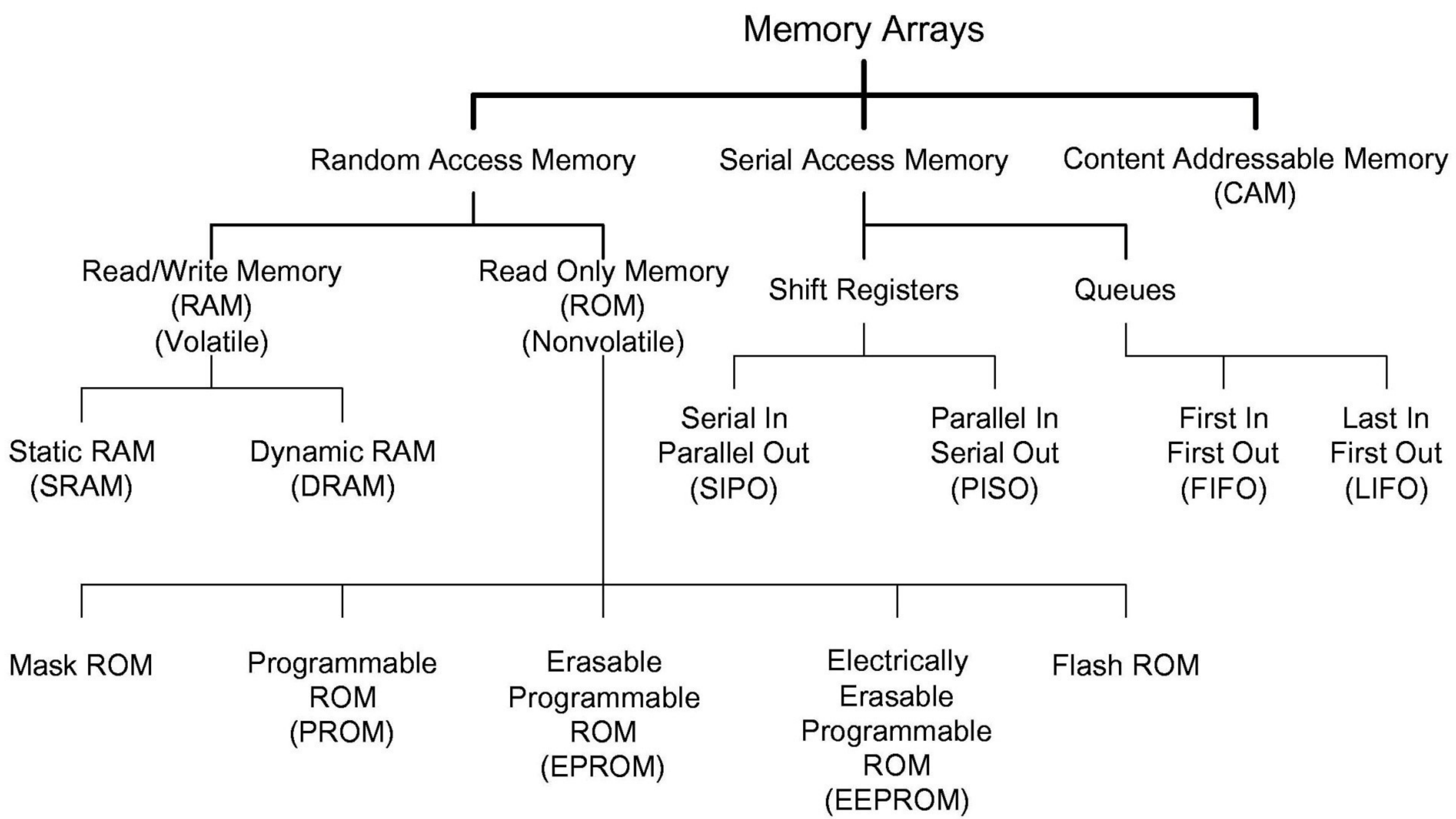
Chapter 9
Memories
(Array Subsystems)

闕志達
台灣大學電機系

Categories of Memory Arrays

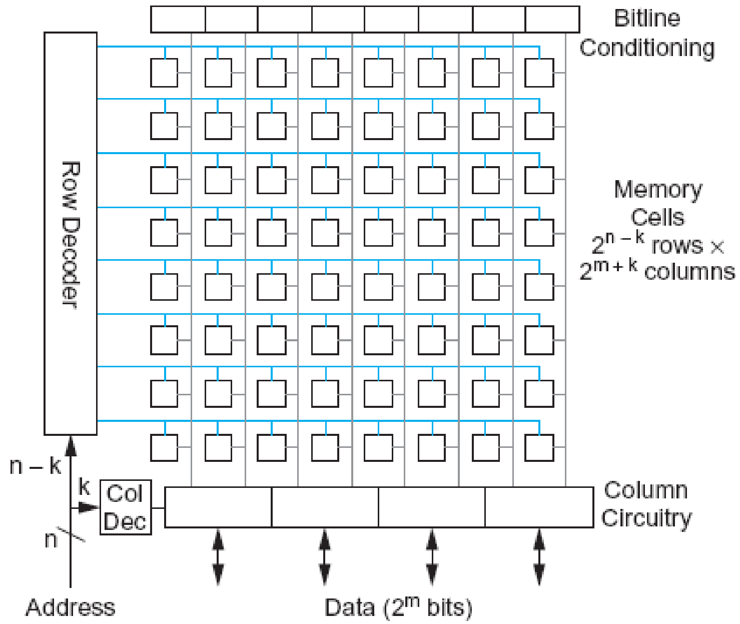


Introduction

- Category of random access memory
  - ROM (read-only memory)
    - Nonvolatile
    - Mask ROM, EPROM, EEPROM, Flash (variant of EEPROM, erases entire blocks rather than individual bits)
  - RAM (read/write memory)
    - Volatile
    - Static vs. dynamic structures
    - Area, speed, leakage, cost tradeoff
- Memory cells can have one or more ports for access

Array Architecture

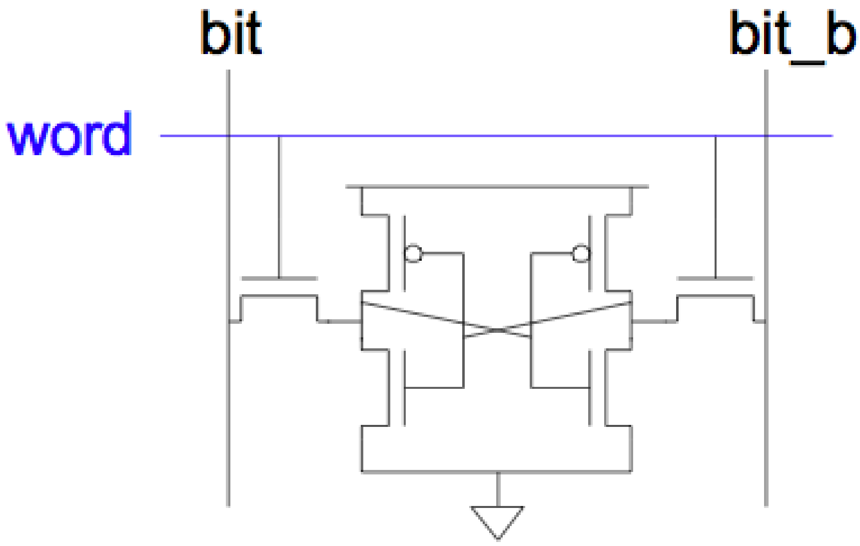
- 2^n words of 2^m bits each
- If n>>m, fold by 2^k into fewer rows of more columns



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

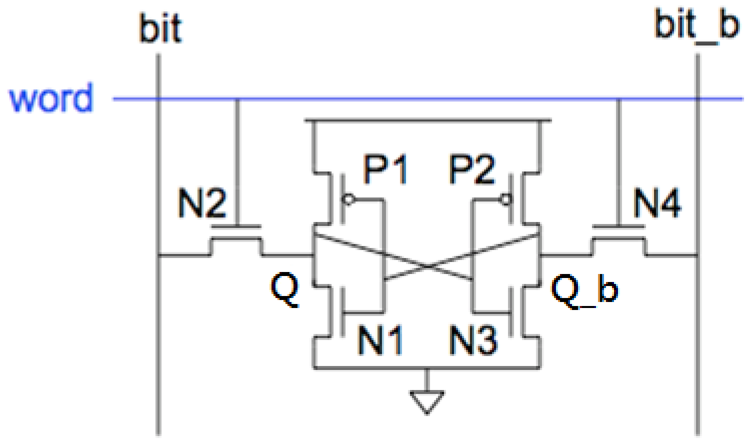
6T SRAM Cell

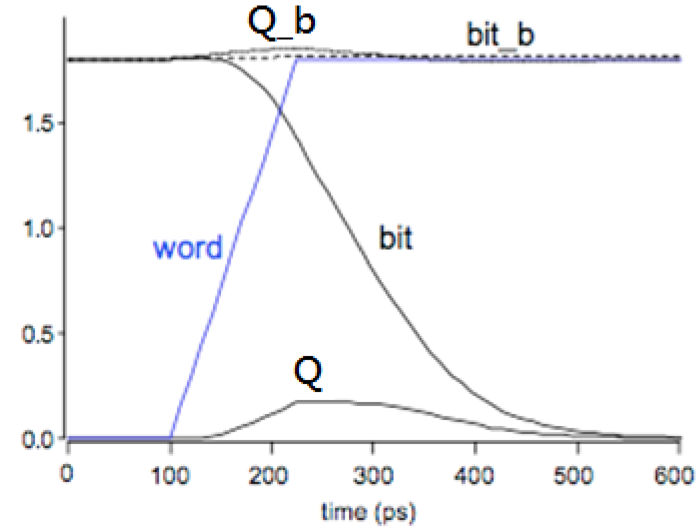
- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read
  - Precharge bit, bit_b
  - Raise wordline
- Write
  - Drive data onto bit, bit_b
  - Raise wordline



