- RA = can access in any order without any loss of performance

Sequential = eg: shift registers
data rotated in a very long chain;
don't require address decoding
→ helpful when we want to keep performing operation on a set of data periodically

- ROM: even this is random access

Bridge = Flash memory
(can't write many times)
- also called 'Write Rarely Read Often'
- Read takes long time

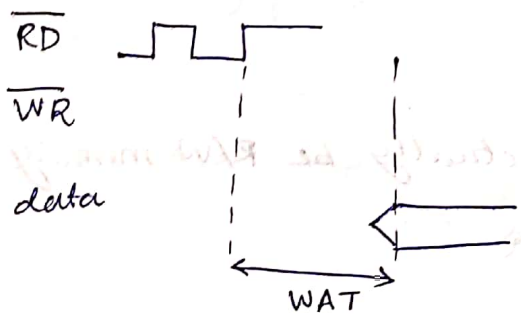
R/W M

Content Addressable Memory: check if any cell has the data

- Read Access time: time between giving read instruction and getting the data output

Write Access time (WAT): time between giving write inst. and how long we need to keep the data unchanged to complete write

Cycle time: time b/w 2 successive read statements
(can be more than read access time)



Cycle time: Worst case time b/w
Read-Read, W-W, R-W, W-R

- when we want large amt. of memory, using lot of D-flipflop is expensive in space (area) and time (control signals) and power

- Rather we make analog ckt: (charge rail-to-rail)
  make sure we get a sufficient voltage change which can be subsequently amplified to a rail value.
  → noise may cause problem hence we use differential amplifier

- Addressing
  - address has to settle befor issuing a $\overline{RD}$ or $\overline{WR}$ (we might otherwise destroy some other data)
  - sometimes memory takes care of it internally

  → having multiple enable signals is inefficient
  → rather we arrange memory in a 2x2 array and get the row and column address

- RAS: Row Address Strobe
  CAS: column —"—

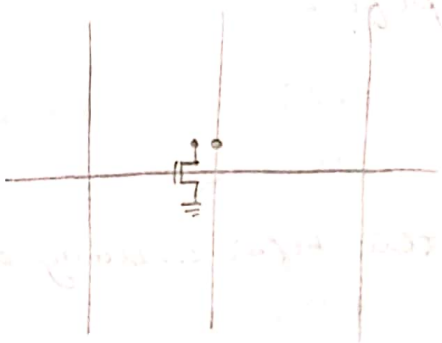
  give RAS first then CAS

- Address line: Bit addressable
  word —"—

- Data line
- RAS/CAS
- $\overline{RD}/\overline{WR}$

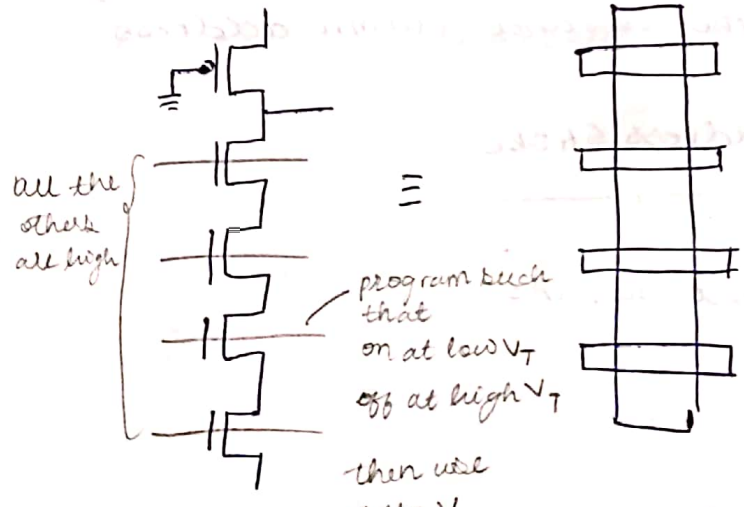
→ use Address to send row
then RAS
use same add. to now send column
then CAS
then $\overline{RD}/\overline{WR}$

- ROM never changed.
  - called NOR memory
  - like programmable logic array



- the selected cell is pulled down to 0.
- may not be pulled to 0, can have a comparator at end → small transistors used

