



**EECS 151/251A**

**Spring 2023**

**Digital Design and Integrated Circuits**

Instructor:

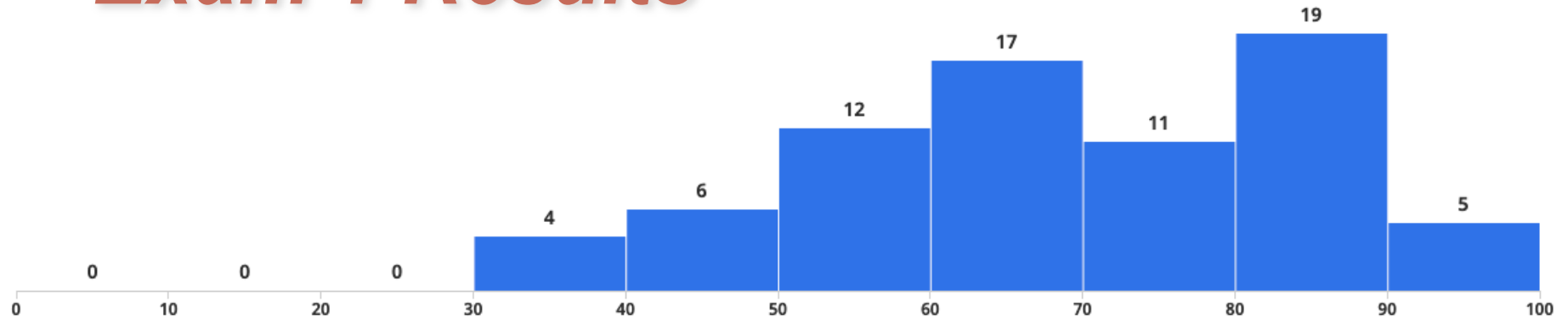
John Wawrzynek

**Lecture 17:  
Memory Circuits  
and Blocks**

# *Announcements*

- ❑ Project spec & HW7 posted this week
- ❑

# Exam 1 Results



Minimum

**35.0**

Median

**68.0**

Maximum

**94.0**

Mean

**68.47**

Std Dev [?](#)

**15.46**

- ❑ Grades and Solutions posted today
- ❑ Regrades open for a week.



# Outline

- ❑ Memory Circuits
  - ❑ SRAM
  - ❑ DRAM
- ❑ Memory Blocks
  - ❑ Multi-ported RAM
  - ❑ Combining Memory blocks
  - ❑ FIFOs
  - ❑ FPGA memory blocks
  - ❑ Caches
  - ❑ Memory Blocks in the Project

# *First, Some Memory Classifications:*

- Hardwired (Read-only-memory- ROM)
- Programmable
  - *Volatile*
    - SRAM - uses positive feedback (and restoration) to hold state
    - DRAM - uses capacitive charge (only) to hold state
  - *Non-volatile*
    - Persistent state without power supplied
    - Ex: Flash Memory

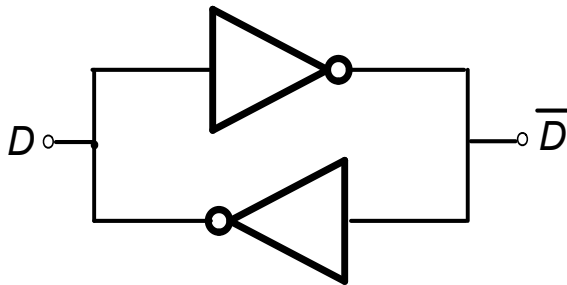


## Memory Circuits

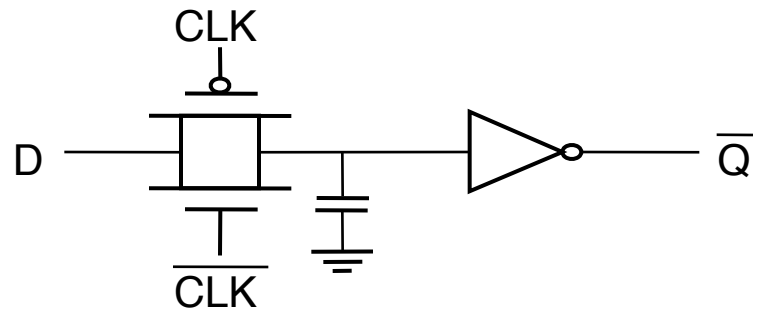
# Volatile Storage Mechanisms

These circuits represent the principles of storing a bit:

**Static - feedback**



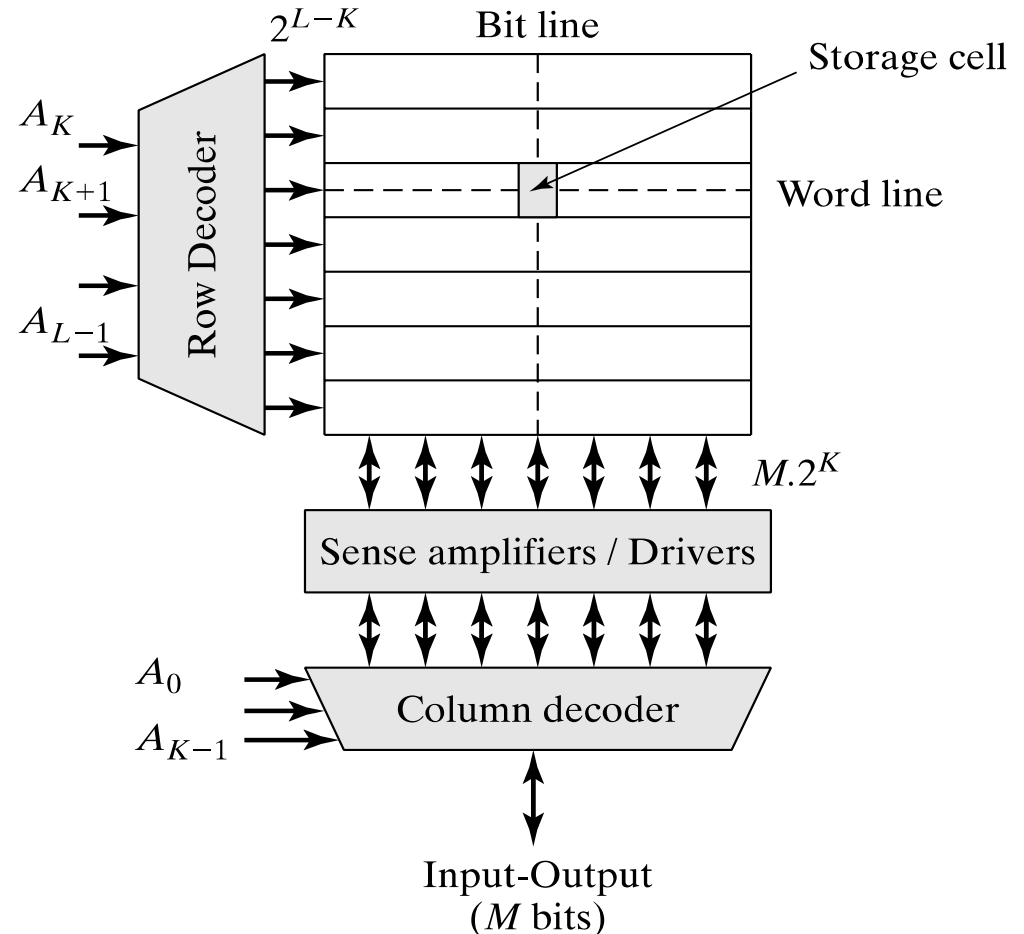
**Dynamic - charge**



Circuit details differ, depending on application.

