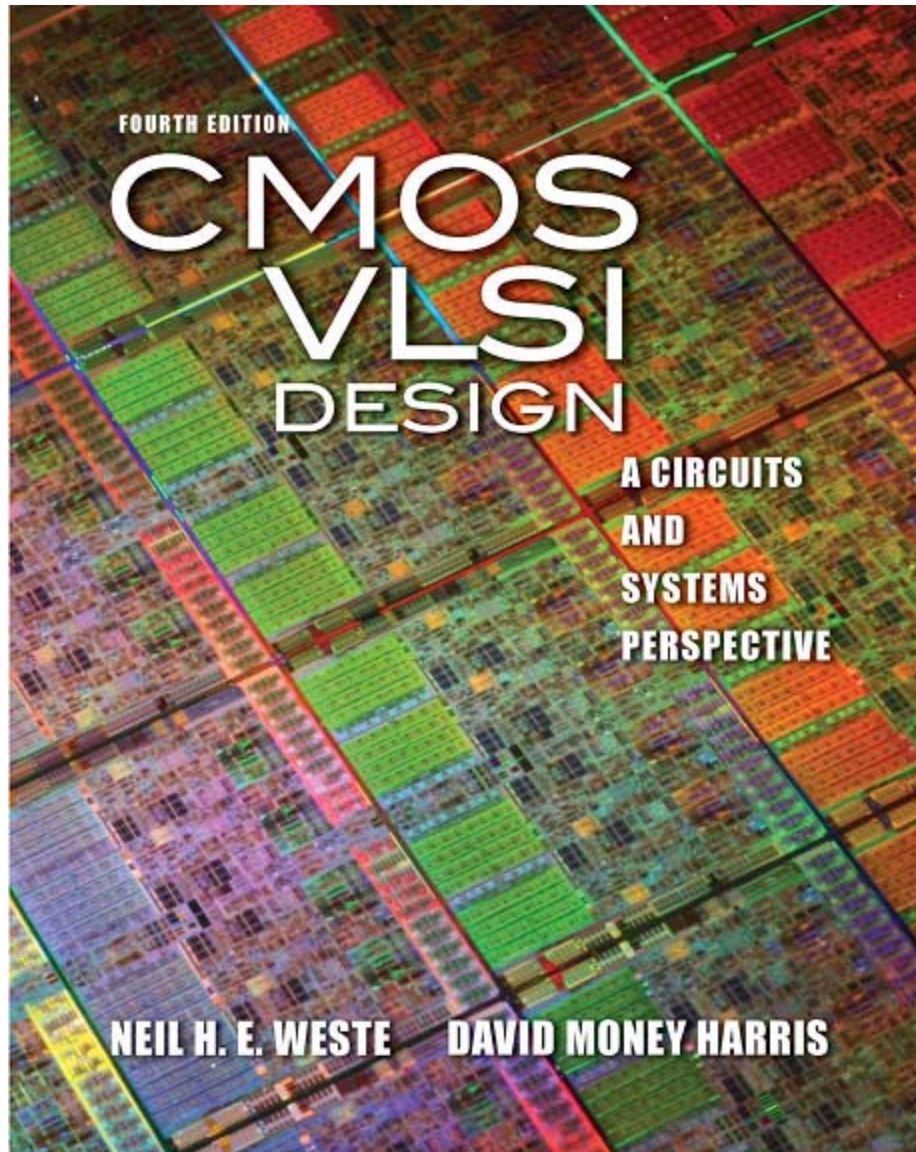
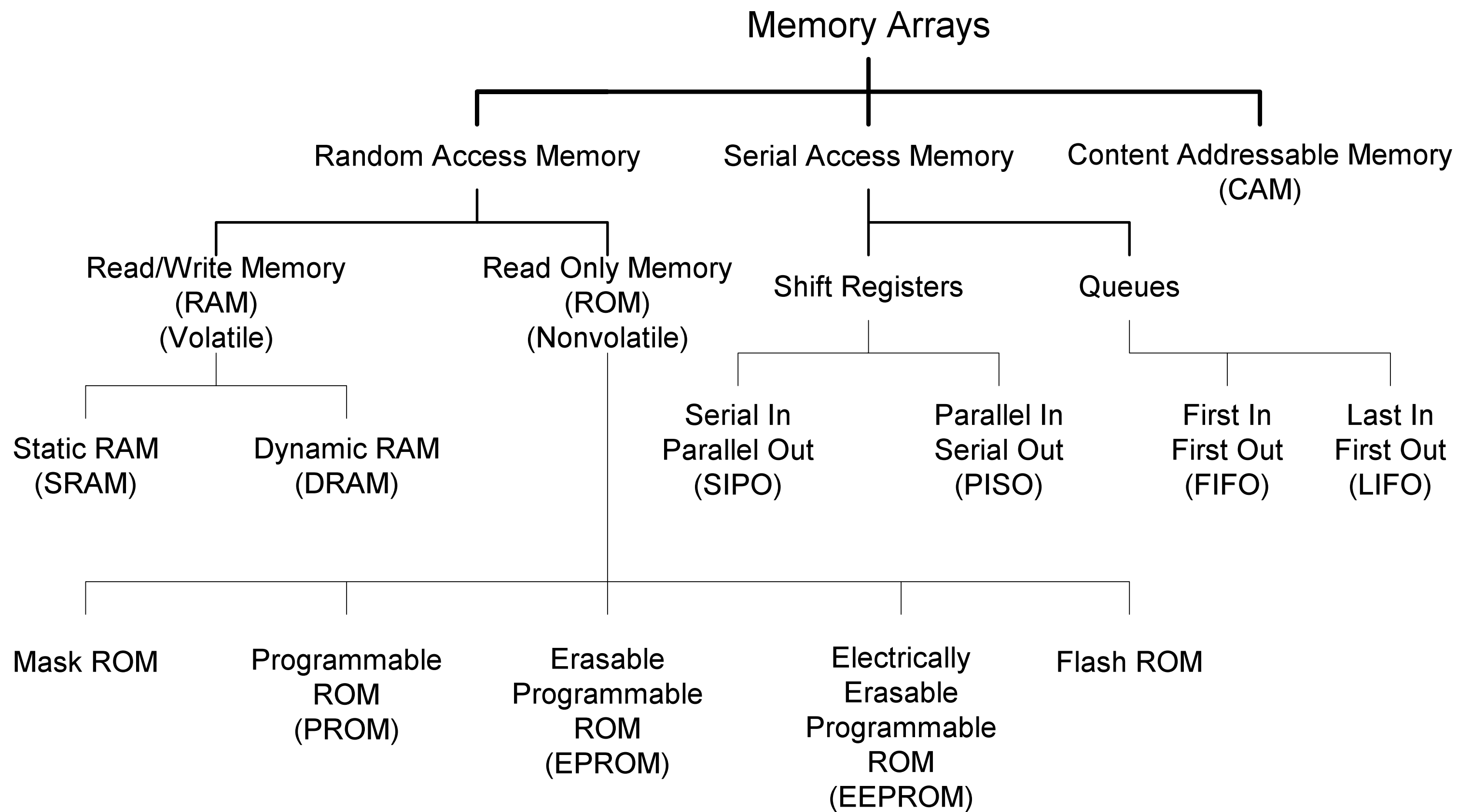


