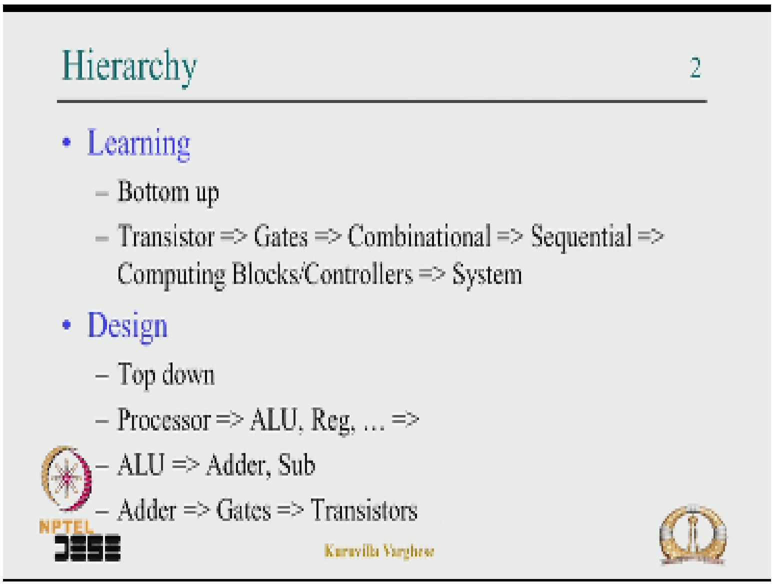
Digital Systems Design with PLDs and FPGAs
Kuruvilla Varghese
Department of Electronic Systems Engineering
Indian Institute of Science – Bangalore

Lecture-02
Revision of Prerequisite

So welcome to the second lecture of digital system design with PLDS and FPGAs. In the last lecture I talked about the contents of the course, the course objective, what are the competencies?, I hope you will develop at the end of the course reference book and some extend in the mode of giving the exercise during the lectures, all that was covered in the course objectives and contents.

Then I started an overview of the field or revision and overview of the field. So quickly will recapture the overview we have d1 in the last lecture, then continue with the revision and overview.

(Refer Slide Time: 01:20)



Hierarchy 2

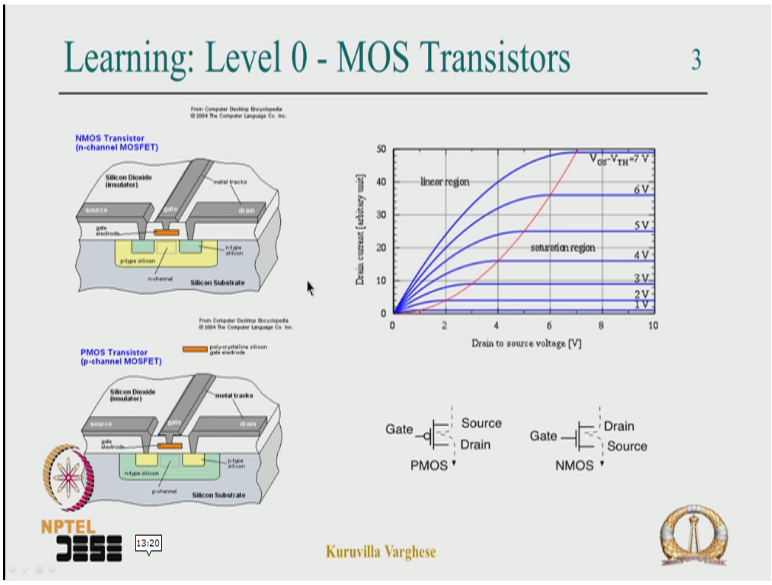
- Learning
  - Bottom up
  - Transistor => Gates => Combinational => Sequential => Computing Blocks/Controllers => System
- Design
  - Top down
  - Processor => ALU, Reg, … =>
  - ALU => Adder, Sub
  - Adder => Gates => Transistors

Kuruvilla Varghese

So let us move on I said in learning you learn always a bottom-up that means from the smart building block to the complete system you go hierarchically that like a transistor gate, combinational sequential and the system. But while designing you go top down like you take a processor break into pieces take each piece and make it in the further pieces and go all the way to the gates and transistor.

So when you design you should remember that this is a hierarchy we use, but while learning the opposite is that the thing.

(Refer Slide Time: 01:54)



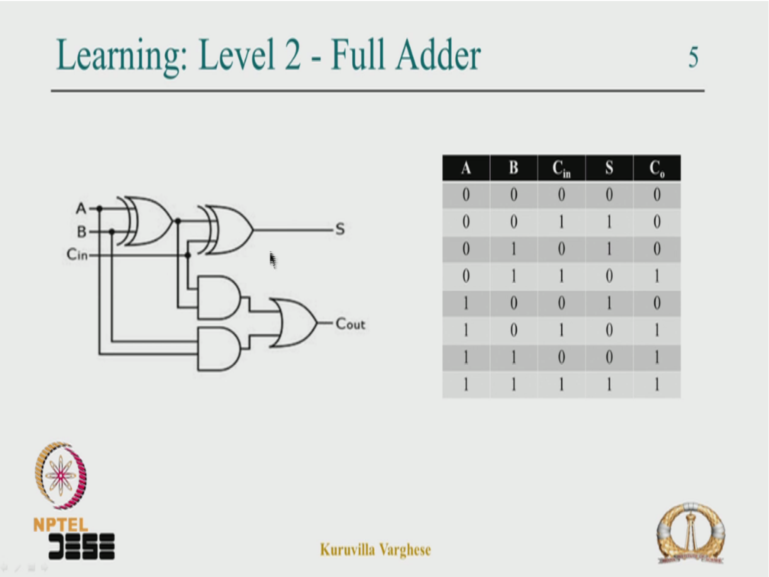
Learning: Level 0 - MOS Transistors 3

From Computer Desktop Encyclopedia
© 2004 The Computer Language Co. Inc.

Kuruvilla Varghese

We have g1 through some example like you start with transistor, construction, characteristic, symbol, then you make gates out of the transistor various gates its input output characteristics.

(Refer Slide Time: 02:11)



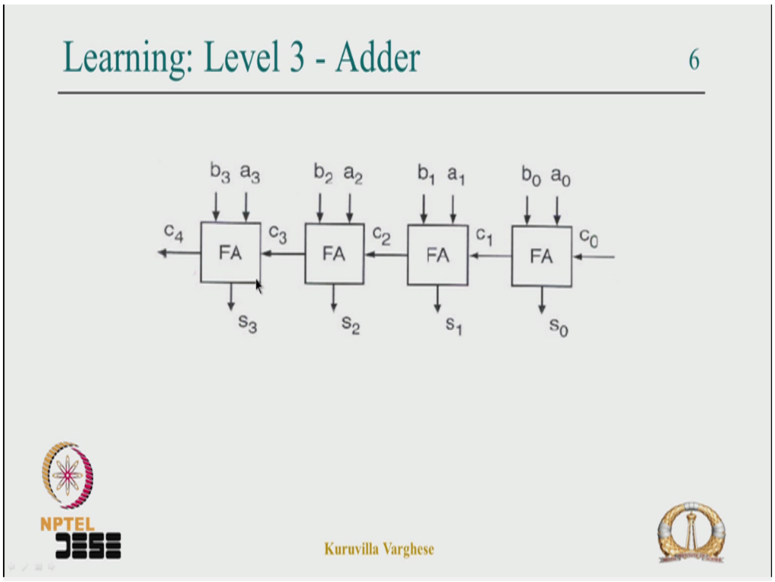
Learning: Level 2 - Full Adder 5

A	B	C_{in}	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Kuruvilla Varghese

Then you built further useful circuits like full adder, using XOR gate, AND gate and OR gate by writing its truth table minimising it and implementing it and so on. At the next level we use this as a building block to build even bigger you know multi bit adder.

(Refer Slide Time: 02:32)

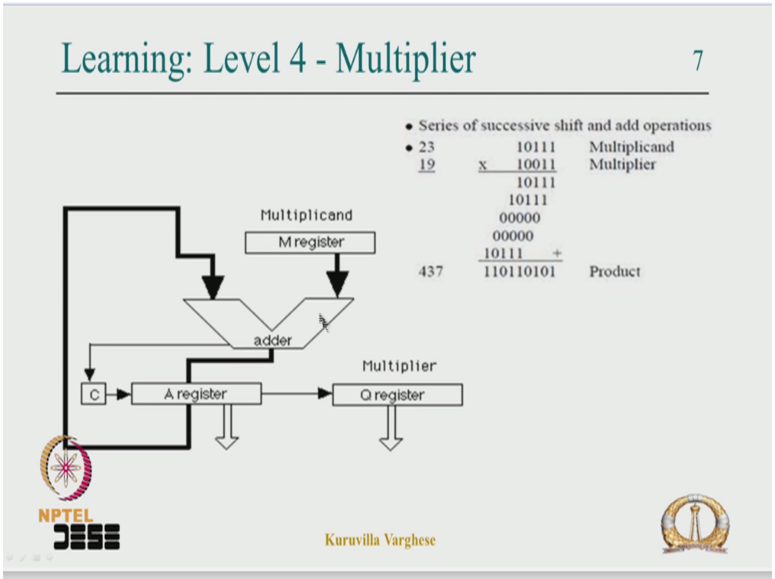


Learning: Level 3 - Adder

6

Kuruvilla Varghese

(Refer Slide Time: 02:42)



Learning: Level 4 - Multiplier

7

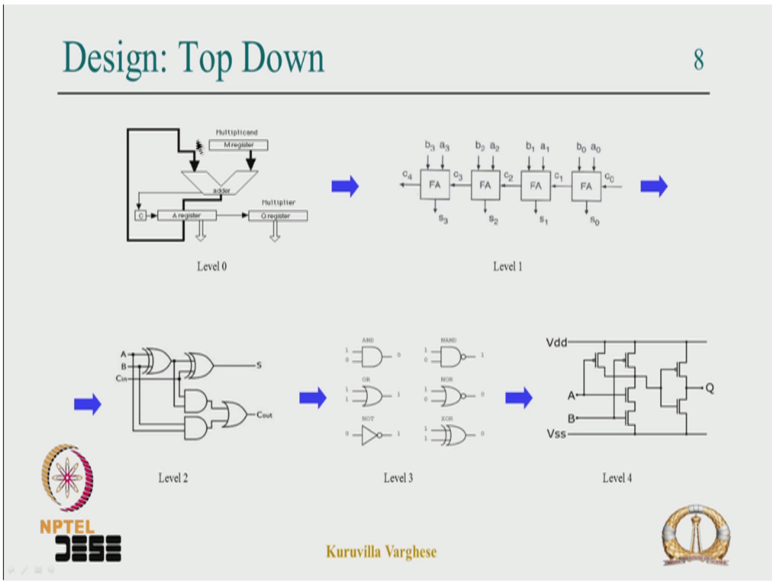
- Series of successive shift and add operations
- 23 19

```
23          10111     Multiplicand
19      x   10011     Multiplier
            10111
           10111
          00000
         00000
        10111    +
437     110110101     Product
```

Kuruvilla Varghese

In this case the 4 bit repel adder made a 4 full adders then having known this then we can go to the next level of building with these building blocks the combinational and sequential building blocks a multiplayer with multiplicand accumulator multiplayer with adder and so on.