# Generic Memory Block Architecture

- **Word lines** used to select a row for reading or writing
- **Bit lines** carry data to/from periphery
- **Core aspect ratio** keep close to 1 to help balance delay on word line versus bit line
- **Address bits** are divided between the two decoders
- **Row decoder** used to select word line
- **Column decoder** used to select one or more columns for input/output of data



*Storage cell could be either static or dynamic*

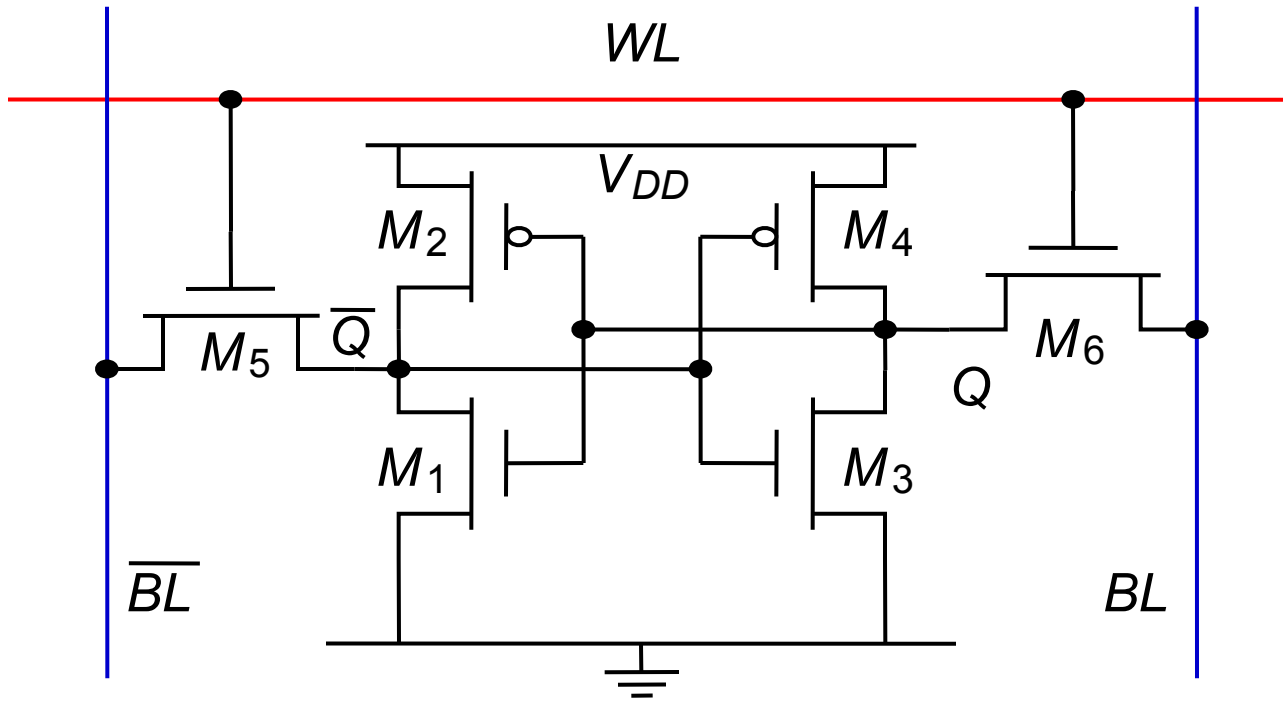
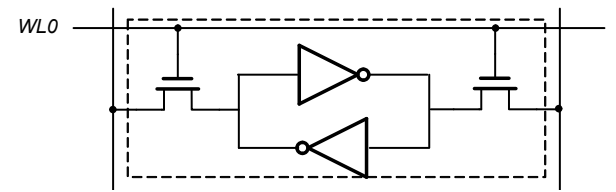




## Memory - SRAM

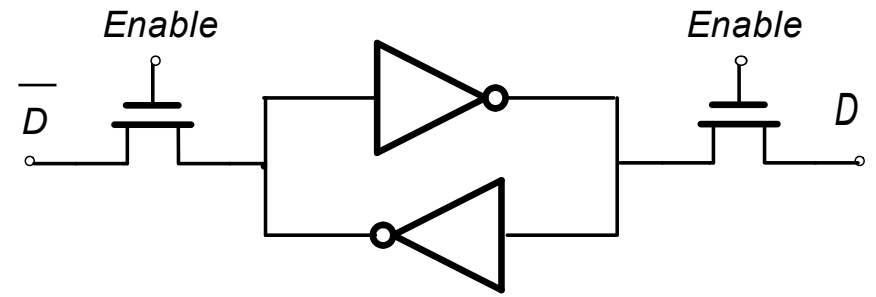
- ❑ Gets used for on-chip memories. Caches, large register files, input/output buffers, ...
- ❑ Compatible with logic processes.

