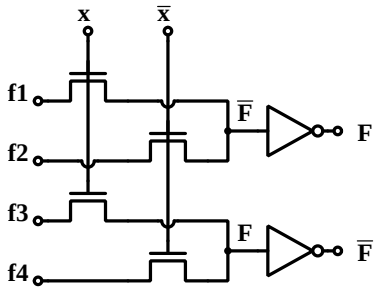
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

End Semester Examination

Wednesday	EE 671: VLSI Design	Time: 0930-1230
18-11-15	Autumn Semester 2015	Marks: 40

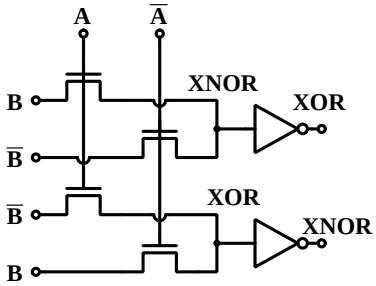
Q–1

The circuit on the right shows the basic structure used by CPL gates. This needs inputs in true and complement forms and provides output in true as well as complement form. Appropriate choices have to be made for x, f1, f2, f3 and f4 for producing the desired logic functions F and $\overline{F}$.



a) Given inputs A and B in true as well as complement form, show how you will connect these to the CPL structure shown above to generate the XOR and XNOR of A and B.

Soln.: When A = 1, XNOR is B, XOR is $\overline{B}$.
When A = 0, XNOR is $\overline{B}$, XOR is B. Therefore the following connection will generate the XOR-XNOR function in CPL.

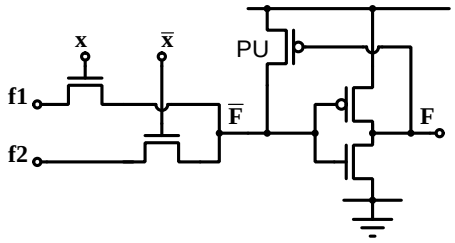


– **[2]**

b) Explain why the basic structure shown in the figure may lead to leakage in the inverters used at the output. Show how it can be prevented by using a PMOS pull-up transistor. (Give a transistor level circuit for this). Explain the pull-up action of the transistor for different output values.

Soln.: When a HIGH input is passed, the PMOS transistor in the inverter is supposed to be OFF and the NMOS transistor is ON, producing a LOW output. The basic structure shown uses NMOS transistors as pass elements. When a HIGH input is being passed, the output will be lower than the input by V_{Tn}. Therefore the HIGH value at the input of the inverter will be below V_{DD} at least by a V_{Tn}. This implies that the PMOS transistor in the inverter cannot be properly turned OFF, which leads to leakage current in the inverter.

This can be corrected by adding a pull up PMOS to the inverter as shown in the figure below:

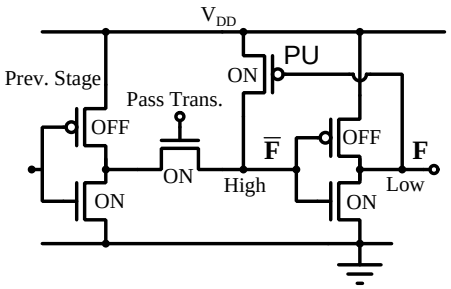


(Only the upper half of the structure is shown. The bottom half will have a similar pull up transistor added.)

Now, when a HIGH is passed to $\overline{\text{F}}$, the NMOS pass transistor is able to take this node only to $V_{DD} - V_{Tn}$. However, this is quite sufficient to take the output F to to a value below $V_{DD} - V_{Tp}$. This turns on the pull up transistor, which now charges up the node $\overline{\text{F}}$ all the way to V_{DD} and leakage is avoided.
When the output is HIGH, the pull up transistor is OFF. – **[3]**

c) The addition of a pull up transistor to reduce inverter leakage requires the transistor widths to be ratioed, otherwise the circuit may not work. Explain why this happens.

Soln.: Take the case when the previous input was HIGH and now we want to pass a LOW value. In the initial condition, the output F is LOW, and the pull up transistor is ON, as described above. Now if we want to take the node $\overline{\text{F}}$ to LOW, we have to fight the pull up transistor PU, which is trying to keep this node HIGH! We can take this node to LOW only if the NMOS transistor can sink more current than what the pull up transistor is sourcing. The equivalent circuit for this condition is shown in the figure below:



As can be seen in this condition, two series connected NMOS transistors need to pull down the node $\overline{\text{F}}$ to LOW, which is being kept at High by the pull up PMOS. This is like a pseudo NMOS NAND gate. The NMOS transistors have to be sufficiently wide to pull the node $\overline{\text{F}}$ low, in spite of the pull up transistor being ON.

Once this node is taken to LOW, the output F will go to HIGH, which will turn the pull up transistor OFF. Thus there is no static power consumed by the circuit. Only during the transition, sufficient drive is required to take the node $\overline{\text{F}}$ to low and this requires that transistor widths be ratioed appropriately. – **[3]**

– **[Q1: 2 + 3 + 3 = 8 marks]**

Q–2 **a)** The delay of a single logic stage may be modeled as

$$d = gh + p$$

where the delays are normalized to the unit inverter delay. The only variable which depends on the size of the gate is h. Why are the logical effort g and the parasitic delay p in this expression independent of the size of the gate?

Soln.: The logical effort compares the capacitive load presented by a gate compared to an inverter with equivalent drive. Therefore, it is not dependent on the size of the gate.

The parasitic delay is largely caused by transistor parasitic capacitances. These are proportional to transistor widths. As we change the size of the gate, the drive it can provide is scaled, but so is the capacitive load in the same ratio. Therefore the parasitic delay associated with the gate remains unchanged as we scale the size of the gate. – **[2]**

b) For multi-stage logic, the total delay is minimized when the stage effort $f = gh$ is the same for all stages. However, some times the total delay may be reduced further by inserting inverters in the logic chain. Derive the relationship which gives the optimum stage effort for each stage when we are free to insert a number of inverters in the logic chain.

Soln.: Let the multistage effort be F. assume that there are M logic gates in the chain. We can add (M-N) inverters to this chain, effectively making a chain of N gates, while the functionality remains the same. (Given that we can accept the output even if it is complemented).

The total delay of this chain will be given by the delay of the m gates plus the delay of (m-n) inverters. Thus,

$$D = \sum_1^M f_i + \sum_1^M p_i + (N-M)f_{inv} + (N-M)p_{inv}$$

The optimum sizing for the logic gates in this new chain will be when each stage has the same single stage effort $= F^{1/N}$. Thus, $f_i = f_{inv} = F^{1/N}$. Then the total delay is:

$$D = MF^{1/N} + \sum_1^M p_i + (N-M)F^{1/N} + (N-M)p_{inv}$$

or

$$D = NF^{1/N} + (N-M)p_{inv} + \sum_1^M p_i$$

Let the optimum stage effort $F^{1/N} \equiv \rho$. Then

$$D = N\rho + (N-M)p_{inv} + \sum_1^M p_i$$

We want to optimize this delay with respect to N. Therefore, we take its derivative with respect to N and equate this to 0. or

$$\rho + N\frac{\partial \rho}{\partial N} + p_{inv} = 0$$

$F = \rho^N$, $\ln F = N \ln \rho$, so $\ln \rho = 1/N \ln F$. Therefore,

$$\frac{1}{\rho}\frac{\partial \rho}{\partial N} = -\frac{1}{N^2}\ln F$$

$$\text{so} \qquad N\frac{\partial \rho}{\partial N} = -\frac{\rho}{N}\ln F = -\rho \ln \rho$$

Substituting in the optimization equation, we get

$$\rho - \rho \ln \rho + p_{inv} = 0$$

This the optimum stage ratio ρ is defined by the relation

$$p_{inv} + \rho(1 - \ln \rho) = 0$$

This equation needs to be solved iteratively to get the optimum stage ratio. – **[4]**

c) Assume that in a given process, the value of the parasitic delay p_{inv} of an inverter is 1.6. How many inverters should we use in a chain to minimize the total delay while driving a load equivalent to 96 inverters, assuming that the input can drive a single minimum sized inverter. (We don't care if the number of inverters is even or odd).

Soln.: The equation to be solved is

$$p_{inv} + \rho(1 - \ln \rho) = 0 \qquad \text{so } \ln \rho = 1 + \frac{p_{inv}}{\rho}$$

$$\text{Therefore} \quad \rho = \exp\left(1 + \frac{p_{inv}}{\rho}\right) = \exp\left(1 + \frac{1.6}{\rho}\right)$$

with $p_{inv} = 1.6$. This equation has to be solved iteratively. Starting with a trial value of 3, we get successive values for rho as:
4.6336,3.8394, 4.1236, 4.0069, 4.0524, 4.0343, 4.0414, 4.0386, 4.0397, 4.0393, 4.0395, 4.0394 ...

Thus $N = \ln(96)/\ln(4.0394) = 3.2694$. Since N must be integral, we should either have N = 3 or N =4.
Now for the inverter chain, G=1, since all stages are inverters with $g = 1$.
$B = 1$, since there is no branching. $H = C_{out}/C_{in} = 96$.
Therefore $F = 1 \times 1 \times 96 = 96$.

For 3 stages, the stage effort should be $96^{1/3} = 4.5789$.
The stage delay for each stage is $4.5789 + 1.6 = 6.1789$
Thus the total delay is $6.1789 \times 3 = 18.54$.

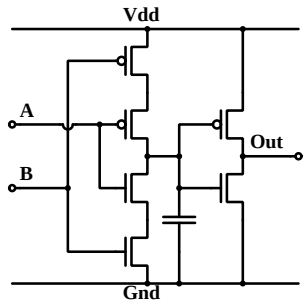
For 4 stages, the stage effort will be $96^{1/4} = 3.1302$
The stage delay is $3.1302 + 1.6 = 4.7302$
The total delay will be $4.7302 \times 4 = 18.92$.

Thus a 3 stage buffer will minimize delay. One can see that the difference in delay is marginal, and in fact both 3 and 4 stage buffers are appropriate. (4 stage buffer will be non-inverting). – **[2]**

– **[Q2: 2 + 4 + 2 = 8 marks]**

Q–3 **a)** What is a C element? Describe the working of a dynamic C element and show that it works effectively as an AND function of events on its inputs.

Soln.:



The C element uses 2 PMOS and 2 NMOS transistors, all in series. The output of this 4 transistor structure goes to a capacitor, which stores state. The logic level on the capacitor is inverted to generate the final output.

When A and B are unequal, one of these should be '0' and the other '1'. Thus, one out of the two series connected NMOS transistors is off. Similarly, one out of the two series connected PMOS transistors is off. In this case both the pull up and pull down are disabled and the the capacitor holds its previous value. So the output remains at its previous value.

When both inputs are '0', the P channel transistors are ON while the N channel transistors are OFF in the first stage. The capacitor charges up to '1' and so the output is '0'.

When both inputs are '1', the N channel transistors are ON while the P channel transistors are OFF. The capacitor is discharged and the output goes to '1'.

Thus when the inputs are equal, the output is the same as inputs. When inputs are unequal, the C element holds its previous state.

Assume that both inputs are at '0' initially. So the output is also at '0'. If either input goes to '1', the inputs become unequal and the output holds its previous value of '0' Subsequently, if the other input also goes to '1', the output will be driven to '1'.

If the other input does not go to '1' but the first one returns to '0', the inputs are equal to '0' and the output remains '0'.

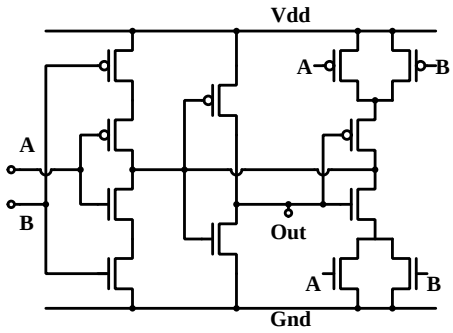
Thus the output has an "event" only when *both* inputs have had an event.
Similarly, if initially both inputs are 1, the output will be '1'. If either input goes to '0', the inputs become unequal and the output holds its previous value of '1'

Subsequently, if the other input also goes to '0', the output will be driven to '0'. The output again has an "event" only when *both* inputs have had an event.

Thus the C element performs an "AND" on events. – [**2**]

b) Show how we can convert the dynamic C element into a static circuit. Describe the operation of the static C element.

Soln.: If we want a static C element, we make use of the fact that the state needs to be stored only when the inputs are unequal. So we can use the following circuit:



When inputs are unequal, at least one of the inputs must be a '0' and the other must be '1'. Therefore the parallel PMOS and parallel NMOS transistors used to power the last inverter must be on, since at least one of the transistors in the pair should be on.

So the last inverter is powered and forms a latch with the first inverter. This provides static storage of the state instead of the capacitor.

When both inputs are '0', the series connected PMOS transistors in the first stage are ON, while the series connected NMOS transistors are off. This outputs a '1', which is inverted to '0' by the first inverter.

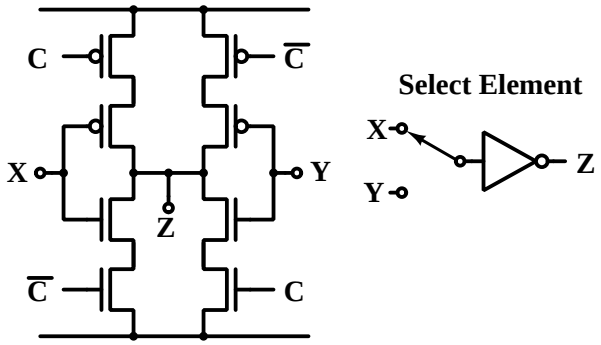
The parallel connected PMOS transistors are ON, while the parallel connected NMOS transistors are OFF. The PMOS pull up of the second inverter is ON because the output of the first inverter is '0'. So the output of the second inverter is '1'. This helps the output of the first stage, to which it is shorted. and the static C element output is '0' when both inputs are '0'.

If both inputs are '1', the series connected PMOS transistors in the first stage are OFF, while the series connected NMOS transistors are ON.

This outputs a '0', which is inverted to '1' by the first inverter.
The parallel connected PMOS transistors are OFF, while the parallel connected NMOS transistors are ON. The NMOS pull down of the second inverter is ON because the output of the first inverter is '1'. So the output of the second inverter is '0', which helps the output of the first stage, to which it is shorted. and the static C element output is '1' when both inputs are '1'. Thus the behaviour of static C element is the same as that of the dynamic C element. – **[2]**

c) Draw the circuit for a select element and show how it works. (The select element is just an inverting mux).

Soln.:



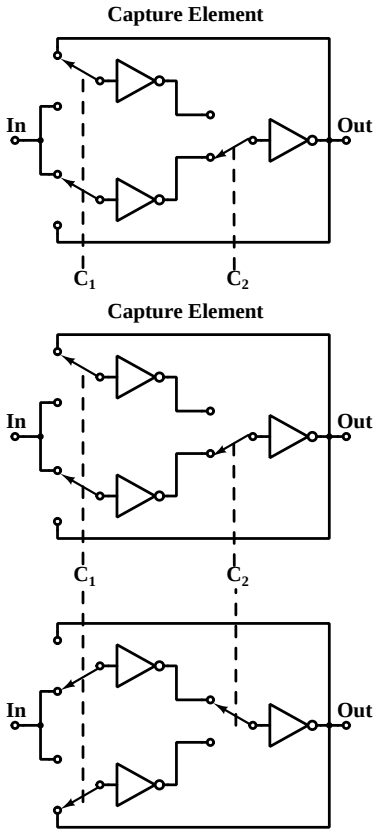
The figure above shows a select element.
When C = '0', the PMOS and NMOS connected to supply and ground in the left half are ON while the right half is OFF. So $Z = \overline{X}$.

When C = '1', the PMOS and NMOS connected to supply and ground in the right half are ON while the left half is OFF. So $Z = \overline{Y}$.

Thus, the behaviour of the circuit is that of a two way switch followed by inversion. – **[1]**

d) Describe the working of an event sensitive data latch using three select elements and two control inputs. Show how the use of two control inputs permits us to make this latch event sensitive.

Soln.:



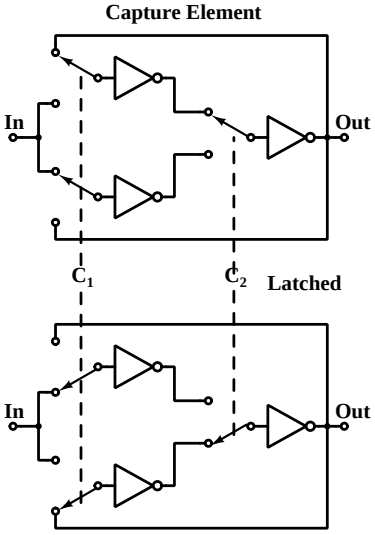
The circuit has two control inputs C_1 and C_2. Assume that C_1 = '1' puts the selectors in the up position, while C_2 = '1' puts its selector in the down position.

When both control inputs are '1', the two switches on the left will be up and the right switch will be down. Data will flow through the lower inverter on the left and output inverter. The circuit acts as a buffer.

When both control inputs are '0', the two switches on the left will be be down and the switch on the right will be up. Data will flow through the upper inverter on the left and the output inverter. So the circuit again acts as a buffer.

Thus when the two control inputs are equal, the circuit behaviour is the same whether $C_1 = C_2 = 0$ or $C_1 = C_2 = 1$

Let us see the case when the two control inputs are unequal.



When $C_1 = 1, C_2 = 0$, and all switches are up. the upper inverter on the left and the output inverter will form a latch.

When $C_1 = 0, C_2 = 1$, all switches are down. the lower inverter on the left and the output inverter will form a latch.

In either case the data present at the input when this condition occurred will be latched and the output will be isolated from the input.

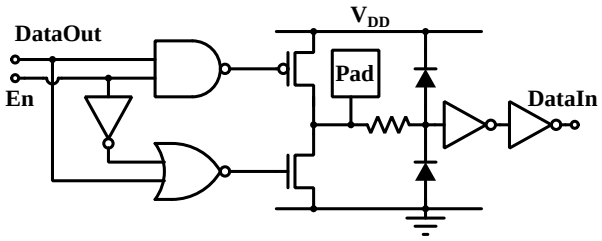
Again, the circuit behaviour is the same when the two control inputs are unequal, whether $C_1 = 1, C_2 = 0$ or $C_1 = 0, C_2 = 1$.

Thus the circuit behaviour is controlled by equality or inequality of C_1 and C_2 and not on their values, which can be 0 or 1. Now any event on either of the inputs will change the equal state to unequal or *vice versa*. Thus the circuit behaviour will change whenever there is an event, irrespective of the value of the control inputs. This makes the latch event sensitive. – **[3]**

– **[Q3: 2 + 2 + 1 + 3 = 8 marks]**

Q–4 **a)** Describe a circuit for bidirectional I-O through a pad. The output part of the circuit can be described at the logic level using NAND-NOR logic.

Soln.: A bidirectional I-O pad has an output circuit as well as an input circuit. The output circuit should be tri-stated when input is being performed. The input circuit need not be disabled during output, because we can always ignore its state when performing output. This functionality can be implemented with the following circuit:



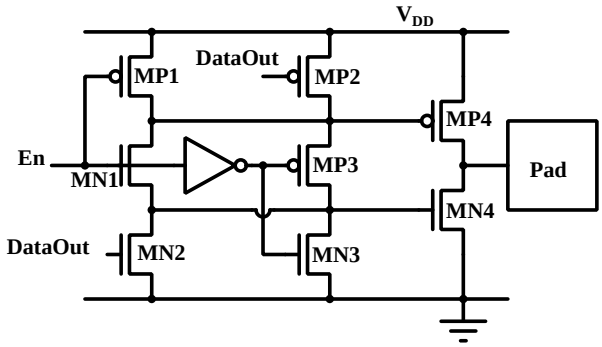
During Input, the Enable signal En is '0'. This forces the NAND output to 1, which turns the PMOS of the output buffer OFF. $\overline{\text{En}}$ is '1', which is applied to the NOR. It forces the NOR output to '0', which turns the NMOS of the output

buffer OFF. Thus when En is '0', the output buffer is tri-stated. The input signal is buffered and made available at DataIn.

When En is '1', both the NAND and NOR act as inverters. The inverted value of DataOut is applied to the output buffer, which inverts it again. Thus the PAD is driven to the value of DataOut. – **[2]**

b) The NAND-NOR functions used for bidirectional I-O can be combined in a single compact circuit. Give the transistor level circuit for this and show its functional equivalence to the discrete NAND-NOR combination.

Soln.: The compact buffer circuit is given below:



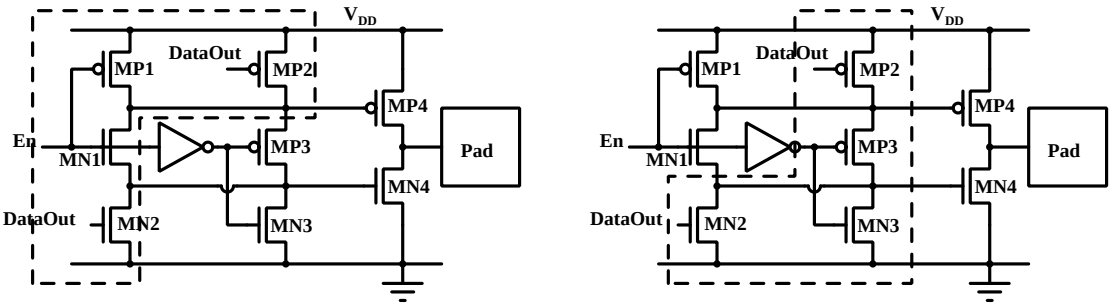
When the Enable signal (En) is '0', MP1 is ON, while MN1 and MP3 are off. Thus the pull up circuit is ON while pull down is disconnected from the upper output line which drives the gate of output buffer transistor MP4. This line is therefore HIGH and MP4 goes OFF.

Also, When En is '0' and $\overline{\text{En}}$ is '1', MN1 is OFF, MP3 is OFF and MN3 is ON. Thus the pull up circuit on the lower line driving MN4 is disconnected, while MN3 pulls it down to '0'. This turns MN4 OFF.

Thus, when En is '0', both MP4 and MN4 are off and the output buffer is tri-stated as desired.

When En is '1' and $\overline{\text{En}}$ is '0', MP1 and MN3 are OFF, while MN1 and MP3 are ON. MN1 and MP3 short the two output lines, which are now controlled by MP2 and MN2. These form an inverter, which takes the lines driving MP4 and MN4 to $\overline{\text{DataOut}}$. This is further inverted by the output buffer MP4-MN4, so DataOut appears at the output pad.

The circuits below show the equivalence of the compact circuit to NAND and NOR respectively in the original logic circuit.

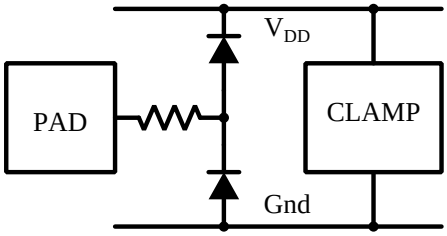


The dotted line enclosure in the circuit on the left shows how MP1, MP2, MN1 and MN2 form a NAND circuit of En and Dataout as in the original logic circuit.

Similarly, the dotted line enclosure on the right shows how MP2, MP3, MN2 and MN3 form a NOR circuit of DataOut and $\overline{\text{En}}$, as was there in the original logic circuit. – **[3]**

c) Describe the input protection circuit using diodes.
Explain why a clamp circuit is required between the supply line and ground for this circuit to work effectively. – **[3]**

Soln.: The circuit below shows the input protection circuit along with a voltage clamp between V_{DD} and Gnd.



While the resistor and diodes are shown as discrete components here, in actual practice these are provided by single elements formed by diffusion resistors of P type in an N well and of N type in a P well. The P well is grounded while the N well is connected to the supply, which leads to the above circuit.

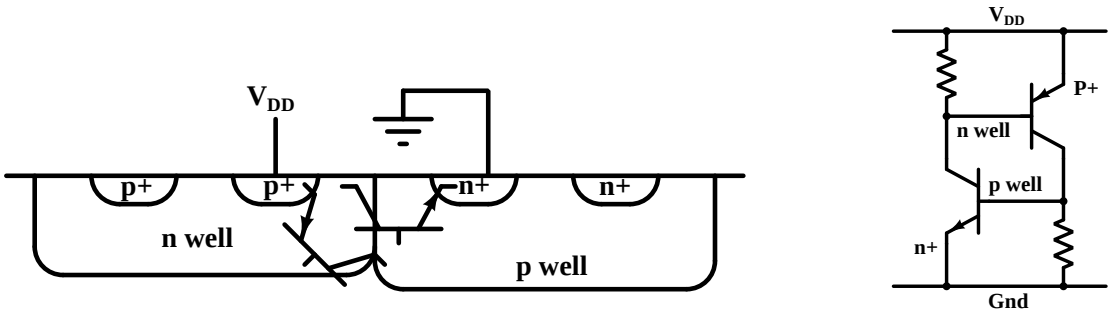
The input can see very high electrostatic voltages during handling. A negative voltage (compared to Gnd) will cause the diode connected to Gnd to be forward biased, which will draw high current. Consequently, most of the voltage will be dropped across the resistor. Thus the input voltage will be clamped to about -0.7 V.

If a large positive voltage appears at the pad, the upper diode will be forward biased, which will convey this voltage to the supply line. We need a clamp circuit which will limit this voltage to reasonable values, (not exceeding 2 to 3 times the value V_{DD}). If the clamp circuit is not there, high voltages will appear on the supply line, which can damage the circuit. The clamp circuit can be a field transistor connected as a diode. Other structures, such as a gated diode designed to breakdown at voltages in the vicinity of 2 to 3 times the supply voltage can also be used.

– **[Q4: 2 + 3 + 3 = 8 marks]**

Q–5 **a)** Show the latch up structure in a CMOS circuit. Using a bipolar transistor equivalent circuit, explain why latch up occurs, and how it can be prevented. Which are the layout design rules to be observed to prevent latch up?

Soln.: The diagrams below show the latchup structure and its bipolar equivalent circuit.



The source of an NMOS transistor connected to ground forms the emitter of the parasitic npn bipolar transistor, with the p well as its base and the n well as its collector.

The source of a PMOS transistor connected to V_{DD} forms the emitter of the parasitic pnp bipolar transistor, with the n well as its base and the p well as its collector.

The p well is normally grounded and the n well is connected to V_{DD}. However, the contact to the well for making connections to ground or V_{DD} may be remote, so the connection is resistive.

The circuit on the right shows how these bipolar transistors are connected to each other and to the resistive contacts of the wells to ground and V_{DD}.

In normal operation, the emitter of the npn transistor is connected to ground and so is the base (p well). Therefore this transistor is in cut off. Similarly, the emitter of the pnp transistor is connected to V_{DD} and so is its base (n well). Therefore this transistor is also in cut off.

However, if some base current is injected into either transistor due to some transient noise, that transistor will become active.

Let us say that some base current is injected into the npn transistor. This current will be multiplied by the β of the npn transistor and will appear as the collector current. This collector current will divide between the n well resistor and the base current of the parasitic pnp transistor. This base current will be multiplied by the β of the pnp transistor and will appear as its collector current. This collector current will divide between the p well resistor and the base current of the parasitic npn transistor. Thus, if the two β values are high enough, the two transistors will amplify their base currents and feed increasing values of base current into each other. A large amount of current will therefore flow between V_{DD} and ground, which can burn off the metallic connections to V_{DD} and ground. This is the latchup event in a CMOS structure.

To prevent latchup, we have to ensure that the current gain β of the two parasitic bipolar transistors is as low as possible and the well resistance is low, so that when the collector current flows in one of the transistors, most of it flows through the well resistor and as small a fraction as possible becomes the base current of the other parasitic transistor.

To ensure this, the base width of the two parasitic transistors should be large. This implies that the n^+ and p^+ regions should be well away from the well boundaries.

To make the well resistance small, a guard ring of p^+ type is put around p wells and a guard ring of n^+ type is put around the n wells. This helps in efficient collection of any current flowing in the wells.

The layout rules affected by this phenomenon are obviously the separation between well boundaries and n^+ and p^+ regions in the p well and n well respectively.

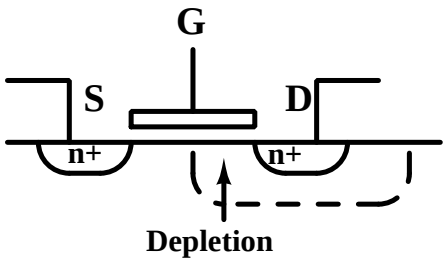
This also leads to the design rule which requires that ground connections to the p well and V_{DD} connections to the n well should be frequent and the distance between such contacts should not exceed a given value. This is done to limit the well resistance values.

– [**4**]

b) Describe the mechanism of punch through and avalanche break down in MOS transistors. Show how these put a limit on the minimum channel length which can be used for these transistors.

Soln.: Let us consider a MOS transistor to which a high value of V_{DS} is being applied.

Punch Through



The drain will have a depletion region around it. The field due to a junction is largely confined to the depletion region. As long as the edge of the depletion region is far away from the source, the drain field will not influence the injection action of the source. However, if the edge of the depletion region reaches the source, a high value of current will be drawn, irrespective of the gate voltage.

Consider an NMOS transistor. The drain has a large positive value applied to it. Due to the field in the depletion region, the voltage will drop across this region. So the p region of the substrate just next to the source will become positive, forward biasing the source junction. If this bias becomes substantial, the forward biased junction will inject a large current into the channel. This is a break-down condition, where the gate terminal has lost control of the amount of current flowing and the

drain itself induces a high current. This type of breakdown is called punch through.

When the edge of the depletion layer reaches the source terminal, $X_D = L$, where L is the channel length. Therefore, to avoid punch through, the channel length must be larger than the depletion width corresponding to the highest admissible drain voltage during normal operation.

Avalanche Breakdown
When a high voltage is applied to the drain, there is a high field in its vicinity. Consider an NMOS transistor as before. The instantaneous velocity of electrons is increased due to this field. This happens till the next collision occurs for the electron. If the velocity increases to a critical value before the next collision, this electron will knock out an electron from an atom at the next collision. Thus an extra electron-hole pair will be generated. This is called carrier multiplication. The newly created electron will also be accelerated by the drain field and may create even more electron hole pairs. This results in an avalanche of additional carriers and hence a sudden increase in drain current. This phenomenon is known as avalanche break down. To avoid this, the drain field must be kept below a critical value. This requires a low channel doping to spread the drain voltage over a higher distance. This is possible only if the channel length is sufficiently long. – **[3]**

c) What is electro-migration? What design rules are affected by this phenomenon?

Soln.: Electro-migration is the phenomenon of atomic movement when there is a high current density through the material. This can cause a metal line to break (in regions where there is a net loss of material) or to short (in regions where there is a net accumulation of material). To avoid this, the current density must be kept below some critical value. This leads to a design rule which requires that wider metal lines be used whenever the current through the conductor (transient or average) is high. – **[1]**

– **[Q5: 4 + 3 +1 = 8 marks]**

Paper Ends

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

Mid Semester Examination

Wednesday	EE 671: VLSI Design	Time: 0830-1030
09-09-15	Autumn Semester 2015	Marks: 25

Q–1 Consider a pseudo NMOS inverter where the ratio of K_n and K_p is β. ($K \equiv \mu C_{ox} W/L$).

a) Assuming perfect saturation for drain currents of transistors, find the input 'High' and output 'Low' logic levels in terms of the supply voltage V_{DD}, turn on voltages V_{Tn} and V_{Tp} and β.
(V_{iH} and V_{oL} are defined by the point on the transfer curve where the gain is -1, with PMOS in saturation and NMOS in linear regime.)

Soln.: Drain currents of PMOS and NMOS must be equal. So

$$\frac{K_p}{2}(V_{DD} - V_{Tp})^2 = K_n\left((V_i - V_{Tn})V_{out} - V_{out}^2/2\right)$$

$$\text{Let } V_1 \equiv V_i - V_{Tn} \quad \text{and } V_2 \equiv V_{DD} - V_{Tp}$$

$$V_2^2 = 2\beta\left(V_1 V_{out} - V_{out}^2/2\right) = 2\beta V_1 V_{out} - \beta V_{out}^2$$

$$\text{So} \quad \beta V_{out}^2 - 2\beta V_1 V_{out} + V_2^2 = 0$$

This leads to

$$V_{out} = \frac{2\beta V_1 \pm \sqrt{4\beta^2 V_1^2 - 4\beta V_2^2}}{2\beta} = V_1 \pm \sqrt{V_1^2 - V_2^2/\beta}$$

$$\text{So} \qquad V_{out} = Vi - V_{Tn} \pm \sqrt{(Vi - V_{Tn})^2 - (V_{DD} - V_{Tp})^2/\beta}$$

We must choose the minus sign, otherwise the NMOS will not be in linear regime and our starting equations will not apply. Thus,

$$V_{out} = Vi - V_{Tn} - \sqrt{(Vi - V_{Tn})^2 - (V_{DD} - V_{Tp})^2/\beta}$$

Taking the derivative with respect V_i and setting it equal to -1 gives

$$-1 = 1 - \frac{2(V_{iH} - V_{Tn})}{2\sqrt{(V_{iH} - V_{Tn})^2 - (V_{DD} - V_{Tp})^2/\beta}}$$

$$2 = \frac{(V_{iH} - V_{Tn})}{\sqrt{(V_{iH} - V_{Tn})^2 - (V_{DD} - V_{Tp})^2/\beta}}$$

Squaring and cross multiplying gives

$$4(V_{iH} - V_{Tn})^2 - \frac{4}{\beta}(V_{DD} - V_{Tp})^2 = (V_{iH} - V_{Tn})^2$$

This gives

$$3(V_{iH} - V_{Tn})^2 = \frac{4}{\beta}(V_{DD} - V_{Tp})^2 \qquad \text{So} \quad V_{iH} - V_{Tn} = \frac{2}{\sqrt{3\beta}}(V_{DD} - V_{Tp})$$

Therefore,

$$V_{iH} = V_{Tn} + \frac{2}{\sqrt{3\beta}}(V_{DD} - V_{Tp})$$

The corresponding output voltage is given by

$$V_{oL} = V_{iH} - V_{Tn} - \sqrt{(V_{iH} - V_{Tn})^2 - (V_{DD} - V_{Tp})^2/\beta}$$

$$= \frac{2}{\sqrt{3\beta}}(V_{DD} - V_{Tp}) - \sqrt{\frac{4}{3\beta}(V_{DD} - V_{Tp})^2 - (V_{DD} - V_{Tp})^2/\beta}$$

So

$$V_{oL} = \frac{2}{\sqrt{3\beta}}(V_{DD} - V_{Tp}) - \frac{1}{\sqrt{3\beta}}(V_{DD} - V_{Tp}) = \frac{V_{DD} - V_{Tp}}{\sqrt{3\beta}}$$

Thus

$$V_{iH} = V_{Tn} + \frac{2}{\sqrt{3\beta}}(V_{DD} - V_{Tp}) \quad \text{and} \quad V_{oL} = \frac{V_{DD} - V_{Tp}}{\sqrt{3\beta}}$$

– **[6]**

b) If $\mu_n = 2.2 \times \mu_p$, find the ratio of W/L values for N and P channel transistors, such that $V_{oL} \leq V_{Tn}$.

Soln.:

$$V_{oL} = \frac{V_{DD} - V_{Tp}}{\sqrt{3\beta}} \leq V_{Tn}$$

So

$$\beta \geq \frac{1}{3}\left(\frac{V_{DD} - V_{Tp}}{V_{Tn}}\right)^2$$

$$\frac{\mu_n C_{ox}(W/L)_n}{\mu_p C_{ox}(W/L)_p} \geq \frac{1}{3}\left(\frac{V_{DD} - V_{Tp}}{V_{Tn}}\right)^2$$

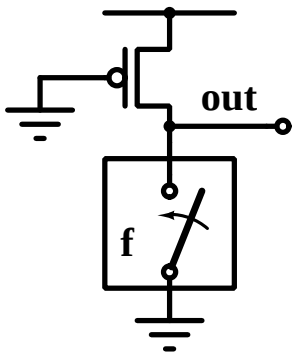
$$\frac{(W/L)_n}{(W/L)_p} \geq \frac{1}{6.6}\left(\frac{V_{DD} - V_{Tp}}{V_{Tn}}\right)^2$$

– **[2]**

– **[Q1: 6 + 2 = 8 marks]**

Q–2 **a)** How is the pseudo NMOS configuration modified to form the dual rail Cascade Voltage Switch Logic, such that static power dissipation is avoided?

Soln.: Consider the pseudo NMOS gate shown below. The switch f represents the entire series-parallel network of NMOS transistors used to generate the logic.



The output of this gate is $\overline{f}$ since the compound switch formed by NMOS transistors is on when f is 'TRUE'. Since the PMOS transistor is always ON, static power is dissipated when f is 'TRUE'. This could be avoided if instead of grounding the PMOS gate, we could drive it with f. Then the PMOS would be OFF whenever f is TRUE. If inputs as well as their complements are available, we can use an additional logic stage and generate f by using the complement of this network and feeding it with complemented inputs. (The complement network replaces series connections by parallel and parallel connections by series).

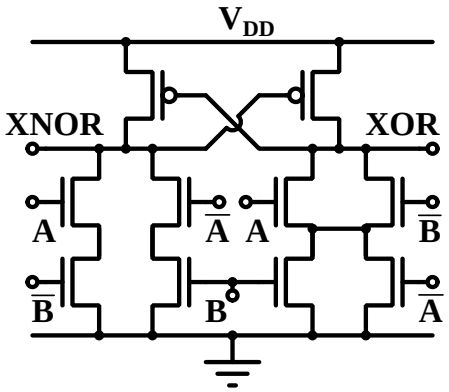
However, what about the static power consumed by this additional stage?

The additional stage will consume static power when its output is LOW, that is when f is 'FALSE'. But we already have $\overline{f}$ available as the output of the original logic stage – so we can use it to turn off the PMOS of the additional logic stage. Now neither stage will consume static power.

This kind of logic is called Cascade Voltage Switch Logic. It needs both true and complemented form of all signals, and generates the output in both true and complemented form. This is called dual rail logic. – **[3]**

b) Draw the transistor level schematic of an XOR/XNOR gate using Cascade Voltage Switch Logic (CVSL).

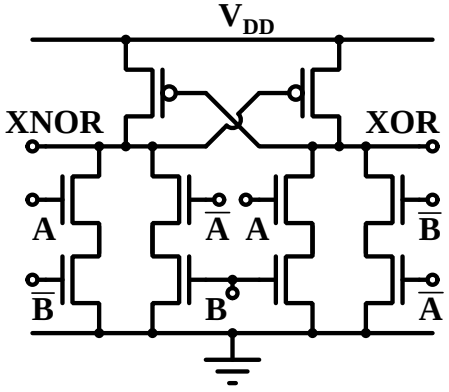
Soln.: The figure below shows the XOR/XNOR gate in CVSL.



In the circuit on the left, The NMOS transistors have A in series with $\overline{B}$ and this series combination is in parallel with the series combination of $\overline{A}$ and B. The output of the left half is therefore LOW when $A \cdot \overline{B} + \overline{A} \cdot B$ is 'TRUE', thus producing XNOR.
The half circuit on the right changes series to parallel, parallel to series and complements the inputs.

Thus, it has $\overline{A}$ in parallel with B and this parallel combination is in series with the parallel combination of A and $\overline{B}$. This produces the XOR output.

The network on the right can be simplified. Consider the lower shorting line used for paralleling NMOS transistors. No current will ever flow through this wire because it provides a path between an NMOS driven by A to another driven by $\overline{A}$. Similarly, it provides a path between an NMOS driven by $\overline{B}$ to another driven by B. Since no current will flow through this, we may as well remove it. This results in the alternative circuit shown below.



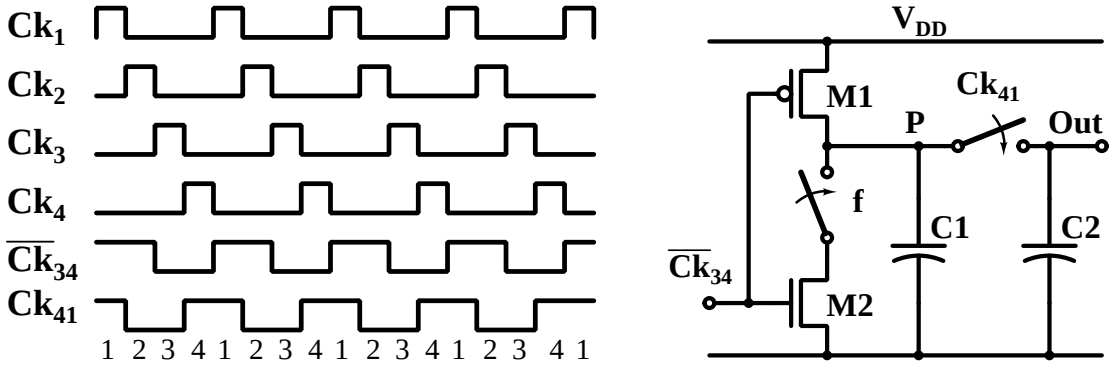
This alternative circuit is not surprising. The right half circuit is seen to implement the series parallel combination corresponding to $A \cdot B + \overline{A} \cdot \overline{B}$, which is just the expression for XNOR. Thus the output of this half circuit is LOW when XNOR is true – that is, the output is XOR, which is just what we wanted.

– **[2]**

– **[Q2: 3 + 2 = 5 marks]**

Q–3 **a)** Show the timing diagram for clocks, internal nodes and output of a 4 phase CMOS dynamic logic gate of type 1. The logic function performed by series/parallel connection of NMOS transistors need not be shown and can be represented by a black box. You should clearly mark the clock phases during which the output is valid.

Soln.: The figure below shows a dynamic 4phase logic gate of type 1.



The signal $\overline{Ck_{34}}$ is LOW during phases 3 and 4 of the clock, while Ck_{41} is HIGH during phases 4 and 1 of the clock..

- During phase 3, the PMOS transistor M1 is ON, the NMOS transistor M2 is OFF and the switch controlled by CK_{41} is OFF. In this phase, capacitor C1 is pre-charged to V_{DD}, while C2 holds its previous value.
- During phase 4, the PMOS transistor M1 remains ON, the NMOS transistor M2 remains OFF and the switch controlled by CK_{41} turns ON. In this phase, capacitors C1 and C2 are pre-charged to V_{DD}.
- During phase 1, the PMOS transistor M1 turns OFF, the NMOS transistor M2 turns ON, while the switch controlled by CK_{41} remains ON. During this phase, capacitors C1 and C2 are conditionally discharged depending on the status of the switch f. Thus the output evaluates $\overline{f}$ during this phase.
- During phase 2, the PMOS transistor M1 remains OFF, the NMOS transistor M2 remains ON, and the switch controlled by CK_{41} turns OFF. The output is valid during this phase, as well as in the next phase.

– **[2]**

b) A circuit module receives external signals ATN (attention) and four address lines A3-A0. It is supposed to respond to incoming data on a data bus D7-D0 if ATN is '1', irrespective of address line values. If ATN is '0', it should respond to the data bits only if the bit pattern on A3-A0 is 0110 (which is its address).

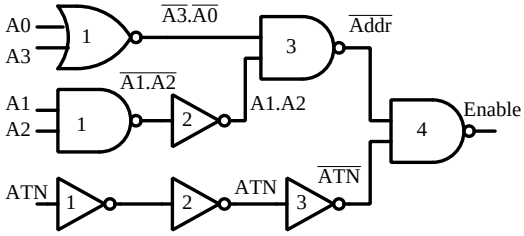
We want to generate an 'Enable' signal which will be 1 only when the circuit module needs to respond to incoming data, using 4 phase CMOS Dynamic logic. Signals ATN and A3-A0 are valid only in phase 1 of the clock. Due to a restriction on series connected transistors, only NAND and NOR gates with a maximum of 3 inputs and inverters can be used.

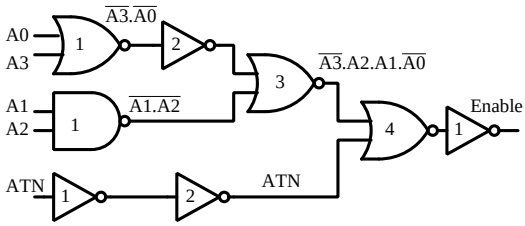
Show a gate level implementation, clearly marking the type of all gates. Specify the clock phases during which the 'Enable' signal is valid. The design should minimize complexity and delay.

Soln.:

$$\text{Enable} = \text{ATN} + \overline{\text{ATN}} \cdot \overline{\text{A3}} \cdot \text{A2} \cdot \text{A1} \cdot \overline{\text{A0}} = \text{ATN} + \overline{\text{A3}} \cdot \text{A2} \cdot \text{A1} \cdot \overline{\text{A0}}$$

This can be generated by either of the circuit shown below:





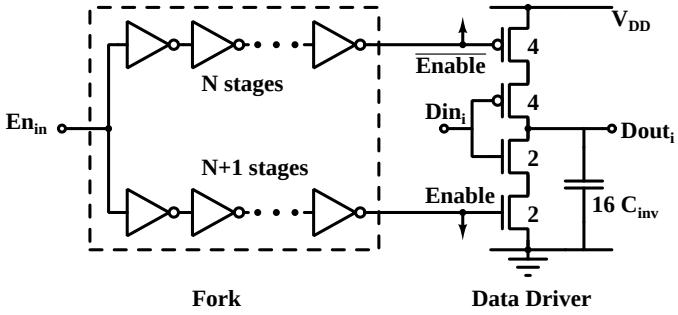
Both circuits use 8 gates. The output of the circuit on the left is available earlier and is valid in phases 1 and 2 of the next clock.

The output of the circuit on the right is valid in phases 2 and 3 of the next clock. If complemented output is acceptable, its output will also be available in phases 1 and 2 of the next clock and it will need one less inverter. – [3]

– **[Q3: 2 + 3 = 5 marks]**

Q–4 The figure below shows a tri-stateable driver which needs Enable and $\overline{\text{Enable}}$ signals generated by a fork. (A fork is a parallel path of N and N+1 inverters with nearly matched total delays).

Each driver sees a load of 16 minimum sized inverters. 8 such drivers are to be driven by a single fork. The input En_{in} can drive a load of 2 minimum sized inverters. Sizes shown for transistors in the data driver are for a minimum sized driver; all of these can be scaled by any factor depending on requirements.



Assume that the mobility correction factor for PMOS transistor widths is 2 and the parasitic delay of inverters (p_{inv}) is 1.

a) Find the optimum stage effort for the whole chain by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0 \qquad \text{iteratively.}$$

Find the number of stages in the logic chain corresponding to this value of ρ. (This number should be adjusted to be an integer just less than the calculated value).

Soln.:

$$p_{inv} + \rho(1 - \ln \rho) = 0 \qquad \text{so } \ln \rho = 1 + \frac{p_{inv}}{\rho}$$

$$\text{Therefore} \quad \rho = \exp\left(1 + \frac{p_{inv}}{\rho}\right) = \exp\left(1 + \frac{1}{\rho}\right)$$

Starting with a trial value of $\rho = 3$, we get successive values of ρ as $3.7937, 3.5381, 3.6061, 3.5869, 3.5923, 3.5908, 3.5912, 3.5911, 3.5911, \cdots$

The logical effort of the data driver is $4/3 = 1.33$
Therefore $G = 1 \times 1 \times \cdots \times 1.33 = 1.33$
$B = 2 \times 1 \times \cdots \times 8 \times 1 = 16$ and $H = 16/2 = 8$.
Therefore $F = GBH = 4/3 \times 16 \times 8 = 170.67$.
Optimum number of stages is therefore $\ln(170.67)/\ln(3.5911) = 4.02$
Therefore the branch with N inverters can have 4 stages, while the branch with N+1 inverters will have 5 stages. Since one of the stages is the data driver itself, the fork will have 3 and 4 stages in the two branches. – [4]

b) Distribute the total effort equally over the logic chain taking the N inverter branch in the fork. Find the transistor widths for all transistors. (The branch with N+1 inverters is not to be designed in this question).

Soln.: Equally distributed effort $= 170.67^{1/4} = 3.6144$.
The scale factor of a stage is given by C_{in}.

$$\text{Since} \quad f = gh = g\frac{C_{out}}{C_{in}}, \qquad C_{in} = g\frac{C_{out}}{f}$$

In our case the value of f is 3.6144 for every stage. The data driver stage should have $C_{in} = 4/3 \times 16/3.6144 = 5.9022$.
Since there are 8 such drivers loading the Enable signal, C_{out} seen by the last inverter is $8 \times 5.9023 = 47.22$
Its C_{in} is therefore $1 \times 47.22/3.6144 = 13.064$. C_{in} of the middle inverter is $1 \times 13.064/3.6144 = 3.6144$,
and C_{in} of the first inverter is $1 \times 3.6144/3.6144 = 1$ as expected.
So the scaling of the 3 inverters is 1, 3.61 and 13.06 respectively. The delay of the 3 inverters is 4.61τ each, leading to a total delay of 12.83τ in the inverter chain. Geometries of transistors are given as:

Stage	Scale Factor	Width Of	
		NMOS	PMOS
Inverter 1	1	1	2
Inverter 2	3.61	3.61	7.22
Inverter 3	13.06	13.06	26.13
Data Driver	5.90	8.85	17.7

In the case of the data driver, only the PMOS is driven. Therefore, just the PMOS presents a capacitive load equivalent to 5.9022 inverters. Therefore, its width should be $3 \times 5.9 = 17.7$ times the minimum transistor size. The NMOS transistors will be half this size and will be driven by the N+1 stage inverter chain. The data driver has two NMOS transistors with $W/L = 8.85$ in series, with corresponding mobility corrected W/L for PMOS transistors. It therefore has the drive strength of $8.85/2 = 4.425$ minimum sized inverters. Since each inverter optimally drives other inverters which are 3.6144 times its own size, this stage can drive $4.425 \times 3.6144 = 16$ inverters, as required. – [3]

– **[Q4: 4 + 3 = 7 marks]**

Paper Ends

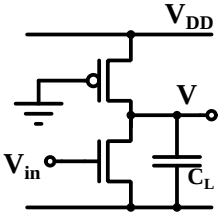
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

End Semester Examination

Tuesday	**EE 671: VLSI Design**	Time: 0930-1230
Nov. 15, 2016	Autumn Semester 2016	Marks: 40

Quantitative answers should be accurate to 1%

Q–1 a) Consider the pseudo-nMOS inverter shown on the right. Derive an expression for the time to charge a load capacitor C_L from 0V to an output voltage V_{oH} when the input is LOW and the nMOS driver transistor is OFF. The expression should be in terms of V_{DD}, C_L and the pMOS transistor parameters.



Use the simple MOS model which ignores channel length modulation. – **[4]**

Soln. 1-a) The charging equation is given by

$$C_L \frac{\mathrm{d}V}{\mathrm{d}t} = I_{dp} \qquad \text{so} \qquad \frac{\mathrm{d}t}{C_L} = \frac{\mathrm{d}V}{I_{dp}}$$

Since the gate of p-MOS is grounded, it is in saturation till the output voltage reaches $|V_{Tp}|$ and in linear region for higher output voltages. In the following, V_{Tp} represents the absolute value of the threshold voltage of p-channel transistors. Integrating from 0V to V_{oH}, we get

$$\frac{\tau_{rise}}{C_L} = \int_0^{V_{Tp}} \frac{\mathrm{d}V}{K_p(V_{DD}-V_{Tp})^2/2} + \int_{V_{Tp}}^{V_{oH}} \frac{\mathrm{d}V}{K_p\left[V_{DD}-V_{Tp})(V_{DD}-V) - \frac{1}{2}(V_{DD}-V)^2\right]}$$

Where $K_p \equiv \mu_p C_{ox}(W_p/L_p)$. Substituting $V_{DD}-V_{Tp} \equiv V_1$ and $V_{DD}-V \equiv V_2$, $\mathrm{d}V = -\mathrm{d}V_2$. We get

$$\frac{\tau_{rise}}{C_L} = \frac{2V_{Tp}}{K_p V_1^2} + \int_{V_{DD}-V_{oH}}^{V_{DD}-V_{Tp}} \frac{\mathrm{d}V_2}{K_p\left(V_1 V_2 - \frac{1}{2}V_2^2\right)}$$

This gives

$$\tau_{rise} = \frac{2C_L V_{Tp}}{K_p V_1^2} + \frac{2C_L}{K_p}\int_{V_{DD}-V_{oH}}^{V_1} \frac{\mathrm{d}V_2}{(2V_1 - V2)V_2}$$

$$\tau_{rise} = \frac{2C_L V_{Tp}}{K_p V_1^2} + \frac{C_L}{K_p V_1}\int_{V_{DD}-V_{oH}}^{V_1} \left(\frac{1}{V_2} + \frac{1}{2V_1 - V_2}\right)\mathrm{d}V_2$$

$$\text{So} \qquad \tau_{rise} = \frac{2C_L V_{Tp}}{K_p V_1^2} + \frac{C_L}{K_p V_1} \ln \frac{V_2}{2V_1 - V_2}\Bigg|_{V_{DD}-V_{oH}}^{V_1}$$

$$\tau_{rise} = \frac{2C_L V_{Tp}}{K_p V_1^2} + \frac{C_L}{K_p V_1} \ln \frac{2V_1 - V_{DD} + V_{oH}}{V_{DD} - V_{oH}}$$

Substituting for V_1, we get

$$\tau_{rise} = \frac{C_L}{K_p(V_{DD} - V_{Tp})}\left(\frac{2V_{Tp}}{V_{DD} - V_{Tp}} + \ln\frac{V_{DD} - 2V_{Tp} + V_{oH}}{V_{DD} - V_{oH}}\right)$$

This is the desired expression for τ_{rise}.

b) In a CMOS process, $C_{ox} = 4.4 \times 10^{-7}\text{F/cm}^2$, $\mu_n = 400\text{cm}^2\text{/Vs}$, $\mu_p = 190\text{cm}^2\text{/Vs}$, $V_{Tp} = -0.65\text{V}$, $V_{Tn} = 0.55\text{V}$. We want to design a pseudo-nMOS inverter using this process, with $V_{DD} = 3.3\text{V}$. What should be the aspect ratio of pMOS transistor (W_p/L_p) such that the inverter charges a load capacitor of 10fF from 0 to 3V in 50ps when the input is LOW (and so the nMOS is OFF). – **[2]**

Soln. 1-b) We have

$$\tau_{rise} = \frac{C_L}{K_p(V_{DD} - V_{Tp})}\left(\frac{2V_{Tp}}{V_{DD} - V_{Tp}} + \ln\frac{V_{DD} - 2V_{Tp} + V_{oH}}{V_{DD} - V_{oH}}\right)$$

Therefore

$$K_p = \frac{C_L}{\tau_{rise}(V_{DD} - V_{Tp})}\left(\frac{2V_{Tp}}{V_{DD} - V_{Tp}} + \ln\frac{V_{DD} - 2V_{Tp} + V_{oH}}{V_{DD} - V_{oH}}\right)$$

Substituting values for various parameters,

$$190 \times 4.4 \times 10^{-7} \times \frac{W_p}{L_p} = \frac{10^{-14}}{5 \times 10^{-11} \times 2.65}\left(\frac{1.3}{2.65} + \ln\frac{3.3 - 1.3 + 3.0}{3.3 - 3.0}\right)$$

$$\text{or} \qquad 8.3610^{-5} \times \frac{W_p}{L_p} = 7.5472 \times 10^{-5}\left(0.49057 + \ln\frac{5.0}{0.3}\right)$$

which gives

$$\frac{W_p}{L_p} = \frac{7.5472}{8.3610} \times (0.49057 + 2.8134) = 2.9827 \approx 3$$

c) What should be the minimum aspect ratio of the n MOS transistor (W_n/L_n) in the above inverter, such that the output voltage is $\leq 0.4V$ when the input voltage is $V_{iH} = 3.0\text{V}$. – **[2]**

Soln. 1-c) When the input voltage is 3.0V and the output voltage is 0.4V, the p-MOS transistor is saturated, whereas the n channel transistor is in linear regime. Equating currents through the two transistors,

$$\frac{1}{2}\mu_p C_{ox}\frac{W_p}{L_p}(3.3 - 0.65)^2 = \mu_n C_{ox}\frac{W_n}{L_n}\left((3 - 0.55) \times 0.4 - 0.4^2/2\right)$$

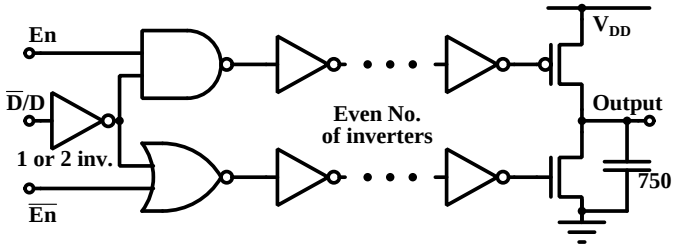
$$\frac{1}{2} \times 190 \times 2.9824 \times 2.65^2 = 400 \times \frac{W_n}{L_n}(2.45 \times 0.4 - .08)$$

which gives

$$\frac{W_n}{L_n} = \frac{95 \times 2.9824 \times 7.0225}{400 \times 0.9} = 5.527$$

– **[Q–1: 4+2+2 = 8 marks]**

Q–2 Consider the driver for a bi-directional pad shown below:



The final load is equivalent to the input capacitance of 750 minimum sized inverters. The number of inverters after the NAND/NOR gates must be even. Depending on whether the *total* number of inverters required to be inserted for optimum delay is odd or even, we put either one or two inverters *before* the NAND/NOR gates to keep the number of inverters *after* the NAND/NOR gates to even. Correspondingly we connect either $\overline{D}$ or D to the input. This input should present a load of 1 minimum inverter to the previous stage.

