Digital Systems Design with PLDs and FPGAs

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

Lecture-36
FPGA Interconnection, Design Methodology

Welcome to this lecture on field programmable gate arrays and the course digital system design with PLDs and FPGAs. In last lecture we have started with the field Programmable Gate array, at that point we kind of contrast the FPGA architecture over the CPLD essentially we told the CPLD has a kind of a central interconnecting switch between the logic blocks.

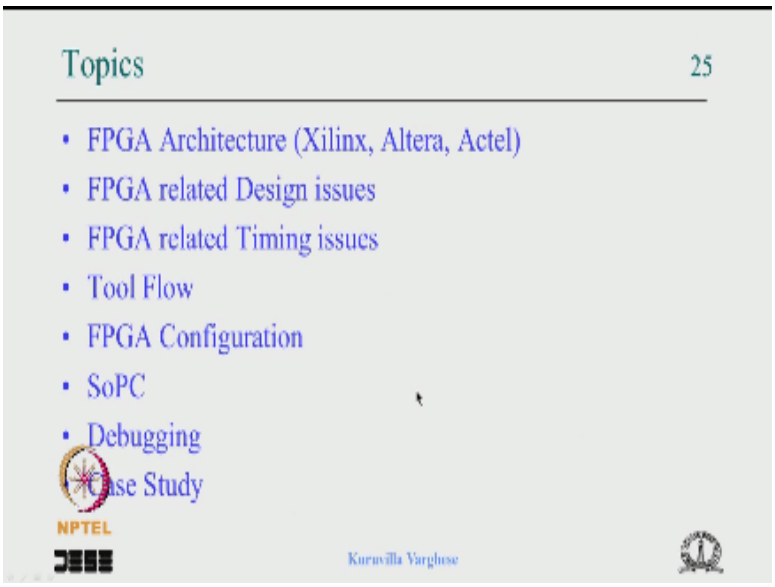
But in FPGA this switch is distributed, so the whole architecture is scalable and second feature of CPLD was very wide products terms which is not required in practice that was a lot of area. So in FPGA you have kind of lesser number of input to the combinational circuits. So essentially that is this to allow other FPGA to scale to larger you know aquatic complex size.

And we have seen the evolution of FPGA from ASIC to the standard cell to the field Programmable Gate array what is in a nutshell, what is the architecture of the FPGA and general architecture of FPGA and we have discuss the programming you know the technologies that means basically how this configuration technology what are the different type of configuration technology.

We have seen some commercial chips, commercial devices from the major vendors ok, maybe those are the only vendors as far as FPGAs are concern, so today we will choose a specific FPGA the Xilinx FPGA and look detail in which architecture and which will basically enable you to tackle understand even though the much more complex FPGAs from Xilinx itself as well as from other manufacturers.

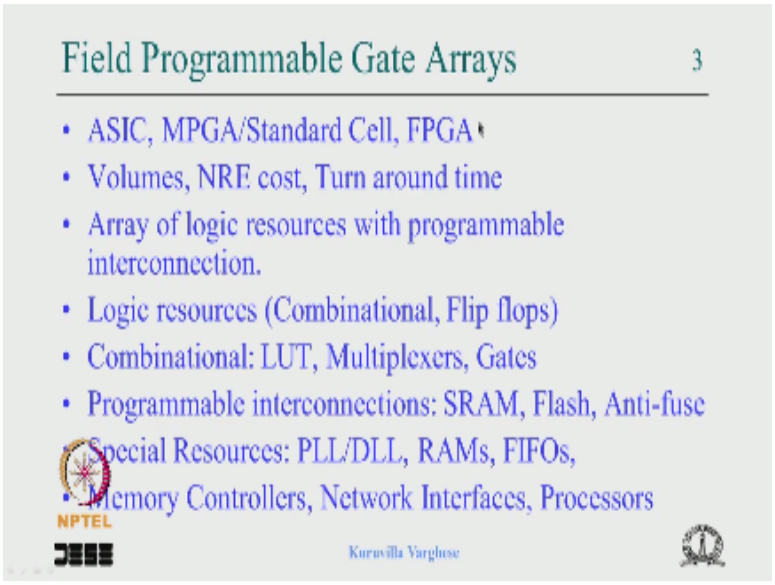
Because most of the people use somewhat which is somewhat similar architecture unless CPLD that there are variation, but I will I would highlight that variation showing some examples where were there is a kind of drastic architecture variation okay. So quickly we will have a look at the last lecture slides then we will get on with today's lecture.

