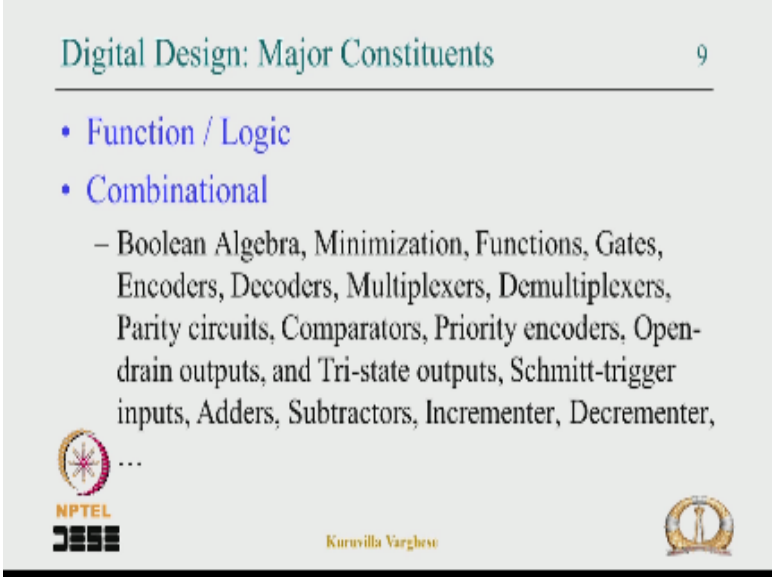
Digital Systems Design with PLDs and FPGAs

Kuruvilla Varghese
Department of Electronic Systems Engineering
Indian Institute of Science - Bangalore

Lecture-03
Design of Synchronous Sequential Circuits

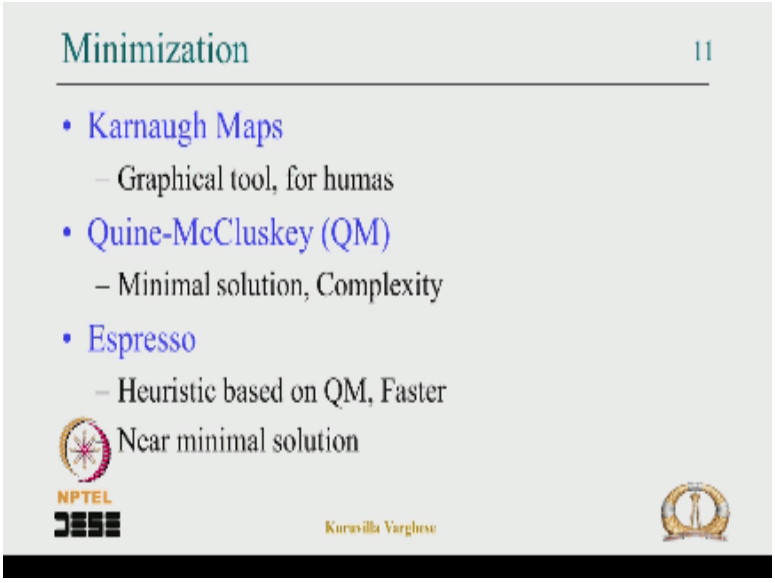
So welcome to the third lecture of digital system design with PLDs and FPGAs. The last class we had a brief for revision of the basic subjects. So let us quickly run through those lines before today's lecture.

(Refer Slide Time: 00:58)



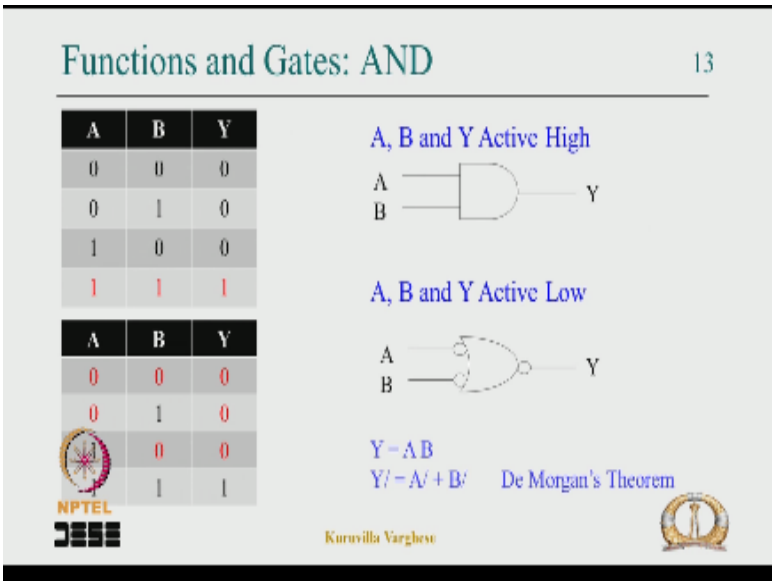
So let us know on essentially we talked about the major constituent of the digital design to start with we will look at the function and there are two part 1 is a combinational logic higher level blocks reset all those blocks which have studied in the in the basic course then we have the sequential elements like flip flops, registers memory is and all that. This is consumed the function on which have to learn thoroughly.

(Refer Slide Time: 01:22)



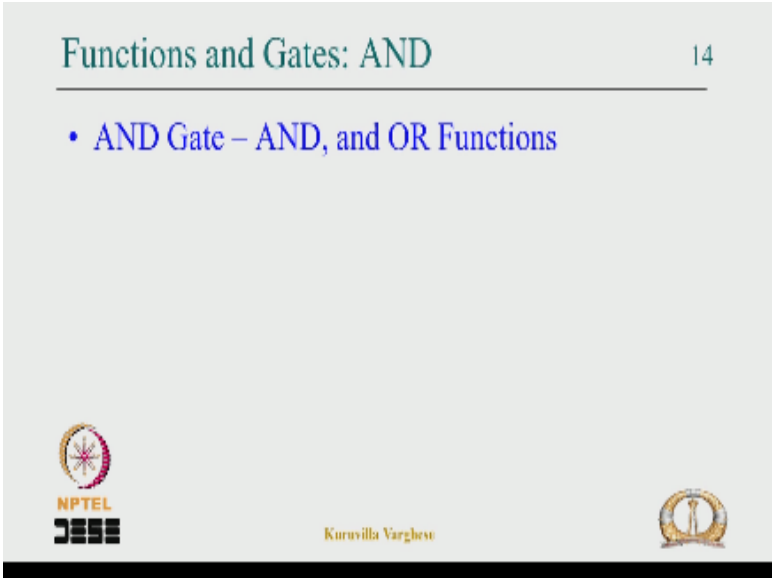
And we have looked at the minimization using sum algorithm single output minimization like Karnaugh maps, McCloskey, espresso and for there are multiple output multi level minimization which is based on this algorithm but then it involves many other steps, we had a brief look at it.

(Refer Slide Time: 01:42)



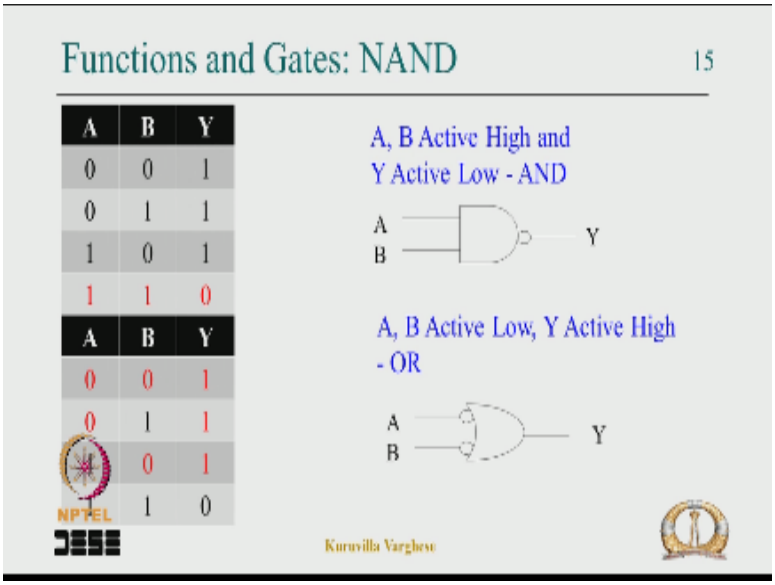
Then we look at differentiating between gates and functions. The gates are able to implement.

(Refer Slide Time: 01:55)



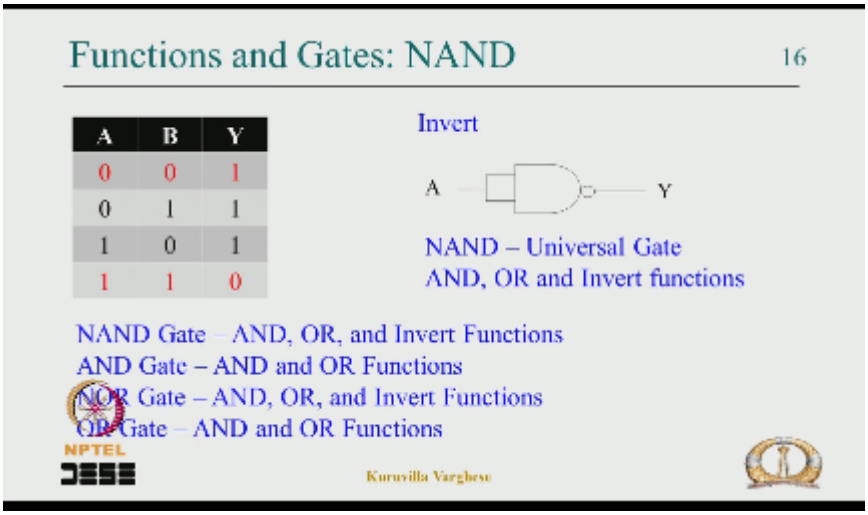
Then we had basically it is bringing the De morgan's theorem in Java concept.

