EECS 151/251A
Spring 2019
Digital Design and Integrated Circuits

Instructor:
John Wawrzynek

Lecture 16

Memory Circuits

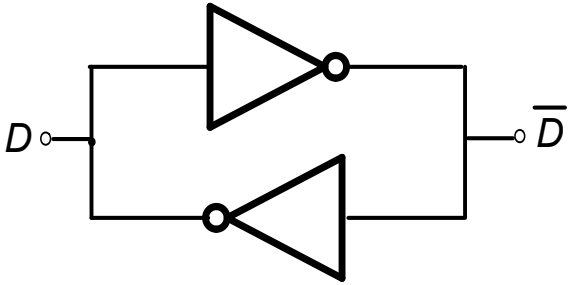
General Latch Design

Storage principles

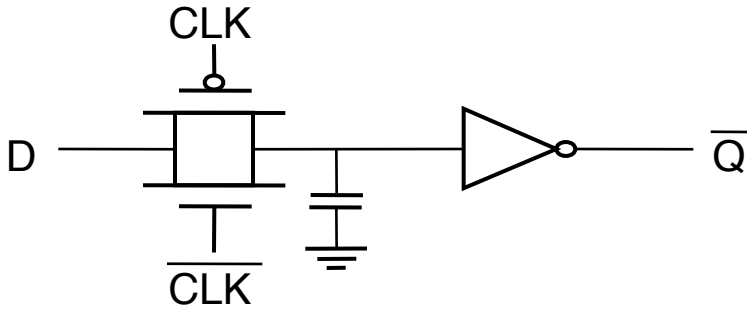
- Hardwired (Read-only-memory- ROM)
- Programmable
  - *Volatile*
    - Feedback to hold state while power is on
  - *Non-volatile*
    - Persistent state without power supplied

Volatile Storage Mechanisms

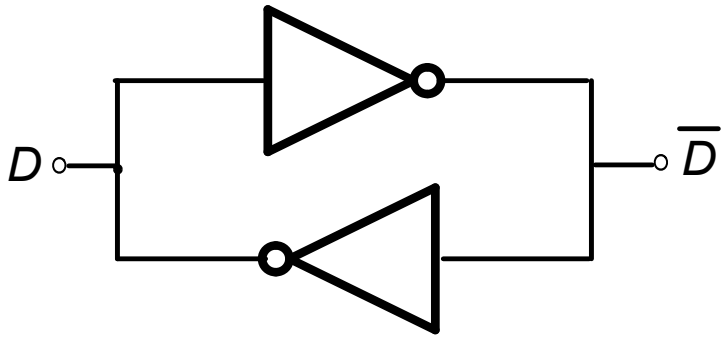
Static - feedback



Dynamic - charge

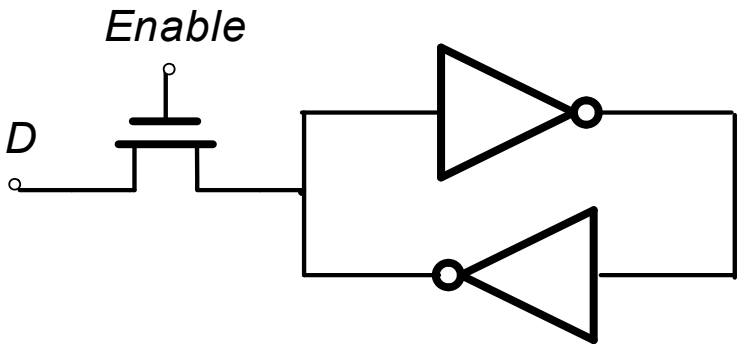


Basic Static Storage Element



- If D is high, D_b will be driven low
  - Which makes D stay high
- Positive feedback
- Tolerant to noise and other small disturbances

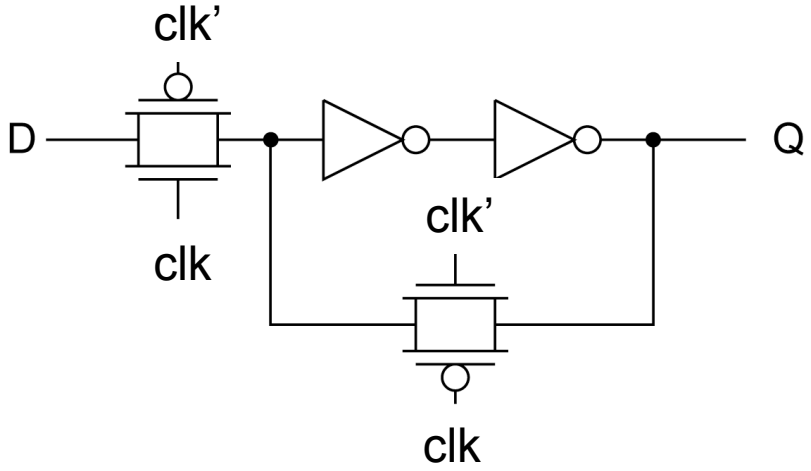
A Static Latch

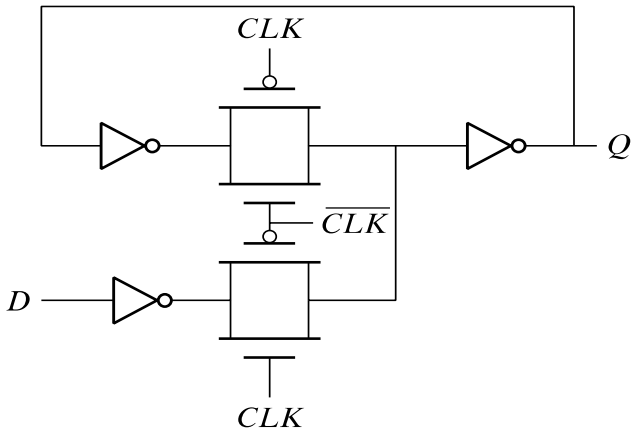


Access transistor must be able to overpower the feedback.

Addressing the write problem

Use the clock as a decoupling signal,
that distinguishes between the transparent and opaque states





Implemented with a MUX.

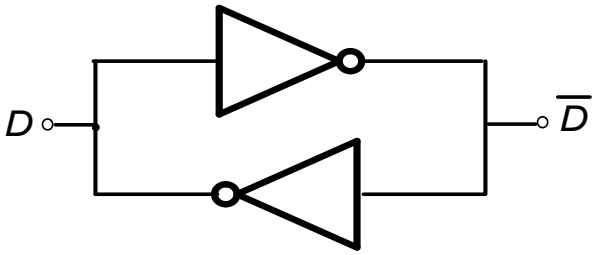
Memories

Semiconductor Memory Classification

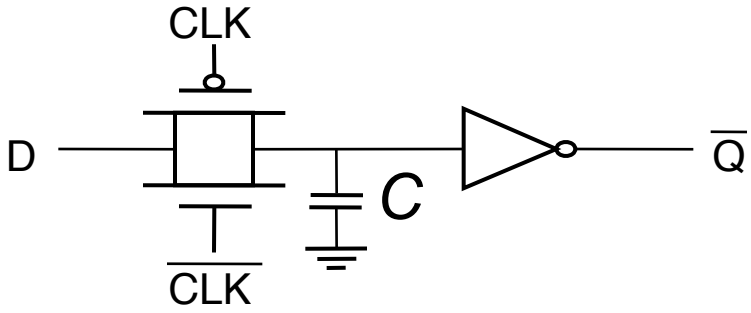
Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	**Non-Random Access**	EPROM	Mask-Programmed
		E^2PROM	Programmable (PROM)
SRAM	FIFO	FLASH	
DRAM	LIFO		
	Shift Register		
	CAM		

Storage Mechanisms

Static (SRAM)

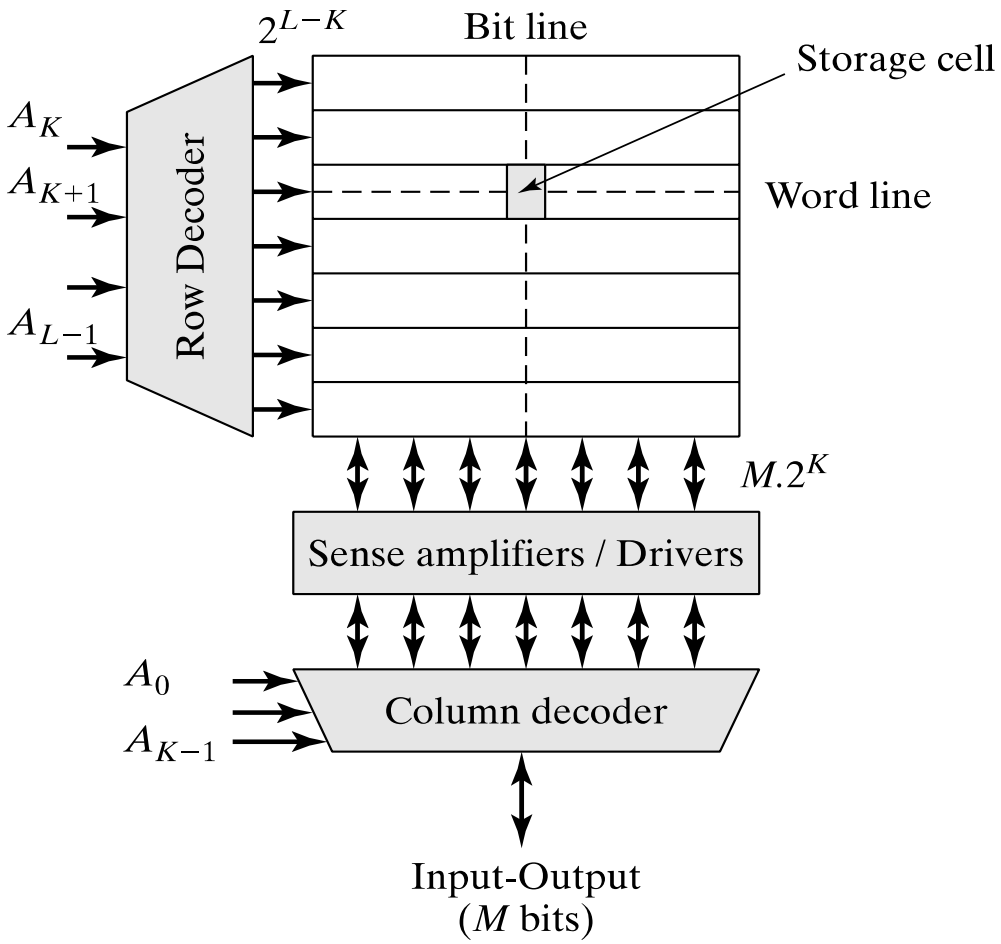


Dynamic (DRAM)



