Topic 7

Memory and Array Circuits

Peter Y. K. Cheung
Department of Electrical & Electronic Engineering
Imperial College London

URL: http://www.ee.ic.ac.uk/pcheung/

Based on slides/material by…

- K. Masselos http://cas.ee.ic.ac.uk/~kostas
- J. Rabaey http://bwrc.eecs.berkeley.edu/Classes/IcBook/instructors.html
  "Digital Integrated Circuits: A Design Perspective", Prentice Hall
- D. Harris http://www.cmosvlsi.com/coursematerials.html
  Weste and Harris, "CMOS VLSI Design: A Circuits and Systems Perspective", Addison Wesley

Recommended Reading:

- J. Rabaey et. al. "Digital Integrated Circuits: A Design Perspective": Chapter 12
- Weste and Harris, "CMOS VLSI Design: A Circuits and Systems Perspective": Chapter 11

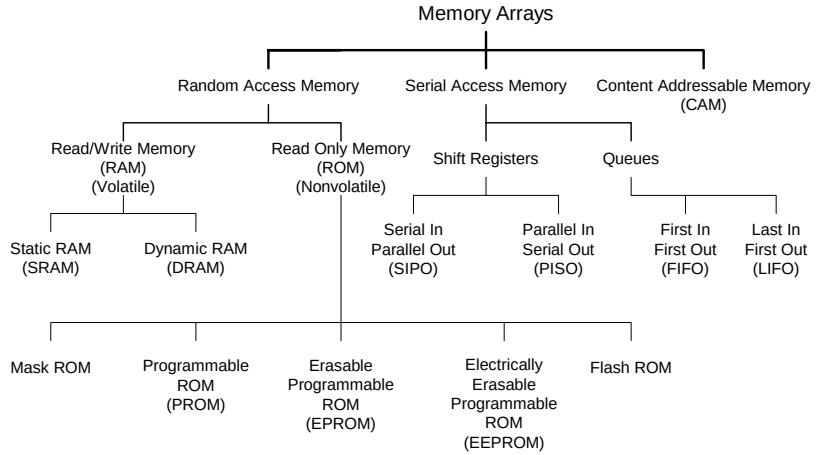
Outline

- **Memory classification**
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Semiconductor Memory Classification

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

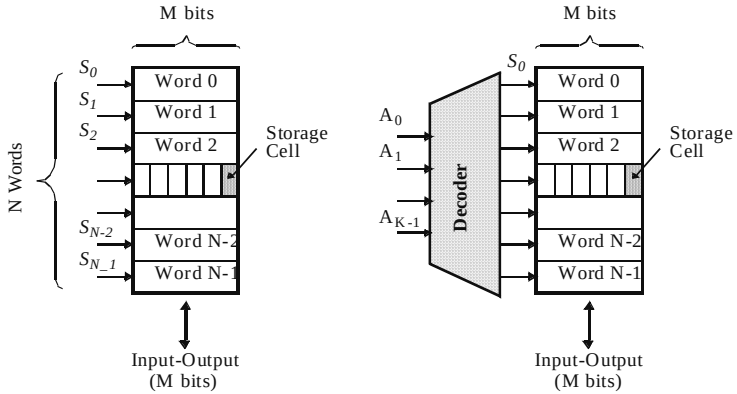
Memory Arrays



Outline

- Memory classification
- **Basic building blocks**
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Memory Architecture: Decoders

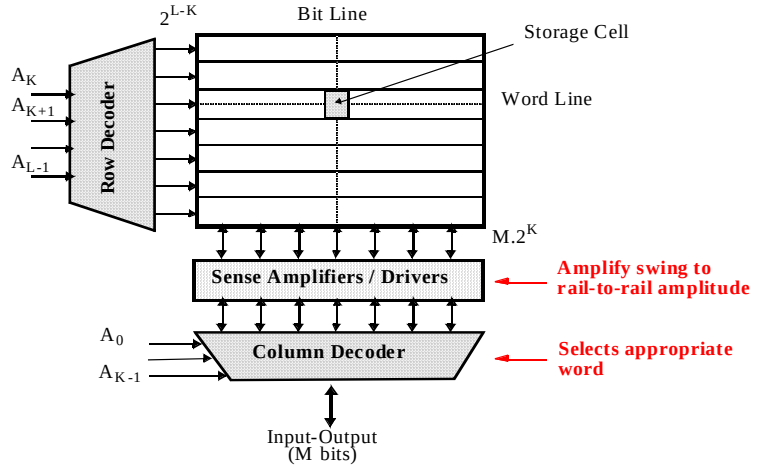


N words => N select signals
Too many select signals

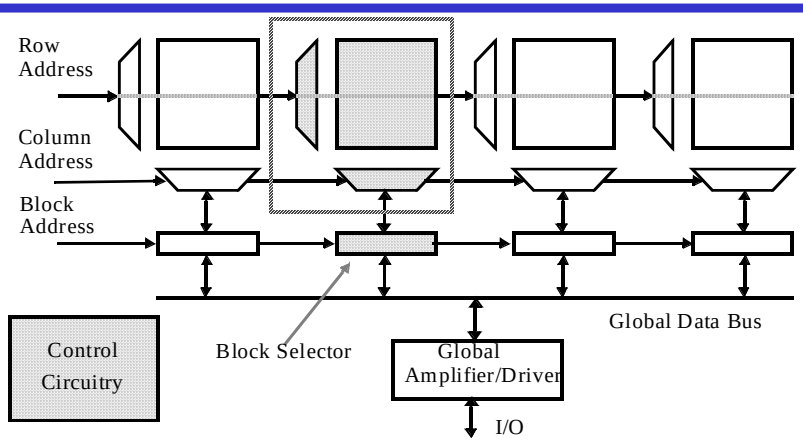
Decoder reduces # of select signals
$K = log_2 N$

Array-Structured Memory Architecture

Problem: ASPECT RATIO or HEIGHT >> WIDTH

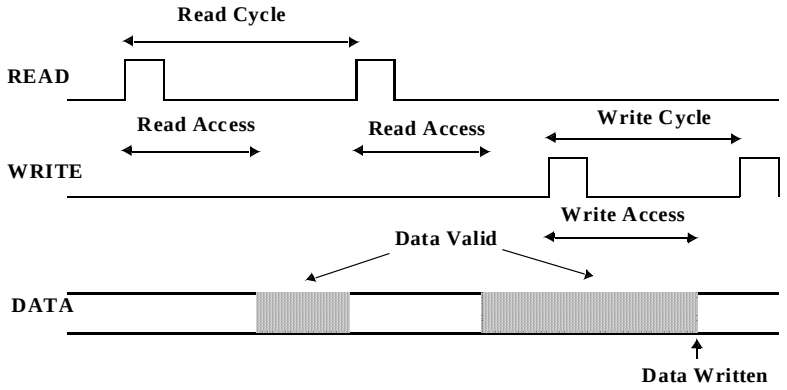


Hierarchical Memory Architecture

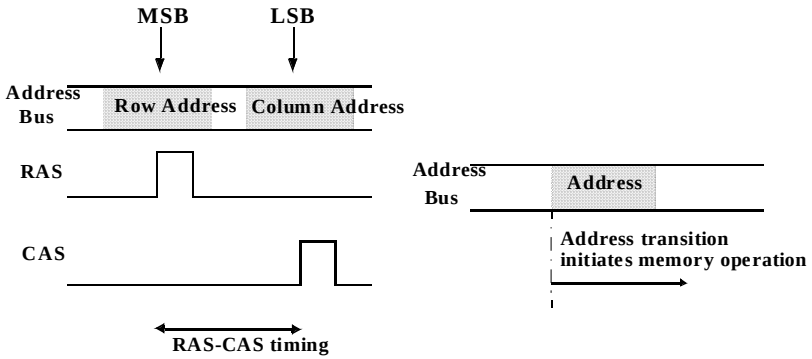


Advantages:
1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Memory Timing: Definitions



Memory Timing: Approaches



DRAM Timing
Multiplexed Adressing

SRAM Timing
Self-timed

Outline

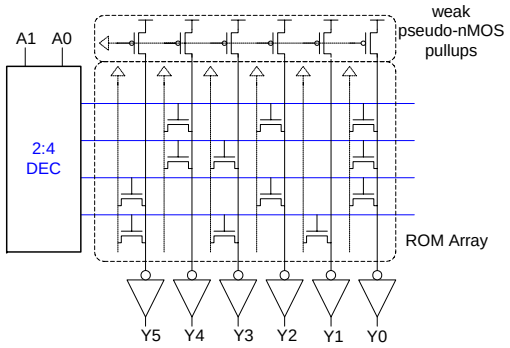
- Memory classification
- Basic building blocks
- **ROM**
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Read-Only Memories

- Read-Only Memories are nonvolatile
  - Retain their contents when power is removed
- Mask-programmed ROMs use one transistor per bit
  - Presence or absence determines 1 or 0

ROM Example

- 4-word x 6-bit ROM
  - Represented with dot diagram
  - Dots indicate 1's in ROM