# 6-Transistor CMOS SRAM Cell

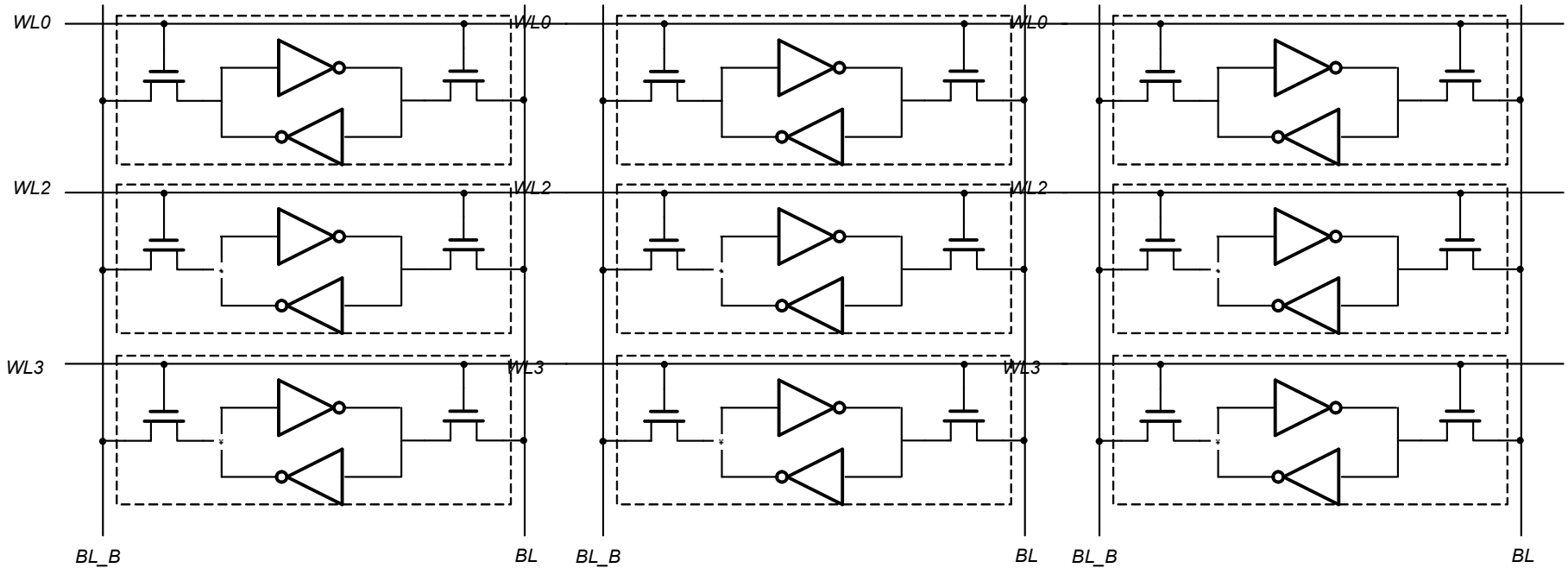


# Memory Cells

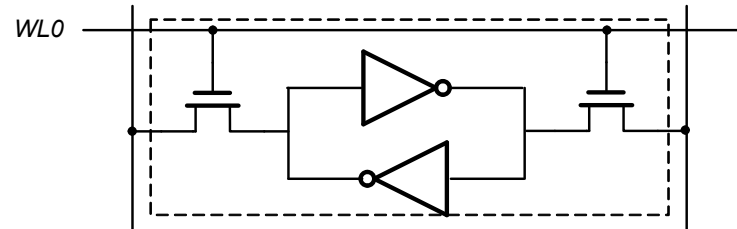
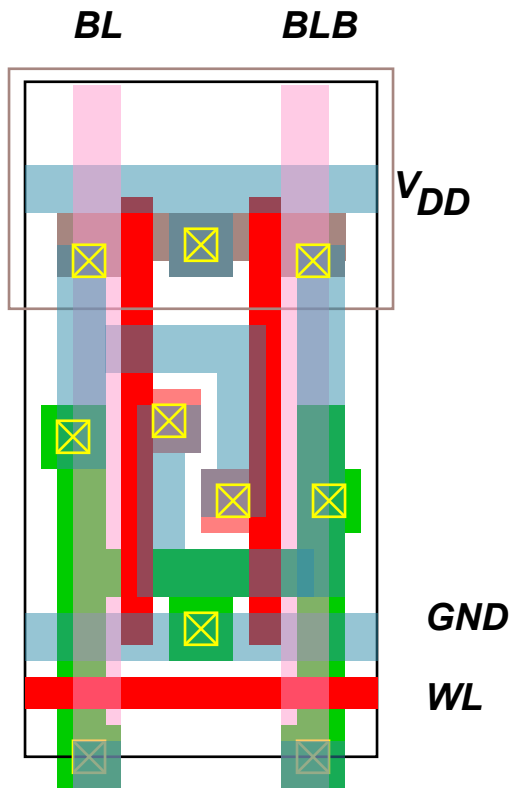
Complementary data values are written (read) from two sides



Cells stacked in 2D to form memory core.



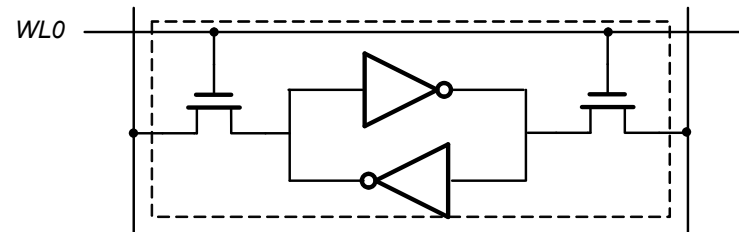
# 6T-SRAM — Older Layout Style



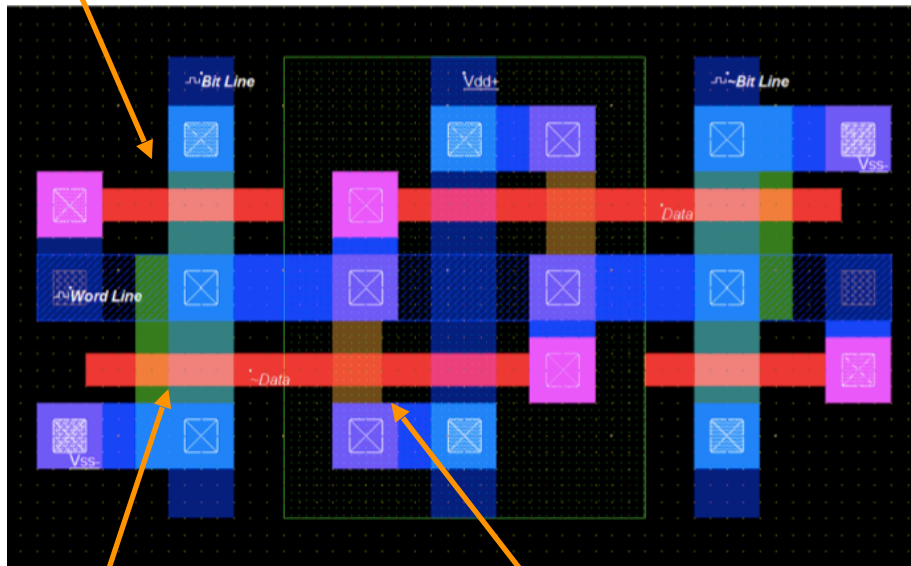
V<sub>DD</sub> and GND: in M1(blue)  
Bitlines: M2 (purple)  
Wordline: poly-silicon (red)

