Digital Integrated Circuit Design

CMPE 530/630

Memory

Dr. Cory Merkel

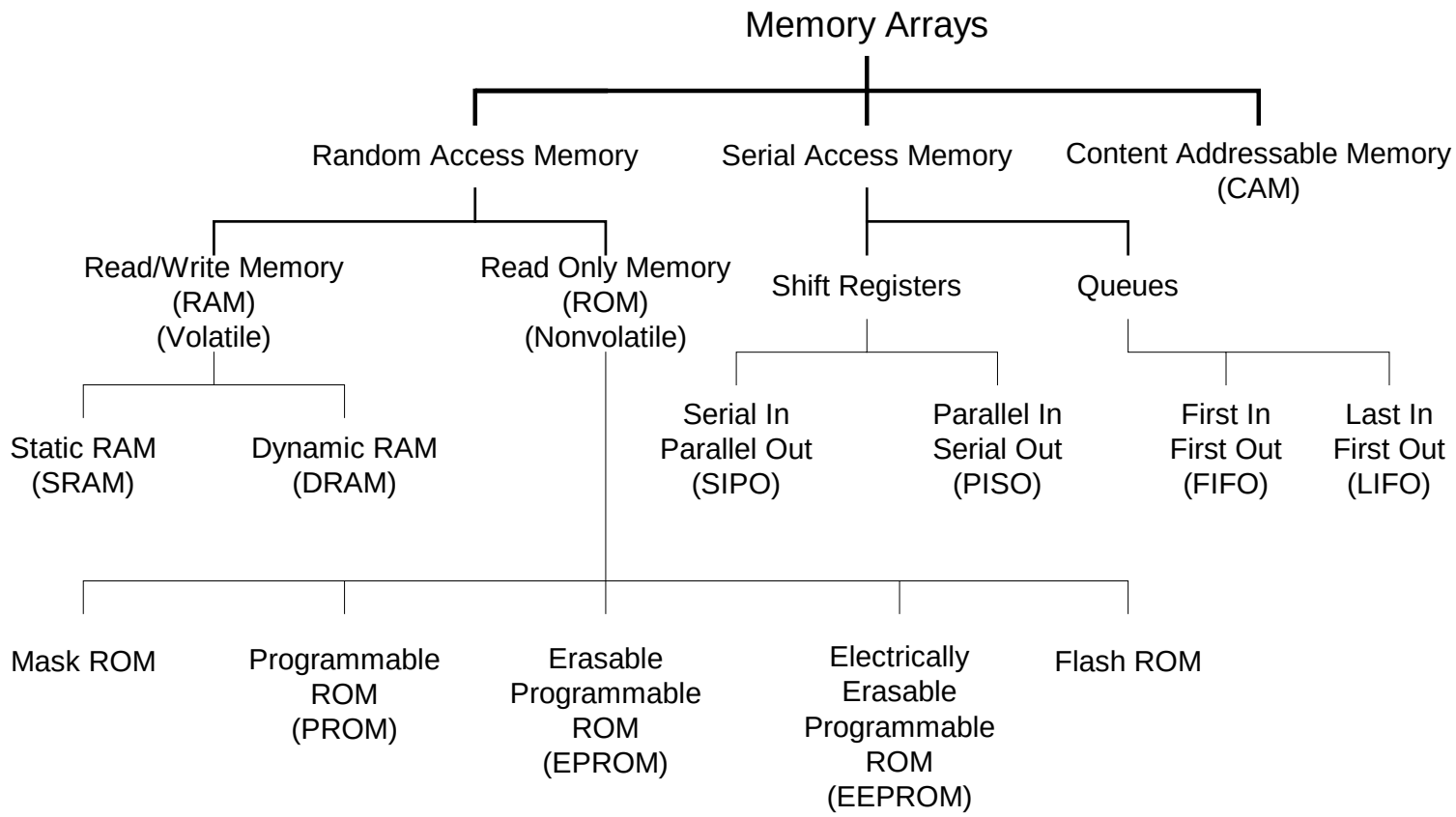
Learning Objectives

- Understand basic architectures and circuit designs for common memories.

Outline

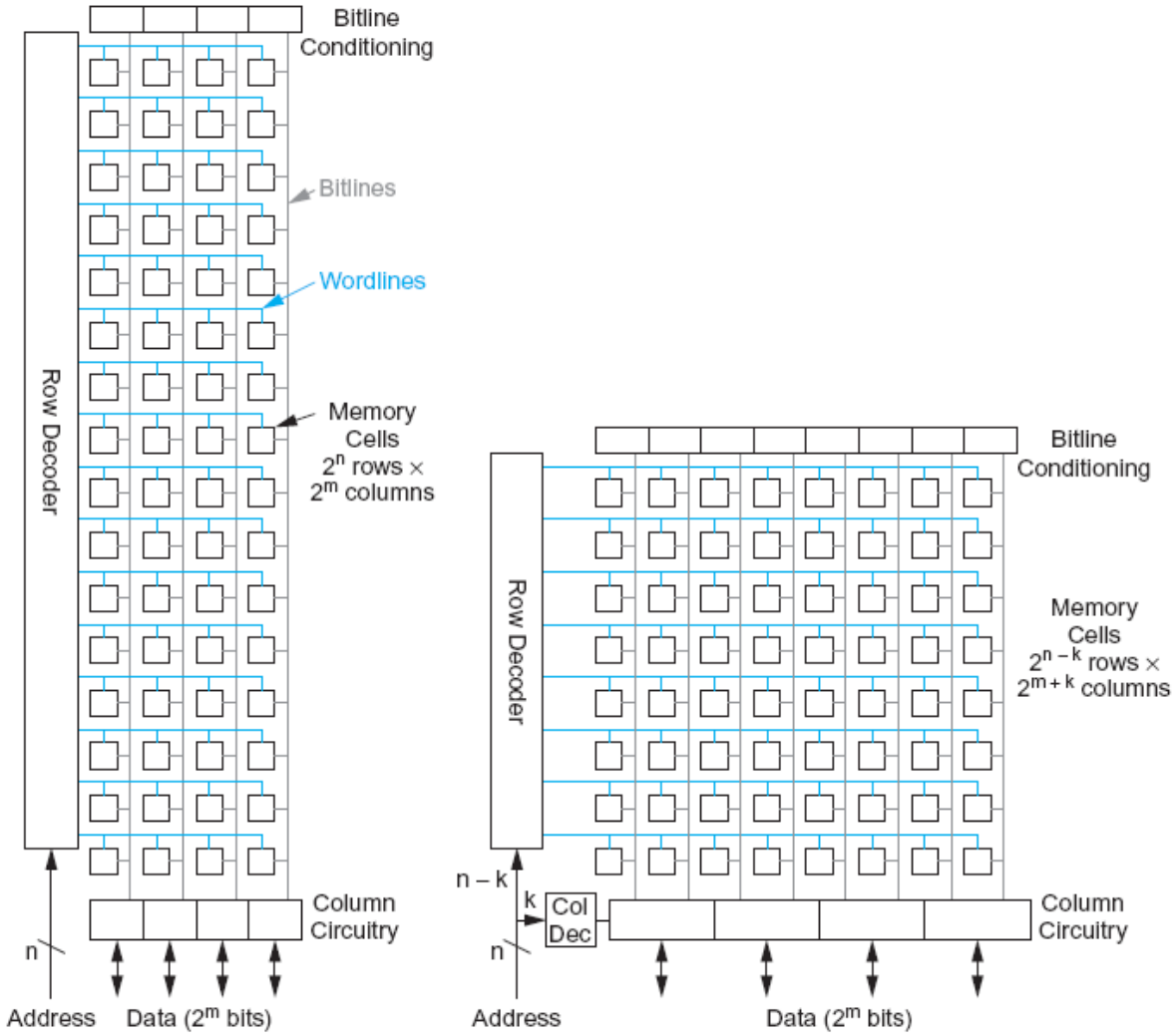
- Memory Arrays
- SRAM Architecture
  - SRAM Cell
  - Decoders
  - Column Circuitry
  - Multiple Ports
- Serial Access Memories