(Refer Slide Time: 01:59)

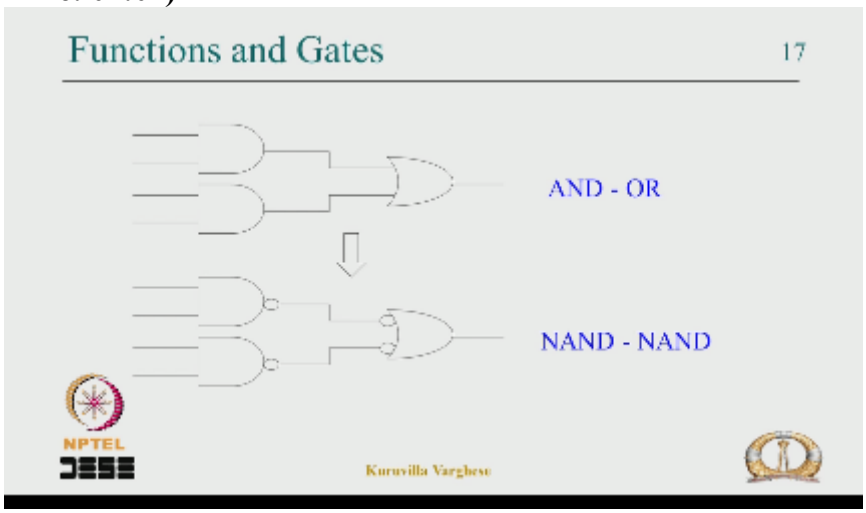


And we have seen that the NAND and NOR are Universal gate.

(Refer Slide Time: 02:06)

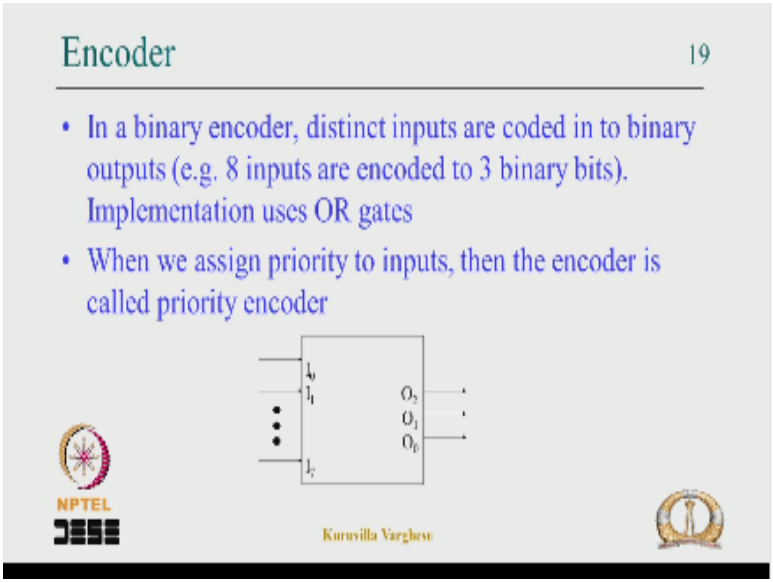


(Refer Slide Time: 02:07)



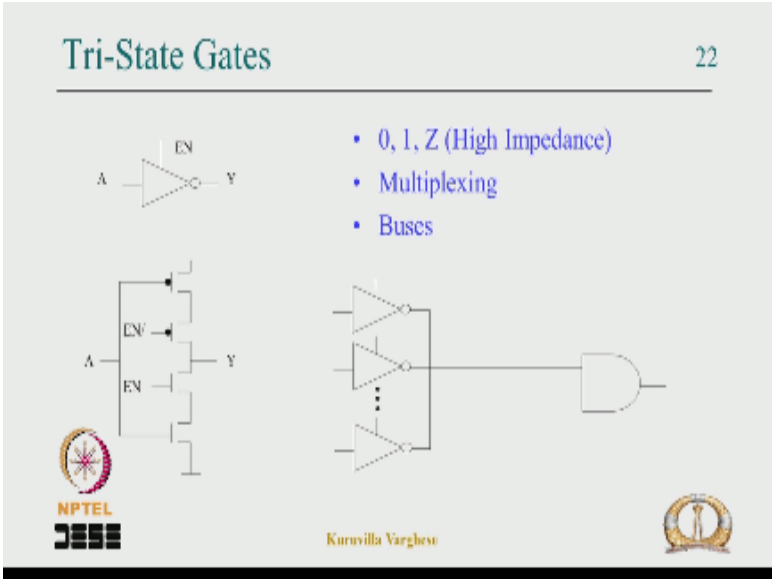
We have seen some example of AND OR is getting implemented using NAND-NAND gates and also we have seen how advantages is to conceptually thing like that then you will not raise gates if you bring this in the concept while designing some simple logic circuits.

(Refer Slide Time: 02:26)



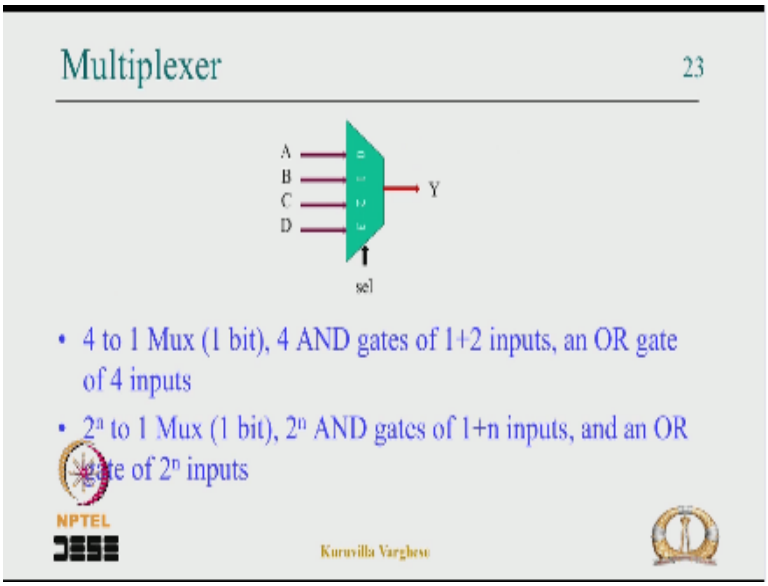
And we are looked at the encoder, decoder, multiplexer, the Tri-state gate.

(Refer Slide Time: 02:32)



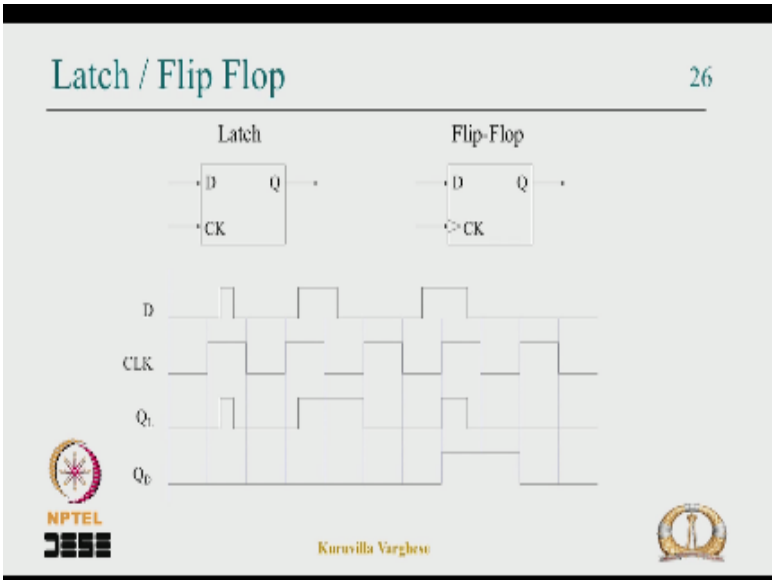
And we have seen what is Tri-state gates and some care I need be taken when you are bussing or multiplexing using the Tri-state gate or pull up pull down.

(Refer Slide Time: 02:42)



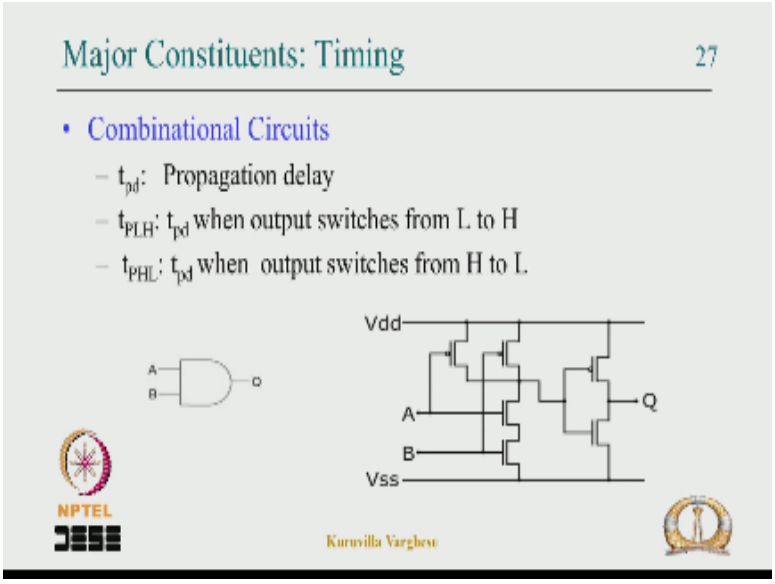
Then we will look that the multiplexer and demultiplexer and real life that it may not be like a text book picture.

