SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB302–VLSI DESIGN

III YEAR/ V SEMESTER

1

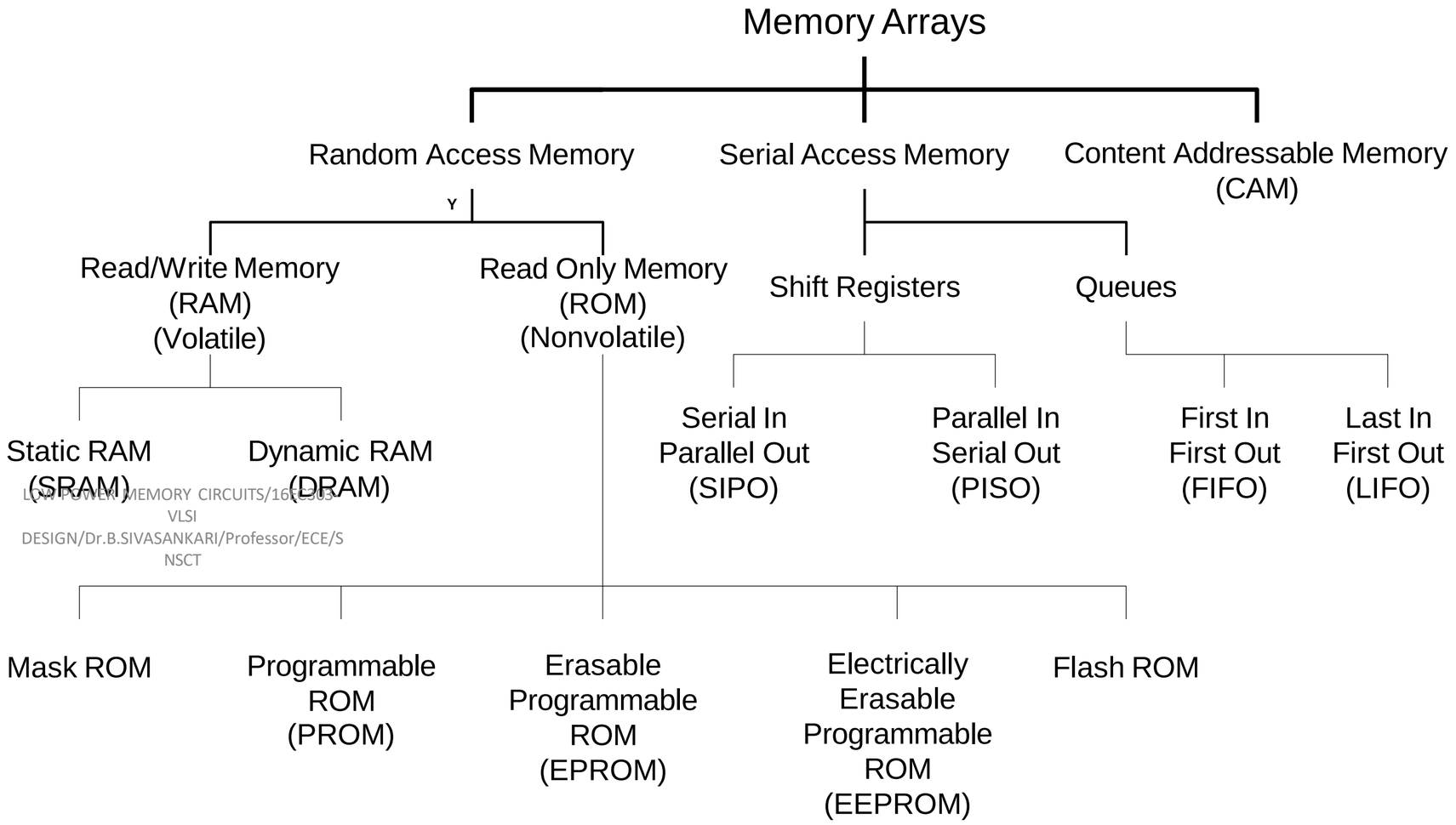
UNIT 3 –SEQUENTIAL LOGIC CIRCUITS

TOPIC 7–LOW POWER MEMORY CIRCUITS

OUTLINE

- MEMORY CLASSIFICATION & MEMORY ARRAYS
- ARRAY ARCHITECTURE
- RAM VS ROM
- DRAM VS SRAM
- 6T SRAM CELL
- SRAM SIZING
- CAMS-CAM IN CACHE MEMORY, 10T CAM Cell, CAM CELL OPERATION
- ACTIVITY
- ROM
- MOS NAND ROM
- ROM EXAMPLE
- NON-VOLATILE MEMORIES
  THE FLOATING-GATE TRANSISTOR (FAMOS)
- PERIPHERY
  - DECODERS
  - SENSE AMPLIFIERS
  - INPUT/OUTPUT BUFFERS
  - CONTROL / TIMING CIRCUITRY
- ASSESSMENT
- SUMMARY

