

CD2IC - Cell Design to Integrated Circuits

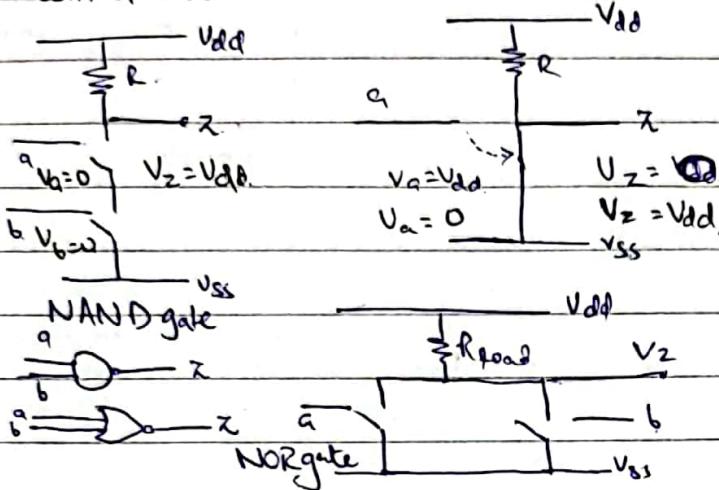
Building a logic gate



- V_{DD} , V_{SS} , R_{load}
- Switch controlled by a voltage (ref V_{GS})

$V_{control} = 0$ open switch

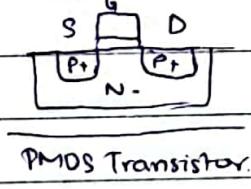
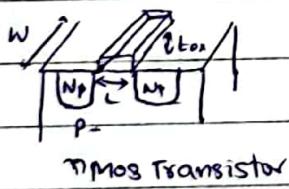
$V_{control} = V_{DD}$ closed switch



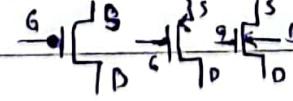
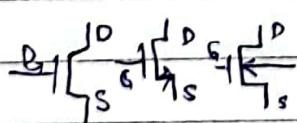
Very simple structure but

- A permanent current flows through the gate when logic output is 0. (useful power consumption should only be when there is activity on gate)
- no ideal switches (at reasonable operating temperature)
 - Logic level 0 does not reach V_{SS} .
 - Safe operation not guaranteed

MOS Transistor (Metal Oxide Semiconductor)



- N-channel
- P-channel
- Electron current
- Holes current
- $V_{GS} > V_{TN}$ (conduction)
- $V_{GS} < V_{TP}$ (conduction)



If $V_{GS} \leq V_{TN} \Rightarrow I_{DS} = 0$ Cut off Region if $V_{GS} \geq V_{TP} \Rightarrow I_{DS} = \omega$

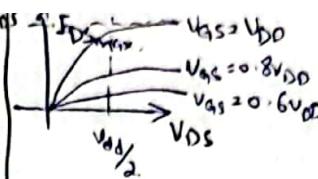
if $V_{GS} > V_{TN}$, $I_{DS} = k_N \cdot (V_{GS} - V_{TN})^2$ Saturation/ conduction Geometrical factor. $k_N = \frac{1}{2} \mu_{nN} \cdot (C_{ox} \cdot W_N / L_N)$

$V_G = 0$ (open switch)

$V_G = 1$ (closed switch)

$V_G = V_{SS}$ (close)

$V_G = V_{DD}$ (open)



Note the effect of L , W , t_{ox} on $I_{DS,max}$.

DATE MONTH YEAR

$V_a = 0$, then $V_z = V_{DD}$.

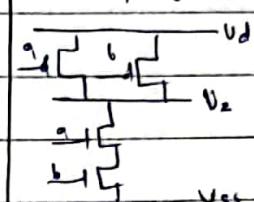
(nMOS cut off, PMOS conducting)

No static power consumption

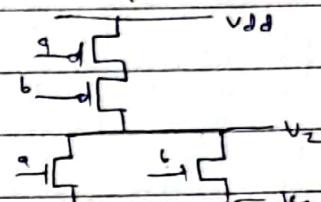
(first order approximation)

Inverter (NOT Gate)

NAND Gate CMOS



NOR Gate CMOS



Performance Criteria

- Area/Cost :- efficiency, lower would be manufacturing cost using less transistors (tech. node) (architectural choice)
- Speed: faster logic gate implies larger processing power. \hookrightarrow power to speed but not good method.
- Power Consumption:- computation means power consumption

Computation time of logic gate

\hookrightarrow rising edge has a null duration

\hookrightarrow only parasitic capacitance taken into account is C_{par} . Current flowing through transistors for charge/discharge of C_{par} is equal to $I_{DS,max}$.

$$I_{DS,max} = k_N \cdot (V_{DD} - V_{TN})^2$$

$$C_{ox} = C_{ox} \cdot W_N \cdot L_N$$

$$C_{par} = C_{par} \cdot \frac{dV_{DD}}{dt}$$

$$I_{Cpar} \approx I_{DS,max} = C_{par} \cdot \frac{dV_{DD}}{dt}$$

Discharge from V_{DD} to 0 :- $t_{comp} = C_{par} \cdot \frac{\Delta V}{I_{DS,max}}$

In order to increase speed, increase power supply

\hookrightarrow bad way (overclocking) $I_C = C \cdot \frac{dV_C}{dt} \Rightarrow t_{comp} = \frac{\Delta V}{I_{DS,max}}$

$$t_{comp} = C_{par} \cdot \frac{V_{DD}}{k_N \cdot (V_{DD} - V_{TN})^2}$$

Power Computation of CMOS logic

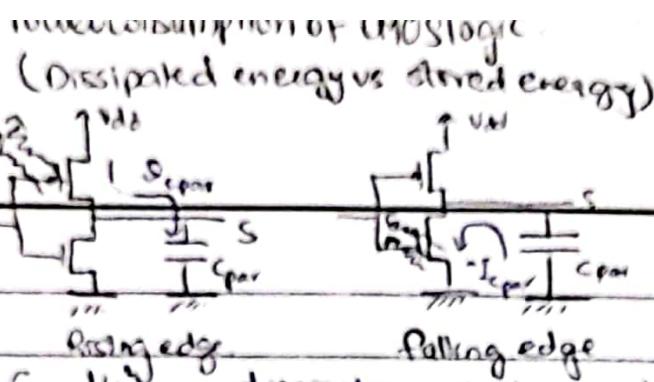
$$E_{VDD} = C_{par} \int_{0}^{V_{DD}} V_{DD} dV_S = C_{par} V_{DD}^2 / 2 \text{ (Charging)}$$

$$E_{VSS} = C_{par} \int_{V_{SS}}^{V_{DD}} V_{DD} dV_S = C_{par} \cdot \frac{V_{DD}^2}{2} \text{ (discharging)}$$

$C_{par} \cdot V_{DD} / 2$ dissipation whatever the edge.

charging: energy comes from supply.

discharging: stored energy in capacitance



Power consumption of full chip

- let C_{par} be overall parasitic capacitance
- f_{CK} → chip operating frequency (synch. logic)
- let T_{act} (activity) be the mean transition probability of signals during a single clock cycle ($T_{act} \approx 0.3$)

$$\text{Percent Oscillations} = \frac{\text{Total fan-out} \cdot C_{par} \cdot V_{DD}^2}{\text{Consumption}}$$

What do you think about over clocking now?

if we increase V_{DD} although $\frac{1}{2}$ but

overall power consumption ↑ at much higher rate.

