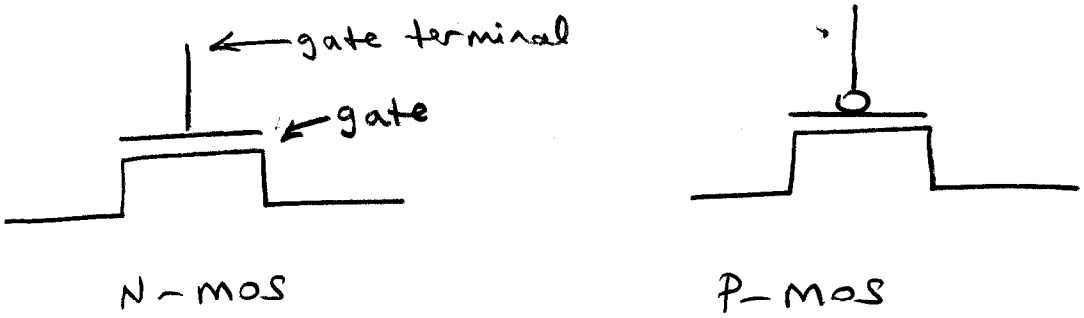
LECTURE : CMOS TECHNOLOGY

AIM: Examine how the logic gates (and memory) are implemented using transistors.

Transistor : basically a switch

CMOS: Complementary MOS (used in integrated circuits)

Two types of transistors

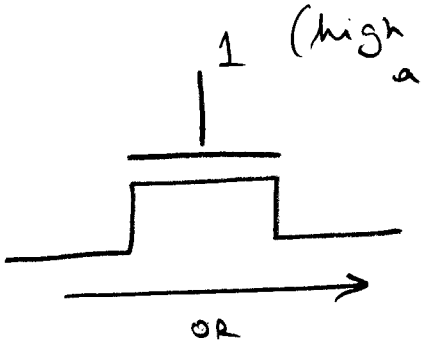


N-MOS

P-MOS

Basic operation :

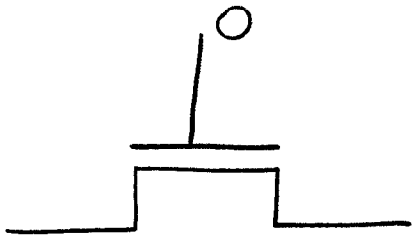
N-mos



OR

passes the signal (current) through

111 equivalent to

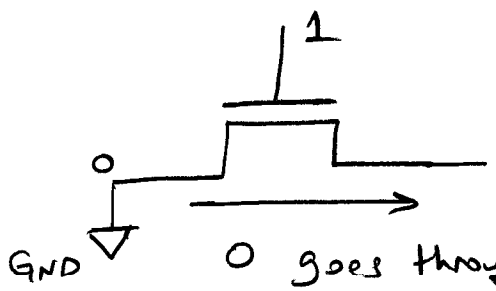


X

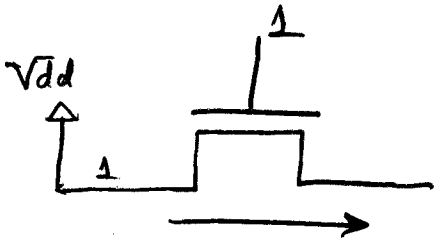
signal blocked

111 equivalent to

N-MOS passes "0" well:



0 goes through without degradation

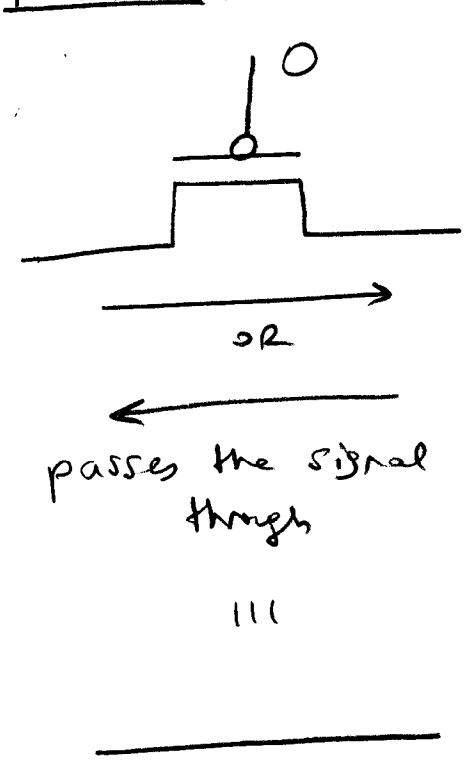


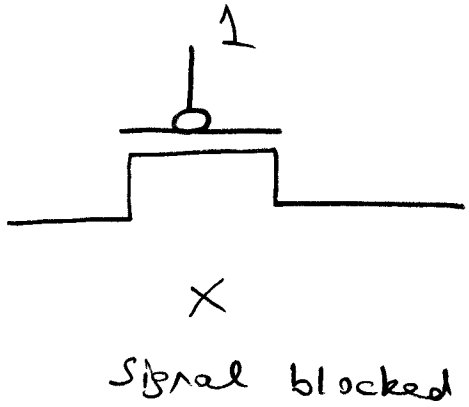
1 is degraded at output

(does not pass "1" well)

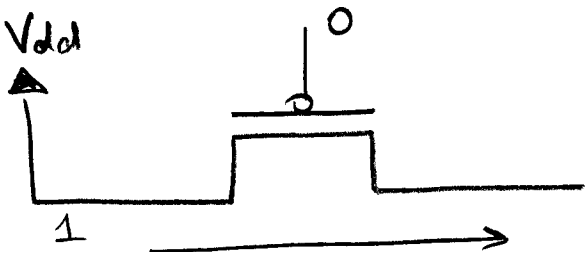
Basic operation :

P-MOS :

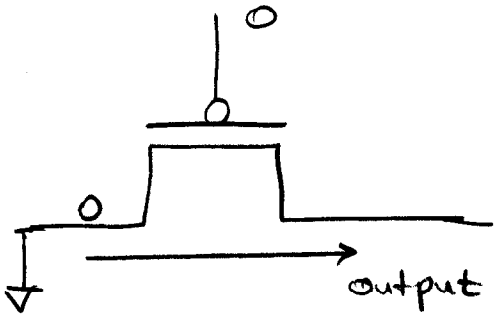




P_mos passes "1" well :



