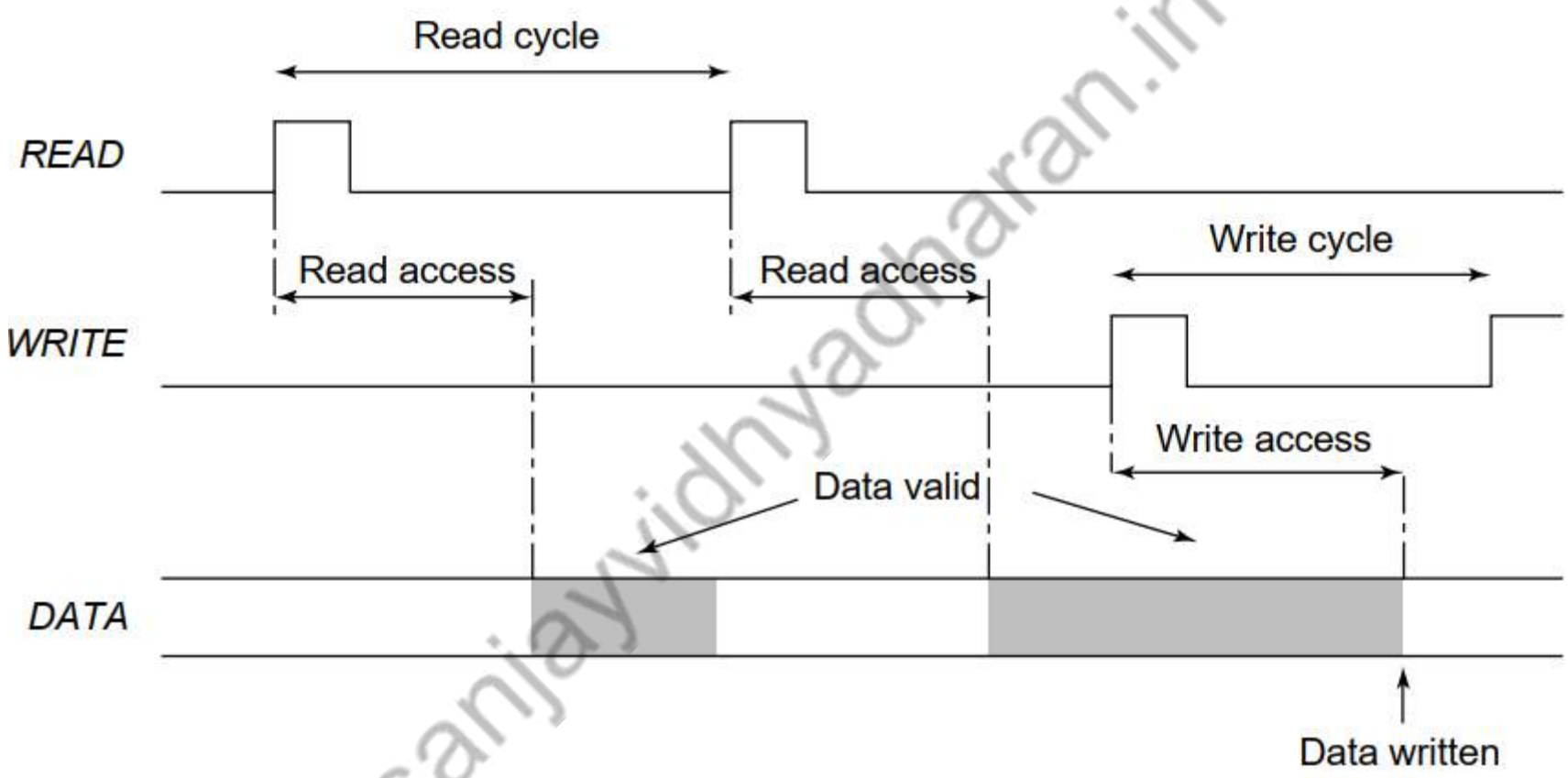
Advanced VLSI Design: 2021-22
Lecture 11: Memory Design

By Dr. Sanjay Vidhyadharan

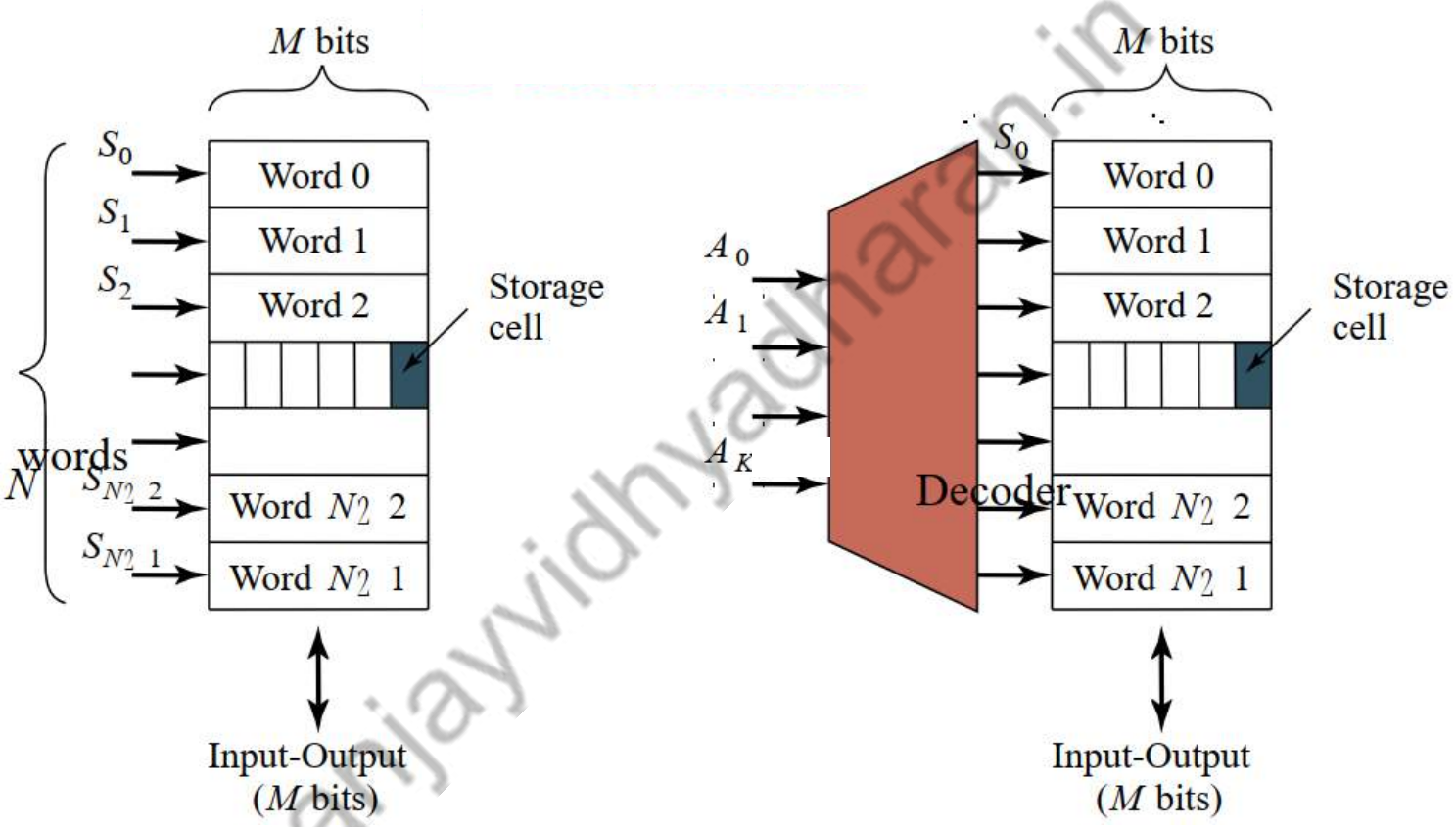
Semiconductor Memory Classification

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

Memory Timing: Definitions



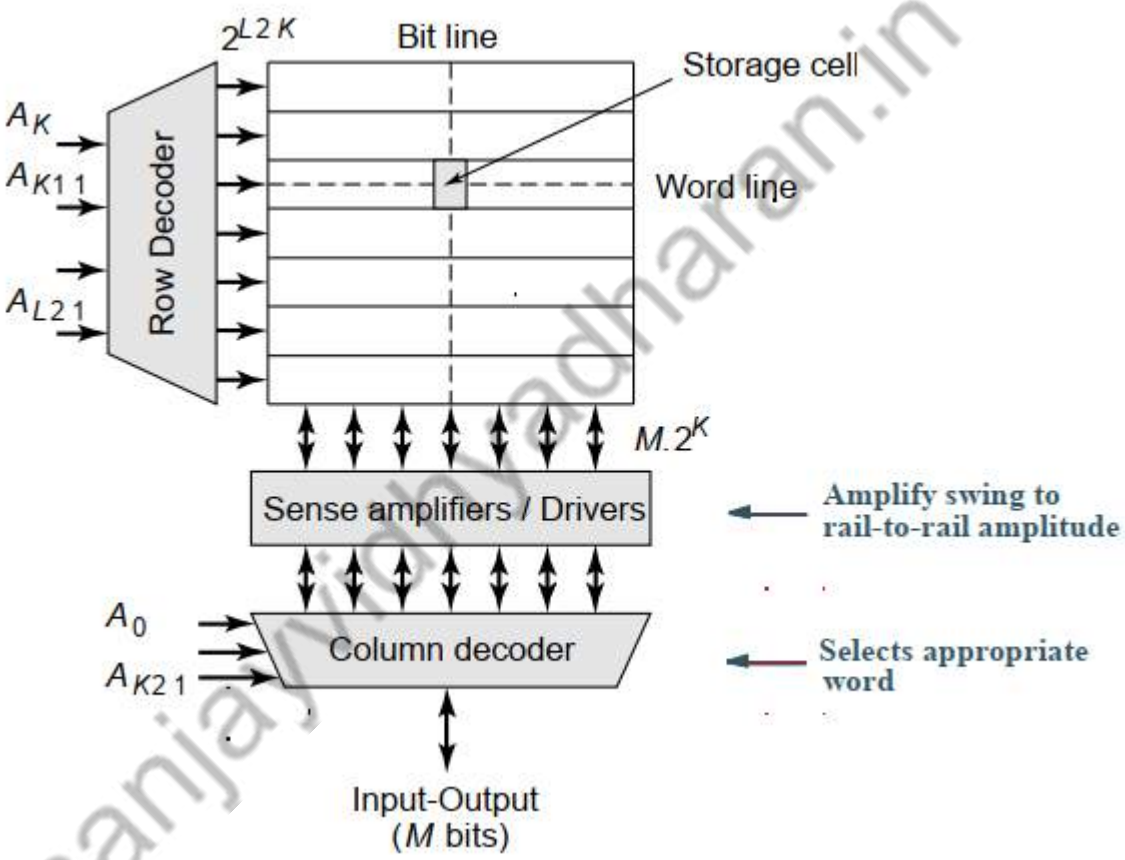
Memory Architecture



Decoder reduces the number of select signals $K = \log_2 N$

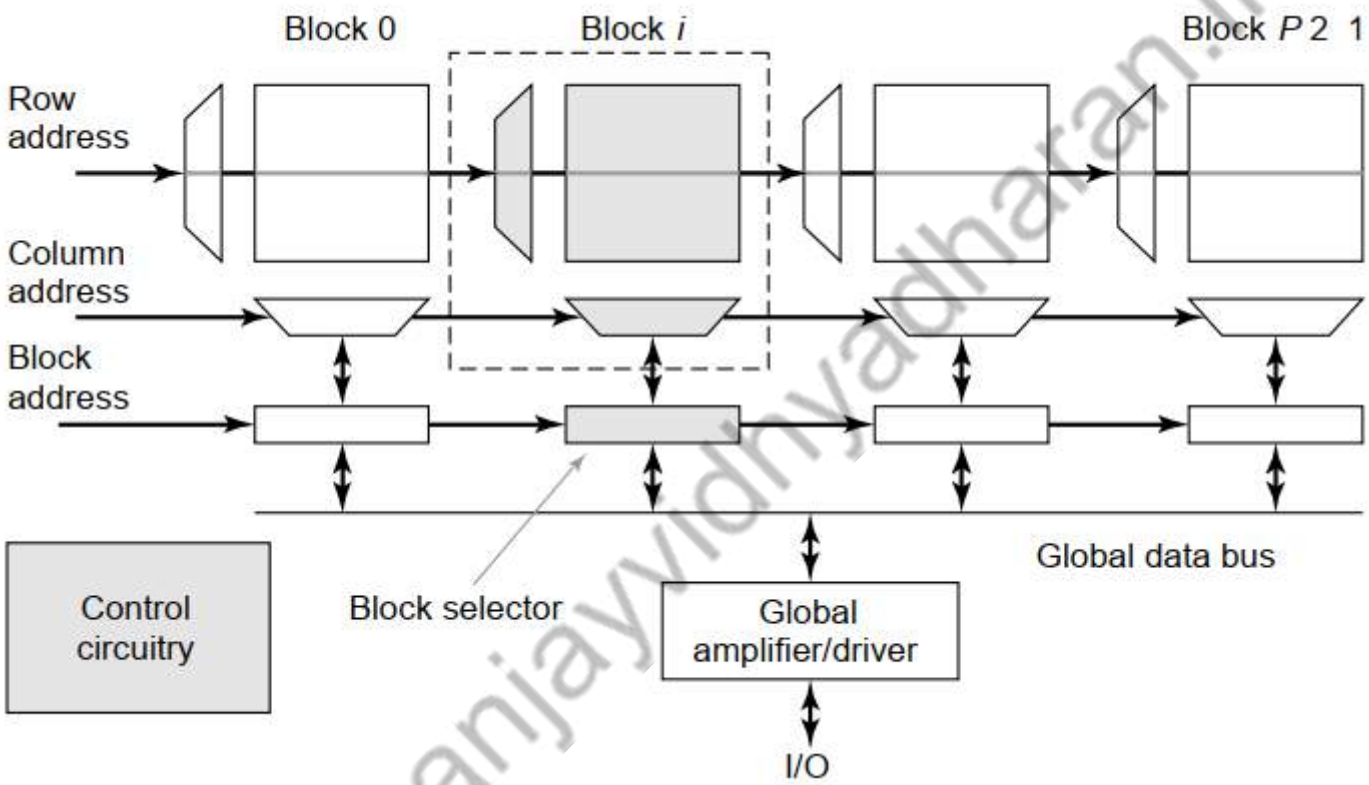
Problem: ASPECT RATIO or HEIGHT >> WIDTH

