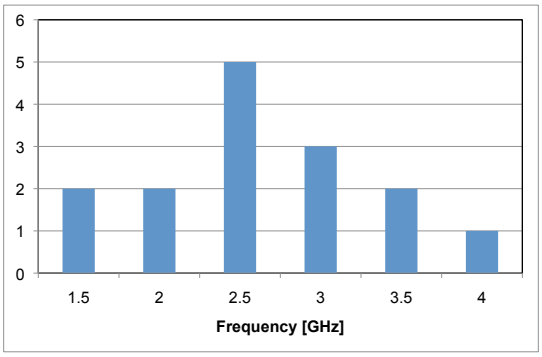
EE141-Spring 2010 Digital Integrated Circuits

Lecture 25
Memory

Administrativia

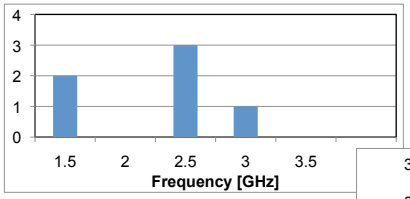
- Hw 8 Posted
  - Last one to be graded
  - Due Friday April 30
- Hw 6 and 7 Graded and available
- Project Phase 2 Graded
- Project Phase 3 Launch Today

Project Phase 2

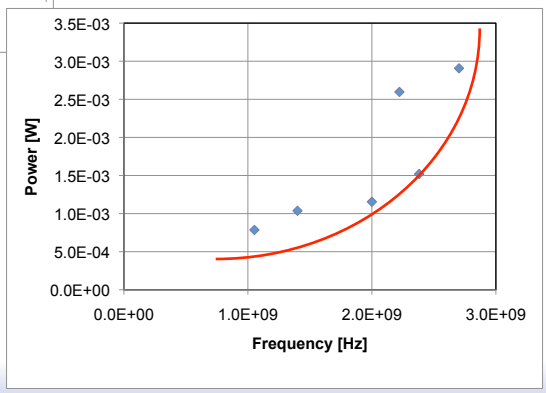


Performance Histogram

Energy-Delay Plot

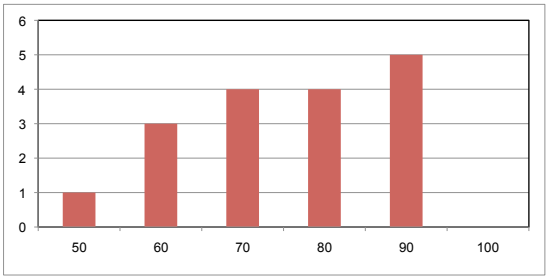


Performance - pruned



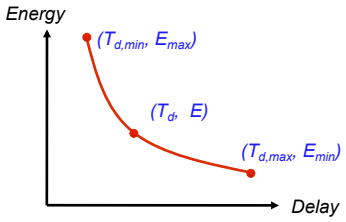
Phase 2- Results

- Max: 98
- Avg: 78.9
- StDev: 12.9
- Median: 81



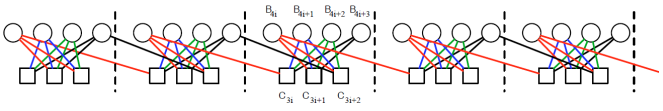
Project Phase 3

- Design implementation of 256-bit code
- Create floorplan to estimate wirelength
- Optimize circuit such that energy is minimized for max delay of 10 nsec
- Determine also Tdmin and Emin for that design
- Present results in poster

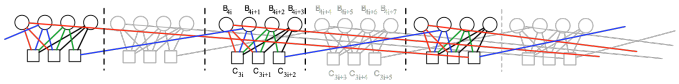


Code Design

Template:



Group A:

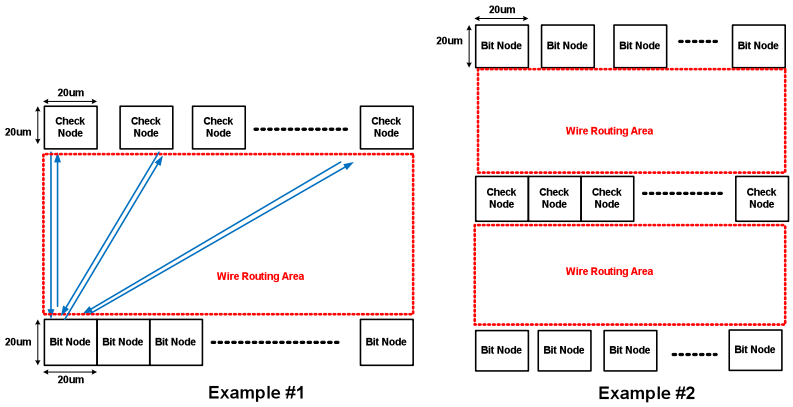


Group B: Same but period is 4

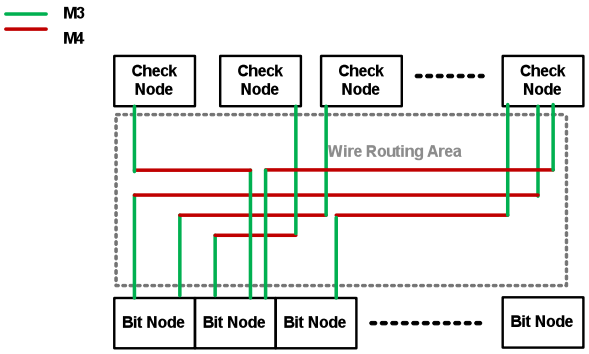
Group C: Same but period is 8

Group D: Same but period is 16

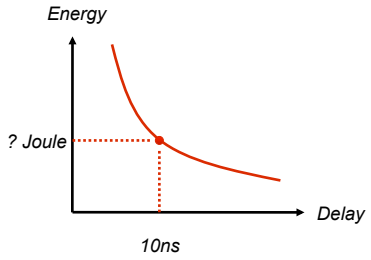
Possible Floorplans



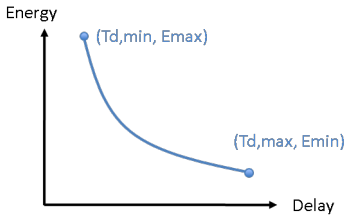
Estimating Wire Length



Optimization Criteria



Optimization target



The extremes

Class Material

- Last lecture
  - Multipliers
- Today's lecture
  - Memory cells
- Reading (Ch 12)

Semiconductor Memory

Semiconductor Memory Classification

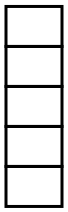
Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	**Non-Random Access**	EPROM	Mask-Programmed
		E^2PROM	Programmable (PROM)
SRAM	FIFO	FLASH	
DRAM	LIFO		
	Shift Register		
	CAM		

Random Access Memories (RAM)

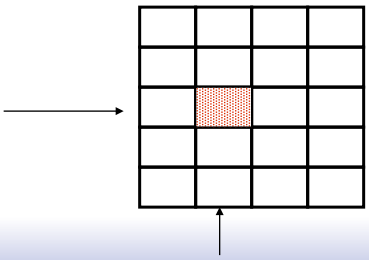
- **STATIC (SRAM)**
  - Data stored as long as supply is applied
  - Larger (6 transistors/cell)
  - Fast
  - Differential (usually)
- **DYNAMIC (DRAM)**
  - Periodic refresh required
  - Smaller (1-3 transistors/cell)
  - Slower
  - Single Ended