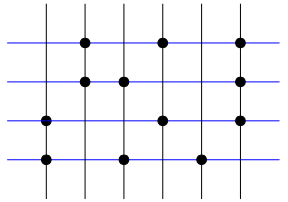


Word 0: **010101**

Word 1: **011001**

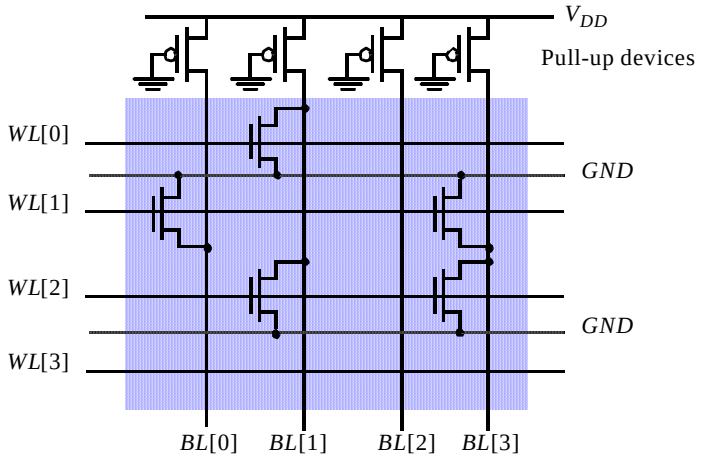
Word 2: **100101**

Word 3: **101010**

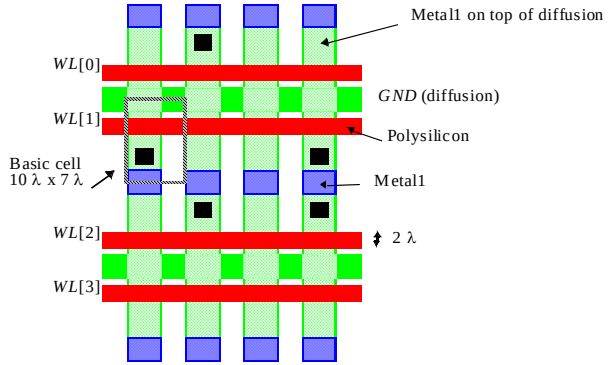


Looks like 6 4-input pseudo-nMOS NORs

MOS NOR ROM

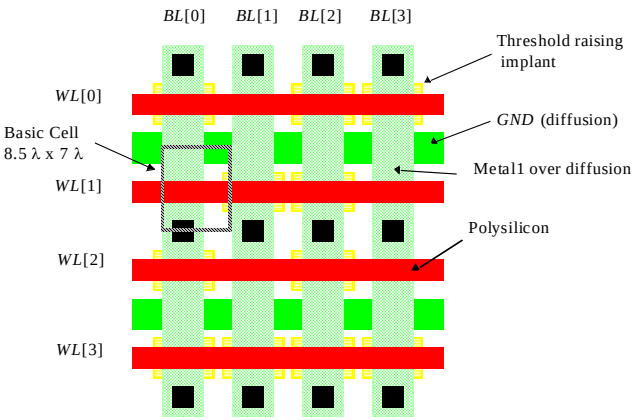


MOS NOR ROM Layout



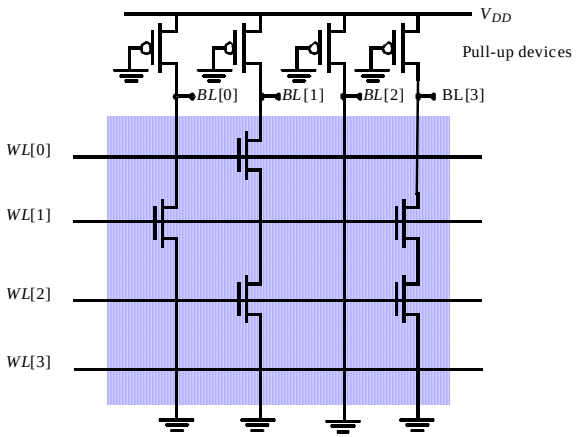
Only 1 layer (contact mask) is used to program memory array
Programming of the memory can be delayed to one of last process steps

MOS NOR ROM Layout



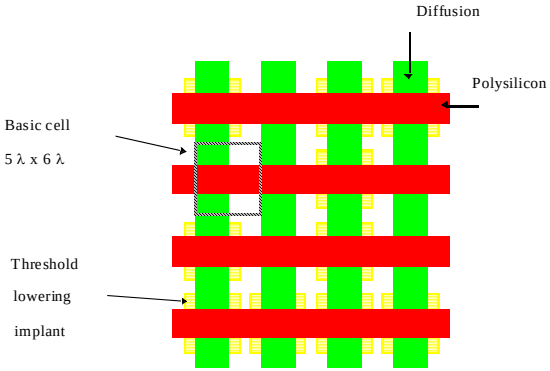
Threshold raising implants disable transistors

MOS NAND ROM



All word lines high by default with exception of selected row

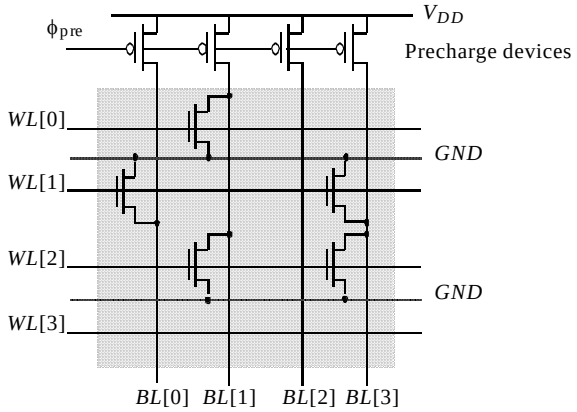
MOS NAND ROM Layout



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

Loss in performance compared to NOR ROM

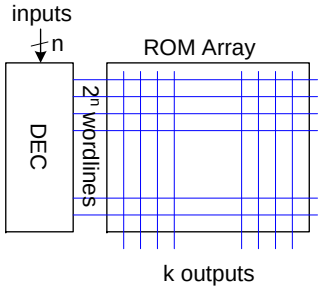
Precharged MOS NOR ROM

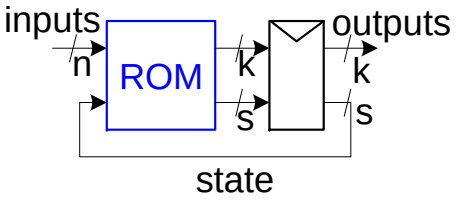


PMOS precharge device can be made as large as necessary, but clock driver becomes harder to design.

Building Logic with ROMs

- Use ROM as lookup table containing truth table
  - n inputs, k outputs requires 2^n words x k bits
  - Changing function is easy – reprogram ROM
- Finite State Machine
  - n inputs, k outputs, s bits of state
  - Build with 2^{n+s} x (k+s) bit ROM and (k+s) bit reg





Outline

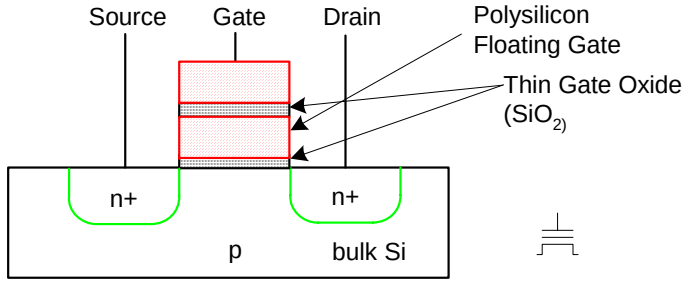
- Memory classification
- Basic building blocks
- ROM
- **Non Volatile Read Write Memories**
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Nonvolatile Read-Write Memories (NVRW)

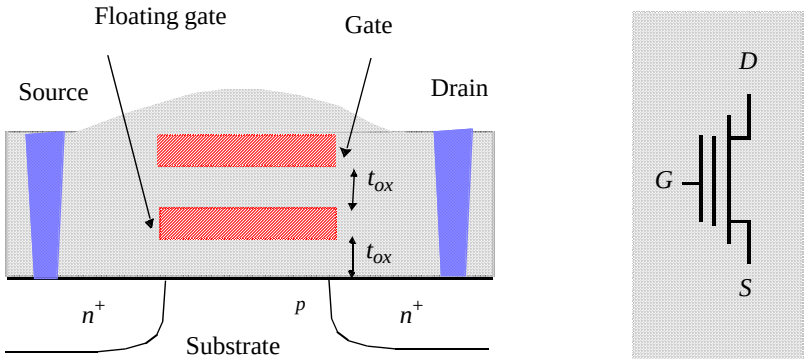
- Architecture virtually identical to the ROM structure
  - the memory core consists of an array of transistors placed on a word-line/bit-line grid
- The memory is programmed by selectively disabling or enabling some of those devices
  - in a ROM this is accomplished by mask level alterations
  - in a NVRW memory a modified transistor that permits its threshold to be altered electrically is used instead – the modified threshold is retained indefinitely (or long) even when the supply voltage is turned off
- To reprogram the memory the programmed values must be erased after which a new programming round can be started
  - The method of erasing is the main differentiating factor between the various classes of reprogrammable non volatile memories
  - The programming of the memory is typically an order of magnitude slower than the reading operation

PROMs and EPROMs

- Programmable ROMs
  - Build array with transistors at every site
  - Burn out fuses to disable unwanted transistors
- Electrically Programmable ROMs
  - Use floating gate to turn off unwanted transistors
  - EPROM, EEPROM, Flash



