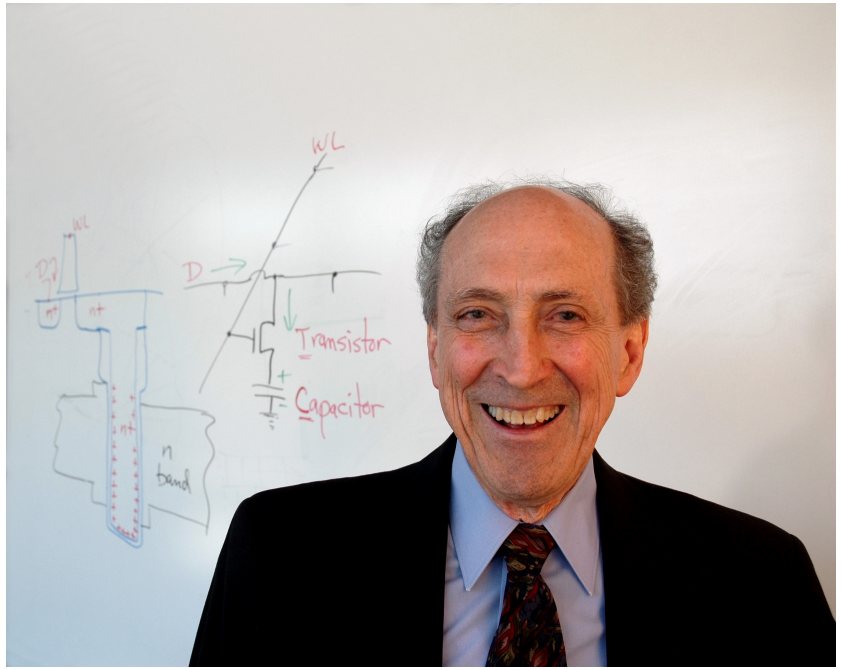
EECS151/251A
Introduction to Digital Design and ICs

Lecture 23: SRAM II

Sophia Shao

Robert Dennard

- Invented DRAM in 1968 at IBM
- Formulate Dennard's scaling: Maintain constant power density with improved frequency/performance
- The end of Dennard's scaling leads to the inability to further increase clock frequencies and multicore processors as an alternative way to improve processor performance

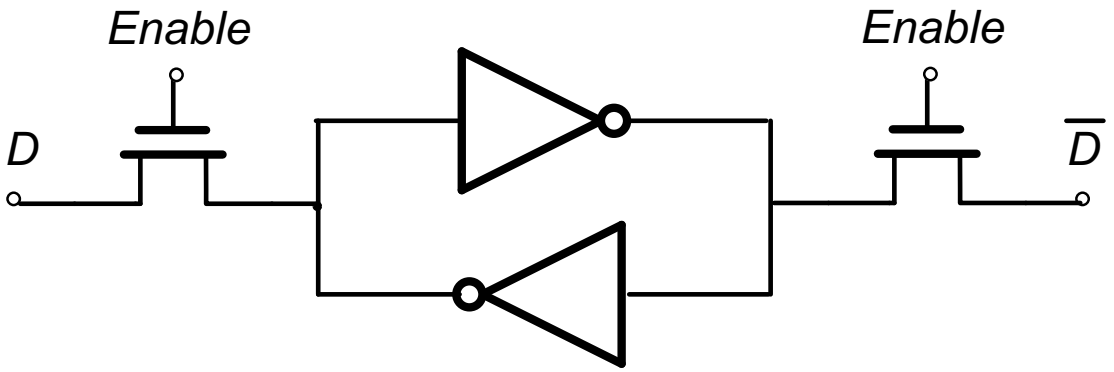


Review

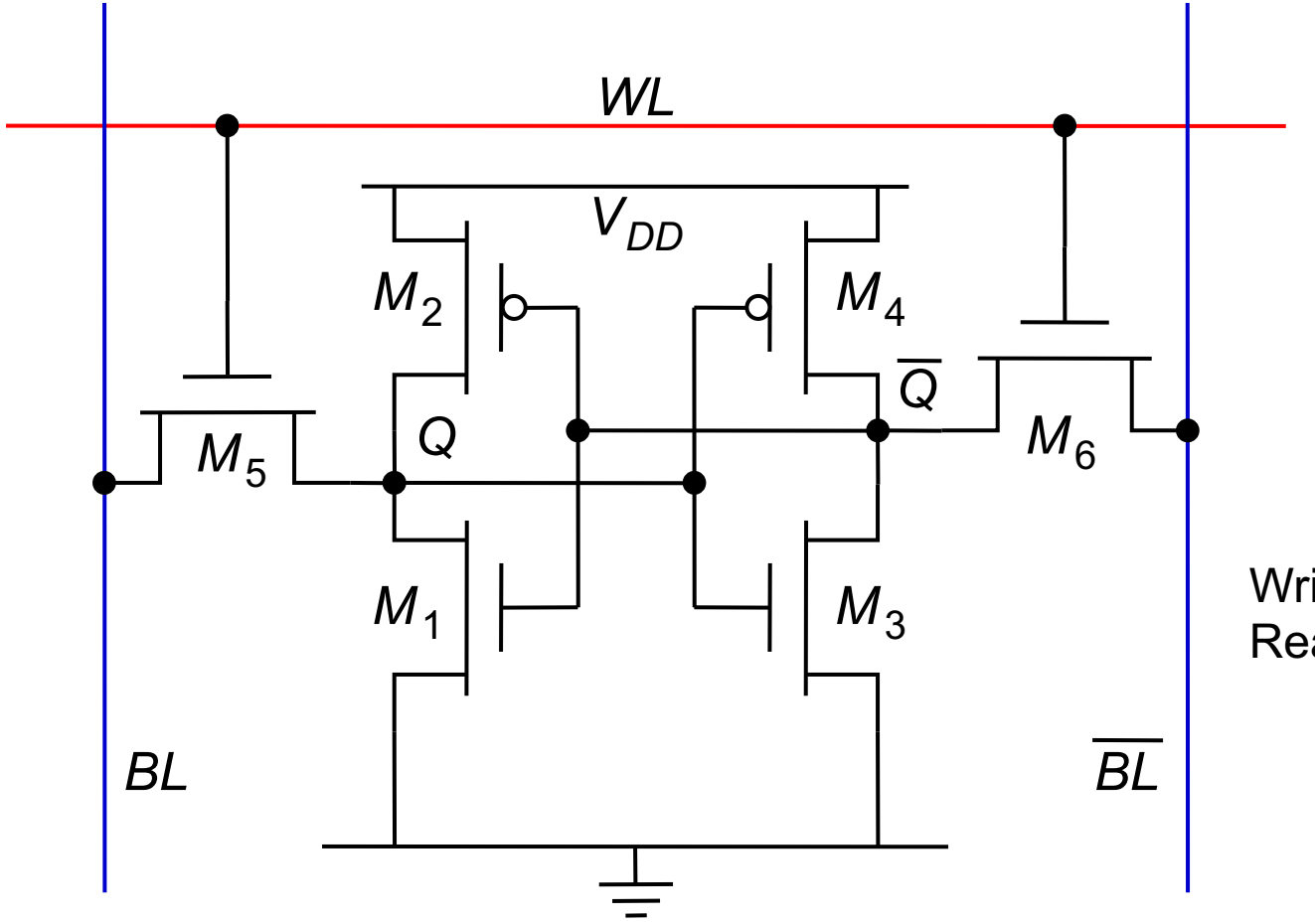
- Flip-flop is typically a latch pair
- Setup and hold times are defined at constant percentage increases over clk-q delay
- SRAM has unique combination of density, speed, power