(Refer Slide Time: 03:16)



So let us move on to the slide so we looked at the evolution kind of from you know very low level design to using higher level blocks and interconnecting and the interconnection made up in the foundry.

(Refer Slide Time: 03:34)



But in FPGA the interconnection is with programmability built into the gate, this Array Logic resources, so that it can be programmed or configured in the field. So the game is between the NRE course which is very high for ASCI medium for this and low for this. So this ASIC workout for huge volumes and this for middle volume and this for low volume and the design turnaround time very high for it and low for it.

This may run to year and this could run through 2 weeks depending on the complexity okay. Apart from that then what should not kind of go all about the definitely this FPGA has

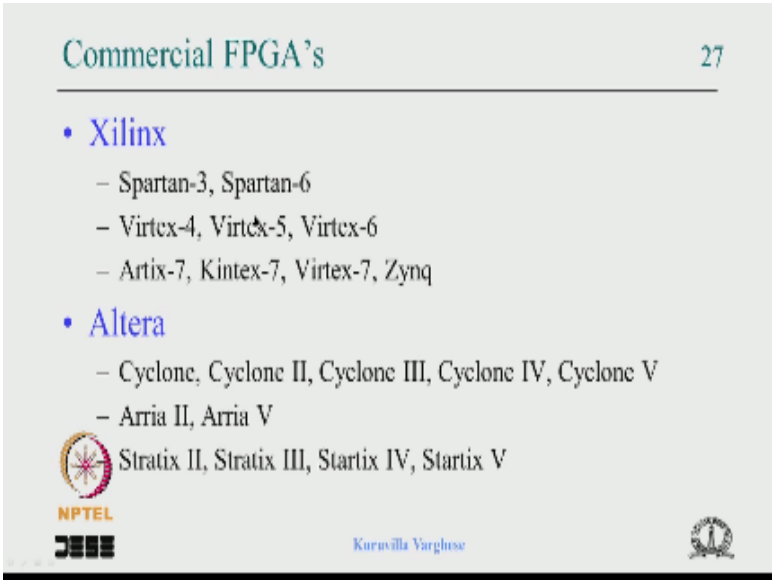
disadvantages and it is quite it can be quite a complex FPGA can be quite a bit and can be costly and one cannot assume the you know if FPGA appearing in kind of the mass produce item like cell phones and all that.

But nevertheless there is a definite role of FPGA in every digital chip design because almost all the chips are kind of implemented fast in a field Programmable Gate array then mostly ASIC almost all of them ok and FPGA as I said as an array of ok though the name suggest array of gates, but that is not true, it is array of logic resources it is higher level logic resource which is used to build combinational circuit with Programmable interconnection.

And when we say logic resources we know that we have to implement data path we have to implement controllers though it has combination circuits and flip flops, when it combinational circuit most of the FPGAs use lookup table as a combinational circuit element and some users multiplexer some uses Gates and mainly I would say these are the Actel, anti-fuse FPGA which uses the multiplexers of gate.

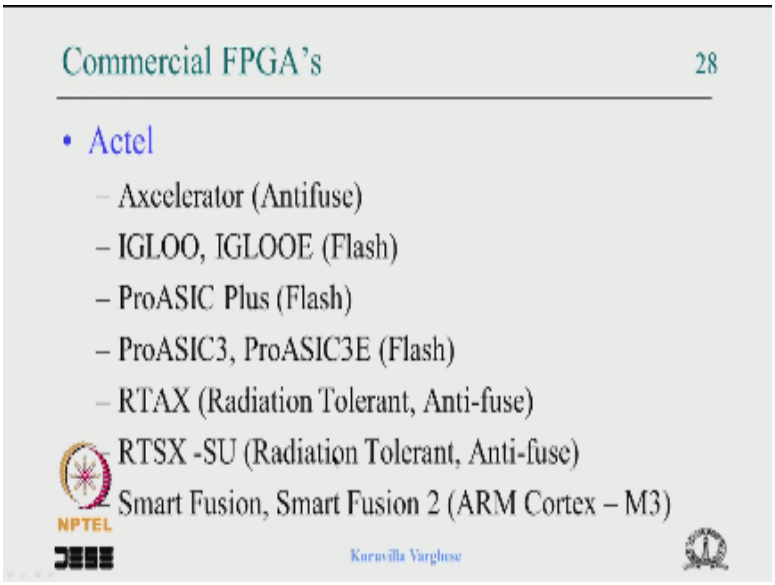
And the Programmable interconnection technologies are called SRAM and flash and Anti-fuse. There are special resources like phase locked loop, delay locked loop, Ram FIFIOs and memory controllers, network interfaces, processes and these are many times depending on the requirement is you know is part of the silicon ok. This is a hard coded devices which I am talking about ok.