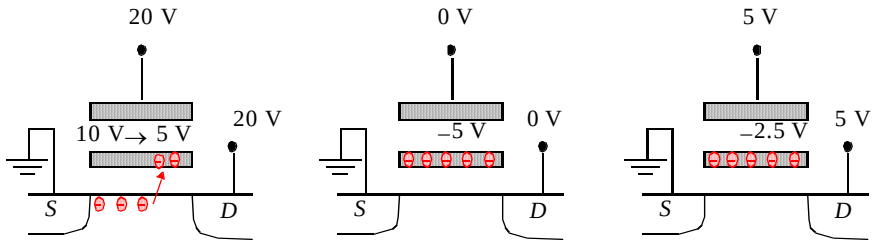
Floating-gate transistor (FAMOS)



(a) Device cross-section

(b) Schematic symbol

Floating-Gate Transistor Programming



Avalanche injection. **Removing programming voltage leaves charge trapped.** **Programming results in higher V_T.**

Characteristics of Non Volatile Memories

	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

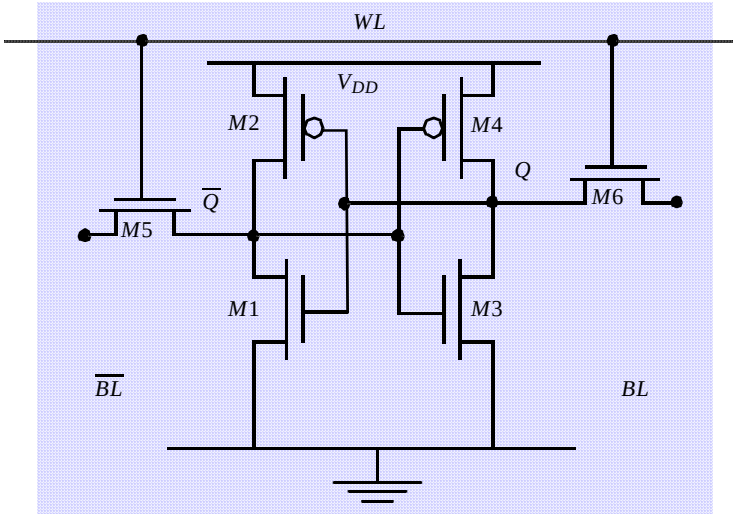
Outline

- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- **Static RAM (SRAM)**
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

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



SRAM Read/Write

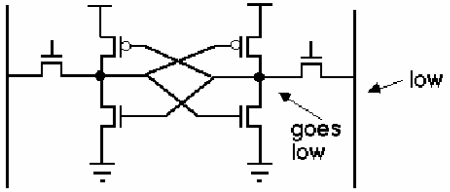
The key issue in an 6T SRAM is how to distinguish between read and writes. There is only one wordline, so it must be high for both reads and writes. The key is to use the fact there are two bitlines.

Read:

- Both Bit and $\overline{\text{Bit}}$ must start high. A high value on the bitline does not change the value in the cell, so the cell will pulls one of the lines low

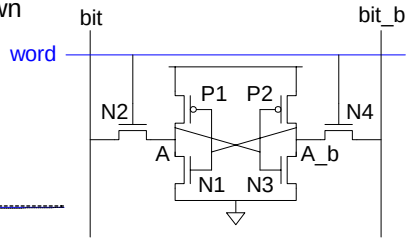
Write:

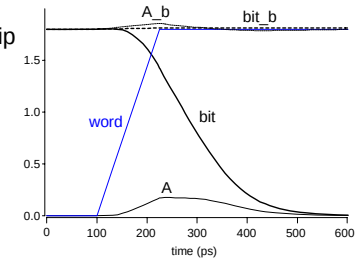
- One (Bit or $\overline{\text{Bit}}$) is forced low, the other is high
- This low value overpowers the pMOS in the inverter, and this will write the cell.



CMOS SRAM Analysis (Read)

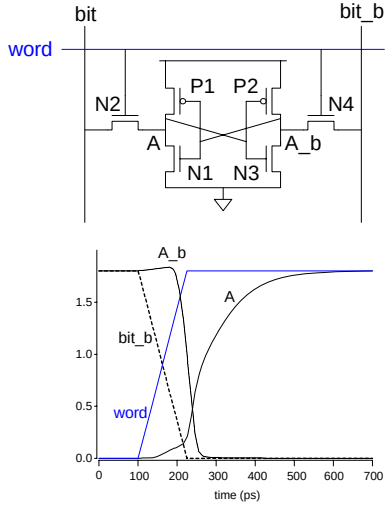
- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- *Read stability*
  - A must not flip
  - N1 >> N2





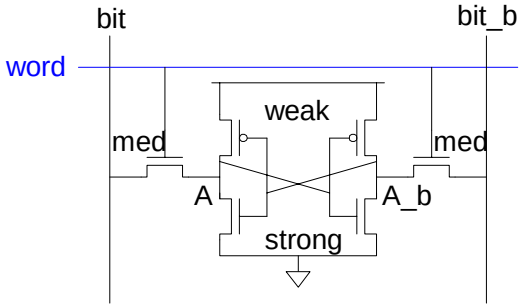
CMOS SRAM Analysis (Write)

- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- Ex: A = 0, A_b = 1, bit = 1, bit_b = 0
  - Force A_b low, then A rises high
- *Writability*
  - Must overpower feedback inverter
  - N2 >> P1

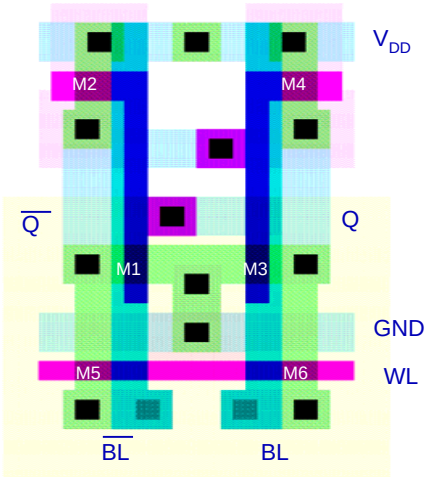


SRAM Sizing

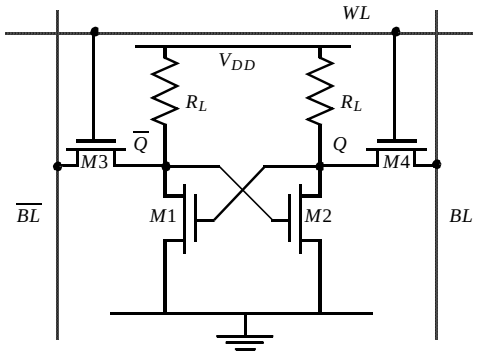
- High bitlines must not overpower inverters during reads
- But low bitlines must write new value into cell



6T-SRAM — Layout



Resistance-load SRAM Cell



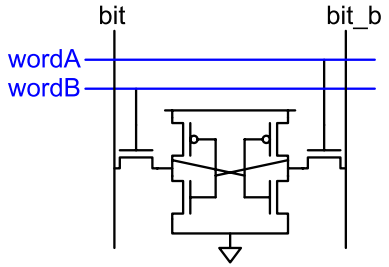
Static power dissipation -- Want R_L large
Bit lines precharged to V_{DD} to address t_p problem

Multiple Ports

- We have considered single-ported SRAM
  - One read or one write on each cycle
- *Multiported* SRAM are needed for register files
- Examples:
  - Multicycle MIPS must read two sources or write a result on some cycles
  - Pipelined MIPS must read two sources and write a third result each cycle
  - Superscalar MIPS must read and write many sources and results each cycle

Dual-Ported SRAM

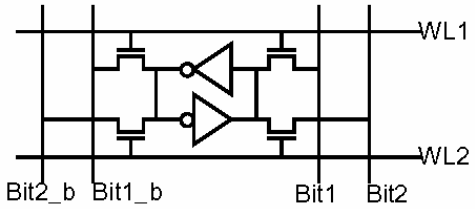
- Simple dual-ported SRAM
  - Two independent single-ended reads
  - Or one differential write



- Do two reads and one write by time multiplexing
  - Read during ph1, write during ph2

True dual port SRAM

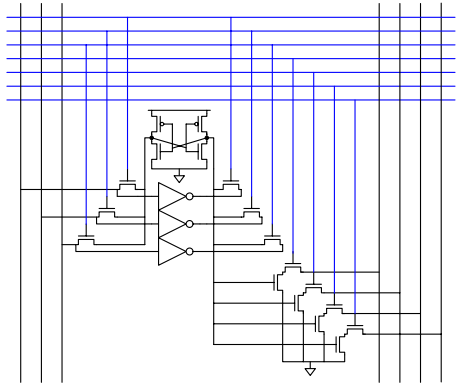
Can build true multi-ported memory cell, by adding more bitline pairs and wordlines to a cell.



Shown in the figure is a true dual port cell. You can read or write on each port every cycle. Since it has more bitlines than the previous cell, it is much larger in area.