SRAM Read

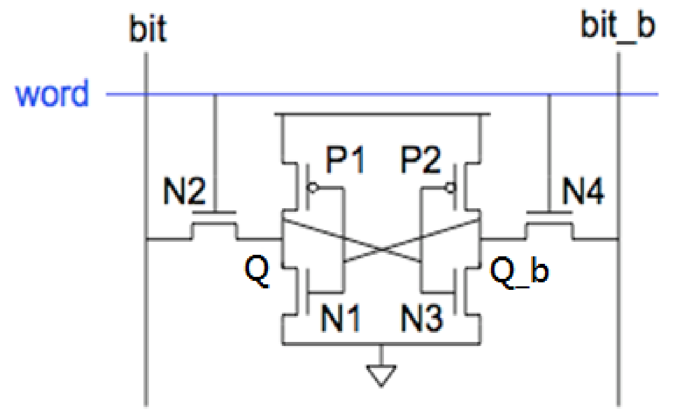
- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: Q=0, Q_b=1
  - bit discharges, bit_b stays high
  - But Q bumps up slightly
- Read stability
  - Q must not flip
  - N1>>N2

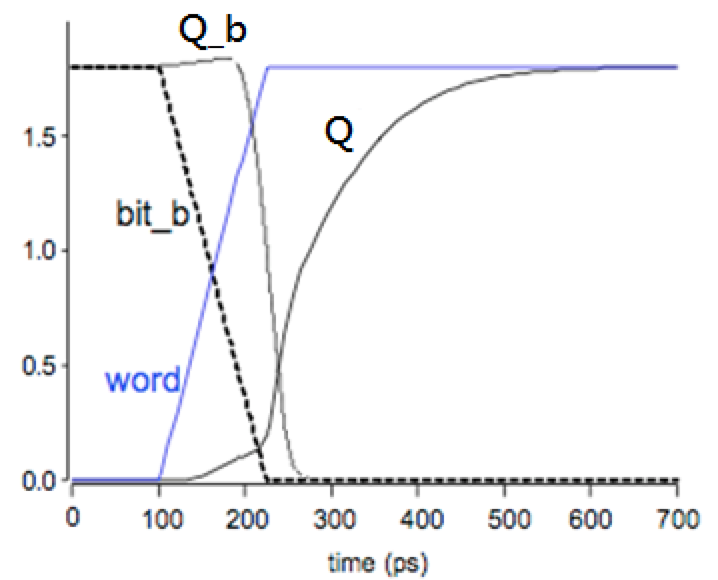




SRAM Write

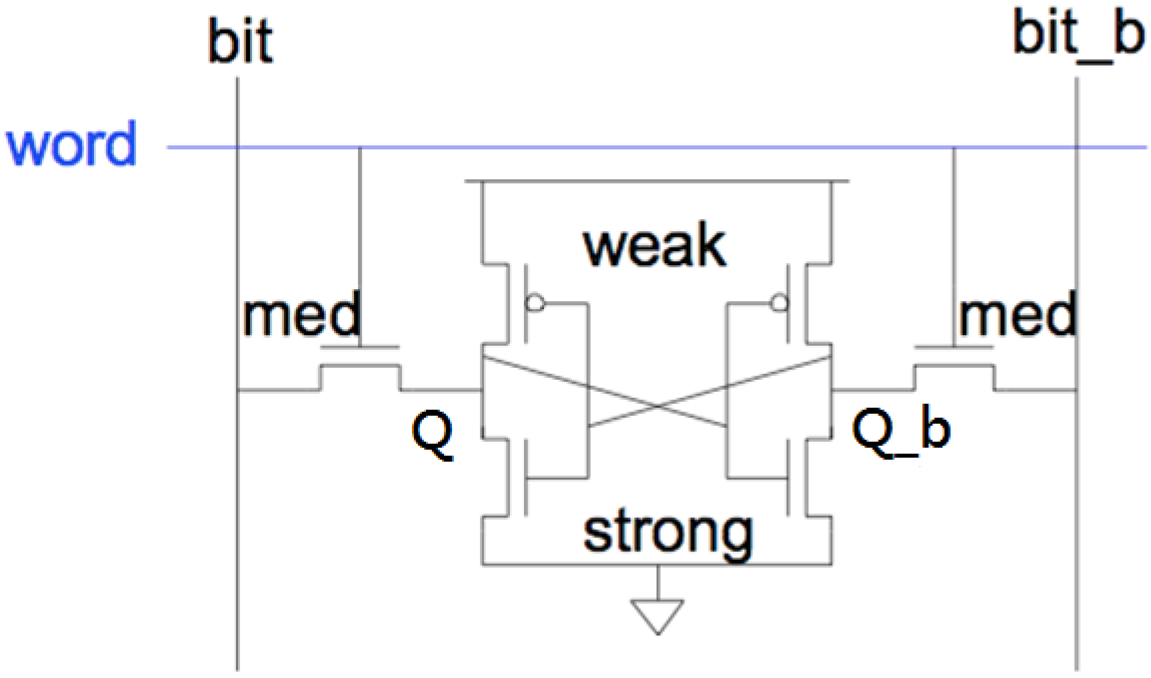
- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
- Ex: Q=0, Q_b=1, bit=1, bit_b=0
  - Force Q_b low, then Q rises high
- Writability
  - Must overpower feedback inverter
  - N2>>P1





