Chapter 12. Semiconductor Memory

Chapter Overview

- Memory Classification
- Memory Architectures
- The Memory Core
- Periphery
- Reliability

Introduction

- Memory is used for storage of data values and program instructions. E.g. cache memories, RAM/ROM, USB flash drive, etc.
- Dense data-storage circuitry is one of the primary concerns of a digital circuit or system designer
- Using register cells (e.g. DFF) for memory → excessively large area, not feasible
- Array structure is used for memory to increase storage density and reduce overhead caused by peripheral circuitry
- Memory design: robustness, performance, power

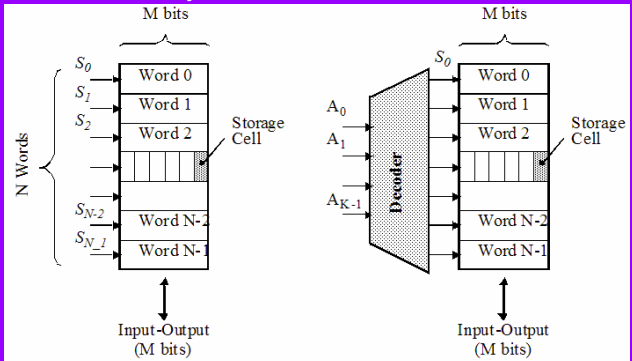
Semiconductor Memory Classification

- ROM: read-only memory, Data cannot be modified once written
  - ✓ Data is encoded into circuit topology
  - ✓ Nonvolatile, stored data is not lost when supply voltage is off
- RWM (read-write memory): data is stored in flip-flops (static) or as charge on capacitor (dynamic, needs refreshing),
  - ✓ Volatile (data is lost when supply voltage is off).
  - ✓ FIFO (first-in first-out), LIFO(last-in first-out), shift register, CAM (contents-addressable memory)
- NVRWM: nonvolatile read-write memory
  - ✓ EPROM: erasable programmable read-only memory
  - ✓ E2PROM: electrically programmable read-only memory
  - ✓ flash memory

RWM		NVRWM	ROM
Random Access	Non-Random Access	EPROM E^2PROM FLASH	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM		

Memory Architecture: Decoders

- Intuitive architecture for N×M memory: only one select signal S_i can be 1 at any time. If $N=2^{20}$, it needs $2^{20}=1048576$ select bits.
- Inserting address decoder reduces number of address bits to $K=\log_2 N$. For $N=2^{20}$, it only needs 20 address bits.



(a). Intuitive architecture for N×M memory

(b). A decoder reduces the number of address bits

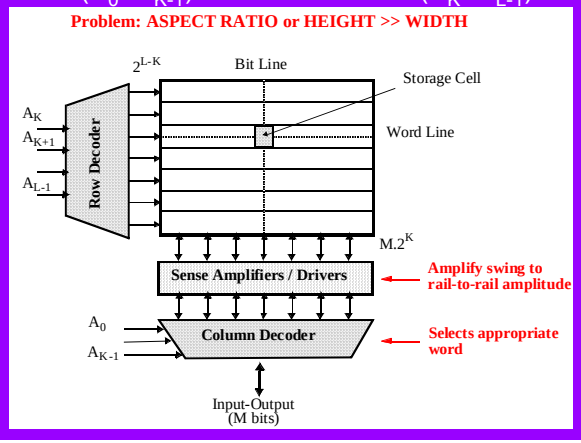
Figure. Architectures for N-word memory (each word is M bits)

N words => N select signals
Too many select signals

Decoder reduces # of select signals
$K = log_2 N$

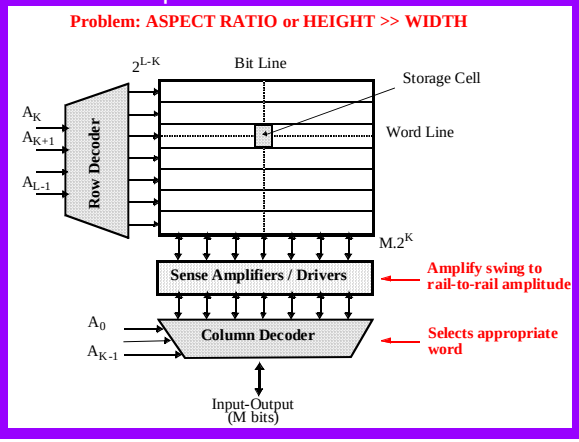
Array-Structured Memory Architecture

- Previous design has extremely large aspect ratio.
- Solution: arrange memory array so that vertical and horizontal dimensions are almost equal → aspect ratio≈1
  - ✓ multiple words stored in a single row and selected simultaneously
  - ✓ column decoder to route correct word to input/output terminals
  - ✓ column address (A_0~A_{K-1}) and row address (A_K~A_{L-1})



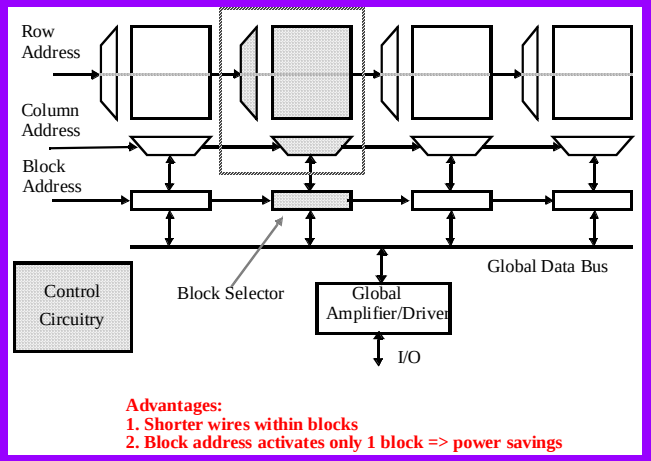
Array-Structured Memory Architecture

- Word line: horizontal select line that enables a single row of cells.
- Bit line: wire connecting cells in a column to input/output circuitry
- For smaller area, we reduce number of transistors in each cell
- Voltage swing on bit lines is reduced to substantially below V_{dd} to reduce delay and power → needs sense amplifier to recover internal swing to full rail-to-rail amplitude.



Hierarchical Memory Architecture

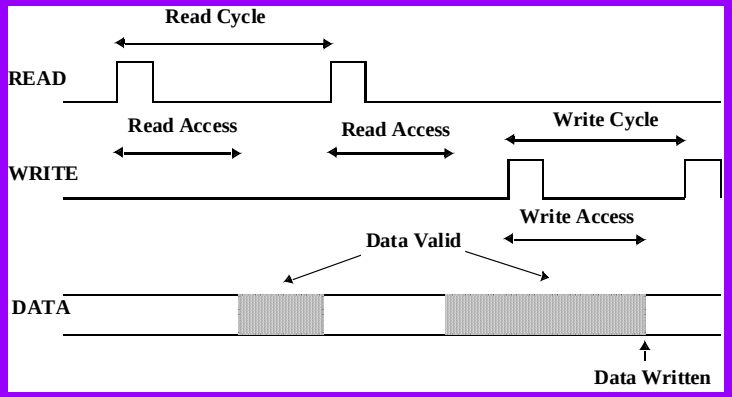
- Large memories suffer from speed degradation due to very long word and bit lines.
- Solution: partition memory to P smaller blocks
  - ✓ Extra block address select 1 of P blocks to be read/written.



Advantages:
1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

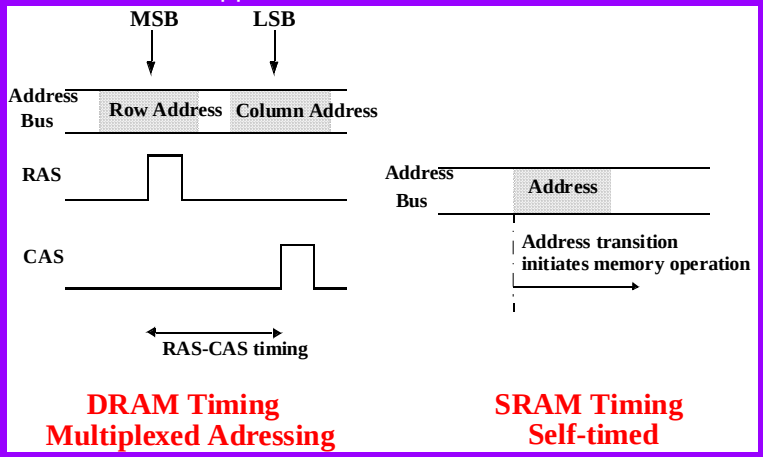
Memory Timing: Definitions

- Read-access time: delay between read request and the moment data is available at output.
- Write-access time: time elapsed between a write request and the final writing of input data into memory.
- (Read or write) cycle time: minimum time required to between successive reads or writes.



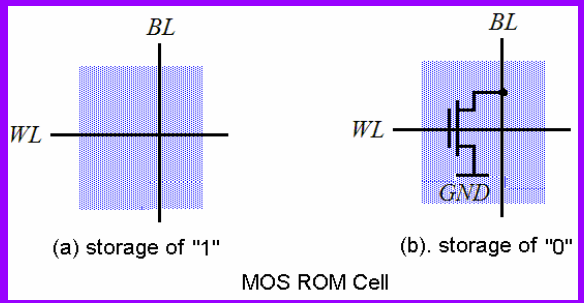
Memory Timing: Approaches

- DRAM timing: multiplexed addressing. Lower and upper halves of address words are presented sequentially on address bus.
  - ✓ RAS (row-access strobe): MSB part of address is present
  - ✓ CAS (column-access strobe): LSB part of address is present
- SRAM: self-timed approach



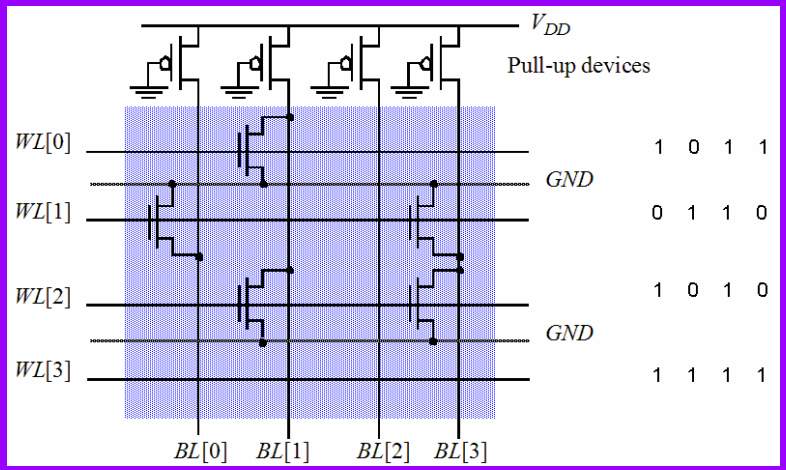
Memory Core Design: ROM Memories

- ROM cell design: 0 or 1 should be presented to bit line upon activation of its word line.
- MOS ROM cell design: bit line is resistively clamped to Vdd → default output=1.
- Absence of a transistor between WL and BL: WL=1, BL=1 (default value) → "1" is stored.
- Providing a MOS transistor between WL and BL: WL=1, transistor is ON, BL is shorted to GND → "0" is stored.