Memory Arrays



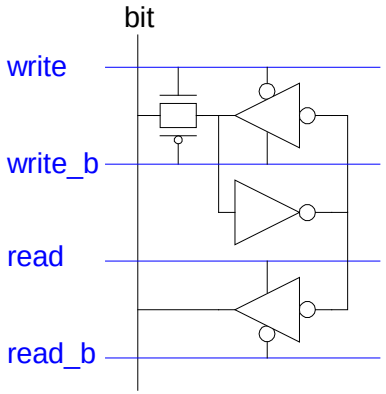
Array Architecture

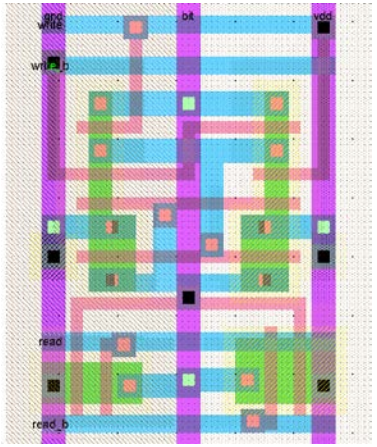
- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*
- Good regularity – easy to design
- Very high density if good cells are used



12T SRAM Cell

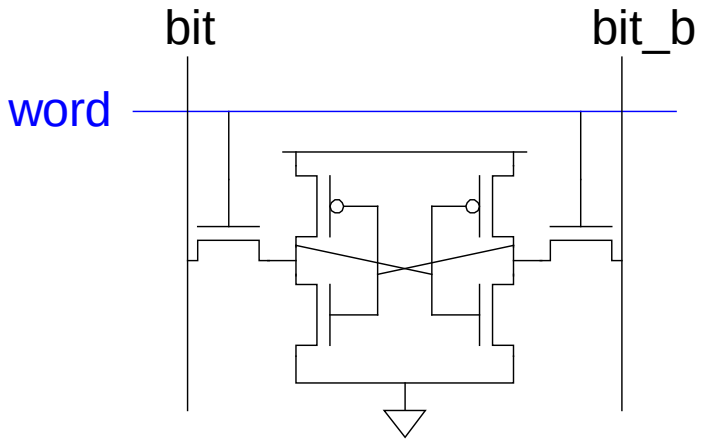
- Basic building block: SRAM Cell
  - Holds one bit of information, like a latch
  - Must be read and written
- 12-transistor (12T) SRAM cell
  - Use a simple latch connected to bitline
  - 46 x 75 λ unit cell





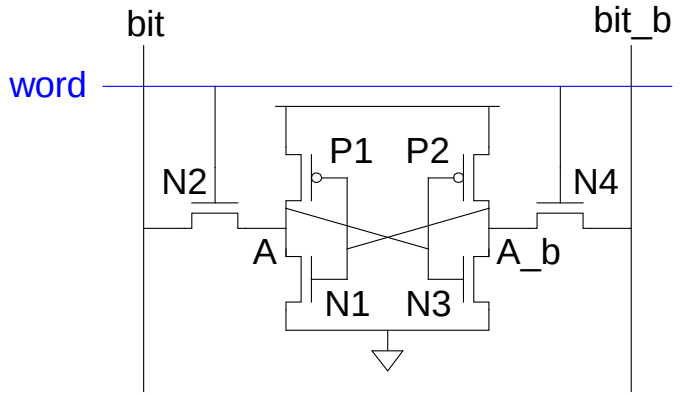
6T SRAM Cell

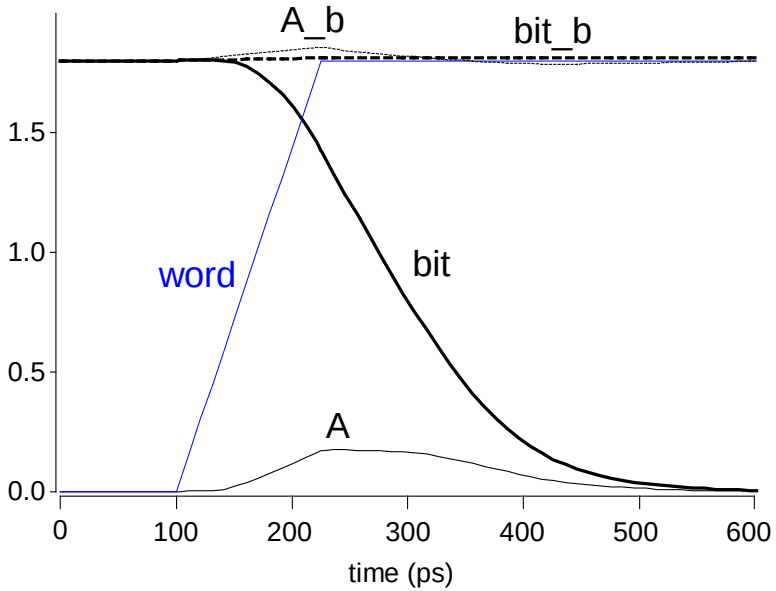
- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge bit, bit_b
  - Raise wordline
- Write:
  - Drive data onto bit, bit_b
  - Raise wordline



SRAM Read

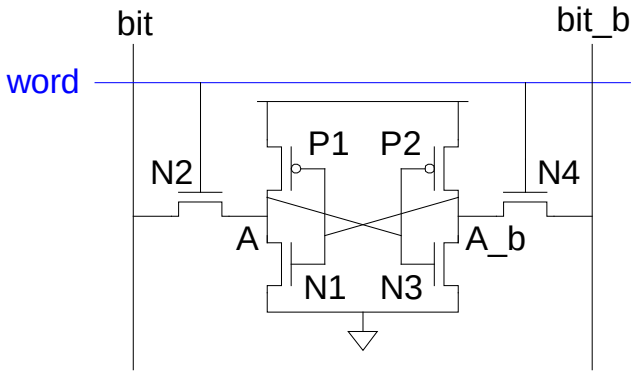
- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- *Read stability*
  - A must not flip
  - N1 >> N2