Moore's law

Bordon Moore, cofounder of Intel.

- A complexity for minimum component costs has increased at a rate of roughly factor 2 per year.

widened to other parameters:-

Processing power of ... doubles every ... year
Power consumption divides by 2 every ... year

Moore's law were exponential & followed during four decades

for less than 2nm tech node does not represent transistor size/gate length density

Tech downscaling (concept of standard node)

Cell area of CPP & MMP × Tracks

CPP (contacted poly pitch), MMP (mini metal pitch)

Today's problem :-
• for CPU frequencies have reached its max value (3-4GHz) at beginning of century due to max heat dissipation of chips.

DATE	MONTH	YEAR

- When lowering V_{DD} , we no longer have "ideal switch" approximation. Chips have larger & larger static dissipation added to computation dissipation.
- Technologists must use more & more complex (costly) manufacturing processes to continue to follow Moore's law.

- Some said end of Moore's law due to scientific reasons (MOS transistor physics) but main difficulty is economical.

Technology evolution model "Theoretical downscaling"

Techniques-

- A tech node is defined by minimum gate length of transistors
- con forders try to reduce transistor area by 2.
- A linear reduction factor $\beta = \frac{1}{2}$ is used
 - W & L of transistors are divided by β .
 - thickness of oxide layer is divided by β .
 - power supply voltage V_{DD} is divided by β .
 - threshold voltage V_T of transistor is divided by β .

Performance evolutions

$$\begin{aligned} C_{par}(\beta) &= (W/\beta) (L/\beta) (\beta C_{ox})^2 = C_{par}/\beta \\ E_{gate}(\beta) &= (C_{par}/\beta) (V_{DD}/\beta)^2 = E_{gate}/\beta^3 \quad (\text{energy consumption}) \\ t_{comp}(\beta) &= t_{comp}/\beta \quad (\text{computation time}) \end{aligned}$$

Reducing costs and power consumption

Keeping clock frequency constant $F_{CK}(\beta) = F_{CK}$

→ Area = Area/ β^2 (area reduction ⇒ price reduction)

$$P_{comp}(\beta) = T_{act} \cdot F_{CK} \cdot E_{gate}/\beta^3 = P_{comp}/\beta^3$$

→ useful for transition from high-end to low-end (multi)

→ ultra-low power devices

Enhancing computational power

Using max Achievable $f = \beta F_{CK}$

- Area(β) = Area, $P_{comp} = P_{comp}$, useful in server CPUs. Computational power ↑ using higher frequency & parallelism.

CMOS Logic

(Generalization to other boolean functions)

- PMOS networks used for logic ones of boolean function.

NMOS function \Rightarrow logic zeros.

• 1 to 1 NMOS/PMOS structure

• Implementation gates function: $\sum \bar{x}_i x_i$

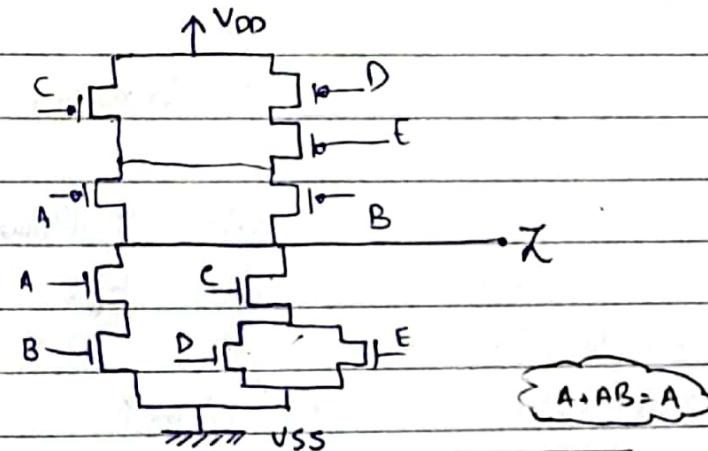
NMOS $\rightarrow \sum x_i$; PMOS $\rightarrow \sum \bar{x}_i$

• CMOS are inverting gates (NAND smaller than AND)

• Dual PMOS/NMOS \rightarrow NMOS parallel PMOS series and vice versa.

• Leads to non-optimal gate (electrical point of view)

$$Z = (\bar{A} \cdot B) + C \cdot (\bar{D} + E)$$



Majority Gate $F = \bar{A} \cdot B + B \cdot C + A \cdot C$

Q.1 Build gate using optimum optimization methods

$$(\bar{A}B) \cdot (\bar{B} \cdot C) \cdot (\bar{A} \cdot \bar{C}) = (\bar{A} + \bar{B}) (\bar{B} + \bar{C}) (\bar{A} + \bar{C})$$

$$= (\bar{A} \cdot \bar{B}) + (\bar{A} \cdot \bar{C}) + \bar{B} + \bar{B} \cdot \bar{C} (\bar{A} + \bar{C})$$

$$= \bar{B} (\bar{A} + \bar{C}) + (\bar{A} + \bar{A} \cdot \bar{C} + \bar{A} \cdot \bar{B} + \bar{B} \cdot \bar{C}) (\bar{A} + \bar{C})$$

$$= (\bar{A} + \bar{A} \cdot \bar{C} + \bar{A} \cdot \bar{B} + \bar{B} \cdot \bar{C} + \bar{B}) (\bar{A} + \bar{C})$$

$$= (\bar{A} + \bar{A} \cdot \bar{C} + \bar{B} \cdot \bar{A} + \bar{B} \cdot \bar{C}) (\bar{B} + \bar{C})$$

$$= (\bar{A} + \bar{B} \cdot \bar{C}) (\bar{A} + \bar{B} \cdot \bar{A} + \bar{B} \cdot \bar{C}) (\bar{B} + \bar{C})$$