Random Access Chip Architecture

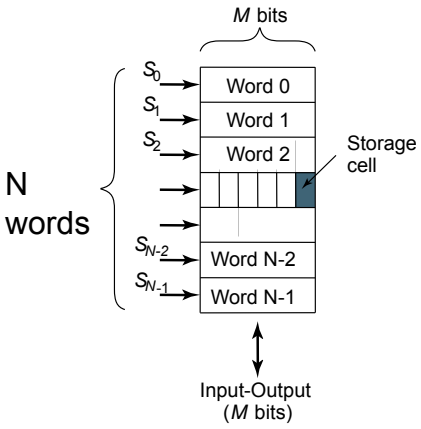
- Conceptual: linear array
  - Each box holds some data
  - But this does not lead to a nice layout shape
  - Too long and skinny



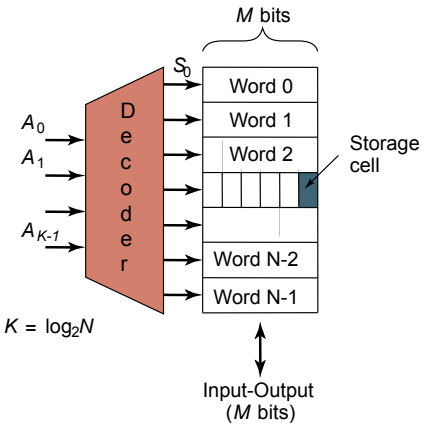
- Create a 2-D array
  - Decode Row and Column address to get data



Memory Architecture: Decoders



Intuitive architecture for N x M memory
Too many select signals:
N words == N select signals



Decoder reduces the number of select signals

$K = log_2 N$

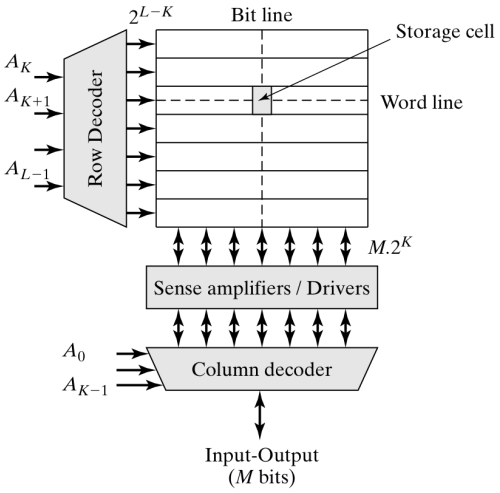
Memory Architecture

CORE:
- keep square within a 2:1 ratio
- rows are **word lines**
- columns are **bit lines**
- data in and out on columns

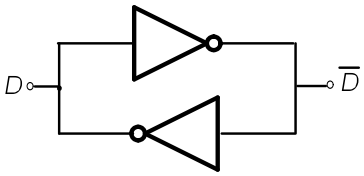
DECODERS:
- needed to reduce total number of pins; N+M address lines for 2^{N+M} bits of storage

Ex: if N+M=20 → 2^{20}= 1Mb

MULTIPLEXING:
- used to select one or more columns for input or output of data

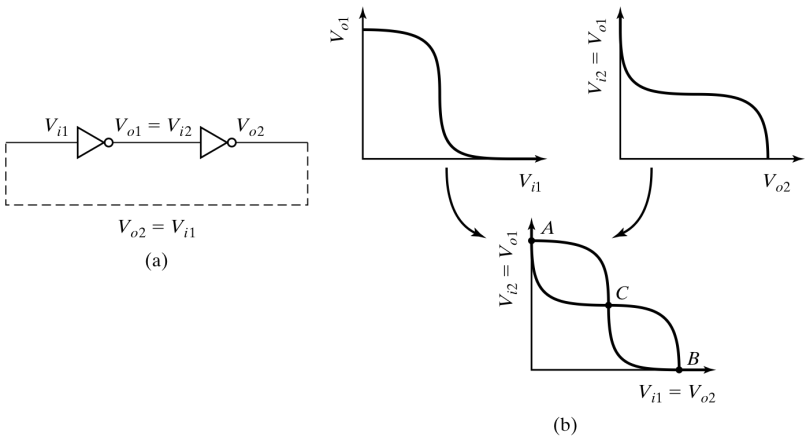


Basic Static Memory Element

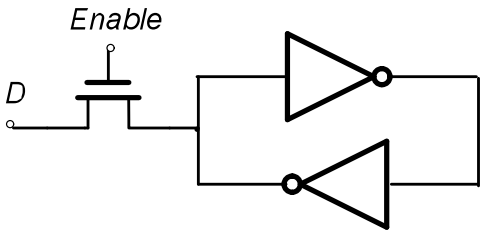


- If D is high, D_b will be driven low
  - Which makes D stay high
- Positive feedback

Positive Feedback: Bi-Stability



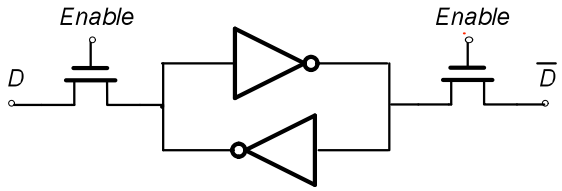
Writing into a Cross-Coupled Pair



Access transistor must be able to overpower the feedback

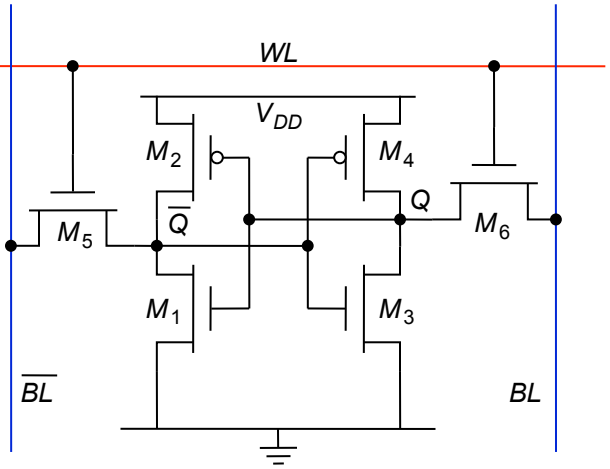
Writing a "1"

Memory Cell



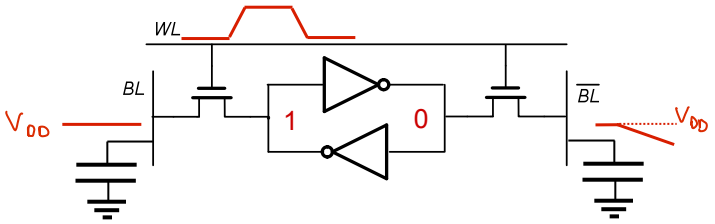
Complementary data values are written (read) from two sides