- SRAM
  - 6-T SRAM Cell
  - Sizing SRAM Cell
- Memory Decoder
  - Overview
  - Decoder Design
  - Pre-decoder
- Multi-Ported SRAM

SRAM Cell



6-transistor CMOS SRAM Cell

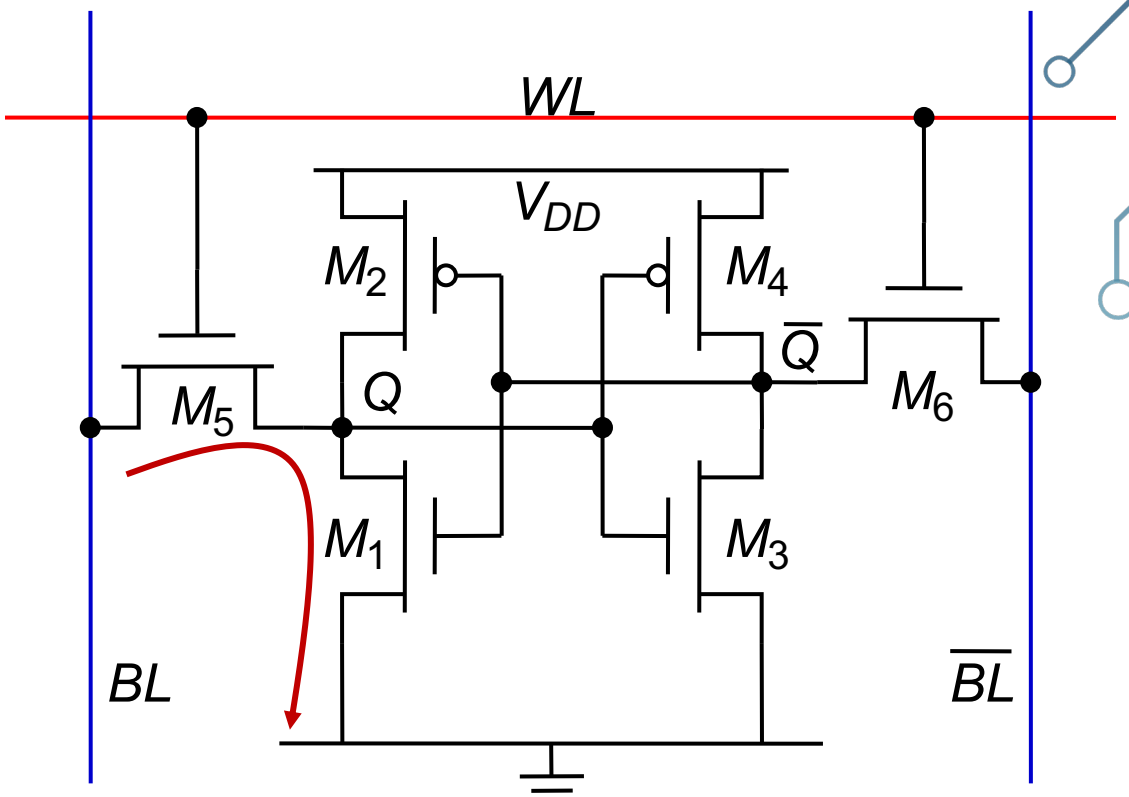


Write: BL and $\overline{BL}$ are inverted.
Read: BL and $\overline{BL}$ are Vdd

- Wordline (WL) enables read/write access for a row
- Data is written/read differentially through shared BL, $\overline{BL}$

