Read/Write Memory Architectures

Dr. Shubhajit Roy Chowdhury,

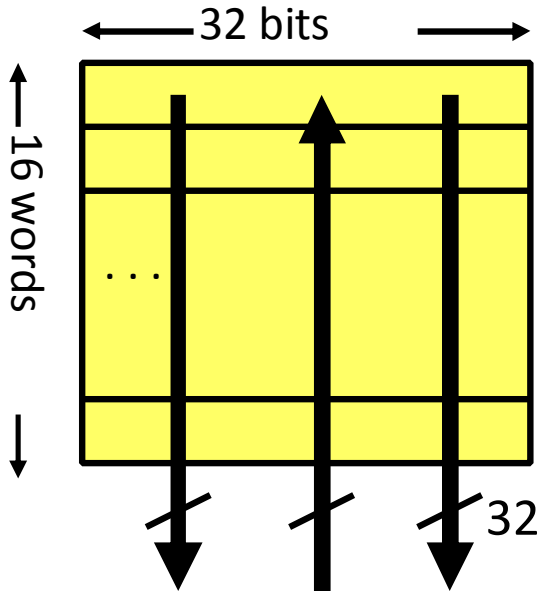
Centre for VLSI and Embedded Systems Technology,

IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

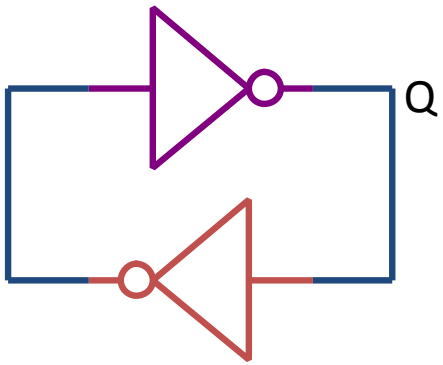
Registers

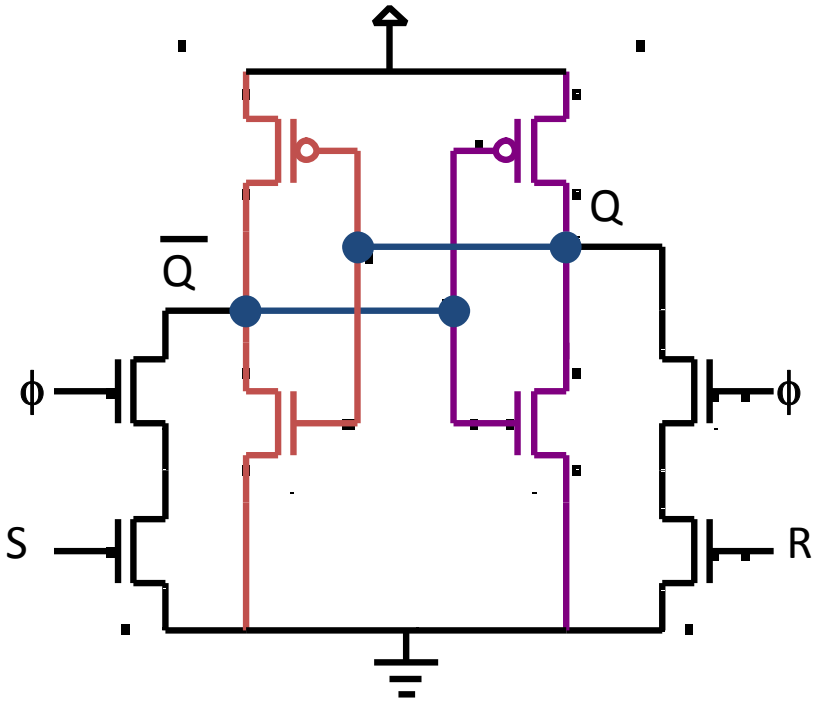
- Used for storing data
- Structure
  - N-bit wide
  - Parallel/serial read/write
  - Clocked
  - Static/dynamic implementation
- Register files
  - Multiple read/write ports possible
  - Example: 32-bit wide by 16-bit deep, dual-port parallel read, single port parallel write register file



Implementing Registers in CMOS

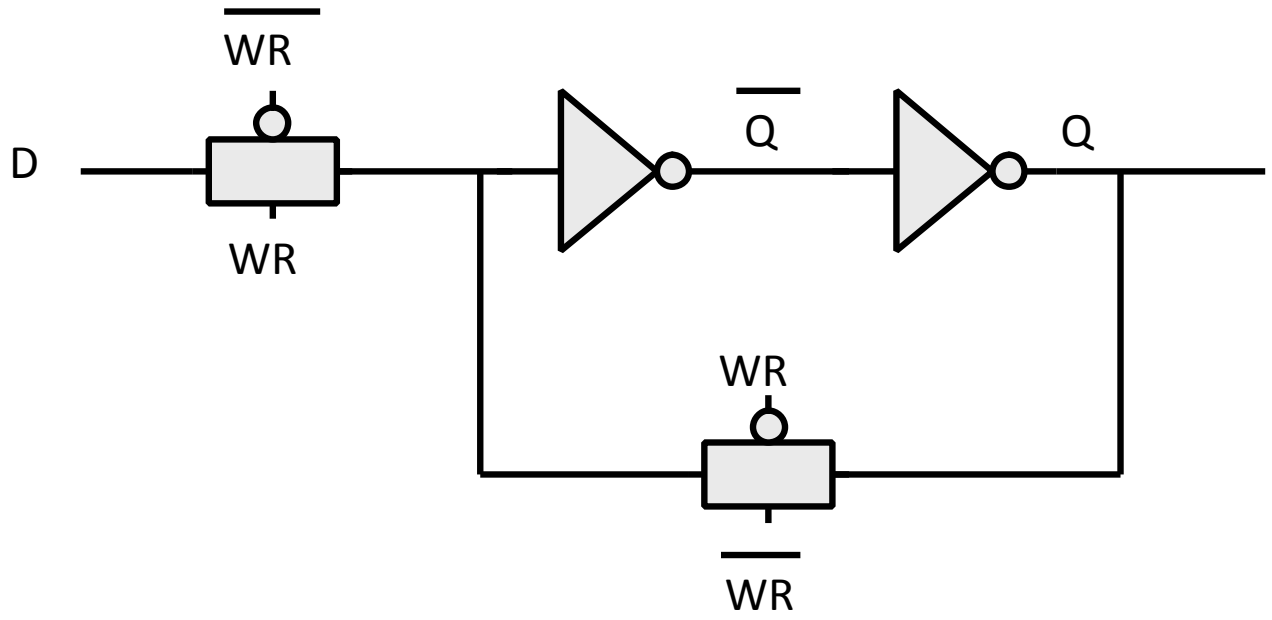
- Direct gate implementation too costly
  - A master-slave JK flip-flop uses 38 CMOS transistors
- Directly implement in transistors
  - Example: clocked SR FF





Implementing Registers in CMOS (cont.)

- Another example: D latch (register)
  - Uses transmission gate
  - When “WR” asserted, “write” operation will take place
  - Stack D latch structures to get n-bit register



Memory (Array) Design

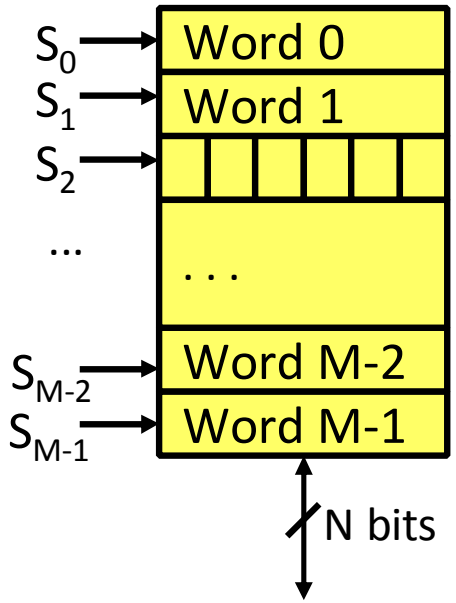
- Array of bits
- Area very important
  - Memory takes considerable area in processor chips
  - Compaction results in fewer memory chip modules, more on-chip cache
- Timing and power consumption of memory blocks have significant impact on the system
- Different types
  - RAM (SRAM, DRAM, CAM)
  - ROM (PROM, EEPROM, FLASH)