Assume that the ratio of p-MOS and n-MOS widths for equal rise and fall times is 2 and the parasitic delay of inverters is 2 units. The parasitic delay of the NAND and NOR gates is 4 units, while that of the final driver stage is 2 units. Assume that the NAND and the NOR gates have the same size factor (*i.e.* actual transistor sizes in these gates are the same multiples of those in the minimum sized NAND/NOR gates).

a) How do we compute the optimum stage effort in a multi-stage logic chain when i) the number of stages is fixed and known and ii) the number of stages can be adjusted by inserting inverters. (You do not have to derive the equation for ρ.) – **[2]**

Soln. 2-a) The stage effort is given by $f = gbh = gb(C_{out}/C_{in})$. The logical effort and branching effort is known for all stages. Only the sizing, and therefore the capacitances of various stages are not known. However, the output capacitance of a stage is the same as the input capacitance of the next stage. Therefore, if we take the product of all stage efforts, the unknown quantities cancel out, leaving only the final load capacitance and the input capacitance of the first stage, which are given as specifications. Therefore the product of stage efforts for all stages can be calculated.

$$F = GBH \quad \text{where} \quad G = \prod g_i, B = \prod b_i, \text{ and } H = \frac{C_{out}}{C_{in}}$$

The optimality condition is that the stage effort should be the same for all stages in a logic chain. If the number of stages is known,

$$\hat{f}^N = F = \prod g_i \prod b_i \, \frac{C_{out}}{C_{in}} \qquad \text{Therefore} \qquad \hat{f} = F^{1/N}$$

where $\hat{f}$ represents the optimum stage effort.

Sometimes we can reduce the total delay by adding inverters to the logic chain. This will not change the value of G, B or H, so F remains the same. However, the optimum number of inverters to be added is not known, so N is not known. Therefore, we cannot compute $\hat{f}$ directly by the above formula.

In this case, we can evaluate the best possible stage effort as ρ, which is computed by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0$$

where p_{inv} is the parasitic delay for an inverter. This non-linear equation needs to be solved either graphically or by iteration. Once ρ is known, We find the optimum number of stages using

$$N = \frac{\ln F}{\ln \rho}$$

The value of N needs to be rounded to the nearest integer. Once N is known, we can compute the optimum stage effort as before, using the relation $\hat{f} = F^{1/N}$.

b) Compute the value of ρ by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0$$

for the given value of $p_{inv} = 2$. Use the Newton Raphson technique to solve the equation. Start with a guess value of 4 and iterate till the solution converges. (All intermediate values for rho should be reported.)

– **[2]**

Soln. 2-b) ρ is a solution of

$$f(\rho) = p_{inv} + \rho(1 - \ln \rho) = 0$$

We can solve this by Newton Raphson iterations.

$$f'(\rho) = (1 - \ln \rho) + \rho\left(-\frac{1}{\rho}\right) = -\ln \rho$$

If we start with a guess solution ρ_g, a refined value of the solution is given by

$$\rho_{next} = \rho_g - \frac{f(\rho_g)}{f'(\rho_g)} = \rho_g + \frac{p_{inv} + \rho_g - \rho_g \ln \rho_g}{\ln \rho_g} = \frac{p_{inv} + \rho_g}{\ln \rho_g}$$

The value of p_{inv} is given to be 2. Let us take a guess value of $\rho = 4$. Starting with this guess value, we get successive values of ρ as:

$$4, \quad \frac{2+4}{\ln 4} = 4.3281, \quad \frac{2+4.3281}{\ln 4.3281} = 4.3191, \quad \frac{2+4.3191}{\ln 4.3191} = 4.3191$$

So the solution for ρ converges to 4.3191.

c) In this question, we shall design only the upper branch of the circuit shown above. How many inverters should we use before and after the NAND gate? – **[1]**

Soln. 2-c) logical effort of inverters = 1, of NAND = 4/3, of pMOS in the final stage = 2/3. (Relative sizes of pMOS and nMOS are 2:1 in the inverter and only pMOS needs to be driven).

$$G = 1 \times 4/3 \times 1 \cdots \times 2/3 = 8/9$$

There is only one node where branching occurs. At this point, the input capacitances of NAND and NOR are in the ratio 4:5. So

$$B = \frac{4+5}{4} = 9/4, \qquad H = 750/1 = 750$$

$$\text{Therefore} \qquad F = GBH = \frac{8}{9} \times \frac{9}{4} \times 750 = 1500$$

The optimum number of stages is

$$N = \frac{\ln F}{\ln \rho} = \frac{\ln 1500}{\ln 4.3191} = 4.9986$$

So we should use a total of 5 stages. This includes the NAND and the final driver stage. So 3 inverters should be inserted. We should therefore put one inverter before the NAND and 2 inverters after the NAND gate. (In this case, the input should be driven by $\overline{D}$).

d) Compute the sizes of all transistors in the logic chain of the upper branch (including the inverter(s) preceding the NAND gate) in units of minimum transistor width. Tabulate the results, giving the input capacitance of each stage in inverter units and the widths of nMOS and pMOS transistors in units of minimum transistor width. – **[6]**

Soln. 2-d) We have a total of 5 stages, with $F = 1500$. Therefore,

$$\hat{f} = 1500^{1/5} = 4.31736$$

We begin from the output and work our way backwards to the input. For the final output driver, only the pMOS needs to be driven. Therefore its g value is 2/3 and $C_{out} = 750$.

$$f = gb\frac{C_{out}}{C_{in}} = \frac{2}{3} \times 1 \times \frac{750}{C_{in}} = 4.31736$$

This gives

$$C_{in} = \frac{2 \times 750}{3 \times 4.31736} = 115.81$$

All capacitances are expressed in inverter units. Therefore the capacitance of the pMOS alone is equivalent to 115.81 inverters. Since each inverter has an input capacitance of 3 minimum sized transistors, the width of the pMOS will be $3 * 115.81 \approx 347.5$ times the minimum transistor width. The nMOS in the final driver will have half this width = 173.8 times the minimum transistor width.

For the second inverter after the NAND, the load capacitance is 115.81 inverters.

$$f = gb\frac{C_{out}}{C_{in}} = 1 \times 1 \times \frac{115.81}{C_{in}} = 4.31736$$

This gives

$$C_{in} = \frac{115.81}{4.31736} = 26.825$$

Capacitance of 26.825 inverters is equivalent to $3 \times 26.825 = 80.5$ minimum transistor widths. Of this, the pMOS will have a width of $2 \times 26.825 = 53.65$ and the nMOS will have a width of 26.825 in units of minimum transistor widths.

For the first inverter after the NAND,

$$f = gb\frac{C_{out}}{C_{in}} = 1 \times 1 \times \frac{26.825}{C_{in}} = 4.31736$$

$$\text{This gives} \qquad C_{in} = \frac{26.825}{4.31736} = 6.21$$

Thus, the pMOS will have a width of $2 \times 6.21 = 12.42$ and the nMOS will have a width of 6.21 times the minimum transistor width.

$$\text{For the NAND gate,} \qquad f = gb\frac{C_{out}}{C_{in}} = \frac{4}{3} \times 1 \times \frac{6.21}{C_{in}} = 4.31736$$

$$\text{This gives} \qquad C_{in} = \frac{4 \times 6.21}{3 \times 4.31736} = 1.92$$

Therefore, the nMOS as well as pMOS transistors will have widths $2 \times 1.92 = 3.84$ times the minimum transistor widths.
For the inverter before the NAND, $g = 1, b = 9/4$. Therefore

$$f = gb\frac{C_{out}}{C_{in}} = \frac{9}{4} \times \frac{1.92}{C_{in}} = 4.31736$$

$$\text{This gives} \qquad C_{in} = \frac{9 \times 1.92}{4 \times 4.31736} = 1$$

This agrees with the specification that the input should present a single inverter load. The transistor widths will be 1 unit for the nMOS and 2 units for the pMOS. These results are summarized in the following table:

Component	C_{in} (inverter units)	pMOS widths (Min Trans. units)	nMOS widths (Min. Trans. Units)
Final Driver	115.81	347.5	173.75
Post Inv.- 2	26.825	53.65	26.825
Post Inv.- 1	6.21	12.42	6.21
NAND gate	1.92	3.84	3.84
Pre inverter	1.00	2.00	1.00

(When $En =$ '1', The final stage can be considered to be an inverter with a scale factor of 173.75, driving a load of 750 inverters.) The logical effort and branching effort for this stage is 1. So the effort f for this stage is:

$$f = 1 \times 1 \times \frac{750}{173.75} = 4.317$$

which is equal to the optimum stage effort.)

e) What is the total delay from $\overline{D}/D$ input to the pad in the upper branch in units of τ? – **[1]**

Soln. 2-e) The effort delay of each of the 5 stages is 4.31736 units. The total delay is the sum of all effort delays and parasitic delays. Therefore

$$d = N\hat{f} + \sum p_i = 5 \times 4.31736 + 2 + 4 + 2 + 2 + 2 = 33.6$$

in units of τ.

– **[Q–2: 2+2+1+6+1 = 12 marks]**

Q–3 **a)** Show the wire reduction scheme for an 8×8 Dadda multiplier. Display the scheme using a table formatted as below:

No. Wires	Stage 1			No. Wires	Stage 2			No. Wires	Stage 3			...
	F	H	P		F	H	P		F	H	P	...
...	..	..	..	...	..	..	..	...	..	..	..	...

Where F is the number of full adders, H is the number of half adders and P is the number of passed through wires. The table will have a row for each weight, starting with the least significant bit.

There should be no more than 2 wires for any weight when the reduction is complete. (Notice that for each stage, the incoming wires should be equal to $3F + 2H + P$ and the outgoing wires should equal $F + H + P + F_{lw} + H_{lw}$ where lw represents the lower weight row of the same stage). – **[3]**

Soln. 3-a) The Dadda reduction scheme for an 8×8 multiplier is shown below:

No. Wires	Stage 1			No. Wires	Stage 2			No. Wires	Stage 3			No. Wires	Stage 4			No. Wires
	F	H	P		F	H	P		F	H	P		F	H	P	
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
3	0	0	3	3	0	0	3	3	0	0	3	3	0	1	0	2
4	0	0	4	4	0	0	4	4	0	1	2	3	1	0	0	2
5	0	0	5	5	0	1	3	4	1	0	1	3	1	0	0	2
6	0	0	6	6	1	1	1	4	1	0	1	3	1	0	0	2
7	0	1	5	6	2	0	0	4	1	0	1	3	1	0	0	2
8	1	1	3	6	2	0	0	4	1	0	1	3	1	0	0	2
7	1	1	2	6	2	0	0	4	1	0	1	3	1	0	0	2
6	1	0	3	6	2	0	0	4	1	0	1	3	1	0	0	2
5	0	0	5	6	2	0	0	4	1	0	1	3	1	0	0	2
4	0	0	4	4	1	0	1	4	1	0	1	3	1	0	0	2
3	0	0	3	3	0	0	3	4	1	0	1	3	1	0	0	2
2	0	0	2	2	0	0	2	2	0	0	2	3	1	0	0	2
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	2

b) It is easy to modify the multiplier wire reduction scheme to implement a "Multiply and Accumulate" function. Show the wire reduction scheme for a multiply and accumulate circuit which computes $A \times B + C$ using the Dadda scheme. It should multiply two 8 bit operands and add the product to a 16 bit number. At the end of reduction, there should be no more than two wires at any weight. Use the same tabular format as described above. – **[3]**

Soln. 3-b) The MAC will have one additional wire for all the 16 weights. The maximum number of wires would have been 8 for an 8×8 multiplier, now it will be 9. The capacity of the subsequent stages will be 6, 4, 3 and 2. Thus, no additional stages will be required for the added functionality.

No. Wires	Stage 1			No. Wires	Stage 2			No. Wires	Stage 3			No. Wires	Stage 4			No. Wires
	F	H	P		F	H	P		F	H	P		F	H	P	
2	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
3	0	0	3	3	0	0	3	3	0	0	3	3	0	1	1	2
4	0	0	4	4	0	0	4	4	0	1	2	3	1	0	0	2
5	0	0	5	5	0	1	3	4	1	0	1	3	1	0	0	2
6	0	0	6	6	1	1	1	4	1	0	1	3	1	0	0	2
7	0	1	5	6	2	0	0	4	1	0	1	3	1	0	0	2
8	1	1	3	6	2	0	0	4	1	0	1	3	1	0	0	2
9	2	1	1	6	2	0	0	4	1	0	1	3	1	0	0	2
8	2	1	0	6	2	0	0	4	1	0	1	3	1	0	0	2
7	2	0	1	6	2	0	0	4	1	0	1	3	1	0	0	2
6	1	0	3	6	2	0	0	4	1	0	1	3	1	0	0	2
5	0	0	5	6	2	0	0	4	1	0	1	3	1	0	0	2
4	0	0	4	4	1	0	1	4	1	0	1	3	1	0	0	2
3	0	0	3	3	0	0	3	4	1	0	1	3	1	0	0	2
2	0	0	2	2	0	0	2	2	0	0	2	3	1	0	0	2
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	2

c) Find the number of *additional* reduction stages, half adders and full adders required to convert the multiplier to a Multiply and Accumulate circuit. (The final fast adder with carry propagation is not to be included in this). – **[2]**

Soln. 3-c) The first stage of reduction for the 8×8 multiplier, as well as the MAC, is to 6 wires. Therefore both circuits use the same number of reduction stages.

The 8×8 Dadda multiplier uses 3 full adders in the first stage, 12 full adders in the second stage, 9 full adders in the third stage and 11 full adders in the fourth stage for a total of 35 full adders.
It also uses 3 half adders in the first stage, two in the second stage, and one half adder each in the third and fourth stages for a total of 7 half adders.

The multiply and accumulate reduction scheme uses 8 full adders in the first stage, 16 full adders in the second stage, 11 full adders in the third stage and 13 full adders in the fourth stage. This adds up to 48 full adders.
It also uses 4 half adders in the first stage, 2 half adders in the second stage and one half adder each in 3rd and fourth stages for a total of 8 half adders.

Thus the MAC circuit uses 13 additional full adders and 1 additional half adder without requiring any additional reduction stages.

– **[Q–3: 3 + 3 + 2 = 8 marks]**

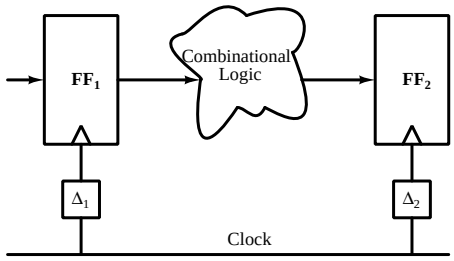
Q–4 a) What do we mean by "skew" and "jitter" in the clock. – **[1]**

Soln. 4-a) Due to routing delays, the clock does not arrive at exactly the same time at all flip flops. The time averaged value of the difference in arrival time for the clock at any two flip flops is the clock skew between these two.
The arrival time of the clock at any point in the clock tree varies with time. This is because the buffer delay depends on the supply voltage to the buffers, which in turn depends on the local current demand through the supply line. This varies with time as the activity of the circuits in the vicinity changes with time. Temperature fluctuations also change the RC delay of the wires. The overall fluctuation in arrival time of the clock at any point is the clock jitter.

b) Show how skew and jitter may impact the performance of a synchronous circuit. – **[1]**

Soln. 4-b)

Consider a stage of a synchronous circuit. Δ_1 and Δ_2 are delays through the clock distribution network. Let us say that the active edge of clock occurs at time 0. This edge will arrive at the first flipflop at time Δ_1. Therefore the data will be available to the combinational logic at $\Delta_1 + CktoQ$, where $CktoQ$ is the delay from the clock edge to the output of the first flipflop. Let the delay through the combinational logic by D_{logic}.



Then the output of the logic is available at the D input of the second flipflop at $\Delta_1 + CktoQ + D_{logic}$. The output data from the combinational logic is to be latched at the next active edge of the clock, which will occur at time T, where T is the clock time period. This edge will arrive at the second flipflop at $T + \Delta_2$. The output of the combinational logic should have settled a setup time t_{su} before the arrival of the clock at the second flipflop. Therefore, we must have

$$\Delta_1 + CktoQ + D_{logic} + t_{su} < T + \Delta_2$$

The worst case for this inequality will occur when Δ_1 is maximum, D_{logic} is maximum and Δ_2 is minimum. This gives,

$$T > (\Delta_{1max} - \Delta_{2min}) + D_{logicmax} + t_{su}$$

Thus for a given logic speed, the fastest clock period is limited by the worst case value of $\Delta_{1max} - \Delta_{2min}$. The time averaged value of this quantity is the skew. Over time, its worst case value could be made higher by jitter, which could be positive for Δ_1 and negative for Δ_2. Thus the ultimate performance of the synchronous system, which is defined by the minimum value of T, is limited by the skew and jitter in the clock distribution network.

c) In some cases skew is intentionally added to the clock of a particular stage in a VLSI circuit. How can it help in improving the performance of the circuit? – **[1]**

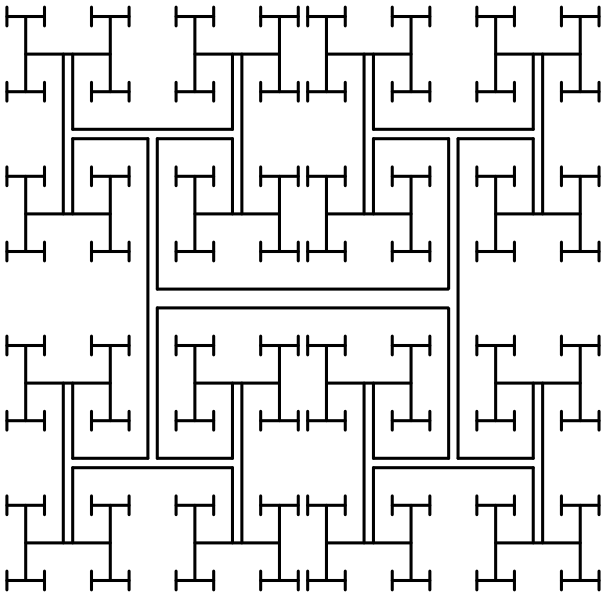
Soln. 4-c) We can see from the expression for T above, that we can use a faster clock (lower value of T) if Δ_2 is high, while Δ_1 is low. However, this cannot be done globally, since the value of Δ_2 for a stage becomes the value of Δ_1 for the next stage.

However, there can be a case when a particular stage needs much more time than other stages. The delay of this stage will then set the minimum limit of the clock period. If the next stage does not take as much time, we can skew the clock to the second stage deliberately to arrive late. Now the clock period can be reduced by the added skew, as this stage, which was limiting the value of the clock period, gets T + added skew for its computation. It must be kept in mind though, that now the next stage will get less time: (T - added skew). Only when this is acceptable, adding skew to a stage will permit us to use a faster clock. (This is known as *"kerning"*.

d) Describe the H-tree and grid distribution networks for the clock. – **[2]**

Soln. 4-d) Since clock skew impacts the performance of a system, we would like to have a clock distribution system which is symmetric and which minimizes skew globally over the whole chip.

One of the ways for doing this is to use an "H tree" for routing the clock over the chip. The diagram below shows the clock distribution scheme.



This is actually a binary tree. Two branches leave on either side from the clock source. At each end of these branches, again two branches leave in a perpendicular direction. This is repeated till the entire chip is covered. Circuits anywhere on the chip tap the clock only from the leaf nodes of this tree. Because of the shape of each branching arrangement, this distribution is called an H tree. Because the total distance from the clock source is the same for each leaf node, the arrival time of the clock edge is nominally the same at each leaf node. Clock needs to be buffered at each leaf node and the load at each leaf node needs to be equalized in order to minimize skew.

The clock signal is at a very high frequency. Therefore it is important to minimize reflections at each node. This is done by impedance matching. The width of each branch is half that of the parent branch, which doubles its characteristic impedance. Since there are two branches in parallel going on either side, this divides the total impedance by two, resulting in impedance matching.

There is a variant of this scheme, in which a buffer is placed at every intermediate node. This is called the buffered H tree.

Another way to distribute the clock is the grid scheme. A grid is made in a level of metallization reserved for clock distribution. All clock inputs to the flipflops over the entire chip are taken from this grid.The grid is fed in parallel by a large number of buffers distributed over the chip. Most buffers are placed around the periphery of the chip. Inputs to the buffers can even be from a distribution scheme like the H tree. Since all buffers are in parallel, this tends to average out variations in their delays. While the grid distribution scheme leads to lower overall skew in the clock distribution, it results in heavy power dissipation. The grid plane over the entire chip presents a huge capacitive load and since the clock is the highest frequency signal on the chip, the CV^2f dynamic power dissipation is quite high.

e) Apart from the clock, why is it important to have low skew on the power-on reset signal distribution on a chip? – **[1]**

Soln. 4-e) Different blocks in a VLSI chip go to their initial state when the power on reset signal is applied. These start working when the reset signal is released. If the reset signal is released at different times for different blocks, they will not be in a consistent state when the reset signal is released.

Some blocks may start working before the blocks which feed them data are operational. These blocks will read invalid data and start operating on that.

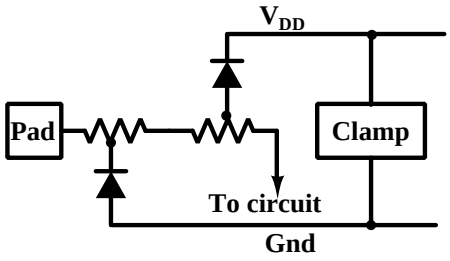
Many blocks expect specific events at certain clock counts. If this count is different for different blocks, data will be misinterpreted.

– **[Q–4: 1 + 1 + 1+ 2 + 1 = 6 marks]**

Q–5 a) Describe the input protection circuit commonly used with pads. What is a clamp circuit and why is it required? What kind of clamp circuits are commonly used? – **[2]**

Soln. 5-a)

The protection circuit commonly used is shown on the right. The diodes steer input voltages higher than V_{DD} to the supply line and negative voltages to the ground line. The resistor and diode pair is often the same structure – a p-type diffused resistor in an n-well also acts as a diode. Similarly, an n-type diffused resistor in a p-well also acts as a diode.

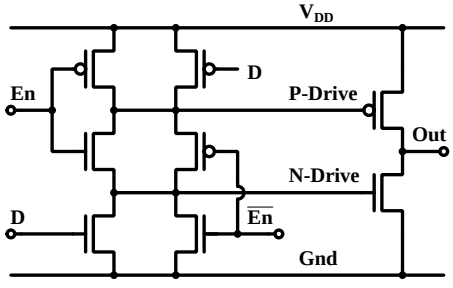


Since the n-wells are connected to V_{DD} and the p-wells to ground, the diodes so formed are connected in proper polarity.

However, dangerous voltages may be applied to inputs when the power supply is not connected. Therefore, a circuit which draws high current when the voltage across it exceeds the supply voltage by some amount is connected between the supply line and ground. When the voltage exceeds its breakdown voltage, it draws heavy current, causing the excess voltage to drop across the series resistors. Thus the maximum voltage across the supply line is limited to the breakdown voltage of this circuit. This circuit is called a clamp circuit.

Different structures can be used as clamp circuits. A latch-up structure without well contacts, a diode connected field transistor whose turn on voltage is higher than the supply voltage and a p^+n^+ zener diode may be employed as the clamp device.

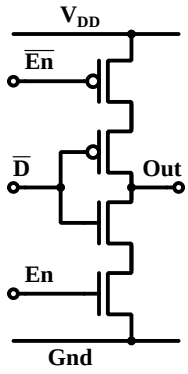
b) Consider the compact NAND-NOR driver for a tri-stateable output shown below.



i) Why is this circuit preferred over the four transistor tri-stateable inverter for driving output pads? – **[1]**

Soln. 5-b i) The given circuit uses a circuit similar to an ordinary inverter for the output driver. The P-Drive lines goes high and the N-Drive line goes low whenever En = '0'. Thus the output is tri-stated. When En = '1', $\overline{D}$ is applied to the output stage, which can then drive the pad and the off-chip loads.

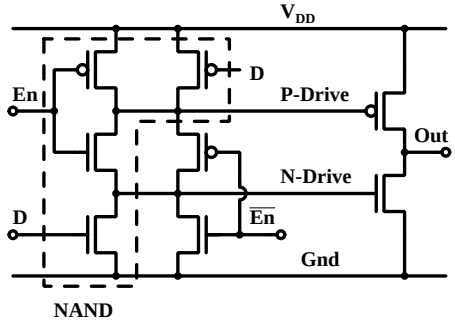
The four transistor tri-state inverter is shown on the right. An output pad driver is generally required to drive heavy off-chip loads. Therefore the driver transistors have to be quite wide. However, in the circuit on the right, the enable transistors are in series with the data driver transistors. Therefore, to get a drive equivalent to an ordinary inverter, transistor widths have to be doubled. Since these are already quite wide, further doubling requires a lot of area and places a heavy capacitive load on the up-stream circuits which drive them.

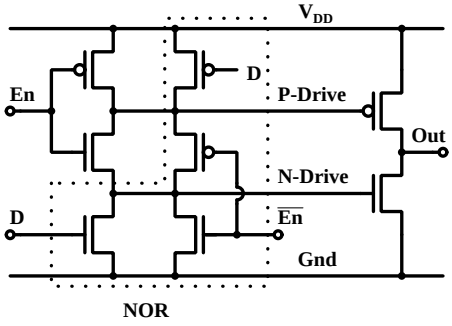


The compact NAND-NOR circuit does not have series transistors in the output stage and thus saves area and presents lower capacitive load to its drivers. This saves power. That is why it is preferred over the 4 transistor tri-stateable driver.

ii) Show the equivalence of this circuit to the discrete NAND and NOR by circling the appropriate sub-circuits in the schematic above. Show that the electrical behaviour of this circuit is identical to the discrete NAND-NOR. – **[1]**

Soln. 5-b ii) The circuit diagrams below show the NAND and NOR functions included in the compact circuit.





When En = '0', The top p-MOS on the left and the bottom n-MOS on the right are ON. The middle n-MOS and p-MOS are OFF. Thus, the P Drive is driven HIGH, while the N Drive is driven LOW, irrespective of the value of D. The output stage is therefore tri-stated. This is what the discrete NAND-NOR circuit would have done.

When En = '1', The top p-MOS on the left and the bottom n-MOS on the right are OFF. The middle n-MOS and p-MOS are ON, forming a pass gate. This connects the right top p-MOS and the bottom left n-MOS driven by D, forming an inverter. This places $\overline{D}$ on the P Drive and N Drive. That is also the same behaviour as the discrete NAND-NOR.

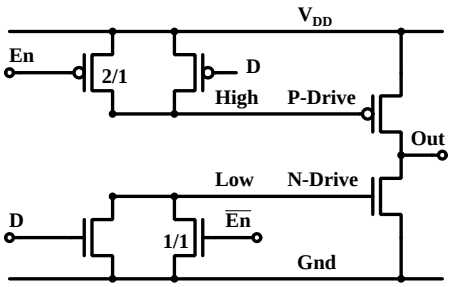
iii) What should be the minimum geometries of the six transistors in the compact

NAND-NOR circuit, so that both P-Drive and N-Drive are driven by at least the drive strength of a minimum inverter. Assume that the pMOS has to be twice as wide as the nMOS to provide equal drive and the ON resistances of pMOS and nMOS transistors forming a pass gate should be equal.

(Hint: Consider the worst case drive requirements of P-Drive and N-Drive for $En =$ '0' and for $En =$ '1' with $D =$ '0' or '1'.) – **[2]**

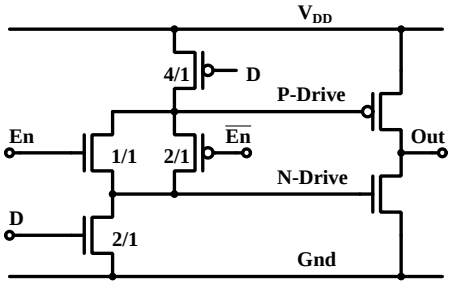
Soln. 5-b iii) Consider the drive requirements when $En =$ '0'. Leaving out the transistors which are OFF, we get the circuit shown on the right below. Irrespective of the value of D, this circuit should be able to drive its outputs with a strength equivalent to an inverter.

Therefore, when $D =$ '1', the top left p-MOS should be equivalent to the p-MOS in an inverter and have a geometry of 2/1. Similarly, when $D =$ '0', the bottom right n-MOS should be equivalent to the n-MOS in an inverter with a geometry of 1/1.



Now consider the case when $En =$ '1'. The equivalent circuit, leaving out the OFF transistors is shown on the right below. When $D =$ '1', the top p-MOS is OFF.

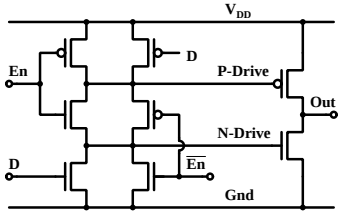
In the worst case, the P Drive line is to be pulled LOW by a series combination of the n-MOS and the pass gate formed by the middle transistors. Therefore the n-MOS should have a geometry of 2/1. Since both transistors of the pass gate are simultaneously ON, a 1/1 n-MOS and a 2/1 p-MOS in parallel will provide the required drive.



When $D =$ '0', the bottom n-MOS is OFF. In the worst case, the N Drive line is to be pulled HIGH by a series combination of the p-MOS and the pass gate. Therefore the top p-MOS should have double the width of the inverter p-MOS, that is 4/1. The pass gate should also have the equivalent of this width. Since both transistors on ON, each transistor can have the same width as in an inverter.

This fixes the relative geometry of all transistors in the circuit. These are tabulated below:

Left column		Right column	
Top p-MOS	2/1	Top p-MOS	4/1
Middle n-MOS	1/1	Middle p-MOS	2/1
Bottom n-MOS	2/1	Bottom n-MOS	1/1



– **[Q–5: 2 + 1 + 1 + 2 = 6 marks]**

Paper Ends

Solution to Mid Semester Examination
EE 671: VLSI Design: Autumn Semester 2016

Q–1 In a given process, the ratio of p channel transistor width to the n channel transistor width in an inverter should be 4 to obtain equal rise and fall times. The parasitic delay of the inverter is 2.3.

a) The ideal stage ratio ρ is a solution to the equation $p_{inv} + \rho(1 - \ln \rho) = 0$. Evaluate the value of ρ using the Newton Raphson iterative technique, starting with a guess value of 4. Values for ρ, $f(\rho)$ and $f'(\rho)$ should be reported for each iteration, till you reach a convergence to 3 decimal places. (The reference section gives a brief description of Newton Raphson method.)

Soln. 1-a)

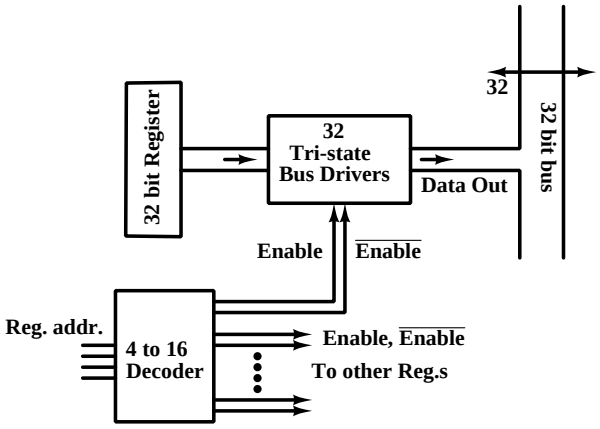
$$\begin{aligned} f(\rho) &= \rho(1 - \ln \rho) + p_{inv} \\ f'(\rho) &= 1 - \ln \rho + \rho(-\frac{1}{\rho}) = -\ln \rho \end{aligned}$$

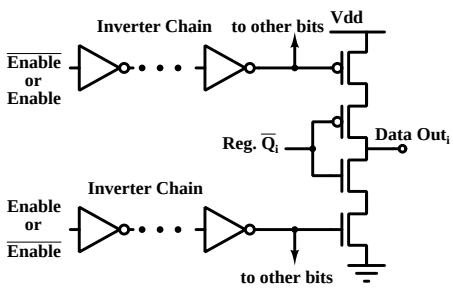
ρ	$f(\rho)$	$f'(\rho)$	next ρ
4.0	0.7548	-1.3863	4.5445
4.5445	-0.0355	-1.5139	4.5211
4.5211	-6.0544×10^{-05}	-1.5087	4.5210
4.5210	-1.7809×10^{-10}	-1.5087	4.5210
4.5210	0	-1.5087	4.5210

– **[3]**

b) The figure on the left below shows a scheme to couple the output of a selected 32 bit register to a bus. Outputs from the decoder (*Enable* / $\overline{Enable}$) drive inverter chains, which in turn drive 32 tri-state output stages (one for each bit). The figure on the right shows inverter chains driving output stages. (Only one output stage is shown, the remaining 31 are indicated by the arrow).

Each tri-state output stage drives a line of the 32 bit bus. The capacitive load presented by the bus line is equivalent to the input capacitance of 40 minimum inverters.





Find the number of inverters we should insert between the p channel transistor

gate and the decoder output to minimize the total delay. (Notice that we have the option of selecting either the *Enable* or the $\overline{Enable}$ output of the decoder depending on whether an even or an odd number of inverters are required).
Assume that *Enable*/$\overline{Enable}$ outputs of the decoder can drive two minimum inverters each.

Soln. 1-b) The tri-state output stage has two n channel transistors in series. Their widths should be 2 each (in units of minimum sized transistors). Two p channel transistors are also in series. Since γ is 4, the p channel transistors should have widths of 8 each. An inverter will have its n channel transistor with width of 1 and the p channel transistor with a width of 4, for a total capacitive load of 5 units. Thus the logical effort for driving the p channel transistor of the output stage is 8/5, whereas for driving the n channel transistor, it is 2/5.
Consider the path from the decoder output to the bus output through the p channel transistor. To compute the path effort, $G = 1 \times 1 \times \cdots 8/5 = 1.6$
$H = 40/2 = 20$
$B = 32$
Therefore the path effort is $F = GBH = 1.6 \times 32 \times 20 = 1024$
The optimum number of stages is
$N = \ln(F)/\ln(\rho) = \ln(1024)/\ln(4.5210) = 4.5942$,
So we require 5 stages in the path. One of these is the output stage itself. Therefore we need to insert 4 inverters. This will be non-inverting, so we should choose the $\overline{Enable}$ output of the decoder to drive this path. – **[3]**

c) Find the scale factor for each of the inverters inserted in the inverter chain. Compute the delay for the path from decoder output to the bus wire in units of τ.

Soln. 1-c) We have total path effort $F = 1028$ and 5 stages. Therefore each stage will have an effort $\hat{f} = 1028^{1/5} = 4$. Let us start with the output stage. Its stage effort must be 4. Therefore,

$$gbh = 8/5 \times 1 \times 40/C_{in} = 4, \text{ so } C_{in} = \frac{8}{5} \times 40/4 = 16$$

Thus this stage should have an input capacitance of 16 inverters. Therefore its scale factor is $16/1.6 = 10$.

We next take up the last of the four inverters, Inv_4. For this stage, $g = 1, b = 32, h = 16/C_{in}$. So we must have

$$32 \times 16/C_{in} = 4, \text{ which gives } \quad C_{in} = 32 \times 16/4 = 128$$

.

The third inverter has $g = 1, b = 1$ and it has to drive a load of 128 inverters. Since the stage effort should be 4, its input capacitance is given by

$$128/C_{in} = 4 \text{ so } \quad C_{in} = 32$$

The second inverter also has $g = 1, b = 1$ and drives a load of 32 inverters. So its input capacitance is $32/4 = 8$.

The first inverter drives a load of 8 inverters and has $g = 1, b = 1$. Therefore its input capacitance is $8/4 = 2$. This agrees with the driving capability of $\overline{Enable}$ output of the decoder.

So the scale factors of the various stages are:

Stage →	1	2	3	4	5
Type	Inv.	Inv.	Inv.	Inv.	Tri-stage
Scale	2	8	32	128	10

The toal delay of this path is $5 \times 4+$ parasitic delay.
The parasitic delay of each inverter is 2.3. The unit tri-state driver has an n channel transistor of geometry 2 and a p channel transistor of geometry 8 connected to the output. This is twice the parasitic capacitance of a minimum inverter (which has $1 + 4 =$) 5. So we can estimate the parasitic delay of the tri-state driver to be twice that of the inverter. Hence we can take the parasitic delay of the tri-state driver to be 4.6. Then the total delay is $20 + 4 \times 2.3 + 4.6 = 33.8$. – [4]

d) How many inverters do we need to insert to drive the n channel transistor gates? Depending on which of *Enable* or $\overline{Enable}$ was chosen for the part above, the other output of the decoder should be used for this chain. Adjust the number of inverters for this and compute the total delay for this path.
Compute the scale factors for all inverters in this chain.
There is no need to equalize delays through the two chains.

Soln. 1-d) To compute the total path effort, the logical effort of the tri-state buffer is to be computed for the n channel transistors. The relative size of the n channel transistor is 2, while the total capacitive load presented by a minimum inverter is $(1+4 = 5)$. Therefore, the logical effort for the n channel transistors is $2/5$. Then, we have $G = 1 \times 1 \times \cdots \times 2/5 = 0.4$. As before, $B = 32$ and $H = 40/2 = 20$. So the path effort is $F = GBH = 0.4 \times 32 \times 20 = 256$

Since the equality of rise time and fall time must be maintained, the scale of the tri-state driver is already fixed by the design of p channel drivers. So the capacitance presented by the tri-state driver to the inverter chain is $10 \times 2 = 20$ minimum transistor widths. In units of inverter capacitances, this is equivalent to 4 inverters. So the stage effort for the tri-state driver is $0.4 \times 1 \times 40/4 = 4$. The path effort from the input to the gate of the n channel transistor in the tri-state buffer is $256/4 = 64$. (This could be calculated independently. The total load on the final inverter inclusive of branching is $32 \times 4 = 128$. Since the input capacitance is 2, $BH = 128/2 = 64$. The logical effort of all inverters is 1, so the path effort is $F = GBH = 64$, excluding the final tri-state buffer.)

The number of inverters to be inserted is given by $\ln 64 / \ln 4.521 = 2.7565$. So we would like to have 3 inverters. However, that would mean that the inverter chain will be inverting and we must use $\overline{Enable}$ as the input. This option is not available to us. We have to use *Enable* as the input and therefore, must have an even number of inverters in the driver chain. So we choose 4 inverters in the chain. (A choice

of 2 inverters is actually closer to 2.76, so we should also consider using 2 inverters).

The optimum stage ratio considering only the 4 inverters is $\hat{f} = 64^{1/4} = 2.8284$. Since the total load on the fourth inverter is $32 \times 4 = 128$ and g and b values for all other stages are 1, we get the scale factors for all stages by dividing 128 repeatedly by $\hat{f}$. So we get the scale factors as 45.26, 16, 5.66 and 2 respectively. The total delay is then

$$4\hat{f} + 4 + 4p_{inv} + p_{tristate} = 11.3136 + 4 + 9.2 + 4.6 = 29.11\tau$$

Additional discussion

Notice that the parasitic delay accounts for about half the buffer delay, inspite of the heavy loading in this problem. It is interesting to check what would happen if chose two stages of inverters instead of four.

In that case, $\hat{f} = 64^{1/2} = 8$ and the scale factors will be 2 and 16 for the two inverters. The total delay in this case will be

$$2\hat{f} + 2p_{inv} + f_{tristate} + p_{tristate} = 16 + 4.6 + 4 + 4.6 = 29.2\tau$$

This is practically the same as the four inverter case. This happens because 2 and 4 are about equidistant from the optimal choice of 3 inverters.

What if we did not respect the p to n ratio in the tri-state buffer?
In that case, the path effort of 256 can be spread equally over all 5 stages. So $\hat{f} = 256^{1/5} = 3.031$. The stage effort for the tri-state buffer is

$$\hat{f} = 3.031 = gbh = 0.4 \times 1 \times 40/C_{in} = 16/C_{in}$$

This gives $C_{in} = 16/3.031 = 5.28$ An input capacitance of 5.28 inverters corresponds to $5.28 \times 5 = 26.4$ minimum transistor widths. So the scale factor is 13.2.

The load on the fourth inverter inclusive of branching is $32 \times 5.28 = 168.9$. Therefore the input capacitance of the fourth inverter is $168.9/3.031 = 55.72$.

Successively dividing by 3.031, we get the scale factors of $55.72/3.03 = 18.38$ for the third, and $18.38/3.03 = 6.062$ for the second, and finally, $6.062/3.031 = 2$ for the first inverter.

The total delay is

$$5 \times 3.031 + 4 \times 2.3 + 4.6 = 28.96$$

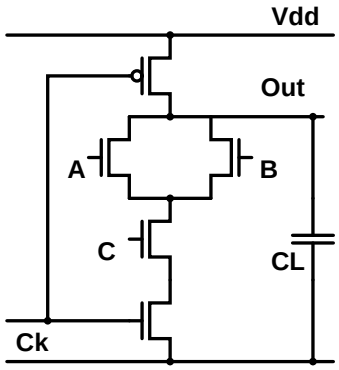
This is not very different from the designs which do maintain equal rise and fall times. So those must be preferred over this design. – [4]

– **[Q1: 3 + 3 + 4 + 4 = 14 marks]**

Q–2 Why does a CMOS dynamic logic gate malfunction if we do not use multiple clocks? How is this problem solved using 4 phase dynamic logic? What is the restriction for driving different types of gates in 4 phase logic?

How does Zipper logic manage to solve this problem without needing multiples phases of the clock?

Soln. 2) Let us consider a basic CMOS dynamic gate inplementing $\overline{(A+B).C}$.

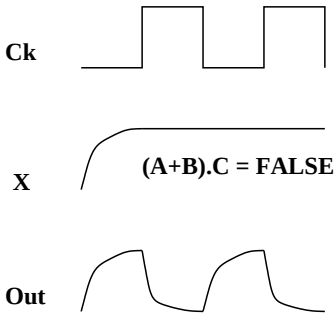


When the clock is low, the pMOS transistor is on and the bottom nMOS is off. The output is 'pre-charged' to '1' unconditionally.

When the clock goes high, the pMOS turns off and the bottom nMOS comes on.
The circuit then conditionally discharges the output node, if (A+B).C is TRUE.

This implements the function $\overline{(A+B).C}$.

However this circuit can malfunction when several CMOS dynamic gates are cascaded. Consider the case when the above circuit is followed by a dynamic inverter.

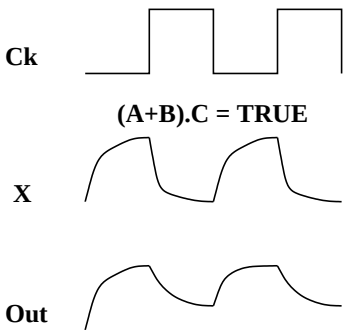


When (A+B).C is FALSE, There is no problem.

X pre-charges to '1' and remains at '1'..

Therefore the inverter sees the correct logic value at its input all the time and discharges the output to '0', which is the expected output.

However, When (A+B).C is TRUE, there is a problem.



The correct value for X is now '0'. After the pre-charge cycle, X takes some time to discharge to '0'. So for some time after pre-charge, its output is held at the wrong value of '1'.

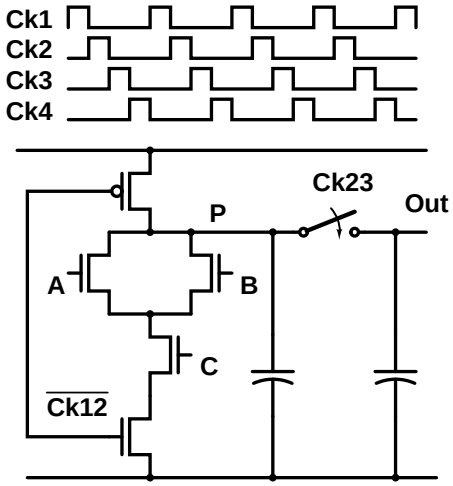
During this time, the nMOS of the inverter is 'on' and charge placed on the output leaks away. This can take the output to the wrong value of '0'. There is no way for the output to go to '1' when X reaches its correct value of '0'.

Thus the simple dynamic CMOS circuit can malfunction on cascading for certain data conditions.

This problem can be soved by using a clock with multiple phases, so that the transiently wrong output of one stage is not fed to the other. We use different phases for pre-charge,

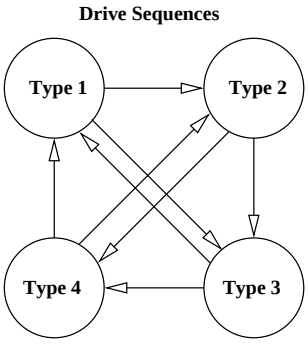
evaluation and for holding a valid output. Now we can sample the previous stage only in the clock phase when evaluation is complete and it is valid.

For the circuit shown below, we use a 4 phase clock. Combined clock signals of the type Ck_{mn} are generated as required. Ck_{mn} is high during the m and n phases of the clock.



1. In phase 1, node P is pre-charged.
2. In phase 2, P as well as the output are pre-charged.
3. In phase 3, The gate evaluates.
4. In phases 4 and 1, the output is isolated from the driver and remains valid.

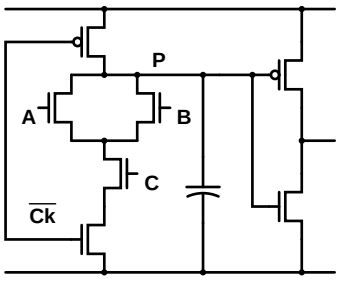
This is called a type 3 gate. It evaluates in phase 3 and is valid in phases 4 and 1. Similarly, we can have type 4, type 1 and type 2 gates.



Since the output of type 3 gate is valid in phase 4 and phase 1, it can drive gates of type 4 and type 1.
Similarly, a type 1 gate can feed gates of types 2 and 3; type 2 gates can drives types 3 and 4; and type 4 gates can drive types 1 and 2.

This ensures that gates receive inputs only when these are valid.

The CMOS dynamic logic gate malfunctions because the output remains at the pre-charge value of '1' for some time, before acquiring it valid value of '0'. We can also solve this problem by putting a static inverter after the CMOS dynamic gate. When the logic gate pre-charges to '1', the output of the inverter goes to '0'.

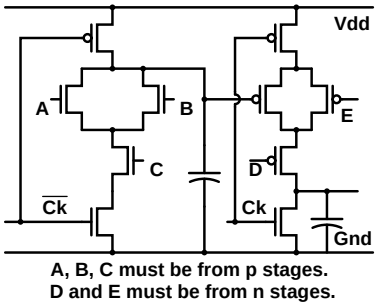


The output of the inverter goes to '0' on pre-charge and may remain at the wrong value of '0' for some time before acquiring its right value of '1'. However this wrong value does not turn on the nMOS input of the next stage and therefore does not lead to a malfunction.

This circuit is non-inverting because of the addition of an inverter. So all logic functions cannot be implemented using it.

We can implement arbitrary logic functions using zipper logic. Just as a 'transiently wrong' value of '0' does not cause a problem for nMOS inputs, a 'transiently wrong'

value of '1' is safe for pMOS inputs. Therefore we can alternate n and p type CMOS dynamic logic without using an inverter.



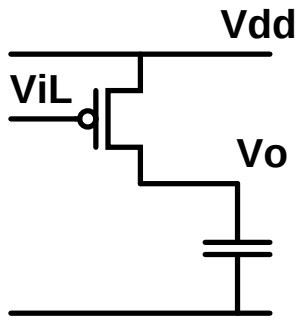
The n stage is pre-charged high, but it drives a p stage. A high pre-charged stage will keep the p evaluation stage off, which will not cause any malfunction. The p stage will be pre-discharged to 'low', which is safe for driving n stages.

– **[7 marks]**

Q–3 Consider a CMOS inverter in which n and p channel transistors have been sized to give equal rise and fall times. Derive an expression in terms of p channel transistor parameters for charging the output capacitance C_L from 0V to 3V with $V_{DD} = 3.3$V. Assume the input voltage to be 0.5V, which is below V_{Tn}. You can assume perfect saturation.

If the value of $K_p \equiv \mu C_{ox} W/L$ is 100μA/V^2, $V_{DD} = 3.3$V, $V_{Tp} = 0.7$V and the load capacitance is 0.1 pF, find the charge time from 0 to 3V. Find the value of the equivalent resistor which will charge the load capacitance in the same amount of time from 0 to 3V.

Soln. 3) Because the input voltage is $< V_{Tn}$, the nMOS is off and need not be considered.



$$I_{dp} = C_L \frac{dV_o}{dt}$$

so,

$$\frac{dt}{C_L} = \frac{dV_o}{I_{dp}}$$

Integrating both sides, we get

$$\frac{\tau_{rise}}{C_L} = \int_0^3 \frac{dV_o}{I_{dp}}$$

The pMOS is saturated till the output reaches $0.5+V_{Tp}$. Between $0.5+V_{Tp}$ to 3.0 V, it is in linear regime. The magnitude of V_{GS} for the p channel transistor is $3.3 - 0.5 = 2.8$V.

$$\begin{aligned}\frac{\tau_{rise}}{C_L} &= \int_0^{0.5+V_{Tp}} \frac{dV_o}{\frac{K_p}{2}(2.8 - V_{Tp})^2} \\ &\quad + \int_{0.5+V_{Tp}}^{3} \frac{dV_o}{K_p\left[(2.8 - V_{Tp})(3.3 - V_o) - \frac{1}{2}(3.3 - V_o)^2\right]}\end{aligned}$$

We define $V_x \equiv 3.3 - V_o$. Then $\mathrm{d}V_o = -\mathrm{d}V_x$. As V_o goes from $0.5 + V_{Tp}$ to 3.0V, V_x will go from $2.8 - V_{Tp}$ to 0.3V.

$$\begin{aligned}\frac{K_p \tau_{rise}}{2C_L} &= \frac{0.5 + V_{Tp}}{(2.8 - V_{Tp})^2} - \int_{2.8-V_{Tp}}^{0.3} \frac{\mathrm{d}V_x}{2V_x(2.8 - V_{Tp}) - V_x^2} \\ &= \frac{0.5 + V_{Tp}}{(2.8 - V_{Tp})^2} + \int_{0.3}^{2.8-V_{Tp}} \frac{\mathrm{d}V_x}{V_x\,(5.6 - 2V_{Tp} - V_x)}\end{aligned}$$

The integration can be carried out using partial fractions.

$$\int_{0.3}^{2.8-V_{Tp}} \frac{\mathrm{d}V_x}{V_x\,(5.6-2V_{Tp}-V_x)} = \frac{1}{5.6-2V_{Tp}} \int_{0.3}^{2.8-V_{Tp}} \left(\frac{1}{5.6-2V_{Tp}-V_x}+\frac{1}{V_x}\right)\mathrm{d}V_o$$

$$= \frac{1}{5.6-2V_{Tp}} \left[\ln \frac{V_x}{5.6-2V_{Tp}-V_x}\right]_{0.3}^{2.8-V_{Tp}}$$

$$= \frac{1}{5.6-2V_{Tp}} \ln \frac{5.3-2V_{Tp}}{0.3}$$

So

$$\frac{K_p \tau_{rise}}{2C_L} = \frac{0.5+V_{Tp}}{(2.8-V_{Tp})^2} + \frac{1}{5.6-2V_{Tp}} \ln \frac{5.3-2V_{Tp}}{0.3}$$

Thus

$$\tau_{rise} = \frac{2C_L}{K_p}\left(\frac{0.5+V_{Tp}}{(2.8-V_{Tp})^2} + \frac{1}{5.6-2V_{Tp}} \ln \frac{5.3-2V_{Tp}}{0.3}\right)$$

This is the expression for τ_{rise} in terms of device parameters.

If the value of $K_p \equiv \mu C_{ox} W/L$ is given to be $100\mu\mathrm{A/V}^2$, $V_{DD} = 3.3\mathrm{V}$, $V_{Tp} = 0.7\mathrm{V}$ and the load capacitance = 0.1 pF, The above expression evaluates to

$$\frac{2\times 10^{-13}}{10^{-4}}\left(\frac{1.2}{2.1^2}+\frac{1}{4.2}\ln\frac{3.9}{0.3}\right) = 1.6861\mathrm{ns}$$

If a resistor charges the same capacitor from 0V to 3V in the same time, we should have

$$3 = 3.3(1-e^{-\tau_{rise}/RC_L}) \quad \text{So} \quad 1-e^{-\tau_{rise}/RC_L} = \frac{3}{3.3}$$

Which gives

$$e^{-\tau_{rise}/RC_L} = 1-\frac{3}{3.3} = \frac{.3}{3.3} = \frac{1}{11} \qquad \text{So} \quad e^{\tau_{rise}/RC_L} = 11$$

Thus,

$$\frac{\tau_{rise}}{RC_L} = \ln(11) \quad \text{and so,} \quad R = \frac{\tau_{rise}}{\ln(11)\times C_L}$$

Since $\tau_{rise} = 1.6861\mathrm{ns}$ and $C_L = 0.1\mathrm{pF}$, we can evaluate R as

$$R = \frac{1.6861\times 10^{-9}}{2.3979\times 10^{-13}} = 7.0399\times 10^3$$

So the equivalent resistor is 7.04KΩ. – **[6 marks]**

Q–4 What is the value of Metal to Semiconductor work function ϕ_{MS} if we are using Aluminum as the metal with p type silicon doped to $10^{17}/\mathrm{cm}^3$ as the substrate? The Fermi level of silicon is 50 meV above the conduction band of silicon. The band gap of silicon is 1.1 eV and the intrinsic carrier density in silicon is $1.5\times 10^{10}/\mathrm{cm}^3$. Take the value of KT/q to be 26 meV.