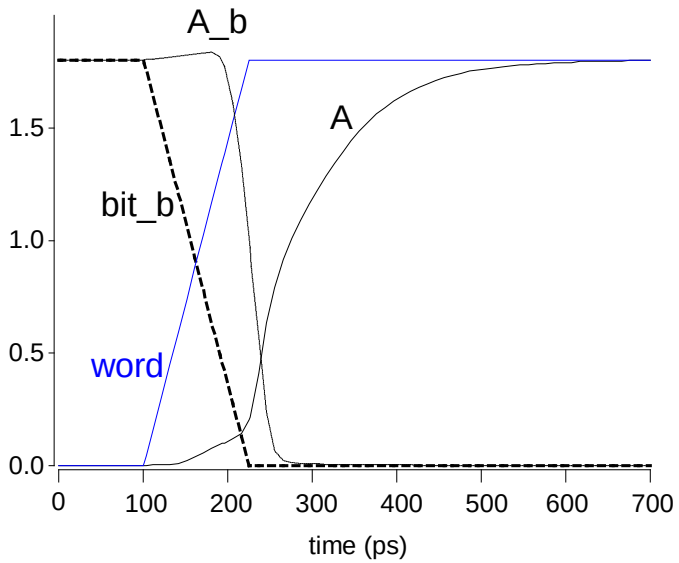




SRAM Write

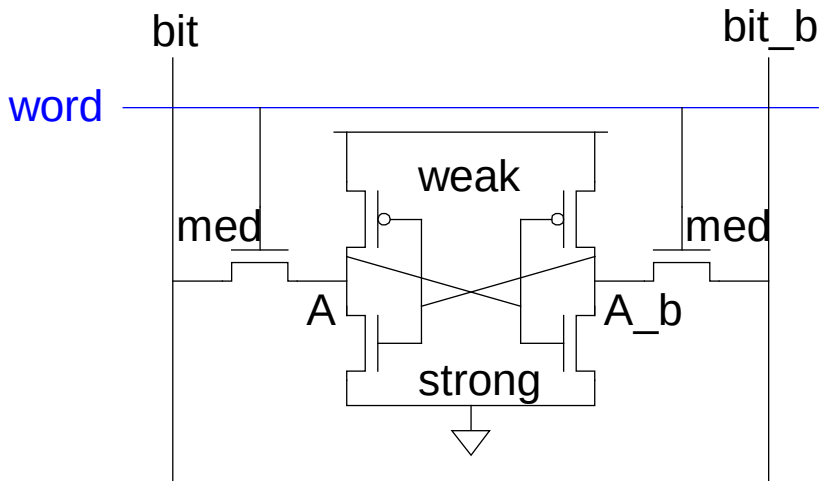
- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- Ex: A = 0, A_b = 1, bit = 1, bit_b = 0
  - Force A_b low, then A rises high
- *Writability*
  - Must overpower feedback inverter
  - N2 >> P1





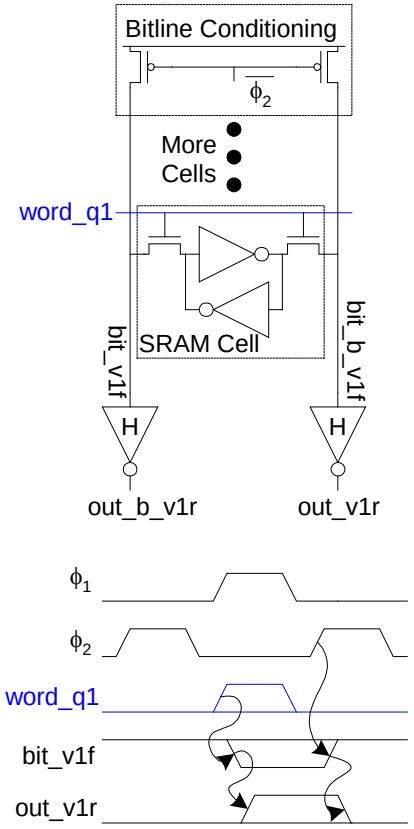
SRAM Sizing

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

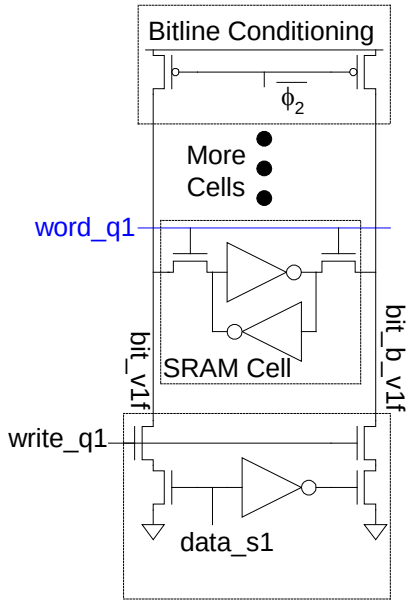


SRAM Column Example

Read

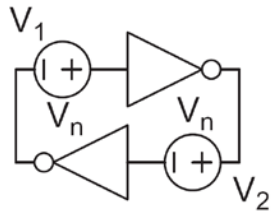


Write

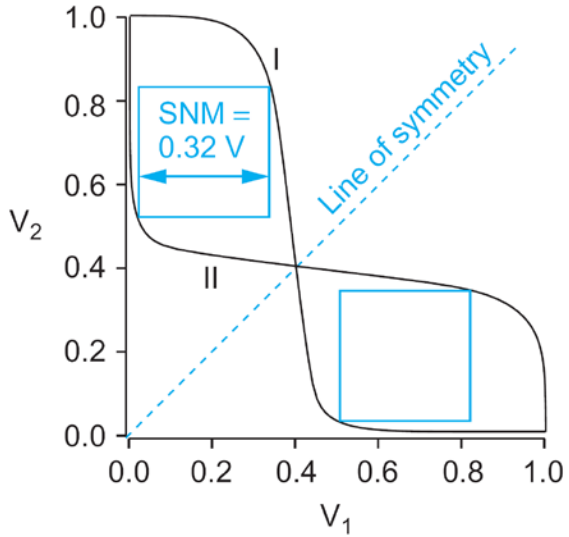


Static Noise Margins

- Static noise margins (SNM) define the amount of noise the SRAM cell can withstand
- Hold margin
  - SNM is the width of the largest square that can be fit between the two inverter voltage transfer characteristics curves (a.k.a. butterfly diagram)



- If inverters are not identical, use smaller of the two squares
- Hold margin increases with V_{dd}, V_{th}, inverter gain

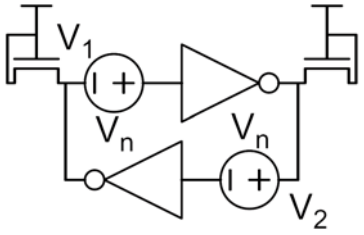


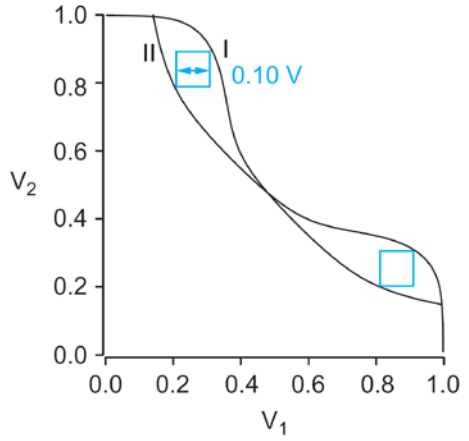
Butterfly Diagram

Static Noise Margins

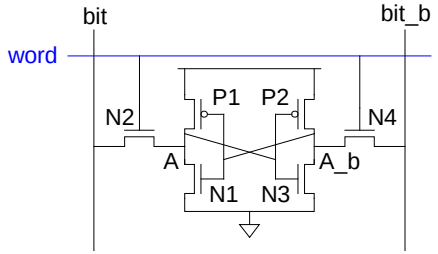
» Read margin

- Minimum voltages on butterfly diagram are >0 due to voltage on bitlines:



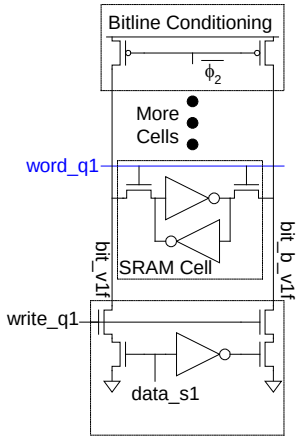


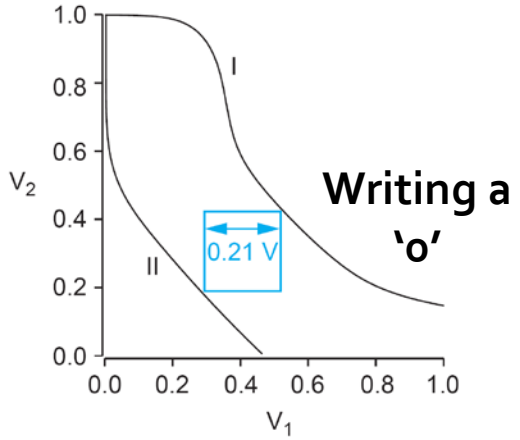
- Read margin increases with relative strength of pulldown to access transistor, or *cell ratio*: W_{N1}/W_{N2}
- Also increases with V_{dd}, V_{th}, and reducing wordline voltage relative to V_{dd}



Static Noise Margins

» Write margin
- *Smallest* square that can be drawn between curves in butterfly diagram

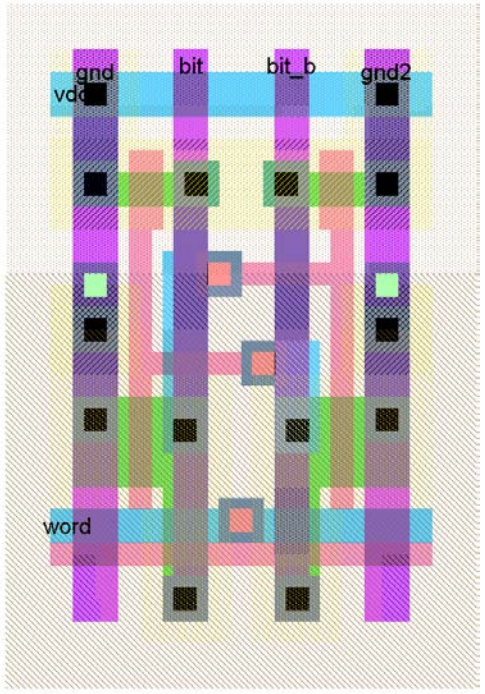


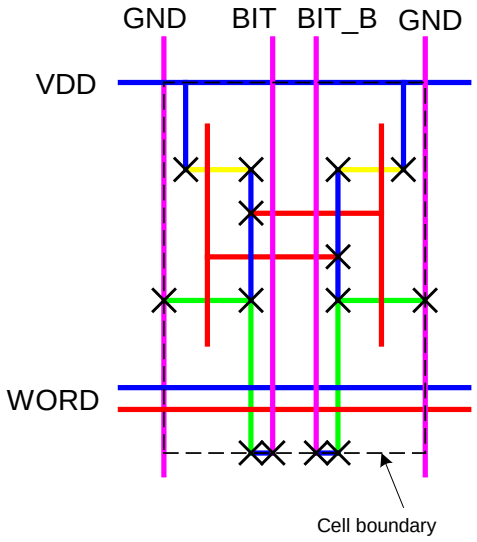


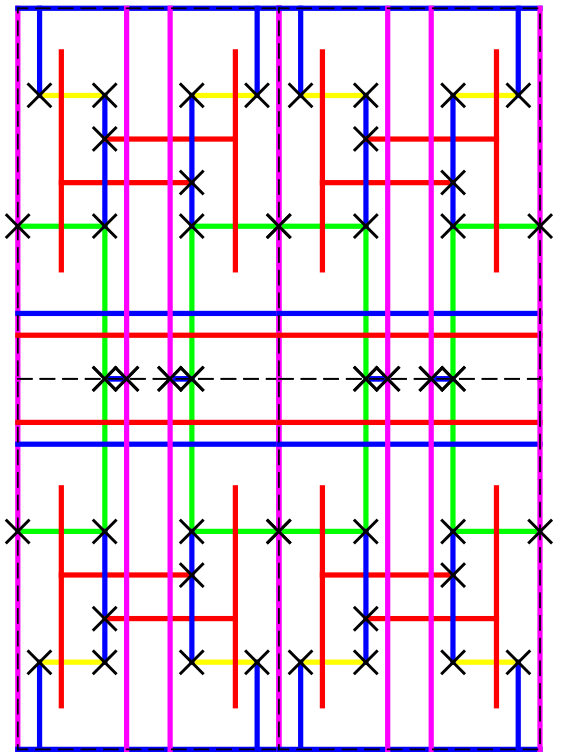
- Write margin increases with increased access transistor strength, weaker pullup transistor, increased wordline voltage
  - In contrast to read margin

SRAM Layout

- Cell size is critical: 26 x 45 λ (even smaller in industry)
- Tile cells sharing V_{DD}, GND, bitline contacts