(Refer Slide Time: 02:52)



Design: Top Down

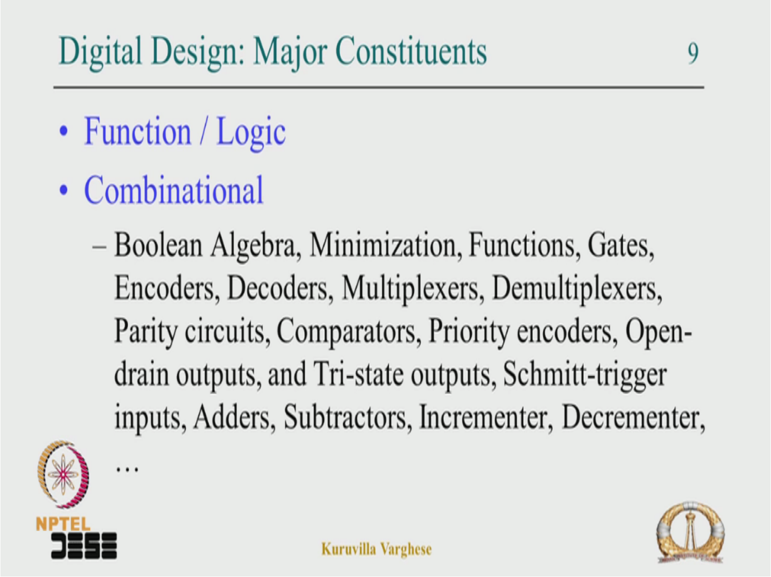
8

Kuruvilla Varghese

But in the case of design we do the opposite having multiplication algorithm we design an architecture with higher level sequential and combinational comp1nts, then each block is design separately. In this case I am illustrating 1 of the block design, so you take the adder then that is shown as a 4 bit ripple adder, in that it is made up 4 full adders and full adders taken that is designed.

And then this is taken the gates design in terms of transistors and so on like when you design a chip even this you have to go further you know you have to make the transfer layout in the chip and mask and so on, but our interest in this course is up to here when we come to this point the front end design is over that we convert the spec or the algorithm into a gate and flip flop level circuit. So that is the front end design that the focus of the course.

(Refer Slide Time: 03:55)



Digital Design: Major Constituents

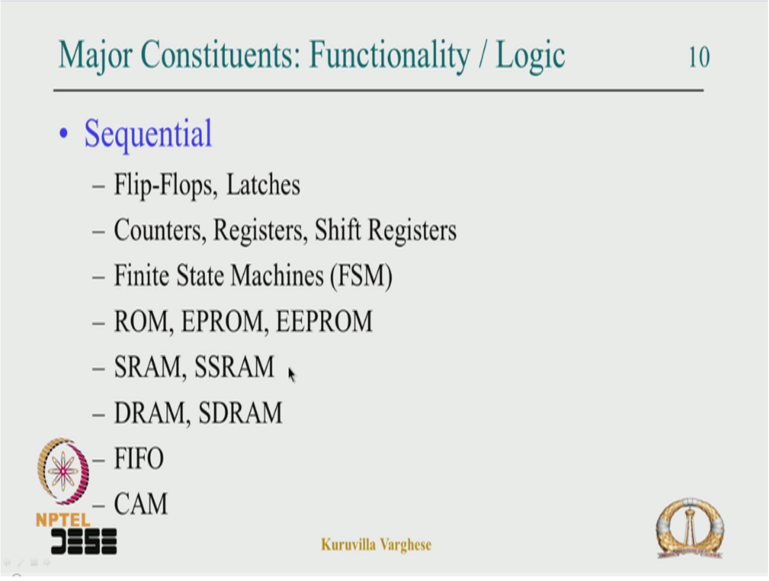
9

- Function / Logic
- Combinational
  - Boolean Algebra, Minimization, Functions, Gates, Encoders, Decoders, Multiplexers, Demultiplexers, Parity circuits, Comparators, Priority encoders, Open-drain outputs, and Tri-state outputs, Schmitt-trigger inputs, Adders, Subtractors, Incrementer, Decrementer, …

Kuruvilla Varghese

So let us move forward and I said when you design the major constituent of the design the first for most important thing is the function and you have learnt all the building blocks in your undergraduate curriculum, there is a combination circuit all these blocks like you know encoder, multiplexer, arithmetic circuit and so on.

(Refer Slide Time: 04:24)



Major Constituents: Functionality / Logic 10

- Sequential
  - Flip-Flops, Latches
  - Counters, Registers, Shift Registers
  - Finite State Machines (FSM)
  - ROM, EPROM, EEPROM
  - SRAM, SSRAM
  - DRAM, SDRAM
  - FIFO
  - CAM

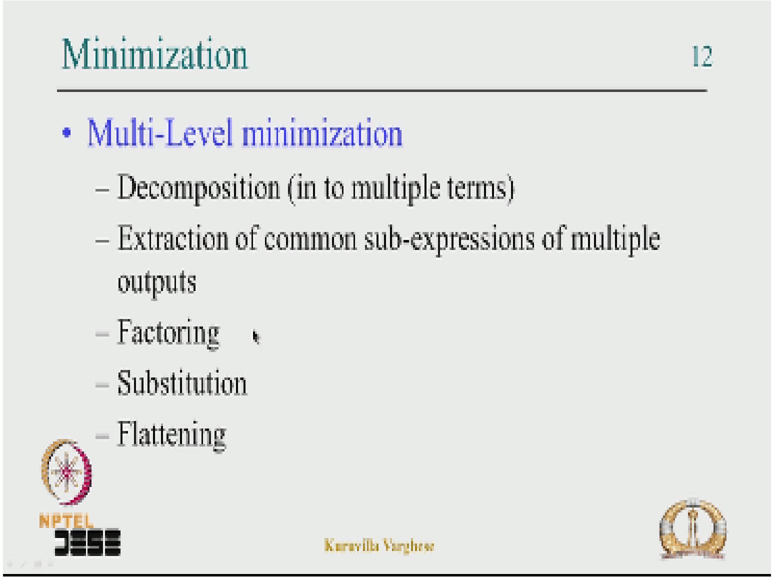
Kuruvilla Varghese

Then comes to flip flop, so you have flip flops, registers, memory. All these need to be thoroughly learn what are the architecture of this?, what is the function of that, what is input output characteristics? and so on. This should be thorough so that we can build higher level system using these blocks.

And I talked briefly about minimization, so you would have learn most of the time this graphical tool called Karnaugh maps, the computer algorithm equal and that is a point McCluskey which is very complex because it starts with the midterm, so there are a fast methods like Espresso, so which is used which does not necessarily reduce the minimal solution, but very near minimal solution.

But the computation time is very very negligible or compare to the Quine McCluskey so this is what is used in real life synthesis tools used this minimization algorithm.

(Refer Slide Time: 05:33)



Minimization

12

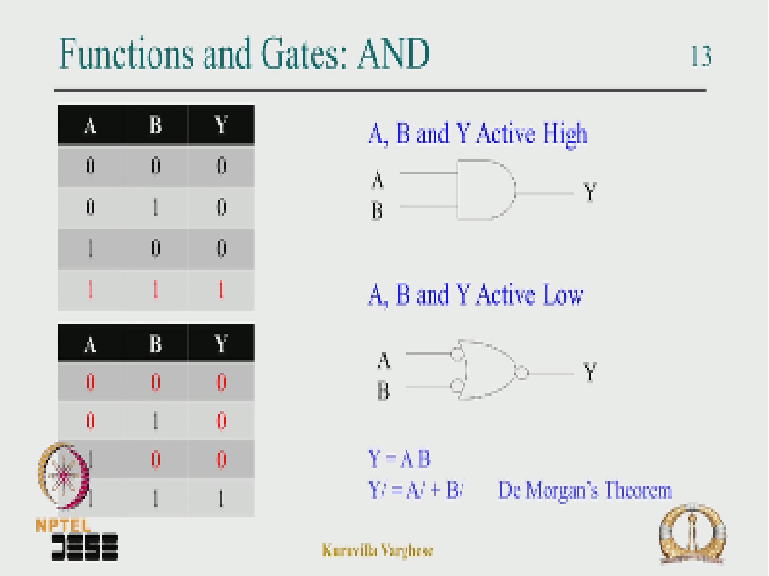
- Multi-Level minimization
  - Decomposition (in to multiple terms)
  - Extraction of common sub-expressions of multiple outputs
  - Factoring
  - Substitution
  - Flattening

Kuruvilla Varghese

Moreover mostly used to get to level implementation. In again in real life circuits we will have multilevel multiple stages of combination circuits and there could be multiple outputs, so when you minimise the multiple output scenario you have to find the common sub-expression among the outputs to be able to share the resources. So that is 1 step in multilevel multi output minimization.

So these are the kind of steps which helps us in multilevel multi output minimization like factoring, substitution, flattening, but as I said this is not our focus those are interested can look at the latest minimization algorithm and digital synthesis.

(Refer Slide Time: 06:22)



Functions and Gates: AND

13

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A, B and Y Active High

A, B and Y Active Low

$Y = A B$

$Y' = A' + B'$ De Morgan's Theorem

Kuruvilla Varghese

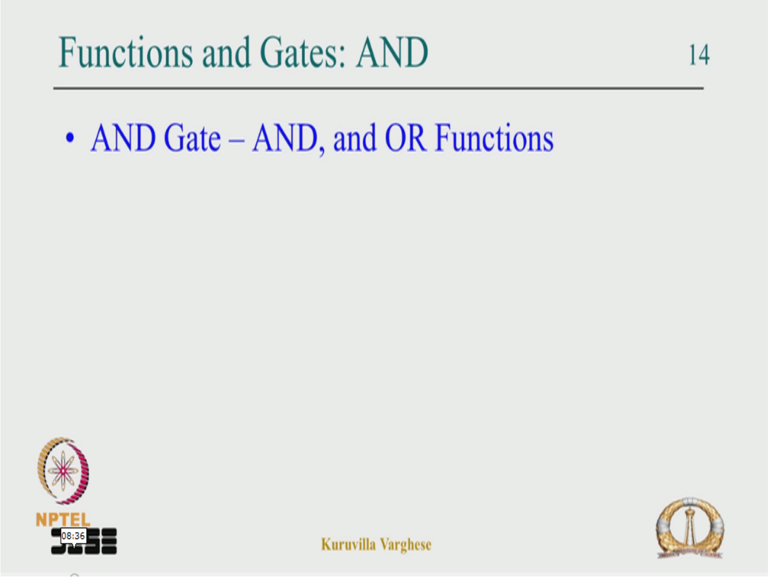
So let us move on and this is where we have kind of stop last time so I would like you to make a little differentiation between the gates and the function. So you take an AND gate depending upon the active levels of the signal it can do other functions. So take this truth table and assume that a b that the input and the output are active high. So you look at it that means that when both inputs are active, output is active.

