Low Power VLSI Circuits and Systems

Prof. Pro. Ajit Pal

Department of Computer Science and Engineering

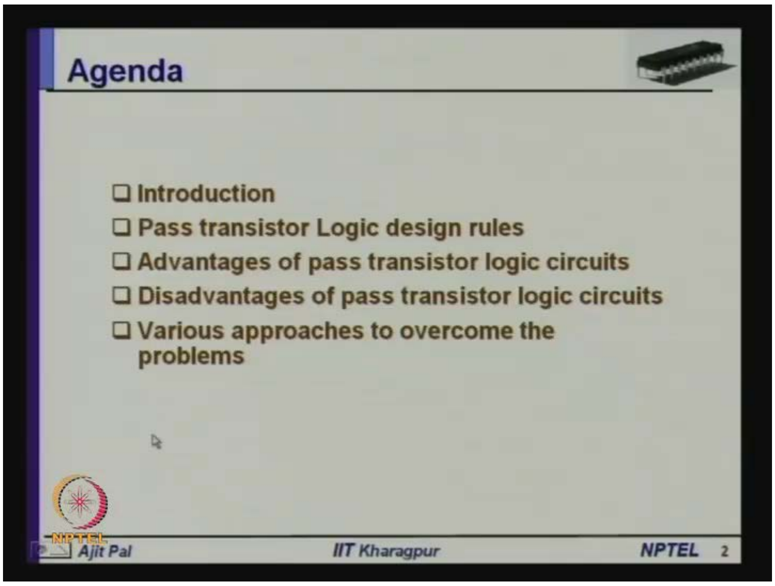
Indian Institute of Technology, Kharagpur

Lecture No. # 14

Pass Transistor Logic Circuits - I

Hello and welcome to today's lecture on pass transistor logic circuits. In the last couple of lectures we have discussed how boolean functions can be realized by using static CMOS circuits and dynamic CMOS circuits. Those are essentially known as gate logic, because inputs are applied to the gate, and you take the output from source or drain of the transistors. And earlier we had discussed about the use of transistors as a switch, today we shall discuss how you can realize boolean functions using transistors as a switch, this is typically known as pass transistor logic circuits.

(Refer Slide Time: 01:06)



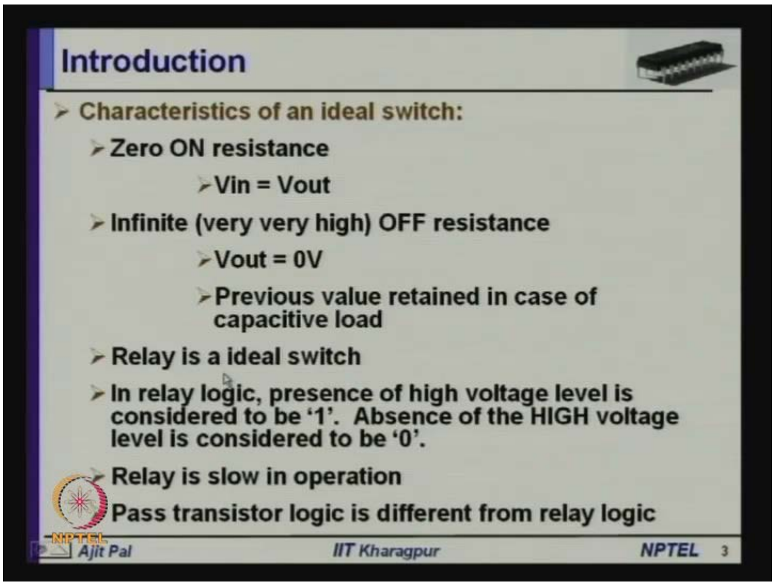
Agenda

- ❑ Introduction
- ❑ Pass transistor Logic design rules
- ❑ Advantages of pass transistor logic circuits
- ❑ Disadvantages of pass transistor logic circuits
- ❑ Various approaches to overcome the problems

Ajit Pal IIT Kharagpur NPTEL 2

So, here is the agenda of today's lecture after giving a brief introduction I shall discuss about pass transistor logic circuit design rules, then discuss about their advantages and disadvantages, and also we shall discuss how various disadvantages can be overcome.

(Refer Slide Time: 01:29)



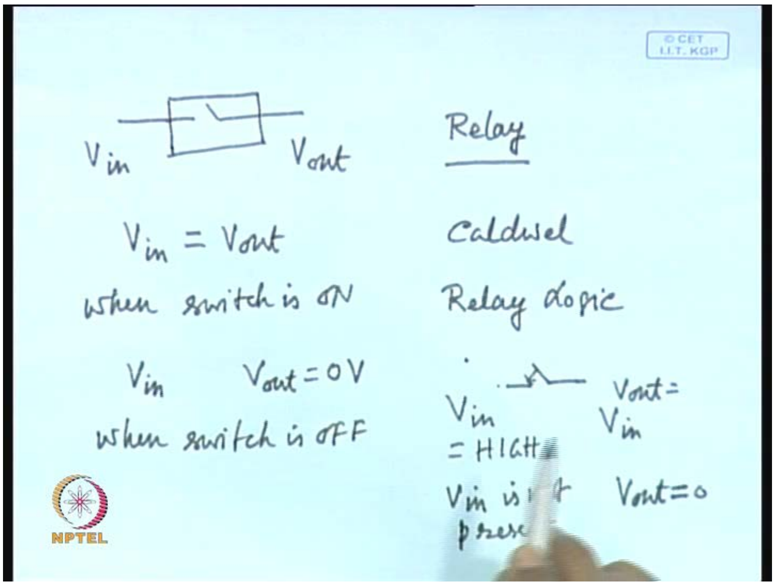
Introduction

- Characteristics of an ideal switch:
  - Zero ON resistance
    - Vin = Vout
  - Infinite (very very high) OFF resistance
    - Vout = 0V
    - Previous value retained in case of capacitive load
  - Relay is a ideal switch
  - In relay logic, presence of high voltage level is considered to be '1'. Absence of the HIGH voltage level is considered to be '0'.
  - Relay is slow in operation
  - Pass transistor logic is different from relay logic

Ajit Pal IIT Kharagpur NPTEL 3

So, let us start with a brief introduction which is essentially a recapitulation of what is discussed earlier. We know that an ideal switch have 0 resistance and that means, whenever you apply an input.

(Refer Slide Time: 01:48)

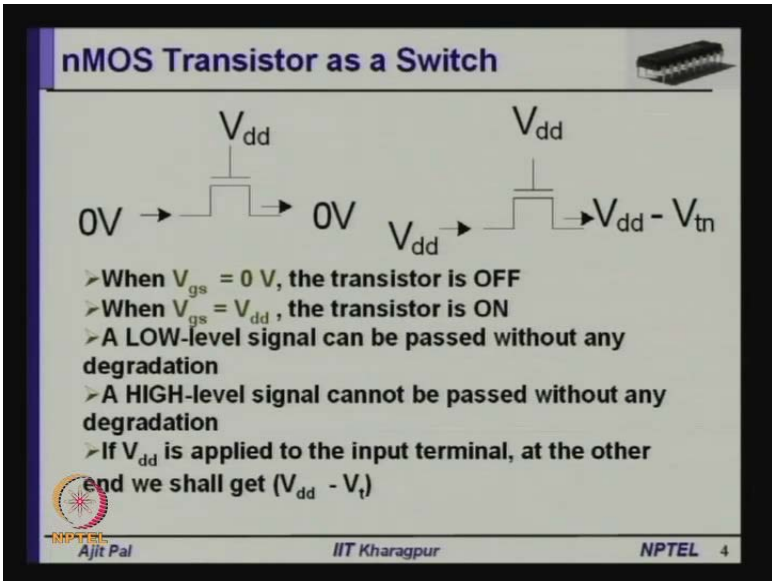


It will reach attenuated say suppose this is a switch, whatever is inside let us forget, so let me represent it by this a switch, and you are applying an input V in, and you are getting an output. Ideally, V in should be equal to V out when switch is on, and whenever switch is off then V in should not be V what irrespective of V in V out should be equal to 0, 0 volt or 0 so when switch is off. So, this is the ideal characteristics as you know, and we also know that a relay which is an electro mechanical device can act as a ideal switch as an ideal switch; however, its switch it is very slow, so switching characteristics is not acceptable in present day circuits.