Soln. 4) The Fermi level in silicon will be $(KT/q)\ln(N_A/n_i)$ below the intrinsic level. Thus it is $.026\times \ln\frac{10^{17}}{1.5\times 10^{10}} = 0.4085$ eV below mid gap.

The Fermi level of Aluminum is 50 meV above the conduction band,
so it is $0.550+0.050 = 0.6$ eV above the midgap.

The work function difference is $-06-0.4085 = -1.0085\mathrm{V}$, $\approx -1.01\mathrm{V}$. – **[3 marks]**

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

EE 671: VLSI Design

Tuesday	Class Test 1	Time: 1130-1300
09-08-16	Autumn Semester 2016	Marks: 10

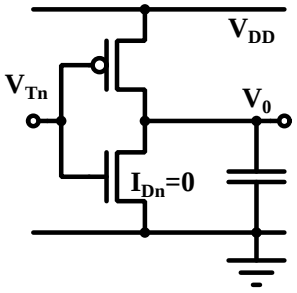
Q–1 What should be the ratio of widths of n and p channel transistors of a CMOS inverter, such that the time taken to charge the output from 0V to $V_{DD} - V_{Tp}$ with the input voltage $= V_{Tn}$ is the same as the time taken to discharge the output from V_{DD} to V_{Tn} with the input voltage $= V_{DD} - V_{Tp}$.

You are given that $V_{DD} = 3.0$V, $V_{Tn} = 0.5$V, $V_{Tp} = 0.7$V, $\mu_n = 450\text{cm}^2/\text{Vs}$, and $\mu_p = 250\text{cm}^2/\text{Vs}$. The n and p channel transistors have the same channel length and gate oxide capacitance per unit area.

(Expressions for charge and discharge times should be derived and *not* quoted from memory).

Soln.:

During the charge cycle, the input voltage is V_{Tn}, so the n channel transistor draws 0 current. As the output charges from 0 to $V_{Tn} + V_{Tp}$, the p channel transistor is in saturation. When the output goes above this voltage, the p channel transistor enters the linear regime.



Because the load capacitor is being charged by the drain current of the p channel transistor,

$$I_{dp} = C\frac{dV_o}{dt}$$

so,

$$\frac{dt}{C} = \frac{dV_o}{I_{dp}}$$

This separates the variables, with the LHS independent of operating voltages and the RHS independent of time. Integrating both sides, we get

$$\frac{\tau_{rise}}{C} = \int_0^{V_{DD}-V_{Tp}} \frac{dV_o}{I_{dp}}$$

The integration range on the right can be broken into saturation and linear regimes. Thus,

$$\begin{aligned}\frac{\tau_{rise}}{C} &= \int_0^{V_{Tn}+V_{Tp}} \frac{dV_o}{\frac{K_p}{2}(V_{DD} - V_{Tn} - V_{Tp})^2} \\ &\quad + \int_{V_{Tn}+V_{Tp}}^{V_{DD}-V_{Tp}} \frac{dV_o}{K_p\left[(V_{DD} - V_{Tn} - V_{Tp})(V_{DD} - V_o) - \frac{1}{2}(V_{DD} - V_o)^2\right]}\end{aligned}$$

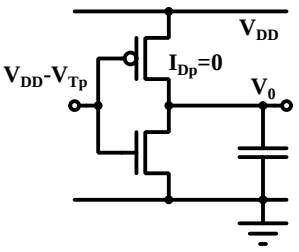
We define $V_1 \equiv V_{DD} - V_o$ and $V_2 \equiv V_{DD} - V_{Tn} - V_{Tp}$, so $dV_o = -dV_1$.
We get

$$\begin{aligned}
\frac{K_p \tau_{rise}}{2C} &= \frac{V_{Tn} + V_{Tp}}{V_2^2} - \int_{V_2}^{V_{Tp}} \frac{dV_1}{2V_1 V_2 - V_1^2} \\
&= \frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \int_{V_{Tp}}^{V_2} \left[\frac{1}{2V_2 - V_1} + \frac{1}{V_1} \right] dV_1 \\
&= \frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \left[\ln \frac{V_1}{2V_2 - V_1} \right]_{V_{Tp}}^{V_2} \\
&= \frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \ln \frac{2V_2 - V_{Tp}}{V_{Tp}}
\end{aligned}$$

Thus we can express τ_{rise} as

$$\begin{aligned}
\tau_{rise} &= \frac{2C}{K_p} \left[\frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \ln \frac{2V_2 - V_{Tp}}{V_{Tp}} \right] \\
\text{where} \quad V_2 &\equiv V_{DD} - V_{Tn} - V_{Tp}
\end{aligned}$$

During the discharge cycle, the input voltage is $V_{DD} - V_{Tp}$. Thus the magnitude of the gate voltage for the p channel transistor is V_{Tp}. So the p channel transistor draws 0 current. The n channel transistor is in saturation while the output voltage falls from V_{DD} to $V_{DD} - V_{Tp} - V_{Tn}$. Below this output voltage, the n channel transistor enters the linear regime.



$$I_{dn} = -C \frac{dV_o}{dt}$$

Separating voltages and integrating from the initial voltage (V_{DD}) to the final voltage (V_{Tn}), we get

$$\frac{\tau_{fall}}{C} = -\int_{V_{DD}}^{V_{Tn}} \frac{dV_o}{I_{dn}}$$

As the voltage falls from V_{DD} to $V_{DD} - V_{Tn} - V_{Tp} \equiv V_2$, the transistor is in saturation and

$$I_{dn} = \frac{K_n}{2} (V_{DD} - V_{Tp} - V_{Tn})^2 = \frac{K_n}{2} V_2^2$$

As the voltage falls below V_2 it enters the linear regime and

$$I_{dn} = K_n \left((V_{DD} - V_{Tp} - V_{Tn}) V_o - \frac{1}{2} V_o^2 \right) = K_n \left(V_2 V_o - \frac{1}{2} V_o^2 \right)$$

So,

$$\begin{aligned}
\frac{\tau_{fall}}{C} &= -\int_{V_{DD}}^{V_2} \frac{dV_o}{\frac{K_n}{2} V_2^2} - \int_{V_2}^{V_{Tn}} \frac{dV_o}{K_n \left(V_2 V_o - \frac{1}{2} V_o^2 \right)} \\
\text{So} \quad \frac{K_n \tau_{fall}}{2C} &= \int_{V_2}^{V_{DD}} \frac{dV_o}{V_2^2} + \int_{V_{Tn}}^{V_2} \frac{dV_o}{2V_2 V_o - V_o^2} \\
&= \frac{V_{DD} - V_2}{V_2^2} + \frac{1}{2V_2} \int_{V_{Tn}}^{V_2} \left[\frac{1}{2V_2 - V_o} + \frac{1}{V_o} \right] dV_o
\end{aligned}$$

Which gives

$$\begin{aligned} \frac{K_n \tau_{fall}}{2C} &= \frac{V_{DD} - V_2}{V_2^2} + \frac{1}{2V_2}\left[\ln \frac{V_o}{2V_2 - V_o}\right]_{V_{Tn}}^{V_2} \\ &= \frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \ln \frac{2V_2 - V_{Tn}}{V_{Tn}} \\ \text{So} \quad \tau_{fall} &= \frac{2C}{K_n}\left[\frac{V_{Tn} + V_{Tp}}{V_2^2} + \frac{1}{2V_2} \ln \frac{2V_2 - V_{Tn}}{V_{Tn}}\right] \end{aligned}$$

In our case, $V_{DD} = 3\text{V}$, $V_{Tn} = 0.5\text{V}$, $VTp = 0.8\text{V}$.
So $V_2 \equiv V_{DD} - V_{Tn} - V_{Tp} = 3 - 0.5 - 0.7 = 1.8\text{V}$.
Equating τ_{rise} and τ_{fall}, we get

$$\begin{aligned} \frac{2C}{K_p}\left[\frac{1.2}{1.8^2} + \frac{1}{3.6} \ln \frac{3.6 - 0.7}{0.7}\right] &= \frac{2C}{K_n}\left[\frac{1.2}{1.8^2} + \frac{1}{3.6} \ln \frac{3.6 - 0.5}{0.5}\right] \\ \text{So} \quad \frac{2C}{K_p}(0.3704 + 0.3948) &= \frac{2C}{K_n}(0.3704 + 0.5068) \end{aligned}$$

This leads to

$$\frac{K_p}{K_n} = \frac{0.3704 + 0.3948}{0.3704 + 0.5068} = 0.8723$$

Therefore

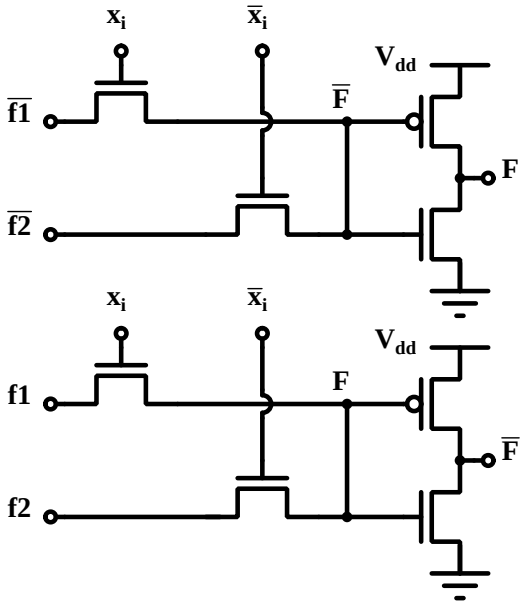
$$\frac{\mu_p C_{ox} W_p / L}{\mu_n C_{ox} W_n / L} = 0.8723$$

So

$$\frac{W_p}{W_n} = \frac{\mu_n}{\mu_p} \times 0.8723 = \frac{450}{250} \times 0.8723 = 1.57$$

So the p channel transistor needs to be about 1.57 times the width of the n channel transistor. – **[7]**

Q–2 Consider the basic CPL gate shown below.



a) Describe the operation of this gate. Why are the inverters required?

Soln.: This gate is based on multiplexer logic. It requires each input in true as well as complement form and provides the output in true as well as complement form. The NMOS pass transistors are controlled by true and complemented forms of one of the inputs. Therefore, the output of the pair of pass transistors is one of the two inputs presented to the multiplexer, depending on the value of the input controlling the pass gates.

Given a Boolean function F(x0, x2, ..., xn), we can express it as:

$$F(x1, x2, \ldots, xn) = xi \cdot f1 + \overline{xi} \cdot f2$$

where f1 and f2 are reduced expressions for F with xi forced to '1' and '0' respectively. Thus, F can be implemented with a multiplexer controlled by xi which selects f1 or f2 depending on xi. f1 and f2 can themselves be decomposed into simpler expressions by the same technique.

As an example, consider the xor of two inputs A and B. Then, $F = A \cdot \overline{B} + \overline{A} \cdot B$. Taking A as the pivot which controls the multiplexers, we have $F = B$ when $A = 0$ and $F = \overline{B}$ when $A = 1$. Therefore we can present B and $\overline{B}$ as inputs to the multiplexer controlled by A to realize the function.

An NMOS pass gate has negative noise margin for 'high' inputs. This is because the pass gate turns off when the output reaches a voltage which is V_{Tn} below the gate voltage. To restore the noise margin, conventional CMOS gates are put after the multiplexers. We use the simplest CMOS gate– an inverter, for this purpose. Thus two multiplexers, each with an inverter, are required to generate true as well as complemented outputs.

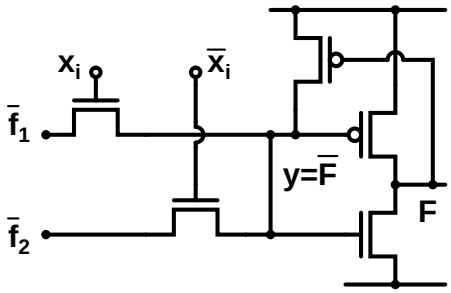
In our example, the multiplexer which produces xor will now be followed by an inverter and so, will give xnor as the output. The multiplexer which produces xnor ($= A \cdot B + \overline{A} \cdot \overline{B}$) will give xor as the eventual output after inversion.

b) Why does this configuration lead to leakage current in the inverters?

Soln.: The output of the NMOS multiplexer is limited to a voltage which is V_{Tn} below the voltage applied to the gate of the pass transistor. Therefore, in the given configuration, the input voltage of the inverter never quite reaches the supply voltage. Thus the PMOS transistor of the inverter has a negative gate bias of V_{Tn} or more applied to it when the input is high. Since V_{Tn} and V_{Tp} are nearly equal in magnitude, the PMOS is either on or on the verge of turning on for an input when it is supposed to be off. Therefore, it has high leakage current. This results in static power consumption in the inverter.

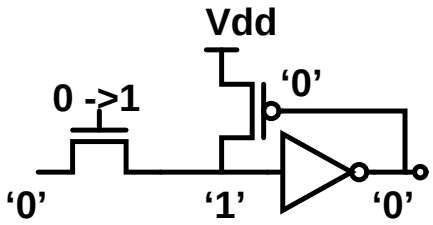
c) Show how we can add a pull-up p channel transistor to remove the leakage problem. Explain how this addition makes the logic configuration ratioed, rather than ratioless.

Soln.: The leakage current can be avoided by adding a pull up PMOS as shown in figure below.



When the multiplexer output (y) is 'low', the inverter output is high. The PMOS is therefore off and has no effect. When the multiplexer output goes 'high', the inverter input charges up, the output starts falling and turns the additional PMOS on. Now, as the multiplexer output (y) approaches $V_{DD} - V_{Tn}$, the NMOS switch in the multiplexer turns off. However, the PMOS pull up remains 'on' and takes the inverter input all the way to V_{DD}. This avoids leakage in the inverter.

However, this solution brings up another problem. Consider the equivalent circuit when the inverter output is 'low' and the PMOS is 'on'.



Now if input wants to change the output to 'high', the multiplexer output should go 'low'. For this, it has to fight the PMOS pullup - which is trying to keep this node 'high'.

In fact, the multiplexer n transistor and the pull up p transistor constitute a pseudo NMOS inverter. Therefore, the multiplexer output cannot be pulled low unless the transistor geometries are appropriately ratioed.

– **[3]**

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

Tuesday	**EE 671: VLSI Design**	Time: 1130-1300
Oct. 18, 2016	Class Test - 2	Marks: 10

Q–1 **a)** What is the motivation for using the generate/kill/propagate signals for carry in an adder? How are these produced?

Soln. 1-a) The main problem in adder design is that more significant bits can't begin their work till the carry has arrived from the less significant bits. Therefore it makes sense to do at least as much of logic computation as possible before the arrival of carry, so that the generation of output carry is as fast as possible.

We know that if both bits at any addition column are '1', output carry should be '1', irrespective of the input carry. In this case, we need not wait for the arrival of carry input, we can output a '1' immediately. This is called the "generate" case.

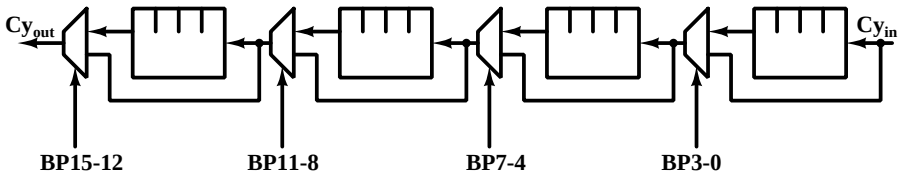
Similarly, if both bits are '0', the output does not depend on the input carry and we can output a '0' immediately. This is called the "kill" case.

In the other two cases when the input bits are "01" or "10", we do have to wait for the input carry. However, since the output carry is equal to the input carry in these cases, we can just propagate the input carry to the output. This is the propagate case.

Thus we have the signals: generate $G = A \cdot B$, kill $K = \overline{A} \cdot \overline{B}$ and propagate $P = A$ xor B. These signals can be produced from A and B in constant time, without having to wait for the input carry. – **[1]**

b) Consider a 16 bit adder with carry bypass over every 4 bits. Show how the worst case delay will be substantially reduced because of breaking up of the critical path.

Soln. 1-b) If no bypass is used, the worst case will occur when all A_is and B_is are unequal. In this case, all stages will need to propagate their input carry, so the output carry will be delayed by 16 propagation delays.



The figure above shows the case when we bypass the carry over every 4 bits. The bypass signals are just ANDs of all the 4 P values. If any P value is '0' near the most significant bit, carry will be either generated or killed at that point and so the total delay up to that point will be irrelevant. Therefore the worst case delay will occur when all P values are '1' near the MSB. For the first group near LSB, if the least significant bit P is '0', this will require propagation of carry over the next 3 bits. Thus the worst case delay will be 3 propagations in the least significant group, followed by 4 propagations through bypass muxes. This is much faster than 16 propagations required without carry bypass. – **[1]**

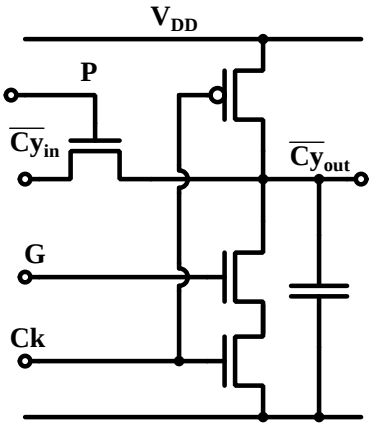
c) Describe the circuit of a dynamic CMOS implementation of a Manchester carry chain adder.

Dynamic CMOS logic has the problem that the output can be at the wrong value for some time after pre-charge (during evaluation), which can lead to malfunction. Will this circuit have the same problem? (Give reasons).

What makes it possible to use OR logic rather than the slower XOR to generate the propagate signal for this adder?

Soln. 1-c)

The figure on the right shows a stage of the dynamic Manchester carry chain.



The signal carried by the chain is $\overline{Cy}$. Width of the transistor driven by G is adjusted so that it can pull the output down to '0' even if the transistor driven by P is passing a '1'. (Ratioed Logic).

When Ck is 'LOW', the output is unconditionally charged to '1'. The output is evaluated when Ck goes 'High'.

During evaluation, the output will discharge unconditionally to '0' if $G = 1$. The only other discharge path is through the pass transistor driven by P. Therefore the output will also discharge to zero if $P = 1$ AND $\overline{Cy_{in}} = 0$. This implements the logic

$$Cy_{out} = G + P \cdot Cy_{in}$$

which is the desired function to generate the output carry. The pass transistor driven by P is required only to discharge the output, not to charge it. Therefore, an n channel transistor can be used instead of a transmission gate.

This circuit does not exhibit the problem associated with CMOS dynamic circuits where a transient wrong value discharges the output. In this case, the pre-charged output does not go the gate terminal of any transistor (which may be erroneously turned on, producing wrong output). The pre-charged output can only be discharged through transistors driven by G and P, which are valid before carry propagation begins.

Ideally P should be the XOR of the corresponding A and B bits. However, XOR and OR outputs differ only when $A = B = 1$. In this particular case, $G = A \cdot B = 1$ and so the output is unconditionally discharged to '0' irrespective of the value of P. Therefore, we can drive the pass transistor with OR of A and B rather than XOR, and still produce the right result. – **[2]**

– **[Q1: 1+1+2 = 4 marks]**

Q–2 a) Describe the wire reduction algorithm used in Dadda multipliers.

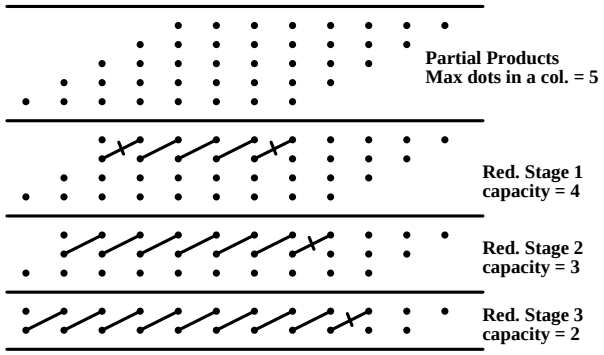
Soln. 2-a) In Dadda multipliers, we first define the maximum number of wires per weight for each stage of reduction. We start from the last stage, where we know there should be no more than 2 wires per weight. Each previous stage can have no more than int(3n/2) wires per weight, where n is the number of wires in the current stage. We continue this process till the allowed number of wires is greater than or equal to the actual number of wires for any weight in the partial products.

For example, in a 5×5 multiplier, the maximum number of wires for any weight is 5. Starting from the last stage, 2 wires are allowed per weight. In the previous stage, a maximum of $\text{int}(2 \times 3/2) = 3$ wires are allowed per weight. Adding another stage, up to $\text{int}(3 \times 3/2) = 4$ wires would be allowed. Since we have 5 wires, yet another stage needs to be added. This will allow $\text{int}(4 \times 3/2) = 6$ wires per weight, which can accommodate all wires without exceeding the limit.

The wire reduction algorithm uses the least number of smallest adders which will reduce the total number of wires going to the next stage (inclusive of incoming carry wires from lower weight) to the allowed numbers. All bunches of wires smaller than or equal to the capacity of next stage are just passed through. This is continued till the last stage, where no more than 2 wires are present for any weight. A wire each is given to two numbers and these two numbers are added using any conventional fast adder. – **[1]**

b) Consider a Dadda multiplier where the multiplicand is 8 bit wide while the multiplier is 5 bit wide. Show the wire reduction scheme for this multiplier using a dot diagram and give a brief description of the reduction at each stage.

Soln. 2-b) The 8×5 multiplier has the partial products aligned as shown. The maximum number of dots in a column is 5.



The wire distribution of partial products (from LSB) is:
1-2-3-4-5-5-5-5-4-3-2-1. The reduction target is 4 wires.
For all columns containing 4 or less wires, all the wires are passed through. The first column with 5 wires can be reduced to 4 using a half adder. The next 3 columns also have 5 wires. Since we anticipate a carry from the adder on the right, we should use a full adder, which takes up 3 wires and outputs a sum wire of the same weight.

The next column has 4 wires, and we anticipate one in-coming carry, so a half adder will reduce the total wires at this weight to 4. The next column has 3 wires and with the in-coming carry, the number will be 4. So nothing has to be done and the remaining columns are also just passed through.

The in-coming wire distribution for stage 2 (from LSB) is
1-2-3-4-4-4-4-4-4-4-2-1. The reduction target is 3 wires.
Wires for the first three columns are just passed through, as the capacity of the stage is 3. The fourth column has 4 wires, which can be reduced to 3 with a half adder. The next six columns have 4 wires each and each of these anticipates a carry from the right. These have to be reduced with a full adder, which takes up 3 wires and outputs one sum at the same weight. One wire is passed through and along

with the in-coming carry, makes up the total for output wires to 3 as desired. The last two columns have 2 and 1 wires, which are just passed through.

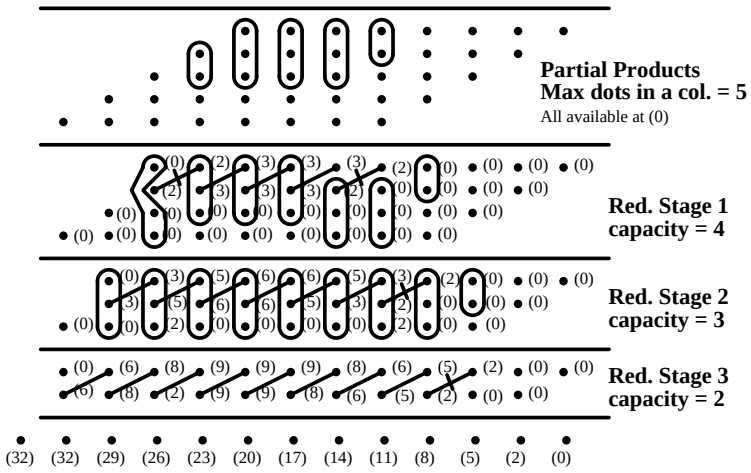
The in-coming wire distribution for stage 2 (from LSB) is
1-2-3-3-3-3-3-3-3-3-1. The reduction target is 2 wires.
The first two columns are just passed through. The next column has 3 wires, which can be reduced to 2 using a half adder. The next 8 columns also have 3 wires and each anticipates a carry from the right. Their outputs can be reduced to 2 using a full adder. (All 3 wires are taken up by the full adder. Its sum output and the in-coming carry wire are the two output wires.) The single wire in the last column with the incoming carry from the right also gives out two wires.

This completes the wire reduction. Each of the wires in columns containing two wires can be fed to a number each and these two numbers can be added using a fast adder. – **[2]**

c) Assume that a half adder provides its sum and carry outputs two units of time after the arrival of the last of its inputs, while a full adder takes 3 units of time to generate sum and carry after the arrival of the latest input. Assume that all partial product bits for the 8×5 multiplier are ready at time 0.

Redraw the dot diagram for the 8×5 Dadda multiplier, placing the time of arrival of each bit in brackets next to each dot.
(At every reduction stage, one can choose which wires go to a full/half adder and which ones are passed through. You should make this choice such that the worst case delay for reaching the conventional adder stage is minimized.)

Soln. 2-c)



– **[2]**

d) Assuming that a ripple carry adder using the same half adder and full adder will be used for the final addition, what is the worst case time at which the product will be ready?

Soln. 2-d) As can be seen from the final row representing the results of ripple carry adder, the product will be ready at 32 units of time. – **[1]**

– **[Q2: 1+2+2+1 = 6 marks]**

Paper Ends

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

End Semester Examination

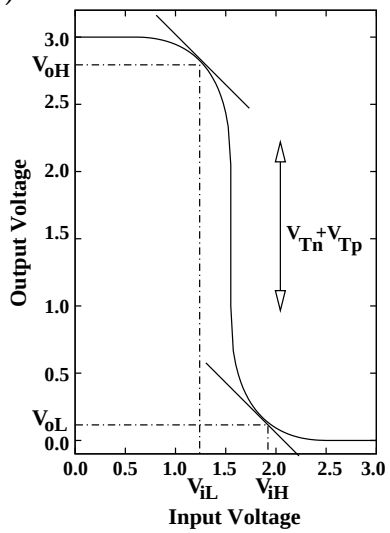
Tuesday	EE 671: VLSI Design	Time: 0930-1230
21-11-17	Autumn Semester 2017	Marks: 50

For all problems in this paper, use the simple transistor model with perfect saturation given at the end of the paper. Unless otherwise specified, take $V_{DD} = 1.8V$, $|V_{Tp}| = 0.45$V and $V_{Tn} = 0.35$V.

Quantitative answers must be accurate at least to 1%.
All intermediate calculations should be reported.

Q–1 **a)** The 'Low' and 'High' logic levels at the input and output are defined by the points on the transfer curve of an inverter where the slope is -1. Justify this choice.
How are static noise margins determined from the 'Low' and 'High' logic levels as defined above?

Soln. 1-a)



The transfer characteristics of a CMOS inverter are shown in the diagram on the left. For all points to the left of the first point and to the right of the second point where the slope $= -1$, the analog gain $\frac{\mathrm{d}V_{out}}{\mathrm{d}V_{in}}$ has an absolute value less than 1. Therefore, any noise or deviation at the input will be diminished at the output. On the other hand, for the region between these two, the gain is > 1. An input in these regions will be amplified at the output. For digital use, we want the output to be insensitive to the actual value of the input voltage, as long as it is within the specified range for the definition of a '0' or a '1'.

Thus points with a slope $= -1$ form natural boundaries for defining the highest voltage for '0' and lowest voltage for '1'. The input and output voltages at the defining points are termed (V_{iL}, V_{oH}) and (V_{iH}, V_{ol}). V_{ol} should be $< V_{iL}$, so that even if the output has some noise added to it, the next stage still treats it as a '0' at its input. Similarly, V_{oH} should be $> V_{iH}$, so that even in the presence of noise, a '1' at the output of one stage is still interpreted as a '1' at the input of the next. Therefore we define the static noise level at '0' as $V_{iL} - V_{ol}$ and the static noise level at '1' as $V_{oH} - V_{iH}$. Both should be positive and as large as possible for good noise immunity. – **[2]**

b) Consider a CMOS inverter. Assume that the geometries of the p and n channel transistors are so chosen that their conductance factors K_p and K_n are equal. Derive an expression for the input 'Low' and output 'High' values in terms of the supply voltage and absolute values of the turn on voltages.

Soln. 1-b) When the input is 'Low' and output is 'High', the n channel transistor will be in saturation and the p channel transistor will be in linear mode of operation. The absolute value of the gate to source voltage for the p channel transistor is

$V_{DD} - V_i$, while the absolute value of its drain to source voltage is $V_{DD} - V_o$ The current through the two transistor should be equal. Therefore,

$$\frac{K_n}{2}(V_i - V_{Tn})^2 = K_p \left((V_{DD} - V_i - V_{Tp})(V_{DD} - V_o) - \frac{1}{2}(V_{DD} - V_o)^2 \right)$$

We define $V_{DP} \equiv V_{DD} - V_o$. Since $K_n = K_p$, we can write,

$$\frac{1}{2}(V_i - V_{Tn})^2 = (V_{DD} - V_i - V_{Tp})V_{DP} - \frac{1}{2}V_{DP}^2$$

Which leads to the quadratic equation

$$\frac{1}{2}V_{DP}^2 - (V_{DD} - V_i - V_{Tp})V_{DP} + \frac{1}{2}(V_i - V_{Tn})^2 = 0$$

with solutions:

$$V_{DP} = (V_{DD} - V_i - V_{Tp}) \pm \sqrt{(V_{DD} - V_i - V_{Tp})^2 - (V_i - V_{Tn})^2}$$

Since the p channel transistor is in the linear regime, we must have $V_{DP} \equiv V_{DD} - V_o < V_{DD} - V_i - V_{Tp}$ and therefore, we must choose the negative sign in the equation. So

$$V_{DP} \equiv V_{DD} - V_o = (V_{DD} - V_i - V_{Tp}) - \sqrt{(V_{DD} - V_i - V_{Tp})^2 - (V_i - V_{Tn})^2}$$

$$\text{Therefore} \quad V_o = V_i + V_{Tp} + \sqrt{(V_{DD} - V_i - V_{Tp})^2 - (V_i - V_{Tn})^2}$$

$$\text{So} \quad V_o = V_i + V_{Tp} + \sqrt{(V_{DD} - V_{Tn} - V_{Tp})(V_{DD} - 2V_i + V_{Tn} - V_{Tp})}$$

Taking the derivative with respect to V_i and putting it equal to -1, we get

$$-1 = 1 - \sqrt{\frac{V_{DD} - V_{Tn} - V_{Tp}}{V_{DD} - 2V_i + V_{Tn} - V_{Tp})}}$$

$$\text{squaring, we get } 4 = \frac{V_{DD} - V_{Tn} - V_{Tp}}{V_{DD} - 2V_i + V_{Tn} - V_{Tp}}$$

$$\text{Therefore } 4V_{DD} - 8V_i + 4V_{Tn} - 4V_{Tp} = V_{DD} - V_{Tn} - V_{Tp}$$

Which gives

$$V_{iL} = \frac{3V_{DD} + 5V_{Tn} - 3V_{Tp}}{8}$$

Substituting for V_i in the equation

$$V_o = V_i + V_{Tp} + \sqrt{(V_{DD} - V_{Tn} - V_{Tp})(V_{DD} - 2V_i + V_{Tn} - V_{Tp})}$$

we get

$$V_{oH} = \frac{7V_{DD} + V_{Tn} + VTp}{8} = V_{DD} - \frac{V_{DD} - V_{Tn} - VTp}{8}$$

These are the values of the input Low and output High voltages. – **[4]**

c) For a CMOS inverter, $K_p = K_n = 100\mu\text{A/V}^2$. For what input voltage is the static current drawn by the inverter maximum? What is the value of this maximum static current?

Soln. 1-c) Both transistor will be on and in saturation for current to be maximum. When both are in saturation, their saturation currents must be equal. So

$$\frac{K_n}{2}(V_i - V_{Tn})^2 = \frac{K_p}{2}(V_{DD} - V_i - V_{Tp})^2$$

For $K_n = K_p$ this gives

$$V_i - V_{Tn} = V_{DD} - V_i - V_{Tp}, \quad \text{and therefore,} \quad V_i = \frac{V_{DD} - V_{Tp} + V_{Tn}}{2}$$

For the given constants,

$$V_i = \frac{1.8 - 0.45 + 0.35}{2} = 0.85\text{V}$$

At this voltage, the saturation current of the nMOS transistor is:

$$I_{max} = \frac{100 \times 10^{-6}}{2}(0.85 - 0.35)^2 = 50 \times 10^{-6}(0.25) = 12.5\mu\text{A}$$

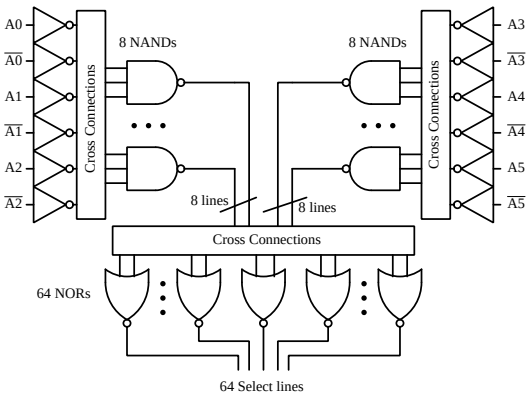
– **[2]**

d) Assume that the widths of the p channel transistors need to be twice the width of n channel transistors for matching their conductance factors K_p and K_n. The minimum width of an n channel transistor is 250 nm. Using thumb rules for scaling geometries, find the widths of the four transistors in a tri-stateable inverter so that it meets the same dynamic specifications as the minimum inverter with equal K_p and K_n values.

Soln. 1-d) The tri-stateable inverter has 2 n channel transistors in series and the 2 p channel transistors are also in series. In the inverter, the p channel transistor is twice as wide as the n channel transistor. Therefore in the tri-stateable inverter, the n width will be twice the n channel transistor width in the minimum inverter = 500 nm. The p channel width will be double this, = 1μm. – **[1]**

– **[Q1: 2+4+2+1 = 9 marks]**

Q–2



A two step decoder for 6 address lines, producing 64 select outputs is shown in the diagram on the left. Combinations of 3 lines out of $A0, \overline{A0}, A1, \overline{A1}, A2, \overline{A2}$ are fed to the first bank of 8 three-input NAND gates. Similarly, combinations of 3 lines out of $A3, \overline{A3}, A4, \overline{A4}, A5, \overline{A5}$ are fed to the other bank of 8 three-input NAND gates. 64 two-input NOR gates then accept one line each from the two banks of NANDs to produce 64 select lines.

The select lines are required to drive heavy loads equivalent to 512 minimum inverters each. Assume that the γ value representing the ratio of p channel widths to n channel widths in an inverter to produce equal rise and fall times is 2, and the parasitic inverter delay is 2.5.

– **P.T.O.**

a) What is the the parasitic delay and the logical effort of 3 input NAND gates and 2 input NOR gates? (Parasitic delay can be taken to be proportional to the total capacitive load at the output node for a gate providing equivalent drive to a minimum inverter). Report the results in a table.

Soln. 2-a) The three input NAND has 3 n channel transistors in series, so each will have a width of 3. Since $\gamma = 2$, the 3 p channel transistors which are in parallel, will have widths of 2. The two input NOR will have two n channel transistors in parallel, so each will have a geometry of 1. There are two p channel transistors in series and $\gamma = 2$, so each will have a width of 4. Geometries and consequently the logical effort and parasitic delay of the components in use will be as follows:

Gate	Width of n Trans.	Width of p Trans.	Total W per input	Total W at output	Logical Effort	Parasitic Delay
Inverter	1	2	3	3	1	2.5
3-input-NAND	3	2	5	9	5/3	7.5
2-input-NOR	1	4	5	6	5/3	5.0

– **[1]**

b) What is the optimum number of stages for minimum delay in this circuit, assuming that each of the input lines can drive 1 minimum sized inverter. Show the recommended configuration for the two step decoder with added inverters if necessary, so that overall delay is minimized.

Soln. 2-b) The input can drive one inverter while the output is required to drive 512 inverters, so $H = 512$.

The logical effort of NANDs as well as NORs is 5/3. Therefore $G = 5/3 \times 5/3 = 25.9$.

There are 6 input inverters which drive 8 three-input NANDs *i.e.* 24 inputs. Therefore the branching factor at the output at the first inverter is $24/6 = 4$. A total of 16 NANDs drive 64 two-input NORs. Therefore the branching factor at the output of NANDs is $64 \times 2/16 = 8$. Thus the overall branching factor is $4 \times 8 = 32$. Thus the overall path effort is $F = GBH = 512 \times 32 \times 25/9 = 45511.11$

We can find the asymptotic stage ratio ρ by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0 \quad \text{with } p_{inv} = 2.5$$

We define $f \equiv 2.5 + \rho(1 - \ln \rho)$. Then

$$f' = (1 - \ln \rho) + \rho\left(-\frac{1}{\rho}\right) = 1 - \ln \rho - 1 = -\ln \rho$$

Then starting with a guess value g for ρ, the next guess is given by Newton Raphson technique as

$$g - \frac{f}{f'} = g + \frac{2.5 + g(1 - \ln g)}{\ln g} = g + \frac{2.5 + g}{\ln g} - g = \frac{2.5 + g}{\ln g}$$

Starting with a guess value of 4, we iterate to get values for ρ as: 4.6888, 4.6524, 4.6523, 4.6523 ...
Thus, $\rho = 4.6523$. The optimum number of stages is then

$$N = \frac{\ln F}{\ln \rho} = \frac{\ln 45511.11}{\ln 4.6523} = 6.9767$$

Therefore the optimum number of stages is 7.

Correspondingly, the actual stage ratio for 7 stages is

$$\hat{f} = F^{1/N} = 45511.11^{1/7} = 4.6286$$

The circuit as given has three stages – an inverter followed by a NAND, followed by a NOR. Therefore four inverters should be added to optimize delay.

It is clear that the sizes will be scaled up as we go from the input side to the final load. Therefore, it is preferable to have larger inverters rather than larger NANDS or NORS. This implies that we should try to put all four additional inverters at the end.

To check if this will run into problems, we start from the beginning.

Since the inputs can drive a single minimal inverter, the first inverter should have unit size. The stage effort of all stages should be $\hat{f} = 4.6286$. Therefore

$$gbh = 1 \times 4 \times \frac{C_{out}}{1} = 4.6286, \text{ So } C_{out} = 4.6286/4 = 1.1571$$

C_{out} of the inverter stage is C_{in} of the NAND stage. Each inverter represents transistor widths of 3 units. Therefore the NANDs can have a total transistor width of $3 \times 1.1571 = 3.4714$ at the input. This total width will be divided in the ratio 3:2 over the n and p channel input transistors. Thus the n channel transistor width will be $3 \times 3.4714/5 = 2.0828$, and the p channel transistor width will be $2 \times 3.4714/5 = 1.3886$. These widths are acceptable.

For the second stage, the NAND drives 8 NOR gates, and has a logical effort of 5/3. Thus

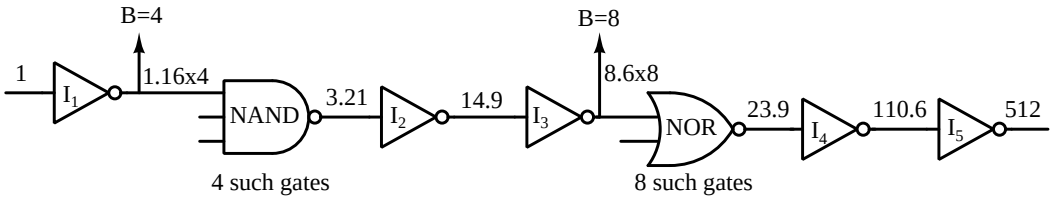
$$gbh = \frac{5}{3} \times 8 \times \frac{C_{out}}{1.1571} = 4.6286, \quad \text{So } C_{out} = \frac{4.6286 \times 3 \times 1.1571}{5 \times 8} = 0.4017$$

If we put NOR gates immediately following the NAND gates, the total input capacitance of each gate should be 0.4017 inverter units – or $3 \times 0.4017 = 1.2051$ transistor widths. This is too small to fit a NOR gate, which requires 5 transistor widths even while using minimum sized nMOS transistors for pull down.

Therefore we need to insert two inverters between the NAND and the NOR and do the 8 way branching to NORs *after* the inverters. (A single inverter will require that we change the NOR gate to AND – which will change the overall logical effort.).
So the final configuration should be – inverter, NAND, inverter, inverter, NOR, inverter, inverter for the seven stages. – **[3]** .

c) Compute the scale factors and absolute geometries for all transistors in the design.

Soln. 2-c)



Since the first inverter I_1 is to be driven by the inputs which can drive only one minimum inverter, its size should be minimum. This inverter drives 4 NAND gates.

$$gbh = 1 \times 4 \times \frac{C_{out}}{1} = 4.6286, \quad \text{So } C_{out} = \frac{4.6286}{4} = 1.1571$$

Therefore the input capacitance of the NAND gates should be equivalent to 1.1571 inverters.

The NAND gate is now followed by an inverter (I_2) without any branching.

$$gbh = \frac{5}{3} \times 1 \times \frac{C_{out}}{1.1571} = 4.6286, \quad \text{So } C_{out} = \frac{4.6286 \times 3 \times 1.1571}{5} = 3.2135$$

Therefore the input capacitance of I_2 should be 3.2135 units.

I_2 is followed by another inverter (I_3) without any branching at the output of I_2.

$$gbh = 1 \times 1 \times \frac{C_{out}}{3.2135} = 4.6286, \quad \text{So } C_{out} = 4.6286 \times 3.2135 = 14.8740$$

I_3 is required to drive 8 NORs. Then,

$$gbh = 1 \times 8 \times \frac{C_{out}}{14.8740} = 4.6286, \quad \text{So } C_{out} = \frac{4.6286 \times 14.8740}{8} = 8.6056$$

Thus the input capacitance of NORs is required to be 8.6056 inverter units. The NOR gate drives a single inverter (I_4).

$$gbh = \frac{5}{3} \times 1 \times \frac{C_{out}}{8.6056} = 4.6286, \quad \text{So } C_{out} = \frac{4.6286 \times 8.6056 \times 3}{5} = 23.899$$

So the input capacitance of I_4 is 23.899. This inverter drives the final inverter (I_5) without branching.

$$gbh = 1 \times 1 \times \frac{C_{out}}{23.899} = 4.6286, \quad \text{So } C_{out} = 4.6286 \times 23.899 = 110.62$$

Therefor the final inverter has input capacitance of 110.62. I_5 drives the final load.

$$gbh = 1 \times 1 \times \frac{C_{out}}{110.62} = 4.6286, \quad \text{So } C_{out} = 4.6286 \times 110.62 = 512$$

Which confirms that the final inverter will drive a load of 512.

Alternative Computation

We could have started from the output end, once the configuration has been decided. In this case we begin with C_{out} values and evaluate C_{in} for each stage.
I_5 drives 512 inverters, so

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{512}{C_{in}}, \quad \text{So } C_{in} = \frac{512}{4.6286} = 110.62$$

I_4 is required to drive a load equivalent to 110.62 min. inverters, so

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{110.62}{C_{in}}, \quad \text{So } C_{in} = \frac{110.62}{4.6286} = 23.899$$

NOR Gate drives a load equivalent to 23.899 min. inverters. Therefore

$$\hat{f} = 4.6286 = \frac{5}{3} \times 1 \times \frac{23.899}{C_{in}}, \quad \text{So } C_{in} = \frac{23.899 \times 5}{3 \times 4.6286} = 8.6056$$

I_3 drives 8 NOR gates. Therefore,

$$\hat{f} = 4.6286 = 1 \times 8 \times \frac{8.6056}{C_{in}}, \quad \text{So } C_{in} = \frac{8 \times 8.6056}{4.6286} = 14.8740$$

I_2 just drives I_3. Therefore

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{14.8740}{C_{in}}, \quad \text{So } C_{in} = \frac{14.8740}{4.6286} = 3.2135$$

NAND drives I_2.

$$\hat{f} = 4.6286 = \frac{5}{3} \times 1 \times \frac{3.2135}{C_{in}}, \quad \text{So } C_{in} = \frac{3.2135 \times 5}{3 \times 4.6286} = 1.1571$$

I_1 drives 4 NANDs.

$$\hat{f} = 4.6286 = 1 \times 4 \times \frac{1.1571}{C_{in}}, \quad \text{So } C_{in} = \frac{4 \times 1.1571}{4.6286} = 1.0$$

So the input capacitance of $I_1 = 1$ as expected.

Calculation of geometries:

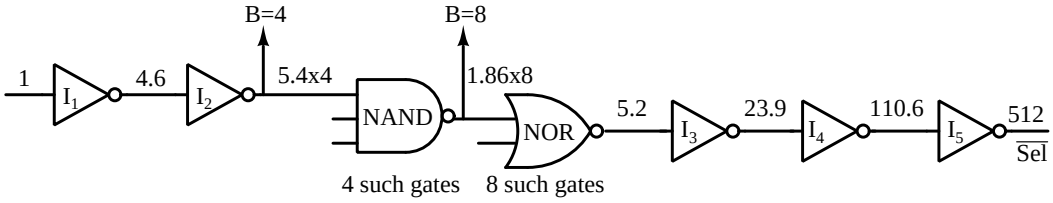
One inverter load is equivalent to 3 minimum transistor widths. We can work out transistor widths from input capacitances as follows:
I_1: $C_{in} = 1$ inv. $= 3$ transistor widths. This is a minimum inverter with n width $= 1$ and p width $=2$.
NAND: $C_{in} = 1.1571$ inv. $= 1.1571 \times 3 = 3.4713$ transistor widths. Therefore n width $= 2.08$, p width $= 1.39$.
I_2: $C_{in} = 3.2135$ inv. So n width $= 3.2135$, p width $= 6.427$
I_3: $C_{in} = 14.874$ inv. So n width $= 14.874$, p width $= 29.748$
NOR: $C_{in} = 8.6056$ inv. $= 25.82$ widths. Therefore n width $= 25.82 \times 1/5 = 5.163$ and p width $= 20.653$
I_4: $C_{in} = 23.899$, so n width $= 23.9$ and p width $= 47.8$
I_5: $C_{in} = 110.62$, so n width $= 110.62$, p width $= 221.24$.

These results are tabulated below:

	I_1	NAND	I_2	I_3	NOR	I_4	I_5
C_{in}	1	1.157	3.21	14.87	8.61	23.9	110.6
n width	1	2.08	3.21	14.87	5.16	23.9	110.6
p width	2	1.39	6.42	29.75	20.65	47.8	221.2

Alternative Configuration

To avoid the problem with heavy branching between the NAND and NOR stages, one can add another inverter *before* the NAND gates. This will make the NAND gate larger and enable it to drive 8 reasonable sized NOR gates. This configuration is shown below:



Notice that putting an extra inverter at the input still provides us with all address lines and their complements. Therefore at the NAND stage, we still have the required combinations to decode and the extra inversion does not matter. However, now there will be 3 inverters after NOR, so the output will be $\overline{Sel}$.

We can compute the scale factors and sizes for this configuration as:
Final stage loading is 512, which is driven by I_5. Therefore

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{512}{C_{in}}, \text{ So } C_{in} = \frac{512}{4.6286} = 110.62$$

I_4 is required to drive a load equivalent to 110.62 min. inverters, so

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{110.62}{C_{in}}, \text{ So } C_{in} = \frac{110.62}{4.6286} = 23.899$$

I_3 is required to drive a load equivalent to 23.899 min. inverters, so

$$\hat{f} = 4.6286 = 1 \times 1 \times \frac{23.899}{C_{in}}, \text{ So } C_{in} = \frac{23.899}{4.6286} = 5.1634$$

NOR Gate drives a load equivalent to 5.1634 min. inverters. Therefore

$$\hat{f} = 4.6286 = \frac{5}{3} \times 1 \times \frac{5.1634}{C_{in}}, \text{ So } C_{in} = \frac{5.1634 \times 5}{3 \times 4.6286} = 1.8592$$

The NAND gate drives 8 such NOR gates.

$$\hat{f} = 4.6286 = \frac{5}{3} \times 8 \times \frac{1.8592}{C_{in}}, \text{ So } C_{in} = \frac{1.8592 \times 5 \times 8}{3 \times 4.6286} = 5.3559$$

I_2 drives 4 such NAND gates. Therefore,

$$\hat{f} = 4.6286 = 1 \times 4 \times \frac{5.3559}{C_{in}}, \text{ So } C_{in} = \frac{4 \times 5.3559}{4.6286} = 4.6286$$

I_1 drives the inverter I_2.

$$\hat{f} = 4.6286 = 1 \times 4 \times \frac{4.6286}{C_{in}}, \text{ So } C_{in} = \frac{4.6286}{4.6286} = 1.0$$

So the input capacitance of I_1 is 1 as expected. From the input capacitances, we can compute transistor geometries as before: capacitance of 1 inverter is equivalent to 3 transistor widths.
For inverters, n width = C_{in}, p width = $2 \times C_{in}$
For the NAND, the width ratio between n and p is 3:2.
So n width = $C_{in} \times 3 \times 3/5$, p width = $C_{in} \times 3 \times 2/5$
For the NOR, the width ratio between n and p is 1:4.
n width = $C_{in} \times 3 \times 1/5$, p width = $C_{in} \times 3 \times 4/5$

	I_1	I_2	NAND	NOR	I_3	I_4	I_5
C_{in}	1	4.63	5.36	1.86	5.16	23.9	110.6
n width	1	4.63	9.64	1.12	5.16	23.9	110.6
p width	2	9.26	6.43	4.48	10.32	47.8	221.2

– **[6]**

d) Compute the delay of each stage and the total delay for the decoder.

Soln. 2-d) The effort delay of each stage is 4.6286. The parasitic delay should be added to it to get the total stage delay.

The delay for each of the 5 inverter stages is $4.6286 + 2.5 = 5.1286$ units.
The delay of the NAND stage is $4.6286 + 7.5 = 12.1286$ units.
The delay of the NOR stages is $4.6286 + 5 = 9.6286$ units
Therefore the total delay is $5 \times 5.1286 + 12.1286 + 9.6286 = 47.4$ units.

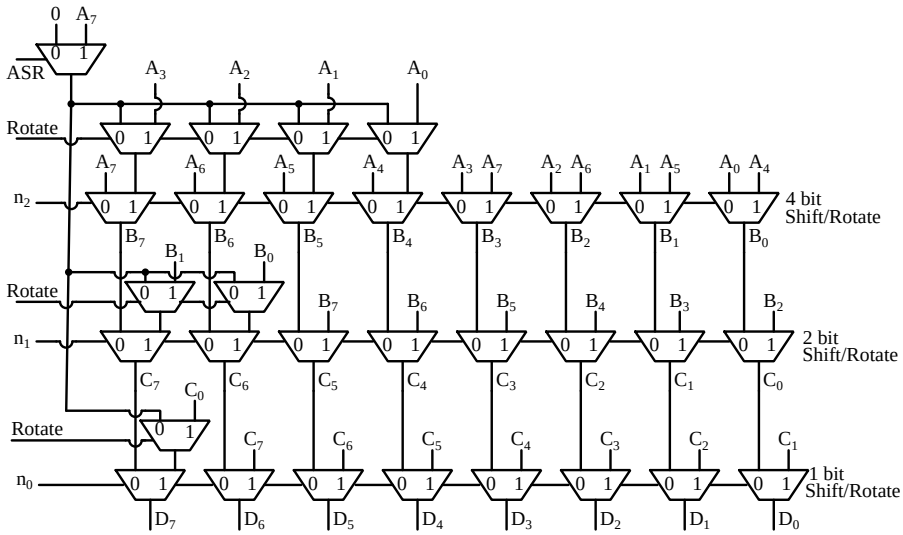
If the inverters were not added, the stage effort for each stage would have been $45511.11^{1/3} = 35.7031$.
The total delay in this case will be 35.7031+2.5+35.7031+7.5+35.7031+5 = 122.11 units.
(This is about 2.5 times the optimal delay!)

– **[Q2: 1+3+6+1 = 11 marks]**

Q–3 a) Show how we can construct a barrel shifter to carry out logical right shift, arithmetic right shift and rotate right, using only 2 input muxes.
Assume the operand to be 8 bit wide.

Soln. 3-a) The rotate operation can be performed by three rows of 2 input muxes. The top row selects between A_i and A_{i+4}, controlled by the top bit (n_2) of the shift amount. (Shift by 4 or 0 positions). Here $i + 4$ is modulo 8.

Similarly the second row selects between A_i and A_{i+2}, controlled by the middle bit of shift amount (n_1). The third row selects between A_i and A_{i+1}, controlled by the lowest bit (n_0) of the shift amount.



If a shift rather than rotate is required (Rotate = '0'), then top 4 bits of the first row, the top two bits of the second row and the top bit of the last row are fed 0. However, if an arithmetic shift is desired, than the value fed to top 4 bits of top row, top 2 bits of the middle row and the top bit of the bottom row is A_7 rather than 0.

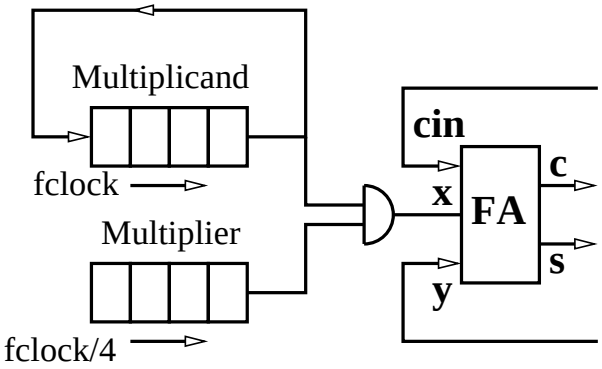
This is implemented by adding a 2 input mux which chooses between A_7 and 0 based on the control input 'ASR'. The selected value is then used by 4 additional muxes in the top row, 2 additional muxes in the middle row and 1 mux in the bottom row to replace the shifted data bits by this value. – **[5]**

b) Describe the operation of a bit serial multiplier, using a 4×4 multiplier as an example. Explain the operations which need to be carried out to take care of exceptions at the end of a row and show how these are implemented in hardware.

Soln. 3-b) The partial product generation as well as addition of partial products must be done serially in a bit serial multiplier. Each bit of the multiplier needs to be ANDed with each bit of the multiplicand to generate the partial products. This requires that all multiplicand bits be presented one after the other, every time a new bit from the multiplier is taken up. This can be managed by using a re-circulating shift register for the multiplicand, which is clocked at a rate which is m times faster than the clock to the multiplier shift register.

These partial products must be presented serially to one input of a full adder. The other input and C_{in} to the full adder have to be appropriately selected and timed to generate the correct product.

Consider a 4×4 bit serial multiplier.

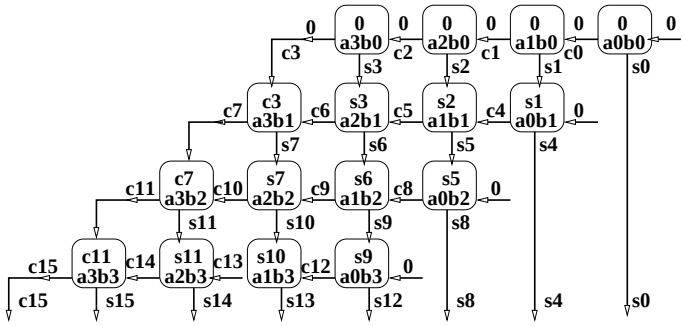


We need additions as follows: (terms in parentheses against each partial product are the times when these should be added).

			a3	a2	a1	a0
		×	b3	b2	b1	b0
			a3b0(3)	a2b0(2)	a1b0(1)	a0b0(0)
		a3b1(7)	a2b1(6)	a1b1(5)	a0b1(4)	
	a3b2(11)	a2b2(10)	a1b2(9)	a0b2(8)		
a3b3(15)	a2b3(14)	a1b3(13)	a0b3(12)			

for all additions, the earlier terms have to wait for 3 clock cycles before the later terms arrive. We can manage this by putting a 3 bit shift register at the sum output and presenting the delayed output at the 'y' input of the full adder. The carry output can be added immediately in the next clock, since it should always go to the next column to its left.