Otherwise output is not active, so it means our if you want to the conventional terminology both inputs are true then the output is true, so the AND gate when all the signals are active high implements and AND function, but look at this same truth table and assume that we have treating A, B, Y as a active low, that means you look at the truth table when any1 of the input is active, the output is active that means when we treat the inputs and outputs are active low the AND gate implements and OR function.

Sao that is what shown here, the same AND gate, but the function is OR and the bubble indicates active level, so in a sense AND gate can implement AND function or OR function and for a clever student we can easily make out this is exactly bringing a de-Morgans theorem into our concept. So here it is Ys A and B, so if you take a Y bar so i use slash for the bar because it is easy to type and text.

Y bar is nothing but AB bar which is nothing but A bar or B bar so essentially what we are doing is bringing this De Morgan theorem into our conceptual level. So that when you work out simple circuit it is very easy that you do not waste gates. So let us take another example let us go and look at the NAND gate.

(Refer Slide Time: 08:34)



Functions and Gates: AND

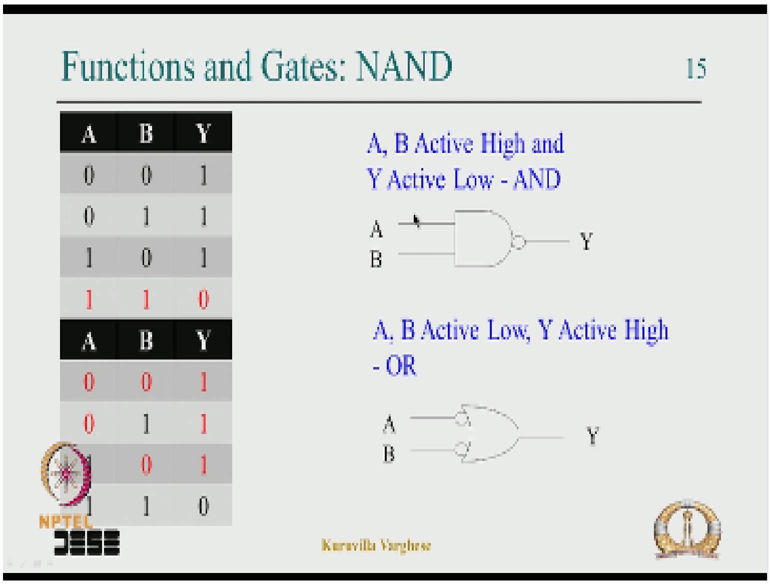
14

- AND Gate – AND, and OR Functions

Kuruvilla Varghese

So AND gate can do AN and OR function depending on the active level of the input and output.

(Refer Slide Time: 08:40)



Functions and Gates: NAND

15

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A, B Active High and Y Active Low - AND

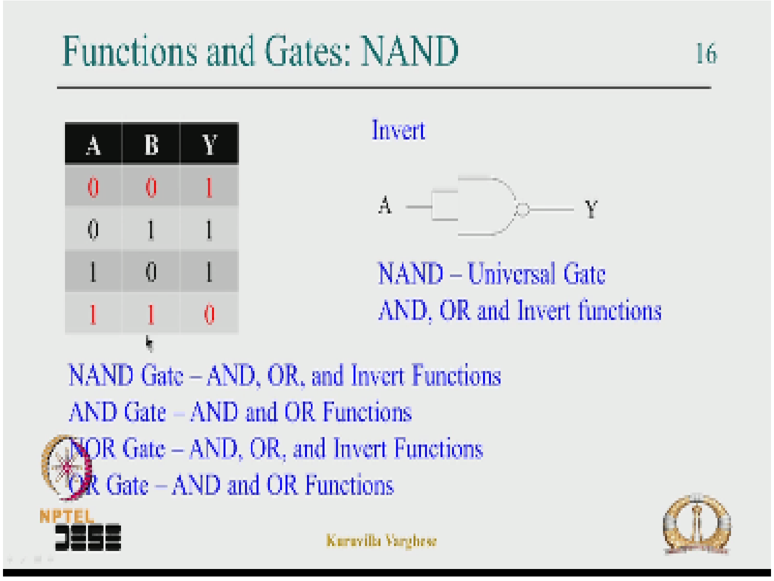
A, B Active Low, Y Active High - OR

Kuruvilla Varghese

So let us take the NAND Gate and you look at that we will treat the inputs are active high and output as active low. Then you see when both inputs are active, the output is active or when both inputs approved output is true that means it is implementing and AND function and look at opposite like when we treat the inputs are active low A and B as active low and Y as active.

Then you look if any 1 of the input is active output is active. Otherwise it is not active. So this implement then OR function, so that is shown as that OR symbol with the bubbles for the active low input. So when you see this symbol it is essentially doing an OR function of the active low A and B but essentially the gate involved is a NAND gate that is idea. NAND gate 1 more property.

(Refer Slide Time: 09:44)



Functions and Gates: NAND

16

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Invert

NAND – Universal Gate
AND, OR and Invert functions

NAND Gate – AND, OR, and Invert Functions
AND Gate – AND and OR Functions
NOR Gate – AND, OR, and Invert Functions
OR Gate – AND and OR Functions

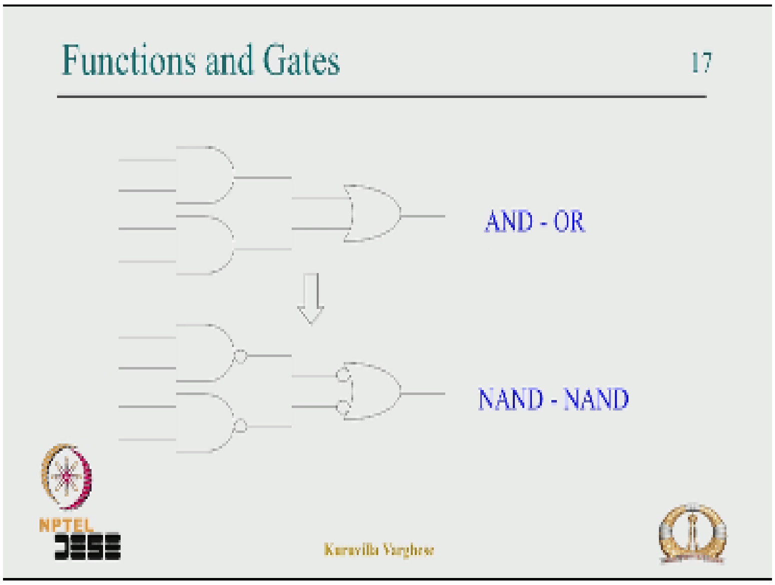
NPTEL

Kuruvilla Varghese

So that let us move to the next slide, so look at this when both inputs are active low output is active high or when both the inputs are false output is true, when both output and inputs of 2 the output is false it shows that if we tie it together, so you tie A and B together like this and you have a symbol in then if input is 0, the output is 1, input is 1 output is 0. So this NAND gate can act as an invert implement invert function or act as a invertor.

So NAND is called universal gate, it can implement AND function OR function and invert function. So in conclusion if you take NAND gate and NOR gate these are universal gates depending on the active levels of the signal, it can implement AND, OR, or invert, but both AND gate and OR gate implement both AND function and OR function. So this is making the difference between the gate and function. And also we are essentially we are bringing in the Demorgan theorem into the concept. So let us move on.

(Refer Slide Time: 11:03)



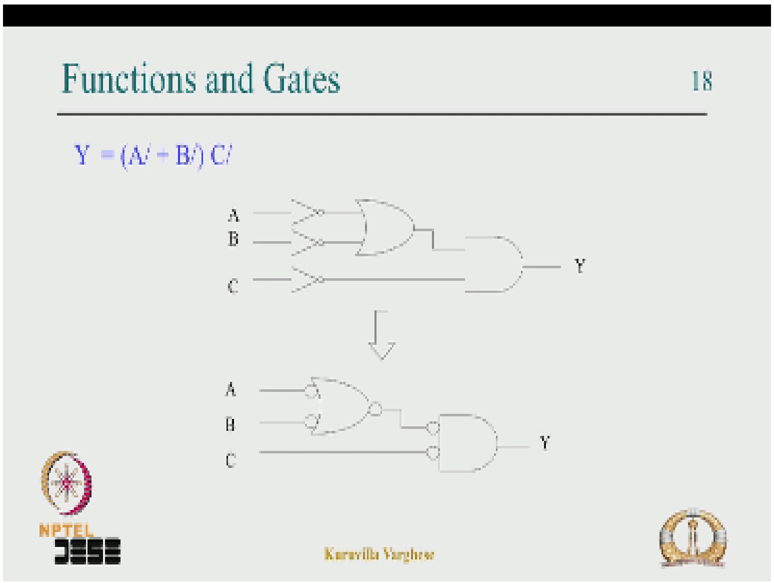
Functions and Gates 17

Kuruvilla Varghese

So an example of this which you often do is that you have AND or implementation a 2 AND gates feeding an OR gate, so let us put a bubble here that means invert it and to compensate for this immersion or bubbling, will put another bubble at the input here. Similarly output a bubble here and bubble here, so that it is equivalent. These 2 circuits equivalent, but the moment you put that you know that this is the NAND gate which is doing AND function and this is also an NAND gate which is doing an OR function ok.

S you essentially convert an AND or OR and NAND and NAND by applying this concept you do not have to do Demogran theorem very simple concept way you can convert this AND-OR into NAND-NAND kind of representation that is why the duty of differentiating the function and gates.

(Refer Slide Time: 12:03)



Functions and Gates 18

Y = (A/ + B/) C/

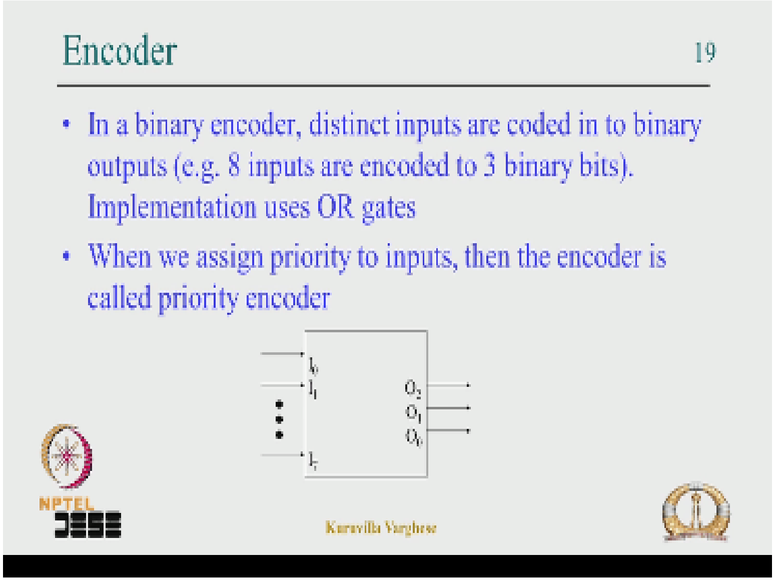
Kuruvilla Varghese

So let us move so again an application is shown here suppose you have an Boolean equation like A bar or B bar and C bar then if you take it in a very way without kind of trying to minimise it then you will put 3 invertors A bar, B bar, C bar to the NOR gate and then everything will be this output and input as handed to get, so end up with 5 gates but if you apply this the method we have discuss then we put these are come as bubbles.