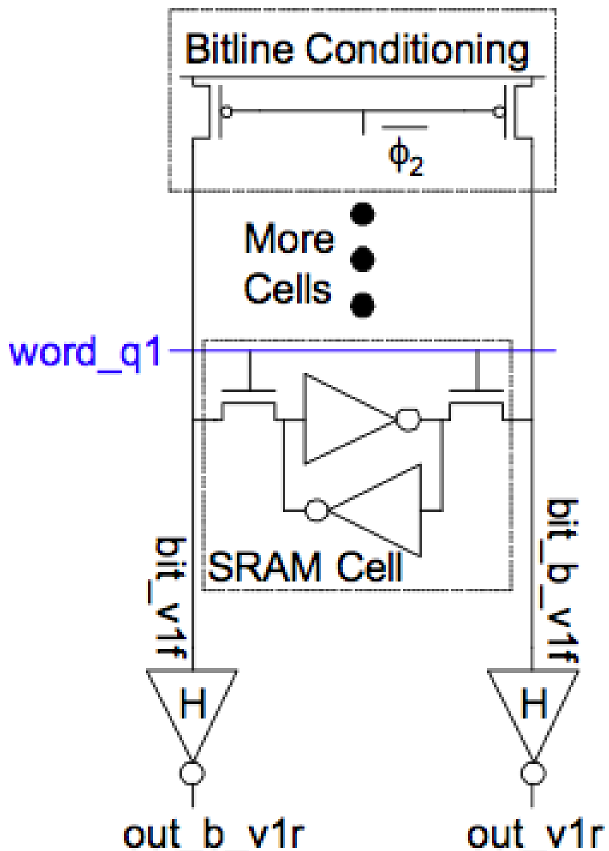
SRAM Sizing

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

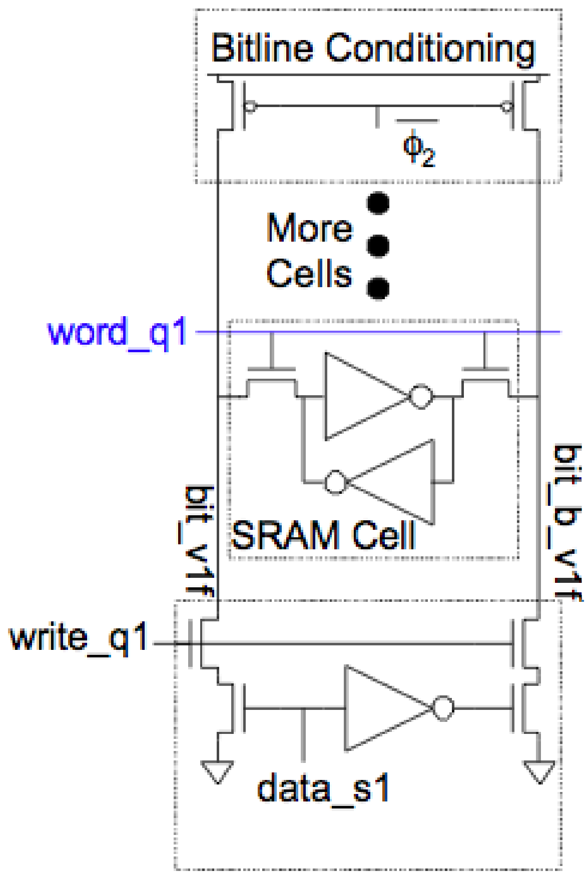


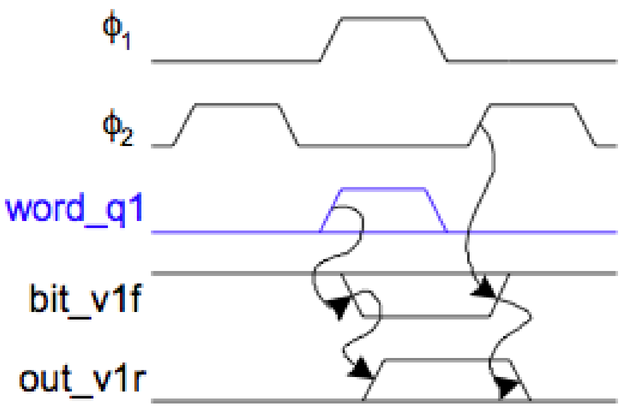
SRAM Column Example

Read



Write

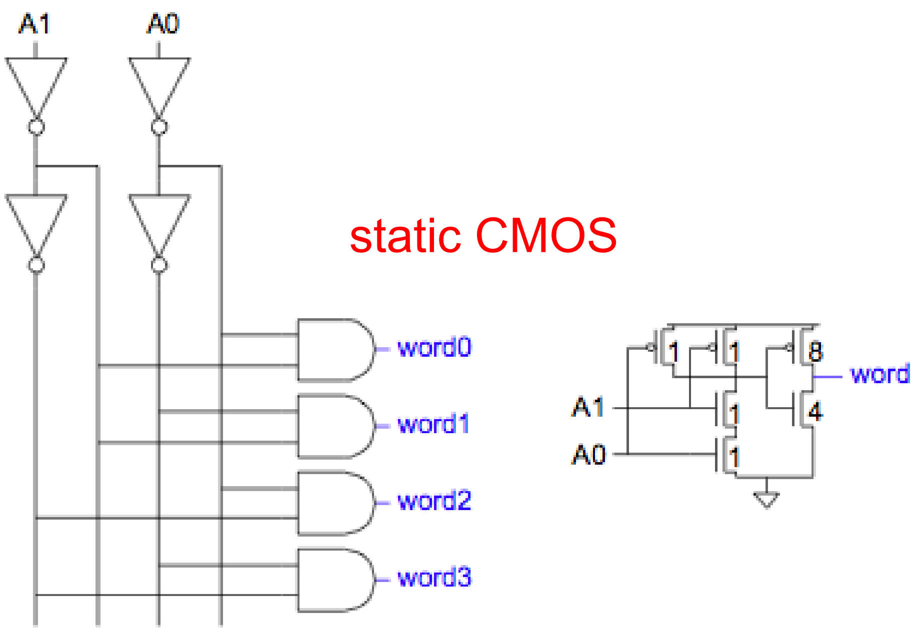




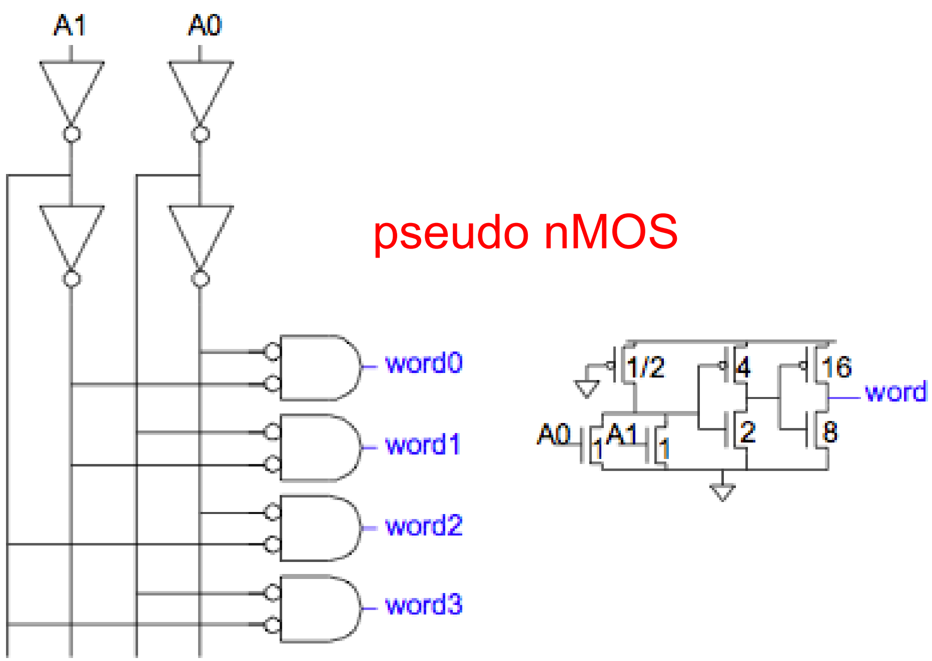
Decoders

- $n{:}2^n$ 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

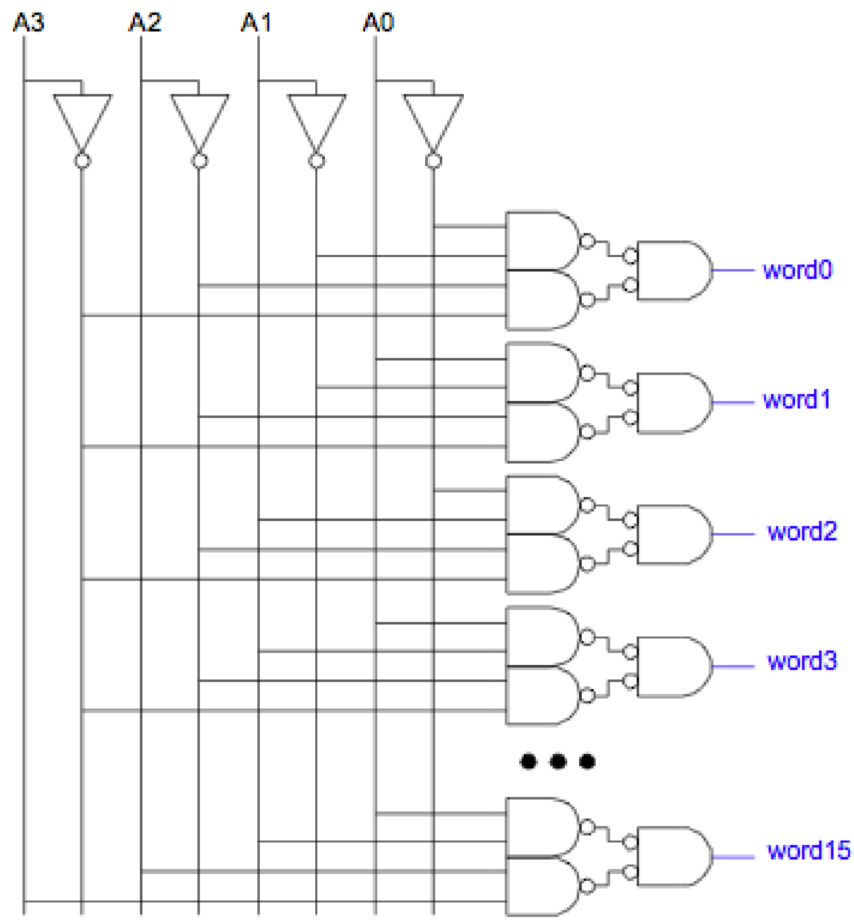


pseudo nMOS



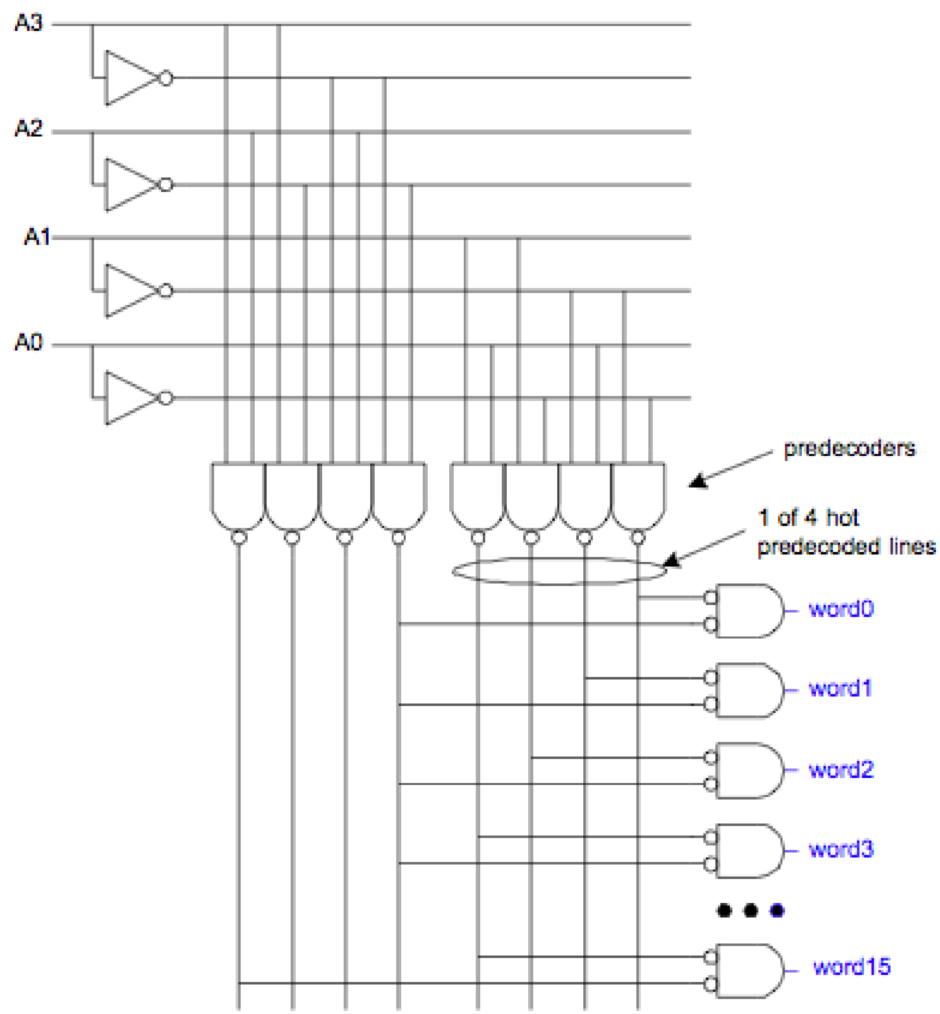
Large Decoders

- 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

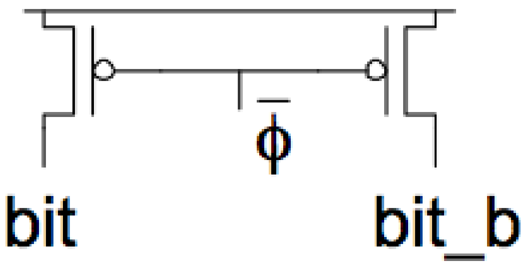


Column Circuitry

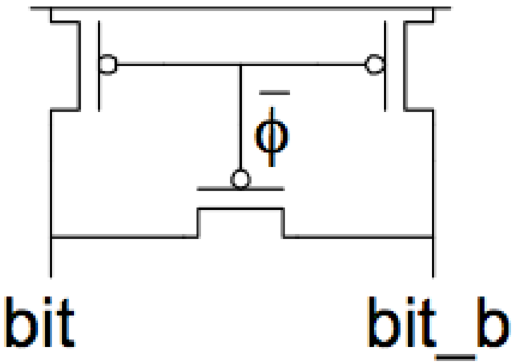
- Some circuitry is required for each column
  - Bitline conditioning
  - Sense amplifiers
  - Column multiplexing