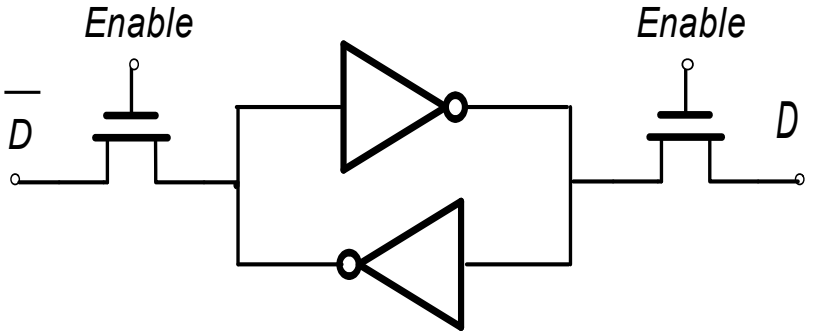
Memory Architecture Overview

- **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



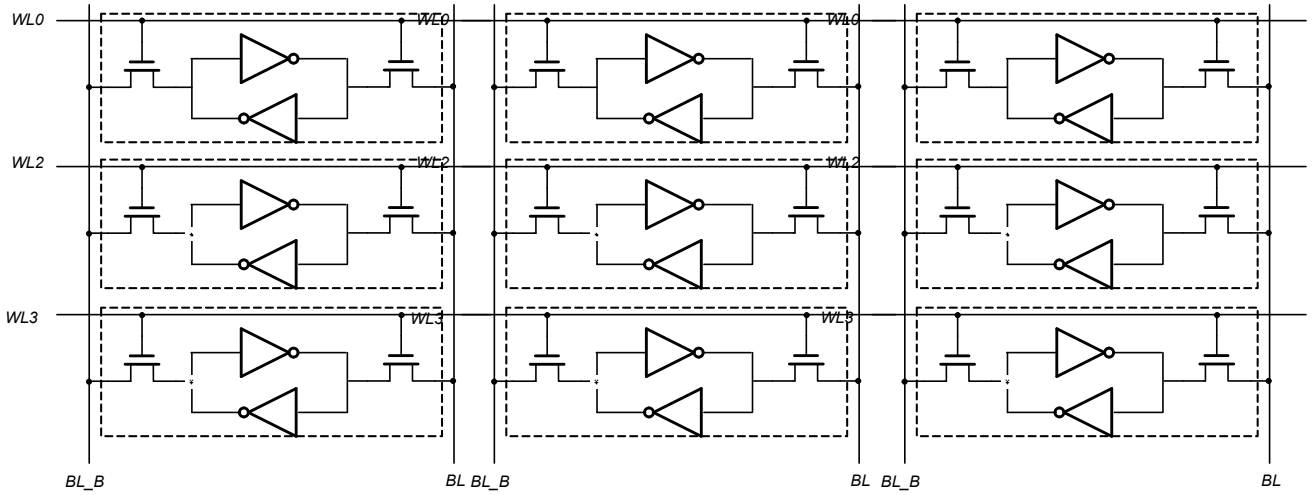
Memory - SRAM

Memory Cells

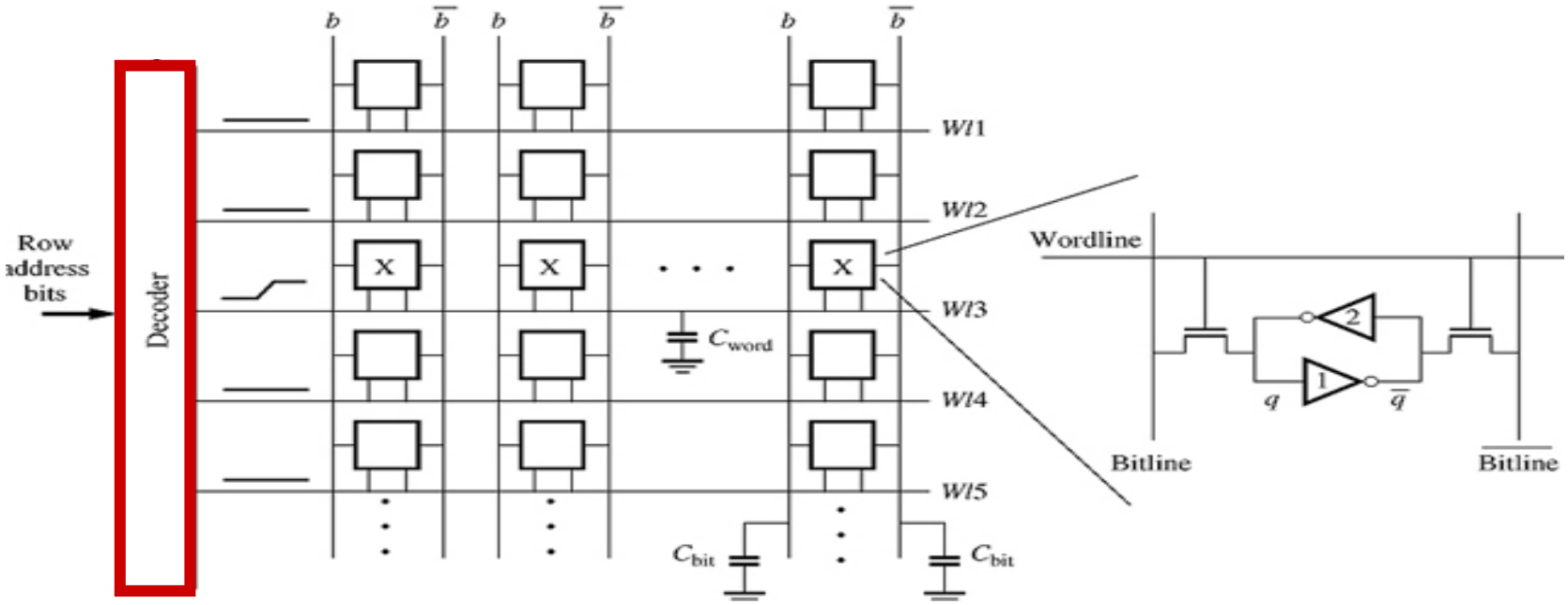


Complementary data values are written (read) from two sides

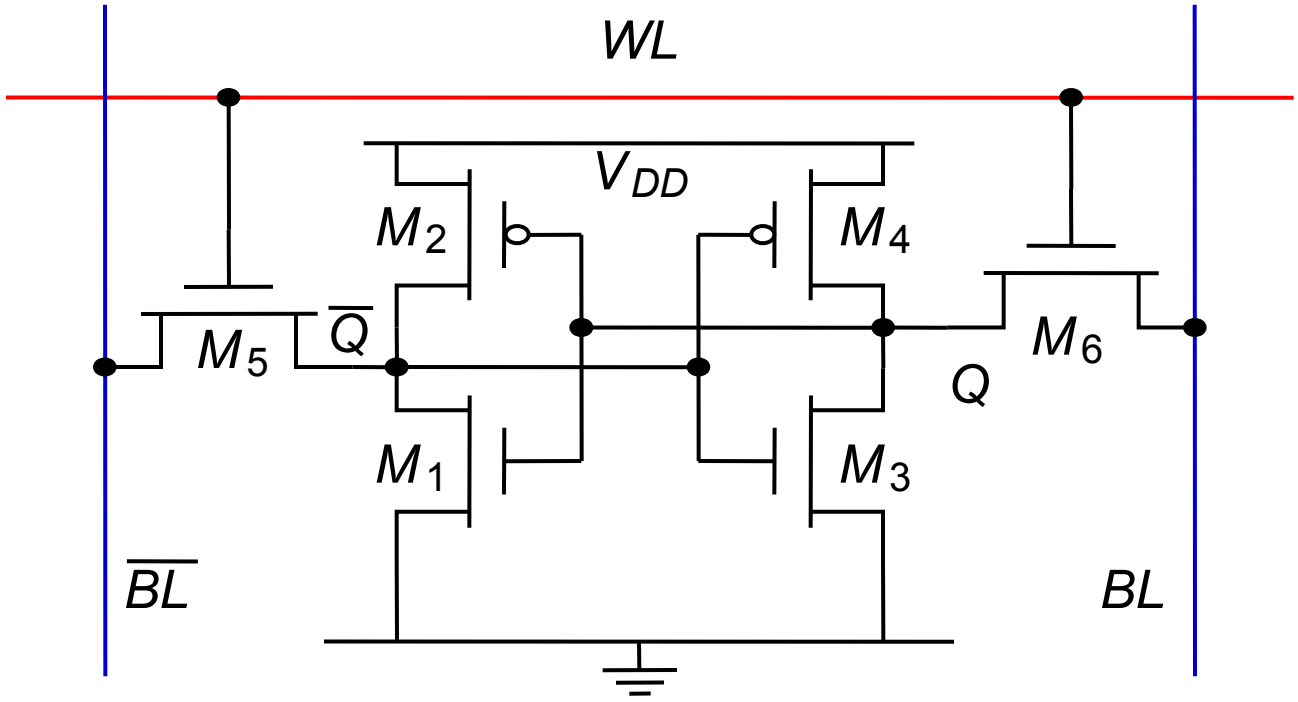
Cells stacked in 2D to form memory core.