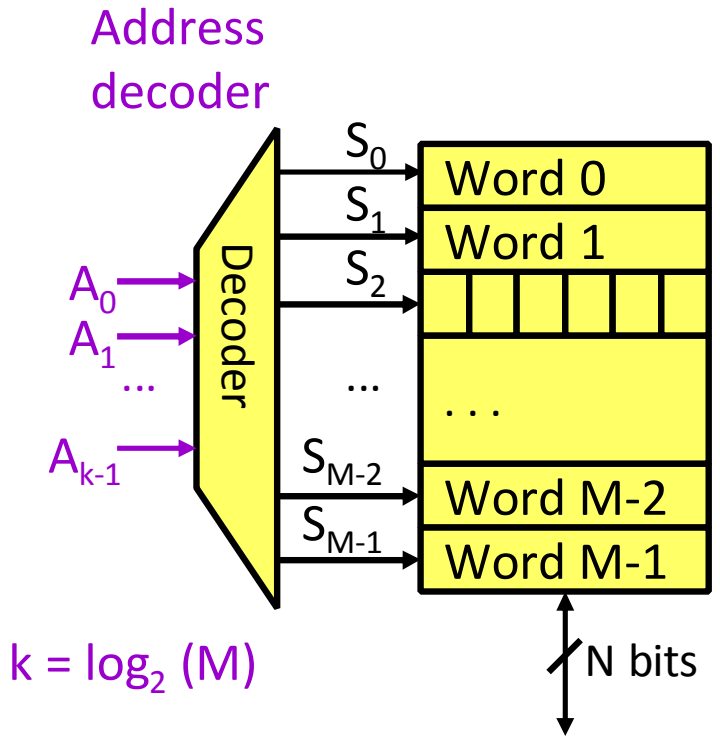
Memory Design (cont.)

- Static vs. dynamic RAM
  - Dynamic needs refreshing
    - Refreshing: read, then write back to restore charge
    - Either periodically or after each read
- Static (SRAM)
  - Data stored as long as supply voltage is applied
  - Large (6 transistors/cell)
  - Fast
- Dynamic (DRAM)
  - Periodic refresh required
  - Small (1-3 transistors/cell)
  - Slower
  - Special fabrication process

Memory Architecture: the Big Picture

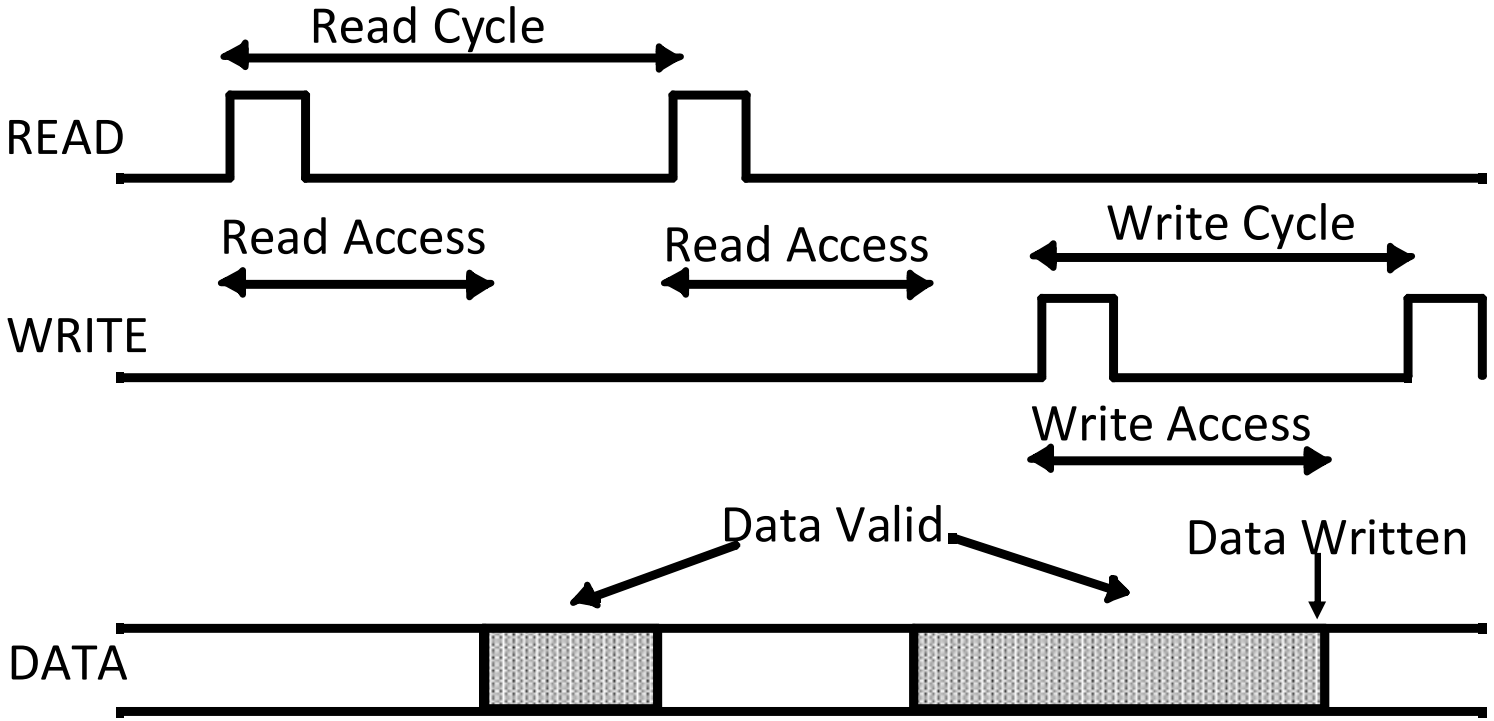
- Address: which one of the M words to access
- Data: the N bits of the word are read/written