6-Transistor CMOS SRAM Cell



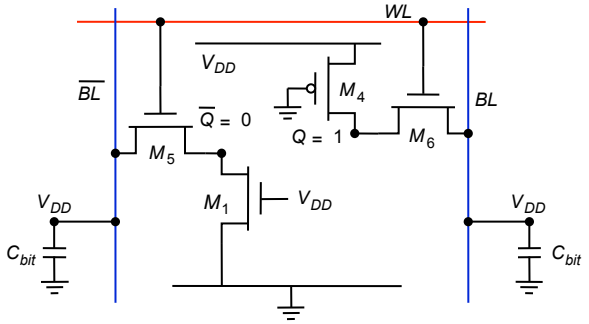
SRAM Operation - Read

Read



- Q_b will get pulled up when WL first goes high
  - Reading the cell should not destroy the stored value

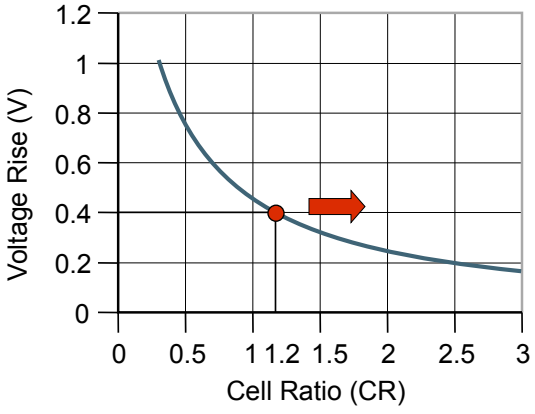
CMOS SRAM Analysis (Read)



$$k_{n,M5}\left((V_{DD}-\Delta V-V_{Tn})V_{DSATn}-\frac{V_{DSATn}^2}{2}\right) = k_{n,M1}\left((V_{DD}-V_{Tn})\Delta V-\frac{\Delta V^2}{2}\right)$$

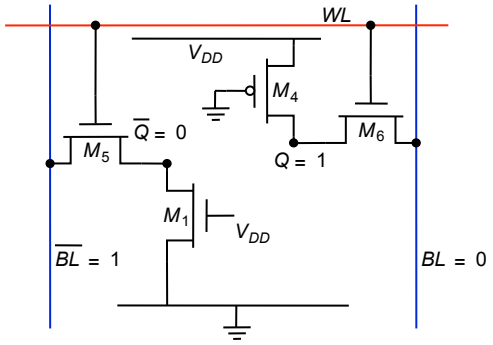
$$\Delta V = \frac{V_{DSATn}+CR(V_{DD}-V_{Tn})-\sqrt{V_{DSATn}^2(1+CR)+CR^2(V_{DD}-V_{Tn})^2}}{CR}$$

CMOS SRAM Analysis (Read)



$$CR = \frac{W_1/L_1}{W_5/L_5}$$

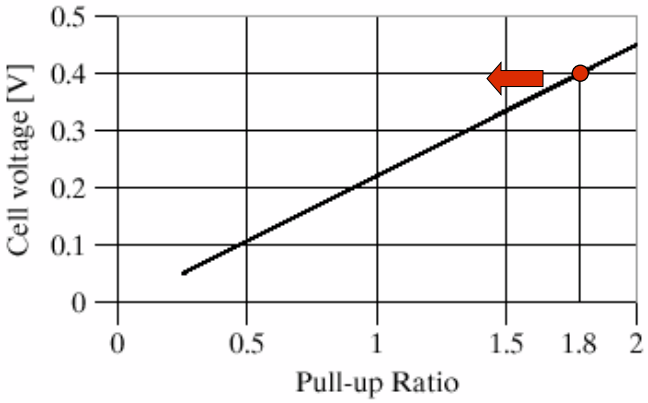
CMOS SRAM Analysis (Write)



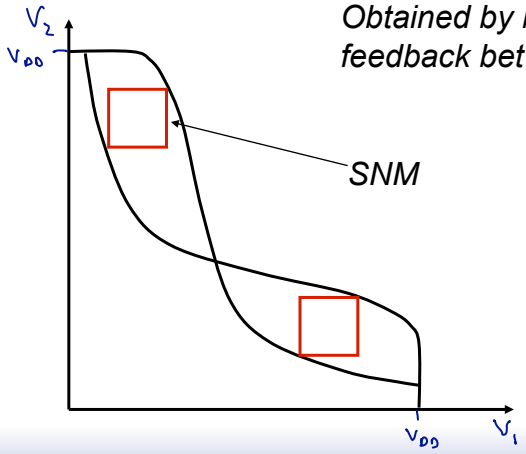
$$k_{n,M6}\left((V_{DD}-V_{Tn})V_Q-\frac{V_Q^2}{2}\right)=k_{p,M4}\left((V_{DD}-|V_{Tp}|)V_{DSATp}-\frac{V_{DSATp}^2}{2}\right)$$

$$V_Q = V_{DD}-V_{Tn}-\sqrt{(V_{DD}-V_{Tn})^2-2\frac{\mu_p}{\mu_n}PR\left((V_{DD}-|V_{Tp}|)V_{DSATp}-\frac{V_{DSATp}^2}{2}\right)},$$

CMOS SRAM Analysis (Write)