output level
not degraded

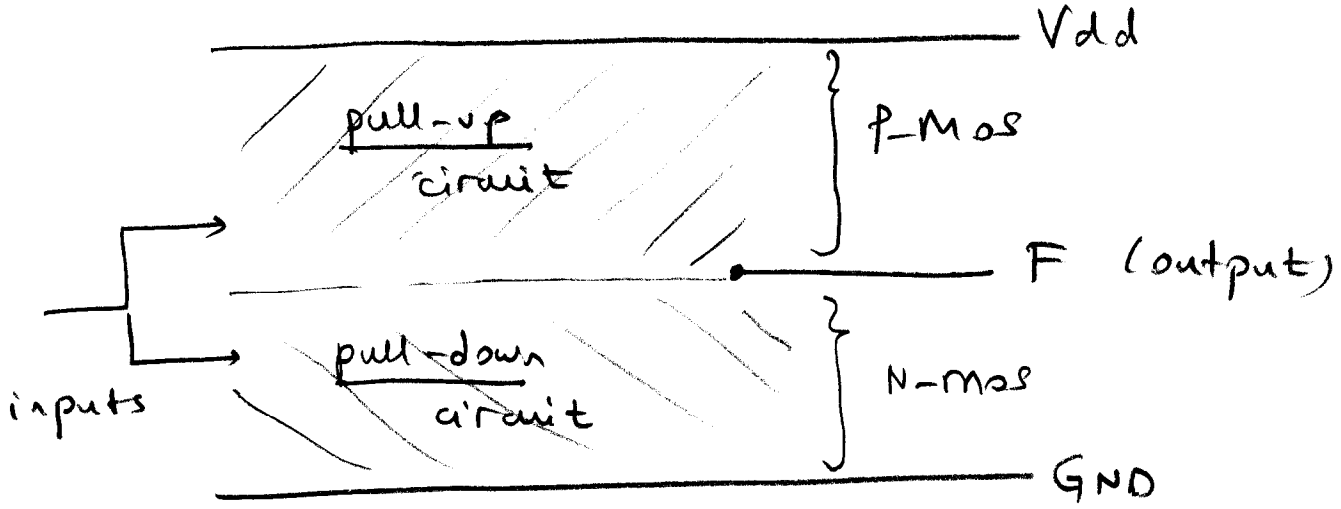


output 0
degraded

(does not pass 0 well)

- ALL Boolean functions can be implemented in CMOS.

Main idea: (chip layout)

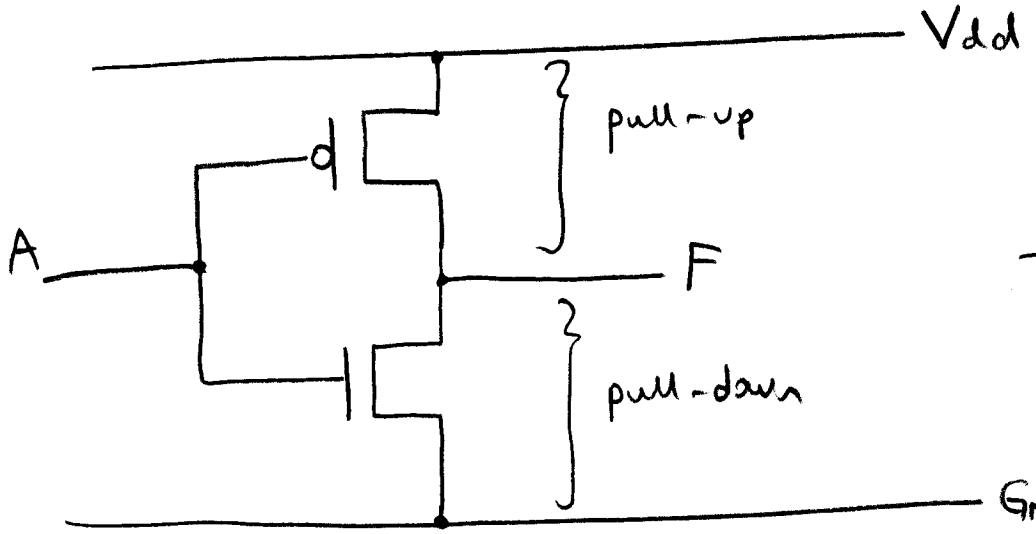


- Pull-up circuit implements the "1"'s (ON-TERMS) of F, using only P-MOS transistors
- Pull-down circuit implements the "0"'s (OFF-TERMS) of F, using only N-MOS transistors.

Example 1: Implement an inverter in CMOS

Solution: $F = A'$ ⟶ implement in P-MOS

$F' = A$ ⟶ (OFF terms) implement in N-MOS



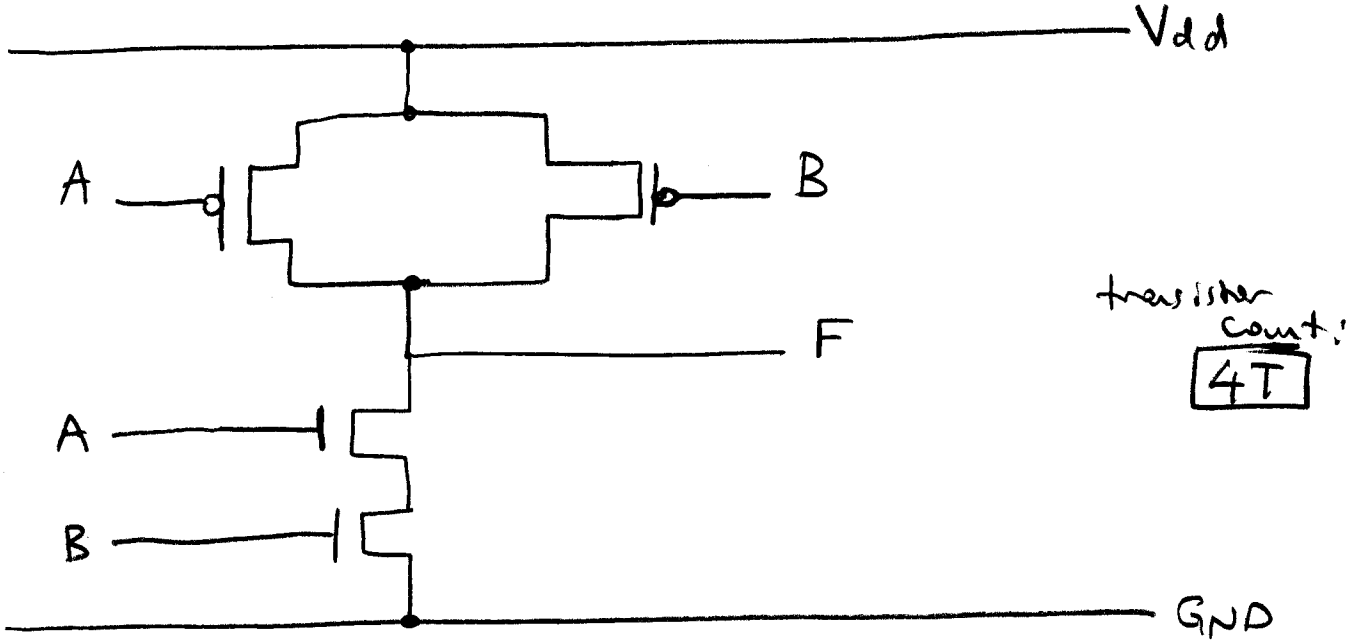
Transistor count:

2T

Example 2: Implement a NAND gate in CMOS.