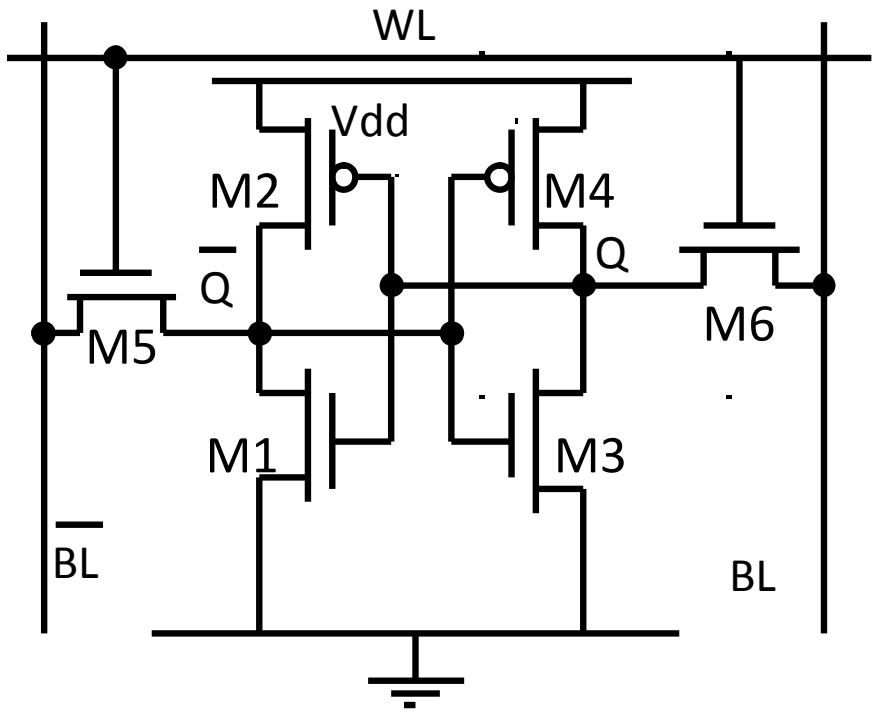
Memory Access Timing: the Big Picture

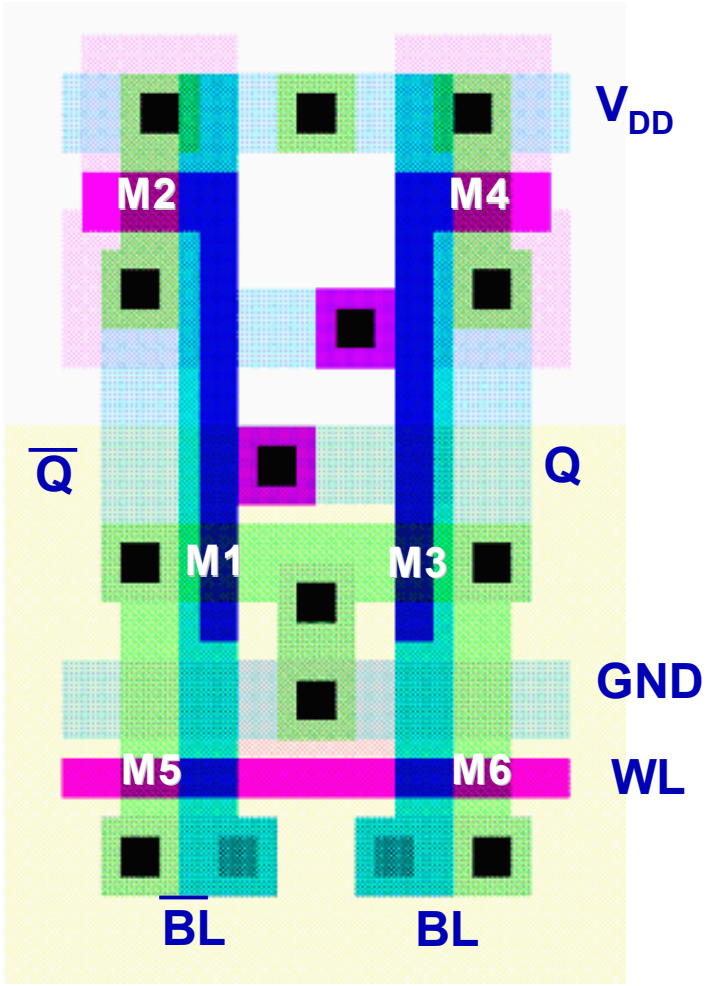
- Timing:
  - Send address on the address lines, wait for the word line to become stable
  - Read/write data on the data lines



6-Transistor SRAM Cell: Layout

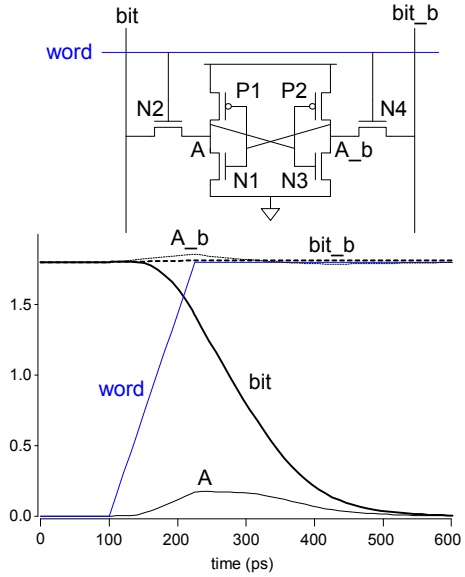
- WL is word line (select line Sj)
- BL is bit line (bit_i)





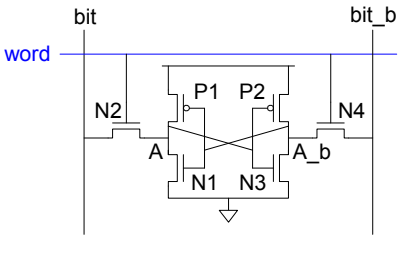
SRAM Read

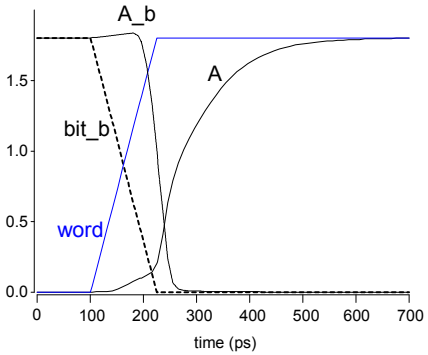
- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- *Read stability*
  - A must not flip



SRAM Write

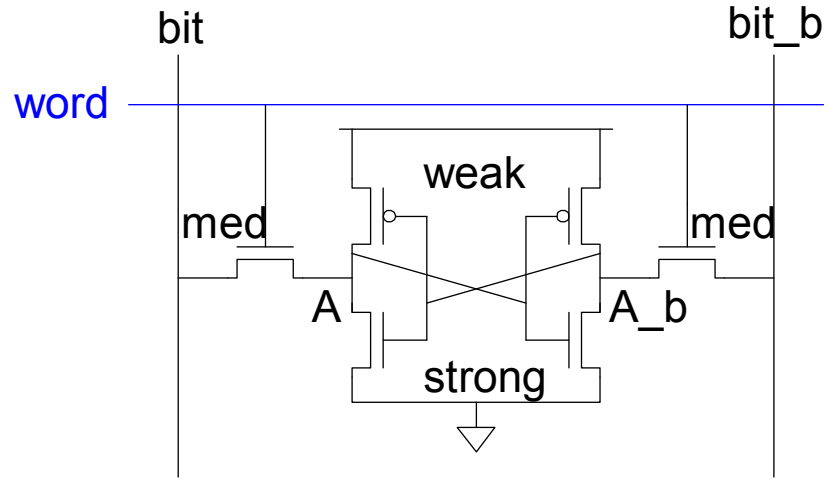
- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- Ex: A = 0, A_b = 1, bit = 1, bit_b = 0
  - Force A_b low, then A rises high
- *Writability*
  - Must overpower feedback inverter
  - N2 >> P1