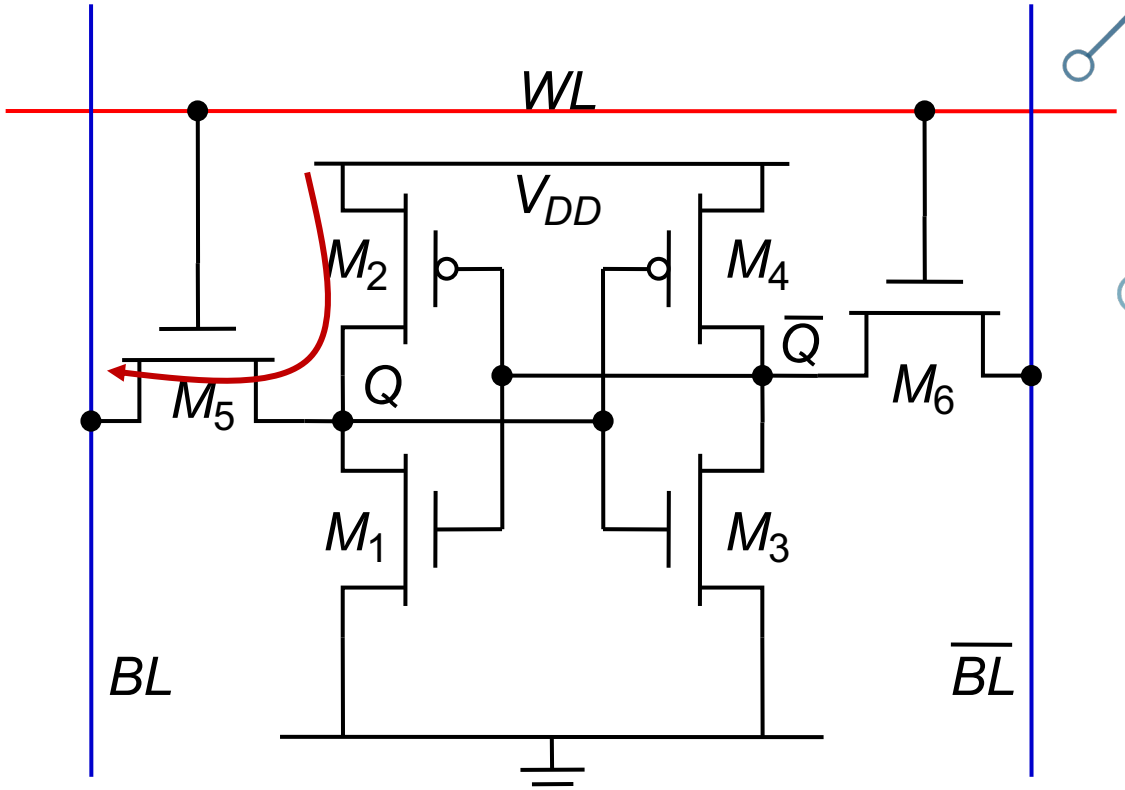
Sizing SRAM Cell

- Read stability: Cell should not change value during read
  - Q = 0: M_5, M_1 both on
  - Voltage divider between M_5, M_1
  - V_Q should stay low, not to flip M_4-M_3 inverter
  - R1 < R5 => $(W/L)_1 > (W/L)_5$
- Typically $(W/L)_1 = 1.5\ (W/L)_5$
  - In FinFETs: $(W/L)_1 = 2(W/L)_5$
- Read speed: Both M_5 and M_1



Sizing SRAM Cell

- Writeability: Cell should be writeable by pulling BL low
  - Q = 1, M_5, M_2 both on
  - Voltage divider between M_5, M_2
  - V_Q should pull below the switching point of M_4-M_3 inverter
  - R5 < R2 => $(W/L)_5 > (W/L)_2$
- Typically $(W/L)_5 = (W/L)_2$ in planar
  - In FinFETs: $(W/L)_5 = 2(W/L)_2$
  - Pull Up: Access: Pull Down:
    - 1:2:2 and 1:2:3 sizing

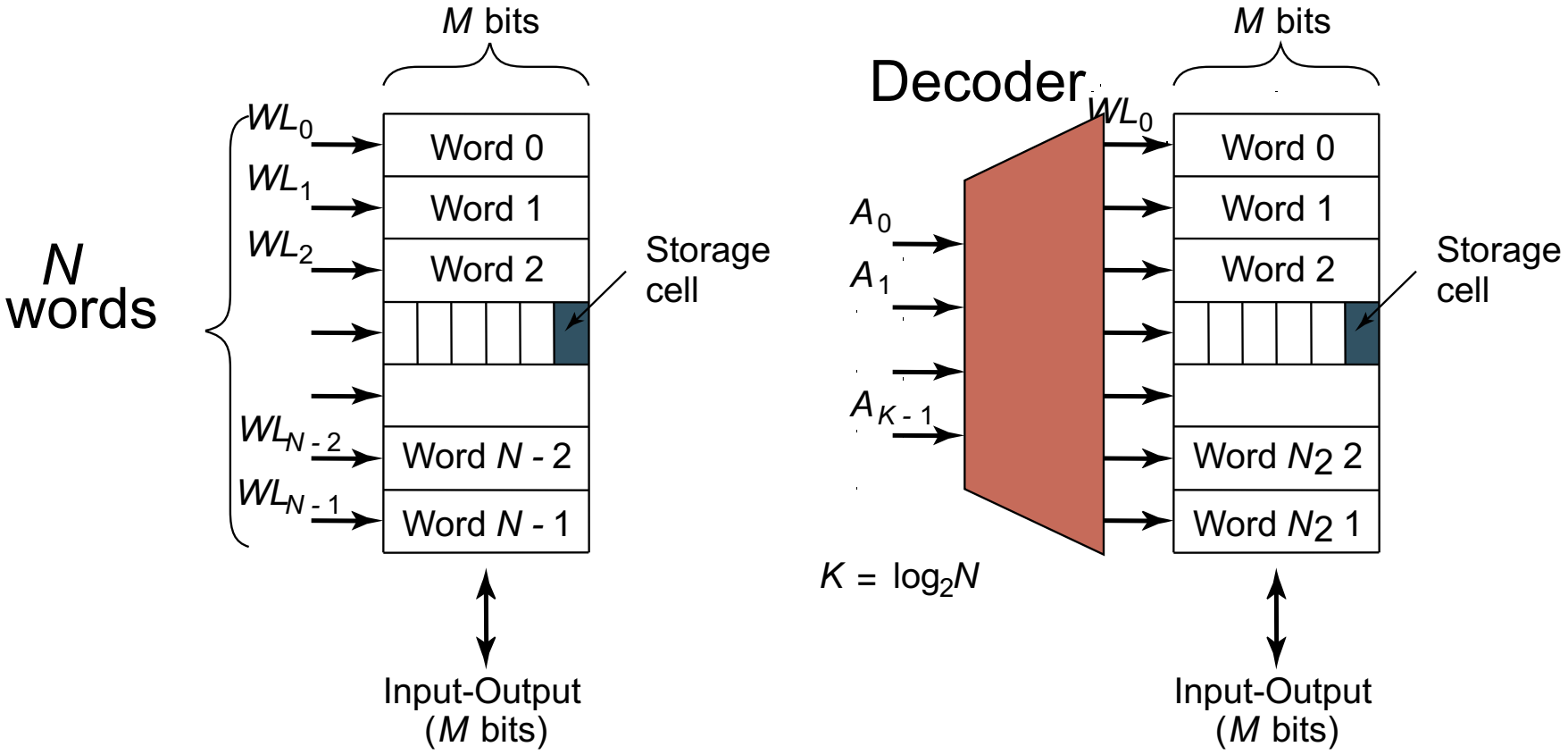


True or False

1. Transistor leakage doesn’t affect SRAM read speed.
2. One should write into an SRAM cell by pulling BL high.
3. One can write into a part of a selected WL.