Solution: $F = (A \cdot B)' = A' + B'$ → OR: implemented as a parallel network of switches

$F' = A \cdot B$ ← series network of switches.



transistor count: 4T

Key Observations:

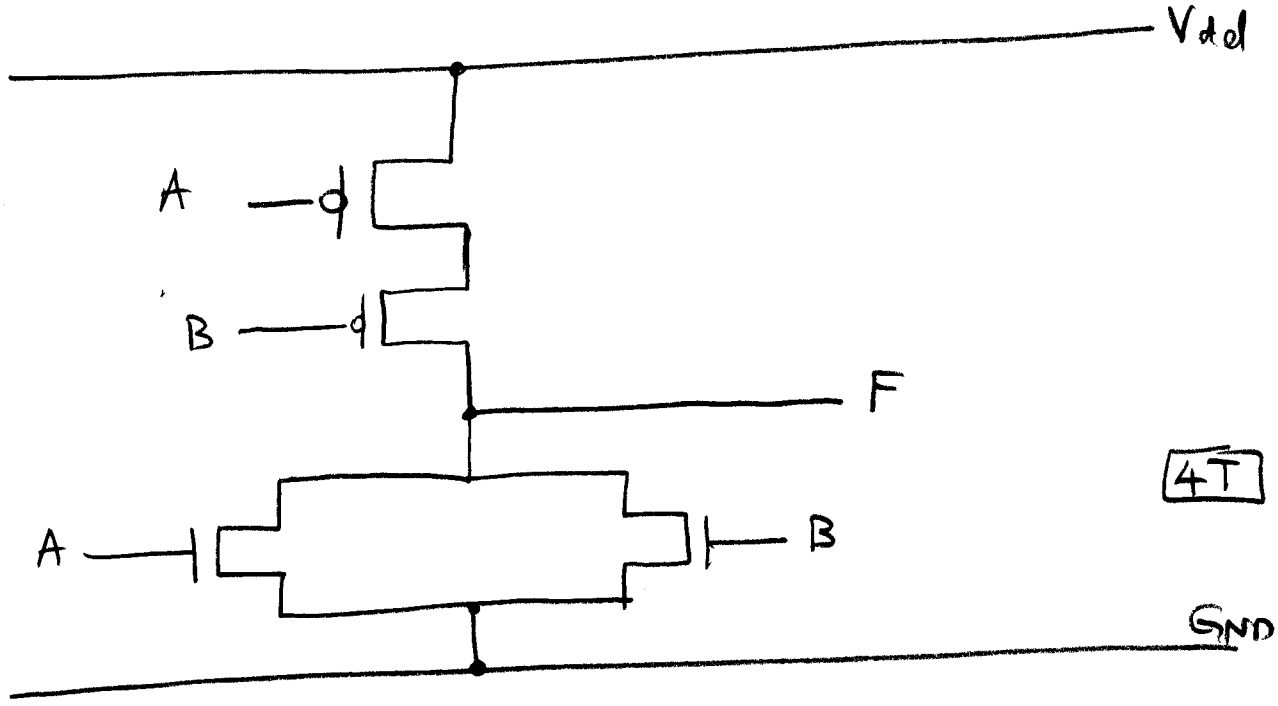
1 - CMOS does not leave the Boolean fn output floating. F is connected either to Vdd or GND. (network is either pulling up to Vdd or down to GND).

2 - Energy is consumed only when the output is switching from Vdd → GND or GND → Vdd. After circuit stabilizes, no static energy consumption. → CMOS used widely for low-power designs

Example 3 : Implement a NOR gate in CMOS

Solution : $F = (A+B)' = A' \cdot B'$

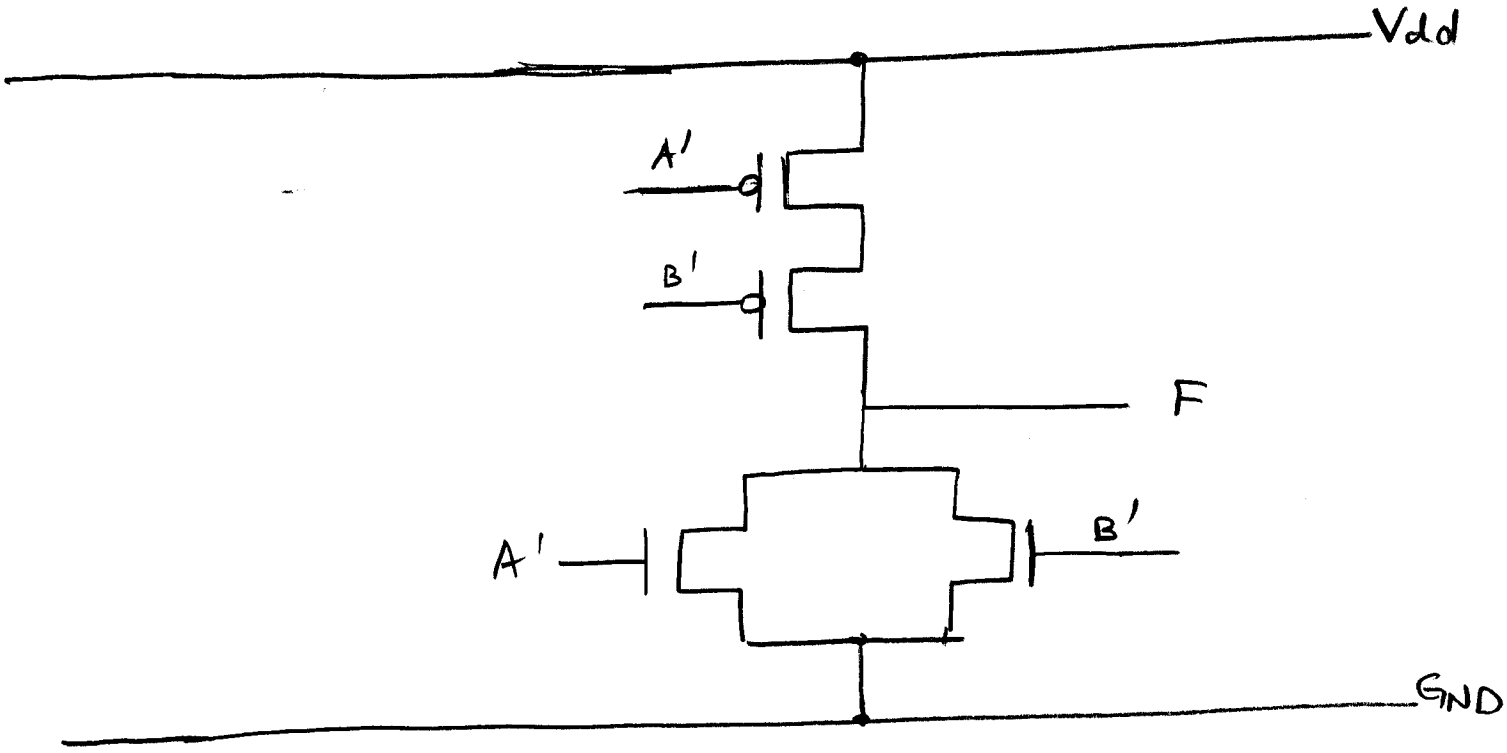
$F' = A + B$



Example 4 : Implement an AND gate in CMOS.