Multi-Ported SRAM

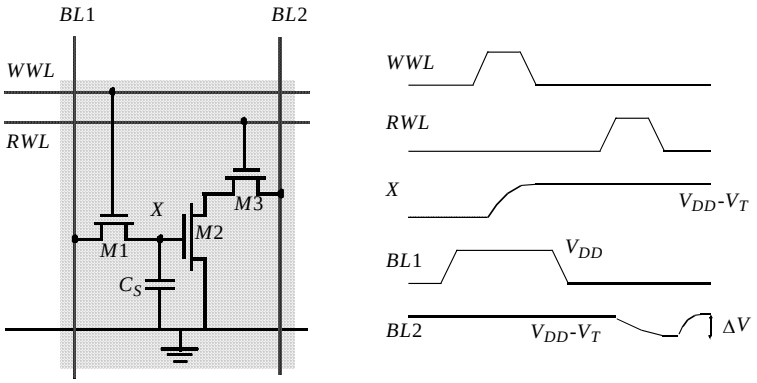
- Adding more access transistors hurts read stability
- Multiported SRAM isolates reads from state node
- Single-ended design minimizes number of bitlines



Outline

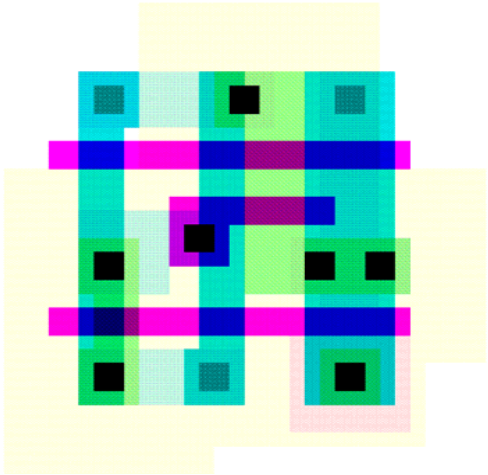
- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- **Dynamic RAM (DRAM)**
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

3-Transistor DRAM Cell

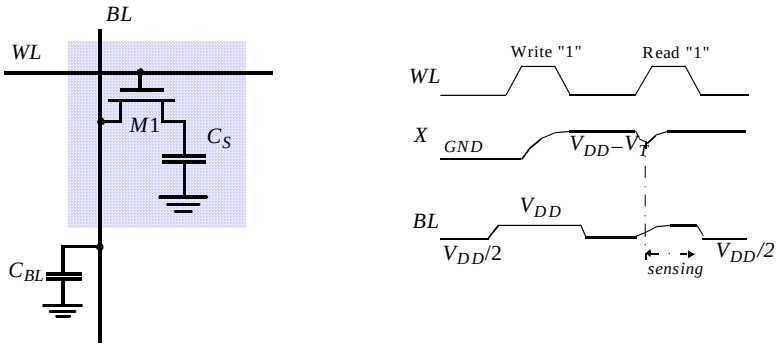


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a "1" = V_{WWL}-V_{Tn}

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.

DRAM Cell Observations

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

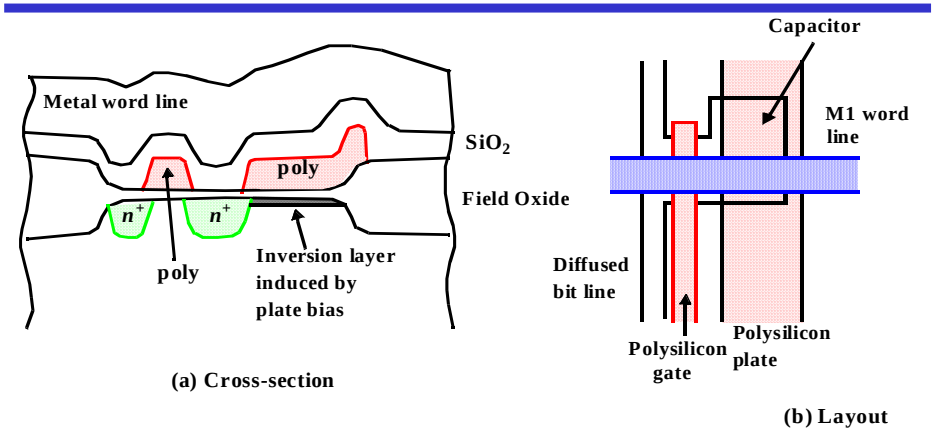
DRAM memory cells are single ended in contrast to SRAM cells.

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



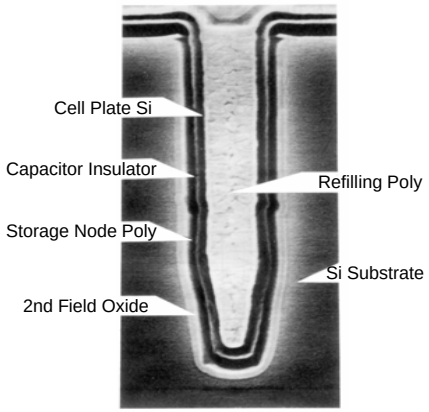
(a) Cross-section

(b) Layout

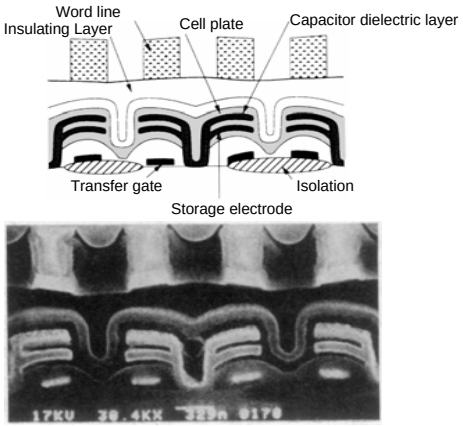
Used Polysilicon-Diffusion Capacitance

Expensive in Area

Advanced 1T DRAM Cells



Trench Cell



Stacked-capacitor Cell

Outline

- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- **Memory peripheral circuit**
- Content Addressable Memory (CAM)
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Periphery

- Decoders
- Sense amplifiers
- Input/output buffers
- Control/timing circuit

Row Decoders

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

(N)AND Decoder

$$WL_0 = A_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

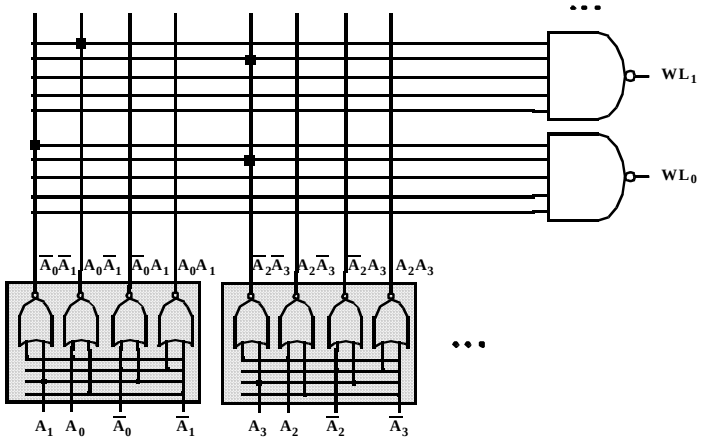
$$WL_{511} = \bar{A}_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

NOR Decoder

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

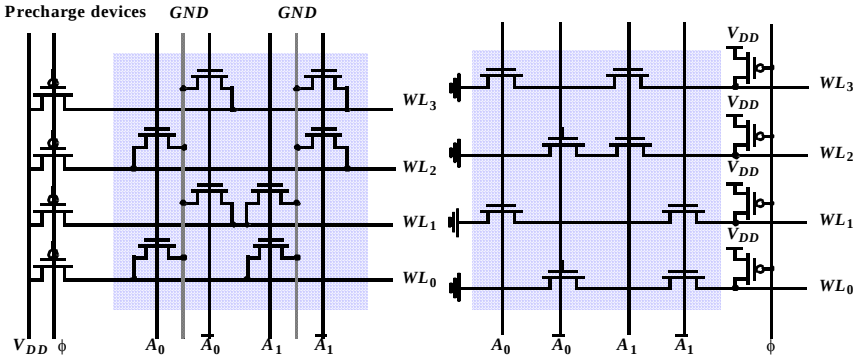
$$WL_{511} = \overline{A_0 + \bar{A}_1 + \bar{A}_2 + \bar{A}_3 + \bar{A}_4 + \bar{A}_5 + \bar{A}_6 + \bar{A}_7 + \bar{A}_8 + \bar{A}_9}$$

A NAND Decoder using 2-input Pre-Decoders



Splitting decoder into two or more logic layers produces a faster and cheaper implementation

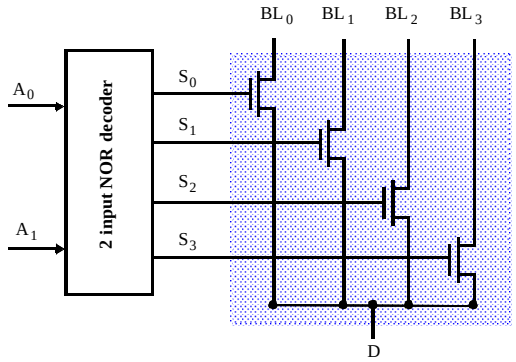
Dynamic Decoders



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

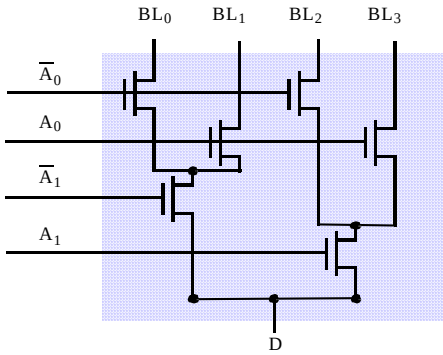
Propagation delay is primary concern

4 input Pass-Transistor based Column Decoder



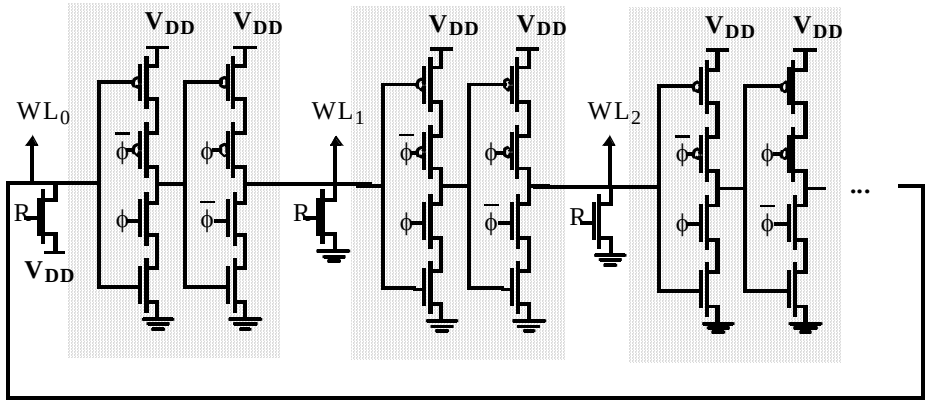
Advantage: speed (t_{pd} does not add to overall memory access time)
only 1 extra transistor in signal path
Disadvantage: large transistor count

