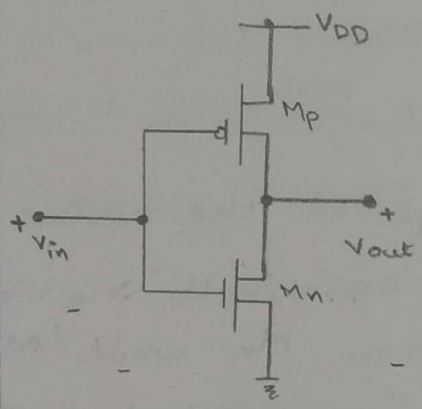
1. Explain D.C. characteristics and derive mid point voltage.

The CMOS inverter gives the basis for calculating the electrical characteristics of logic gates. The input voltage V_{in} determines the conduction states of the two FETs Mn and Mp. This produces the output voltage V_{out} of the gate. Two types of calculations are needed to characterize a digital logic circuit. A DC analysis determines V_{out} for a given value of V_{in}. In this type of calculation, it is assumed that V_{in} is changed very slowly, and that V_{out} is allowed to stabilize before a measurement is made. A DC analysis provides a direct mapping of the input to the output, which in turns tells the voltage ranges that define Boolean logic 0 and logic 1 values.



The CMOS inverter circuit.

PFET: $V_{TP} < 0$

$$\beta_p = k'_p \left[\frac{W}{L}\right]_p$$

NFET: $V_{TN} > 0$

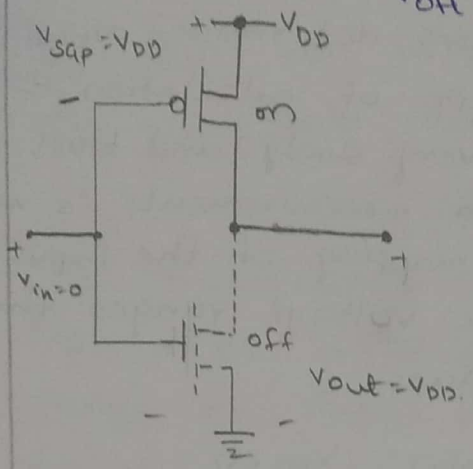
$$\beta_n = k'_n \left[\frac{W}{L}\right]_n$$

The second type of characterization is called a transient analysis. The input voltage is an explicit function of time. $V_{in}(t)$ corresponding to a changing logic value. The response of the circuit is contained in $V_{out}(t)$. The delay between a change in the input and the corresponding change at the output is the fundamental limiting factor for high-speed design.

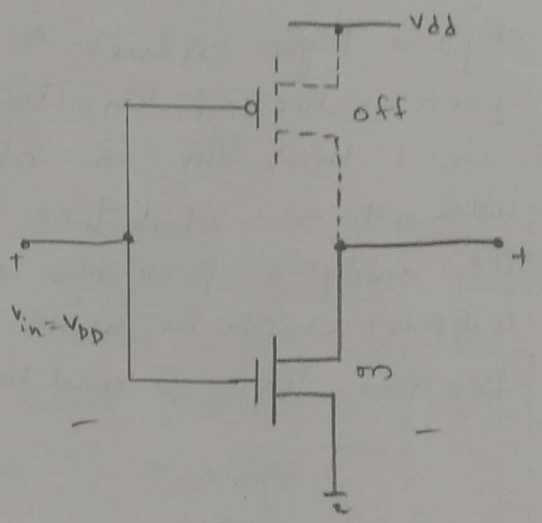
The DC characteristics of the inverter are potrayed in the voltage tranfer characteristic (VTC), which is a plot of V_{out} as a function of V_{in}. This is obtained

by varying the input voltage v_{in} in the range from 0V to V_{DD} and finding the output voltage V_{out}. The end point values are easily found. If v_{in} is equal to 0V as in figure, Mn is off while Mp is on. Since PFET is on, it connects the output voltage to the power supply and gives $v_{out} = V_{DD}$. This defines the output high voltage of the circuit as

$$V_{OH} = V_{DD}.$$



Low input voltage



High input voltage

i.e; the highest output voltage is the value of the power supply V_{DD}. The opposite case with $v_{in} = V_{DD}$ is illustrated in figure. This turns on Mn while Mp is in cut off. The output node is then connected to 0V (ground) through nFET, defining the output low voltage.

$$V_{OL} = 0V.$$

The logic swing at the output is

$$V_L = V_{OH} - V_{OL}$$

$$= V_{DD}.$$

Since this is equal to the full value of the power supply, this is called a full-rail output.

The VTC for the circuit is obtained by starting with an input voltage of $v_{in} = 0V$ and then increasing it up to a value of $v_{in} = V_{DD}$. This results in the plot. The

details can be understood by writing the device voltages in terms of input and output voltages.

$$V_{GSn} = V_{in}$$

$$V_{GSGP} = V_{DD} - V_{in}.$$

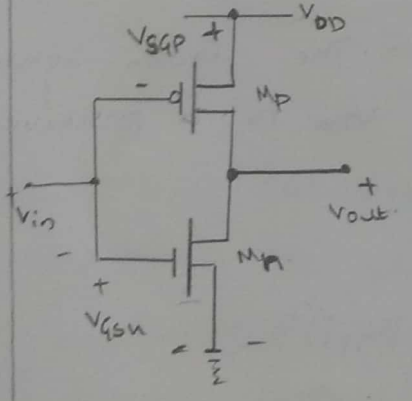
M_N is in cut off so long as $V_{in} \leq V_{Tn}$.
Since the output voltage is high with a value $V_{out} = V_{DD}$, any input voltage is in the range labeled as "0" can be interpreted as a logic 0 input.
M_P goes into cutoff when.