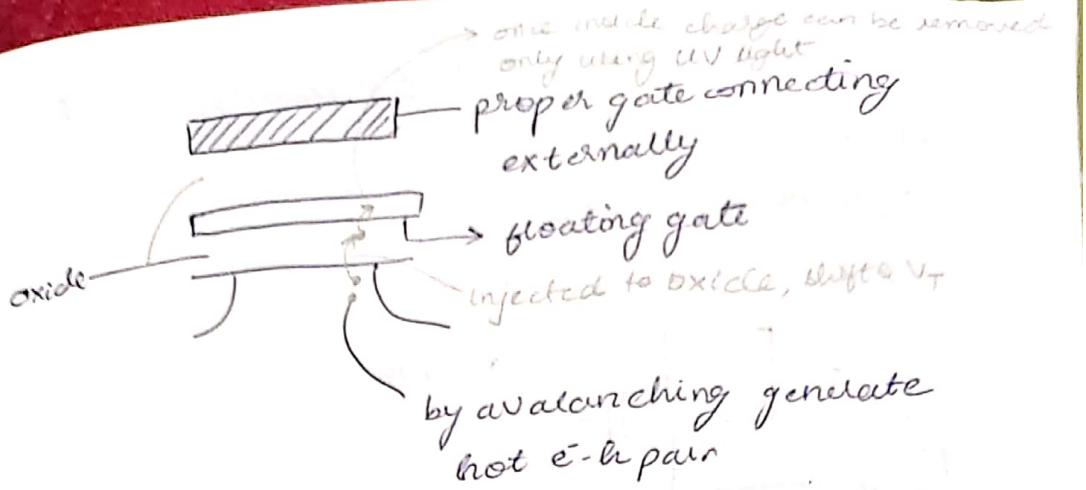
→ NAND memory



→ used for thumb drives, SSD

→ NAND has less density, but slower than NOR

no contact window



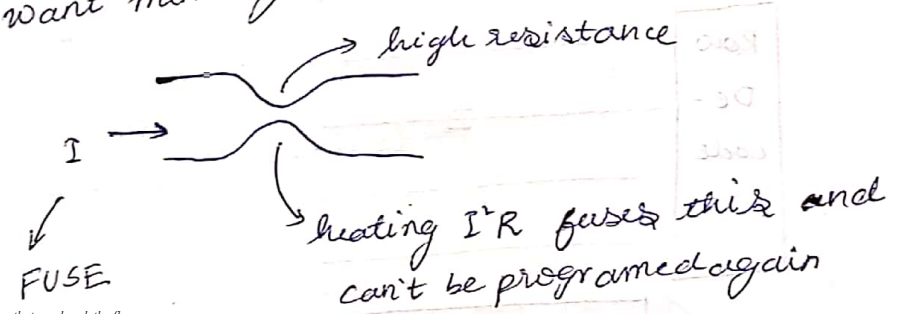
→ removal not perfect, damage producing
hence can only programme a no. of times.

↳ PROM

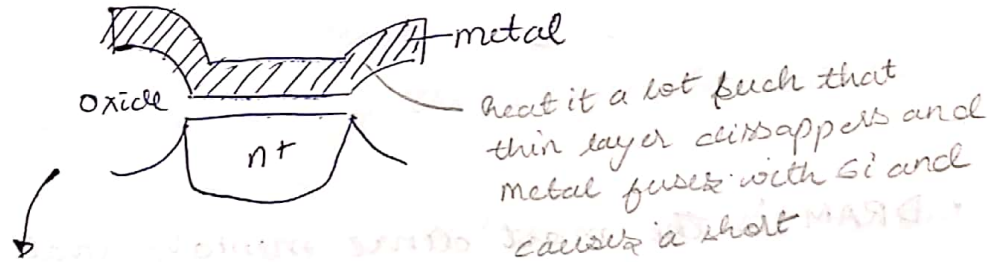
→ modern devices are small enough that tunneling occurs and it can be erased electrically

↳ EEPROM

• want memory that can be programed once

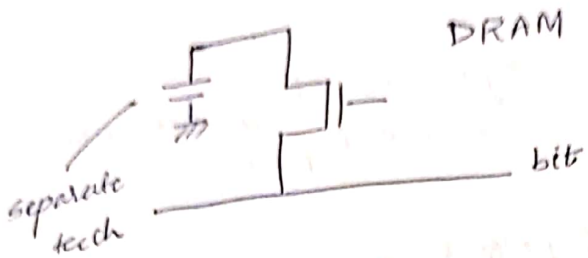


fuse something that can break the flow

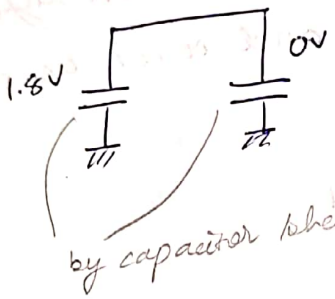


ANTI-FUSE

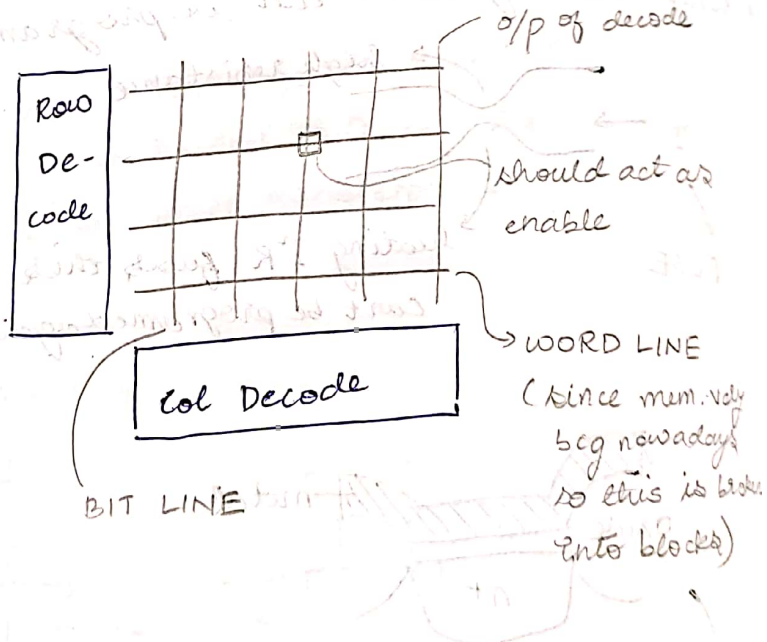
- If we want to program frequently we have
  - static memory
  - dynamic memory