Array-Structured Memory Architecture



Better ASPECT RATIO

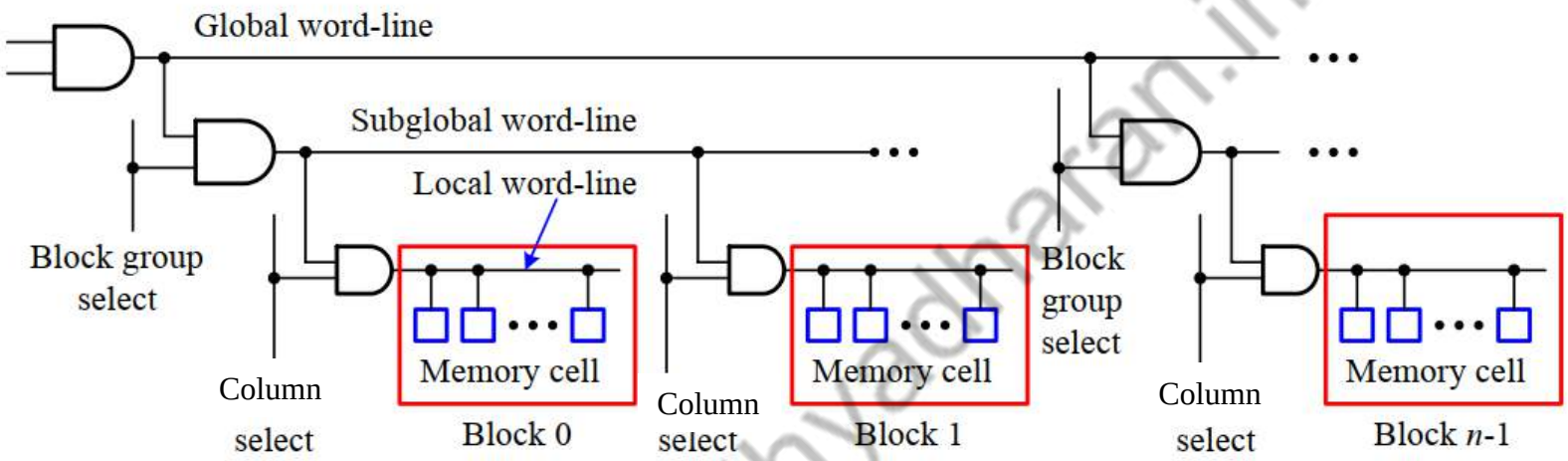
Hierarchical Memory Architecture



Advantages:

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

Hierarchical Memory Architecture



Advantages:

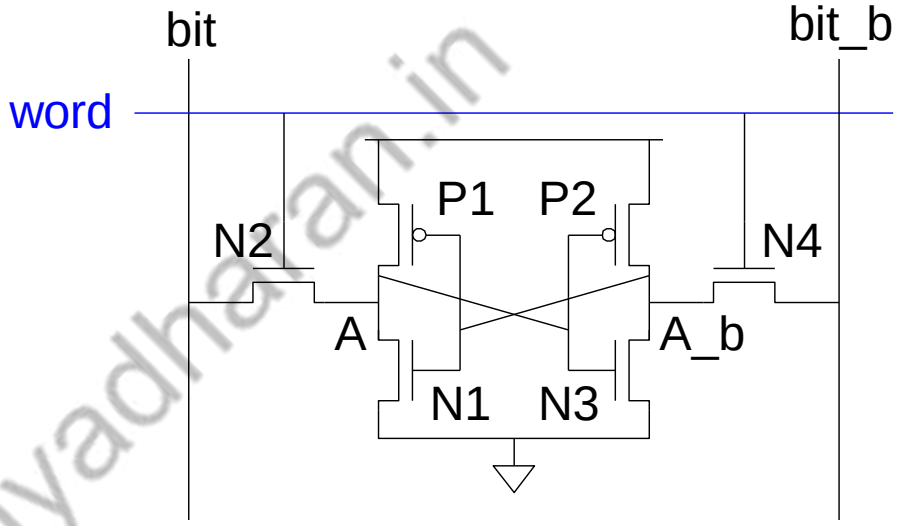
1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

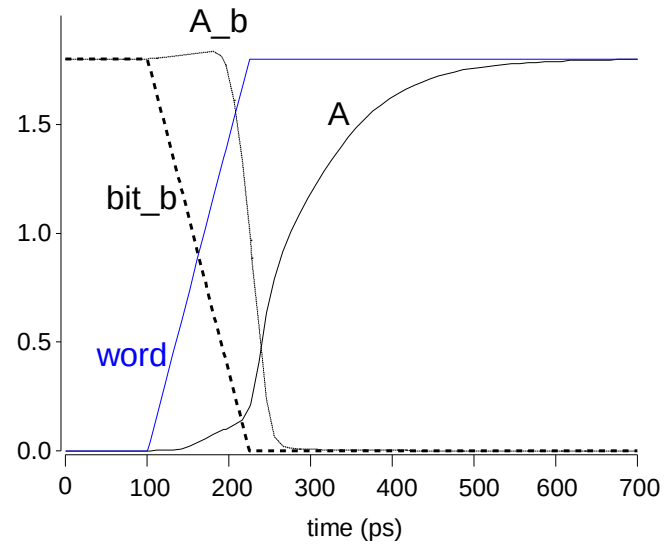
6T SRAM

Write

- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- *Writability*
  - Must overpower feedback inverter
  - N2 >> P1

Ex: A = 0, A_b = 1,
bit = 1, bit_b = 0
Force A_b low, then A rises high



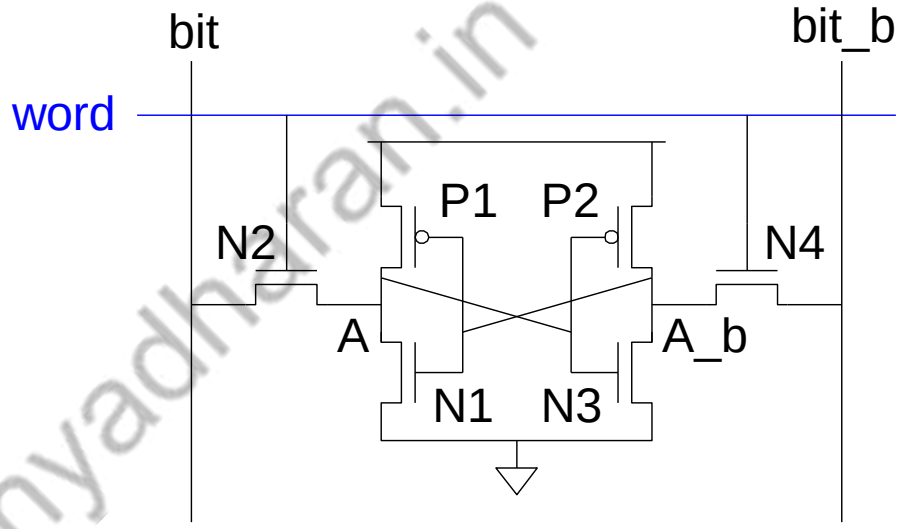


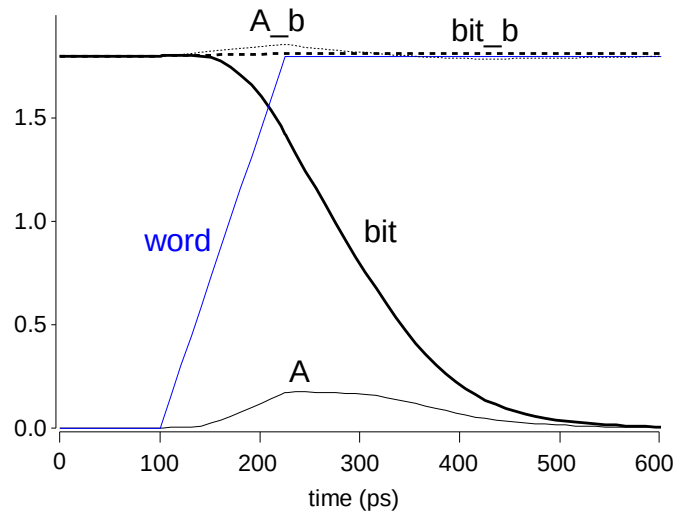
6T SRAM

Read

- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- *Read stability*
  - A must not flip
  - N1 >> N2

Ex: A = 0, A_b = 1
bit discharges, bit_b stays high
But A bumps up slightly



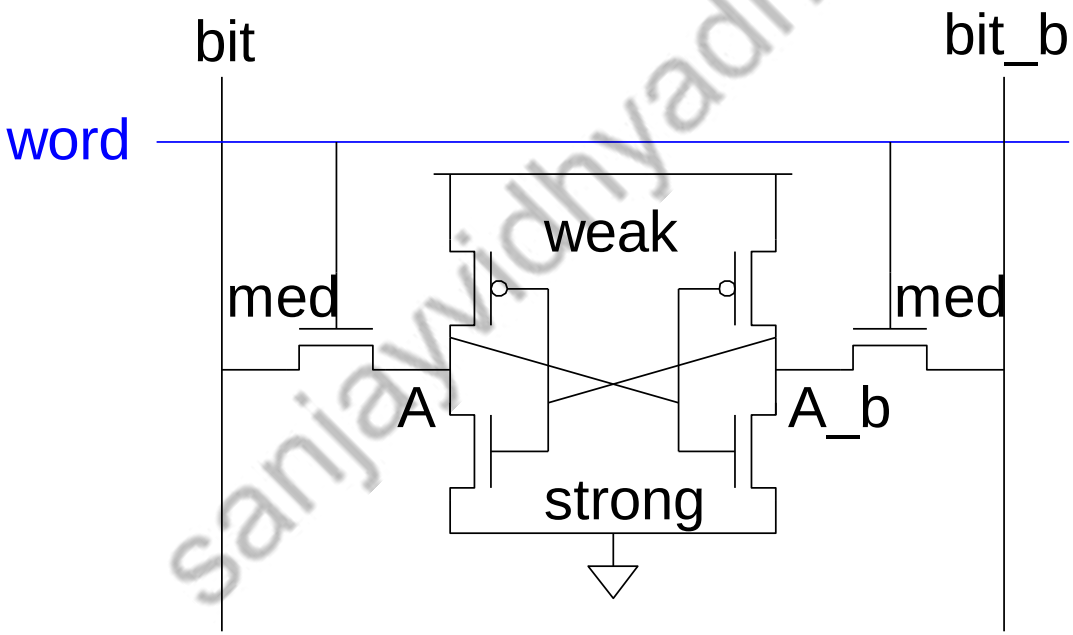


6T SRAM

SRAM Sizing

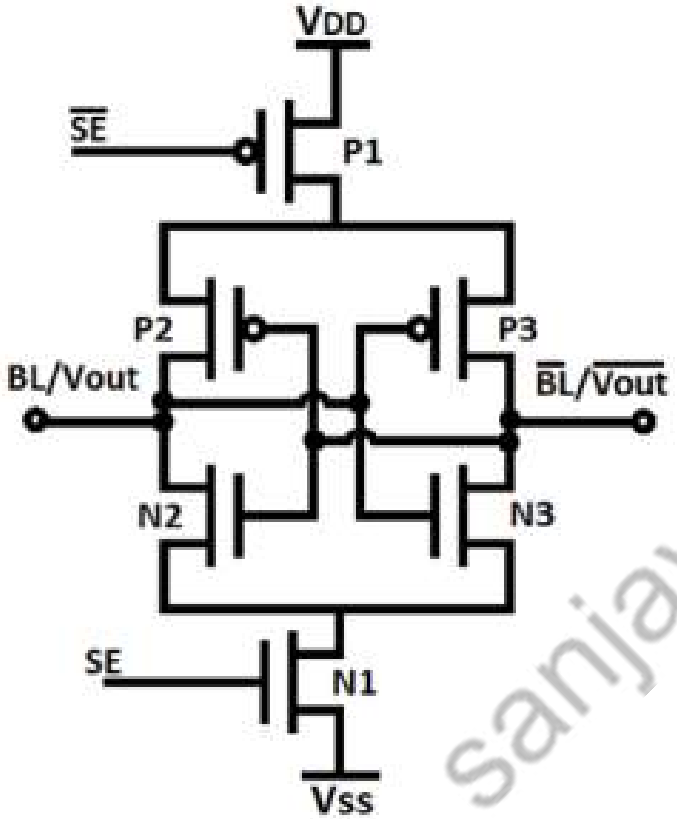
High bitlines must not overpower inverters during reads

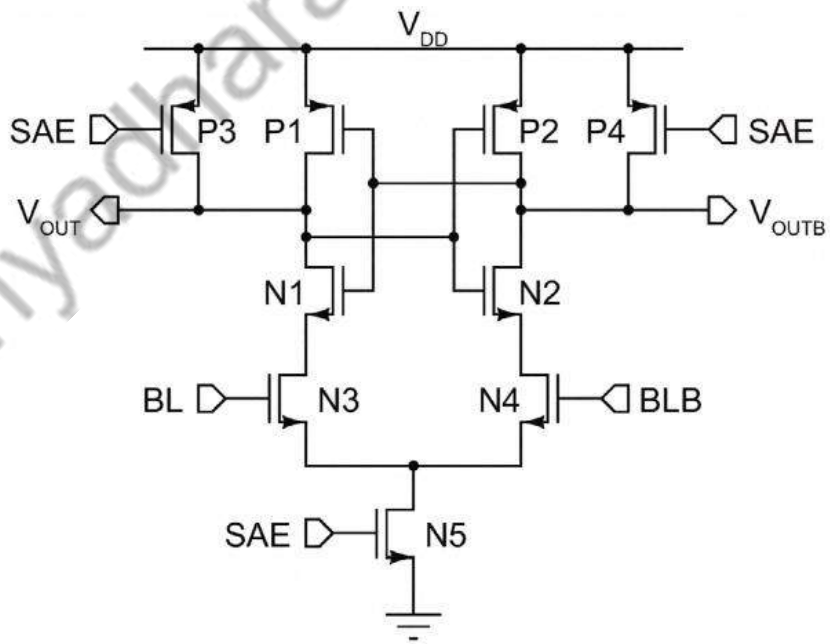
But low bitlines must write new value into cell