- SRAM
  - 6-T SRAM Cell
  - Sizing SRAM Cell
- Memory Decoder
  - Overview
  - Decoder Design
  - Pre-decoder
- Multi-Ported SRAM

Decoders

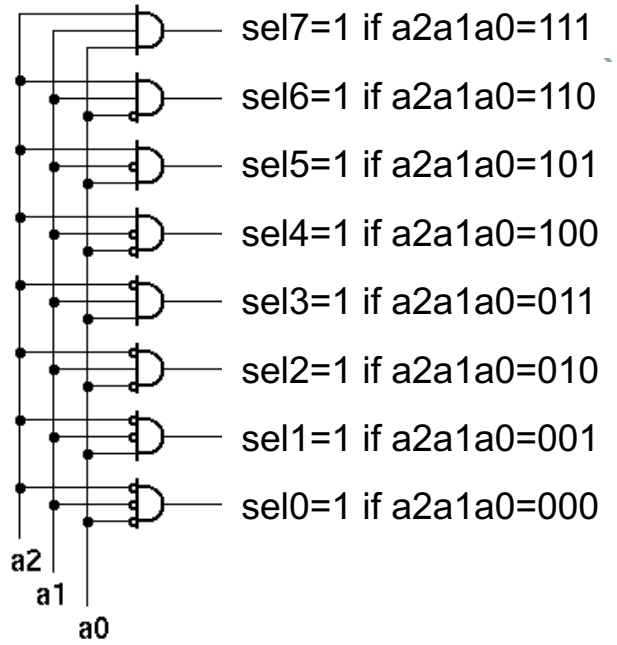


Intuitive architecture for N x M memory
Too many select signals:
N words = N select signals

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

Row Decoders

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



(N)AND Decoder

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

$$WL_{511} = A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 \overline{A_9}$$

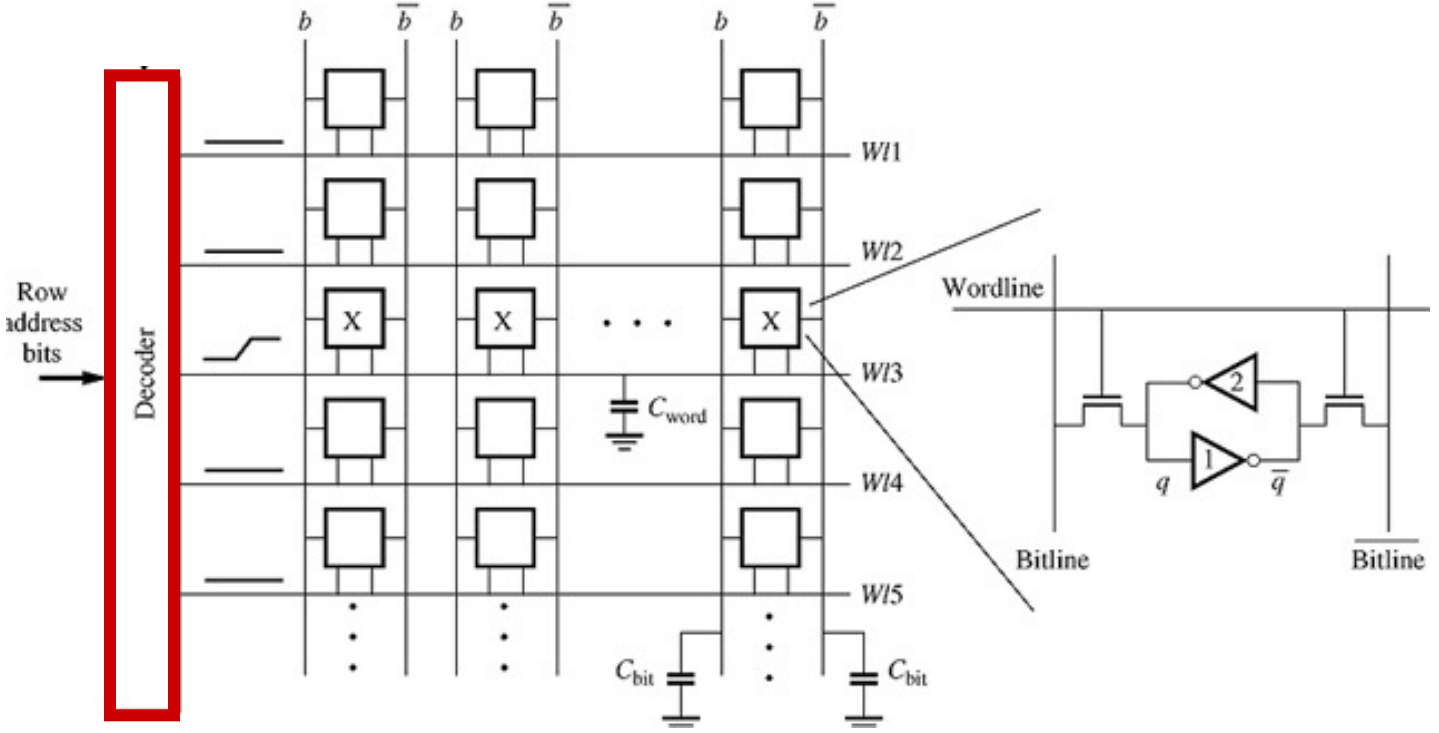
NOR Decoder

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

$$WL_{511} = \overline{\overline{A_0} + \overline{A_1} + \overline{A_2} + \overline{A_3} + \overline{A_4} + \overline{A_5} + \overline{A_6} + \overline{A_7} + \overline{A_8} + A_9}$$