However just a 3 bit delay for the sum will not do as there has to be some exception handling at the end of each row of partial sums.

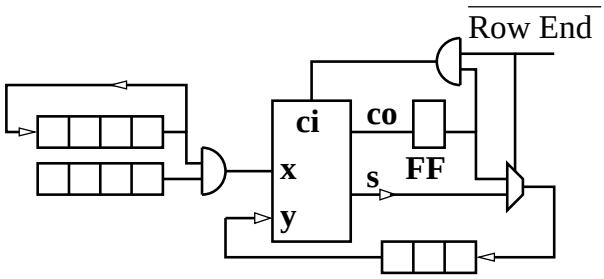


In the figure above, sum and carry terms are indexed by the clock interval in which these were generated. At clocks 0, 4, 8 and 12, carry input should be forced to 0. At clocks 7, 11 and 15, the adder y input should receive carry terms (c3, c7 and c11) instead of sum terms (s4, s8 and s12).

At clocks 0, 4, 8 and 12:

- Carry input should be forced to 0.
- The carry FF output (which is a 1 clock delayed version of cout) should be inserted in the 3-bit shift register.
  Thus C3 (which is always 0), C7 and C11 will emerge at clocks 7, 11 and 15 respectively.
- The sum terms should be taken out as result bits.

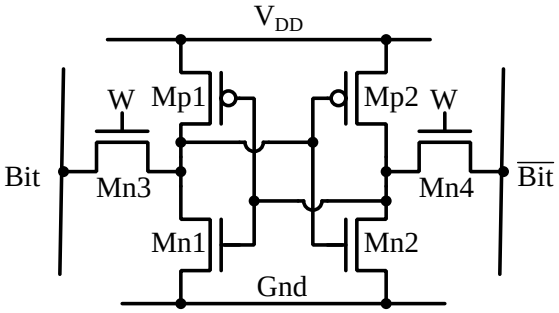
With this exception handling, the bit serial multiplier will be:



This circuit will do a 4x4 multiplication serially. – **[5]**

– **[Q3: 5+5 = 10 marks]**

Q–4



A six transistor static memory cell includes two cross-connected inverters and two N type access transistors (MN3 and MN4) connected to bit and $\overline{\text{bit}}$ lines. Gates of the access transistors are raised to V_{DD} by the word line W when the row containing the cell is selected.

a) Why do we use bit as well as $\overline{\text{bit}}$ lines? What would be the problem if we used only one of these?

Soln. 4-a) The change in voltage at the bit lines during the read operation is very small. This voltage cannot be sensed easily using single ended circuits. Differential circuits cancel out common mode noise and variations in pre-charge values of the bit line. Therefore we use bit and $\overline{\text{bit}}$ lines and a differential sense amplifier which brings up the small voltage differential to a rail to rail digital value.

If we can reliably read a smaller voltage difference, the time to discharge the bit / $\overline{\text{bit}}$ lines through the memory cell can be reduced, thus making the memory faster. – **[1]**

b) How is a "butterfly" diagram used for describing the behaviour of a cascade of two inverters? How is it used to find the stable and meta-stable equilibrium points of cross connected inverters? – **[2]**

c) The capacitance of the bit line is 2pF and it is initially charged to V_{DD}. Transistor MN1 is ON while MN2 is OFF. When the word line goes to V_{DD}, the bit line needs to be discharged to 1.6V for reliable reading by the sensing circuit. Assume that $K_n = 100\mu\text{A/V}^2$ for MN3.

i) If Mn1 is twice as wide as Mn3, find the voltage at the source of MN3 just as the discharge current starts flowing.

Soln. 4-c i) If the voltage at the source of Mn3 is V_s, its drain-source voltage as well as the gate-source voltage is $V_{DD} - V_s$. This means that Mn3 is in saturation. The gate-source voltage of Mn1 V_{DD}, while its drain voltage is small ($= V_s$). Therefore, this transistor is in the linear mode. The load transistor Mp1 is

OFF, because its gate is also at V_{DD}. Since Mn3 and Mn1 are in series, their drain currents must be equal. This gives:

$$\frac{K_{n3}}{2}(V_{DD} - V_s - V_{Tn})^2 = K_{n1}\left((V_{DD} - V_{Tn})V_s - \frac{1}{2}V_S^2\right)$$

Defining $V_1 \equiv V_{DD} - V_{Tn}$ and $\beta \equiv K_{n1}/K_{n3}$, we get

$$(V_1 - V_s)^2 = 2\beta\left(V_1 V_s - \frac{1}{2}V_s^2\right)$$

Which gives

$$V_1^2 + V_s^2 - 2V_1V_s = 2\beta V_1 V_s - \beta V_s^2$$

$$\text{Hence } (1+\beta)V_s^2 - 2(1+\beta)V_1V_s + V_1^2 = 0$$

Solving this quadratic equation, we get

$$V_s = \frac{2(1+\beta)V_1 \pm \sqrt{4(1+\beta)^2V_1^2 - 4(1+\beta)V_1^2}}{2(1+\beta)} = V_1 \pm \sqrt{V_1^2 - \frac{V_1^2}{1+\beta}}$$

Since Mn1 is in linear mode, its drain voltage should be less than $V_{DD} - V_{Tn} (\equiv V_1)$. Therefore,

$$V_s = V_1\left(1 - \sqrt{1 - \frac{1}{1+\beta}}\right) = V_1\left(1 - \sqrt{\frac{\beta}{1+\beta}}\right)$$

For $\beta = 2$, it evaluates to

$$V_s = (1.8 - 0.35)(1 - \sqrt{2/3}) = 0.2661\text{V}$$

– **[4]**

ii) Assuming that the current through MN3 remains constant at its initial value, find the time required to discharge the bit line to 1.6V.

Soln. 4-c ii) MN3 is in saturation with $V_{GS} = 1.8 - 0.2661$ V. Therefore current through MN3, which discharges the bit line, is

$$\frac{K_{n3}}{2}(V_{GS} - V_{Tn})^2 = 50 \times 10^{-6}(1.8 - 0.2661 - 0.35)^2 = 70.08\mu\text{A}$$

The time taken to discharge the capacitor by (1.8-1.6) = 0.2 V will be $C\Delta V/I$ $= 2.0 \times 10^{-12} \times 0.2/70.08 \times 10^{-6}$ or 5.7075 ns. – **[1]**

d) Describe the sequence of operations during read cycle in a static RAM. How is it possible for the stored data to be destroyed during read if the RAM cell is not carefully designed?

Soln. 4-d) During the read cycle, the following actions take place.

1. The address is placed by the processor on the address lines.
2. The row and column address are decoded.
3. Bit and $\overline{\text{Bit}}$ lines are pre-charged.
4. The sense amplifier of the selected column is activated.

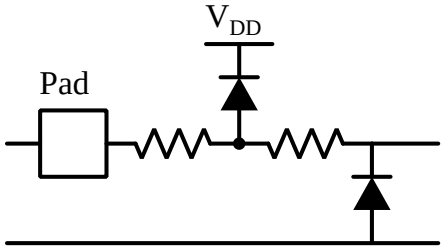
5. Word line for the selected row is pulled high.
6. After the time required to discharge one of the Bit and $\overline{\text{Bit}}$ lines, the Word line is brought low again.
7. The column amplifier determines which of the Bit and $\overline{\text{Bit}}$ lines is lower and outputs a '0' or a '1' accordingly.
8. This digital data is then buffered to the output pad.

As seen in the part above, during the read operation when we are trying to discharge the capacitance associated with the Bit or $\overline{\text{Bit}}$ lines through Mn3 and Mn1, the drain voltage of Mn1 rises by some amount. (It was 0.2661 V in the part above). This voltage is dependent on the value of β. If β is not large enough, this voltage rise can exceed V_Tn. If that happens, Mn2 will be turned ON, which will reduce the output voltage of the inverter Mn2-Mp2. This reduces the gate voltage on Mn1-Mp1 - which will further raise the voltage at the drain of Mn1. This positive feed back action can flip the memory cell such that Mn1 of OFF and Mn2 is ON. Thus the action of reading a cell could destroy the data stored in the cell. – **[2]**

– **[Q4: 1+2+4+1+2 = 10 marks]**

Q–5 a) At the input pads of a CMOS IC, we need to protect against high electrostatic voltages during handling and voltage excursions below ground and above the supply voltage during operation. What kind of device structures are used for protection against these hazards?

Soln. 5-a) Electrostatic voltage from human and machine handling can reach kilo volts in magnitudes. To protect thin oxides and junctions in the IC from these high voltages, special structures are incorporated on the chip.



A commonly used structure is the diode clamp shown in the figure on the left. A p+ diffusion in an n well acts a resistor as well as a diode to V_{DD}. Similarly, an n+ diffusion in a p well provides the resistor as well a diode to ground. The diode to V_{DD} limits any excursion about V_{DD} to within a diode drop and the diode to ground limits negative excursions.

However the IC may be exposed to handling before it is connected to a supply. In that case, high voltages will appear at the supply line through the upper diode. A clamp circuit has to be provided between V_{DD} and ground to prevent this from damaging the circuit. This clamp circuit is often a modified SCR (latchup) structure which occurs naturally in a CMOS circuit, or a field MOSFET, which used the field oxide instead of the gate oxide as the gate dielectric. This will turn on at voltages higher than V_{DD} and clamp the voltage on the supply line to its turn on voltage.

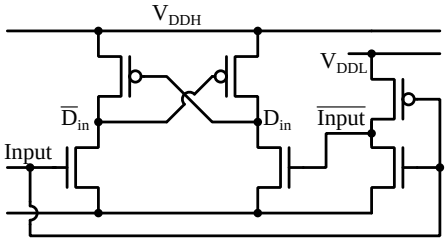
Since external voltages may transiently go below ground or above V_{DD}, it may forward bias the normally reverse biased junction of transistor source/drain with substrate or well. This injected current may turn on the latchup structures, which will damage the IC. To prevent this, all substrate and well regions around the transistors connected to external pads need to have guard bands of p+ regions in the p substrate and n+ regions around the n wells. These guard bands are

connected to ground and V_{DD} respectively through multiple contacts, so that they can collect the injected currents. – **[3]**

b) A chip designed for a supply voltage of 3.3V has to accept inputs from a source which provides logic levels of 0 to 1.8V. How can we use CVSL logic to translate the low swing logic at the input pads to the higher swing logic for use inside a chip? (Assume that a low voltage supply compatible with the low swing input is available on-chip).

Soln. 5-b) Since the input is low swing, the voltage corresponding to a '1' is only 1.8V. However, the internal CMOS circuits are operating at 3.3V. The PMOS transistor of an inverter will have its source connected to 3.3V and the gate connected to this input can only reach 1.8V. Thus the gate sees a negative bias of -3.3 V when the input is '0' and a negative bias of -1.5V while the input is '1'. (V_{Tp} is typically about half a volt). So the PMOS cannot be turned OFF by a '1' at the input. Then both the PMOS and NMOS transistors of a CMOS inverter will be ON when the input is '1' and static power will be wasted.

Thus we need a circuit in which PMOS is not driven by the input. We could have used a pseudo NMOS design, but that uses static power any way. Therefore a CVSL inverter, which does not waste static power and does not drive the PMOS by the input is an attractive choice.



The circuit on the left receives a low swing input and inverts it using an inverter fed from the low voltage V_DD. Since the source of the pMOS of this inverter is at 1.8V, it will be turned off properly when the low swing input is '1'. The input and its complement so generated then drive a CVSL inverter as shown. The output provides a high swing version of the input data (and its complement). – **[2]**

c) How does the large output driver of an output pad also act as a protection device?

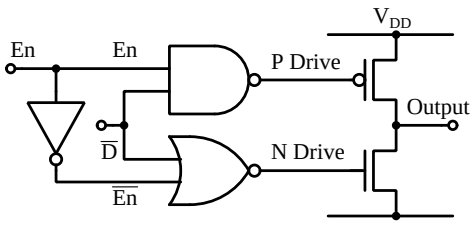
Soln. 5-c) The nMOS driver has a large n+ drain region in a p substrate connected to ground. This constitutes a diode to ground with the same polarity as the one shown for the input pad in part (a) above. The pMOS transistor has a large p+ drain region in an n well connected to V_{DD}. This constitutes a diode connected to supply with the same polarity as shown for the input pad above in part (a). Thus the output buffer itself acts as a protection device. – **[1]**

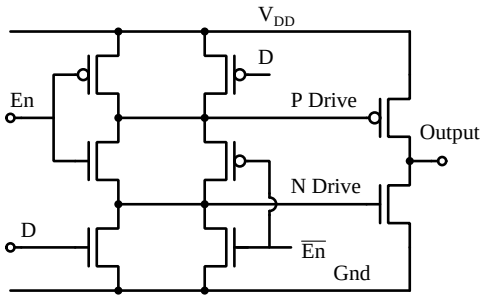
d) In a bidirectional pad, the output drivers need to be tri-stateable. Why is a NAND-NOR based driver structure preferred over 4 transistor tri-stateable inverters at the output?

Soln. 5-d) The tri-stateable inverter uses two pMOS transistors in series and two nMOS transistors in series. Output buffers need to have large widths in order to drive external loads. However, due to the series connection, these widths have to be doubled for both transistors. This requires a large area.

The NAND-NOR circuit shown below or its compact version can provide the ability to tri-state the output buffer, without needing a series transistor. As can be seen from the figure, when En = 0, the NAND output is forced to '1' which turns off the pMOS. At the same time, $\overline{\text{En}}$ is '1', which forces the NOR output to '0' and turns off the nMOS. Thus the output has both pMOS and nMOS turned off when

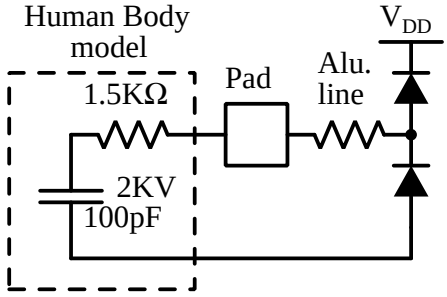
En=0. This accomplishes the tri-stating requirement.





The circuit on the right performs the same function as a Nand-Nor in a single 6 transistor circuit. – **[1]**

e) The human body model is used to emulate the electrostatic hazard to integrated circuits due to handling by human beings.



A 100 pF capacitor is charged to 2KV and discharged through a series resistor of 1.5KΩ and the IC input. Inside the IC, the bond pad size is 70μm ×70μm, which leads to an aluminium line which is 100μm long and 1μm wide. Aluminium thickness is 0.5μm in this layer. The aluminium line leads to a protection diode, whose breakdown voltage is negligible compared to 2KV, so the line may be considered to be terminated to ground.

Estimate the temperature rise in this line when the capacitor is discharged through the external 1.5KΩ resistor and this line to ground. The following assumptions may be made:
– The pad contributes negligibly to the resistance of the line.
– The entire mass of aluminium in the pad and line will be uniformly heated.
– No heat is lost to other structures in this short duration.
– The entire energy stored in the capacitor is used for heating the external resistor, pad and the line.

The relative density of aluminium is 2.7, its specific heat is 0.9J per gram per degree K and its resistivity is $2.7 \times 10^{-6}\Omega$Cm.

Soln. 5-e) Energy stored on the capacitor is $1/2 \times 100 \times 10^{-12} \times 4 \times 10^6 = 200\mu$ J.
Resistance of the Aluminium line is given by

$$R_{Al} = 2.7 \times 10^{-6} \times \frac{100 \times 10^{-4}}{1 \times 0.5 \times 10^{-8}} = 5.4\Omega$$

Since the current through the external resistor and the aluminium line is the same, the energy will divide in the voltage ratio. Therefore, energy dissipated inside the chip is $200 \times 5.4/1505.4\ \mu\text{J} = 0.71742\mu$J.
Total volume of Aluminium being heated is $(70 \times 70 + 100 \times 1) \times 0.5 \times 10^{-12}$ cm^3 = 2500×10^{-12} cm^3
Therefore mass of aluminium = $2.7 \times 25 \times 10^{-10}$g = 67.5×10^{-10}g
Temperature rise is given by

$$\Delta T = \frac{0.71742 \times 10^{-6}}{67.5 \times 10^{-10} \times 0.9} = 118.1^o\text{C}$$

– **[3]**

– **[Q5: 3+2+1+1+3 = 10 marks]**

Paper Ends

Reference

You can use the following transistor model for all questions in this paper:

$$\begin{aligned}
\text{For } V_{GS} \leq V_T, \quad & I_{DS} = \quad 0 \\
\text{For } V_{GS} \geq V_T \text{ and } V_{DS} \leq V_{GS} - V_T, \quad & I_{DS} = \quad K\left((V_{GS} - V_T)V_{DS} - \frac{1}{2}V_{DS}^2\right) \\
\text{For } V_{GS} \geq V_T \text{ and } V_{DS} \geq V_{GS} - V_T, \quad & I_{DS} = \quad \frac{K}{2}\left(V_{GS} - V_T\right)^2
\end{aligned}$$

$$\text{Here } K = \mu C_{ox}\,\frac{W}{L}$$

The asymptotic optimum stage effort for a chain of logic gates is given by:

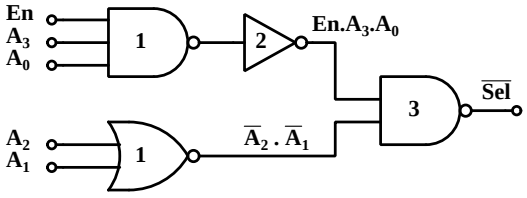
$$p_{inv} + \rho(1 - \ln \rho) = 0$$

Solution to Midsem question paper for EE 671: VLSI Design, Autumn Semester 2017

Q–1 A 4 bit decoder needs to be designed using 4 phase dynamic logic. Inputs to the decoder are an Enable signal En and a 4 bit address $A_3A_2A_1A_0$. These input signals are valid only in phase $\phi 1$. (Complemented address bits are not available and should be generated when required). Design a decoding circuit which will produce an output signal $\overline{\text{Sel}}$ which becomes '0' only when En is '1' and the address bits have the value '1001'. The output should be available in as early a phase as possible. You are allowed to use only NAND, NOR and Inverter circuits with a maximum of 3 logic inputs (not counting clock). Give the logic diagram and the 'type' of each gate which specifies in which phase it evaluates.
(Slow and unnecessarily complex circuits will get no credit).

Soln.: We need to decode $\overline{En \cdot A_3 \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0}$.
To do as much work as possible in phase $\phi 1$ itself, we can feed En, A_3 and A_0 to a 3 input NAND and A_2 and A_1 to a 2 input NOR gate. The output of the 3 input NAND is valid in phases $\phi 2$ and $\phi 3$. This output can be inverted in phase $\phi 2$, so that $En \cdot A_3 \cdot A_0$ is available in phases $\phi 3$ and $\phi 4$.

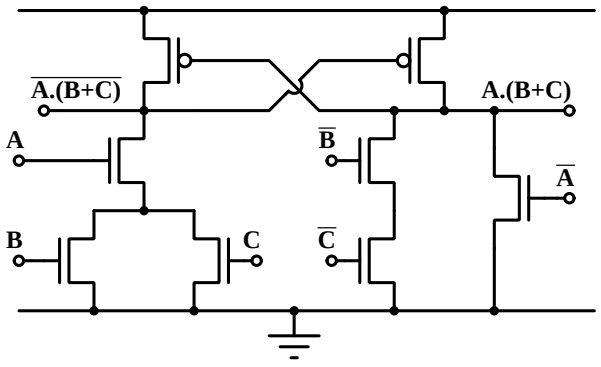
The output of the 2 input NOR gate is $\overline{A_2} \cdot \overline{A_1}$, valid in phases $\phi 2$ and $\phi 3$. Thus we have two intermediate products which are both valid in phase $\phi 3$. These can be combined with a 2-input NAND gate of type 3 to produce the desired $\overline{Sel}$ output.



– **[Q1: 2 marks]**

Q–2 Give the transistor level circuit diagram of a static Cascade Voltage Switch logic (CVSL) gate which implements the logic function $A.(B+C)$ and its complement, given A, B, C and their complements as inputs. How does this logic style avoid static current when the output is '0' while still needing to drive mostly n type transistors?

Soln.: The figure below shows the logic function $A.(B+C)$ implemented in CVSL logic style.



– **P.T.O.**

The nMOS network on the left is ON and pulls the left side output low when $A = 1$ AND either B or C or both are 1. Thus the left side output is $\overline{A.(B + C)}$ provided the pMOS load is 'ON' when $\overline{A.(B + C)}$ is '0'.

Similarly, the right side nMOS network is ON and will pull the right side output low when either $\overline{A} = 1$ or $\overline{B}$ as well as $\overline{C}$ are '1'. Thus the right side output is $\overline{\overline{A} + \overline{B} \cdot \overline{C}} = A.(B + C)$ provided the right side pMOS is on when $A.(B + C)$ is '1'

Whenever $A.(B + C) = 1$, the nMOS transistors on the left side pull the output LOW, turning the pMOS on the right ON as required. At the same time, the nMOS switch combination on the right is OFF and the right side output is '1', which turns the left side pMOS OFF.

Whenever $A.(B+C) = 0$, $\overline{A}+\overline{B}\cdot\overline{C} = 1$. This turns on the nMOS swich network on the right, pulling this output low. A low output on the right turns on the pMOS transistor on the left. At the same time, the nMOS switch combination on the left is OFF, and hence the left output is '1'.

Thre is no static power consumption in either state because the nMOS switch combinations and their pMOS loads are complementary. When $A.(B + C) = 1$, the left nMOS switch combination is ON, but its pMOS is OFF. At the same time, since $\overline{A}+\overline{B}\cdot\overline{C} = 0$, the nMOS combination on the right is OFF, while its pMOS is ON.

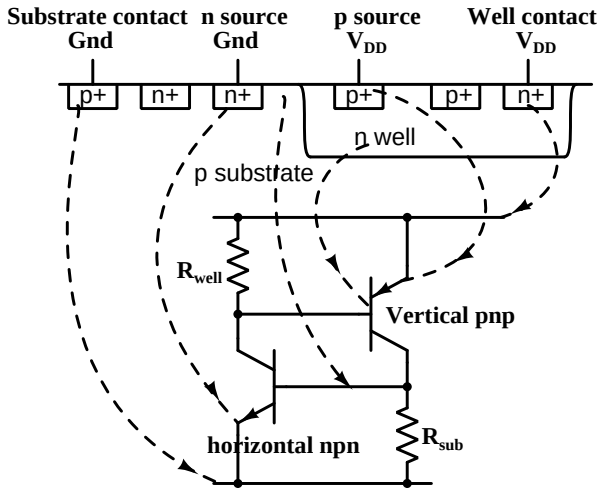
When $A.(B+C) = 0$, $\overline{A}+\overline{B}\cdot\overline{C} = 1$ and the right side nMOS switch combination is ON, but its pMOS is OFF. At the same time, since $A.(B + C) = 0$, the nMOS combination on the left is OFF, while its pMOS is ON.

In this way, even though we are driving only nMOS transistors, there is no static power consumption unlike the case for pseudo nMOS logic. – **[Q2: 2 marks]**

Q–3 Show how latch up occurs in CMOS circuits. Give a cross section diagram and the equivalent circuit, showing the correspondence between the regions in the cross section and the nodes of the equivalent circuit.

What are the suggested methods for avoiding latch up in the process (doping profile) and layout (design rules).

Soln.: The figure below shows a cross section of a CMOS circuit and the parasitic bipolar transistors which form the latchup structure.



The vertical pnp transistor is formed by the p+ source of a pMOS transistor connected to V_{DD} (which becomes the emitter), the n well (which becomes the base) and the p substrate (which becomes the collector of this transistor). The n well is connected to V_{DD} through a resistive path, which represents the resistance of the n well to the well contact.

The horizontal npn transistor is formed by the n+ source of an nMOS transistor connected to ground (which becomes the emitter), the p substrate, (which becomes the base) and the n well, (which becomes the collector).

Since the collector of the npn and the base of the pnp are both formed by the n well, these two are connected. Similarly, the collector of the pnp and the base of the npn are formed by the p substrate, so these are connected too.

Looking at the equivalent circuit, one can see that it forms a positive feedback system. An increase in the base current of the pnp will be amplified by its β_p and a large part of it will flow through the base emitter junction of the npn transistor. This part will be amplified by the β_n of the npn and a substantial part of it will go through the base emitter junction of the pnp. If the product of the two amplification factors β_p and β_n and the current division ratios between the resistors and the base emitter junctions exceeds 1, the currents will keep increasing due to this feedback, till there is a dead short between V_{DD} and ground. This is called latch up.

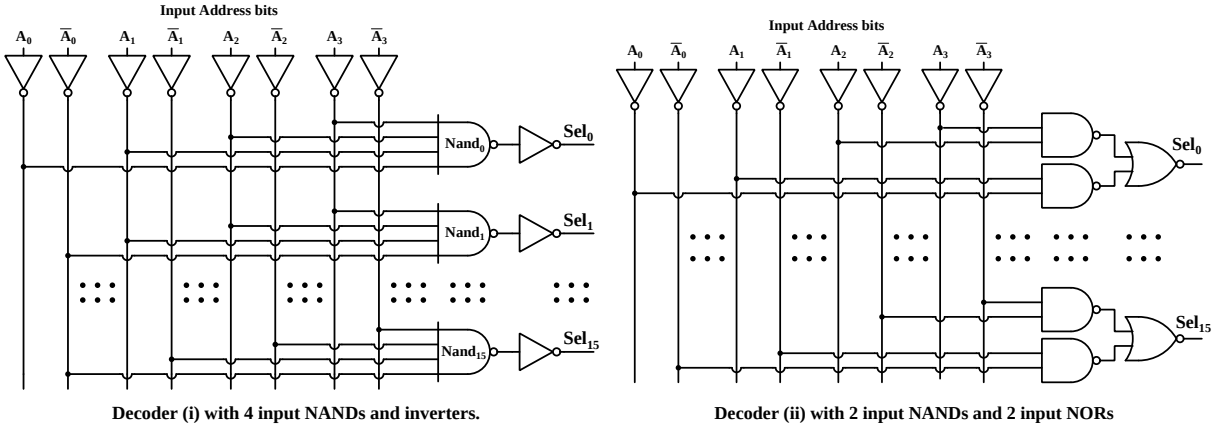
To prevent latch up, we must reduce the β of the parasitic bipolar transistors and make sure that most of the collector current of either transistor is directed to the resistor and not to the base-emitter junction of the other transistor. This can be done through process steps as well as through design rules.

1. The doping gradient of the n well should be made retrograde. (Doping should increase as we go deeper). This kills the current gain β_p of the pnp transistor.
2. The n well should have a guard ring connected to V_{DD}, which will collect any current which could form the base current of the pnp.
3. In layout, substrate and well contacts should be placed frequently, to reduce the value of R_{well} and $R_{substrate}$.
4. n channel transistors should be placed far from the edge of the n well. This increase the base width of the npn transistor and kills its current gain.
5. p channel transistors should also be placed far from the well edge and the p well should be deep to kill the gain of the npn transistor.

– **[Q3: 3 marks]**

Q–4 For a given CMOS process, the mobility correction factor γ for PMOS transistor widths is 2.5. The parasitic delay of gates may be taken to be proportional to the sum of the widths of transistors directly connected to the output terminal in a minimum sized gate. The parasitic delay of an inverter (p_{inv}) is 2 in units of τ, the propagation delay of a minimum sized inverter driving another minimum sized inverter without including the parasitic delay.

We want to compare two circuits to implement a 4 to 16 decoder. In circuit (i), appropriate combinations of address bits and their complements are given to 4-input-NAND gates, and their outputs are connected to inverters. In circuit (ii), combinations of address bits and their complements go to 2-input-NAND gates and their outputs are combined pair wise by 2-input-NOR gates to generate the select outputs as shown.



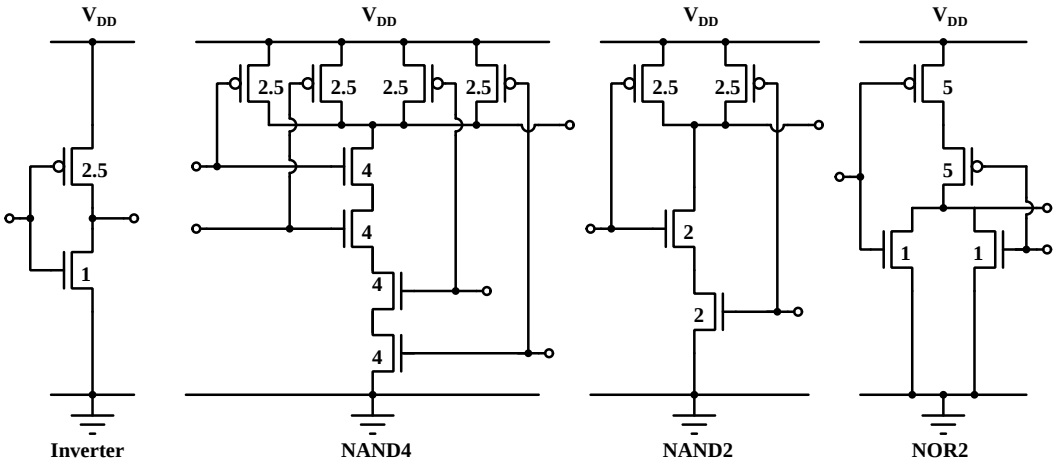
Decoder (i) with 4 input NANDs and inverters.

Decoder (ii) with 2 input NANDs and 2 input NORs

In both circuits, the inverters at the input are minimum sized and each select output is loaded with capacitance equivalent to 128 minimum sized inverters. All transistors use minimum channel length.

a) Compute the logical effort and parasitic delay for all the types of gates involved in the above circuits.

Soln.: The figure below shows the gates with transistor widths to be used for providing the same output drive as a minimum inverter.



In case of 4 input NAND, there are 4 n channel transistors in series. So each must be sized to 4 times the width of the n channel transistor used in the minimum inverter. 4 p channel transistors are in parallel, so each has the same size as the p

channel transistor in the minimum inverter – that is, 2.5 times the width of the n channel transistor in the minimum inverter.

Similarly, the 2 input NAND has two n channel transistors in series with a size of 2 and two p channel transistor with a size of 2.5 in parallel. The 2 input NOR has two n channel transistors in parallel, so each is the same size as the n channel transistor in the minimum inverter. The p channel transistors are in series, so each must be sized to 5.

The logical effort of an inverter is 1 by definition and its parasitic delay has been given to be 2. The logical effort for a gate is proportional to the input capacitance (and hence, total transistor width) connected to a given input. Therefore

$$g = \frac{1}{3.5} \times \text{sum of transistor widths connected to the input.}$$

The parasitic delay can be estimated to be proportional to the total transistor width connected to the output terminal. Therefore,

$$p = \frac{2}{3.5} \times \text{sum of transistor widths connected to the output}$$

This gives:

Gate	Width of n Trans.	Width of p Trans.	Total W at input	Total W at output	Logical Effort	Parasitic Delay
Inverter	1	2.5	3.5	3.5	1	2
4 input NAND	4	2.5	6.5	14	13/7	8
2 input NAND	2	2.5	4.5	7	9/7	4
2 input NOR	1	5	6	7	12/7	4

– **[1]**

b) Find the widths for n and p channel transistors in all the gates of both circuits to minimize the total delay. (Specify the widths in units of the width of the n channel transistor in a minimum inverter).

Soln.: Circuit (i) with NAND4 gates

The output of each input inverter goes to 8 NAND inputs. (There are 8 inverter outputs and 64 NAND inputs and these are equally divided). the logical effort g for 4-input-NAND gates is 13/7. Therefor the path effort for this circuit is given by

$$F = GBH = (1 \times \frac{13}{7} \times 1) \times 8 \times \frac{128}{1} = 1901.714$$

Since the circuit has 3 stages, the optimum stage effort is

$$\hat{f} = 1901.714^{1/3} = 12.38935$$

For the final inverters,

$$gbh = 1 \times 1 \times \frac{128}{C_{in}} = 12.38935, \quad \text{which gives} \quad C_{in} = 10.33146$$

For NAND gates with 4 inputs,

$$gbh = \frac{13}{7} \times 1 \times 10.33146/C_{in} = 12.38935, \quad \text{which gives} \quad C_{in} = 1.548668$$

For the input inverters,

$$gbh = 1 \times 8 \times 1.548668/C_{in} = 12.38935, \quad \text{which gives} \quad C_{in} = 1 \quad \text{as expected}$$

The final inverter is 10.33146 times the size of the minimum inverter. Therefore the n channel transistor width is 10.33146, while the p channel transistor width is $10.33146 \times 2.5 = 25.829$.

The input capacitance is in units of input capacitance of a minimum inverter. Thus, each capacitance unit represents 3.5 units of transistor width. The total transistor width at each input of the 4-input-NAND should be $3.5 \times 1.548668 = 5.420339$ width units. This width is divided in the ratio of $4 : 2.5$ between the n and p channel transistors. Therefore the n channel transistor width is $5.420339 \times 4/6.5 = 3.3356$ and the p channel transistor width is $5.420339 \times 2.5/6.5 = 2.0847$.

The input inverter is of course unit sized, and therefor n and p channel transistor widths are 1 and 2.5 respectively.

Circuit (ii) with NAND2 and NOR2 gates

The output of each input inverter again goes to 8 NAND inputs. (There are a total of 64 AND gate inputs fed by 8 inverter outputs and these are divided equally). Therefore the branch factor for the input inverters is 8 again.

The logical effort for 2-input-NAND gates is 9/7, while that for 2-input-NOR Gates is 12/7. Therefor the path effort for this circuit is given by

$$F = GBH = (1 \times \frac{9}{7} \times \frac{12}{7}) \times 8 \times \frac{128}{1} = 2256.98$$

Since the circuit has 3 stages, the optimum stage effort is

$$\hat{f} = 2256.98^{1/3} = 13.11724$$

For the last stage with NOR gates,

$$gbh = \frac{12}{7} \times 1 \times \frac{128}{C_{in}} = 13.11724 \quad \text{which gives} \quad C_{in} = 16.72825$$

For 2-input-NAND gates

$$gbh = \frac{9}{7} \times 1 \times 16.72825/C_{in} = 13.11724 \quad \text{which gives} \quad C_{in} = 1.639655$$

For the input inverters,

$$gbh = 1 \times 8 \times 1.639655/C_{in} = 13.11724 \quad \text{which gives} \quad C_{in} = 1 \quad \text{as expected}$$

Again, each capacitance unit represents 3.5 units of transistor width. Therefore the total input transistor width for NOR gates is $16.72825 \times 3.5 = 58.54889$. This is divided in the ratio $1 : 5$ between n and p channel transistors. Therefore, n channel transistor width is $58.54889/6 = 9.758149$ and the p channel transistor width is $58.54889 \times 5/6 = 48.79074$.

The total input transistor width for 2-input-NAND gates is $1.639655 \times 3.5 = 5.738794$. This is divided in the ratio $2 : 2.5$ between the n and p channel transistors. Therefore the n channel transistor width is $5.738794 \times 2/4.5 = 2.550574$ and the p channel transistor width is $5.738794 \times 2.5/4.5 = 3.188219$.

The input inverters are of course unit sized and so the n and p channel transistors have widths of 1 and 2.5 respectively.

The transistor sizes for all gates may be summarized as:

Circuit	gate	n width	p width
With NAND4	Input Inv.	1	2.5
	NAND 4	3.33	2.08
	Final Inv.	10.33	25.83
With NAND-NOR	Input Inv.	1	2.5
	NAND2	2.55	3.19
	NOR2	9.76	48.79

– [8]

c) Compute the total delay in units of τ for both circuits.

Soln.: In case of the first circuit, $\hat{f} = 12.39$. Therefore

$$D_{total} = 3\hat{f} + p_{inv} + p_{\text{NAND4}} + p_{inv} = 3 \times 12.39 + 2 + 8 + 2 = 49.17$$

For the second circuit, $\hat{f} = 13.12$. Therefore,

$$D_{total} = 3\hat{f} + p_{inv} + p_{\text{NAND2}} + p_{\text{NOR2}} = 3 \times 13.12 + 2 + 4 + 4 = 49.36$$

So the total delay is about the same in the two cases. – **[2]**

d) The optimum stage ratio ρ is a solution to the equation $\rho(1 - \ln \rho) + p_{inv} = 0$. Find the value of ρ and the optimum logic depth for the two decoders for the specified loading. What is the total delay for the two circuits if the logic depth is made optimum by adding inverters?

Soln.: p_{inv} is given to be 2. So the equation to be solved is:

$$f \equiv \rho - \rho \ln \rho + 2 = 0$$

We have

$$f' = 1 - \ln \rho - \rho \frac{1}{\rho} = -\ln \rho$$

Therefore given a guess g, the next improved guess is

$$g - \frac{f(g)}{f'(g)} = g + \frac{g - g \ln g + 2}{\ln g} = \frac{g + 2}{\ln g}$$

Starting with a guess value of 4, the successive guesses for ρ are: 4.328085, 4.319143, 4.319137, 4.319137 ....

Therefore the optimum number of stages is $\ln F / \ln \rho$.

For the circuit with NAND4, this is

$$\frac{\ln 1901.714}{\ln 4.319137} = 5.16$$

The optimum number of stages is 5, though one should also evaluate the delay for 6 stages to see which is better. The stage effort for a 5 stage design is $1901.714^{1/5} = 4.527193$. This design will add two additional inverters to the existing 2 inverters. So the parasitic delay will be $4p_{inv} + p_{\text{NAND4}} = 8 + 8 = 16$. Thus the total delay is $5 \times 4.527193 + 16 = 38.64$.

In case of a 6 stage design, 3 additional inverters will be inserted, so the parasitic delay will be $5p_{inv} + p_{\text{NAND4}} = 10 + 8 = 18$. The stage effort is $1901.714^{1/6} = 3.51985$ and so the total delay is $6 \times 3.51985 + 18 = 39.12$. Thus a 5 stage delay is optimum, with a total delay of 38.64.

For the circuits with NAND2 and NOR2, the optimum number of stages is

$$\frac{\ln 2256.98}{\ln 4.319137} = 5.28$$

Again, 5 stages should be optimum, though one should also evaluate the delay for 6 stages to see which is better. For a 5 stage design, two additional inverters will be inserted, so the parasitic delay will be $3 \times p_{inv} + p_{\text{NAND2}} + p_{\text{NOR2}} = 6 + 4 + 4 = 14$. The stage effort in this case is $2256.98^{1/5} = 4.684956$ and so the total delay is $5 \times 4.684956 + 14 = 37.42$.

For a six stage design, three inverters will be inserted, so the parasitic delay will be $4 \times p_{inv} + p_{\text{NAND2}} + p_{\text{NOR2}} = 8 + 4 + 4 = 16$. The stage effort is $2256.98^{1/6} = 3.621773$, so the total delay will be $6 \times 3.621773 + 16 = 37.73$.
In this case, the delay is about the same for a 5 stage or 6 stage design. However 5 stage design will be optimum because it has lower complexity (and marginally lower delay). – **[2]**

– **[Q3: 1 + 8 + 2 + 2 = 13 marks]**

Paper Ends

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

End Semester Examination

Tuesday	**EE 671: VLSI Design**	Time: 1130-1230
Oct 24, 2017	Autumn Semester 2017	Marks: 10

Q–1 **a)** Explain how the modified Booth algorithm reduces the number of partial products in an unsigned multiplier by looking at groups of 3 bits of the multiplier at a time with one overlapping bit. Give a table of operations to be performed corresponding to all combinations of the 3 bits.

Soln. 1-a) The booth algorithm multiplies the multiplicand with two bits of the multiplier at a time. If the multiplicand is A, the partial product on multiplication by a 2 bit number can be 0, A, 2A or 3A. While 0, A and 2 A can be generated easily from A, 3A can not be generated directly. Instead, we generate -A and increment the multiplier of the next group of 2 bits in the multiplier by 1. This is equivalent to subtracting A and adding 4A.

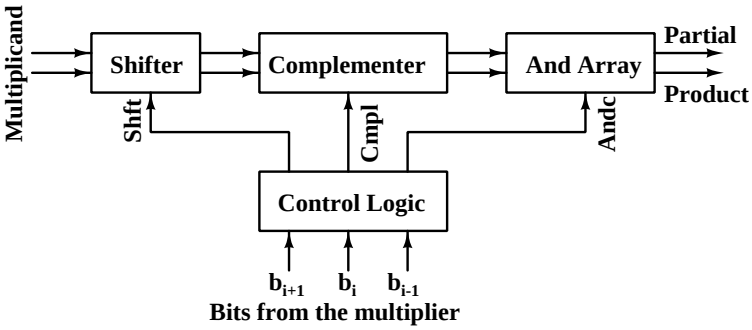
In modified Booth encoding, we handle multiplication by 2 also in the same way – subtracting 2A and adding 4A when handling the next group of 2 bits. This has the advantage that one needs to examine only the more significant bit of the previous two bit group to decide whether the current multiplier should be incremented by 1 or not. Since incrementing is to be done whenever the previous group was 2 or 3, the condition simplifies to the more significant bit of the previous group being '1'.

Thus the generation of partial products for eventual addition depends on the two multiplier bits and the more significant bit of the previous group of 2 bits. The partial product to be generated can be summarized by the following table:

Current 2 bits	MSb of Prev. 2 bits	Effective Multiplier	Partial Product
00	0	0+0	0
01	0	1+0	A
10	0	2+0	-2A
11	0	3+0	-A
00	1	0+1	A
01	1	1+1	2A
10	1	2+1	-A
11	1	3+1	0

– **[3]**

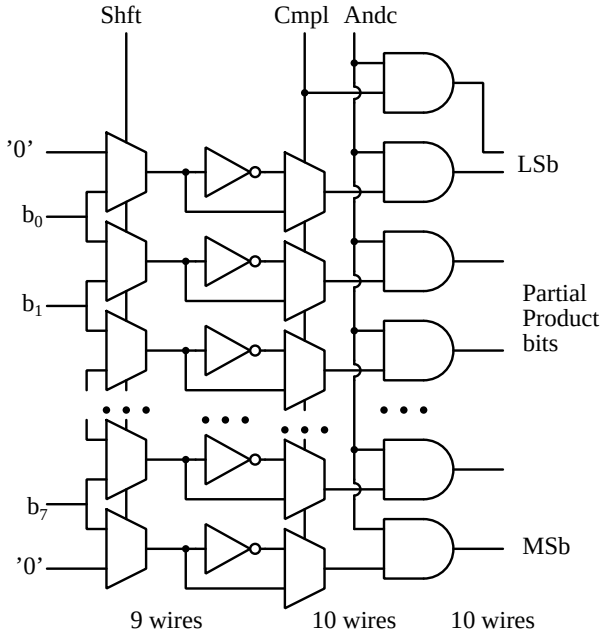
b) We want to implement a Booth partial product generator using a 'Shifter', a 'Complementer' and an 'And Array' as shown in the figure below:



The Shifter shifts the multiplicand left by 1 bit if the control input 'Shft' = 1, otherwise it copies its input (with an appended 0 as msb) to the output. The Complementer inverts its input bits if the control input 'Cmpl' =1, otherwise it copies its input to the output. (It will also provide the 'Cmpl' input as a Carry input data bit to implement a 2's complement). The And Array ands all its input bits (inclusive of Carry in) with the control input 'Andc' to produce the partial product.

i) Show a schematic for the Shifter, Complementer and And Array, (using multiplexers and basic gates) for the above circuit in an 8×8 multiplier. The number of inputs/outputs for each block should be explicitly shown.

Soln. 1-b i) The figure below shows the three arrays and their interconnection.

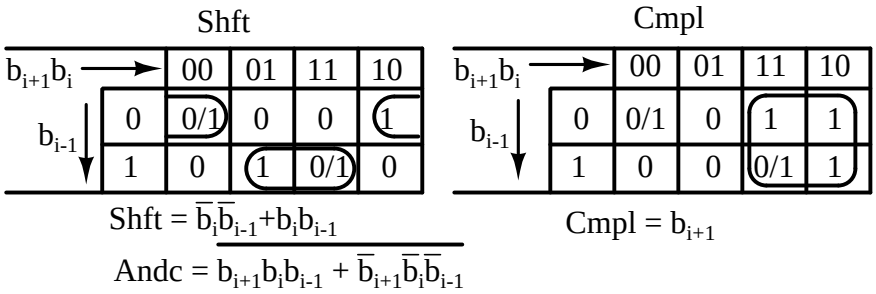


The shifter produces 9 output wires, which can be "$b_7 b_6 \ldots b_0 0$" or "$0 b_7 b_6 \ldots b_0$". Since the shifter effectively multiplies the input by 2 (if Shft = '1'), the most significant bit of the output has one position higher weight than the input.
The Complementer effectively changes the sign of its input when the control input Cmpl=1. It does so by generating a 1's complement first and providing an extra wire at the least significant bit position (to add 1 to the 1's complement). The complementer circuit chooses between the 9 bits at its input or their complements and adds the control input Cmpl as an output at the least significant bit position. Thus, it outputs two wires for the least significant bit and one wire each for for the other 8 place values. This means that there are a total of 10 wires output by the Complementer. The And array just ands all

these 10 wires with the control input AndC, to produce 10 wires as the partial product. Two of the wires have the same weight (lease significant bit). There is a wire each for each higher weight, ending with the most significant bit which has a weight 1 position higher than the input. – **[3]**

ii) Derive the logic expressions to generate the control signals 'Shft', 'Cmpl' and 'Andc' from the three multiplier bits. (b_{i-1} is the overlap bit.) You must use the fact that 'Shft' and 'Cmpl' can be '0' or '1' (Don't Care) when 'Andc' is '0' to simplify the logic expressions.

Soln. **1**-b ii) Based on the table for Booth algorithm, we can set up the following Karnaugh maps for the control inputs to the Shifter, Complementer and And Array:



The Karnaugh maps use the fact that when all three bits are 0 or 1, Andc will be 0 and therefore the Shft and Cmpl signals can be either 0 or 1. Using this, the expressions for the control inputs are:

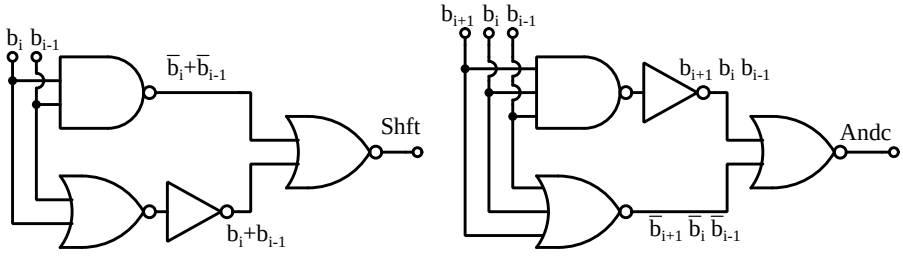
$$\text{Shft} = \overline{b_i} \cdot \overline{b_{i-1}} + b_i \cdot b_{i-1}, \qquad \text{Cmpl} = b_{i+1}$$

$$\text{and} \qquad \text{Andc} = \overline{b_{i+1} \cdot b_i \cdot b_{i-1} + \overline{b_{i+1}} \cdot \overline{b_i} \cdot \overline{b_{i-1}}}$$

– **[3]**

iii) Suggest a gate level implementation to generate the three control inputs. Only Nand and Nor gates (with no more than three inputs) and inverters should be used and the critical path for the control signal generation should be no more than 3 gates deep.

Soln. **1**-b iii) Cmpl requires no logic and b_{i+1} can be used directly for this control line. The other two control signals can be generated as shown in the logic diagram below:



– **[2]**

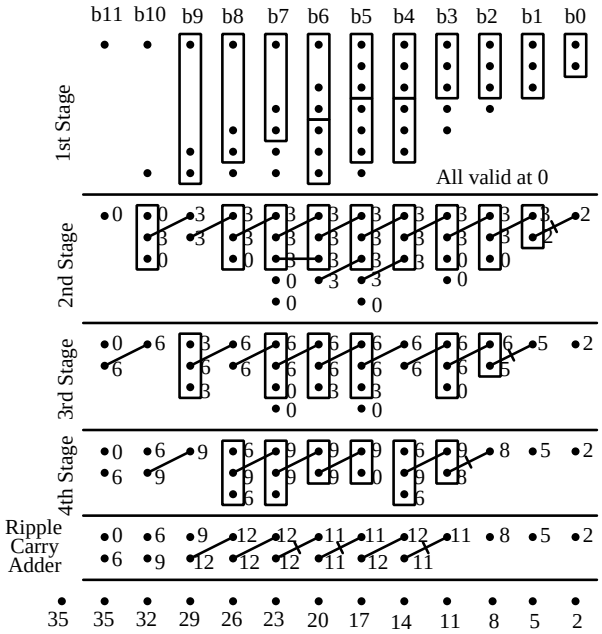
Q–2 We want to design an unsigned multiply and accumulate circuit (MAC) in which two 6 bit operands are to be multiplied and a 12 bit number added to the product to generate a 13 bit result. Partial products are generated by a matrix of 6×6 AND gates and are assumed to be available in parallel at t=0. These partial products are to be reduced to two for each weight using a Wallace tree using half and full adders. Assume that a half

adder takes 2 units of time while a full adder takes 3 units of time after the arrival of the latest input to produce their sum and carry outputs.

a) Show the Wallace tree reduction scheme for this circuit using a dot diagram. Clearly state the policy for using half adders such that no redundant bits will be generated in the result. Mark the time of arrival of each bit in brackets next to each dot.

Soln. 2-a) The policy for Wallace tree reduction scheme may be stated as follows:
Combine all wires for any particular weight. As long as there are 3 wires available at any weight, reduce these using a full adder. Finally, less than 3 wires will be left.
If no wire is left, there is nothing else to do at this weight.
If one wire is left, pass it through to next reduction stage.
If there are two wires left, upto the column with maximum number of wires: if there is no carry coming in from the less significant bit column in the next stage, reduce the wires using a half adder. Otherwise pass the two wires through to the next stage (hoping for reduction using a full adder then), unless passing the wire through will exceed the 'capacity' of the level. The capacity of the level is 2 at the last stage and floor(1.5*n) for other stages where n is the capacity of the next stage. Thus the capacities will be 2,3,4,6,9 ... counting from the last stage of reduction.

In the MAC, the adder contributes one extra wire at each weight from b0 to b11. Partial products provide 1,2,3,4,5,6,5,4,3,2 and 1 wires starting from the least significant bit (b0) up to the 11th bit (b10). Using this rule, a dot diagram for wire reduction can be shown as below:



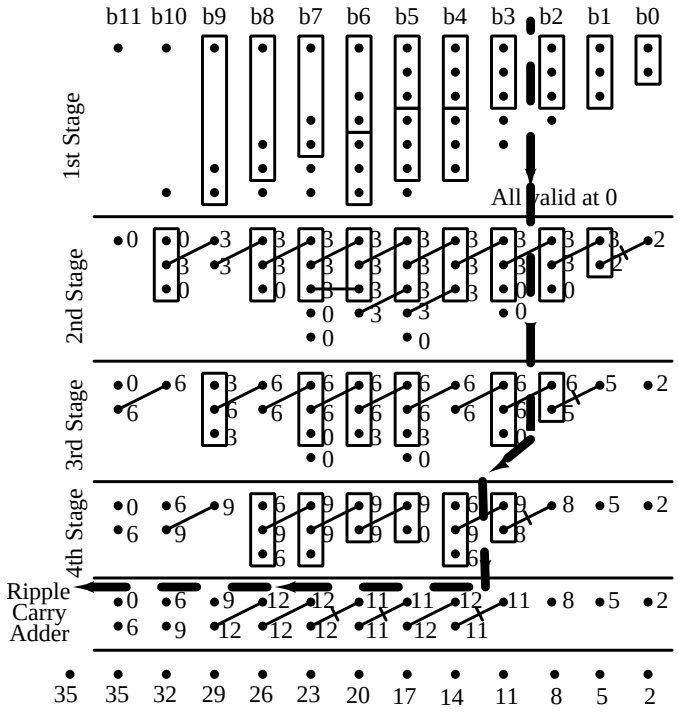
Assuming that a half adder takes 2 units of time while a full adder takes 3 units of time, the latest arrival time of bits at different stages is shown in the figure next to the corresponding dots. – **[4]**

b) Assume that a ripple carry adder will be used for the final addition. How many full and half adders will be used for implementing this circuit (inclusive of the final ripple carry adder)?

Soln. 2-b) The first reduction stage uses 12 full adders and 1 half adder. The second stage requires 8 full adders and 1 half adder. The third stage uses 5 full adders and a half adder. The last reduction stages uses 3 full adders and 3 half adders. Thus the reduction stages use 28 full adders and 6 half adders. The reduction stage results in single wire outputs for the 4 least significant bits. Therefore, the ripple carry adder is 8 bits wide, needing one half adder and 7 full adders. Thus, in all, we need 35 full adders and 7 half adders. – **[2]**

c) Assume that a half adder takes 2 units of time and a full adder takes 3 units of time to generate their sum and carry outputs. What is the critical path for this circuit? How many units of time are required after the generation of partial products for the final result to be ready?

Soln. 2-c) The figure above gives the arrival times for various bits. Notice that after the ripple carry addition, the last two bits are the sum and carry outputs of the full adder at the most significant bit full adder. Therefore these two will arrive at the same time. So the final result will be ready after 35 units of time. (Notice that additions for the reduction stage are all over at 12 units of time and the ripple carry adder accounts for the remaining 23 units of delay. Using a ripple carry adder is not such a good choice even for this 13 bit unit!).



If we start from the last full adder in the ripple carry adder, the b12 and b11 of the result arrive at 35 units of time because the rippling carry from the b10 arrives only at 32 units of time. (The other inputs to this adder are ready at 0 and 6 units of time). Similarly, the arrival time of 32 units at b10 is caused by the rippling carry being available only at 29 units of time from b9. We can continue this argument similarly, till we reach b4. This bit is ready at 14 units since the latest input to the half adder at this position arrives at 12 units of time.

The latest input to the half adder arrives at 12 units of time because the latest input to the full adder which produces it gets its latest input at 9 units of time. This input is produced by the full adder at b3 in the third stage of reduction, whose

latest inputs arrive at 6 units of time. These two inputs come from the sum output of the full adder at b3 and the carry output of the full adder at b2 in the 2nd stage. Both these adders receive their latest input at 3 units of time due to full adders at b1, b2 and b3.

Thus the critical path(s) are:
Full adders at bits b3, b2, b1 in the first stage to full adders at b3 and b2 in the second stage, to full adder at b3 in the 3rd stage, to the full adder at b4 in the 4th stage to the half adder in the ripple carry adder and then through the ripple carry adder to the most significant bit of the output as shown in the diagram. – **[3]**

Paper Ends

Credit for Q1 and Q2 adds to 20 marks. Total marks will be scaled to 10 for final grading.

Reference

In a dot diagram for a multiplier, all wires of a particular weight are shown as dots lined up in a column. Thus the sum and carry outputs from an adder are shown as dots in adjacent columns, connected by a line. This line is crossed if these are from a half adder and the line is not crossed if these are the outputs from a full adder.