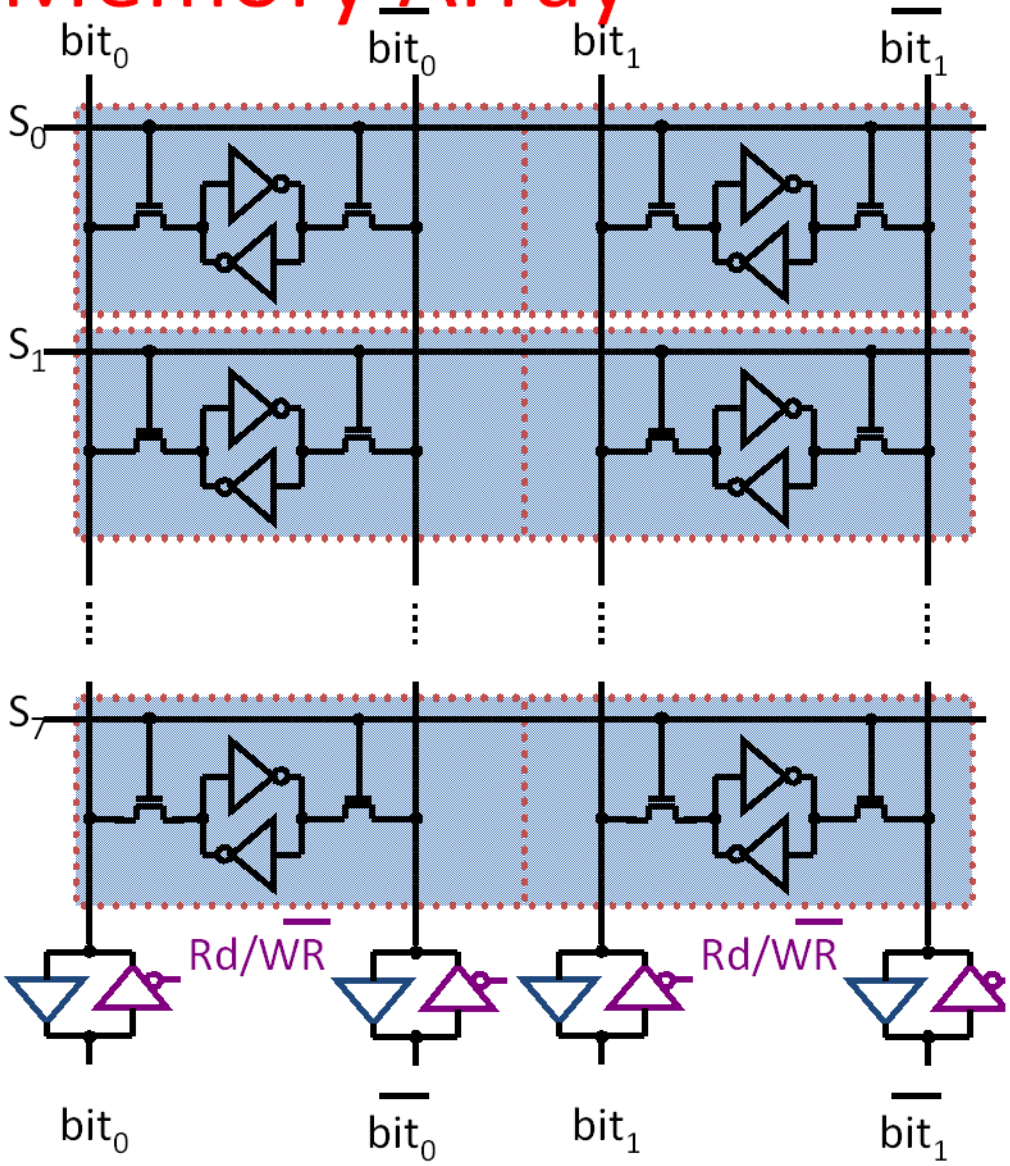
SRAM sizing

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



6-Transistor Memory Array

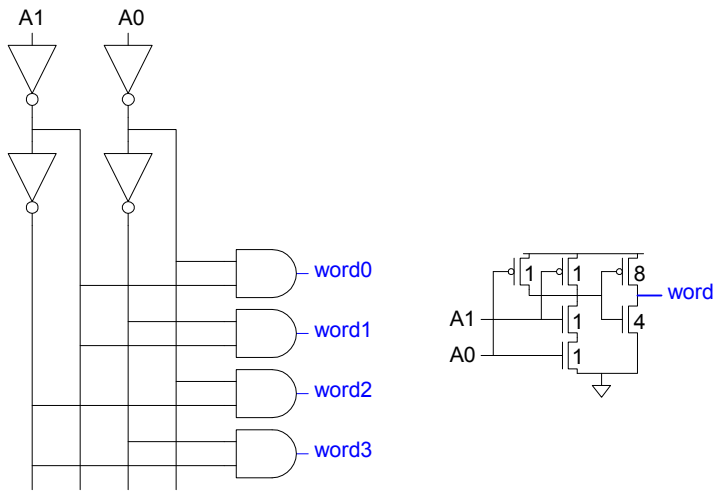
- 8 words deep RAM, 2 bits wide words
- To write to word j:
  - Set S_j=1, all other S lines to 0
  - Send data on the global bit_0, bit_0', bit_1, bit_1'
- To read word k:
  - Set S_k=1, all other S lines to 0
  - Sense data on bit_0 and bit_1.



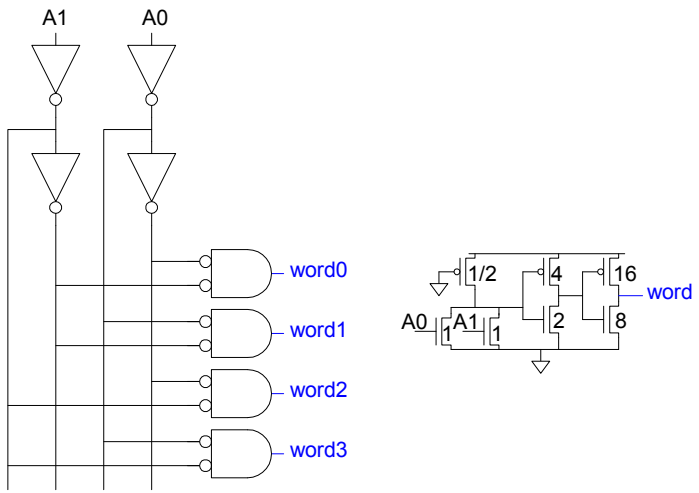
Decoders

- 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

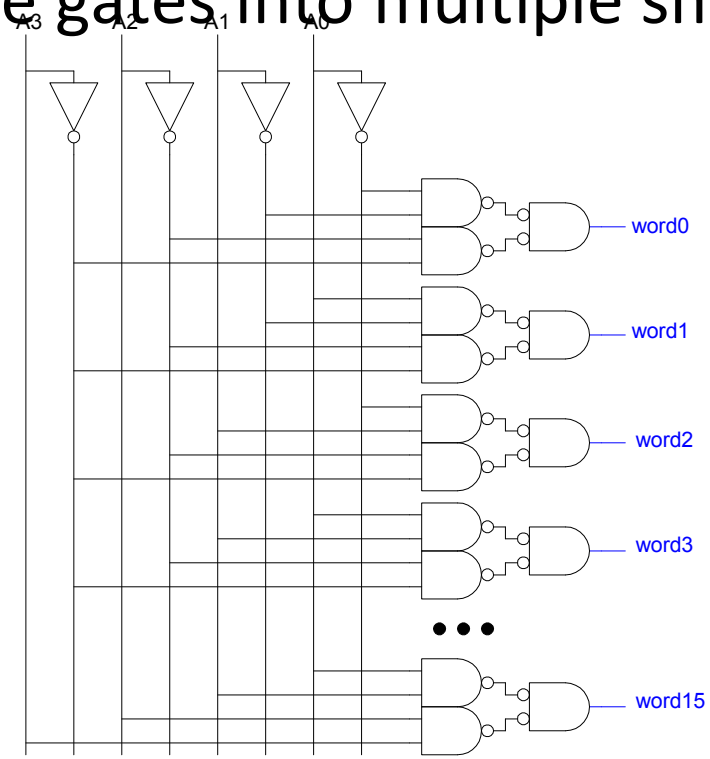


Pseudo-nMOS



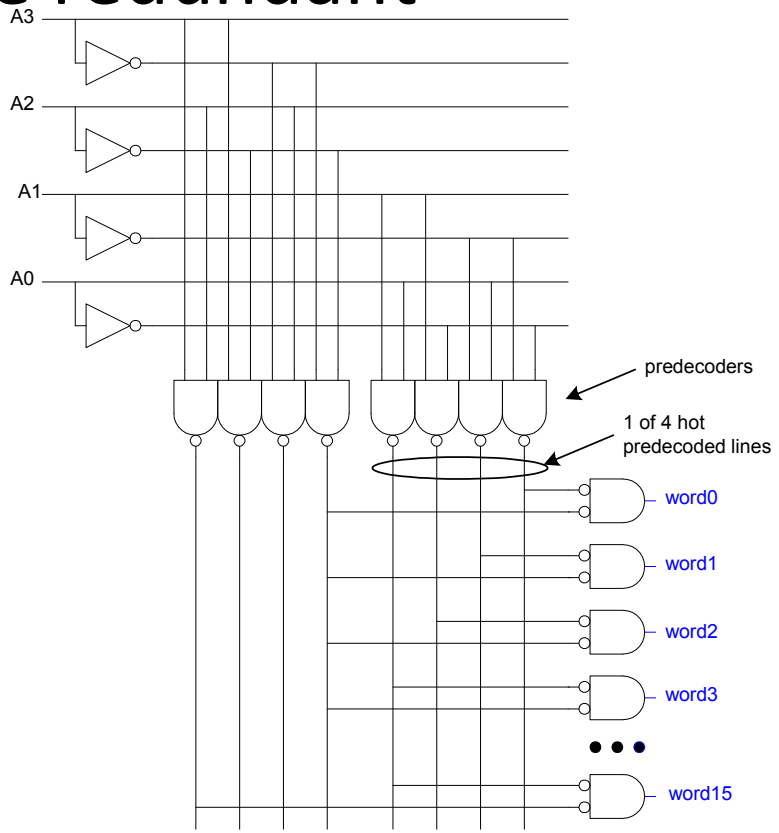
Large Decoders

- 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

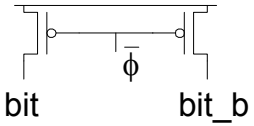


Column Circuitry

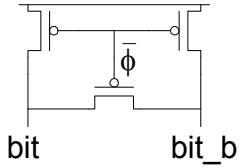
- Some circuitry is required for each column
  - Bitline conditioning
  - Sense amplifiers
  - Column multiplexing

Bitline Conditioning

- Precharge bitlines high before reads



- Equalize bitlines to minimize voltage difference when using sense amplifiers

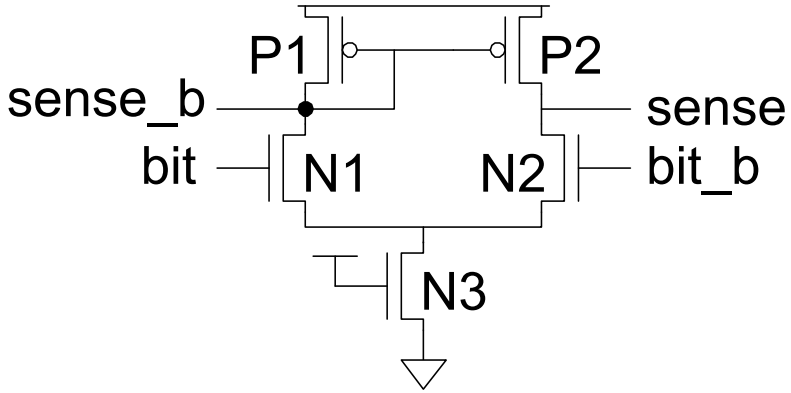


Sense Amplifiers

- Bitlines have many cells attached
  - Ex: 32-kbit SRAM has 256 rows x 128 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)