# Modern SRAM

□ ST/Philips/Motorola

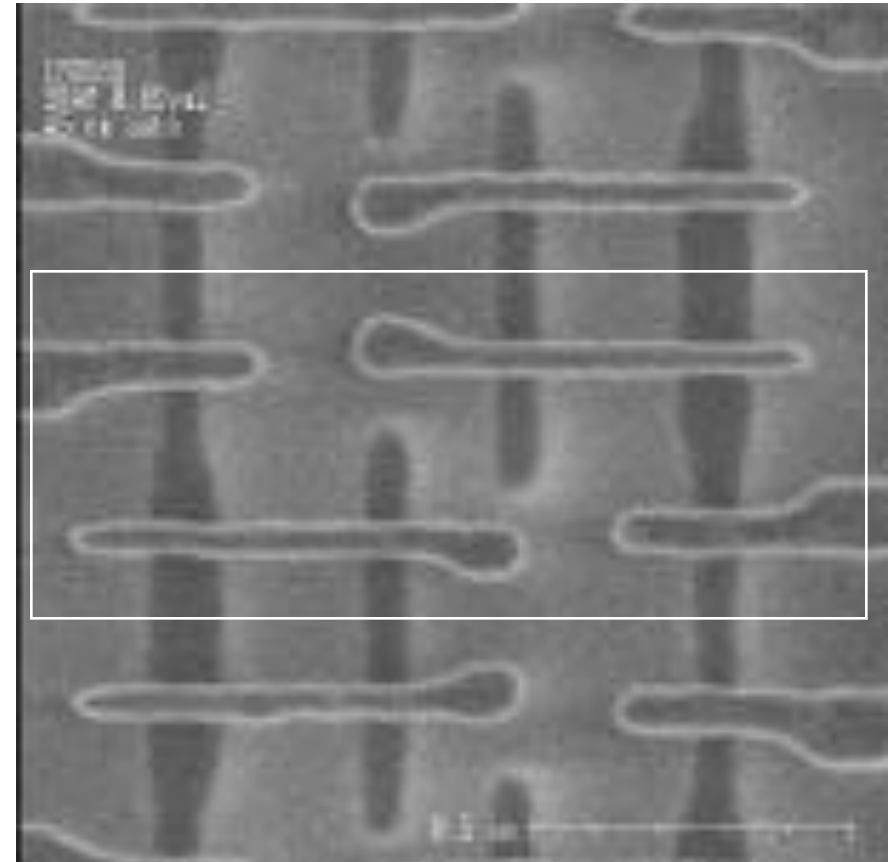


*Access Transistor*

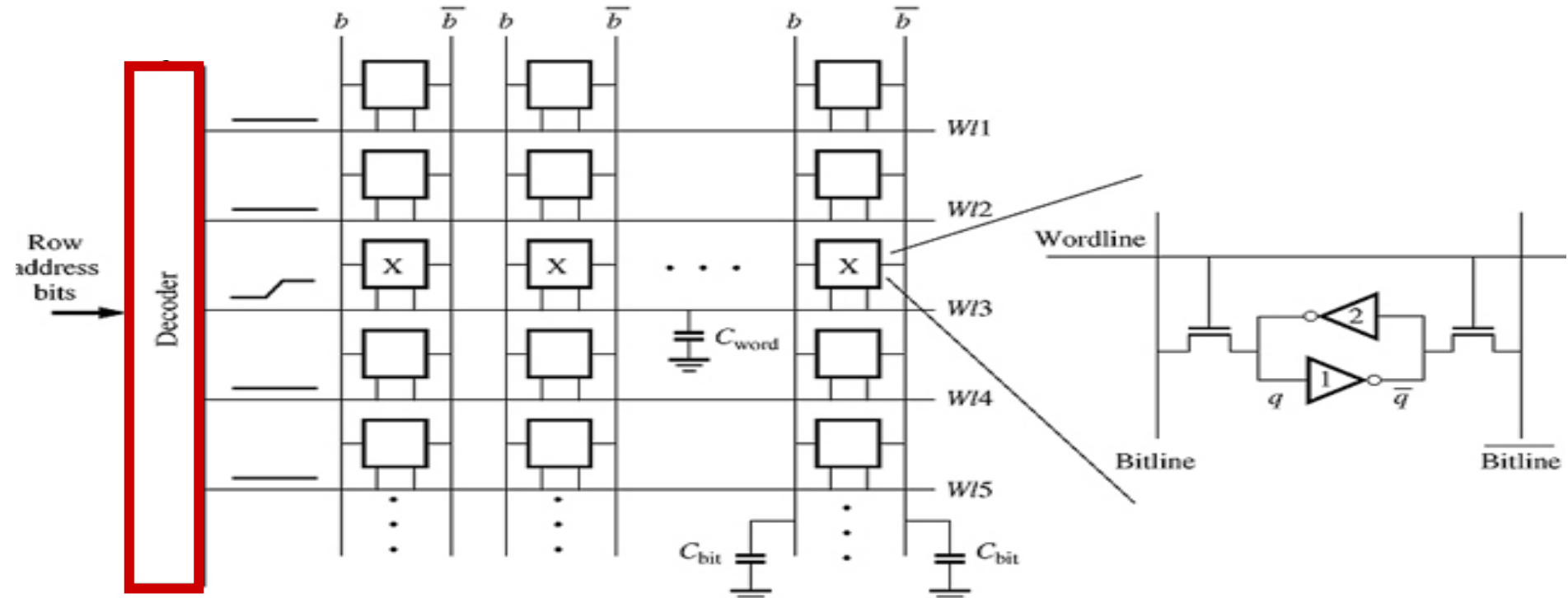


*Pull down*

*Pull up*

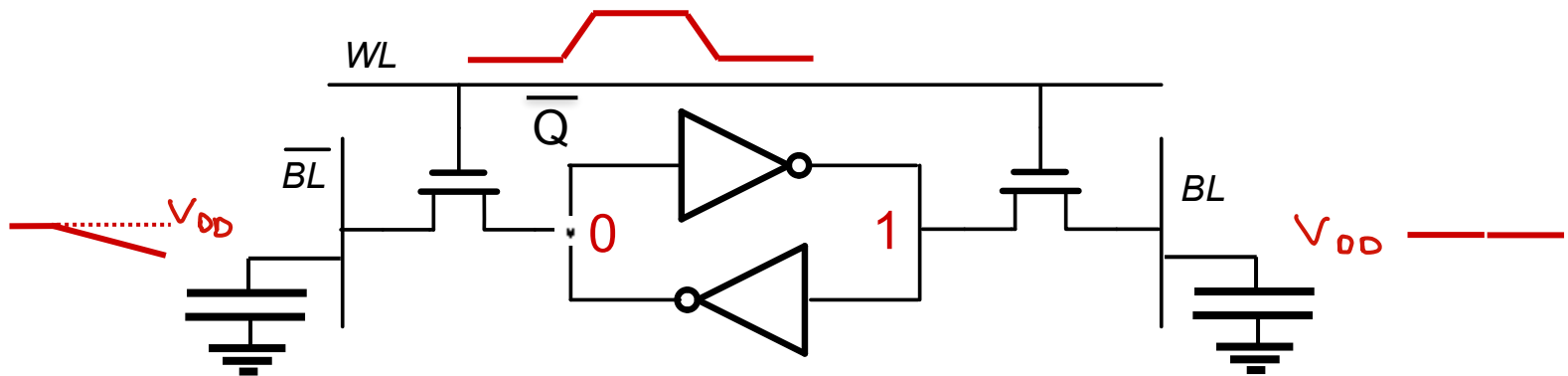


# SRAM read/write operations



# SRAM Operation - Read

1. Bit lines are “pre-charged” to VDD
2. Word line is driven high (pre-charger is turned off)
3. Cell pulls-down one bit line
4. Differential sensing circuit on periphery is activated to capture value on bit lines.

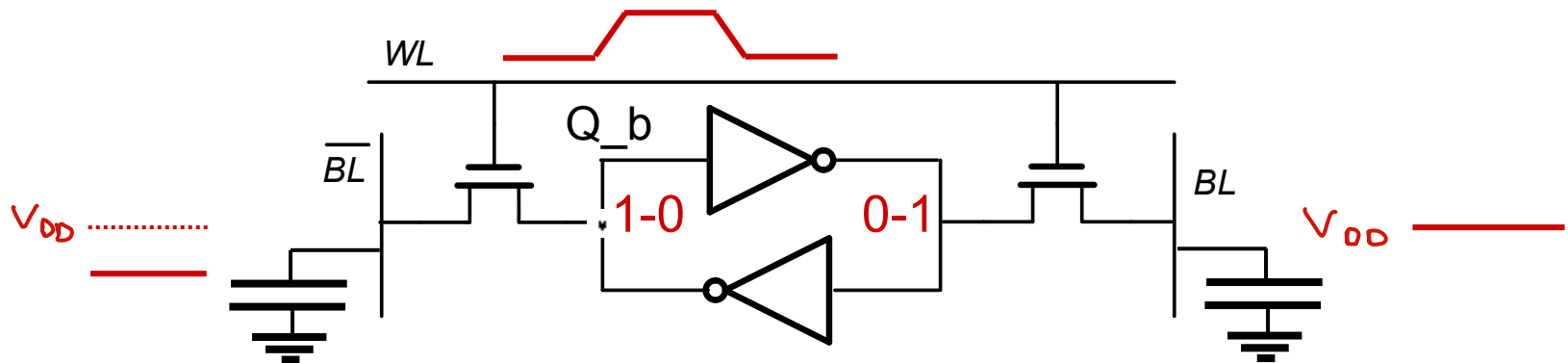


*During read  $\bar{Q}$  will get slightly pulled up when WL first goes high, but ...*