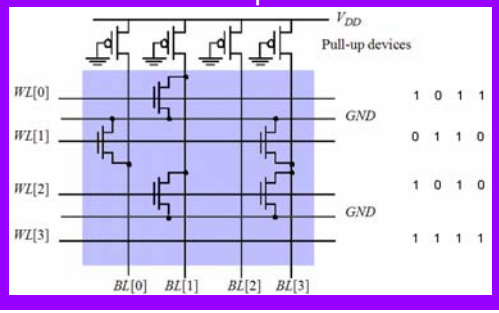
MOS NOR ROM

- Example 4×4 MOS ROM array
  - ✓ PMOS load is used to pull up bit lines in case none of attached NMOS devices is enabled.
  - ✓ GND lines are shared between 2 consecutive WL lines.



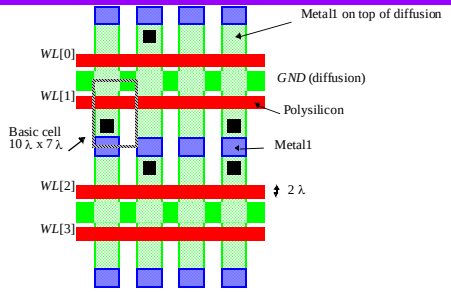
MOS NOR ROM

- MOS NOR ROM: combination of a bit line, PMOS pull-up and NMOS pull-downs constitutes pseudo-NMOS NOR gate with word lines as inputs → MOS NOR ROM.
- N×M ROM: a combination of M NOR gates with at most N inputs (for a fully populated column).
  - ✓ Only 1 word line goes "1" → at most 1 of pull-down devices is ON.
  - ✓ Resistance of pull-up device must be larger than pull-down resistance to ensure adequate low level of logic "0"



MOS NOR ROM Layout

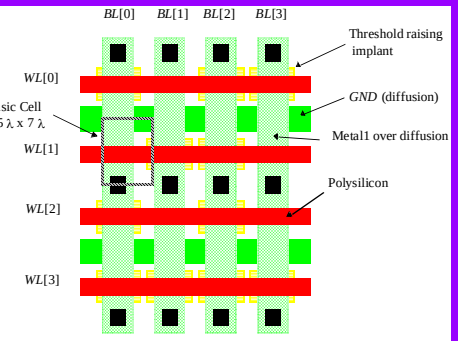
- Layout of 4×4 NOR ROM (contact-mask programming)
  - ✓ Constructed by repeating same cell in both X, Y directions
  - ✓ Mirroring odd cells around horizontal axis to share GND.
  - ✓ Memory is programmed by selective addition of metal-to-diffusion contacts
  - ✓ Presence of metal contact to BL: 0-cell; absence: 1-cell.



Only 1 layer (contact mask) is used to program memory array
Programming of the memory can be delayed to one of last process steps

MOS NOR ROM Layout

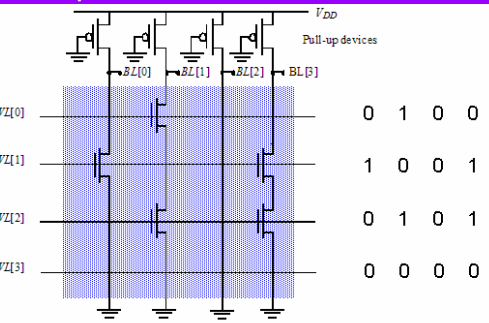
- Layout of 4×4 NOR ROM (threshold-raising programming)
  - ✓ Thresholds of some transistors are selectively raised to be higher than V_{dd} (e.g. to 7V for V_{dd}=5V) by implanting extra p-type impurities
  - ✓ Transistors with higher V_{th} can never be turned ON → equivalent to be eliminated.



Threshold raising implants disable transistors

MOS NAND ROM

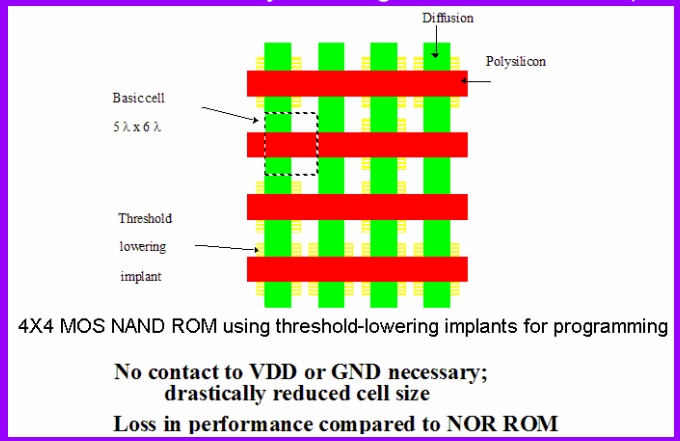
- MOS NAND ROM (word lines operated in reverse-logic)
  - ✓ All word lines are 1 by default, only for selected row WL=0
  - ✓ All transistors on nonselected rows are turned ON
  - ✓ If no transistor present on intersection, all other transistors are turned ON since WL=1→ output of BL=0.
  - ✓ If a transistor present at intersection, it's turned off due to selected WL=0 → output of BL=1



All word lines high by default with exception of selected row

MOS NAND ROM Layout

- Advantage of MOS NAND ROM: no contact to Vdd or Gnd is needed → cell size is reduced substantially.
- Eliminating a transistor means replacing it with short-circuit → threshold-lowering implant using n-type impurities (depletion transistor, always on regardless WL value)



4X4 MOS NAND ROM using threshold-lowering implants for programming

No contact to VDD or GND necessary; drastically reduced cell size

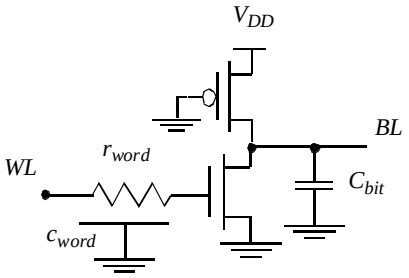
Loss in performance compared to NOR ROM

ROM Transient Performance

- Transient response of a memory array: the time it takes from the time a word line switches until the point where the bit line has traversed a certain voltage swing ΔV.
- Since bit line feeds into sense amplifier, it is not necessary to traverse full voltage swing → Generally ΔV=0.5V.
- Most of the delay of a memory array is attributable to interconnect parasitic resistance/capacitance.

Equivalent Transient Model for MOS NOR ROM

Model for NOR ROM



Word line parasitics

Resistance/cell: (7/2) x 10 Ω/q = 35 Ω

Wire capacitance/cell: $(7\lambda \times 2\lambda)$ $(0.6)^2$ 0.058 + 2 × $(7\lambda \times 0.6)$ × 0.043 = 0.65 fF

Gate Capacitance/cell: $(4\lambda \times 2\lambda)$ $(0.6)^2$ 1.76 = 5.1 fF.

Bit line parasitics:

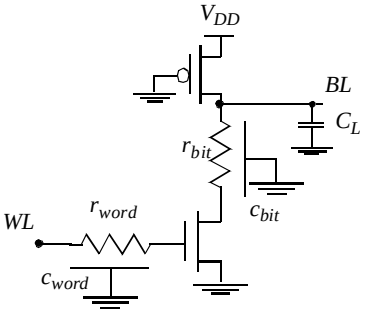
Resistance/cell: (8.5/4) x 0.07 Ω/q = 0.15 Ω (which is negligible)

Wire capacitance/cell: $(8.5\lambda \times 4\lambda)$ $(0.6)^2$ 0.031 + 2 × $(8.5\lambda \times 0.6)$ × 0.044 = 0.83 fF

Drain capacitance/cell: $((3\lambda \times 4\lambda)$ $(0.6)^2$ × 0.3 + 2 × 3λ × 0.6 × 0.8) × 0.375 + 4λ × 0.6 × 0.43 = 2.6 fF

Equivalent Transient Model for MOS NAND ROM

Model for NAND ROM



Word line parasitics:

Resistance/cell: (6/2) x 10 Ω/q = 30 Ω

Wire capacitance/cell: $(6\lambda \times 2\lambda)$ $(0.6)^2$ 0.058 + 2 × $(6\lambda \times 0.6)$ × 0.043 = 0.56 fF

Gate Capacitance/cell: $(3\lambda \times 2\lambda)$ $(0.6)^2$ 1.76 = 3.8 fF.

Bit line parasitics:

Resistance/cell: ~ 10 kΩ, the average transistor resistance over the range of interest.