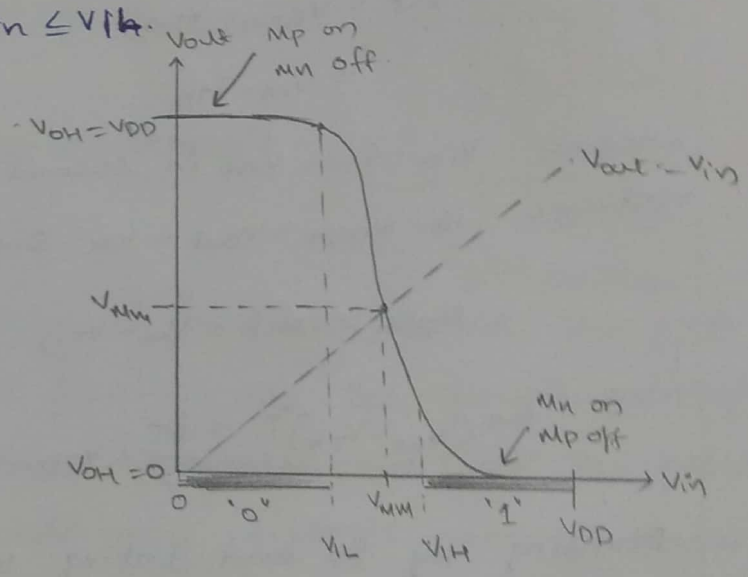
$$V_{in} = V_{DD} - |V_{TP}|.$$

For V_{in} greater than this value, $V_{out} = 0V$ since only the nFET is active. This shows that there is a range of input voltages that act as logic 1 input values as indicated by the "1" on the VTC.

The logic 0 and 1 voltage ranges are defined by the changing slope of VTC.

$$0 \leq V_{in} \leq V_{IL}.$$





$$V_{IH} \leq V_{in} \leq V_{DD}.$$

The voltage noise margins are

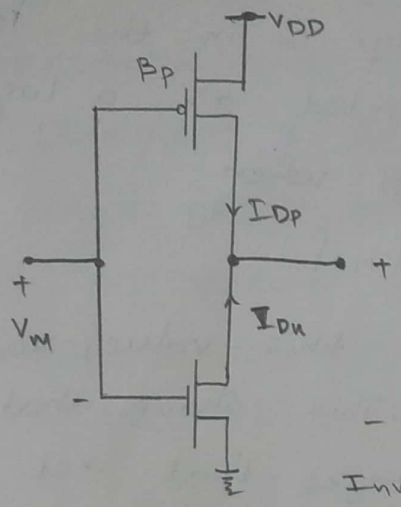
$$V_{NM_H} = V_{OH} - V_{IH}$$

$$V_{NM_L} = V_{IL} - V_{OL}.$$

for high and low states, respectively. The noise margins give a quantitative measure of how stable the inputs are with respect to coupled electromagnetic signal interference.

To calculate midpoint voltage $v_{in} = v_{out} = v_M$ is set. Equating the drain currents of the FETs gives

$$I_{Dn} = I_{Dp}.$$



Inverter voltages for V_M calculation.

but we need to find operating region (saturation or non-saturation) of each FET before we can use the expression.

$$V_{sat} = V_{Gsn} - V_{Tn}$$

$$= V_M - V_{Tn}$$

where $v_{in} = v_{Gsn} = v_M$ in second line. The drain-source voltage is $v_{DSn} = v_{out} = v_M$. Since V_{Tn} is a positive number.

$$V_{DSn} > V_{sat} = V_M - V_{Tn}$$

$$\frac{\beta_n}{2}(V_M - V_{Tn})^2 = \frac{\beta_p}{2}(V_{DD} - V_M - |V_{Tp}|)^2.$$

Dividing by β_p and taking the square root gives.

$$\sqrt{\frac{\beta_n}{\beta_p}}(V_M - V_{Tn}) = V_{DD} - V_M - |V_{Tp}|$$

Simple algebra then gives midpoint voltage as.

$$V_M = \left(V_{DD} - |V_{Tp}| + \sqrt{\frac{\beta_n}{\beta_p}}\, V_{Tn}\right) \Big/ \left(1 + \sqrt{\frac{\beta_n}{\beta_p}}\right)$$