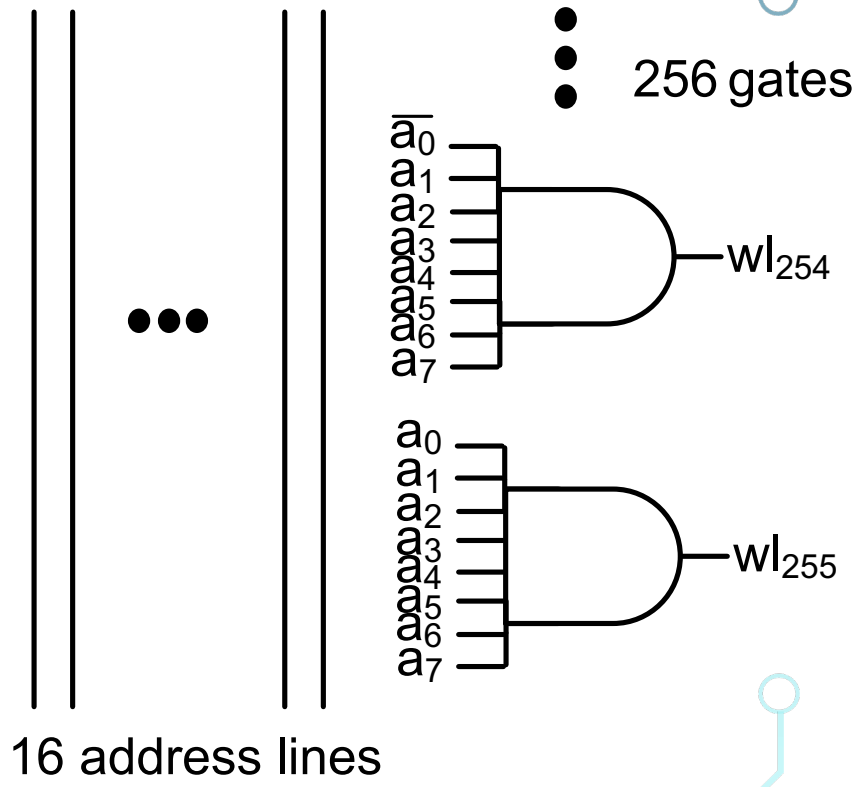
Decoder Design Example

- Look at decoder for 256x256 memory block (8KBytes)

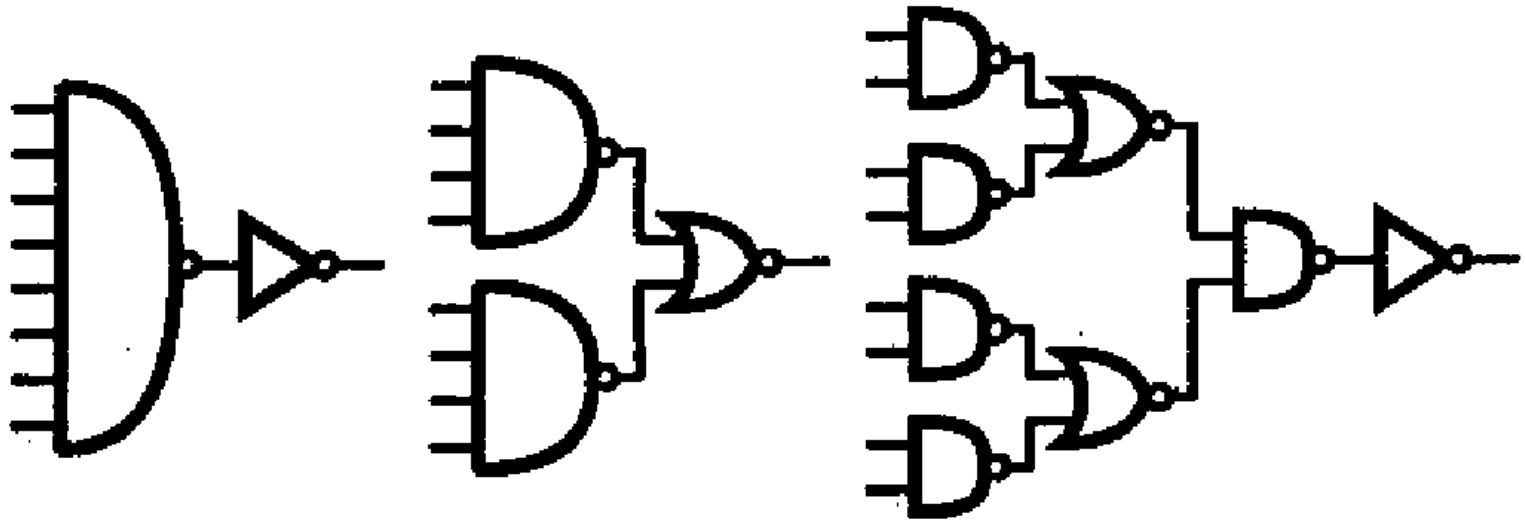


Possible Decoder

- 256 8-input AND gates
  - Each built out of a tree of NAND gates and inverters
- Need to drive a lot of capacitance (SRAM cells)
  - What's the best way to do this?



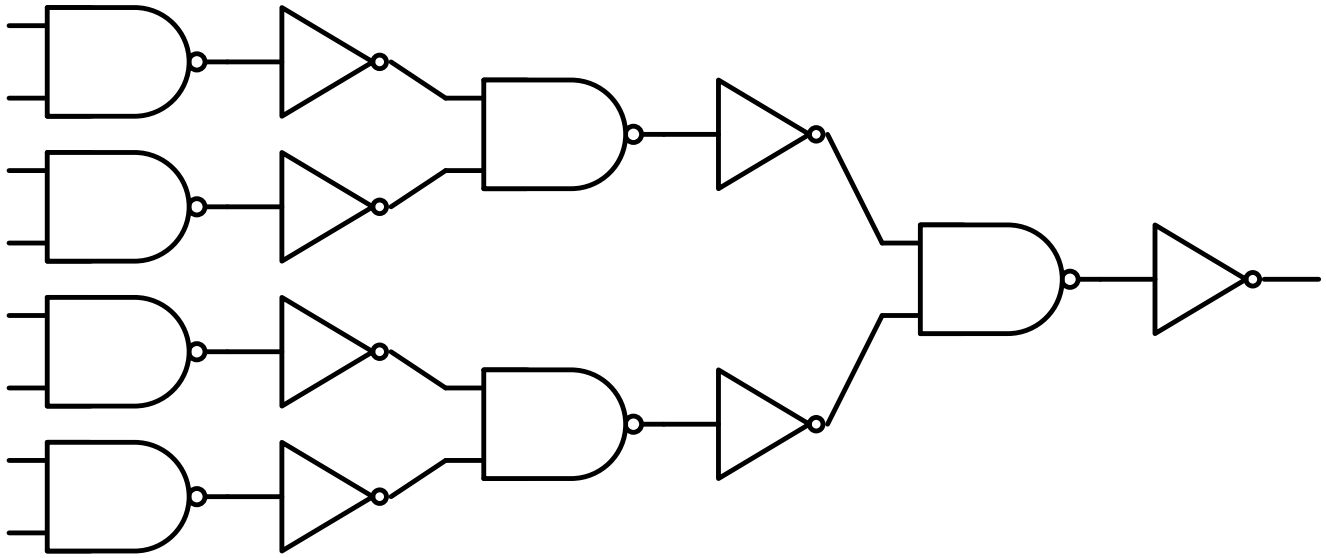
8-Input AND



LE : 9/2 1	***LE*** : 5/2 3/2	***LE*** : 3/2 3/2 3/2 1
Π***LE*** = 9/2	Π***LE*** = 15/4	Π***LE*** = 27/8
P = 8 + 1	P = 4 + 2	P = 2 + 2 + 2 + 1