Wire capacitance/cell: Included in diffusion capacitance

Source/Drain capacitance/cell: $((3\lambda \times 3\lambda)$ $(0.6)^2$ × 0.3 + 2 × 3λ × 0.6 × 0.8) × 0.375 + $(3\lambda \times 2\lambda)$ $(0.6)^2$ × 1.76 = 5.2 fF

Propagation Delay of NOR ROM

Word line delay

Consider the 512×512 case. The delay of the distributed *rc*-line containing *M* cells can be approximated using the expressions derived in Chapter 8.

$$t_{word} = 0.38\ (r_{word} \times c_{word})\ M^2 = 0.38\ (35\ \Omega \times (0.65 + 5.1)\ \text{fF})\ 512^2 = 20\ \text{nsec}$$

Bit line delay

Assume a (2.4/1.2) pull-down device and a (8/1.2) pull-up transistor. The bit line switches between 5 V and 2.5 V.

$$C_{bit} = 512 \times (2.6 + 0.8)\ \text{fF} = 1.7\ \text{pF}$$

$$I_{avHL} = 1/2\ (2.4/0.9)\ (19.6\ 10^{-6})((4.25)^2/2 + (4.25 \times 3.75 - (3.75)^2/2)) -$$
$$1/2\ (8/0.9)\ (5.3\ 10^{-6})\ (4.25 \times 1.25 - (1.25)^2/2) = 0.36\ \text{mA}$$

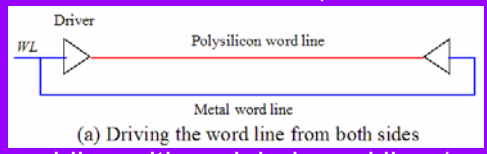
$$t_{HL} = (1.7\ \text{pF} \times 1.25\ \text{V}) / 0.36\ \text{mA} = 5.9\ \text{nsec}$$

The low-to-high response time can be computed using a similar approach.

$$t_{LH} = (1.7\ \text{pF} \times 1.25\ \text{V}) / 0.36\ \text{mA} = 5.9\ \text{nsec}$$

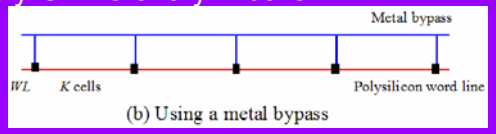
Decreasing Word Line Delay

- ❑ Word-line delay is larger than bit-line delay due to large resistance of poly-Si wire
- ❑ Methods to reduce word-line delay
- ✓ Partition word-line into multiple sessions and insert buffers
- ✓ Drive word-line from both ends (reduces delay by factor of 4)



(a) Driving the word line from both sides

- ✓ Bypass word-line with a global word line (metal wire) and connect to Poly-Si line every K cells.

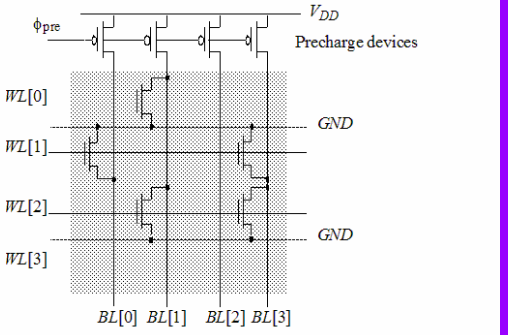


(b) Using a metal bypass

- ✓ Use other interconnect material: silicides (e.g. WSi_2). Al needs extra metal-to-Poly contacts → large area, not good.

Precharged MOS NOR ROM

- ❑ Disadvantages of previous NAND/NOR ROM
- ✓ ratioed logic: V_{OL} depends on ratio of pull-up/pull-down devices
- ✓ static power consumption: a static current path exists between V_{dd} and G_{nd} when output is low
- ❑ Solution: Use precharged logic (dynamic logic)



Precharged 4X4 MOS NOR ROM

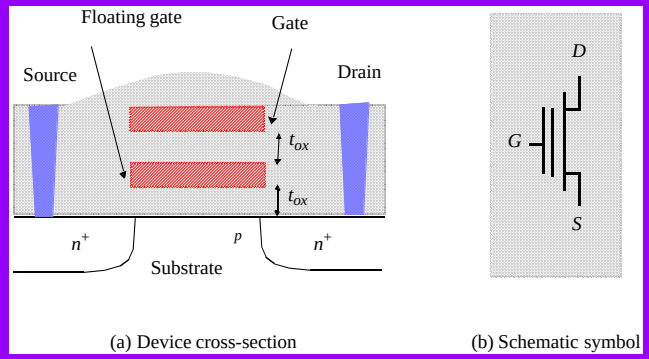
PMOS precharge device can be made as large as necessary, but clock driver becomes harder to design.

ROM Memories – A User Perspective

- ❑ Application-specific ROMs: memory is part of a custom design and programmed for that particular application only.
- ❑ Commodity ROM chips: vendor mass-produces memory and customized according to customer specifications.
- ✓ mask-programmable using contact or extra implant mask.
- ✓ programming involves manufacturer → undesirable delay
- ❑ PROM (Programmable ROM): allow customer to program the memory one time (write once) by fuses.
- ✓ a single error in programming makes the device unusable
- ❑ NVRW (Nonvolatile Read-Write Memory):
- ✓ Memory programming: selectively disabling/enabling memory cell by electrically altering transistor threshold
- ✓ Modified threshold is retained indefinitely even power is off
- ✓ Programmed value can be erased and reprogrammed.

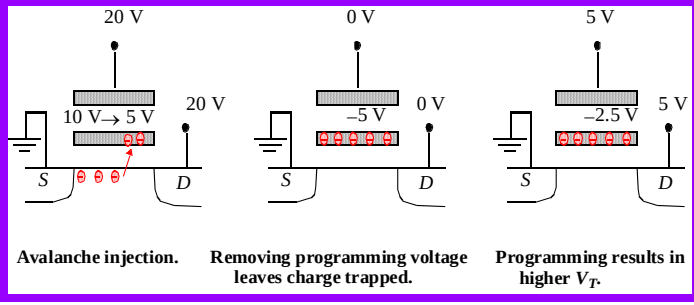
Nonvolatile RWM: Floating-gate Transistor (FAMOS)

❑ FAMOS: Floating-gate avalanche-injection MOS
✓ extra poly-Si strip inserted between gate and channel: floating gate.



Floating-Gate Transistor Programming

❑ Applying high voltage (15~20V) between S and G-D → high E-field → avalanche injection of electrons through oxide and get trapped on floating gate
❑ Removing voltage leaves induced negative charge in place → negative voltage on floating gate
❑ To turn on the device, a higher voltage is needed to overcome the effect of induced negative charge
❑ Threshold voltage is increased (~7V). A 5V voltage is not sufficient to turn on the transistor → transistor disabled.

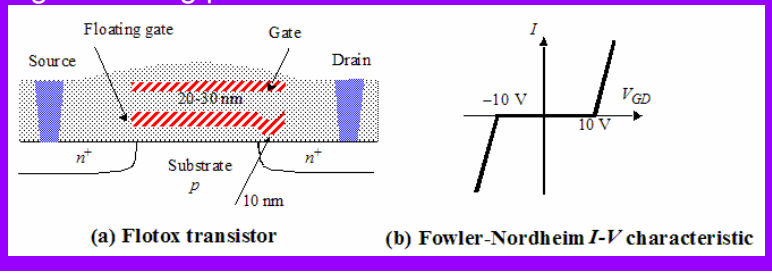


EPROM

❑ EPROM: Erasable-Programmable Read-Only Memory
✓ erased by shining UV light on cells through a transparent window in the package.
✓ UV light generate electron-hole pairs in oxide to make it slightly conductive.
✓ extremely simple and dense, good to fabricate large memories at low cost
❑ Disadvantages
✓ "off-system" erasure: memory must be removed from board and placed in EPROM programmer for programming
✓ slow erasure: 5-10μsec/word
✓ limited endurance: maximum 1000 erase/program cycles
✓ reliability issue: device threshold changes after many cycles

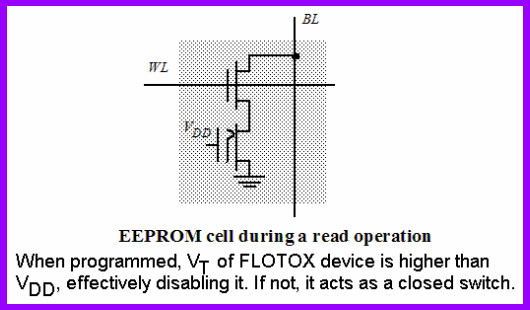
FLOTOX EEPROM

❑ EEPROM (E^2PROM): Electrically-Erasable Programmable Read-Only Memory
✓ use FLOTOX (floating-gate tunneling oxide) transistor
✓ a portion of dielectric separating the floating gate from the channel and drain is reduced in thickness to about 10nm
✓ Voltage of 10V ($E=10^9$V/m) applied over thin insulator → electrons travel to/from floating gate by Fowler-Nordheim tunneling
✓ Reversible → erased by reversing the voltage applied during the writing process



FLOTOX EEPROM

❑ Bidirectionality of FLOTOX EEPROM: injecting electrons onto floating gate raises V_T, while reverse operation lowers V_T.
❑ Resulting threshold voltage depends on initial gate charge→ threshold control problem
✓ Removing too much charge from floating gate results in depletion device → cannot be turned off when WL=0.
✓ Solution: add an extra transistor in series with FLOTOX as access device during read operation, while FLOTOX for storage.



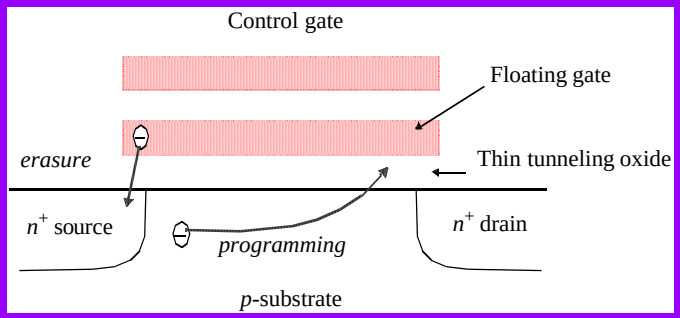
EEPROM cell during a read operation
When programmed, V_T of FLOTOX device is higher than V_{DD}, effectively disabling it. If not, it acts as a closed switch.