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

Decoder for Circular Shift-Register



Sense Amplifiers

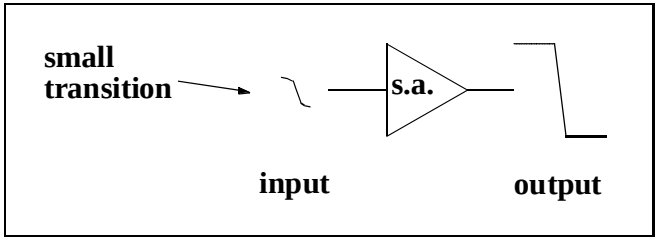
$$t_p = \frac{C \cdot \Delta V}{I_{av}}$$

make ΔV as small as possible

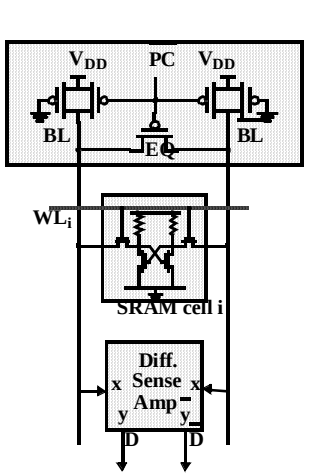
large

small

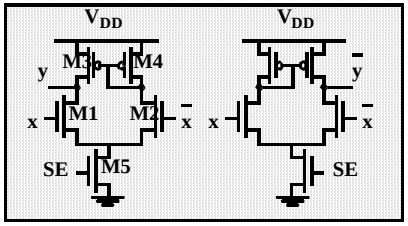
Idea: Use Sense Amplifer



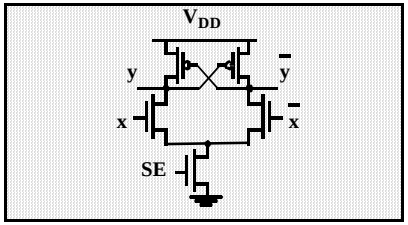
Differential Sensing - SRAM



(a) SRAM sensing scheme.

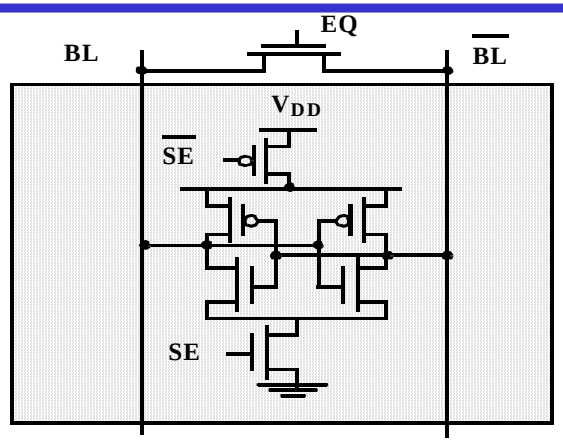


(b) Doubled-ended Current Mirror Amplifier



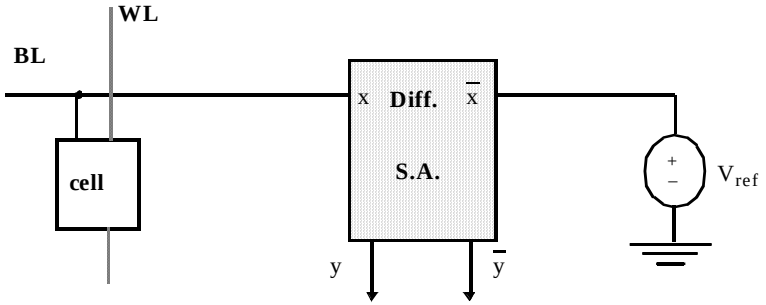
(c) Cross-Coupled Amplifier

Latch-Based Sense Amplifier



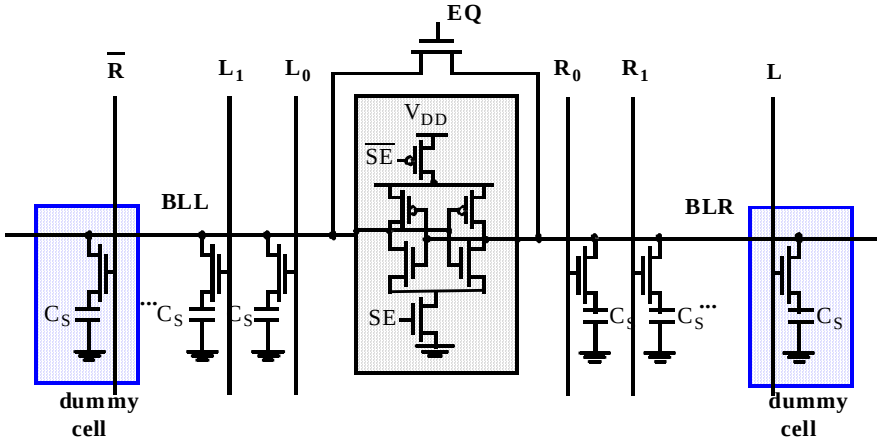
Initialized in its meta-stable point with EQ
Once adequate voltage gap created, sense amp enabled with SE
Positive feedback quickly forces output to a stable operating point.

Single-to-Differential Conversion

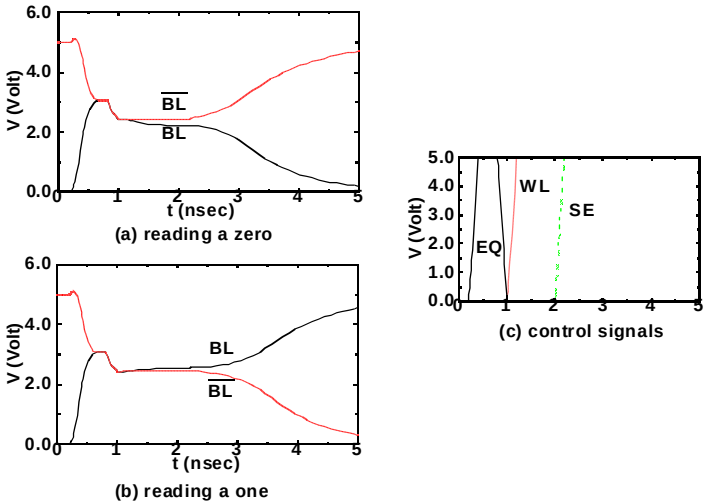


How to make good V_{ref}?

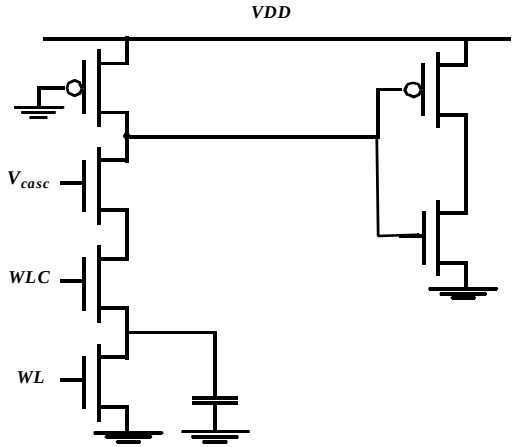
Open Bitline Architecture



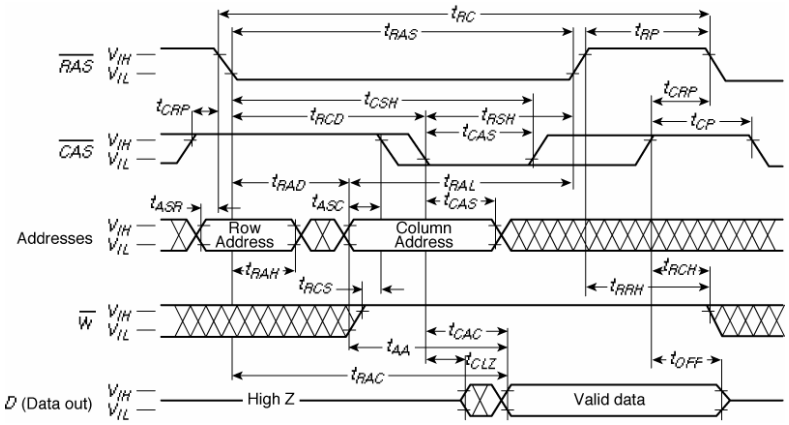
DRAM Read Process with Dummy Cell



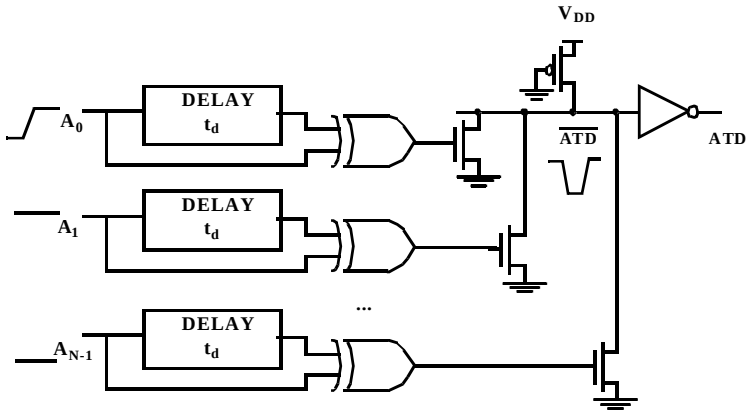
Single-Ended Cascode Amplifier



DRAM Timing (nightmare!)