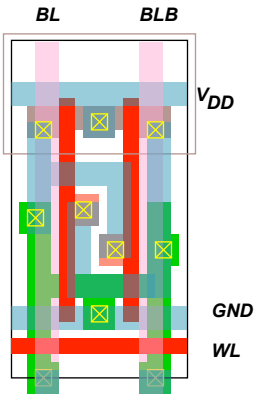


Read Static Noise Margin



Obtained by breaking the feedback between the inverters

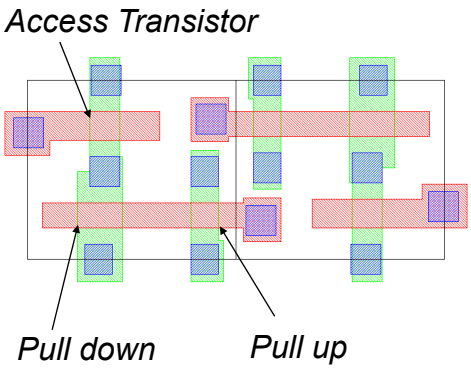
6T-SRAM — Layout

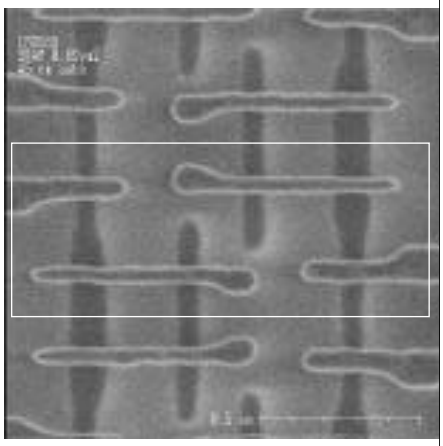


Compact cell
- Bitlines: M2
- Wordline: bootstrapped in M3

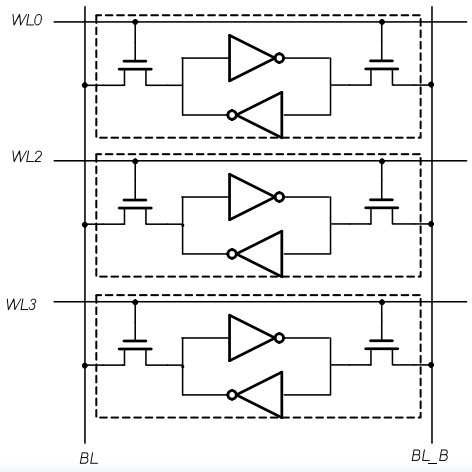
65nm SRAM

- ST/Philips/Motorola

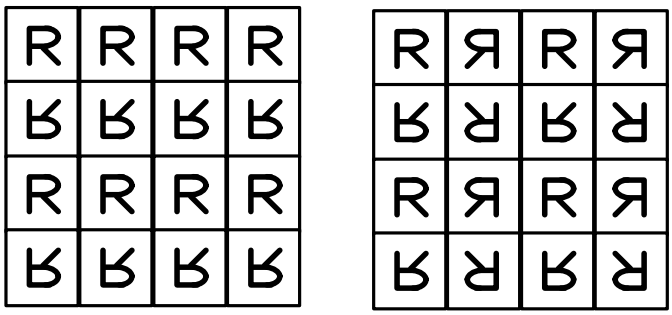




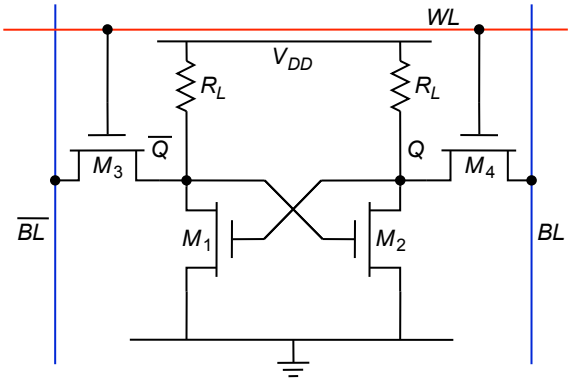
SRAM Column



SRAM Array Layout

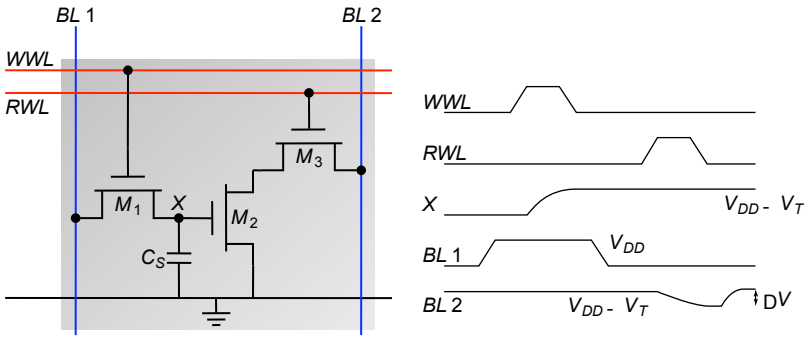


Resistive Pull-Ups



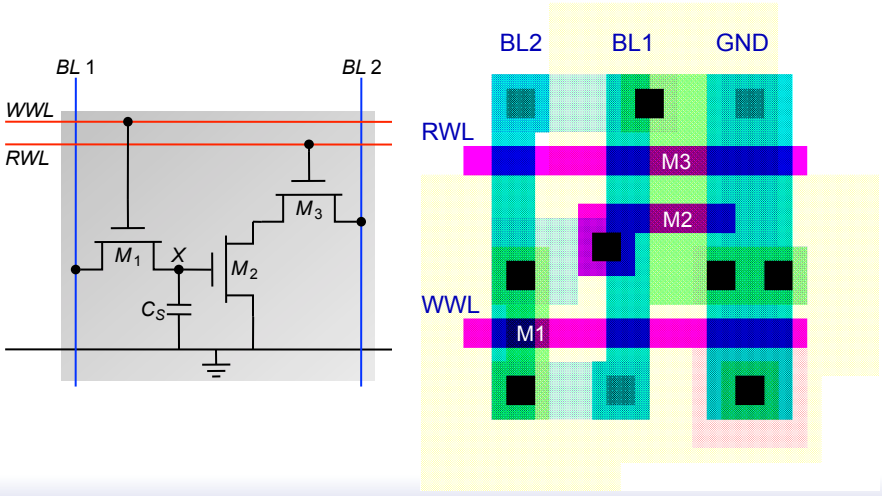
Static power dissipation -- Want R_L large
Bit lines precharged to V_{DD} to address t_p problem

3-Transistor DRAM Cell

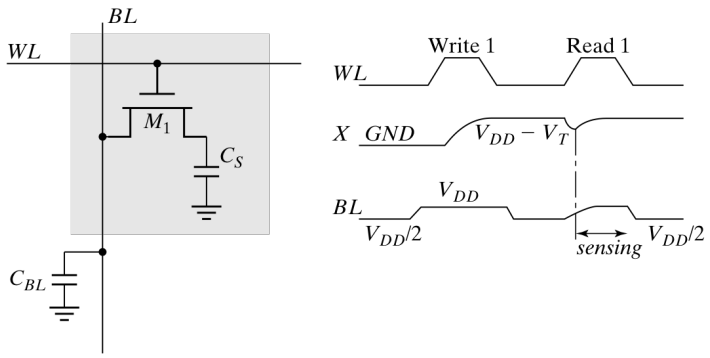


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a "1" = V $_{WWL}$-V_{Tn}

3T DRAM Layout



1-Transistor DRAM Cell

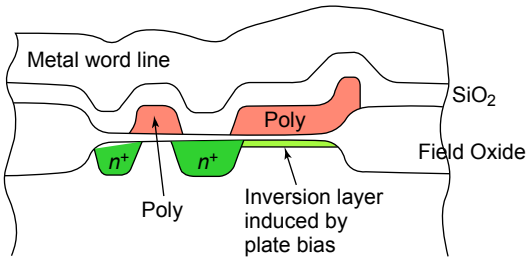


Write: C_S is charged or discharged by asserting WL and BL.
Read: Charge redistribution takes places between bit line and storage capacitance

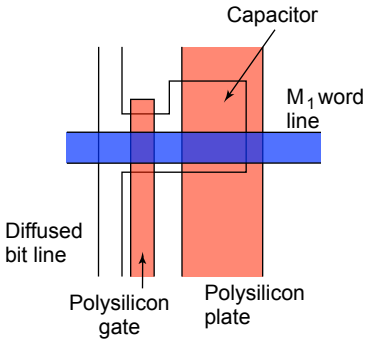
$$\Delta V = V_{BL} - V_{PRE} = V_{BIT} - V_{PRE} \frac{C_S}{C_S + C_{BL}}$$

Voltage swing is small; typically around 250 mV.