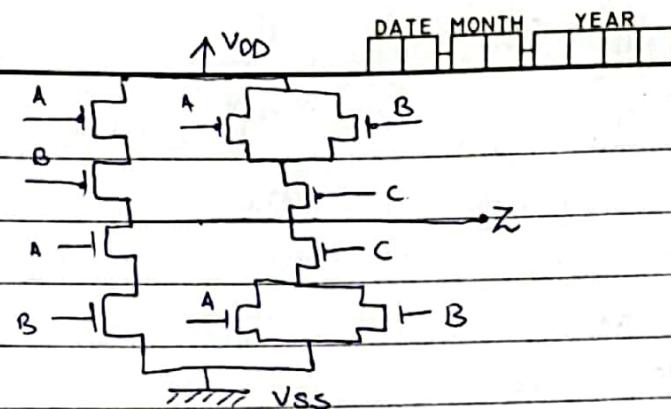
$$= (\bar{A} + \bar{B} \cdot \bar{C}) (\bar{B} + \bar{C})$$

$$= \bar{A} \cdot (\bar{B} + \bar{C}) + \bar{B} \cdot \bar{C} (\bar{B} + \bar{C})$$

$$= \boxed{\bar{A} \cdot \bar{B} + \bar{A} \bar{C} + \bar{B} \cdot \bar{C}} = \bar{A} \cdot \bar{B} + \bar{C} \cdot (\bar{A} + \bar{B})$$

$A \cdot B + B \cdot C + A \cdot C$ for PMOS: - - -

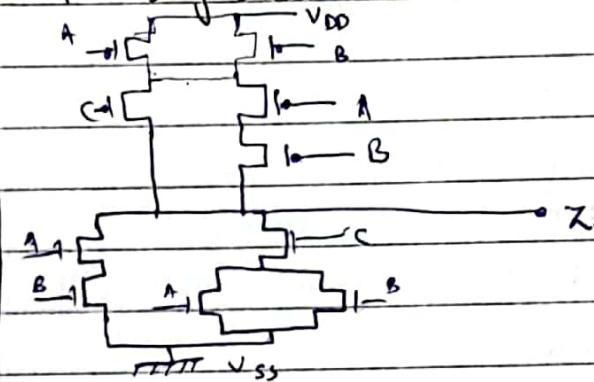
$$A \cdot B + B \cdot C + A \cdot C = A \cdot B + C \cdot (A + B)$$



If we exchange CMOS with A/B NMOS?

Logical part no diff. but diff in electrical/timing part.

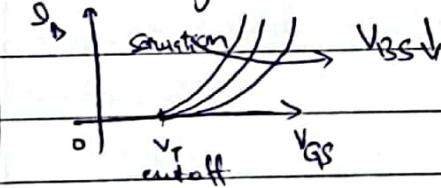
Build using dual network?



Disadvantage: - 3 transistors at every node, delay, zada ana.

CMOS logic (low/high input counts)

series of transistors: - the body effect leads to slow high V_t transistors.

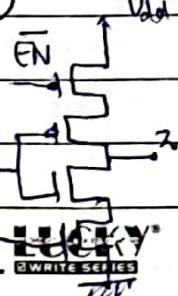


• Assembly of low count input gates may be faster than high input count gates.

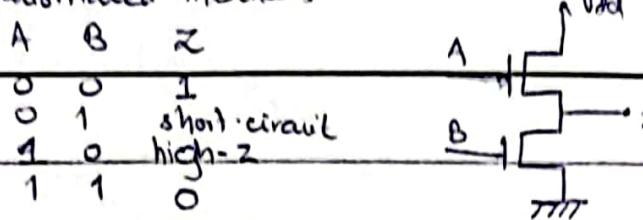
CMOS (non-complementary N/P MOS)

• NMOS/PMOS simultaneously off.

EN	A	Z	• 9/10 transceivers for external buses
0	0	high Z	
0	1	high Z	
1	0	V_{DD}	
1	1	V_{SS}	• tri-state inverter



- NMOS/PMOS may be simultaneously ON.
- avoid inputs which short circuits
- only for full custom design (not included in automated methods)



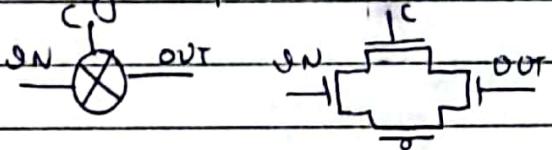
CMOS logic summary

- ↳ Robustness, noise immunity
- ↳ sizing of transistors affect only performance not functionality
- ↳ in practical case, series of transistors limited to 3-4 transistors

But: - only 1 boolean function but 2 transistor networks

Pass transistor logic (Analog switch)

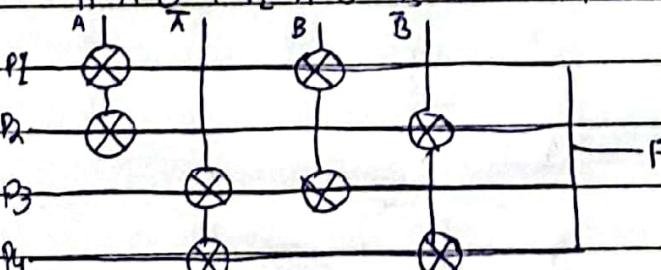
- NMOS efficient for falling transistors
- PMOS for rising transistors
- Passive circuit, no signal regeneration along the path "IN" to "OUT"



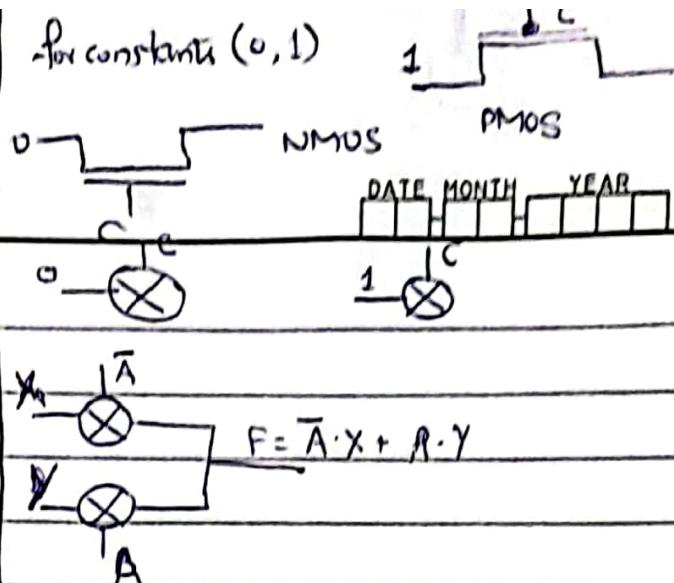
Inputs signals may be constants or variables

- Input is a constant :- CMOS logic
- Input is a variable :- Pass logic
- conditions should not lead to open/ short circuits
- Any form of Boolean expression

$$F = P_1 \cdot A \cdot B + P_2 \cdot A \cdot \bar{B} + P_3 \cdot \bar{A} \cdot B + P_4 \cdot \bar{A} \cdot \bar{B}$$