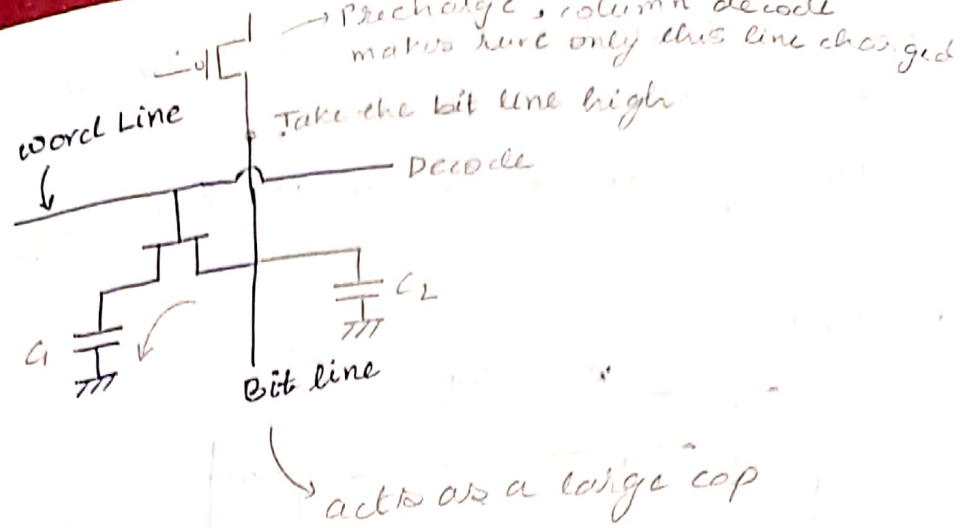
- DRAM should be a separate chip than the rest of the ckt
- when we read we need to write back a 1



24/10

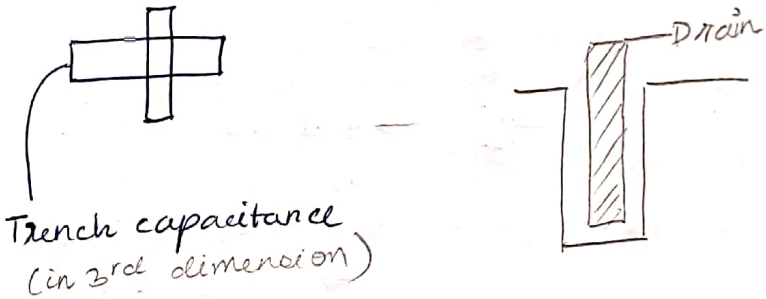


- DRAM is the most dense memory that we can have.
- Read in a DRAM is destructive, so after reading we must write it back
  Thus, cycle time for reading and writing di



acts as a large cap

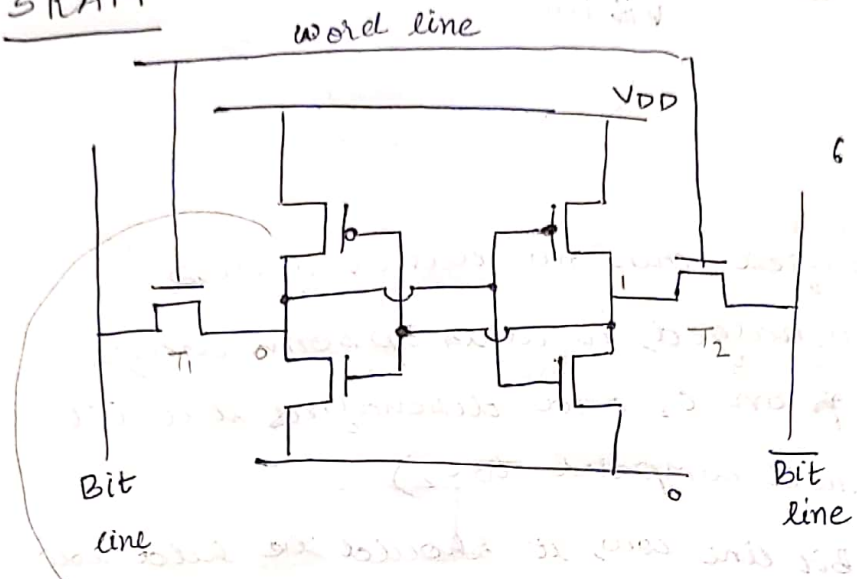
i) if C_1 charged, then no delta V occurs
ii) if C_1 uncharged, it takes up some charge present on C_2 and discharges it a bit
(C_1 small compared to C_2)
iii) when Bit line low, it should be held low long enough to discharge both