Address Transition Detection

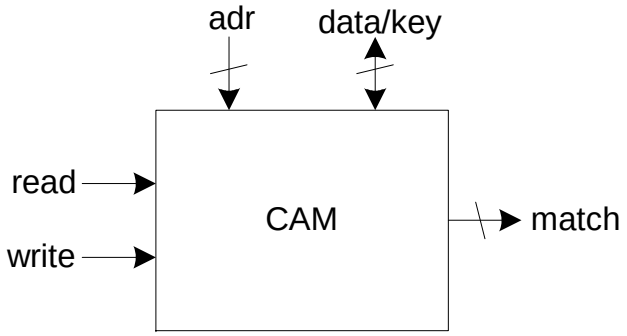


Outline

- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- **Content Addressable Memory (CAM)**
- Serial access memories
- Programmable Logic Array
- Reliability and Yield
- Memory trends

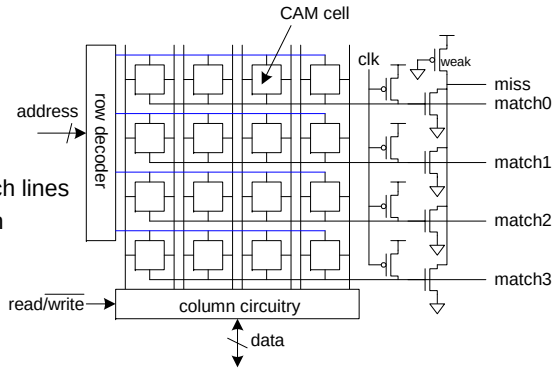
Content Addressable Memories (CAMs)

- Extension of ordinary memory (e.g. SRAM)
  - Read and write memory as usual
  - Also *match* to see which words contain a *key*



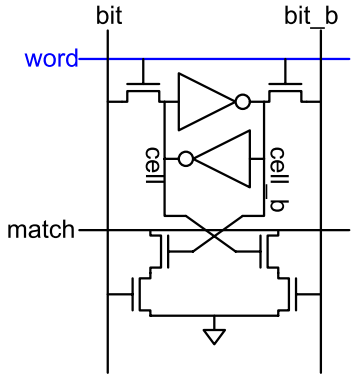
CAM Cell Operation

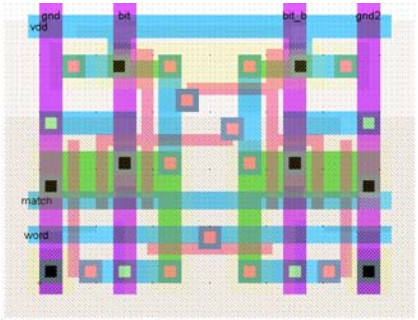
- Read and write like ordinary SRAM
- For matching:
  - Leave wordline low
  - Precharge matchlines
  - Place key on bitlines
  - Matchlines evaluate
- Miss line
  - Pseudo-nMOS NOR of match lines
  - Goes high if no words match



10T CAM Cell

- Add four match transistors to 6T SRAM





Outline

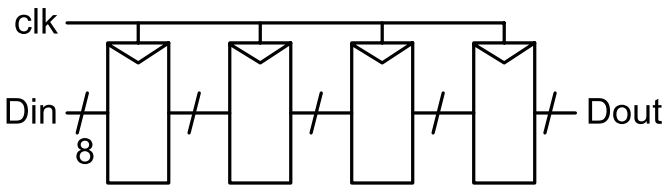
- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- **Serial access memories**
- Programmable Logic Array
- Reliability and Yield
- Memory trends

Serial Access Memories

- Serial access memories do not use an address
  - Shift Registers
  - Tapped Delay Lines
  - Serial In Parallel Out (SIPO)
  - Parallel In Serial Out (PISO)
  - Queues (FIFO, LIFO)

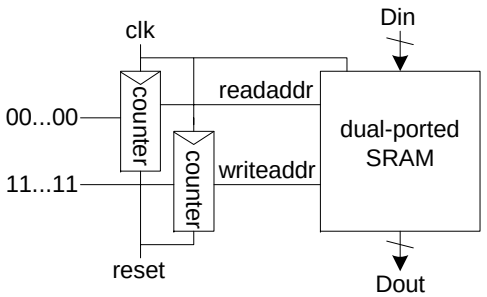
Shift Register

- *Shift registers* store and delay data
- Simple design: cascade of registers
  - Watch your hold times!



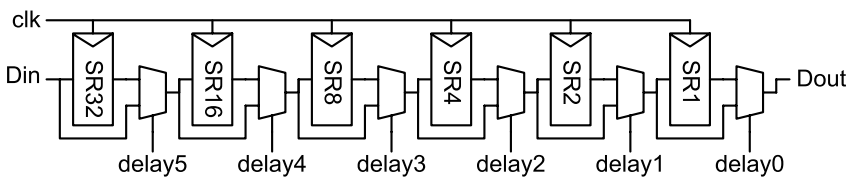
Denser Shift Registers

- Flip-flops aren't very area-efficient
- For large shift registers, keep data in SRAM instead
- Move read/write pointers to RAM rather than data
  - Initialize read address to first entry, write to last
  - Increment address on each cycle



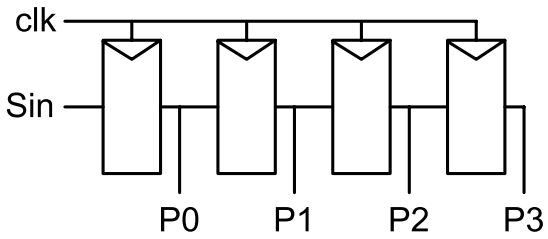
Tapped Delay Line

- A *tapped delay line* is a shift register with a programmable number of stages
- Set number of stages with delay controls to mux
  - Ex: 0 – 63 stages of delay



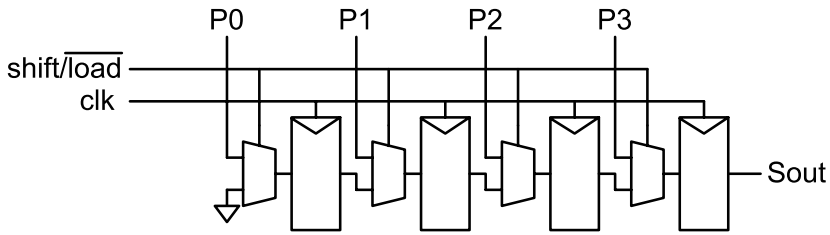
Serial In Parallel Out

- 1-bit shift register reads in serial data
  - After N steps, presents N-bit parallel output



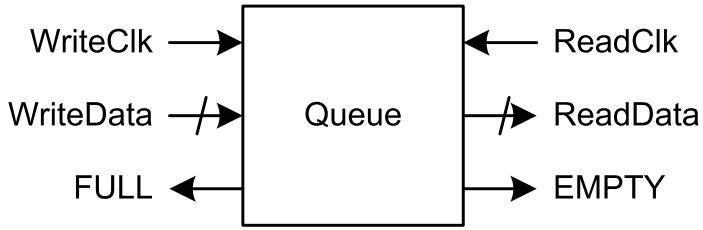
Parallel In Serial Out

- Load all N bits in parallel when shift = 0
  - Then shift one bit out per cycle



Queues

- *Queues* allow data to be read and written at different rates.
- Read and write each use their own clock, data
- Queue indicates whether it is full or empty
- Build with SRAM and read/write counters (pointers)



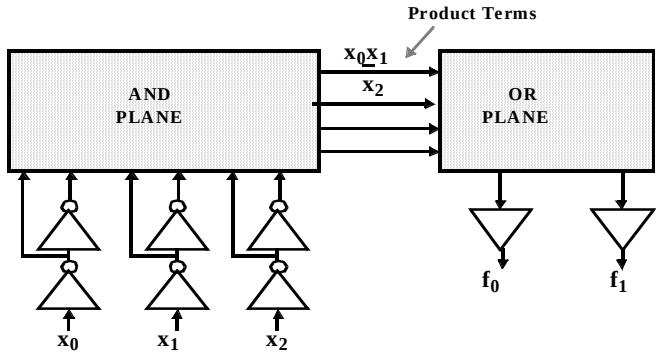
FIFO, LIFO Queues

- *First In First Out* (FIFO)
  - Initialize read and write pointers to first element
  - Queue is EMPTY
  - On write, increment write pointer
  - If write almost catches read, Queue is FULL
  - On read, increment read pointer
- *Last In First Out* (LIFO)
  - Also called a *stack*
  - Use a single *stack pointer* for read and write