- *But by sizing the transistors correctly, reading the cell will not destroy the stored value*

# SRAM Operation - Write

1. Column driver circuit on periphery differentially drives the bit lines
2. Word line is driven high (column driver stays on)
3. One side of cell is driven low, flips the other side



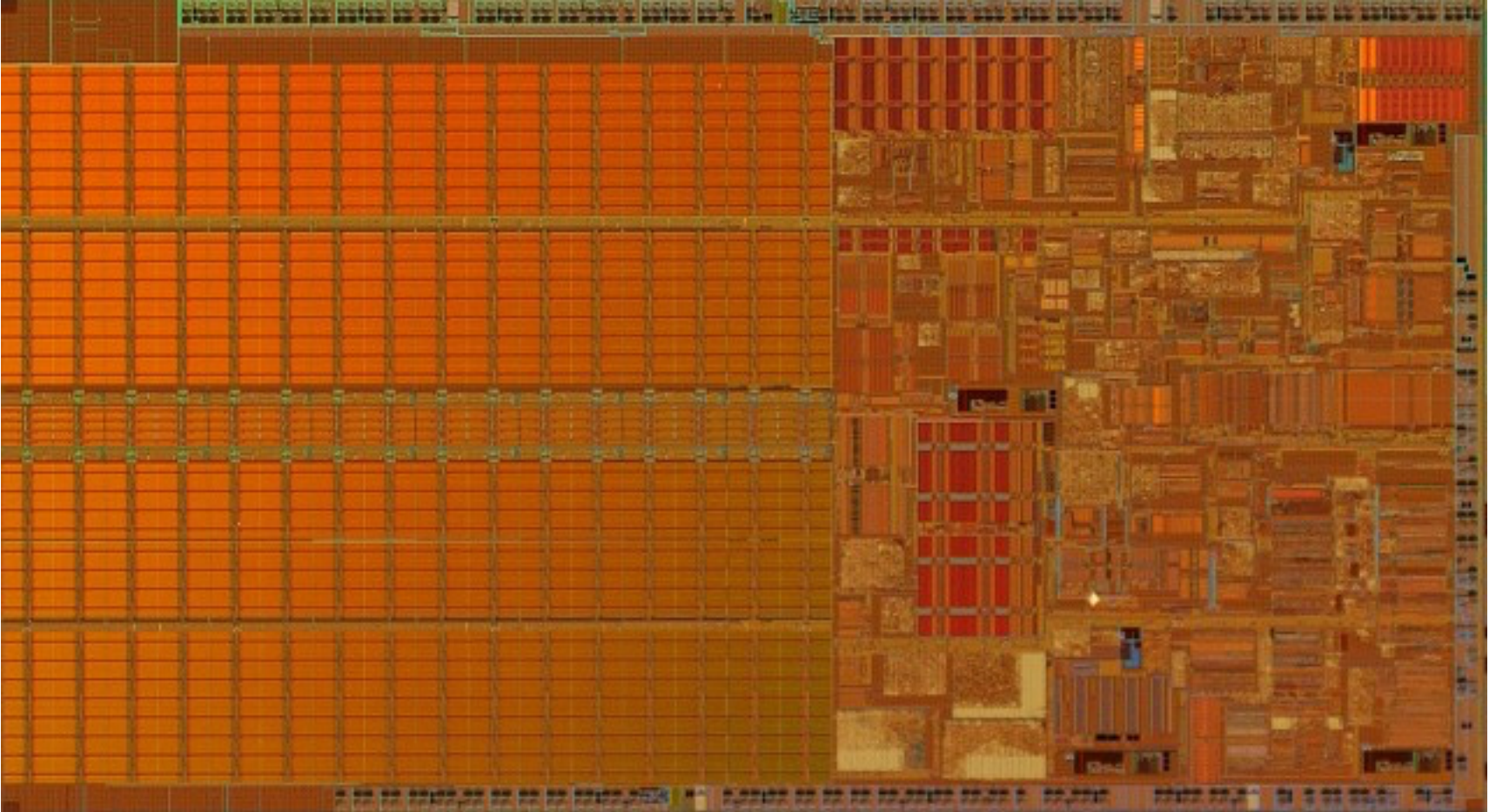
*For successful write the access transistor needs to overpower the cell pullup. The transistors are sized to allow this to happen.*