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

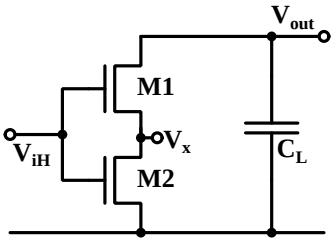
Mid Semester Examination

Saturday	EE 671: VLSI Design	Time: 1330-1530
15-09-18	Autumn Semester 2018	Marks: 25

For iterative solutions, all intermediate values must be reported.
Quantitative answers must be accurate at least to 0.1%.

Q–1 Consider a 2 input CMOS NAND gate acting as an inverter with both its inputs tied to a logic 'High' value. Since the inputs are 'High', the pMOS transistors are OFF and may be ignored for this problem. The two n channel transistors M1 and M2 have identical geometries and electrical parameters. We wish to analyse this circuit without approximating the behaviour of the transistors as equivalent resistors.

Assume that the load capacitor C_L is initially charged to $V_{DD} =$ 3.3V. Both inputs are tied to $V_{iH} = 3.0$V. Assume $V_{Tn} = 0.6$V. Dependence of V_{Tn} on the source voltage (bulk effect) is to be ignored. As the output discharges, V_{out} goes from V_{DD} towards 0 V. Use the simple MOS model with perfect saturation for MOS transistor currents.



a) In what modes (saturated or linear) are the two transistors for different values of V_{out} as it drops from V_{DD} to 0?

Soln. Q1a: Condition for saturation is $V_{ds} \geq V_{gs} - V_{Tn}$. For M1, this translates to $V_{out} - V_x \geq V_{iH} - V_x - V_{Tn}$. Canceling V_x from both sides, the condition becomes $V_{out} \geq V_{iH} - V_{Tn}$. Thus M1 is in saturation for $V_{out} \geq V_{iH} - V_{Tn}$ and is in linear mode for $V_{out} \leq V_{iH} - V_{Tn}$. The single transistor replacing M1 and M2 will also be in saturation for $V_{out} \geq V_{iH} - V_{Tn}$ and in linear mode for $V_{out} \leq V_{iH} - V_{Tn}$.

Since the input is 'High', both transistors are conducting.
Therefore V_{gs} for M1 is $> V_{Tn}$. This implies that the drain of M2 has lower voltage compared to its gate by $> V_{Tn}$. Therefore M2 is always in linear mode. – **[1]**

b) Derive expressions for the voltage at the source of M1 (V_x) in terms of V_{iH}, V_{Tn} and V_{out} for different combinations of modes of M1 and M2 which will occur during the discharge. Taking $V_{iH} = 3.0$V and $V_{Tn} = 0.6$V, tabulate values of V_x for:
$V_{out} =$ i) 3.3 V, ii) 3.0 V, iii) 2.4 V, iv) 1.8 V and v) 1.2 V.

Soln. Q1b: For $V_{out} \geq V_{iH} - V_{Tn}$, M1 is in saturation, while M2 is in linear mode. Since the two are in series, their currents must be equal. Therefore,

$$\frac{K_n}{2}(V_{iH} - V_x - V_{Tn})^2 = K_n\left((V_{iH} - V_{Tn})V_x - \frac{1}{2}V_x^2\right)$$

$$\text{This gives} \quad (V_{iH} - V_x - V_{Tn})^2 = 2\left((V_{iH} - V_{Tn})V_x - \frac{1}{2}V_x^2\right)$$

$$\text{Defining } V_1 \equiv V_{iH} - V_{Tn}, \text{ we get} \quad (V_1 - V_x)^2 = 2V_1V_x - V_x^2$$

$$\text{Therefore} \quad V_1^2 + V_x^2 - 2V_1V_x = 2V_1V_x - V_x^2 \quad \text{So,} \quad 2V_x^2 - 4V_1V_x + V_1^2 = 0$$

$$\text{This can be solved to give} \quad V_x = \frac{4V_1 \pm \sqrt{16V_1^2 - 8V_1^2}}{4} = V_1 \pm \sqrt{\frac{V_1^2}{2}}$$

Since V_x must be $< V_{iH} - V_{Tn}$, the negative sign should be chosen.

$$\text{So} \quad V_x = V_1\left(1 - \frac{1}{\sqrt{2}}\right) = \left(1 - \frac{1}{\sqrt{2}}\right)(V_{iH} - V_{Tn})$$

Thus the voltage V_x remains constant at $(1-1/\sqrt{2})(V_{iH} - V_{Tn})$ till V_{out} drops below $V_{iH} - V_{Tn}$.

For $V_{out} \leq V_{iH} - V_{Tn}$, both M1 and M2 are in linear mode. Since these are in series, their currents must be equal. This gives

$$K_n\left((V_{iH} - V_x - V_{Tn})(V_{out} - V_x) - \frac{1}{2}(V_{out} - V_x)^2\right) = K_n\left((V_{iH} - V_{Tn})V_x - \frac{1}{2}V_x^2\right)$$

$$\text{So} \quad (V_1 - V_x)(V_{out} - V_x) - \frac{1}{2}(V_{out} - V_x)^2 = V_1V_x - \frac{1}{2}V_x^2$$

$$\text{Or} \quad V_1V_{out} - V_xV_{out} - V_1V_x + V_x^2 - \frac{1}{2}(V_{out}^2 + V_x^2 - 2V_{out}V_x) = V_1V_x - \frac{1}{2}V_x^2$$

$$\text{This leads to} \quad V_x^2 - 2V_1V_x + V_1V_{out} - \frac{1}{2}V_{out}^2 = 0$$

We can solve this quadratic equation to give

$$V_x = \frac{2V_1 \pm \sqrt{4V_1^2 - 4(V_1V_{out} - \frac{1}{2}V_{out}^2)}}{2} = V_1 \pm \sqrt{V_1^2 - V_1V_{out} + \frac{1}{2}V_{out}^2)}$$

Again, since $V_x < V_{iH} - V_{Tn}$, the negative sign must be chosen.

$$\text{Then} \quad V_x = V_1 - \sqrt{V_1^2 - V_1V_{out} + \frac{1}{2}V_{out}^2}$$

$$\text{Or} \quad V_x = V_1 - \sqrt{V_1(V_1 - V_{out}) + \frac{1}{2}V_{out}^2}$$

It is interesting to evaluate this at $V_{out} = V_1 = V_{iH} - V_{Tn}$ when M1 is at the edge of saturation. We get

$$V_x = V_1\left(1 - \frac{1}{\sqrt{2}}\right)$$

This matches with the value we obtained in the saturation case, as indeed it should. In our case, $V_{iH} = 3.0$V, $V_{Tn} = 0.6$V, so $V_1 = 2.4$V. We can now compute and tabulate the values of V_x for all the given values of V_{out}, noticing that for $V_{out} \geq 2.4$V, M1 is saturated and V_x is constant at $(1 - 1/\sqrt{2})V_1 = 0.703$V.

V_{out}	3.3	3.0	2.4	1.8	1.2
V_x	0.703	0.703	0.703	0.651	0.503

– **[4]**

c) How much is the discharge current through the series connected transistors M1 and M2, when expressed as a fraction of the discharge current through a single nMOS transistor with identical dimensions replacing the series connected transistors M1 and M2, with the same input voltage V_{iH} applied to its gate? Evaluate this ratio for all combinations of operating modes of M1 and M2 which occur during discharge.

Soln. Q1c:

$$\text{When M1 is saturated,} \quad V_x = V_1\left(1 - \frac{1}{\sqrt{2}}\right)$$

The discharge current is the same as the current through M1, so

$$I_{ds} = \frac{K_n}{2}(V_1 - V_x)^2 = \frac{K_n}{2}\frac{V_1^2}{2}$$

The current through a single transistor would have been $\frac{K_n}{2}V_1^2$. So the current through M1 and M2 is half as much as that through a single transistor with identical geometry.

$$\text{When M1 is linear,} \quad V_x = V_1 - \sqrt{V_1(V_1 - V_{out}) + \frac{1}{2}V_{out}^2}$$

The discharge current through M1 and M2 is the same as the current through M2, given by $K_n(V_1V_x - 1/2V_x^2)$. The current through a single transistor would have been $K_n(V_1V_{out} - 1/2\,V_{out}^2)$. The quadratic equation for V_x was

$$V_x^2 - 2V_1V_x + V_1V_{out} - \frac{1}{2}V_{out}^2 = 0$$

$$\text{Therefore} \quad V_1V_x - \frac{1}{2}V_x^2 = \frac{1}{2}\left(V_1V_{out} - \frac{1}{2}V_{out}^2\right)$$

So, current through M1 and M2 is

$$I = K_n(V_1V_x - \frac{1}{2}V_x^2) = \frac{K_n}{2}\left(V_1V_{out} - \frac{1}{2}V_{out}^2\right)$$

Thus the current through the series connected transistors M1 and M2 is half of what a single transistor with identical geometry replacing them would have been.

This can be shown in another way: Let I be the current through the series connected transistors. Since the currents through the two transistors must be equal, the sum of these currents must equal $2I$.

$$2I = K_n\left((V_1 - V_x)(V_{out} - V_x) - \frac{1}{2}(V_{out} - V_x)^2\right) + K_n\left((V_1V_x - \frac{1}{2}V_x^2\right)$$

$$\text{Therefore} \quad \frac{2I}{K_n} = (V_1 - V_x)(V_{out} - V_x) - \frac{1}{2}(V_{out} - V_x)^2 + V_1V_x - \frac{1}{2}V_x^2$$

$$\text{So} \quad \frac{2I}{K_n} = V_1V_{out} - V_xV_{out} - V_xV_1 + V_x^2 - \frac{1}{2}(V_{out}^2 + V_x^2 - 2V_{out}V_x) + V_1V_x - \frac{1}{2}V_x^2$$

All terms involving V_x cancel and we are left with

$$\frac{2I}{K_n} = V_1V_{out} - \frac{1}{2}V_{out}^2$$

$$\text{Hence} \quad 2I = K_n\left(V_1V_{out} - \frac{1}{2}V_{out}^2\right)$$

The expression on the right is just the current through a single transistor with identical K_n and V_{Tn} and with V_{iH} applied to its gate.

Thus *for all output voltages*, the discharge current provided by M1 in series with M2 is half the current which would have been provided by a single transistor of identical geometry replacing them. – **[3]**

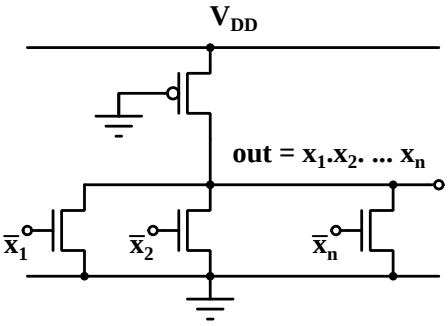
– **[Q1: 1+4+3=8 marks]**

Q–2 In a programmable logic array using pseudo NMOS style logic, we generate programmable products in one array and then add programmably selected products in the other.

a) NAND functions need to size the nMOS transistors depending on the number of inputs. This presents problems with programmability of the product array. How is this problem solved in PLAs?

Soln. Q2a: The number of inputs to be included in a product is variable in programmable logic. If we implement the product by using NAND gates, we shall have the problem that the nMOS transistor will have to be sized depending on the number of terms in the product – which is variable.

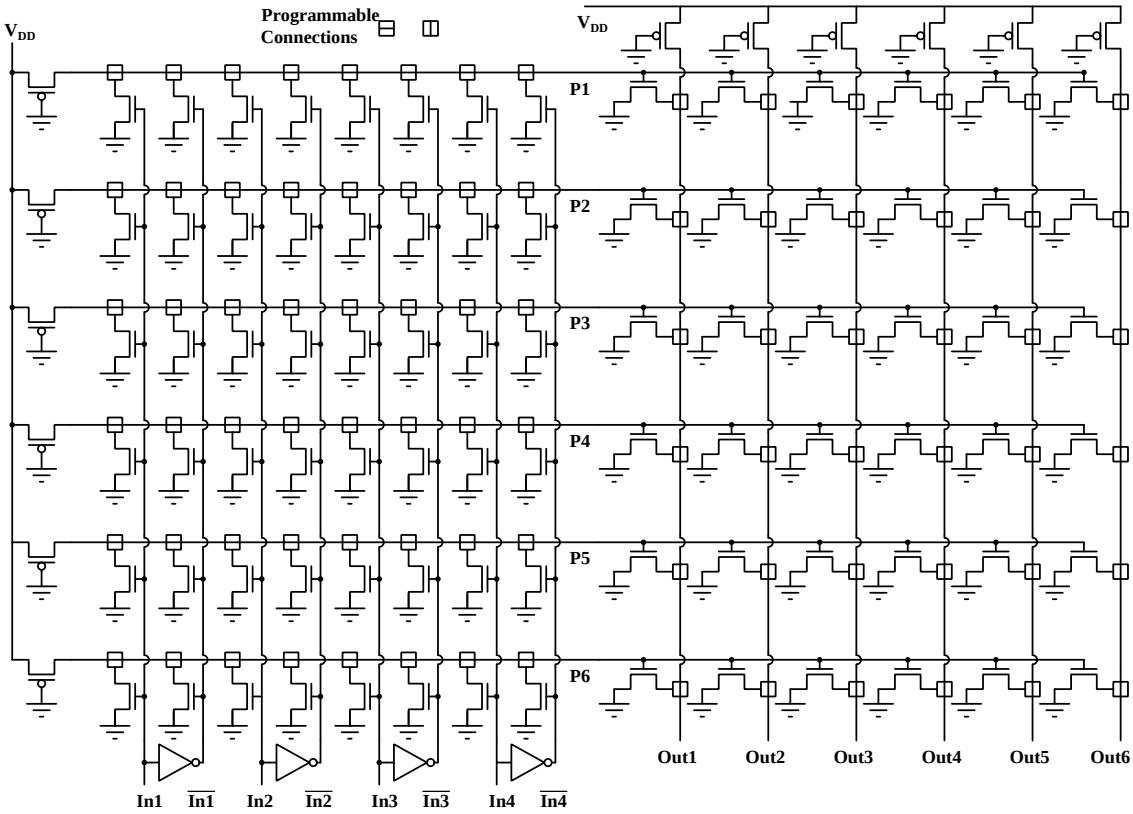
This problem is solved by implementing the product as the NOR of complemented inputs. The transistor size is independent of the number of inputs in a NOR gate, so we can include a variable number of (complemented) inputs without having to re-size the transistors.



– [**1**]

b) Show a transistor level circuit for a PLA implemented with pseudo NMOS logic with four primary inputs, six possible products and six outputs.

Soln. Q2b: The Product array will have 8 columns (for four inputs and their complements) and six rows for the six programmable products.
The Sum array will have six rows (from the six product inputs) and six columns for the six outputs. A circuit diagram is shown below:

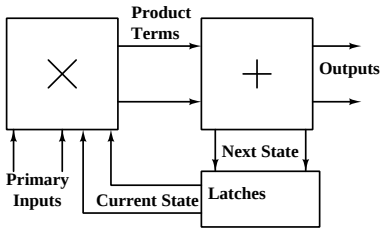


– [**1**]

c) Show how this configuration can be enhanced with latches to implement generic finite state machines.

Soln. Q2c: A finite state machine has a storage block (which encodes the current state) and two blocks of random logic – one to compute the next state from the current state and inputs, and the other to generate the outputs from the current state and inputs. Inputs to these two blocks is the same: current state information from the storage block and primary inputs to the fsm.

Since the PLA can generate arbitrary sums of products, the two random logic blocks can be easily implemented. Thus we only have to add latches for storage of current state and feed their outputs as additional inputs to the programmable logic array to implement any finite state machine. The PLA will generate the next state as well as outputs from latch outputs and primary inputs.

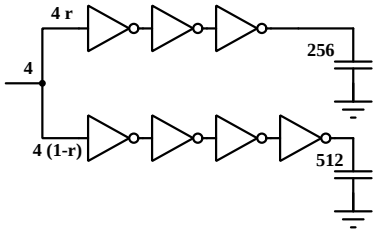


– **[1]**

– **[Q2: 1+1+1=3 marks]**

Q–3 We want to design a 3-4 fork with the total input capacitance (to be driven by the upstream driver) equal to 4 times the minimum inverter input capacitance.
Assume $\mathbf{p_{inv}} = \mathbf{2.0}, \gamma = \mathbf{2.2}$.

The input capacitance is divided in the ratio r:(1-r) for the 3 and 4 inverter branches of the fork respectively. The final load on the branch with 3 inverters is equivalent to 256 minimum inverters, while that on the 4 inverter branch is equivalent to 512 minimum inverters.



a) Evaluate the value of r such that the optimum delay in the two branches is equal, using Newton Raphson technique (starting with a guess value of r=0.5).

Soln. Q3a: For the upper branch, the input capacitance is $4r$, while the output capacitance is 256. Thus $H_1 = 256/4r = 64/r$. All g and b values are 1.

$$\text{Therefore,} \quad F_1 = 64/r, \text{ and correspondingly,} \quad \hat{f}_1 = \left(\frac{64}{r}\right)^{1/3} = 4r^{-1/3}$$

The delay through the upper arm of the fork is

$$D_1 = 3\hat{f}_1 + 3p_{inv} = 12r^{-1/3} + 3p_{inv}$$

For the lower branch, the input capacitance is $4(1-r)$, while the output capacitance is 512. Thus $H_2 = 512/4(1-r) = 128/(1-r)$. Since all g and b values are 1,

$$F_2 = \frac{128}{1-r}, \text{ and correspondingly,} \quad \hat{f}_2 = \left(\frac{128}{1-r}\right)^{1/4} = 3.3636(1-r)^{-1/4}$$

The delay through the lower arm of the fork is

$$D_2 = 4\hat{f}_2 + 4p_{inv} = 13.4543(1-r)^{-1/4} + 4p_{inv}$$

Condition for the two delays to be equal is: $13.4543(1-r)^{-1/4} + p_{inv} - 12r^{-1/3} = 0$

$$\text{Defining} \quad f(r) \equiv 13.4543(1-r)^{-1/4} + p_{inv} - 12r^{-1/3}$$

We seek the value of r which will make $f(r) = 0$. The derivative of $f(r)$ may be written as

$$f'(r) = -\frac{13.4543}{4}(1-r)^{-5/4}(-1) + \frac{12}{3}r^{-4/3} = 3.3636(1-r)^{-5/4} + 4r^{-4/3}$$

Taking the initial guess for r as 0.5, successive values for r can be tabulated as:

r	f(r)	f'(r)	next r
0.5	2.88095	18.0794	0.34065
0.34065	-0.251373	22.4743	0.351835
0.351835	-0.00333406	21.8878	0.351987
0.351987	-5.78369e-07	21.8803	0.351987
0.351987	-1.42109e-14	21.8803	0.351987

Thus, r = 0.352 will equalize delays. – **[3]**

b) Calculate the sizes of all transistors in the fork. Transistor widths are to be specified in units of the width of nMOS in the unit inverter.

Soln. Q3b:

$$\text{For the upper branch,} \quad \hat{f}_1 = \frac{4}{r^{1/3}} = \frac{4}{0.351987^{1/3}} = \frac{4}{0.70606} = 5.665232$$

All stages are inverters with $g = 1, b = 1$. Since $\hat{f} = gbh = 5.665232$, $h = 5.665232$ for all stages.

The first inverter should have an input capacitance of $4r = 1.408$
The next inverter should have an input capacitance of $1.408 \times h = 1.408 \times 5.665232 = 7.976$.
Input capacitance for the final inverter will be $7.976 \times h = 7.976 \times 5.665232 = 45.188$.
The final inverter can drive a load of $45.188 \times 5.665232 = 256$ as required.

Alternatively, we could have started with the output. As before:
$\hat{f} = gbh = 5.665232, g = 1, b = 1$, so $h = 5.665232$ for all stages. Final $C_{out} = 256$.
For the last inverter, $C_{in} = 256/5.665232 = 45.188$. This becomes the output capacitance of the second inverter. Since $h = 5.665232$ for all stages, $C_{in} = 45.188/5.665232 = 7.976$. for the second inverter. Finally, since the output capacitance of the first inverter is 7.976, its input capacitance is $7.976/5.665232 = 1.408$. This agrees with the value $4r$ as required.

Either way, we get the input capacitances of $1.408, 7.976$ and 45.188 respectively, for the three inverters.

Transistor geometries for the upper branch
The unit of width is the n channel transistor width in the minimal inverter. Thus, input capacitance of 1 corresponds to n channel width of 1 and p channel width of γ. Therefore an inverter stage with input capacitance of C_{in} will have n channel transistor width of C_{in} and p channel transistor width of $\gamma \times C_{in}$. Thus we can tabulate the transistor geometries as

First Inverter		Second Inverter		Third Inverter	
$C_{in} = 1.408$		$C_{in} = 7.976$		$C_{in} = 45.188$	
n width	p width	n width	p width	n width	p width
1.408	3.10	7.976	17.548	45.188	99.413

For the lower branch, $\hat{f}_2 = 3.3636/(1-r)^{1/4} = 3.749$.
Again all stages are inverters with $g = 1, b = 1$. Therefore, for all stages, $h = 3.749$.
Input capacitance of the first inverter is $4 \times (1-r) = 2.592$
Input capacitance of the following three inverters should be $2.592 \times 3.749 = 9.717$, $9.717 \times 3.749 = 36.43$ and $36.43 \times 3.749 = 136.57$.
The final inverter can drive a capacitance of $136.57 \times 3.749 = 512$ as expected.

We could have started with the output capacitance of 512 and successively divided by $h = 3.749$ to get input capacitances of the four inverters.

Given these input capacitance values, the n channel widths are equal to the capacitance, while the p channel widths are 2.2 times this value. Thus we can tabulate the transistor geometries for the lower branch as

First Inverter		Second Inverter		Third Inverter		Fourth Inverter	
$C_{in} = 2.592$		$C_{in} = 9.717$		$C_{in} = 36.43$		$C_{in} = 136.57$	
n width	p width	n width	p width	n width	p width	n width	p width
2.592	5.703	9.717	21.378	36.43	80.146	136.57	300.46

– **[4]**

c) Compute delays for both the branches of the fork.

Soln. Q3c: Delay for the upper branch is $3\hat{f}_1 + 3p_{inv} = 3 \times 5.665 + 6 = 22.995$.
Delay for the lower branch is $4\hat{f}_2 + 4p_{inv} = 4 \times 3.749 + 8 = 22.995$. – **[1]**

d) Without changing inverter sizes, assume that the actual load capacitors in both the branches are higher by 10%. Now what are the delays and how much is the difference in delays of the two branches?

Soln. Q3d: Since inverter sizes remain the same, all inverters except the final one see the same load. Therefore change in the final load capacitor will change the delay of the last stage only. The remaining delays will remain the same. therefore delays in the two branches are:

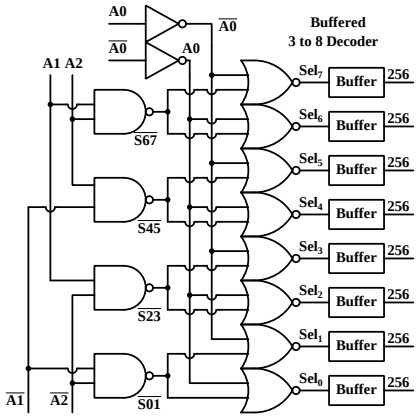
$$\begin{aligned} D_1 &= 2\hat{f}_1 + 1.1 \times \hat{f}_1 + 3p_{inv} = 3.1 \times 5.665 + 6 = 17.562 + 6 = 23.562 \\ D_2 &= 3\hat{f}_2 + 1.1 \times \hat{f}_2 + 4p_{inv} = 4.1 \times 3.749 + 8 = 17.562 + 6 = 23.371 \end{aligned}$$

The difference in delays is therefore $23.562 - 23.371 = 0.192$. (The upper branch is slower by this amount). – **[1]**

– **[Q3: 3+4+1+1=9 marks]**

Q–4 We want to design a 3 bit decoder where the decoder outputs have to be buffered to drive a load equivalent to 256 minimal inverters. Assume $\mathbf{p_{inv}} = \mathbf{2.0}, \gamma = \mathbf{2.2}$.

Assume that the 3 bits to be decoded and their complemented values are available as inputs and can drive loads equivalent to 4 minimal inverters. Decoding is done using a 2 step NAND-NOR circuit as shown. The decoded outputs Sel_i are buffered using inverters to drive the load. The number of inverters in the buffer is to be chosen to minimise the delay of the path involving NAND-NOR-Buffer. The number of inverters in the buffer can be even or odd, since true or complemented values of select outputs are equally acceptable.



a) Find the optimum value of ρ using Newton Raphson technique, starting with a guess value of $\rho = 4$.

Soln. Q4a: We define $f(\rho) \equiv \rho(1 - \ln \rho) + p_{inv}$.
Then the optimum value of ρ is the one which makes $f(\rho) = 0$.
We can solve this non-linear equation using Newton Raphson iterations. Derivative of $f(\rho)$ is given by

$$f'(\rho) = 1 - \ln \rho - \rho \frac{1}{\rho} = -ln(\rho)$$

Therefore given a guess value g for ρ, is the next refined value of g is given by

$$g_{next} = g - \frac{f(g)}{f'(g)} = g + \frac{g - g \ln g + p_{inv}}{\ln g} = \frac{g + p_{inv}}{\ln g}$$

Starting with $g = 4$, we get successive values for ρ as
4.0, 4.3281, 4.3191, 4.3191.
Thus the optimum stage ratio is 4.3191. – **[1]**

b) Find the optimum number of stages in the path through NAND-NOR and Buffer. How many inverters should be used in the buffer?

Soln. Q4b: Since each input drives two NAND gates, each NAND gate should have an input capacitance of 2. Thus H = 256/2 = 128.
The output of each NAND drives two NOR gates. Therefore $b = 2$ for the NAND stage, while all other stages have $b = 1$. Therefore B = 2.
g for the NAND gates is $(2 + \gamma)/(1 + \gamma) = 4.2/3.2 = 1.3125$.
g for NOR gates is $(1 + 2\gamma)/((1 + \gamma) = 5.4/3.2 = 1.6875$.
Therefore G for the path is $1.3125 \times 1.6875 \times 1 \times 1 = 2.214844$.
Thus the path effort $F = GBH = 2.21488 \times 2 \times 128 = 567$.
The optimum number of stages for this path effort is

$$N = \frac{\ln F}{\ln \rho} = \frac{6.34036}{1.46305} = 4.333667$$

This suggests a total of 4 or 5 stages. For 4 stages, $\hat{f} = 567^{1/4} = 4.87973$ and the delay is $4 \times 4.87973 + p_{NAND} + p_{NOR} + 2p_{inv}$.
With $p_{inv} = 2$, this is $23.5189 + p_{NAND} + p_{NOR}$.

For 5 stages, $\hat{f} = 567^{1/5} = 3.55399$. This gives a delay of $5 \times 3.55399 + 3p_{inv} + p_{NAND} + p_{NOR}$. With $p_{inv} = 2$, this is $23.76997 + p_{NAND} + p_{NOR}$.

Thus there is slightly higher delay for a 5 stage implementation. Not only is the 4 stage decoder somewhat faster, it is smaller and is likely to consume less power.

Hence we choose a 4 stage design. Two of these are the NAND and NOR gates, so we should buffer Sel_i outputs with 2 inverters. – **[1]**

c) Find the transistor sizes for NAND, NOR and all the inverters in the buffer such that the path delay is minimum.

Soln. Q4c: The path to be optimized has four stages: NAND, NOR, Inverter1 and Inverter2. $F = 567$ and correspondingly $\hat{f} = 567^{1/4} = 4.87973$.

Parameter	NAND	NOR	Inv.1	Inv.2
$\hat{f}$	4.87973			
g	$4.2/3.2 = 1.3125$	$5.2/3.2 = 1.6875$	1	1
b	2	1	1	1
$h = \hat{f}/gb$	1.858945	2.8917	4.87973	4.87973
C_{in}	2	3.718	10.751	52.462
$C_{out} = hC_{in}$	3.718	10.751	52.462	256
W_n	3.048	2.203	10.751	52.462
W_p	3.352	9.694	23.652	115.42
$W_n + W_p$	6.4	11.897	34.403	167.882
$(W_n + W_p)/C_{in}$	3.2	3.2	3.2	3.2

1. **NAND stage:** $g = 4.2/3.2 = 1.3125, b = 2, \hat{f} = gbh = 4.87973$.
   So $h = 4.87973/(2 \times 1.3125) = 1.858945$.
   C_{in} for NAND is 2, Hence $C_{out} = C_{in} \times h = 2 \times 1.858945 = 3.718$.
   A NAND gate with n width of 2 and p width of $\gamma(= 2.2)$ will have
   $C_{in} = 4.2/3.2 = 1.3125$ in units of minimal inverter capacitance.
   Thus, $C_{in} = 1.3125$ corresponds to nMOS width of 2 and pMOS width of 2.2.
   Since actual C_{in} for NAND is 2, the nMOS width should be $2 \times 2/1.3125 = 3.048$, and pMOS width should be $2 \times 2.2/1.3125 = 3.352$.
2. **NOR stage:** $g = 1.6875, b = 1, \hat{f} = gbh = 4.87973$.
   So $h = 4.87973/(1 \times 1.6875) = 2.8917$
   C_{in} for NOR C_{out} for NAND $= 3.718$.
   Hence $C_{out} = C_{in} \times h = 3.718 \times 2.8917 = 10.751$.
   A NOR gate with n width of 1 and p width of $2\gamma = 4.4$ has $C_{in} = 5.4/3.2$ in units of inverter capacitance. Thus, $C_{in} = 1.6875$ corresponds to nMOS width of 1 and pMOS width of $2\gamma = 4.4$. Since actual $C_{in} = 3.718$ for the NOR gate, nMOS width should be $3.718/1.6875 = 2.203$. and pMOS width should be $4.4 \times 3.718/1.6875 = 9.694$.
3. **First inverter** $g = 1, b = 1, \hat{f} = gbh = 4.87973$. So $h = 4.87973$
   $C_{in} = 10.751$. $C_{out} = C_{in} \times h = 10.751 \times 4.87973 = 52.462$.
   nMOS width for the first inverter should be 10.751,
   while the pMOS width should be $2.2 \times 10.751 = 23.652$.
4. **Second inverter** $g = 1, b = 1, \hat{f} = gbh = 4.87973$. So $h = 4.87973$
   $C_{in} = 52.462$, $C_{out} = C_{in} \times h = 52.462 \times 4.87973 = 256$. This is as desired.
   For $C_{in} = 52.462$, nMOS width should be 52.462 and pMOS width should be $2.2 \times 52.462 = 115.42$

– **[3]**

– **[Q4: 1+1+3=5 marks]**

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

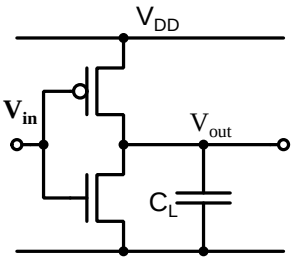
EE 671: VLSI Design

Saturday	Class Test 1	Time: 1730-1830
18-08-18	Autumn Semester 2018	Marks: 10

All numerical answers should be accurate to 1%.

Q–1

Consider a CMOS inverter as shown on the right. Assume that the supply voltage V_{DD} is 3.3V and the external load capacitance C_L is 0.1 pF. Parameters for the n and p channel transistors are :



Parameter	N Channel	P Channel
μC_{ox}	$45\mu A/V^2$	$22\mu A/V^2$
V_T	0.6 V	-0.6 V

The rise time for a CMOS inverter is given by

$$\frac{K_p \tau_{rise}}{C} = \frac{2(V_{iL} + V_{Tp})}{(V_{DD} - V_{iL} - V_{Tp})^2} + \frac{1}{(V_{DD} - V_{iL} - V_{Tp})} \ln \frac{V_{DD} + V_{oH} - 2V_{iL} - 2V_{Tp}}{V_{DD} - V_{oH}}$$

Here C is the total capacitance given by $C_L + C_p$, where C_p is the parasitic capacitance. K_p is the conductance factor for the P channel transistor, with $K_p = \mu_p C_{ox} W/L$. V_{Tp} represents the absolute value of the p channel threshold voltage. Channel length L for all transistors is 0.35μm.
The parasitic capacitance is given by $C_p = \alpha W$, with $\alpha = 10^{-14} F/\mu m$.
Other symbols have their usual meanings.

Find the width for the pMOS transistor such that the output will charge from 0V to 3.0V in 5 ns when the input voltage is 0.3V.

Soln. 1)

$$\frac{K_p \tau_{rise}}{C} = \frac{2(V_{iL} + V_{Tp})}{(V_{DD} - V_{iL} - V_{Tp})^2} + \frac{1}{(V_{DD} - V_{iL} - V_{Tp})} \ln \frac{V_{DD} + V_{oH} - 2V_{iL} - 2V_{Tp}}{V_{DD} - V_{oH}}$$

$V_{DD} - V_{iL} - V_{Tp} = 3.3 - 0.3 - 0.6 = 2.4$V, So

$$\frac{22 \times 10^{-6} \times (W/.35) \times 5 \times 10^{-9}}{10^{-13} + 10^{-14} \times W} = \frac{2(0.3 + 0.6)}{2.4^2} + \frac{1}{2.4} \ln \frac{3.3 + 3.0 - 2 \times 0.3 - 2 \times 0.6}{3.3 - 3.0}$$

$$\text{Therefore} \qquad \frac{3.142857W}{1 + 0.1W} = 0.3125 + \frac{1}{2.4} \ln 15 = 1.440854$$

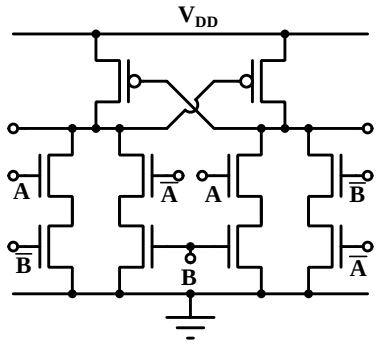
This gives

$$3.142857W = 1.440854 + 0.1440854W \qquad \text{or} \qquad 2.998772W = 1.440854$$

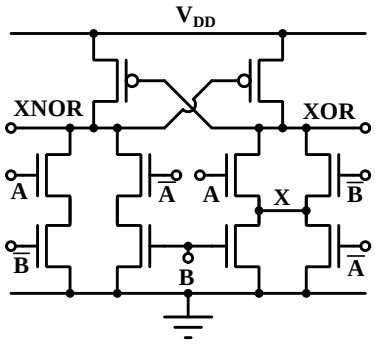
And therefore $W = 0.4805\mu$m. – **4**

Q–2

Where does the circuit given on the right for a CVSL gate deviate from the usual series-parallel rule?
Does it still provide a valid implementation for some logic function? If so, identify the function and describe how it works.
Otherwise suggest the necessary changes to the circuit to make it operate properly.



Soln. 2) The left side of the circuit has transistors A and $\overline{B}$ in series, and this is in parallel with transistors $\overline{A}$ and B in series.
By series parallel rule, we should have transistors $\overline{A}$ and B in parallel which should be in series with transistors A and $\overline{B}$ in parallel. This would give the following circuit:

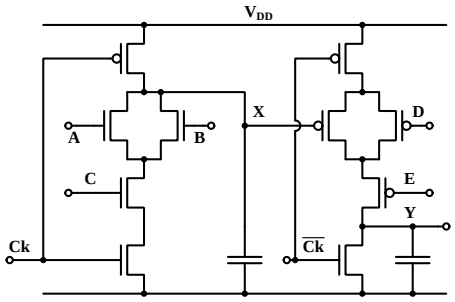


This iplements $\overline{A \cdot \overline{B} + \overline{A} \cdot B}$ or XNOR of A and B on the left, with XOR on the right output. The only difference between this circuit and the given circuit is the shorting wire to form the two parallel combinations (marked X in the figure).

However, this wire is redundant. It connects transistor A with transistor $\overline{A}$ which will never be on simultaneously and so no current will flow through the wire because of these. Similarly, it connects transistor $\overline{B}$ with transistor B. These will also never be on simultaneously and so no current will flow through this path either. Thus this wire can be removed without changeing the functionality of the circuit.

Indeed, if one looks at the circuit on the right after removing the wire, it has transistors A and B in series which is in parallel with series connected transistors $\overline{A}$ and $\overline{B}$. This implements the function $\overline{A \cdot B + \overline{A} \cdot \overline{B}}$, which is $\overline{A \text{ XNOR } B}$ or A XOR B, which is the complement of the left side function. Hence this circuit will work properly as the XNOR/XOR of A and B. – **[2]**

Q–3 Consider the two stage np zipper circuit given below. Identify the logic functions appearing at nodes x and y. Draw a timing diagram showing $Ck, \overline{Ck}, X$ and Y on the same time scale, when A = C = D = E = '1' and B = '0'.



– **[2]**

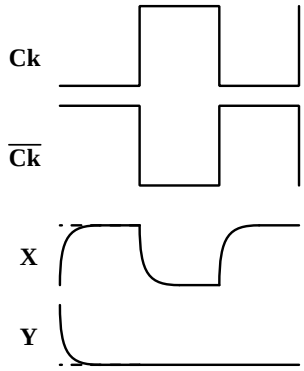
Soln. 3) When Ck = '0', the first stage pre-charges, while the second stage discharges its output capacitor.
When Ck = '1', the two stages evaluate. The first stage will discharge its output capacitor if either A or B is '1' and C is '1'. Therefore,

$$X = \overline{(A+B).C}$$

The second stage evaluation begins with a discharged output capacitor. It will charge up to '1' only if either X or D is '0' and E is '0'. Thus $Y = (\overline{X} + \overline{D}) \cdot \overline{E}$. So

$$Y = ((A+B).C + \overline{D}) \cdot \overline{E}$$

The figure below shows a timing diagram for the two outputs when A = C = D = E = '1' and B = '0'.



– **[2]**

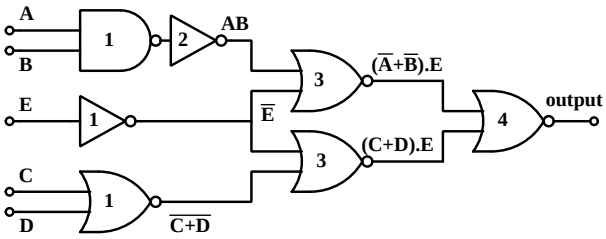
Q–4 Show a logic diagram for the function $A \cdot B \cdot (\overline{C+D}) + \overline{E}$ implemented in 4 phase dynamic logic using only inverters, two input NANDs and two input NORs.

A,B,C,D and E are available in phase 1 in uncomplemented form only. Transistor level circuits are not required. Just draw a logic diagram with the type of each gate inscribed in its gate symbol. The circuit should produce the output as quickly as possible.

Soln. 4) The final gate should be inverting. Therefore we evaluate the function as

$$\overline{(\overline{A} + \overline{B} + (C+D)) \cdot E} = \overline{(\overline{A} + \overline{B}) \cdot E + (C+D).E}$$

The figure below shown an implementation for this function.



The output is available in phase 1 of the next cycle. – [**2**]

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

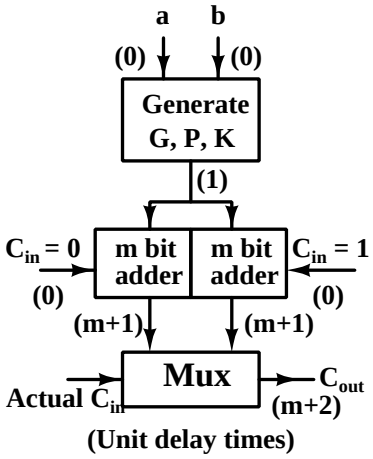
EE 671: VLSI Design

Saturday	Class Test 2	Time: 1730-1830
13-10-18	Autumn Semester 2018	Marks: 10

Q–1 Show how carry select addition can speed up the addition of wide words. How can the addition be made faster by using a variable number of bits in each group of carry select adders? How are the number of bits chosen in each group for square root tiling of carry select adders? Why is this scheme called square root tiling?

Soln. 1) Adders are slow because component adders have to wait for carry to arrive from the previous bits. In carry select adders, we divide the width of the adder into sub-adders of smaller size.

Each sub-adder uses two adder circuits. One of these assumes the input carry to be '0', while the other assumes it to be '1'. The two possible outputs are thus pre-computed and the actual carry input just selects the correct value. When the actual carry arrives, it just has to operate a 2 way mux to select the correct carry. The two possible outputs are computed in constant time, because these don't wait for carry from the previous bits. The actual carry ripples across groups (sub-adders) and not across each bit. Also, the delay of the rippling carry at each group is just the mux delay. Hence the carry select adder is much faster.



The first group at LSB does not have to use two adders, as the input carry is already available and no selection is required. Therefore its output is available after (m+1) units of time. All subsequent sub adders take (m+2) units of time. Because the actual carry arrives later and later (due to mux delay) at each sub-adder as we go from LSB to MSB, each sub-adder can be of a larger size and can still produce alternative results just in time as the actual carry arrives. In case of square root tiling, each sub-adder adds one more bit than the previous one.

Assuming the first group (which has no mux) has s bits. Its output will be available in (s+1) units of time in unit delay model. The next sub-adder can also be s bits wide, so that its alternatives are available at (s+1). After mux delay, the carry output will be valid at (s+2). Therefore, the next group can be of (s+1) bits and have the alternatives available at (s+2) just as its input carry arrives.

The total number of bits added will be given by

$$n = s + s + (s+1) + (s+2) + (s+3) + \cdots$$

Each group adds just one unit of time to produce its output. Therefore, (ignoring the initial group), the delay is proportional to the number of groups. For g groups, we have

$$n = gs + 1 + 2 + 3 + \cdots + (g-2) = gs + \frac{(g-2)(g-1)}{2} \approx g^2/2 \text{ For large g}$$

So $g \approx 2\sqrt{n}$

Since the delay is proportional to g, it becomes proportional to $\sqrt{n}$. Hence the name. – **[Q1: 2 marks]**

Marking Key:
Description, $t \propto g$: 1
$n \propto g^2$ so $t \propto \sqrt{n}$:1

Q–2 Describe the working of modified Booth algorithm for multiplication.
Illustrate it by working out the multiplication of unsigned binary numbers 100101 and 1101. All partial products should be shown as binary numbers and sign extension should be carried out where required. Show that by adding the binary partial products, you get the expected answer.

Soln. 2) In booth algortithm, we multiply two bits at a time. This reduces the number of partial product rows to be added, which speeds up a multiplier.
The two bit combination can be "00", "01", "10" or "11". If the multiplicand is A, the partial product corresponding to multiplication by these 2 bits is 0, A, 2A or 3A. The first three can be generated easily from A. Instead of generating 3A, we generate 2's complement of A (which is -A) as the partial product and increment the next significant two bit group. This is equivalent to adding 4A and subtracting A, so the final sum will be correct (*i.e.* 3A). Implementation of this technique will require decoding the previous two bits for each group to see if it was "11" and accordingly, whether the current multiplier should increment or not. To ease this task, in modified Booth algortithm, for "10" also we subtract 2A and increment the next two bit group. Now the next group will always increment if the more significant bit of the previous group is '1'. Thus the partial product generation logic is:

Current 2 bits	Previous MSB	Partial Product	Remark
00	0	0	
01	0	A	
10	0	-2A	Sign extension reqd
11	0	-A	Sign extension reqd
00	1	A	
01	1	2A	Left Shift A
10	1	-A	Sign extension reqd
11	1	0	

In the given example, we are required to multiply 100101 by 1101 or 37 by 13. Since these are unsigned (positive) numbers, we shall append a '0' to the left to avoid interpreting these as negative numbers. Thus $A = 0100101$ and multiplier is 01101. The possible partial products are $-A = 1011011$, $2A = 01001010$ and $-2A = 10110110$. The most significant bit of negative partial products needs to be sign extended to the maximum word size used for addition.

Overlapping groups of 3 bits for the multiplier are 01 : 0, 11 : 0 and00 : 1 where the bit following the colon is the MSB of the previous group of 2 bits. The least significant group receives no request from the preceding group and therefore assumes a 0 for the MSB of the previous group. Thus the partial products are:

Current 2 bit gp.	MSB of Prev. gp.	Partial Product	Binary PP
01	0	A	0100101
11	0	-A	1011011
00	1	A	0100101

Adding these partial products with the correct place values and sign extension for the $-A$ term gives:

$$\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & & \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & & & & \\ \hline 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

The final carry out has to be complemented in signed addition. Thus the result is

$$00111100001 = 1E1\ 'H\ = 256 + 14 \times 16 + 1 = 256 + 224 + 1 = 481$$

The given multiplication was $37 \times 13 = 481$. So the result agrees. – **[Q2: 2 marks]**

Marking Key: Description, Booth table: 1 mark
Numerical verification: 1 mark

Q–3 A multiply and accumulate circuit uses a multiplier and an accumulator to compute expressions of the type $a_i = a_i + c_i x_i$ in a single unit. Since the process for multiplication implements multi-bit addition anyway, the bits of the accumulator just provide additional wires to the partial products at the corresponding weights.

a) Show a dot diagram for wire reduction using Wallace scheme for a multiply and accumulate circuit which has an 8 bit wide multiplicand, 6 bit wide multiplier and 15 bit wide accumulator. Use the scheme which does not produce a redundant MSB. (When two wires are left after allocating full adders, feed these through unless all bits at lower weights have a single wire or feeding through will exceed the capacity of the next layer.)
What is the width of the final adder to be used after wire reduction to ≤ 2 wires at each weight?

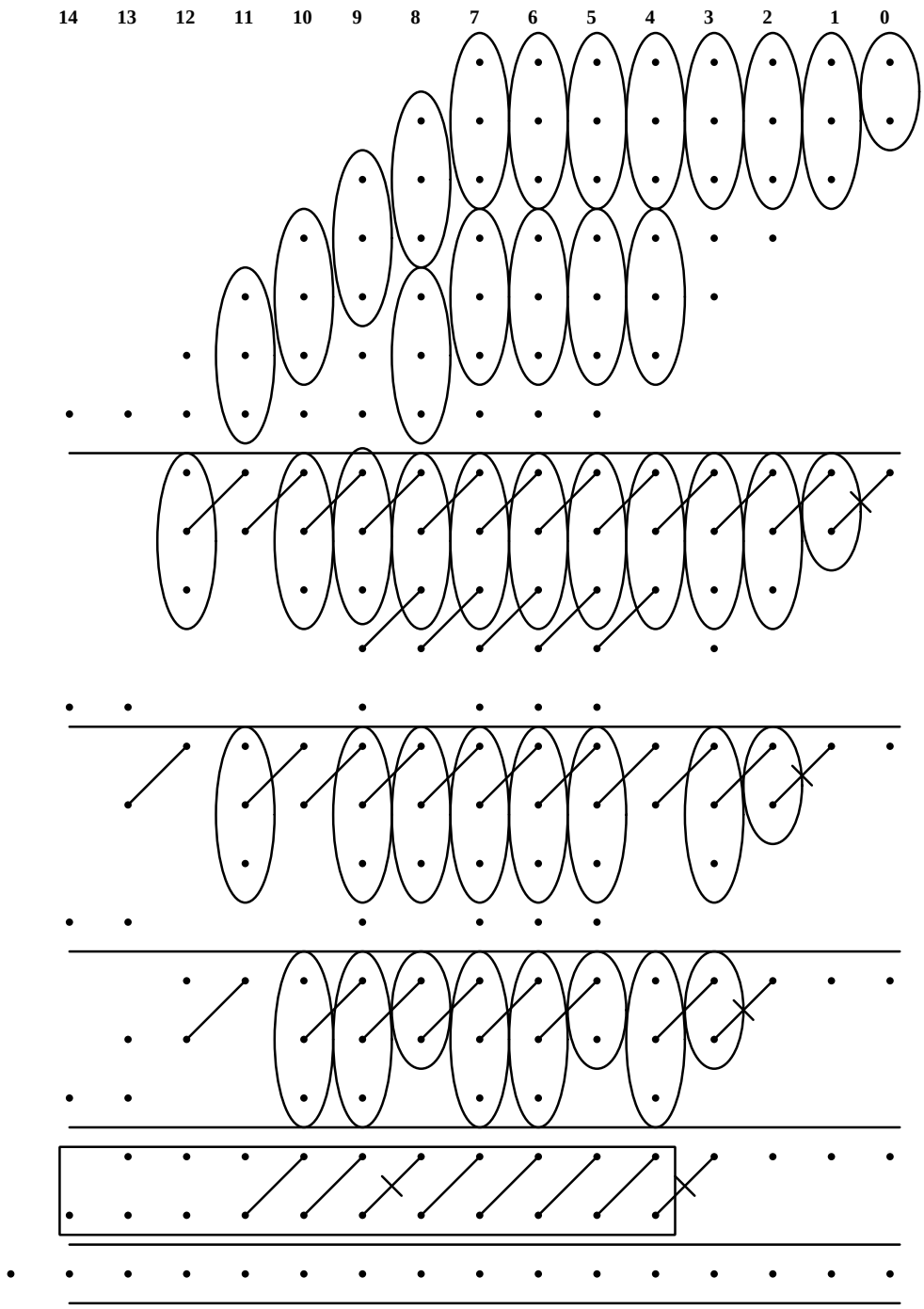
Soln. 3-a) The number of wires of just the multiplier at different bit positions are:

Bit	12	11	10	9	8	7	6	5	4	3	2	1	0
Wires	1	2	3	4	5	6	6	6	5	4	3	2	1

Including an additional wire from Bit 14 to Bit 0 for the accumulator gives the total wire count to begin with as

Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wires	1	1	2	3	4	5	6	7	7	7	6	5	4	3	2

A dot diagram for reduction by Wallace procedure is give below. Notice that the least significant bit has 2 wires. These should be reduced by a half adder because there are no bits to the right of this.



4 least significant bits are already reduced to a single wire. Therefore an 11 bit adder is required. – **[2]**

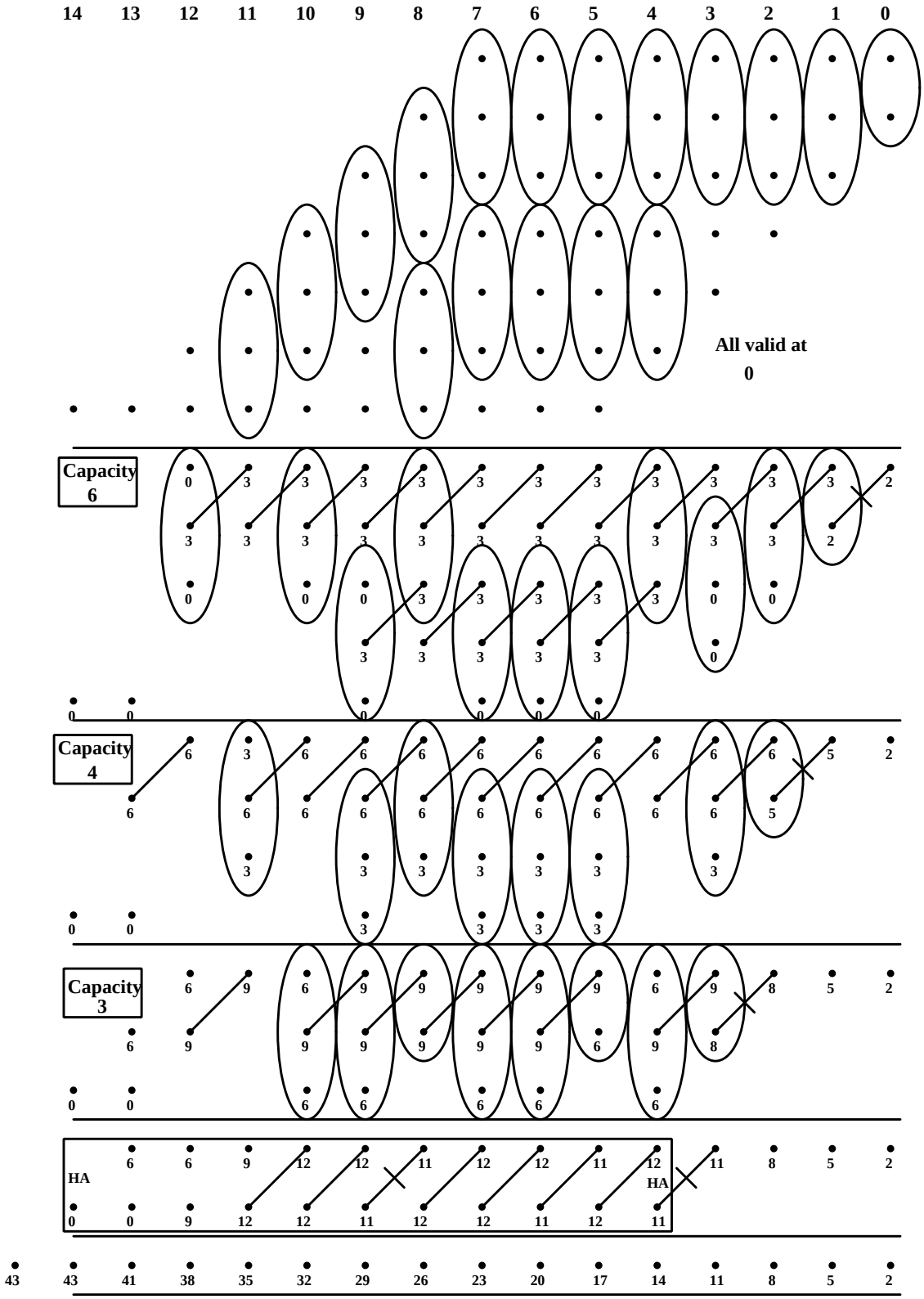
Marking Key: wire reduction, width of final adder: 2

b) Assume that all partial product bits are ready at time 0.

Assume that a half adder produces its sum and carry outputs in 2 units of time, while a full adder produces its sum and carry outputs in 3 units of time after the arrival of the last of its inputs.

Redraw the dot diagram, placing the arrival time of the signal below each dot. (Draw a neat new diagram, do not re-use the diagram in the part above). You must choose wires to feed to full/half adders or for passing through in such a way that the late arrivals incur less additional delay. (This optimizes the critical path delay).