VLSI Assignment

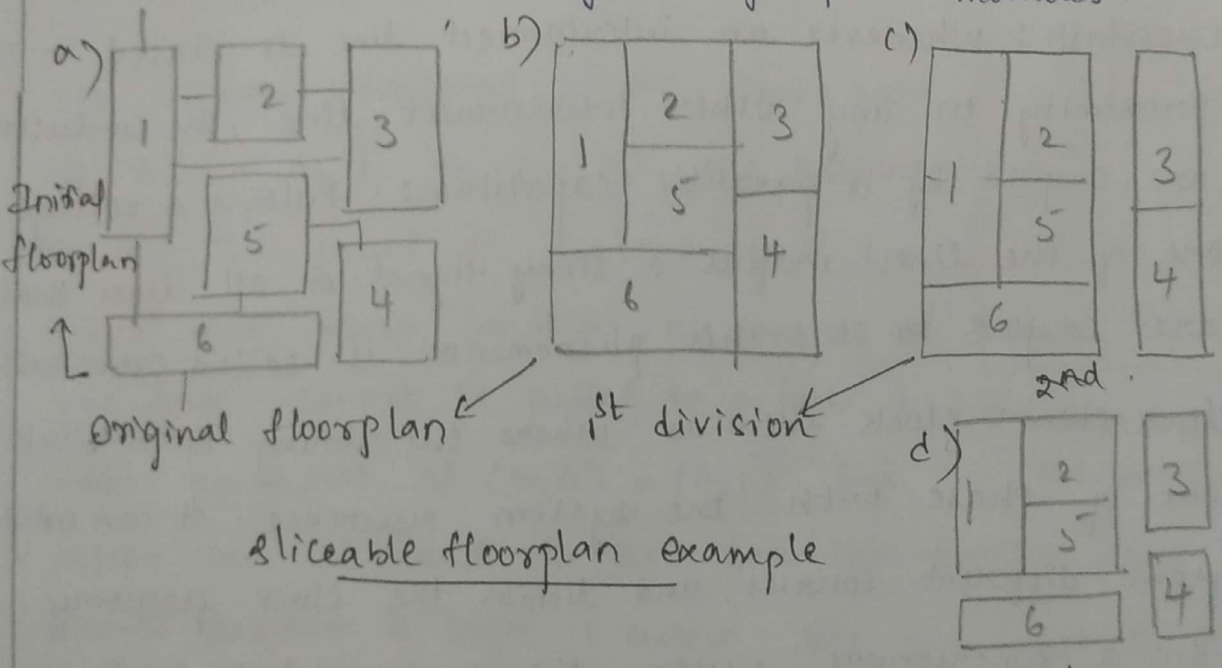
M. Jyothsna
160115737014
IT-1 4/4

(1) Write a short note on the following:

a) Floor planning and Routing

Floor planning is the starting step of the physical design in the VLSI design flow. In this step, it is planned to accomodate all the design components and their interconnects within a minimum area.

→ Sliceable floorplan: A sliceable floorplan is defined as either a single module, or a floor plan that can be partitioned into modules using a vertical or horizontal line that traverses a contiguous group of modules.

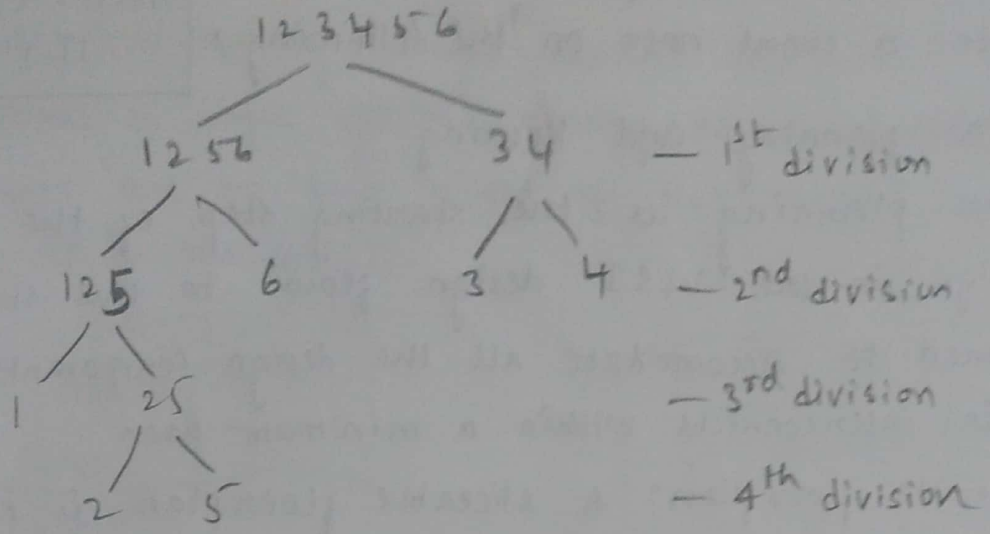


Sliceable floorplan example

A vertical cut line may be used to obtain the first division. The second division into the groups portrayed is obtained using two horizontal cut lines. This process may be continued until only seperate modules remain

→ Hierachially defined Floorplan: In this the divisions can be described using the tree structure drawn below. The numbers denote the connected module groups, and the slicing process can be seen by the division level

indicated at the bottom of each branch.

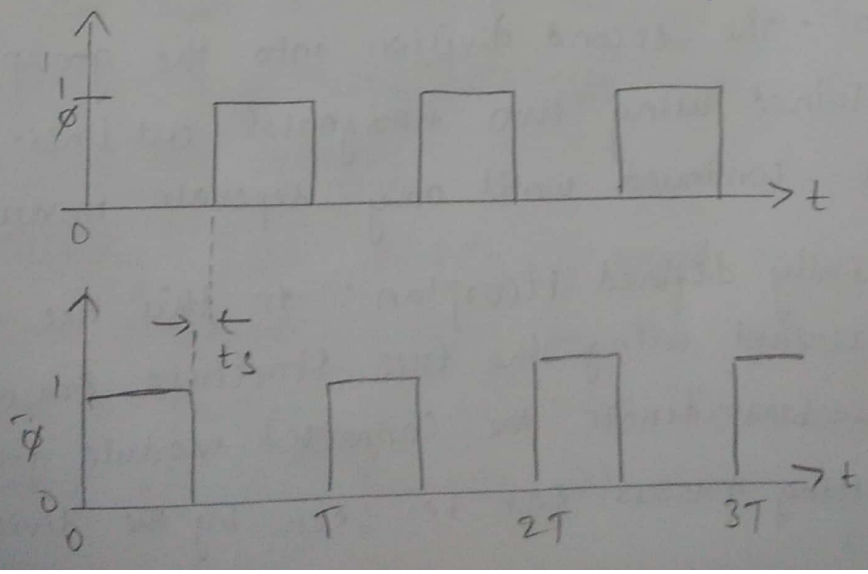


Binary division tree

b) Crosstalk and clock skew

Crosstalk: Whenever an interconnect line is placed in close proximity to any other interconnect line, the conductors are coupled by a parasitic capacitance. Pulsing a voltage on one of the lines induces a stray signal on all lines that are coupled to it. This phenomenon is called crosstalk.

