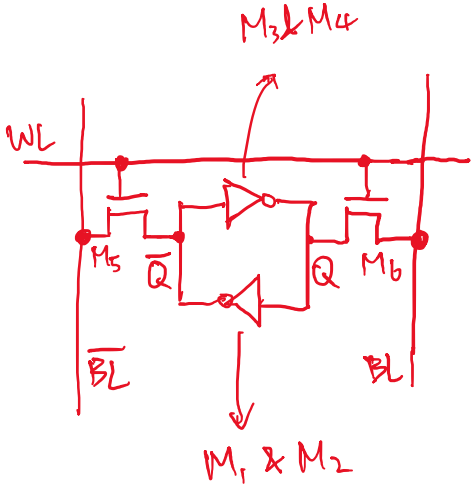
$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

$$WL_{511} = \overline{A_0 + \bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \bar{A}_4 + \bar{A}_5 + \bar{A}_6 + \bar{A}_7 + \bar{A}_8 + \bar{A}_9}$$

Static RAM (SRAM)



6-transistor CMOS SRAM Cell

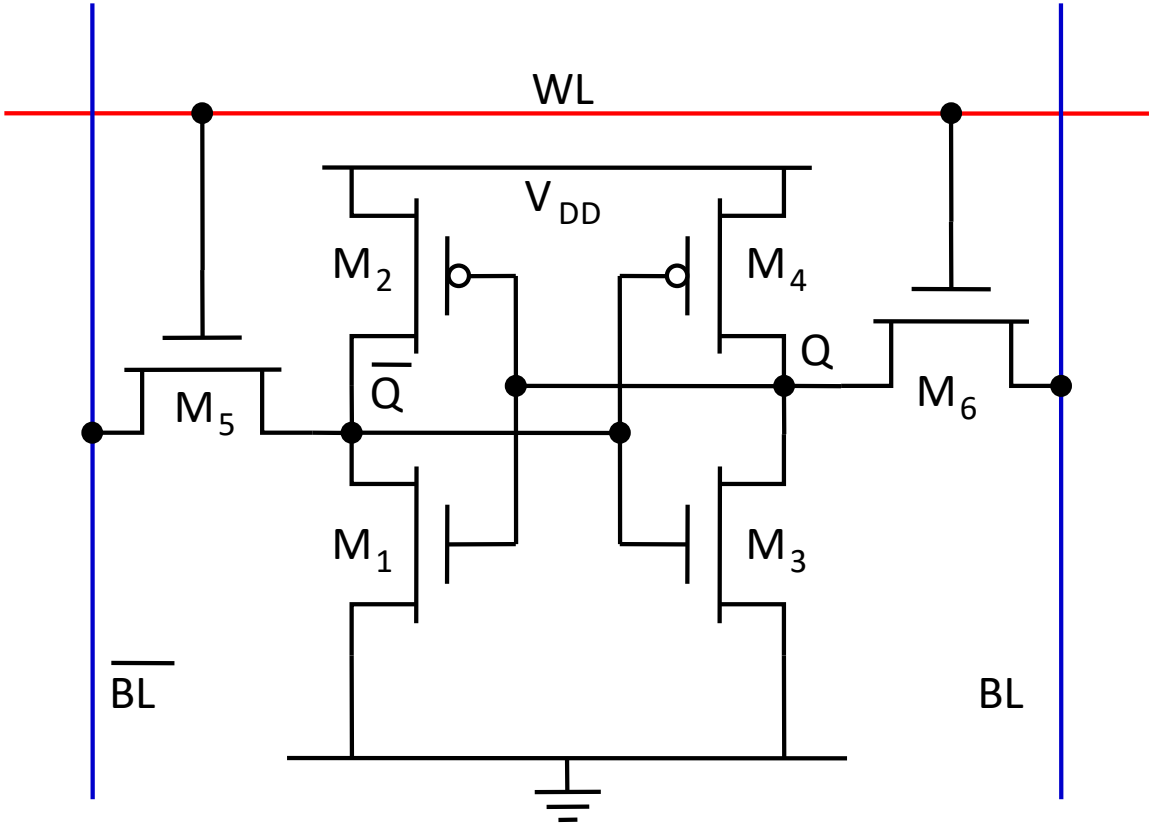


read

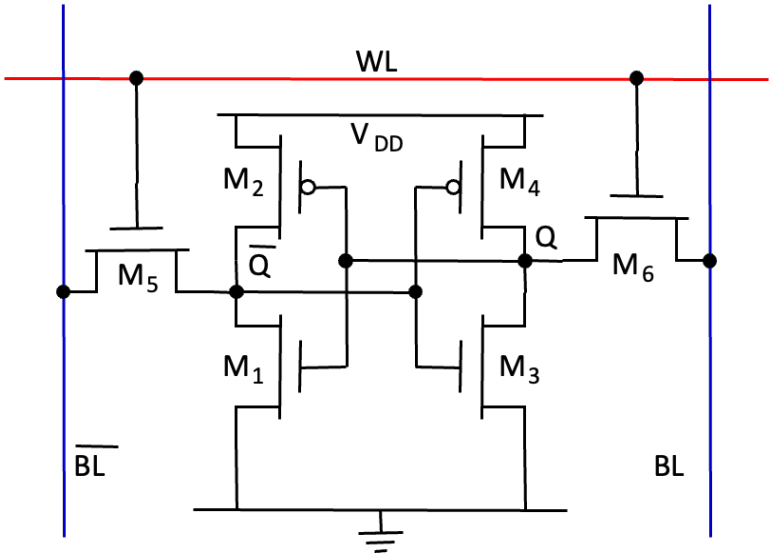
precharge BL & $\overline{BL}$