There was a classical book still it may not be available now, written by Caldwell, who provided a chapter where you can realize circuits using relay logic; that means, using relays as the building block bullion functions can be realized, so relay logic is provided And in case of relay presence of input; that means, whenever you apply say suppose this is the relay, and if you apply an input which is equal to say high; that means, if input is available then you get if the switch is close you get V, whatever is V in you get as V out; that means, V out is equal to V in respective of high or low.

Whenever, V in is not present; that means, there is no input applied to it than V out is equal to zero; that means, in case of relay logic absence of input provides a 0 output; that means, that absence of input is treated as low input and at the output you get 0 whenever the switch is closed. So however, we shall see pass transistor logic is different from relay logic, because we cannot really use the techniques that has been used to realize a bullion functions by using relay logic to pass transistor logic, and in what way they differ we shall discuss in detail.

(Refer Slide Time: 04:58)



nMOS Transistor as a Switch

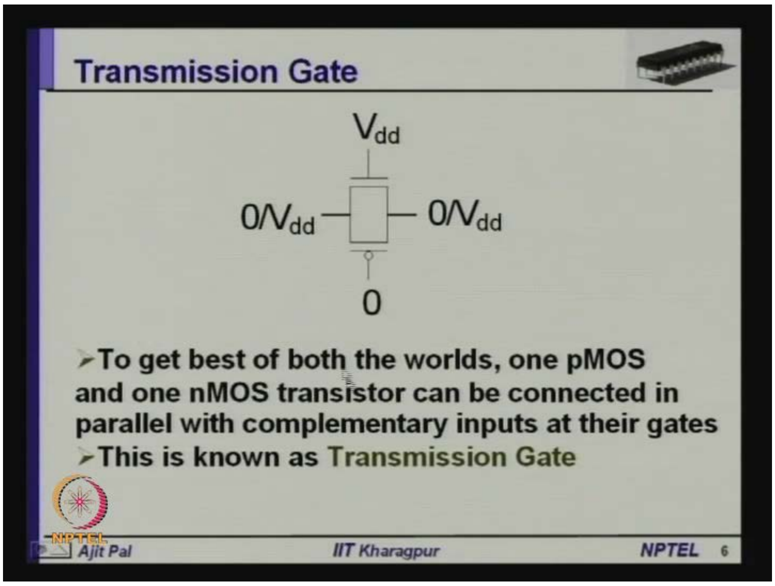
- When V_{gs} = 0 V, the transistor is OFF
- When $V_{gs} = V_{dd}$, the transistor is ON
- A LOW-level signal can be passed without any degradation
- A HIGH-level signal cannot be passed without any degradation
- If V_{dd} is applied to the input terminal, at the other end we shall get (V_{dd} - V_t)

Ajit Pal IIT Kharagpur NPTEL 4

But, before that as I told we shall be using n MOS transistors as switch very quickly, let us recapitulate what we discussed earlier. We know that an n MOS transistor as a switch can be used, and when the gate voltage is 0 than the transistor is off, and when the gate voltage is 0 than the transistor is on, and when gate volt is V dd the transistor is on, and we also know that a low level signal can be passed through the switch without any degradation, as it is shown here if 0 volt is applied you get 0 volt, and a high level signal cannot be passed without any degradation as we know. The output you will get V dd minus V tn, where V tn is the threshold voltages of the n MOS transistors.

So, this is the characteristics of a n MOS transistor as a switch, and similarly if we use a p MOS transistor as a switch when V gs is equal to V dd the transistor is off, if you apply V dd here the transistor is off, and when V g s is equal to 0 volt the transistor is on, and as you know a low level signal cannot be passed without any degradation, because you will get a voltage which is equal to V tp; I mean, absolute value of V tp you apply a volt V dd here; that means, if it if you apply 0 volt here you will not get 0 here but you will get a minimum voltage of V tp, because this transistor will turn off when the voltage gets discharged beyond V tp point. Than high level; however, a high level signal can be passed without any degradation as we know, so if we apply V dd you get V dd with a 0 and the 0 applied to the gate.

(Refer Slide Time: 06:45)



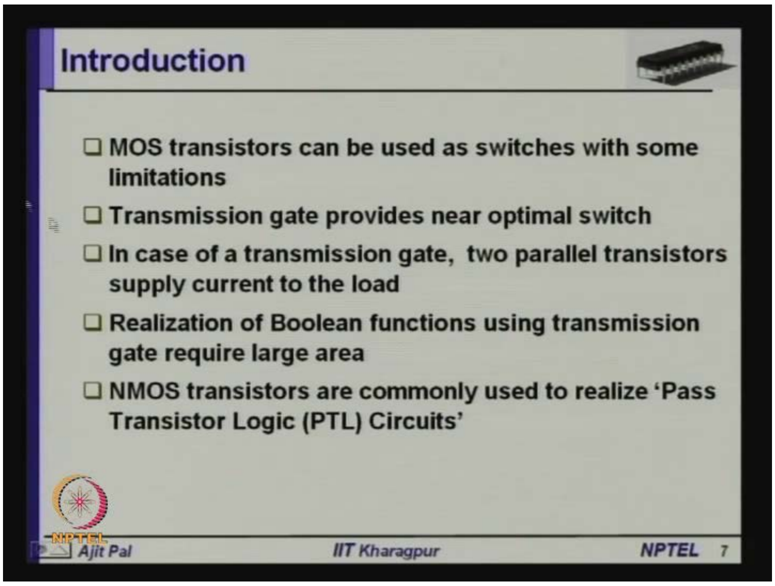
Transmission Gate

- To get best of both the worlds, one pMOS and one nMOS transistor can be connected in parallel with complementary inputs at their gates
- This is known as Transmission Gate

Ajit Pal IIT Kharagpur NPTEL 6

So, minimum voltage that it can pass is V tp, we can make a conclusion from this. And we can combine the advantages of n MOS and p MOS transistors to realize a transmission gate, where both 0 and high level, low and high level can pass without any degradation, and where we use two pass transistors, two transistors one n MOS and one p MOS in parallel; however, this will require control signals you can see complementary control signal; that means, if this is the switch, this is the transmission gate, than if you apply some say control signal if you apply here this is your V in, this is V out; that means, if you apply 0 here you have to apply one; that means, you may have to use an inverter, and if this is the control signal coming from some other source to apply to the two gates, so you require either complementary signals or you have to use an inverter to compliment the control signal to be apply to the other input. So, the number of transistors that you require in realizing a switch is quite large in case of transmission gate used as a switch and that that.