Flash EEPROM

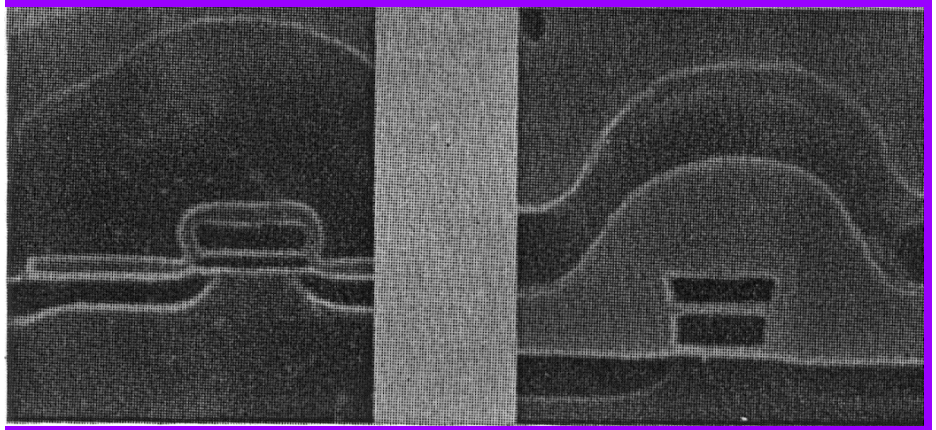
❑ EEPROM: large area (2 transistors in each cell), fabrication of very thin oxide is challenging and costly. But it has high versatility, more resilient against wear-out.
❑ Flash EEPROM: combines density of EPROM and versatility of EEPROM, with cost and functionality between the two.
❑ Flash EEPROM: programmed by hot-electron-injection, erasure by Fowler-Nordheim tunneling.
❑ Erasure performed in bulk for complete chip→ no need for extra access transistor of EEPROM.
❑ Monitoring control hardware checks threshold during erasure, dynamically adjusting the erasure time → no depletion device

Flash EEPROM

❑ ETOX flash cell (Intel): very thin tunneling gate oxide (10nm)
✓ different areas of gate oxide used for programming and erasure
✓ Programming: apply high voltage (12V) on gate and drain for a grounded source.
✓ Erasure: gate grounded and source at 12V.



Cross-sections of NVM cells



Flash **EPROM**

Courtesy Intel

Characteristics of State-of-the-art NVM

	EPROM [Tomita91]	EEPROM [Terada89, Pashley89]	Flash EEPROM [Jinbo92]
Memory size	16 Mbit (0.6 μm)	1 Mbit (0.8 μm)	16 Mbit (0.6 μm)
Chip size	7.18 x 17.39 mm^2	11.8 x 7.7 mm^2	6.3 x 18.5 mm^2
Cell size	3.8 μm^2	30 μm^2	3.4 μm^2
Access time	62 nsec	120 nsec	58 nsec
Erasure time	minutes	N.A.	4 sec
Programming time/word	5 μsec	8 msec/word, 4 sec /chip	5 μsec
Erase/Write cycles [Pashley89]	100	10^5	10^3-10^5

Read-Write Memories (RAM)

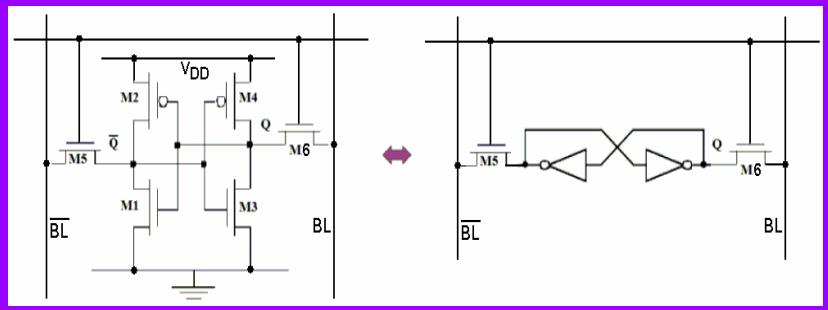
- **STATIC (SRAM)**
  - **Data stored as long as supply is applied**
  - **Large (6 transistors/cell)**
  - **Fast**
  - **Differential**
- **DYNAMIC (DRAM)**
  - **Periodic refresh required**
  - **Small (1-3 transistors/cell)**
  - **Slower**
  - **Single Ended**

6-transistor CMOS SRAM Cell

❑ Generic SRAM cell: 6 transistors.

✓ Access to cell is enabled by word line

✓ Two bit lines (BL and !BL) are required to improve noise margins during read and write operations.



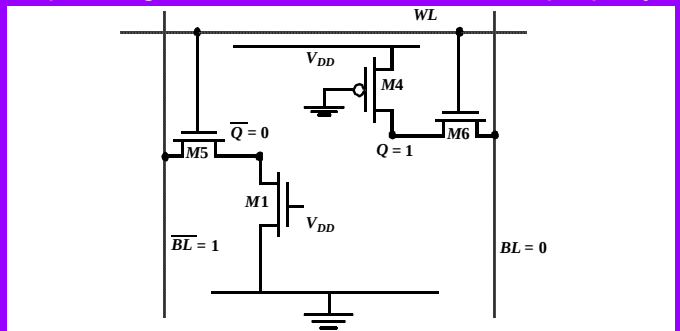
CMOS SRAM Analysis (Write)

❑ CMOS SRAM write operation

✓ Assume 1 is stored in the cell (Q=1)

✓ A 0 is written in the cell by setting BL=0 and !BL=1.

✓ flip-flop changes state if devices are sized properly.



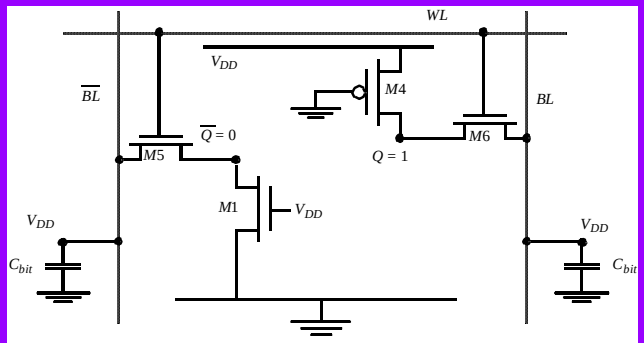
$$k_{n,M6}\left((V_{DD}-V_{Tn})\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right) = k_{p,M4}\left((V_{DD}-|V_{Tp}|)\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right)$$

$$(W/L)_{n,M6} \geq 0.33\,(W/L)_{p,M4}$$

$$\frac{k_{n,M5}}{2}\left(\frac{V_{DD}}{2}-V_{Tn}\right)^2 = k_{n,M1}\left((V_{DD}-|V_{Tn}|)\frac{V_{DD}}{2}-\frac{V_{DD}^2}{8}\right)$$

$$(W/L)_{n,M5} \geq 10\,(W/L)_{n,M1}$$

CMOS SRAM Analysis (Read)

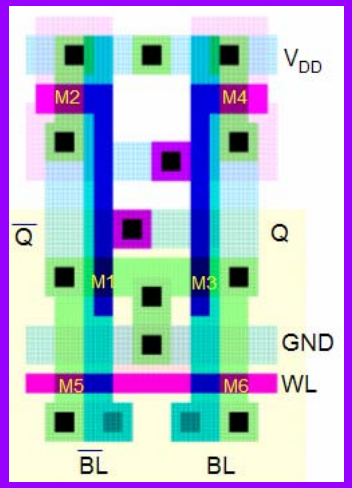


$$\frac{k_{n,M5}}{2}\left(\frac{V_{DD}}{2} - V_{Tn}\left(\frac{V_{DD}}{2}\right)\right)^2 = k_{n,M1}\left((V_{DD} - |V_{Tn}|)\frac{V_{DD}}{2} - \frac{V_{DD}^2}{8}\right)$$

$(W/L)_{n,M5} \leq 10\,(W/L)_{n,M1}$ **(supercedes read constraint)**

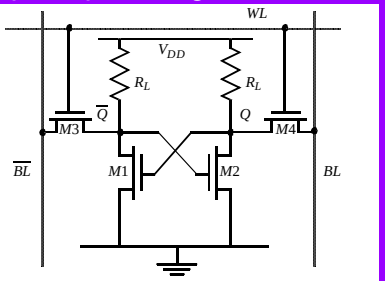
6T-SRAM — Layout

- ❑ CMOS SRAM with 6 transistors
- ✓ area-hungury: besides the devices, it needs signal routing and connections to two bit lines, a word line and both supply rails.
- ✓ Total area: 1092λ²



Resistance-load SRAM Cell

- ❑ Resistive load SRAM: four-transistors
- ✓ replace cross-coupled CMOS inverter pair by a pair of resistive-load NMOS inverters
- ✓ R_L must be as high as possible for reasonable noise margin NML and to reduce static power consumption
- ✓ But if R_L is very large, t_{pLH} and cell size are also increased.
- ✓ Solution for large tpLH: precharge bit lines to VDD



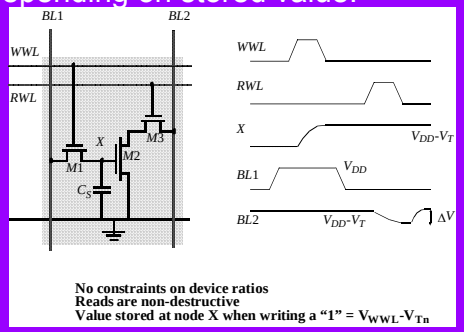
Static power dissipation -- Want R_L large
Bit lines precharged to V_{DD} to address t_p problem

Dynamic Random-Access Memory (DRAM)

- ❑ Dynamic Random-Access Memory (DRAM): based on charge storage on capacitor
- ❑ Needs continuous refresh operation to prevent the contents of memory cells from being corrupted by leakage.
- ❑ Refresh should occur every 1 to 4 msec.

3-Transistor DRAM Cell

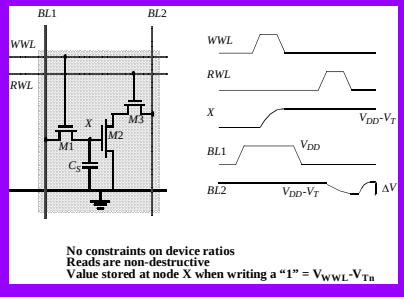
- 3T DRAM cell: resistive load SRAM → eliminating load resistors, remove redundancy of BL and !BL → 3 transistors.
- Two bit lines: BL1 (for write), BL2 (for read, get opposite value as stored data)
- Writing cell: placing appropriate data value on BL1 and asserting write-word line (WWL). Data is retained as charge on capacitor CS once WWL is lowered. Storage transistor M2 is either on or off depending on stored value.