So here is a bubble and we have 1 bubble here, so it compensate like here for this invertor to make a symmetric we put a bubble here and bubble here, so immediately you know that this is nothing but a AND gate and this is nothing but a NOR gate, so you end up 2 gates instead of 5 gates, so though we are not going to do any kind of gate level implementation, but for small circuit very quickly can work out.

Sometime it is useful you are very look logic very simple logic you can quickly write VHDL code by converting like this not that who can handle it then sometime in some places you can work out the concept wise what you learn in mean forces that is why discuss this and let us look at 1 or 2 other combination comp1nts.

(Refer Slide Time: 13:38)



Encoder 19

- In a binary encoder, distinct inputs are coded in to binary outputs (e.g. 8 inputs are encoded to 3 binary bits). Implementation uses OR gates
- When we assign priority to inputs, then the encoder is called priority encoder

Kuruvilla Varghese

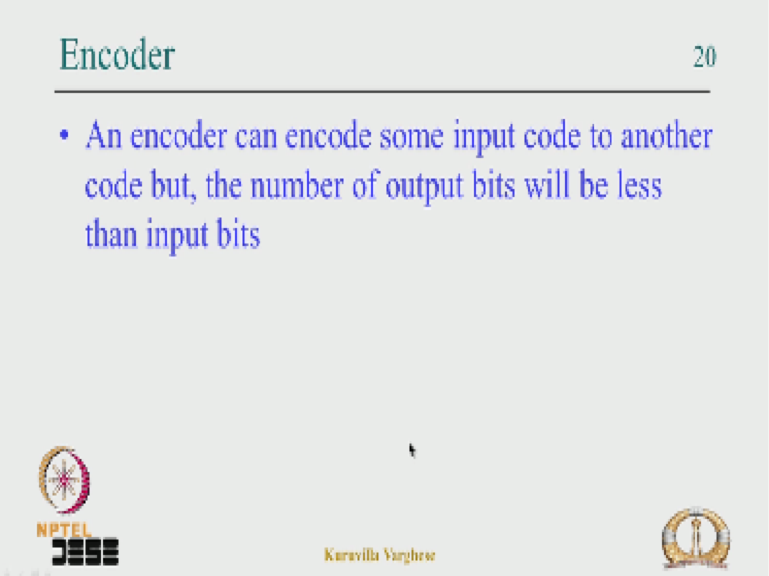
Take an encoder you would have studied an encoder let us take the example of a binary encoder, so here in the binary encoder in this encoder you have 8 distinct input at a time any 1 of the input can be active, active high, so we code the output of a equal and binary number that means if this is active this will be 000, if I1 is active 001, I7 is active 111 like that, so it is an 8 input 3 output 3 binary output encoder and you can imagine that internal circuit is nothing but OR gate.

Suppose I7 is the 1 then there be 3 OR gates and I7 will go to the input OR gate so that you get 11, so encoder essentially uses OR gate and your priority suppose there is a situation that more than 1 of this can be activate any time then we have 2 kind of give priority to 1 of them because we assumed that only 1 will be activate at any time. In that case there will be AND gates of you know coming here the top.

So that when the I0 is active which mask the I1 out, so there be invert of I0 will be going to mask it the I1 and when comes to I2 invertor of I0 and I1 will be going and masking the Iq and so on, so the last gate will have huge AND gate with which mask with all the complement of all these input you have studied priority encoders, so this is what is encoder about.

And 1 topic of encoder is that the number of bits in input will be less than the number of bits in input, so this is what is encoder and basically is nothing but the OR gate equal to the number of output.

(Refer Slide Time: 15:41)



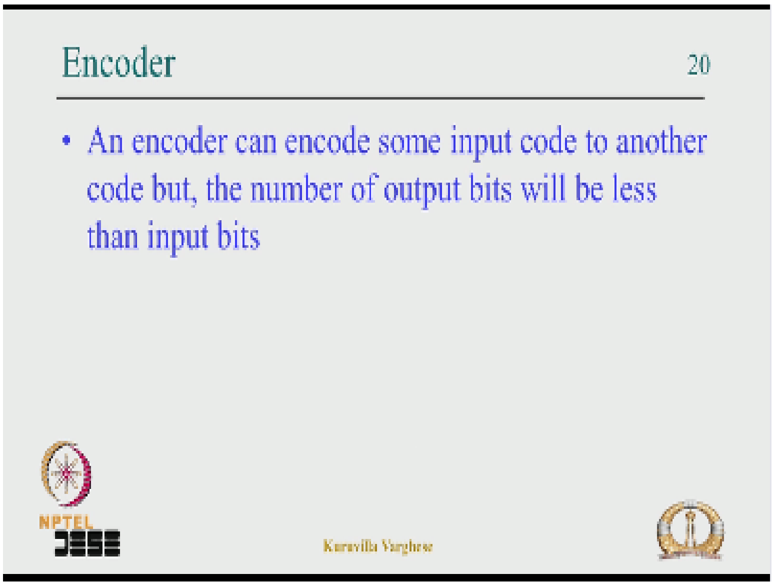
Encoder 20

- An encoder can encode some input code to another code but, the number of output bits will be less than input bits

Kuruvilla Varghese

In case of simple encoder let us look at the Decoder.

(Refer Slide Time: 15:45)



Encoder

20

- An encoder can encode some input code to another code but, the number of output bits will be less than input bits

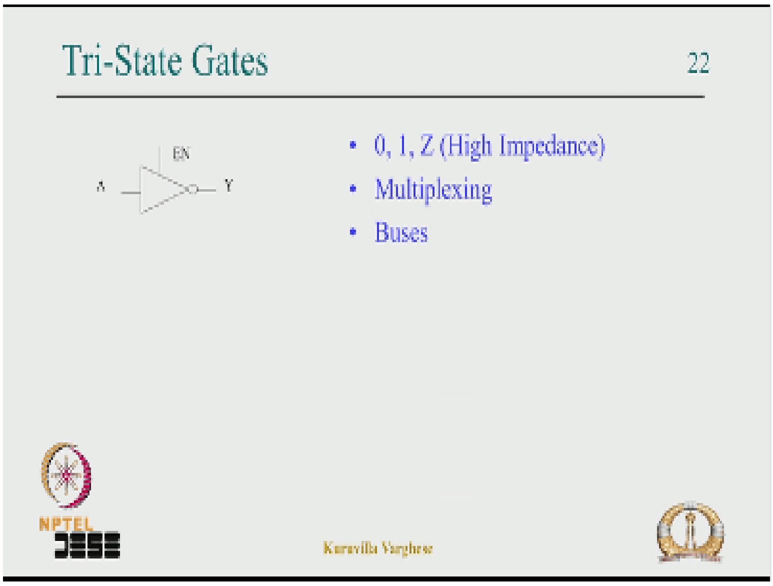
Kuruvilla Varghese

I am showing an example of a binary decoder so here we have 3 binary inputs that means it will go from 000 to 1111, so that means 0-7 and we have distinct 8 outputs, supposed the number here is 101 then O5 will be active, so if the number is 5 fifth output will be active ok. So this basically uses AND gate that means we will put an AND gate suppose number is 101 here put an AND here this 1 goes directly 1 goes directly.

This go through invertor or bubble to the AND gate, so for 5 only this AND gate is active and as goes to O5, so you can image that there will be 8 AND gates with 3 input each with the decoding all the mean term so that is a decoder. In a general decoder the number of bits in the input will be less than the number of bits in the output. So normal in digital design and high level design will be using.

There will be need to use the decoder, may not have much encoder but we will be definitely using more decoder and encoder but depending on the need that could be encoders also, so let us move on.

(Refer Slide Time: 17:13)


Tri-State Gates 22

- 0, 1, Z (High Impedance)
- Multiplexing
- Buses

Kuruvilla Varghese

This is 1 case the tri-state gate which cause often quite a lot confusion, so this shows a Tri-state invertor so where it is like normal invertor when this enable is high, so if enable is high then A is 1, y is 0 A is 0 Y is 1, but when enable is 0 then we say Y is Tri-state and most people do not know what is or we say high impedance and it is indicated by the symbol Z so to understand what is high impedance then if I ask a student what is high impedance many people are not clear so let us look at the impedance of this tri state invertor.

So it is like a usual normal invertor PMOS on top and NMOS on at the bottom but in between you can 1 more NMOS and 1 more PMOS it is in the views, when enable is active this transistor and this transfer will be on, so that when A is 1 this path is active pulling the Y to 0 when A is 0 this part is active pulling the y to the doubling making it 1 ok. So what happens if this enable is off in an inverter case either this path is active or this path is active.

But when enable is active you can see that this is cut off and this is cut off so this particular point this output has a very high impedance part to the (()) (18:54) that means you can say if the rest ends are equal maybe as a 10 make to here 10 make to here and if the supply is a 5 volts then this will be floating at 2.5 volts any coupling any small signal coupling can pull it to the 1 or 0.

That is why this is called high impedance and this is use for multiplexing and busing and all that let us see the scenario here, so here I show say multiple Tri-state invertors are tie together to form a bus. The idea is that only at any time only 1 if the output drives this output so all others will be cut off and if both are 2 anymore than any 1 of them is on then there will be a clash here.

That means it can be 1 and this is 0 then there is a problem, so 1 is active like that and the 0 so there will be short circuit from the VVD to the ground, so that should be avoided but there is a problem if all are cut off then this input will be this output will be kind of in between and if this bus is connected to some input then this will be floating and any small noise can make it switch go up and down and there will be lot of power wastage in this, may be in the circuit design this is taken care.