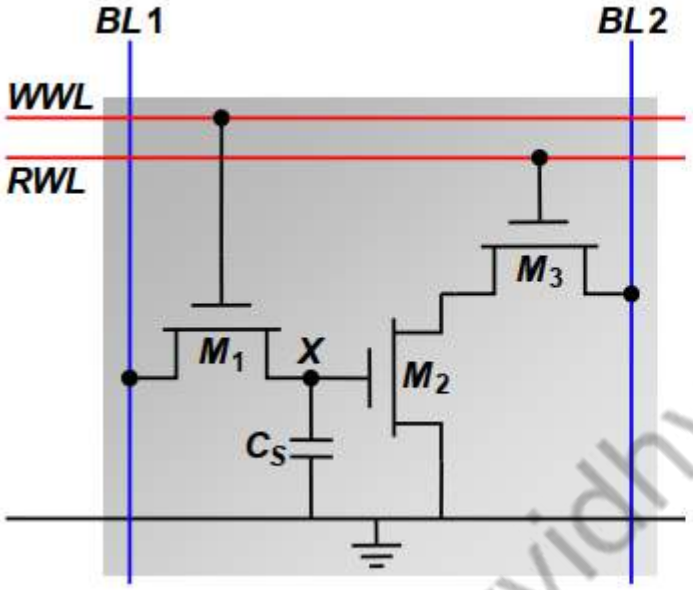
6T SRAM

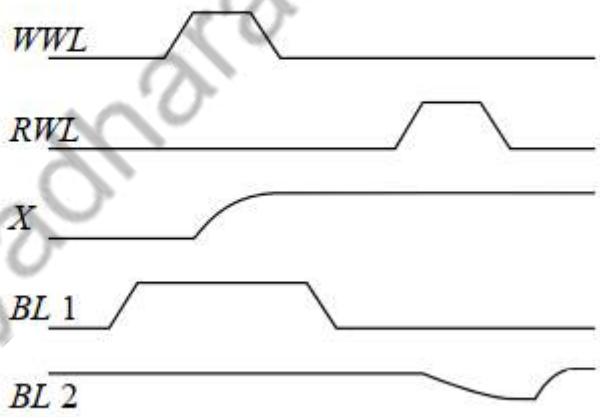
SRAM Sense Amplifier





3-Transistor DRAM Cell

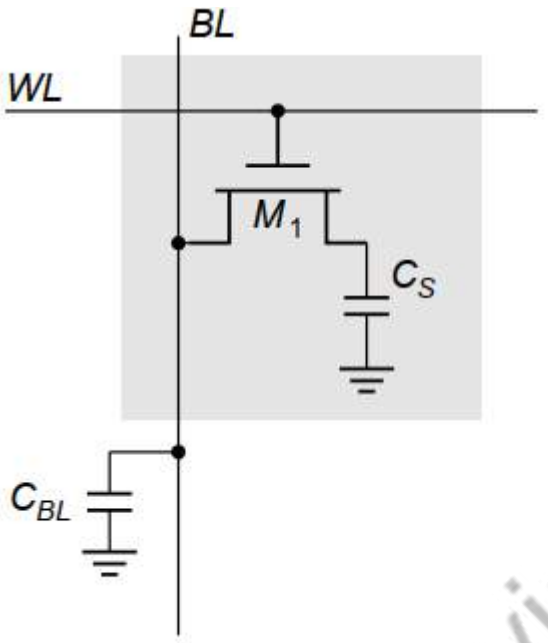


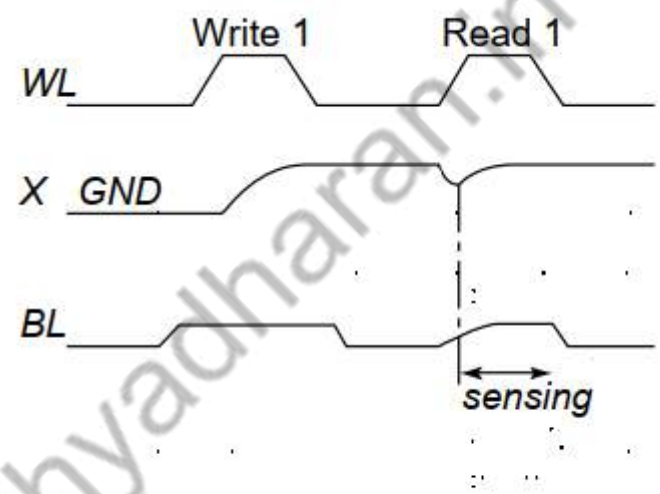


No constraints on device ratios Reads are non-destructive

Value stored at node X when writing a "1" = V_{WWL}-V_{Tn}

1-Transistor DRAM Cell





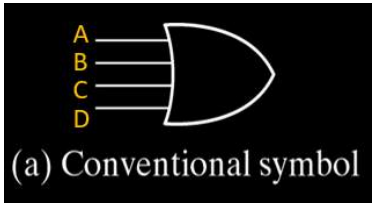
Write: CS is charged or discharged by asserting WL and BL.
Read: Charge redistribution takes places between bit line and storage capacitance

Voltage swing is small; typically around 250 mV

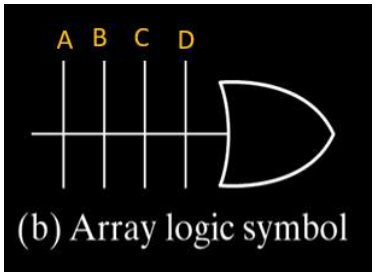
1-Transistor DRAM Cell

- 1T DRAM requires a sense amplifier for each bit line, due to charge redistribution read-out
- DRAM memory cells are single-end 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}

Read-Only Memory

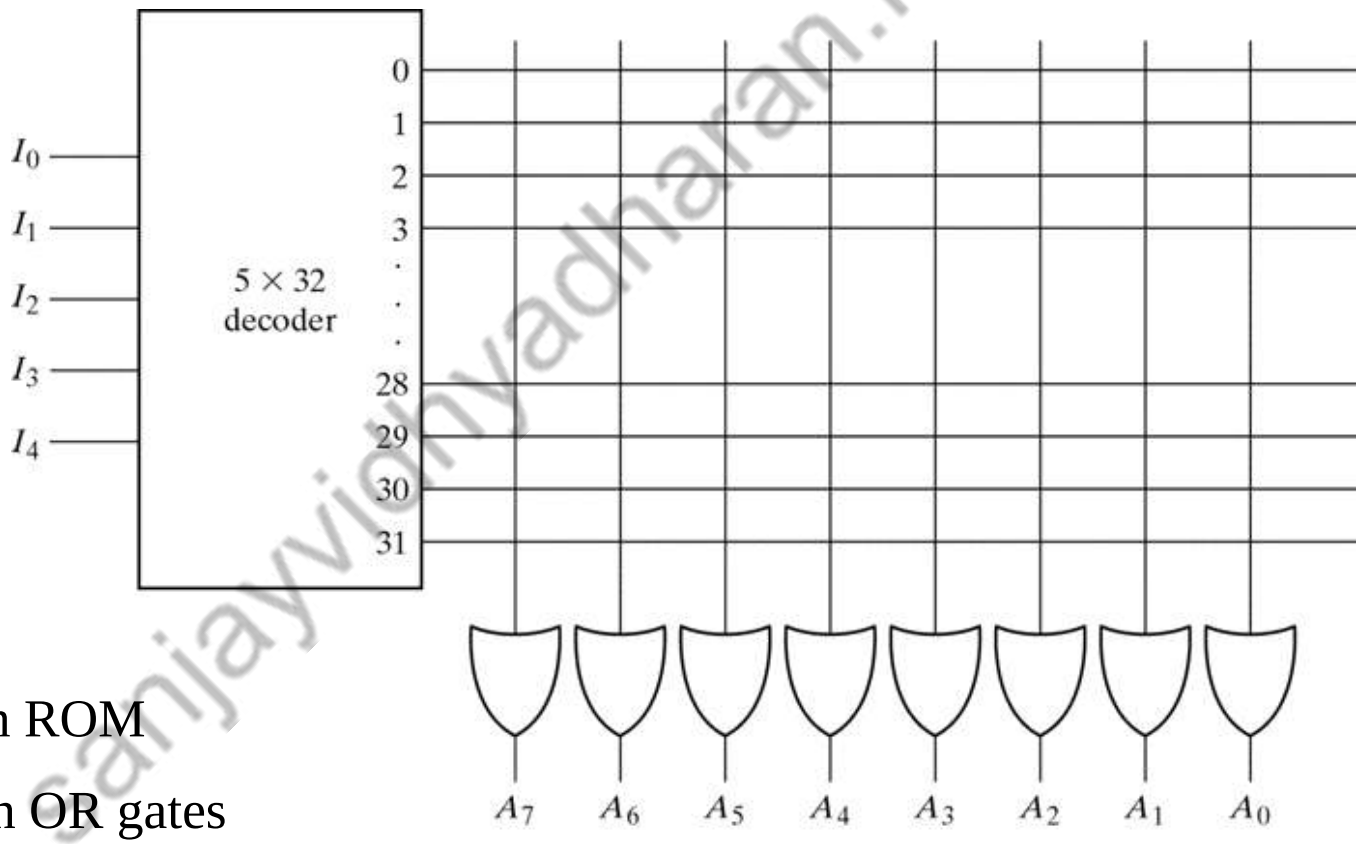


(a) Conventional symbol



(b) Array logic symbol

Internal Logic 32 X 8 ROM



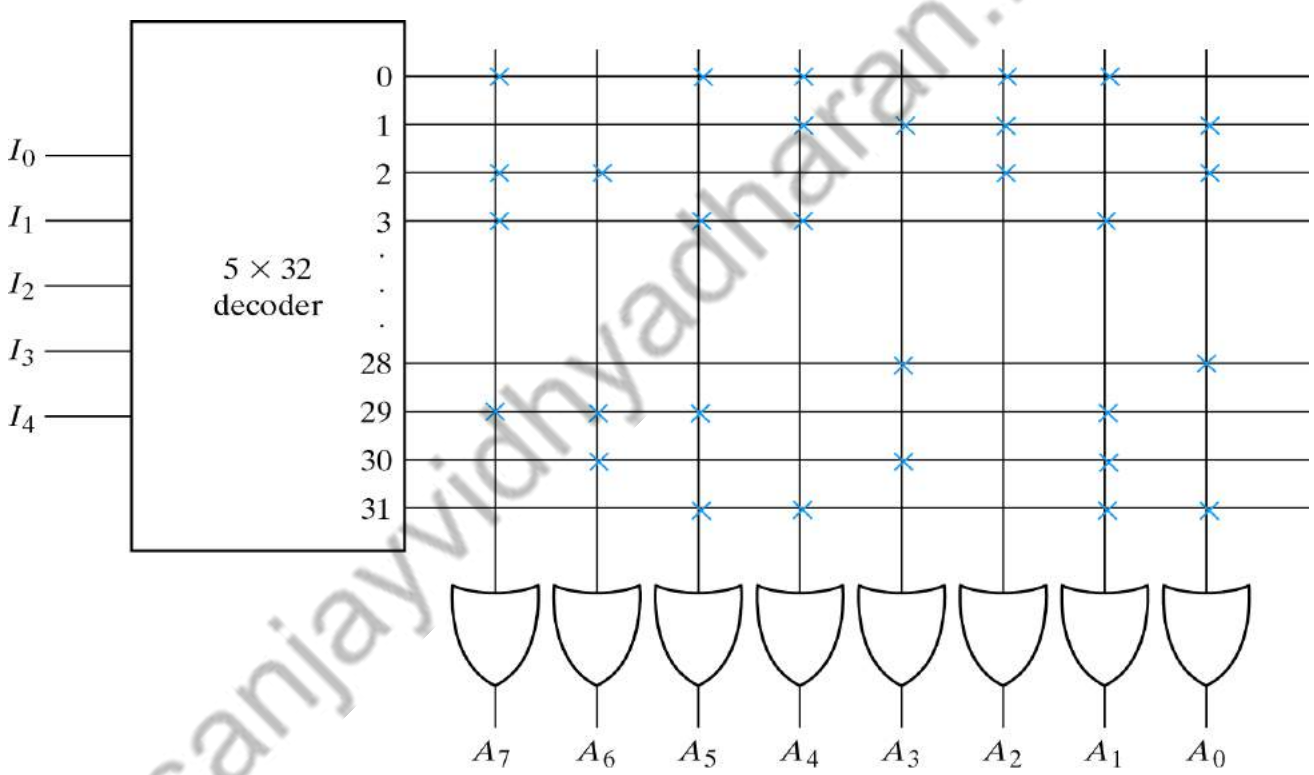
In General for 2^k X n ROM

K x 2^k Decoder and n OR gates

Each OR gate has 2^k inputs

Programming the ROM

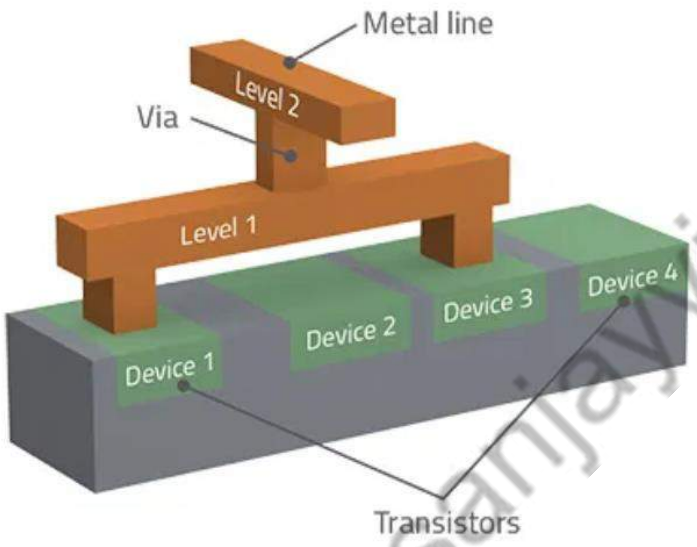
Address 3 = 10110010 is permanent storage using fuse link