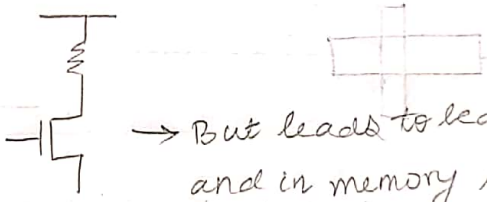
- Need to refreshing the charge on the capacitance (it's leaky).
- So we do a read
- During reading the DRAM should become inaccessible
- Goes through all the rows one-by-one
- So while doing an actual read, do a dummy read (refresh) for the other cells.

- Since DRAM requires a trench cap. and auto control for reading, it is usually kept separate from the std. technology chip

- SRAM



↳ Replace by P-sub N-doped poly ≡ very high resistance (to reduce no. of transistors)



→ But leads to leakage and in memory leakage is important concern
⇒ not used nowadays

- If word line not connected, then this cell is isolated
- column decoder decides the value of Bit & $\overline{Bit}$.
  $\overline{Bit}$ & Bit are taken high when not used
  word line high when selecting that row
- If column is 1, then Bit = 1, $\overline{Bit}$ = 0

- T_1 & T_2 should be strong compared to the other transistors in case of write. However, the requirement is opposite in case of read. Thus, designing their ratios is important

- i) Charge Bit & $\overline{Bit}$ to a high value then turn it off

  ii) Now, charge word line and turn it off

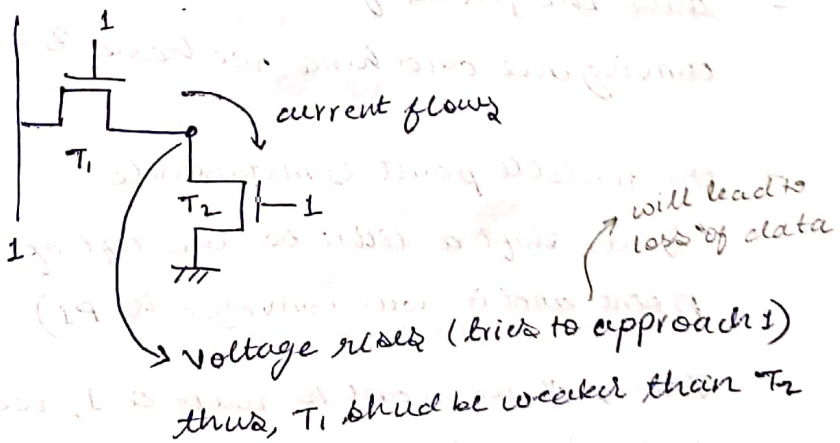
  iii) Suppose, T_1 sees a 1 so it tries to discharge the big capacitance of Bit line

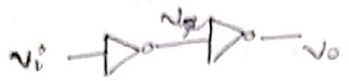
  iv) the Bit and $\overline{Bit}$ lines sent to comparator

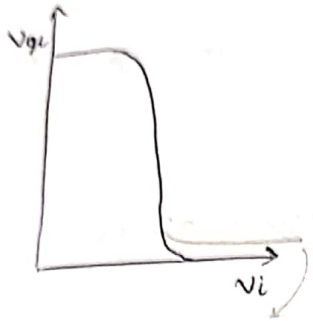
  v) when we turn the word line off, the cell becomes isolated and by positive feedback restore the value

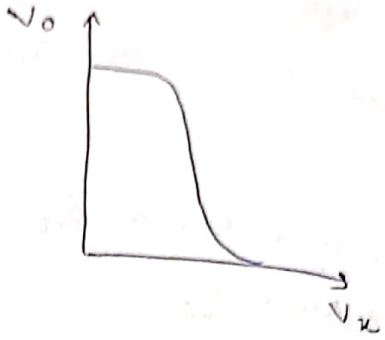
- when we turn a row on, the cell activated is being written into while the others (in that row) are being read.