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

3-Transistor DRAM Cell

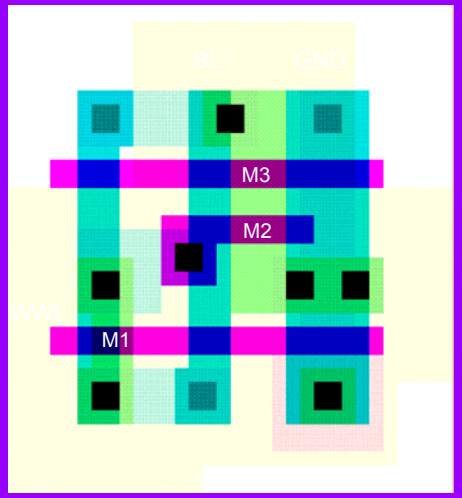
- 3T DRAM cell: When reading the cell, read-word line (RWL)=1, storage transistor M2 is on or off depending on the stored value.
- BL2 is either clamped to V_{DD} with a load device or precharged to either V_{DD} or V_{DD}-V_T.
- The series connection of M2 and M3 pulls BL2 low when a 1 is stored. BL2=1 in the opposite case.
- Cell is inverting: the inverse value of the stored signal is sensed on the bit line BL2.
- Refresh cells by reading the stored data, put its inverse on BL1 and assert WWL in consecutive order.



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

- Total 3T DRAM cell area: 576λ²



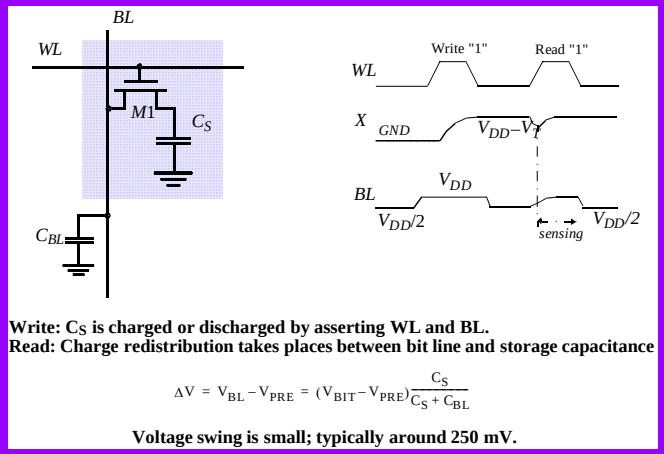
Properties of 3T-DRAM

- Properties of 3T-DRAM
- ✓ In contrast to SRAM cell, no constraints exist on device ratios. The choice of device sizes is solely based on performance and reliability considerations.
- ✓ In contrast to other DRAM cells, reading 3T-cell contents is nondestructive. That is, the data value stored in the cell is not affected by a read.
- ✓ The value stored on the storage node X when writing a 1 equals V_{WWL}-V_{Tn}

1-Transistor DRAM Cell

❑ 1-Transistor DRAM cell:

✓ Write: data value is placed on bit line BL, word line WL=1.Depending on the data value, cell capacitance CS is either charged or discharged.



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

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

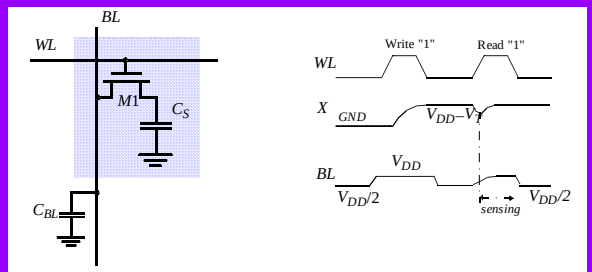
Voltage swing is small; typically around 250 mV.

1-Transistor DRAM Cell

❑ 1-Transistor DRAM cell:

✓ Before a read operation, BL is precharged to V_{PRE}. Upon WL=1, charge redistribution occurs between bit line and strorage capacitance.

✓ C_{BL}: bit-line capacitance. V_{BL}: potential of bit-line after charge redistribution. V_{BIT}: initial voltage on CS.



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.

DRAM Cell Observations

1T DRAM requires a sense amplifier for each bit line, due to charge redistribution read-out.

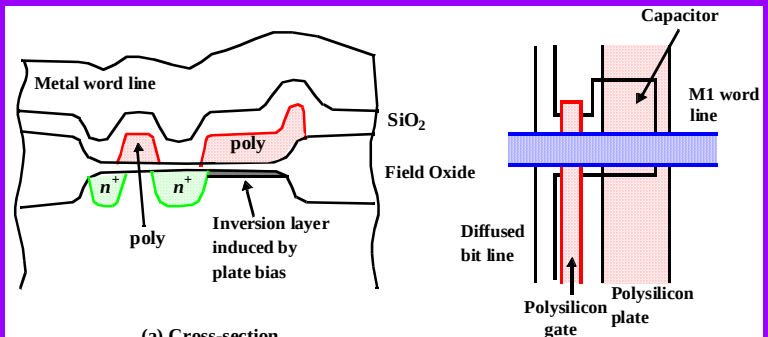
DRAM memory cells are single ended in contrast to SRAM cells.

The read-out of the 1T DRAM cell is destructive; read and refresh operations are necessary for correct operation.

Unlike 3T cell, 1T cell requires presence of an extra capacitance that must be explicitly included in the design.

When writing a "1" into a DRAM cell, a threshold voltage is lost. This charge loss can be circumvented by bootstrapping the word lines to a higher value than V_{DD}.

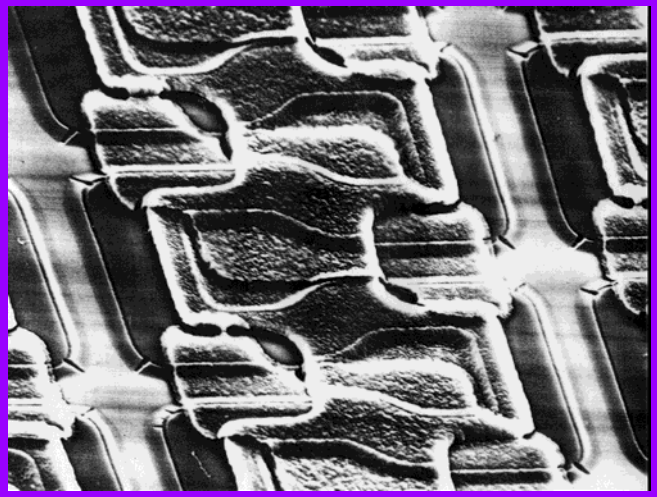
1-T DRAM Cell



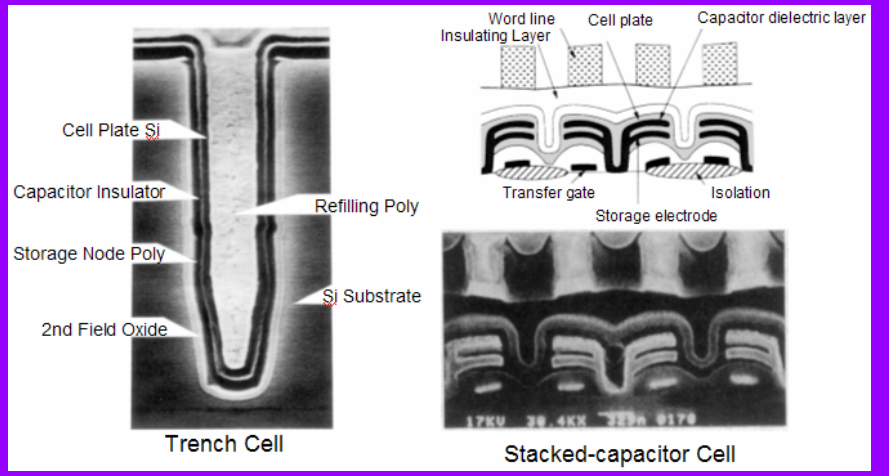
Used Polysilicon-Diffusion Capacitance

Expensive in Area

SEM of Poly-diffusion Capacitor 1T-DRAM



Advanced 1T DRAM Cells



Periphery

- ❑ Memory core trades performance and reliability for reduced area → needs peripheral circuitry to recover both speed and electrical integrity
- ❑ Memory peripheral circuitry
  - ✓ Decoders
  - ✓ Sense Amplifiers
  - ✓ Input/Output Buffers
  - ✓ Control/Timing Circuitry

Address Decoders - Row Decoders

- ❑ 1-out-of-2^M decoder: Collection of 2^M complex logic gates organized in regular and dense fashion
- ❑ Ex: 10-input (A_0 to A_9) address decoder
  - ✓ (N)AND decoder:

$$WL_0 = A_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$
$$WL_{511} = \overline{A}_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

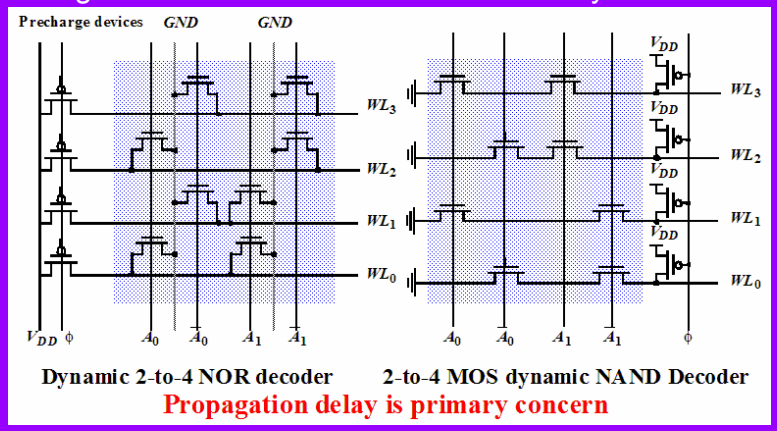
  - ✓ NOR decoder (single-stage CMOS design):

$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$
$$WL_{511} = \overline{A_0 + \overline{A}_1 + \overline{A}_2 + \overline{A}_3 + \overline{A}_4 + \overline{A}_5 + \overline{A}_6 + \overline{A}_7 + \overline{A}_8 + \overline{A}_9}$$

- ❑ For 10-input NOR decoder, if using pseudo-NMOS or dynamic gate for each row → totally 11×1024=11264 transistors.
- ❑ It can be implemented in a regular and dense fashion like ROM design

Dynamic Decoders with ROM Array Structure

- Dynamic 2-to-4 NOR decoder: word line of selected row is 1, all other word lines are 0.
- Dynamic 2-to-4 NAND decoder: word line of selected row is 0, all other word lines are 1. → "active low" signaling, put inverting buffer between decoder and memory.