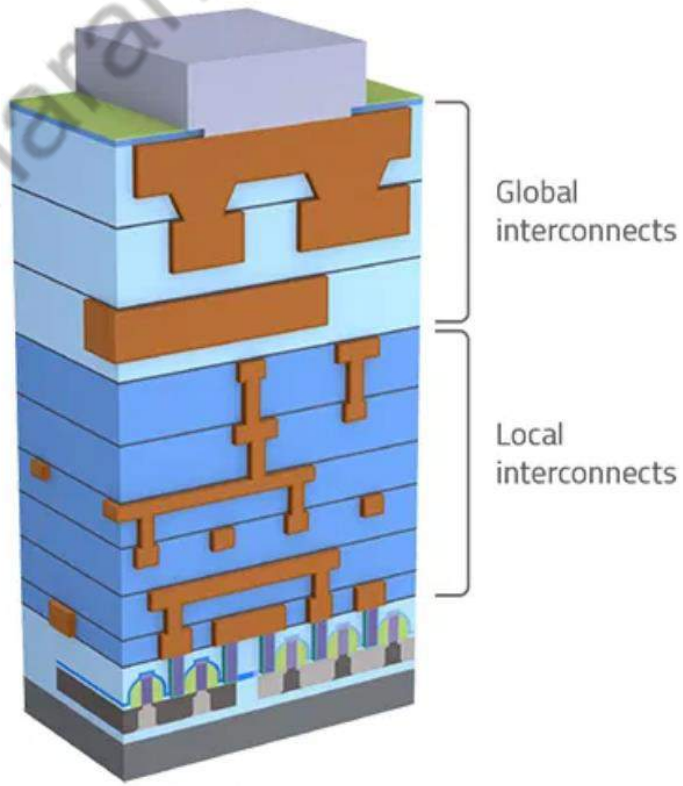


X : means connection

Programming the ROM

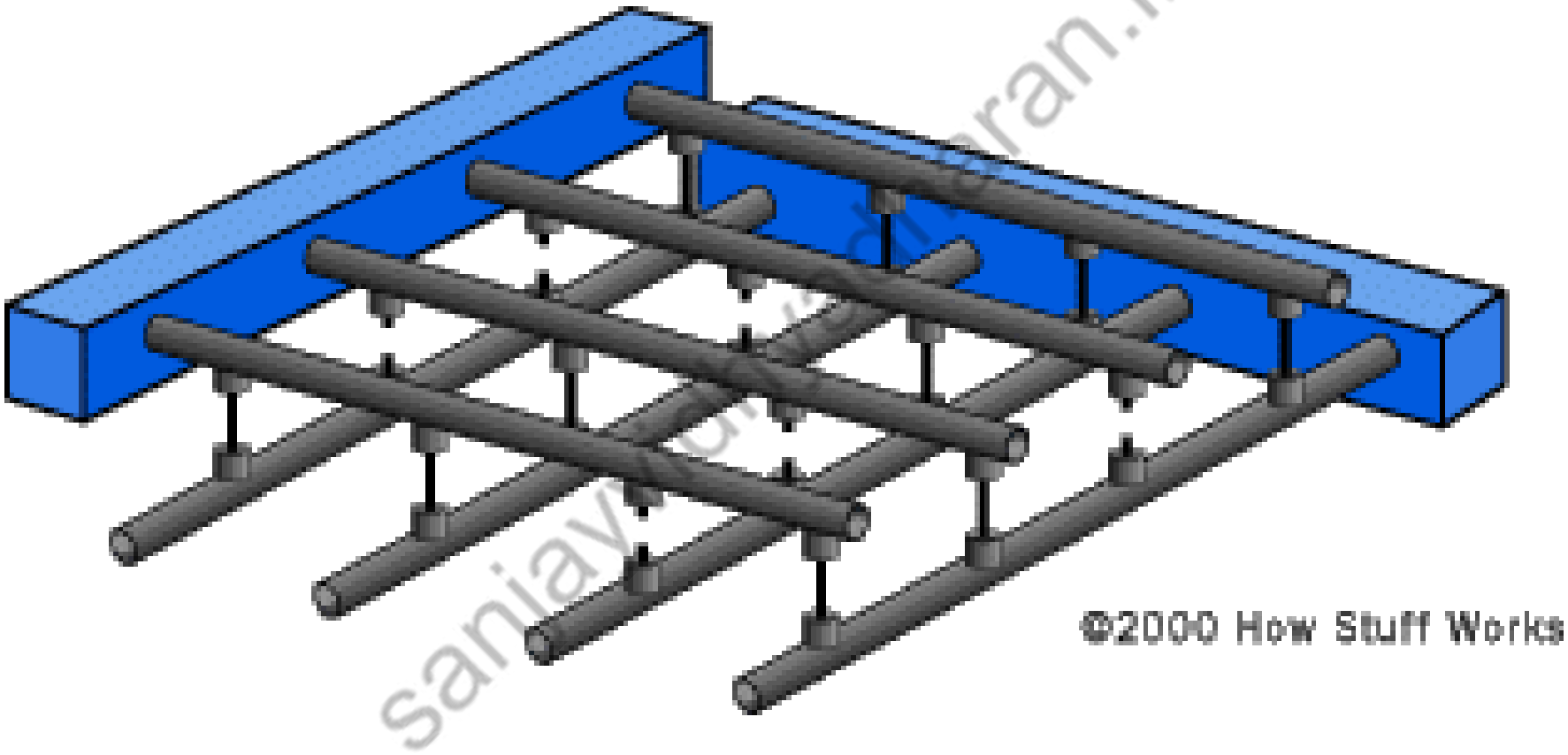
1. Masking During Metallization





Programming the ROM

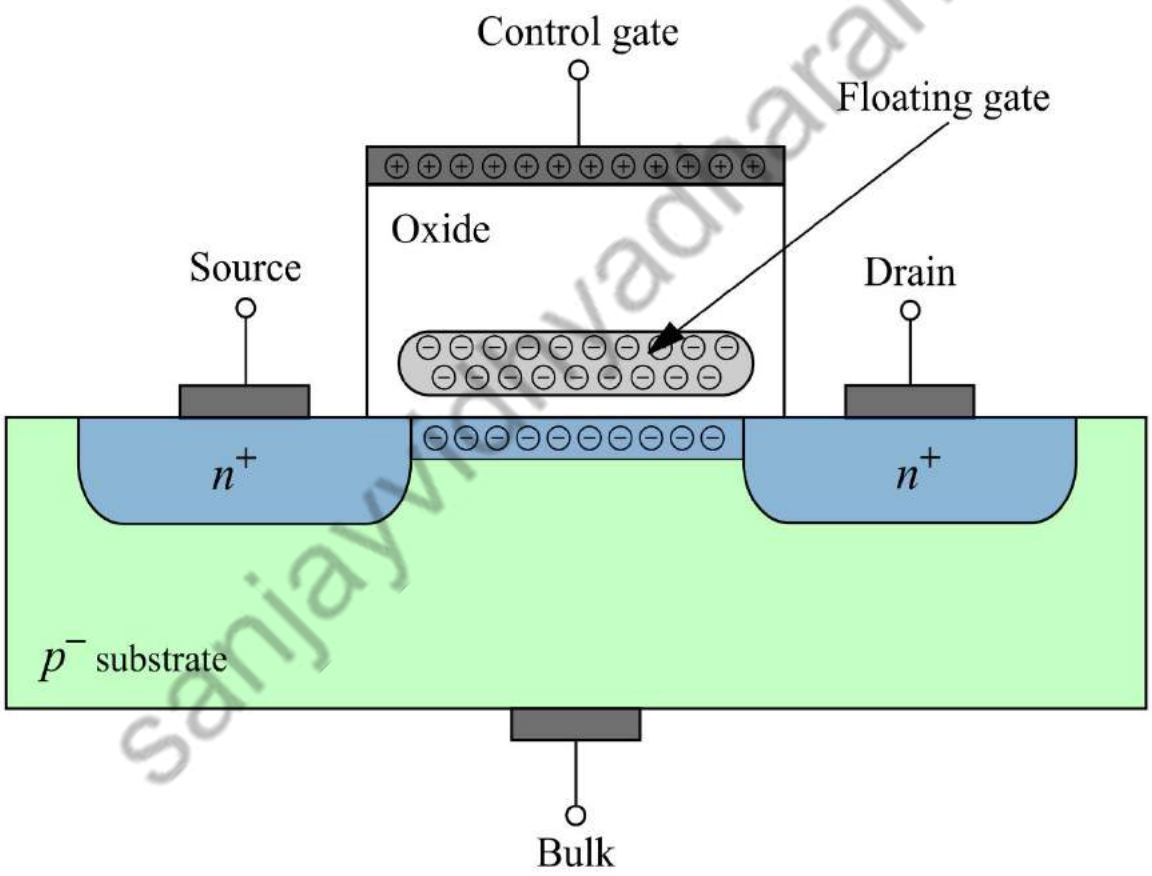
2. Fuse (PROM)



PROM

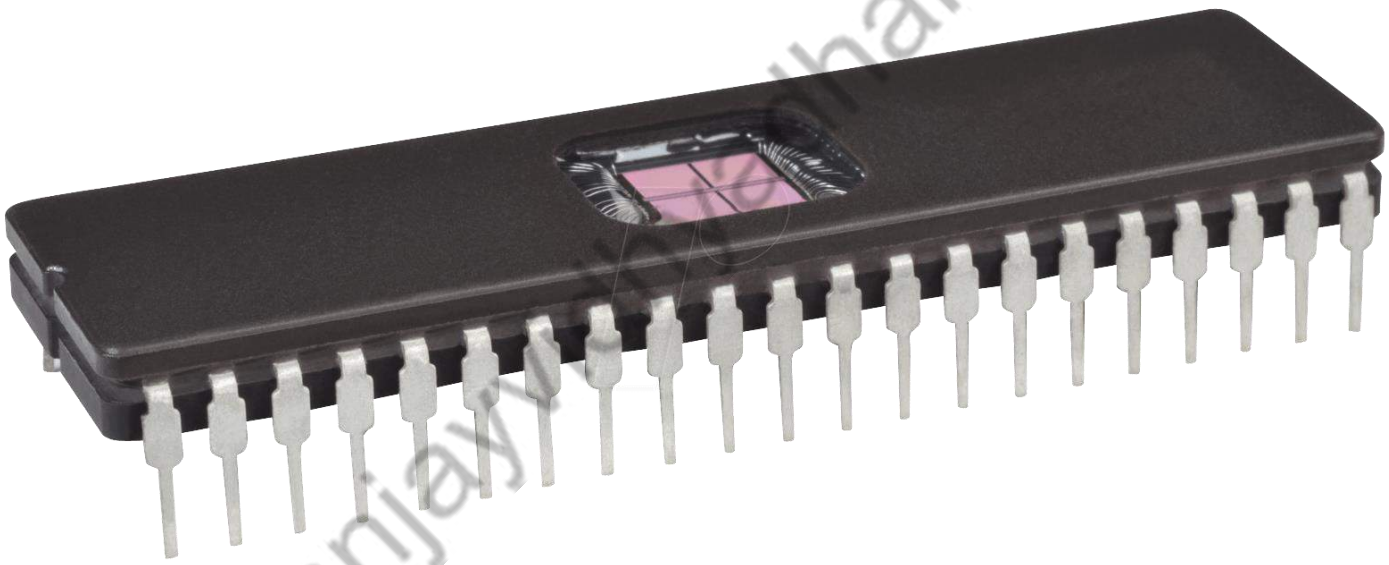
Programming the ROM

3. EPROM



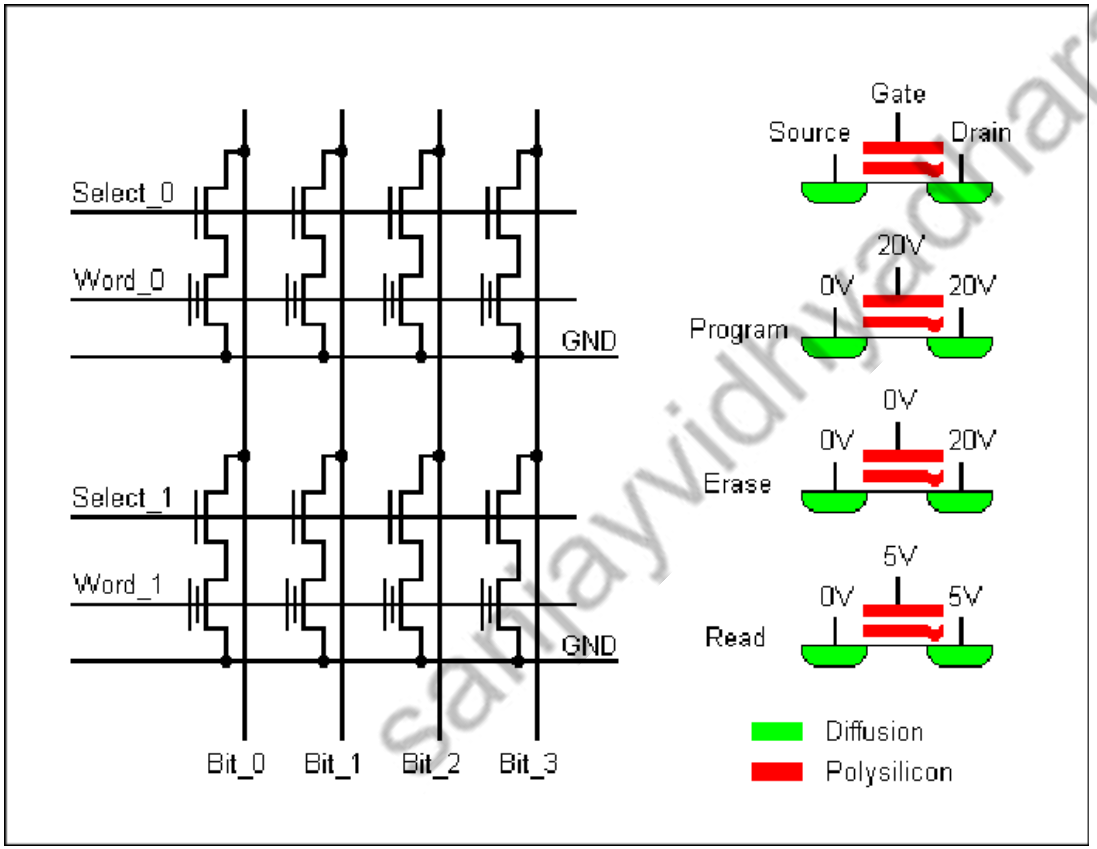
Programming the ROM

3. EPROM



Programming the ROM

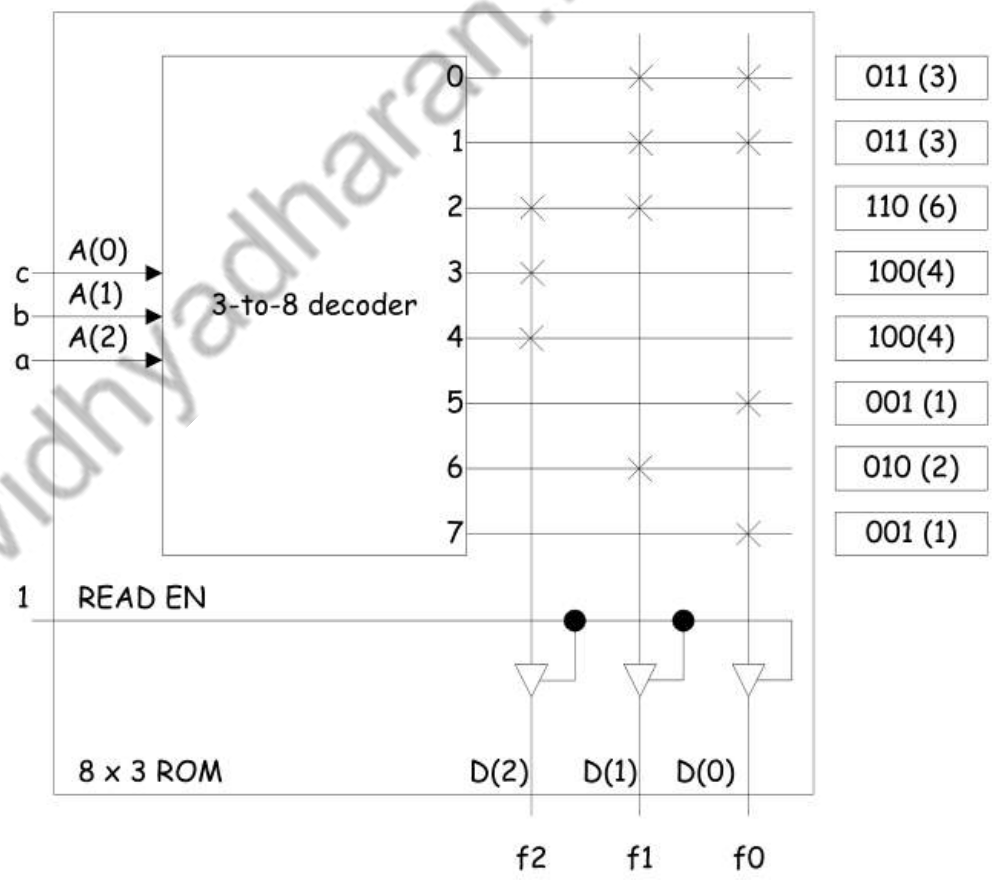
4. EEPROM



Programming the ROM

- E.g., Implement the 3-input logics $f_0 = \Sigma(0,1,5,7)$, $f_1 = \Sigma(0,1,2,6)$ and $f_2 = \Sigma(2,3,4)$ using a ROM.

a	b	c	f_2	f_1	f_0
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	0	0	1



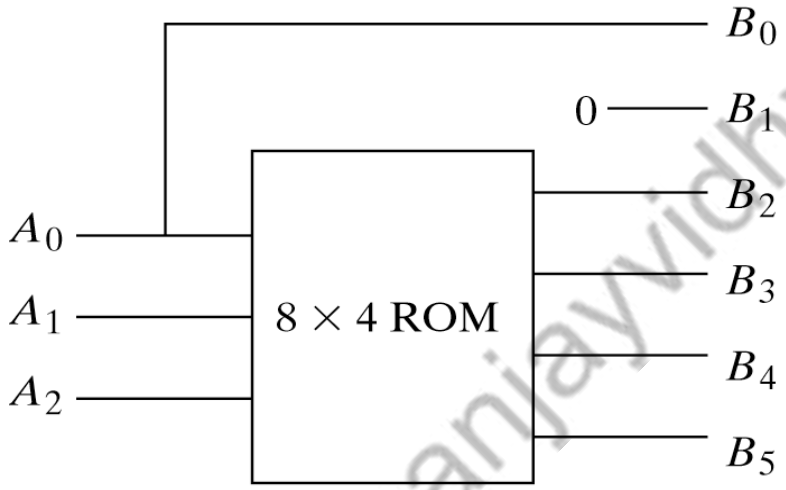
Programming the ROM

Example: Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input number.