(Refer Slide Time: 02:52)



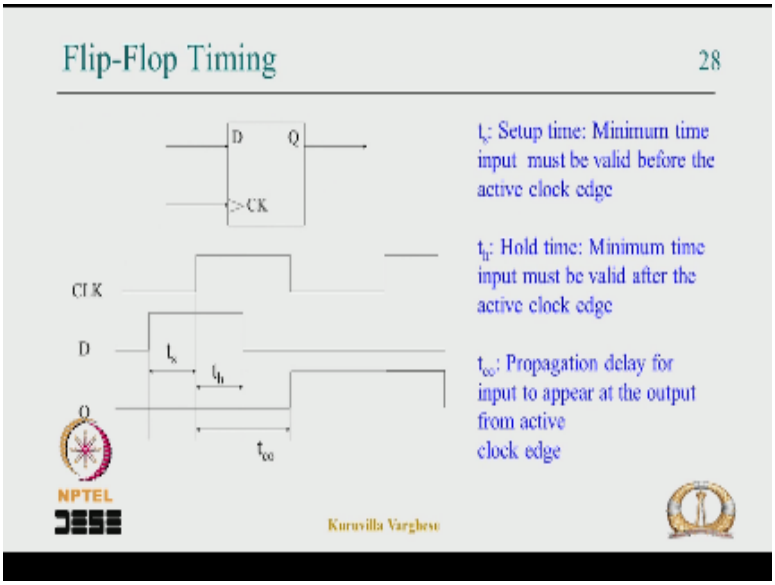
Then we have looked at the difference between latch and flip flop, latch is transparent, flip flop is edge to that.

(Refer Slide Time: 02:58)



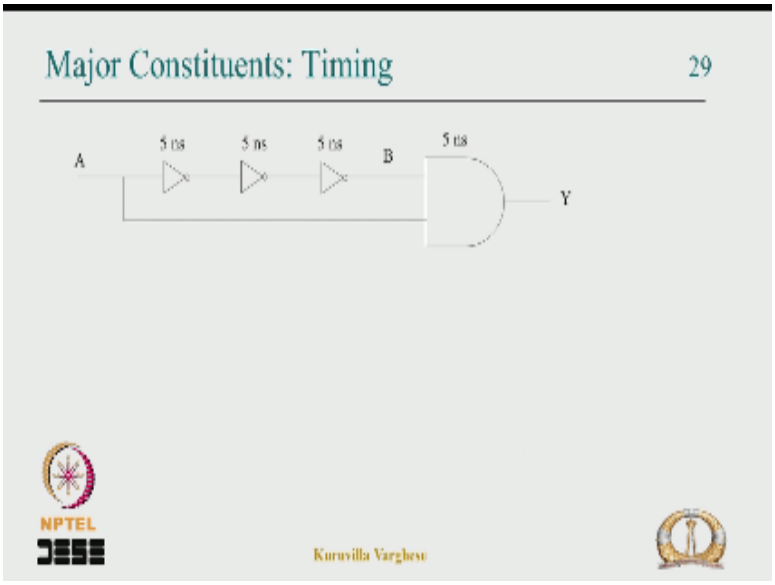
We have looked at the timing that is a second important constituent of the design and for combinational circuit what is the timing, it is a propagation delay.

(Refer Slide Time: 03:12)



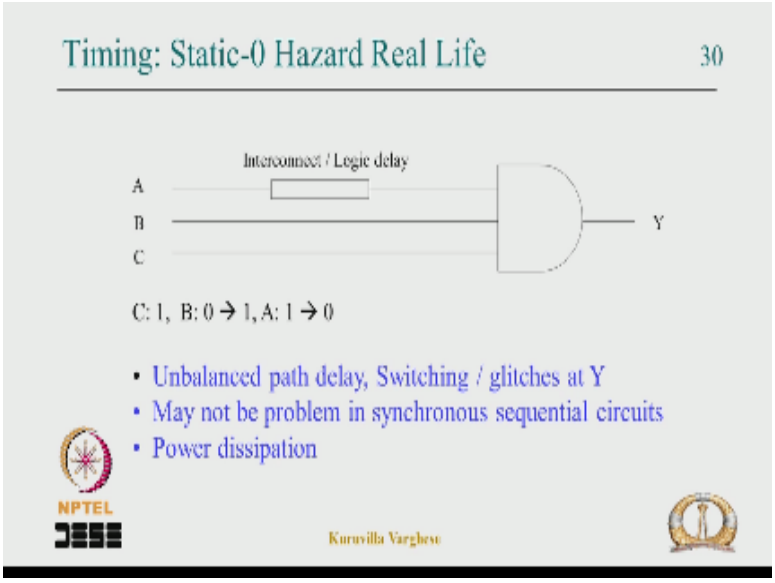
And for sequential circuit flip flop there is a propagation delay called TCO, and at the input we have meet sometime call setup and whole time to for the flip flop to transfer the input to the to the output properly.

(Refer Slide Time: 03:28)



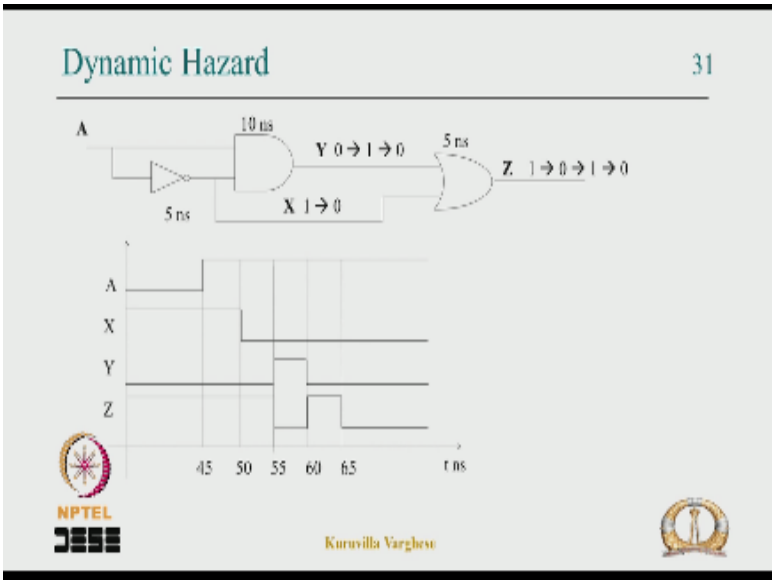
Then we have looked at something called as hazard ware in where in unexpected output you will get glitches which I said there is static 0 and hazard in a sequential circuits is not a problem, because we lost the output after settling.

(Refer Slide Time: 03:46)



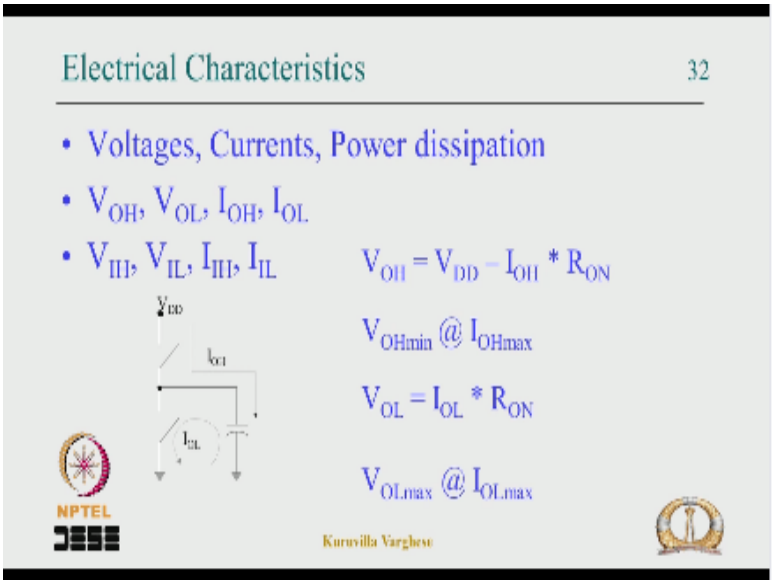
But this is just a model in real life when there is an unbalanced path delay at the input of an AND Gate or NOR gate. This glitches can happen, so one has to wait for it to settle before launching on do a flip flop or a register.

(Refer Slide Time: 04:04)



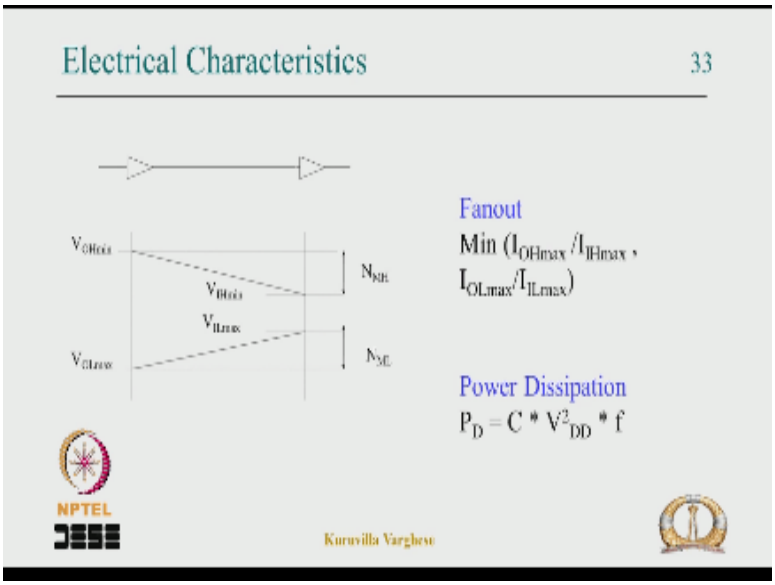
You can have dynamic website for a single change in the input that could be multiple glitches at output instead of 1 pulses.