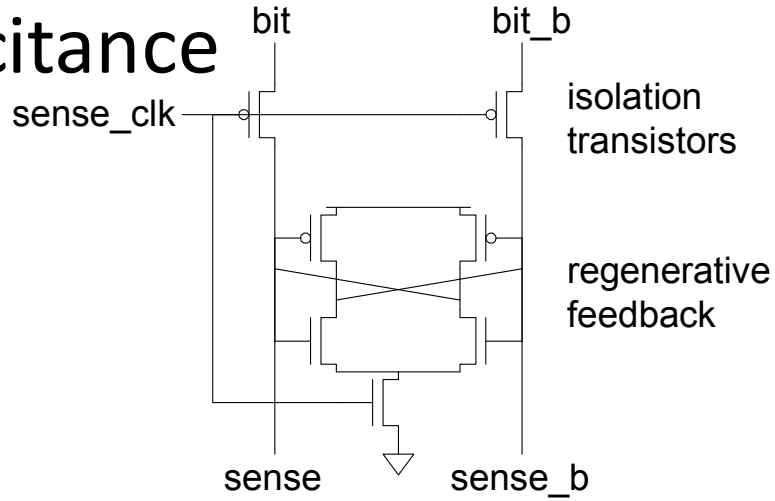
Differential Pair Amp

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



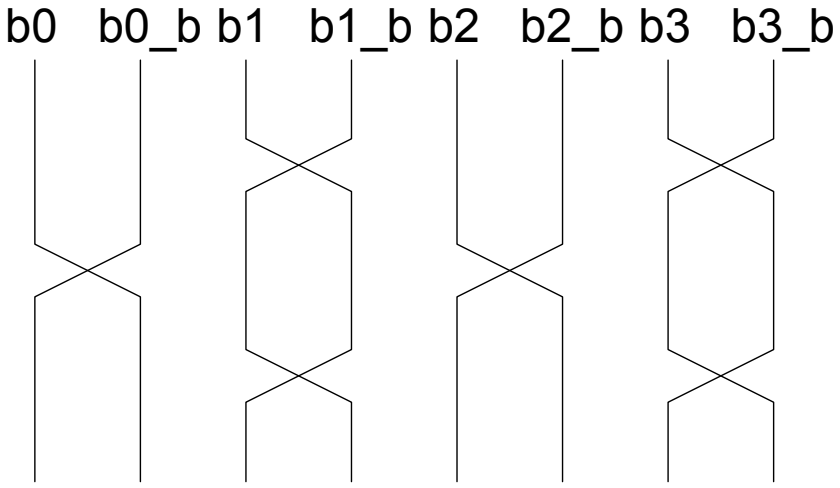
Clocked Sense Amp

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



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

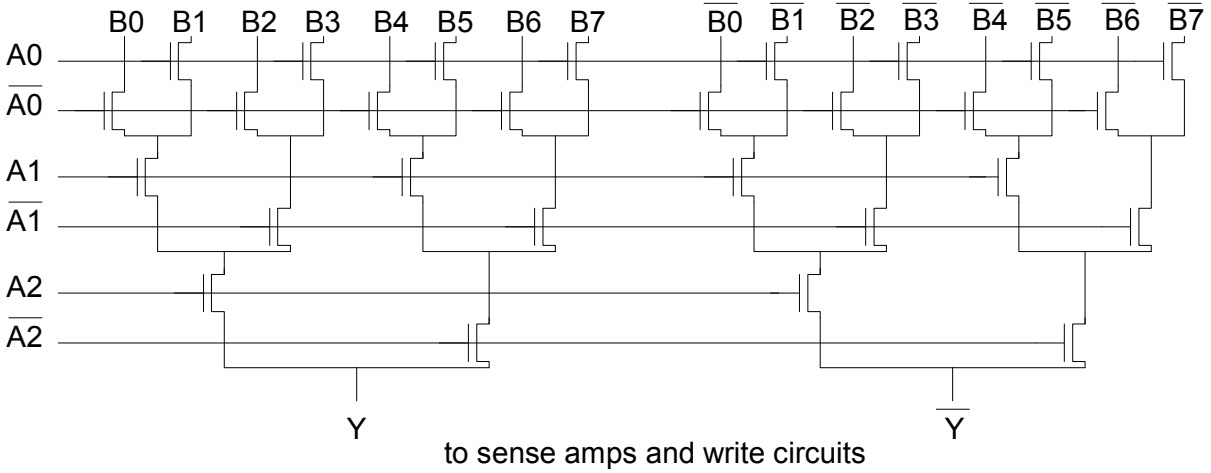


Column Multiplexing

- 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

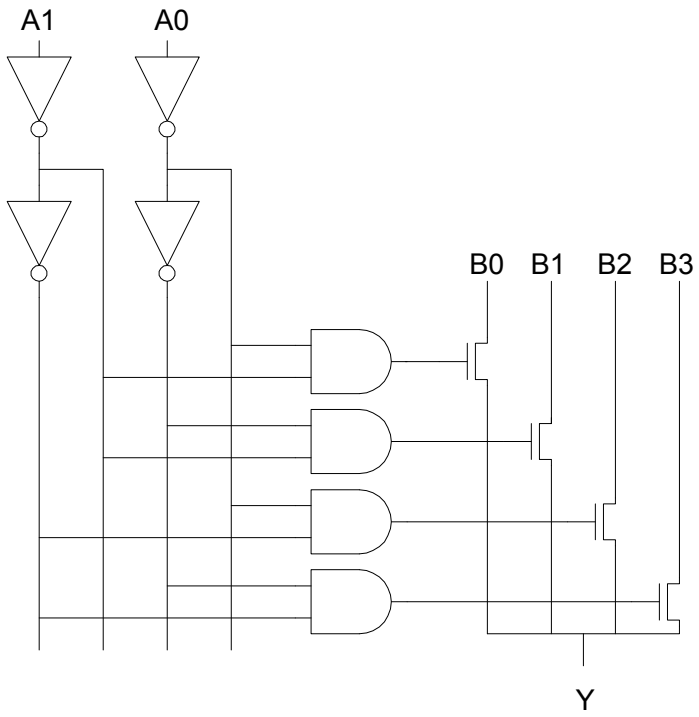
Tree Decoder Mux

- 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

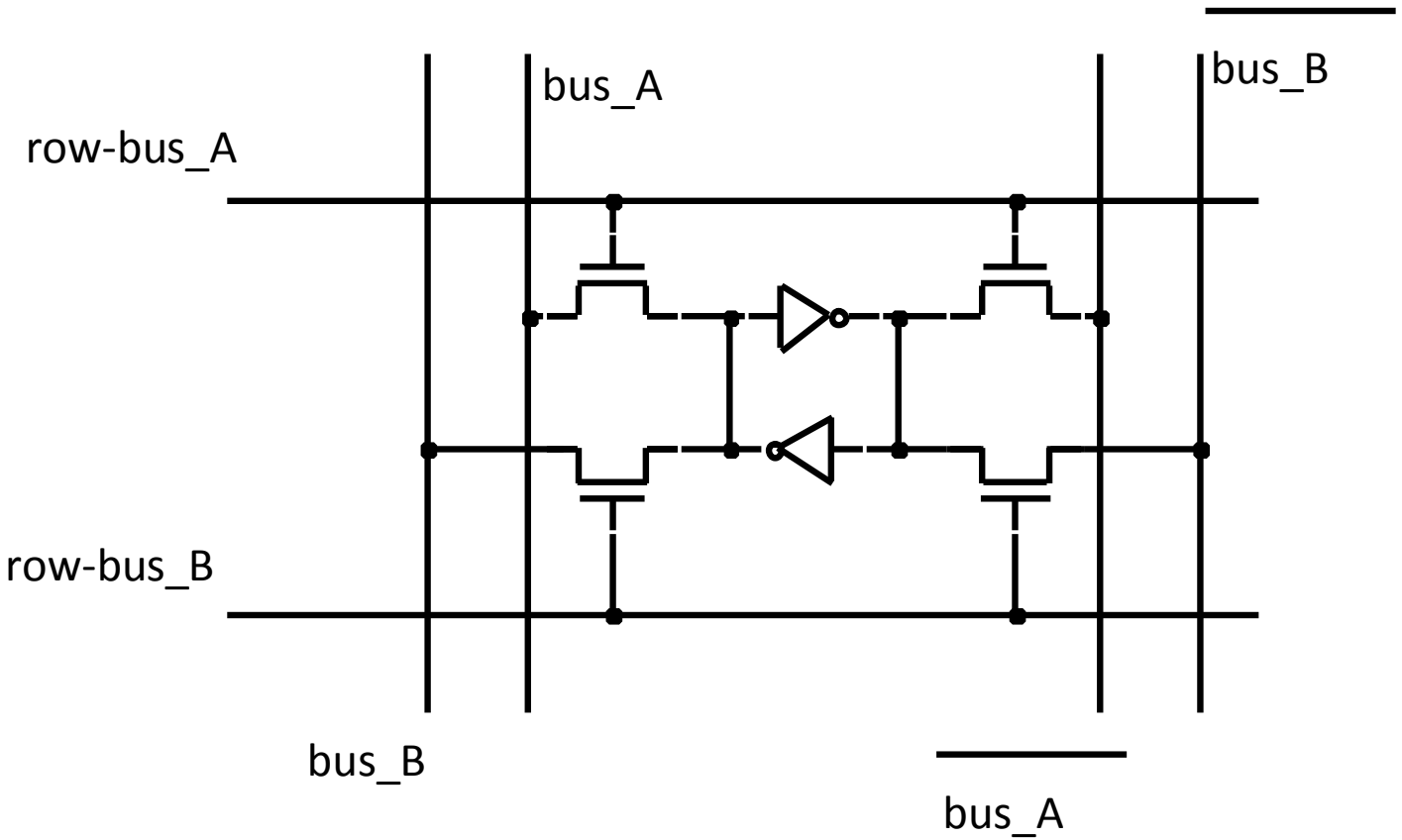
- Or eliminate series transistors with separate decoder