Lecture 19: SRAM

Outline

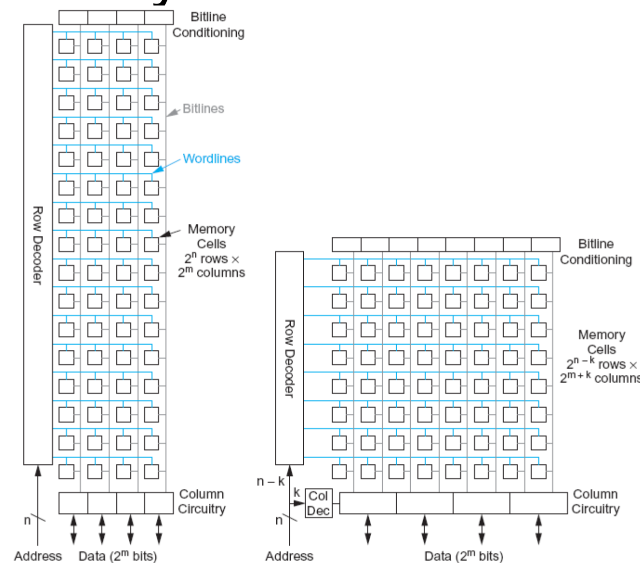
- Memory Arrays
- SRAM Architecture
  - SRAM Cell
  - Decoders
  - Column Circuitry