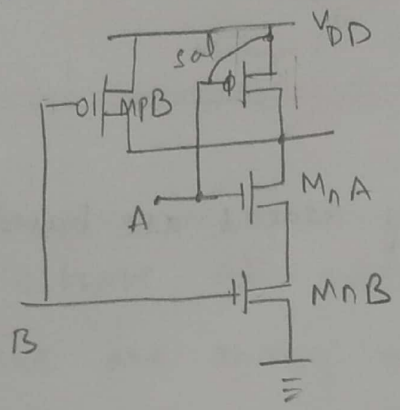
Clock skew: Clock skew is where the timing of a clock is out of phase with the system reference. It can originate from different sources, and limits the clock frequency. In a synchronous system, this is equivalent to limiting the data flow rate and the overall speed.



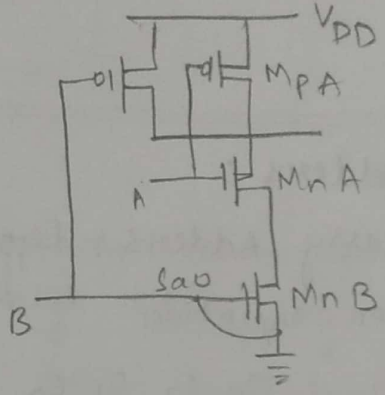
Clock skew

c) Testing of VLSI circuits

→ Gate level testing: Fault models are used to characterize features in logic gates. Creating different fault circuits allows one to find a set of test vectors that can be applied to the circuit.



a) Stuck at 1 fault



b) Stuck at 0 fault

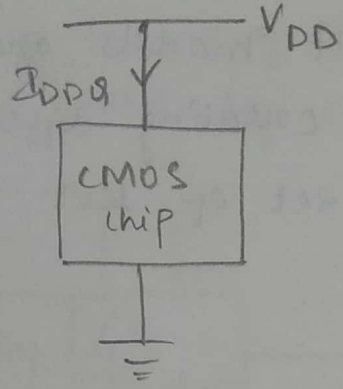
The normal response of the NAND gate is shown as F. In fig a), the response of the sa1 fault is denoted by F_{sa1}. Since MpA never conducts, the output of the gate cannot be pulled to a logic 1 with an input of (A, B) = (0, 1). This vector can be used to test for this problem since it should produce a logic 1 output. The sa0 fault in fig b) causes the output to behave as summarized in the F_{sa0} column. In this case, MpB is always on so that output is stuck at a 1. Using an input vector of (A, B) = (1, 1) would find this fault.

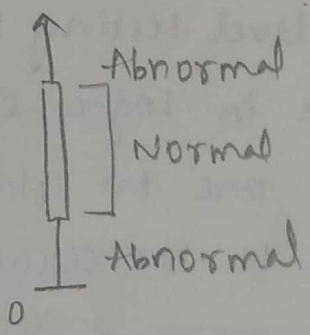
A	B	F	F_{sa1}	F_{sa0}
0	0	1	1	1
0	1	1	0	1
1	0	1	1	1
1	1	0	0	1

Function tables for the NAND2 gate

→ I_{DDQ} Testing: I_{DDQ} testing is based on the assumption that an abnormal reading of the leakage current indicates a problem on the chip. I_{DDQ} testing is usually performed at the beginning of the testing cycle. If a

a die fails, it is rejected and no further tests are performed.





Basic I_{DDQ} test

d) Fast Adders :

→ Ripple carry Adders : Ripple carry adders are based on the addition equation

$$\begin{array}{r} c_3\ c_2\ c_1\ c_0 \\ +\ a_3\ a_2\ a_1\ a_0 \\ +\ b_3\ b_2\ b_1\ b_0 \\ \hline c_4\ s_3\ s_2\ s_1\ s_0 \end{array}$$

where c_i represents the carry-in bit from the previous column. we will keep the 0th carry in bit c_0 for generality. An n-bit ripple carry adder requires n full adders with the carry out bit c_{i+1} used as in the carry-in bit to the next column.

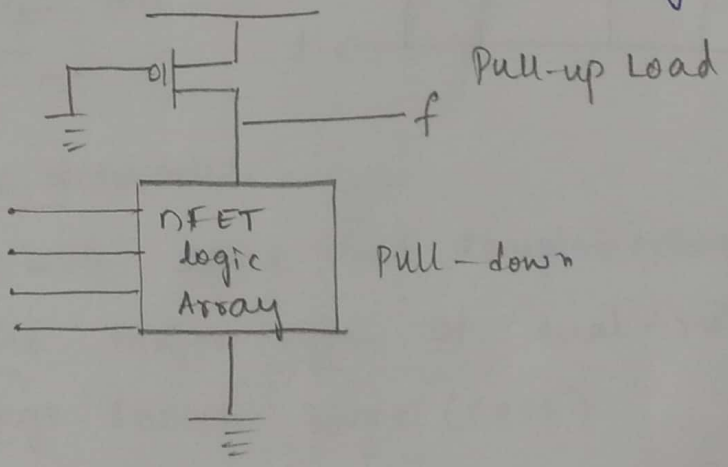
→ Carry Look-Ahead Adders : These are designed to overcome the latency introduced by the rippling effect of the carry bits. The CLA algorithm is based on the origin of the carry-out bit in the equation

$$c_{i+1} = a_i \cdot b_i + (a_i + b_i) \cdot c_i$$

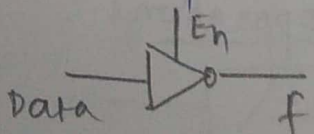
(2) Illustrate all advanced high speed logic designs.