Soln. 3-b)



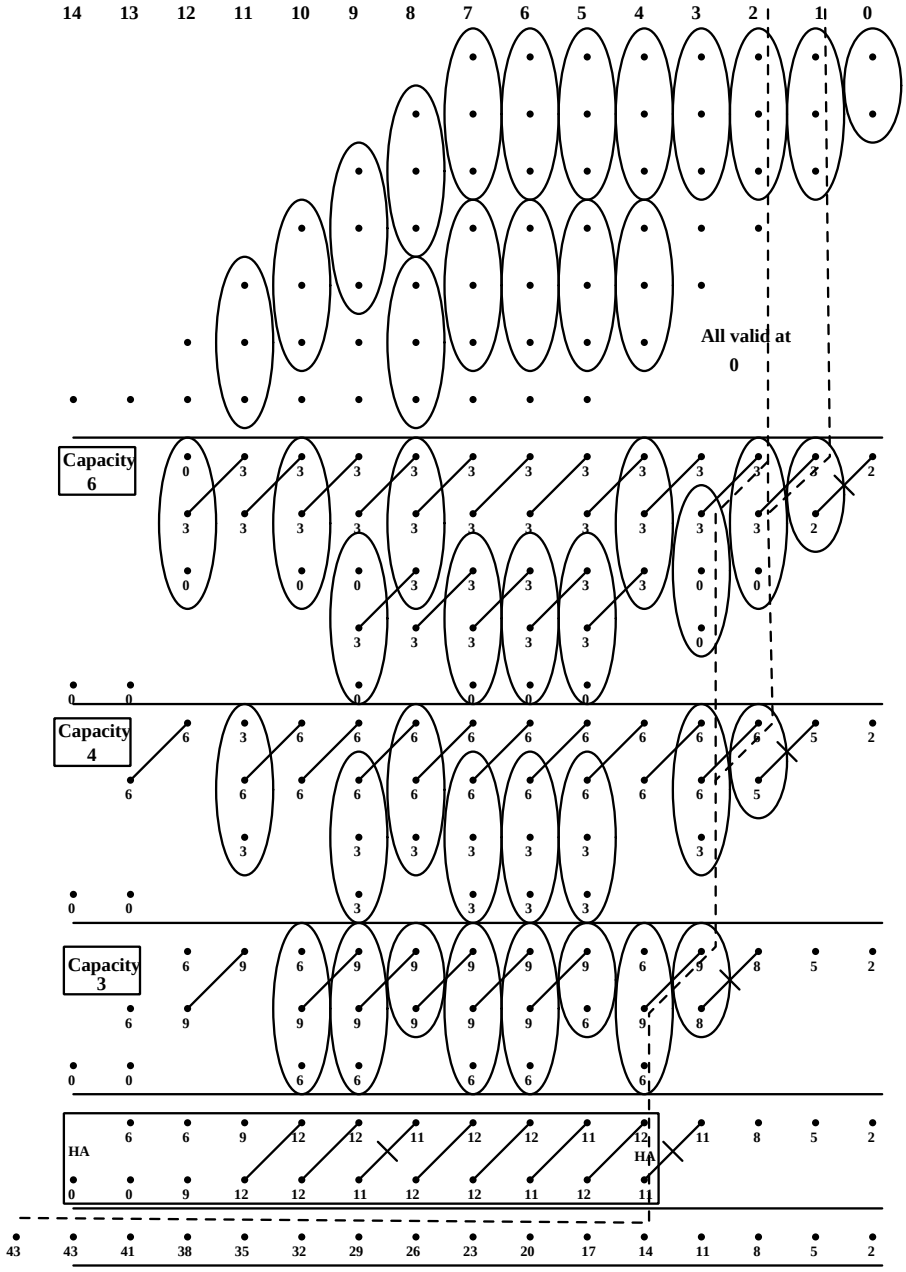
– [3]

Marking Key: Time computation for reduction: 2 marks
Choice of wires: 1 mark

c) If a ripple carry adder is used for the final addition, show the time for the final

output dots to become valid. Show the critical path of the MAC circuit.

Soln. 3-c) Time for final output dots are shown in the dot diagram above.
The last bits to become active are bits 14 and 15 at 43. This time is the result of bit 13, an input to the half adder, being vaild at 41. This bit becomes valid at 41 because carry for the full adder was provided at 38 by bit 12 and so on down the ripple carry adder at bit 4. This is valid at 14 because the half adder producing it got a reduced input at 12. Now tracking this up through the reduction, (Critical path is not unique - late arrival of either bit 1 or bit 2 will delay the final result. we get the critical path as shown:



– **[1]**

– **[Q3: 2+3+1=6 marks]**

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

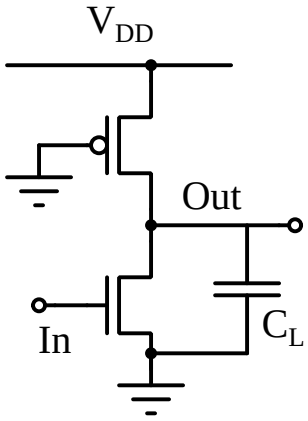
EE 671: VLSI Design

Tuesday	Class Test 1	Time: 1130-1300
22-08-16	Autumn Semester 2017	Marks: 10

All numerical answers should be accurate to 1%.

Q–1

Consider a pseudo-nMOS inverter as shown on the right. Assume that the supply voltage V_{DD} is 3.3V and the load capacitance C_L is 0.1 pF. Parameters for the n and p channel transistors are :



Parameter	N Channel	P Channel
μC_{ox}	$45\mu A/V^2$	$22\mu A/V^2$
V_T	0.6 V	-0.6 V

a) Find the W/L value for the pMOS transistor which will charge the load capacitor from 0V to 3.0V in 5 ns when the nMOS transistor is OFF. The rise time for a pseudo-nMOS inverter is given by

$$\tau_{rise} = \frac{C_L}{\mu_p C_{ox}(W_p/L_p)(V_{DD} - V_{Tp})}\left[\frac{2V_{Tp}}{V_{DD} - V_{Tp}} + \ln\frac{V_{DD} + V_{oH} - 2V_{Tp}}{V_{DD} - V_{oH}}\right]$$

V_{Tp} in the above expression represents the absolute value of the p channel threshold voltage.

Soln.: Using the given expression, we get

$$5 \times 10^{-9} = \frac{10^{-13}}{22 \times 10^{-6}(W_p/L_p)(3.3 - 0.6)}\left[\frac{1.2}{3.3 - 0.6} + \ln\frac{3.3 + 3.0 - 1.2}{3.3 - 3.0}\right]$$

$$\text{So} \quad \frac{W_p}{L_p} = \frac{20}{22 \times 2.7}\left[\frac{1.2}{2.7} + \ln\frac{5.1}{0.3}\right] = 3.3670 \times (0.4444 + 2.8332)$$

Which gives

$$\frac{W_p}{L_p} = 1.1036$$

– 2

b) Find the value of the equivalent resistor which will charge the load capacitor from 0V to 3.0V in the same amount of time (5 ns).

Soln.: The voltage across the capacitor C_L being charged by resistor R from the supply is given by:

$$V_{Out} = V_{DD}\left(1 - e^{-t/RC_L}\right)$$

This gives

$$e^{-t/RC_L} = 1 - \frac{V_{Out}}{V_{DD}} = \frac{V_{DD} - V_{Out}}{V_{DD}} \qquad \text{So} \quad e^{t/RC_L} = \frac{V_{DD}}{V_{DD} - V_{Out}}$$

$$\text{Therefore} \quad t = RC_L \ln \frac{V_{DD}}{V_{DD} - V_{Out}} \qquad \text{and so} \quad RC_L = \frac{t}{\ln \frac{V_{DD}}{V_{DD} - V_{Out}}}$$

If the resistor is so chosen that C_L charges to 3.0V in 5 ns, we get

$$R = \frac{5 \times 10^{-9}}{10^{-13} \times \ln \frac{3.3}{3.3-3.0}} = \frac{50 \times 10^3}{\ln 11} = 20.8516\text{K}\Omega$$

– **1**

c) Find the ratio of (W/L) values for the n channel and p channel transistors such that the static output voltage is = 0.3V when the input voltage is 3.0V. (No memorized expressions should be used. Find the output voltage by equating currents through the two transistors.)

Soln.: When the input voltage is 3.0V and the output is 0.3 V, the nMOS transistor is in linear mode and the pMOS is in saturation. Equating currents, we get

$$K_n \left((3.0 - 0.6) \times 0.3 - \frac{1}{2}(0.3)^2 \right) = \frac{K_p}{2}(3.3 - 0.6)^2$$

This gives

$$\frac{K_n}{K_p} = \frac{2.7^2}{2 \times 2.4 \times 0.3 - 0.09} = 5.4$$

Therefore,

$$\frac{W_n/L_n}{W_p/L_p} = 5.4 \times 22/45 = 2.64$$

– **2**

d) We represent the inverter as a voltage divider, with the p channel transistor replaced by its equivalent resistor computed in part b) above and the n channel transistor by another resistor, such that the static output is = 0.3V. What is the ratio of the equivalent resistors for n channel and p channel transistors?

Soln.: Since R_p and R_n form a voltage divider across V_{DD} to give an output of 0.3 V, we must have

$$0.3 = 3.3\frac{R_n}{R_n + R_p} \qquad \text{Or} \quad 11 = 1 + \frac{R_p}{R_n} \qquad \text{So} \quad \frac{R_p}{R_n} = 10$$

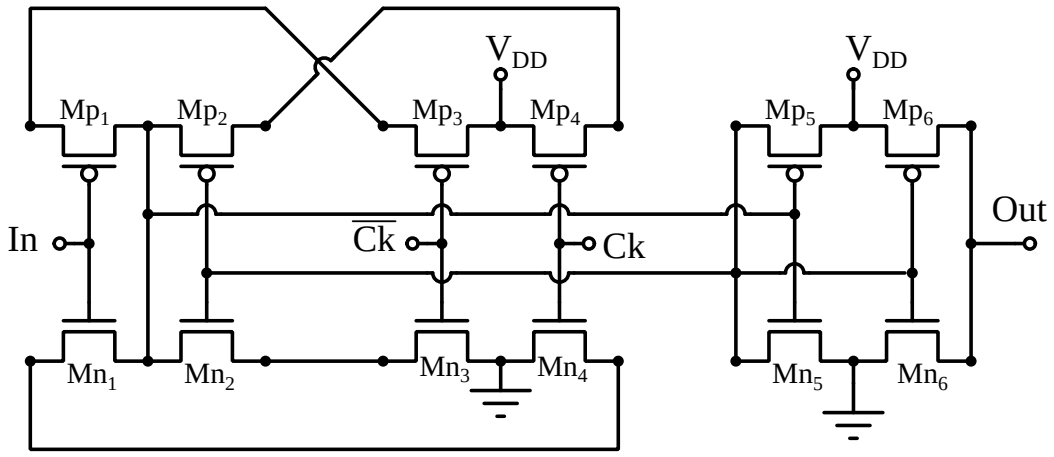
Therefore

$$\frac{R_n}{R_p} = 0.1$$

and accordingly, $R_n = 2.085\text{K}\Omega$. – **1**

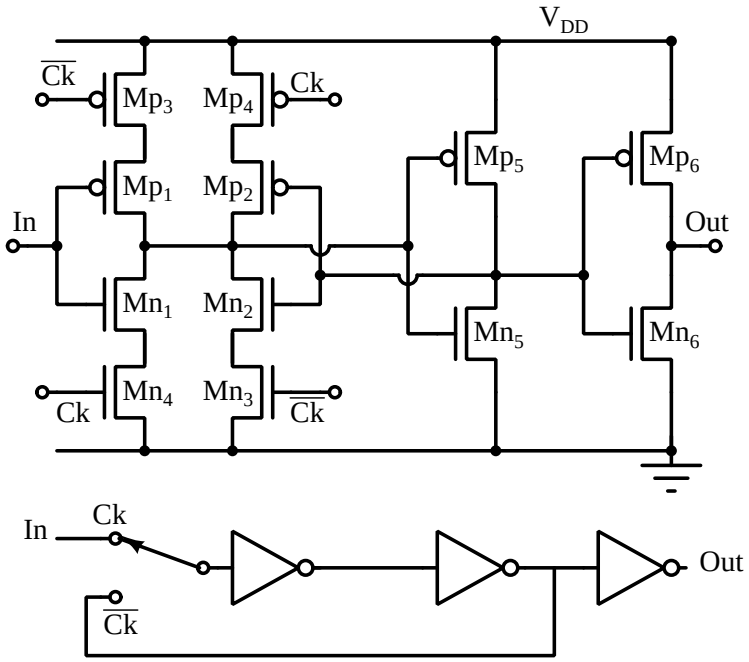
– **[Q–1: 2+1+2+1= 6 marks]**

Q–2 A circuit has been designed in "sea of gates" style, using interconnects as shown below:



a) Re-draw the schematic in conventional style, separating all gates and with V_{DD} on top and ground at the bottom. (Your schematic should clearly identify the labels for all transistors and signals corresponding to the labeling above).

Soln.: The circuit can be re-drawn as follows:



The two tri-stateable inverters are enabled by Ck and $\overline{Ck}$ respectively and have their outputs shorted. This constitutes a multiplexer with inverter. The equivalent logic diagram is also given in the figure above. **– 1**

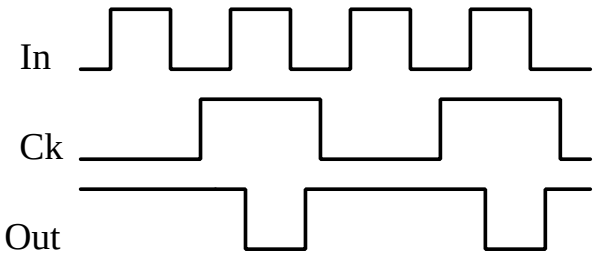
b) What function does this circuit perform? Describe how it works. When Ck is high, the multiplexer chooses In, which appears at the output after three inversions. Thus the output changes as the input changes, providing an inverted version of it.

When Ck is low, the multiplexer forms a latch with the first two inverters. Its output is inverted by the third inverter and appears as the output.

Thus the circuit is a transparent D latch, providing its $\overline{Q}$ as the output. It follows the input (with an inversion) when Ck is high, and is latched when Ck goes low. – **2**

c) For the input and clock waveforms given below, sketch the expected output showing the timing relationship with respect to the clock and the input (In).

Soln.:



The clock has just gone low at the start. Therefore the inverted value of input is latched. When the clock goes high, the latch becomes transparent and the output is the inverted value of input. Again when clock drops low, the output remains latched and ignores further changes in the input. – **1**

– **[Q–2: 1+2+1= 4 marks]**

Paper Ends

Reference

You can use the following MOS model:

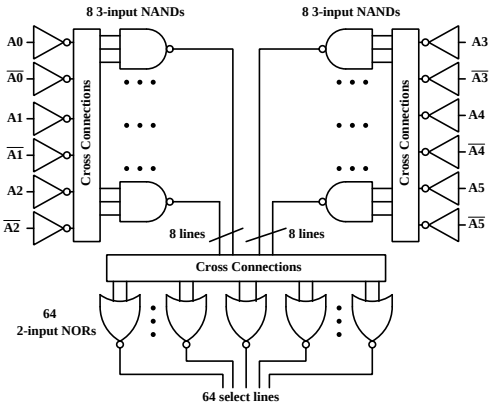
$$
\begin{aligned}
I_{DS} &= 0 && \text{when } V_{GS} \le V_T \\
I_{DS} &= K\left[(V_{GS}-V_T)V_{DS} - \tfrac{1}{2}V_{DS}^2\right] && \text{when } V_{GS} > V_T, \quad V_{DS} \le V_{GS}-V_T \\
I_{DS} &= \tfrac{K}{2}\left(V_{GS}-V_T\right)^2 && \text{when } V_{GS} > V_T, \quad V_{DS} \ge V_{GS}-V_T
\end{aligned}
$$

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

Solution to End Semester Examination

Monday	EE 671: VLSI Design	Time: 17:30-20:30
18-11-19	Autumn Semester 2019	Marks: 40

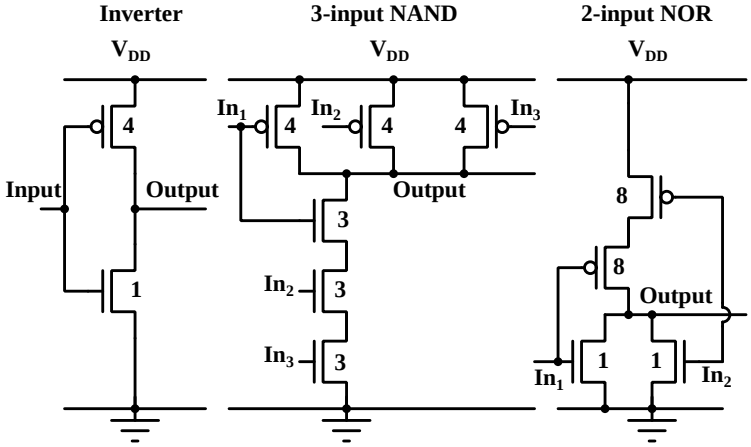
Q–1 A two step decoder uses 6 address lines and their complements and produces 64 select outputs as shown in the diagram on the right. Combinations of 1 line each out of $(A0, \overline{A0})$, $(A1, \overline{A1})$ and $(A2, \overline{A2})$ are fed to the left bank of 8 three-input NAND gates. Similarly, combinations of 1 line each out of $(A3, \overline{A3})$, $(A4, \overline{A4})$ and $(A5, \overline{A5})$ are fed to the right bank of 8 three-input NAND gates. 64 two-input NOR gates then accept one line each from the 8 outputs of the two banks of NANDs to produce 64 select lines.



The select lines are required to drive heavy loads equivalent to 512 minimum inverters each. Assume that the γ value representing the ratio of p channel widths to n channel widths in an inverter to produce equal rise and fall times is 4 and the parasitic inverter delay is 2.5.

a) What is the the parasitic delay and the logical effort of 3-input NAND gates and 2-input NOR gates? (Parasitic delay can be taken to be proportional to the total capacitive load at the output node for a gate providing equivalent drive to a minimum inverter).

Soln. 1-a) The inverter will have n width = 1, while the width of the p channel transistor will be $\gamma = 4$. Thus the input capacitance of the minimum inverter will be equivalent to 5 width units.



Since the 3 input NAND will have 3 n channel transistor in series, each will have a width of 3. The 3 p channel transistors will be in parallel and each will have the same width as the inverter p channel transistor, that is 4. Thus the input capacitance of the 3 input NAND will be $3 + 4 = 7$ in width units and will be 7/5 times the input capacitance of the minimum inverter. Thus, its logical effort will be $7/5 = 1.4$. Total width at the output node will be $3 + 3 \times 4 = 15$. Since the inverter has a total width of $1 + 4 = 5$ at the output node and this corresponds to a parasitic delay of 2.5, the parasitic delay of the 3 input NAND will be $15 \times 2.5/5 = 7.5$.

The 2 input NOR will have two n channel transistors in parallel, so each will have the same width as the n channel transistor in the minimum inverter. Thus, n width = 1 for the 2 input NOR. The two p channel transistors will be in series, so each should

have a width of $2 \times \gamma = 2 \times 4 = 8$. Thus the input capacitance of the 2 input NOR will be $1 + 8 = 9$ in width units and will be 9/5 times the input capacitance of the minimum inverter. Thus, its logical effort will be $9/5 = 1.8$.
Total width at the output node will be $2 \times 1 + 8 = 10$. Since the inverter with a total width of 5 at the output node has a parasitic delay of 2.5, the parasitic delay of the 2 input NOR will be $10 \times 2.5/5 = 5$. Thus:

Gate	Width of n Trans.	Width of p Trans.	Total W per input	Total W at output	Logical Effort	Parasitic Delay
Inverter	1	4	5	5	1	2.5
3-input-NAND	3	4	7	15	7/5	7.5
2-input-NOR	1	8	9	10	9/5	5.0

– **[1]**

b) What is the optimum number of stages (inclusive of added inverters if required) for minimum delay in this circuit, assuming that the inverters shown connected to the input address lines in the circuit above need to be minimum sized.

Soln. 1-b) Since $p_{inv} = 2.5$, The asymptotic stage ratio ρ is given by

$$p_{inv} + \rho(1 - \ln \rho) = 0 \quad \text{with } p_{inv} = 2.5$$

We define $f \equiv 2.5 + \rho(1 - \ln \rho)$. Then

$$f' = (1 - \ln \rho) + \rho\left(-\frac{1}{\rho}\right) = 1 - \ln \rho - 1 = -\ln \rho$$

Then starting with a guess value g for ρ, the next guess is given by Newton Raphson technique as

$$g - \frac{f}{f'} = g + \frac{2.5 + g(1 - \ln g)}{\ln g} = g + \frac{2.5 + g}{\ln g} - g = \frac{2.5 + g}{\ln g}$$

Starting with a guess value of 4, we iterate to get values for ρ as: 4.6888, 4.6524, 4.6523, 4.6523 .... Thus, $\rho = 4.6523$.

The path to the output as given in the question involves an inverter, a 3 input NAND and a 2 input NOR. Additional inverters may be added to reduce delay. Thus we have

$$G = g_{inv} \times g_{NAND3} \times g_{NOR2} = 1 \times 7/5 \times 9/5 = 63/25 = 2.52$$

Additional inverters will not change the value of G, since each of those will have a logical effort of 1.

On either side of the decoder, 6 inverters drive 24 inputs. (8 NAND gates with 3 inputs each). Therefore the branch factor at the output of the first inverter in the logic chain is $24/6 = 4$. The output of each NAND gate goes to 8 NOR gates. Therefore the branch factor for each of the 3-input NAND gates is 8. Thus $B = 4 \times 8 = 32$.
The output load is 512 inverters, while the input capacitance should be that of 1 inverter. Therefore $H = 512/1 = 512$.

$$F = GBH = \frac{63}{25} \times 32 \times 512 = 41287.68$$

$$\text{Optimum number of stages } N = \frac{\ln F}{\ln \rho} = \frac{10.62832}{1.537366} = 6.91333 \approx 7$$

Thus the optimum configuration will have 7 stages from the input to output. The logic chain as shown has an inverter, a 3-input NAND and a 2-input NOR. We should add 4 inverters to this logic chain for optimum delay. – **[3]**

c) Show the recommended configuration for the two step decoder with appropriately placed added inverters if necessary, so that the overall delay is minimized, and no transistor has (W/L) smaller than the n channel transistor in a minimum inverter. Compute the input and on-path output capacitance of each stage. 'Select' outputs can be positive TRUE or negative TRUE.
(Hint: extra inverters should be placed at locations such that no transistor in any gate has a width < 1.)

Soln. 1-c)

$$F = GBH = \frac{63}{25} \times 32 \times 512 = 41287.68$$

Number of stages = 7, So $\hat{f} = 41287.68^{1/7} = 4.5646$. For optimum delay, the product of gbh for each stage should be $\hat{f}$.

The minimum n transistor size is 1. Therefore the smallest possible 3 input NAND gate will have the three series connected n channel transistors with size = 1. The drive capacity of this gate will be 1/3 times that of the minimum inverter and the three parallel p channel transistors will have a size of 4/3. The input capacitance of this gate will be $1 + 4/3 = 7/3$ width units, or $7/15 = 0.46667$ times the input capacitance of the minimum inverter. (4 such gates will load the previous stage with the equivalent capacitance of $28/15 = 1.86667$ inverters).

Similarly, the smallest possible 2 input NOR gate will have the two parallel connected n channel transistors with size = 1. This will have the drive capacity of 1 minimum inverter and the p channel transistor size will be 8. The input capacitance of this gate will be $1 + 8 = 9$ width units, or $9/5 = 1.8$ times the input capacitance of the unit inverter. (8 such gates will load the previous stage with the equivalent capacitance of $72/5 = 14.4$ inverters).

It is convenient to start from the input and work our way towards the output to keep track of minimum transistor sizes. If at any stage we find that a stage cannot drive the total input capacitance of the next stage, we shall insert an even number of inverters there.

$$\hat{f} = gbh = gbC_{out}/C_{in} \quad \text{So } C_{out} = \frac{\hat{f}C_{in}}{gb}$$

The C_{out} value in this expression is the 'on path' capacitance of the next stage, since we have already divided by b.

Stage 1: inverter
This inverter is required to have $C_{in} = 1$. Let us check if it can drive 4 smallest possible NAND gates.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 1}{1 \times 4} = 1.141151$$

This is greater than the input capacitance of the smallest NAND (which is 0.46667), So this stage can drive 4 NAND gates in the second stage directly.

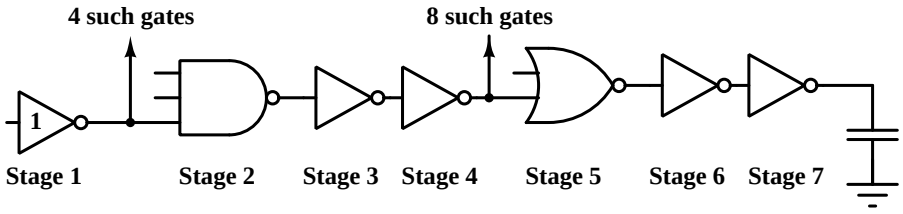
Stage 2: 3 input NAND gate
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. Let us check if with this size, it can drive 8 smallest NOR gates.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 1.141151}{1.4 \times 8} = 0.46508$$

This is less than the input capacitance of the smallest possible 2 input NOR gate (which is $9/5 = 1.8$). So this stage will not be able to drive 8 smallest possible NOR gates directly and we insert two inverters here. Now this stage is required to drive a single inverter and so the branch factor is just 1.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 1.141151}{1.4 \times 1} = 3.72064$$

Thus the configuration corresponding to this choice is:



Stage 3: Inverter
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. The next stage is also an inverter with branch factor of 1. Therefore,

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 3.72064}{1 \times 1} = 16.98325$$

Stage 4: Inverter
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. After buffering with these two inverters (stages 3 and 4), it will be possible to drive 8 NOR gates quite easily. Thus now we can support a branch factor of 8.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 16.98325}{1 \times 8} = 9.69022$$

This is well above the input capacitance of a minimum NOR (= 1.8). So the next stage can be 8 NOR gates.

Stage 5: 2 input NOR
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. It will now drive the first of the two remaining inverters. So the branch factor is just 1.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 9.69022}{1.8 \times 1} = 24.57334$$

Stage 6: Inverter
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. It will just drive the final inverter with a branch factor of 1

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 24.57334}{1 \times 1} = 112.1675$$

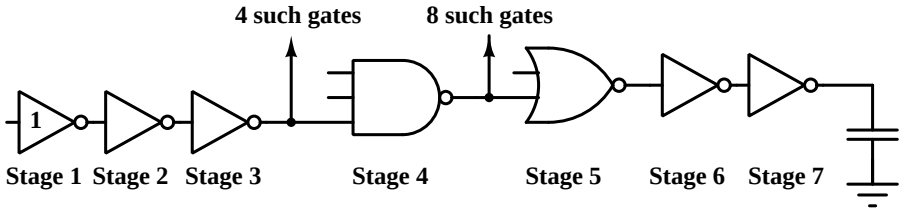
Stage 7: Inverter
C_{in} for this stage is the 'on-path' C_{out} of the previous stage. This stage will drive the final load.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 112.1675}{1 \times 1} = 512$$

This confirms that the design can indeed drive the required load.

Alternative Configuration

Multiple choices exist for placing inverters. The main design concern is the ability to drive 8 NOR gates by the hardware before this. So, some of the inverters should be placed before the NOR gates. To keep the logic intact, we may choose to insert inverters in pairs. Consider the case when we insert two inverters right after the first stage and the remaining two at the end. This configuration is shown below.



We can compute the sizing of all stages for this configuration.

Stage 1: inverter
This inverter is required to have $C_{in} = 1$. $g = 1, b = 1$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 1}{1 \times 1} = 4.5646$$

Stage 2: inverter
C_{in} for this inverter is the C_{out} of the previous one. $g = 1, b = 1$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 4.5646}{1 \times 1} = 20.83559$$

Stage 3: inverter
$C_{in} = 20.83559$, $g = 1, b = 4$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 20.83559}{1 \times 4} = 23.77655$$

This is quite adequate for driving the NAND gates.

Stage 4: 3-input NAND
This stage needs to drive 8 NOR gates. $C_{in} = 23.77655$, $g = 7/5 = 1.4, b = 8$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 23.77655}{1.4 \times 8} = 9.690221$$

This value of C_{out} should comfortably drive 8 NOR gates. (C_{in} for the smallest NOR is 1.8). Stage 5: 2-input NOR
$C_{in} = 9.690221$, $g = 9/5 = 1.8, b = 1$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 9.690221}{1.8 \times 1} = 24.57334$$

Stage 6: Inverter
$C_{in} = 24.57334$, $g = 1, b = 1$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 24.57334}{1 \times 1} = 112.1675$$

Stage 7: Inverter
$C_{in} = 112.1675$, $g = 1, b = 1$.

$$C_{out} = \frac{\hat{f}C_{in}}{gb} = \frac{4.5646 \times 112.1675}{1 \times 1} = 512$$

This shows that we can drive the final load properly.

Other configurations are also possible. For example, A single additional inverter at the input will still provide a selection of all address lines and their complements. So we could use 2 inverters, followed by 3 input NANDs, 2 input NORs and ending with three inverters. 'Select' lines will now be negative TRUE. – **[4]**

d) Compute the geometries for all transistors in the design. (All geometries are to be specified in units of width of the n channel transistor in the minimum inverter). Report the results in a neat table.

Soln. 1-d) The scale factors and geometries can be calculated from the input capacitance of each stage. The scale factor can be calculated as the ratio of actual C_{in} and the logical effort (which is the C_{in} value for the gate with a unit drive.) The n and p channel transistor geometries of the unit drive gate can then just be multiplied by the scale factor to give the actual geometries.
Thus, Scale Factor = C_{in}/g.
For inverters, nsize = Scale Factor $\times 1$, psize = Scale Factor $\times 4$.
For 3 input NANDs, nsize = Scale Factor $\times 3$, psize = Scale Factor $\times 4$.
For 2-input NORs, nsize = Scale Factor $\times 1$, psize = Scale Factor $\times 8$.

The following table summarizes the results for the first configuration:

Stage No.	Gate Type	C_{in}	g	Scale Factor	Size of	
					nMOS	pMOS
1	Inverter	1.0	1.0	1.00	1.00	4.00
2	3-in NAND	1.14115	1.4	0.8151	2.45	3.26
3	Inverter	3.72064	1.0	3.7206	3.72	14.88
4	Inverter	16.9833	1.0	16.9833	16.98	67.93
5	2-in NOR	9.69022	1.8	5.3835	5.38	43.07
6	Inverter	24.5733	1.0	24.5733	24.57	98.29
7	Inverter	112.168	1.0	112.168	112.17	448.67

Results for the second configuration are summarized in the table below:

Stage No.	Gate Type	C_{in}	g	Scale Factor	Size of	
					nMOS	pMOS
1	Inverter	1.0	1.0	1.00	1.00	4.00
2	Inverter	4.5646	1.0	4.5646	4.56	18.26
3	Inverter	20.8356	1.0	20.8356	20.84	83.34
4	3-in NAND	23.7766	1.4	16.9833	50.95	67.93
5	2-in NOR	9.69022	1.8	5.3835	5.38	43.07
6	Inverter	24.5733	1.0	24.5733	24.57	98.29
7	Inverter	112.168	1.0	112.168	112.17	448.67

– **[3]**

e) Compute the delay of each stage and the total delay for the decoder.

Soln. 1-e) The delay for each stage is just $\hat{f} + p = 4.5646 + p$. The parasitic delay is 2.5 for inverters, 5 for 2-input NOR and 7.5 for 3-input NAND.

Stage wise delay is tabulated below:

Configuration-1			
Stage No.	Gate Type	Par. Delay	Stage Delay
1	Inverter	2.5	7.0646
2	3-in NAND	7.5	12.0646
3	Inverter	2.5	7.0646
4	Inverter	2.5	7.0646
5	2-in NOR	5.0	9.5646
6	Inverter	2.5	7.0646
7	Inverter	2.5	7.0646
	Total	25.0	56.9522

Configuration-2			
Stage No.	Gate Type	Par. Delay	Stage Delay
1	Inverter	2.5	7.0646
2	Inverter	2.5	7.0646
3	Inverter	2.5	7.0646
4	3-in NAND	7.5	12.0646
5	2-in NOR	5.0	9.5646
6	Inverter	2.5	7.0646
7	Inverter	2.5	7.0646
	Total	25.0	56.9522

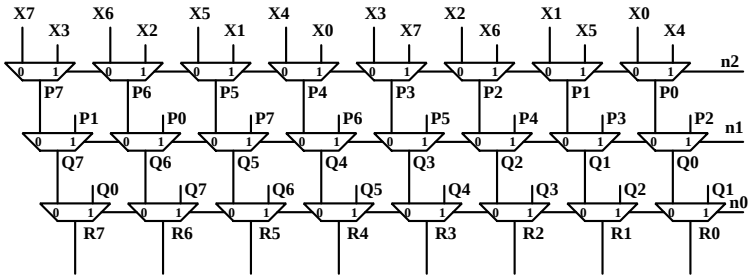
So the total delay is about 57 units in either case. – **[1]**

– **[Q1: 1+3+4+2+1 = 11 marks]**

Q–2 We want to design a logarithmic barrel shifter which can do right/left rotate/logical shift/arithmetic shift operations on 8 bit operands. Inputs to the circuit are an 8 bit operand, 3 bit shift/rotate amount and three single bit control inputs:
i) Dir ('1' for left, '0' for right), ii) Shift ('1' for shift, '0' for rotate) and iii) Arithmetic ('1' for arithmetic shifts, '0' for logical shifts). Outputs are the 8 bit result and carry.

a) Give the circuit for a logarithmic barrel shifter for performing right rotate operations on 8 bit operands using rows of 2-way multiplexers. (You need not draw all muxes. Indicate regular repetitions by $\cdots$ symbols.)

Soln. 2-a) To perform a right rotate operation, we use the scheme shown in in the figure below.



For an 8 bit operand, the amount of rotation will be between 0 to 7 places, which can be represented by 3 bits. Let the rotate amount be represented by bits $n_2n_1n_0$. n_2 controls whether the operand will be rotated right by 4 bits or by zero bit depending on whether n_2 is '1' or '0'. Similarly, n_1 controls whether the operand will be rotated right by 2 bits or by zero bit depending on whether n_1 is '1' or '0'. Finally, n_0 controls whether the operand will be rotated right by 1 bit or by zero bit depending on whether n_0 is '1' or '0'.

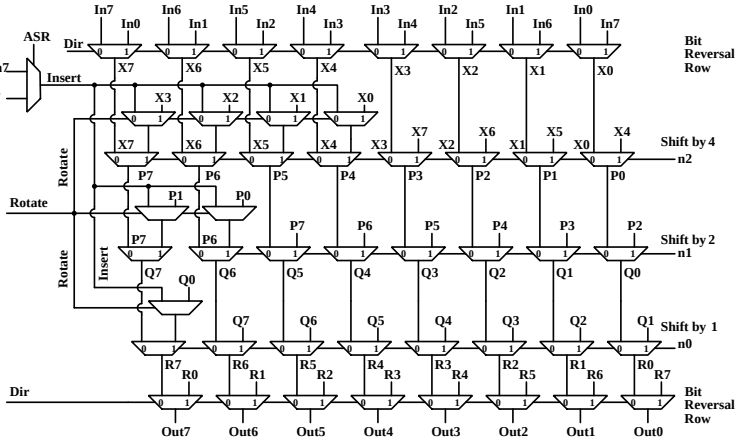
This is implemented through rows of eight 2-way muxes (one for each bit of the operand). The i'th mux in the row controlled by n_2 chooses between bit $(i+4)$ mod 8 and bit i, depending on whether n_2 is '1' or '0'. Similarly, the i'th mux in the row controlled by n_1 chooses between bit $(i+2)$ mod 8 and bit i, depending on whether n_2 is '1' or '0'. Finally, the i'th mux in the row controlled by n_0 chooses between bit $(i+1)$ mod 8 and bit i, depending on whether n_0 is '1' or '0'. Thus the top row rotates right by 4 bits or 0 and the next row further rotates the output of top row by 2 bits of 0. Finally the third row rotates the output of second row by 1 bit or 0. The cumulative right rotation is thus by $n_2n_1n_0$ bits. – **[3]**

b) Show how by inserting additional 2-way muxes in the circuit above, we can perform logical shift/arithmetic shift/rotate operations in the right or left direction.

Additional inputs to this circuit are three single bit control inputs: i) Dir ('1' for left, '0' for right), ii) rotate ('1' for rotate, '0' for shift) and iii) ASR ('1' for arithmetic shifts, '0' for logical shifts).
(Rows of the 8 bit rotate circuit above need not be re-drawn in detail).

Soln. 2-b) A right rotate can be converted to a right logical shift by forcing the inputs to the top 4 bits of the top row of the rotate circuit, the top 2 bits of the middle row and the top bit of the last row to '0'.

Further, we can switch between Arithmetic and Logical right shift by choosing the bit to be forced at these locations to be the most significant bit or '0'. This can be accomplished by an additional mux as shown in the figure below.



'ASR' chooses the bit to insert to be In7 or '0'. Using 4 muxes in the top row, 2 in the middle row and 1 in the bottom row, we can replace the input bits at these positions by the bit chosen by 'ASR' if the control signal 'Rotate' is '0', implying that a shift rather than rotate is desired. (Otherwise, the same input as was shown for the right rotate circuit is presented to these muxes).

Finally, we can convert right shift/rotate operations to left shift/rotate by reversing the bits before shift/rotate and reversing these again after the shift/rotate operation. This is shown in the diagram above through the additional bit reversal rows at the top and bottom of the structure. These rows are controlled by the signal 'Dir', which reversed the bit order if it is '1' and leaves the operand unchanged if it is '0'. – **[3]**

– [Q2: 3+3 = 6 marks]

Q–3 a) Describe the Dadda scheme for wire reduction in a multiplier. What are the advantages and disadvantages of using this scheme compared to the Wallace scheme in a multiplier?

Soln. 3-a) Both Dadda and Wallace schemes carry out the following three steps:

1. Create partial product bits using an array of ANDs.
2. Using full and half adders, successively reduce the number of wires at any place value. This is continued till we have no more than 2 wires at any position.
3. allocate one wire from each position to one register and the other wire to the other register and add these registers using the fastest adder available.

The two schemes differ from each other in the way they reduce wires.

Wire reduction stops when we have no more than 2 wires at any place value. Thus the maximum no. of wires at any position in the last stage is 2. The stage prior to that can have $int(1.5 \times 2)$ or 3 wires. The one before that can have $int(1.5 \times 3)$ or 4 wires, and the one before this one can have $int(1.5 \times 4)$ or 6 wires. We continue this process till we reach a number which is larger than or equal to the maximum number

of partial product bits at any position. We identify this as the starting stage. The numbers obtained by successive multiplication by 1.5 and taking the integer part are called the "capacity" of a reduction stage.

In Dadda reduction scheme, we reduce the wires as late as possible. If the anticipated number of wires at any place value (inclusive of carries expected from the less significant position) at the next stage is less than or equal to the capacity of the next stage, we just pass through the wires to the next stage. If the number of wires exceeds the capacity, we reduce these by placing half and full adders. A half adder takes two input wires and produces a sum at the same place value and a carry at the next place value. Thus it reduces the number of wires at the current place value by 1 and increases the number of wires at the next value by 1. Similarly, a full adder takes 3 wires and outputs a sum at the same place value and a carry to the next higher place value. Thus it reduces the wires at the current place value by 2 and adds a carry wire to the next higher place value.

In Dadda scheme of wire reduction, we use the smallest number of smallest adders which will reduce the number of wires at the current position to the capacity of the stage. Any left over wires not going to these adders are passed through to the next stage.

This operation is repeated stage by stage till we have no more than two wires at any place value. These two constitute the final addition to be performed after wire reduction. – **[2]**

b) A Multiply and Accumulate circuit with a 10 bit multiplicand, a 6 bit multiplier and a 16 bit accumulator needs to be designed using the Dadda scheme for wire reduction with full and half adders. Show the number of wires and the number of full and half adders to be used at each stage of reduction.

Soln. 3-b) The multiplier itself gives partial products as shown below:

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wires	0	1	2	3	4	5	6	6	6	6	6	5	4	3	2	1

The accumulator provides an additional wire at each position. Therefore the wire distribution becomes:

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wires	1	2	3	4	5	6	7	7	7	7	7	6	5	4	3	2

The maximum number of wires at any place value is 7 here.
The stage capacity sequence (obtained by repeated multiplication by 1.5 and retention of just the integer part) is: 2, 3, 4, 6, 9 $\cdots$. We identify the starting number of wires with a stage capacity of 9. So the capacity of the next stage is 6

We can now reduce these wires according to the Dadda scheme described above.

Stage 1. Next stage capacity = 6																
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In wires	1	2	3	4	5	6	7	7	7	7	7	6	5	4	3	2
Full Adders	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
Half Adders	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
Pass through	1	2	3	4	5	4	4	4	4	4	5	6	5	4	3	2
Out wires	1	2	3	4	6	6	6	6	6	6	6	6	5	4	3	2

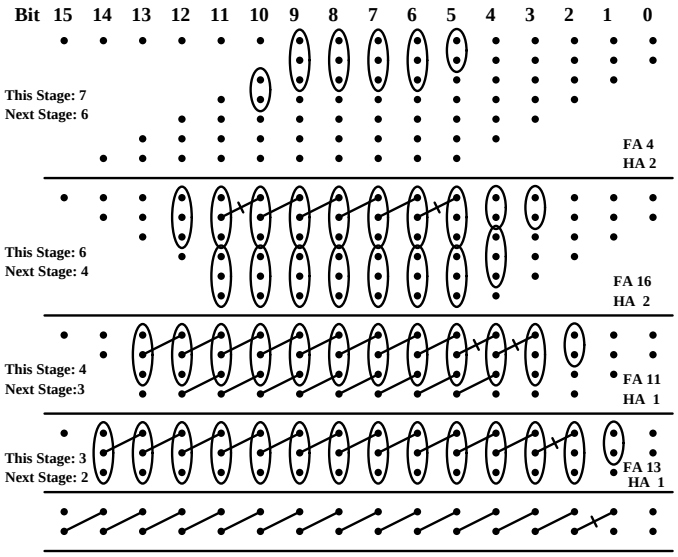
Stage 2. Next stage capacity = 4																
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In wires	1	2	3	4	6	6	6	6	6	6	6	6	5	4	3	2
Full Adders	0	0	0	1	2	2	2	2	2	2	2	1	0	0	0	0
Half Adders	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
Pass through	1	2	3	1	0	0	0	0	0	0	0	1	3	4	3	2
Out wires	1	2	4	4	4	4	4	4	4	4	4	4	4	4	3	2

Stage 3. Next stage capacity = 3																
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In wires	1	2	4	4	4	4	4	4	4	4	4	4	4	4	3	2
Full Adders	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Half Adders	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Pass through	1	2	1	1	1	1	1	1	1	1	1	1	1	2	3	2
Out wires	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2

Stage 4. Next stage capacity = 2																
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
In wires	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2
Full Adders	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
Half Adders	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Pass through	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
Out wires	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

As can be seen from the tables, the first stage uses 4 Full adders and 2 Half Adders, the second stage uses 16 Full Adders and 2 Half Adders, the third stage uses 11 Full adders and 1 Half adder and the last stage uses 13 Full adders and 1 Half adder. In all, 44 Full adders and 6 Half adders are used.

The following figure shows the same reduction using a dot diagram.



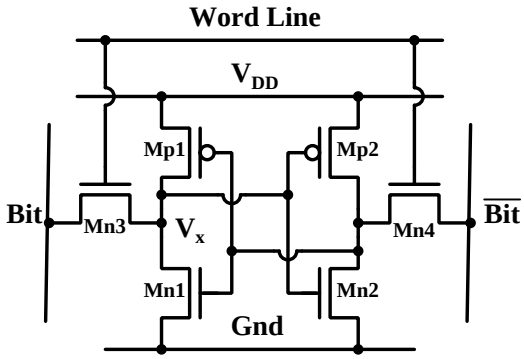
– **[3]**

– **[Q3: 2+3 = 5 marks]**

Q–4 A static RAM uses a 6 transistor cell as shown below on the right.

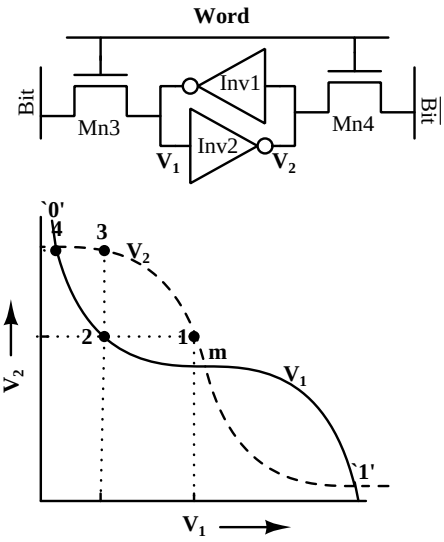
a)

Explain the butterfly diagram used for describing the behaviour of the cross connected inverters in the cell. Identify the '1', '0' and the meta-stable points on the butterfly diagram. Show graphically that if one deviates by even a small amount from the meta-stability point towards the '0' point, the circuit will end up at '0'.



Soln. 4-a) The butterfly diagram is the combined transfer curve of the two inverters in the 6T RAM cell.

The dotted line is the conventional transfer characteristic of Inv2 formed by Mp2 and Mn2, with its input plotted along the X axis and the output along the Y axis. The output of this inverter drives Inv1 formed by Mp1, Mn1. Since the output of Inv2 is the same as the input of Inv1, we use the Y axis as the input axis of inverter Inv1 and plot its output along the X axis using a solid line. All points on the dotted curve satisfy the static conditions for Inv2 – that is currents through Mp2 and Mn2 are equal and therefore there is no output current from Inv2. Similarly, for all points on the solid line curve, currents through Mp1 and Mn1 are equal and there is no output current from Inv1.



This means that for all input-output voltage pairs lying on the dotted curve, there is no output current from Inv2 and hence, $dV2/dt$ is zero. Similarly, for all input-output voltage pairs lying on the solid line curve, there is no output current from Inv1 and hence, $dV1/dt$ is zero.

The points of intersection of these curves lie on both curves and therefore, satisfy both conditions. For these points, V2 as well as V1 are constant with time and hence these represent the solutions of circuit equations.

The intersection point on the left corresponds to low V1 and High V2, so Bit = '0' and $\overline{\text{Bit}}$ = '1'. This represents a '0' stored in the memory. The intersection point on the right corresponds to high V1 and low V2. Here Bit = '1' and $\overline{\text{Bit}}$ = '0'. This means a '1' is stored in the memory.

The middle intersection point represents unstable equilibrium and is called the meta-stable point. This is because any small deviation from this solution results in the solution moving further and further away from this point till it settles either at '0' point or at '1' point.

Take a value of V1 just to the left of the meta-stable point. The output V2 is shown with a dot marked 1. This now becomes the input of Inv1, Since this point is not on the solid line curve, the output of Inv1 must change to a point on the transfer curve with this as its input. So the output goes to a point shown with a dot marked as 2 on the solid line curve. Now this becomes the new value of V1. Since this point is not on the dotted curve, the output of Inv2 should change to a point on the dotted line curve

with this new input. The output V2 of Inv2 is shown by the dot marked 3. This then becomes the input of Inv2, whose output is dot marked 4 on the dotted curve and so on till we reach the point of intersection of the two curves on the left. As this point lies on both characteristics, this is the solution to the circuit equations. Thus if we perturb the input voltage by a minute amount from the meta-stable point to the left, the solution moves to the Bit=0 point.

On the other hand, a small deviation from the Bit = '0' point comes back to the same point and similarly, a small deviation from the Bit = '1' point also returns to the same point, Hence these two are stable solutions. – **[2]**

b) describe the sequence of events that take place during the read and write cycles for the static RAM. Show the data path between the storage cell and the data pin for these operations. (Pass gates in the data path should be shown along with their control signals).

Soln. 4-b) During the read cycle, the following actions take place.

1. The address is placed by the processor on the address lines.
2. The row and column address are decoded.
3. Bit and $\overline{\text{Bit}}$ lines are pre-charged.
4. Word line for the selected row is pulled HIGH.
5. The sense amplifier of the selected column is activated.
6. After the time required to discharge one of the Bit and $\overline{\text{Bit}}$ lines, the Word line is brought low again.
7. The sense amplifier determines which of the Bit and $\overline{\text{Bit}}$ lines is lower and outputs a '0' or a '1' accordingly.
8. This digital data is then buffered to the output pad by a tristateable driver, which is activated only during read cycles.

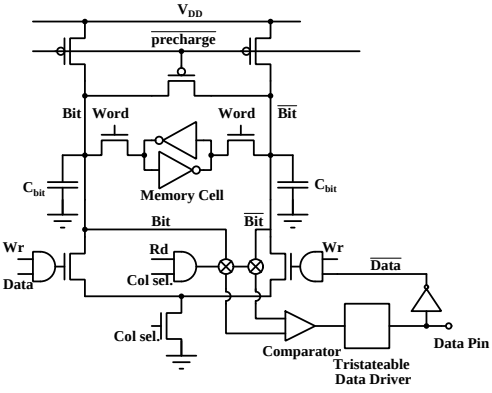
Similarly during the write cycle,

1. The address is placed by the processor on the address lines.
2. The row and column address are decoded.
3. When $\overline{\text{Wr}}$ is asserted, the Bit and $\overline{\text{Bit}}$ lines are driven to data and $\overline{\text{data}}$ respectively in the selected column.
4. Both Bit and $\overline{\text{Bit}}$ lines are driven High in the columns which have not been selected.
5. When the word line goes HIGH, which ever of the Bit and $\overline{\text{Bit}}$ lines is LOW, discharges the node connected to it through the pass transistors MN3/MN4. Once the voltage at this node goes below the meta-stability point, the positive feed back ensures that it will go all the way to '0'. Thus the state of Bit and $\overline{\text{Bit}}$ lines is copied to the RAM cell in the selected column.
6. After the write process is complete, the Word line is brought low again.

The data path during read and write is shown below on the right.

Bit and $\overline{\text{Bit}}$ lines are pulled up by the Precharge signal, which is activated during read as well as write for all columns. The shorting transistor across the two lines ensures that the lines are at the same potential after precharge, so that small differences can be detected during read.

The pass gates connected to Bit and $\overline{\text{Bit}}$ lines are activated only if this is a read cycle and this column has been selected. When activated, these feed the sense amplifier/comparator, which then drives a tri-stateable buffer to the data pin.

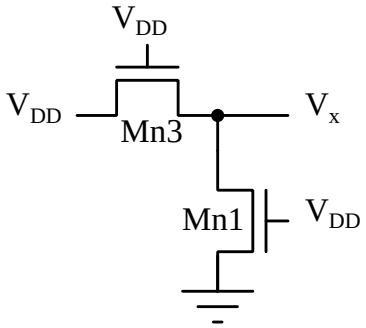


The output data buffer is enabled only when this is a read cycle and this column has been selected. During the write cycle, the pull down transistors are enabled only when this is a write cycle and the column has been selected. – **[2]**

c) An SRAM cell has a '0' stored in it. (Mn1, Mp2 are 'ON', Mn2, Mp1 are 'OFF'). During the read cycle, all the Bit and $\overline{\text{Bit}}$ lines will be pre-charged to V_{DD} and the word line for the selected row will be raised to V_{DD}. Let the ratio of conductance factors for Mn1 and Mn3 (K_{Mn1}/K_{Mn3}) be β.
Find the minimum value of β in terms of V_{DD} and V_{Tn} such that the voltage rise at the junction of the drains of Mn1, Mp1 and the source of Mn3 (V_x) remains below V_{Tn} during the read cycle.
Evaluate this value of β if $V_{Tn} = V_{DD}/5$.

Soln. 4-c) During the read cycle, both Bit and $\overline{\text{Bit}}$ lines are pre-charged to V_{DD}. The word line is also raised to V_{DD}.

The equivalent circuit of the memory cell during a read cycle when a '0' is stored in the cell is shown on the right. MN2 and MP1 are OFF. Since we are solving for the case when V_x remains below V_{Tn}, Mn2 will remain OFF during the read cycle. Thus the voltage at the gate of Mn1 will remain at V_{DD} for the entire cycle.



Since the drain and gate voltages of Mn3 are equal (and at V_{DD}), it is in saturation. The gate of Mn1 is at V_{DD} while its drain is below V_{Tn}, so Mn2 is in linear regime. Equating current through the two,

$$\frac{K_{n3}}{2}\,(V_{DD} - V_x - V_{Tn})^2 = K_{n1}\left((V_{DD} - V_{Tn})V_x - \frac{1}{2}V_x^2\right)$$

$$\text{So} \quad (V_{DD} - V_x - V_{Tn})^2 = 2\beta V_x(V_{DD} - V_{Tn} - \frac{1}{2}V_x)$$

In the limiting case, $V_x = V_{Tn}$. This gives

$$(V_{DD} - 2V_{Tn})^2 = 2\beta V_{Tn}(V_{DD} - \frac{3}{2}V_{Tn}) = \beta V_{Tn}(2V_{DD} - 3V_{Tn})$$

$$\text{Hence,} \quad \beta = \frac{(V_{DD} - 2V_{Tn})^2}{V_{Tn}(2V_{DD} - 3V_{Tn})}$$

If $V_{Tn} = V_{DD}/5$, we get

$$\beta = \frac{(5V_{Tn} - 2V_{Tn})^2}{V_{Tn}(10V_{Tn} - 3V_{Tn})} = \frac{9V_{Tn}^2}{7}V_{Tn}^2 = 9/7 = 1.2857$$

Thus the value of β should be ≥ 1.29 to keep V_x below V_{Tn} – **[3]**

d) When the word line goes high during the write cycle, the pass transistors (Mn3 and Mn4) of *all* cells in this row will be turned on. How does the data stored in those cells of the selected row whose column has *not* been selected for writing, remain unchanged?

Soln. 4-d) A write is performed on the selected column by driving Bit line to data and $\overline{\text{Bit}}$ line to $\overline{\text{data}}$. The other columns in this row have both Bit and $\overline{\text{Bit}}$ lines pulled up and therefore go through a read like cycle.

Aspect ratios of pass transistors and pull down transistors are calculated carefully (as in the problem above) such that the voltage rise at the 'low' node does not rise sufficiently to cause a read upset. With proper ratio selection for pass transistors and pull down transistors as seen in the problem above, the stored data retains the value which was originally stored in the cell. – **[1]**

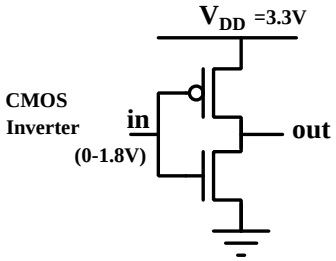
– **[Q4: 2+2+3+1 = 8 marks]**

Q–5 a) An IC operates with $V_{DD} = 3.3$V. However, it receives its digital inputs from a source which has a swing of 0 to 1.8V. The IC has access to an internally generated V_{DDL} supply of 1.8V.

Why is a CMOS inverter at the input node not a good choice to amplify this swing to 0 to 3.3V? Show why a CVSL logic buffer is a better choice for bringing up the input swing to the required level. Give a complete transistor level schematic for the CVSL circuit, showing connections to V_{DD} and V_{DDL} clearly.

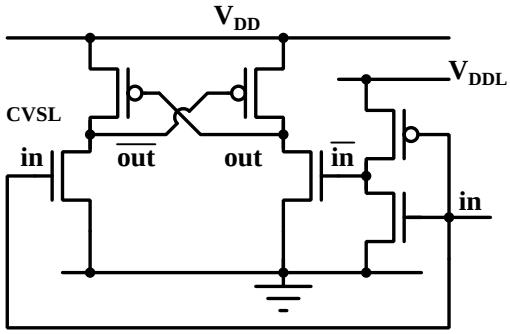
Soln. 5-a) input swing: 0 to 1.8V, $V_{DD} = 3.3$V.

If we connect an input swing of 0 to 1.8V to the input of a CMOS inverter with $V_{DD} = 3.3$V, even the most positive input will take the pMOS gate to 1.8V, which is negative with respect to V_{DD} by 1.5V. As a result, the pMOS can never be OFF. This results in considerable static power consumption.



Since the problem is with pMOS, we could consider a pseudo nMOS input buffer which has the pMOS gate permanently grounded. Unfortunately, here also the pMOS transistor is continuously ON and static power is consumed.

A CVSL input buffer solves this problem. Low swing inputs go only to nMOS transistors, which can be turned on easily by the 1.8V input. The pMOS gates are driven by the full swing output, so there is no static power consumption.