Memory Arrays



Array Architecture

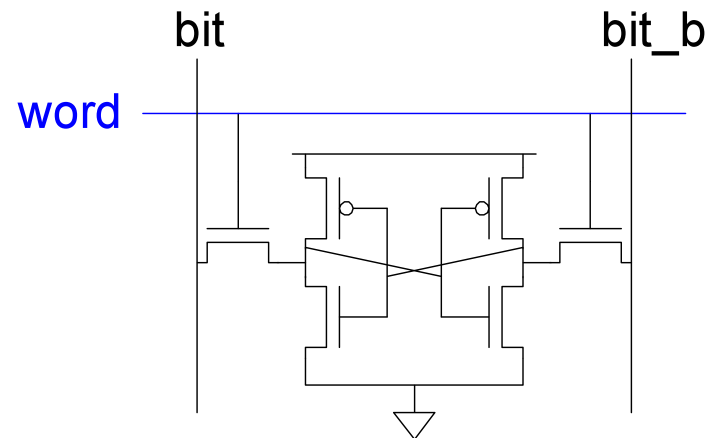
- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*



- Good regularity – easy to design
- Very high density if good cells are used

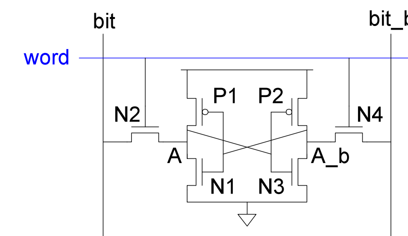
6T SRAM Cell

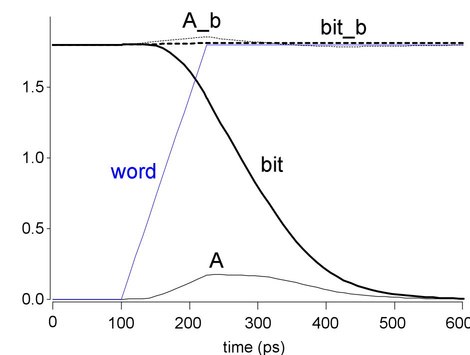
- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge bit, bit_b
  - Raise wordline
- Write:
  - Drive data onto bit, bit_b
  - Raise wordline



SRAM Read

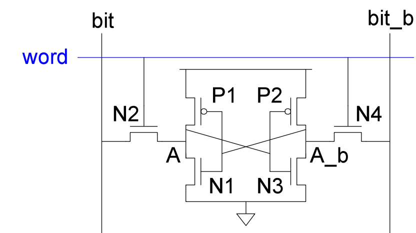
- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- *Read stability*
  - A must not flip
  - N1 >> N2

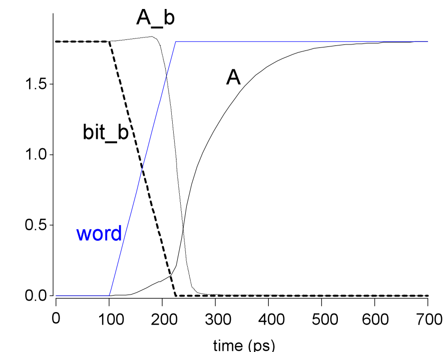




SRAM Write

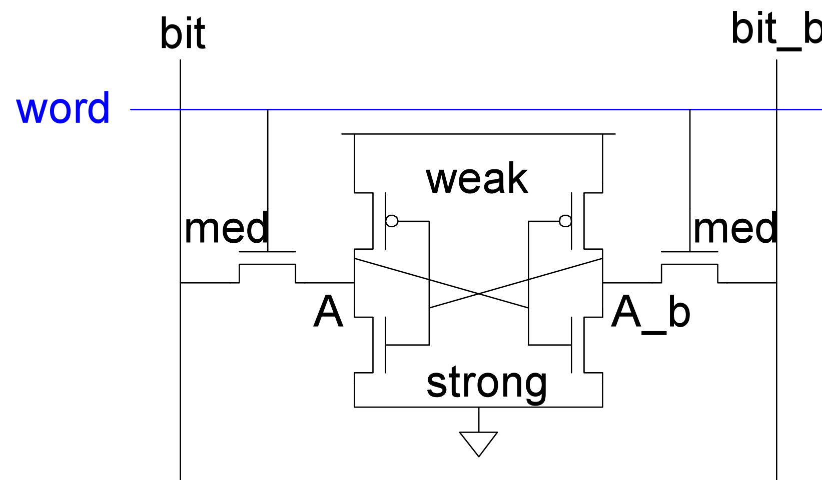
- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- Ex: A = 0, A_b = 1, bit = 1, bit_b = 0
  - Force A_b low, then A rises high
- *Writability*
  - Must overpower feedback inverter
  - N2 >> P1