Solution : $F = A \cdot B$

$F' = (A \cdot B)' = A' + B'$



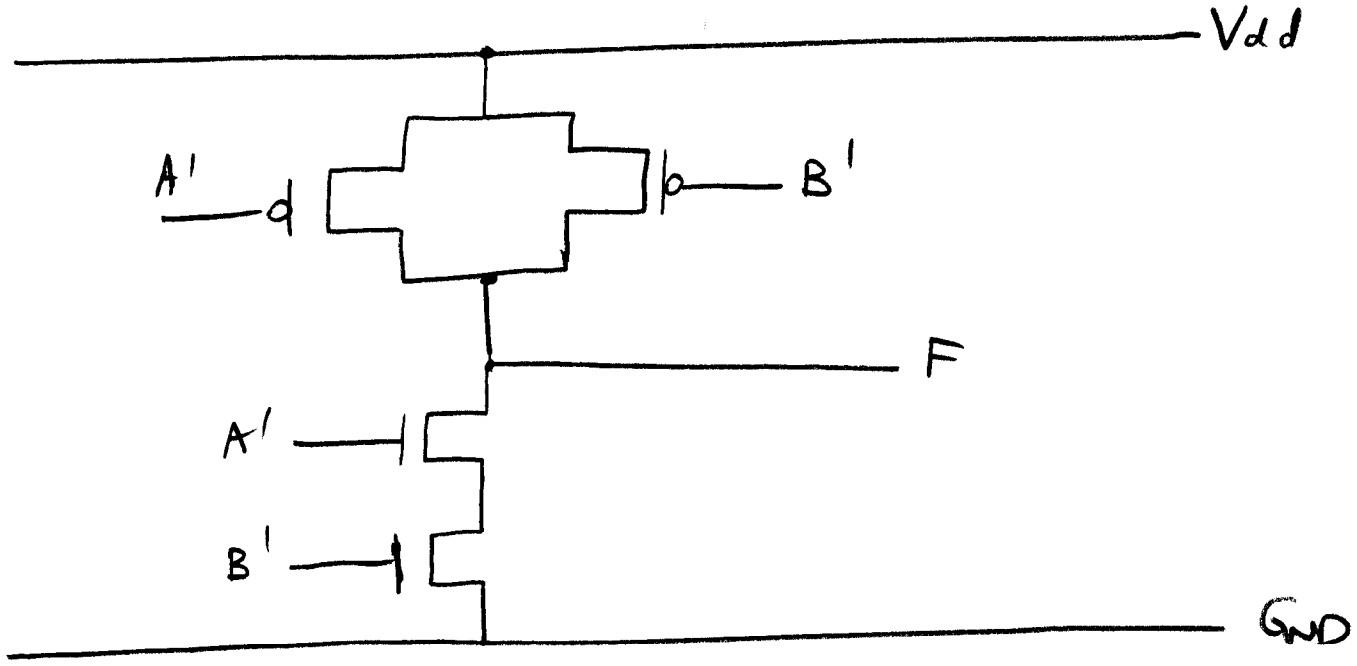
Transistor count = $4 + \underbrace{2}_{\text{to invert A}} + \underbrace{2}_{\text{invert B}} = \boxed{8T}$

[Contrast with NAND gate: 4T]

Example 5: Implement an OR gate in CMOS

Solution: $F = A + B$

$F' = (A+B)' = A' \cdot B'$



Transistor count = $4 + 2 + 2 = \boxed{8T}$

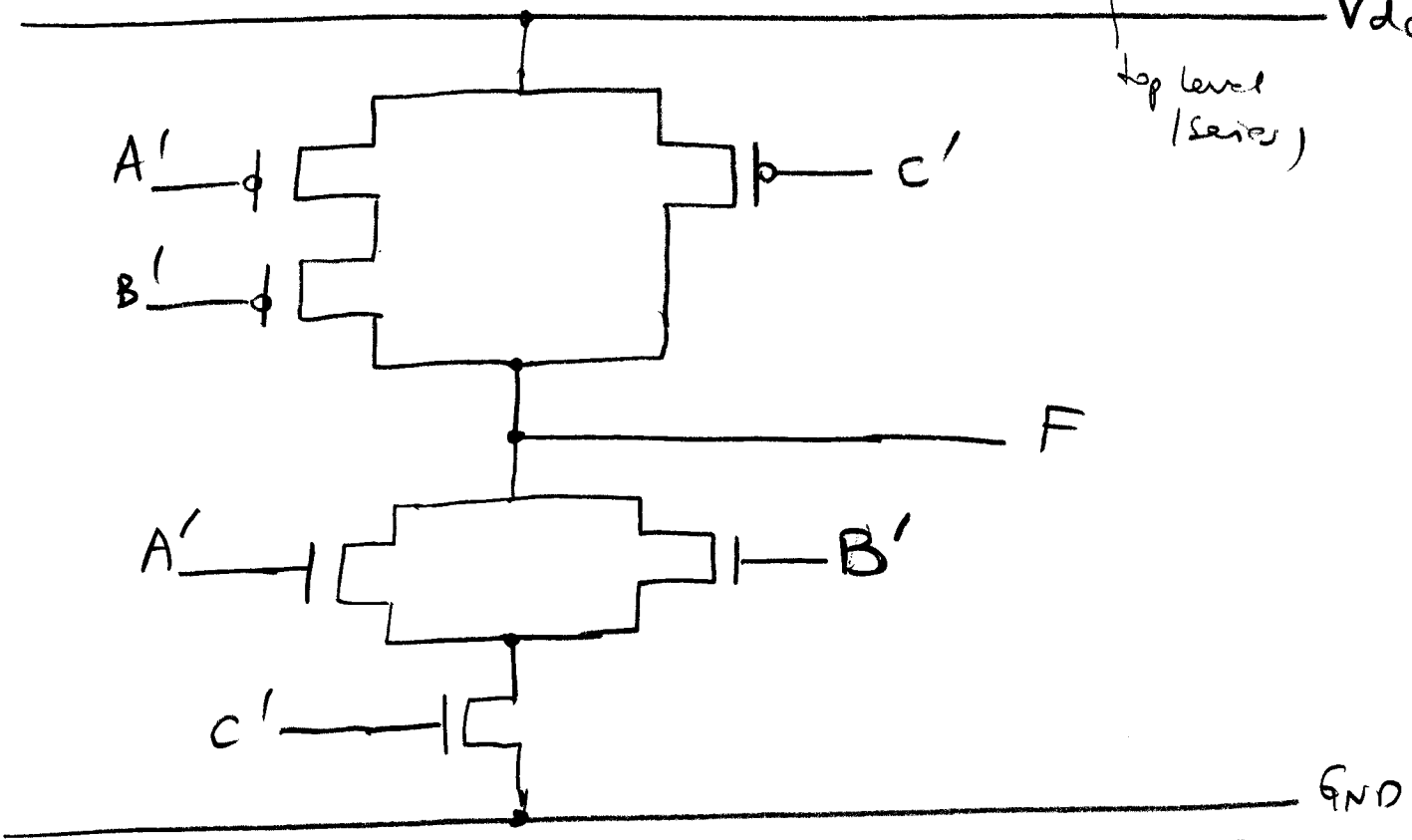
Example 6: Implement $F = A \cdot B + C$

[This shows the general procedure to implement any Boolean function.]