(Refer Slide Time: 08:08)



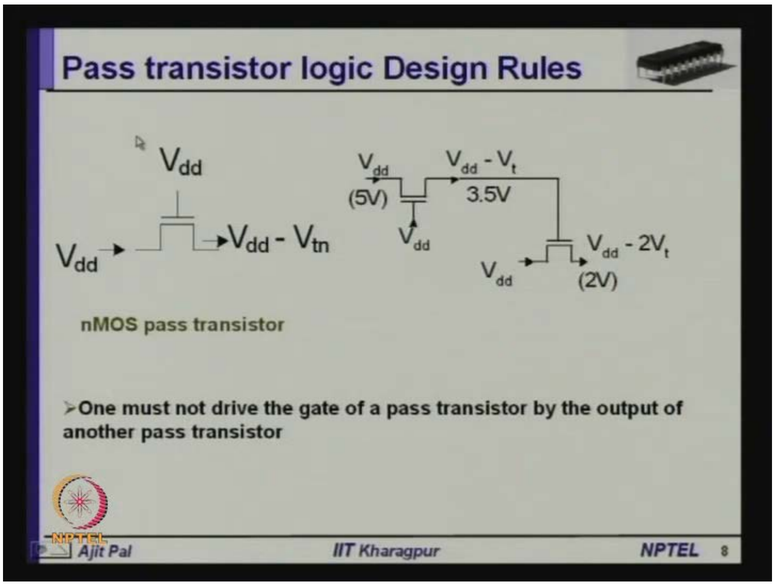
Introduction

- □ MOS transistors can be used as switches with some limitations
- □ Transmission gate provides near optimal switch
- □ In case of a transmission gate, two parallel transistors supply current to the load
- □ Realization of Boolean functions using transmission gate require large area
- □ NMOS transistors are commonly used to realize 'Pass Transistor Logic (PTL) Circuits'

Ajit Pal IIT Kharagpur NPTEL 7

So, we can say that… You can from this we can… Our observation is most transistors can be used as switches with some limitation, transmission gate provides near optimal switch in case, of transmission gate as we have seen two parallel transistors supply current to the load, and realization of Boolean functions using transmission gate requires larger area, because as we have seen you not only require two transistors to realize a switch, to generate control signals you may require another inverter. So, that is the reason why instead of using transmission gate, n MOS transistors are commonly used to realize pass transistor logic circuits; that means, in switch logic whenever you realize Boolean functions using transistor as switch usually a, and usually an n MOS transistor is used as a switch, instead of using transmission gate although transmission gate is closer to an ideal switch. With this brief introduction let us now see in what way; I mean, how you can realize pass transistor logic circuits.

(Refer Slide Time: 09:20)



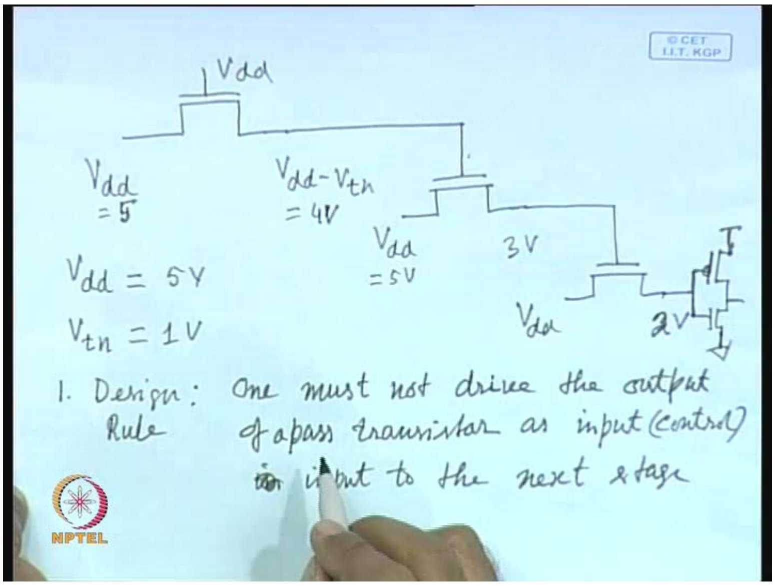
Pass transistor logic Design Rules

- One must not drive the gate of a pass transistor by the output of another pass transistor

Ajit Pal IIT Kharagpur NPTEL 8

First, let us discuss about some logic design rules as we know. So, whenever you are using n MOS transistor as a switch than

(Refer Slide Time: 09:35)



$V_{dd} = 5V$

$V_{tn} = 1V$

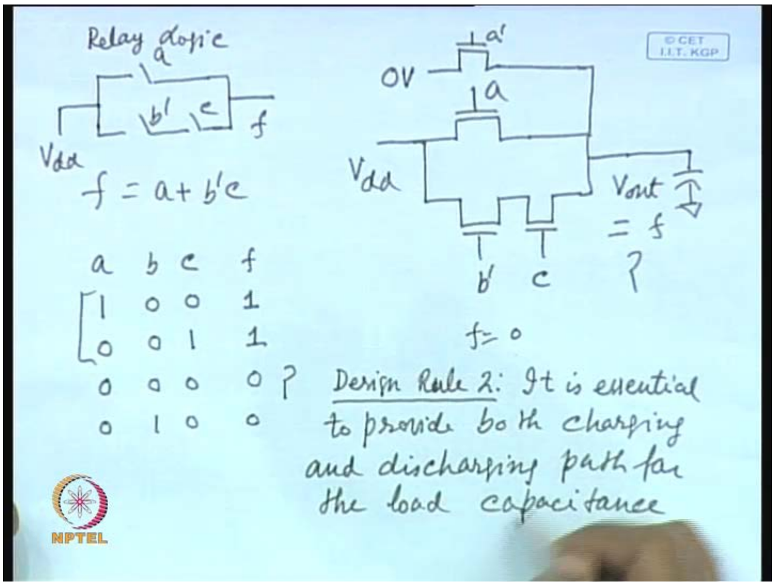
1. Design Rule: One must not drive the output of a pass transistor as input (control) ~~to~~ input to the next stage

if we apply V dd here, let us assume the transistor is on, so if you apply V dd here the maximum voltage that you can use here is V dd minus V tn. Suppose, V dd is equal to 5 volt and V tn is equal to say 0.2 V dd let us assume it is one volt, than what will happen? If this is equal to 5 volt here you will get four volt, now if this is used as a control signal to another stage, what will happen here if we apply V dd is equal to 5 volt, here we shall get V dd we shall get 3 volt, the reason is you see here we are applying 4 volt; so that means, the transistor will turn off when this voltage will charge to 3 volt level. In a similar way suppose, this we apply as a control signal to another stage than what will happen, and a V dd will get degraded too much and here we shall get 3 volt; that means, if the output of a pass transistor logic I mean n MOS transistor logic is switch, if it is applied as a input to a gate to the next stage than the voltage further degrades; that means, so here it will be 2 volt, so 4, 3 and 2 so 2 volt.

So, it is getting degraded because if it is 3 volt here, this voltage this point cannot change beyond 2 volt. So, we can see there is severe degradation, and it may so happen that the next stage may not treat this if we apply to some gate logic circuit, than it may not be treated as a high level. Moreover, if it is driving to say suppose if it is driving to a inverter. Than what will happen, the drive that it will provide to this n MOS transistor will be quite low, and as a consequence the because the drive even if it is treated as a high level, the drive to this transistor will be quite low when the input is high, and as a consequence this will affect the switching characteristic of the device; that means, with 2 volt as the input voltage to the gate as you know, the saturation current is equal to V dd minus V tn square proportion to that, and as a consequence the current will be significantly less, and as a consequence the switching characteristic will be very inferior.