There are many advanced high speed logic designs:

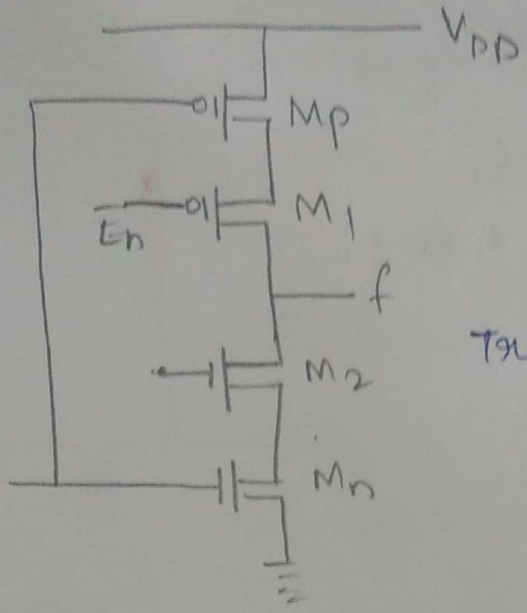
1. Pseudo-nMOS: The large area requirements of complex CMOS gates present a problem in high density designs, since two complementary transistors, one nMOS and one PMOS are needed for every input. Pseudo nMOS logic requires less transistors due to the fact that, only nFET logic block is required to create the logic.



2. Tri-state circuits: In normal logic circuits there are two states of the output, LOW & HIGH. If the output is not in the low state it is definitely in other HIGH state.



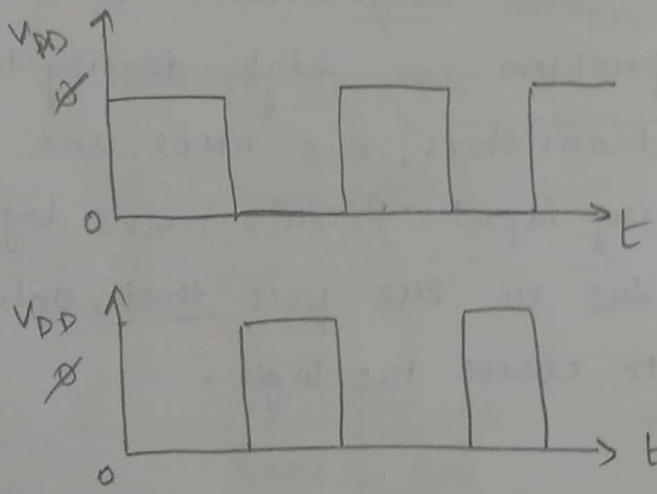
En	f
0	Z
1	Data



Tri state Inverter

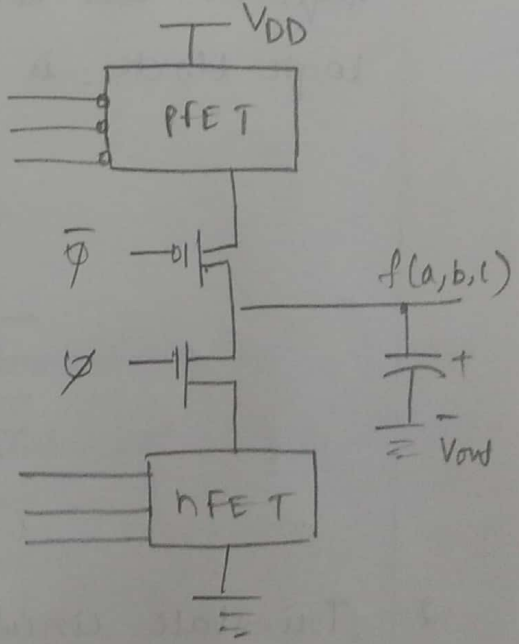
3. Clocked CMOS: The clock signal (ϕ) is a periodic waveform with certain period (T) and frequency (f)