(Refer Slide Time: 06:36)



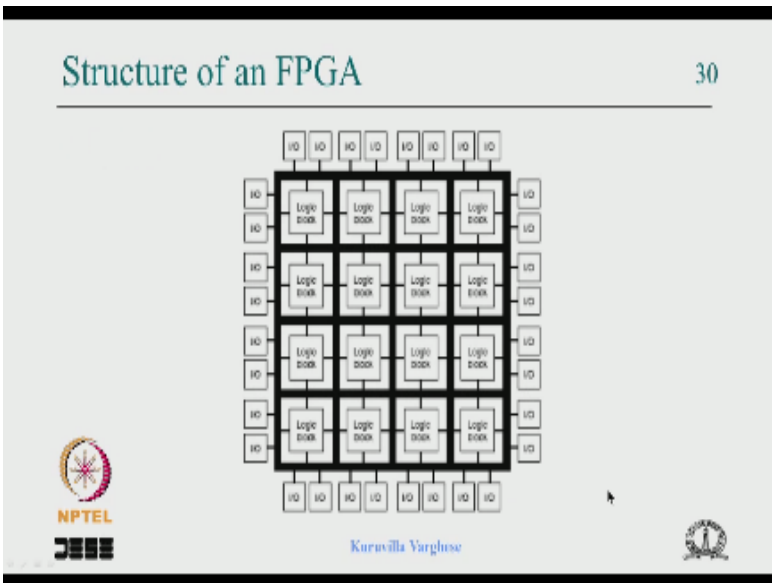
And also that is filed programme of get natural we have seen commercial FPGA Spartan and Virtex from Xilinnx and cyclone and Stratix and area from Altera.

(Refer Slide Time: 06:46)



And Actel has this ProAsIC and accelerator and radiation tolerant version of the accelerator smart fusion with Arm cortex and things like that and of course as I said the Xilinx Zynq with dual core RAM and the Altera has soft core.

(Refer Slide Time: 07:10)



And also processor, so this is general structure of FPGA, I/O pins around, array of logic blocks normally this will run into 10s and 100s okay like across and across the row and across a column, ok that this may be typical numbers will be for small 150 by 52 maybe 100 by 100 it may go all the way to huge numbers ok and that shows that you cannot interconnect those blocks with a single switch.

So there are wires laid out in a regular pattern like at the junctions are the switches and the wires run so and these left and the left and the bottom into the logic block from this wires and the top and the right of the output and this is a very general I am not talking about a specific FPGA but maybe that when you take a specific FPGA the input is from the bottom and the output is from the right hand side.

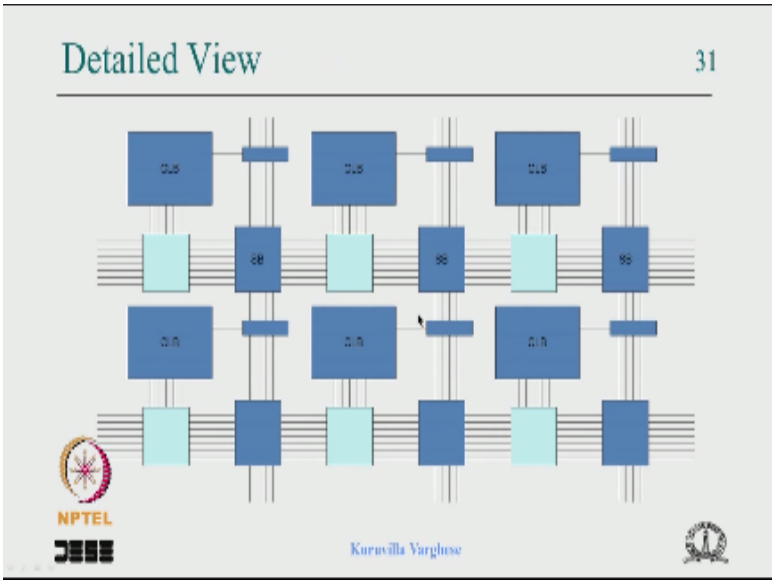
And I have not verified the chip layout it may not be available in the data sheet exactly these are kind of schematic you know when you look very exactly not look exactly like that if exactly in this fashion that nevertheless it resembles the implemented architecture. So you can imagine there are output coming there is a switch here where the output wires are connected to any of the wires here.