So, from this we can say that one first design rule that you should use is one must not drive, so this is the design rule one must not drive the output of a pass transistor of a pass transistor one must not drive the output as input rather control input we should say control input…. Control input to the next stage; that means, what we shall do we shall never use this; that means, this signal we shall never use at control signal to the next stage, so either we shall put an inverter here buffer to restore the level and then we shall apply to the next stage. So, we can this is the first rule that you have to follow while realizing circuits using n MOS as switch. This is the first rule that you have to follow, than the second rule we shall come from this observation.

(Refer Slide Time: 14:08)



$f = a + b'c$

a	b	c	f
1	0	0	1
0	0	1	1
0	0	0	0 ?
0	1	0	0

$f = 0$

Design Rule 2: It is essential to provide both charging and discharging path for the load capacitance

Suppose, we have to realize a function f is equal to a plus b square c, conventionally if we realize it by a relay logic the representation is like this, this is controlled by a, and this is controlled by b dash, and this is controlled by a another switch c, and we get the output from here. That means, if we apply an input here and through these two switches; that means, when a is one this switch is close so signal will pass through it, and when b das is equal to one and c is equal to one the signal will pass through it, and this is how it will work. And this is the realization to realize this function f to of course, here you will apply V dd or I mean in terms of our MOS circuits; so that means, if we apply here V dd, and then use these three switches we shall will realize this function f. So, as we can see whenever the switches are off we shall be getting 0 volt, either a is I mean either b is off, or c is off, or a b das and c das all are 1, in that all are 0, in that case also the switch will be off; that means, if there is no path than output will be 0, so this is how you can realize by using relay logic.

Now, suppose we use the same approach to realize the function by using pass transistor; Let us assume, we use three n MOS transistors three n MOS transistors here we apply a, here we apply b das, here we apply c, and following the convention of relay logic here we apply V dd, and here we are expecting the output V out, Question is, will V out be equal to, what will be V out will it be equal to f? That is the question, will it be equal to f the function that is shown here, what is the answer?

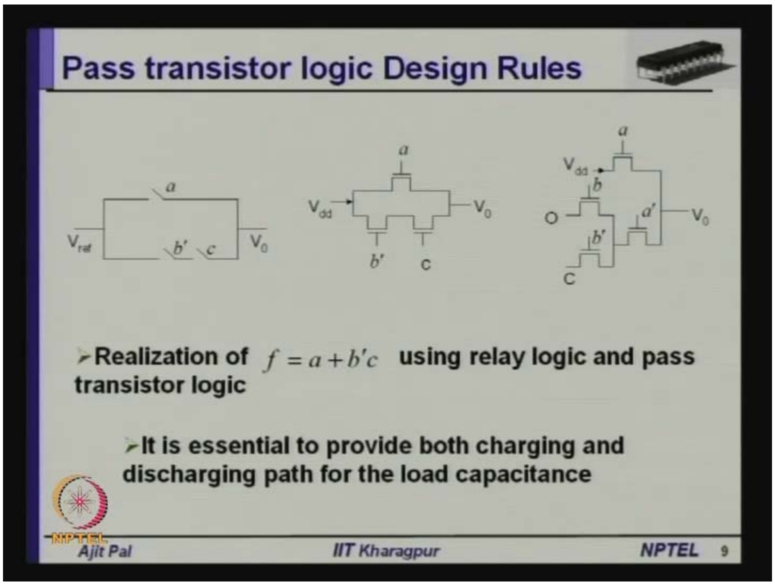
Unfortunately, if we realize the function based on relay logic, it will not produce correct output. Why it will not produce correct output? The reason for that is, you see it will produce correct output, so for as the high output level is concerned; that means, whenever this switch is closed it will produce V out of course, it will not produce V out but V out minus V dd, let us assume we shall take it as a high level, but let us assume the case whenever so; that means, whenever the output is say a b c, so when the input is 1 0 0 than it will produce one, or whenever it will the output is say 0 0 1 than it will produce 1, 0 0 1 b das is equal to so this path is closed it will produce one; that means, this is one, this is one then also it will produce one, but about the case say whenever we apply say 0 0 0 all are 0, or let us assume whenever the input say 0 1 0.

I am not writing all possible combinations. So, for these inputs it is supposed to produce 0, let us assume after applying anyone of these inputs we apply 0 0 0, than what will happen? Shall we get 0 at the output, the answer is no, why? The reason for that is now whenever, after applying anyone of the inputs, whenever we apply either 0 0 0 or 0 1 0 what will happen? There is no path to V dd yes this is off, this is off, this is I mean whenever, it is 0 0 0 neither this path, nor this path is close, so you are supposed to get a 0 here. But unfortunately, capacitor is now charged to V dd because of the previous input.

So, as you know it retains the charge at the output, and as a consequence it will continue to produce high if this input is applied after this, or this input is applied after any one of these two input; that means, after producing output I mean producing high at the output, if any input combination is applied for which we are suppose to get 0 it will not produce 0. In that case, what is the solution? Solution is, that we have to realize the circuit in such a way that it will provide path for 0 also; that means, for one it is getting path through this; that means, to charge this capacitor to high level the path is this I mean, when a is one this is the path, when b das c is one this is the path, but what about when f is suppose to be 0. So, we have to provide a path for discharging the capacitor, how can you provide that? That means, we have to realize, we have to put additional path such that the output is I mean it should produce 0 by discharging the capacitor when the input combination is input combination is such output is 0.

So, what we can do we can provide this kind of things, say a das is connected and we provide 0 here, 0 means 0 volt. Now, as you can see if we apply any other input combinations, than you will see that it will provide for this input, or this input it will produce a 0 output, because this switch will now close and discharge the capacitor to 0 level; that means,… So, what is the design rule two? what is our design rule two Design rule two is, it is essential to provide both charging and discharging path for the load capacitance; that means, when the output is suppose to be 0 there should be a discharge path of the capacitor, when the output is suppose to be one there will be charging path. So, this is how we have to use, realize the circuit by using pass transistor logic.

(Refer Slide Time: 21:36)



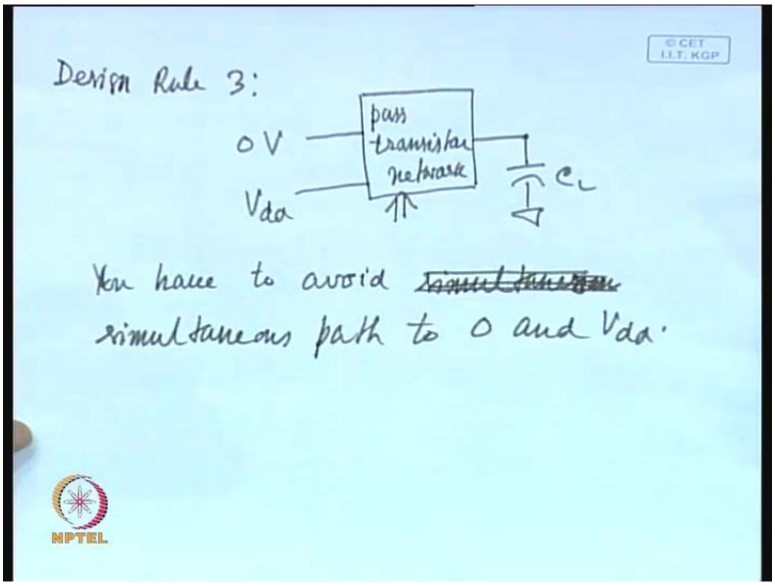
Pass transistor logic Design Rules

- Realization of $f = a + b'c$ using relay logic and pass transistor logic
- It is essential to provide both charging and discharging path for the load capacitance