For constants (0,1)



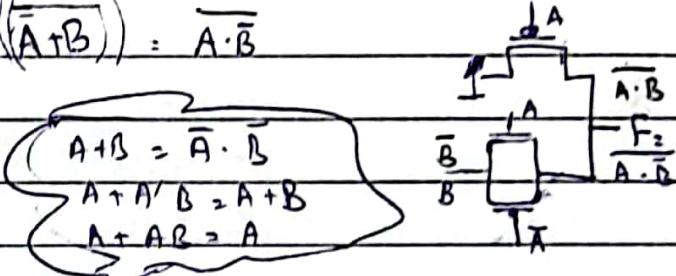
Generate function for $A \cdot \bar{B}$, $\bar{A} \cdot B$,

$$A \cdot B, A \cdot \bar{B} \quad \bar{A} \cdot B + \bar{A} \cdot \bar{B} = \bar{A} \cdot B + A \cdot \bar{B}$$

$$A \cdot \bar{B} \quad F$$

$$\begin{array}{ccc} 0 & 1 & \Rightarrow (\bar{A} + B) = \bar{A} \cdot B + A \cdot \bar{B} \\ 1 & 0 & \Rightarrow (\bar{A} + B) = \bar{A} \cdot B \\ 1 & 1 & \Rightarrow \bar{A} \cdot B = A \cdot \bar{B} \end{array}$$

$$(\bar{A} + B) = A \cdot \bar{B}$$



$$\bar{A} + \bar{B} = A \cdot B$$

$$A \cdot B \cdot \bar{B} \quad F$$

$$0 \quad 0$$

$$1 \quad 0$$

$$1 \quad 1$$

$$(\bar{A} + \bar{B}) = \bar{A} \cdot \bar{B}$$

$$= A \cdot B$$

$$(\bar{A} + \bar{B}) = \bar{A} \cdot \bar{B} = A \cdot B$$

$$= A \cdot B$$

$$(\bar{A} + \bar{B}) = \bar{A} \cdot \bar{B} = A \cdot B$$

$$= A \cdot B$$

$$\bar{A} \cdot \bar{B} \neq A \cdot B$$

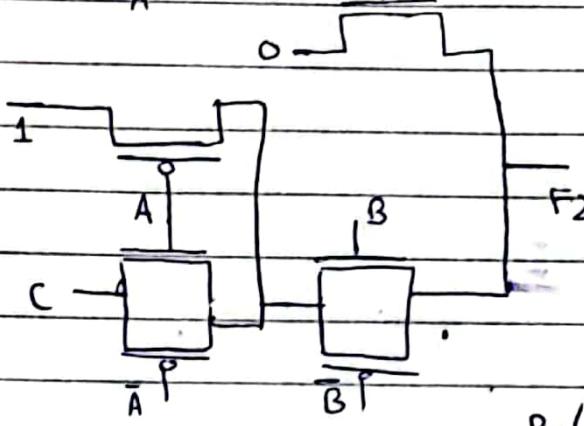
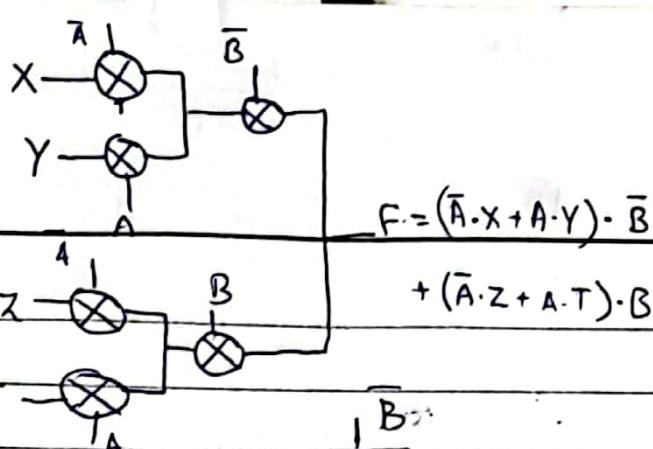
$$\text{e.g. } \bar{A} \cdot \bar{B} = \bar{A} \cdot \bar{B} \quad 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1$$

$$1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0$$

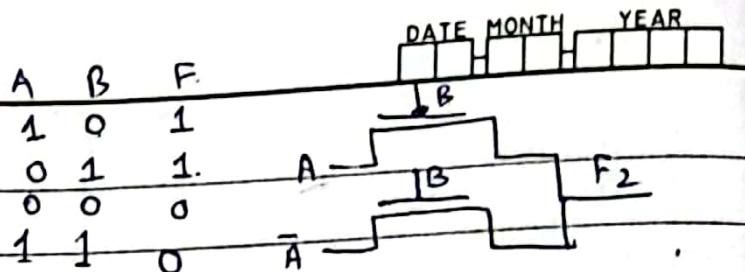
$$1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1$$

$$0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0$$

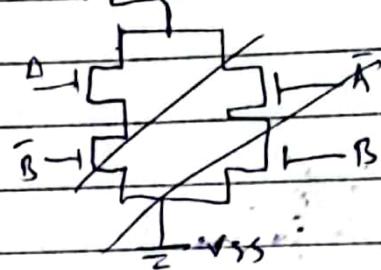
$$0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0$$



XOR Gate $A \cdot \bar{B} + \bar{A} \cdot B$
In classical Pass transistor



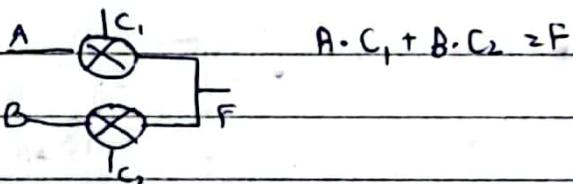
CMOS XOR



A	B	C	F2
-	0	-	0
0	1	-	1
1	1	C	C

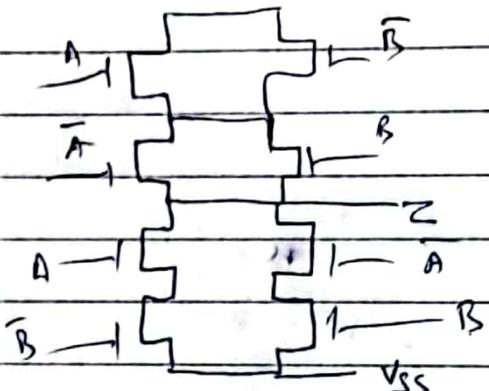
Less Transistors than Pass logic using decoder
heavily used in LUTs in FPGA