Dynamic 2-to-4 NOR decoder 2-to-4 MOS dynamic NAND Decoder

Propagation delay is primary concern

Dynamic Decoders with ROM Array Structure

- NOR decoders are much faster, but consume more area and power than NAND decoders.
- Propagation delay:
  - ✓ R_{on} of NOR decoders are in parallel → smaller RC delay
  - ✓ R_{on} of NAND decoders are in series → larger RC delay
- Power consumption: after word lines are precharged to 1
  - ✓ NOR decoder: only 1 WL stays "1", all other WLs discharged to 0 → more power
  - ✓ NAND decoder: only 1 WL discharged to 0, all other WLs stay at "1" → less power

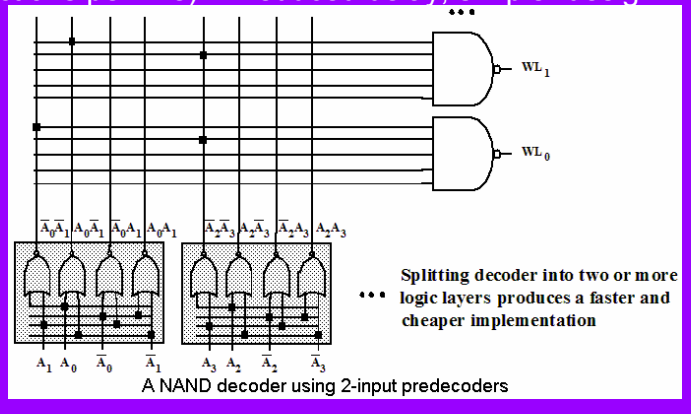
NAND Decoder Using 2-input Pre-decoders

- Decoder propagation delay: important to read/write access times
- To make large decoders faster: use predecoder which decodes segments of address in a first logic layer. A second layer of logic gates produces final word-line signals.
- Reason: splitting a complex gate into 2 or more logic layers often produces faster and cheaper implementation.
- Ex: 10-input NAND decoder:

$$WL_0 = \overline{\bar{A}_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9}$$
$$= \overline{(A_0 + A_1)(A_2 + A_3)(A_4 + A_5)(A_6 + A_7)(A_8 + A_9)}$$

NAND Decoder Using 2-input Pre-decoders

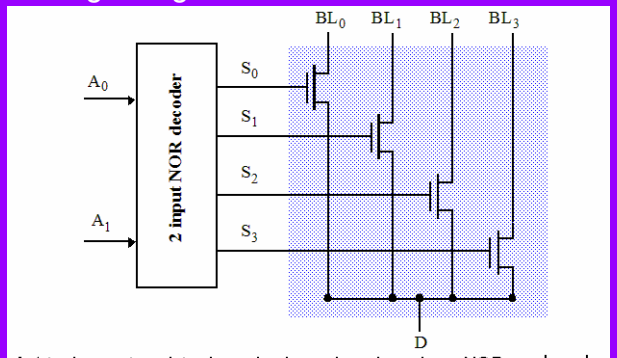
- NAND decoder using 2-input predecoders:
  - ✓ It reduces number of transistors required
  - ✓ Number of inputs to NAND gates is halved → delay reduced by factor of 4.
  - ✓ Load on vertical address lines is halved (only 256 connections per line) → reduced delay, simpler design.



A NAND decoder using 2-input predecoders

4-to-1 Pass-transistor Based Column Decoder

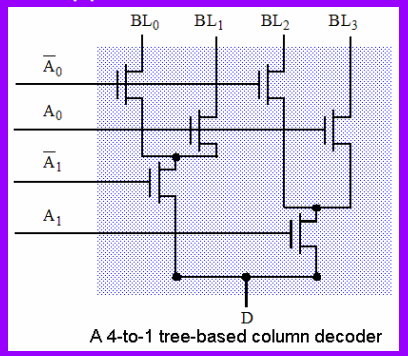
- Column/block decoder: a 2^K-input multiplexer (K: size of address word). Generally 2 implementations.
- CMOS pass-transistor MUX
- ✓ Advantage: speed (t_{pd} does not add to overall memory access time), only 1 extra transistor in signal path
- ✓ Disadvantage: large transistor count



A 4-to-1 pass-transistor-based column decoder using a NOR predecoder

4-to-1 Tree Based Column Decoder

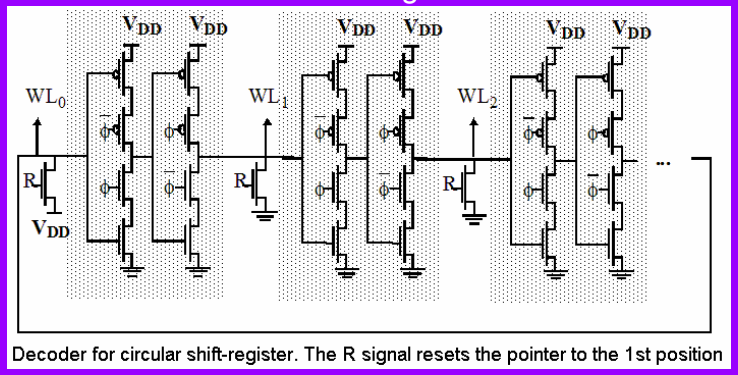
- Tree decoder: no predecoder is required.
- ✓ No. of devices required is drastically reduced.
  $N_{tree}=2^K+2^{K-1}+\ldots+4+2=2\times(2^K-1)$
- ✓ Disadvantage: delay increases quadratically with # of sections; prohibitive for large decoders.
- ✓ Solutions: buffers, progressive sizing, combination of tree and pass transistor approaches.



A 4-to-1 tree-based column decoder

Decoder for Non-Random-Access Memories

- Non-random-access memory: no need for full decoder.
- Example: in a serial-access memory, decoder degrades to M-bit shift-register with M the number of rows.
- ✓ only 1 of the bits is "1" at a time (pointer)
- ✓ Pointer moves to next position after every access operation
- ✓ C^2MOS D-FF is used in shift register



Decoder for circular shift-register. The R signal resets the pointer to the 1st position

Sense Amplifiers

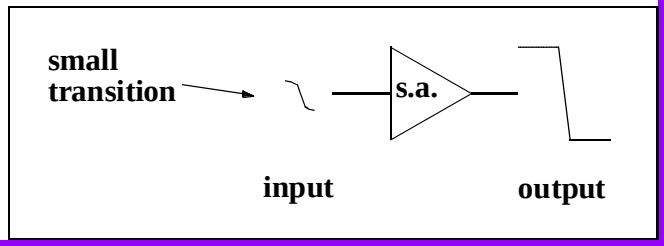
$$t_p = \frac{C \cdot \Delta V}{I_{av}}$$

make ΔV as small as possible

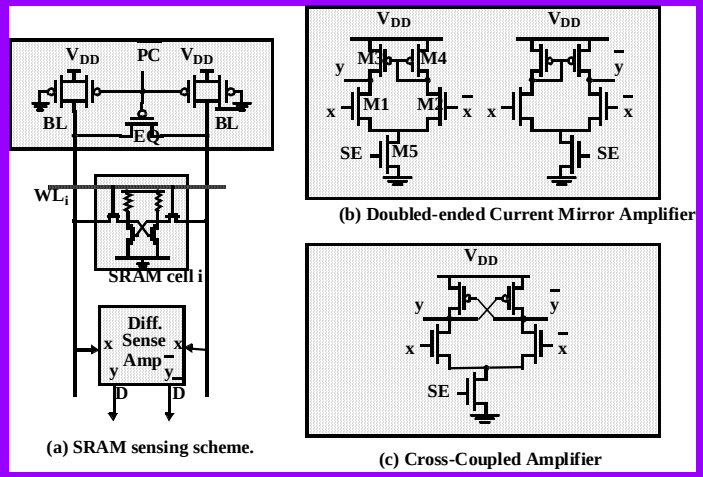
large

small

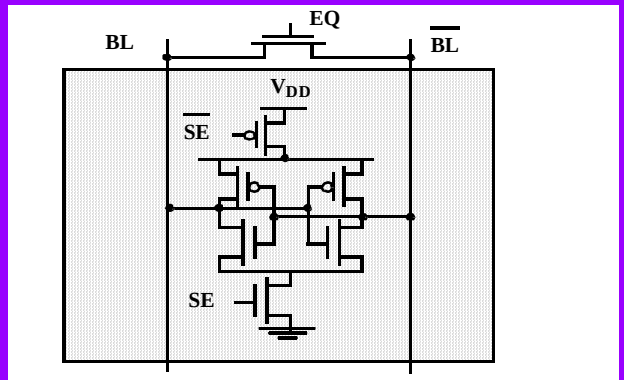
Idea: Use Sense Amplifer



Differential Sensing - SRAM

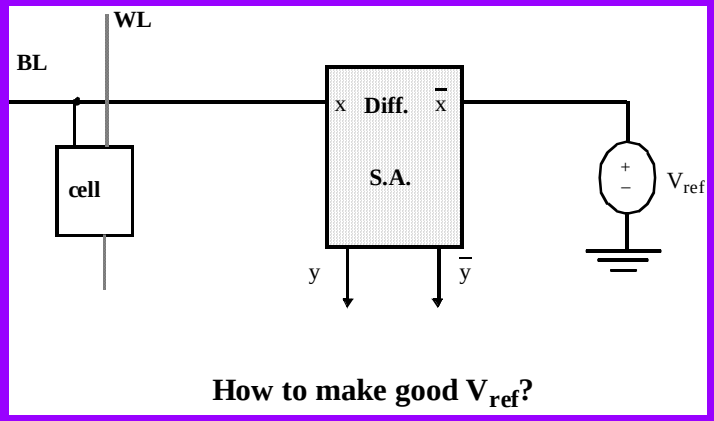


Latch-Based Sense Amplifier



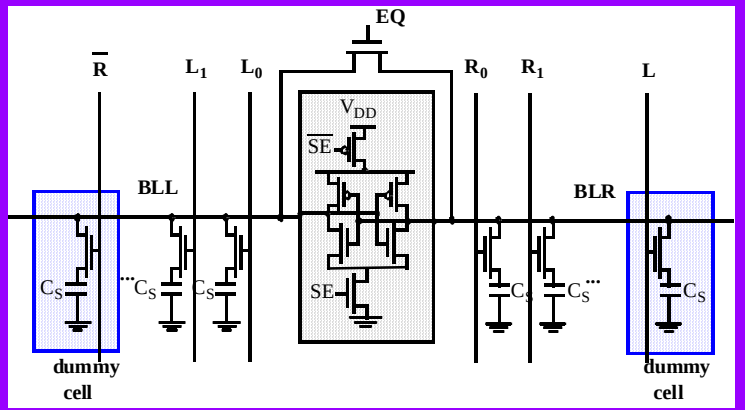
Initialized in its meta-stable point with EQ
Once adequate voltage gap created, sense amp enabled with SE
Positive feedback quickly forces output to a stable operating point.