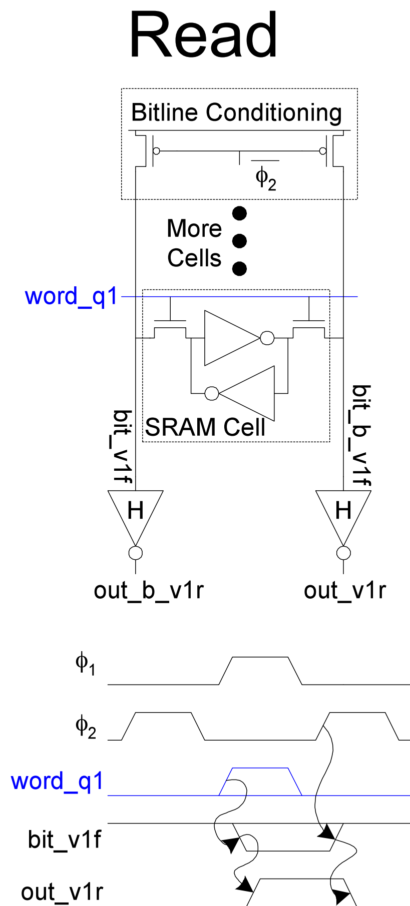
SRAM Sizing

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

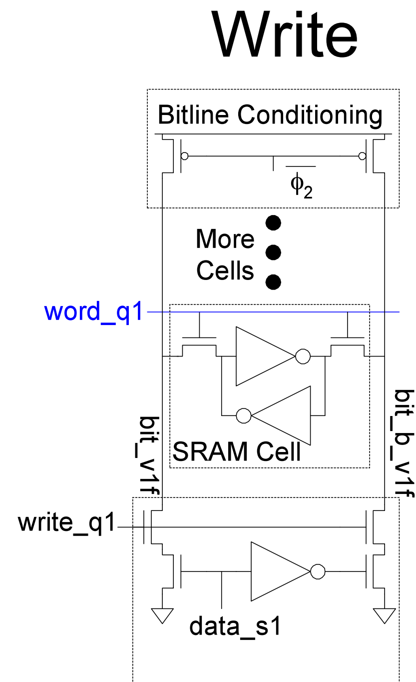


SRAM Column Example

Read

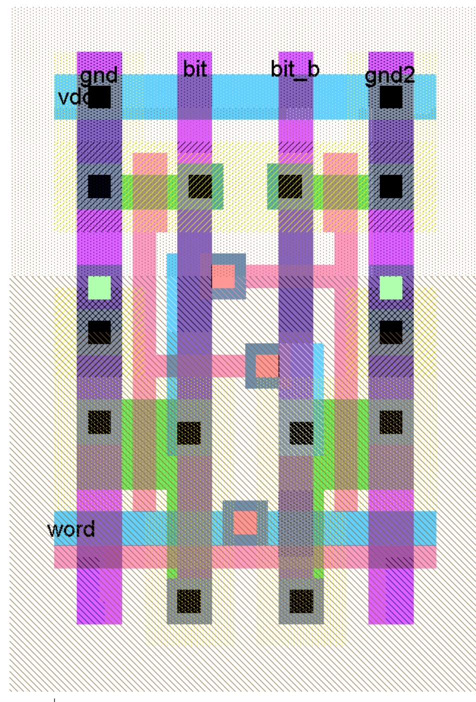