(Refer Slide Time: 04:14)



Then electrical characteristics basically voltages and current we have seen output voltage, output current how it is the voltage is related to the current.

(Refer Slide Time: 04:27)



Then we have seen the noise margin the fanout, how to calculate the fanout, the power dispassion in the power for switching, power dissipation, power dissipation leakage, power dissipation expression for the power dissipation in CMOS output stage.

So that is where we have expression of the whole circle, so we can attack at a lower level at the transistor level, the power dissipation and the circular level we can reduce the power dispassion. The third level is that when you have a system complete embedded system say with the process with lot of peripherals and all that and a particular circuit, particular system may not use all the peripheral at the same time.

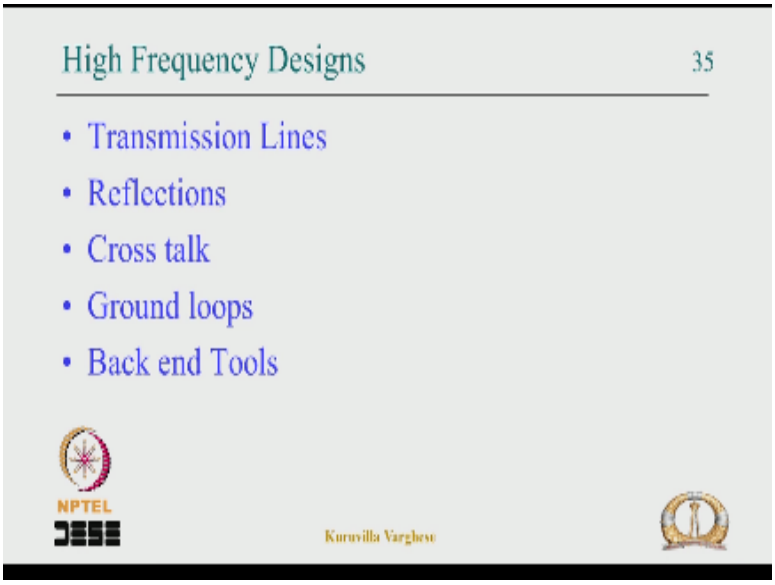
So it is possible to power down the peripherals of the module which are not working or I can freeze the clock of some register memory so that it reduces the power dispassion or you can reduce a clock frequency of the unused blocks. So that the power can be brought down and does not stop there maybe when you are designing the whole architecture you can take care that the architectural design such a way that the power dissipation in the system is reduced.

So that can be handle that the architectural level ultimately one might think it stops here but does not stop there when you choose the algorithm for some computation, you can choose an algorithm which reduces the power dispassion that could be like you have a multiplication there are different algorithms we can use for multiplication, but there maybe algorithms which reduces the power dispassion.

But it may be slow but if our primary aim is to reduce the power dispassion then you can choose an algorithm which reduces the power dispassion. So the low power design can attack at all these friends, you know these transistor level, circuit level, the system level, architecture level and algorithmic level the power can be reduced and this is a very major focus on the current the VLSI design even in is like a FPGA one would like to have low power dissipation.

But this was not the focus of our course, so I will leave I am in have given a taste of it and if you want low power to know about low power VLSI design have to go and credit at course with talks about it or read a book in the low power design domain. So let us move on what is the next issue in design.

(Refer Slide Time: 07:20)

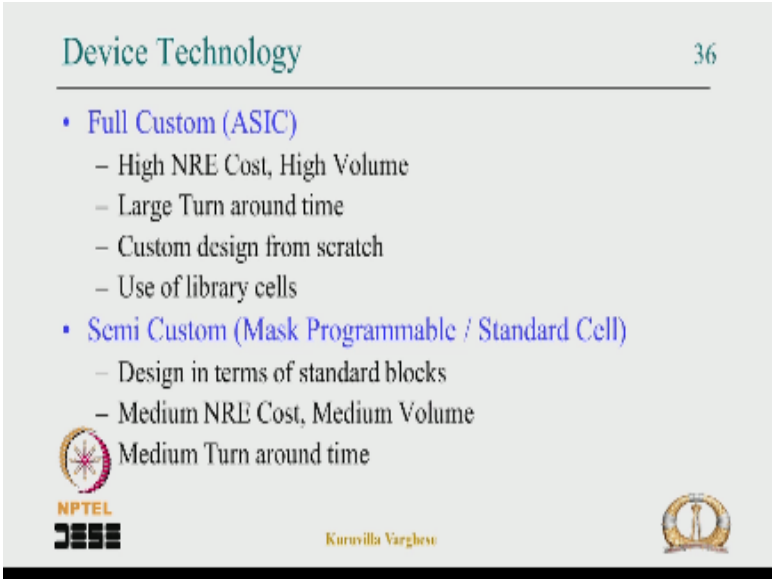


So that is high frequency design you know that in high frequency the tracks or the lines of the wires within on a PCB or a chip can act like transmission lines. So there could be reflections, so unless we take care maybe your logic is correct, the timing is correct, everything else is taken care, but because of the high frequency the wires can act like transmission line. If there is impedance matching problem there can be reflections which can corrupt the signal.

So this has to be taken care and I frequency design similarly that can be caused talk like when you have two wires running parallel for a long distance then 1 signal in the wire I can be compared to the next signal. So you can imagine what can happen if the clock line is running I close to the outline of a process then that is disaster, so this has to be taken care of the cross talk.

Similarly there is in the noise analysis there are ground loops, the ground current say there is an analogue component and there is a digital components of the chip the switching current of the ground can pass through the digital ground can pass through the analogue that can create problem in analog circuit. All this has been taken care and that is done in the backend tool and backend design. There are special tools for it and this is also an important area we need to know VLSI.

(Refer Slide Time: 08:58)



So let us move on to the next technology that the device technology. So our courses about PLDs and FPGAs. So it is let us have a look at what are the device technology available, one thing is the application specific in the greatest circuit or ASIC and this is normally supposed to be full custom. So the that means it is design from scratch you know almost everything is the sign at the gate or flip flop level.

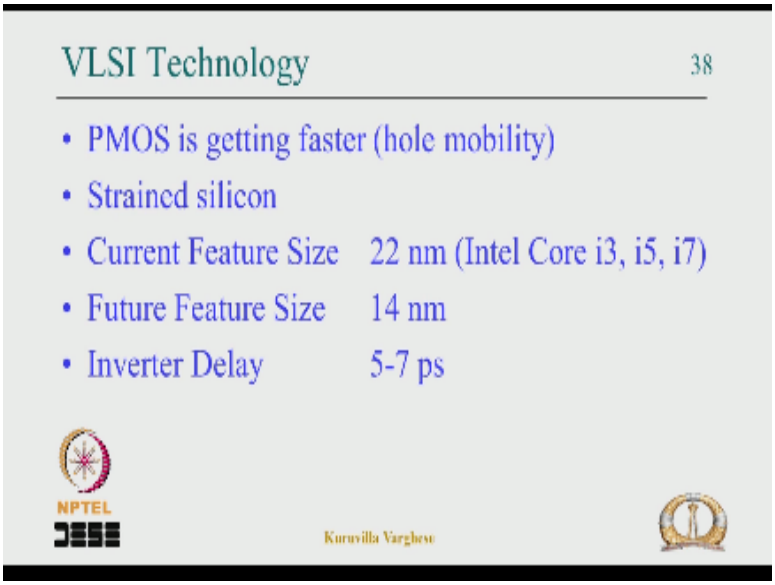
Maybe they will use some library cell but most of it is designed from the scratch. The problem is that it has a very huge cost because one has to setup the EDA tool for design tools for it, people have to work on it and the foundry has to setup and the cheapest be fabricated., So it has a huge non-recurring engineering cost, that means it is a onetime cost this huge and that has to be amortized over the number of the chief produce.

So if the cheapest is produced only 10000 you cannot possibly make an AC can get money out of it because each chip will cost so high. So ASIC are design only where there is high volume like you have the desktop CPU or the mobile phone SOC chip and so on and this also take a large turnaround time from the you know finalisation of specification to the first chip it may take 1 year, 1 and half year, and 2 years and so on.

So this ASIC is a huge costly affair but once it is design it can be made work and very high frequency like you have 3 hertz processor chip that is because ASIC is custom design, many times people do not have such luxury, they do not have the number to justify such a cause then one can go for semi custom design that is that is a mask programmable or standard Android cell that means of the foundry will give you some standard design.

And you make your system based on that standard design and the foundry has all the layout for those blocks, so it is quick when you send out the interconnection of the standard cell or the array weak it is quickly in the connected at the foundry, so this works for the medium volume, it has NRE cost, medium turnaround time. The field Programmable Gate array. So this is the device technology.

(Refer Slide Time: 11:44)



So the next I want to search on the current state of the VLSI technology, you must learn that the PMOS is slower than NMOS. Because the whole mobility in PMOS is less than the electron mobility in NMOS. But you know that the current technology like strained silicon the PMOS is getting fastest. Strained silicon means a silicon lattice structure is strained by some deposit on top of silicon.

So that the later structure you kind of structures, so that the electrons and holes can pass faster, so the PMOS is getting faster and the current feature size is the technology wise 22 nanometer and if you look at the Intel Core I3, core i5, i7 is fabricated in the study to nanometer and the next technology which is coming with people walking the manufacturers are walking is 14 nanometer where the inverted delay is of the order of 5-7 ps.