- while reading

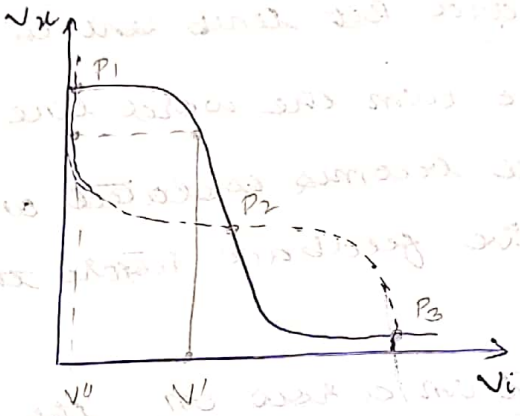








doesn't go to 0 since connected to rail but, bit and would line

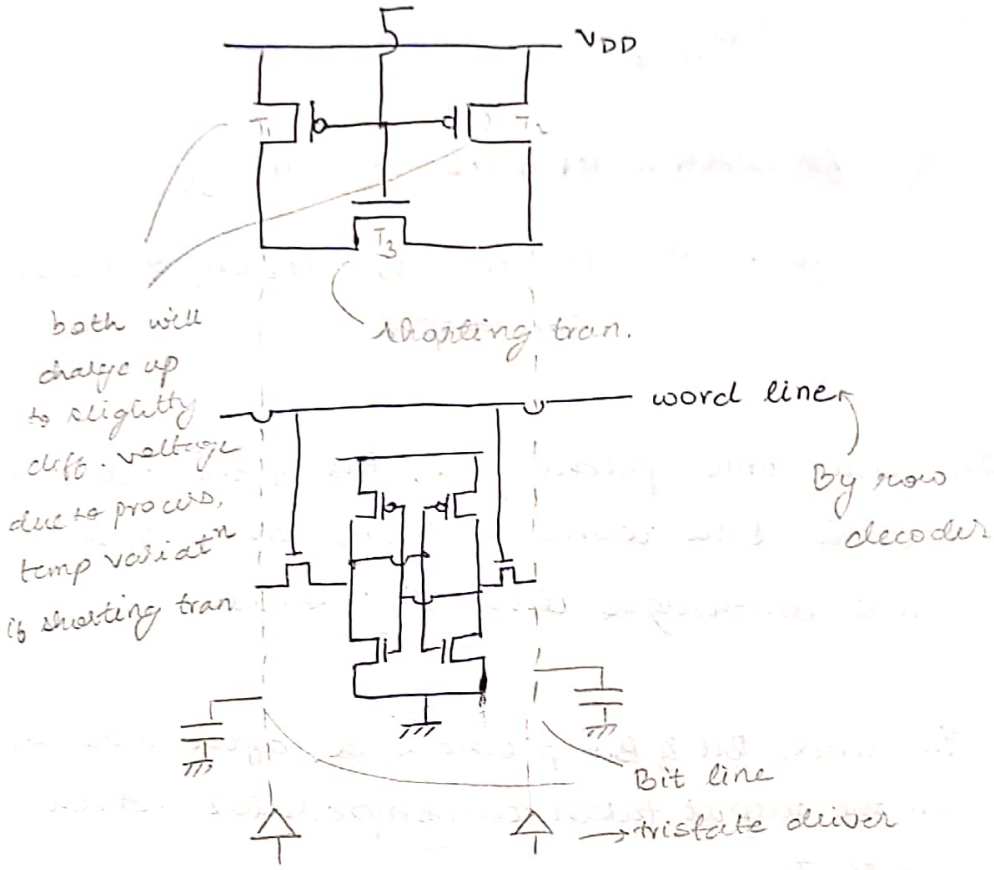


- if V_i connected to V_o, then on starting with V' we end up in V''. Thus, this goes on till we converge at a point
- Thus the points of intersection lead to convergence and hence we have 3 solution.
- the middle point is metastable (if we shift a little to the left of this point and it will converge to P_1)
- Thus, if we want to write a 1, we put our voltage to the right of P_2. However, if it is too close to P_2 it will

take a lot of time to reach 1 so we pull it close to P3

18/10 i) Precharge

(use p-mos, also use a shorting tran.



→ all 3 pmos turned on, Bit lines turn on

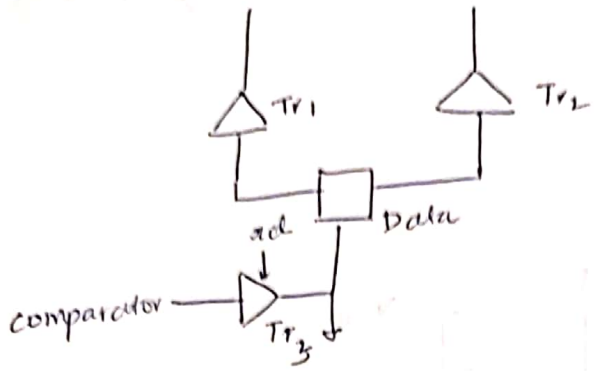
→ turn off T_1, T_2 first then T_3

→ then turn word line on

If T_1, T_2 are pmos, then common mode range rises up to V_{DD}. So we use nmos instead (compact to use nmos) which doesn't charge fully upto V_{DD}.