Non-Exclusive conditions



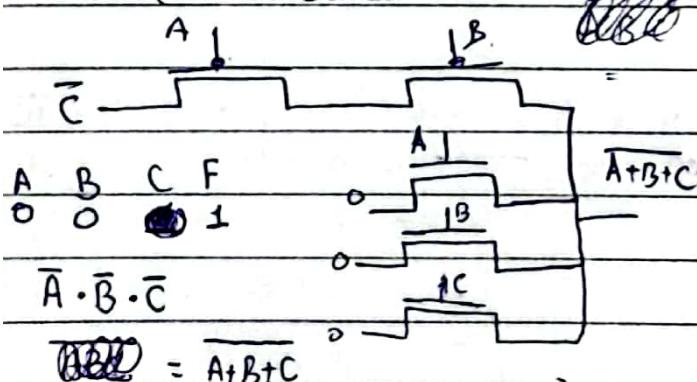
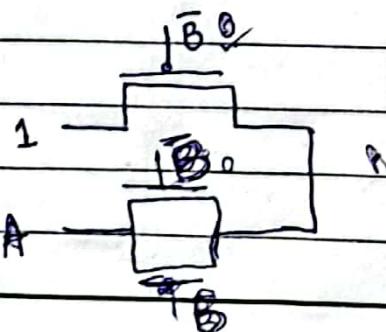
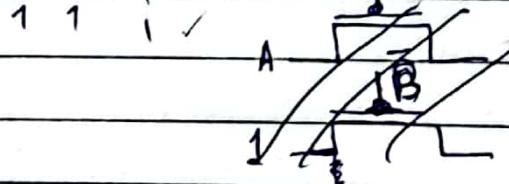
Several branches simultaneously on.

Short circuits & high impedance state
should be avoided



$A + B \rightarrow$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



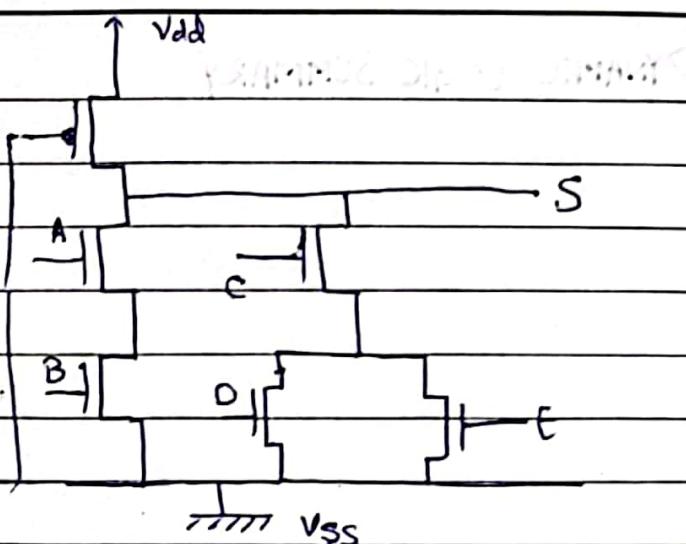
$$\bar{A} \cdot \bar{B} \cdot \bar{C}$$

$$\text{NOR} = \bar{A} + \bar{B} + \bar{C}$$

L3 - Structural Design of digital circuit (2/2)

Dynamic NMOS logic

Pseudo-NMOS logic



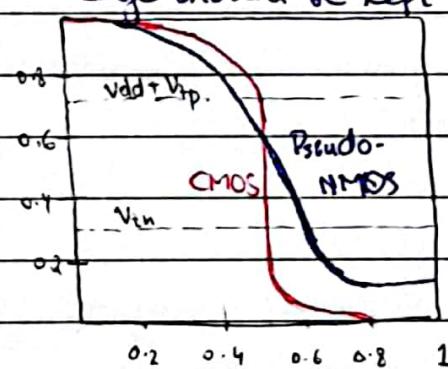
• Replacement of PMOS network by a passive load.

• Smaller ~ The "1" values of truth table are implicit values

• Conflict ~ When NMOS network is on \rightarrow Steady state current.

NMOS
PMOS-width balancing:-

- low output voltage should be less than threshold voltage of NMOS transistor.
- Propagation time for output rising edge should be kept small.



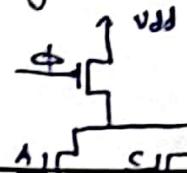
How to avoid choice between speed & robustness

Dynamic logic key points:-

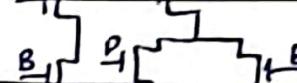
- very high speed (small capacitive loads) (high speed carry chain in sep)
- But power consumption of clock network
- Precharge phase is wasted time.
- Complex design automation (no direct RT synthesis tools)
- ⇒ only used in optimized "full-custom" design.

Dynamic logic :- Precharge logic

- Only 1 NMOS network.
- 1 clock (sync. control)



DATE MONTH YEAR



- $\Phi = 0 \Rightarrow$ precharge phase (output is charged to "1")

- $\Phi = 1$ (conditional computation of output) \Rightarrow evaluation phase

\Rightarrow clock should be fast so that state '1' of precharge does not get lost. because of leakage current of transistors (limits $f_{CK \text{ min}}$) \Rightarrow else state '0' disappears.

\Rightarrow State 1 is a high impedance state.

Constraints:-

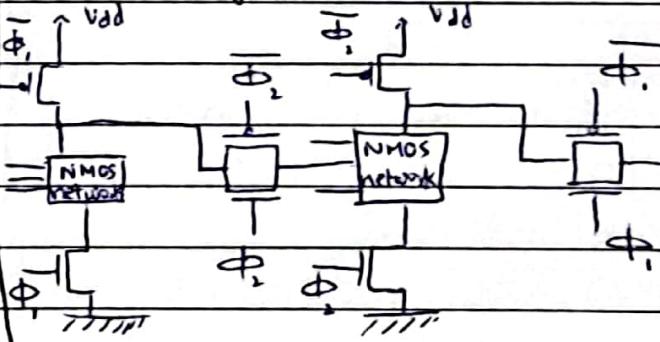
\hookrightarrow if final state is "1", output state should stay "1" during evaluation phase.

\hookrightarrow inputs should be stable during evaluation phase.

\hookrightarrow outputs of such cells can not be used as inputs of another cell.

\hookrightarrow even if we meet constraints, final output voltage may be less than V_{dd} . (charge sharing inside NMOS network)

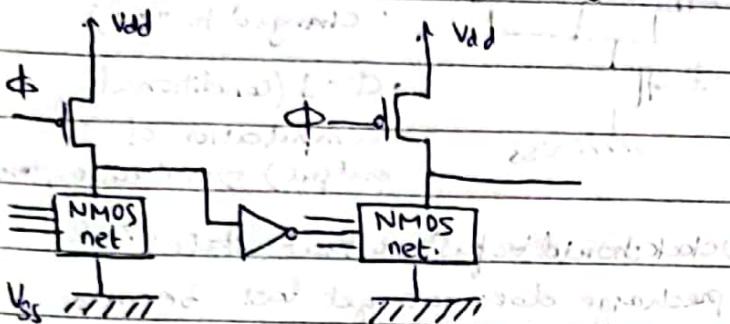
Dynamic logic: Two Phases DL



- Φ_1, Φ_2, Φ , should be with non-overlapping phases.
- Cell isolation using pass transistor logic & acting as a register.