8-Input AND

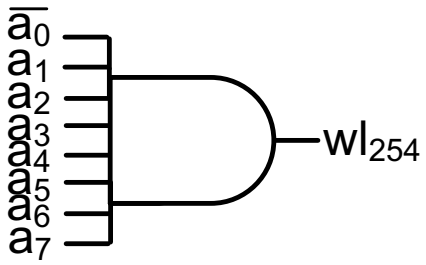
- Using 2-input NAND gates
  - 8-input gate takes 6 stages
- Total LE is $(3/2)^3 \approx 3.4$

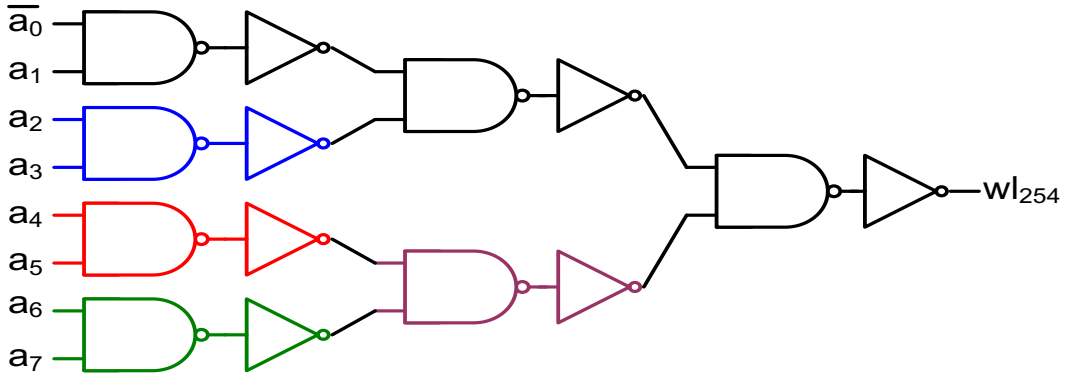


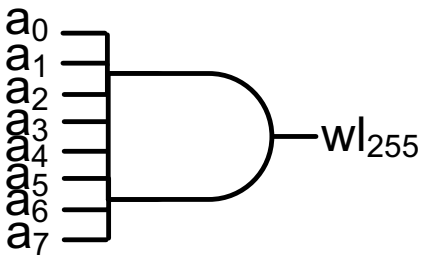
Decoder So Far

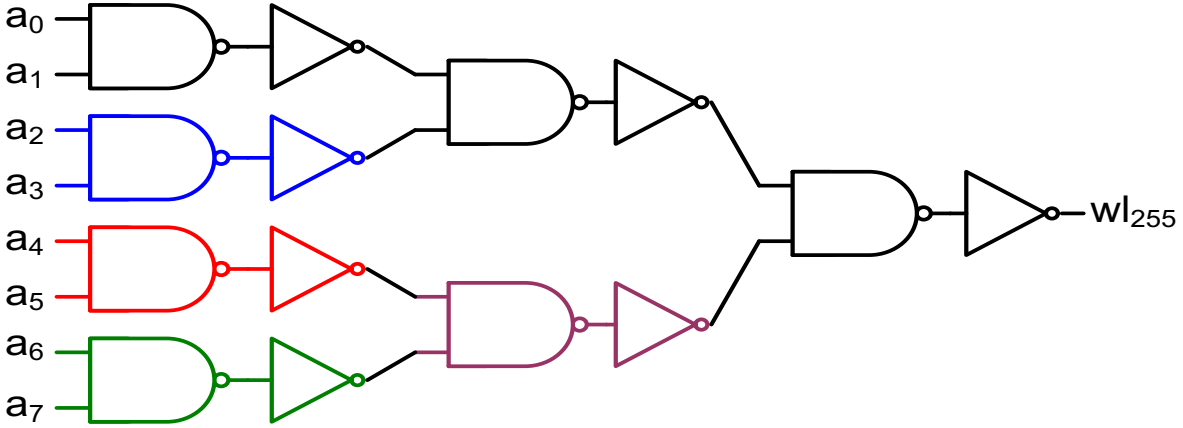
- 256 8-input AND gates
  - Each built out of tree of NAND gates and inverters
- Issue:
  - Every address line has to drive 128 gates (and wire) right away
  - Forces us to add buffers just to drive address inputs