Outline

- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Content Addressable Memory (CAM)
- Serial access memories
- **Programmable Logic Arrays**
- Reliability and Yield
- Memory trends

Programmable Logic Array

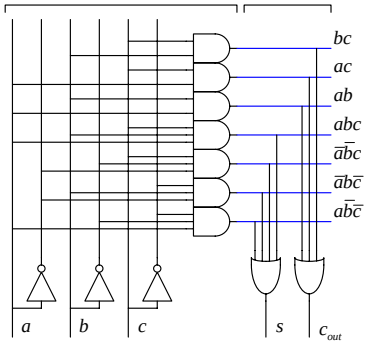


PLAs

- A *Programmable Logic Array* performs any function in sum-of-products form.
- *Literals*: inputs & complements
- *Products / Minterms*: AND of literals
- *Outputs*: OR of Minterms
- Example: Full Adder

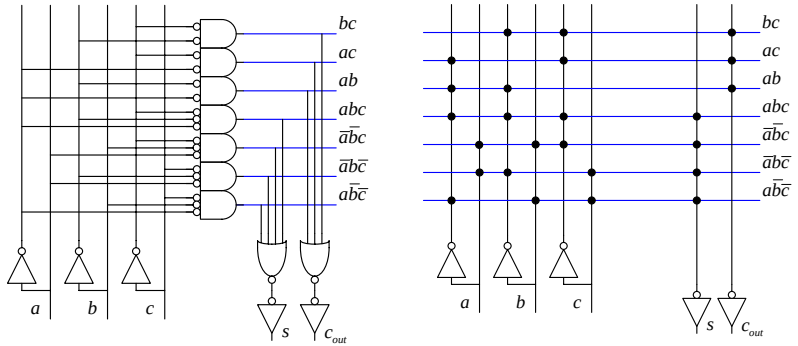
$$s = a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c + abc$$

$$c_{out} = ab + bc + ac$$

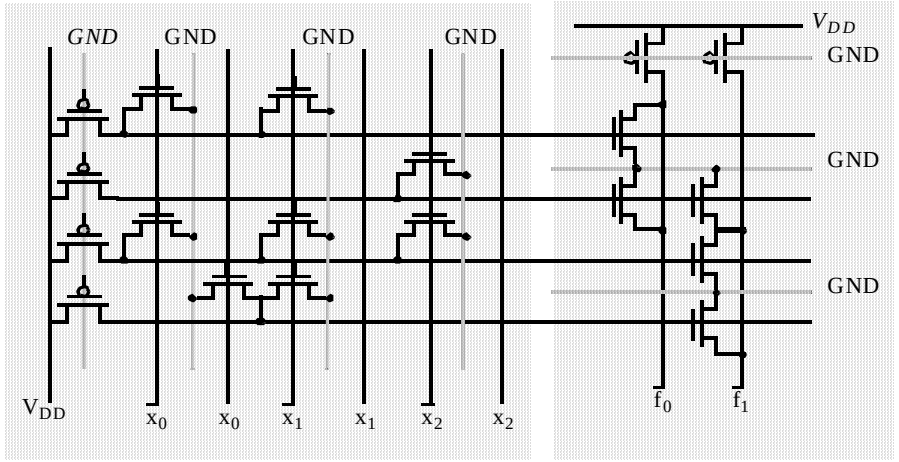


NOR-NOR PLAs

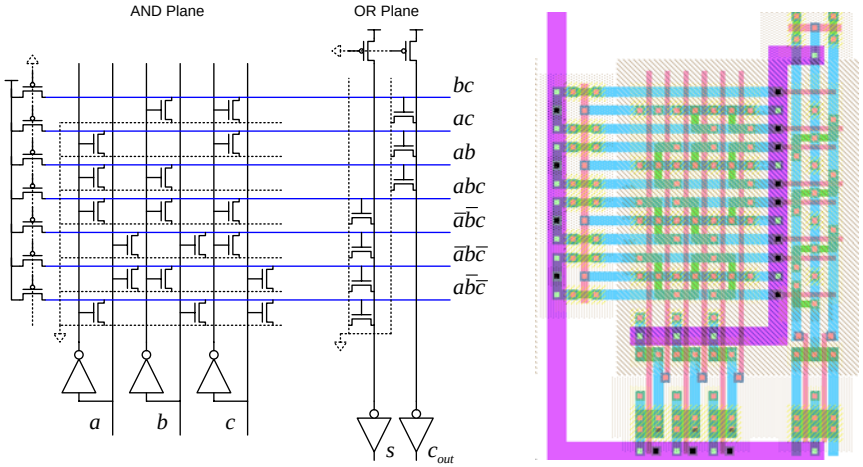
- ANDs and ORs are not very efficient in CMOS
- Dynamic or Pseudo-nMOS NORs are very efficient
- Use DeMorgan's Law to convert to all NORs



Pseudo-Static PLA

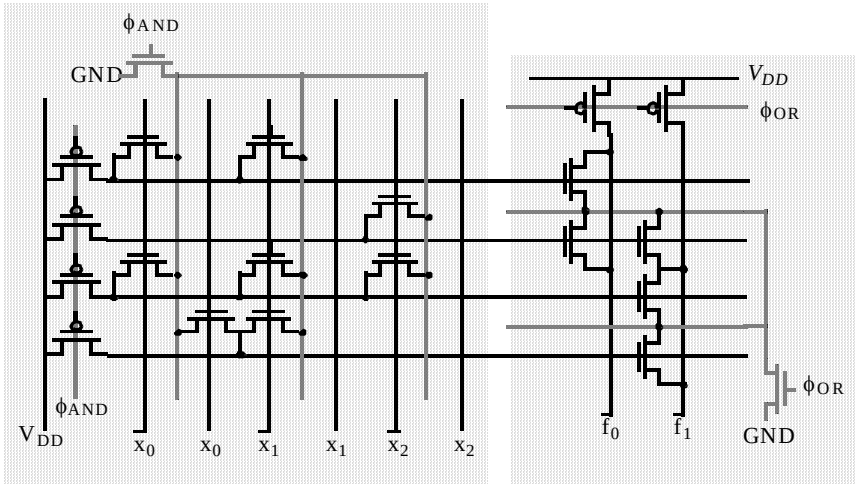


PLA Schematic & Layout

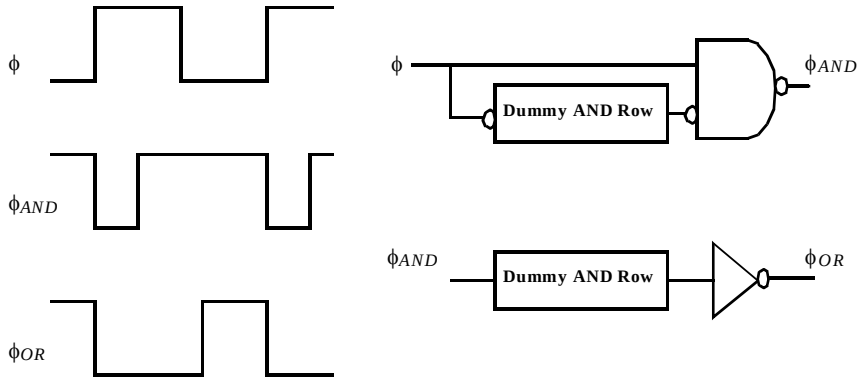


OR

Dynamic PLA

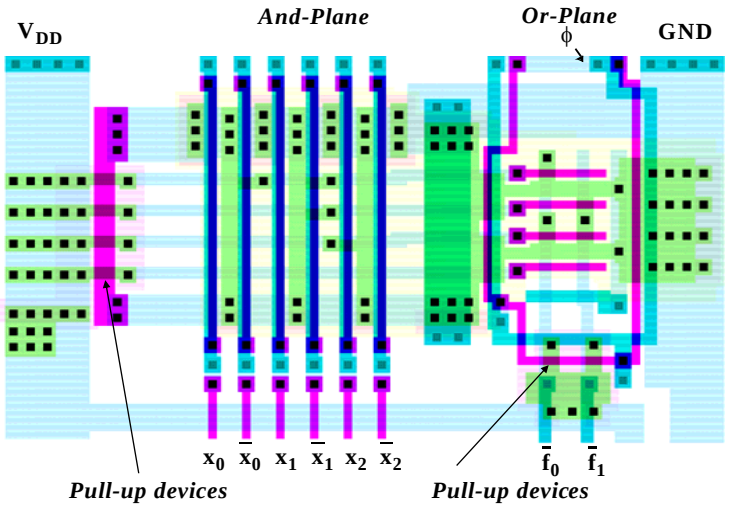


Clock Signal Generation for self-timed dynamic PLA



(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)**

Outline

- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Serial access memories
- Content Addressable Memory (CAM)
- Programmable Logic Array
- **Reliability and Yield**
- Memory trends

Reliability and Yield

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

Highly Sensitive to Noise (Crosstalk, Supply Noise)