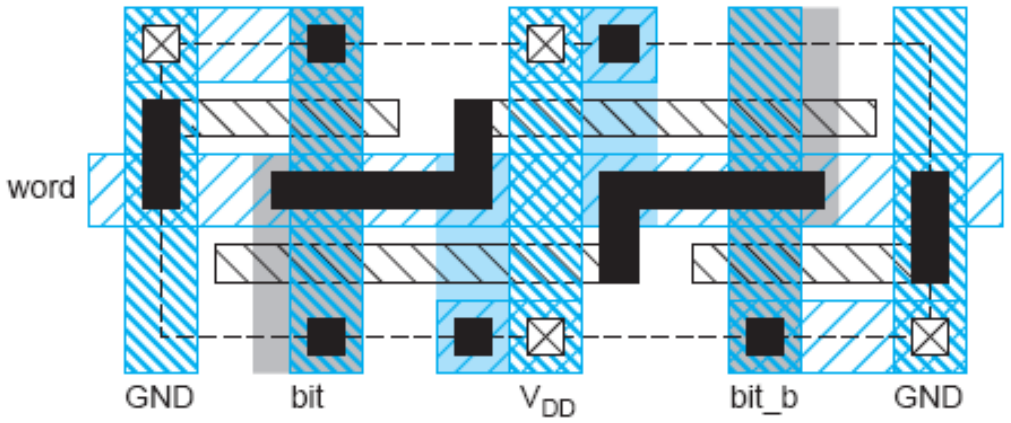






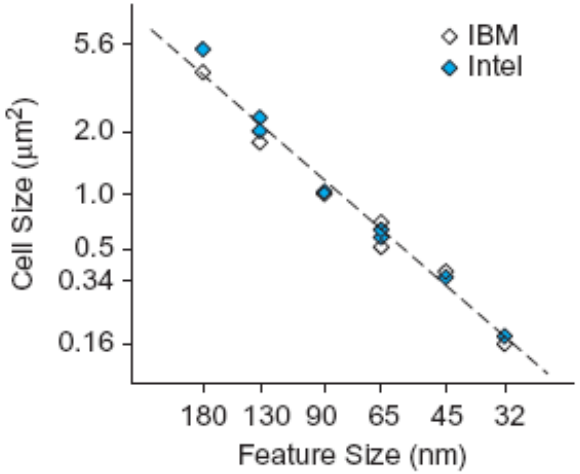
Thin Cell

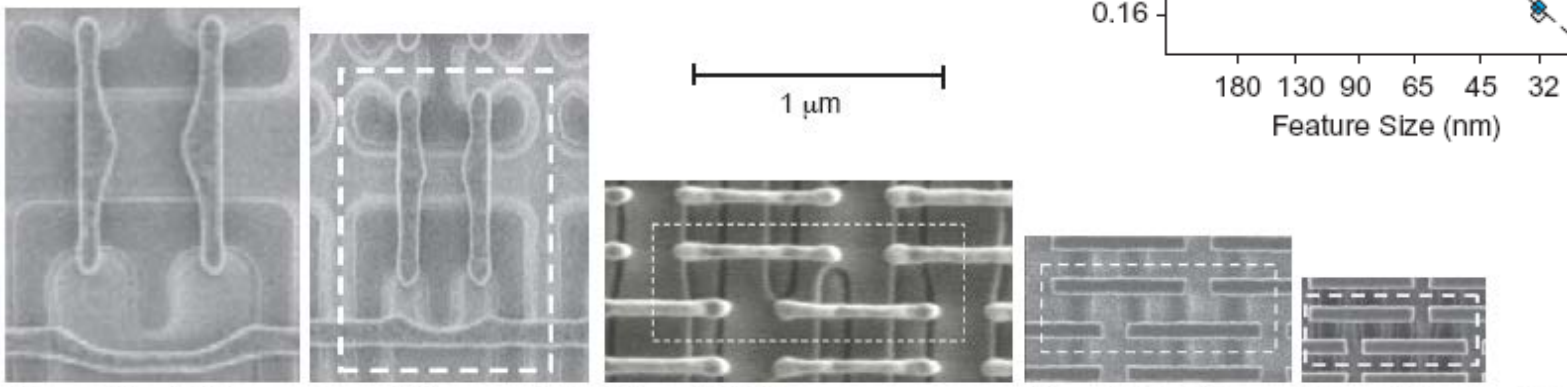
- In nanometer CMOS
  - Avoid bends in polysilicon and diffusion
  - Orient all transistors in one direction
- *Lithographically friendly* or *thin cell* layout fixes this
  - Also reduces length and capacitance of bitlines



Commercial SRAMs

- Five generations of Intel SRAM cell micrographs
  - Transition to thin cell at 65 nm
  - Steady scaling of cell area



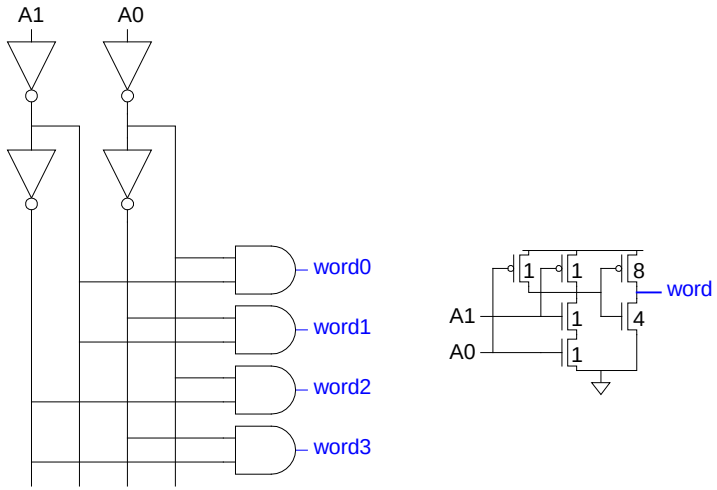


130 nm [Tyagi00] 90 nm [Thompson02] 65 nm [Bai04] 45 nm [Mistry07] 32 nm [Natarajan08]

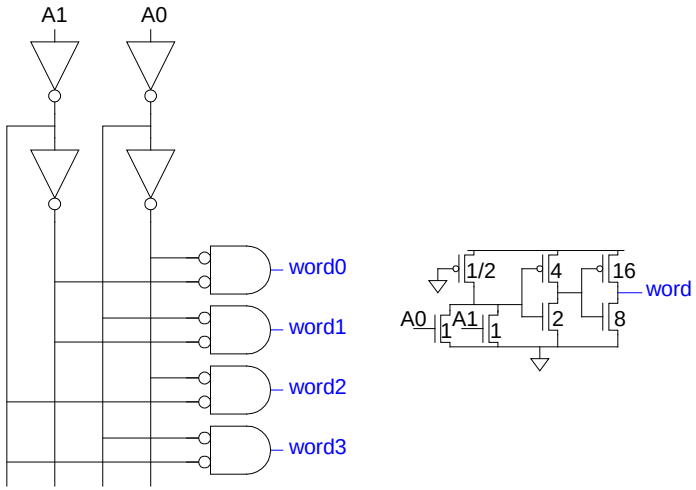
Decoders

- 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
    - Lower logical effort → Reduced delay