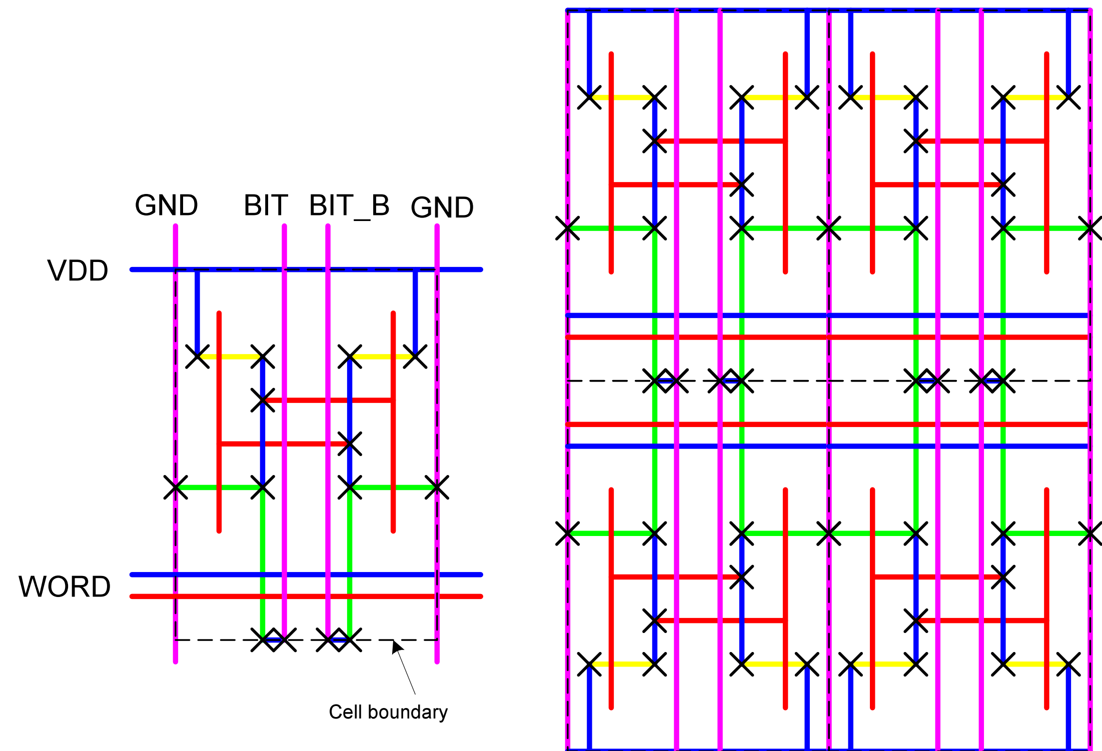
Write



SRAM Layout

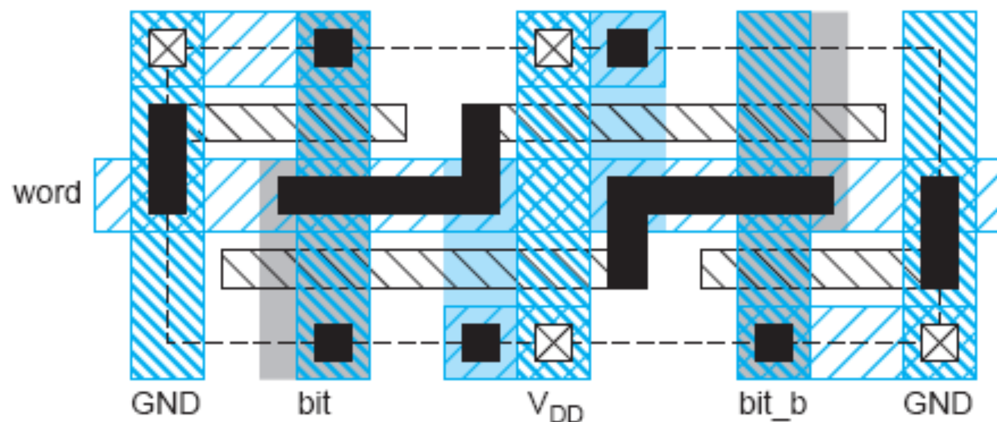
- Cell size is critical: 26 x 45 λ (even smaller in industry)
- Tile cells sharing V_{DD}, GND, bitline contacts