## Memory Periphery

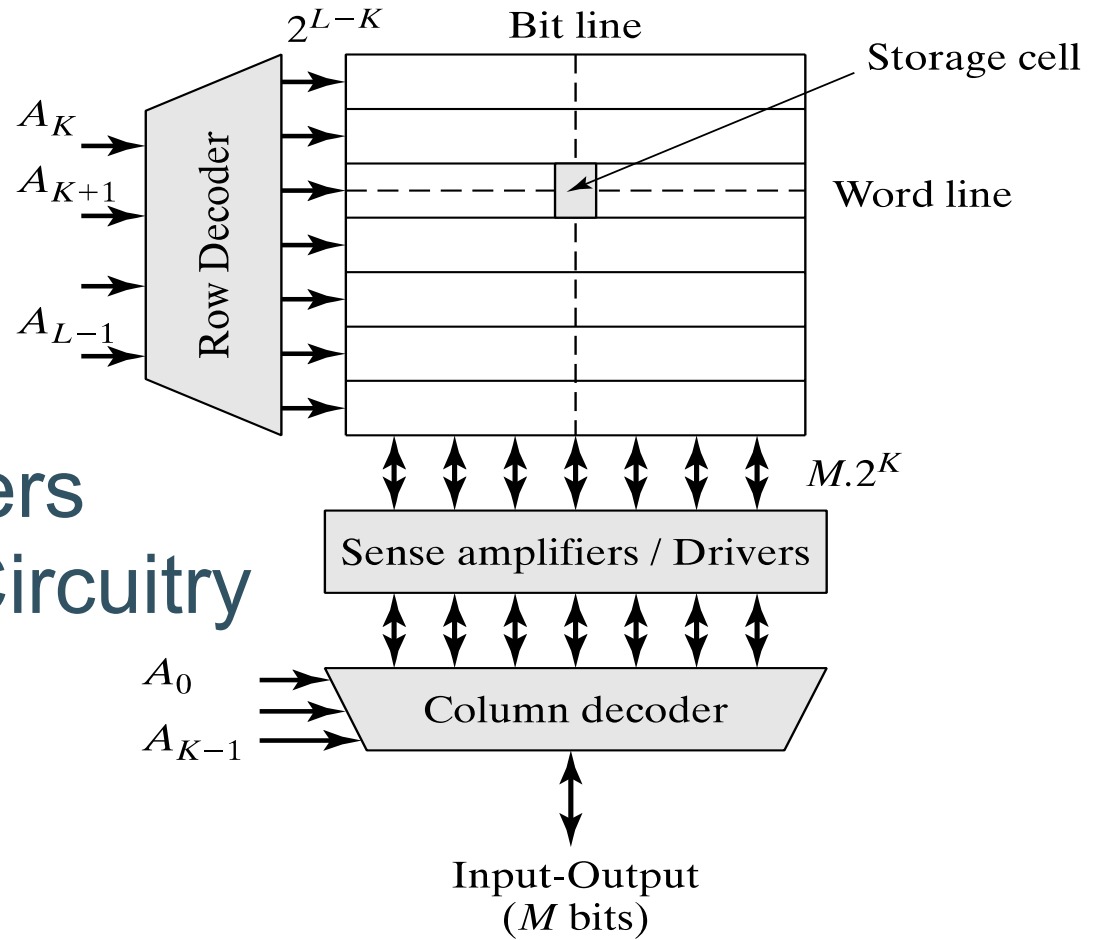
# *On-chip Memory*



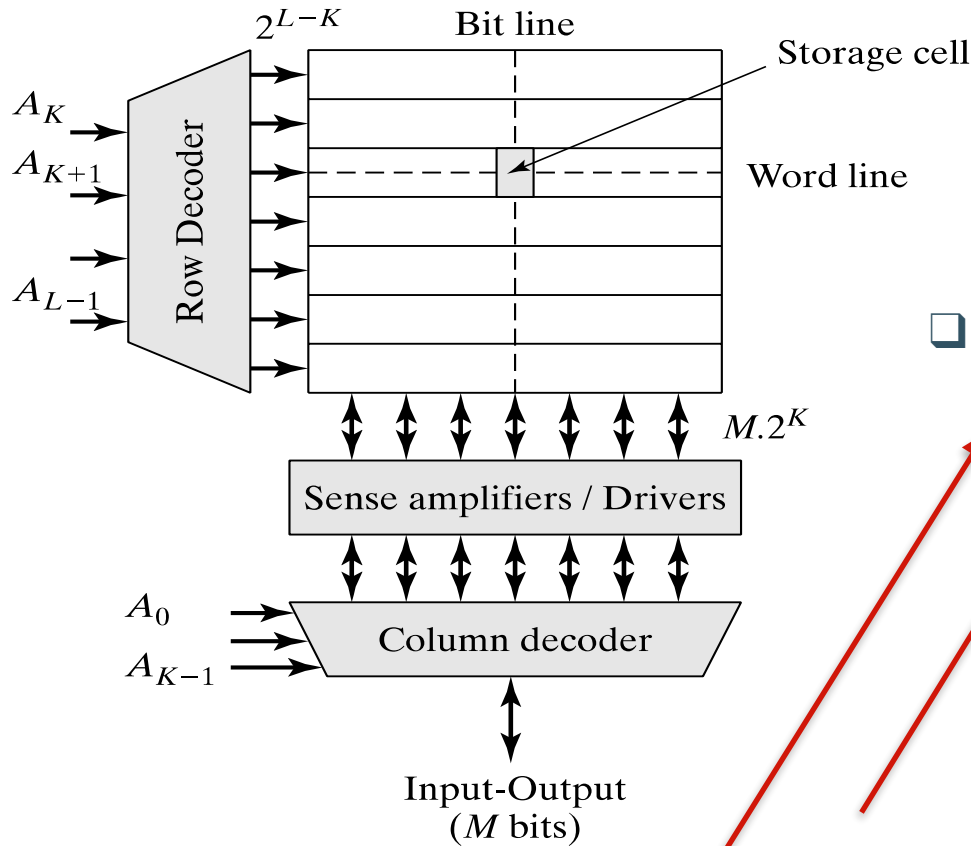
*ARM A5 Photo*

# Periphery

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



# Row Decoder



- $L$  total address bits
- $K$  for column decoding
- $L-K$  for row decoding
- Row decoder expands  $L-K$  address lines into  $2^{L-K}$  word lines
- $M$  bits per word

□ Example: decoder for  $8K \times 8$  memory block

- core arranged as  $256 \times 256$  cells
- Need 256 AND gates, each driving one word line

*each row has 32 8-bit words ( $8 \times 32 = 256$ )*

*$8K \times 8$  means  $8K$  words of 8-bits each*

*In this case:  $L=13$  total address bits ( $2^L=8K$ ),  $K=5$  ( $2^K=32$ ),  $L-K=8$  ( $2^{L-K}=256$ )*

# Row Decoders

## (N)AND Decoder

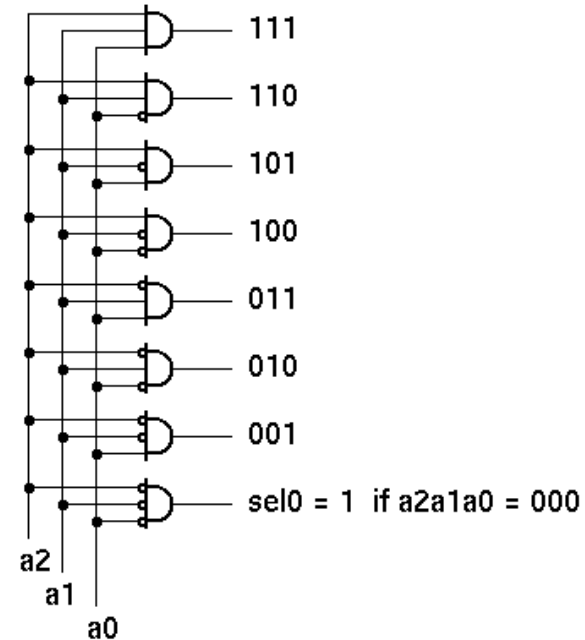
$$WL_0 = A_0A_1A_2A_3A_4A_5A_6A_7A_8A_9$$

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

## NOR Decoder

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

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



Collection of  $2^{L-K}$  logic gates, but need to be dense and fast.

Naive solution would require  $L-K$  input gates: *Too big to pitch match to storage cells and too slow.*

# Predecoders

$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$\overline{a_2}$	$\overline{a_1}$	$\overline{a_0}$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$\overline{a_2}$	$\overline{a_1}$	$a_0$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$\overline{a_2}$	$a_1$	$\overline{a_0}$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$\overline{a_2}$	$a_1$	$a_0$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$a_2$	$\overline{a_1}$	$\overline{a_0}$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$a_2$	$\overline{a_1}$	$a_0$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$a_2$	$a_1$	$\overline{a_0}$
$\overline{a_5}$	$\overline{a_4}$	$\overline{a_3}$	$a_2$	$a_1$	$a_0$