Bitline Conditioning

- Precharge bitlines high before reads



- Equalize bitlines to minimize voltage difference and thus ensure correct sensing using sense amplifiers

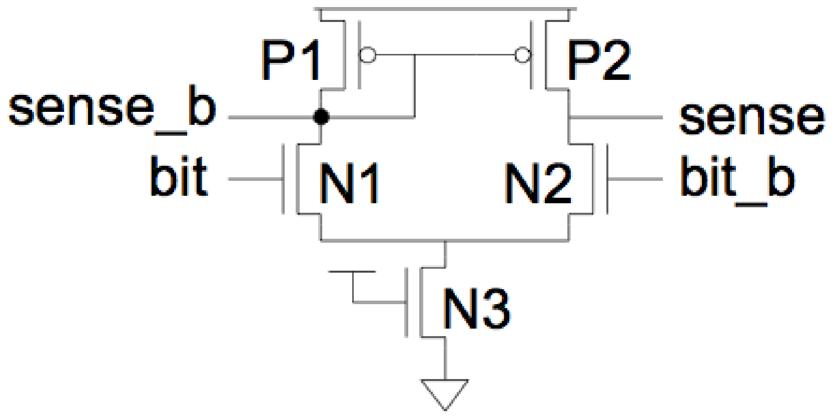


Sense Amplifiers

- Bitlines have many cells attached
  - Ex: 64-kbit SRAM has 256 rows x 256 cols
  - 256 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)

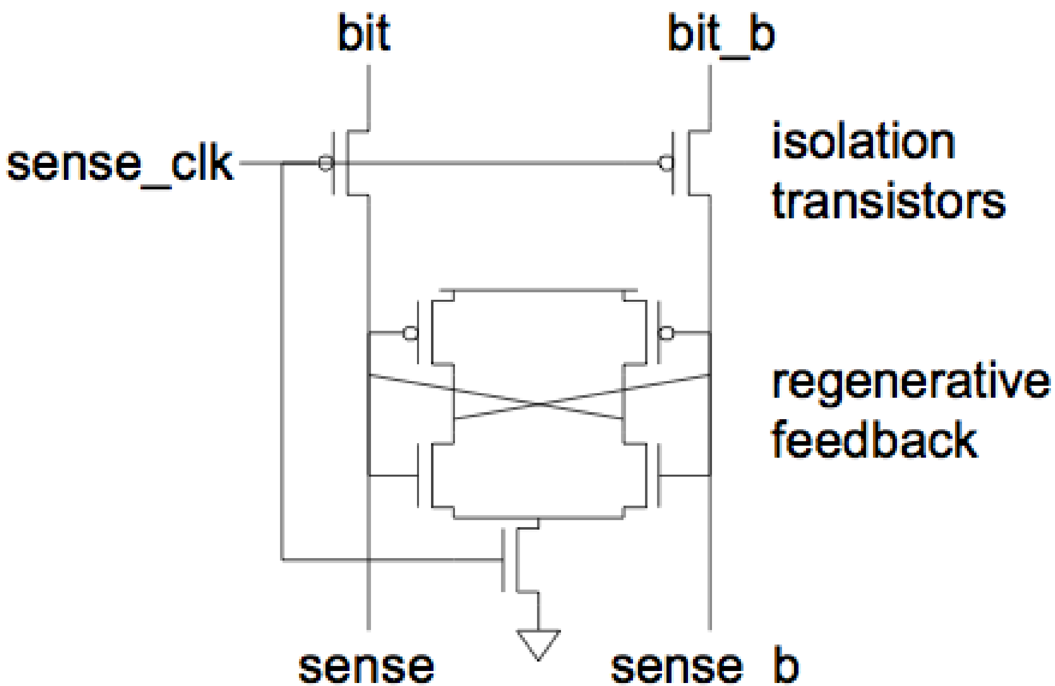
Differential Pair Amp

- 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 during sensing

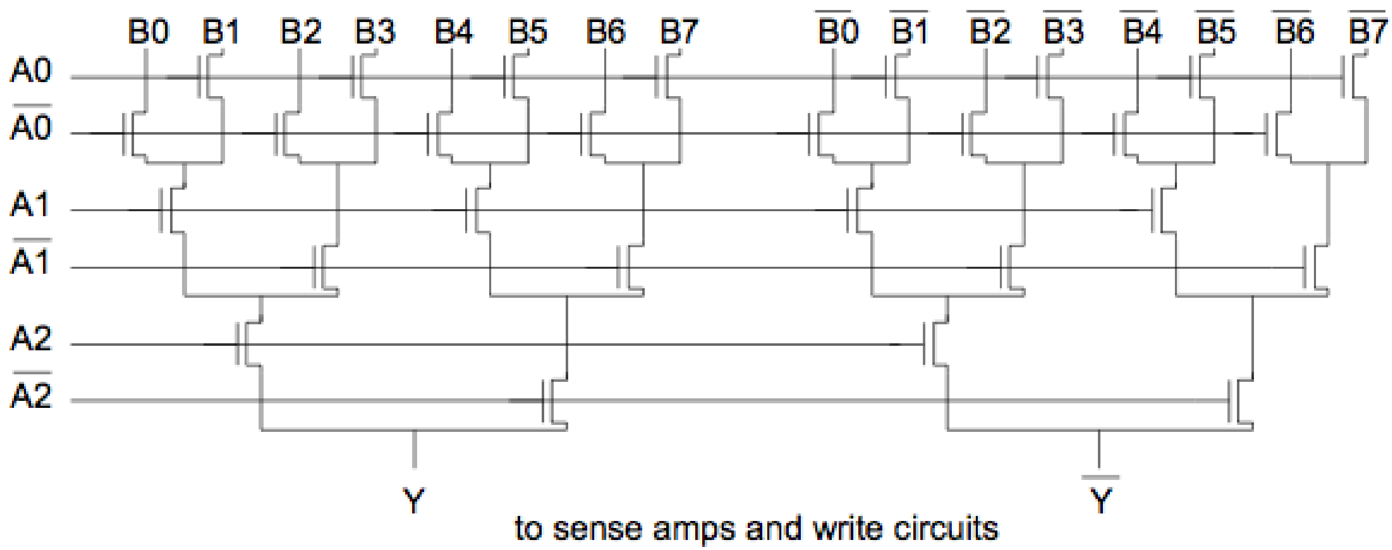


Column Multiplexing

- Recall that array maybe 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

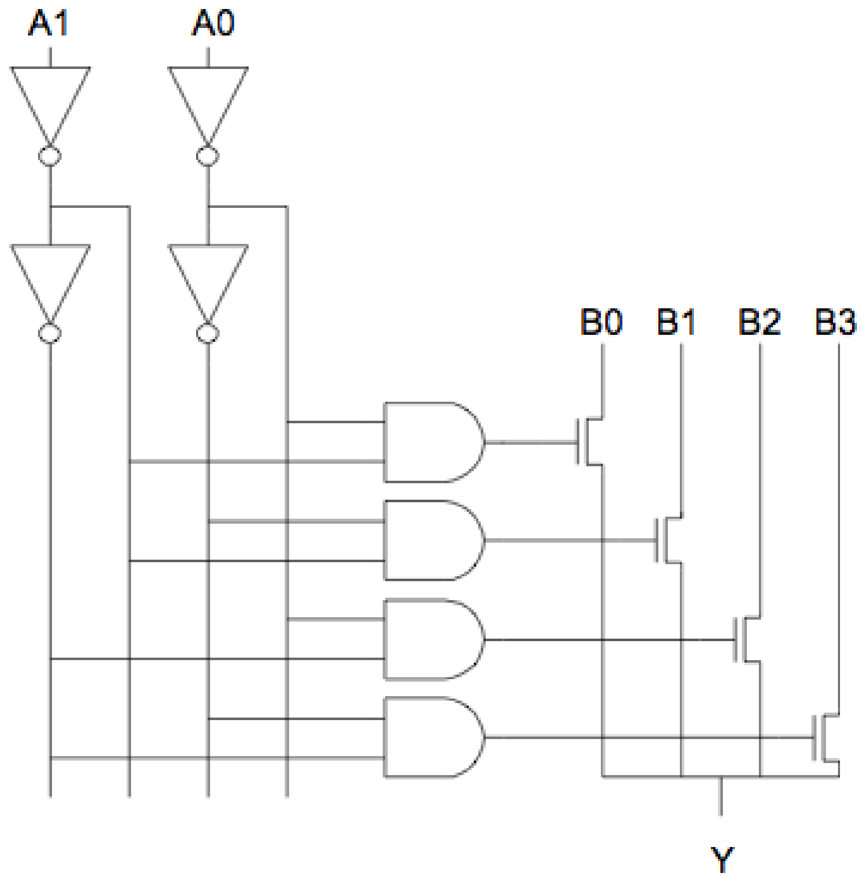
Tree Decoder MUX

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



Single Pass-Gate MUX

- Eliminate series transistors with separate decoder to do the binary-unary decoding.