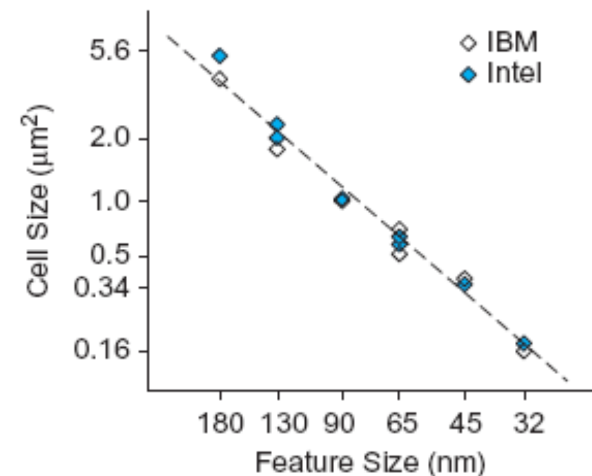
Thin Cell

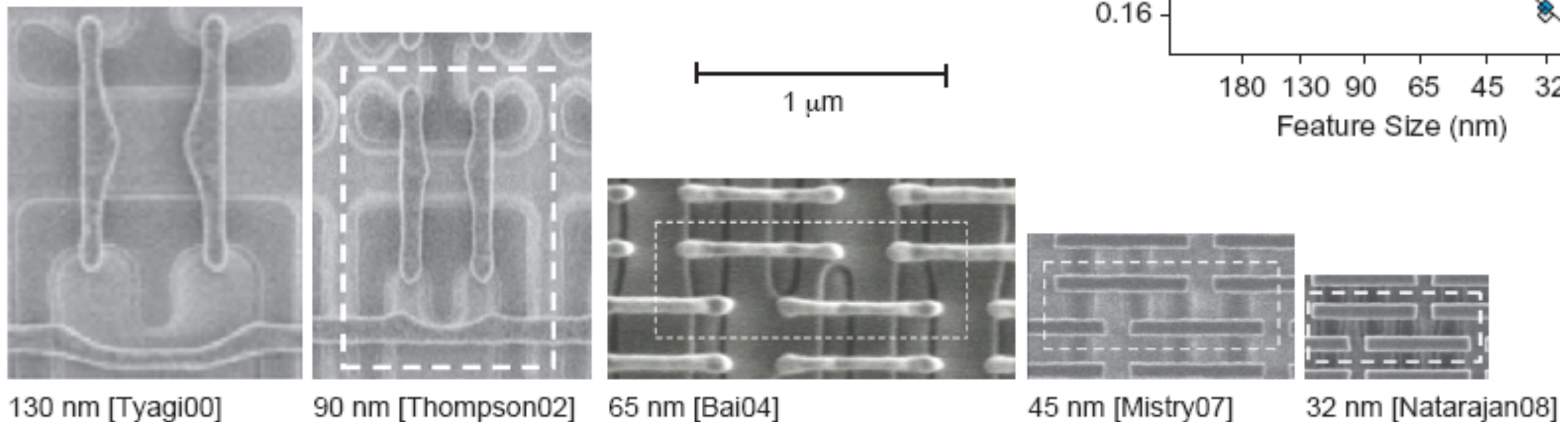
- In nanometer CMOS
  - Avoid bends in polysilicon and diffusion
  - Orient all transistors in one direction
- *Lithographically friendly* or *thin cell* layout fixes this
  - Also reduces length and capacitance of bitlines



Commercial SRAMs

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



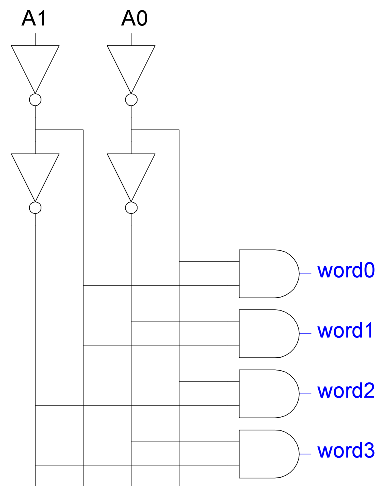


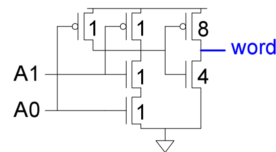
130 nm [Tyagi00] 90 nm [Thompson02] 65 nm [Bai04] 45 nm [Mistry07] 32 nm [Natarajan08]