→ assert WL high

→ sense BL & $\overline{BL}$ (read)

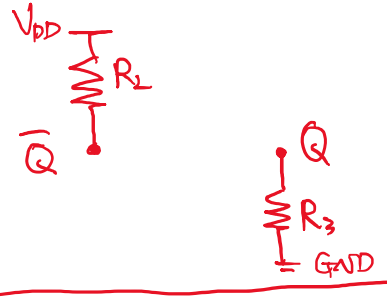


6-transistor CMOS SRAM Cell



Writing $Q = 0 \rightarrow 1$

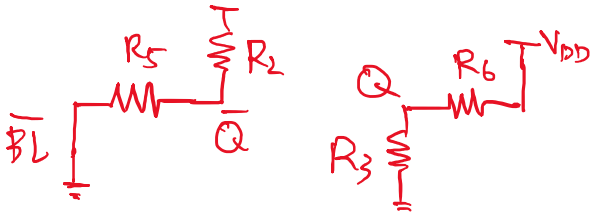
When $Q=0$, $\overline{Q}=1$: M_2 & M_3 : on, M_1 & M_4 : OFF



$V_{\overline{Q}} = V_{DD}$

$V_Q = 0$

to write $Q=1$, drive $\begin{cases} WL = V_{DD} \\ BL = V_{DD},\ \overline{BL} = 0 \end{cases}$