Solution: $F = (A \cdot B) + C$] pull-up (top level)

$F' = (A \cdot B + C)' = (A \cdot B)' \cdot C'$

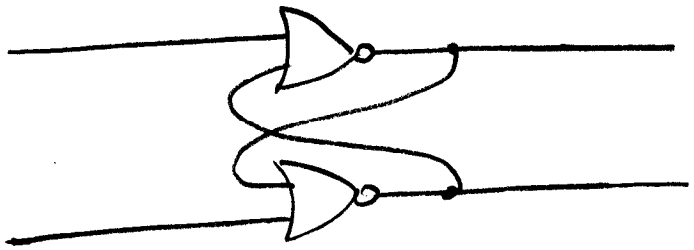
$= (A' + B') \cdot C'$] pull-down (top level / series)



Transistor count = $6 + \underbrace{2}_{A'} + \underbrace{2}_{B'} + \underbrace{2}_{C'} =$ **12T**

Memory elements in CMOS

- Cross-coupled NOR gates:



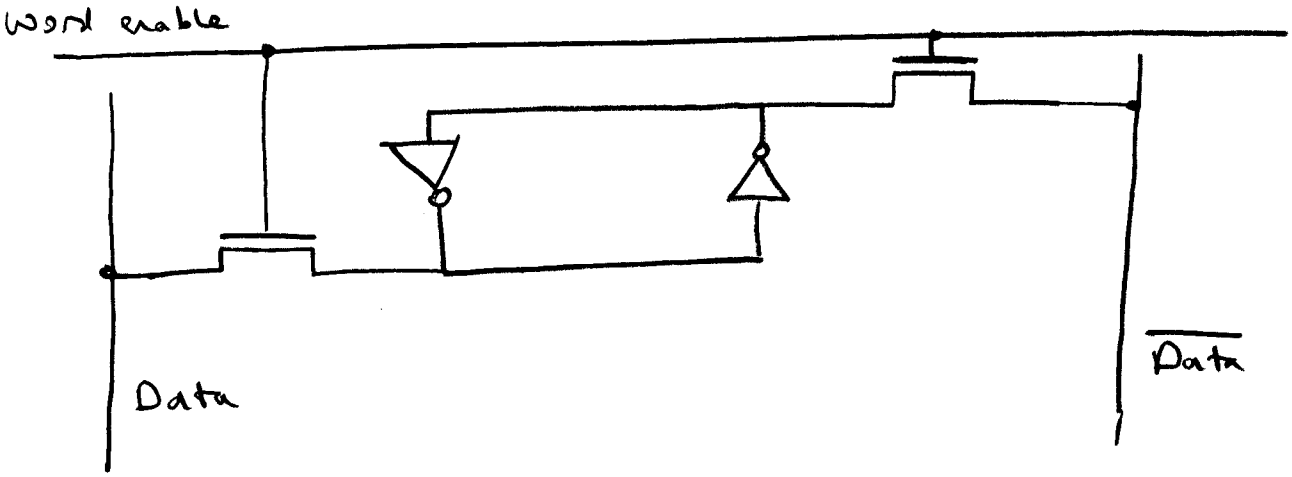
- Transistor count in CMOS = $2 \times \underbrace{(4T)}_{\text{per NOR gate}} = \boxed{8T}$

- For integrated memories, we need high integration density (i.e. a high # of bits stored per unit area of silicon).
- ∴ Must reduce the area of a "cell" (a storage unit that holds 1 bit).
- Area: depends on # transistors + wiring complexity.

S-RAM cell (6 T S-RAM cell)

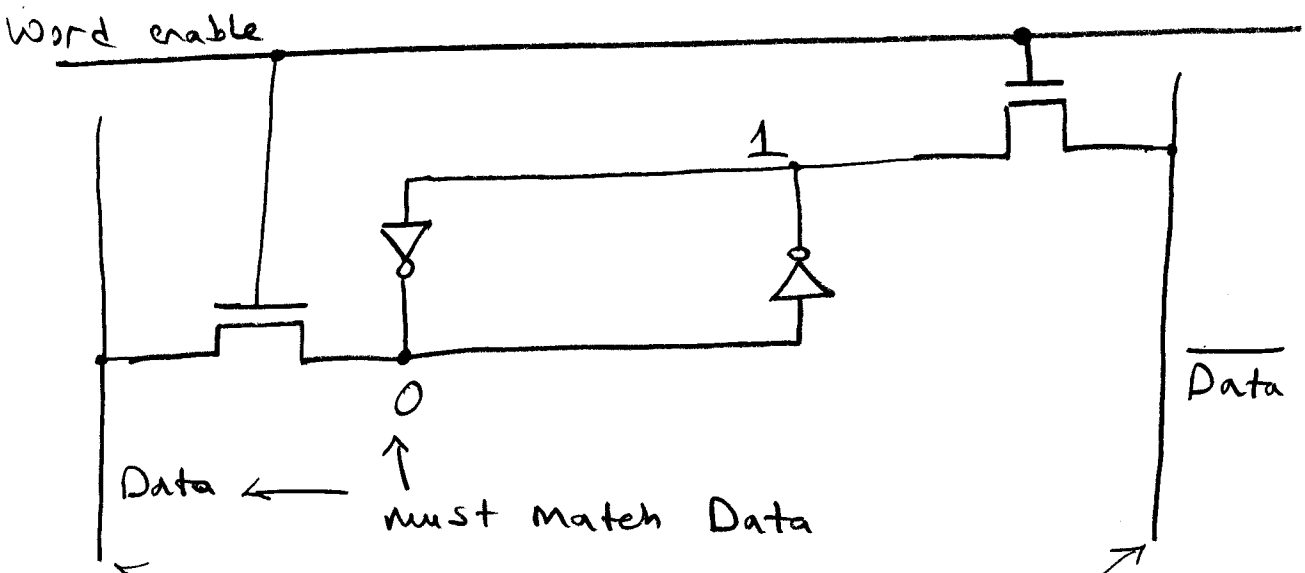
→ Random Access memory

[Static RAM: Static means that Vdd, GND used to reinforce the value stored inside. ∴ will hold the bit as long as power is on.]



- Holds 1 bit.
- cross coupled inverters
- (2T/inverter) * 2 inverters + 2 n-Mos = 6T

Operation of S-RAM cell :



- Assume the cell is initially storing a "0".

Sense amplifier detects difference.
(Reading)

READING from the S-RAM cell:

① Both Data and $\overline{\text{Data}}$ are pre-charged to 1.
- (Word Enable originally low)

② Word Enable put high.
- The "0" in the cell pulls the Data line to 0 (since N-Mos passes 0 well.)

③ The sense amplifier senses the difference between Data and $\overline{\text{Data}}$.
- If Data has voltage < (sufficiently less than) $\overline{\text{Data}}$, then a value of "0" declared.
- Otherwise a "1" is declared.

WRITE : to the S-RAM cell.