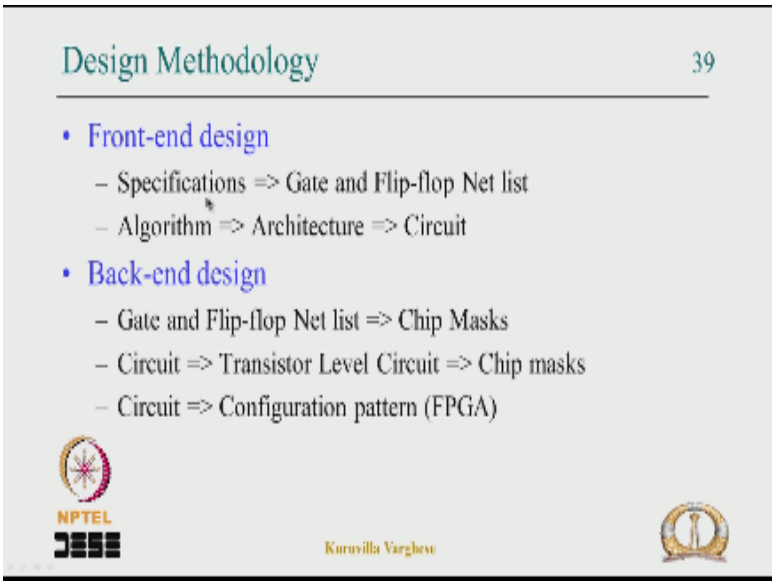
So you know how fast this technology is going to be, but now the present trend is not to make things faster you know that this is we are reaching the limit so one cannot expect this to go down considerably in the same scale as before, but people are using parallel multiple core for

computation you know that the Intel current chips as the quad core and you know 8 cores and things like that.

The present trend is not on the rose speed but to have multiple computational units parallel working whenever possible ok. So which is many a times possible because there are the inner process or even in a mobile phone does multitasking. So he will be good if you can you know listen to the music on a cell phone at the same time you going for a walk and you want to capture the picture.

So you both can work parallel it is very good you know, it is very bad you are listening to music, then you click a picture if the music stops yes, nothing catastrophic happen but the user experience might get interrupted. So I am not greatly talking about the social implication of the technology, but the good things can happen without with the multiple core parallel computing and things like that.

(Refer Slide Time: 14:36)



So let us move on next the design methodology will have a look at it you've already that is it up on it but then let us have a look at the design methodology of the VLSI and FPGA, so as I said there is something called front end design and which goes basically from specification to the gate and flip flop level circular or the net list or we can save from algorithm.

We design and architecture and we go to the logic circuit in terms of the gates and flip flops and this is many times for logic synthesis. This chapter was going from an architecture to the logic level circuit is called logic level synthesis. Then the moment you have at the circuits

here that the front end design is over, then we go for the backend design. The back end design this gate and flip flop level net list is converted to the chip mask not in one step.

First the logic circuit is converted to the transistor level secured and this is called physical synthesis that is the physical transistors are synthesized out of the logic circuits and thereafter the each transistor and interconnection is design in detail in terms of the mask which is required for the fabrication. So that is the backend design as far as the VLSI integrated circuits are concerned.

But if you take the FPGA we do not go for this step the FPGA has already the logic circuit built-in as a logic resources with the with the wire laid out with switches in between. So you literally you know go in one step from the logic circuits to the configuring the FPGA. So far FPGA this physical changes and cheap mask is not that we directly go from this circuit level to the configuration of the FPGA directly with of course there is a **a** step involved in translating the circuit and mapping it to the resources of FPGA interconnecting it and so on. So that is a back and design for a FPGA. So we talked about the frontend design and backend design.

(Refer Slide Time: 17:03)

So let us continue as I said in a digital design there is a data path where the computation happens it is made of registers and combination circuits. Registers holding the value, combination circuit computing it and there are controllers are finite state machine which control the data movement in the data path, there could be multiple controllers or single

controllers depending on the need and VLSI design we have metric or we have constraints, these are the constraints the area and normally one would like to have a smaller area.

So that the chip is smaller can be more chips and reduce the cost, and when the chip is small it will be come back and the delay will be less the wire delay will be less, and the power dispassion will be less. So these are conflicting requirements you know that we can small area small delay and small power. So where one go for small area some block you my get a larger delay.

And when you go for a very fast circuit you may end up with the large area and large power dissipation and if you have look that even very simple example like an adder so you have study ripple adder you know that suppose we have any 8 bit ripple adder, it use 8 full adder, cascade. So the area is very less.

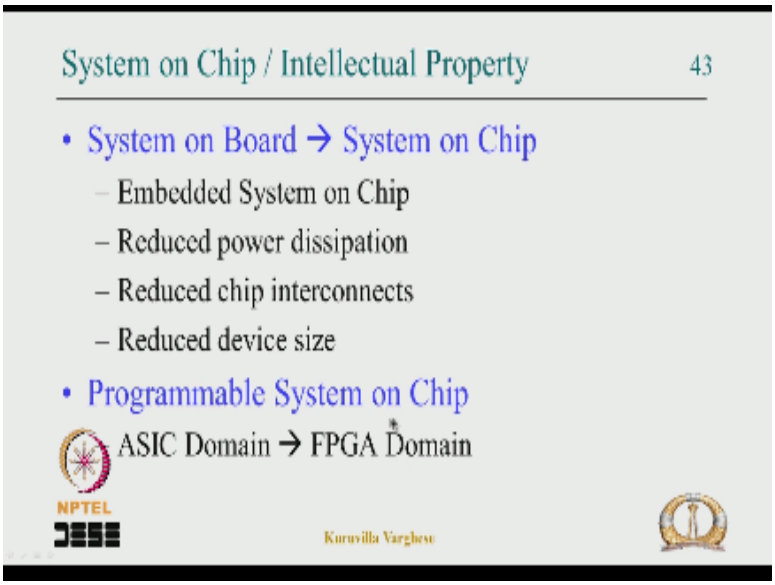
You know what is in a full adder. Full adder used 2 exclusive OR gate and 3 AND gate and an OR gate okay at the maximum. So such each stages but it is you know that it is true that changes in the worst stage you say one stay delay is a 1 nanosecond. The worst case delay of additional will be 8 nanosecond, but you can go for a carry look adder, which is which has only 2step delay.

So you anything you add you will get in 2 steps maybe 2 nanoseconds, but you know that when you go to ask you go from the 0 bit to the 7 bit the whole circuit gets exponentially larger and larger and it keep lot of area that carry look adder is not very seriously used because of this problem, it occupies huge area, so it is just an example saying that these are conflicting requirements.

You do not get everything low it is very difficult, you have to make a trade-off between the area and delay. So let us move on so the next thing I want to talk to is this thing cause system on chip which is are SOC and many times have a word called intellectual property or IP. So you know that earlier maybe 10 years back system was designed in on number of boards you have a box right based system for the multiple boards like earlier if you look at the computers.

There are motherboard, three are data board, but now it is fun to a single mother board everything is built on the board and if you look nowadays on the motherboard there are very few chips your processor chip and a peripheral chip which is also getting kind of integrated into a single step. So the complete system of the desktop CPU can go as a single chip maybe the De Ram DDR3 memory will be outside.

(Refer Slide Time: 20:42)



But things are singing the complete system for me on a chip, it will reduce the power dissipation, reduce the area, the cost will be reduced, the delays will be less and the power dispassion will be less. And same thing is happening in the FPGA dominos also. In the ASIC domain a system on board, in the FPGA domain you have Programmable system on chip ok, that means you have in the FPGA domain also.

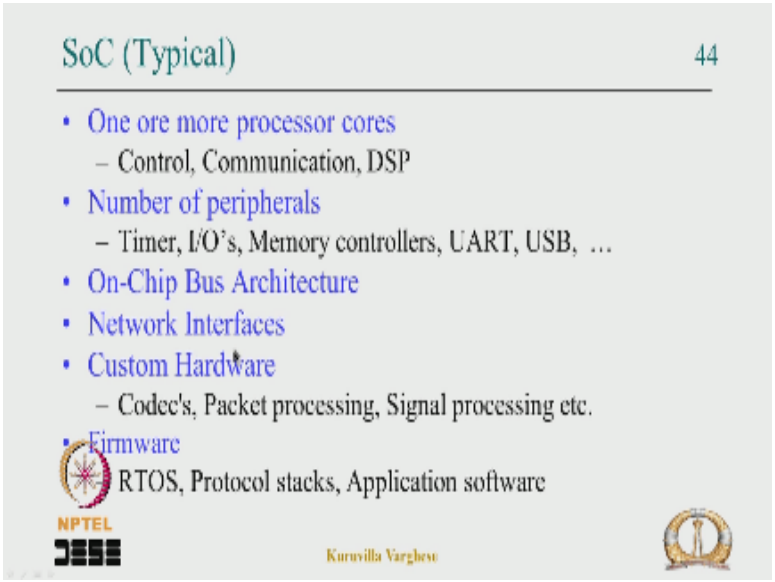
You have lot of processes peripherals are built into FPGA and you design your custom design on the FPGA, put it together for a system which is called Programmable system on chip and manufactures called system on programmable system chip so on ok. So if it takes a system on chip it has lot of things typical and I am taking an example if we have 1 or more processor Core, so you will have a communication maybe there is a processor for basic communication,

There is a process control, there is some DSP for competition, there could be multiple cores of PSP or control processes and so on. There are lot of peripheral devices like memory controller, UART, USB, the timer and so on, iOS and so on. There will be 1 on chip the bus architecture with lot of peripherals need to be inter connected, so there will be some kind of bus architecture on the chip like if you have Xi bus in a may be like in a PC you have the

other PC chips the local bus within for connecting the memories and your network in the faces maybe you have Ethernet interface or Wifi interface and Bluetooth interface and so on.