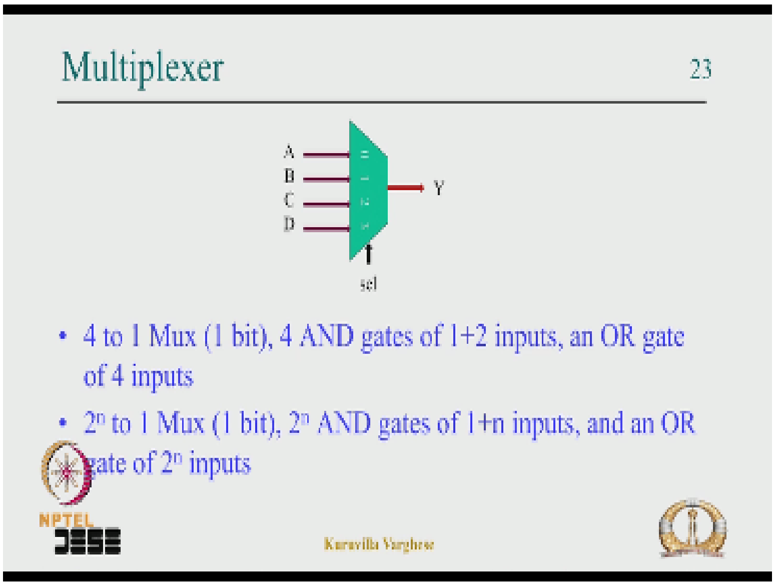
This is ignored when everything is cut out, but still there will be lot of participation due to switching so when you Tri-state a bus normally either you pull it up or pull it down to a high resistant so that when everything is cut off it is safely kind of 1 or 0 a weak 1 or 0 if there, so either you pull it up to VVD through a high resistant or pull it down to the ground using a high resistant.

So that is about the Tri-state gate, so please remember this Tri-state gate as 3 state at the output 1 is 0 then the next 1 is 1 and the high impedance, high impedance means the output is flopping, and if you are driving any input it should be pulled up or pulled down and it is a very bad strategy to keep any output at high impedance to mean inactive stage, many times student write VHDL code.

If enable is if some condition is not bad the output is Tri-state, it is a dangerous thing to do if you are not sure what you are doing, that means that output will be in between some state and that is driving from input and it can produce incorrect logic levels at the output of those circuits which it is driving. So you have to treat Tri-state gate.

Tri-state gate should be used only when you are multiplexing or you are bussing otherwise it is not be used it is not an inactive state like 0 or something like that. So just like that if you not put Z in a encoding when you are coding that in circuit. So let us move on to the next comp1nt circuit.

(Refer Slide Time: 22:33)



Multiplexer 23

- 4 to 1 Mux (1 bit), 4 AND gates of 1+2 inputs, an OR gate of 4 inputs
- 2^n to 1 Mux (1 bit), 2^n AND gates of 1+n inputs, and an OR gate of 2^n inputs

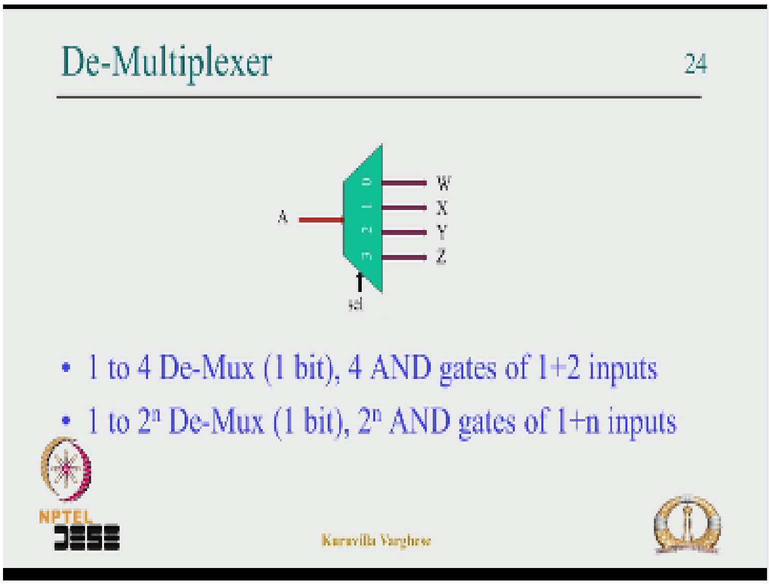
Kuruvilla Varghese

You would have known that is multiplexer so the multiplex are you know that multiplex is the input. So now in this case I am showing a 4 to 1 multiplexer, so 4 to 1 multiplexer will have 2 bit select line, when the select line is 0 A goes to Y when select line is 1B then 2C and 3 is D. This can be like 1 bit or multiple bits does not matter and you know that internal it is nothing but AN OR gates and the select line will enable particular AND gate.

So in the case of 4 to 1 mux I have written here for 1 bit case, there are 4 AND Gate corresponding to 4 inputs and each AND gate has this particular 1 bit input and the select line going to enable this the min term decoding of the select line like here the select 1 and select 0 will go with a bubble, here select 1 and select 0 will go directly, so that when in the 11 this path is selected.

And there is an OR gate here which combines output of all the 4 AND gates, so in general if you have 1 to raise to N to 1 multiplexer then you have at the input 2 raise to N AND gate with what each AND gate is a 1+N inputs that is a log logarithm of the number of inputs ok to the base 2 1+N input and an OR gate of 2 raise to an input to combine all the 2 raise to N output of the AND gate. So that is a multiplexer.

(Refer Slide Time: 24:14)



De-Multiplexer

24

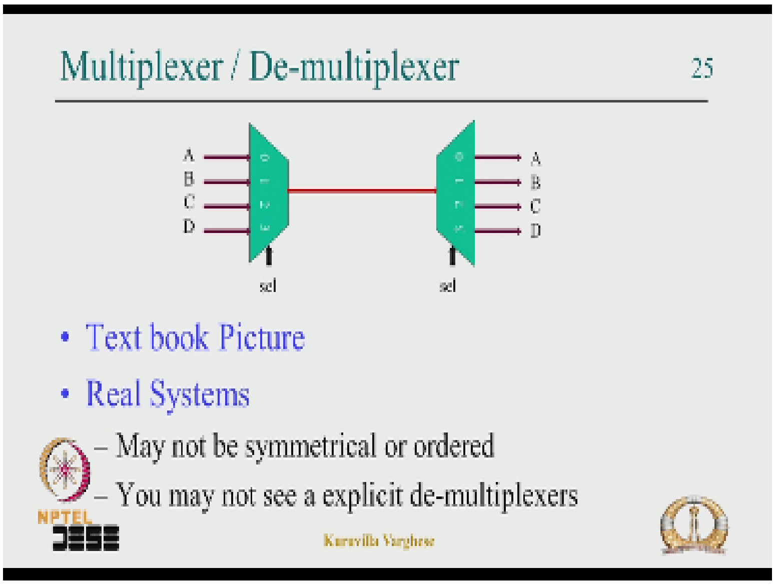
- 1 to 4 De-Mux (1 bit), 4 AND gates of 1+2 inputs
- 1 to 2^n De-Mux (1 bit), 2^n AND gates of 1+n inputs

Kuruvilla Varghese

So let us move on to the de-multiplexer, it is opposite of de-multiplexer, so you have 1 input depending on the select line it puts it to any 1 of the 4 output. So if the select line is 0 it goes here, if it is to it goes here and so on and basically it is an AND gate you know that there is an AND gate. In this case there are 4 AND gates and output and this A goes to all the AND gates of 1 bit in the 1 bit case.

And the select line goes to all the AND gates with the proper decoding like here select 1 bar and select the 0 bar here, select 1 and select 0. So for a 1 to 4 demarks there will be 4 AND gates of 1+2 input for 1 to 2 raise to N 1 demarks you have 2 raise to N AND gates with each AND gate has 1+N input, so that be kept in mind, so multiplexer is AND OR gates and de-multiplexer is nothing but the AND gates.

(Refer Slide Time: 25:14)



Multiplexer / De-multiplexer

25

- Text book Picture
- Real Systems
  - May not be symmetrical or ordered
  - You may not see a explicit de-multiplexers

Kuruvilla Varghese

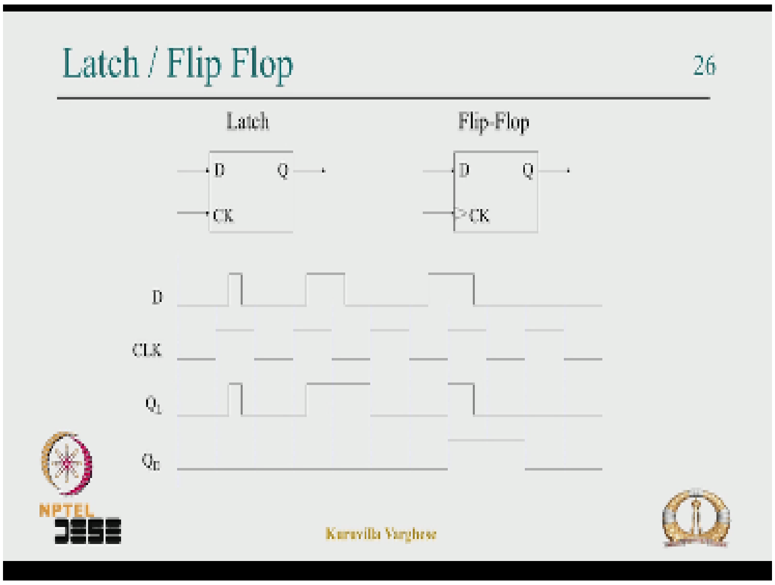
So and this is 1 picture that you see in a textbook normally the application of the multiplexer and de-multiplexer is shown like that, there is a 4 to 1 multiplexer with multiplexers with the select line and at other end it is de-multiplexer with the same kind of select line okay, so it is a textbook conceptual picture and many times some kind of a rotary kind of arrangement is shown.

But in real life in real system this may not be very symmetry that means symmetry I mean that at 1 end it may be 4 to 1 multiplexer at the other end it may be 1 to 3 de-multiplexer second thing is that it may not be ordered like it is not that A goes to A need not be like that may be A and B will go to B here and C go to A and so on. It need not be very much ordered.

And the best thing you can go and see some serious digital logic design you will not see any symbol of a de-multiplexer ok, you want see an explicit demultiplexer in the secure ok, so you can think over at please think over why that happens ok, you do now know that the demultiplexing is not happening it is happening but you would not see a combinational circuits which is acting as a demultiplexer.

So you can moreover it can come out with the answer maybe we can communicate with me or find the answer yourself, so that is about multiplexer and demultiplexer. Let us move on.

(Refer Slide Time: 26:57)



Latch / Flip Flop

26

Kuruvilla Varghese

So that is the kind of quick overview of some of the comp1nts of the combination circuit, so I have told you about the gates and function. So as I said an AND gate can do AND or OR function. NAND gate can do AND, OR or invert function, similarly NOR gates. So they are universal gate, and essentially we are building the de-morgan's theorem into our concept and second I have told you is about that Tri-state gate**.**

What is the Tri-state high impedance and the care need be taken while using the Tri-state gate of what are the application of Tri-state gate and the next thing I talked about the encoder and Decoder. Encoder what is a structure of encoder, what is the structure of Decoder then we had look at the multiplexer and demultiplexer. What is multiplexer and what is demultiplexer and the really life circuit is not like textbook picture.

You may not see a demultiplexer symbol and as I said you please think over and come out with an answer why there is may not be a combination demultiplexer you know expressively the symbol is not there in the real circuit, so let us look at the flip flop and latches. So this is important, so this shows a latch, a transparent latch, it is input D and output Q and there is a clock.