SRAM read/write operations

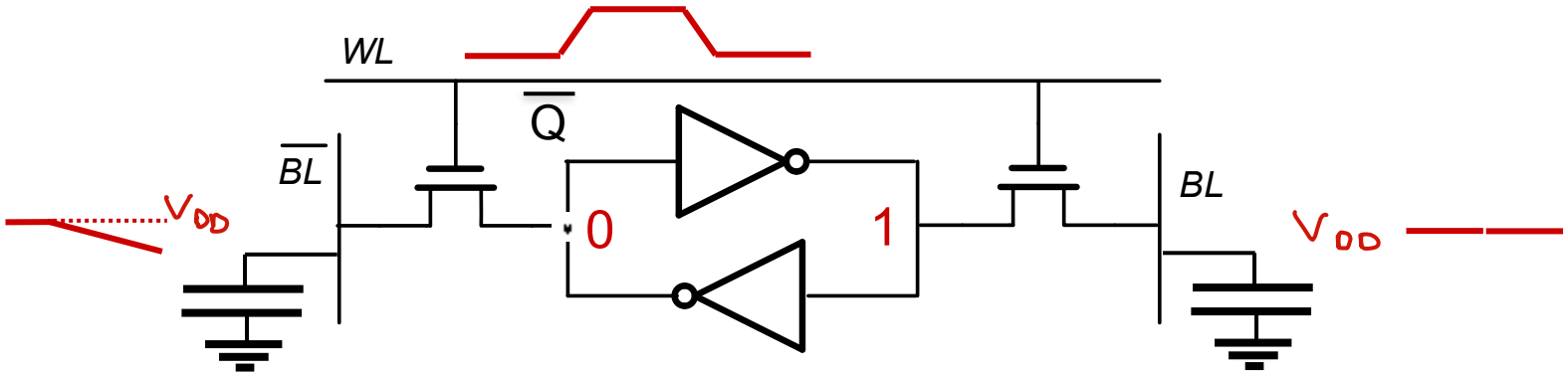


6-Transistor CMOS SRAM Cell



SRAM Operation - Read

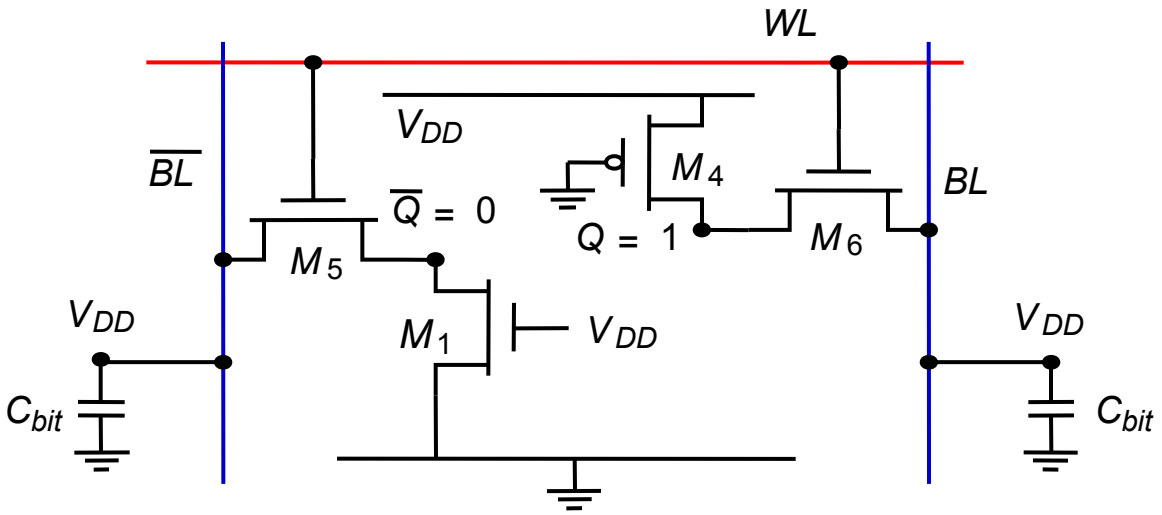
1. Bit lines are “pre-charged” to VDD
2. Word line is driven high (pre-charger is turned off)
3. Cell pulls-down one bit line
4. Differential sensing circuit on periphery is activated to capture value on bit lines.

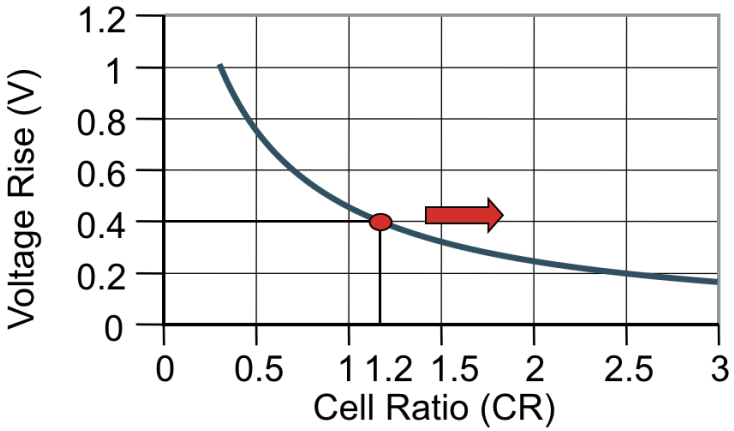


During read $\overline{Q}$ will get pulled up when WL first goes high, but …

- Reading the cell should not destroy the stored value

CMOS SRAM Analysis (Read)