Inputs			Outputs						
A_1	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

Programming the ROM

Example: Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input number.

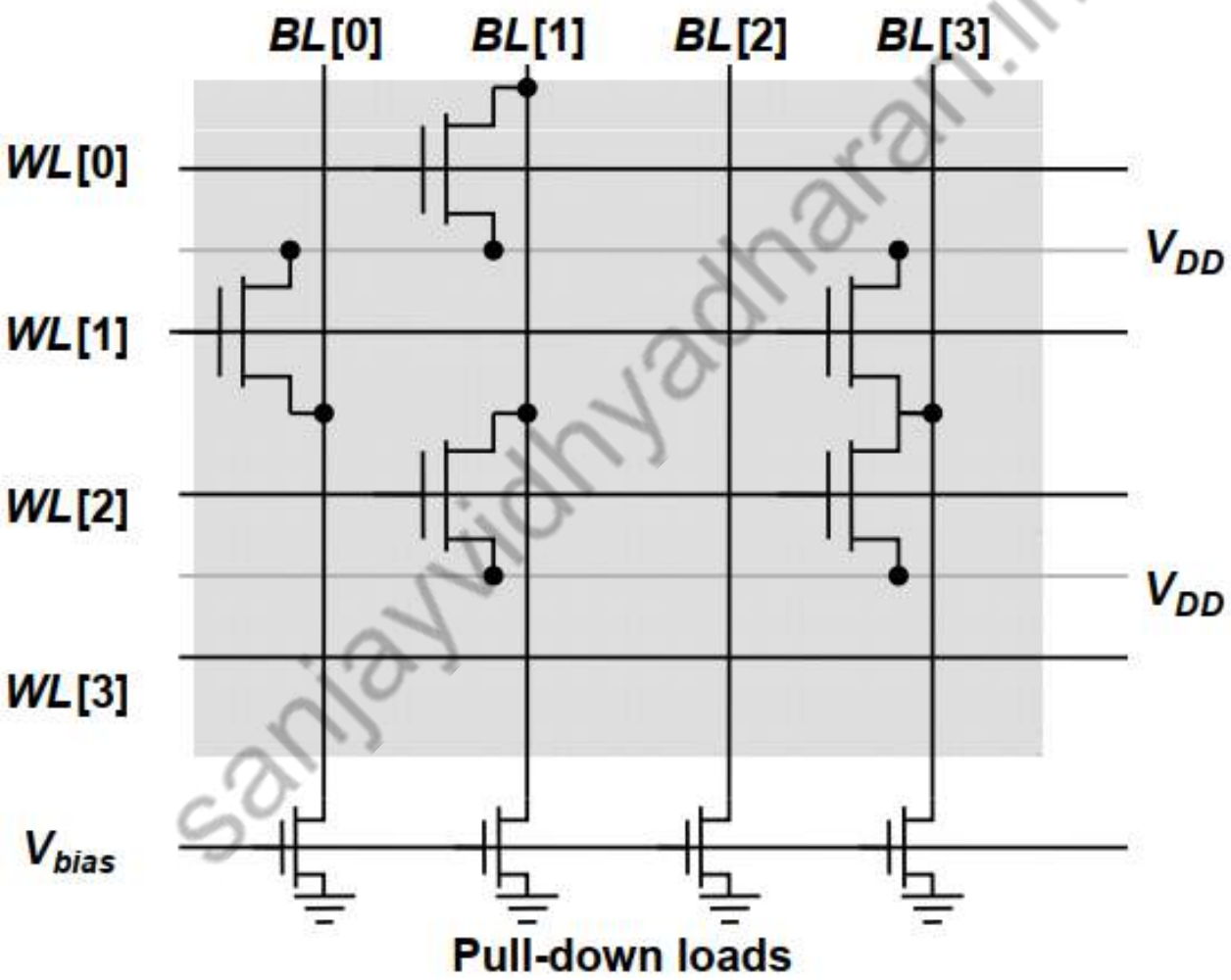


(a) Block diagram

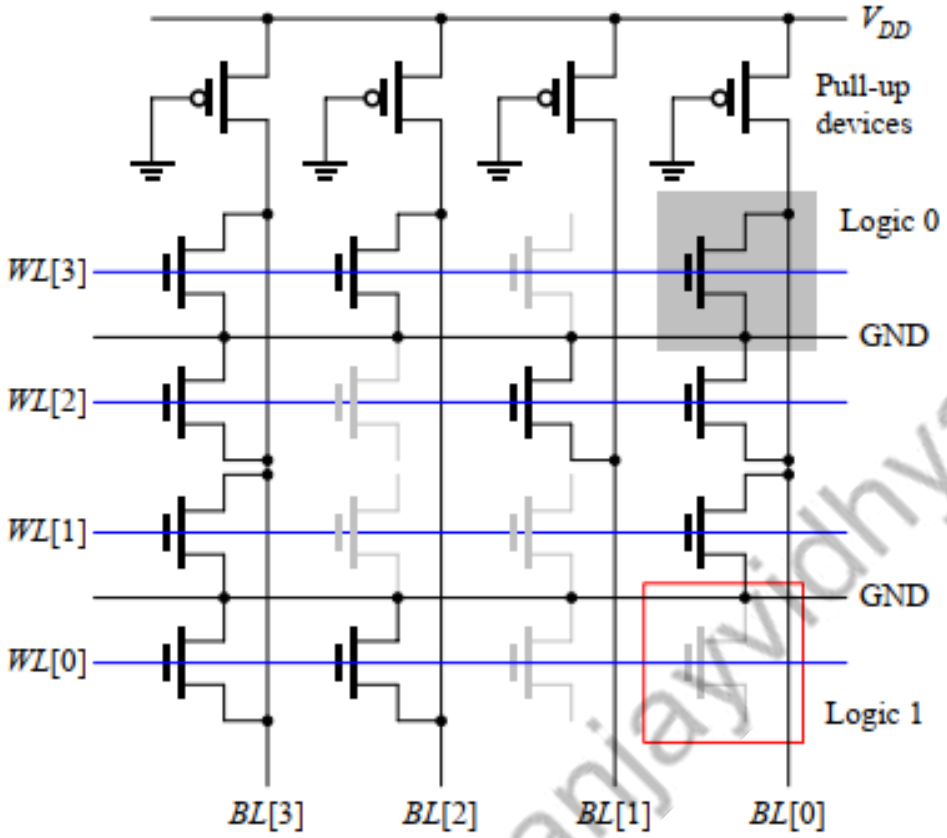
A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

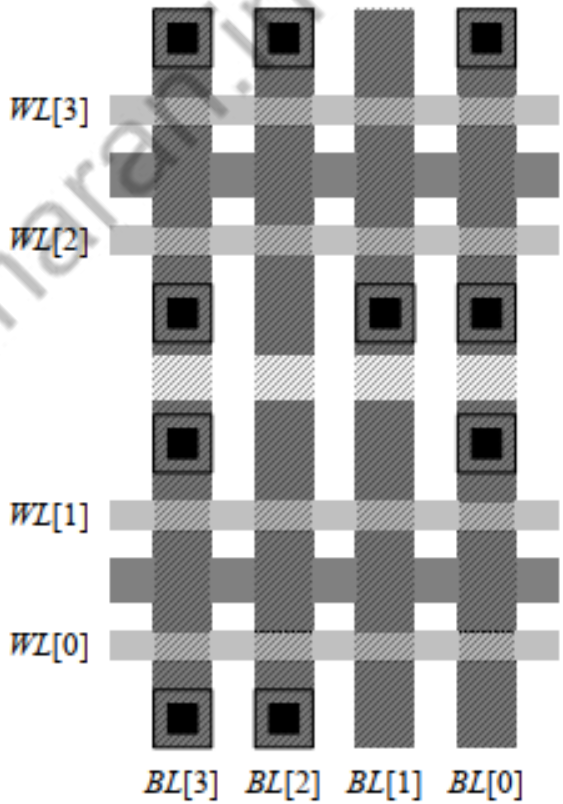
(b) ROM truth table

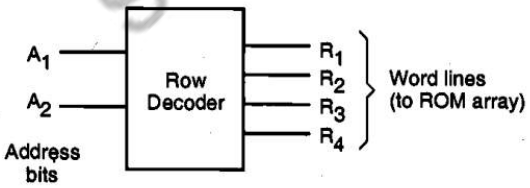
MOS OR ROM



MOS NOR ROM

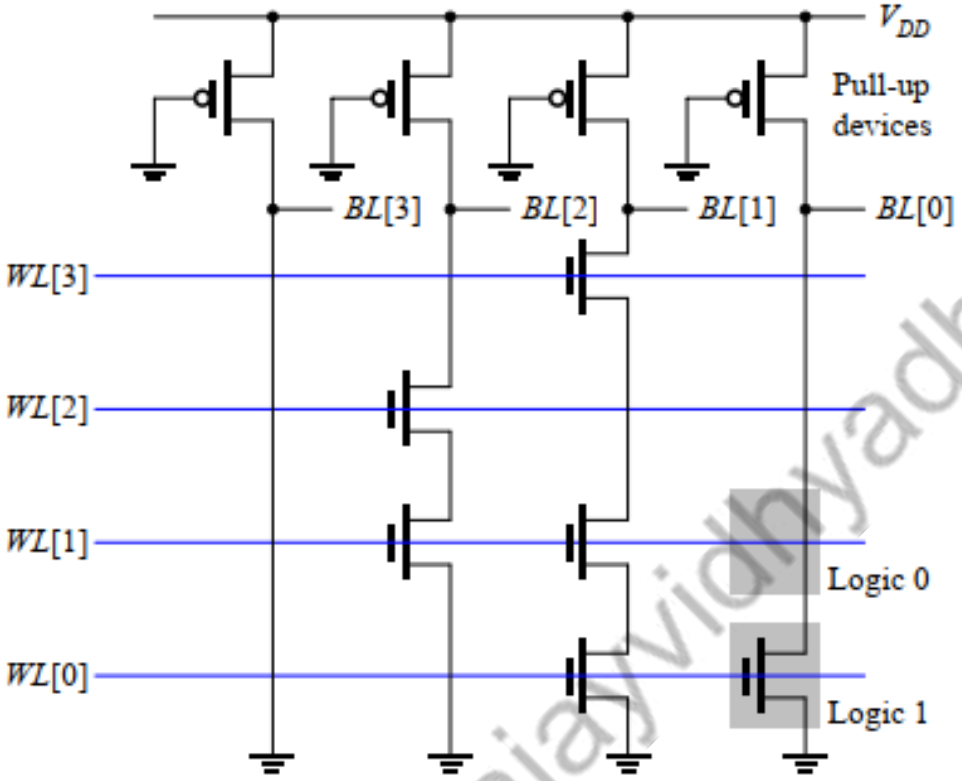




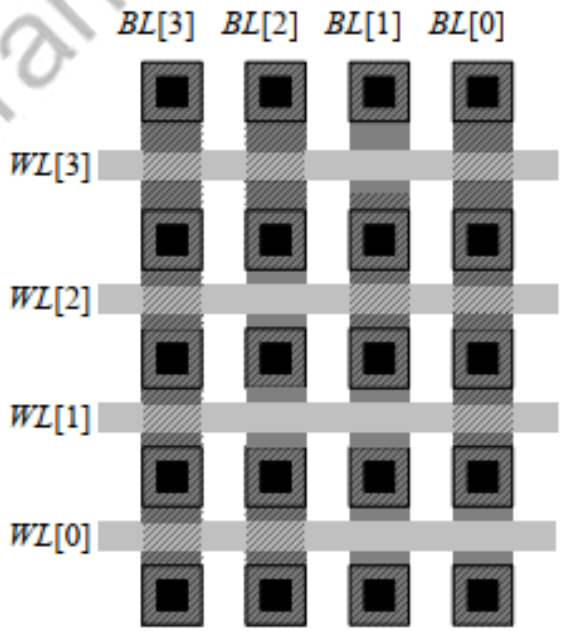


A_1	A_2	R_1	R_2	R_3	R_4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

MOS NAND ROM



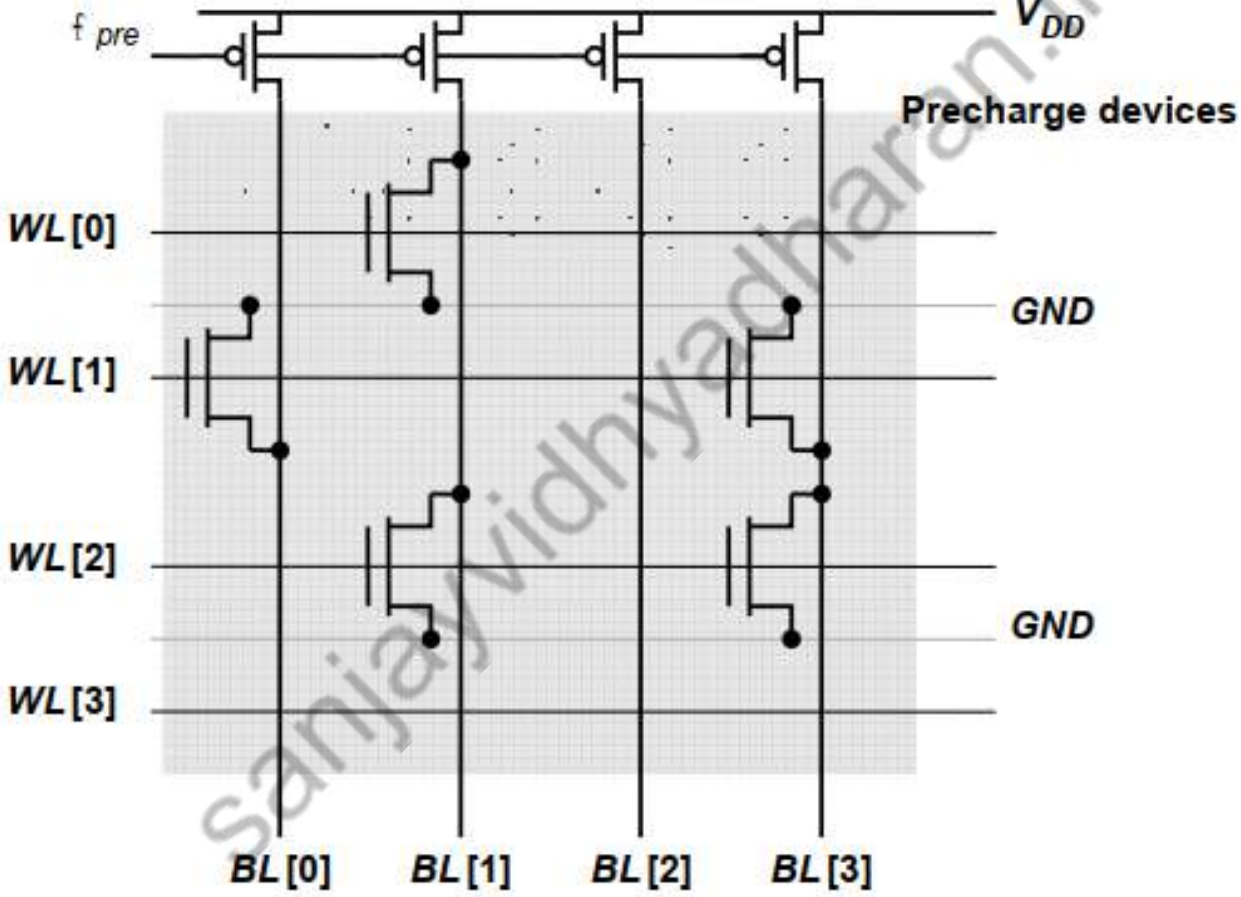
(a)