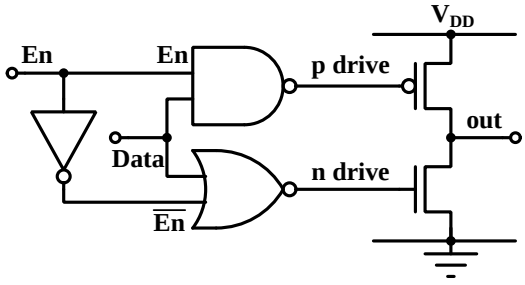


We use a low voltage CMOS inverter using $V_{DDL} = 1.8V$ as the supply. The input swing is adequate for this supply voltage. Now we have the input in true and complement form, but with low swing. These two low swing signals drive the nMOS transistors of the two branches of CVSL. The CVSL stage is fed by the full supply voltage. Therefore the swing at the output is 3.3V, which can drive the pMOS adequately. – **[2]**

b) A tri-stateable buffer is needed for driving a bi-direction pad. Why is the traditional 4 transistor tri-stateable inverter not a good choice for this task? Show how a two transistor output stage with a NAND-NOR driver can be used for a bi-directional pad.

Soln. 5-b) Because the output driver drives heavy loads, the drive transistors are large in size. The traditional 4 transistor tri-stateable driver uses two nMOS and two pMOS transistors in series. Because of the series connection, transistor sizes have to be doubled. This is a big overhead in area because of the large size of transistors. To avoid the series connection overhead, we use the configuration shown below on the right.

This is an inverter like configuration in which the p and n channel driver transistors are driven independently. The drive to p and n channel transistors is applied through logic which ensures that the output driver will be tri-stated when the Enable signal En is '0', and will drive the output to the data value when En is '1'.

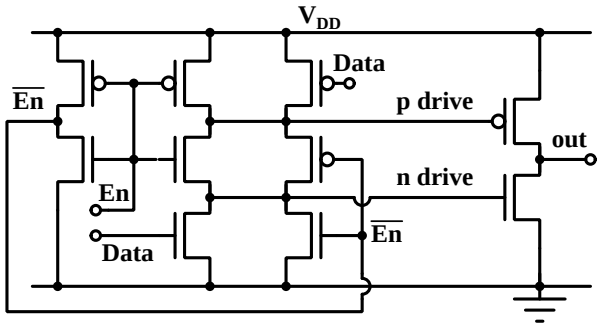


When the enable signal En is '0', the output of NAND (p drive) is forced to '1' so the p channel driver transistor is OFF. The NOR gate uses $\overline{\text{En}}$ as its input. When enable is '0', this input is '1', which forces the output of NOR gate to'0'. Therefore the n channel final driver is also OFF. Thus the circuit is tri-stated when En is '0'.

When En is '1', $\overline{\text{En}}$ is '0'. Now the NAND has '1' and Data as its inputs, while NOR has '0' and Data as its inputs. Thus NAND as well as NOR drive their outputs to $\overline{\text{Data}}$. So the output driver acts as an inverter with $\overline{\text{Data}}$ applied to its input and thus drives the output to the data value. – **[2]**

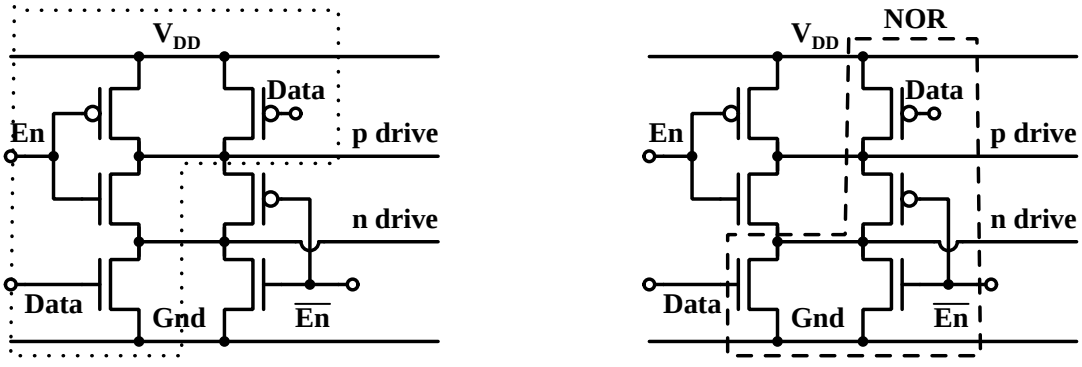
c) The discrete NAND and NOR gates in the driver above can be merged into a compact driver with the same functionality. Give a transistor level circuit for this driver and describe its functioning. Show a circuit level equivalence of parts of this circuit with traditional NAND and NOR logic gates.

Soln. 5-c) The NAND-NOR logic shown above can be combined into a single circuit. An inverter is used to generate $\overline{\text{En}}$ from En. The combined circuit is shown just to the right of the inverter.



The middle n channel transistor receives En as its input while the middle p channel transistor is driven by $\overline{\text{En}}$. When En is '0', both of these transistors are OFF and p drive is isolated from n drive. The upper p channel transistor driven by En is ON and pulls the p drive to '1', turning the output p driver OFF. Similarly, the lower n channel driven by $\overline{\text{En}}$ is also ON and pulls n drive to '0', turning the output n transistor OFF. Thus the output is tri-stated when En is '0'.
When En is '1', both of the middle transistors are ON, shorting the p drive with n drive. The upper p transistor driven by En and the lower n transistor driven by $\overline{\text{En}}$ are OFF. So both p drive and n drive are driven to $\overline{\text{Data}}$. The output stage acts like an inverter and drives the output to the Data value.

The equivalence of the compact circuit to NAND and NOR structures can be seen by paying attention to the sub-circuits enclosed by dotted lines in the figure below.



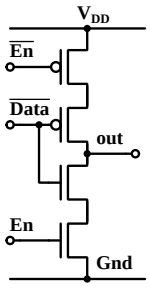
– **[2]**

d) Why is careful delay matching required for the 'p drive' and 'n drive' outputs of the NAND/NOR circuit? Why is the timing of enable and $\overline{\text{enable}}$ signals for the 4 transistor tri-stateable inverter not so critical?

Soln. 5-d) As can be seen in the circuit above, the wide p and n channel transistors are connected right across the power supply. If there is a timing mismatch between p and n drive, both transistors can be ON for a short time, drawing huge current.

If a 4 transistor tristateable inverter is used as the output driver, we do have the disadvantage of having to double the size of all transistors.

However, this circuit is more robust against timing mismatch between En and $\overline{\text{En}}$ signals. This is because one of the two transistors driven by $\overline{\text{Data}}$ is always OFF. So even if both the top and bottom transistor are turned on due to a timing mismatch, heavy current will not flow from the supply to ground.



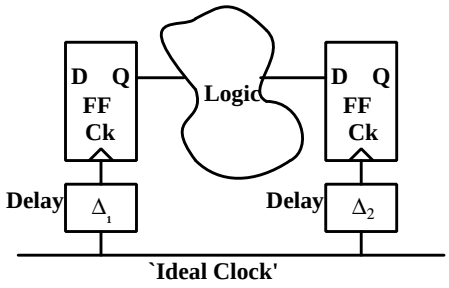
– **[1]**

– **[Q5: 2+2+2+1 = 7 marks]**

Q–6 Show how the fastest clock speed at which we can operate a synchronous digital circuit depends on the skew and jitter of the clock. Briefly describe the various techniques used for distributing the clock over a large chip with minimal skew.

Soln. 6) Consider the model of a stage of a synchronous circuit.

Let the delay for the clock signal to arrive at the input flipflop be Δ_1 while the delay for the output flopflop is Δ_2. In a practical case, these two will not be equal due to skew and jitter. In the worst case for clock period, Δ_1 will be maximum (so that the data is applied late to the logic) and Δ_2 will be minimum, (so that the result is latched early by the output flipflop).



For proper operation of the circuit, the result should be ready at least a set up time earlier than the arrival of the clock at the second flipflop. If the ideal clock ticks at zero time,

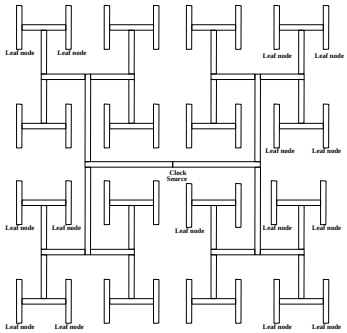
$$\Delta_1 + \text{Ck-to-Q} + \text{logic}_{\max} + T_{setup} \leq T_{Ck} + \Delta_2$$

$$\text{Which leads to} \quad T_{Ck} \geq (\Delta_1 - \Delta_2) + \text{Ck-to-Q} + \text{logic}_{\max} + T_{setup}$$

Thus the clock skew and jitter, which decides $\Delta_1 - \Delta_2$, directly adds to the minimum clock period we can have for a given logic design and technology. It is therefore important that clock distribution should ensure minimum skew and jitter.

Methods for minimizing clock skew include:
H-tree clock distribution:

In this scheme, clock routing is symmetric around the origin of the clock. At each point the clock bifurcates into two symmetric branches. This tree is extended till it covers the entire chip. Clock feed is taken only from the leaf nodes of this tree, all of which are at equal distance from the clock source. This equalizes the delay for clock arrival at all points and hence minimizes skew.



A similar arrangement where there is a 4 way diagonal branching at each point is possible. It is known as the X tree.

Clock Grid:
A different approach for clock distribution involves driving a grid from all end points in parallel. This tends to average out the skew of the clock drivers and the grid as a whole has very little skew.

– **[Q6: 3 marks]**

Paper Ends.

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

Mid Semester Examination

Wednesday	EE 671: VLSI Design	Time: 1330-1530
18-09-19	Autumn Semester 2019	Marks: 25

For iterative solutions, *all* intermediate values must be reported.
Quantitative answers must be accurate to 0.1% at least.

Q–1 The output of a CMOS static inverter is given by:

$$V_{out} = V_{in} + V_{Tp} + \sqrt{(V_{DD} - V_{in} - V_{Tp})^2 - \beta(V_{in} - V_{Tn})^2} \quad \text{for} \quad \text{Low input voltage}$$

$$V_{out} = V_{in} - V_{Tn} - \sqrt{(V_{in} - V_{Tn})^2 - \frac{(V_{DD} - V_{in} - V_{Tp})^2}{\beta}} \quad \text{for} \quad \text{High input voltage}$$

a) Justify the choice of points on the transfer characteristics of an inverter where the slope of V_{out} versus V_{in} is -1, to define the input and output logic 'Low' and 'High' levels.

Soln. 1-a) The requirement from a digital circuit is that it should clearly distinguish logic levels and at the same time, be insensitive to the exact analog voltage at the input as long as it is within the definition of being a proper '0' or '1' value. Therefore, the rate of change of the output with respect to the input should be low for input voltages in valid digital ranges.

The slope $\mathrm{d}V_{out}/\mathrm{d}V_{in}$ represents the analog voltage gain of a stage. Its magnitude should be ≤ 1 to ensure that noise riding on digital values does not get amplified by the chain of logic gates that it traverses. We select two points on the transfer curve where the slope $(\frac{\mathrm{d}V_{out}}{\mathrm{d}V_{in}})$ is -1.0.

The coordinates of these two points define the values of (V_{iL}, V_{oH}) and (V_{iH}, V_{oL}). The magnitude of the slope for $0 \leq V_{in} \leq V_{iL}$ and for $V_{iH} \leq V_{in} \leq V_{DD}$ is now ≤ 1 and hence meets our requirements. – **[1]**

b) Evaluate the value of V_{iL}, V_{0L}, V_{iH} and V_{oH} in terms of V_{DD}, V_{Tn}, V_{Tp} using the equations given above, assuming $\beta = 1$.

Soln. 1-b) For low input value, we have

$$\begin{aligned} V_{out} &= V_{in} + V_{Tp} + \sqrt{(V_{DD} - V_{in} - V_{Tp})^2 - \beta(V_{in} - V_{Tn})^2} \\ &= V_{in} + V_{Tp} + \sqrt{(V_{DD} - V_{Tn} - V_{Tp})(V_{DD} - 2V_{in} + V_{Tn} - V_{Tp})} \end{aligned}$$

Let $V_1 \equiv V_{DD} - V_{Tn} - V_{Tp}$. Taking the derivative with respect to V_{in} and putting it equal to -1,

$$\begin{aligned} -1 &= 1 + \sqrt{V_1}\ (1/2)(V_{DD} - 2V_{iL} + V_{Tn} - V_{Tp})^{-1/2}(-2) \\ -2 &= -\sqrt{\frac{V_{DD} - V_{Tn} - V_{Tp}}{V_{DD} - 2V_{iL} + V_{Tn} - V_{Tp}}} \\ \text{So} \quad 4 &= \frac{V_{DD} - V_{Tn} - V_{Tp}}{V_{DD} - 2V_{iL} + V_{Tn} - V_{Tp}} \end{aligned}$$

This leads to: $4V_{DD} - 8V_{iL} + 4V_{Tn} - 4V_{Tp} = V_{DD} - V_{Tn} - V_{Tp}$

$$\text{So} \quad V_{iL} = \frac{3V_{DD} + 5V_{Tn} - 3V_{Tp}}{8}$$

Substituting this in the expression for V_{out}, we get

$$\begin{aligned} V_{oH} &= \frac{3V_{DD}+5V_{Tn}-3V_{Tp}}{8} + V_{Tp} + \\ &\quad \sqrt{V_1\left(V_{DD}+V_{Tn}-V_{Tp}-\frac{3V_{DD}+5V_{Tn}-3V_{Tp}}{4}\right)} \\ &= \frac{3V_{DD}+5V_{Tn}+5V_{Tp}}{8} + \sqrt{(V_{DD}-V_{Tn}-V_{Tp})\frac{V_{DD}-V_{Tn}-V_{Tp}}{4}} \\ &= \frac{3V_{DD}+5V_{Tn}+5V_{Tp}}{8} + \frac{V_{DD}-V_{Tn}-V_{Tp}}{2} \\ &= \frac{7V_{DD}+V_{Tn}+V_{Tp}}{8} = V_{DD} - \frac{V_{DD}-V_{Tn}-V_{Tp}}{8} \end{aligned}$$

When the input voltage is 'High', we have

$$\begin{aligned} V_{out} &= V_{in} - V_{Tn} - \sqrt{(V_{in}-V_{Tn})^2 - (V_{DD}-V_{in}-V_{Tp})^2} \\ &= V_{in} - V_{Tn} - \sqrt{(V_{DD}-V_{Tn}-V_{Tp})(2V_{in}-V_{DD}-V_{Tn}+V_{Tp})} \end{aligned}$$

We again define $V_1 \equiv V_{DD} - V_{Tn} - V_{Tp}$. Then

$$V_{out} = V_{in} - V_{Tn} - \sqrt{V_1(2V_{in}-V_{DD}-V_{Tn}+V_{Tp})}$$

Taking the derivative of V_{out} with respect to V_{in} and putting it equal to -1 gives:

$$\begin{aligned} -1 &= 1 - \sqrt{V_1}\ (1/2)(2V_{iH}-V_{DD}-V_{Tn}+V_{Tp})^{-1/2}(2) \\ \text{So} \quad -2 &= -\sqrt{\frac{V_{DD}-V_{Tn}-V_{Tp}}{2V_{iH}-V_{DD}-V_{Tn}+V_{Tp}}} \\ \text{Or} \quad 4 &= \frac{V_{DD}-V_{Tn}-V_{Tp}}{2V_{iH}-V_{DD}-V_{Tn}+V_{Tp}} \end{aligned}$$

$$\text{Thus} \quad 8V_{iH} - 4V_{DD} - 4V_{Tn} + 4V_{Tp} = V_{DD} - V_{Tn} - V_{Tp}$$

$$\text{So} \quad V_{iH} = \frac{5V_{DD}+3V_{Tn}-5V_{Tp}}{8}$$

Substituting in the expression for V_{out}, we get

$$\begin{aligned} V_{oL} &= \frac{5V_{DD}+3V_{Tn}-5V_{Tp}}{8} - V_{Tn} - \sqrt{V_1\left(\frac{5V_{DD}+3V_{Tn}-5V_{Tp}}{4} - V_{DD} - V_{Tn} + V_{Tp}\right)} \\ &= \frac{5V_{DD}-5V_{Tn}-5V_{Tp}}{8} - \sqrt{V_1\left(\frac{V_{DD}-V_{Tn}-V_{Tp}}{4}\right)} \\ &= \frac{5}{8}(V_{DD}-V_{Tn}-V_{Tp}) - \frac{1}{2}(V_{DD}-V_{Tn}-V_{Tp}) = \frac{V_{DD}-V_{Tn}-V_{Tp}}{8} \end{aligned}$$

Thus we have

$$V_{iL} = \frac{3V_{DD}+5V_{Tn}-3V_{Tp}}{8} \qquad V_{oH} = V_{DD} - \frac{V_{DD}-V_{Tn}-V_{Tp}}{8}$$

$$V_{iH} = \frac{5V_{DD}+3V_{Tn}-5V_{Tp}}{8} \qquad V_{oL} = \frac{V_{DD}-V_{Tn}-V_{Tp}}{8}$$

– **[6]**

c) Compare the voltage difference between V_{DD} and V_{oH} with the voltage difference between V_{oL} and ground for $\beta = 1, V_{Tn} \neq V_{Tp}$.

Soln. 1-c) From the values of V_{oH} and V_{oL} derived above, we can see that

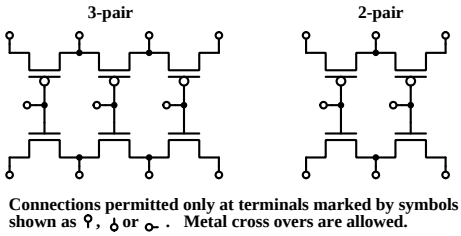
$$V_{DD} - V_{oH} = V_{oL} - 0 = \frac{V_{DD}-V_{Tn}-V_{Tp}}{8}$$

in spite of having $V_{Tn} \neq V_{Tp}$, when $\beta = 1$. – **[1]**

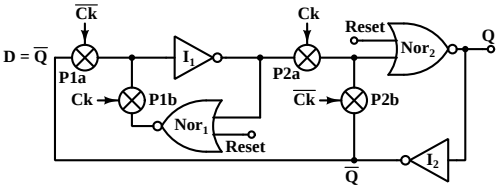
– **[Q1: 1+6+1 = 8 marks]**

Q–2 A toggle flip flop (which changes its output at every positive clock edge) can be constructed by feeding the $\overline{Q}$ output of a D flip flop to its D input.
Give the logic diagram for a toggle flip flop with asynchronous reset, using NOR gates, inverters and pass gates.

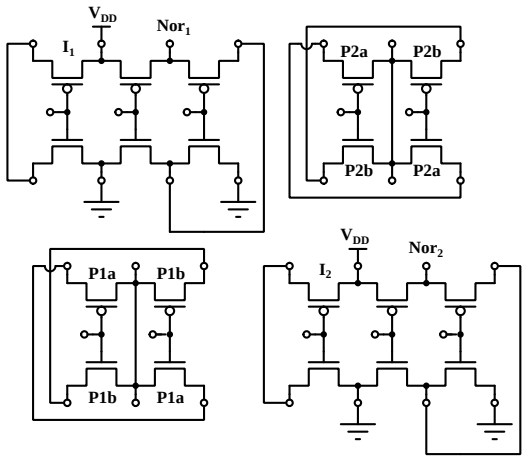
A "Sea of Gates" semi-custom logic chip contains repeated instances of 3-pairs and 2-pairs as shown on the right. Show the interconnection diagram which should be used to implement a toggle flip flop with asynchronous clear using two 3-pair and two 2-pair structures.



Soln. 2) We can show the implementation in two steps. First we implement the required gates etc. and then we interconnect them as desired.
The figure at the top on the right shows the logic diagram required for constructing the toggle flipflop. It is a master slave D flip flop whose $\overline{Q}$ output is connected to its D input. The Reset input clears the flipflop asynchronously.

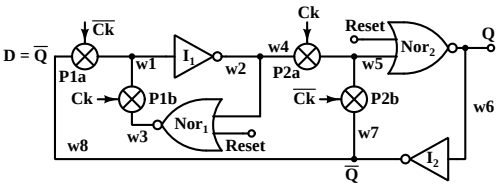


Thus, we need two Nor gates, two inverters and four pass gates for constructing the toggle flipflop. we can use two 3-pair and two 2-pair structures for this.



Each 3-pair can be converted to form a 2 input NOR + an Inverter as shown. The two 2-pairs provide the four pass gates. Diagonally opposite p and n transistors constitute a pass gate. Thus P1a, P1b, P2a and P2b can be implemented using two 2-pairs.

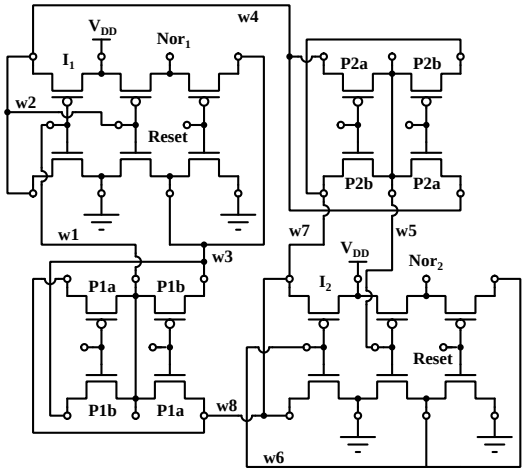
We now proceed to interconnect these gates as desired. Wire w1 takes the common point of pass gates P1a and P1b to the input of the inverter. Wire w2 connects the output of the inverter I1 to one of Nor1 inputs. (The other input is connected to Reset). The output of Nor1 is taken to the input of pass gate P1b by wire w3.



Output of inverter I1 is taken to pass gate P2a by wire w4.

Wire w5 connects the common point of pass gates P2a and P2b to one of the inputs of Nor2. The other input is connected to Reset. Wire w6 takes the output of Nor2 to inverter I2. w7 takes the output of inverter I2 to pass gate P2b.

Finally, wire w8 takes the output of inverter I2 ($\overline{Q}$) to pass gate P1a (D input).



– **[Q2: 3 marks]**

Q–3 Answer the following briefly:

a) Give a transistor level circuit for the 'tiny' XOR implementation which uses a combination of static CMOS and pass gate elements. Describe its working for different input combinations. Show how this circuit can be modified to generate the XNOR output.

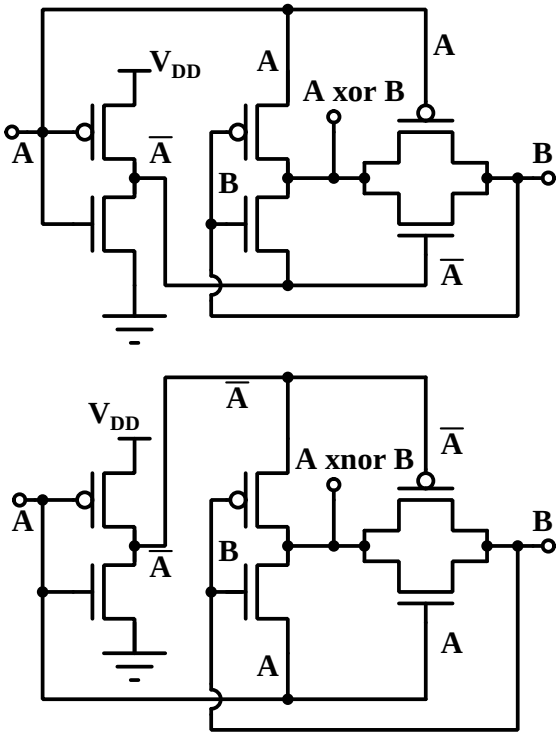
Soln. 3-a) The circuit below shows the tiny XOR and tiny XNOR.

The first stage is an inverter which produces $\overline{A}$ from A.

In the first circuit, when A is 'High', $\overline{A}$ is 'Low' and these provide the supply to the second stage which now acts as an inverter and produces $\overline{B}$ at the output. Since A is 'High' and $\overline{A}$ is 'Low', the pass gate on the right is OFF. so the output is $\overline{B}$

When A is 'Low', and $\overline{A}$ is high, transistors in the second stage act as source followers and thus produce B at the output. However, these will not produce a rail to rail swing. However the pass gate at the right is now 'ON', which couples B to the output which can now swing rail to rail. Thus the output is $\overline{B}$ for A = 0 and is B when A = 1. So the circuit implements $\overline{A} \cdot B + A \cdot \overline{B}$ which is the xor function.

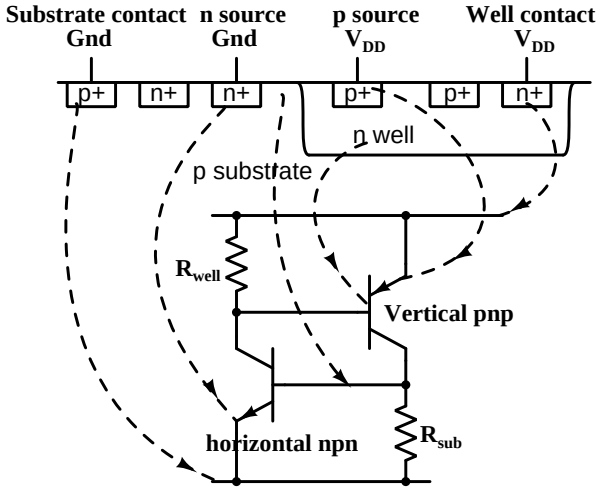
If we interchange the connections for A and $\overline{A}$ as shown in the second circuit, the second stage will invert and pass gate will be OFF when A = 0. For A = 1, the second stage will act as source follower and pass gate will be ON. Thus the output will be $\overline{A} \cdot \overline{B} + A \cdot B$ which is the xnor function.



– **[2]**

b) Draw a cross-section diagram for the latch up structure formed in CMOS process. Show the equivalent bipolar transistor circuit formed by this structure and describe how it can lead to a destructive breakdown. What steps are taken to avoid this?

Soln. 3-b) The figure below shows a cross section of a CMOS circuit and the parasitic bipolar transistors which form the latchup structure.



The vertical pnp transistor is formed by the p+ source of a pMOS transistor connected to V_{DD} (which becomes the emitter), the n well (which becomes the base) and the p substrate (which becomes the collector of this transistor). The n well is connected to V_{DD} through a resistive path, which represents the resistance of the n well to the well contact.

The horizontal npn transistor is formed by the n+ source of an nMOS transistor connected to ground (which becomes the emitter), the p substrate, (which becomes the base) and the n well, (which becomes the collector).

Since the collector of the npn and the base of the pnp are both formed by the n well, these two are connected. Similarly, the collector of the pnp and the base of the npn are formed by the p substrate, so these are connected too.

Looking at the equivalent circuit, one can see that it forms a positive feedback system. An increase in the base current of the pnp will be amplified by its β_p and a large part of it will flow through the base emitter junction of the npn transistor. This part will be amplified by the β_n of the npn and a substantial part of it will go through the base emitter junction of the pnp. If the product of the two amplification factors β_p and β_n and the current division ratios between the resistors and the base emitter junctions exceeds 1, the currents will keep increasing due to this feedback, till there is a dead short between V_{DD} and ground. This is called latch up.

To prevent latch up, we must reduce the β of the parasitic bipolar transistors and make sure that most of the collector current of either transistor is directed to the resistor and not to the base-emitter junction of the other transistor. This can be done through process steps as well as through design rules.

1. The doping gradient of the n well should be made retrograde. (Doping should increase as we go deeper). This kills the current gain β_p of the pnp transistor.
2. The n well should have a guard ring connected to V_{DD}, which will collect any current which could form the base current of the pnp.
3. In layout, substrate and well contacts should be placed frequently, to reduce the value of R_{well} and $R_{substrate}$.
4. n channel transistors should be placed far from the edge of the n well. This increase the base width of the npn transistor and kills its current gain.
5. p channel transistors should also be placed far from the well edge and the p well should be deep to kill the gain of the npn transistor.

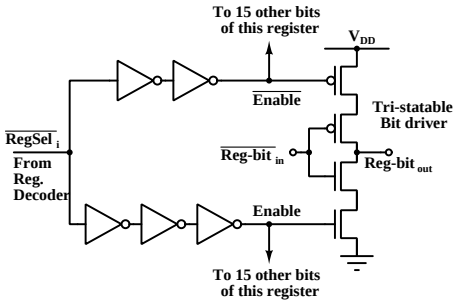
– **[2]**

– **[Q3: 2+2 = 4 marks]**

– **P.T.O**

Q–4 A system contains 8 registers, each of which is 16 bits wide. Outputs of all registers are connected to a common 16 bit bus through tri-stateable drivers. A 3 to 8 decoder provides negative true $\overline{\text{RegSel}_i}$ outputs, which are used to enable one out of the 8 registers to place its contents on the bus.

Each $\overline{\text{RegSel}_i}$ line from the decoder drives a fork whose branches contain 2 and 3 inverters to provide $\overline{\text{Enable}}$ and Enable signals, which in turn drive the p and n channel transistors of the 16 tristateable drivers for all the bits of the selected register i.
The total load presented by the two arms of the fork on the $\overline{\text{Sel}_i}$ line should be equivalent to 4 minimum inverters. Each tri-stateable driver should provide a drive strength equivalent to 3 minimum inverters.



For all parts of this question, assume that the parasitic delay of an inverter is 1.55, and p channel transistors should be twice as wide as n channel transistors to provide the same drive current ($\gamma = 2$).

a) What is the difference between stage effort values represented by $\hat{f}$ and ρ?
Find the optimum stage effort ρ for inverter parasitic delay $p_{inv} = 1.55$.
Use Newton Raphson iterations to solve the equation: $\rho(\ln \rho - 1) - p_{inv} = 0$
with a starting guess of 3.0.

Soln. 4-a) $\hat{f}$ is the optimum stage effort when the number of stages is fixed. If we can add a number of inverters in the design to possibly reduce the delay even further, a different stage effort will result since the number of stages is now higher. This optimum stage effort, with additional optimization of number of stages, is ρ.

$$\begin{aligned} \text{We define} \quad f(\rho) &\equiv \rho(\ln\rho - 1) - p_{inv} \\ \text{Then} \quad f'(\rho) &= \ln\rho - 1 + \rho(\frac{1}{\rho}) = \ln\rho \end{aligned}$$

We can now solve the non-linear equation $f(\rho) = 0$ with a starting guess of g. Then

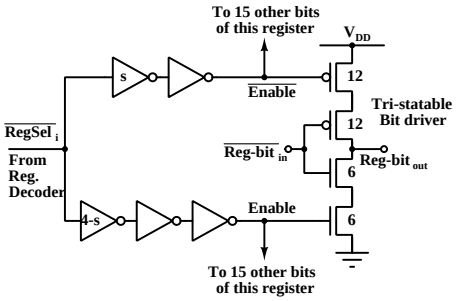
$$g_{next} = g - \frac{f(g)}{f'(g)} = g - \frac{g\ln g - g - p_{inv}}{\ln g} = \frac{g + p_{inv}}{\ln g}$$

Starting with a guess value of 3, we get successive values of g_{next} as:
4.141588, 4.005116, 4.003478, 4.003478.
So the value of $\rho = 4.003478 \approx 4$ – **[2]**

b) Find the size of the first inverters in the two branches of the fork such that the delay through the two branches is equal and the total load on the $\overline{\text{Sel}_i}$ line is equivalent to 4 minimum inverters.

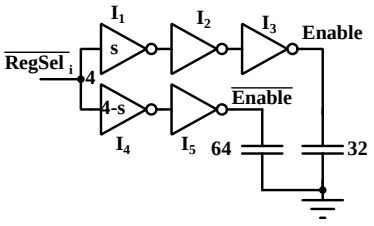
Soln. 4-b) The output tri-stateable driver should provide a driving strength of 3 minimum inverters.

Since the tri-stateable driver has two n channel transistors in series, to provide the pull down capability of 3 minimum sized n transistors, these should be 6 units wide. The p channel transistors are also in series, so for $\gamma = 2$, their widths should be 12 units. There are 16 such drivers in parallel.



Therefore the 'Enable' output of the fork is loaded with 96 min. sized n channel transistors. Since a minimum inverter presents a load of 3 width units (for $\gamma = 2$), the load on the 'Enable' line is equivalent to 32 min. inverters.
Similarly, there are 16 p channel transistors of width 12 in parallel. So these present a load of $16 \times 12/3 = 64$ min, inverters.

Thus the problem reduces to equalization of optimum delays of the two arms of the fork.
The arm with 3 inverters produces 'Enable' and is loaded with 32 inverters, while the arm with 2 inverters provides $\overline{\text{Enable}}$ which drives p channel transistors and is loaded with equivalent of 64 inverters. Assume that the first inverter in the 3 inverters branch is scaled by s. Then the first inverter of the 2 inverter branch should be scaled by (4-s), in order to present a load of 4 units to the input.



For the upper branch:

$$F = 1 \times 1 \times \frac{32}{s},\ \text{So}\ \hat{f}_1 = \left(\frac{32}{s}\right)^{1/3},\ \text{Therefore delay}\ D_1 = 3 \times \left(\frac{32}{s}\right)^{1/3} + 3 \times 1.55$$

For the lower branch:

$$F = 1{\times}1{\times}\frac{64}{4-s},\ \text{So}\ \hat{f}_2 = \sqrt{\frac{64}{4-s}} = \frac{8}{\sqrt{4-s}}\ \text{Therefore delay}\ D_2 = \frac{16}{\sqrt{4-s}} + 2{\times}1.55$$

Equating the two delays, we get

$$3 \times \left(\frac{32}{s}\right)^{1/3} + 3 \times 1.55 = \frac{16}{\sqrt{4-s}} + 2 \times 1.55$$

Which gives

$$3 \times \left(\frac{32}{s}\right)^{1/3} - \frac{16}{\sqrt{4-s}} + 1.55 = 0$$

We can solve this non-linear equation in s using Newton Raphson iterations.
We define

$$f(s) \equiv 3 \times \left(\frac{32}{s}\right)^{1/3} - \frac{16}{\sqrt{4-s}} + 1.55$$

$$\text{Then } f'(s) = 3 \times 32^{1/3}(-\frac{1}{3})s^{-4/3} - 16 \times (-\frac{1}{2})(4-s)^{-3/2}(-1)$$

$$\text{So} \quad f'(s) = -\left(\frac{32}{s^4}\right)^{1/3} - \frac{8}{(4-s)^{3/2}}$$

To begin with, we can assign equal sizes to the first inverter of the two branches. So, initially, $s = 4 - s = 2$.

guess	f(g)	f'(g)
2	-2.204182	-4.088348
1.460862	-9.701352×10^{-2}	-3.892549
1.435940	1.903964×10^{-4}	-3.908242
1.435988	8.066477×10^{-10}	-3.908209
1.435988		

At this time the value of s has stabilized and $f(s)$ has become close enough to zero. So we can accept this value of s.
Thus $s = 1.435988$ and $4 - s = 2.564012$ are the values by which the first inverter in the two branches should be scaled up. – **[5]**

c) What is the total delay through either branch of the fork?

Soln. 4-c) With the given values of scale factors, we have for the upper branch:

$$\hat{f}_1 = \left(\frac{32}{s}\right)^{1/3} = 2.814058, \quad \text{So delay } D_1 = 3\hat{f}_1 + 3p_{inv} = 13.09217$$

For the lower branch,

$$\hat{f}_2 = \sqrt{\frac{64}{4-s}} = 4.996087, \quad \text{So delay } D_2 = 2\hat{f}_2 + 2p_{inv} = 13.09217$$

Thus we see that the two delays have indeed been equalized and the load on the input is 4 inverter units. – **[1]**

d) Find the width of all transistors in the circuit shown above, in units of the width of the n channel transistor in the minimum inverter. Label all components and present your results in a tabular form.

Soln. 4-d) The size of the first inverter in the upper branch is $1.435988 \approx 1.436$. Since $\hat{f}_1 = 2.814058$, the sizes of second and third inverters are: $s \times \hat{f}_1 = 4.041$ and $s \times \hat{f}_1{}^2 = 11.371$ respectively. Just to check, $s \times \hat{f}_1{}^3 = 32$ as expected.
For the lower branch, the size of the first inverter is $4 - s = 2.564012 \approx 2.564$. Since $\hat{f}_2 = 4.996087$, the size of the second inverter is $(4-s)\hat{f}_2 = 11.281$. To check the calculation, further multiplication by $\hat{f}_2$ gives 64, as indeed it should.

In every inverter, the width of n channel transistor should be the same as its scale factor, while the p channel transistor should be twice as wide. Using this, we can summarize the geometry of all transistors as follows:

component	n size	p size
I1	1.44	2.88
I2	4.04	8.08
I1	11.37	22.74
I4	2.56	5.12
I5	11.28	22.56

Sizes of transistors in the tri-stateable driver were already calculated as 6,6,12 and 12. – [**2**]

– [**Q4: 2+5+1+2 = 10 marks**]

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

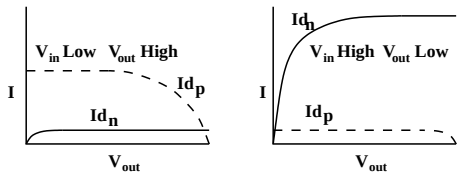
EE 671: VLSI Design

All numerical answers should be accurate to 0.1%.

Q–1 Derive an expression in terms of V_{DD}, V_{Tn}, V_{Tp}, K_n and K_p for the input voltage to a CMOS inverter such that the current from V_{DD} to ground in static conditions is maximum. Also, derive the expression for this maximum current.
(The operating regions of the two transistors at maximum inverter current can be found from the graphical solution for the output voltage of the inverter).
Calculate the value of this input voltage and maximum current for a CMOS inverter which uses both n and p channel transistors with (W/L) = 1. Values of other device and circuit parameters are given at the end of this question paper.

Soln. 1)

To find the region of operation of the two transistors when the current is maximum, we look at a graphical representation of n and p channel currents for different output voltages.



The solution lies where the currents through the two transistors are equal – i.e. at the point of intersection of the two current curves. The current at the solution point is low when the input voltage is low and the output voltage is high, because the n channel transistor limits the current to low values.
Similarly, the current at the solution point is low for high input voltage and low output voltage. This is because now the p channel transistor limits the current.
Clearly the current at the solution point will be maximum when the two transistors are saturated.
Thus the input voltage for maximum current can be found by equating the saturation current of the two transistors. Then

$$\frac{K_n}{2}(V_{in} - V_{Tn})^2 = \frac{K_p}{2}(V_{DD} - V_{in} - V_{Tp})^2$$

Defining $\beta \equiv K_n/K_p$, we get

$$\begin{aligned} \beta(V_{in} - V_{Tn})^2 &= (V_{DD} - V_{in} - V_{Tp})^2 \\ \text{So } \sqrt{\beta}(V_{in} - V_{Tn}) &= (V_{DD} - V_{in} - V_{Tp}) \end{aligned}$$

We can write this because the quantities in brackets on both sides are positive. This leads to

$$\begin{aligned} (1 + \sqrt{\beta})V_{in} &= V_{DD} + \sqrt{\beta}V_{Tn} - V_{Tp} \\ \text{So } V_{in} &= \frac{V_{DD} + \sqrt{\beta}V_{Tn} - V_{Tp}}{1 + \sqrt{\beta}} \end{aligned}$$

This is the input voltage when maximum static current flows. At this input voltage, the current is equal to either saturation current. So,

$$I_{max} = \frac{K_n}{2}(V_{in} - V_{Tn})^2$$

$$= \frac{K_n}{2}\left(\frac{V_{DD}+\sqrt{\beta}V_{Tn}-V_{Tp}}{1+\sqrt{\beta}}-V_{Tn}\right)^2$$
$$= \frac{K_n}{2}\left(\frac{V_{DD}-V_{Tn}-V_{Tp}}{1+\sqrt{\beta}}\right)^2$$

This is the expression for the maximum current drawn.
For the given parameters, $W/L = 1$ for both transistors. So,

$$\beta = \frac{\mu_n C_{ox}\cdot 1}{\mu_p C_{ox}\cdot 1} = \frac{\mu_n}{\mu_p} = 2.25;\ \text{So}\ \sqrt{\beta} = 1.5$$

$$\text{Therefore } V_{in} = \frac{1.8+1.5\times 0.35-0.4}{1+1.5} = \frac{1.925}{2.5} = 0.77\text{V}$$
$$\text{and } I_{max} = 40\times 10^{-6}\left(\frac{1.8-0.35-0.4}{2.5}\right)^2 = 7.056\mu\text{A}$$

– **[3]**

Q–2 Time to discharge the output of a CMOS inverter from V_{DD} to V_{oL} is given by

$$\frac{K_n\tau_{fall}}{C} = \frac{2(V_{DD}-V_{in}+V_{Tn})}{(V_{in}-V_{Tn})^2} + \frac{1}{(V_{in}-V_{Tn})}\ln\frac{2(V_{in}-V_{Tn})-V_{oL}}{V_{oL}}$$

Here C is the total capacitance given by $C_L + C_p$, where C_L is the load capacitance from the next stage = 20fF. C_p is the parasitic capacitance due to this stage itself, which is given by $C_p = \alpha W_n$ with $\alpha = 10\text{fF}/\mu\text{m}$. (1 fF = 10^{-15}F). The channel length is 0.2μm for both n and p channel transistors.

Using the model and device parameters at the end of this paper, find the width for the nMOS transistor such that the output will discharge from V_{DD} to V_{Tn} in 100 ps with the input voltage at 1.6V.

Soln. 2) We take W_n and L_n values in μm. $V_{in} - V_{Tn} = 1.6 - 0.35 = 1.25$V.

$$\frac{K_n\tau_{fall}}{C} = \frac{2(1.8-1.25)}{(1.25)^2} + \frac{1}{1.25}\ln\frac{2\times 1.25-0.35}{0.35}$$
$$\frac{80\times 10^{-6}(W_n/0.2)10^{-10}}{(20+10W_n)10^{-15}} = \frac{1.1}{1.25^2} + \frac{1}{1.25}\ln 6.142857$$
$$\frac{40W_n}{20+10W_n} = 0.704 + \frac{1.81529}{1.25} = 2.156232$$
$$\frac{4W_n}{2+W_n} = 2.156232$$

This leads to

$$4W_n = 4.312464 + 2.156232W_n$$
$$\text{So}\quad 1.843768W_n = 4.312464$$
$$\text{Or}\quad W_n = 2.3389 \approx 2.34\mu\text{m}$$

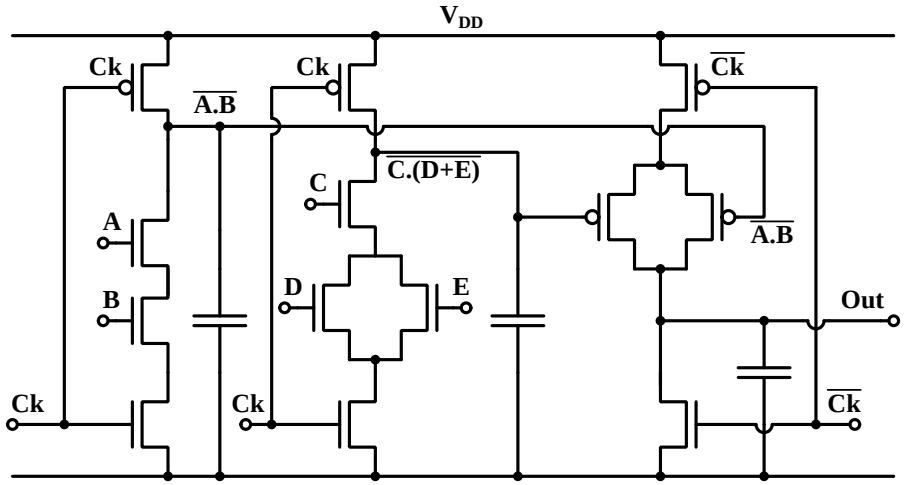
Confirmation:
With $W = 2.3389\mu$m, $W/L = 11.6947$ and $K_n = 80\times 11.6947 = 935.5763\ \mu\text{A/V}^2$.
$C = 20 + 10\times 2.3389 = 43.3894$ fF.
Therefore

$$\frac{K_n\tau_{fall}}{C} = \frac{935.5763\times 10^{-6}\times 10^{-10}}{43.3894\times 10^{-15}} = 2.156232$$

– **[4]**

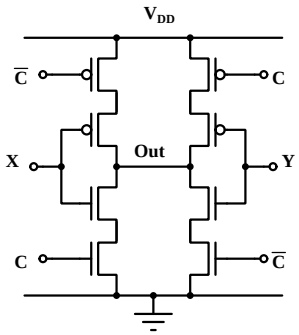
Q–3 We break up the logic expression $A.B+C.(D+E)$ as $X = \overline{A.B}$, $Y = \overline{C.(D+E)}$ and the final output as $\overline{X.Y}$. Draw a transistor level schematic for this implementation in zipper logic, labeling the clock, inputs, all the intermediate signals and the output clearly.

Soln. 3)



– [**2**]

Q–4

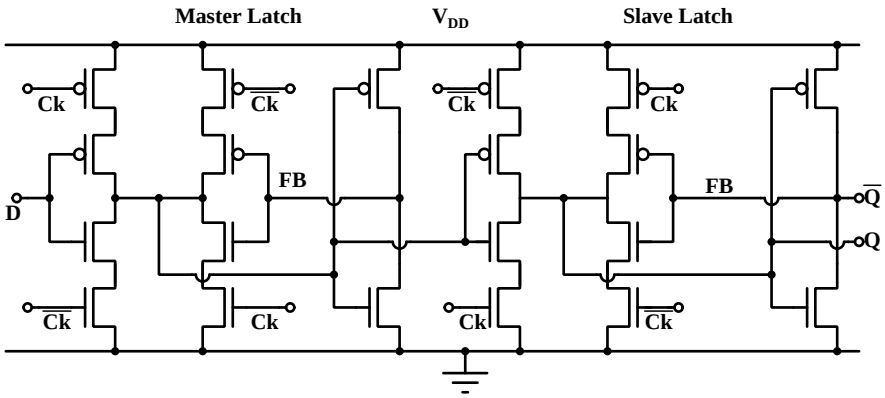


The circuit on the left provides either $\overline{X}$ or $\overline{Y}$ depending on the value of C.

Draw the transistor level schematic of a D type flip flop which uses this inverting mux element instead of pass transistors.

Soln. 4)

•



– [**1**]

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY
ELECTRICAL ENGINEERING DEPARTMENT

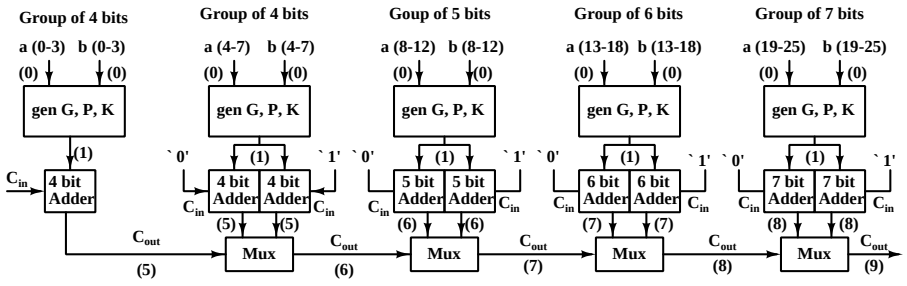
EE 671: VLSI Design

Sunday	Class Test 2	Time: 1600-1700
20-10-19	Autumn Semester 2019	Marks: 10

All numerical answers should be accurate to 0.1%.

Q–1 Show how a 26 bit carry select adder can be constructed using square root tiling, using a 4 bit adder for the first stage. Find the time when the final result and all the intermediate carries will be ready in this case, assuming that each of the following operations takes one unit of time:
–generation of G and P signals,
–carry output from G, P and carry input,
–multiplexing the carry,
–generation of sum from P and carry input.

Soln. 1) The first stage adds 4 bits. Since the input carry is already known for this stage, there is no need for redundant adders and a mux at this stage. Therefore, G, P symbols will be ready at 1, the output carry will be ready at 1+4 = 5 units.



The next stage carry should be ready by the time this carry reaches its mux. Therefore the next stage should also add 4 bits. Each subsequent stage can afford to add an additional bit, since the carry controlling the mux will arrive one unit later. Hence the stages should add 4, 4, 5, 6 and 7 bits.

The time when various signals become valid is shown in parentheses in the figure above. The last of the bit wise carry input signals in the group of 7 bits will arrive at 7 units of time. The sum and carry from this will be available at 8 units of time. One more unit of time will be required to choose the sum and carry outputs from the correct sub-adder. Thus, the final sum and output carry will be ready at 9 units of time.

– **[Q1: 2 marks]**

Q–2 a) Describe the modified Booth algorithm which is used to reduce the number of partial products generated in a multiplier. Tabulate the actions to be taken for different bit combinations of multiplier.

Soln. 2-a) The algorithm reduces the number of partial products by multiplying by two bits of the multiplier at a time. Multiplication by each group of 2 bits produces partial products whose place value is 4 times the place value of the previous group. Therefore each partial product is shifted two places to the left before being added to produce the result.

Let the multiplicand be A. Multiplication by two bits of the multiplier involves multiplying A by 0, 1, 2 or 3. The result of multiplying A by 0, 1 or 2 can be generated trivially. In Booth algorithm, instead of generating a partial product which is 3A, we subtract A and ask the next group of two bits (with place value 4 times that of this group) to add 1 to the multiplier, thus effectively adding (-1 + 4 = 3)A.

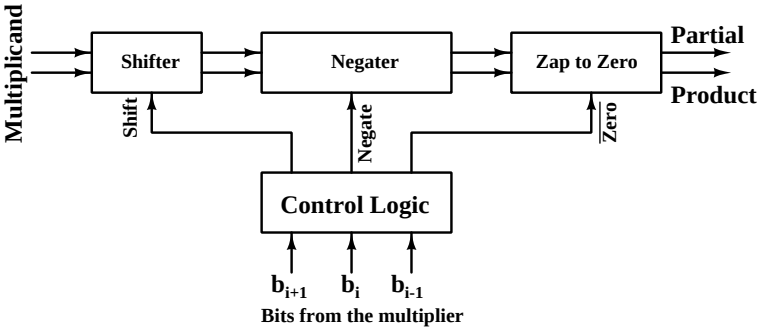
To make it easier for the next group of 2 bits to detect if they are required to add 1 to the multiplier or not, we use this trick also if the current multiplier bits are '10'. Instead of adding 2A, we subtract 2A, and ask the next group of 2 bits to increment by 1. Since incrementing is now required for both '10' and '11' cases, the next group of two bits just needs to look at the more significant bit of the previous group and if it is a '1', then the partial product to be generated corresponds to the incremented value of the current two bits. For the least significant two bits, there is an implied '0' to the right of the least significant bit. Similarly to finish any pending request from the last two bits, the group '00' may need to be added at the most significant bit end. This is the modified booth algorithm. Thus the partial product generation logic is:

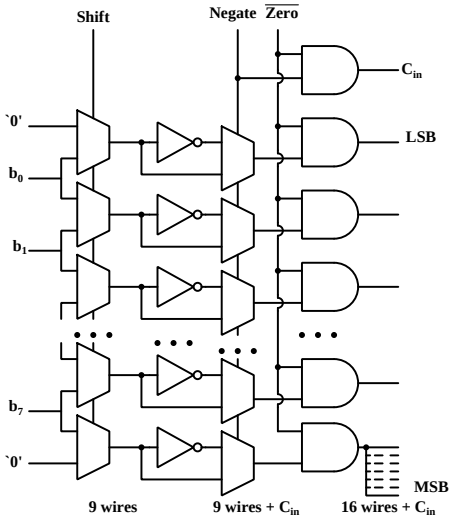
Current 2 bits	Previous MSB	Partial Product	Remark
00	0	0	
01	0	A	
10	0	-2A	Sign extension reqd
11	0	-A	Sign extension reqd
00	1	A	
01	1	2A	Left Shift A
10	1	-A	Sign extension reqd
11	1	0	

– **[2]**

b) We want to produce the partial products in parallel, using three successive stages (for each multiplicand bit group). These stages are for i) shifting the multiplicand, ii) negating the multiplicand and iii) zeroing the whole partial product.
Using 8 bit operands (with a 16 bit product) as an example, give a logic diagram for generating the above actions, as well as for producing the control signals required for carrying out these actions.

Soln. 2-b) The following diagram shows how the Booth algorithm may be implemented in hardware.





The shifter produces 9 output wires, which can be "$b_7b_6 \ldots b_0$0" or"0$b_7b_6 \ldots b_0$". Since the shifter effectively multiplies the input by 2 (if Shft = '1'), the most significant bit of the output has one position higher weight than the input.
The negater chooses between a bit or its complement. (This could also be done with an xor gate). The control bit goes through as 'carry in' of the adder.
The and array forces the output of all bits to zero if required (when the control signal $\overline{\text{Zero}}$ is '0'. The final output from this array goes as a 16 bit partial product, with the most significant bit replicated to make up 16 bits. This ensures sign extension.

To generate the required control signals, we use the fact that 'Shift' and 'Negate' can be don't care if $\overline{\text{Zero}}$ is '0'.
Based on the table for Booth algorithm, we can set up the following Karnaugh maps for the control inputs to the Shifter, Negater and Zero Array:

- **Shift**

$b_{i+1}b_i$ → / b_{i-1} ↓	00	01	11	10
0	0/1	0	0	1
1	0	1	0/1	0

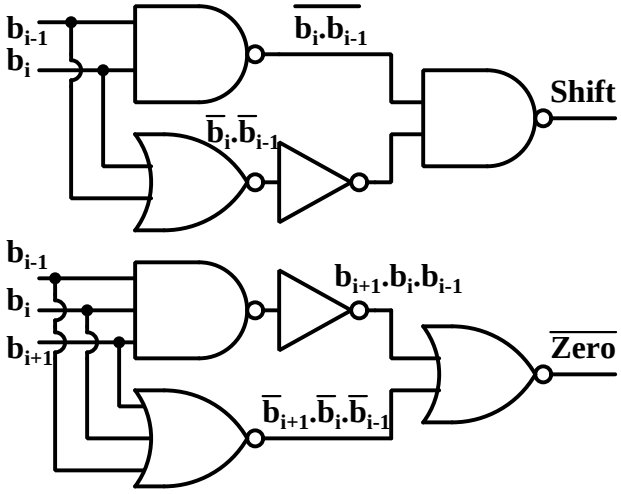
Shift = $\overline{b_i}\,\overline{b_{i-1}} + b_i\,b_{i-1}$

Negate

$b_{i+1}b_i$ → / b_{i-1} ↓	00	01	11	10
0	0/1	0	1	1
1	0	0	0/1	1

Negate = b_{i+1}

$\overline{\text{Zero}} = \overline{b_{i+1}\,b_i\,b_{i-1} + \overline{b}_{i+1}\,\overline{b}_i\,\overline{b}_{i-1}}$



The Karnaugh maps use the fact that when all three bits are 0 or 1, $\overline{\text{Zero}}$ will be '0'and therefore the Shift and Negate signals are 'don't care' i.e. can be either 0 or 1. Using this, the expressions for the control inputs are:

$$\text{Shift} = \overline{b_i} \cdot \overline{b_{i-1}} + b_i \cdot b_{i-1}, \qquad \text{Negate} = b_{i+1}$$

and $$\overline{\text{Zero}} = \overline{b_{i+1} \cdot b_i \cdot b_{i-1} + \overline{b_{i+1}} \cdot \overline{b_i} \cdot \overline{b_{i-1}}}$$

– **[2]**

c) Assume A and B are two unsigned 8 bit operands to be multiplied. If the multiplicand A is 01110111 and the multiplier B is 10100111, what will be the partial products generated using modified Booth algorithm? Show by comparison with the product of decimal equivalents of the two numbers that the addition of the binary partial products does produce the correct result.