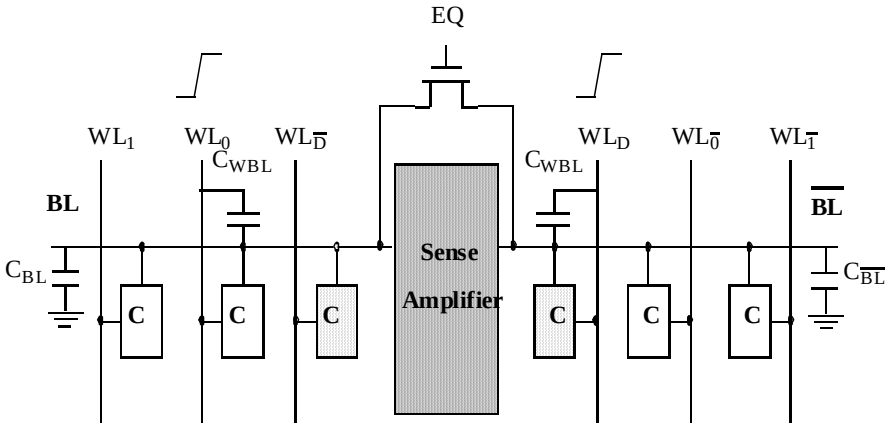
- High Density and Large Die size cause Yield Problems

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

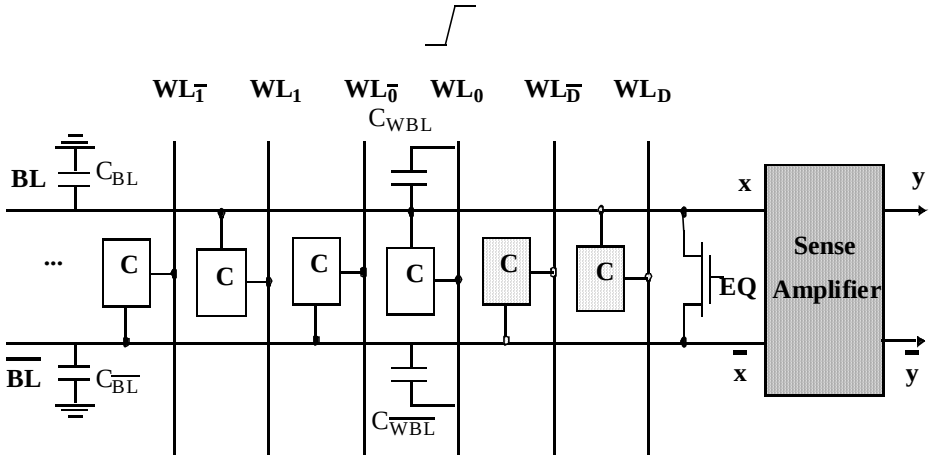
$$Y = \left[\frac{1-e^{-AD}}{AD}\right]^2$$

Increase Yield using Error Correction and Redundancy

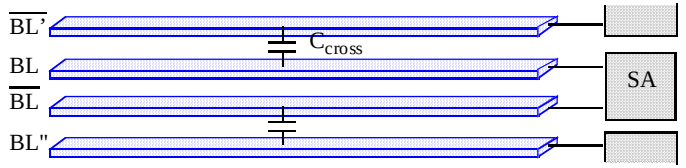
Open Bit-line Architecture —Cross Coupling



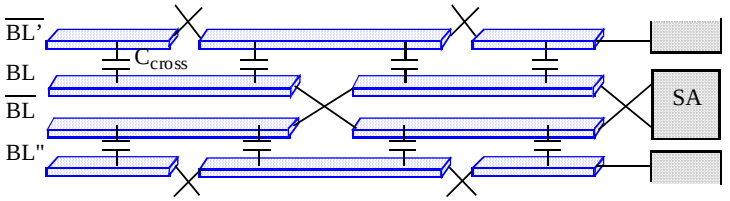
Folded-Bitline Architecture



Transposed-Bitline Architecture

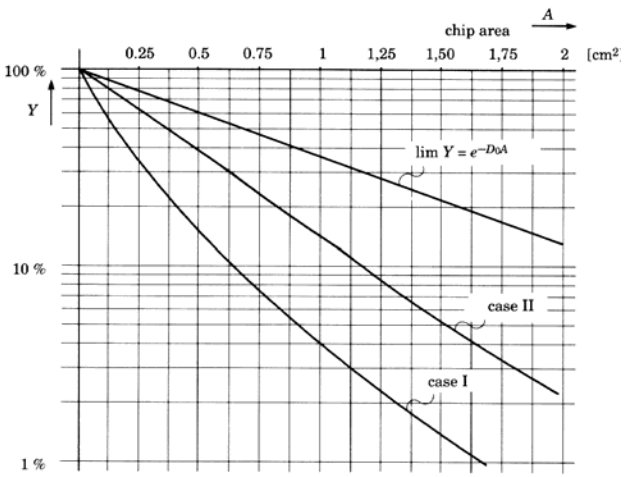


(a) Straightforward bitline routing.



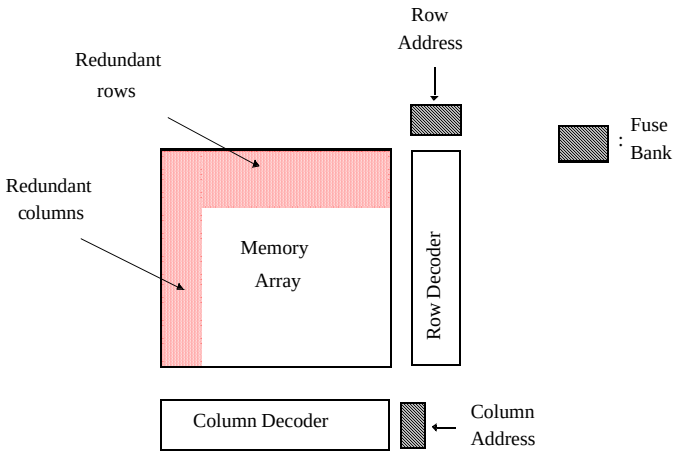
(b) Transposed bitline architecture.

Yield

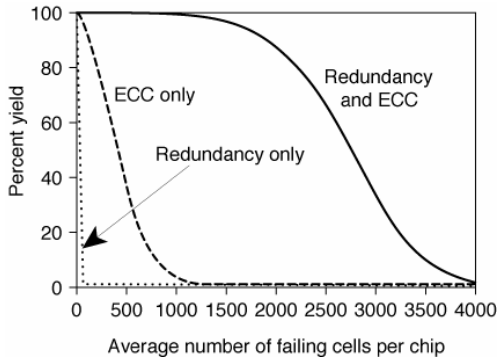


Yield curves at different stages of process maturity (from [Veendrick92])

Redundancy



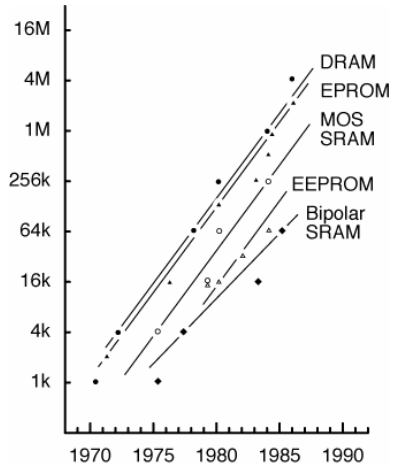
Redundancy and Error Correction



Outline

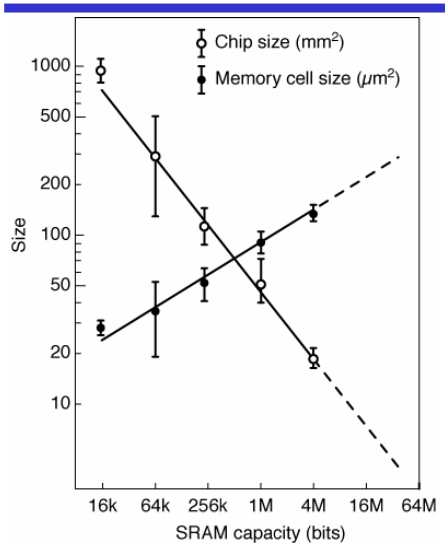
- Memory classification
- Basic building blocks
- ROM
- Non Volatile Read Write Memories
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Memory peripheral circuit
- Serial access memories
- Content Addressable Memory (CAM)
- Programmable Logic Array
- Reliability and Yield
- **Memory trends**

Semiconductor Memory Trends



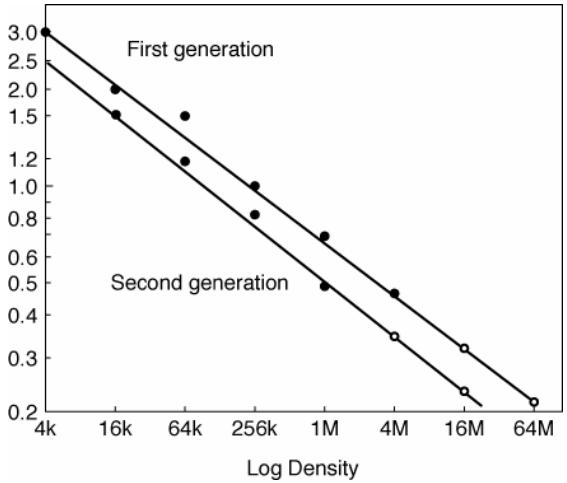
Memory Size as a function of time: x 4 every three years

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