Ajit Pal IIT Kharagpur NPTEL 9

So, this is the second rule that we have to follow while realizing circuits by using pass transistor logic. What about the third rule, is there any other problem here? Actually, another rule is there design rule three, design

(Refer Slide Time: 21:46)



Design Rule 3:

You have to avoid ~~simultaneous~~ simultaneous path to 0 and Vdd.

rule three comes from say suppose, this is a pass transistor network pass transistor where you will be applying 0 volt and V dd as input and of course, those various inputs you will apply as control signals, and here it will produce output where let us assume we have a load capacitance here. Now, it may so happen that the output there is a path directly to this 0 volt and also there is a path directly to the V d d; that means, both paths are realized advertently or inadvertently, it may so happen because of some mistake in the design, what has happen there is path both from 0 from the low level as well as from high level, this will lead to an output which is undefined; that means, it will depend on the… They will act as a kind of potential divider and you will get an output which is neither 0 nor 1, so we will get some undefined output.

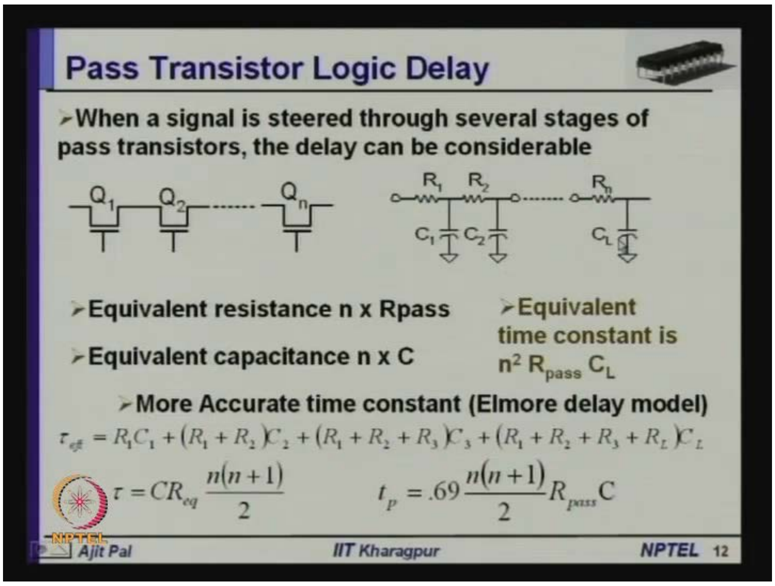
So, what is you design rule two? So, you have to avoid simultaneous path simultaneous path to 0 and V dd, how later on we shall see. So, this is your design rule we followed, and if we realize circuits by using three design rules than it will produce correct output, and you can realize any general Boolean function. Later on we can discuss on detail how we can realize general Boolean using pass transistor logic, but before that let us look at the advantages and disadvantages. First advantage is ratio less, why it is ratio less? It is ratio less because the output is not dependent on the size of the n MOS or p MOS transistors. So, unlike your gate logic realized by using n MOS circuit or pseudo-n MOS circuit, where the w by L ratio of the pull-up to pull-down device effects the output voltage at low level, and we have to maintain some ratio such that you get proper low level output. Here of course, that is not true; that means, it is ratio less, what is the impact of this? The impact is you can realize circuits with transistors of minimum dimension; that means, 2 lambda by 2 lambda, lambda is the minimum feature size for a particular process technology.

You can realize circuits by using transistors of minimum dimensions; that means, 2 lambdas by 2 lambda, both for n MOS and p MOS transistors if you require. So, in this case; however, you are you realizing using only n MOS transistors, so all of them will be of smallest dimensions, smallest dimension means 2 lambda by 2 lambda. And obviously, it will have smaller area this will lead to smaller area of realization this is the outcome of this. Second is lower area due to smaller number of transistors, there is another important feature of pass transistor logic circuit is that, you can realize Boolean functions with smaller number of transistors, transistors are of smaller size and also you will require smaller number of transistors, later on we shall discuss about the realization of some benchmark circuit say (()) or another things. You will see the pass transistor logic realization will require smaller number of transistors, and as a consequence area required will be small. So, lower area due to smaller number of transistors, than lesser power dissipation, there is no static power dissipation and short-circuits power dissipation.

As you have seen here although we are having path 2 0 volt, path 2 V dd, but as we know only one of the two paths will be closed; that means, either there will be path to 0 or there will be path to V dd, there will be no path to the output simultaneously, to 0 0 and V dd, as a result it will never form, never need to short-circuit power dissipation, we know that short-circuit power dissipation occurs because there is a path from V dd to ground, but here that kind of path will never be established of course, if we follow design rule three. That means, if we follow design rule three only than this will happen, if we do not follow it than this problem may arise, but you have to follow design rule three whenever you are realizing circuits using pass transistor logic.

So, these are the advantages and obviously, you will be tempted to realize digital circuits or realize Boolean functions by using pass transistor logic, but before we venture to do that let us have a look at the disadvantages. First disadvantage is higher delay in long chain of pass transistors, this is the first disadvantage.

(Refer Slide Time: 28:20)



Pass Transistor Logic Delay

- When a signal is steered through several stages of pass transistors, the delay can be considerable

Q1
Q2
Qn
R1
R2
Rn
C1
C2
CL

- Equivalent resistance n x Rpass
- Equivalent capacitance n x C
- Equivalent time constant is $n^2 R_{pass} C_L$
- More Accurate time constant (Elmore delay model)

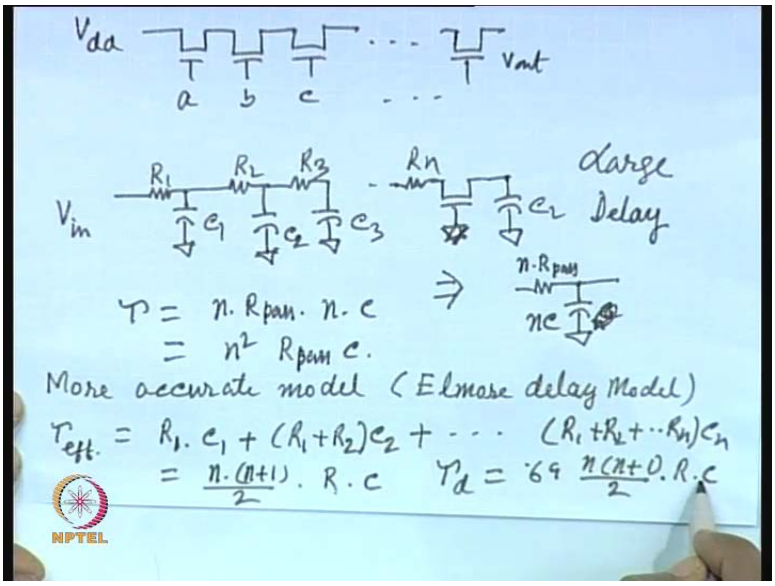
$$\tau_{eff} = R_1C_1 + (R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3 + (R_1 + R_2 + R_3 + R_L)C_L$$

$$\tau = CR_{eq}\frac{n(n+1)}{2} \qquad t_p = .69\frac{n(n+1)}{2}R_{pass}C$$

Ajit Pal IIT Kharagpur NPTEL 12

You see, whenever you apply a number of transistors in series, say you will be applying let us assume V dd, and you will be realizing say there may be large number of inputs.