1T DRAM Cell



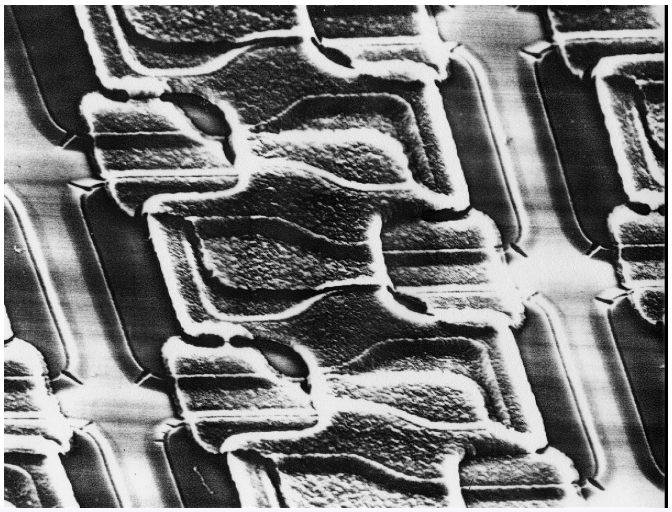
Cross-section



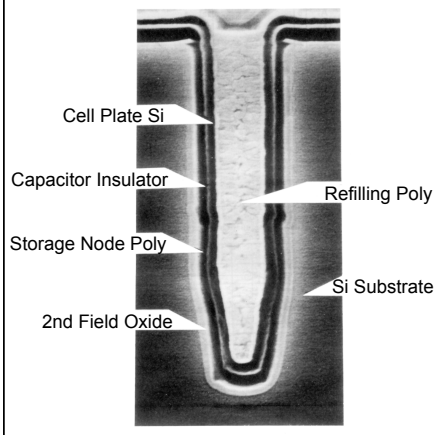
Layout

Uses Polysilicon-Diffusion Capacitance
Expensive in Area

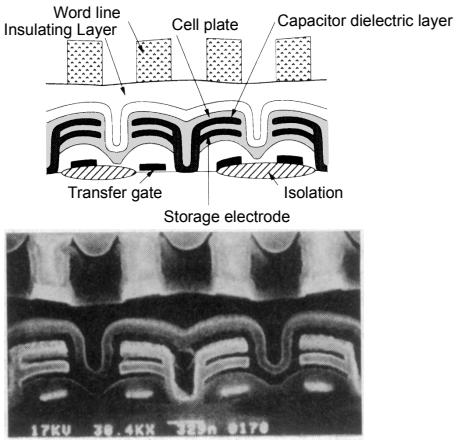
Micrograph of 1T DRAM



Advanced 1T DRAM Cells

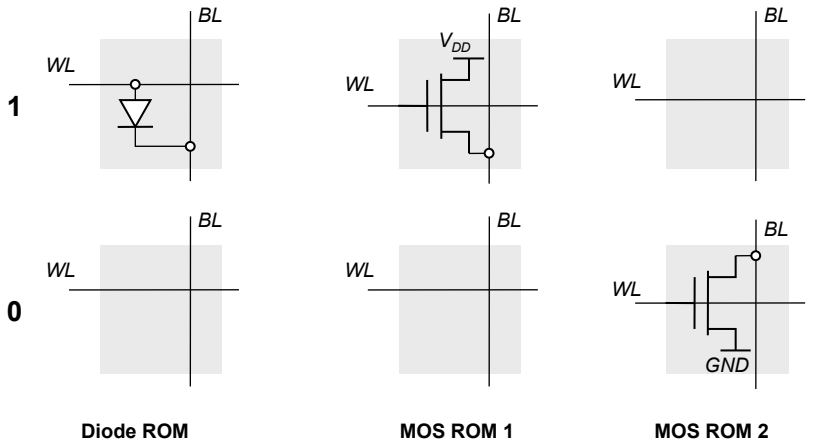


Trench Cell

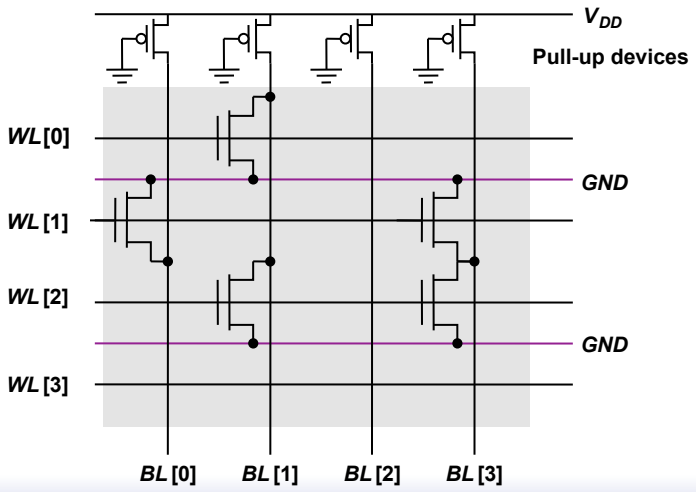


Stacked-capacitor Cell

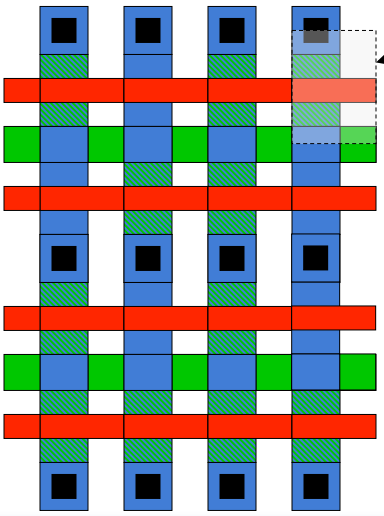
Read-Only Memory



MOS NOR ROM



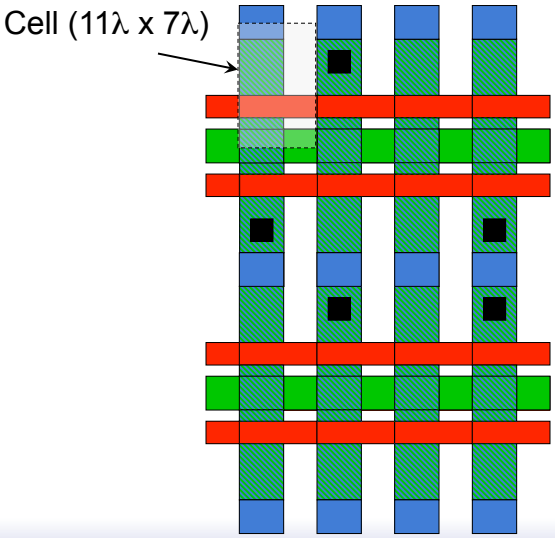
MOS NOR ROM Layout



Programmming using the Active Layer Only

Polysilicon
Metal1
Diffusion
Metal1 on Diffusion

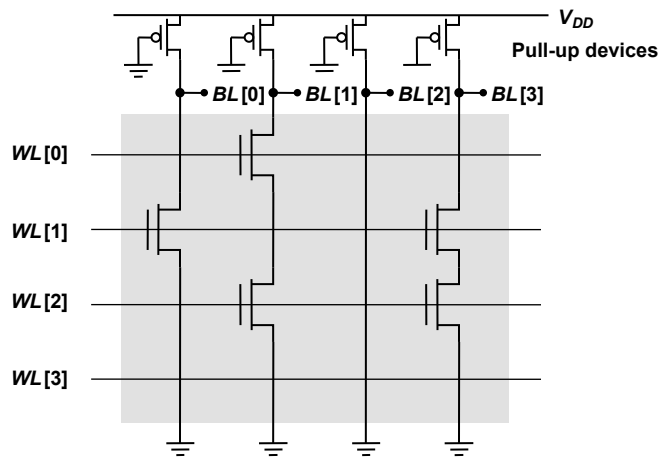
MOS NOR ROM Layout



Programmming using the Contact Layer Only

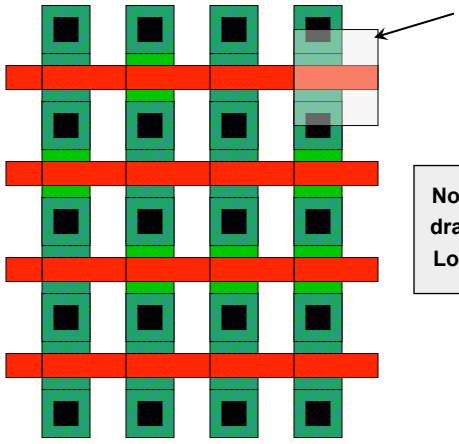
Polysilicon
Metal1
Diffusion
Metal1 on Diffusion

MOS NAND ROM



All word lines high by default with exception of selected row

MOS NAND ROM Layout



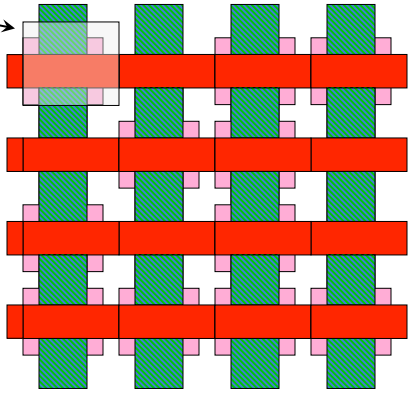
Programmming using the Metal-1 Layer Only

No contact to VDD or GND necessary; drastically reduced cell size