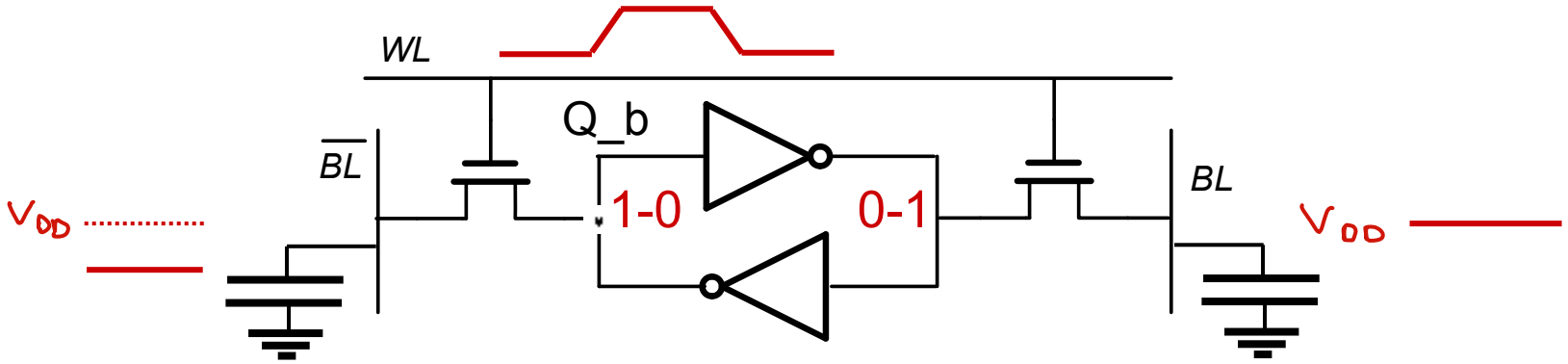




$$CR = \frac{W_1/L_1}{W_5/L_5}$$

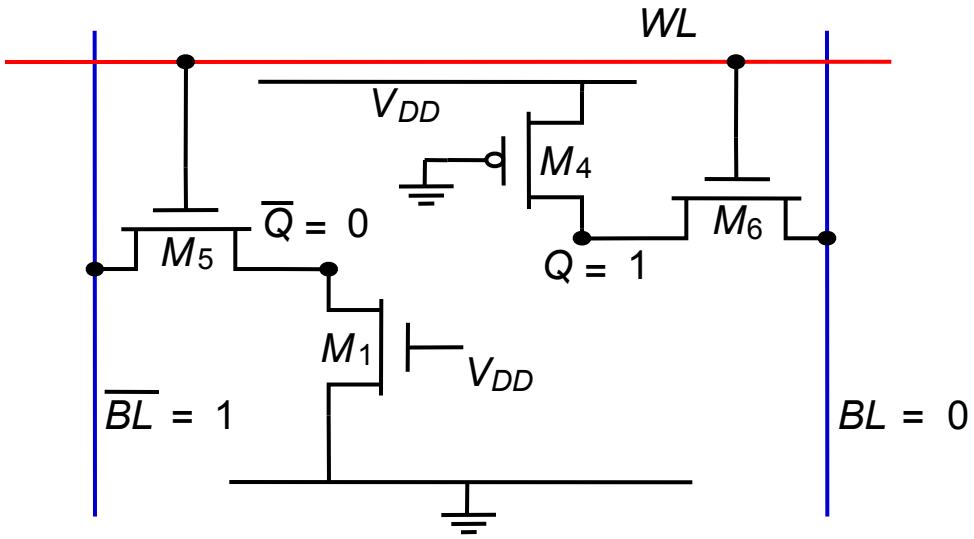
SRAM Operation - Write

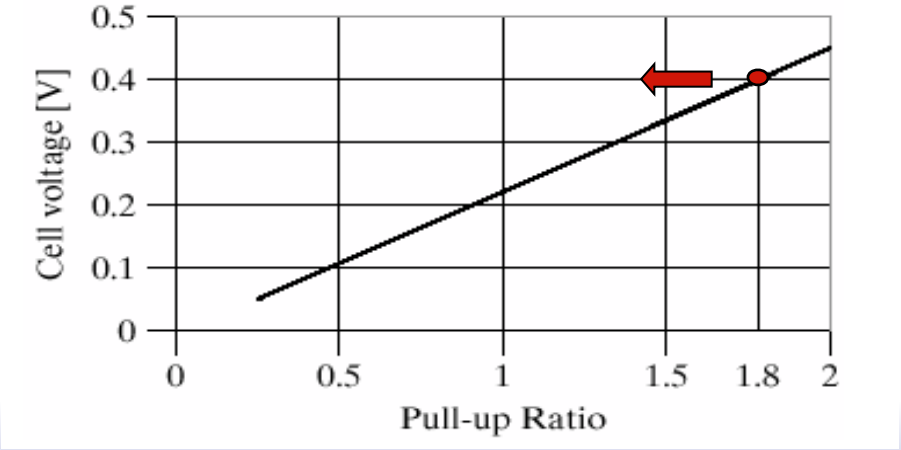
1. Column driver circuit on periphery differentially drives the bit lines
2. Word line is driven high (column driver stays on)
3. One side of cell is driven low, flips the other side



For successful write the access transistor needs to overpower the cell pullup

CMOS SRAM Analysis (Write)



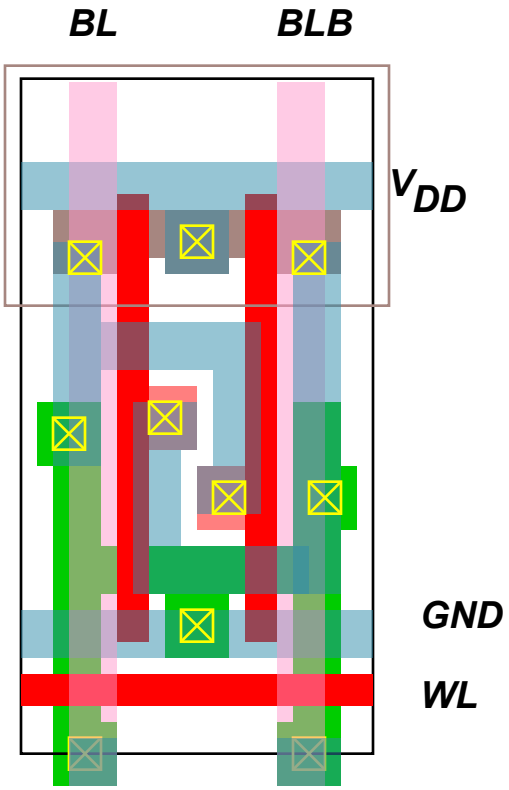


$$\frac{W_4/L_4}{W_6/L_6}$$

Size width ratio between PMOS pull-up and NMOS access

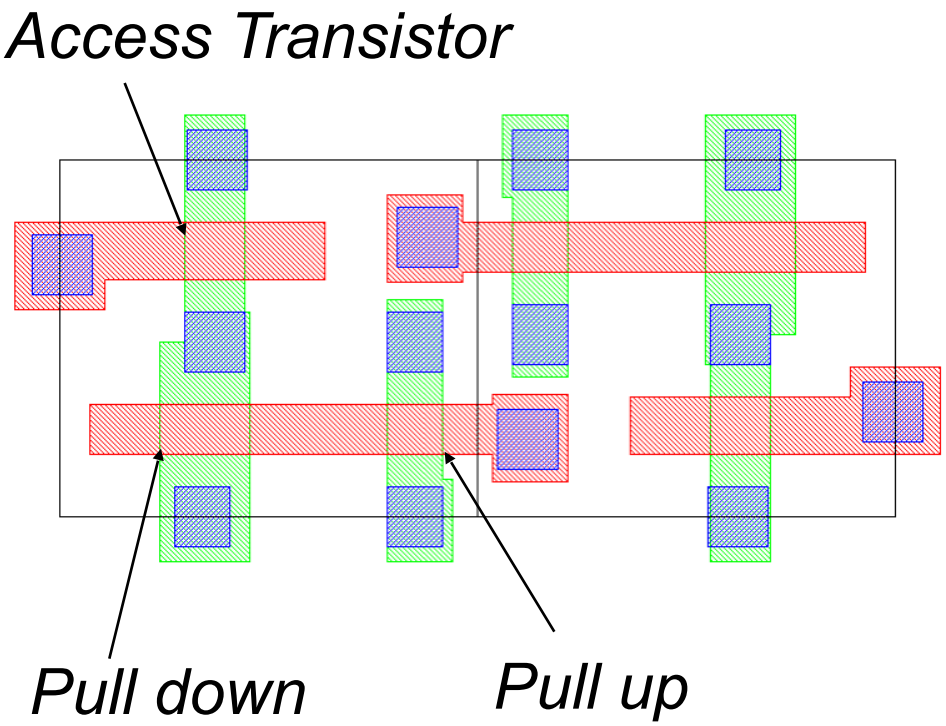
6T-SRAM — Layout

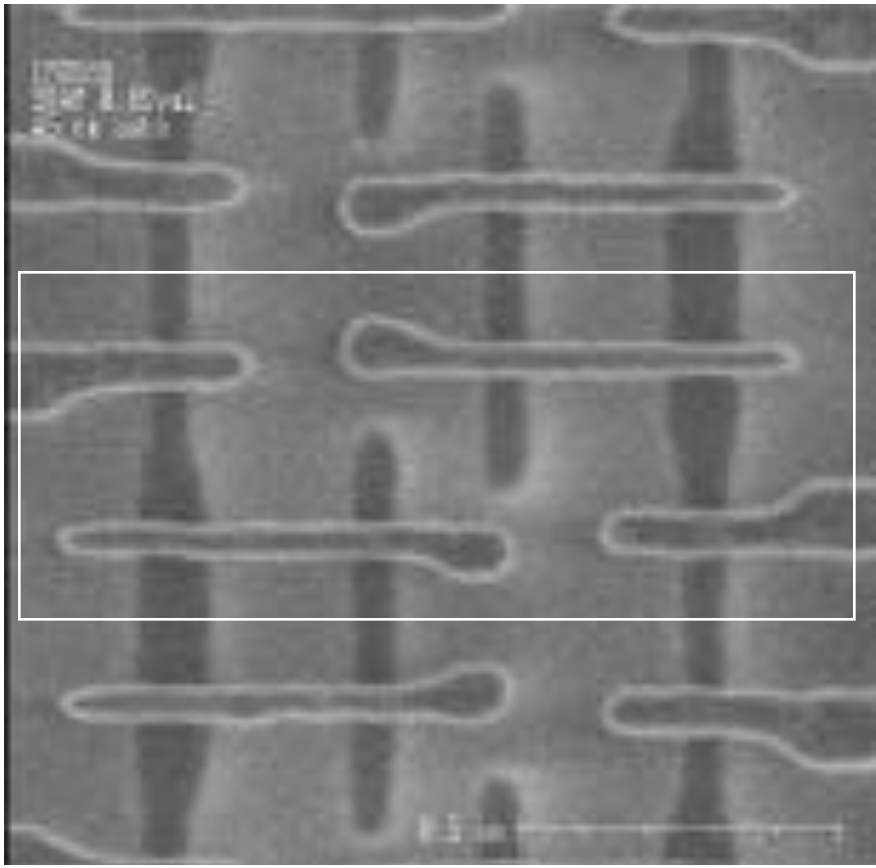
V_{DD} and GND: in M1
Bitlines: M2
Wordline: poly-silicon