(Refer Slide Time: 28:25)



$$\tau = n.\,R_{pass}.\,n.\,C$$
$$= n^2\,R_{pass}\,C. \quad \Rightarrow$$

More accurate model (Elmore delay Model)

$$\tau_{eff.} = R_1.\,C_1 + (R_1+R_2)C_2 + \cdots\ (R_1+R_2+\cdots R_n)C_n$$
$$= \frac{n.(n+1)}{2}.\,R.\,C \qquad \tau_d = .69\,\frac{n(n+1)}{2}.\,R.\,C$$

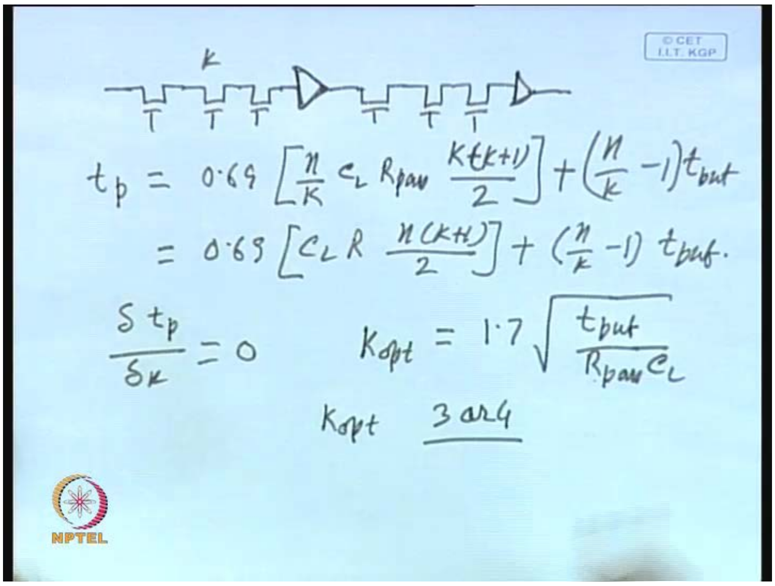
So, let us assume here you apply b c d and so on. So, this is your input and this is your output V out; that means, whenever the signal is passing through a large number of pass transistors, what is the outcome of this? When the signal is passing through a large number of pass transistors it will lead to long delay, because each transistor is associated with a resistance and a capacitance, we can model this as a somewhat like this, a resistance and a capacitance, and a resistance and a capacitance, and a resistance and a capacitance, capacitance is connected to ground, this is connected to ground, if we have n transistors in series we shall have n n r and c and this is connected to your load resistance. So, you are applying say input here, and the it is passing through n number of resistors and capacitor to the output, this is your C L. What is a consequence of this? This will lead to large delay. So, how do you estimate the delay whenever the signal is passing through a large number of transistors?

This can be modeled as a single equivalent to, as if you have got one resistance and one capacitance. So, these network can be modeled as, this part can be modeled as a single resistance and a single capacitance, where the resistance value is equal to n into say r pass, this is the resistance of the pass transistors, and this is c is equal to I mean n into c n c equivalent capacitance. So, equivalent capacitance as if n c and equivalent resistance is n r passes. So, what is the time constant tau? Time constant tau in this particular case is equal to n into r pass into n into c; that means, this is equal to n square r pass into c. So, whenever we follow this simple model we find that delay is proportional to n square and obviously, for 4 input delay will be 16 times that of a single transistor, or when there are 8 transistors in casket delay will be 64 times obviously, it will be very large.

However, it has been found that it is not so high, because this model that you have used is a kind of pessimistic model and not correct model. A more accurate model can be used more accurate model based on what is known as L more delay model L more delay model. If we use this L more delay model, this can be modeled as if this time constant tau will be equal to effective you can say, is equal to this is say R 1, this is R 2, R 3 in this way, let us assume there are in transistors and similarly, let us assume this is C 1, this is C 2, this is C 3, later on we shall assume that they are all same. So, than this delay and time constant will be equal to R 1 into C 1 plus, this 2 resistances are in series to which is charging or discharging das capacitor. So, R 1 plus R 2 into C 2 plus, in this way we can say R 1 plus R 2, in this way R n into c n, we assume that all r and n all are same, than this becomes equal to this will become equal to n into n minus 1 by 2 into r into c, this can be if all c are same which is equal to c, and all r are same r than the value will be like this; that means, R C into n into n minus 1 by 2.

So, this will be this will be the time constant. So, the time delay based on this model will be equal to 0.69 n into n plus 1 by 2, it will be n into n plus one I believe plus 1 by 2 into r into c, so this will be the time constant. So, even with this value let us assume the value of n is equal to 4, so when the value is equal to n is equal to 4 this is 4 into 5 by 2; that means, it is 4 into 5 by 2; that means, value is 10; that means, it will be 10 into 0.69, so roughly 7 R C. So, it may not be 16, but 7 R C; that means, when the number of any value of n is equal to 4, instead of 16 R C it is becoming 7 R C. So, in this way we find that the delay is increasing, may not be at the rate of I mean square following square law, but it is definitely increasing at a high rate. So, what you have to do to overcome this problem? To overcome this problem the solution is you will insert buffers, you will insert buffers in between the pass transistors.

(Refer Slide Time: 35:15)



$$t_p = 0.69\left[\frac{n}{K} C_L R_{pass} \frac{K(k+1)}{2}\right] + \left(\frac{n}{k} - 1\right) t_{buf}$$

$$= 0.69\left[C_L R \frac{n(k+1)}{2}\right] + \left(\frac{n}{k} - 1\right) t_{buf}.$$

$$\frac{\delta t_p}{\delta k} = 0 \qquad K_{opt} = 1.7\sqrt{\frac{t_{buf}}{R_{pass} C_L}}$$

K_{opt} $\underline{3 \text{ or } 4}$

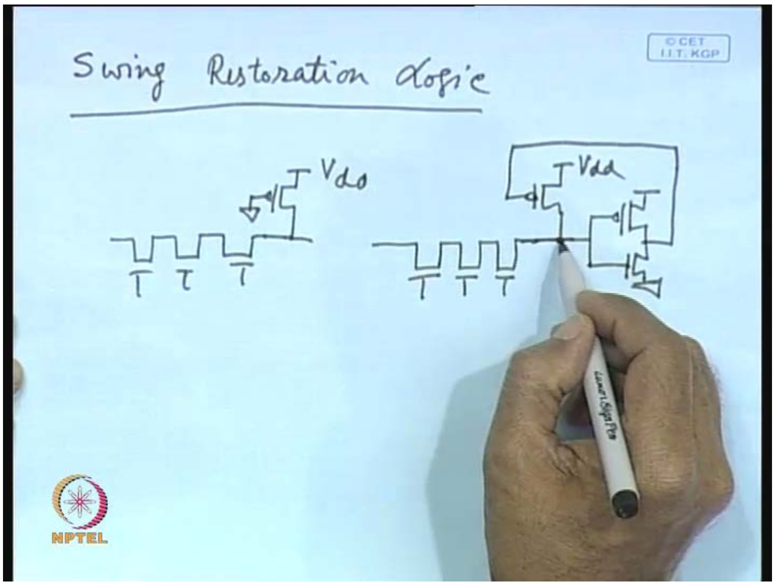
Let us assume we use buffers at regular interval, may be after few pass transistors a buffer is inserted, than again you have got few such pass transistors, here we have assume that buffer is inserted after three transistors. So, whenever we do this and if the delay of this buffers is not very large, than this the overall delay can be reduced, it has been estimated following that L bar model this delay will be equal to 0.69 n by k where you have got n pass transistors, and delay it it is introduced after k stages C L into r pass into k k into k plus 1 by 2 plus n by k minus 1 I am not deriving this, but this is the delay of the buffer, this is the expression that you will get, if you solve it you will get 0.69 this will be equal to C L r pass, so this k this k will cancel out into n into k plus 1 by 2, and this part will remain same n minus k minus 1 into t buffer.