(b)

A_1	A_2	R_1	R_2	R_3	R_4
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Pre-charged MOS NOR ROM



Row Decoders

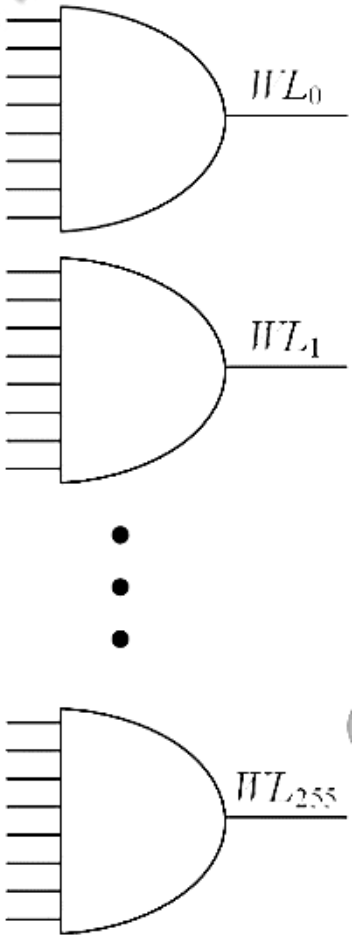
Example 8 bit Decoder

$$WL_0 = \overline{A}_7\overline{A}_6\overline{A}_5\overline{A}_4\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$$

$$WL_{255} = A_7A_6A_5A_4A_3A_2A_1A_0$$

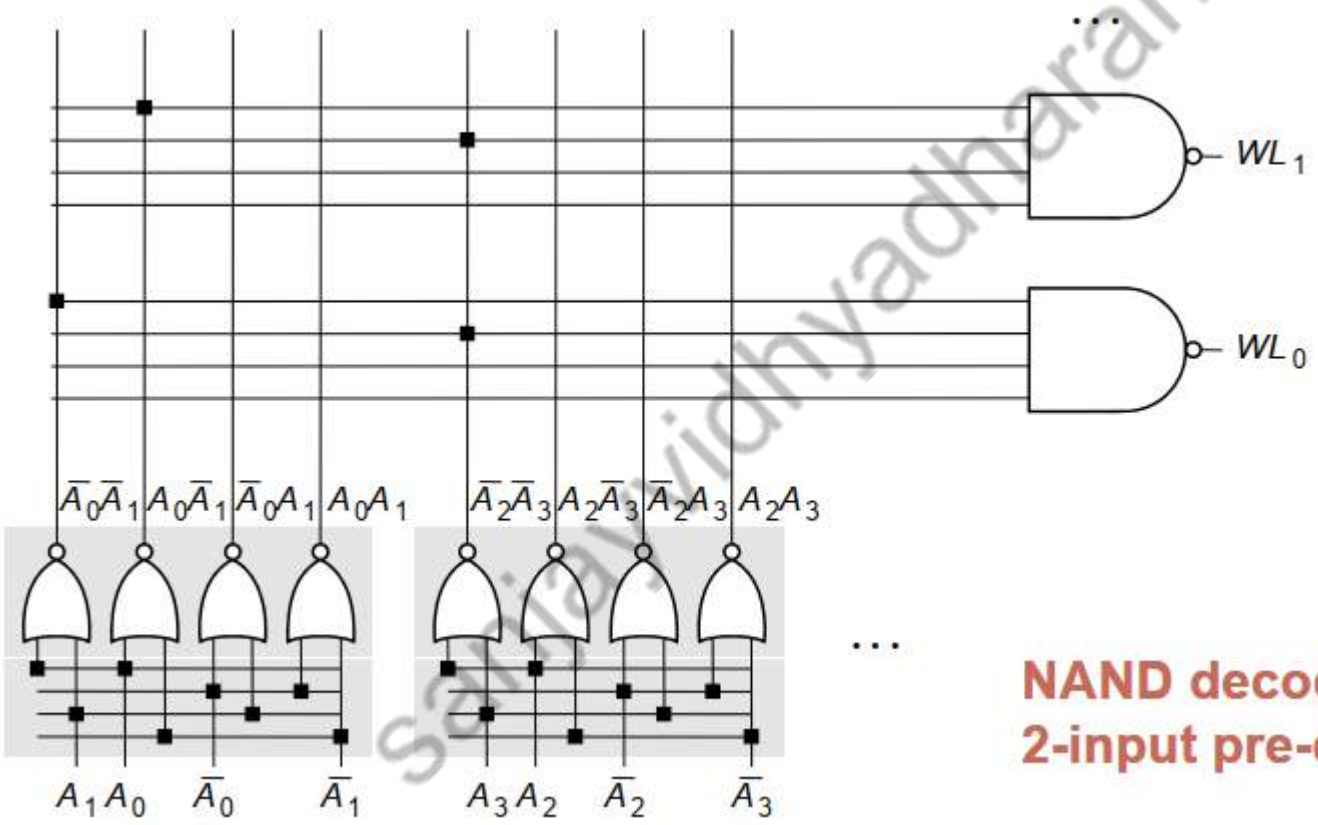
1. Implementation 8 Inverters + 256 NAND

2. Implementation 8 Inverters + 256 NOR



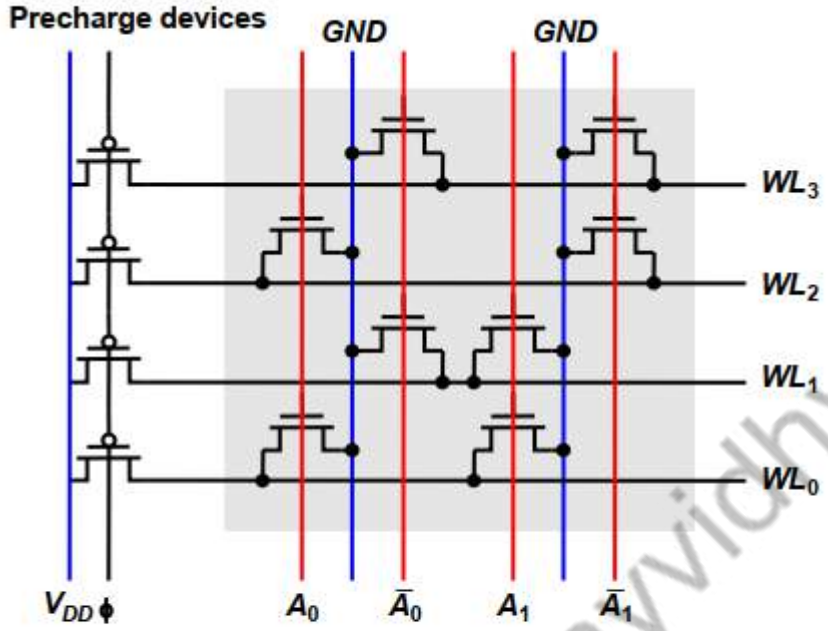
Row Decoders

Multi-stage implementation improves performance

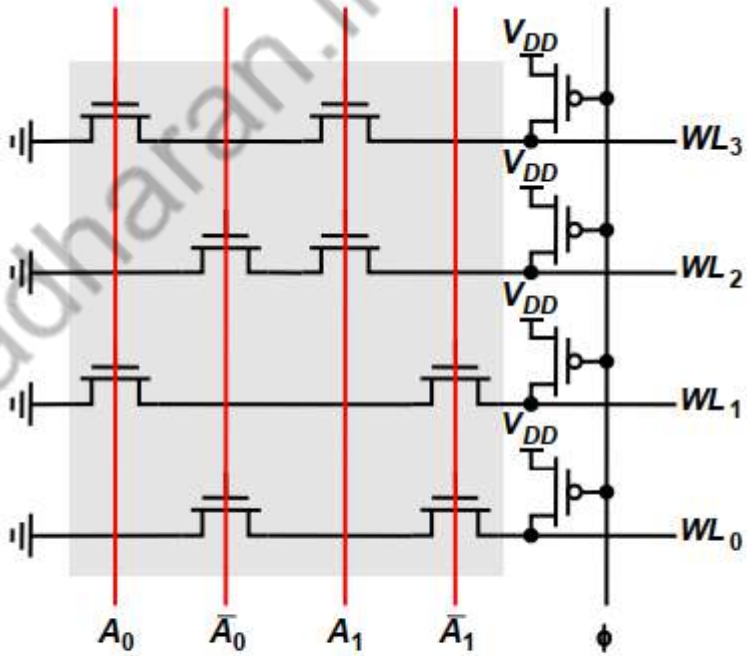


NAND decoder using 2-input pre-decoders

Row Decoders

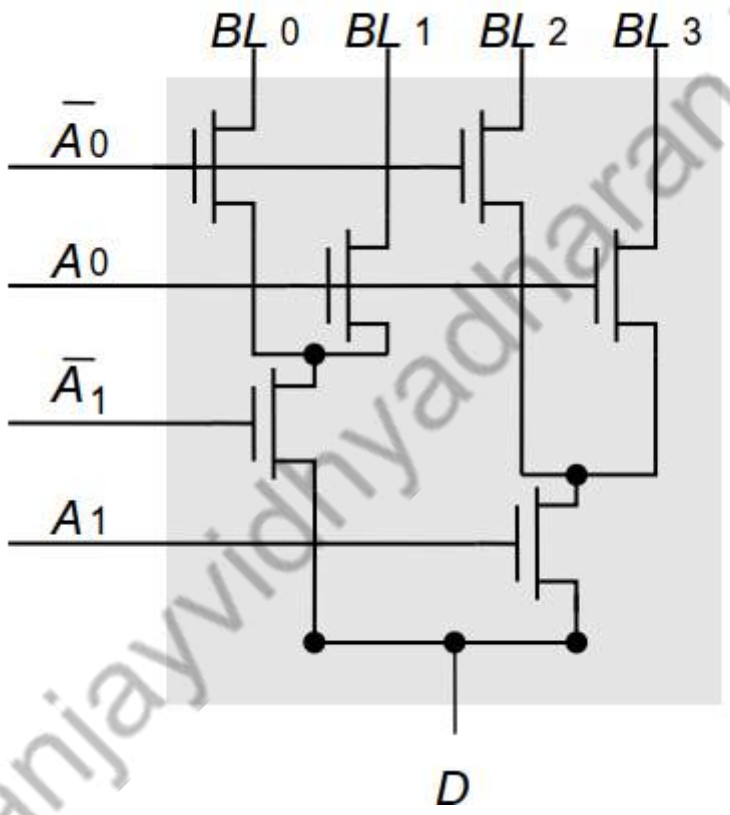


2-input NOR decoder



2-input NAND decoder

4-to-1 tree based column decoder



Number of devices drastically reduced. Delay increases quadratically with # of sections; prohibitive for large decoders

Solution : Buffers

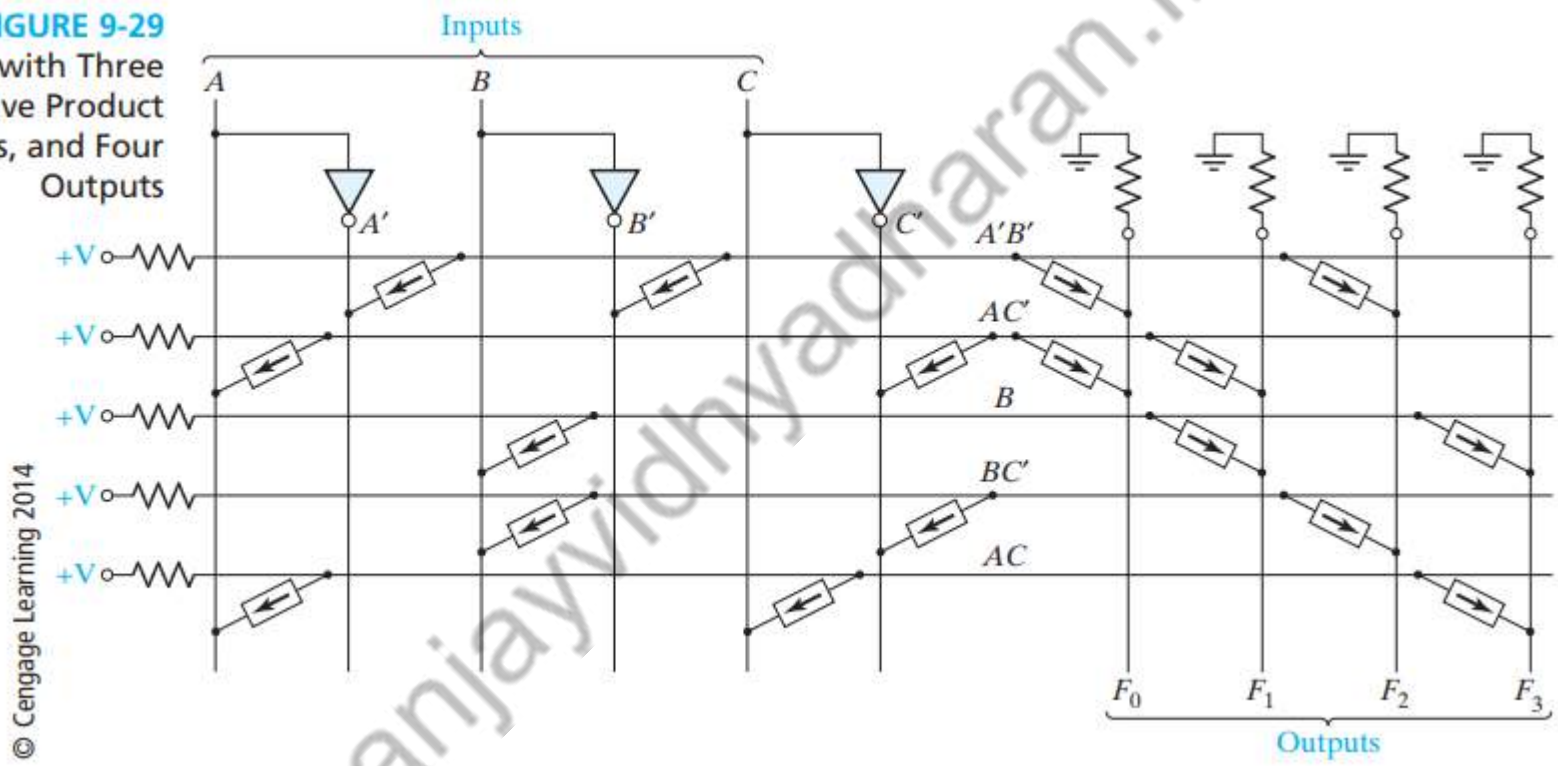
Programable Logic Array (PLA)

Programmable device capable of implementing functions expressed in SOP.

- Consists of input buffers and inverters followed by:
- Programmable AND plane, followed by
- Programmable OR plane.
- Can implement m logic functions of n variables. Limit is the number of product terms that can be generated inside of the device.