Static CMOS

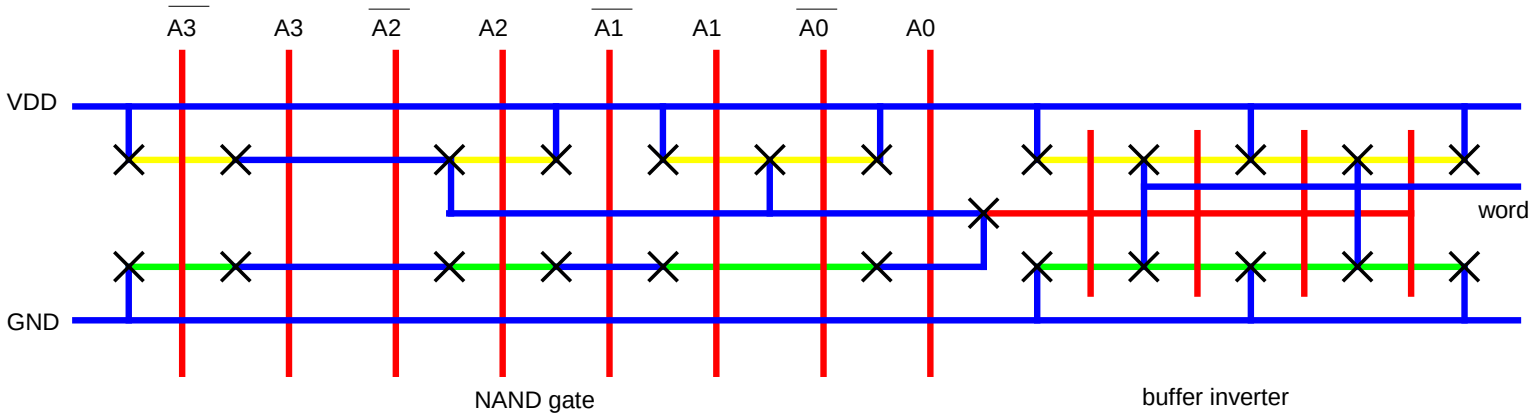


Pseudo-nMOS



Decoder Layout

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



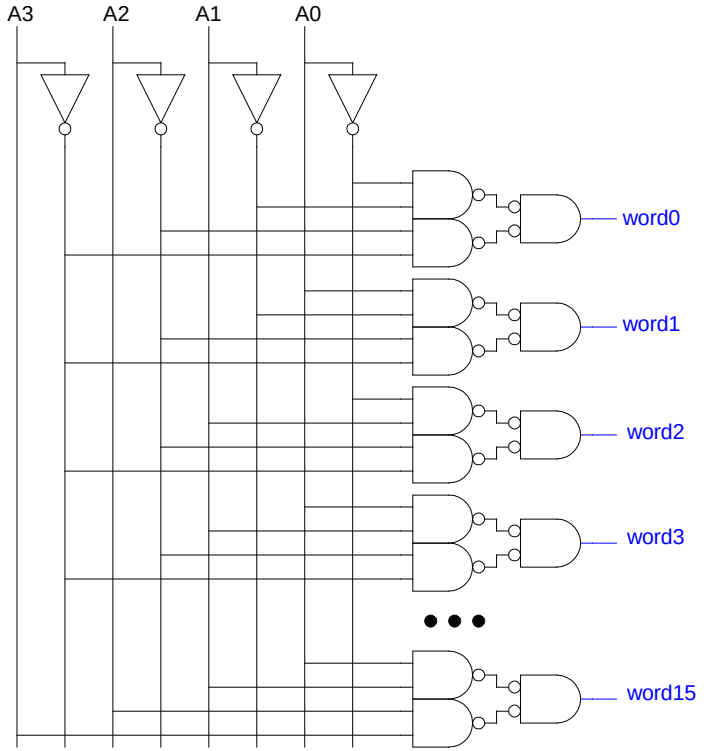
Large Decoders

- For n > 4, NAND gates become slow
  - Break large gates into multiple smaller gates
- Delay of inverter→4-input NAND

$$D = d_{INV} + d_{NAND4}$$
$$= g_{INV}h_{INV} + p_{INV} + g_{NAND4}h_{NAND4} + p_{NAND4}$$
$$1\left(\frac{6}{3} \times 2^{n-1}\right) + 1 + \left(\frac{6}{3}\right)\left(\frac{C_{word}}{6}\right) + \frac{12}{3}$$

- Delay of inverter→2-input NAND→2-input NOR

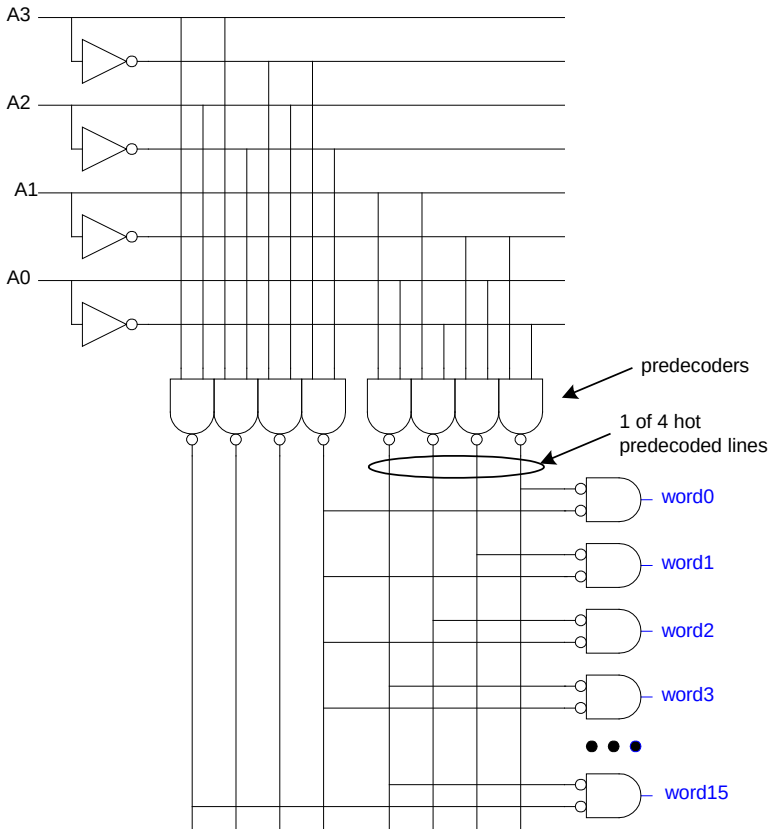
$$D = d_{INV} + d_{NAND4}$$
$$= g_{INV}h_{INV} + p_{INV} + g_{NAND2}h_{NAND2} + p_{NAND2}$$
$$+ g_{NOR2}h_{NOR2} + p_{NOR2}$$
$$1\left(\frac{4}{3} \times 2^{n-1}\right) + 1 + \left(\frac{4}{3}\right)\left(\frac{5}{4}\right) + \frac{6}{3} + \left(\frac{5}{3}\right)\left(\frac{C_{word}}{5}\right) + \frac{6}{3}$$



- For large n, 2-input NAND implementation is $\approx 1.5 \times$ faster…

Predecoding

- Many of these gates are redundant
  - Factor out common gates into predecoder
  - Saves area
  - Same path effort

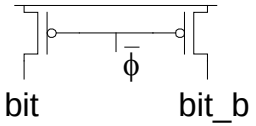


Column Circuitry

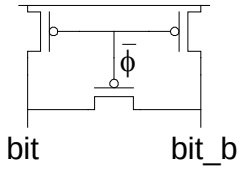
- Some circuitry is required for each column
  - Bitline conditioning
  - Sense amplifiers
  - Column multiplexing

Bitline Conditioning

» Precharge bitlines high before reads

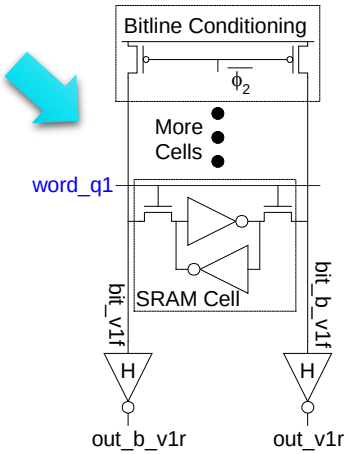


» Equalize bitlines to minimize voltage difference when using sense amplifiers



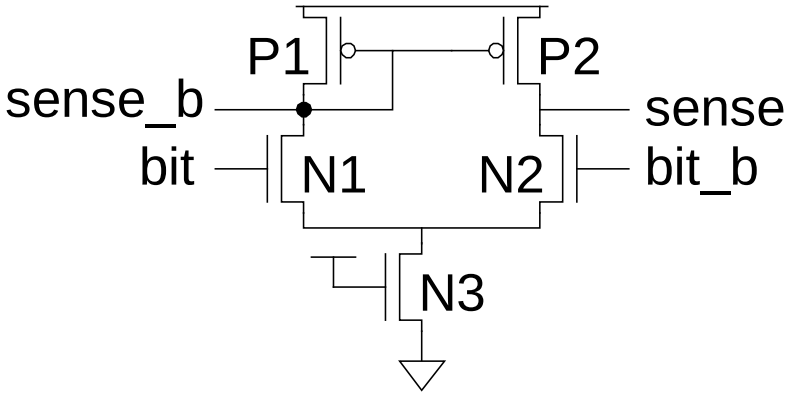
Sense Amplifiers

- One approach is to use high-skew inverters
  - **Large signal approach** → Bitlines have full voltage swing
- Bitlines have many cells attached
  - Ex: 32-kbit SRAM has 128 rows x 256 cols
  - 128 cells on each bitline
- $t_{pd} \propto (C/I)\,\Delta V$
  - Even with shared diffusion contacts, 64C of diffusion capacitance (big C)
  - Discharged slowly through small transistors (small I)
- *Sense amplifiers*
  - Triggered on small voltage swing (reduce ΔV)
  - **Small signal approach**