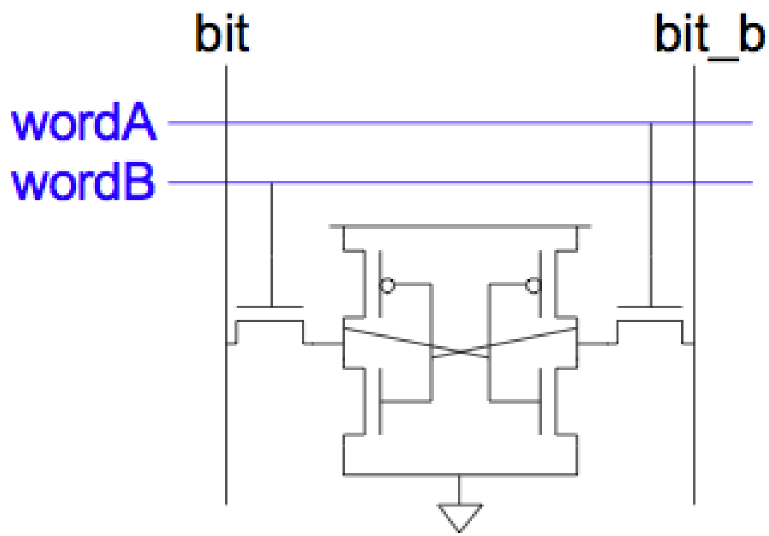


Multiple Ports

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

Dual-Ported SRAM

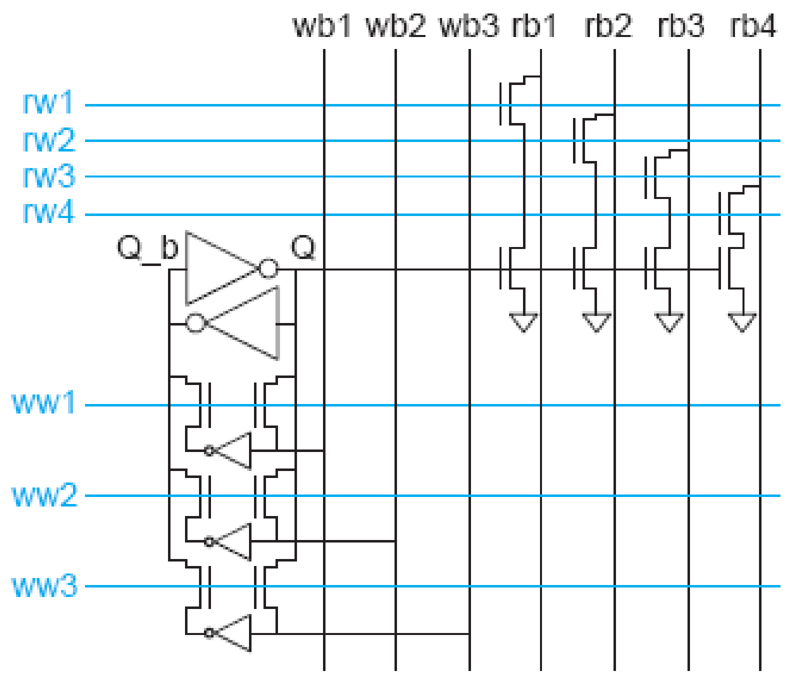
- Simple dual-ported SRAM
  - Two independent single-ended reads
  - Or one differential write



- Do two reads and one write by time multiplexing
  - Read during $\phi 1$, write during $\phi 2$

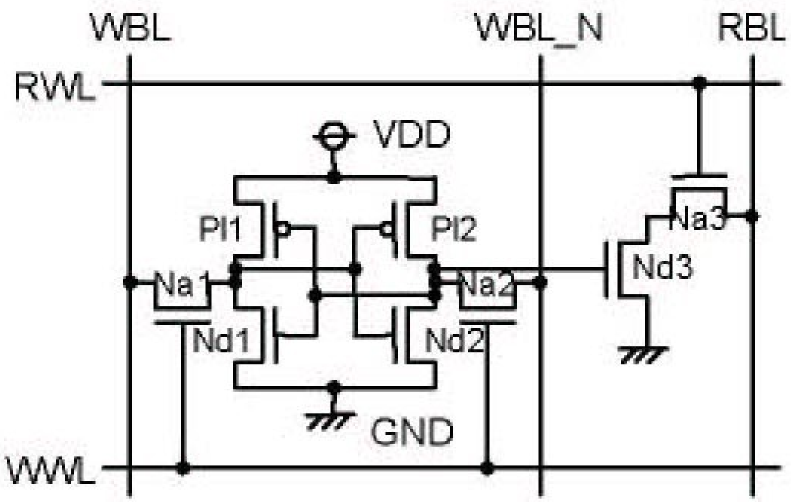
Multi-Ported SRAM

- Add more access transistors hurts read stability
- Multiported SRAM isolates reads from state nodes
- Single-ended design minimizes number of bitlines



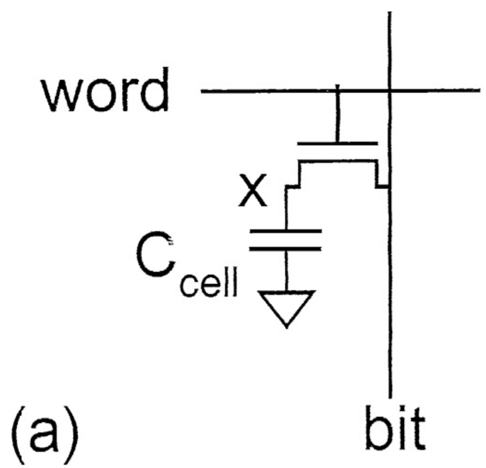
Future Trend

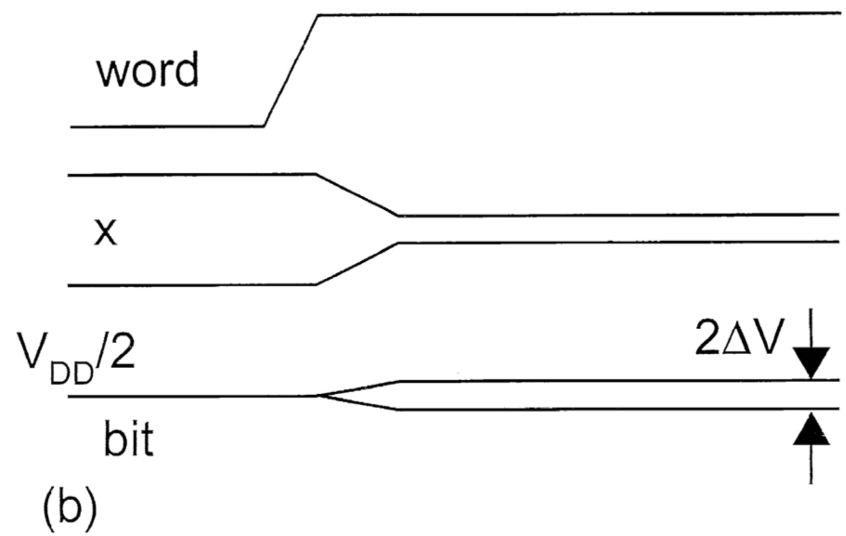
- Use 8T SRAM cell in 65nm-32nm technology
  - Morita VLSI Circuits Symp. 2007.
  - Separate read port from write port
  - Can support Vdd down to 0.42V



Dynamic RAM

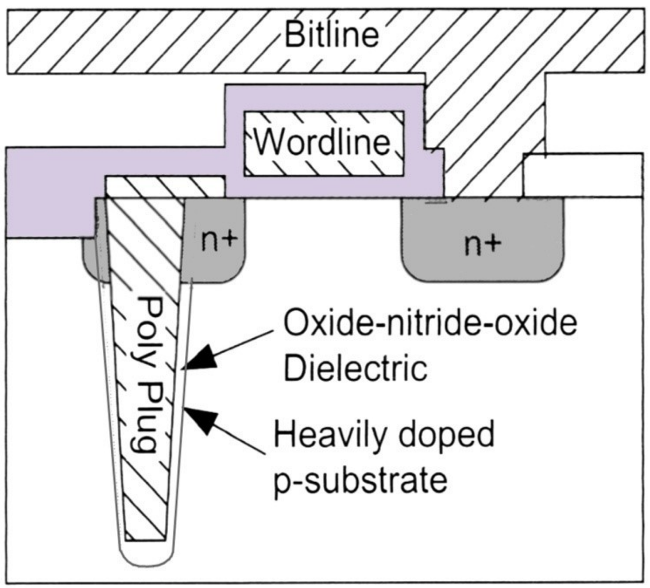
- Cell charge is dynamic and cell needs periodically refreshing
- Destructive reading and voltage swing is very small.
- Bit lines are precharge to $V_{DD}/2$.
- $\Delta V = (V_{DD}/2)[C_{cell}/(C_{cell} + C_{bit})]$

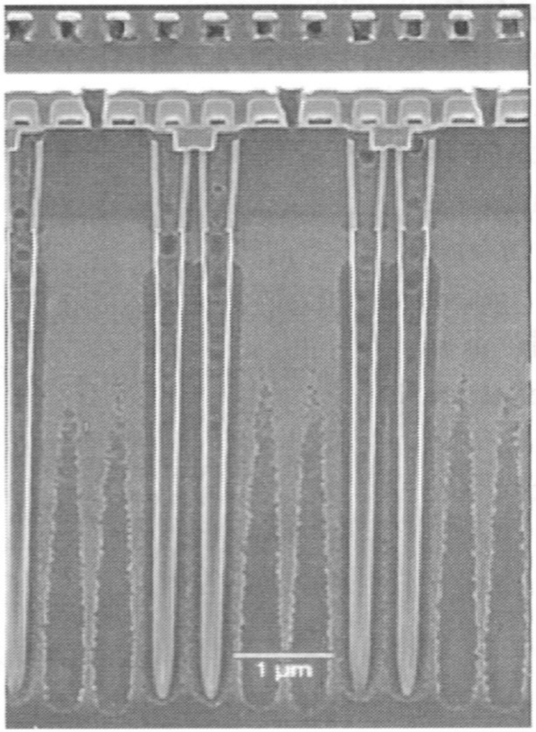




DRAM

- Trench capacitor can drastically decrease the cell area
- Schematic and scanning electron microscope (SEM) photo.