And you have a big switch here which allows interconnection of these vertical to bottom part or horizontal part left right and things like that. So these are switches all junction switches which allows connection. So when you put when a designer, when the tool put some hardware in to FPGA to be place across multiple logic blocks and depending on the circuits synthesize that will be interconnected using this general wires.

There are wires which span a single logic block, the wires with one maybe 2 logic blocks, 4 logic blocks, 6 and the one end to the Other ok, now these are these comes under the statistics it depends on what is inside a logic block and how many need to be interconnected, how close you know for a normal circuit you know like in an average in might be that you take a 4 by 4 kind of matrix the mostly the connection interconnection for in here.

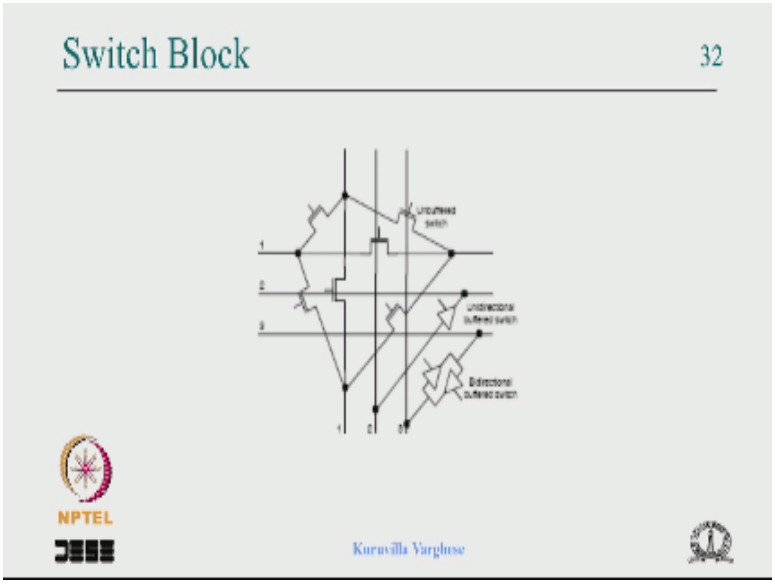
Now then somewhat you know you take this boundary then it on an average it goes all the way up to the 4 to 6 then such kind of wires are useful and as I said like when you interconnect then be lot of switches brand on the way to the interconnect delay in FPGA could be quite high compared to a CPLD, but the advantage of FPGA is a complex itself a complex circuit can be implemented.

(Refer Slide Time: 10:36)



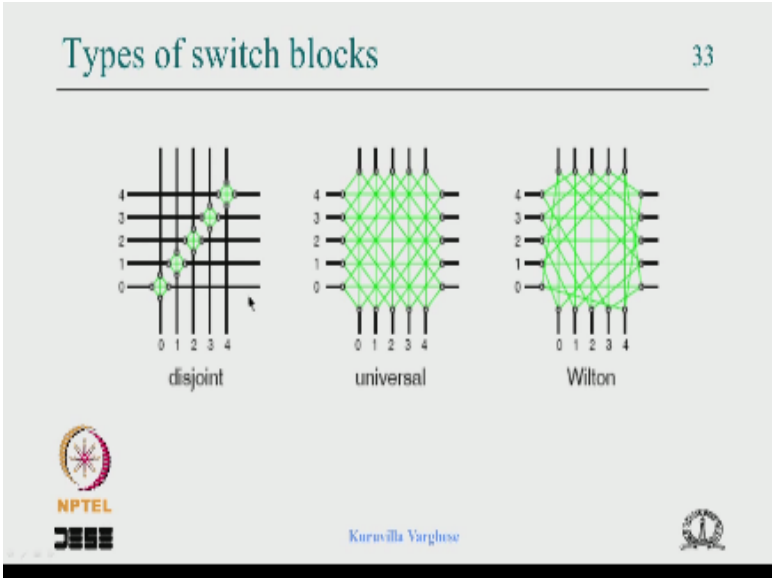
And this shows a very detailed diagram where these are the switch lock with horizontal and vertical voice of this could be connected here, here or here and you can program all that only output can be connected to any of the wires there are 5 input going to the logic block and that can be connected to any of these wires, so there are lot of program of this is configurable logic block because the logic block itself is quite big. We will discuss why they big, so it need a lot of configuration ok.