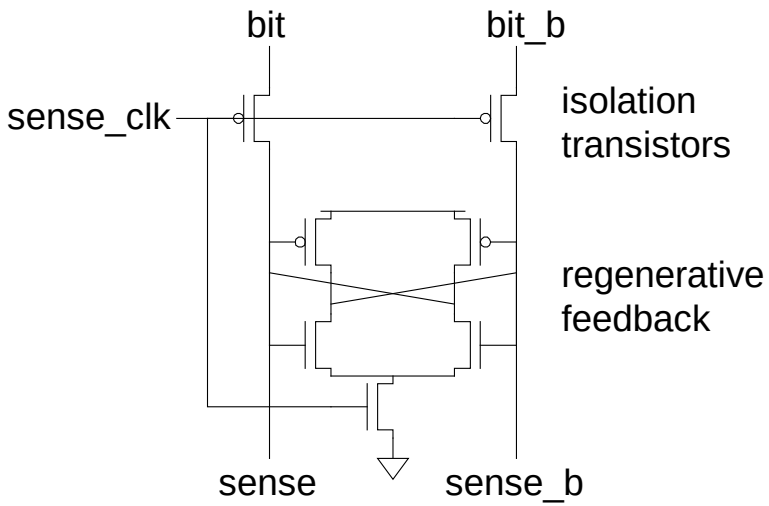
Differential Pair Amp

- Differential pair requires no clock
- But always dissipates static power



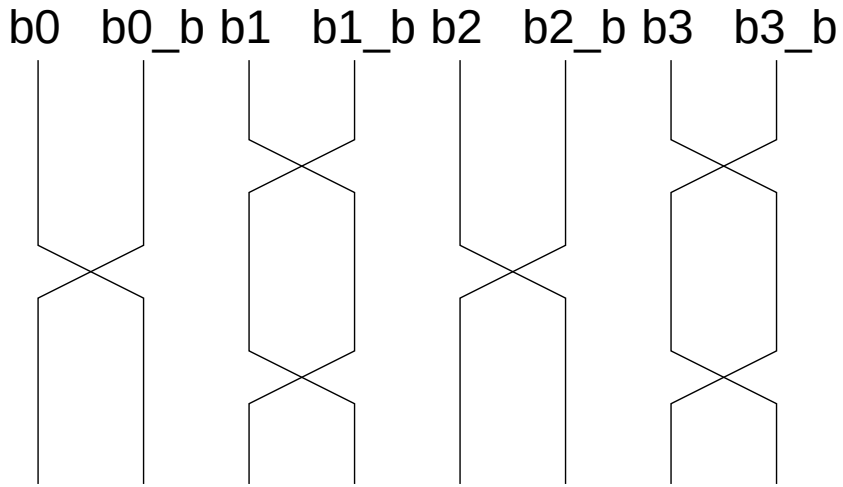
Clocked Sense Amp

- Clocked sense amp saves power
- Requires sense_clk after enough bitline swing
- Isolation transistors cut off large bitline capacitance



Twisted Bitlines

» Sense amplifiers also amplify noise
  - Coupling noise is severe in modern processes
  - Try to couple equally onto bit and bit_b
  - Done by *twisting* bitlines

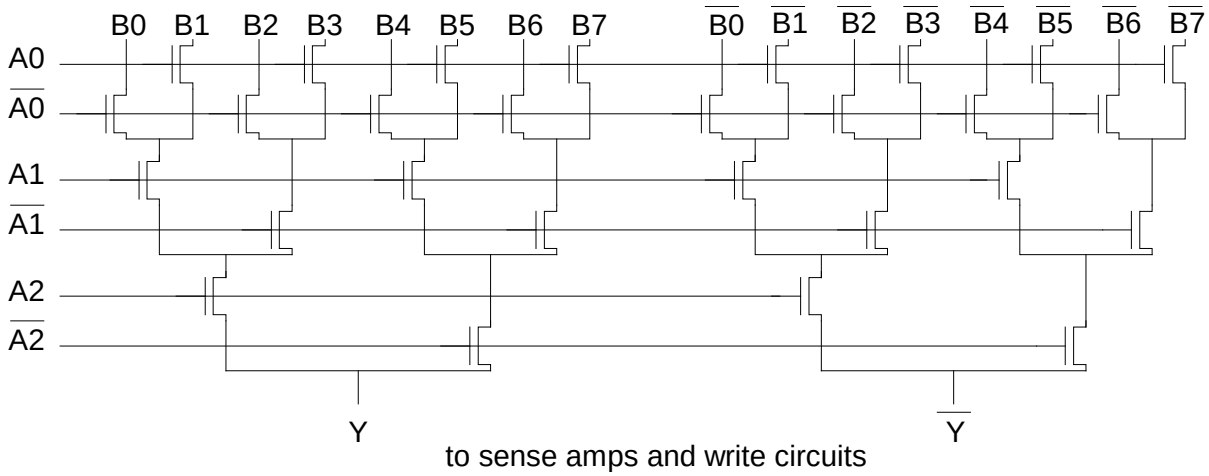


Column Multiplexing

- Recall that array may be folded for good aspect ratio
- Ex: 2 kword x 16 folded into 256 rows x 128 columns
  - Must select 16 output bits from the 128 columns
  - Requires 16 8:1 column multiplexers

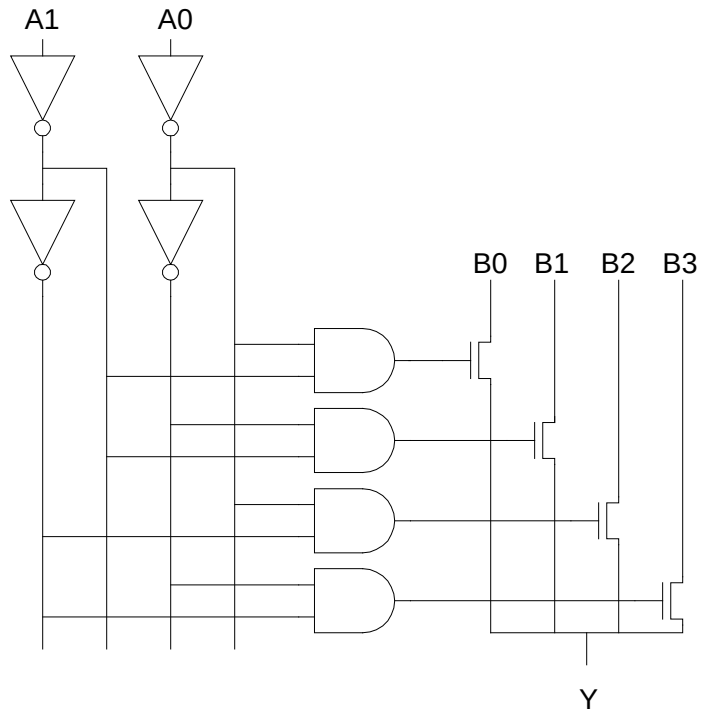
Tree Decoder Mux

- Column mux can use pass transistors
  - Use nMOS only, precharge outputs
- One design is to use k series transistors for 2^k:1 mux
  - No external decoder logic needed