MEMORY ARRAYS

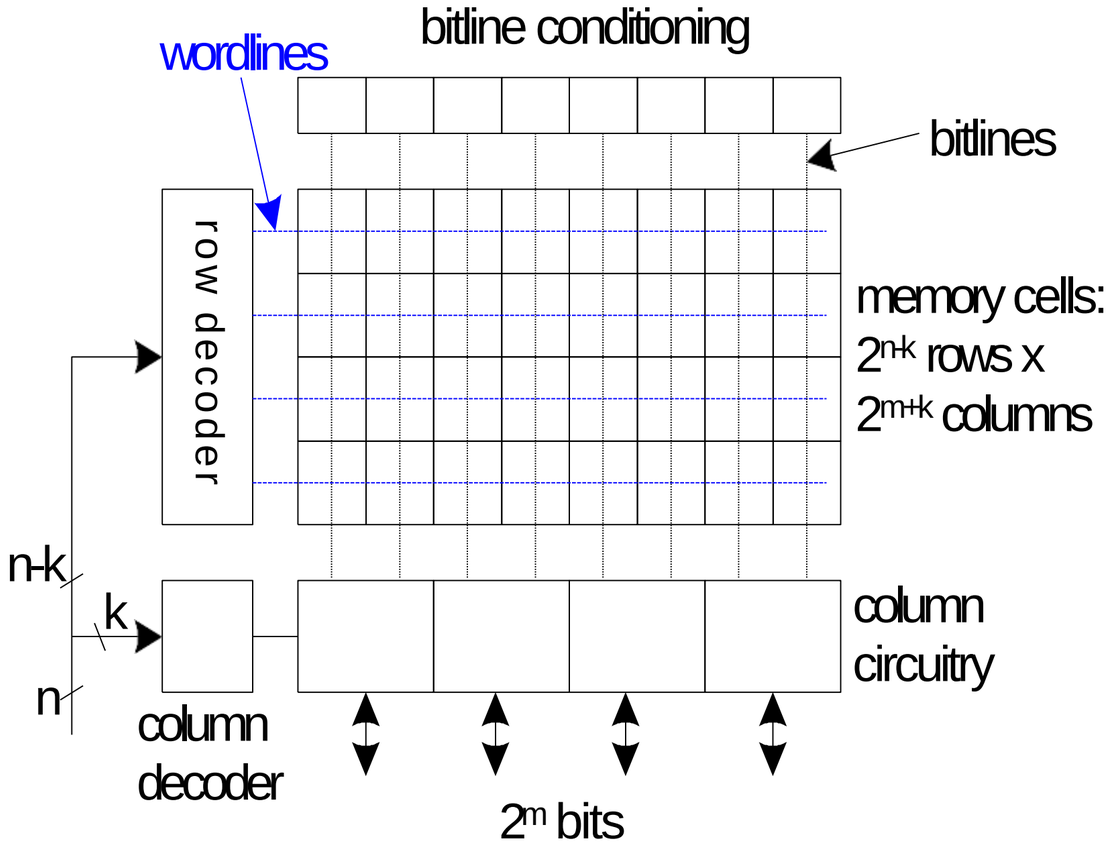


MEMORY CLASSIFICATION

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

ARRAY ARCHITECTURE

- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*



- Good regularity – easy to design
- Very high density if good cells are used

RAM VS ROM

RAM

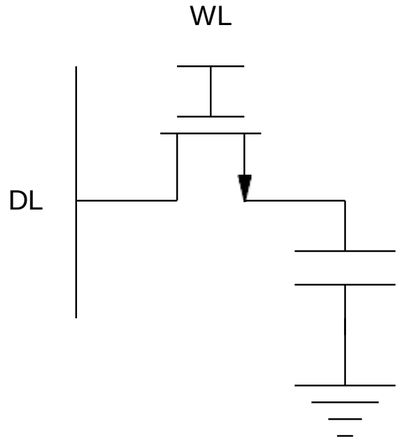
- Random write and read operation for any cell
- Volatile data
- Most of computer memory
- DRAM
  - Low Cost
  - High Density
  - Medium Speed
- SRAM
  - High Speed
  - Ease of use
  - Medium Cost

ROM

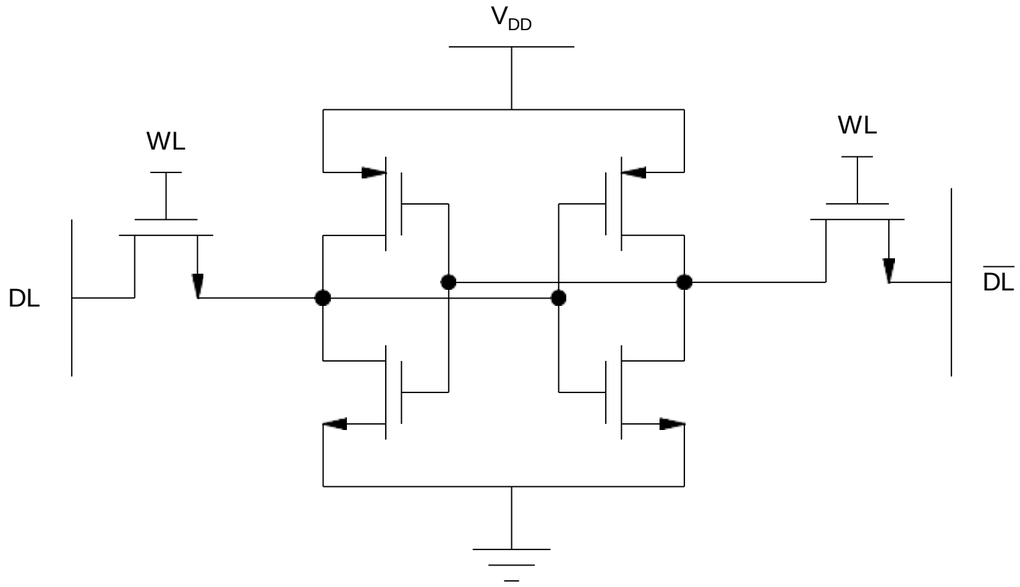
- Non-volatile Data
- Method of Data Writing
- Mask ROM
  - Data written during chip fabrication
- PROM
  - Fuse ROM: Non-rewritable
  - EPROM: Erase data by UV rays
  - EEPROM: Erase and write through electrical means
    - Speed 2-3 times slower than RAM
    - Upper limit on write operations
    - Flash Memory – High density, Low Cost

DRAM VS SRAM

DRAM

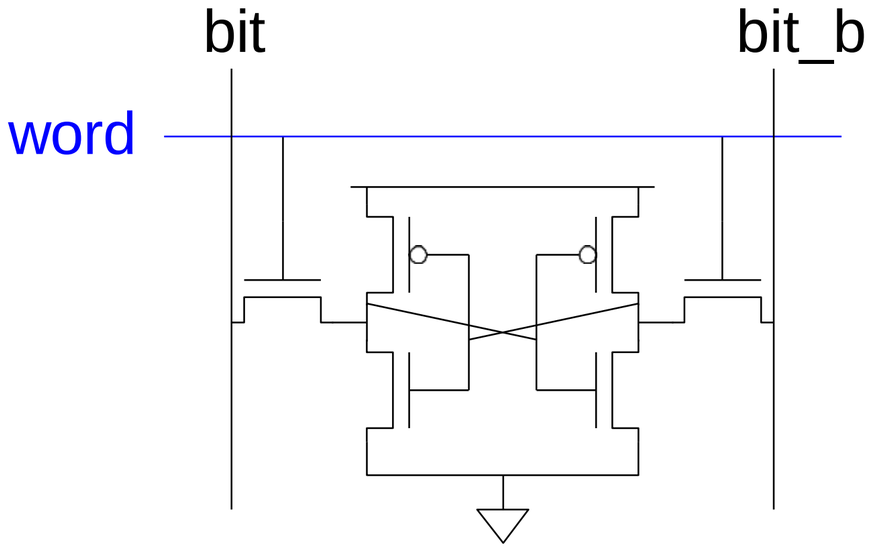


SRAM



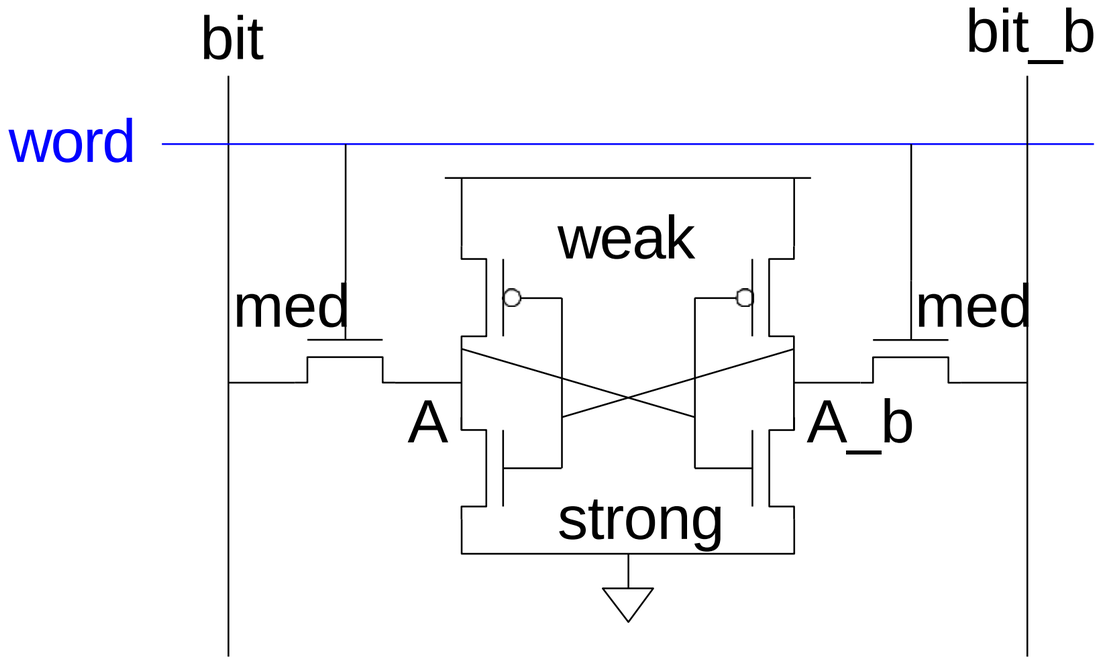
6T SRAM CELL

- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge bit, bit_b
  - Raise wordline
- Write:
  - Drive data onto bit, bit_b
  - Raise wordline