- Until decoding done can't do R/W
  - ⇒ speed of decoder imp.
  - ⇒ make decoder multistaged (by logical effort multistage faster)

→ typically comparators are 2 stages, acts like a linear amplifier



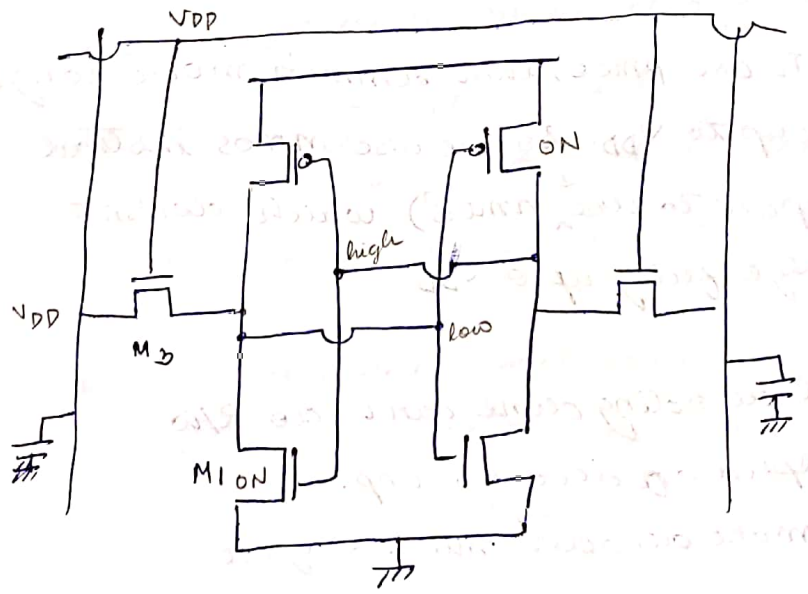
for write, Tr_1 & Tr_2 on, Tr_3 off

for read, Tr_1, Tr_2 off, Tr_3 on, data becomes an output

In write, pmos precharges, Bit & $\overline{Bit}$ get data & $\overline{data}$. once word line on, which of bit & $\overline{bit}$ is 0, discharges that transistor

In read, Bit & $\overline{Bit}$ precharged, after word line on the output taken to comparator which sends it to Tr_3

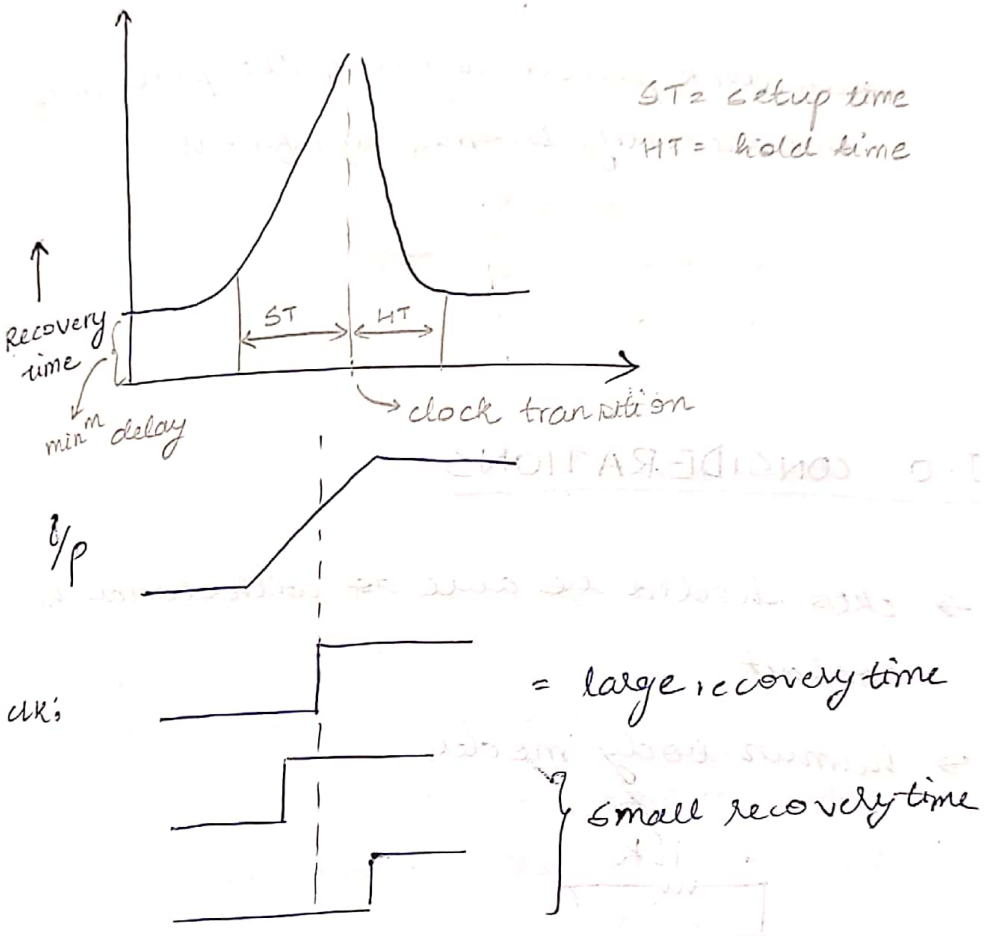
Read conflict

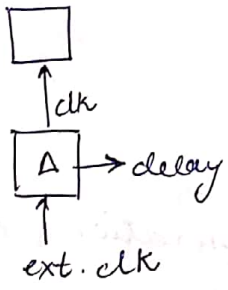


If initially 0 & 1 were written and we want to read that.

o/p : external o/p transition can occur at any time w.r.t clock, so it might change when clock changes. so while sampling, ext. o/p might be close to metastable pt.