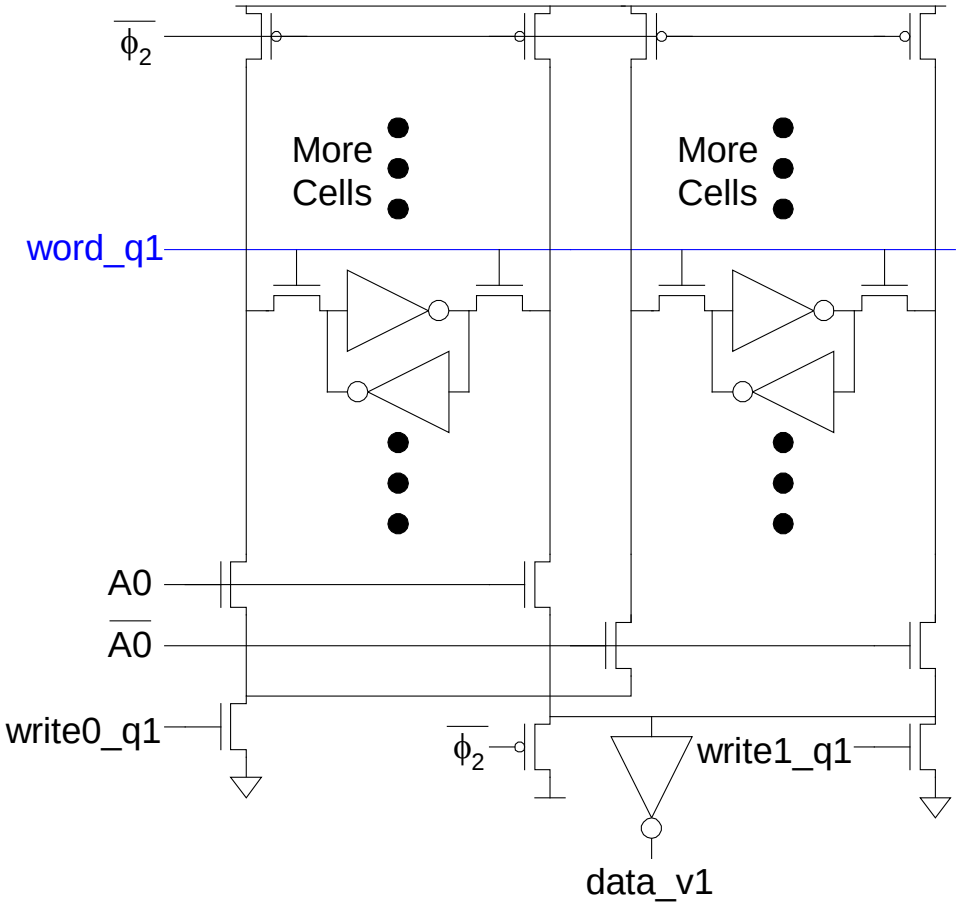


Single Pass-Gate Mux

» Or eliminate series transistors with separate decoder



Ex: 2-way Muxed SRAM

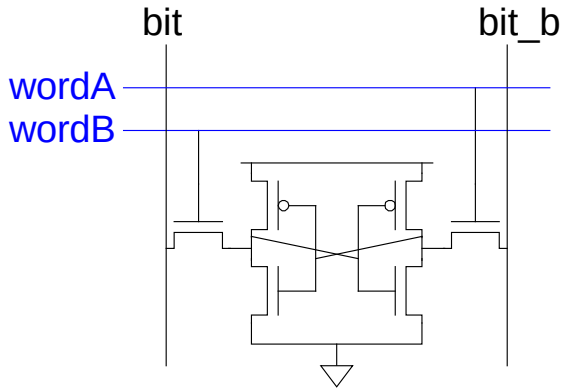


Multiple Ports

- We have considered single-ported SRAM
  - One read or one write on each cycle
- *Multiported* SRAM are needed for register files
- Examples:
  - Multicycle MIPS must read two sources or write a result on some cycles
  - Pipelined MIPS must read two sources and write a third result each cycle
  - Superscalar MIPS must read and write many sources and results each cycle

Dual-Ported SRAM

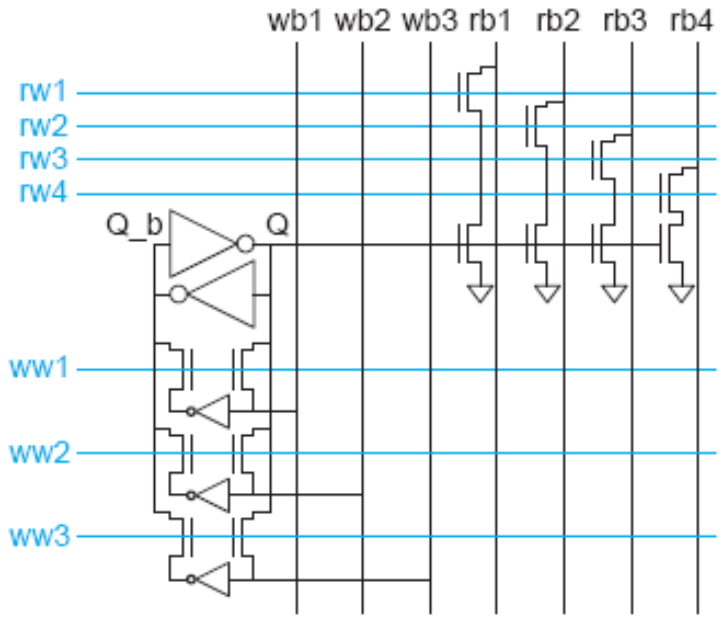
- Simple dual-ported SRAM
  - Two independent single-ended reads
  - Or one differential write



- Do two reads and one write by time multiplexing
  - Read during ph1, write during ph2

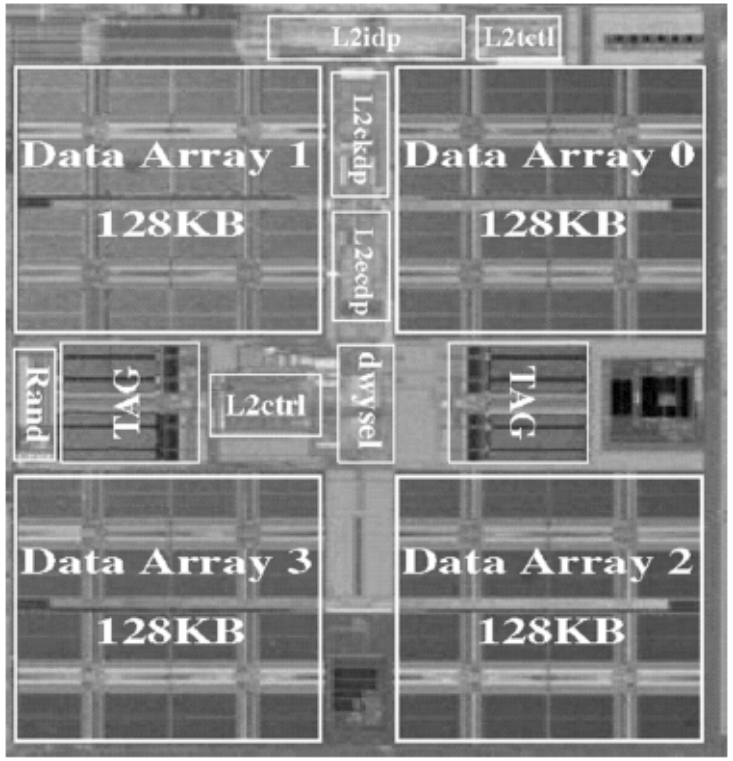
Multi-Ported SRAM

- Adding more access transistors hurts read stability
- Multiported SRAM isolates reads from state node
- Single-ended bitlines save area



Large SRAMs

- Large SRAMs are split into subarrays for speed
- Ex: UltraSparc 512KB cache
  - 4 128 KB subarrays
  - Each have 16 8KB banks
  - 256 rows x 256 cols / bank
  - 60% subarray area efficiency
  - Also space for tags & control



[Shin05]