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

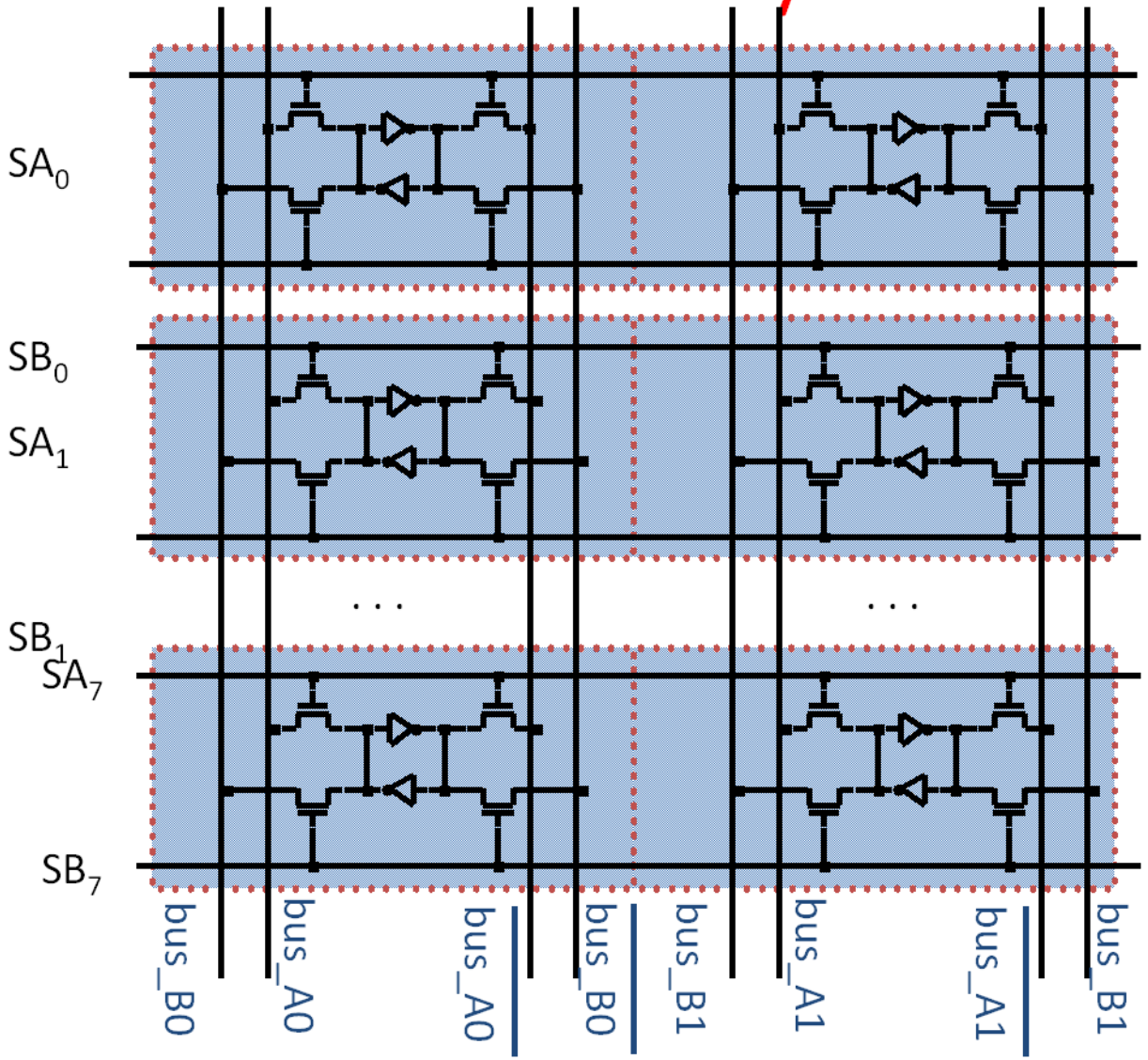
Multi-Port SRAM Cells



- Idea: add more input and output transistors
- Can be applied to all variants
  - Usually not done for 1T cells

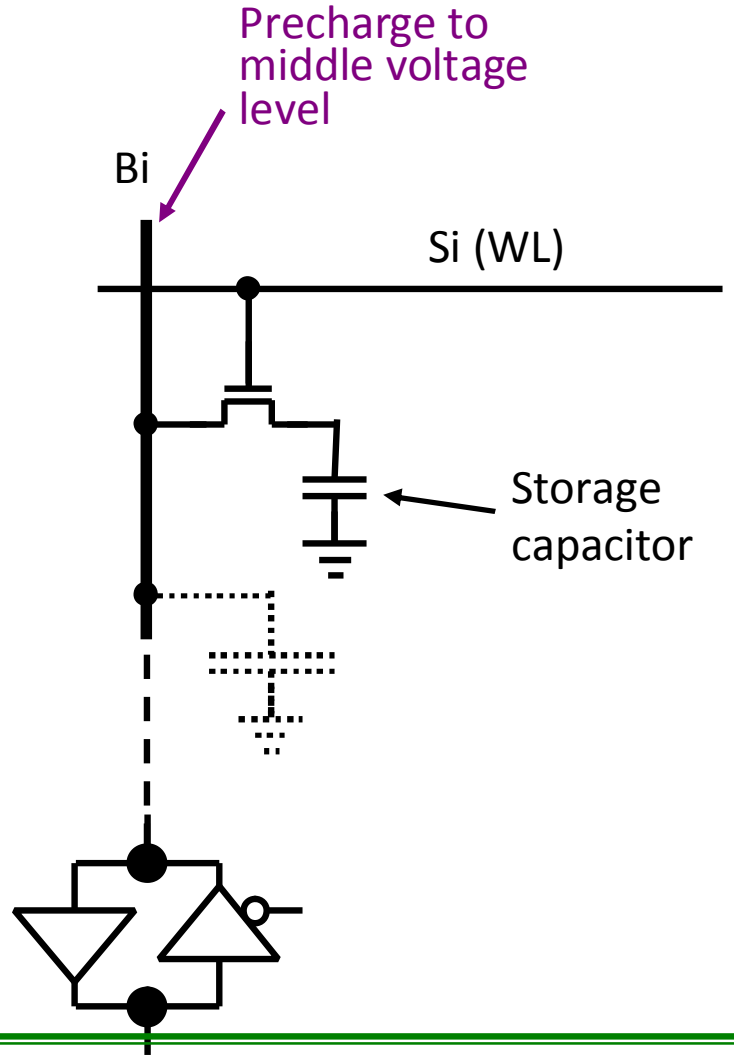
Multi-Port SRAM Cells Array

- 7 words deep, 2 wide words, dual port mem
- To read from word j and write "d_1d_0" to word k simultaneously:
  - Set SA_j=1, and all other SA's=0
  - Set SB_k=1, and all other SB's=0
  - Sense the values on bus_A0 and bus_A1
  - Write d_1d_0 to bus_B0 and bus_B1