SRAM SIZING

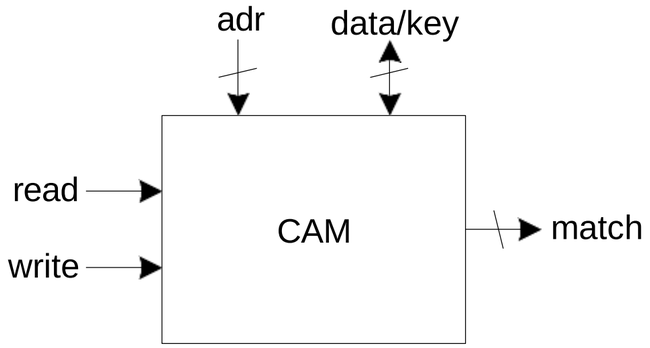
- High bitlines must not overpower inverters during reads
- But low bitlines must write new value into cell



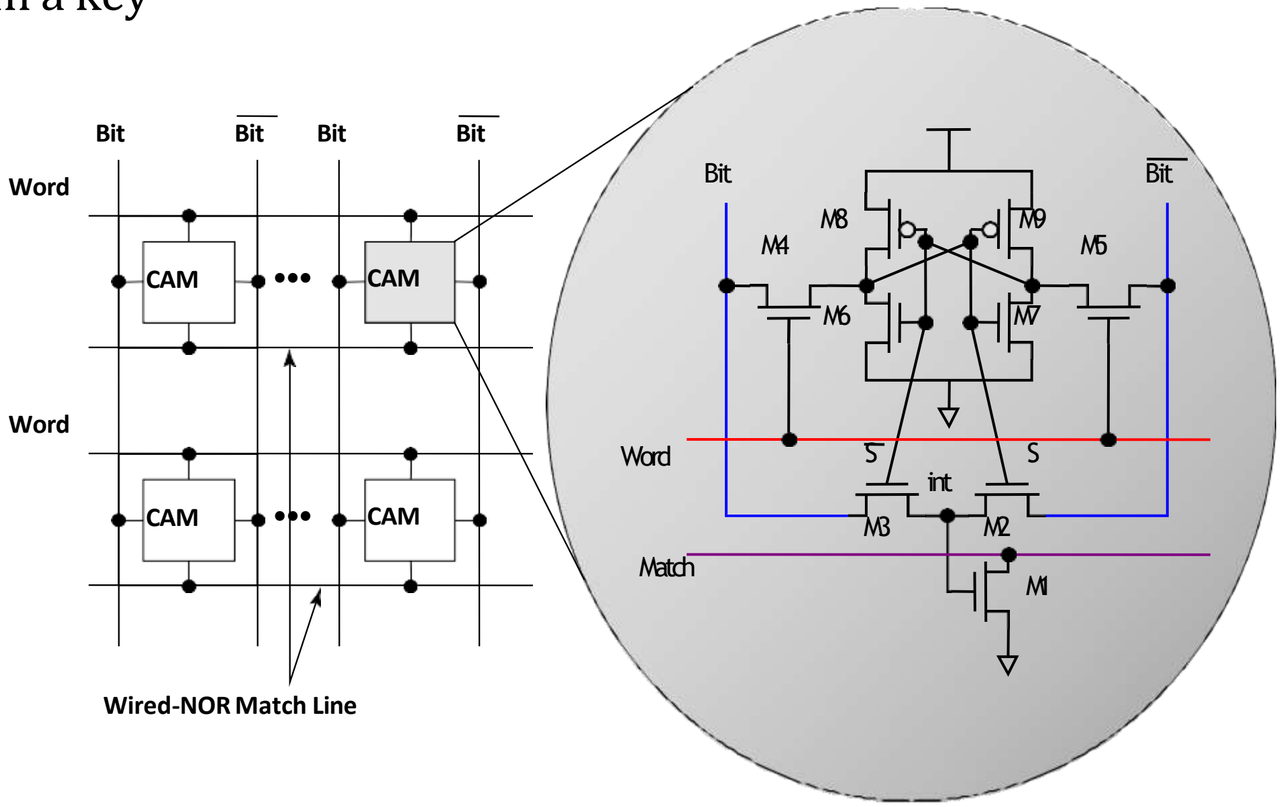
CAMS

Extension of ordinary memory (e.g. SRAM)

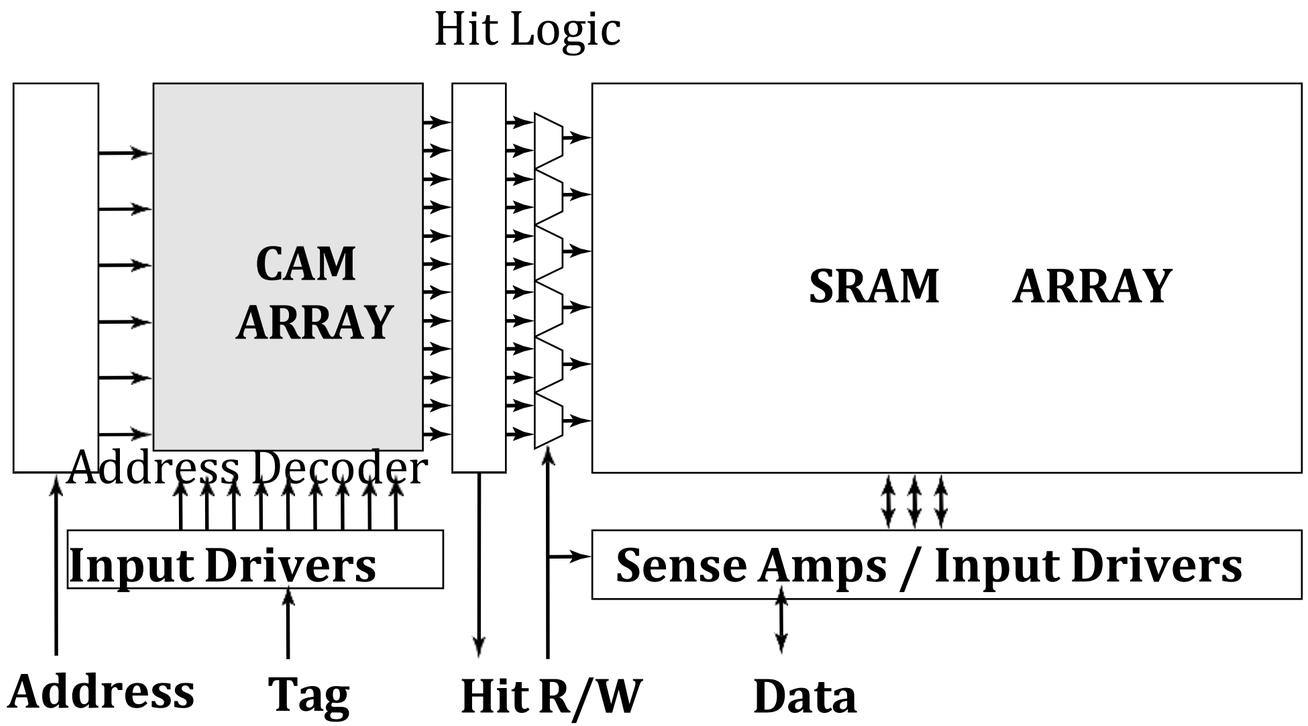
- Read and write memory as usual
- Also match to see which words contain a key