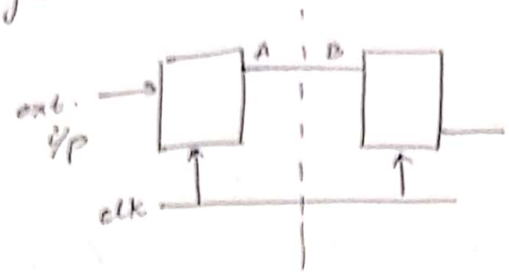
→ it can take an unpredictable amt. of time to become stable.

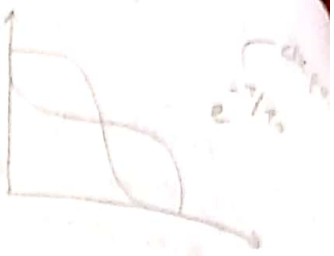




⇒ ext. clock will shift it delta(Δ) before the clock time

- Synchronizer





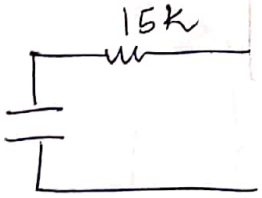
A might be metastable, but B would latch onto a value, instability vastly reduced.
↳ by the time next clock comes

→ as clock period reduces, the problem of metastability becomes significant.
(if T small, then by the time next clock comes, we won't move far enough from metastable point (V_m))

I-O CONSIDERATIONS

→ ckts should be able to withstand human contact

→ human body model



① ⇒ protection imp.

② voltage translation
- chips from diff. generations can be together
- voltage translation needed at input

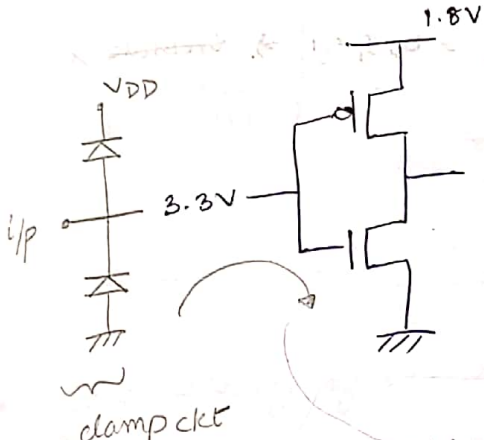
(B) signal conditioning
- robust signal that removes noise

(A) drive large capacitive loads
- for bidirectional parts

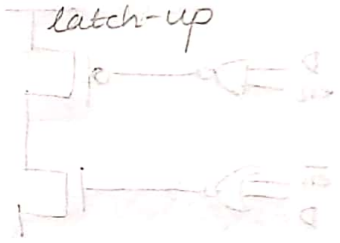
voltage translation

(A) at input

- input swing higher than V_{DD}
  ⇒ safety issue



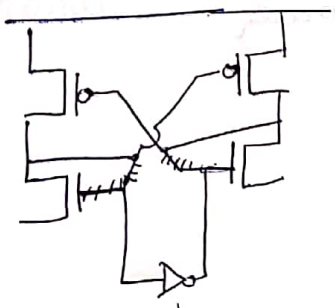
⇒ this could lead to latch-up



price to pay: Retime cont logic
⇒ delay

- input lower than V_{DD}
  - problem with high, low is same
  - also, pmos is the one who can't drive with a small i/p
  - so we use CVSL (doesn't need to be driven by input)

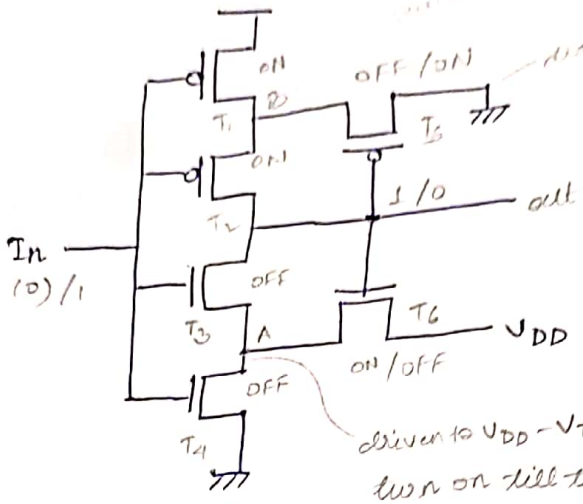
anyways present in pad which are large, hence area not a big concern



[pseudo nmos can be used but it consumes static power]

→ low voltage inverter

⇒ high thresh. higher than average and low thresh. lower than avg.



driven to $V_{DD} - V_{Tn}$ ⇒ T_2 & T_4 won't turn on till the input goes above $V_{DD} - V_{Tn}$