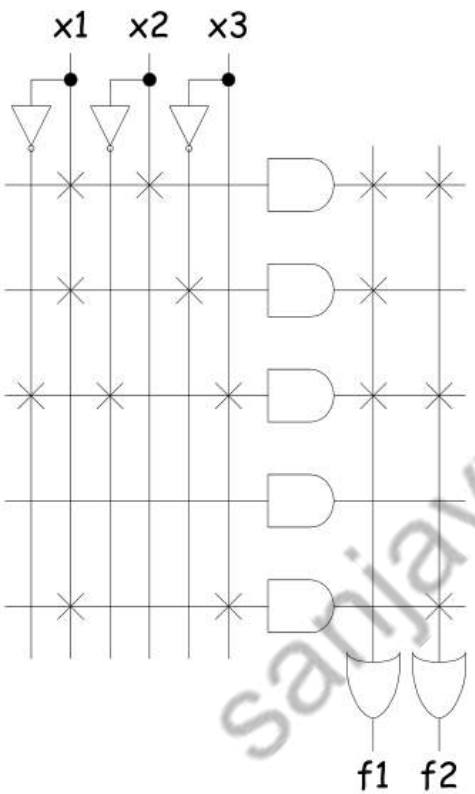
Programable Logic Array (PLA)

FIGURE 9-29
PLA with Three Inputs, Five Product Terms, and Four Outputs



Programable Logic Array (PLA)

- Example implementing 2 logic functions of 3 inputs using a **3-5-2 PLA**.



$$f_1 = x_1x_2 + x_1\overline{x}_3 + \overline{x}_1\overline{x}_2x_3$$

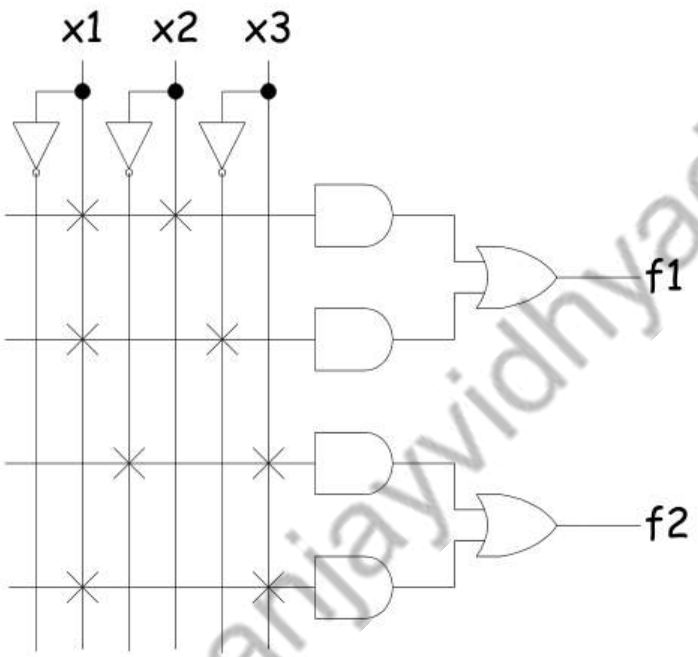
$$f_2 = x_1x_2 + \overline{x}_1\overline{x}_2x_3 + x_1x_3$$

Programable Array Logic Array (PAL)

- Similar to a PLA, but only has a programmable AND plane.
  - The OR plane is fixed.
- Not as flexible as a PLA since only certain AND gates feed each OR gate, but has fewer things that need programming.

Programable Array Logic Array (PAL)

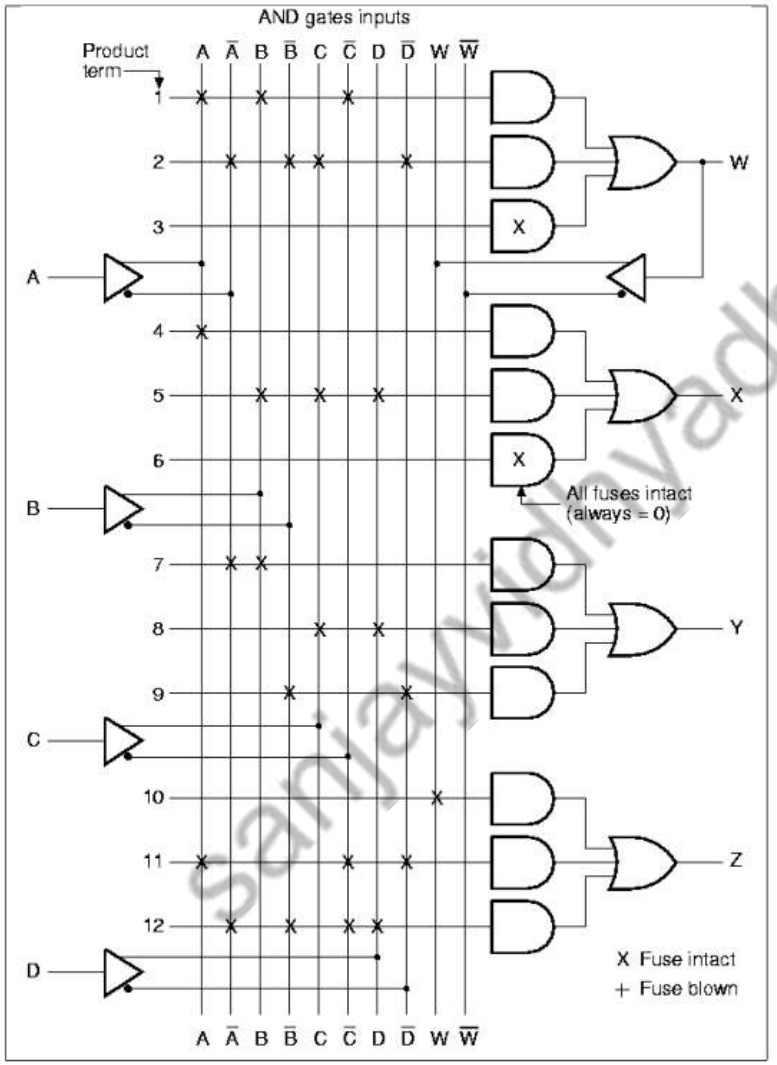
- Example of a PAL:



$$f_1 = x_1x_2 + x_1\overline{x}_3$$

$$f_2 = \overline{x}_2x_3 + x_1x_3$$

Programable Array Logic Array (PAL)



Programable Array Logic Array (PAL)

Example

$W(A, B, C, D) = \sum (2, 12, 13)$

$X(A, B, C, D) = \sum (7,8,9,10,11,12,13,14,15)$

$Y(A, B, C, D) = \sum (1,2,8,12,13)$

$W(A, B, C, D) = ABC' + A'B'CD'$

$X(A, B, C, D) = A + BCD$

$Y(A, B, C, D) = ABC' + A'B'CD' + AC'D' + A'B'C'D$

$= W + AC'D' + A'B'C'D$

Programable Array Logic Array (PAL)

Example

W(A, B, C, D) = ABC’ + A’B’CD’

X(A, B, C, D) = A + BCD

Y(A, B, C, D) = W + AC’D’ + A’B’C’D

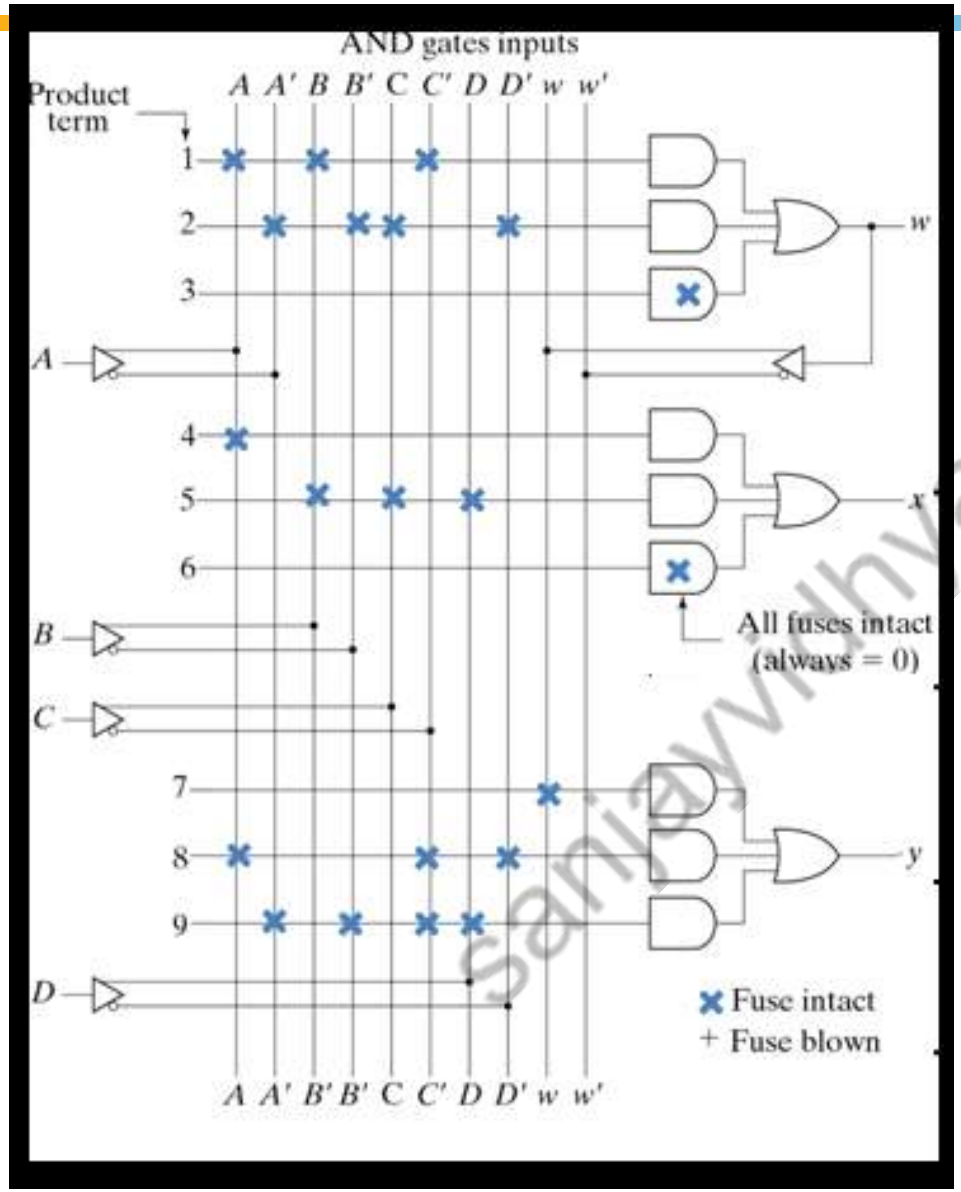
Product	AND Inputs					Outputs
	A	B	C	D	w	
1	1	1	0	--	--	**W = ABC’ + A’B’CD’**
2	0	0	1	0	--	
3	--	--	--	--	--	
4	1	--	--	--	--	**X = A + BCD**
5	--	1	1	1	--	
6	--	--	--	--	--	
7	--	--	--	--	1	**Y = W + AC’D’ + A’B’C’D**
8	1	--	0	0	--	
9	0	0	0	1	--	

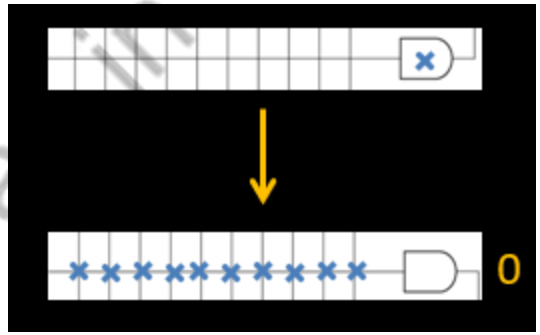
Section 1

Section 2

Section 3

Programable Array Logic Array (PAL)





Example

W(A, B, C, D) = ABC’ + A’B’CD’

X(A, B, C, D) = A + BCD

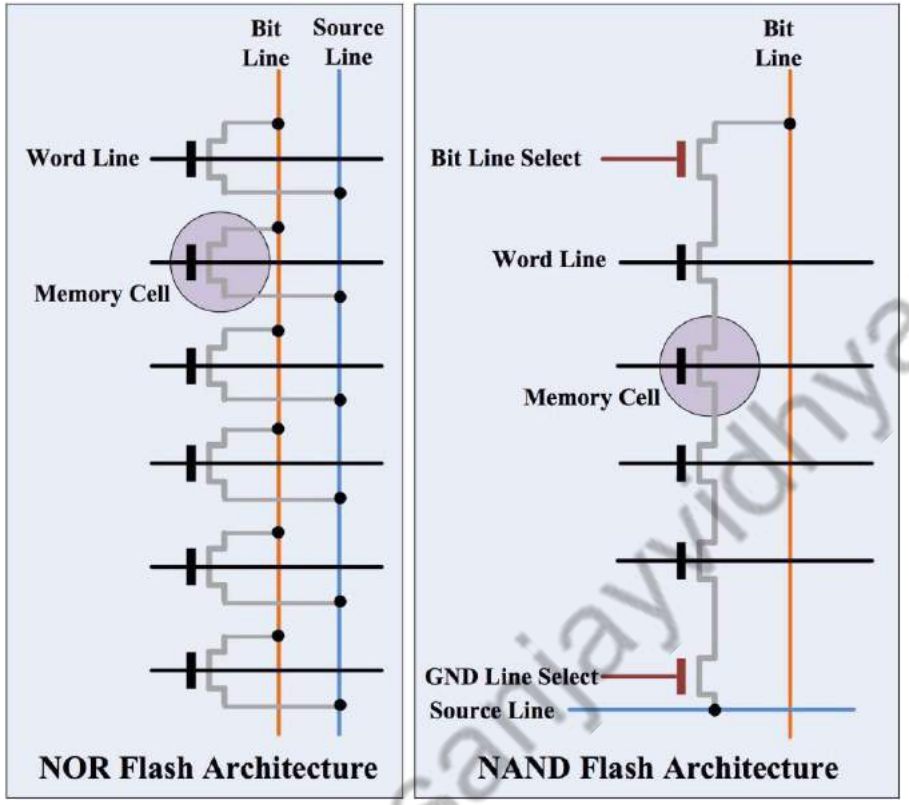
Y(A, B, C, D) = W + AC’D’ + A’B’C’D

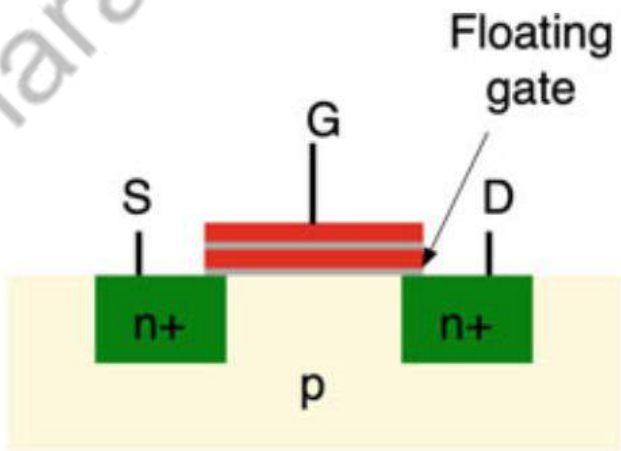
Flash Storage

- Most prominent solid state storage technology
  - No other technology is available at scale
- NAND- and NOR- flash types available
  - NOR-flash can be byte-addressed, expensive
  - NAND-flash is page addressed, cheap
  - Except in very special circumstances, all flash-storage we see are NAND-flash

Flash Storage

Flash memories store information in memory cells made from floating gate transistors.