Static CAM Memory Cell

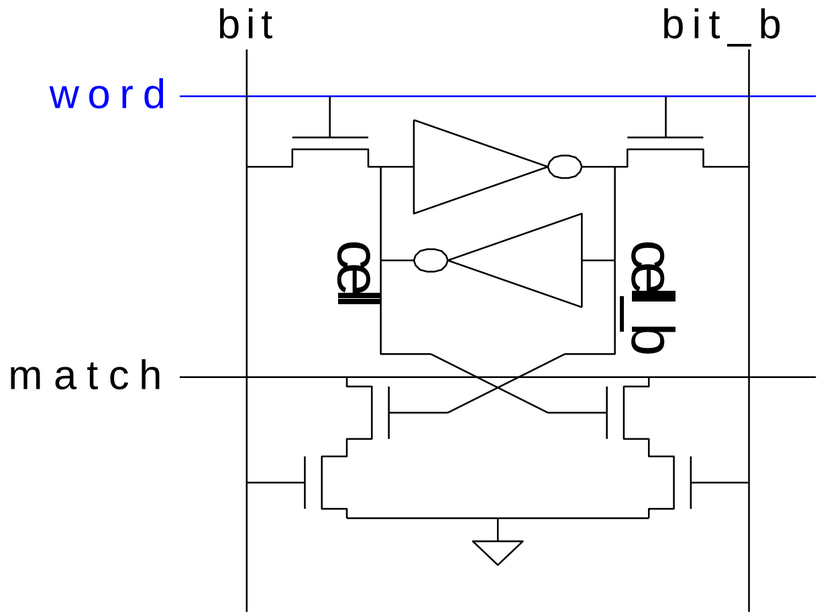


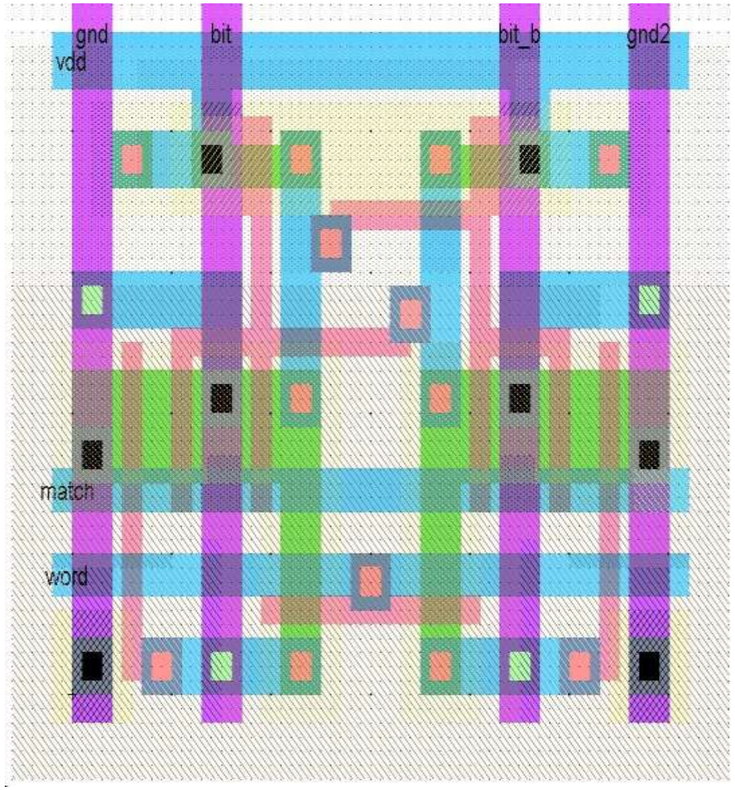
CAM IN CACHE MEMORY



10T CAM Cell

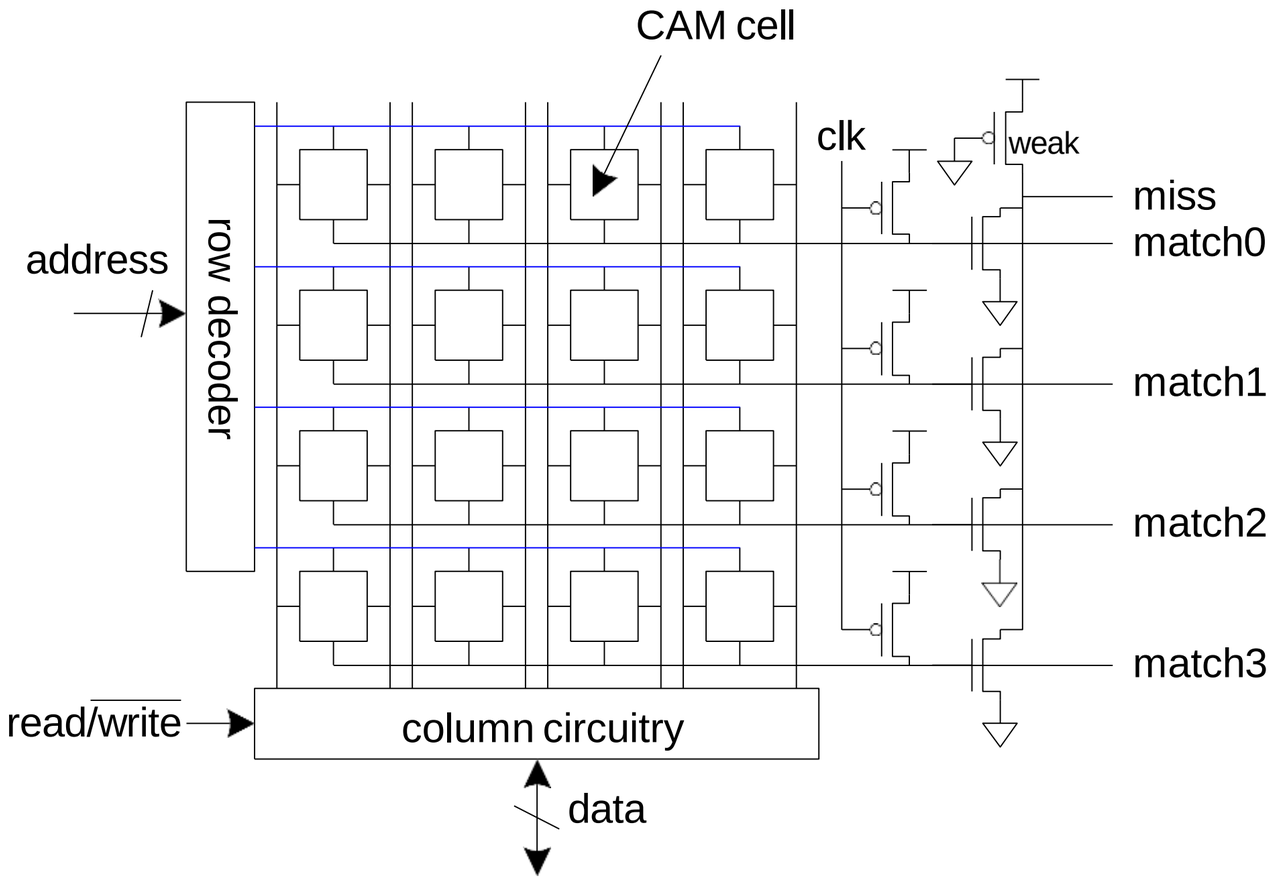
Add four match transistors to 6T SRAM
56 x 43 l unit cell