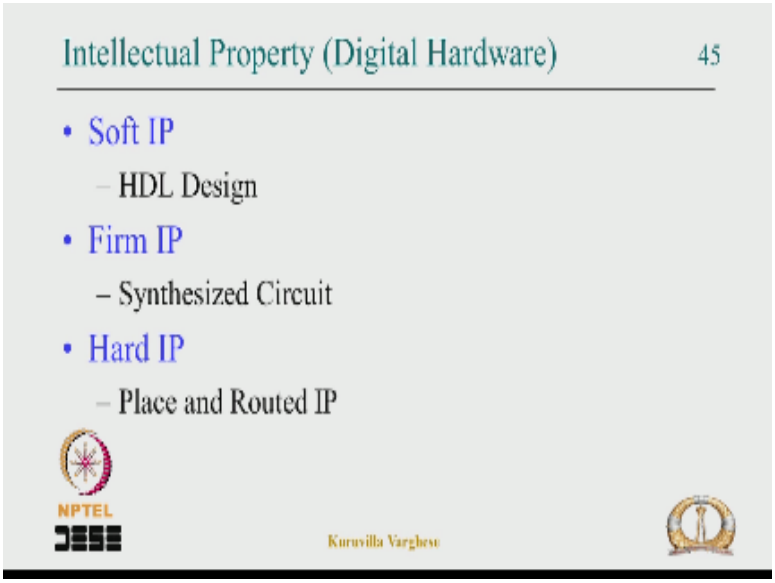
Then there could be custom hardware for packet processing, Signal Processing like Kodak you know you have a TV set top box which has an extra 264 or decoder, decoder that is required ultimately the software the real time OS or the OS, the network protocol stack application software anything in a flash memory in the chip**,** so that is an SOC, so do not get you know if you do not know what is system on chip a typical system on chip is like this.

(Refer Slide Time: 22:55)



And let us move on to the next keyword which is many a times of intellectual property which is related to somewhat related to system on chip.

(Refer Slide Time: 23:04)



When you have a complete embedded system it is very difficult design from the scratch and you know that the way that short time duration in which the cell phones are coming into the market with newer and newer chips and is very difficult to design a complete system in 6 months. So most manufacturers when they design a system on chip, they used already design components of CPUs of peripherals of Codex and so on.

And they by it from somebody already designed it and just put it together for as the as they required, so that is called intellectual property. These designs which is already done is called the intellectual property. There are different types of soft you know intellectual property one is soft IP. Soft IP would mean suppose you have a supposed memory controller in VHDL or Verilog you can buy.

And you said very flexible because you can change it depending on the license, but then you have to synthesise you have to do the frontend design and backend design, maybe it may not be depend on your tool or your the way you design in may not end up in an optimised assign and may not give the decide delays, it can be slow. So soft type is very flexible, but may not give the performance.

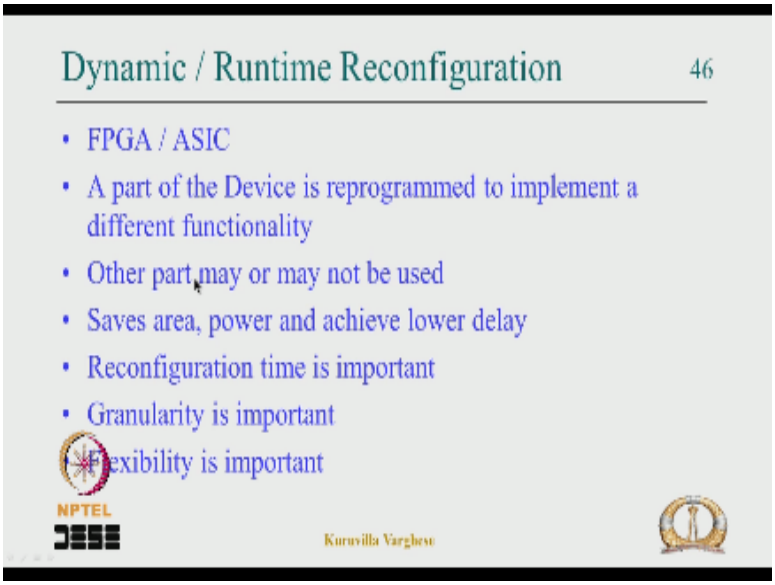
The next is a firm IP which is nothing but a synthesise I period means the logic circuits synthesise, so maybe optimally synthesise with the minimal logic circuit. So it can work faster but the changes are difficult now, because earlier it was a hardware description language you could easily change the design. Now you have to change in the synthesise design in terms of the gate and flip flops.

It comes some it is a better option than soft type in terms of performance but flexibility is less. The last category as a hard IP where this is really not the synthesizer could literally place and router IP. so you can go directly place it in the Silicon, the Silicon the mask is ready for that circuit, so you integrate this hard IP with the other hard IP the backend 2 level.

So the advantage with this hard IP is it is already tightly integrated very optimally please run out, so it can be very fast but you know that any change in this hard IP is very difficult because it's already kind of physical synthesis over, the transistors are over, chief master ready to change it is quite a tough. So that these are life is but the present the design find goes for the IP design and people make it is and sell it to other people.

You know the arm chip, the arm processor, this is an IP address not a cheap mostly you buy from that manufactures, the Arm Consortium give it to whoever want set the licensor and they ask for some royalty and people are very happy with it because there a lot of peripherals and process along with the peripherals can be integrated and there are lot of standard IPs available, standard OS available for arm processor.

(Refer Slide Time: 26:38)



So let us move on this is the next technologies that is something called dynamic or runtime reconfiguration which is very much talked about research pretty much. So this can walk in FPGA and ASIC level and the basic idea is that a part of a device reprogram to implement a different functionality, that means take an example say you have a SOC which is used in a mobile phone and suppose you are working with bluetooth ok.

Then that bluetooth part of the hardware is programmed into the chip ok. Now suppose after sometime instead of bluetooth you are connected to the WiFi okay, the wireless network not bluetooth and you are not using the bluetooth at the same time. Then what is done is that at the runtime dynamically this bluetooth part of the hardware is removed and the Wi-Fi part of the hardware is come.

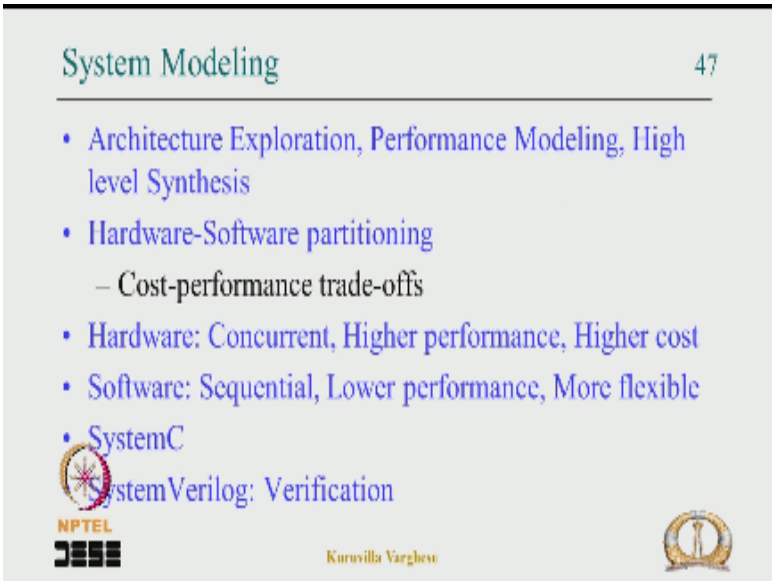
So you can save lot of area and reduce the participation and the cost anything like that this can happen this is supported in FPGA, but can even do this clearly in ASIC which says area which allows reconfiguration maybe you can even think of a system level reconfiguration

like you have a general-purpose device may be with can act as a cell phone, after sometime it can act as a general computing device like tab or pad like that.

So it is quite advantages, here the important thing is that the reconfiguration time is important, you should be able to reconfigure a very fast suppose use and of the bluetooth and you switched on the Wi-Fi then if it takes some 10 second for the hardware configuration then just may not you know satisfy the use of the reconfiguration has to be fast and very important is a granularity is important.

What part of the circuit can be reprogrammed is very important, suppose the manufacture restrict saying that only half the part of the chip can be reconfigured and all of the half part has to be reconfigured and think like, that other granularity is important and the flexibility how flexible it is like we should be able to reprogram in part of the chip or any area of the chip flexibly. So this dynamic reconfiguration this granularity the reconfiguration time the flexibility is very important.

(Refer Slide Time: 29:33)



So let us move on this is another area is embedded system is concerned like also when system on chip is considered at top level you need something call system modelling, when you start with the system on chip you are not sure what part should be in hardware and what part should be in software. So there are many times you know that the hardware design is done in HTL and software design is done in C or C++Opportunities

And is very (()) (30:02) thing to like once a partition is made the hardware designer start designing the hardware and software designer start designing software and half the way you realise some software parts has to be moved to the hardware, it is a very tough thing. So it will be good at a system lovely can model and study the performance at the cost. That means hardware you know that is concurrent we can make it very fast but is very less flexible and the cost is higher.

But software is sequential it has a lower performance compared to the hardware but it is more flexible you want to change the software we can change it is easily, so it will be good to have a system modelling language we can model it together in 1 language and you know workout on the cost performance paid off and preset then go for the you know the hardware design and software design after partitioning.

So the languages like system C allow the system modelling and there is a system Verilog language used mainly used for the system verification. So that is all about the system modelling. Let us move on your have learnt you know like suppose your learnt the VLSI design or FPGA design that in itself is not an end because if you know how to design VLSI chip and FPGA chip.

But it is of no use but you know that you have to design the cheap in some area. So it is very important know that where is these chips are used in which domain it is used ok. So maximum the ASIC or FPGAs are used in communication networks. So if you are an VLSI or FPGA designer you should have some background in communication networks, so let us turn to the communication network.