65nm SRAM

- ST/Philips/Motorola





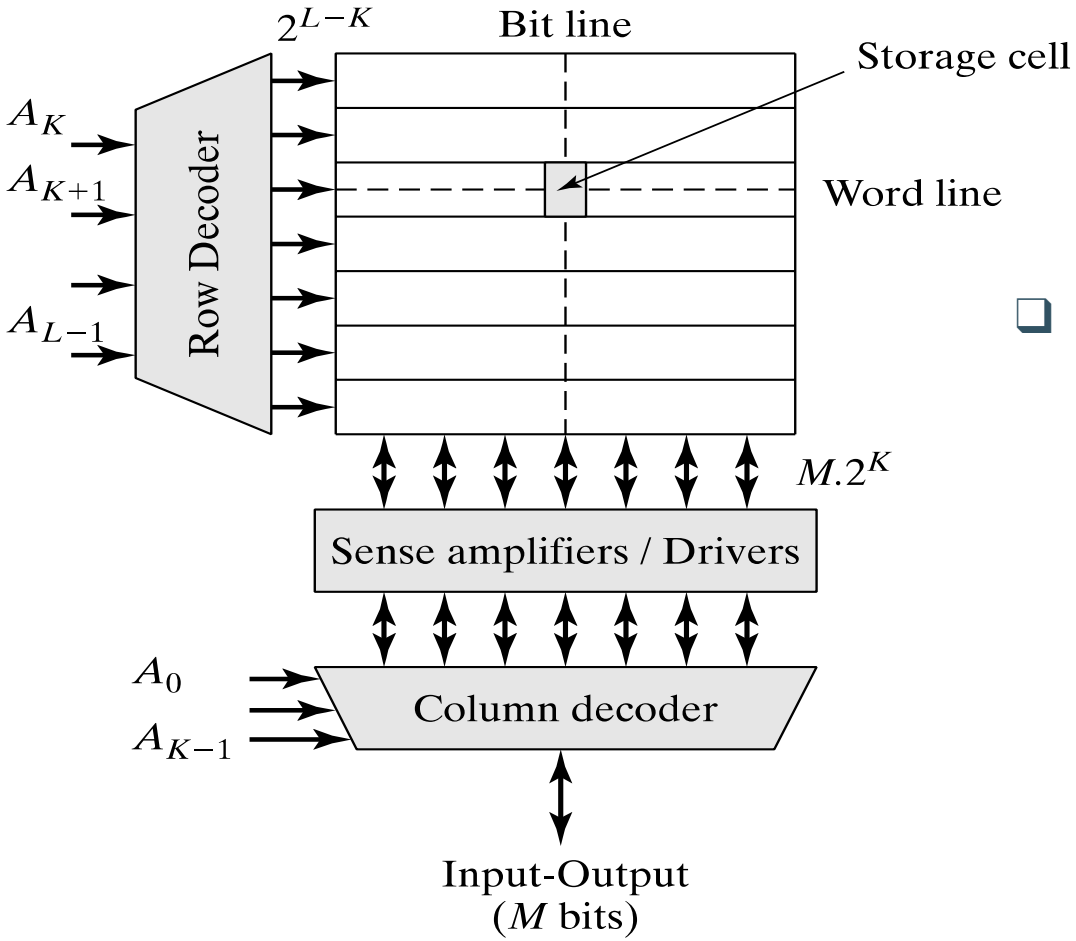
Memory Periphery

Periphery

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

Row Decoder

- Expands L-K address lines into 2^{L-K} word lines



- Example: decoder for 8Kx8 memory block
  - core arranged as 256x256 cells
  - Need 256 AND gates, each driving one word line

Row Decoders

(N)AND Decoder

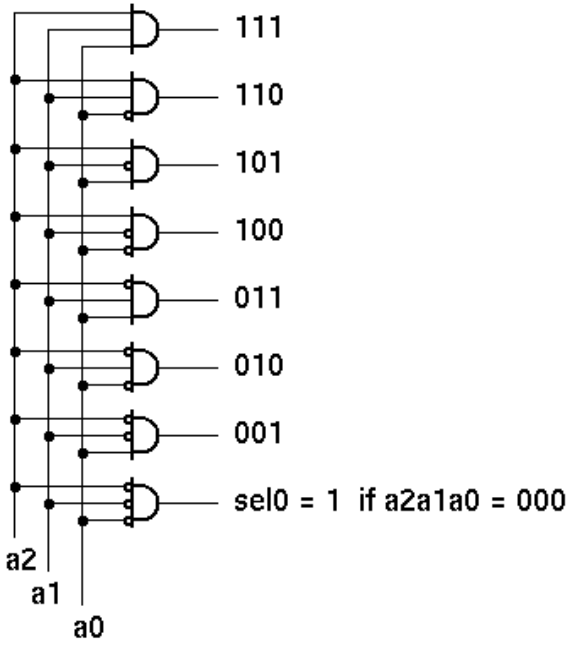
$$WL_0 = A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9$$

$$WL_{511} = \bar{A}_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9$$

NOR Decoder

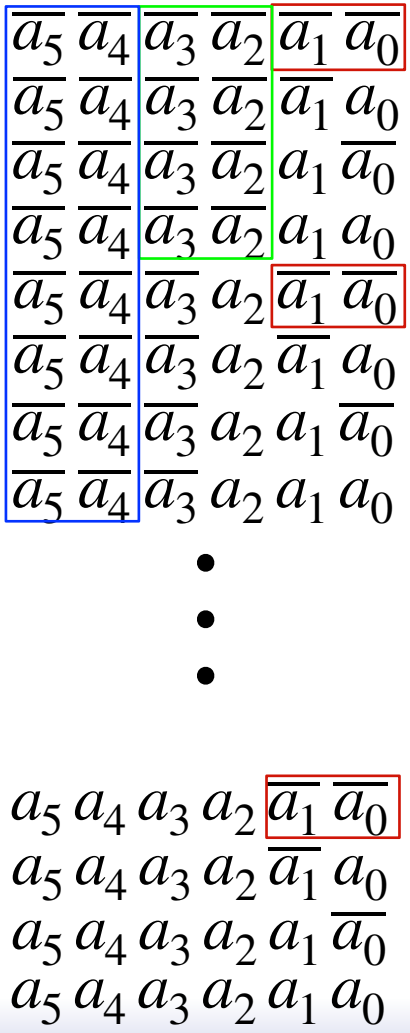
$$WL_0 = \overline{A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

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



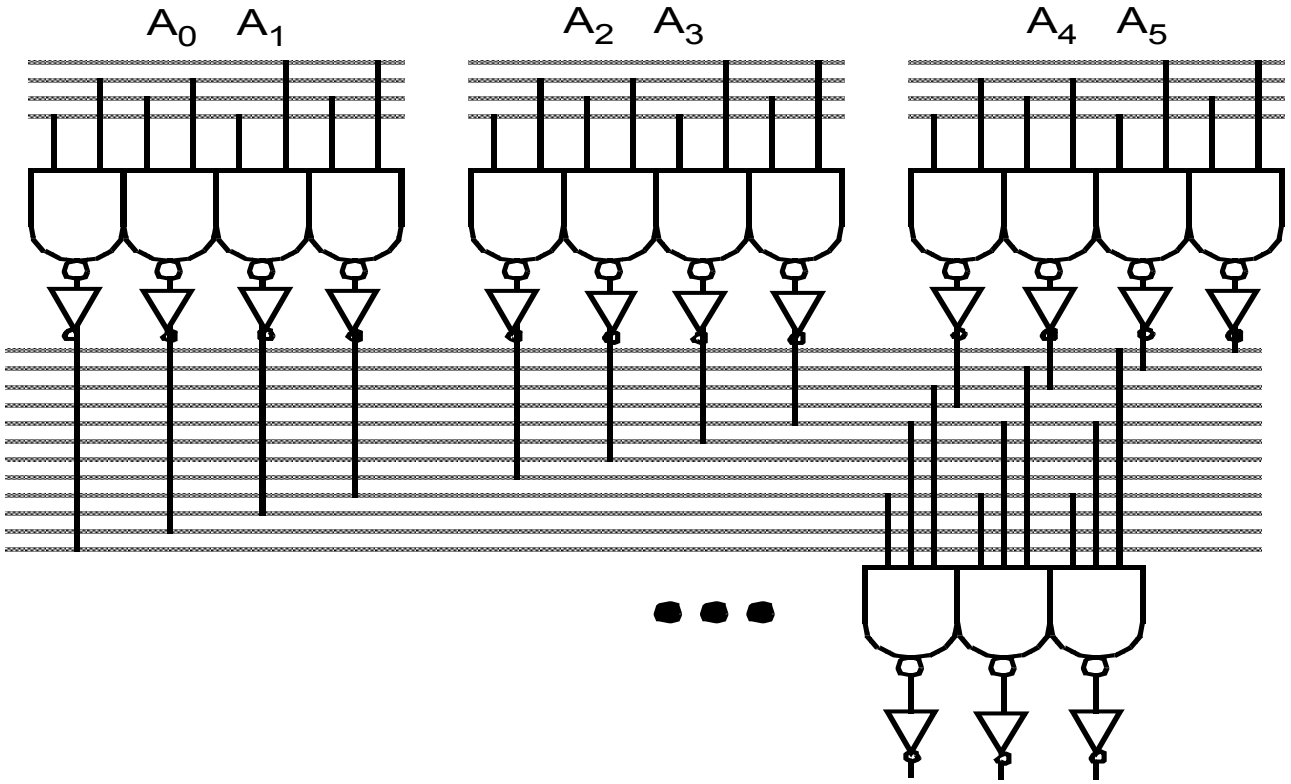
Collection of 2^K logic gates, but need to be dense and fast.