(Refer Slide Time: 11:16)



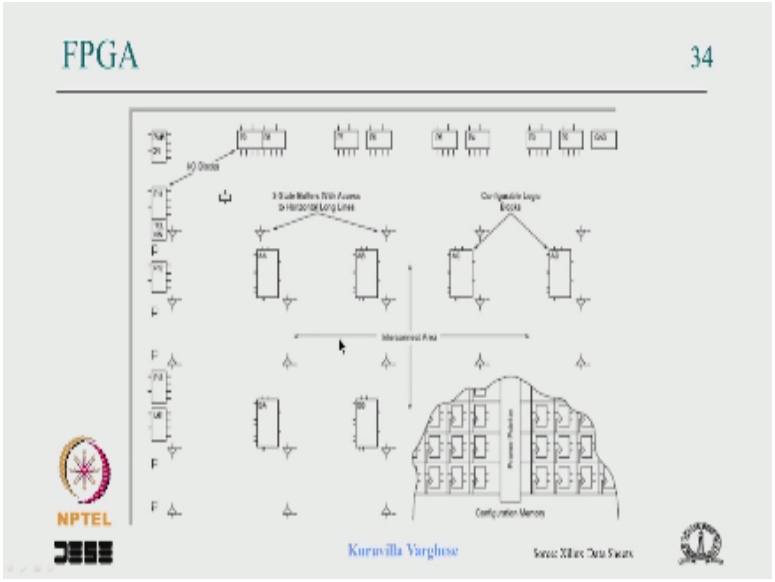
So that is shows a switch diagram like a horizontal wire, a vertical wire is connected to the horizontal you know this wire and this wire, so at least 1 wire is connect to left right and bottom and that this shows you know various different types of connection depending on the statistics of each vendors use .

(Refer Slide Time: 11:42)



Different you know type of connections maybe we will not be seriously worried because our aim is to understand the FPGA and use it in our design that are designed FPGA.

(Refer Slide Time: 11:56)



So I will skip and this shows a diagram with which from the Xilinx data sheet with very old but this kind of you know a kind of highlight a thing which we should not forget there are logic blocks which can stop combinational circuit and the flip flop there are interconnect wires, but underlying is a configuration circuitry to configure all these and so you should not forget about that.

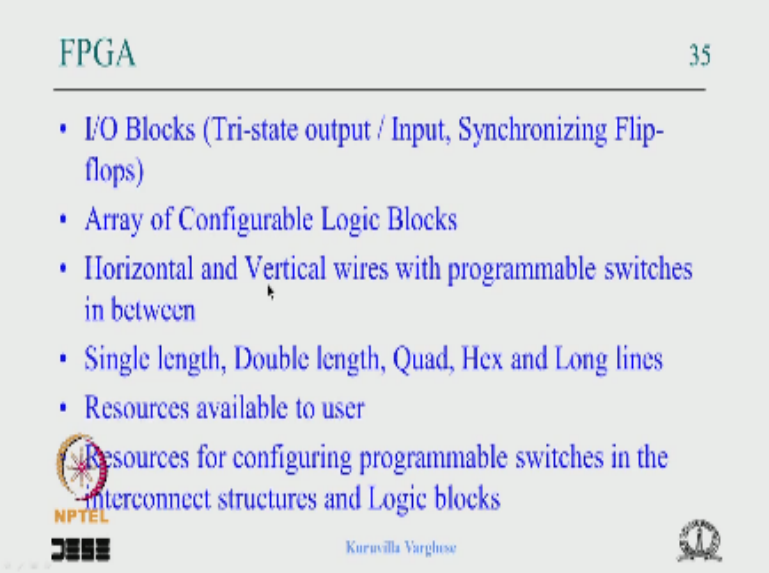
There is lot of configuration overhead in terms of address in these switches and programming at on off and all that. So that makes you know compare to a normal ASIC this makes the

FPGA quite big maybe despite more power, so the row the speed of an FPGA may not be as high as say 1d or 2 d may be the serial transceivers go at high speed because we see design.