Loss in performance compared to NOR ROM

Polysilicon
Diffusion
Metal1 on Diffusion

NAND ROM Layout

Cell (5λ x 6λ)



Programmming using Implants Only

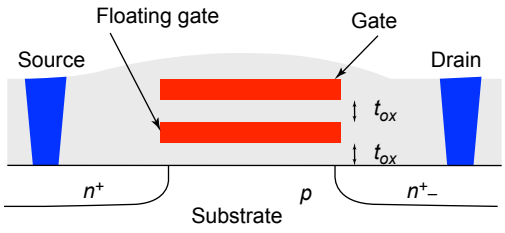
Polysilicon

Threshold-altering implant

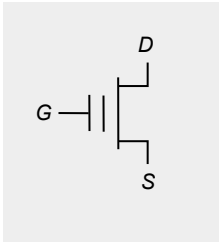
Metal1 on Diffusion

Non-Volatile Memories

The Floating-gate transistor (FAMOS)

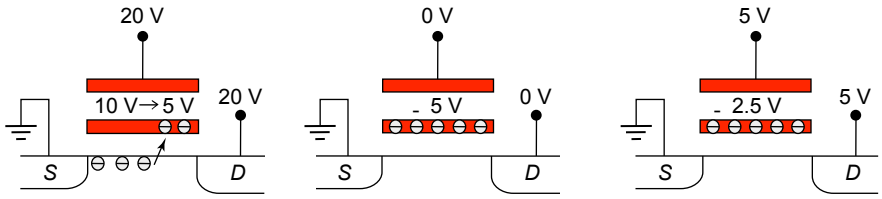


Device cross-section



Schematic symbol

Floating-Gate Transistor Programming

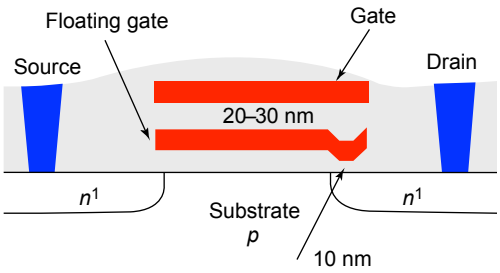


Avalanche injection

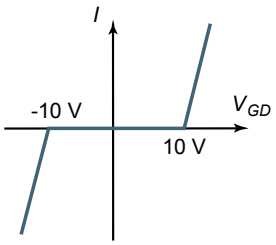
Removing programming voltage leaves charge trapped

Programming results in higher V_T.

FLOTOX EEPROM

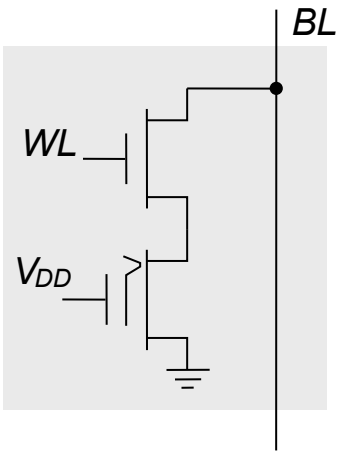


FLOTOX transistor



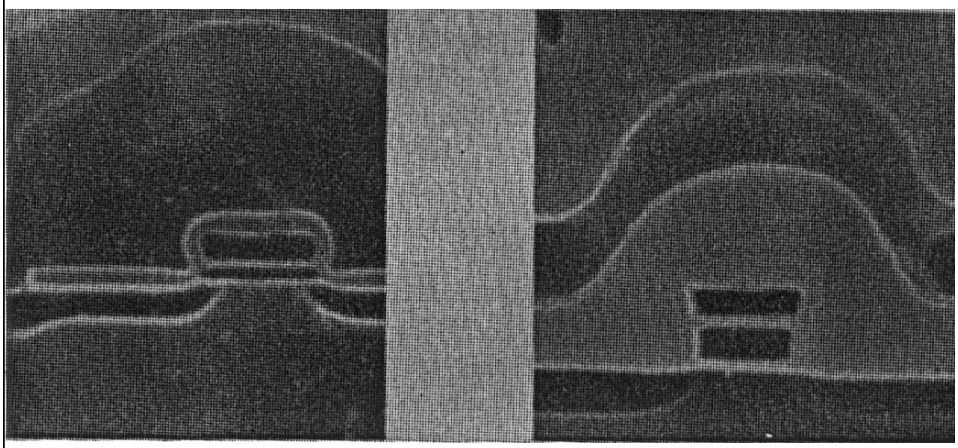
Fowler-Nordheim *I-V* characteristic

EEPROM Cell



Absolute threshold control is hard
Unprogrammed transistor might be depletion
⇨ 2 transistor cell

Cross Sections of NVM Cells



Flash EPROM

Courtesy Intel

Periphery

- Decoders
- Sense Amplifiers
- Input/Output Buffers
- Control / Timing Circuitry