Example: Writing a "0"

① Data = 0 and $\overline{Data}$ = 1 (driven by external circuitry)

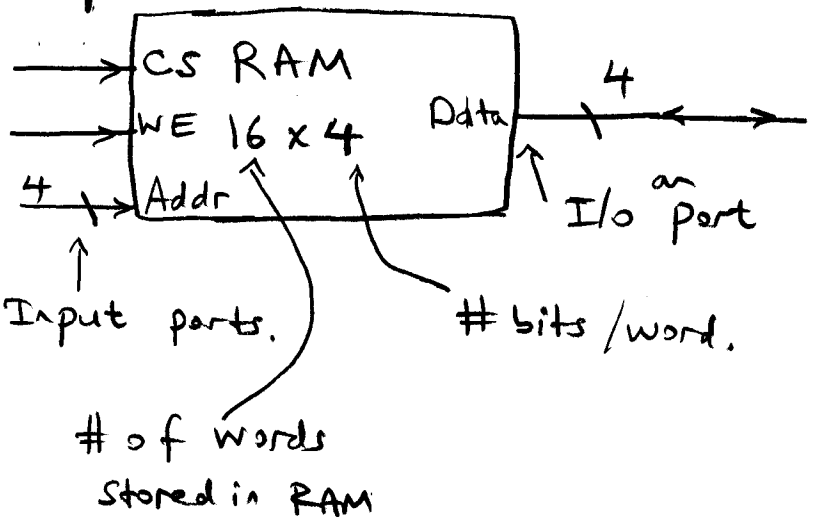
② Word Enable = 1

- The external circuitry drives these values into the cell.

③ Word Enable = 0.

S-RAM chip:

Example:



CS: Chip Select (if low, cannot read or write)

WE: Write Enable (if CS == 1, and WE == 1 write
CS == 1, and WE == 0 read.)

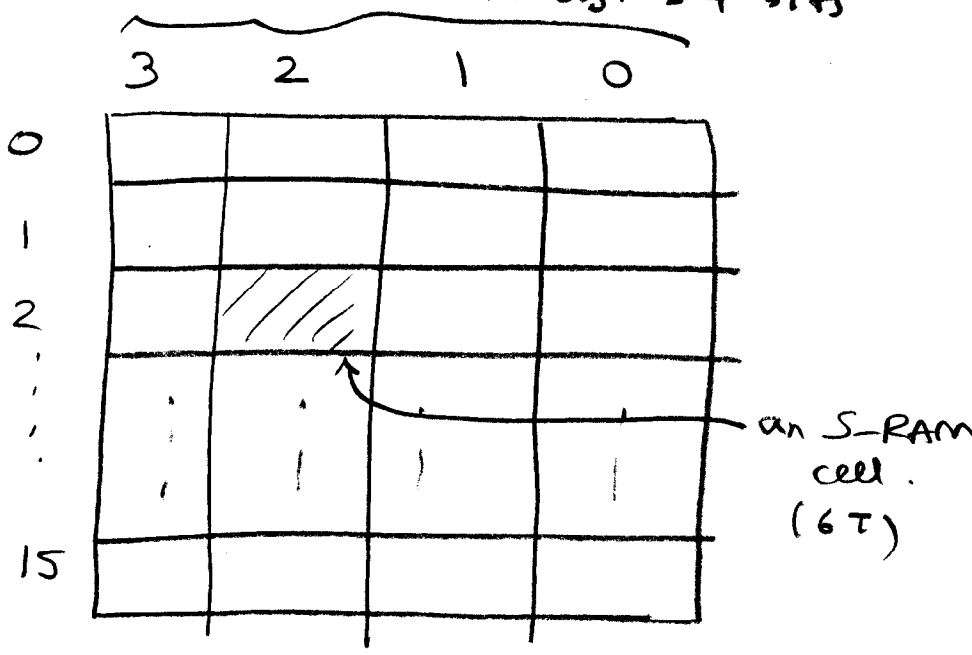
Address: Need $\log_2(16) = 4$ bits to address each word.

Conceptual organization:

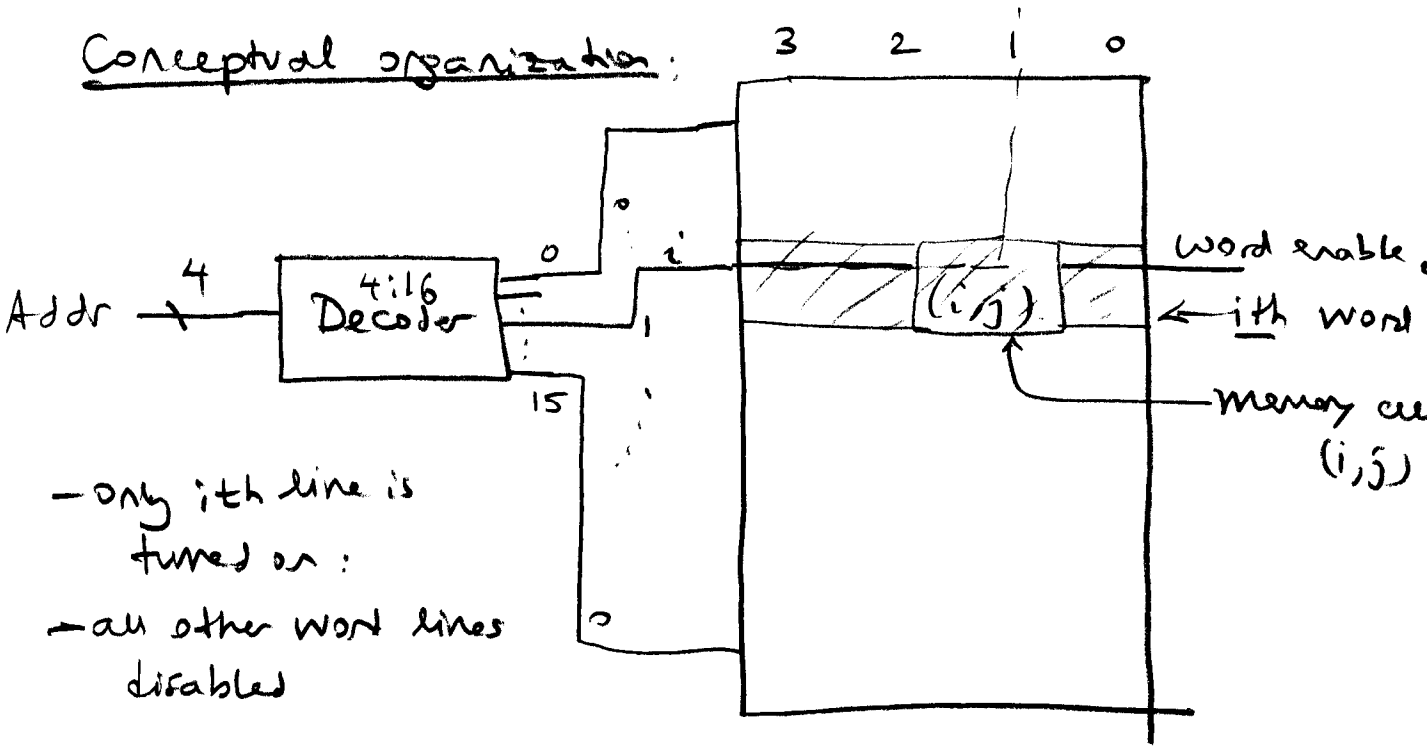
16 x 4 RAM:

- $16 \cdot 4 = \underline{64}$ cells in total

- Idea: Address one of the 16 words (to write in) or (read out) an entire word.