So you have learnt about the layering of the protocol stack, say layer 0 is a physical layer, layer 1 is a physical layer, layer 2 is a data link layer, layer 3 is the network clearance and so on. So we will take the physical layer most it is analogue and digital mix. If you take the data link layer or many times it is called Mac layer or media access control layer.

It is digital mostly and in the case of computer hardware but it is controlled by a network driver, the operating system driver to control this data link layer. So whenever a new networking products are come like a you know you have the Wi-Fi or of the latest protocol stack then a lot of design has to happen in the data link layer to implement this in a chip.

And similarly network layer that is a layer 3 where the packets are switched or router and that is as far as a N systems are concerned it is done in software but if you look at the routers and switches on the main network this will be done in hardware. So there is lot of hardware work within the communication network, so it is worthwhile for VLSI or FPGA designer to learn about communication network, communication protocols.

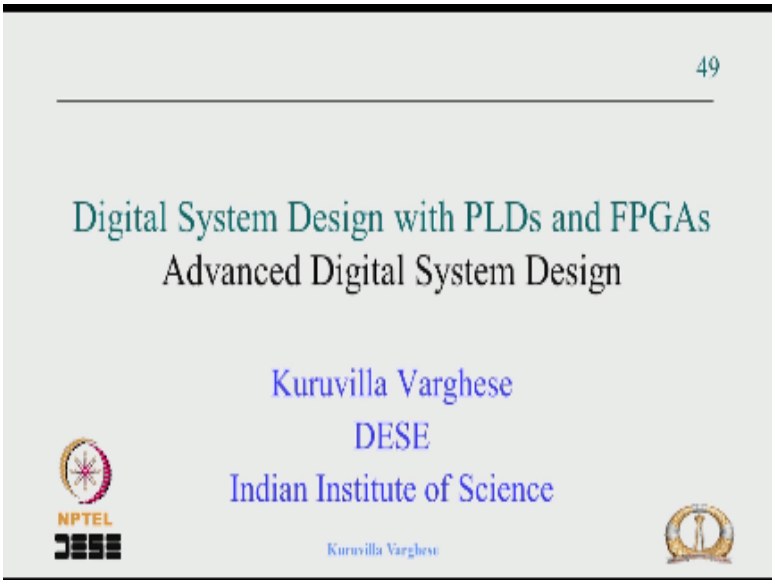
Because lot of communication network based chips are designed. Similarly lot of things are happening in the signal processing like multimedia, the video, encoding video, decoding, h 264 and various other contacts need to be designed filters need to be design lot of compression need to be done all this happen in the signal processing area. So far and VLSI or FPGA design it is very important to have some basic background the algorithm of signal processing.

So that you can implement it and ultimately computer architecture within the day like designing a CPU the nuts job of VLSI design are many times it is prototype then they FPGA that then move on basic the mind. So you are a FPGA design can move ultimately to the ASIC design. So the computer architecture, so 3 areas one is communication network or computer networks, then signal processing computer architecture.

So that you should learn the basics of it, so that when you design for a particular domain this knowledge and you will be a good designer in the field. So better know this so that gives an overview of the field, so we started talking about the various issues like application areas, design methodology, device technology, then the system modelling ,then the high frequency design, low power design and so on.

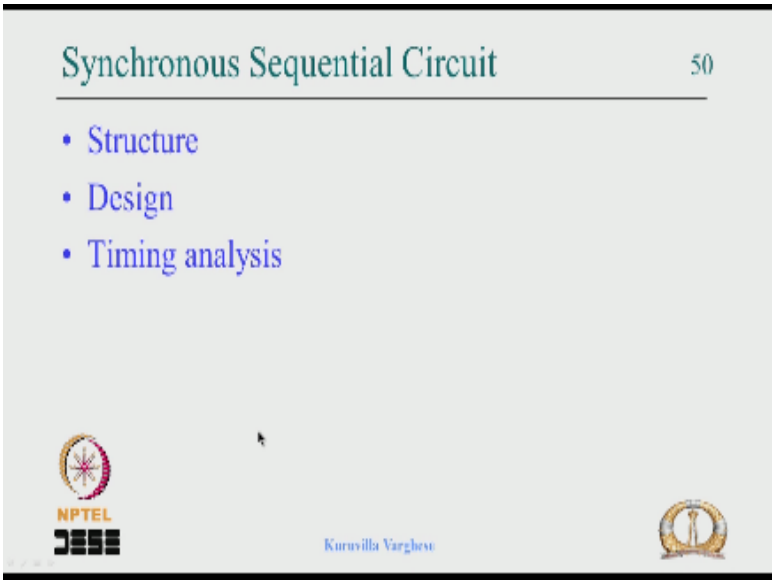
Just give a glimpse of the state of the art of the current technology to be aware before moving into the nitty-gritty of digital design. So we have some time left. So let us move on to the to the digital design**.** So let me put the slide.

(Refer Slide Time: 35:24)



So I call advanced digital design, so in this advanced digital design we will you know conceptually developed from lower level to the higher level . We will start with small system then built, we will see how are the biggest systems are built upon these basic building blocks and what are the standard basic building blocks to understand the structure, timing and so on ok.

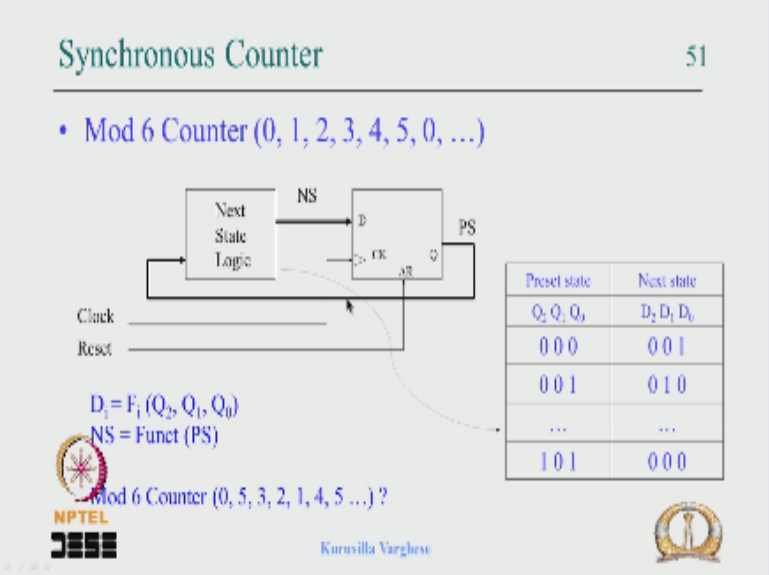
(Refer Slide Time: 36:02)



So let us know on the first thing let us take the synchronous sequential circuits ok. So that is the first thing, so I assume as I said before I assume all the background that you have learnt the combinational circuit, sequential circuits and now you are coming to the serious design sequential circuits and synchronous sequential circuits and we will learn 3 things essentially.

The structure of a synchronous sequential circuits, for what is architecture of it, how to design it snowing the structure how to design it and do the timing analysis of the synchronous sequential circuits. So that is what is shown in the slide so we will study the structure of asynchronous sequential circuit design of it and the timing analysis of it and you must have learn.

(Refer Slide Time: 37:00)



We will start with the basic very simple thing you have learnt that is asynchronous counter, you must have learnt in your undergraduate program something called ripple counter ok. This is a very cute so cute circuit, which has less use you know you have suppose ripple counter you have the least significant bit output of which will talk the next bit and so on ok. This is not timing wise not a very good circuit.

So forget about the ripple counter, let us take the synchronous counter. Synchronous counter is a simplest synchronous sequential circuits we can study and the structure is same for all the synchronous sequential circuits or at least we can build on the basic synchronous counter. So let us try to design a very simple counter. So let me ask you how to design a mod Mod low 6 counter ok. So Mod 6 counter has 6 States or 6 count which start with 0 is a binary counter.

It start with 0 then it count 1 2 3 4 5 and then goes to 0 okay and then it continues it looks ok that is the Mod 6 counter. So how to design this counter ok. So let us go step by step. So we use binary encoding, so that means we will and got this 6 in Flip Flop. So we need logarithm to the base 2 number of swaps which comes you now some three point some numbers, so 2 point some number.

So we have to use 3 flip flops, so let me put 3 flip flops ok. Now I am already have starting in your basic course you would have already put 3 flip flops separate and try to design. So the first thing the notice is that we cannot go you know the complex system you know separate flip flops you know because you try to design something 32bit you cannot put 32 flip flops and in a paper or anywhere and design.

And we are also not you know that we are also not going to detail Gate level design we are going to boot use a higher level blocks so we do not need to know all the all the details which s done by the to lock images is done by the tools of we go to a higher level, so I have put 3 flip flops together and you can see there is clocks are tight together in asynchronous circuits all the clocks of the flip flops are same.

So you get a single clock is connected to all three flip flops, a reset is connected to all the synchronous reset. Now there are three Qs to Q2, Q1 Q0 and 3D D2, D1, D0 ok. Now assume that the power on, we power on this the power on this has become 0,0,0, three 0 that means a value is 0. Now when the clock comes we should get the 1 here ok. So that is basic counter design.

So how to make it 1 ok. So we only have, we have really made the 0 and somehow we have to derive in the next clock we have to get the 1 and you know that when the clock comes whatever the as far as the flip flop or registers are concerned whatever is here is transferred to the Q. So it is worthwhile, it is enough if you have already here so when the clock comes 1 comes here.