And the operation of that the latch is that when the clock is active hen the clock is high ahh so I am sorry for this picture this is these are lines these are real way form and these are the lines which shows the edges, so do not confuse and these are the kind of lines dotted lines to show the clock edges to match it, so these are the way form, so you see when the clock is high the Q, Q of the latch follows the input ok.

So when the clock is high you can see the Q is exactly same as the D but when the clock goes low whatever was input is latched and so it remains, so you can see here in the QL next time when the clock comes that clock goes active, it again produce that D, but when the clock goes low here it remember the last stage till that the clock becomes active.

So it remains 1 though the input has g1 low here, it remembers whatever is during this period and then its samples and it allows the input to come to the output and when again when the clock is active you can see that before the clock is activity it is going high but that does not come to the output when clock goes high then you get output and when clock goes low it is latched.

Ok, so that is the behaviour of the latch and the clock is high it is transparent, clock goes low it remember still the next active clock comes but in the case of flip flop this is H to the that means whenever clock at comes whatever is the input is transferred to the output with some timing constraints which will see later. So here you can see the behaviour of it and this triangle represent the edge together in action, so this is a positive edge to the flip flop.

So you can see that when the positive outcomes and this is the flip flop output, the input is same so when the positive edge comes input is 0 so the output is 0 and if remember till the next clock edge so here also the input is 0, the output is 0, the next clock at input is 0, the output is 0 and here you can see that input is going high but it is not reflected but when the clock edge active clock edge comes that is transferred to the output and it remembers, ok.

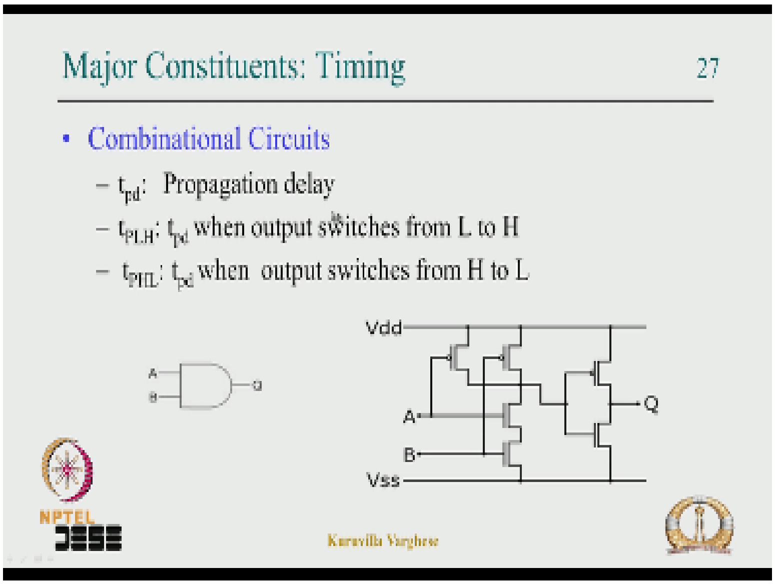
Till the next clock edge though it is g1 low it retains till the next active clock at the next clock edge it capture the 0 and it becomes 0. So in the case of the flip flop you get output in synchronous with the clock edge, you can see the warners is going high, the output is going high at this clock edge and it remains there till the clock edge not so in the case of the latch because when the clock is active whatever is the input changes will be reflected in the output.

So there is a difference between latch and flip flop and most cases will be using the edge triggered flip flop in the design. There are special cases where the latch can be used in advance cases we may not see those case in this course but we will stick with the use of flip flop in this course. So let us move so that is the sequential part we have seen and overview of the comp1nts circuit.

Now we have seen the sequential circuit, what is the difference between latch and flip flop and the flip flop is edge triggered it can be positive at triggered or negative edge triggered and it is synchronous with the clock edge and so that is the 1 important part of the digital design the function of the logic and we have seen some basic sequential circuits and the combination circuit you have learnt and this is very useful.

So keep in mind there are other parts which I am not touching upon I hope you revise and understand that and let us move on to the next important requirement, so that is nothing but the timing ok, once you are tackled the function once a circuit is working then the next important thing is the timing and now we will look at the timing for the basic timing parameters for a combination circuit and a flip flop. From there we can build the more complex timing details of the regional circuits.

(Refer Slide Time: 33:24)



Major Constituents: Timing 27

- Combinational Circuits
  - t_{pd}: Propagation delay
  - t_{PLH}: t_{pd} when output switches from L to H
  - t_{PHL}: t_{pd} when output switches from H to L

So let us turn to our slide. So combinational circuit essentially say like a take an AND gate essentially it is a propagation delay that means if 1 of the input changes how much time it takes for the input that change to propagate to the output, so from A to Q how much it takes that is the t pd or propagation delay. Now there is a difference TPD is divided into 2 sections or 2 parts.

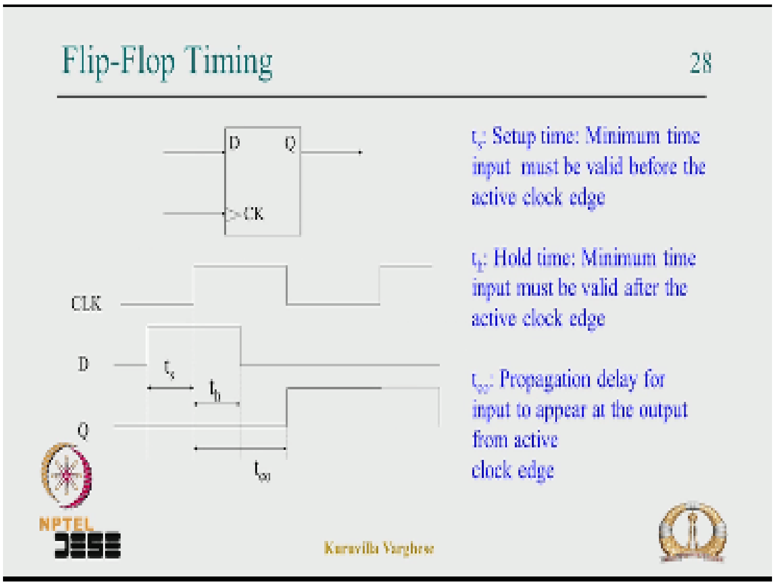
1 is TPLX and TPHL ok, so this is the propagation delay TPLH is a propagation delay when the output switches from low to high and TPHL is when the output switches from high to low, so this will be different because you see the circuit is not symmetric CMOS circuit is made of PMOS and the NMOS, NMOS circuit and depending on the input signal maybe PMOS is on maybe the NMOS is on.

You can see here the NMOS is in series PMOS is in parallel, and so depending on how the output switches whether is going from low to high or high or low some part of the circuit is switched and some NMOS is involved, some PMOS is involved, you know that mobility of electrons and volts are different, maybe the sizes of this transistors depending on the design is different.

So that makes difference in the propagation delay. So normally in a combinational circuit you need to specify the propagation delay from each input to each output, so most of the time manufacturers will specify in the symmetrical case, the worst case delay. That means from if we have 10 inputs to 10 outputs that they will specify what is the worst case input output delay which is TPLH and TPHL.

But in if there is a large variation it is worthwhile to specify the slowest input and fastest input, so that we can maybe take that into account and come out with some clever design to take advantage of it, so let us move on to the flip flop.

(Refer Slide Time: 35:42)



Flip-Flop Timing

28

t_s: Setup time: Minimum time input must be valid before the active clock edge

t_h: Hold time: Minimum time input must be valid after the active clock edge

t_{co}: Propagation delay for input to appear at the output from active clock edge

Kuruvilla Varghese

In the case of H to get flip flop there are some important timing parameters, so here it is not that when the clock comes whatever is that the input is immediately transfer to the output. When the clock comes first thing to notice is that for the input here when the clock comes the input is 1, for the output to become 1 it takes some time, that is called PCO or the propagation delay for input to appear at the output from active clock edge ok.

So that is called TCO clock 2 output or sometimes it is called PCQ which is clock to Q that means not that the data is going from the top to Q but enabling path from clock that is why is called clock to Q, the input goes to the Q, but for this to happen for the input to come to the output the input itself should needs some timing requirements with respect to the clock edge ok, so that means if the input has to go to the output when the clock comes the input has to be there at the at this point sometime before the clock edge.,

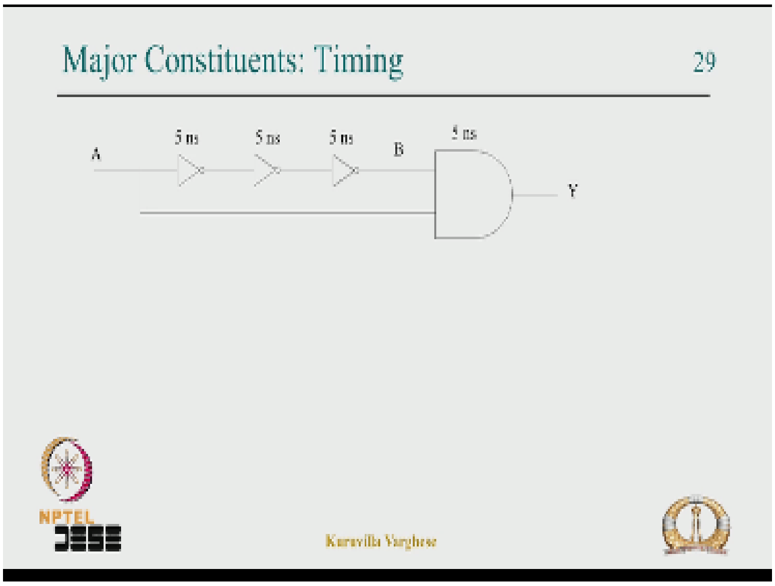
That means that time is called setup time, so input should be set up sometime before the clock edge, not only that after the clock edge it should remain there for some time, then only properly the input will be transferred to the output. If this is not met we cannot guarantee what is a state of the output, it could become 1, or it could become 0 in the worst case you can even get stuck in between ok which probably would not of study.

Again in this course I do not have time to deal with I mean the concept related that but I will at least tell you how to handle that situation we will study in synchronisation how to handle it this scenario but essentially in flip flop there is a setup time that means input has to come sometime before the clock thus to remain there sometime after the clock, that is call whole time.

You hold the input after the clock edge for some time and if that is met with TCO or PCQ delay the outfit ideas at the I mean in the input appears at output with the delay and this is the basic timing parameters of a flip flop, so knowing this that is basically the combinational delay which is PPD propagation delay which is TPLH and TPHL and this the TCQ in the case of flip flop.

And setup and hold time we can build or other timings whichever is required for the sequential circuits from this basic timing. The next level we can build on this understanding so if have not learnt this part of the time please learn it now.