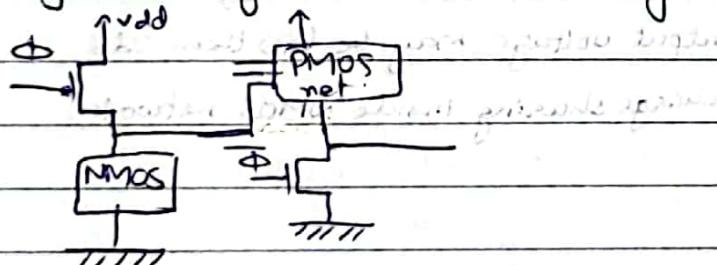
- fully cascadable, but odd & even cells should alternate
- Deep pipelining (no more than 1 gate level between each pipeline register)

Dynamic logic :- Domino logic



- During precharge phases, all gates are '0' (all NMOS networks are off)
- During evaluation phase, some inputs switch to '1'.
- Then some NMOS network switches to ON state (some gates outputs switch to 0; some switch to 1)
- 1 cycle $> \sum T_{\text{propagation}}$
- ⇒ Warning: - only non-inverting gates can be implemented

Dynamic logic: - Domino N-P logic



- Simplified: - no more inverter
- Warning: - only inverting gates can be implemented
- NMOS gates should alternate PMOS.

Differential logic

- All signals are duplicated = $A \Rightarrow A_I, A_F$.
- Parallel computations of F and \bar{F}
- No inverter needed
- Reduces complexity of arithmetic computations

- 2 NMOS transistors networks of equal size
- no limitation to inverting / non-inverting gates
- less # of Gates but DATE MONTH YEAR more wire

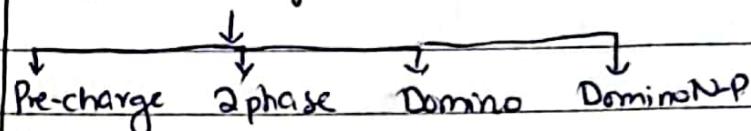
DYNAMIC LOGIC SUMMARY

* Pseudo-NMOS ..

(PMOS ground & sat connected)

⇒ Agr sarey NMOS on to steady state current (conflict state)

* Dynamic logic.



PMOS aur NMOS network kneechni aur NMOS P & N dono ko phi dena.

$\Rightarrow \phi = 0$ (precharge phase); output charged to 1

$\Rightarrow \phi = 1$ (evaluation phase);

\Rightarrow clock should be fast enough so that state '1' (high impedance state) does not disappear due to leakage current

2 phase:-

- cascading possible
- alternate even odd block
- $\phi, \phi_2, \text{PMOS } \bar{\phi}, \text{PMOS } \phi, \text{PMOS } \bar{\phi}, \text{NMOS } \phi, \text{NMOS } \bar{\phi}$ beech mai alk pass transistor logic (1st ki output is se connect)
- ϕ, ϕ_2 = out of phase
- non-overlapping

Domino

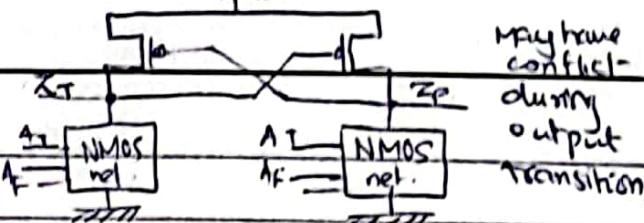
- no extra NMOS; pass logic ki jaga inverter lagado; stiff non-inverting logic implement ho skte.
- stiff PMOS ko phi dena.

Domino N-P:-

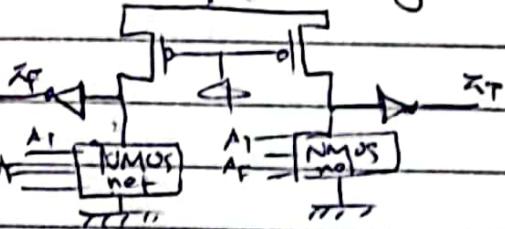
- Alternate NMOS & PMOS; only inverting gates can be implemented.
- PMOS ko phi & NMOS ko $\bar{\phi}$.

Differential logic

Cascade Voltage Switch Logic



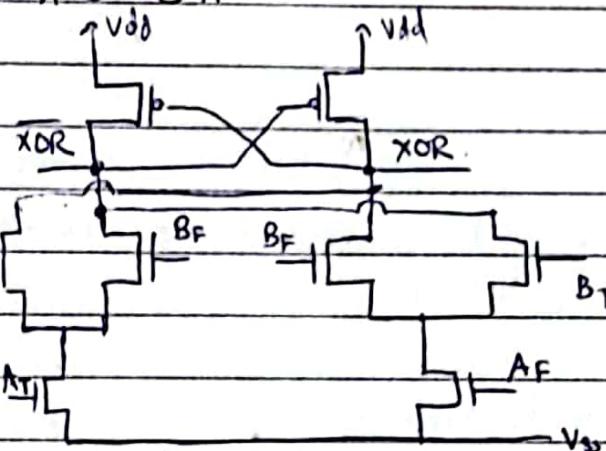
Dynamic Cascade Voltage Switch logic



Q. Design CVSL 2 input XOR gate.

try to minimize # of transistors.

$$A \cdot B + B \cdot \bar{A}$$



Differential logic - Complementary Pass logic



⇒ Pass transistor logic using only NMOS - slow degraded logic but ...

weak pullups PMOS restore full scale swing and outputs are buffered by CMOS inverters.

• using low V_T transistors for the network (high speed but more leakage); using high V_T for inverters helps speed optimization

- said to be one of the fastest logic
- Application Specific logic
- current mode logic
- adiabatic logic
- subthreshold logic

DATE MONTH YEAR

Positive Edge Triggered D-Flip flop:-

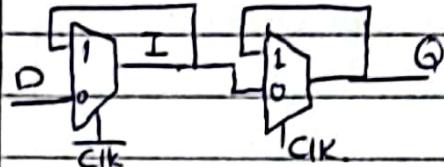
(Theoretical Master/Slave flip-flop)

⇒ Muxes with loops defined 2 storage elements (master and slave).