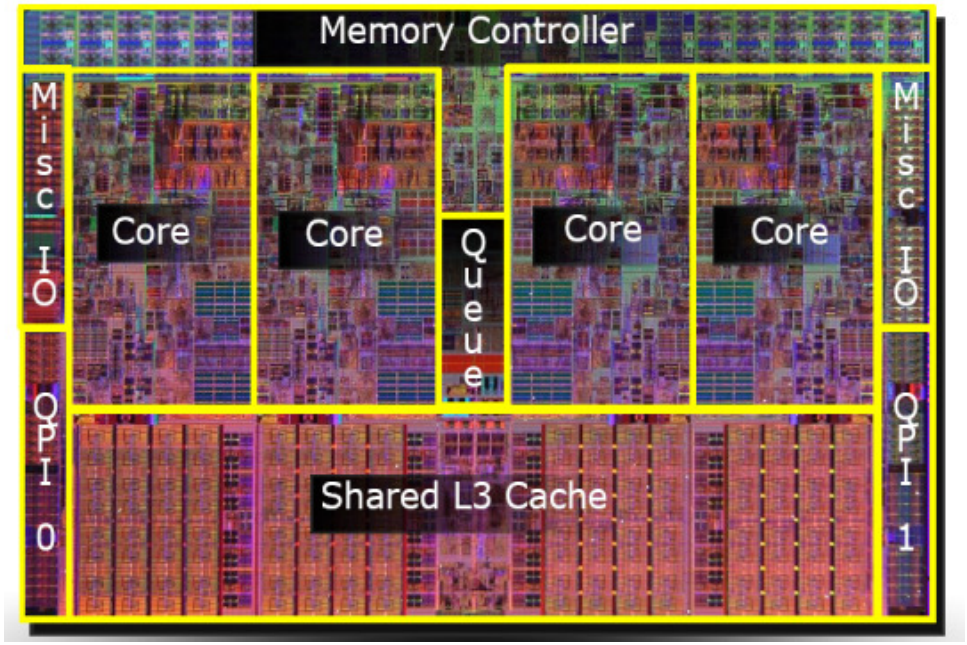


When $R_5 \ll R_2$

$V_{\overline{Q}}$ becomes $< V_{DD}/2$

M_4 turns on
M_3 turns off
$\Rightarrow$ Q becomes 1
$\overline{Q}$ " 0

On-chip SRAM is very common



Intel Core i7

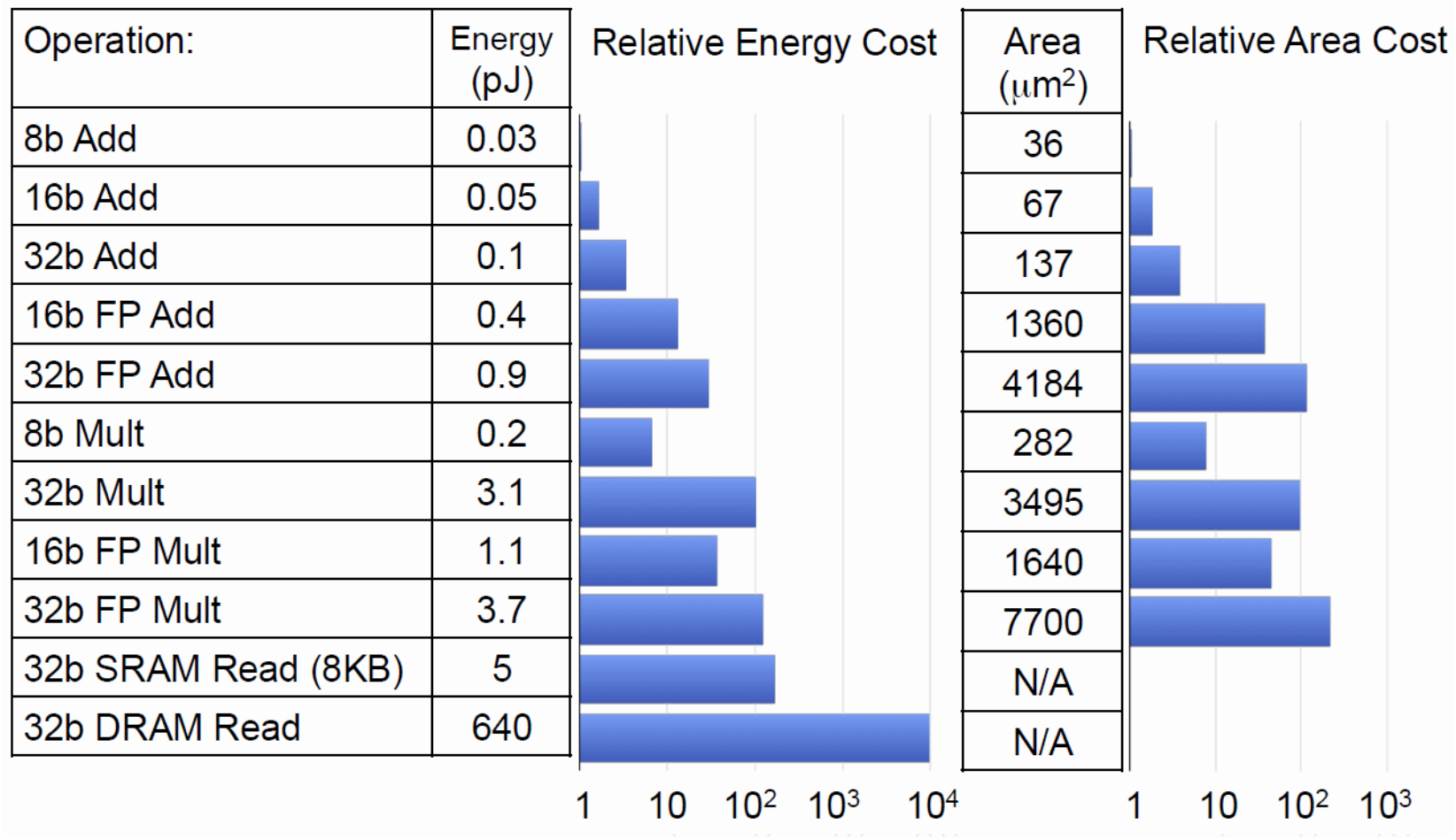
Recently used data is stored in on-chip *cache* memory

The larger the cache, the less frequently you need to go off-chip to access data (SLOW)

SRAM is technology compatible with logic

Normalized Energy and Area

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A



[Horowitz, "Computing's Energy Problem (and what we can do about it)", ISSCC 2014]