(Refer Slide Time: 38:46)



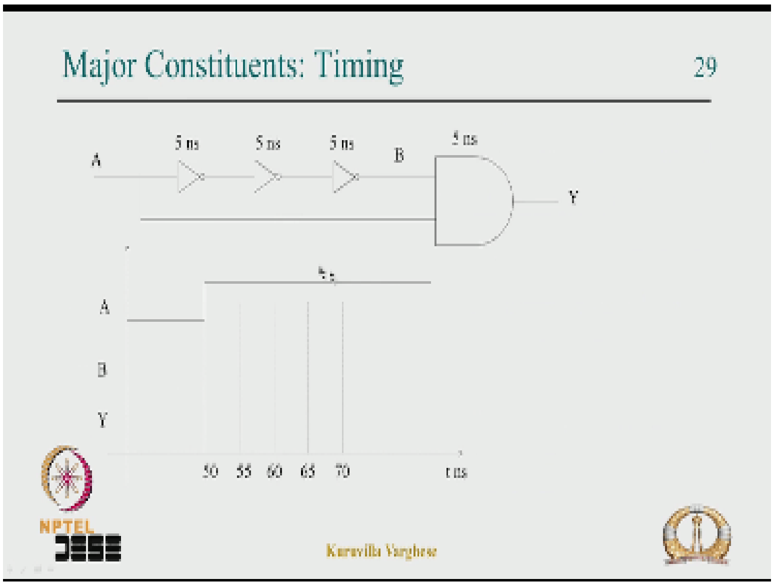
Major Constituents: Timing

29

Kuruvilla Varghese

So let us move on, so I want to show an example some kind of application of the timing, let us look at take this circuit even AND gate with 5 nanoseconds can delay, input 1 of the input is **is** coming directly to 1 of the inputs of the AND gate. The other input of the AND gate goes through 3 invertors each of 5 nanosecond delay. So this point is called A, this is intermediate point is called B and output is 5Y.

(Refer Slide Time: 39:24)



Major Constituents: Timing

29

Kuruvilla Varghese

So assume the A is changing at 50 nanosecond from 0-1 ok. Before that we assume it is 1. So if you do a static analysis of this circuit what happens is that typically you learn in the undergraduate this is 0 that mean this is 1 because of the ward number of him what does this 1 and this is 0. If this becomes 1, this input is 1, this is 0, again the output is 0. So we are expecting you know at constant static 0 at the output.

But you see because of this 15 nanosecond delay for B to change from say 1-0 something interesting can happen, let us look at what is the B when A go from 0-1. So B will change from 1- 0 i mean -10 like this after 15 nanoseconds, so at 65 nanoseconds B goes from 1-0 but you know that A as directly connected here, so A changes here from 0-1, so for 15 nanosecond time both A and B are 1 here.

And so that tall will come at the output as a 1 with a delay of 5 nanoseconds, so the output Y look like this, so you get a paths of 15 nanoseconds duration which was not intended like we were not like of tral 0 and 1 in an AND gate is 0 but we get something called a pulse, and many times it is called as Glitch and I do not whether you have studied this, this is called a static 0 hazard.

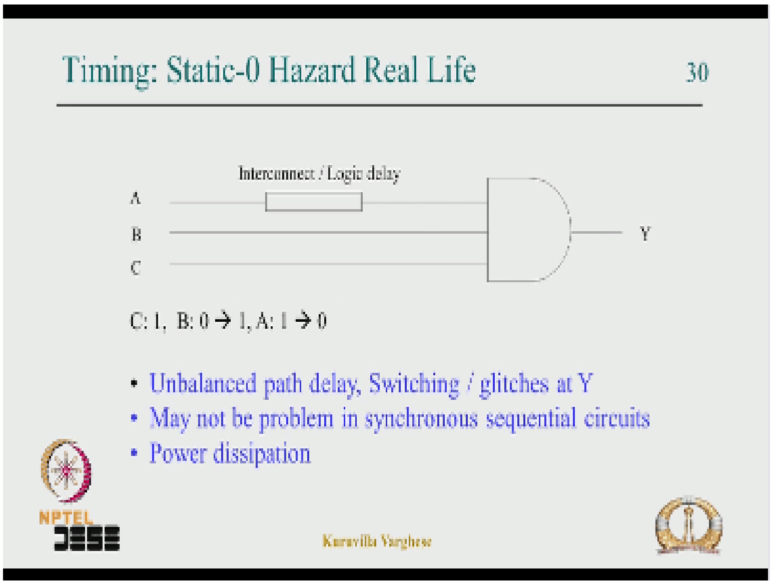
So when whenever you have an AND gate this static 0 hazard can happen. Now you can imagine an opposite of that a static 1 hazard, so if you have for that you have to have an OR gate here and similar set up but instead of A going from 0-1 A has to go from 1-0 ok. So here an OR gate input of OR gate as a same circuit and the A moves from 1-0 then you are normally expecting a constant static 1.

But you will get a 0 Glitch or receive calls. So this is call static 1 hazard, and you must have studied some ways of eliminating the static hazard and things like that using adding some redundant terms product terms and so on. But the important thing is that why are we studying this, because nobody will cook up a circular like this everybody knows that 0, 1 is 0 or in the case of OR gate 01 is 1.

So there is no need of cooking up a circuit like this only thing is that the only used for the circuit is that if you want a pulse out of an edge then you can use this circuit, and definitely we will be using this kind of structure as we proceed for precisely for same application. You have an edge which is going like something is going from 0-1 for a long time but we want convert into a pulse you can use it.

But other than that why are we studying this hazard is that it is just a model okay. In real life this can happen in very kind of very nice way which you may not detect.

(Refer Slide Time: 43:04)



Timing: Static-0 Hazard Real Life 30

C: 1, B: 0 → 1, A: 1 → 0

- Unbalanced path delay, Switching / glitches at Y
- May not be problem in synchronous sequential circuits
- Power dissipation

Kuruvilla Varghese

So let us move on to the real life case, you taken AND gate with 3 inputs which is coming from some circuit through some Is in a in a chip and assume that B is C is permanently 1 at some time incense B is going from 0-1 and A is going from 1-0. Now C is 1, B goes from 0-1 initially it was 0 and A goes from 1-0 and there i some delay hear, some logic is there here and maybe it is a wire delay.

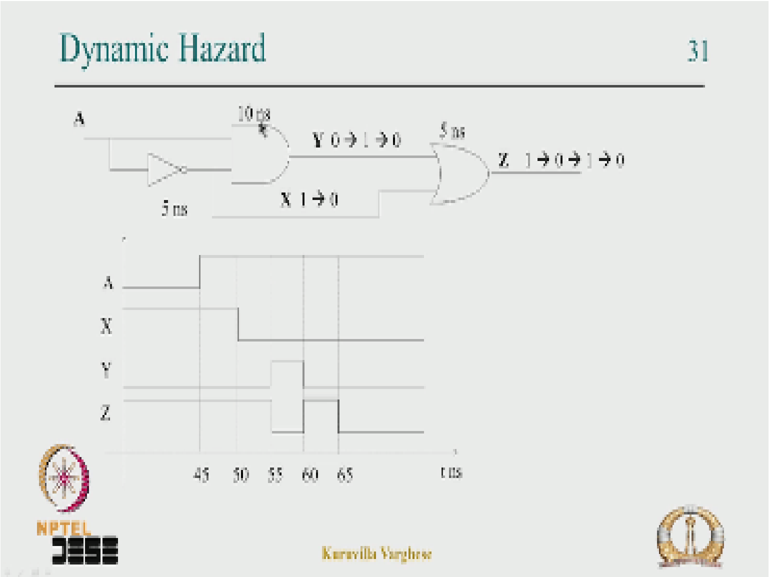
So the A does not go from 1-0 immediately it stay there for some time, so you get a Glitch here. So whenever there is an AND gate and the input does not arrive at the same time or there is an unbalance part delay you will get the glitches at the output ok, similarly whenever you have an OR gate with unbalance path delay you will get glitches ok. So that is the first thing you to understand most of the time in your basic course.

You have taken everything static that you apply some kind of set of input then you get some active high or low for a particular input. But is not going to be like that the outputs are going to Glitch and will settle to some value. So this is quite normal you need not be worried about in a moment I will tell how to handle that, so you will see lot of glitches in combination, lot of switching before getting the correct output is not a problem at all.

But the main thing is that 1 side effect of that is that is there a lot of switching then there will be lot of participation. Because the power dispassion is proportional to the switching, so if there is unnecessary switching so if they will be good from my participation point of you if you can balance delay then the participation in this gate will be less.

So that is about the hazard. And in this case we have seen that in case 1 and there is 1 and there is 1 Glitz. So but you can imagine a situation where input changes 1, once and their could be multiple Glitches at the output.

(Refer Slide Time: 45:19)



Dynamic Hazard

31

Kuruvilla Varghese

And that can happen in a multilevel secured I say take the case of an AND gate with the static hazard case which gives a 010 Glitch, but do you see there is an OR gate following which get this 1-0 transition before this 010. So I am showing A here this particular x, X is not mark I am sorry. So this is X, X is shown here so the Y is shown here this Glitch but you know that this 1-0 comes before that and both combine you will get a 1010 2 glitches at the output ok.

So it is supposed to transit from 1-0 but it goes to 0, then goes to 1 and then goes to 0. But this got normal in the case of combinational circuit and you can imagine if there is this is followed with another level of the circuit in Glitch 3 times depending on the logic. So that should be kept in mind but it normally in sequential circuit what we do is that we will have a flip flop here.

When everything settles we will give enough time for all these thing to settle then we will latch the correct output to that flip flop. So in a sequence circuit we are not worried about the glitches only thing is at when you simulate circuit you will see lot of glitches you do not worry this is quite normal. So that is about the hazard. So let us move on, so that the second factor we have talked about after the function, functional aspect I have.

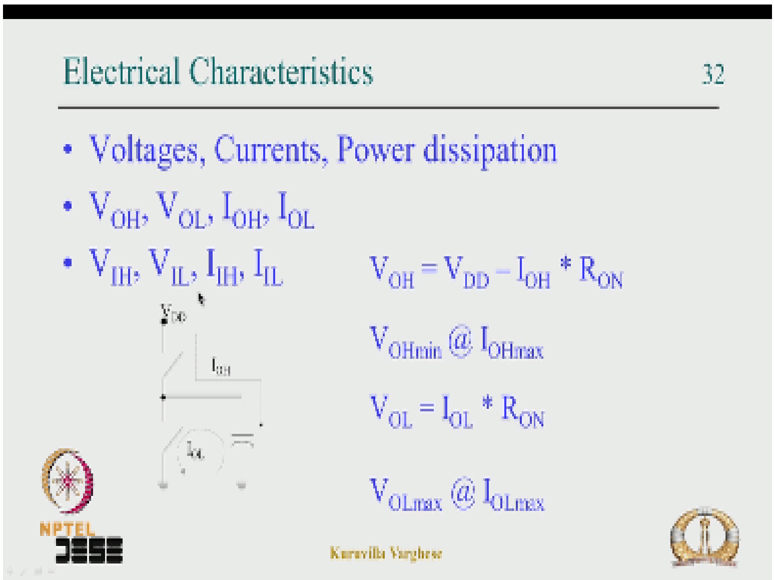
You know talked about the timing and delay parameter for the combination circuit is propagation delay and there are 2 parts, 1 is TPLX and TPHL. That means the propagation delay when the output switches from high to low or output switches from low to high because the circuit is not symmetric not all the transistors are switching at the same time and some are PMOS and some are NMOS.