Row Decoders

Collection of 2^M complex logic gates
Organized in regular and dense fashion

(N)AND Decoder

$$WL_0 = A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9$$

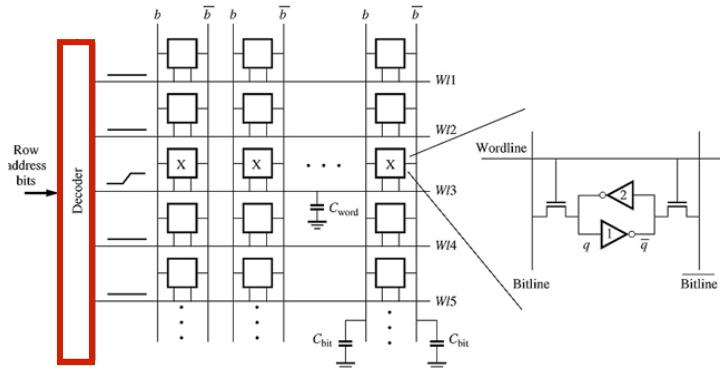
$$WL_{511} = \bar{A}_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9$$

NOR Decoder

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

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

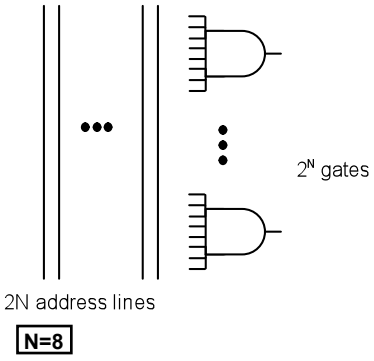
Decoder Design Example



- Look at decoder for 256x256 memory block (8KBytes)

Problem Setup

- Goal: Build fastest possible decoder with static CMOS logic
- What we know
  - Basically need 256 AND gates, each one of them drives one word line



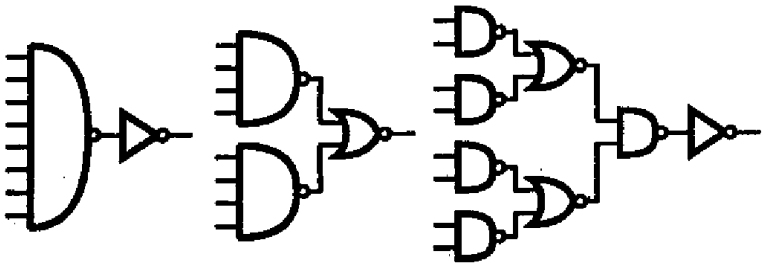
Problem Setup (1)

- Each word line has 256 cells connected to it
- Total output load is $256*C_{cell} + C_{wire}$
- Assume that decoder input capacitance is $C_{address}=4*C_{cell}$
- Each address drives $2^8/2$ AND gates
  - A0 drives ½ of the gates, A0_b the other ½ of the gates
- Neglecting C_{wire}, the fan-out on each one of the 16 address wires is: $BF = \frac{C_{load}}{C_{in}} = \frac{(2^8/2)(2^8 C_{cell})}{4C_{cell}} = 2^{13}$

Decoder Fan-Out

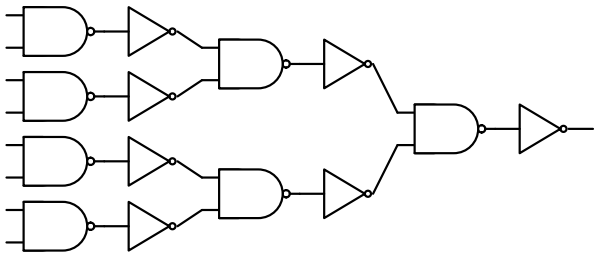
- FB of at least 2^{13} means that we will want to use more than $\log_4(2^{13}) = 6.5$ stages to implement the AND8
- Need many stages anyways
  - So what is the best way to implement the AND gate?
  - Will see next that it's the one with the most stages and least complicated gates

8-Input AND



LE=10/3 1	*LE*=2 5/3	*LE*=4/3 5/3 4/3 1
ΠLE = 10/3	ΠLE = 10/3	ΠLE = 80/27
P = 8 + 1	P = 4 + 2	P = 2 + 2 + 2 + 1

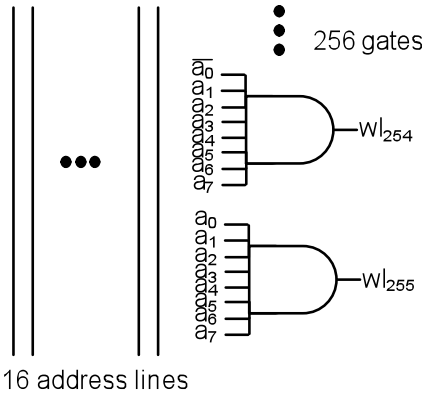
8-Input AND



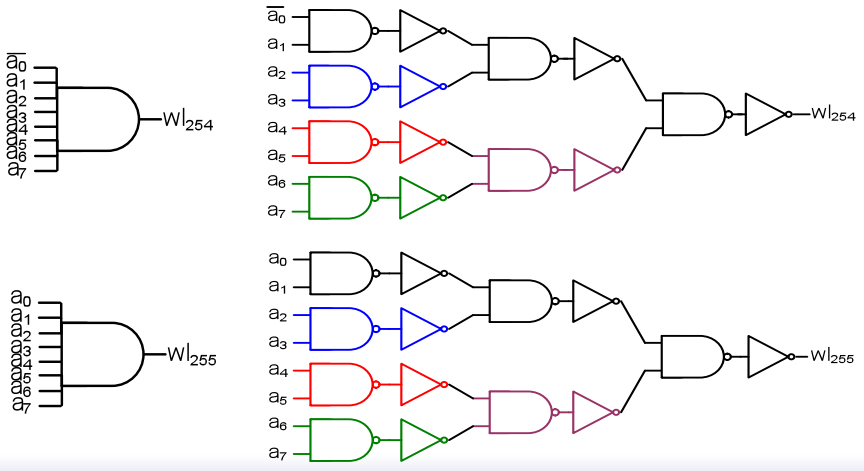
- Using 2-input NAND gates
  - 8-input gate takes 6 stages
- Total LE is $(4/3)^3 \approx 2.4$
- So PE is $2.4*2^{13}$ – optimal N of ~7.1

Decoder So Far

- 256 8-input AND gates
  - Each built out of tree of NAND gates and inverters
- Issue:
  - Every address line has to drive 128 gates (and wire) right away
  - Can't build gates small enough - Forces us to add buffers just to drive address inputs