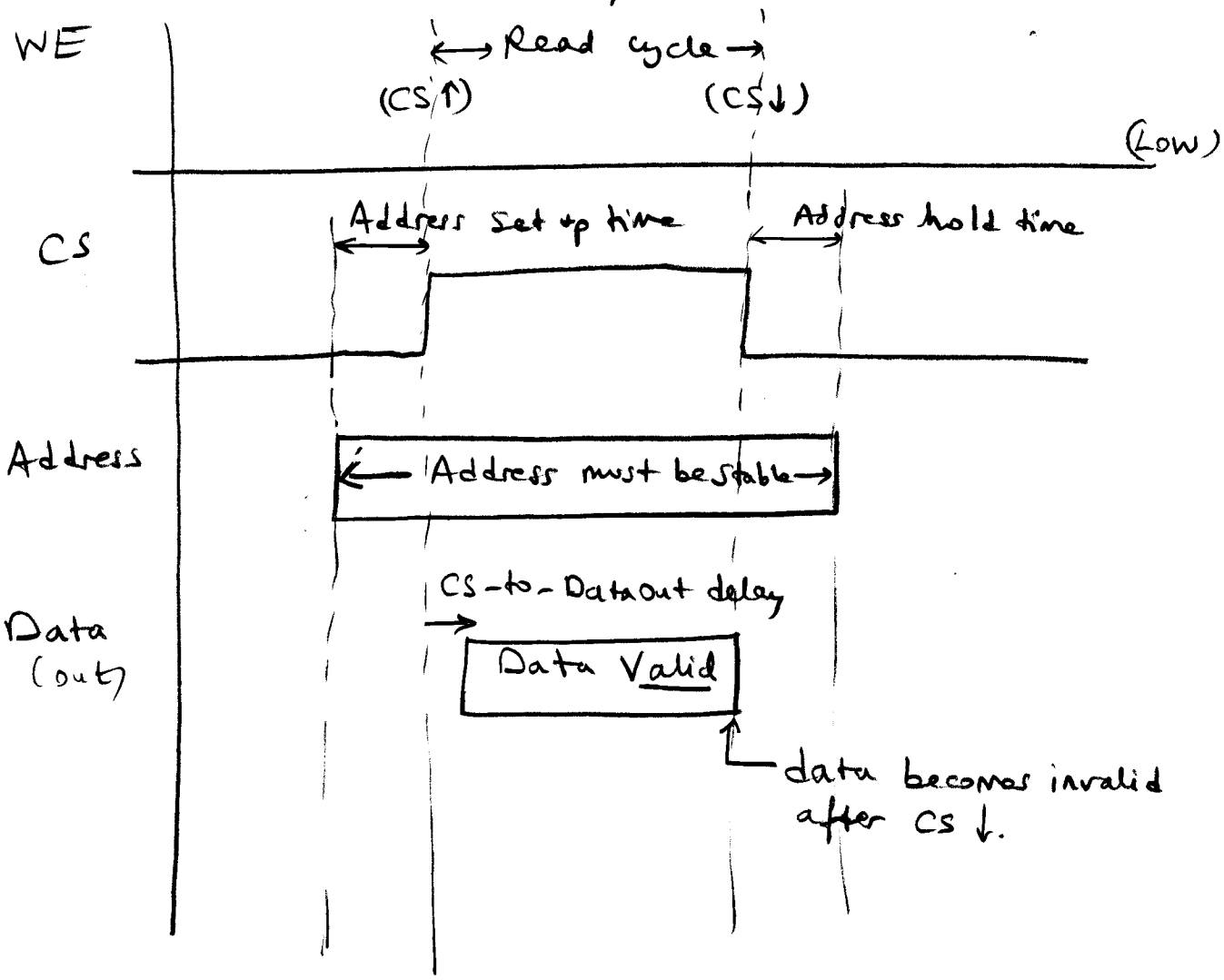


Conceptual organization:

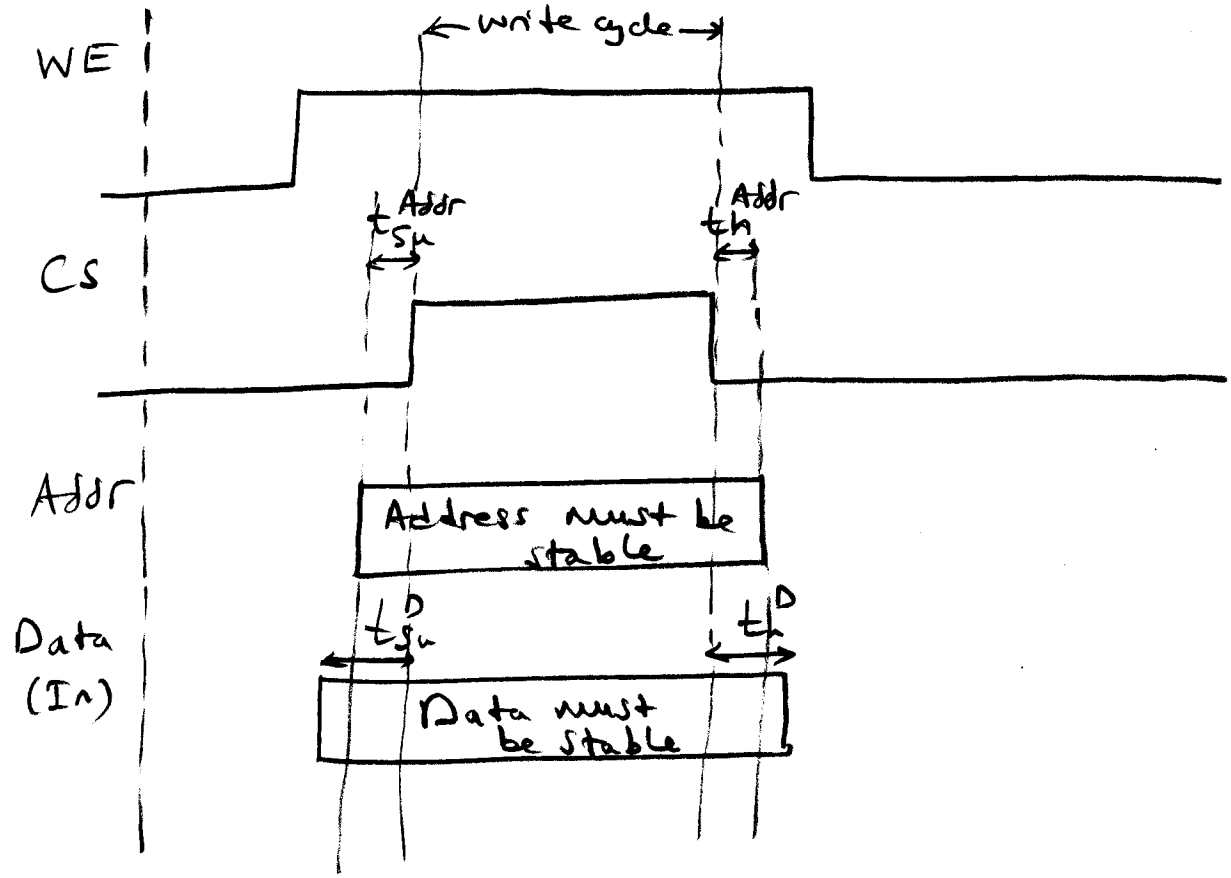


- only ith line is turned on:
- all other word lines disabled

SRAM Read Cycle Sequencing :