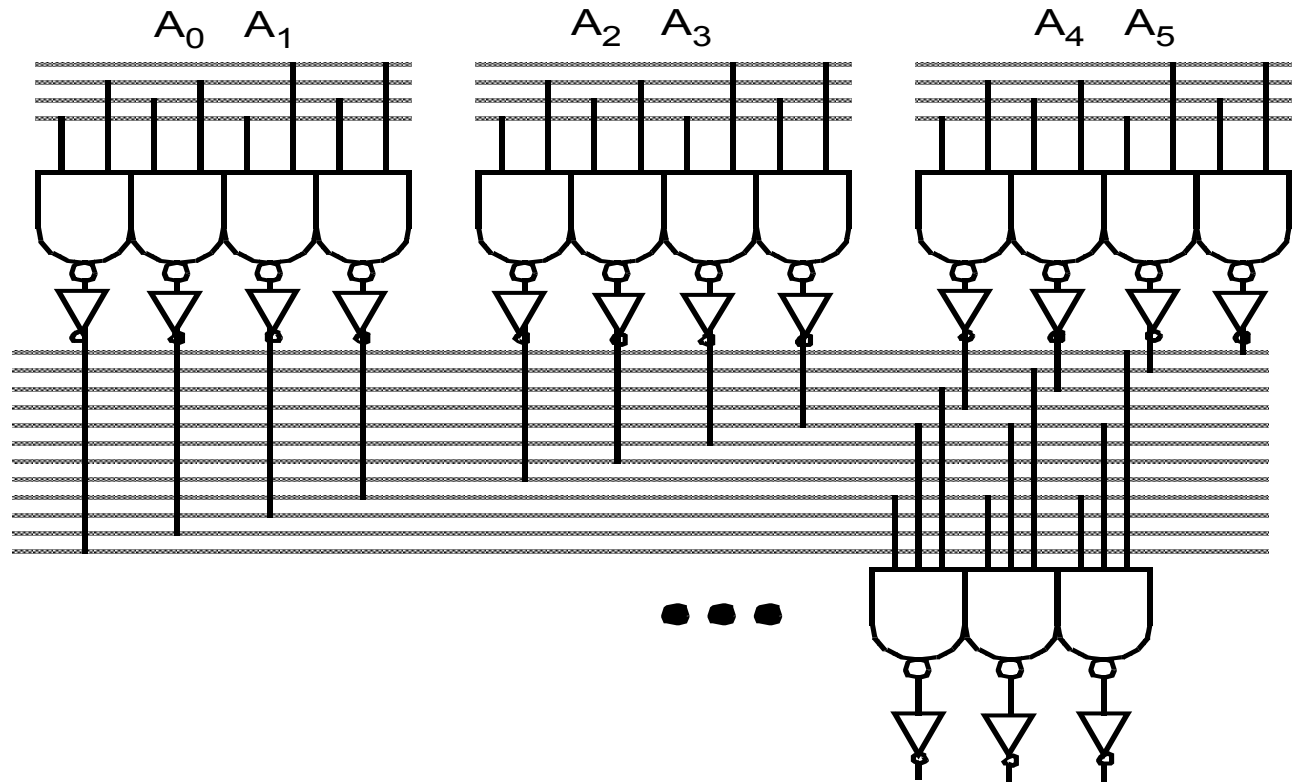
•  
•  
•

$a_5$	$a_4$	$a_3$	$a_2$	$\overline{a_1}$	$\overline{a_0}$
$a_5$	$a_4$	$a_3$	$a_2$	$\overline{a_1}$	$a_0$
$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$\overline{a_0}$
$a_5$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$

- Use a single gate for each of the shared terms
  - E.g., from  $a_1, \overline{a_1}, a_0, \overline{a_0}$  generate four signals:
    - $\overline{a_1} \overline{a_0}, \overline{a_1} a_0, a_1 \overline{a_0}, a_1 a_0$
- Do same for  $a_5, a_4, a_3, a_2$
- In other words, we decode smaller groups of address bits first
  - And using the “predecoded” outputs to do the rest of the decoding

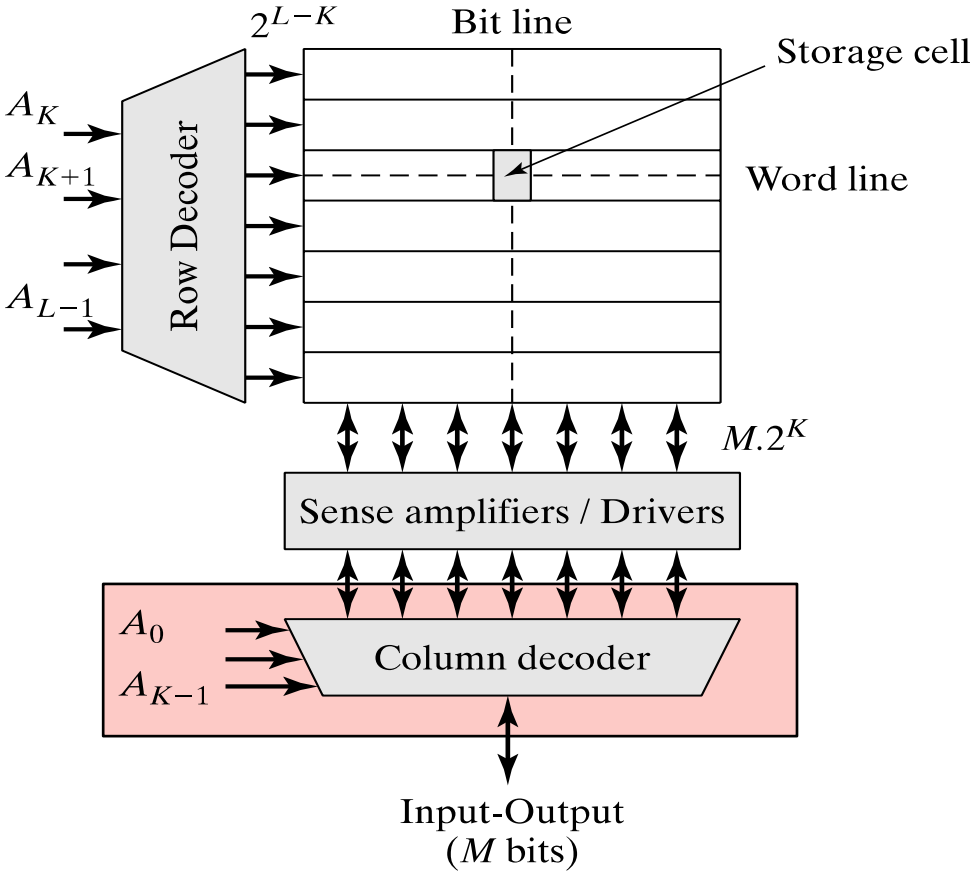
# Predecoder and Decoder

*Predecoders*



*Final Decoder*

# Column “Decoder”



- ❑ Is basically a multiplexer
- ❑ Each row contains  $2^K$  words each  $M$  bits wide.
- ❑ Bit of each of the  $2^K$  are interleaved
  - ❑ ex:  $K=2, M=8$

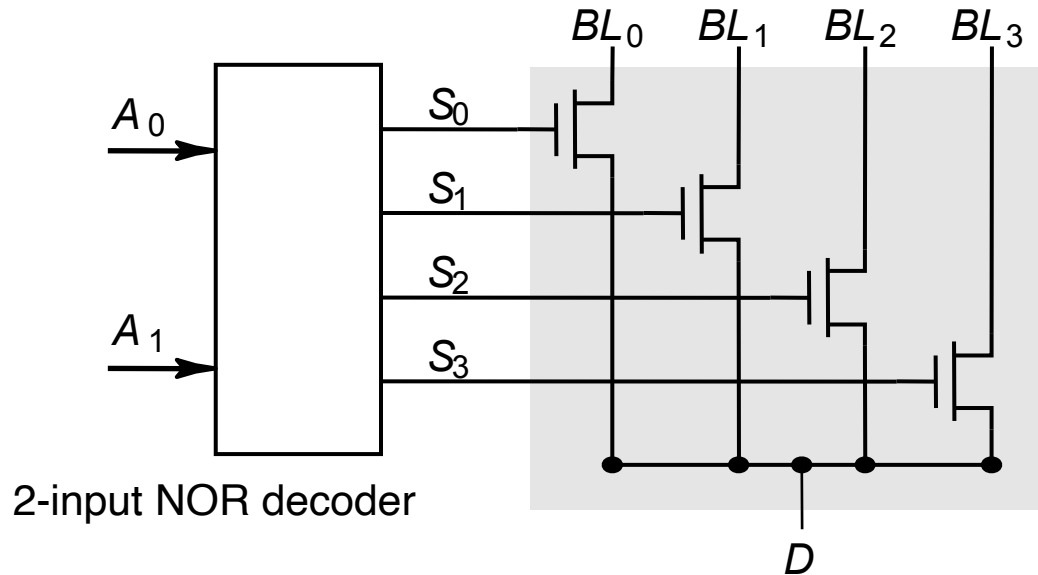


$d_7c_7b_7a_7d_6c_6b_6a_6d_5c_5b_5a_5d_4c_4b_4a_4d_3c_3b_3a_3d_2c_2b_2a_2d_1c_1b_1a_1d_0c_0b_0a_0$