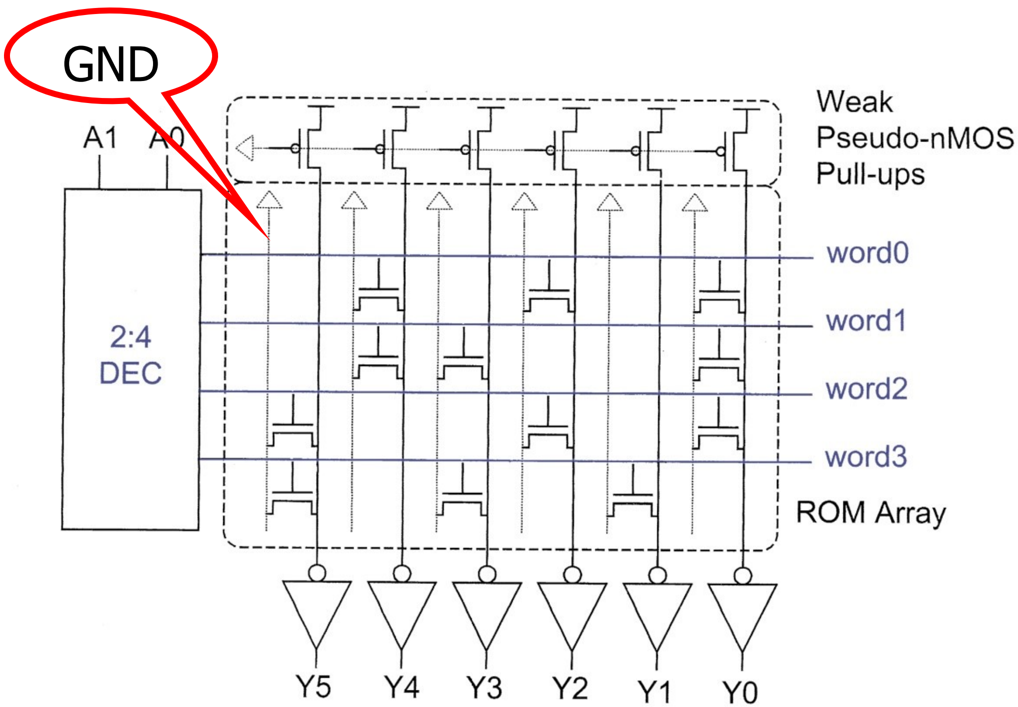


Read-Only Memories

- Read-only memories are nonvolatile
  - Retain their contents when power is removed
- Mask-programmed ROMs use one transistor per bit
  - Presence or absence determines 1 or 0

ROM Example

- 4-word x 6-bit ROM
  - Represented with dot diagram
  - Dots indicate 1's in ROM

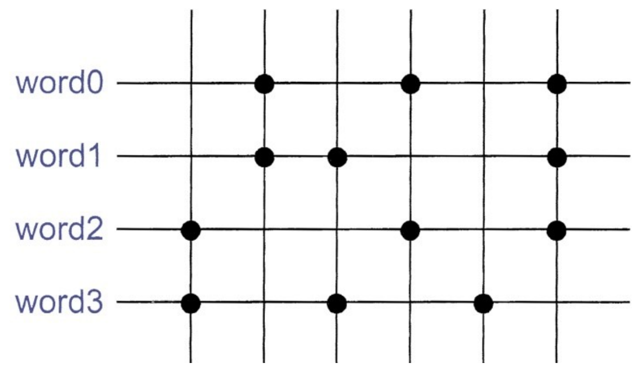


Word 0: 010101

Word 1: 011001

Word 2: 100101

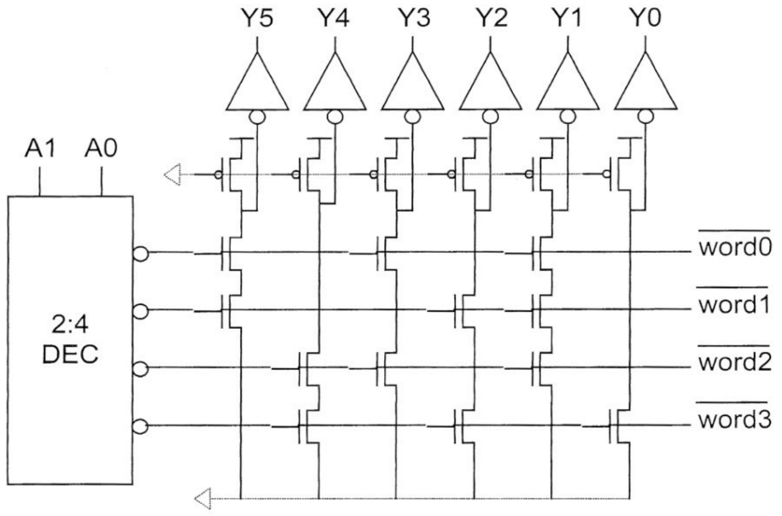
Word 3: 101010

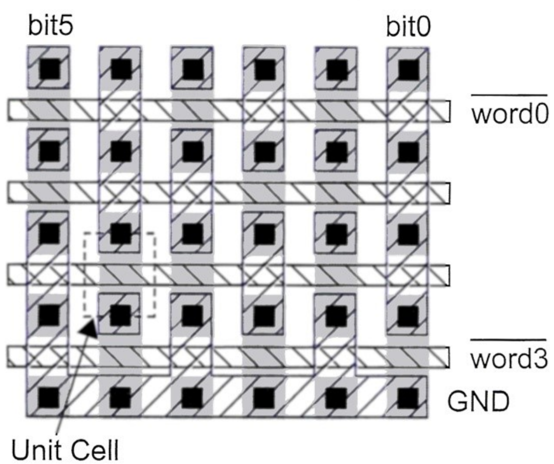


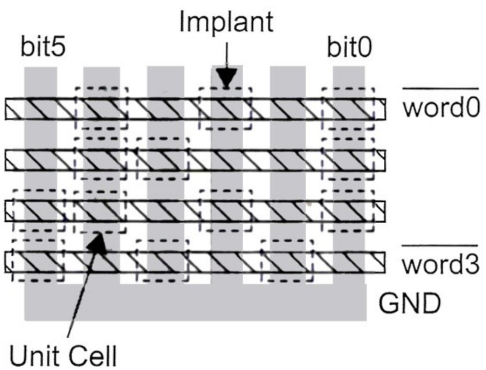
Looks like 6 4-input pseudo-nMOS NORs

NAND ROMs

- Smaller area due to no ground lines.
- Cell area can be further reduced by eliminating the contacts and using an extra implant step that makes the Vth negative and turns on the transistor permanently.

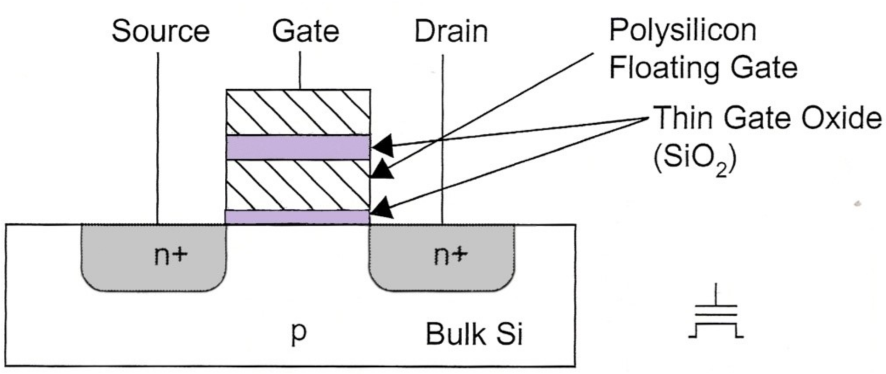






PROMs and EPROMs

- Programmable ROMs
  - Build array with pull-down transistors at every site
  - Burn out fuses to disable unwanted transistors; one-time programmable (OTP).
- Erasable Programmable ROMs
  - Use floating gate to turn off unwanted transistors
  - Negative charges are drawn into the isolated floating gate, which effectively increases the threshold voltage and turns off the transistor.
  - EPROM, EEPROM, Flash

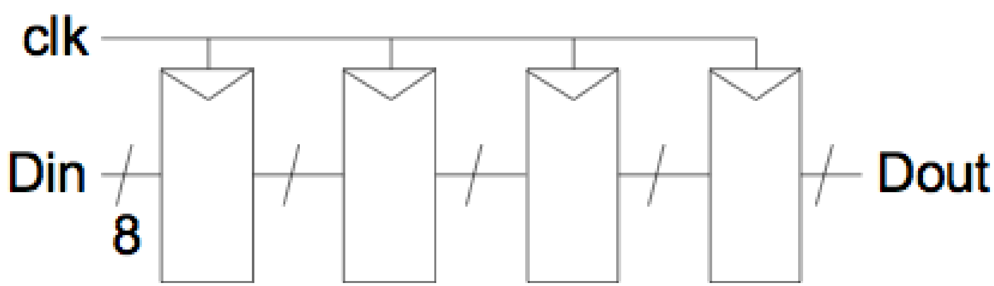


Serial Access Memories

- Serial access memories do not use an address
  - Shift registers
  - Serial In Parallel Out (SIPO)
  - Parallel In Serial Out (PISO)
  - Queues (FIFO, LIFO)

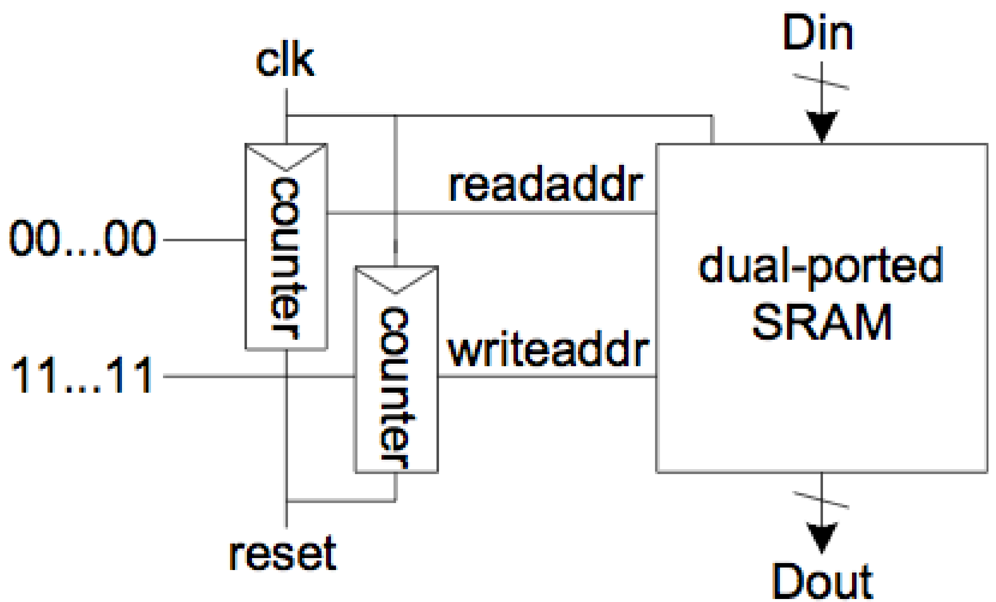
Shift Register

- Shift register store and delay data
- Simple design: cascade of registers
  - Watch your hold times, i.e., the signal may travel too fast.