⇒ Master (resp slave) hold value while slave is transparent

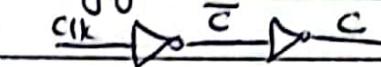
⇒ Skew between 2 clocks (should be avoided) ↴ (race condition)

⇒ D signal should respect hold/setup time



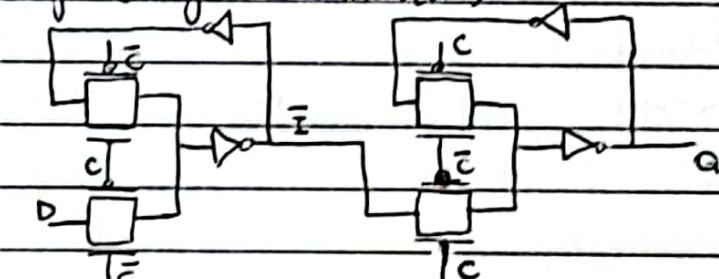
Practical Design:-

⇒ clocks are internally generated to avoid unwanted skew



⇒ Warning D input is on the drain of transistors.

⇒ One can mix CMOS logic and pass-transistor logic (using tristate inverters)



Standard Cells

• layout design limited to generic cells

• electrical simulation for cell properties extraction

• small logic styles choices

• automation of synthesis, place & route phases

• suboptimal for speed, power, area

• when time to market is main criteria

Full Custom

• layout design limited to generic cells

• manual layout of all needed cells

• long electrical simulations for verification of whole block

• wide logic style choice

• scripting may help layout phases

• ultimate optimization for speed, power, area

LUCKY

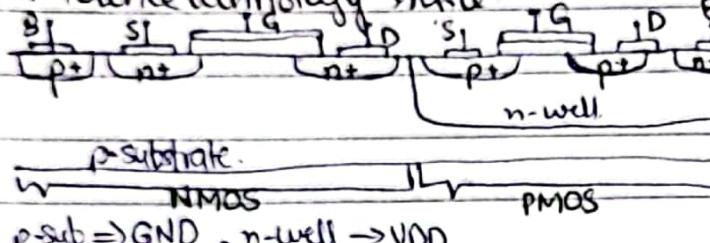
WHITE SERIES

Standard cell Principles

- ⇒ All cell have same height
- ⇒ Power supply & GND connected by abutment
- ⇒ Cell design should be free of DRC
- ⇒ Any abutment of any couple of cells should be free of DRC.

⇒ wiring inside cell limited to Metal 1 layer

Reference technology profile



qpk045 standard cell Template

- NMOS & PMOS areas already filled
- NMOS & PMOS body-ties areas already filled.
- Body ties connected already NMOS bulk (GND), PMOS bulk (VDD).
- Simple abutment of cells fill row of cells with PMOS & NMOS areas.

qpk045 adder

- ⇒ all transistors have horizontal orientation
- maximal width defined by NMOS/PMOS areas height
- use parallel transistors for larger widths
- drain/source implants (high resistance) may be used for local short wires (beware of resistivity)
- Global optimization of (PMOS/NMOS subcircuits) are grafted which visits every edge exactly once (eulerian paths).

Performance Metrics

- area of logic gates
- power consumption
- EDP (energy delay product), robustness of design
- Near threshold and subthreshold behavior.

How to transmit data on long wire

⇒ long wire as a distributed RC model; distributed inverters help minimize overall propagation time; same kind of optimization but with non-linear model of T_p through the line.

BUFFER OPTIMIZATION EXAMPLE

Simple Inverter

(sink to ground) when space issue too pr implant to enlarge to dense, very resistive

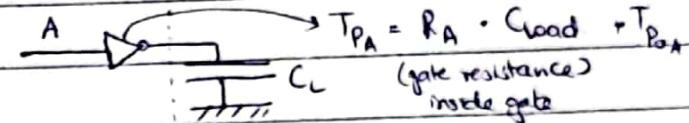
DATE MONTH YEAR

(like bad contact resistive thru metal layers)

Short connections between transistor preferred

- away of contacts (large) jatay to reduce resistivity ⇒ parallel of main jatay

Q.1 What is the fastest way to transmit data from input of Gate A; to inputs of gates connected to A?



$T_{PA} \Rightarrow$ even when no external load; gate has parasitic capacitance inside it.

⇒ if parametrized inverter used :-

(gate width multiplied by α) $\alpha \geq 10$

$$T_{PA} = \frac{R_A}{\alpha} \cdot C_{load} + T_{P_{PA}}$$

T_{PA} same like ratio of α internal capacitance

increase α by α & resistance decreased by α .

$$C_{pmode} = \frac{C}{\alpha}; R_{pmode} = R/\alpha.$$

With increasing α , T_{PA} decreases; but it

would load previous cells as C_{load}

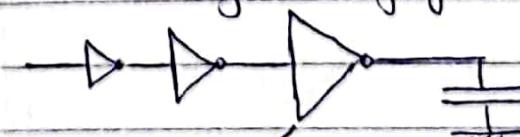
for previous circuitry would increase.

what is the value of α for minimum propagation time.

TM Using inverter of good size helps in

minimizing propagation time

Is logic cell se pechay aik $\alpha = 1$ ka inverter lagga do. ta k pechay circuitry ke load na aye aur aik balanced inverter



This enormous connected to all D-flip-flops. Is point pe bkt zada current ajana.

Lecture 4:- Design Automation

Standard Cell Library

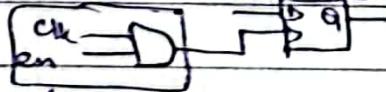
- a wide selection of cells
- facilitate synthesis tools usage
- non-obvious cells (AND, OR, BUF, INV, XOR, ...)
- for synthesis and P&R timing optimization.
- No way to resize transistors
- Growing sizes of predefined output transistors
- Small choice for complex cells.
- extra choice for INV/BUF (load adaptation)
- ! Speed/Dynamic Power/Leakage power tradeoff
 - several versions of library with same cell layout
- \Rightarrow Based on Available Transistors Threshold choice
- \hookrightarrow Standard cell: General Purpose Logic
 - \hookrightarrow high V_t : low leakage, ~~high~~ speed slow (critical paths)
 - \hookrightarrow low V_t : high leakage, fast (critical paths)
 - "Back-bias" versions: dynamically adapt speed to external conditions

Special Purpose Cells: Clock Tree, Delay

Cells, Clock Gating, Low Power Cells

- During synthesis: Clock signals are considered as ideal clock. Designer defines period, skew.
- Usage of Clock gating cells (based on activity simulation) \Rightarrow minimize power consumption

\hookrightarrow clock enable & latch do.



enable 1 to clock D-flipflop ko janiyon power consumption reduced.

- During P&R tools, clock tree is generated by tools in order to fulfil expected behavior.
- \Rightarrow clock buffers used (in a transition time t_{th} t_{max}) with specific timing constraints
- P&R tools may insert delay cells to meet hold timing constraints (output stable after clock)

• In multi-domain power supply designs, "level adapter cells" are inserted between cells of different power domains.