(1) Input 0 for long time ⇒ out → 1

When In reaches V_{Tn}, T_4 turns on and tries to discharge A, but A is supplied by T_6 so, A is somewhat above gnd. so the In that turns on T_2 is somewhat above V_{Tn}

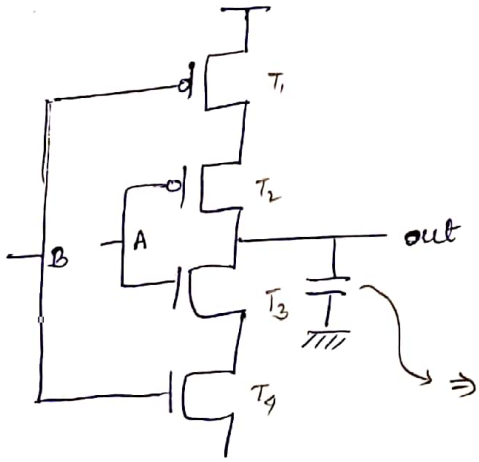
T_4, T_6 form a potential divider

(2) A is T_5 turns on, B becomes V_{Tp}, as T_1 turns on it tries to make B V_{DD} but due to T_5 it is a little below V_{DD}

⇒ T_2 requires an i/p lower than V_B by V_{Tp}, hence the turning on voltage reduces.

- **C element**

→ used for asynchronous design



⇒ dynamic implementation

A	B	out	
0	0	1	
1	1	0	
0	1	prev	Tristate, neither of p or n mos are on as a whole so not connected to power ⇒ take value of capacitance
1	0	prev	

→ equal input ⇒ inverter

unequal inputs ⇒ latch

⇒ called "Event logic"

→ need event on both inputs (not necessarily simultaneous) for the output to change

{std logic = level logic}

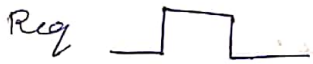
→ C element

(AND of events rather than AND of levels)

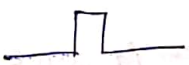
- Asynchronous transfer of data

→ request - acknowledge

↳ talker puts data on data bus, once data settles, it puts request



the other side uses this as a clk and latch the data. Once ~~see~~ latched it acknowledges and allows next data to be sent



→ since 4- transitions ⇒ "4-phase request - ack"

The 2nd edge on ~~a~~ both transitions only restores the value, doesn't do any job.

so we use event logic

→ change of state now tells us both whether req or ack

⇒ less switching ⇒ less power consumed

→ however not used coz as it is event driven and hence we would need to change the entire logic family.

• DFT (Design for testing)

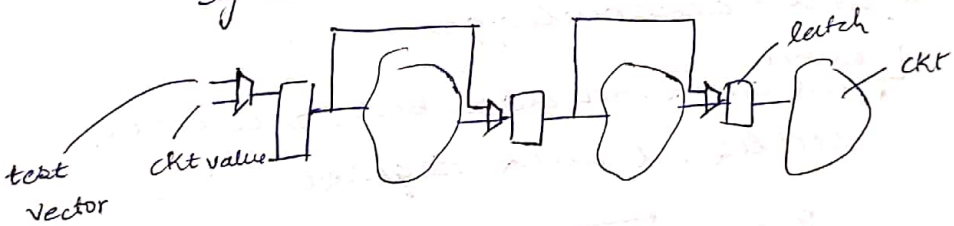
- generation of test vectors that encompass all possibilities
- approachability of nodes

if $S \rightarrow$ scalability

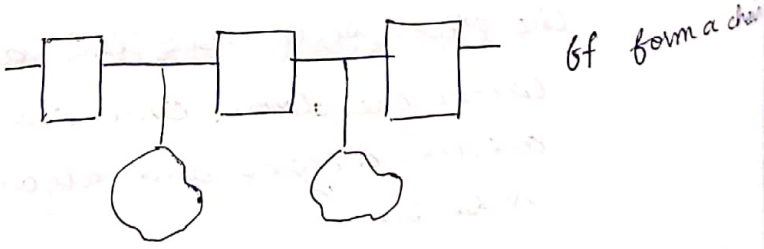
pads $\propto$ 4S ⇒ no. of pins reduces, hence difficulty to test

(A) Border Scan

- test vectors put in series
- synchronous design



⇒ in test mode:

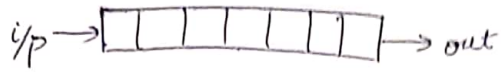


put input in each ff and run for 1 clk cycle, new values stored in ff's.

Now, put in test mode and view the outputs of all subckts

→ as we take out one output at a time,

put in one input vector for the next test



→ since test vector passes through all combinations need to make sure none of them make any subckt invalid
  → never use asynchronous set-reset in such a struct.

→ In border scan can have upto 5 inputs
  - Digital In
  - Digital Out
  - Test In
  - Test Out
  - Scan
  - reset

(B) on-chip test units — req. memory
  - test. vector generator (pseudo-random)