Single-to-Differential Conversion

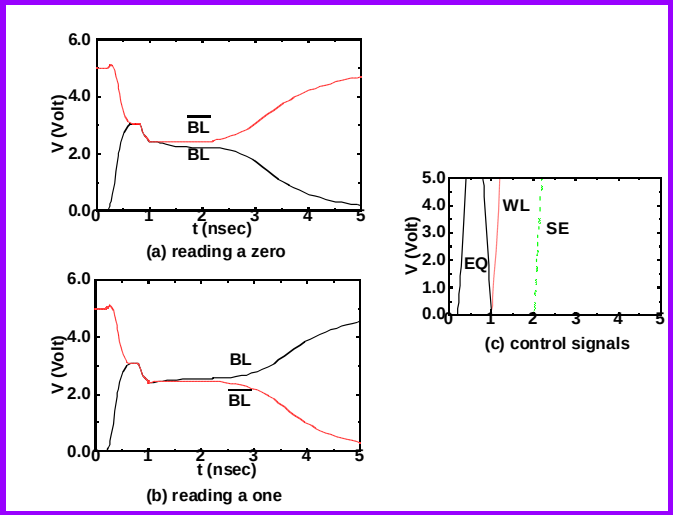


How to make good V_{ref}?

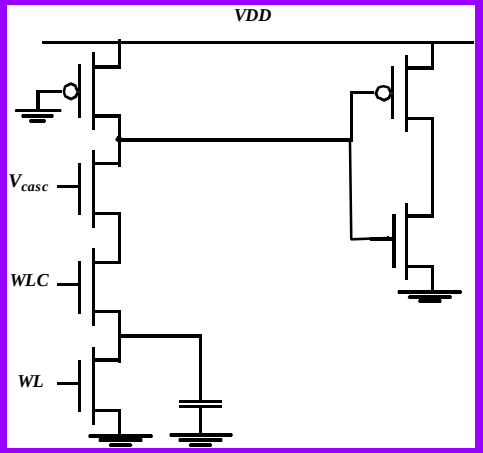
Open Bitline Architecture



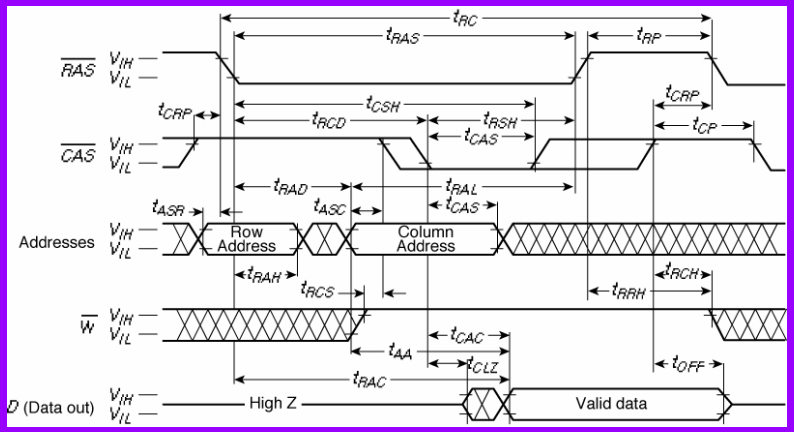
DRAM Read Process with Dummy Cell



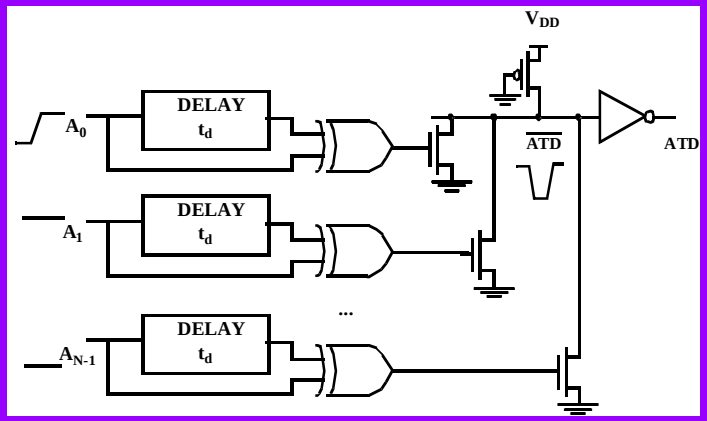
Single-Ended Cascode Amplifier



DRAM Timing



Address Transition Detection



Reliability and Yield

- Semiconductor memories trade off noise-margin for density and performance

Highly Sensitive to Noise (Crosstalk, Supply Noise)

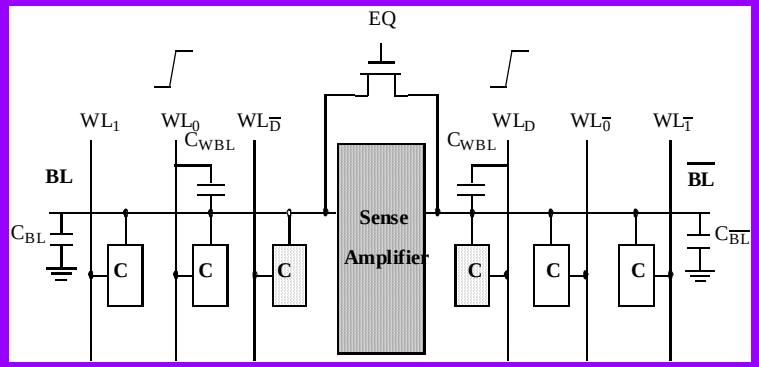
- High Density and Large Die size cause Yield Problems

$$Y = 100\frac{Number\ of\ Good\ Chips\ on\ Wafer}{Number\ of\ Chips\ on\ Wafer}$$

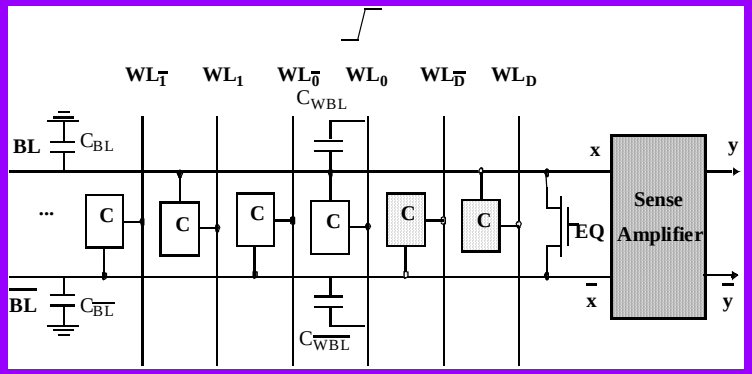
$$Y = \left[\frac{1 - e^{-AD}}{AD}\right]^2$$

Increase Yield using Error Correction and Redundancy

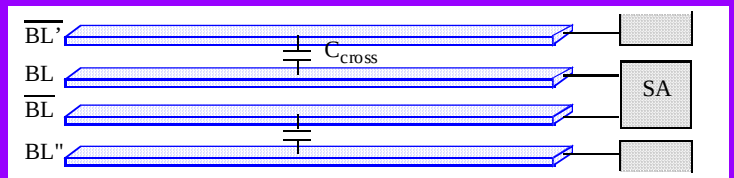
Open Bit-line Architecture —Cross Coupling



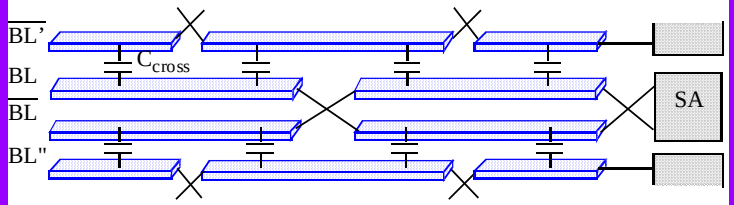
Folded-Bitline Architecture



Transposed-Bitline Architecture

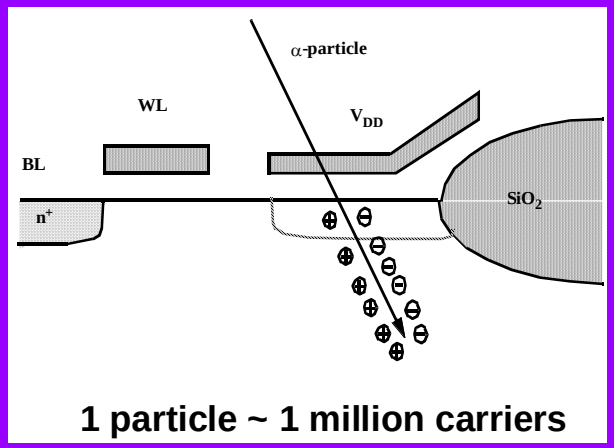


(a) Straightforward bitline routing.



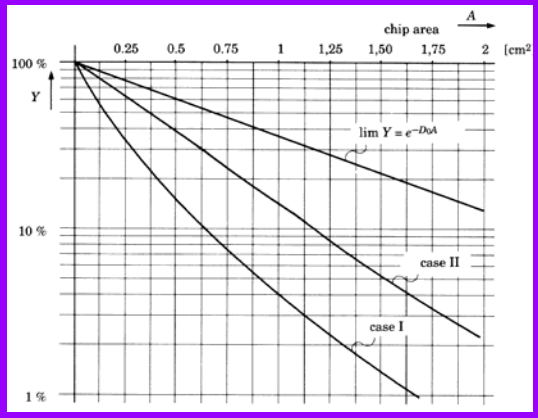
(b) Transposed bitline architecture.