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

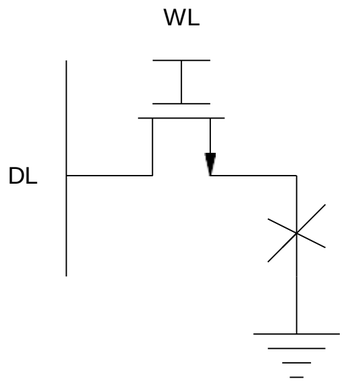


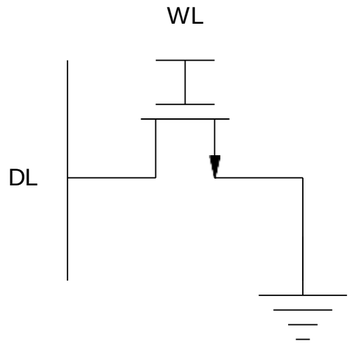
ROM

•To store constants, control information and program instructions in digital systems.
•To provide a fixed, specified binary output for every binary input.
•simple combinational Boolean network, which produces a specified output value for each input combination, i.e. for each address.
• storing binary information at a particular address location can be achieved by the presence or absence of a data path from the selected row (word line) to the selected column (bit line), which is equivalent to the presence or absence of a device at that particular location.

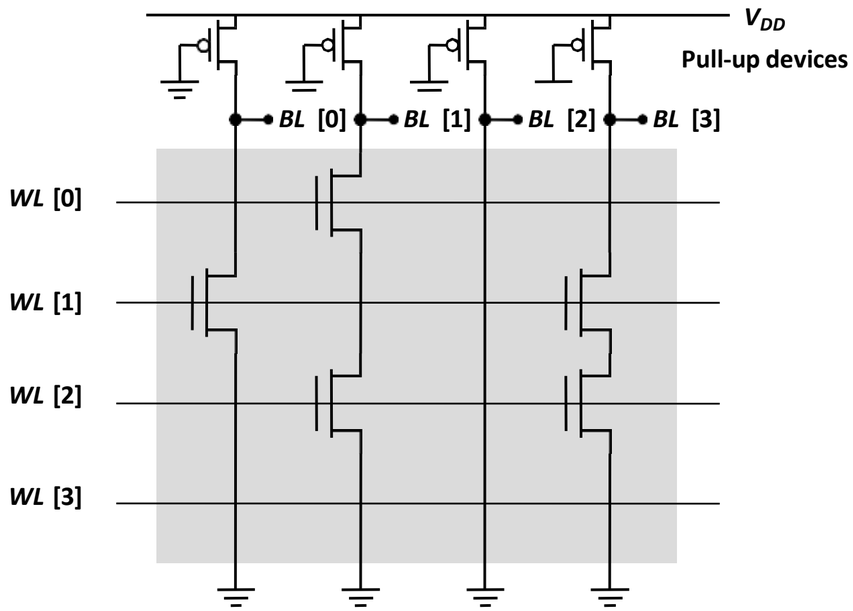
- Fuse ROM

EEPROM

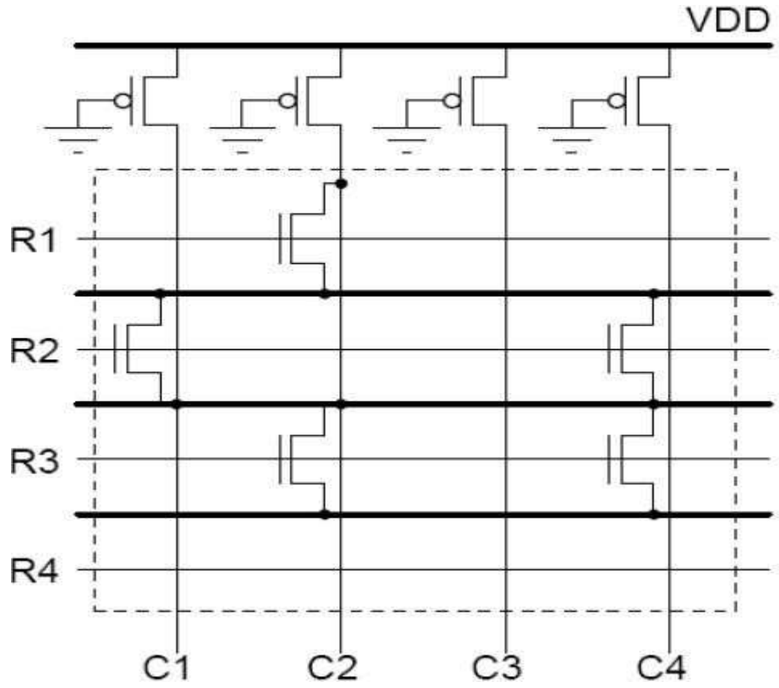




NAND-based ROM Array



NOR-based ROM Array



All word lines high by default with exception of selected row

TRUTH TABLES

NAND-based ROM Array

R1	R2	R3	R4	C1	C2	C3	C4
0	1	1	1	0	1	0	1
1	0	1	1	0	0	1	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	0

NOR-based ROM Array

R1	R2	R3	R4	C1	C2	C3	C4
1	0	0	0	0	1	0	1
0	1	0	0	0	0	1	1
0	0	1	0	1	0	0	1
0	0	0	1	0	1	1	0

ROM EXAMPLE

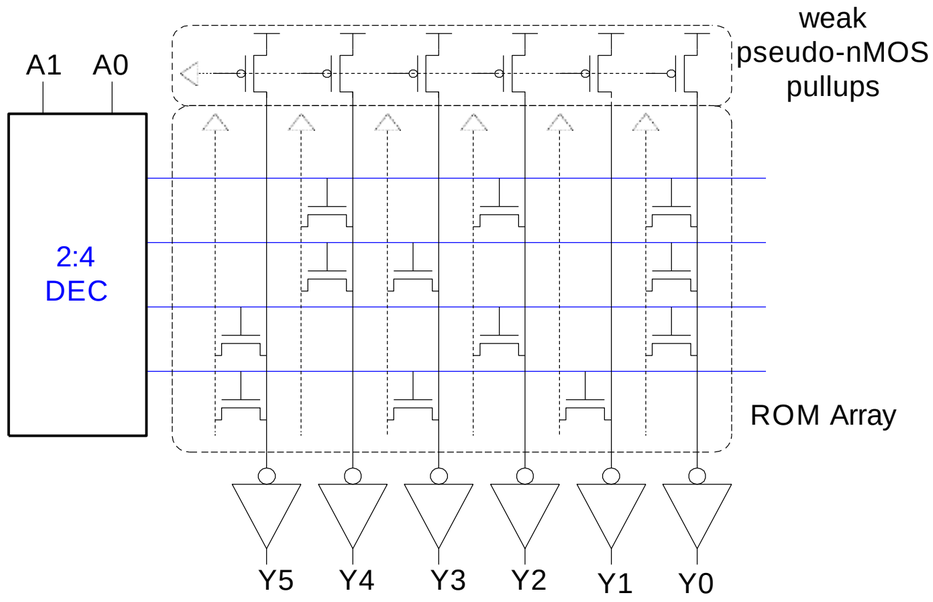
- 4-word x 6-bit ROM
  - Represented with dot diagram
  - Dots indicate 1's in ROM