Look Inside Each AND8 Gate





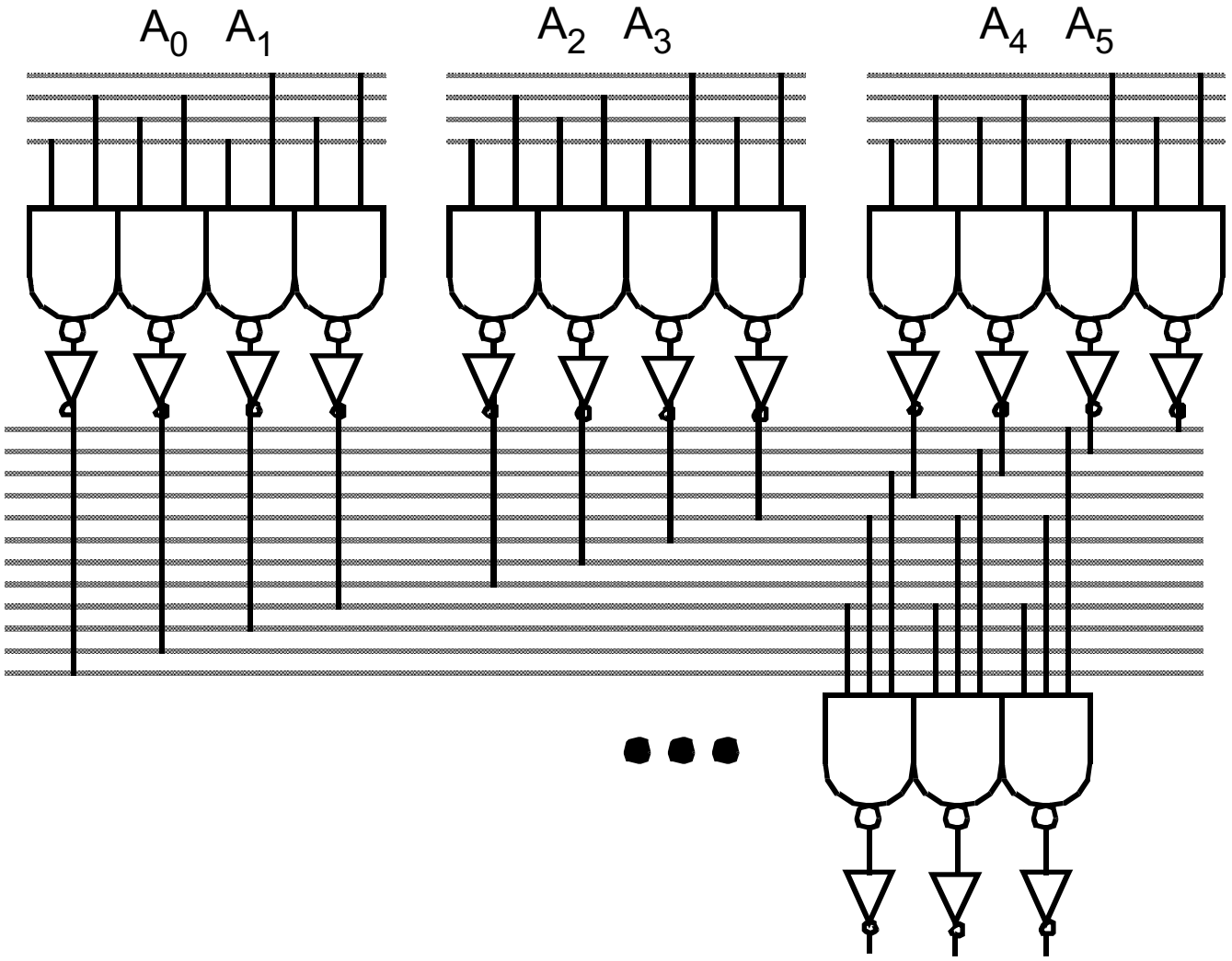




Predecoders

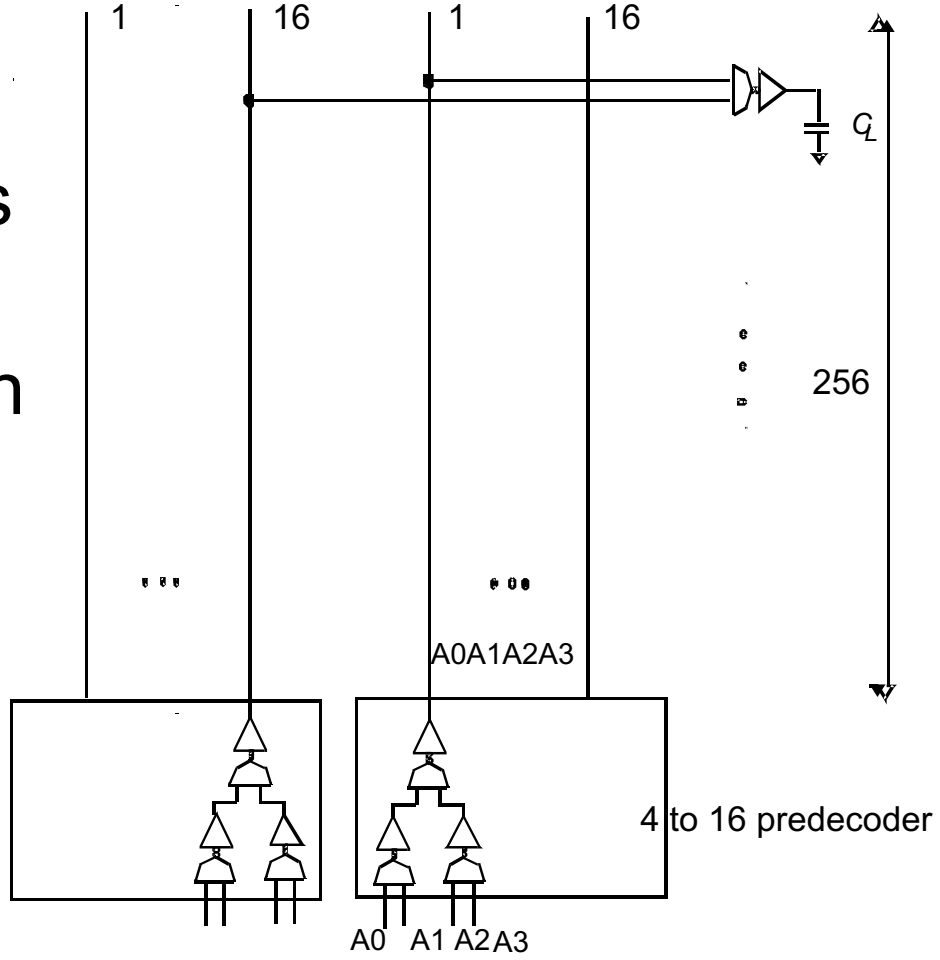
- Use a single gate for each of the shared terms
  - E.g., from A_0, $\overline{A_0}$, A_1, and $\overline{A_1}$, generate four signals: A_0A_1, $\overline{A_0}A_1$, $A_0\overline{A_1}$, $\overline{A_0}\overline{A_1}$
- In other words, we are decoding smaller groups of address bits first
  - And using the “predecoded” outputs to do the rest of the decoding