*4 interleaved words A, B, C, D*



# 4-input pass-transistor based Column Decoder (for read)



(actual circuit would use a “differential signaling”)

*decoder shared across all  $2^K \times M$  row bits*

Advantages: speed (Only one extra transistor in signal path, share sense amp)

# Sense Amplifiers Speed Reading

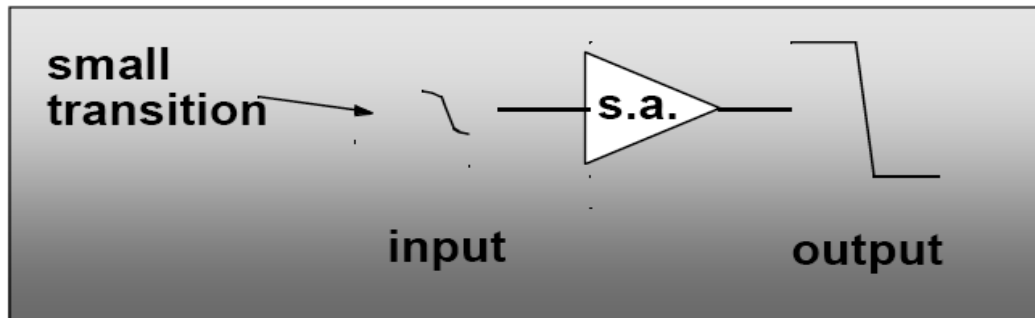
large Capacitance of bit lines

$$\tau_p \propto \frac{C \cdot \Delta V}{I_{av}}$$

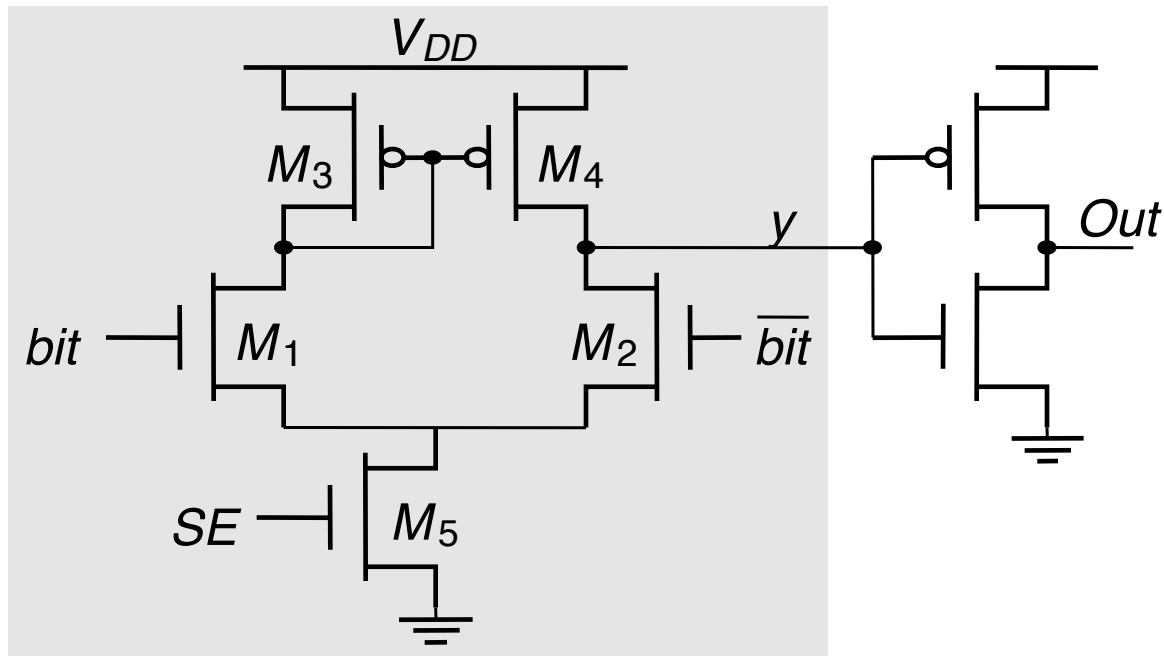
*make as small as possible*

small

Idea: Use “Sense Amplifier”

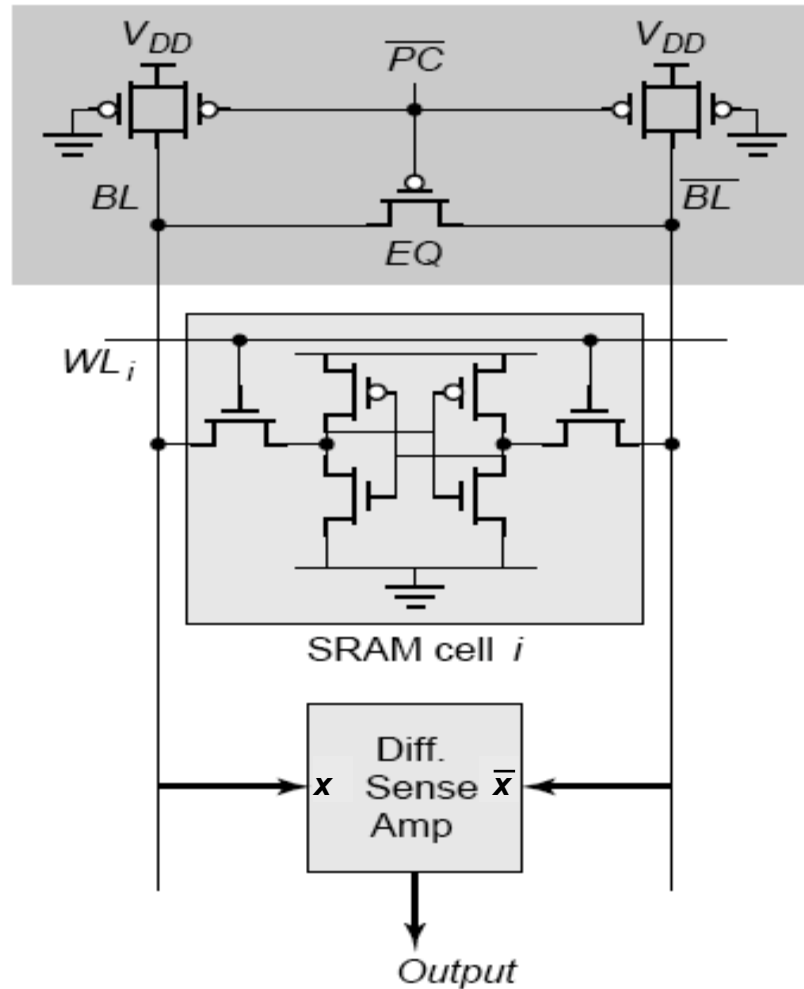


# Differential Sense Amplifier



Classic Differential Amp structure - basis of opAmp

# Differential Sensing – SRAM



(a) SRAM sensing scheme