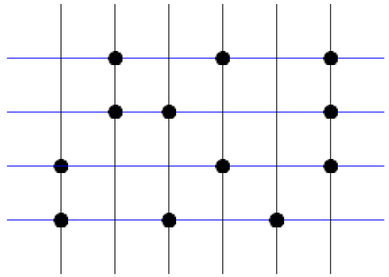
Word 0: **010101**

Word 1: **011001**

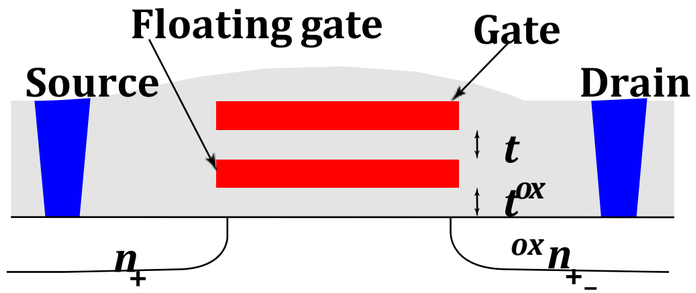
Word 2: **100101**

Word 3: **101010**



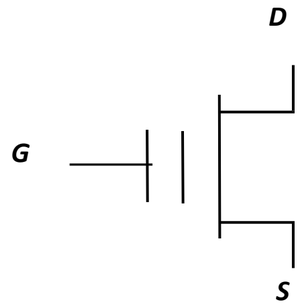


NON-VOLATILE MEMORIES
THE FLOATING-GATE TRANSISTOR (FAMOS)

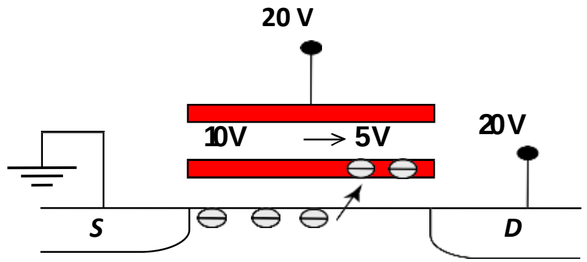


p Substrate

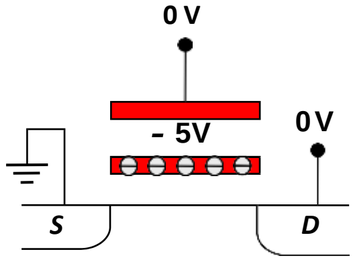
Device cross-section



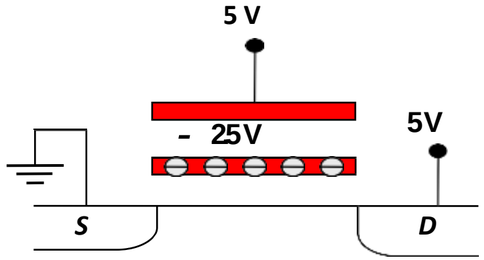
Schematic symbol



Avalanche injection



Removing programming voltage leaves charge trapped



Programming results in higher V_T

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

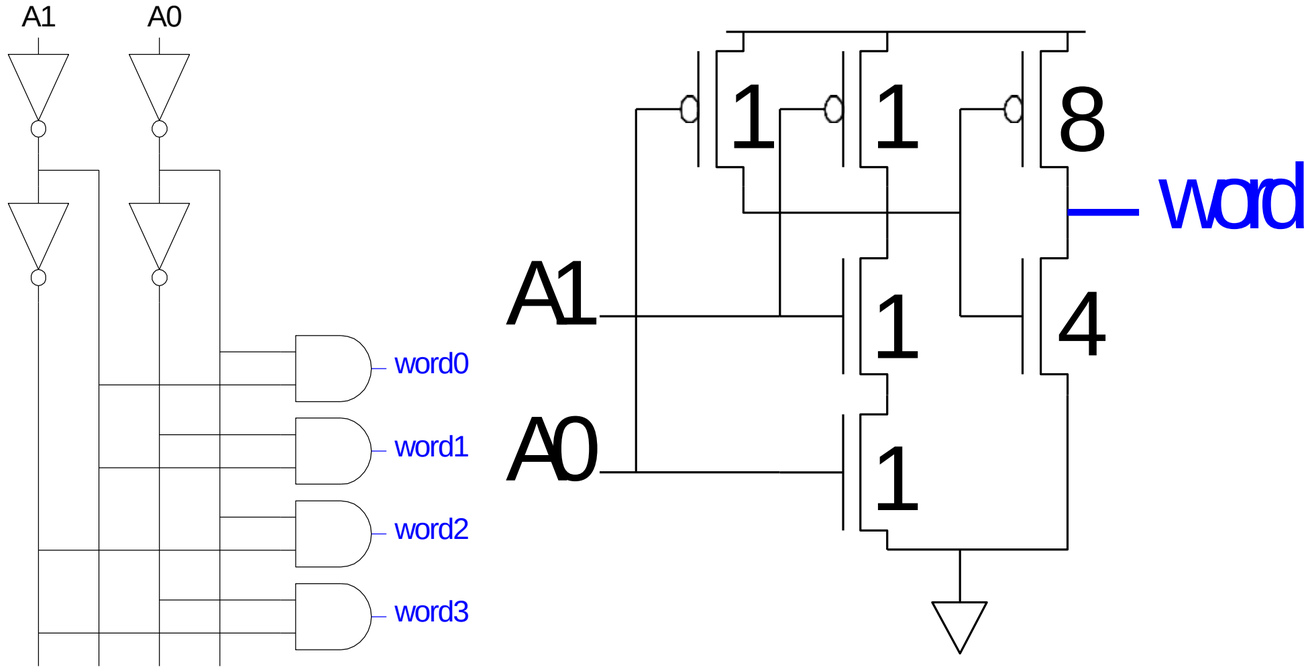
$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