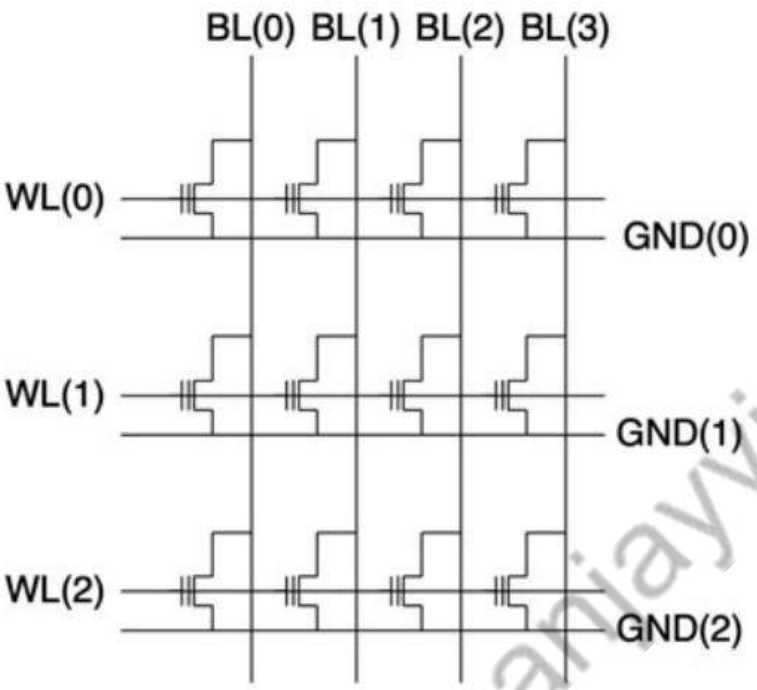


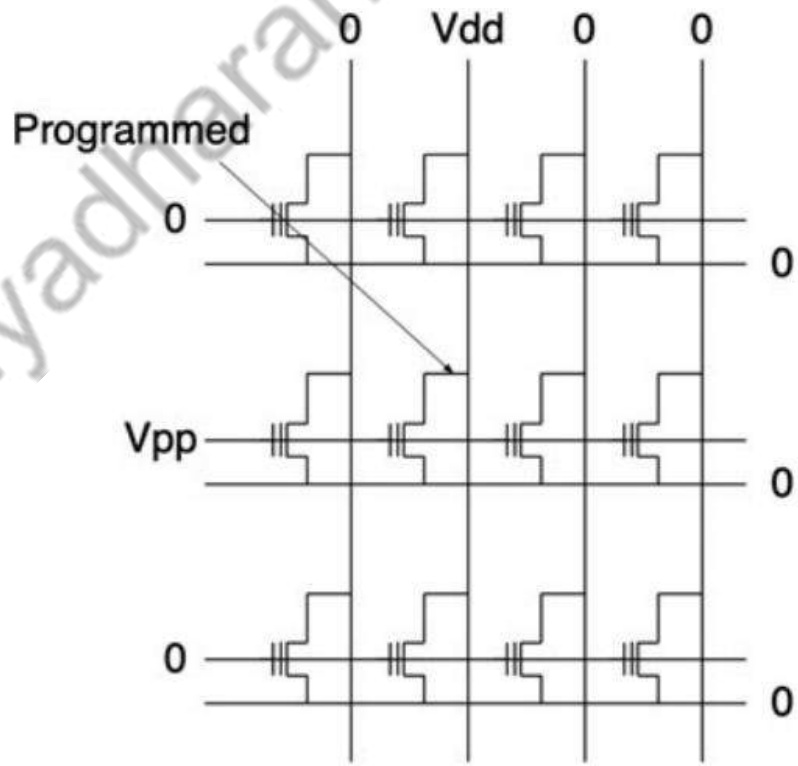


NOR flash is faster to read than NAND flash, but it's also more expensive.
NAND has a higher memory capacity than NOR.

Flash Storage

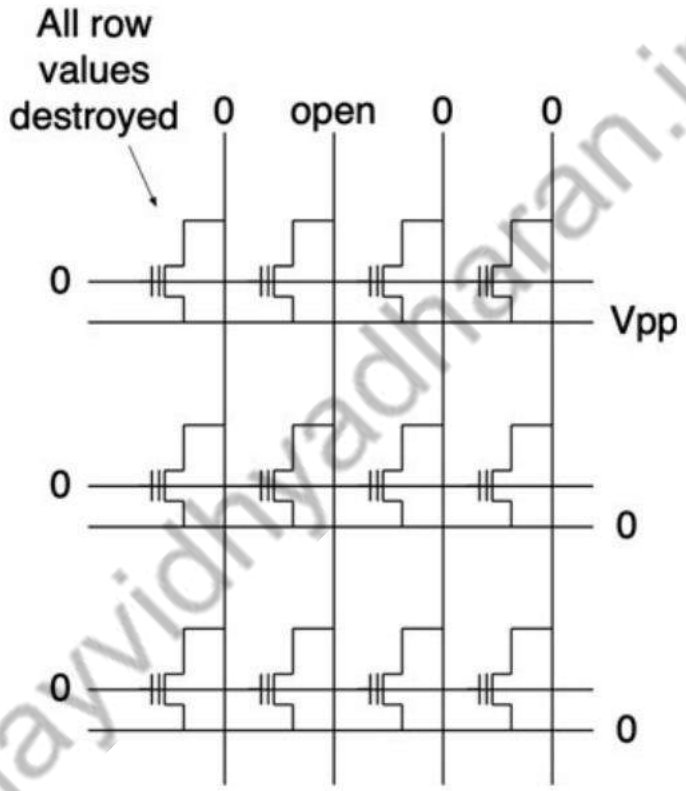
NOR flash





Flash Storage

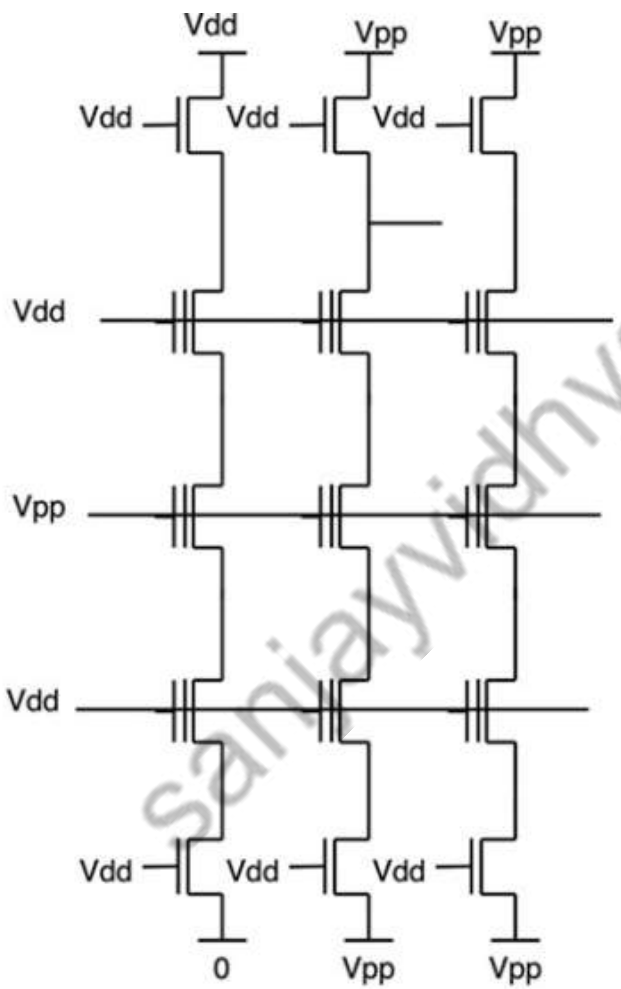
NOR flash

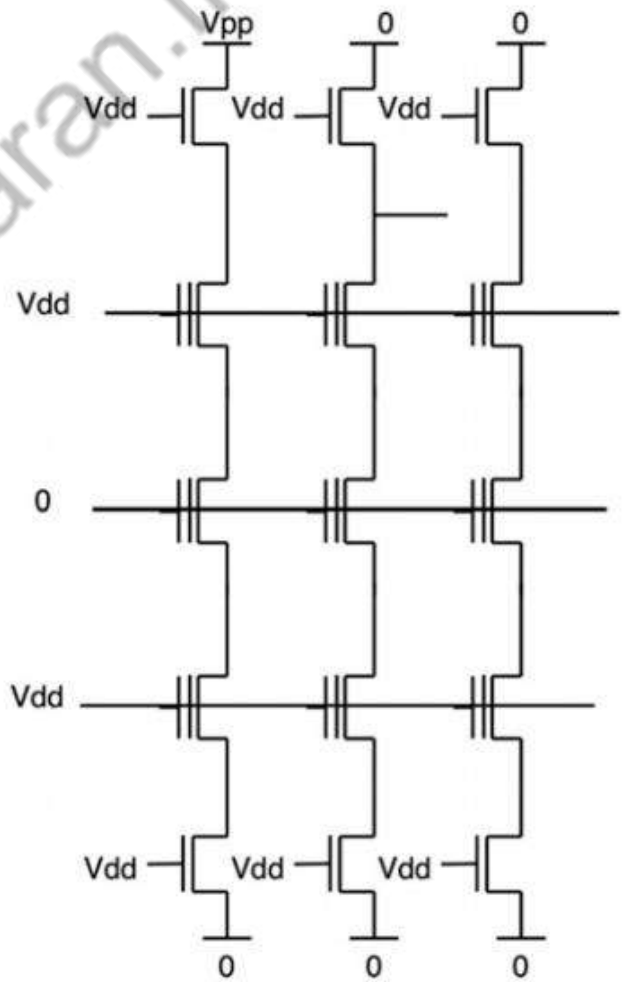


NOR FLASH memories are very fast to program and read. Erasure through tunneling is much slower. However, this kind of array suffers from low density due to the same reason that impacts NOR ROM density the need for multiple grounds.

Flash Storage

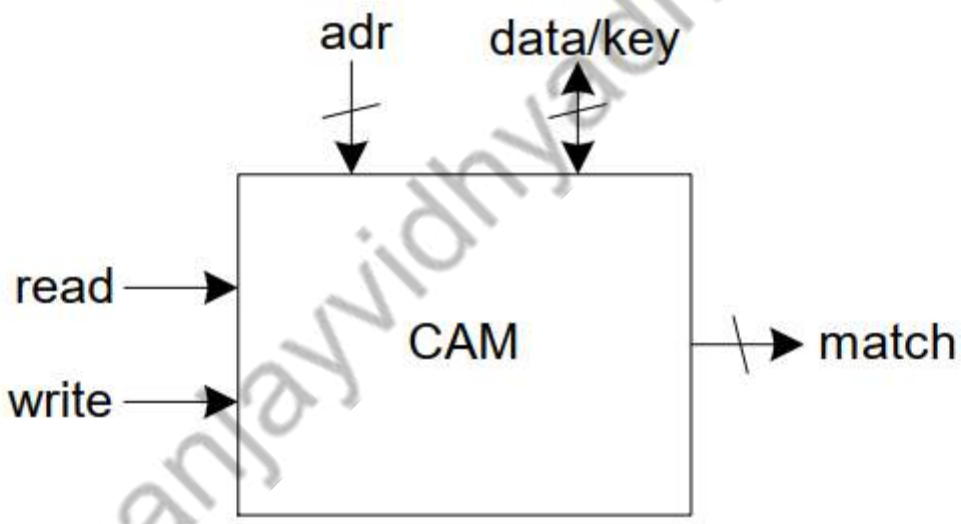
NAND flash





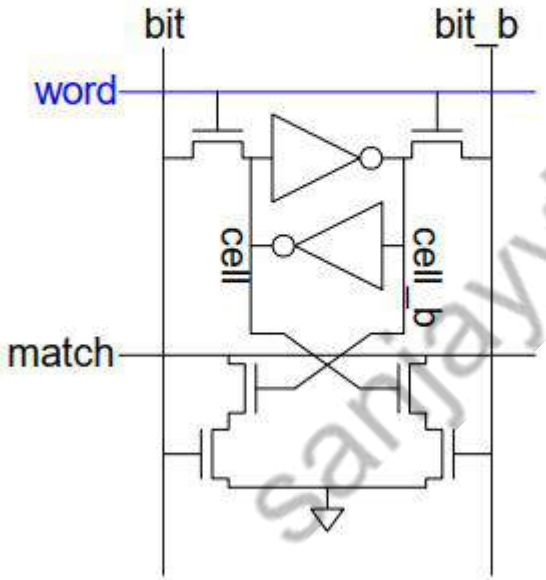
Content Addressable Memory (CAM)

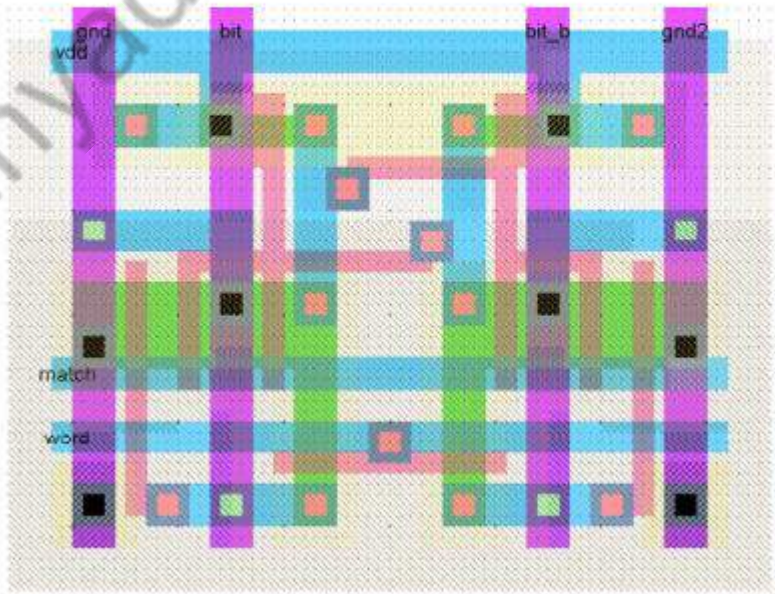
- Extension of ordinary memory (e.g. SRAM)
  - Read and write memory as usual
  - Also *match* to see which words contain a *key*



10T CAM Cell

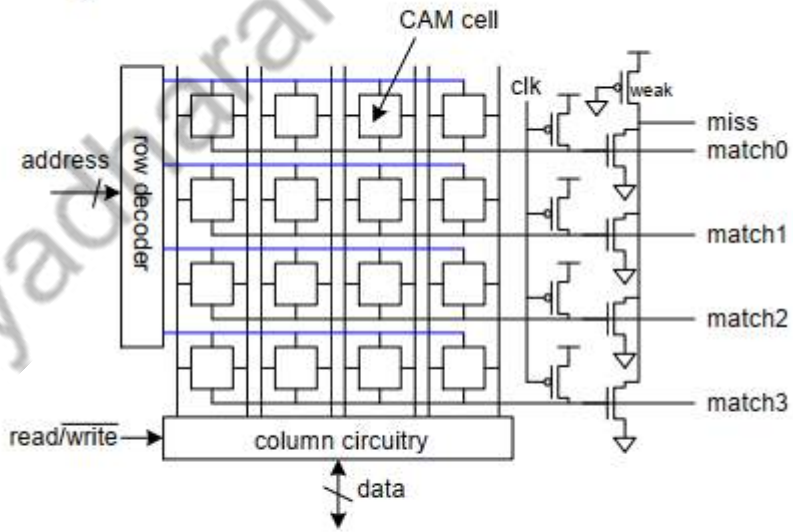
- Add four match transistors to 6T SRAM
  - 56 x 43 λ unit cell





CAMs

- 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



Thank you