Alpha-particles



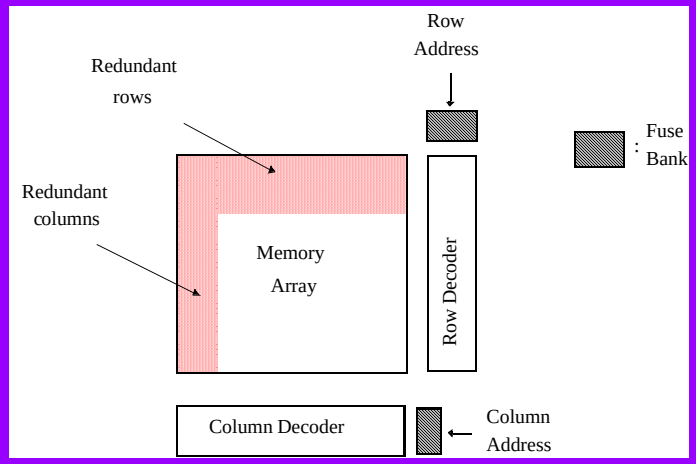
1 particle ~ 1 million carriers

Yield

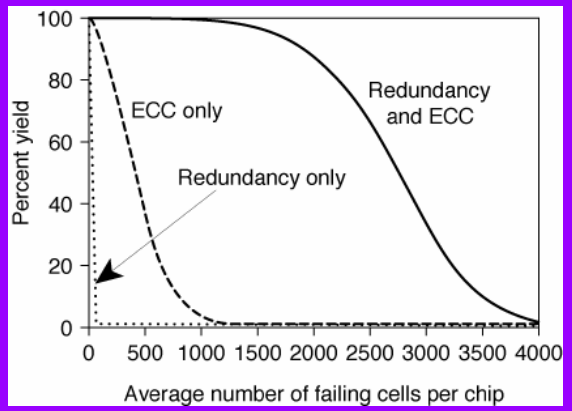


Yield curves at different stages of process maturity (from [Veendrick92])

Redundancy

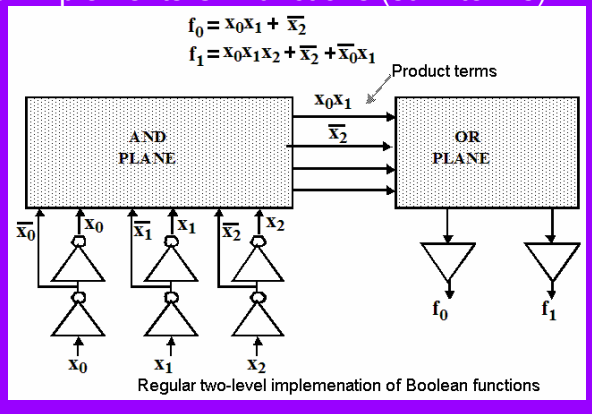


Redundancy and Error Correction



Programmable Logic Array

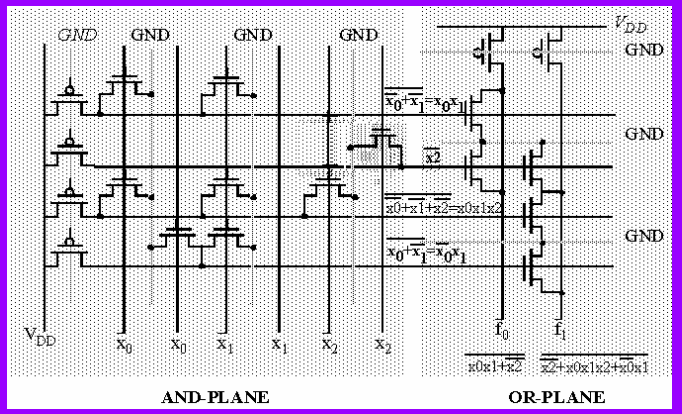
- Programmable logic array (PLA): using two-level sum-of-products representation of logic function
- Regular 2-level implementation of Boolean functions:
- ✓ 1st layer of gates implements AND operations (product-terms or min-terms)
- ✓ 2nd layer implements OR functions (sum-terms)



Regular two-level implemenation of Boolean functions

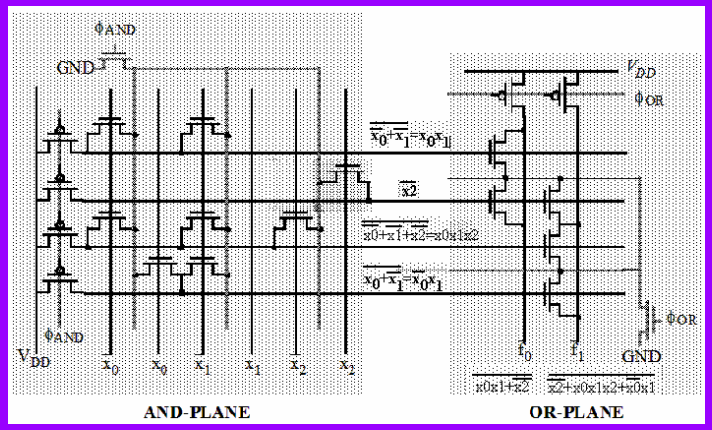
Pseudo-Static PLA

- Left portion: AND-plane, each row is a pseudo-NMOS NOR gate to implement a product term. $a \cdot b = \overline{\overline{a}+\overline{b}}$
- Right portion: OR-plane, each column is a pseudo-NMOS NOR gate (hence an extra inverter is needed) to implement a sum term. $a+b = inv(\overline{a+b})$

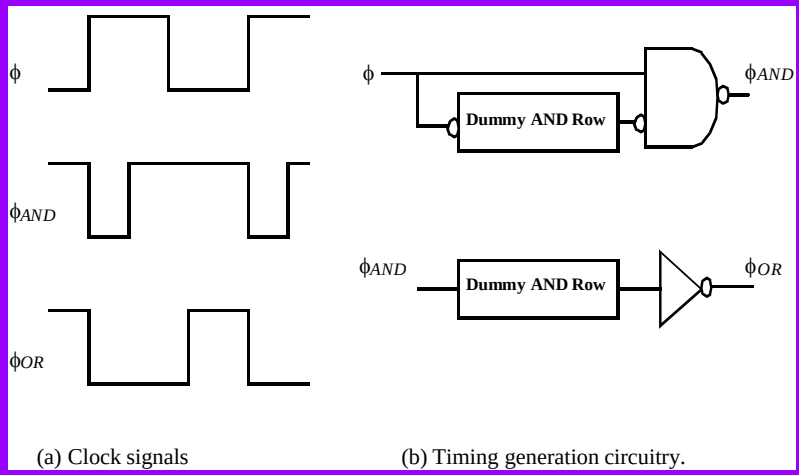


Dynamic PLA

- Left portion: AND-plane, each row is a dynamic ΦN NOR gate to implement a product term. $a \cdot b = \overline{\overline{a}+\overline{b}}$
- Right portion: OR-plane, each column is a dynamic ΦN NOR gate (hence an extra inverter is needed) to implement a sum term. $a+b = inv(\overline{a+b})$

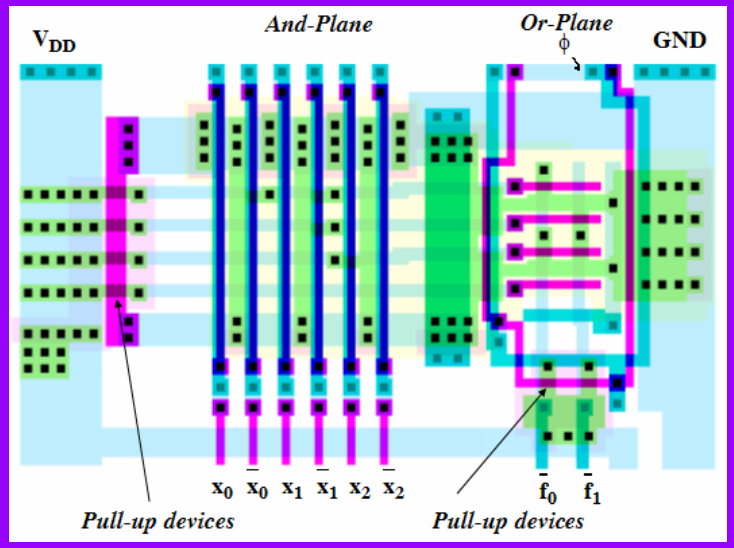


Clock Signal Generation for self-timed dynamic PLA



(a) Clock signals (b) Timing generation circuitry.

PLA Layout



PLA versus ROM

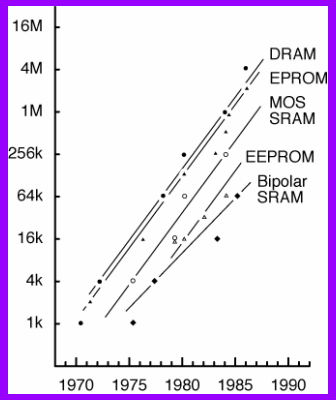
Programmable Logic Array
- **structured approach to random logic**
- **"two level logic implementation"**
  - **NOR-NOR (product of sums)**
  - **NAND-NAND (sum of products)**

IDENTICAL TO ROM!

Main difference
- **ROM: fully populated**
- **PLA: one element per minterm**

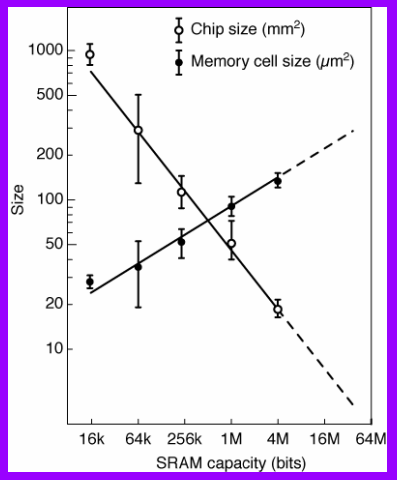
Note: Importance of PLA's has drastically reduced
1. **slow**
2. **better software techniques (mutli-level logic synthesis)**

Semiconductor Memory Trends



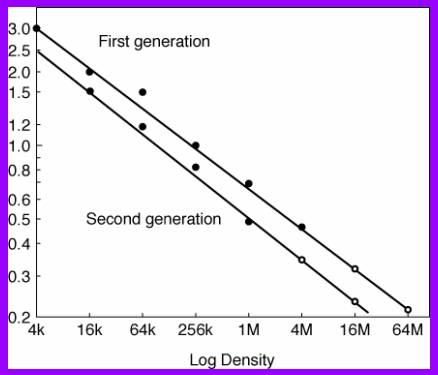
Memory Size as a function of time: x 4 every three years

Semiconductor Memory Trends



Increasing die size
factor 1.5 per generation
Combined with reducing cell size
factor 2.6 per generation

Semiconductor Memory Trends



Technology feature size for different SRAM generations