Predecoder and Decoder



Predecode Options

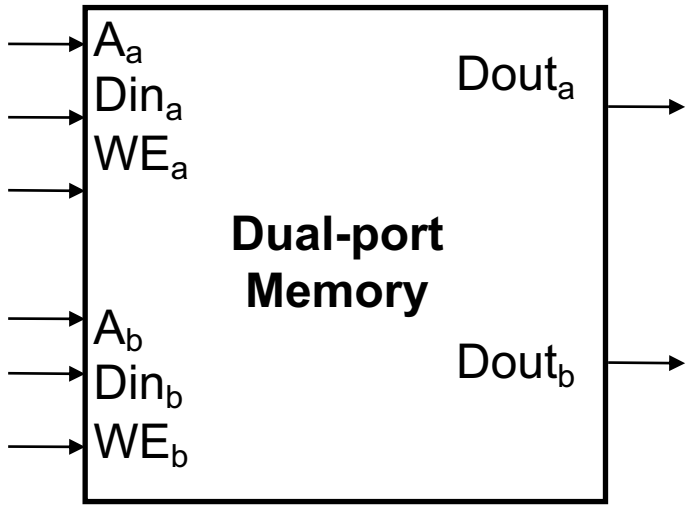
- Larger predecode usually better:
- More stages before the long wires
  - Decreases their effect on the circuit
- Fewer number of long wires switches
  - Lower power
- Easier to fit 2-input gate into cell pitch



- SRAM
  - 6-T SRAM Cell
  - Sizing SRAM Cell
- Memory Decoder
  - Overview
  - Decoder Design
  - Pre-decoder
- Multi-Ported SRAM

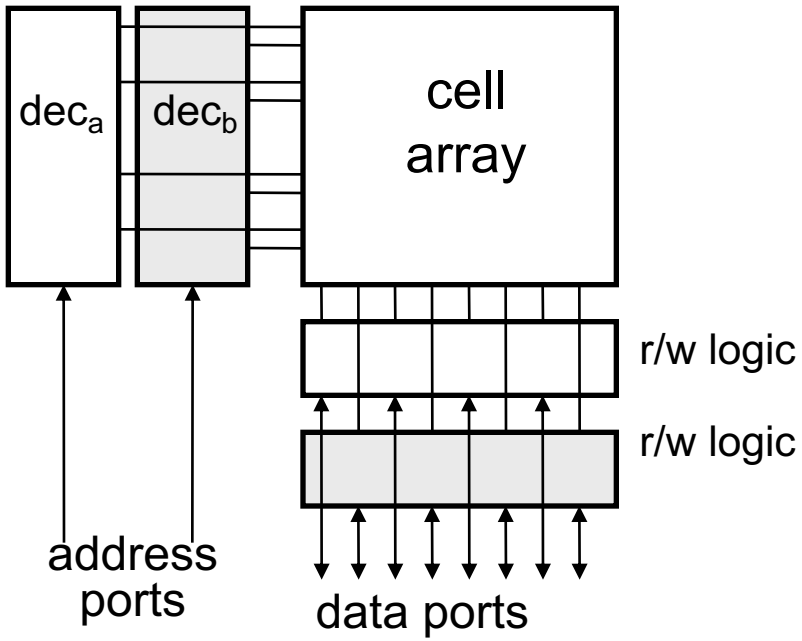
Multi-Ported Memory

- 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.
    - Single-issue pipelined CPUs usually needs 2 read ports and 1 write port (2R/1W).
    - Superscalar processors have more (e.g. 6R/3W)

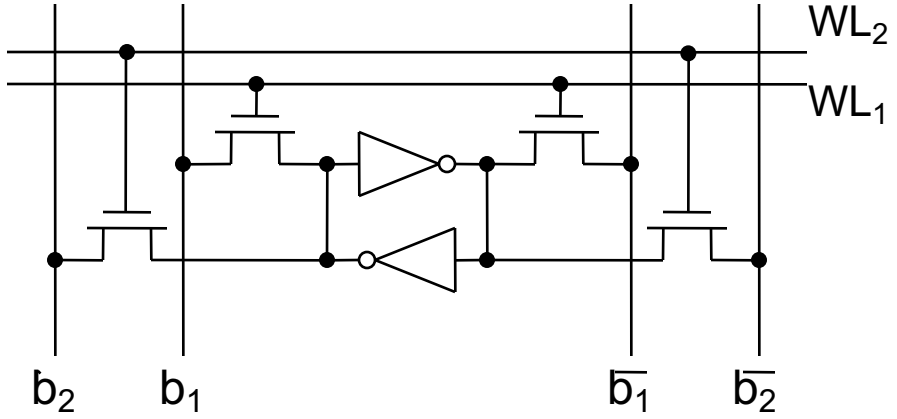


Dual-Ported Memory Internals

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



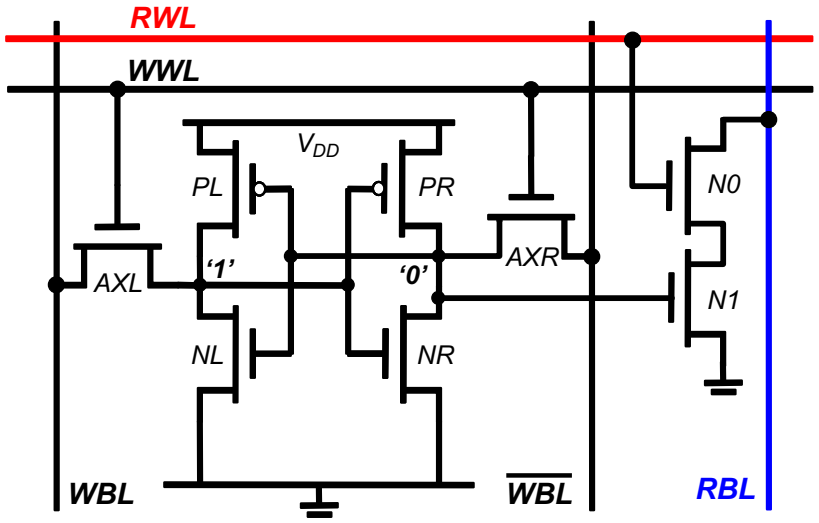
- Example cell: SRAM



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

1R/1W 8T SRAM

8-T SRAM



- Write:
  - RWL = 0
  - WWL = 1
- Read:
  - WWL = 0
  - RWL = 1, RBL = Vdd

- Dual-port read/write capability
- Single-cycle read and write, timed appropriately
- Often found in register files, first level (L1) of cache

Summary

- SRAM cells sized for stability and writeability.
- Memory decoder & pre-decoder:
  - Decoder is a series of AND gates that drive word lines.
  - One decoder per read/write port.
  - Broken into predecoder for better area and delay.
- Multi-ported SRAM
  - Improve bandwidth requirement
  - Typically lead to duplicate resources like decoders, access transistors.