So this makes a difference when a particular output switches from low to high and depending on the number of inputs for each input and each output the delays will be different. But manufacturers of unspecified the worst case delays or if there is variation it can be large variation the fastest and the slowest can be specified. We have seen the timing parameters of a flip flop, the setup time, whole time.

The input has to set up sometime before the clock at input has to be there for some time after the clock edge, and if that happens with propagation delay TCO the input appears that the output properly that does not happen then the output is not. We cannot say what will be the output it could be 0, it could we 1, it could get stuck in between then we have seen hazard static 1 hazard, static 0 hazard.

It is useful because in real life because of the unbalanced path delay AND gate and OR gate can produce glitches with normal and it can be dynamic hazard, multiple glitches but normally we put in sequence is secured the flip flop and output when everything we give enough time, when everything settles the settle value is latch onto the flip flop.

(Refer Slide Time: 48:51)



Electrical Characteristics 32

- Voltages, Currents, Power dissipation
- V_{OH}, V_{OL}, I_{OH}, I_{OL}
- V_{IH}, V_{IL}, I_{IH}, I_{IL}

$V_{OH} = V_{DD} - I_{OH} * R_{ON}$

V_{OHmin} @ I_{OHmax}

$V_{OL} = I_{OL} * R_{ON}$

V_{OLmax} @ I_{OLmax}

Kuruvilla Varghese

So the next we will move on the next important consequent is the electrical characteristics ok and these are the voltage. current and power dissipation. And again in your basic course of study this voltage levels VOH, VOL, IOH and IOL. VOH is output voltage when the output is high, VOL is the output voltage when the output is low, IOH is output current which is flowing out when the output is high.

IL is a output current which is flowing inside when the output is 0. So that is the output voltages or output voltages and currents. Input voltage VHI, that is the voltage with the circuit considered as a high voltage or 1. Similarly this is a VIL is voltage which is considered as 0, IIH is the input current flowing in when the input is high and IIL is the input current which is flowing out when the input is low and things like that.

To understand we should look at the output stage of any gate, all the gates output stage will look like this, they will B and PMOS transistor here and NMOS transistor here. Any 1 of them will be on, if the output is 1 this PMOS is on and NMOS is off output is 0, the NMOS is on and the PMOS is off ok. So when the output is high this IOH current flows and charger output capacitance.

Output capacitance means all the capacitance is there a lot of inputs connected everything lump together as a lump together as a lump capacitance it is showing. Similarly IOL is when the output is 0 this capacitor discharge through this particular transistor, now you can see that if the VOH is nothing but the VDD-the IOH into the own resistance of this transistor.

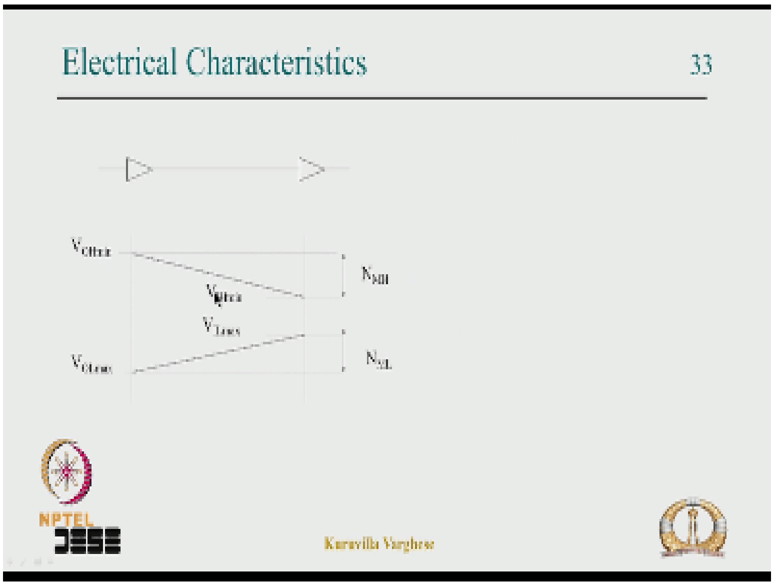
So it is VOH is nothing but VVD-IOH and RON, so as current increases this comes down. So if the current is increasing because of the dissipation across this transistor increases the voltage here goes down. So but then you know that there is a danger that it should not go below some voltage because then even input is connected here it should recognise it as high voltage.

So there is a limit below with its not go. So the manufacturers will make sure that the maximum current specified they will say that it will not go below a particular minimum value so that the input can recognise it as high. So opposite cases when the output is low the VOL is nothing but the IOL in the RON resistance of this transistor, so VOL is IOL x RON. Now you know that if the IOL goes up the VOL goes up.

So when the IOL is going up the VOL will move up. Again it is important that this not exceed the limit maximum limit of the VIL, so the manufacturer for the maximum current specials does not exceed some fix value, not only that is not that there has to be some gap between the VOL max and the VAL max. Some gap between the VOH min and VAH min.

So that some noise come that does not corrupt the logic operation. So there has to be some margin between this VOH min and VIH min as call noise margin.

(Refer Slide Time: 52:36)



Electrical Characteristics

33

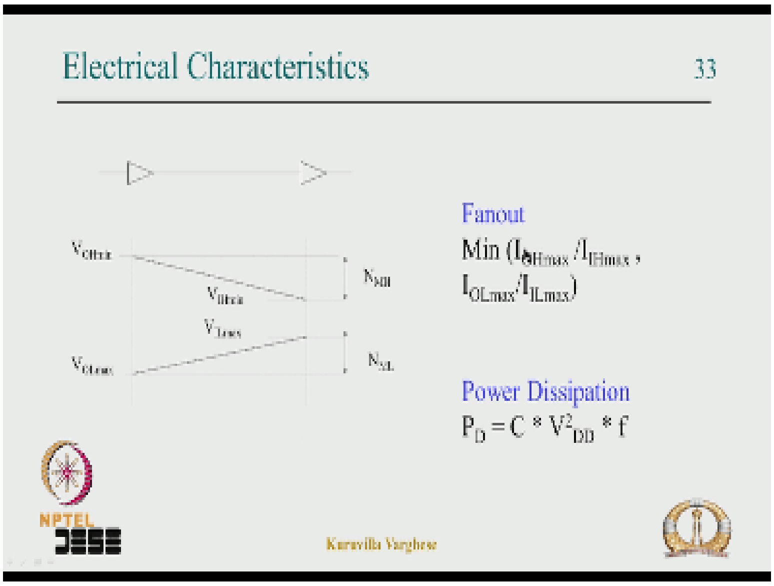
Kuruvilla Varghese

So let us look at this, so this is noise margin when an output drive an input the manufacture specify at the maximum IOH what is the minimum output voltage and manufactured make sure that the VIH min mean that means the minimum voltage which is recognised as high or the 1 is below the VOH min and that is called NMH as a noise margin. These accommodate some noise voltage.

So that if a noise couple here it does not corrupt the logic operation. Similarly the VOL max has to be less than the VAL max and the difference is called the noise margin. The next thing we bought into noise what is the fun out find out is nothing but the number of inputs a particular output can drive. So that is fine out. So how do we find out because here we have IOH and IOL.

So that means when the output is high this IOH is supplying the IOH of mini input ok. So similarly this IOL is supplying the mini IOL. So basically to find out the number of the input a particular output can support we have to find the ratio of IOH to IAH and IOL to IAL and find the minimum value that will give us the fun out.

(Refer Slide Time: 54:13)



Electrical Characteristics 33

Fanout

Min (I_{OHmax}/I_{IHmax}, I_{OLmax}/I_{ILmax})

Power Dissipation

$P_D = C * V^2_{DD} * f$

Kuruvilla Varghese

So the fanout is nothing but the minimum of IOH max by IAH Max and IOL Max and IAL max, so that is the fanout in the case of digital circuit, and you know that when a output stage like take this case the supplies VDD assumed that this capacitors C then when you know that when suppose this is switching on and off with the frequency of F you know the energy stored here is half CV squared F.

But there are 2 path with switching on and off, so half CV squared up + half CV squad F then you will get the kind of switching dissipation is not the leakage the switching dissipation is nothing but the C which is output capacitance VDD is a supply voltage and F is a frequency for an output single output stage**.** Power dissipation is proportional to the capacitance, all the capacitance lamp together the power supply voltage and frequency.

So 1 good thing is that the participation is you know quadratic related to the power supply voltage so that is why many times does it apply reduction the feature size reduction happens when you reduce the feature size you make transistor small this capacitance becomes small, so the power dissipation reduces similarly since the transistor become small the supply can be reduce and if supply is reduced from 5 volt to 2.5 volt there is a 4 times reduction in the power supply.

But this frequency many times people are not interested in reducing at it goes higher and higher the year after year. So that is about the participation equation. So here in the electrical characteristics we have seen basically the voltages and currents VOH, VIH, VOL, VIL and

IOH, IAL and we have seen that the VOH comes down as a current goes up. So there is a minimum for VOH and VOL goes up when the current goes up.

So there is a max for VOL and there has to be a margin between VOH min and VIH min, VOL max and VAL max that is called noise margin and the fanout is the ratio of the current, the minimum value of and participation is CV squared F, so the participation is proportional the capacitance that is why the transistors are made small 2 purposes the power is reduced and also the area comes less.

So that is within the same area more transistors can be put and the power supply and the feature size reduce the power supply can be reduced to for a quadratic reduction in the participation and similarly participation is proportional to frequency many times you are not able to reduce the frequency because you know the computation requirement are high but you can see that nowadays the device is your mobile ph1s or the laptops or even the desktop.

When it is not operational the frequency of the CPU is scale down, that means if it is ideal the frequency of the CPU core is reduced so that the power dissipation is reduced. So these are the 3 main constituent of least the important constituent which cannot be ignored at all that is the function the timing and electrical characteristics. And if you look at any data sheet you take care you must have seen some symbol gates like 74000 I do not know whether it is used now.

But if you look at it the first thing is the function in the data sheet. So you have the truth table of the gate followed with all the timing parameters like TPLH, TPHL followed with all the electrical characteristics and you take the case of a processor all this will be specified that function, the timing and electrical characteristics only thing is that the case of complex circuits this will be quite a lot of spec and many times nowadays separate data sheets are you know issued for each of this thing.

So that is that is I can say a brief revision of the basic. Now what is left in this overview part is that a quick review of the present state of the digital VLSI or FPGA so that you are in touch with the field before we get on with the serious design, so I hope you enjoy this lecture. Please go back review refer to the reference book, learn well and I wish you all the best thank you.