Predecoders

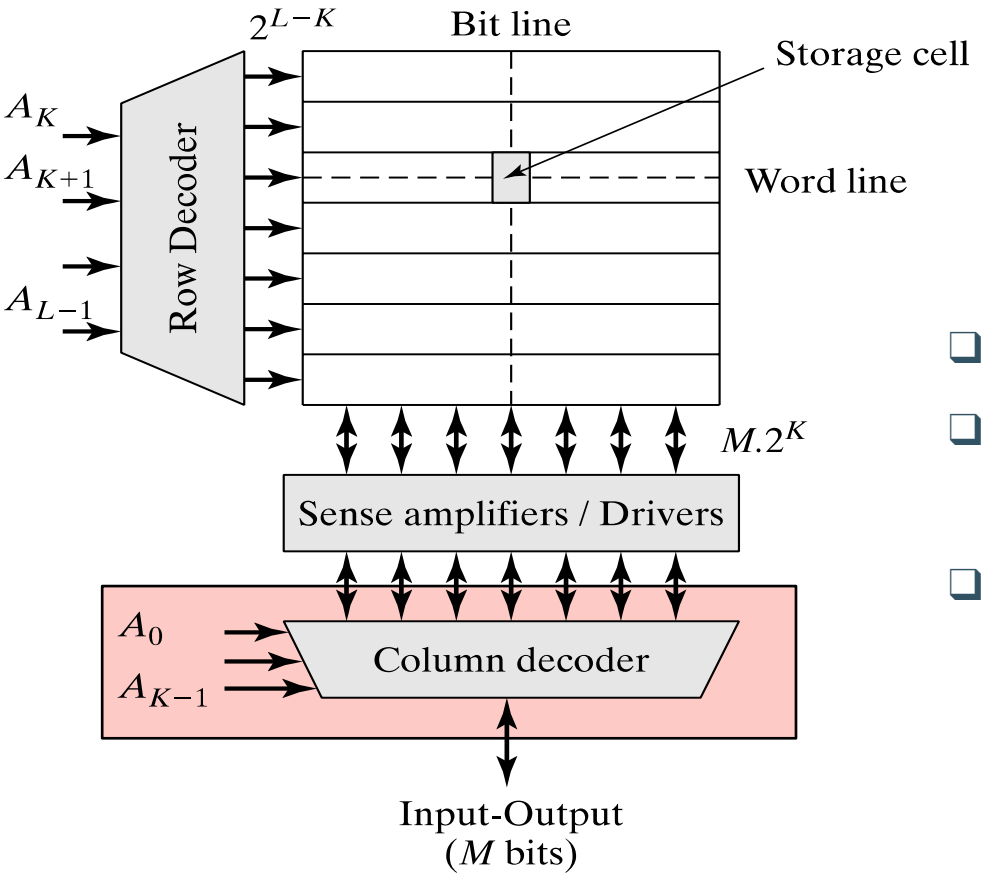


- Use a single gate for each of the shared terms
  - E.g., from $a_1, \overline{a_1}, a_0, \overline{a_0}$ generate four signals:
  - $\overline{a_1}\ \overline{a_0}, \overline{a_1}\ a_0, a_1\ \overline{a_0}, a_1\ a_0$
- In other words, we are decoding smaller groups of address bits first
  - And using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

Predecoder and Decoder



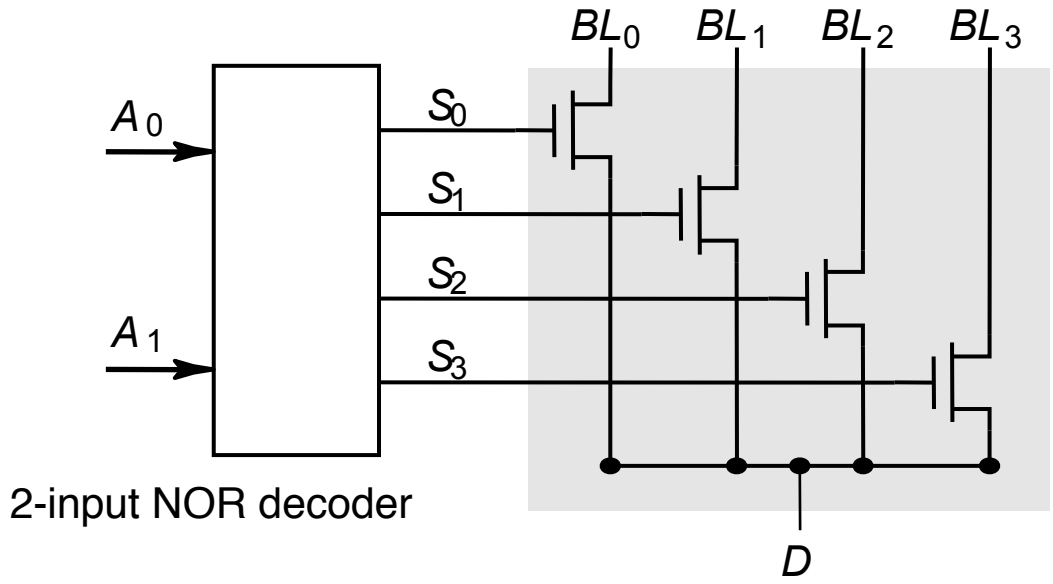
Column "Decoder"



- Is basically a multiplexer
- Each row contains 2^K words each M bit 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-input pass-transistor based Column Decoder



decoder shared across all 2^K x M row bits

Advantages: speed (t_{pd} does not add to overall memory access time)

Only one extra transistor in signal path

Sense Amplifiers

Sense Amplifiers

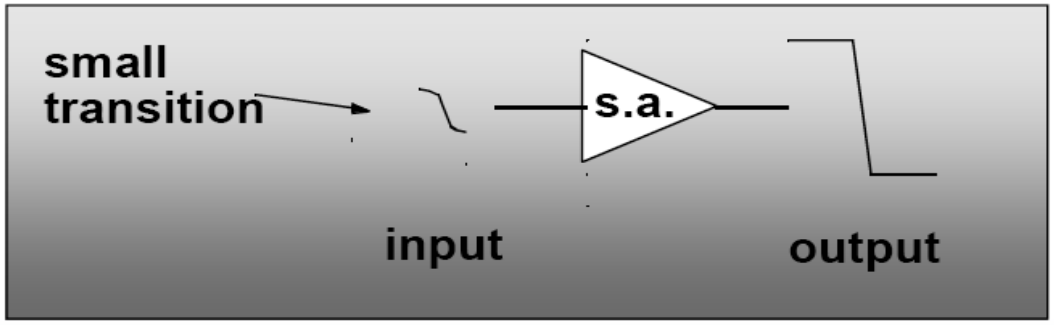
large

make as small as possible

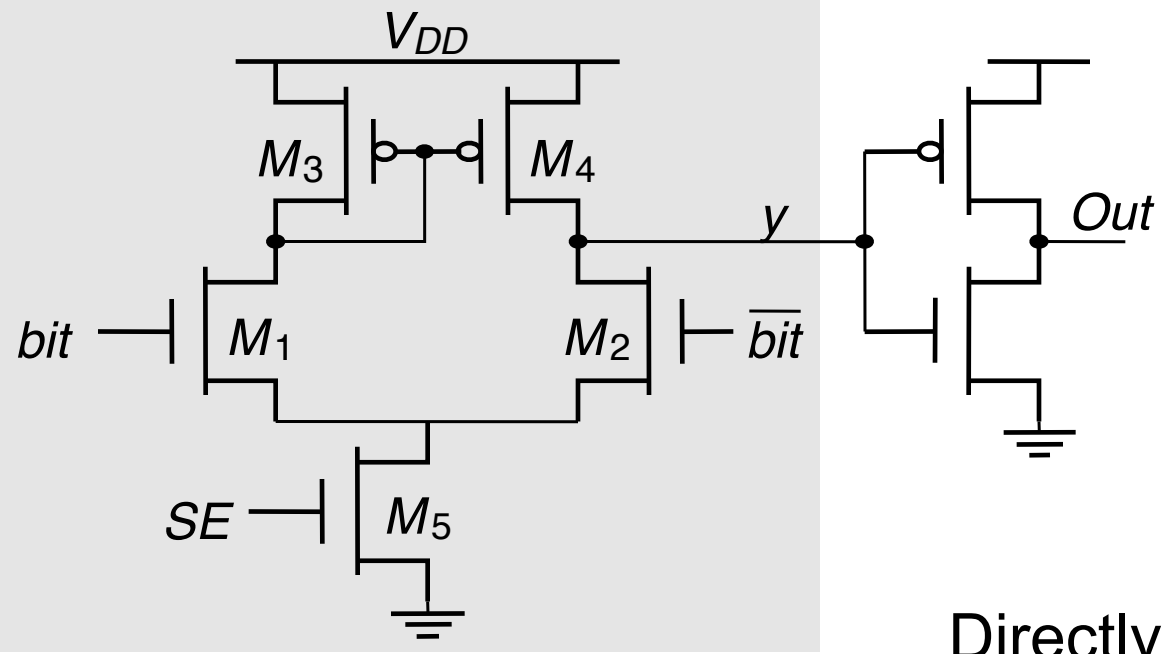
$$\tau_p = \frac{C \cdot \Delta V}{I_{av}}$$