Soln. 2-c) The decimal equivalents of A and B are: 119 and 167 respectively. The expected product is $119 \times 167 = 19873$. This is 4DA1 in Hex or 0100 1101 1010 0001 in binary

The possible partial products are 0, A, 2A, -A and -2A. It is convenient to precompute these. Sign extension may be required up to 16 bits.

A	0000000001110111
2A	0000000011101110
-A	1111111110001001
-2A	1111111100010010

1. Scanning B with a zero to the right gives the first group of 3 bits as '110', with the partial product as -A.
2. The next group of 3 bits (with 1 bit overlap on the right) is '011'. This gives the partial product as 2A. (This will be shifted two places to the left, so effectively, this is 8A).
3. The next group of 3 bits is '100', so the partial product is -2A. (Since this will be shifted left by 4 places, it is effectively -32A).
4. The next group of 3 bits is '101', so the partial product is -A. (This will be shifted left by 6 places – so this is effectively -64A).
5. Finally, since the last partial product was negative, we add two zeros to the left to finish off the pending work of the last group. Thus, the 3 bits are '001', giving the partial product as A. (This will be shifted 8 place to the left, so effectively, this is 256A).

When we add up the partial products, we shall get
$-A + 8A - 32A - 64A + 256A = 264A - 97A = 167A$, which is what we want.

In binary format:

3 bit gp.	PP	Binary PP with place value															
110	-A	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
011	2A	0	0	0	0	0	0	1	1	1	0	1	1	1	0	X	X
100	-2A	1	1	1	1	0	0	0	1	0	0	1	0	X	X	X	X
101	-A	1	1	1	0	0	0	1	0	0	1	X	X	X	X	X	X
001	A	0	1	1	1	0	1	1	1	X	X	X	X	X	X	X	X

We can perform a tree addition for these numbers. We add the top two rows, then the next two rows, add these together and finally add the fifth row to that sum. Carry outs to beyond the 16 bit answer can be ignored.

Partial Sums	Binary value															
Rows 1 + 2	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
Rows 3 + 4	1	1	0	1	0	0	1	1	0	1	1	0	X	X	X	X
Rows 1+2+3+4	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	1
Row 5	0	1	1	1	0	1	1	1	X	X	X	X	X	X	X	X
Final Sum	0	1	0	0	1	1	0	1	1	0	1	0	0	0	0	1

So the final product is 4DA1 in Hex – which is $77 \times 256 + 161 = 19873$. This agrees with the expected product: $119 \times 167 = 19873$. – **[2]**

– **[Q2: 2 + 2 +2 = 6 marks]**

Q–3 Show a wire reduction scheme using a Wallace tree for a multiply and accumulate circuit in which two 8 bit operands are to be multiplied and added to a 16 bit accumulator. Show a dot diagram for the operation. Use the Wallace tree scheme which does not produce a redundant bit.

Soln. 3) The multiplier partial products will give 1,2,3,4,5,6,7,8,7,6,5,4,3,2,1 wires from b_{14} to b_0. The accumulator will provide 1 wire each from b_15 to b_0. The table below shows the wire count at each reduction stage, following the Wallace technique.

Stage 1: capacity of next stage = 6

Bit pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Wires	1	2	3	4	5	6	7	8	9	8	7	6	5	4	3	2
Full Adders	0	0	1	1	1	2	2	2	3	2	2	2	1	1	1	0
Half Adders	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Pass Through	1	2	0	1	2	0	1	0	0	2	1	0	2	1	0	0
Output Wires	1	3	2	3	5	4	6	6	5	6	5	3	4	3	2	1

Stage 2: capacity of next stage = 4

Bit pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Wires	1	3	2	3	5	4	6	6	5	6	5	3	4	3	2	1
Full Adders	0	1	0	1	1	1	2	2	1	2	1	1	1	1	0	0
Half Adders	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
Pass Through	1	0	2	0	2	1	0	0	0	0	2	0	1	0	0	1
Output Wires	2	1	3	2	4	4	4	4	4	3	4	2	3	2	1	1

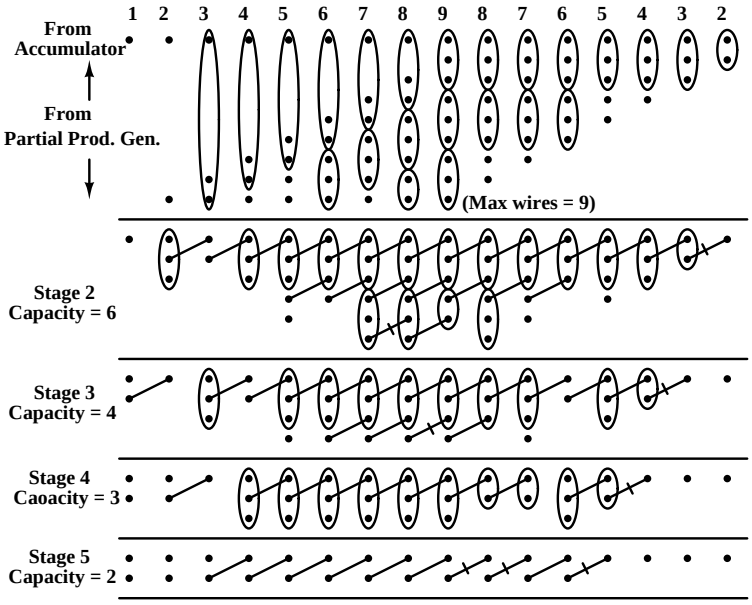
Stage 3: capacity of next stage = 3

Bit pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Wires	2	1	3	2	4	4	4	4	4	3	4	2	3	2	1	1
Full Adders	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0
Half Adders	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Pass Through	2	1	0	2	1	1	1	1	1	0	1	2	0	0	1	1
Output Wires	2	2	1	3	3	3	3	3	3	2	2	3	2	1	1	1

Stage 4: capacity of next stage = 2

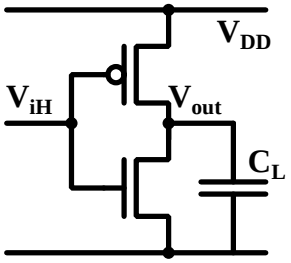
Bit pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input Wires	2	2	1	3	3	3	3	3	3	2	2	3	2	1	1	1
Full Adders	0	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0
Half Adders	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
Pass Through	2	2	1	0	0	0	0	0	0	0	0	0	0	1	1	1
Output Wires	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1

The dot diagram for this scheme is shown below:


1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2
From Accumulator
From Partial Prod. Gen.
(Max wires = 9)
Stage 2 Capacity = 6
Stage 3 Capacity = 4
Stage 4 Caoacity = 3
Stage 5 Capacity = 2

– [**Q3: 2 marks**]

Q–1 (Class Test-1: 2015)
We want to calculate the discharge time of a CMOS inverter using a transistor model which does *not* assume perfect saturation.



For computation of the discharge time, we take the input voltage V_{iH} to be $V_{DD} - V_{Tp}$, so that the P channel transistor is OFF.

The transistor model is given by:

$$V_g^{'} \equiv V_{gs} - V_{Tn}, \qquad V_{dss} = V_g^{'}\left(1 - \frac{V_g^{'}}{2V_E}\right) \qquad \text{where } V_E \text{ is the Early Voltage.}$$

$$K_n \equiv \mu_n C_{ox}\frac{W}{L} \quad \text{where } \mu_n = \text{electron mobility, } C_{ox} = \text{gate capacitance per unit area}$$

$$\begin{aligned}
I_{ds} &= 0 && \text{when } V_g^{'} \le 0 \\
I_{ds} &= K_n(V_g^{'}V_{ds} - \tfrac{1}{2}V_{ds}^2) && \text{when } V_g^{'} > 0, \quad V_{ds} \le V_{dss} \\
I_{dss} &\equiv K_n(V_g^{'}V_{dss} - \tfrac{1}{2}V_{dss}^2) && \text{(Drain current at onset of saturation)} \\
I_{ds} &= I_{dss}\tfrac{V_{ds}+V_E}{V_{dss}+V_E} && \text{when } V_g^{'} > 0, \quad V_{ds} \ge V_{dss}
\end{aligned}$$

a) Derive the expression for time taken to discharge the load capacitor from V_{DD} to V_{Tn} in terms of the load capacitor value C_L, V_{DD} and the transistor parameters defined above.

b) Given $C_L = 1$pF, $V_{DD} = 3.0$V, $\mu_n C_{ox} = 600\mu$A/V^2, $V_{Tn} = 350$mV, $V_{Tp} = 400$mV and Early voltage $V_E = 10$V, find the (W/L) value for the n channel transistor such that we can discharge a load capacitor of 1 pF from V_{DD} to V_{Tn} in 1 ns when the input voltage is $V_{DD} - V_{Tp}$.
(You must use the results derived in part (a) above.)

Q–2 (Test-1:2016)
What should be the ratio of widths of n and p channel transistors of a CMOS inverter, such that the time taken to charge the output from 0V to $V_{DD} - V_{Tp}$ with the input voltage $= V_{Tn}$ is the same as the time taken to discharge the output from V_{DD} to V_{Tn} with the input voltage $= V_{DD} - V_{Tp}$.

You are given that $V_{DD} = 3.0$V, $V_{Tn} = 0.5$V, $V_{Tp} = 0.7$V, $\mu_n = 450$cm^2/Vs, and $\mu_p = 250$cm^2/Vs. The n and p channel transistors have the same channel length and gate oxide capacitance per unit area.

(Expressions for charge and discharge times should be derived and *not* quoted from memory).

Q–3 (Mid-sem:2017)
Consider a CMOS inverter in which n and p channel transistors have been sized to give equal rise and fall times. Derive an expression in terms of p channel transistor parameters for charging the output capacitance C_L from 0V to 3V with $V_{DD} = 3.3$V. Assume the input voltage to be 0.5V, which is below V_{Tn}. You can assume perfect saturation.

If the value of $K_p \equiv \mu C_{ox} W/L$ is 100μA/V^2, $V_{DD} = 3.3$V, $V_{Tp} = 0.7$V and the load capacitance is 0.1 pF, find the charge time from 0 to 3V. Find the value of the equivalent resistor which will charge the load capacitance in the same amount of time from 0 to 3V.

Q–4 (End-sem: 2017)

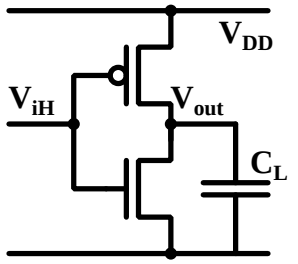
a) Consider a CMOS inverter. Assume that the geometries of the p and n channel transistors are so chosen that their conductance factors K_p and K_n are equal. Derive an expression for the input 'Low' and output 'High' values in terms of the supply voltage and absolute values of the turn on voltages.

b) For a CMOS inverter, $K_p = K_n = 100\mu$A/V^2. For what input voltage is the static current drawn by the inverter maximum? What is the value of this maximum static current?

c) Assume that the widths of the p channel transistors need to be twice the width of n channel transistors for matching their conductance factors K_p and K_n. The minimum width of an n channel transistor is 250 nm. Using thumb rules for scaling geometries, find the widths of the four transistors in a tri-stateable inverter so that it meets the same dynamic specifications as the minimum inverter with equal K_p and K_n values.

Q–5 (Test-1:2018)

Consider a CMOS inverter as shown on the right. Assume that the supply voltage V_{DD} is 3.3V and the external load capacitance C_L is 0.1 pF. Parameters for the n and p channel transistors are :



Parameter	N Channel	P Channel
μC_{ox}	45μA/V^2	22μA/V^2
V_T	0.6 V	-0.6 V

The rise time for a CMOS inverter is given by

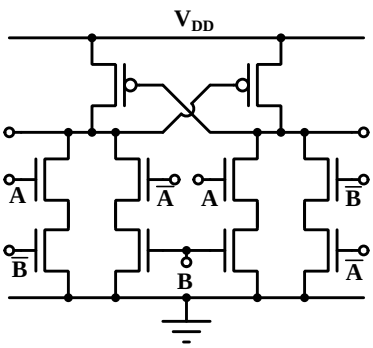
$$\frac{K_p \tau_{rise}}{C} = \frac{2(V_{iL} + V_{Tp})}{(V_{DD} - V_{iL} - V_{Tp})^2} + \frac{1}{(V_{DD} - V_{iL} - V_{Tp})} \ln \frac{V_{DD} + V_{oH} - 2V_{iL} - 2V_{Tp}}{V_{DD} - V_{oH}}$$

Here C is the total capacitance given by $C_L + C_p$, where C_p is the parasitic capacitance. K_p is the conductance factor for the P channel transistor, with $K_p = \mu_p C_{ox} W/L$. V_{Tp} represents the absolute value of the p channel threshold voltage. Channel length L for all transistors is 0.35μm.
The parasitic capacitance is given by $C_p = \alpha W$, with $\alpha = 10^{-14} F/\mu m$.
Other symbols have their usual meanings.

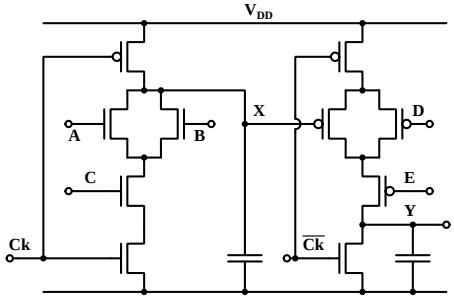
Find the width for the pMOS transistor such that the output will charge from 0V to 3.0V in 5 ns when the input voltage is 0.3V.

Q–6

Where does the circuit given on the right for a CVSL gate deviate from the usual series-parallel rule?
Does it still provide a valid implementation for some logic function? If so, identify the function and describe how it works.
Otherwise suggest the necessary changes to the circuit to make it operate properly.



Q–7 Consider the two stage np zipper circuit given below. Identify the logic functions appearing at nodes x and y. Draw a timing diagram showing $Ck, \overline{Ck}, X$ and Y on the same time scale, when A = C = D = E = '1' and B = '0'.



– **[2]**

Q–8 Show a logic diagram for the function $A \cdot B \cdot (\overline{C+D}) + \overline{E}$ implemented in 4 phase dynamic logic using only inverters, two input NANDs and two input NORs.

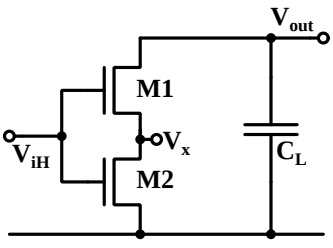
A,B,C,D and E are available in phase 1 in uncomplemented form only. Transistor level circuits are not required. Just draw a logic diagram with the type of each gate inscribed in its gate symbol. The circuit should produce the output as quickly as possible.

Q–9 (Mid-sem: 2015)
Consider a pseudo NMOS inverter where the ratio of K_n and K_p is β. ($K \equiv \mu C_{ox} W/L$).

a) Assuming perfect saturation for drain currents of transistors, find the input 'High' and output 'Low' logic levels in terms of the supply voltage V_{DD}, turn on voltages V_{Tn} and V_{Tp} and β.
(V_{iH} and V_{oL} are defined by the point on the transfer curve where the gain is -1, with PMOS in saturation and NMOS in linear regime.)

Q–10 (Mid-sem: 2018)
Consider a 2 input CMOS NAND gate acting as an inverter with both its inputs tied to a logic 'High' value. Since the inputs are 'High', the pMOS transistors are OFF and may be ignored for this problem. The two n channel transistors M1 and M2 have identical geometries and electrical parameters. We wish to analyse this circuit without approximating the behaviour of the transistors as equivalent resistors.

Assume that the load capacitor C_L is initially charged to $V_{DD} =$ 3.3V. Both inputs are tied to $V_{iH} = 3.0$V. Assume $V_{Tn} = 0.6$V. Dependence of V_{Tn} on the source voltage (bulk effect) is to be ignored. As the output discharges, V_{out} goes from V_{DD} towards 0 V. Use the simple MOS model with perfect saturation for MOS transistor currents.



a) In what modes (saturated or linear) are the two transistors for different values of V_{out} as it drops from V_{DD} to 0?

b) Derive expressions for the voltage at the source of M1 (V_x) in terms of V_{iH}, V_{Tn} and V_{out} for different combinations of modes of M1 and M2 which will occur during the discharge. Taking $V_{iH} = 3.0$V and $V_{Tn} = 0.6$V, tabulate values of V_x for:

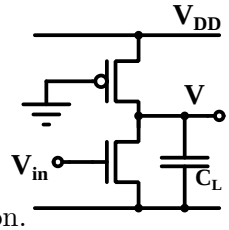
V_{out} = i) 3.3 V, ii) 3.0 V, iii) 2.4 V, iv) 1.8 V and v) 1.2 V.

c) How much is the discharge current through the series connected transistors M1 and M2, when expressed as a fraction of the discharge current through a single nMOS transistor with identical dimensions replacing the series connected transistors M1 and M2, with the same input voltage V_{iH} applied to its gate? Evaluate this ratio for all combinations of operating modes of M1 and M2 which occur during discharge.

Q–11 (End-sem:2016)

a)

Consider the pseudo-nMOS inverter shown on the right. Derive an expression for the time to charge a load capacitor C_L from 0V to an output voltage V_{oH} when the input is LOW and the nMOS driver transistor is OFF. The expression should be in terms of V_{DD}, C_L and the pMOS transistor parameters.
Use the simple MOS model which ignores channel length modulation.



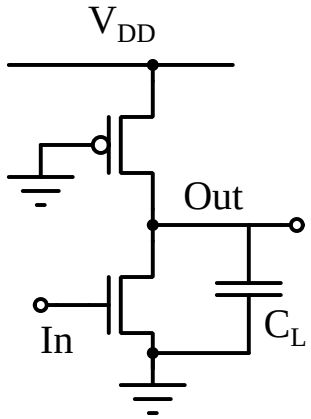
b) In a CMOS process, $C_{ox} = 4.4\times10^{-7}$F/cm^2, $\mu_n = 400$cm^2/Vs, $\mu_p = 190$cm^2/Vs, $V_{Tp} = -0.65$V, $V_{Tn} = 0.55$V. We want to design a pseudo-nMOS inverter using this process, with $V_{DD} = 3.3$V. What should be the aspect ratio of pMOS transistor (W_p/L_p) such that the inverter charges a load capacitor of 10fF from 0 to 3V in 50ps when the input is LOW (and so the nMOS is OFF).

c) What should be the minimum aspect ratio of the n MOS transistor (W_n/L_n) in the above inverter, such that the output voltage is $\leq 0.4V$ when the input voltage is $V_{iH} = 3.0$V.

Q–12 (Test-1: 2017)

Consider a pseudo-nMOS inverter as shown on the right. Assume that the supply voltage V_{DD} is 3.3V and the load capacitance C_L is 0.1 pF. Parameters for the n and p channel transistors are :

Parameter	N Channel	P Channel
μC_{ox}	45μA/V^2	22μA/V^2
V_T	0.6 V	-0.6 V



a) Find the W/L value for the pMOS transistor which will charge the load capacitor from 0V to 3.0V in 5 ns when the nMOS transistor is OFF. The rise time for a pseudo-nMOS inverter is given by

$$\tau_{rise} = \frac{C_L}{\mu_p C_{ox}(W_p/L_p)(V_{DD} - V_{Tp})}\left[\frac{2V_{Tp}}{V_{DD} - V_{Tp}} + \ln\frac{V_{DD} + V_{oH} - 2V_{Tp}}{V_{DD} - V_{oH}}\right]$$

V_{Tp} in the above expression represents the absolute value of the p channel threshold voltage.

b) Find the value of the equivalent resistor which will charge the load capacitor from 0V to 3.0V in the same amount of time (5 ns).

c) Find the ratio of (W/L) values for the n channel and p channel transistors such that the static output voltage is = 0.3V when the input voltage is 3.0V. (No memorized expressions should be used. Find the output voltage by equating currents through the two transistors.)

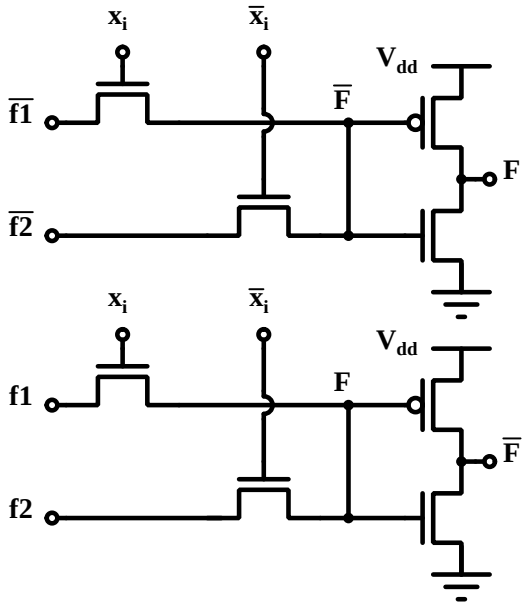
d) We represent the inverter as a voltage divider, with the p channel transistor replaced by its equivalent resistor computed in part b) above and the n channel transistor by another resistor, such that the static output is = 0.3V. What is the ratio of the equivalent resistors for n channel and p channel transistors?

Q–13 (Midsem: 2015)

a) How is the pseudo NMOS configuration modified to form the dual rail Cascade Voltage Switch Logic, such that static power dissipation is avoided?

b) Draw the transistor level schematic of an XOR/XNOR gate using Cascade Voltage Switch Logic (CVSL).

Q–14 (Test-1:2016)
Consider the basic CPL gate shown below.



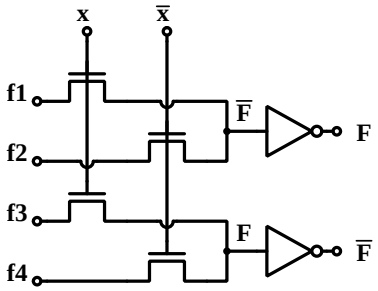
a) Describe the operation of this gate. Why are the inverters required?

b) Why does this configuration lead to leakage current in the inverters?

c) Show how we can add a pull-up p channel transistor to remove the leakage problem. Explain how this addition makes the logic configuration ratioed, rather than ratio-less.

Q–15 End-Sem: 2015

The circuit on the right shows the basic structure used by CPL gates. This needs inputs in true and complement forms and provides output in true as well as complement form. Appropriate choices have to be made for x, f1, f2, f3 and f4 for producing the desired logic functions F and $\overline{F}$.



a) Given inputs A and B in true as well as complement form, show how you will connect these to the CPL structure shown above to generate the XOR and XNOR of A and B.

b) Explain why the basic structure shown in the figure may lead to leakage in the inverters used at the output. Show how it can be prevented by using a PMOS pull-up transistor. (Give a transistor level circuit for this). Explain the pull-up action of the transistor for different output values.

c) The addition of a pull up transistor to reduce inverter leakage requires the transistor widths to be ratioed, otherwise the circuit may not work. Explain why this happens.

Q–16 (Midsem: 2015)

a) Show the timing diagram for clocks, internal nodes and output of a 4 phase CMOS dynamic logic gate of type 1. The logic function performed by series/parallel connection of NMOS transistors need not be shown and can be represented by a black box. You should clearly mark the clock phases during which the output is valid.

b) A circuit module receives external signals ATN (attention) and four address lines A3-A0. It is supposed to respond to incoming data on a data bus D7-D0 if ATN is '1', irrespective of address line values. If ATN is '0', it should respond to the data bits only if the bit pattern on A3-A0 is 0110 (which is its address).

We want to generate an 'Enable' signal which will be 1 only when the circuit module needs to respond to incoming data, using 4 phase CMOS Dynamic logic. Signals ATN and A3-A0 are valid only in phase 1 of the clock. Due to a restriction on series connected transistors, only NAND and NOR gates with a maximum of 3 inputs and inverters can be used.

Show a gate level implementation, clearly marking the type of all gates. Specify the clock phases during which the 'Enable' signal is valid. The design should minimize complexity and delay.

– **[Q3: 2 + 3 = 5 marks]**

Q–17 (Mid-sem: 2016)

Why does a CMOS dynamic logic gate malfunction if we do not use multiple clocks? How is this problem solved using 4 phase dynamic logic? What is the restriction for driving different types of gates in 4 phase logic?

How does Zipper logic manage to solve this problem without needing multiples phases of the clock?

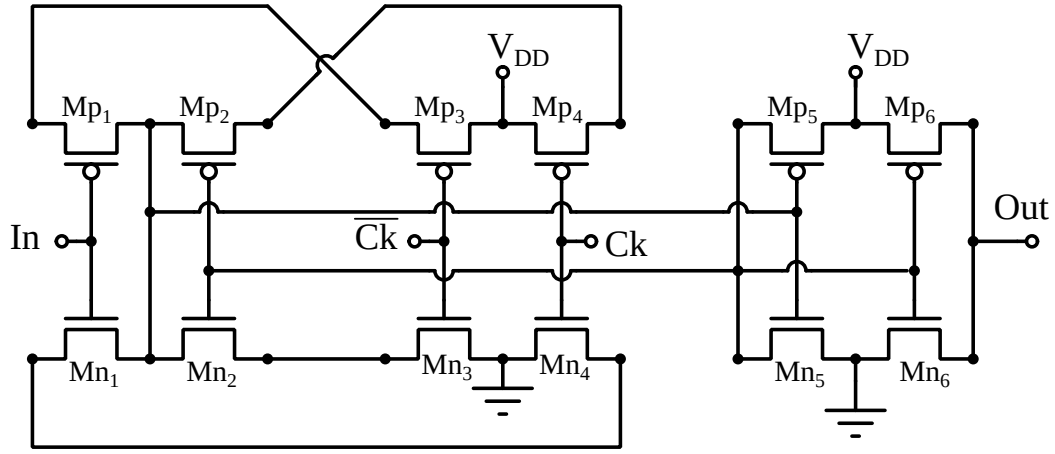
Q–18 (Mid-sem:2017)

A 4 bit decoder needs to be designed using 4 phase dynamic logic. Inputs to the decoder are an Enable signal En and a 4 bit address $A_3A_2A_1A_0$. These input signals are valid only in phase $\phi 1$. (Complemented address bits are not available and should be generated when

required). Design a decoding circuit which will produce an output signal $\overline{\text{Sel}}$ which becomes '0' only when En is '1' and the address bits have the value '1001'. The output should be available in as early a phase as possible. You are allowed to use only NAND, NOR and Inverter circuits with a maximum of 3 logic inputs (not counting clock). Give the logic diagram and the 'type' of each gate which specifies in which phase it evaluates.
(Slow and unnecessarily complex circuits will get no credit).

Q–19 Give the transistor level circuit diagram of a static Cascade Voltage Switch logic (CVSL) gate which implements the logic function $A.(B+C)$ and its complement, given A, B, C and their complements as inputs. How does this logic style avoid static current when the output is '0' while still needing to drive mostly n type transistors?

Q–20 (Test-1:2017)
A circuit has been designed in "sea of gates" style, using interconnects as shown below:



a) Re-draw the schematic in conventional style, separating all gates and with V_{DD} on top and ground at the bottom. (Your schematic should clearly identify the labels for all transistors and signals corresponding to the labeling above).

b) What function does this circuit perform? Describe how it works.

c) For the input and clock waveforms given below, sketch the expected output showing the timing relationship with respect to the clock and the input (In).

Q–21 (Mid-sem:2018)
In a programmable logic array using pseudo NMOS style logic, we generate programmable products in one array and then add programmably selected products in the other.

a) NAND functions need to size the nMOS transistors depending on the number of inputs. This presents problems with programmability of the product array. How is this problem solved in PLAs?

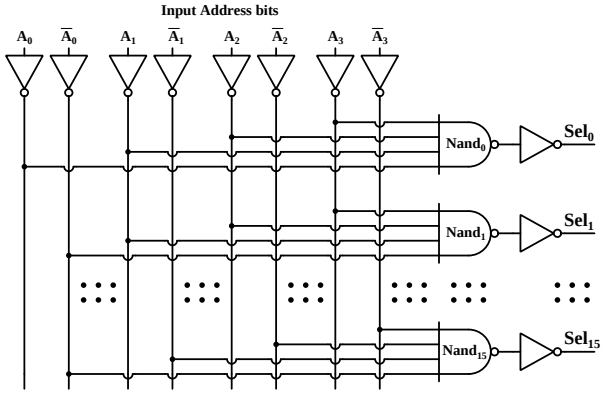
b) Show a transistor level circuit for a PLA implemented with pseudo NMOS logic with four primary inputs, six possible products and six outputs.

c) Show how this configuration can be enhanced with latches to implement generic finite state machines.

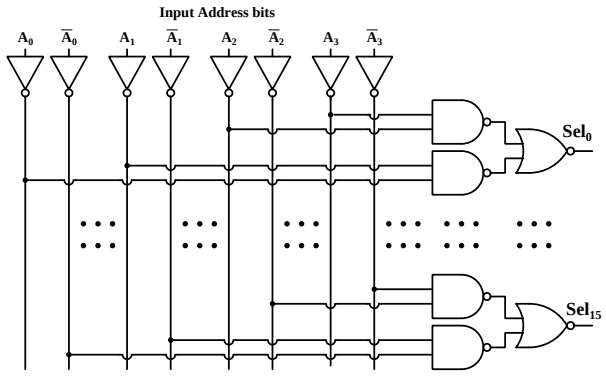
Q–22 (Midsem: 2017)

For a given CMOS process, the mobility correction factor γ for PMOS transistor widths is 2.5. The parasitic delay of gates may be taken to be proportional to the sum of the widths of transistors directly connected to the output terminal in a minimum sized gate. The parasitic delay of an inverter (p_{inv}) is 2 in units of τ, the propagation delay of a minimum sized inverter driving another minimum sized inverter without including the parasitic delay.

We want to compare two circuits to implement a 4 to 16 decoder. In circuit (i), appropriate combinations of address bits and their complements are given to 4-input-NAND gates, and their outputs are connected to inverters. In circuit (ii), combinations of address bits and their complements go to 2-input-NAND gates and their outputs are combined pair wise by 2-input-NOR gates to generate the select outputs as shown.



Decoder (i) with 4 input NANDs and inverters.



Decoder (ii) with 2 input NANDs and 2 input NORs

In both circuits, the inverters at the input are minimum sized and each select output is loaded with capacitance equivalent to 128 minimum sized inverters. All transistors use minimum channel length.

a) Compute the logical effort and parasitic delay for all the types of gates involved in the above circuits.

b) Find the widths for n and p channel transistors in all the gates of both circuits to minimize the total delay. (Specify the widths in units of the width of the n channel transistor in a minimum inverter).

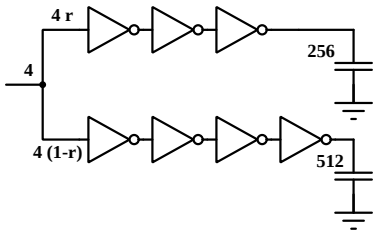
c) Compute the total delay in units of τ for both circuits.

d) The optimum stage ratio ρ is a solution to the equation $\rho(1 - \ln \rho) + p_{inv} = 0$.
Find the value of ρ and the optimum logic depth for the two decoders for the specified loading. What is the total delay for the two circuits if the logic depth is made optimum by adding inverters?

Q–23 (Mid-sem:2018)

We want to design a 3-4 fork with the total input capacitance (to be driven by the up-stream driver) equal to 4 times the minimum inverter input capacitance.
Assume $\mathbf{p_{inv}} = \mathbf{2.0}, \gamma = \mathbf{2.2}$.

The input capacitance is divided in the ratio r:(1-r) for the 3 and 4 inverter branches of the fork respectively. The final load on the branch with 3 inverters is equivalent to 256 minimum inverters, while that on the 4 inverter branch is equivalent to 512 minimum inverters.



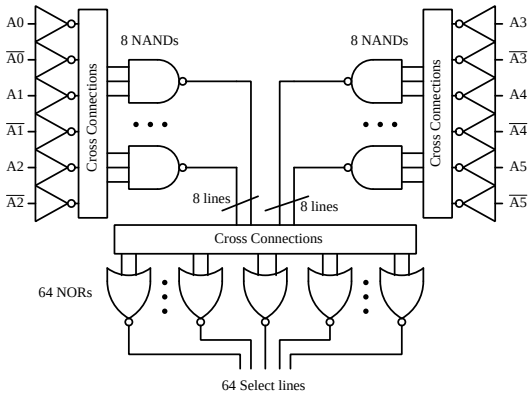
a) Evaluate the value of r such that the optimum delay in the two branches is equal, using Newton Raphson technique (starting with a guess value of r=0.5).

b) Calculate the sizes of all transistors in the fork. Transistor widths are to be specified in units of the width of nMOS in the unit inverter.

c) Compute delays for both the branches of the fork.

d) Without changing inverter sizes, assume that the actual load capacitors in both the branches are higher by 10%. Now what are the delays and how much is the difference in delays of the two branches?

Q–24 (End-sem:2017)



A two step decoder for 6 address lines, producing 64 select outputs is shown in the diagram on the left. Combinations of 3 lines out of $A0, \overline{A0}, A1, \overline{A1}, A2, \overline{A2}$ are fed to the first bank of 8 three-input NAND gates. Similarly, combinations of 3 lines out of $A3, \overline{A3}, A4, \overline{A4}, A5, \overline{A5}$ are fed to the other bank of 8 three-input NAND gates. 64 two-input NOR gates then accept one line each from the two banks of NANDs to produce 64 select lines.

The select lines are required to drive heavy loads equivalent to 512 minimum inverters each. Assume that the γ value representing the ratio of p channel widths to n channel widths in an inverter to produce equal rise and fall times is 2, and the parasitic inverter delay is 2.5.

a) What is the the parasitic delay and the logical effort of 3 input NAND gates and 2 input NOR gates? (Parasitic delay can be taken to be proportional to the total capacitive load at the output node for a gate providing equivalent drive to a minimum inverter). Report the results in a table.

b) What is the optimum number of stages for minimum delay in this circuit, assuming that each of the input lines can drive 1 minimum sized inverter. Show the recommended configuration for the two step decoder with added inverters if necessary, so that overall delay is minimized.

c) Compute the scale factors and absolute geometries for all transistors in the design.

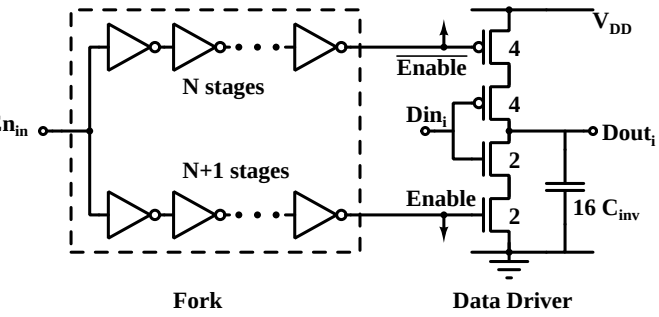
d) Compute the delay of each stage and the total delay for the decoder.

Q–25 (Mid-sem: 2015)

The figure below shows a tri-stateable driver which needs Enable and $\overline{\text{Enable}}$ signals generated by a fork. (A fork is a parallel path of N and N+1 inverters with nearly matched total

delays).

Each driver sees a load of 16 minimum sized inverters. 8 such drivers are to be driven by a single fork. The input En_{in} can drive a load of 2 minimum sized inverters. Sizes shown for transistors in the data driver are for a minimum sized driver; all of these can be scaled by any factor depending on requirements.



Assume that the mobility correction factor for PMOS transistor widths is 2 and the parasitic delay of inverters (p_{inv}) is 1.

a) Find the optimum stage effort for the whole chain by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0 \qquad \text{iteratively.}$$

Find the number of stages in the logic chain corresponding to this value of ρ. (This number should be adjusted to be an integer just less than the calculated value).

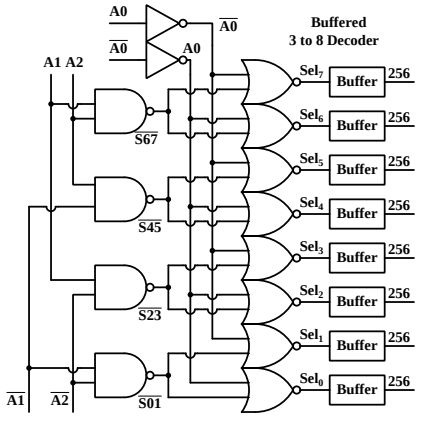
b) Distribute the total effort equally over the logic chain taking the N inverter branch in the fork. Find the transistor widths for all transistors. (The branch with N+1 inverters is not to be designed in this question).

Q–26 (Mid-sem:2018)

We want to design a 3 bit decoder where the decoder outputs have to be buffered to drive a load equivalent to 256 minimal inverters. Assume $\mathbf{p_{inv} = 2.0}, \gamma = \mathbf{2.2}$.

t

Assume that the 3 bits to be decoded and their complemented values are available as inputs and can drive loads equivalent to 4 minimal inverters. Decoding is done using a 2 step NAND-NOR circuit as shown. The decoded outputs Sel_i are buffered using inverters to drive the load. The number of inverters in the buffer is to be chosen to minimise the delay of the path involving NAND-NOR-Buffer. The number of inverters in the buffer can be even or odd, since true or complemented values of select outputs are equally acceptable.



a) Find the optimum value of ρ using Newton Raphson technique, starting with a guess value of $\rho = 4$.

b) Find the optimum number of stages in the path through NAND-NOR and Buffer. How many inverters should be used in the buffer?

c) Find the transistor sizes for NAND, NOR and all the inverters in the buffer such that the path delay is minimum.

– **[Q26: 1+1+3=5 marks]**

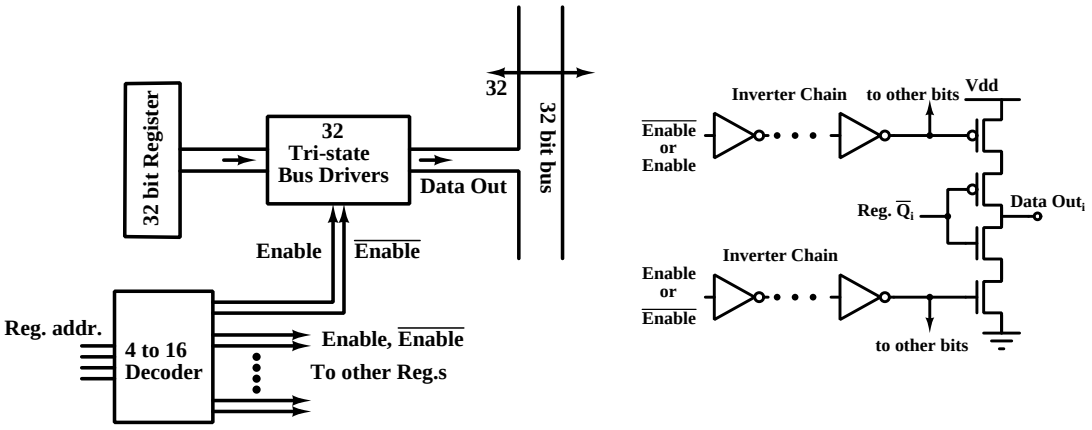
Q–27 (Mid-sem:2016)

In a given process, the ratio of p channel transistor width to the n channel transistor width in an inverter should be 4 to obtain equal rise and fall times. The parasitic delay of the inverter is 2.3.

a) The ideal stage ratio ρ is a solution to the equation $p_{inv} + \rho(1 - \ln \rho) = 0$. Evaluate the value of ρ using the Newton Raphson iterative technique, starting with a guess value of 4. Values for ρ, $f(\rho)$ and $f'(\rho)$ should be reported for each iteration, till you reach a convergence to 3 decimal places. (The reference section gives a brief description of Newton Raphson method.)

b) The figure on the left below shows a scheme to couple the output of a selected 32 bit register to a bus. Outputs from the decoder ($Enable$ / $\overline{Enable}$) drive inverter chains, which in turn drive 32 tri-state output stages (one for each bit). The figure on the right shows inverter chains driving output stages. (Only one output stage is shown, the remaining 31 are indicated by the arrow).

Each tri-state output stage drives a line of the 32 bit bus. The capacitive load presented by the bus line is equivalent to the input capacitance of 40 minimum inverters.



Find the number of inverters we should insert between the p channel transistor gate and the decoder output to minimize the total delay. (Notice that we have the option of selecting either the $Enable$ or the $\overline{Enable}$ output of the decoder depending on whether an even or an odd number of inverters are required).
Assume that $Enable/\overline{Enable}$ outputs of the decoder can drive two minimum inverters each.

c) Find the scale factor for each of the inverters inserted in the inverter chain. Compute the delay for the path from decoder output to the bus wire in units of τ.

d) How many inverters do we need to insert to drive the n channel transistor gates? Depending on which of $Enable$ or $\overline{Enable}$ was chosen for the part above, the other output of the decoder should be used for this chain. Adjust the number of inverters for this and compute the total delay for this path.
Compute the scale factors for all inverters in this chain.
There is no need to equalize delays through the two chains.

Q–28 (Class Test 2: 2015)

Consider a 32 bit adder with provision for a Carry-in signal. It is implemented as 8 groups

of 4 bit adders. Each group of 4 bit adders is a ripple carry adder but with a carry bypass for the group. The generation of control signals Generate/Kill/Pass is carried out in parallel. We use a simple delay model, in which the generation of control signals takes one unit of time, carry generation in each single bit full adder takes 3 units of time and when all Pass signals in a group of 4 bits are TRUE, the bypass around the group of 4 bits takes 2 units of time (inclusive of evaluation of bypass condition). What is the critical path for final carry generation for this arrangement? What is the total delay for this critical path?

Q–29 (Test-2:2018)
Show how carry select addition can speed up the addition of wide words. How can the addition be made faster by using a variable number of bits in each group of carry select adders? How are the number of bits chosen in each group for square root tiling of carry select adders? Why is this scheme called square root tiling?

Q–30 Describe the working of modified Booth algorithm for multiplication.
Illustrate it by working out the multiplication of unsigned binary numbers 100101 and 1101. All partial products should be shown as binary numbers and sign extension should be carried out where required. Show that by adding the binary partial products, you get the expected answer.

Q–31 A multiply and accumulate circuit uses a multiplier and an accumulator to compute expressions of the type $a_i = a_i + c_i x_i$ in a single unit. Since the process for multiplication implements multi-bit addition anyway, the bits of the accumulator just provide additional wires to the partial products at the corresponding weights.

a) Show a dot diagram for wire reduction using Wallace scheme for a multiply and accumulate circuit which has an 8 bit wide multiplicand, 6 bit wide multiplier and 15 bit wide accumulator. Use the scheme which does not produce a redundant MSB. (When two wires are left after allocating full adders, feed these through unless all bits at lower weights have a single wire or feeding through will exceed the capacity of the next layer.) What is the width of the final adder to be used after wire reduction to ≤ 2 wires at each weight?

b) (End-sem-2017)
Show how we can construct a circuit to carry out logical right shift, arithmetic right shift and rotate right, using only 2 bit muxes.
Assume the operand to be 8 bit wide.

c) Describe the operation of a bit serial multiplier, using a 4×4 multiplier as an example. Explain the operations which need to be carried out to take care of exceptions at the end of a row and show how these are implemented in hardware.

Q–32 End-sem: 2015

a) The delay of a single logic stage may be modeled as

$$d = gh + p$$

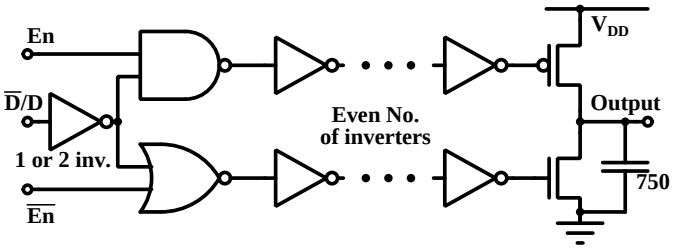
where the delays are normalized to the unit inverter delay. The only variable which depends on the size of the gate is h. Why are the logical effort g and the parasitic delay p in this expression independent of the size of the gate?

b) For multi-stage logic, the total delay is minimized when the stage effort $f = gh$ is the same for all stages. However, some times the total delay may be reduced further by inserting inverters in the logic chain. Derive the relationship which gives the optimum stage effort for each stage when we are free to insert a number of inverters in the logic chain.

c) Assume that in a given process, the value of the parasitic delay p_{inv} of an inverter is 1.6. How many inverters should we use in a chain to minimize the total delay while driving a load equivalent to 96 inverters, assuming that the input can drive a single minimum sized inverter. (We don't care if the number of inverters is even or odd).

Q–33 Explain how the modified Booth algorithm can be used to generate partial products for a multiplier. Derive the table to be used to decide what operation is to be carried out on the multiplicand based on 3 successive bits of the multiplier (with one bit overlapping).

Q–34 (End-sem:2016) Consider the driver for a bi-directional pad shown below:



The final load is equivalent to the input capacitance of 750 minimum sized inverters. The number of inverters after the NAND/NOR gates must be even. Depending on whether the *total* number of inverters required to be inserted for optimum delay is odd or even, we put either one or two inverters *before* the NAND/NOR gates to keep the number of inverters *after* the NAND/NOR gates to even. Correspondingly we connect either $\overline{D}$ or D to the input. This input should present a load of 1 minimum inverter to the previous stage.

Assume that the ratio of p-MOS and n-MOS widths for equal rise and fall times is 2 and the parasitic delay of inverters is 2 units. The parasitic delay of the NAND and NOR gates is 4 units, while that of the final driver stage is 2 units. Assume that the NAND and the NOR gates have the same size factor (*i.e.* actual transistor sizes in these gates are the same multiples of those in the minimum sized NAND/NOR gates).

a) How do we compute the optimum stage effort in a multi-stage logic chain when i) the number of stages is fixed and known and ii) the number of stages can be adjusted by inserting inverters. (You do not have to derive the equation for ρ.)

– **[2]**

b) Compute the value of ρ by solving the equation

$$p_{inv} + \rho(1 - \ln \rho) = 0$$

for the given value of $p_{inv} = 2$. Use the Newton Raphson technique to solve the equation. Start with a guess value of 4 and iterate till the solution converges. (All intermediate values for *rho* should be reported.)

c) In this question, we shall design only the upper branch of the circuit shown above. How many inverters should we use before and after the NAND gate? – **[1]**

d) Compute the sizes of all transistors in the logic chain of the upper branch (including the inverter(s) preceding the NAND gate) in units of minimum transistor width. Tabulate the results, giving the input capacitance of each stage in inverter units and the widths of nMOS and pMOS transistors in units of minimum transistor width.

e) What is the total delay from $\overline{D}/D$ input to the pad in the upper branch in units of τ? – **[1]**

Q–35 (Test-2:2016)

a) What is the motivation for using the generate/kill/propagate signals for carry in an adder? How are these produced?

b) Consider a 16 bit adder with carry bypass over every 4 bits. Show how the worst case delay will be substantially reduced because of breaking up of the critical path.

c) Describe the circuit of a dynamic CMOS implementation of a Manchester carry chain adder.

Dynamic CMOS logic has the problem that the output can be at the wrong value for some time after pre-charge (during evaluation), which can lead to malfunction. Will this circuit have the same problem? (Give reasons).

What makes it possible to use OR logic rather than the slower XOR to generate the propagate signal for this adder?

Q–36 (Test-2:2016)

a) Describe the wire reduction algorithm used in Dadda multipliers.

b) Consider a Dadda multiplier where the multiplicand is 8 bit wide while the multiplier is 5 bit wide. Show the wire reduction scheme for this multiplier using a dot diagram and give a brief description of the reduction at each stage.

c) Assume that a half adder provides its sum and carry outputs two units of time after the arrival of the last of its inputs, while a full adder takes 3 units of time to generate sum and carry after the arrival of the latest input. Assume that all partial product bits for the 8×5 multiplier are ready at time 0.

Redraw the dot diagram for the 8×5 Dadda multiplier, placing the time of arrival of each bit in brackets next to each dot.
(At every reduction stage, one can choose which wires go to a full/half adder and which ones are passed through. You should make this choice such that the worst case delay for reaching the conventional adder stage is minimized.)

d) Assuming that a ripple carry adder using the same half adder and full adder will be used for the final addition, what is the worst case time at which the product will be ready?

Q–37 (End-sem:2016)

a) Show the wire reduction scheme for an 8×8 Dadda multiplier. Display the scheme using a table formatted as below:

No.	Stage 1			No.	Stage 2			No.	Stage 3			...
Wires	F	H	P	Wires	F	H	P	Wires	F	H	P	...
...	..	..	..	...	..	..	..	...	..	..	..	...

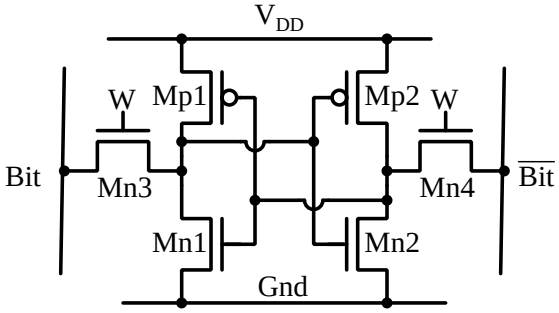
Where F is the number of full adders, H is the number of half adders and P is the number of passed through wires. The table will have a row for each weight, starting with the least significant bit.

There should be no more than 2 wires for any weight when the reduction is complete. (Notice that for each stage, the incoming wires should be equal to $3F + 2H + P$ and the outgoing wires should equal $F + H + P + F_{lw} + H_{lw}$ where lw represents the lower weight row of the same stage). – **[3]**

b) It is easy to modify the multiplier wire reduction scheme to implement a "Multiply and Accumulate" function. Show the wire reduction scheme for a multiply and accumulate circuit which computes $A \times B + C$ using the Dadda scheme. It should multiply two 8 bit operands and add the product to a 16 bit number. At the end of reduction, there should be no more than two wires at any weight. Use the same tabular format as described above.

c) Find the number of *additional* reduction stages, half adders and full adders required to convert the multiplier to a Multiply and Accumulate circuit. (The final fast adder with carry propagation is not to be included in this).

Q–38 (End-sem:2017)



A six transistor static memory cell includes two cross-connected inverters and two N type access transistors (MN3 and MN4) connected to bit and $\overline{\text{bit}}$ lines. Gates of the access transistors are raised to V_{DD} by the word line W when the row containing the cell is selected.

a) Why do we use bit as well as $\overline{\text{bit}}$ lines? What would be the problem if we used only one of these?

b) How is a "butterfly" diagram used for describing the behaviour of a cascade of two inverters? How is it used to find the stable and meta-stable equilibrium points of cross connected inverters?

c) The capacitance of the bit line is 2pF and it is initially charged to V_{DD}. Transistor MN1 is ON while MN2 is OFF. When the word line goes to V_{DD}, the bit line needs to be discharged to 1.6V for reliable reading by the sensing circuit. Assume that $K_n = 100\mu\text{A/V}^2$ for MN3.

i) If Mn1 is twice as wide as Mn3, find the voltage at the source of MN3 just as the discharge current starts flowing.

ii) Assuming that the current through MN3 remains constant at its initial value, find the time required to discharge the bit line to 1.6V.

d) Describe the sequence of operations during read cycle in a static RAM. How is it possible for the stored data to be destroyed during read if the RAM cell is not carefully designed?

Q–39 (End-sem:2015)

a) What is a C element? Describe the working of a dynamic C element and show that it works effectively as an AND function of events on its inputs.

b) Show how we can convert the dynamic C element into a static circuit. Describe the operation of the static C element.

c) Draw the circuit for a select element and show how it works. (The select element is just an inverting mux).

d) Describe the working of an event sensitive data latch using three select elements and two control inputs. Show how the use of two control inputs permits us to make this latch event sensitive.

Q–40 (End-sem:2016)

a) What do we mean by "skew" and "jitter" in the clock. – **[1]**

b) Show how skew and jitter may impact the performance of a synchronous circuit.

c) In some cases skew is intentionally added to the clock of a particular stage in a VLSI circuit. How can it help in improving the performance of the circuit?

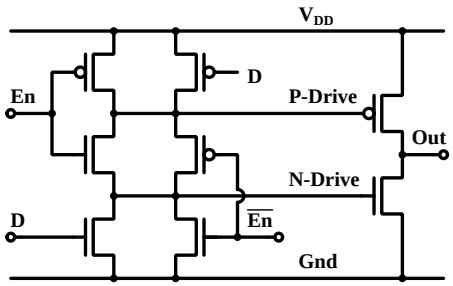
d) Describe the H-tree and grid distribution networks for the clock.

e) Apart from the clock, why is it important to have low skew on the power-on reset signal distribution on a chip?

Q–41 (End-sem: 2016)

a) Describe the input protection circuit commonly used with pads. What is a clamp circuit and why is it required? What kind of clamp circuits are commonly used?

b) Consider the compact NAND-NOR driver for a tri-stateable output shown below.



i) Why is this circuit preferred over the four transistor tri-stateable inverter for driving output pads?

ii) Show the equivalence of this circuit to the discrete NAND and NOR by circling the appropriate sub-circuits in the schematic above. Show that the electrical behaviour of this circuit is identical to the discrete NAND-NOR.

iii) What should be the minimum geometries of the six transistors in the compact NAND-NOR circuit, so that both P-Drive and N-Drive are driven by at least the drive strength of a minimum inverter. Assume that the pMOS has to be twice as wide as the nMOS to provide equal drive and the ON resistances of pMOS and nMOS transistors forming a pass gate should be equal.

(Hint: Consider the worst case drive requirements of P-Drive and N-Drive for $En = '0'$ and for $En = '1'$ with $D = '0'$ or $'1'$.)

Q–42 (End-sem: 2017)

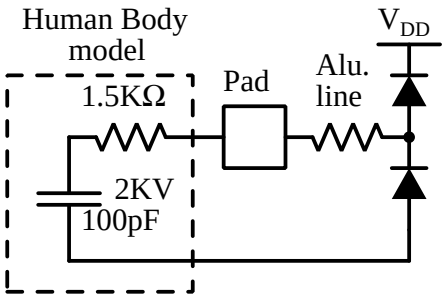
a) At the input pads of a CMOS IC, we need to protect against high electrostatic voltages during handling and voltage excursions below ground and above the supply voltage during operation. What kind of device structures are used for protection against these hazards?

b) A chip designed for a supply voltage of 3.3V has to accept inputs from a source which provides logic levels of 0 to 1.8V. How can we use CVSL logic to translate the low swing logic at the input pads to the higher swing logic for use inside a chip? (Assume that a low voltage supply compatible with the low swing input is available on-chip).

c) How does the large output driver of an output pad also act as a protection device?

d) In a bidirectional pad, the output drivers need to be tri-stateable. Why is a NAND-NOR based driver structure preferred over 4 transistor tri-stateable inverters at the output?

e) The human body model is used to emulate the electrostatic hazard to integrated circuits due to handling by human beings.



A 100 pF capacitor is charged to 2KV and discharged through a series resistor of 1.5KΩ and the IC input. Inside the IC, the bond pad size is 70μm $\times$70μm, which leads to an aluminium line which is 100μm long and 1μm wide. Aluminium thickness is 0.5μm in this layer. The aluminium line leads to a protection diode, whose breakdown voltage is negligible compared to 2KV, so the line may be considered to be terminated to ground.

Estimate the temperature rise in this line when the capacitor is discharged through the external 1.5KΩ resistor and this line to ground. The following assumptions may be made:

– The pad contributes negligibly to the resistance of the line.
– The entire mass of aluminium in the pad and line will be uniformly heated.
– No heat is lost to other structures in this short duration.
– The entire energy stored in the capacitor is used for heating the external resistor, pad and the line.

The relative density of aluminium is 2.7, its specific heat is 0.9J per gram per degree K and its resistivity is $2.7 \times 10^{-6}\Omega$Cm.