And in the next clock 2 should be ready here. So you get the idea you at the beginning have 0 here but we want 1 of the input, so it is then if you put a combination circuits here and feedback this the present count to the input of that circuit. So that then we design that so circuit such that given 0 that will give 1, given 1 that will give 2 and so son ok. So what I am saying is that we will put a combination cute before the flip flop and the input of that is the output of the flip flops.

And output of that component secured goes to the all the input of the flip flops, ok. The basic idea is that we design this component circuit such that given 0 it gives 1, given 1 it gives 2,

and so on ok. This was then when the cock comes initially it was 0 and next clock if the one comes here and that 1 gives 2 here. Next clock 2 come here and 2 come here and so on ok. So you design this circuit then you get a counter which count you know sequentially 0, 1, 2, 3, 4, 5.

And you must have leant again in the basic course something like you want to Mod 6 counter and you have Mod 8 counter, then what you do this that which counts are 2 to 8 part then you catch 6 here fixed decoder and reset it and then it becomes 0 and so on, on phone but there is a problem if you reset it using an asynchronous reset for a brief duration it might go to the next state and so on.

Like you wait for the 7 to come here then you reset it then 7 there is a Glitch 7 at output but it is a very neat design, it works perfectly. But next question is so that is a structure of asynchronous counter you have flip flops and many times we called this is a present state of the flip flop and it goes to a combination circuit and the output of the combination circuit is called next state.

Because up on the clock the next state become present that is why it called as next state and this combination circuit is often called next stage logical because it is the logic are the components circuit which decode the next state from the person state. Now it is very easy to design, we have to design, basically we have an design this combination circuit and we know how to design a combination circuit or write the truth table in terms of the input.

Present state is a input and the next state, so you write the truth table, so we take these components circuit, develop the truth table, the present state has three inputs Q2, Q1, Q0. The state as let us call it D2, D1, D0 because it is going to the D2, D1, D0. So the moment you know that you write the truth table we have to look at the D2 separate, D1 separate and D0 separate.

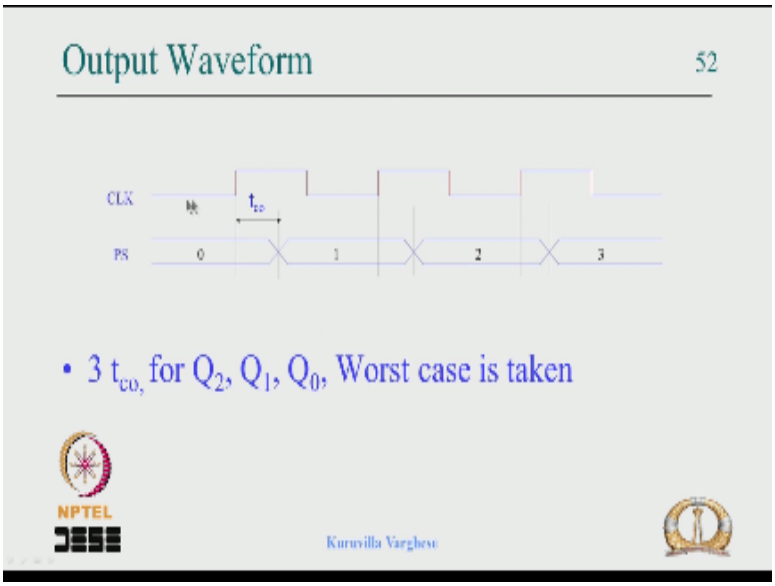
The normal course of time you would have picked all the 1s in this D2 column and go to the corner of map minimise it and you come out with an equation or a sum of product equation in terms of Q2, Q1, Q0. So ultimately you will get three Boolean equation in terms of D2, D1, and D0. So let us write that I said Di is FI of Q2, Q1, Q0. So I is nothing but 2, 1 and 0. So you have D2 is F2 of Q2, Q1, Q0.

D1 is F1 of Q2, Q1, Q0. D0 is F0 of Q2, Q1, Q0. SO you get 3 equation that can be implemented, so normally we are not going to do this hand level design, it is enough if you describe the truth table in terms of the hardware description language in a high level design then you get the design ok. So many times this is also called the next state is a function of the person state.

So we call it next state is a function of the present state. So mind you the next question I want to ask is that we still take the Mod 6 counter, but it does not count sequentially like 0 1 2 3 4 5 0, it counts in a crazy way that it start with 0 then 5,3,2,1,4,5 ok. So the question is can we design such a counter, you know already the answer very simple absolutely no worry, anybody specify anything.

You change this next state logic, you change the truth table, you write start with 0, the next state is 5, you write 5 here, the next it is 3, you write here, the next it is 2 and so on, then you take from the question or write the truth table in VHDL code, then this should be you you know perfectly ok can be designed and so let us look at the little bit timing how this present state changes with respect to the clock to give you an idea. You may already know it but then let us look at the timing, the simple timing of a counter.

(Refer Slide Time: 46:21)



So let us move on, so here this is a clock wave form query form, you have the clock edge and this is a positive clock edge triggered flip flop, so at the positive clock edge the present state changes its state and that cannot happen immediately in many times in text book will be

shown that this state change will have you know coincide with this active clock edge, practical circuit with delays it does not happen.

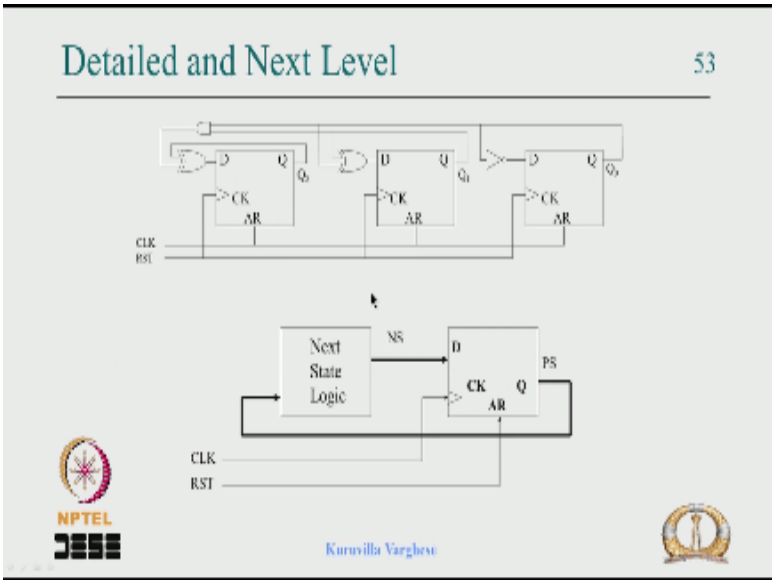
You know that you are the flip flop as delay called TCO, when the clock comes this D is transferred to Q with delay called TCO. So after the clock edge it takes TCO time for the present state to change. So when the clock edge comes the state will change, the count will change only after the TCO. So similarly here it is 1, then but when the clock edge comes after TCO delay it changes to 2.

And after TCO delay it changes to 3 and so on. Mind you there are three flip flops, so there is Q2, Q1, Q0. So there Tc 01 TCO 0 TCO2, so I am showing leave on time, so there are 3 TCOs we are taking the showing the worst case timing delay okay. So this is one disadvantage when you go for little abstract high level block diagrams it heights lot of details.

Actually you are used to maybe showing the 3 flip flops separate in thinking that clarity comes, but you should be very careful when you put higher and higher level blocks more and more abstract you should not forget the details involve, so here you should know that when you said TCO the three TCOs we take the maximum for the maximum like 3 outputs are changing, we wait for the latest output to settle.

There will be lot of glitches here, I am showing just state change, but there can be glitches depending on the state change like from 1 to 2 there could be glitches ok, because there are two bits are changing like 1 is 0, 1 and 2 is 1, 0. So there are Q0 and Q1 is changing state there could be transit free state in between. So there could be glitches which I am not showing here.

(Refer Slide Time: 49:00)



But this real life it can happen. So let us look at a picture of a binary counter I just want to show that picture ok. So this is a real picture, this is what your views I am showing a mod 8 counter which account from 0 to 7. So you know that this is a picture we have shown in abstract next level higher level block diagram, but here you D0 is Q0 bar, D1 is Q1 exclusive or Q0 and D2 is Q2 exclusive or Q1 and Q0 and so on ok.

So when we put it you should not forget that D0 is a function of Q2, Q1, Q0. D1 is a function of Q2, Q1 and Q0 and so on. In the case of a binary counter you know that the Q2 only depend, D2 only depends on you know the Q2, Q1, Q0 and D1 only depends on the Q1 and Q0 does not depend on Q2. But if you take a counter with count not very sequential e all this can change maybe the D0 can depend on Q2 also.

So that is just to say that do not forget when you such block diagram do not forget the detail picture you should have in your head in the mind. Otherwise you can make mistake in analysis when you analyse them circuit, debug some circuit you can forget something and make you know it can create trouble. So we have looked at the basically at the structure of a synchronous counter.

How what is the structure it consists of flip flop and combinational circuit and flip flop output is called present state and combination circuit is called next stage logic, it is basically designing the next stage logic. All the intelligence is in the combinational circuit, the flip flops are used to control it, control whole the value at every clock this looks you know this is like gate which allows you know gate is a open.

Then the next account comes in at the precise interval you know it is precisely control every clock edge the count changes, then we have looked at the output waveform, so in real life the output changes after if you delay which is the worst case delay of the multiple TCO of the output and finally I have told about that you should do we abstract to the next level of the diagram.

You should not lose sight of the detailed diagram because if you forget it then you can make mistake in analysis in understanding and so on. So I stop here, in the next lecture we will continue the timing analysis of this and we will build on this, you know we will go on to the real life design, system level design very soon. So I suggest you go back to the this lecture, revise do some simple design of the counters look at the structure of that, try to understand the timing details and so on. I wish you all the best and thank you.