DATE	MONTH	YEAR

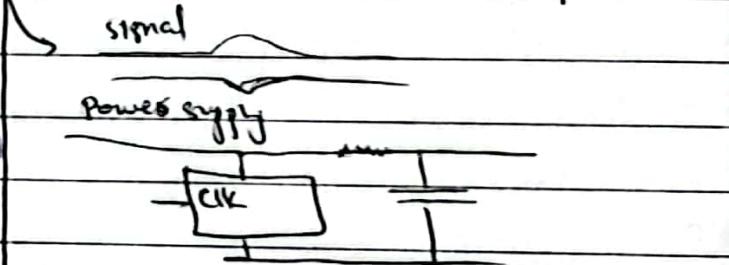
"Power switch cells": - to completely on/off blocks (power block)

Tools handles this automatically.

Special Purpose Cells: Physical only cell

Physical only cells: - cells that have no logic behavior. They are only needed by physical constraints of layout.

- filler cells: - fills holes between standard cells. Ensures supply & well continuity.
- well tap cells: connects Nwell to VDDP, VSSP, ~~VSSA~~ to GND.
- antenna cells: used to limit antenna effect during manufacturing.
- decap cells: used to limit GND & Power bounce called IR drop.



\hookrightarrow during transition, voltage dropped, propagation time would change not what you expected

\hookrightarrow near D-flipflop him ne decap logo dia

\Rightarrow more D-cap more leakage current

Why he jaga thi ni fasty dehna. Antenna effect:-

During manufacturing phases, plasma etching leads to charge accumulation in conductor layers already created.

\hookrightarrow if this layer is floating (not connected to drain/sources)

of transistors/wells) and connected to gates then gate oxide may break
↳ breakage of transistor possible due to this long charge
↳ α area of gate

⇒ Place & route tools are able to evaluate this problem & correct it.
↳ Estimate risk of breakdown, based on areas of metal/gates/implants connected to conductors.
↳ Correction 1:- add & connect specific diode (^{Antenna} cells) in order to avoid floating layers during manufacturing.
↳ Correction 2:- modify the routing in order to connect long lines of metal only during last step of manufacturing.

Digital Integrated Circuit Testing

Goal of Test

↳ Filtering out defective devices during manufacturing.

↳ more you want to detect faults, more it costs.

↳ Tests are done at wafer & packaging level

↳ Specification-oriented test, - checks conformance to design specifications

↳ application-oriented test, - checks design works properly in its application environment

↳ Structural test, - checks that there is no physical defect in chip.

* Structural test is more efficient, more easy to implement with generic methods.

STRUCTURAL TEST

what kind of defects?

• hard shorts, hard open

• sensitive bridge, resistive shorts

• wiring defects, component (transistor) defects

* Fault :- fault is the undesired behavior of a chip as a result of defect

* Fault models should -

↳ accurately reflect the effect of defect

↳ represents defects that are typical for technology used

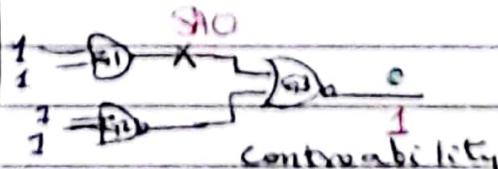
↳ be easy to implement in tools

The "Stuck-at" fault model
we assume that the only defect is a node that is stuck at low/high logic level.

we assume there is only 1 "stuck-at" fault in circuit.

⇒ no need to find where defect is, input send to, output not good (reject the chip)

⇒ How to detect stuck-at "0" "between G1, G2, G3" -



Try to put 1 on faulty node.

Output of G1 (1) \rightarrow both inputs, '1'

Observability

G3 used to transmit value

Put zero on G3 input; G2 (1, 1)

Stuck-at Model:-

↳ usable for any combinational block based on simple combinational gates.

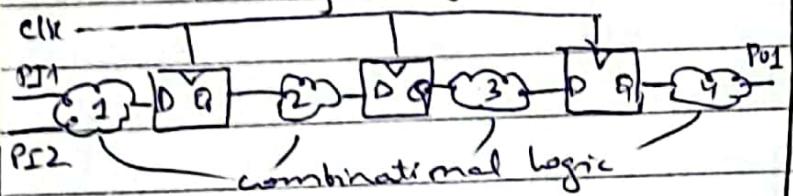
↳ inputs are called primary inputs, outputs are called primary outputs.

↳ 100% coverage (stuck-at-fault) can be achieved

↳ for any combinational netlist algorithmic tools can compute a test program :-

- test program is a set of test vectors
- test vectors is union of stimuli on primary inputs and expected values at primary outputs.

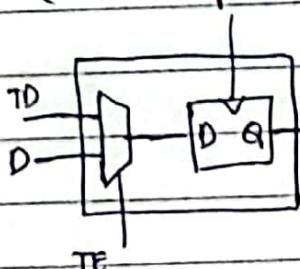
How to get full observability & full controllability in sync. circuits?



- a digital circuit is build of combinational blocks synchronized by D flip-flop
- only few primary inputs or outputs are usable
- some combinational blocks are completely isolated from PIs or POs

Scan-Chain Insertion

(scan flip flop)



- Each DFF replaced by Scan-DFF.

- In test mode ($TE=1$)

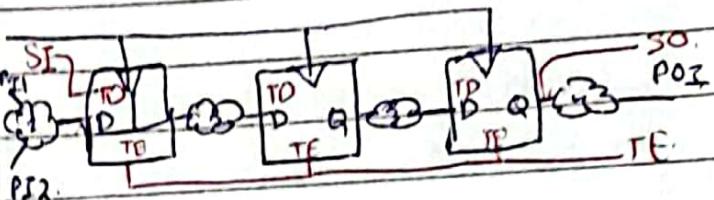
Input of flip flop replaced by TD input.

- ScanDFFs are chained in long shift register.

• A new SI serial input controllability

• A new SO output is used for observability DATE MONTH YEAR

• A test enable mode is inserted



• A test program consists of loops of following procedure:-

- $TE=1$; test vector is loaded via SI using shift register.
- $TE=0$; a one cycle computation is done in normal mode. All registers are loaded by computed values.
- $TE=1$; Shift register dumped via SO; results are compared to the expected results.

⇒ Scan chain insertion can be fully automatic (during synthesis).

⇒ test vector generation can be fully automatic (after synthesis).

⇒ Warning:- Scan-bkst reduces performances (lower clock frequency; higher power consumption).

Fault model ke jaga use nahi; such as carry skip adder if stuck at zero. to sort propagation time se pata chalna (not very useful/effective method).

Add some specific logic to help HCKY the test. (Adding chain of XOR gates)