Decoders

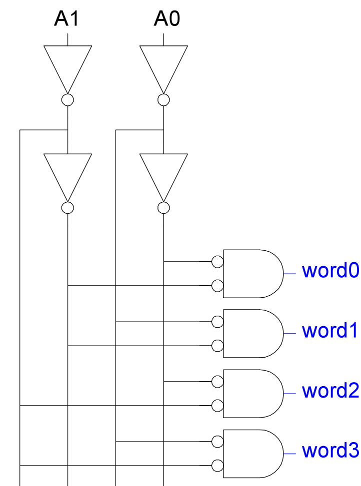
- n:2^n decoder consists of 2^n n-input AND gates
  - One needed for each row of memory
  - Build AND from NAND or NOR gates

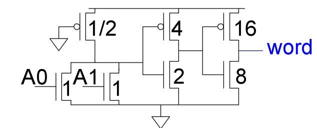
Static CMOS





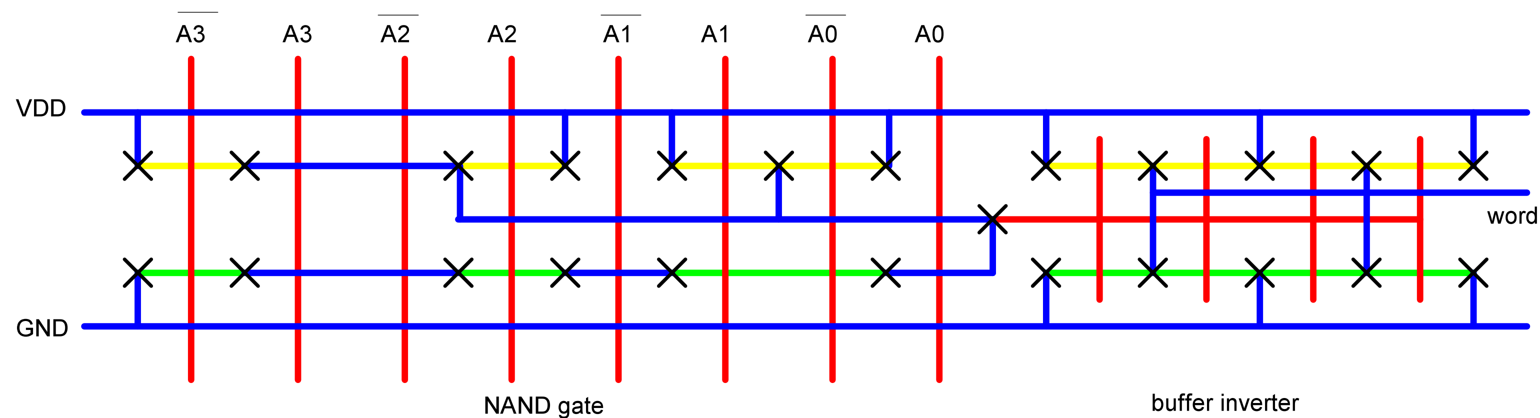
Pseudo-nMOS





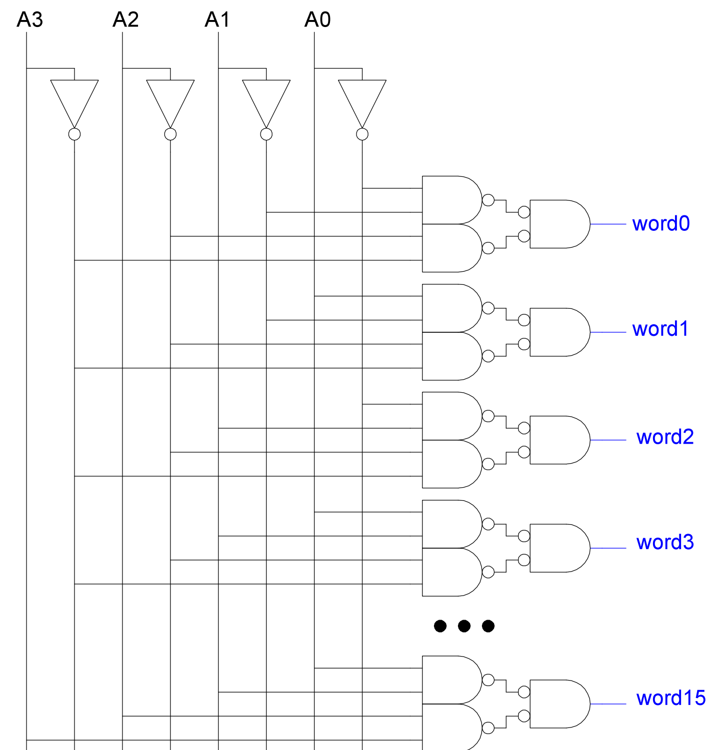
Decoder Layout

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



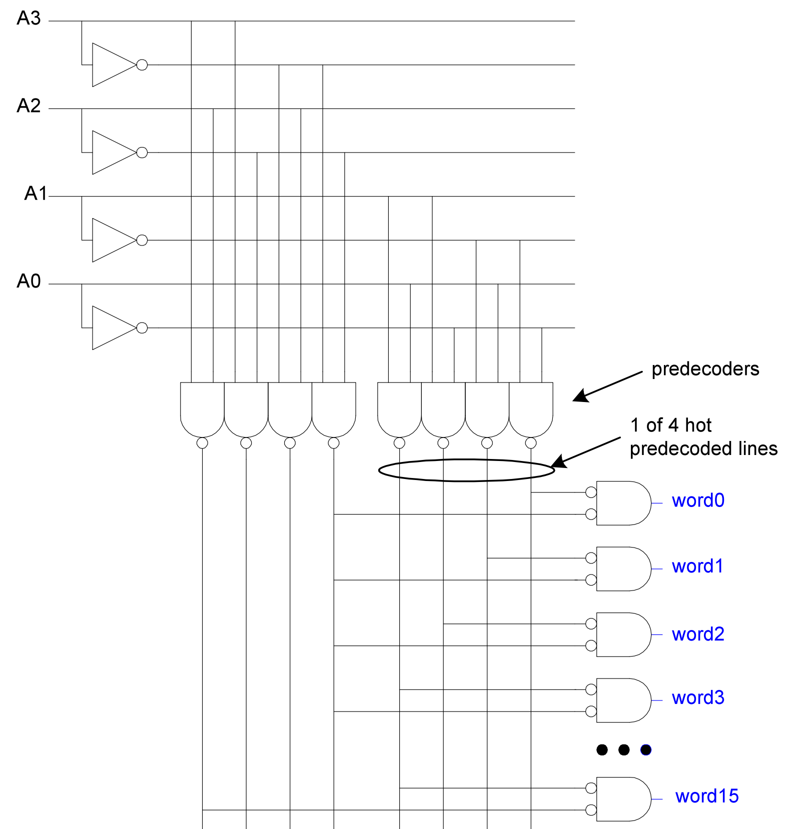
Large Decoders

- For n > 4, NAND gates become slow
  - Break large gates into multiple smaller gates



Predecoding

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