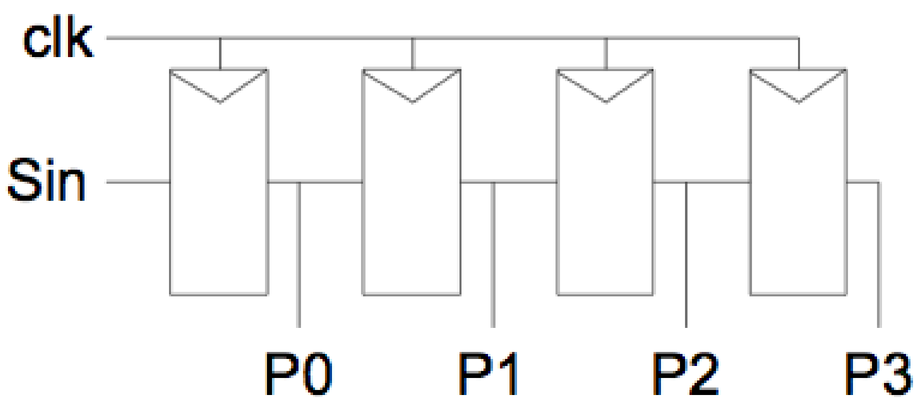
Dense Shift Registers

- Flip-flops are not very area-efficient
- For large shift registers, keep data in SRAM instead
- Move read/write pointers to RAM rather than data
  - Initialize read address to first entry, write to last
  - Increment address on each cycle
  - Can use ring counters to replace counters and address decoders



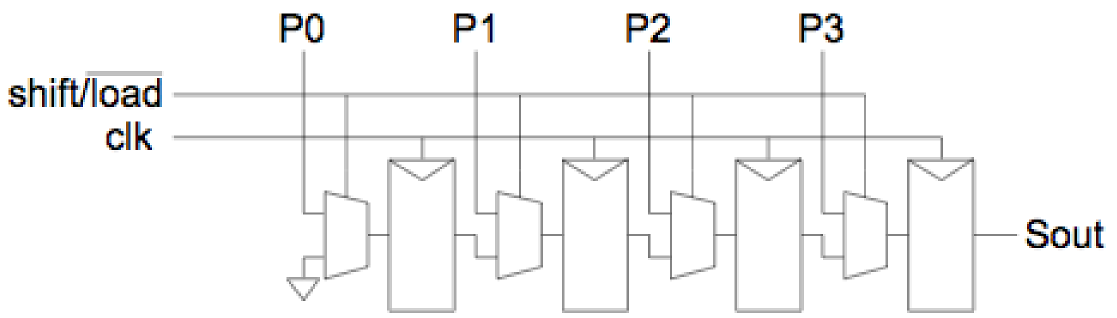
Serial In Parallel Out

- 1-bit shift register reads in serial data
  - After N steps, presents N-bit parallel output



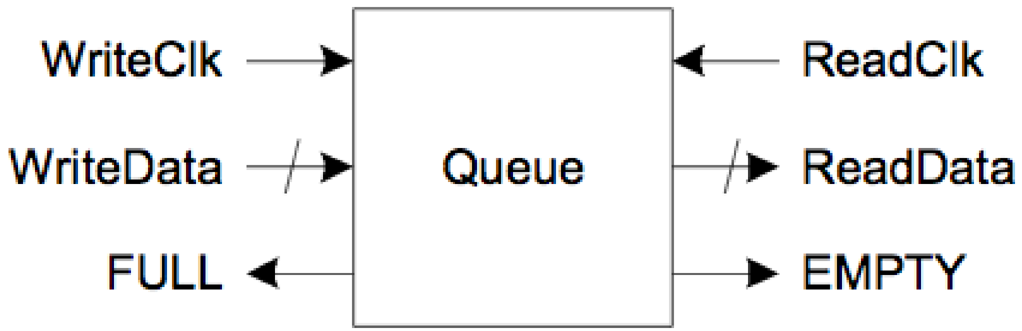
Parallel In Serial Out

- Load all N bits in parallel when shift=0
  - Then shift one bit out per cycle



Queues

- Queues allow data to be read and written at different rates
- Read and write each use their own clock and data
- Queue indicates whether it is full or empty
- Build with SRAM and read/write counters (pointers)

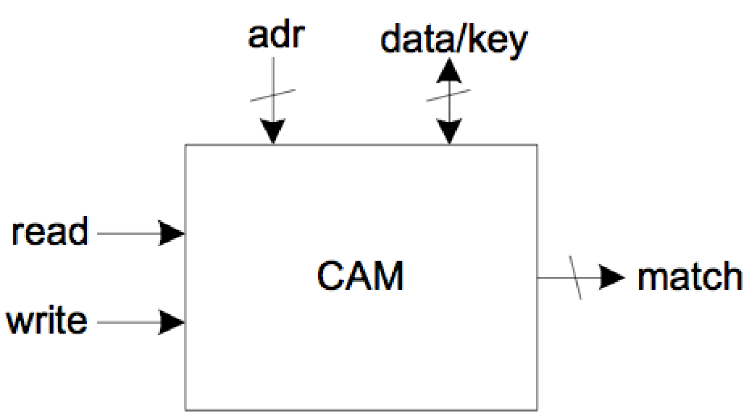


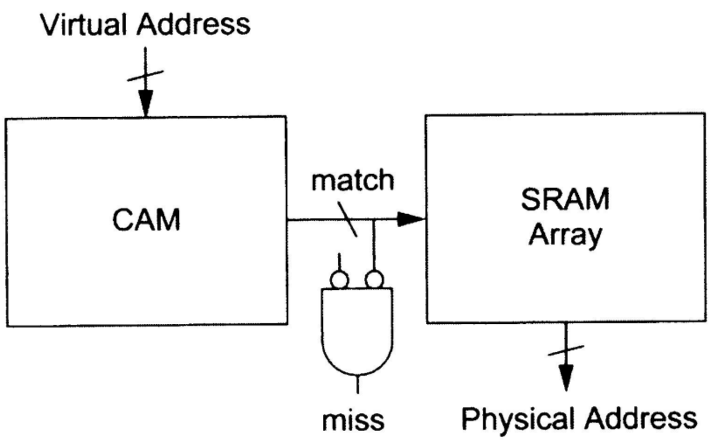
FIFO, LIFO Queues

- First In First Out (FIFO)
  - Initialize read and write pointers to first element
  - Queue is EMPTY
  - On write, increment write pointer
  - If write almost catches read, Queue is FULL
  - On read, increment read pointer
- Last In First Out (LIFO)
  - Also called a stack
  - Use a single stack pointer for read and write

Content Addressable Memory

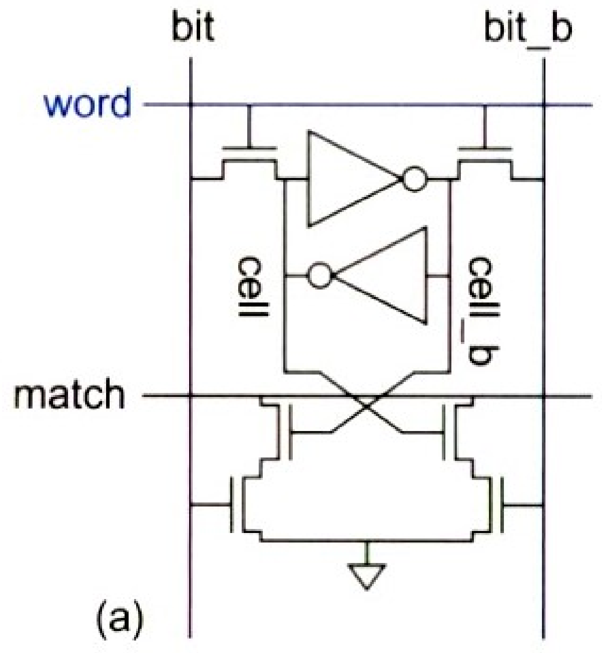
- Read and write memory as usual
- A content addressable memory takes in a data word and compare it with stored words and generates a match signal, which can be used to address a RAM. This is how the translation look-aside buffer works in a cache.