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

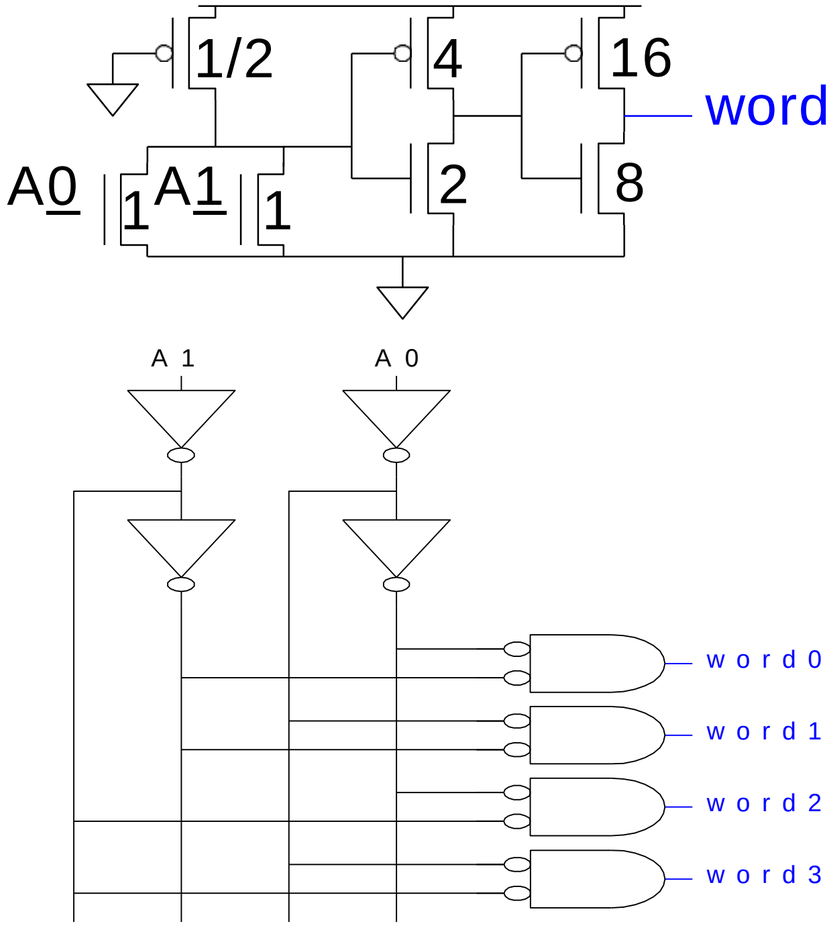
DECODERS

- n:2^n decoder consists of 2^n n-input AND gates
  - One needed for each row of memory
  - Build AND from NAND or NOR gates

Static CMOS

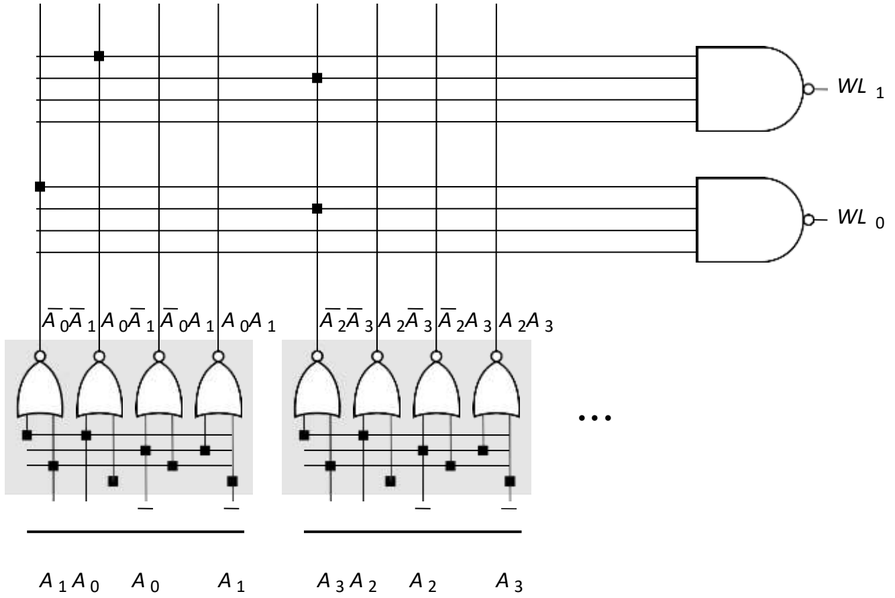


Pseudo-nMOS

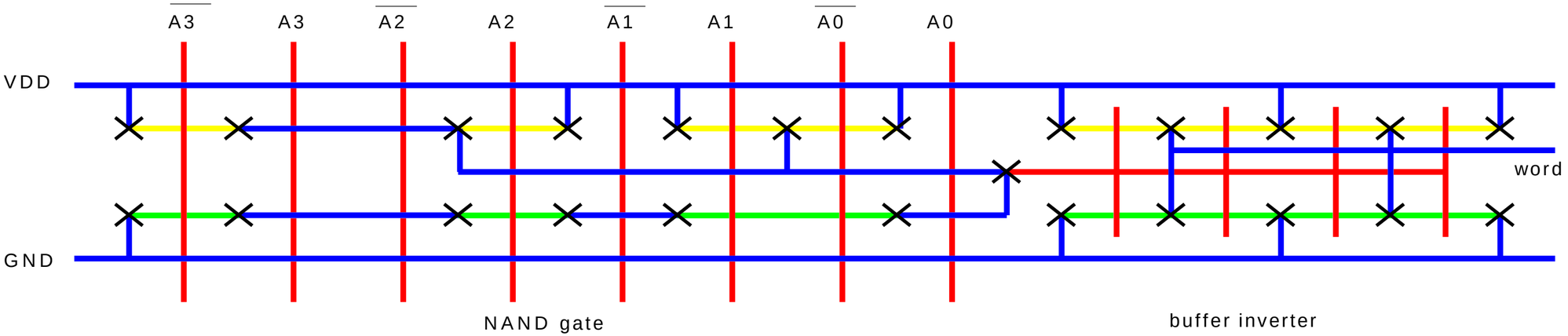


DECODER LAYOUT & HIERARCHICAL DECODERS

NAND decoder using
2-input pre-decoders



- Decoders must be pitch-matched to SRAM cell
  Requires very skinny gates



SENSE AMPLIFIERS

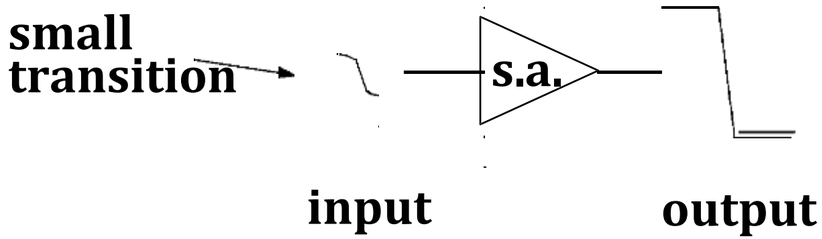
- Bitlines have many cells attached
  - Ex: 32-kbit SRAM has 256 rows x 128 cols
  - 128 cells on each bitline
- $t_{pd} \propto (C/I)\ DV$
  - Even with shared diffusion contacts, 64C of diffusion capacitance (big C)
  - Discharged slowly through small transistors (small I)
- Sense amplifiers are triggered on small voltage swing (reduce DV)