$$f = \frac{1}{T}$$



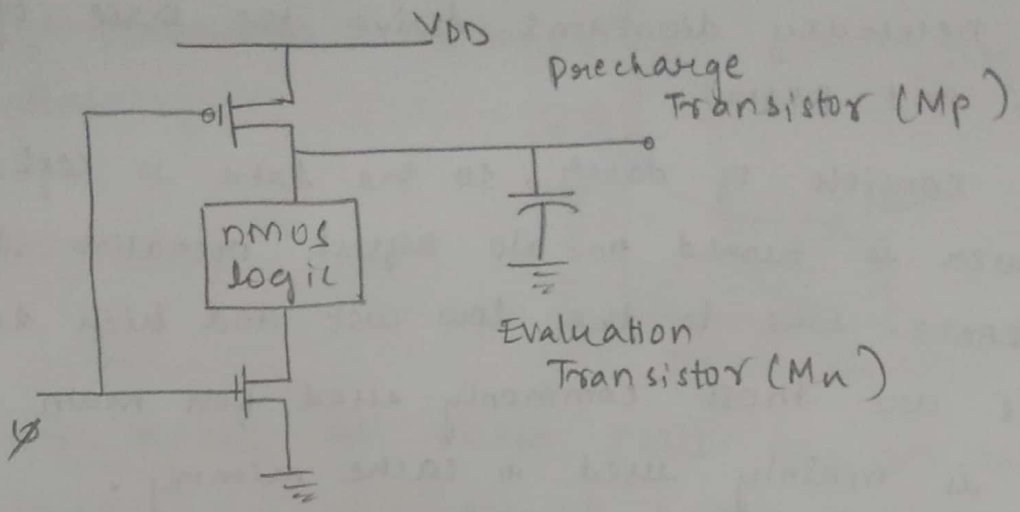
$$\phi(t) \cdot \bar{\phi}(t) = 0$$

$$\bar{\phi}(t) = V_{DD} - \phi(t)$$

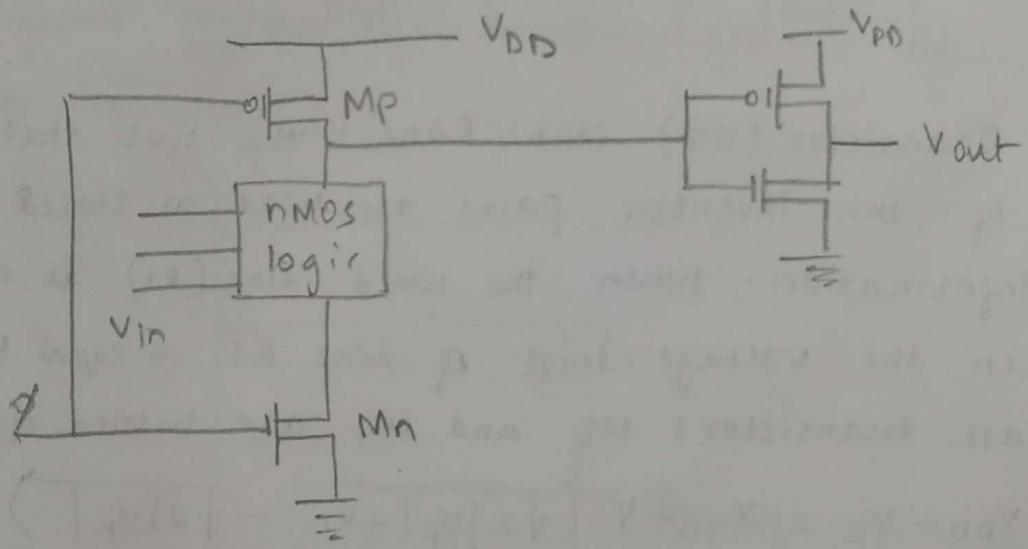


4. Dynamic CMOS:

It uses charge storage and clocking properties of MOS transistors to implement logic operations.



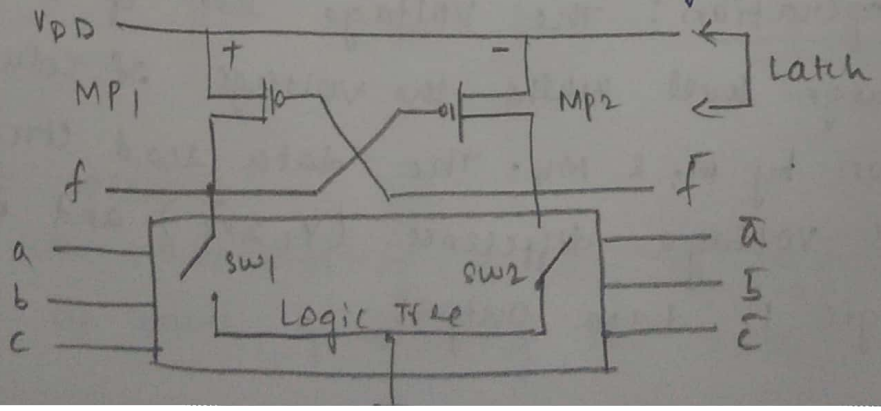
5. Domino Logic: It is a slightly modified version of the dynamic CMOS. Here, a static inverter is connected at the output of each dynamic CMOS logic block.



6. Dual-Rail logic Networks:

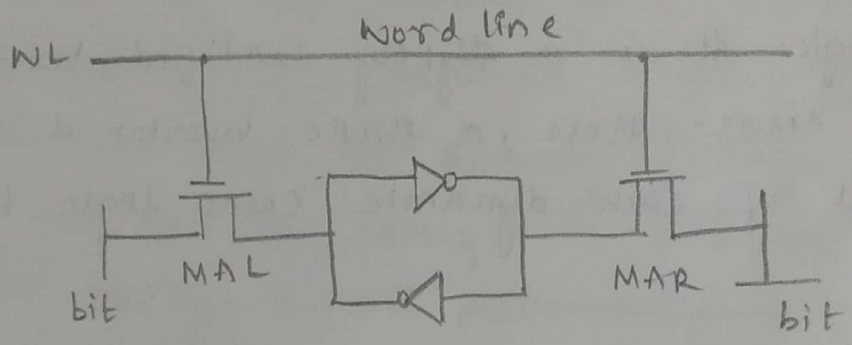
It handles both true and complementary signals at any time. The major types of dual-rail logic circuits:

1. Cascade voltage switch logic (CVSL)
2. Complementary pass Transistor logic (CPL).



(3) With necessary diagrams derive the three operations of SRAM and DRAM?