But in general lookup table speed may not be that high, so any reasonable circuit you place and route it may not clock in a 1 gig maybe 200 megahertz, 300 megahertz sometime it is too complex if you are not opening it properly it might even go down up to hundred, so but then it is not a processor so you can it is up to the design up to exploit the available resources.

And you know do parallel computing you know pipelining and so on and kind of implement a very high throughput design to exploit this available sources ok, that is how the FPGA can achieve performances.

(Refer Slide Time: 14:02)



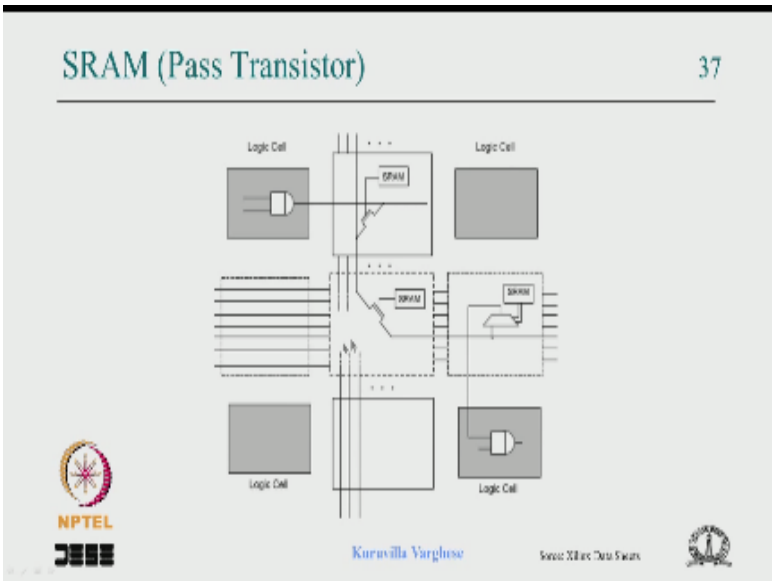
So FPGA comes I/O blocks and that has tri state output and input and it has synchronizing flip flop because the input can come from another chip for another clock domain and array of configurable logic blocks, the horizontal vertical wires with program which is in between these wires are single end, double end, quad, hex and long lines, it depends on as a statistic or how statics the circuit of the connection.

And there are resources available to the user in terms of logic clock, the memory, the PLL and all that and that are the sources for configuring the Programmable switches in the interconnect structures and logic blocks ok. We will see what is there to configure in a logic block.

(Refer Slide Time: 14:59)

And this is where we have stopped, the Programmable Technologies are SRAM flash and Anti-fuse.

(Refer Slide Time: 15:06)



And we said that the SRAM basically used a transistor and NMOS transistor for interconnecting the two wires but then at the gate is like NMOS, the gate is 1 it makes a connection gate is 0, it cut off the connection ok, now the gate is a flip flop which holds that that status of the connection if the flip flop is set to 1 then it conducts if it is 0 it does not conduct.

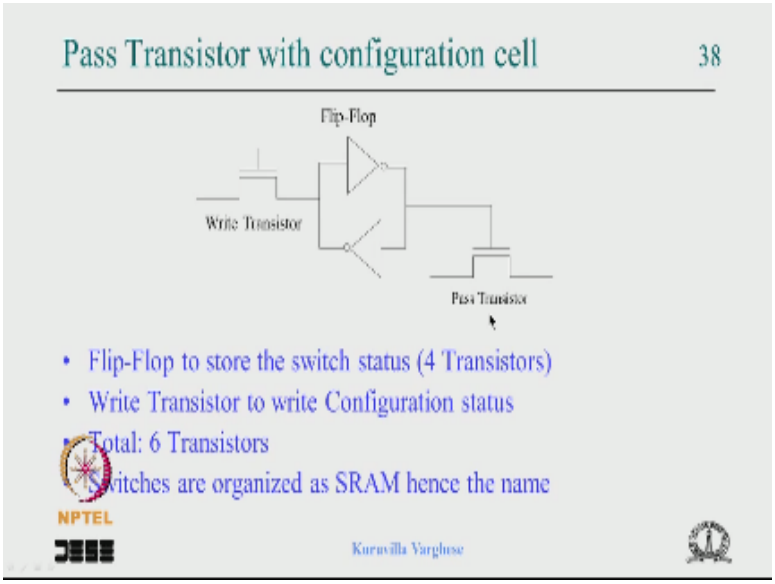
So that means that these are flip flops, in all these of flip flops which need to be programmed at the power on because the power on there no specific state and now to the program allow it like it is too much of an overhead too kind of independently address and you know program

it. So you do not program it is a really you program parallel just flip flops are organised as static RAM of with 8 or 16 bit.