small

Idea: Use Sense Amplifer

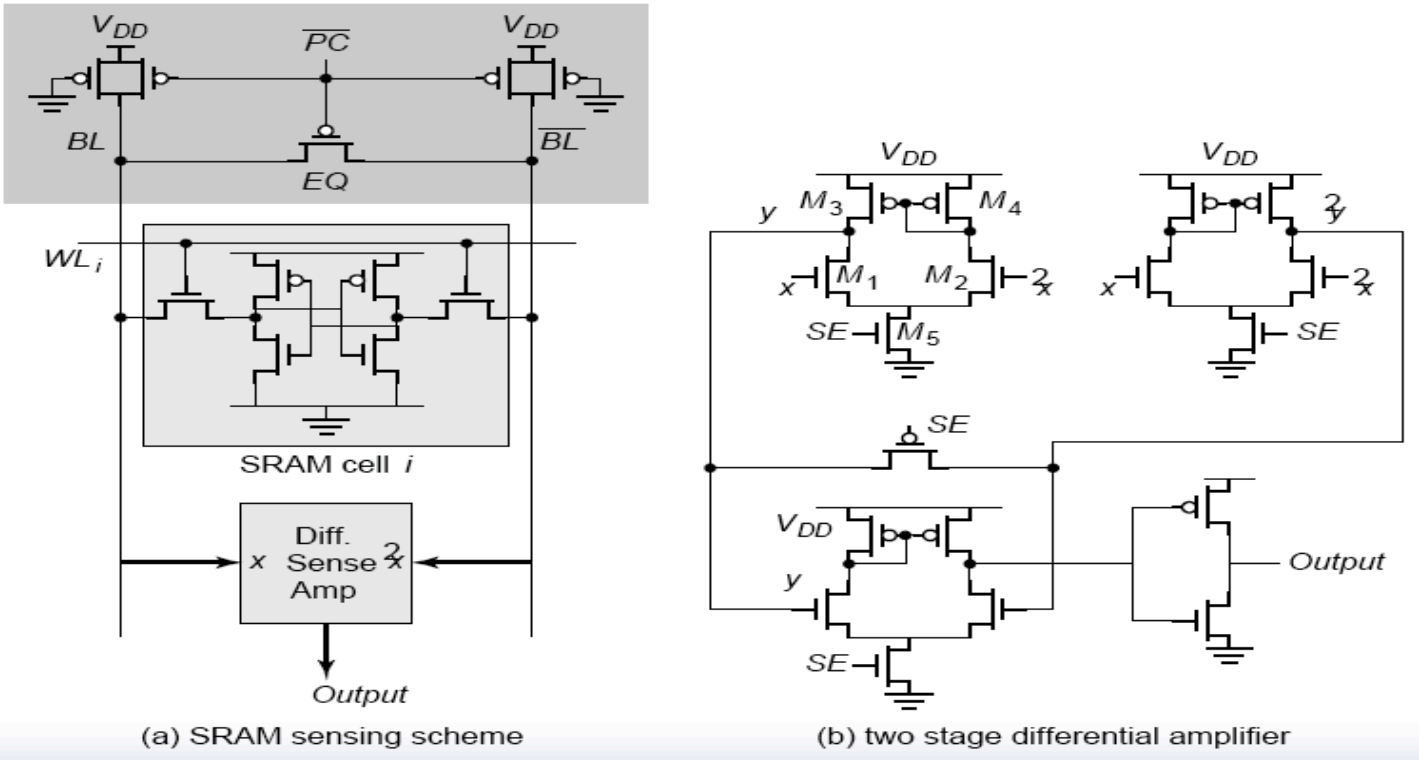


Differential Sense Amplifier



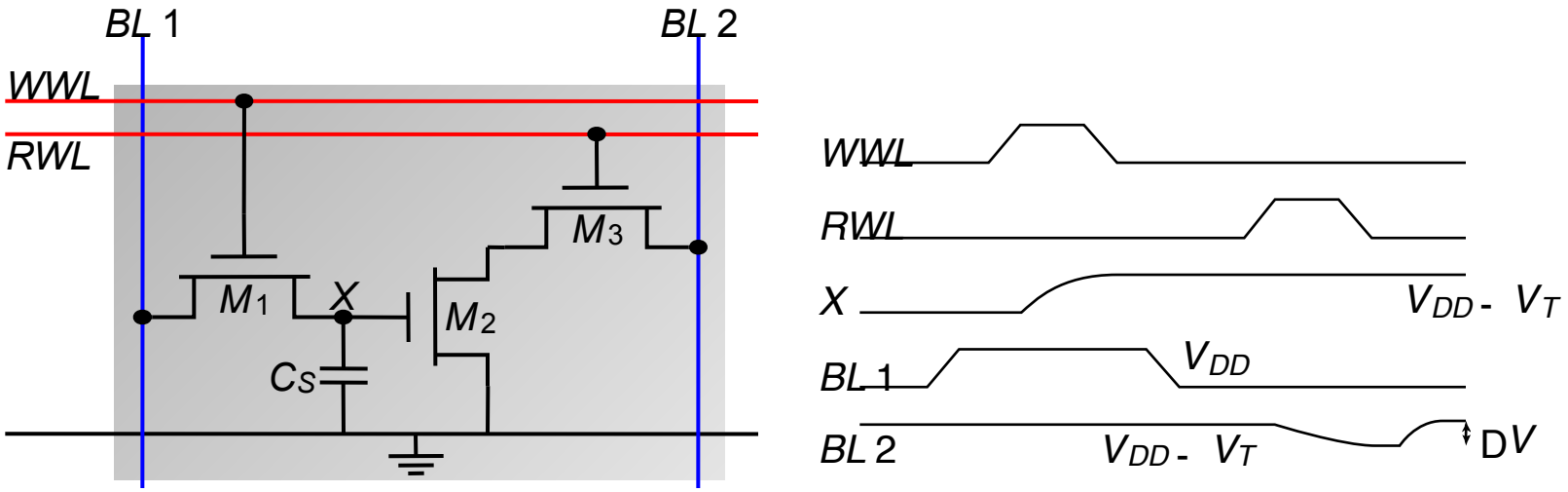
Directly applicable to SRAMs

Differential Sensing – SRAM



Memory - DRAM

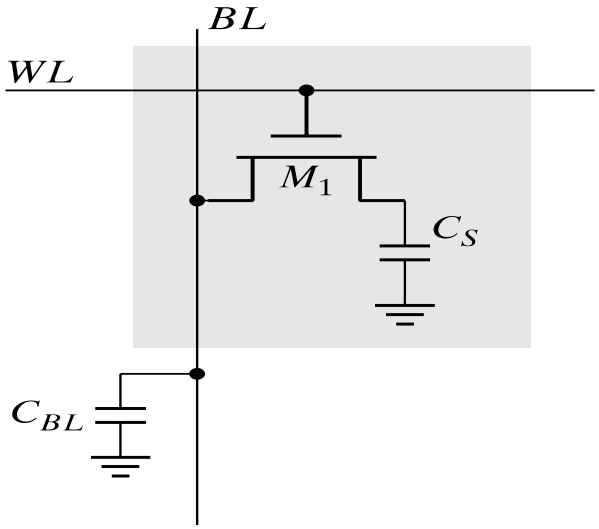
3-Transistor DRAM Cell

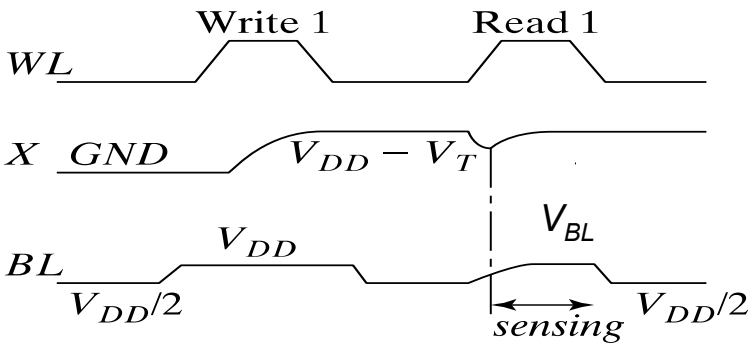


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a "1" = $V_{WWL} - V_{Tn}$

Can work with a normal logic IC process

1-Transistor DRAM Cell





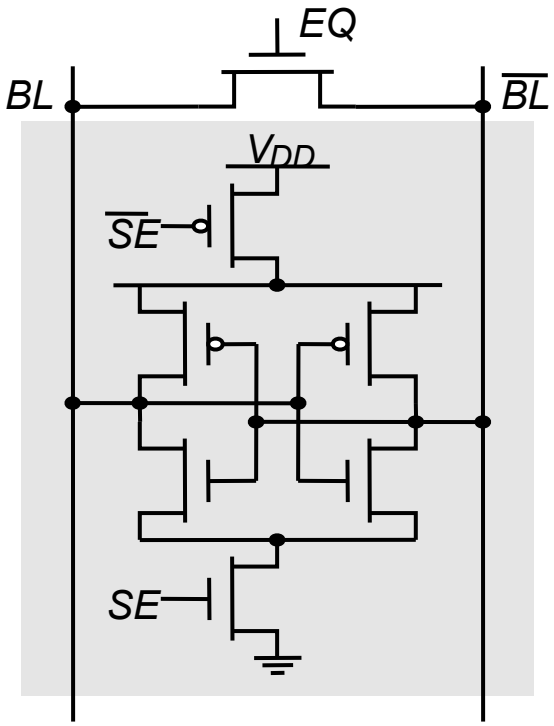
V_{BIT}= 0 or $(V_{DD} - V_T)$

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

$C_S << C_{BL}$ Voltage swing is small; typically around 250 mV.