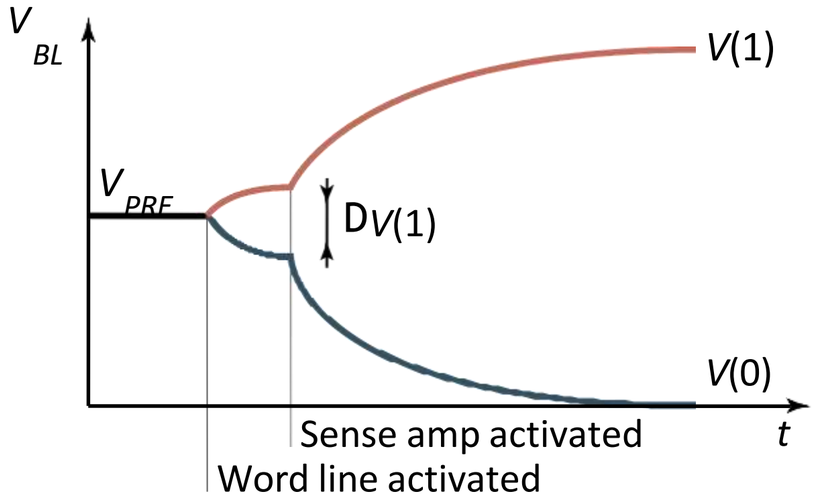
$$t_p = \frac{C \times \Delta V}{I_{av}}$$

make ΔV as small as possible

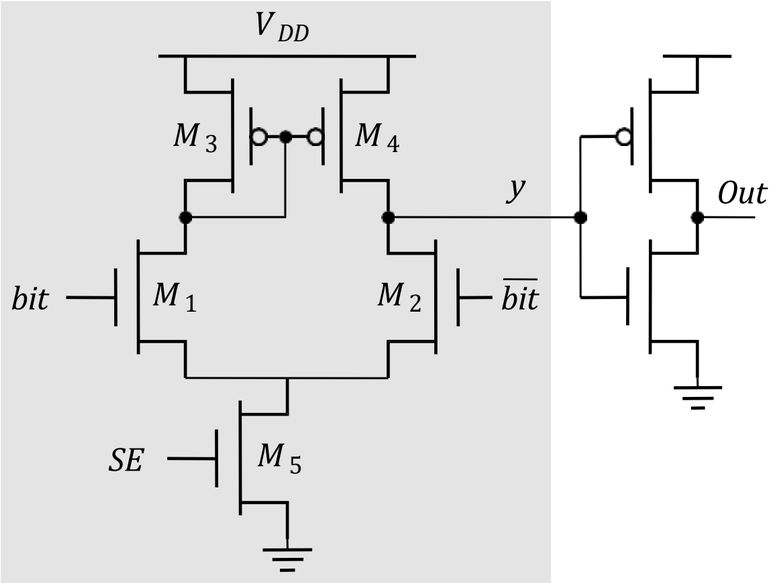
large

small





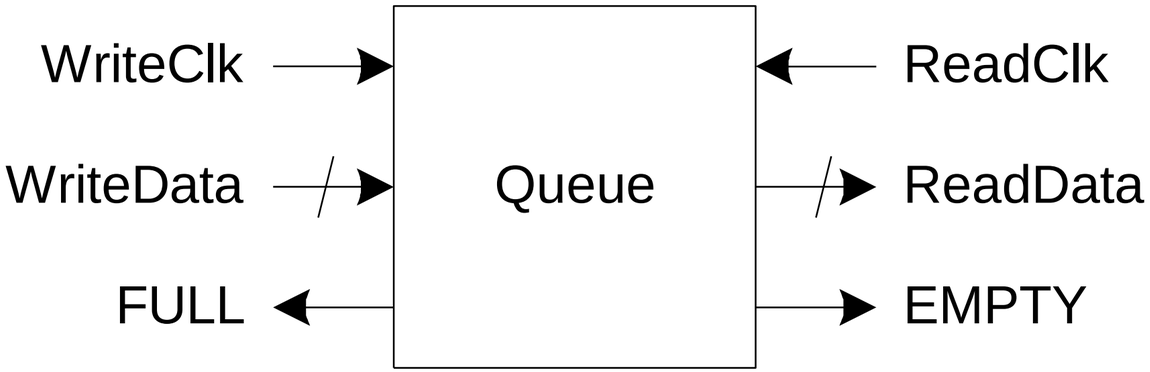
DIFFERENTIAL SENSE AMPLIFIER



Directly applicable to SRAMs

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)



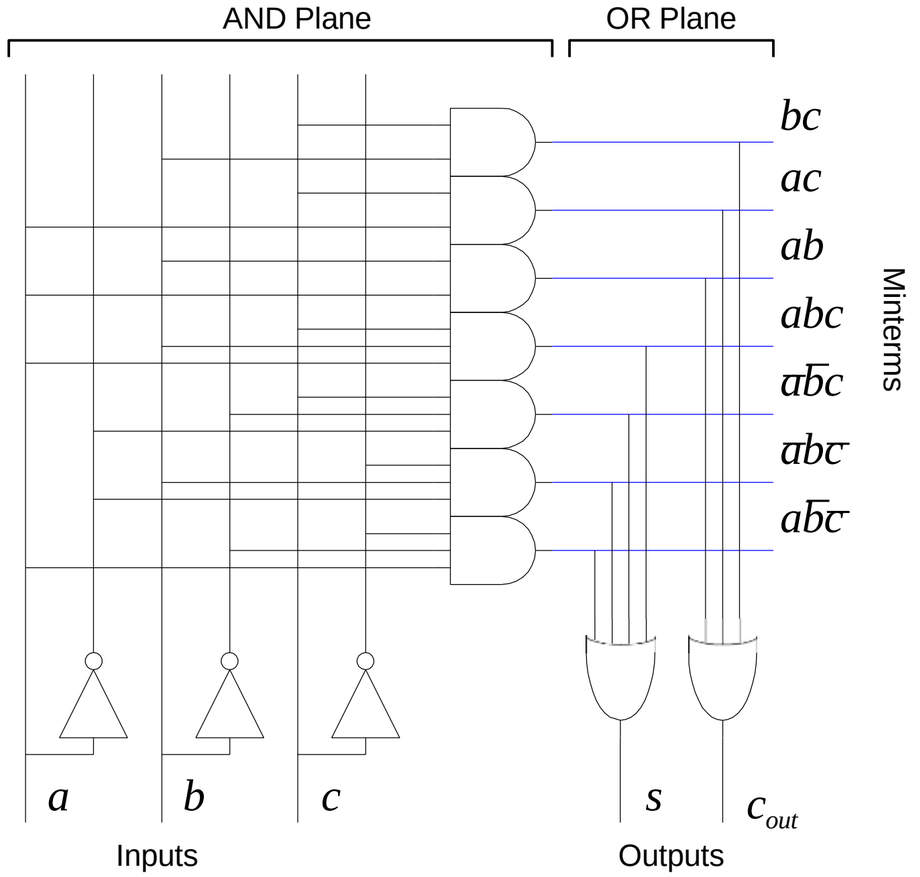
PLAs

- 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

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

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



ASSESSMENTS

1. List out the memory classification
2. Compare ROM VS RAM
3. Differentiate DRAM VS SRAM
4. Draw the 6T SRAM CELL
5. Draw the Static CAM Memory Cell
6. Compare NAND-based & NOR based ROM Array
7. Write short notes on SENSE AMPLIFIERS
8. Draw the PLAs logic diagram.

SUMMARY& THANK YOU