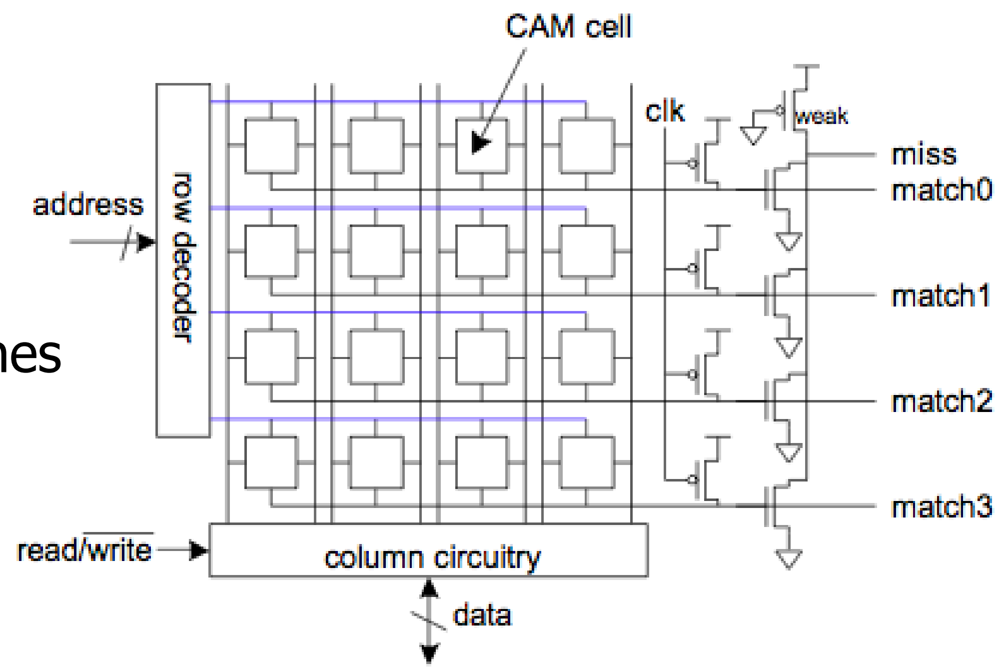
10-T CAM Cell

- Add four match transistors to 6T SRAM
- Match line will be pulled low when ‘cell’ and ‘bit’ mismatch.



CAM Cell Operation

- Read and write like ordinary SRAM
- For matching:
  - Leave wordline low
  - Precharge matchlines
  - Place key on bitlines
  - Matchlines evaluate
- Miss line
  - Pseudo-nMOS NOR match lines
  - Goes high if no words match

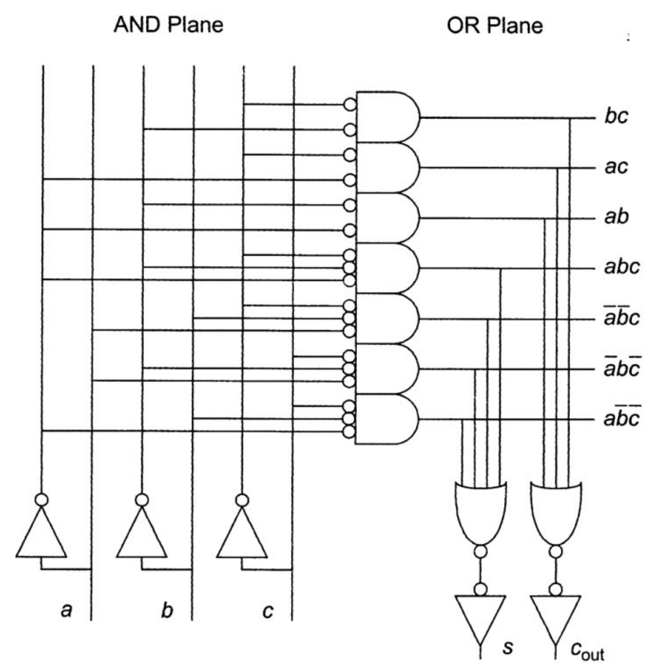


PLAs

- A Programmable Logic Array performs any function in sum-of-products form
- Literals: inputs & complements
- Products/Minterms: AND of literals
- Outputs: OR of Minterms
- Example: Full Adder

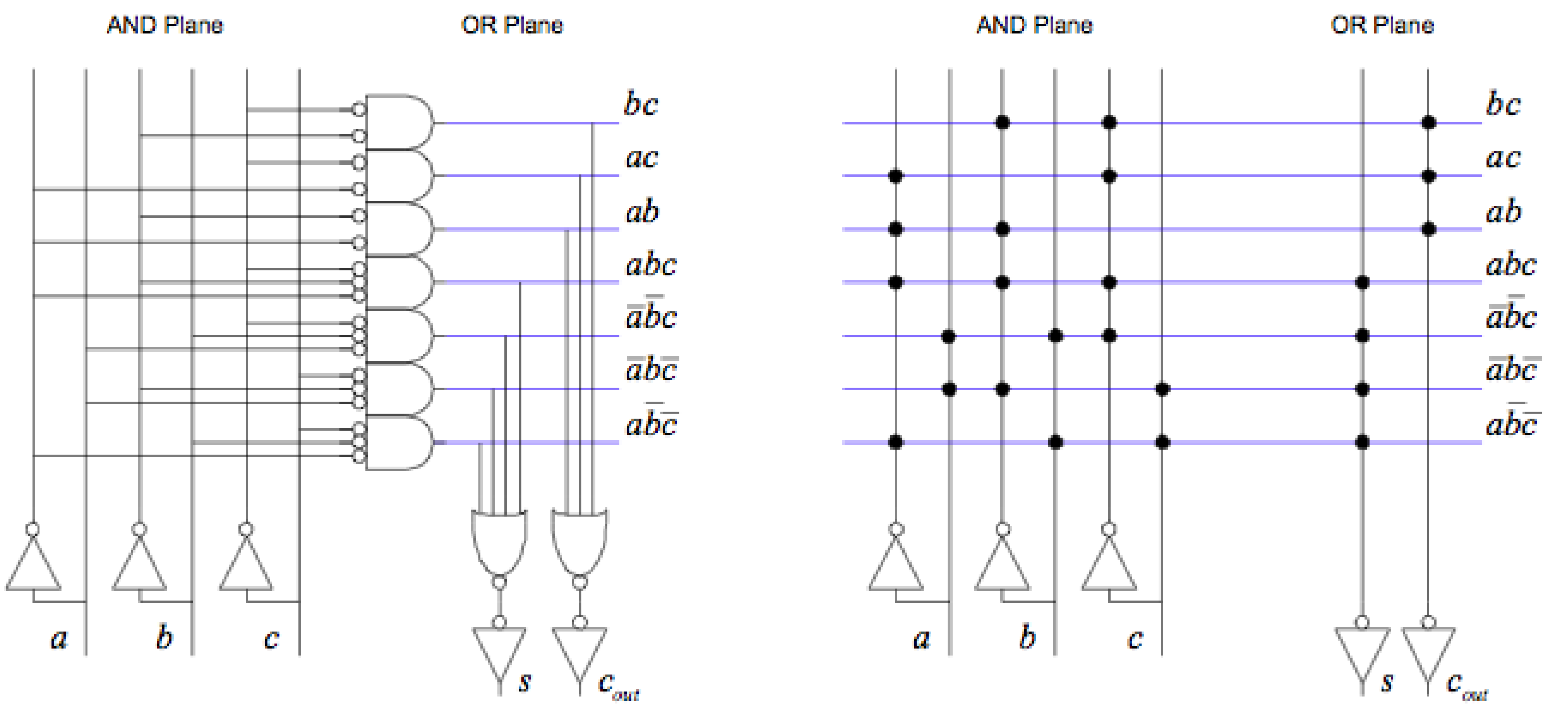
$$s = a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c + abc$$

$$c_{\text{out}} = ab + bc + ac$$



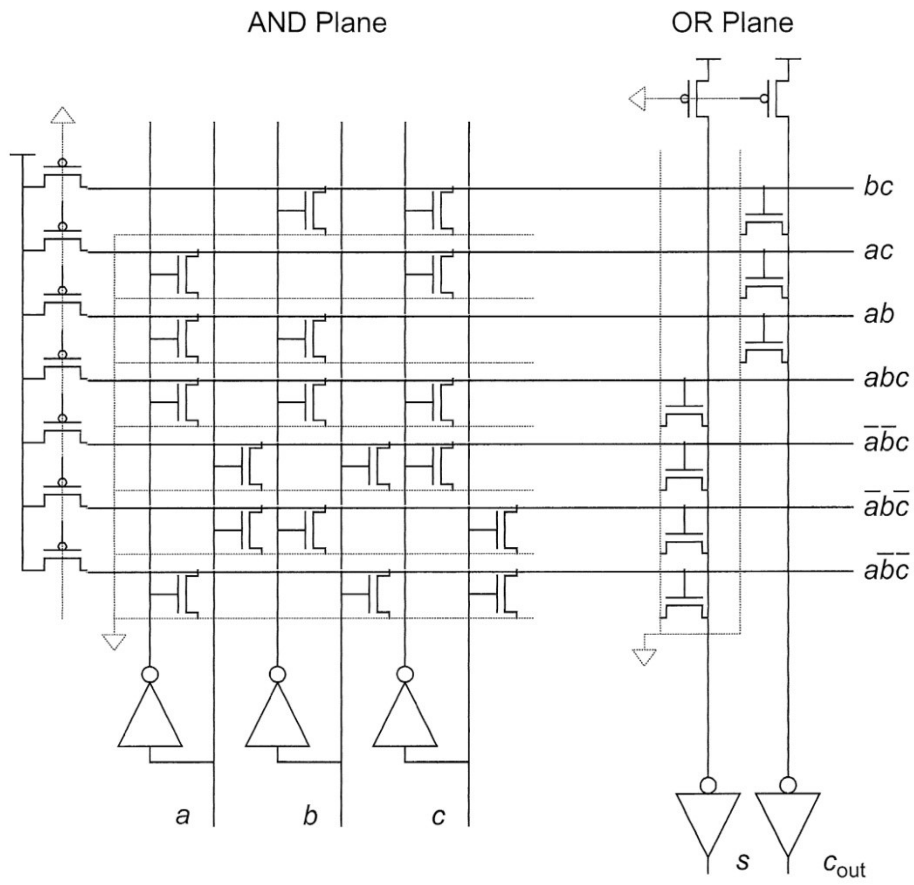
NOR-NOR PLAs

- ANDs and ORs are not very efficient in CMOS
- Dynamic or Pseudo-nMOS NORs are very efficient
- Use DeMorgan's Law to convert to all NORs



PLA Schematic

- Pseudo-NMOS version
- Dynamic and self-timed version can be faster.



PLAs vs. ROMs

- The OR plane of the PLA is like the ROM array
- The AND plane of the PLA is like the ROM decoder
- PLAs are more flexible than ROMs
  - No need to have 2^n rows for n inputs
  - Only generate the minterms that are needed
  - Take advantage of logic simplification