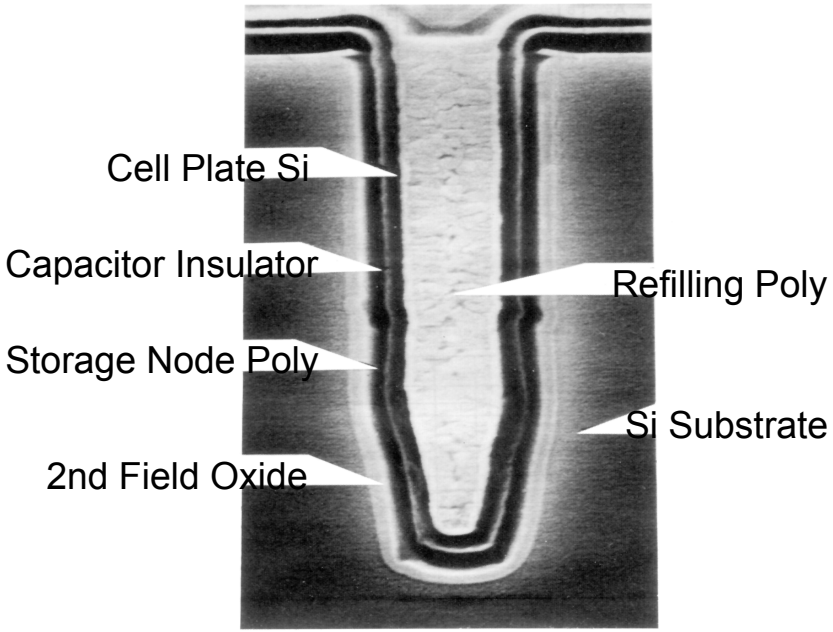
- To get sufficient C_s, special IC process is used
- Cell reading is destructive, therefore read operation always is followed by a write-back
- Cell looses charge (leaks away in ms - highly temperature dependent), therefore cells occasionally need to be “refreshed” - read/write cycle

Latch-Based Sense Amplifier (DRAM)

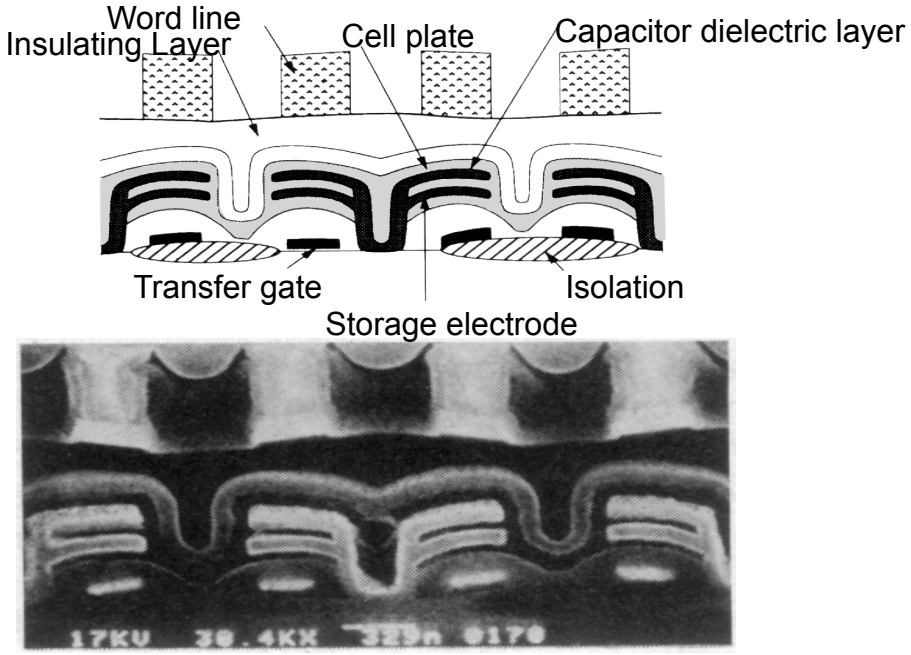


- Initialized in its meta-stable point with EQ
- Once adequate voltage gap created, sense amp enabled with SE
- Positive feedback quickly forces output to a stable operating point.

Advanced 1T DRAM Cells



Trench Cell

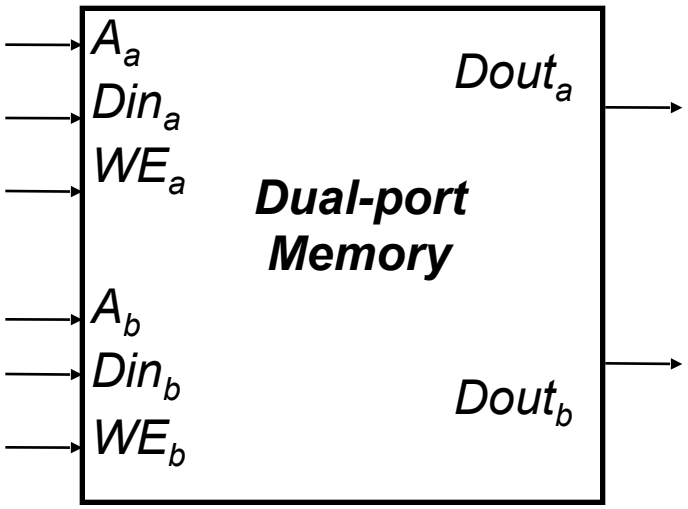


Stacked-capacitor Cell

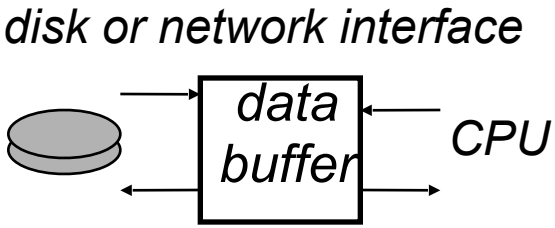
Multi-ported memory

Multi-ported Memory

- Motivation:
  - Consider CPU core register file:
    - 1 read or write per cycle limits processor performance.
    - Complicates pipelining. Difficult for different instructions to simultaneously read or write regfile.
    - Common arrangement in pipelined CPUs is 2 read ports and 1 write port.



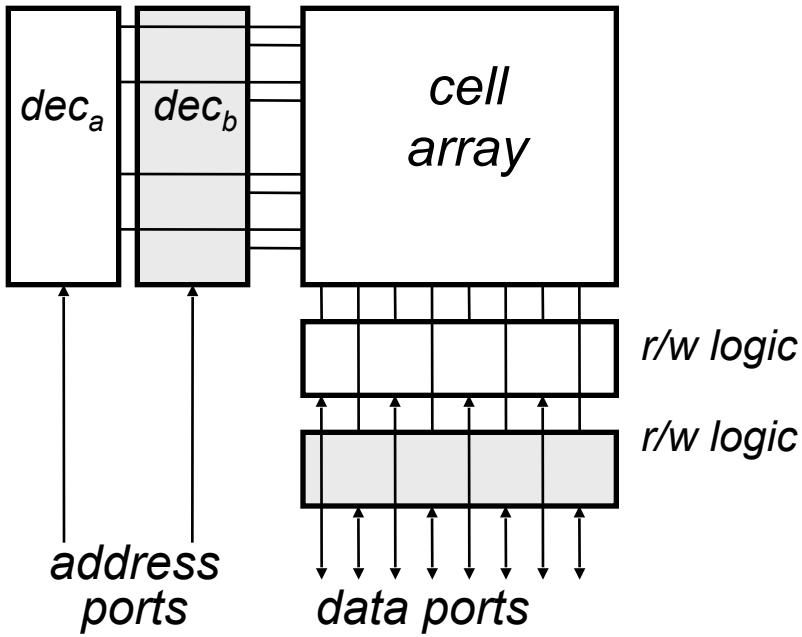
- I/O data buffering:



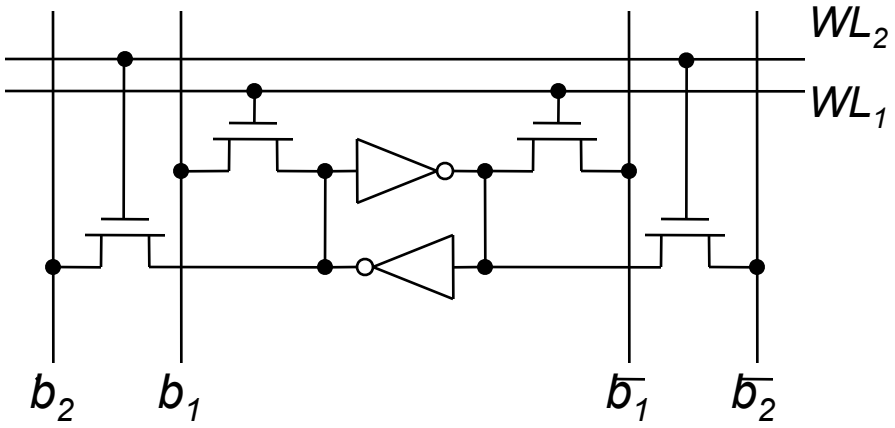
- dual-porting allows both sides to simultaneously access memory at full bandwidth.

Dual-ported Memory Internals

- Add decoder, another set of read/write logic, bits lines, word lines:



- *Example cell: SRAM*



- *Repeat everything but cross-coupled inverters.*
- *This scheme extends up to a couple more ports, then need to add additional transistors.*