How do you minimize this? To minimize this delay, as we know the simple way of doing it differentiate it and equal make it equal to 0. So, what you can do to minimize the value of this delay, we can have delta t p k by delta k; that means, after how many stages k is the number of stages, after how many stages you will insert a buffer? We have shown non-inverting buffer here. So, if we put non-inverting buffer, and to get the value of p k we shall differentiate it and then equate it to 0, and has been found that k optimal value that you can get will be equal to 1.7, again I have not derived in it t buffer by r pass into c l.

So, this is the optimal value of delay, I mean the optimal number of stages that will produce minimum delay, and it has been by putting realistic values of r pass C L and a delays of buffer it has been that the value of k optimal is 3 or 4; that means, if we insert buffers at the interval of 3 or 4 pass transistors than the delay will be minimized. So, this is how the impact of long delay of pass transistors can be minimized; obviously, you cannot nullified it completely there is no way of doing this, but you can minimize it by using this. So, whenever you are realizing circuits by using pass transistor logic, you will see that after 4 or 5 stages a buffer is inserted, and buffer will reduce the delay.

Now, what are the what is the second disadvantage? Second disadvantage is multi-threshold voltage drop, multi-threshold voltage drop that problem we have already discussed. Here as you see the transistor, here we are getting V dd minus V tn, now this voltage is not really good in a sense, that if we use it to drive an inverter it will not be it will lead to slow operation of the I mean, discharging; that means, if this capacitor will be discharge slower rate, because instead of 5 volt we are driving with 4 volt. So, how this problem can be overcome? Because normally, you will see you will be combing pass transistor logic with static CMOS, you may have to put inverters static CMOS inverters along with pass transistor logic, so in such a case you will you will face problem. So, this multi-threshold voltage drop particularly, if we use say two pass transistors in series, and there this one is used as an input to a next stage than problem will be more severe. How we can overcome this problem? This problem can be overcome by using what is known as swing restoration logic. swing restoration.

(Refer Slide Time: 40:21)

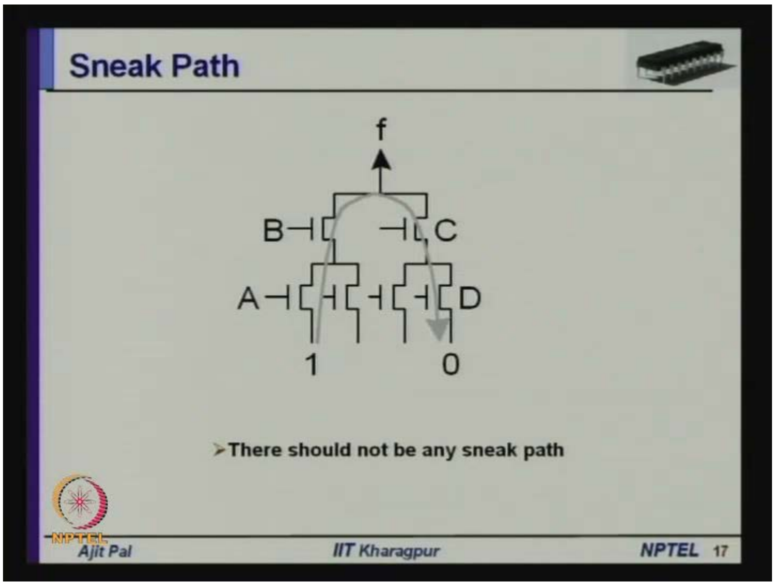


So, you have to provide a swing restoration logic after whenever you take the output to the next stage, what is the role of the swing restoration logic? It will raise the voltage level to particularly the high level to V dd. So, how it can be done? Normally, say suppose this is a pass transistor logic and this is going to an output. So, what will you do what you can do, you can put a weak p MOS transistor and ground it. So, if you do that as you know the voltage level will be pulled to V dd through this weak p MOS transistor, and this is one approach that can be followed, another approach is in addition to this weak p MOS transistor you can put an inverter along with a weak p MOS transistor; that means, this will be applied to an inverter, then you will having a weak p MOS transistor this is connected here, and this output will be coming from here.

So, what will happen in this? Even this inverter will get high level voltage, this is V dd. So, this weak p MOS will pull the output to high V dd level. So, this is kind of swing restoration logic, either by using a single weak p MOS transistor or along with I mean weak p MOS transistor along with an inverter can be used to restore the voltage level, here at this point we can see we have raise the level. So, this way this will serve as not only swing restoration logic, so swing restoration logic along with inverting buffer, so this will act as and buffer, so this can be applied to the next stage. So, you see this type of thing you have to do when realizing circuit’s, realistic circuits by using pass transistor logic, so this is the second problem that we shall face.

Third problem is sneak path, sneak path I have already mention about it, sneak path occurs when because of the problem of logic design there is path to V dd and ground. As

(Refer Slide Time: 42:58)



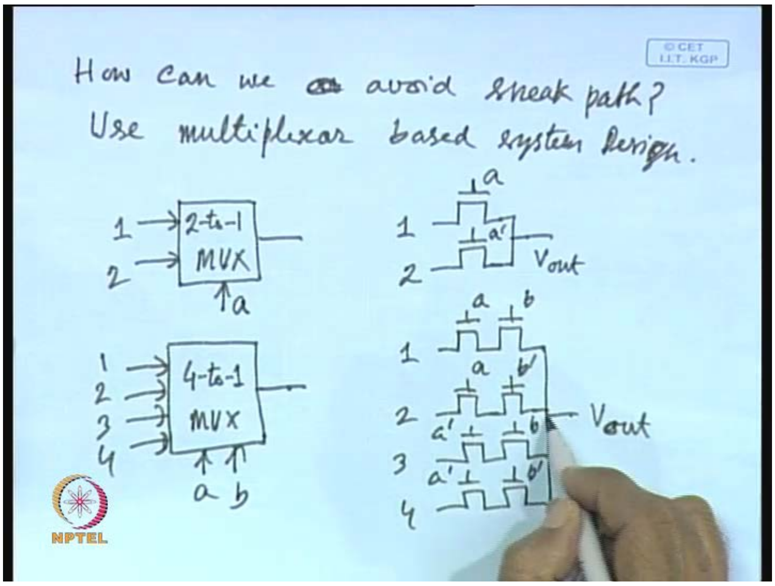
Sneak Path

➢There should not be any sneak path

Ajit Pal IIT Kharagpur NPTEL 17

you can see here, say here you are suppose to get f, but what has happens simultaneously, for some input combination in this particular case a is equal to 1, b is equal to 1, c is equal to 1, d is equal to 1 you can see there is a path from 1 to 0. So, you can see here the voltage will be neither high, nor low something in between, and not only that there will be that kind of it will let you short circuit power dissipation. So, this... So, you have to design such that there should not be any sneak path which I have already mentioned as the third rule. So, you have to design in such a way so that there is no sneak path. How can it be avoided?