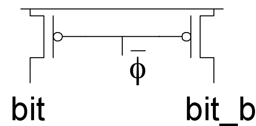


Column Circuitry

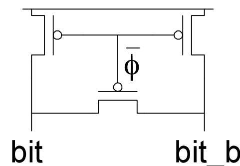
- Some circuitry is required for each column
  - Bitline conditioning
  - Sense amplifiers
  - Column multiplexing

Bitline Conditioning

- Precharge bitlines high before reads



- Equalize bitlines to minimize voltage difference when using sense amplifiers

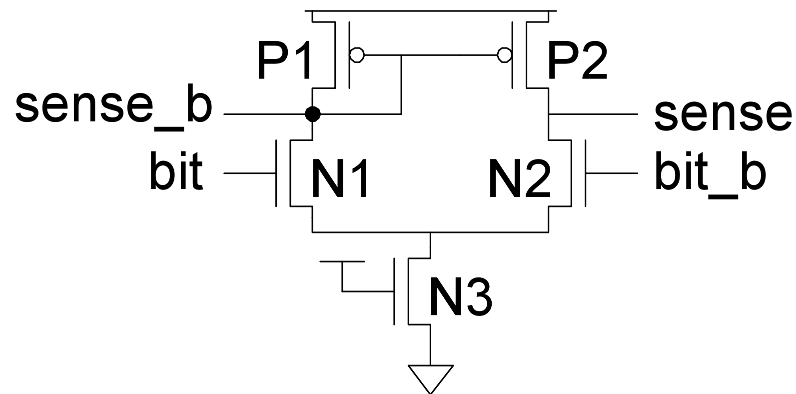


Sense Amplifiers

- 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* are triggered on small voltage swing (reduce ΔV)

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