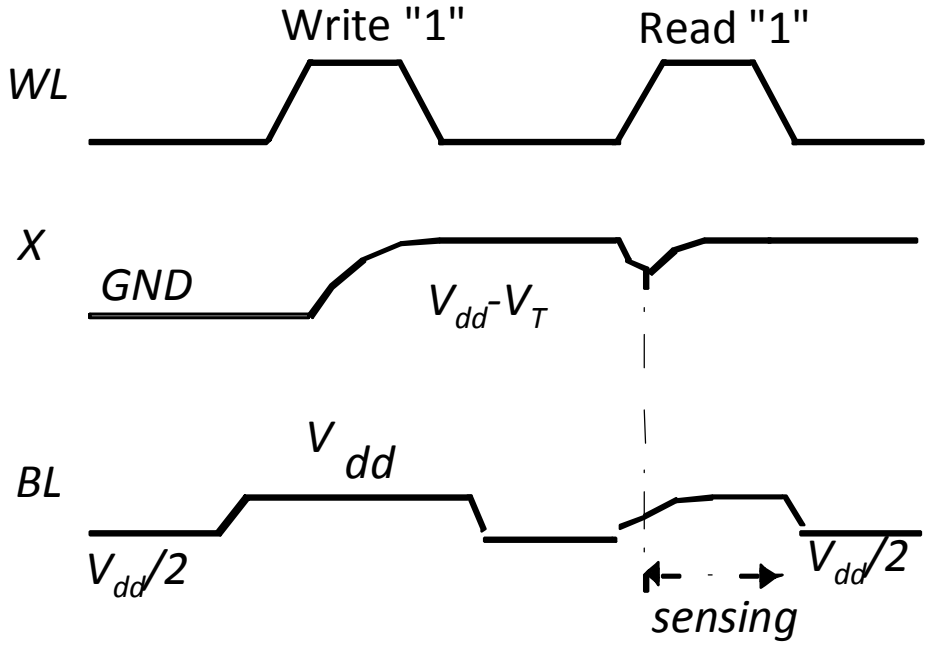


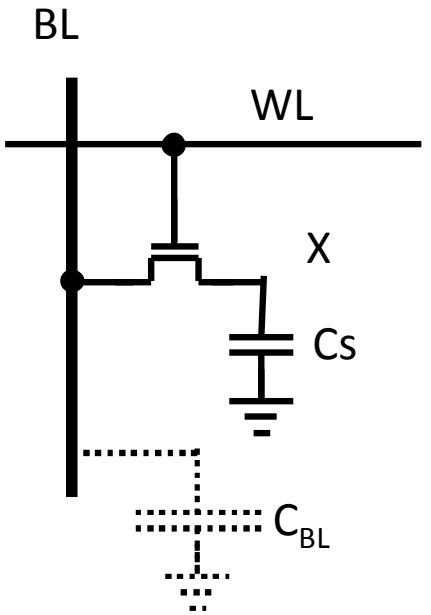
Dynamic RAM 1-Transistor Cell

- 1-transistor cell
  - Storage capacitor is source of cell transistor
  - Special processing steps to make the storage capacitor large
  - Charge sharing with bus capacitance ($C_{cell} << C_{bus}$)
  - Extra demand on sense amplifier to detect small changes
  - Destructive read (must write immediately)



Dynamic RAM 1-Transistor Cell: Timing



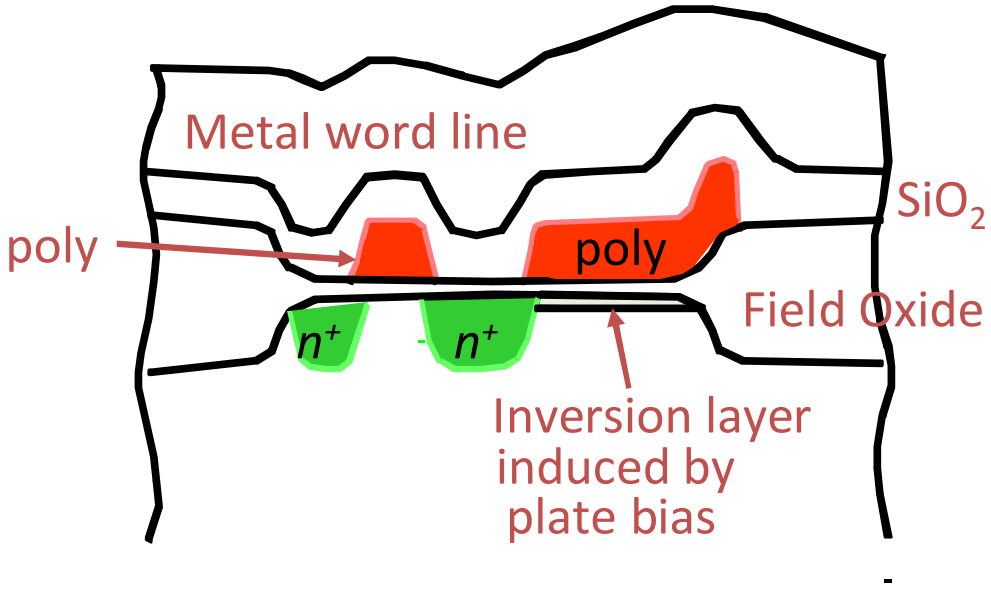


- Write: Cs is charged/discharged
- Read
  - Voltage swing is small (~250 mV)
  - $\Delta V = V_{BL} - V_{PRE} = (V_X - V_{PRE}) \cdot Cs / (Cs+C_{BL})$

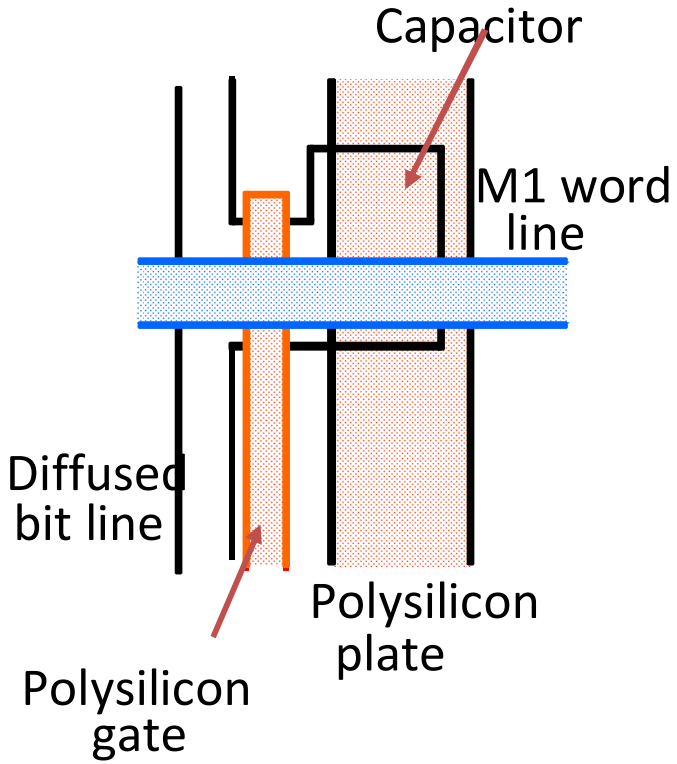
Dynamic RAM 1-Transistor Cell: Observations

- DRAM memory cell is single-ended
- Read operation is destructive
- 1T cell requires presence of an extra capacitance that must be explicitly included in the design
  - Polysilicon-diffusion plate capacitor
- When writing a “1” into a DRAM cell, a threshold voltage is lost
  - Set WL to a higher value than Vdd