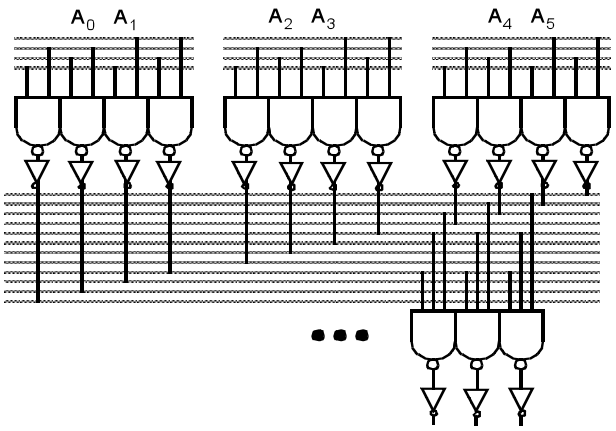
Look Inside Each AND8 Gate



Predecoders

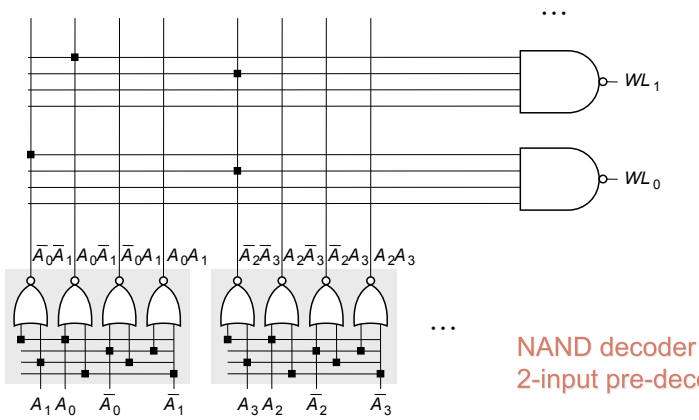
- Use a single gate for each of the shared terms
  - E.g., from A_0, $\overline{A_0}$, A_1, and $\overline{A_1}$, generate four signals: A_0A_1, $\overline{A_0}A_1$, $A_0\overline{A_1}$, $\overline{A_0}\overline{A_1}$
- In other words, we are decoding smaller groups of address bits first
  - And using the "predecoded" outputs to do the rest of the decoding

Predecoder and Decoder



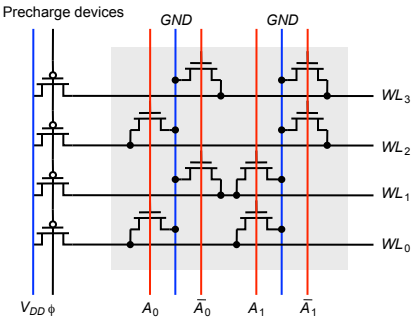
Hierarchical Decoders

Multi-stage implementation improves performance

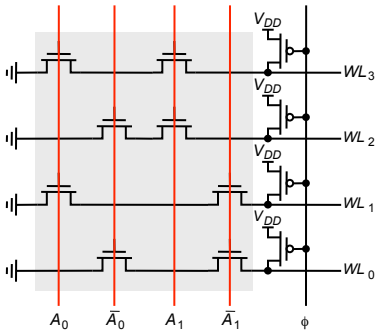


NAND decoder using
2-input pre-decoders

Dynamic Decoders

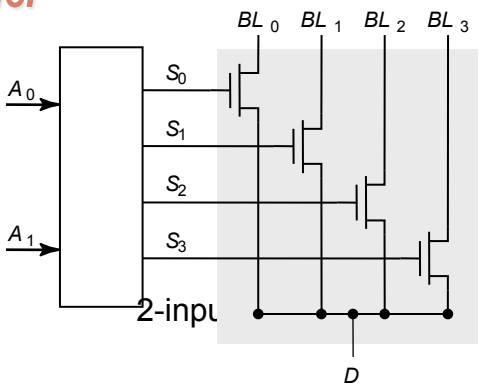


2-input NOR decoder



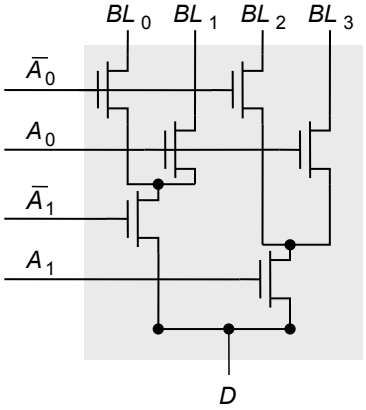
2-input NAND decoder

4-input pass-transistor based column decoder



Advantages: speed (t_{pd} does not add to overall memory access time)
Only one extra transistor in signal path
Disadvantage: Large transistor count

4-to-1 tree based column decoder



Number of devices drastically reduced
Delay increases quadratically with # of sections; prohibitive for large decoders
Solutions: buffers
progressive sizing
combination of tree and pass transistor approaches

Sense Amplifiers

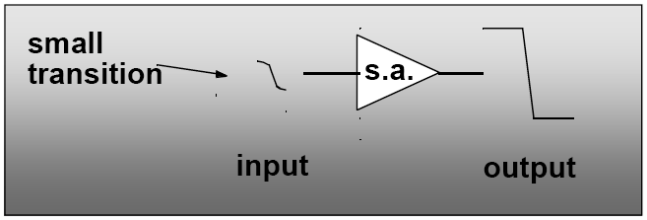
$$t_p = \frac{C \cdot \Delta V}{I_{av}}$$

make ΔV as small as possible

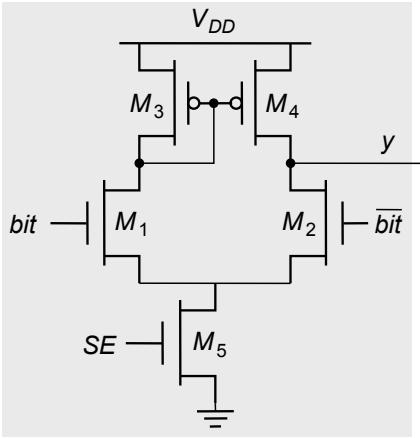
large

small

Idea: Use Sense Amplifer

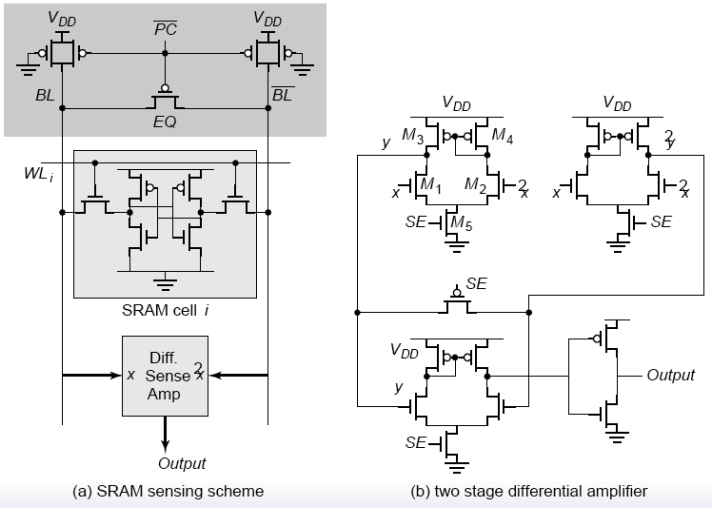


Differential Sense Amplifier

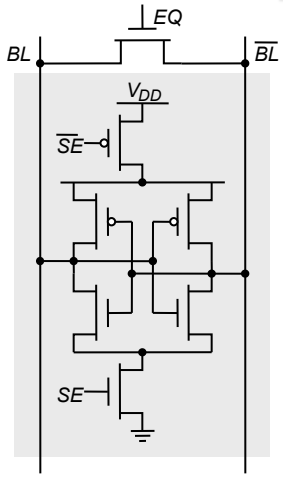


Directly applicable to SRAMs

Differential Sensing – SRAM



Latch-Based Sense Amplifier (DRAM)



Initialized in its meta-stable point with EQ

Once adequate voltage gap created, sense amp enabled with SE

Positive feedback quickly forces output to a stable operating point.