(Refer Slide Time: 43:49)



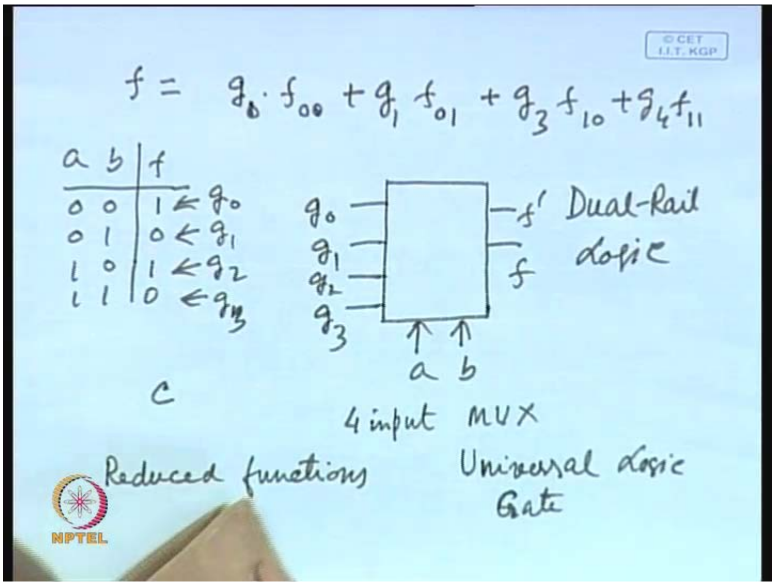
So, how can we avoid sneak path? Sneak path can be avoided by using multiplexer based circuit design. So, use multiplexer based system design. What is a multiplexer? Multiplexers as you know, it has got multiple input and a single output, you can multiplex the inputs. So, suppose you have a you are familiar with say two input multiplexor, so here there are two inputs and there is a control input and you produce a output. So, this is the control signal, and these are the two inputs that you apply, this is known as 2 to 1 MVX.

So, in a similar way you can have 4 to 1 MVX, say 4 to 1 multiplexor in this case, you will be having 4 inputs and there will be a single output and of course, you will require two signals to control which input will be selected to be connected to the output. So, essentially the role of this control signal is to control the path to one of the two inputs in this case, and here to one of the paths one of the four paths in this particular case. How can you realize this multiplexor? You can realize multiplexor by using switch logic, using pass transistors for example, this particular 2 to 1 multiplexor can be realized in this way. So, this is a 2 to 1 MVX, here is one inputs first input, here is second input, what you do here you apply a here and in this case apply the control signal a bar. So, control signal a bar is here, and control signal a is here; that means, your control signal is a this is one, this is two, and here you are getting output V out. So that means, when a is equal to 1 the one input is connected to the output, when a is equal to 0 a is equal to 0 than this input two is connected to the output.

This is how you can realize a Boolean function I mean, you can multiplexor is realized. In a similar way a 4 to 1 multiplexor can be realize in this manner, here you will require two pass transistors in each path, and there will be 4 paths there will be four paths. So, all the 4 paths are tied together, and here it produces V out, and 1, 2, 1, 2, 3, 4, and let us assume these are A and B. So, what you will do, this will be A B, and this is A B bar A bar B, and A bar B bar. So, what will happen when both the inputs as 0 than, I mean both the inputs are 1, path 1 will be connected, when a is 1 b is 0 than this path will be connected, path 2 will be connected when a is 0 b is 1, than path 3 will be connected to the output, when both are 0 than path 4 will be connected to the output.

Question is, how this multiplexor can be used to realize Boolean functions?

(Refer Slide Time: 48:02)



$f = g_0 \cdot f_{00} + g_1 f_{01} + g_3 f_{10} + g_4 f_{11}$

a	b	f
0	0	1 ← g_0
0	1	0 ← g_1
1	0	1 ← g_2
1	1	0 ← g_4

It can be very easily done from the truth table for example, say a function f can be, let us assume it is a function of 2 variables, can be always expressed as in terms of a say g 1 g 1 f say 0 0, g 0 f 0 0 plus g 1 f 0 1 plus g 3 f 1 0, and plus g 4 f 1 1; that means, suppose A and B these are the two inputs and 0 0 0 1 1 0 1 1, than depending on whatever is the say this is f, say let us assume we are realizing 1 0 1 0, than g 0 g 1 g 3 g 4 can be directly taken from the truth table; that means, f 0 g 0 value of g 0 is equal to 1, value of g 1 is equal to is equal to 0, value of g 2 is equal to 1, value of g 4 is g 3 is equal to 0; that means, what will you do if this is the multiplexor, you will apply a and b as control input, and you will apply g 0, g 1, g 2 and g 3, which are essentially Boolean function, I mean Boolean variables you can see depending on the function you are realizing they will be having 0 or one, so accordingly we apply and this will realize f.

So, it can be proved that a multiplexor, 4 input multiplexor can realize any function of two variables, not only I mean later on I shall prove also, instead of applying 0 and 1 here, if you can apply 1 variable here also, instead of say suppose there is another variable c, what you can do you can apply 0 1 or for c and c dash, if you are allowed to do that; that means, a 4 input MVX can realize any function of 3 variables, you can expand around a and b, than this function g 0, g 1, g 2, g 3, these are known as reduced functions. This reduced functions you can find out, those reduced functions will be either constant; that means, 0 or 1 or function of the third variable c; that means, either c or c das.

So, those you can apply 0 1 c and c dash, and you can realize any function; that means, using pass transistor logic you can realize bullion function using multiplexor as the kind of building block, where multiplexor is realized using pass transistors. So, this is how it can be done, and it can be proved that this is free from all the problems that I discussed; that mean, all the designed rules which I have mentioned is followed here, none of the three designs rules that I have told is violated in this case. So, you can realize functions in this way, and that is the reason why multiplexors known as, they are known as you know if they can since they can realize any function, they are considered as universal logic gate. So, since they can realize any function.

So, in fact, later on if you study FPGA field-programmable gate array, there are multiplexor based field-programmable gate array by active, there multiplexor is used as building block of digital circuits. So, how multiplexor is realized that is no concern there, but any Boolean function can be expressed in terms of multiplexors and this is how it can be done. So, we find that this is a technique by which you can realize Boolean functions, and using multiplexor as building block. Let me consider another very important disadvantages complementary control signals.

We have already seen that in some cases you have to apply control signals in the complemented form, if we consider the network within the multiplexor network we can see b bar, a bar these are also present; that means, you have to apply both the variable, control variable, and it is compliment to control the pass transistor logic network; that means, this is a disadvantage, you have to it must you have to generate the control signal. In other words, you have to use what is known as dual-rail logic dual-rail logic. So, you will generate f as well as f bar, so that f and f bar can use as control signal to the next stage. So, these are the main drawbacks and advantages that we have discussed, now you can summarize what we have discussed in today’s lecture. I have discussed about introduce the basic concept of pass transistor logic, discussed about the logic design rules, discussed about their advantages and disadvantages, and discussed techniques how to overcome these problems

In my next lecture, I shall discuss about how you can realize complicated Boolean functions using pass transistor logic; that means, systematic technique for logic design, and also there are some logic families, pass transistor logic families which have been proposed in the last couple of years. So, I shall introduce the pass transistor logic family to you. Thank you.