Dynamic RAM 1-Transistor Cell: Layout



(a) Cross-section

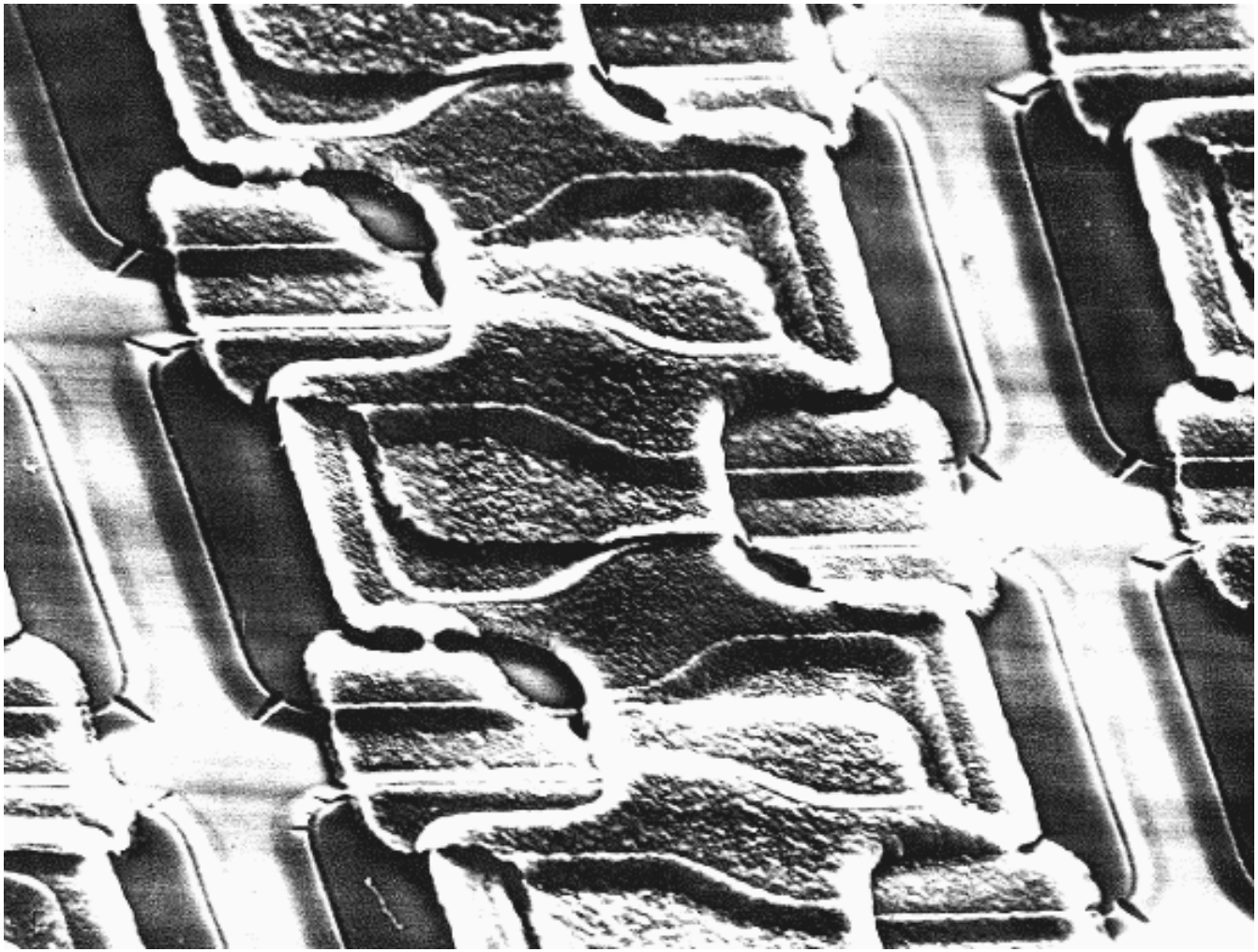


(b) Layout

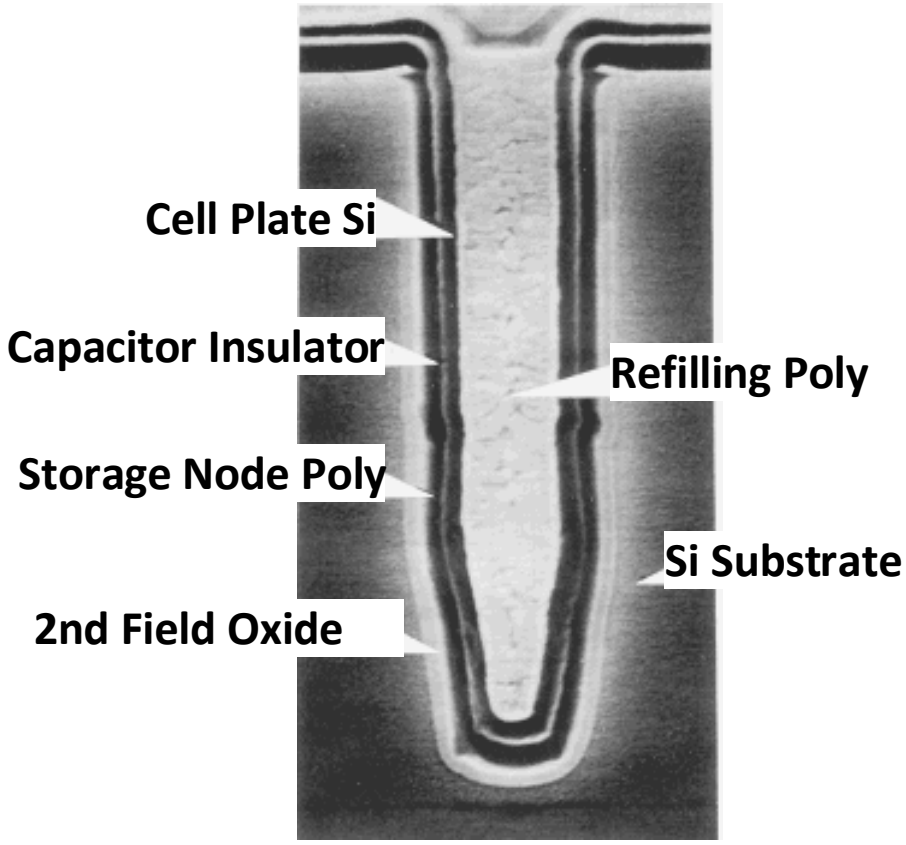
Used Polysilicon-Diffusion Capacitance

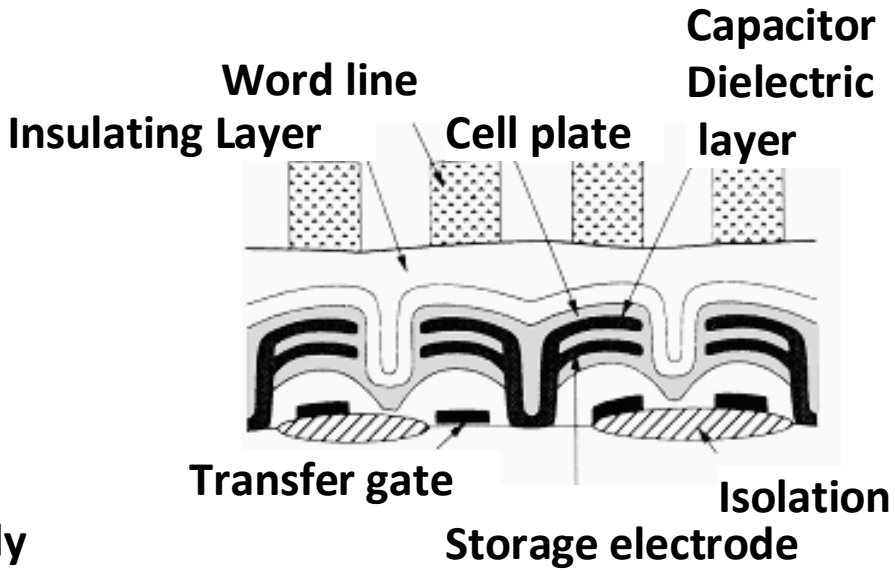
Expensive in Area

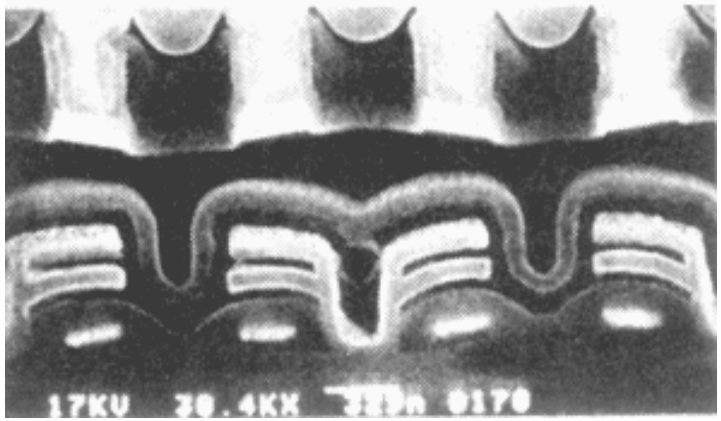
Dynamic RAM 1-Transistor Cell: Layout



Dynamic RAM 1-Transistor Cell: Layout







RAM Cells: Summary

- Static
  - Fastest (no refresh)
  - Simple design
  - Right solution for small memory arrays such as register files
- Dynamic
  - Densest: 1T is best and is the way to go for large memory arrays
  - Built-in circuitry to step through cells and refresh (can do more than one word at a time)
  - Sense amplifier needed for fast read operation

THANK YOU