SRAM consists of datch, so the data is kept as long as power is turned on. No refresh operation is required in SRAMs. Due to the low cost and high density, DRAMs are most commonly used for main memory. SRAM is mainly used in cache memory.



Four Transistor (4T) static RAM : the two stable operations points of the inverter pairs are used to stored one bit piece information. When the word line (RS) is not selected ie. when the voltage level of line RS is equal to logic '0', the pass transistors M3 and M4 are turned off.

$$V_{DD} - V_c = V_{TO} + \gamma \left(\sqrt{2|\phi_F| + V_c} - \sqrt{2|\phi_F|} \right)$$

where $V_{DD} = 5V$, $V_{TO} = 1V$, $|\phi_F| = 0.6V$, $g = 0.4V^{\frac{1}{2}}$

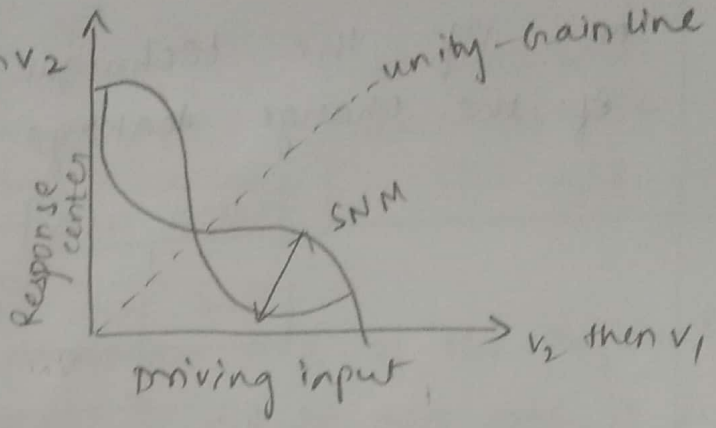
$= 3.5V$

Write '1' operation: The voltage level of columns $\bar{c}$ is forced to logic low by the data write circuitry. the driver transistor M_1 turns off.

Read '1' operation: The voltage level of column c retains its precharge level while the voltage of column $\bar{c}$ is pulled down by M_2 & M_4. The data read circuitry detects the small voltage difference ($v_c > v_{\bar{c}}$) and amplifies it as a logic '1' data output.

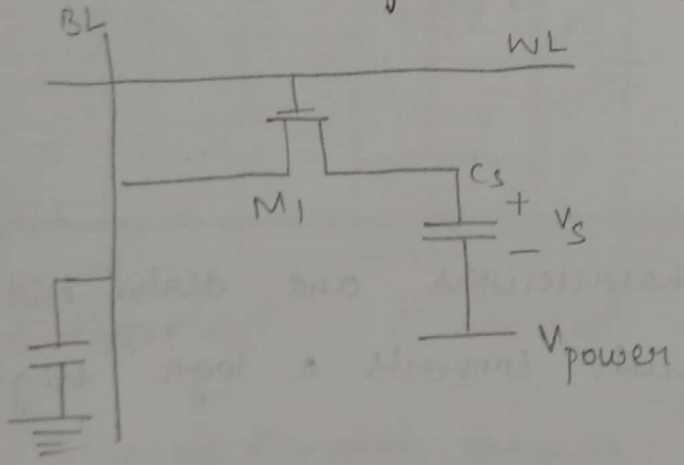
Design parameters :

1. cell stability
2. speed
3. Layout area

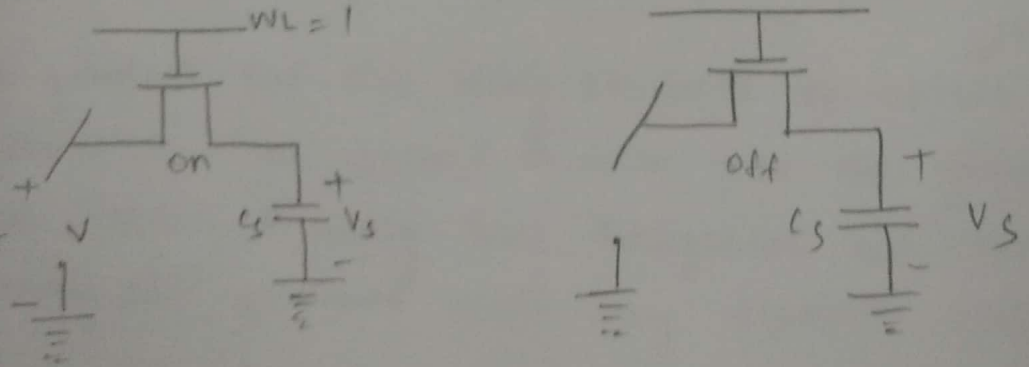


Dynamic RAMS: All DRAM cells requires a period refreshing of the stored data, so unwanted modifications due to leakage are prevented before they occur. the capacitor is used as primary storage device, so the DRAM cell can be realized in a much smaller area compared to the typical SRAM cell.

one-Transistor (1T) Dynamic RAM:



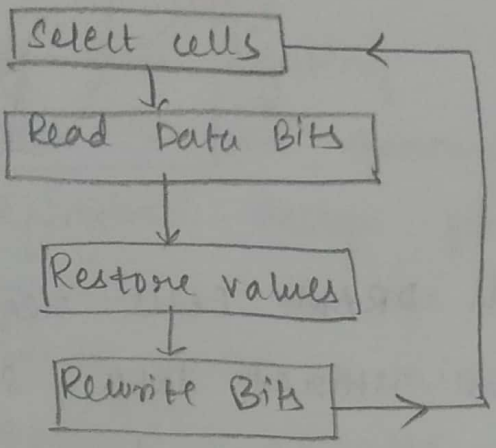
write and hold operations in a DRAM cell:



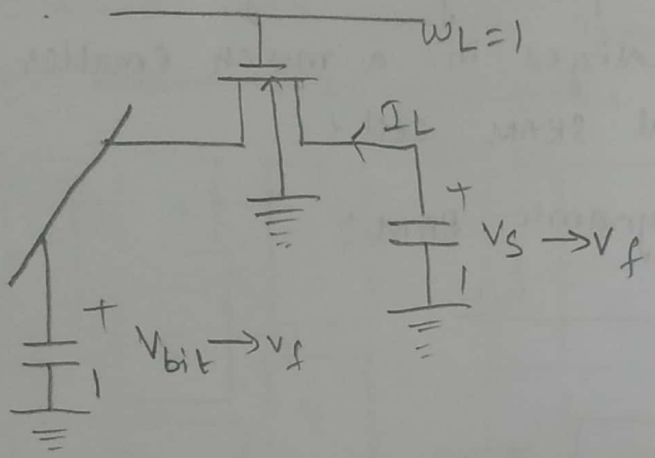
Refresh operation:

until the availability of applied power, memory units must be capable of storing the data. This operation is

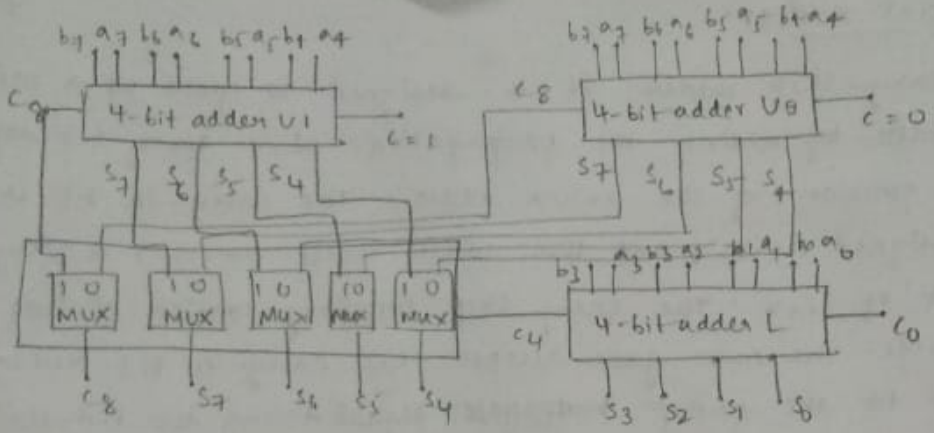
one of the technique that overcomes the limitations of the charge leakage problem



Read operation

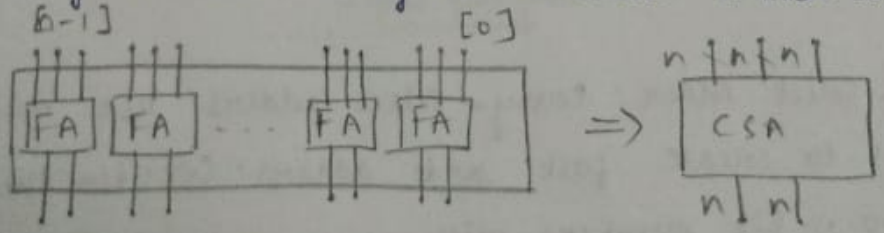


(4) Explain DC characteristics and derive



8-bit carry select adder

3. Carry-save Adders: Carry-save adders are based on the idea that a full adder really has three inputs and produces two outputs. While we usually associate the third input with a carry-in, it could equally well be used as a regular value. We can build an n-bit carry save adder by using n seperate adders. The name "carry-save" arises from the fact that we save the carry out word instead of using it immediately to calculate a final sum.



Creation of an n-bit carry save adder

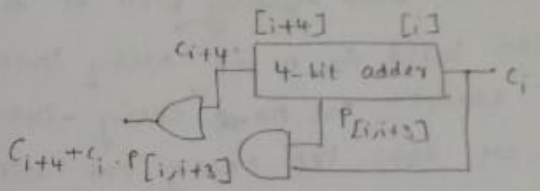
d) Fast Adders

1. Carry-skip adder: It is designed to speed up a wide adder by aiding the propagation of a carry bit around a portion of the entire adder. The carry-in bit is designed as c_i, and the adder itself produces a carry-out bit of c_{i+4}. The carry skip circuitry consists of two logic gates. The AND gate accepts the carry-in bit and compares it to the group propagate signal using the individual

$$P_{[i,i+3]} = P_{i+3} \cdot P_{i+2} \cdot P_{i+1} \cdot P_i$$

propagate values. The output from the AND gate is ORed with c_{i+4} to produce a stage output of

$$\text{carry} = c_{i+4} + P_{[i,i+3]} \cdot c_i$$



Carry-skip Logic

2. Carry-Select Adder: Carry-select adders use multiple narrow adders to create fast wide adders. Consider the addition of two n-bit numbers with $a = a_{n-1} \cdots a_0$ and $b = b_{n-1} \cdots b_0$. At the bit level, the adder delay increases from the least significant 0th position upward, with the $(n-1)$th requiring the most complex logic. A carry select adder breaks the addition problem into smaller groups. There are only two possibilities for the carry bit:

$$c_{n/2} = 0 \quad \text{or} \quad c_{n/2} = 1$$

A carry-select adder provides two seperate adders for the upper words, one for each possibility.