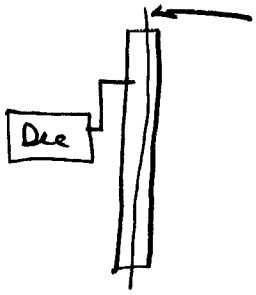
S-RAM WRITE CYCLE SEQUENCING



Wider S-RAM's have a more efficient internal organization

Example: 1024 x 4 SRAM.

not a good idea to:

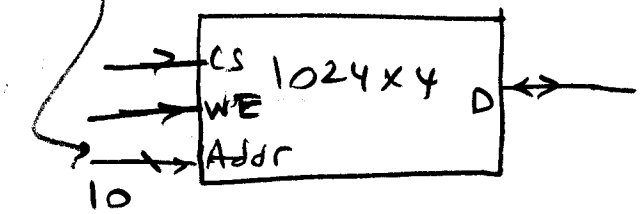


data lines would have to be v. long

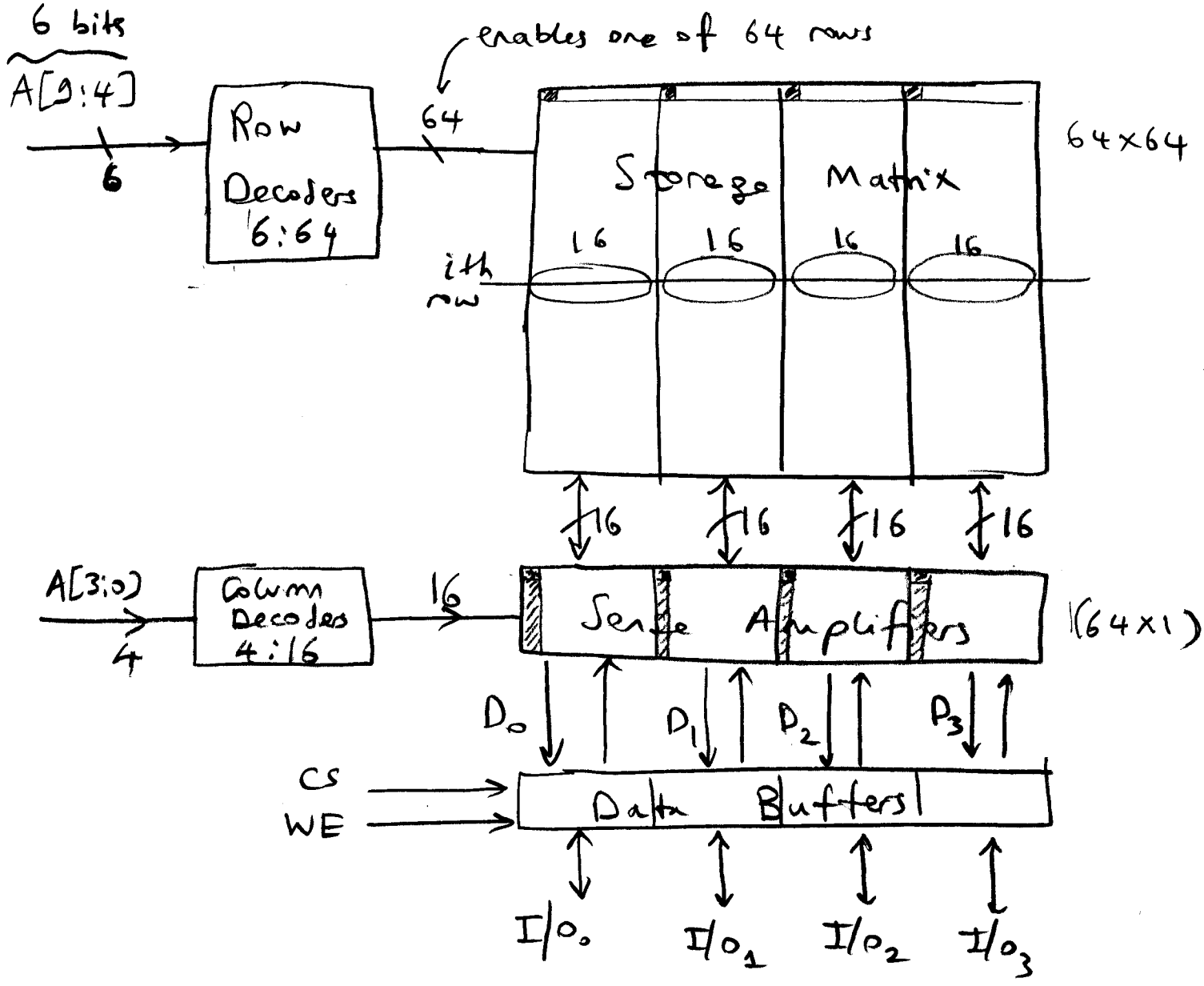
Preferred organization: uses a square storage matrix.

1024 x 4 = 64 x 64. (square)

$(2^{10}) \times (2^2) = 2^6 \times 2^6$ ← internal



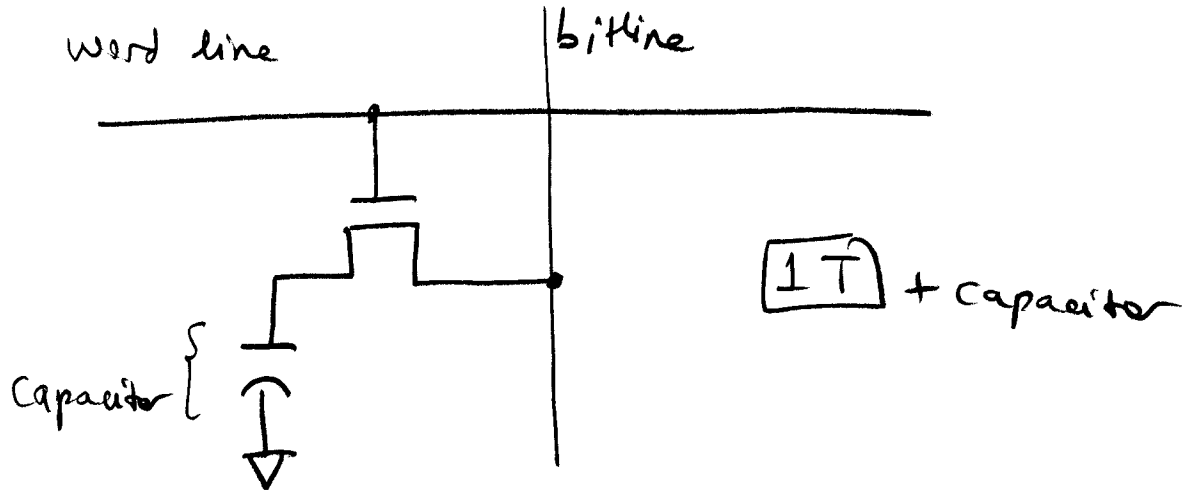
Internal organization with STORAGE MATRIX:



- Note that the 4 bits of a word do not sit next to each other in storage matrix (i.e. physical storage location different from that in conceptual diagram.)
- The word at address 0000000000 is shown shaded above.

Dynamic RAM

- smaller space per cell ⟶ higher integration density
- slower than S-RAM
- needs refreshing (∴ dynamic memory)



- Capacitor holds the charge
- charge leaks over time ∴ need to read out all content & write them back in ("refresh")