And the vendors knows the kind of the position to decide on the format of the configuration file and that is why it is called as SRAM ok. When you say the programming Technologies SRAM nowhere SRAM can interconnect something the past answer is interconnect Technologies the flip flops are used to support the status of the gate and flip flops are you know organised as static RAM is a programming.

So in the previous families of FPGA it was 8 bit wide and the current FPGA with 16 bit wide programming and that I am telling the kind of the row internal structure but then there is a way to program CLE as well as useless confirm we will we will see that ok.

(Refer Slide Time: 17:06)



So but one issue with the SRAM is that this is a simple flip flop ok. So you look at this whole block ok. Now I hope you remember that this is a logic cell, this is a logical cell, these 4 are logics and these are those are the horizontal wires, these are the vertical wires to get whole interconnection at this big switches and output interconnecting to the to the vertical wire ok, just an explanation on the diagram.

Now we are looking at this way of doing it up this one, so we have a right transistor and we have a kind of switch with transistor and a flip flop ok. So that is what is shown here your way switch the gate of that was connected to a flip flop nothing but a cross connected

invertors ok, so now you have two forces invertor to 1 or 0 with this is a right transistor suppose you want to force this to 0 then you what you do is that you put a one here.

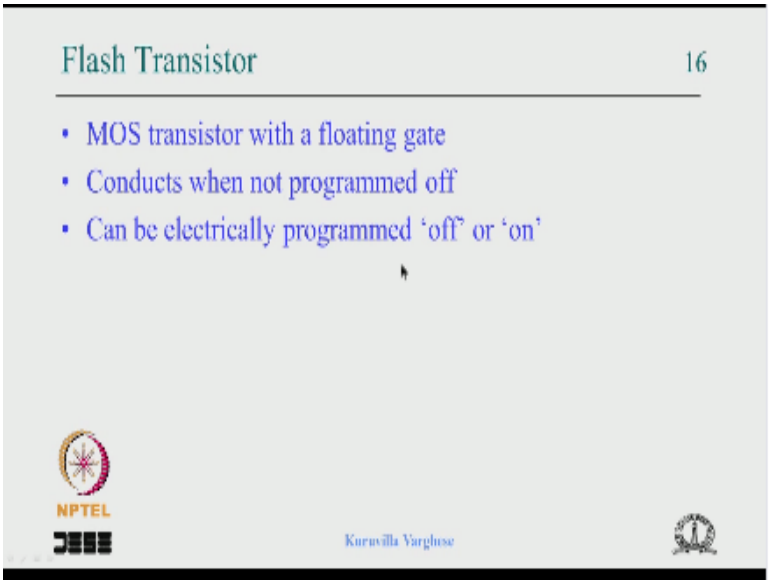
Enable the right transistor want to want this becomes 0 and 0 come here it is latched and that can be removed that means you give up on you give and past answer is owner of depending on what you give you ok only thing is that you give the opposite of what you want as far as the structure of the important thing to remember is that the flip flop are made of 2 inverters and 1 inverter is made of two transistor one NMOS and 1 PMOS, PMOS to VDD NMOS to the ground.

So that this invertor is composed of two transistors, this 2 and this one is you know the 2+2 5 and 6 and this can be sometime right answer with isolation can be big. So one switch means 6 transistor ok. So you can imagine it occupies quite a lot of area. So if you assume say here it is shown 7 wires, so it is a 7*4 28 wires, even if one wire is connected to 3 know you like you can say 28.

See 2 kind of connections are available for it to connect to everything else. So it is a quite 28*27 by two connections maybe there. So and that involves lot of current, so switches occupies quite a lot of area. So that is one kind of disadvantage of the SRAM kind of technology that this is a quite lot of area and of course there is a delay because once you there is a delay of the channel ok.

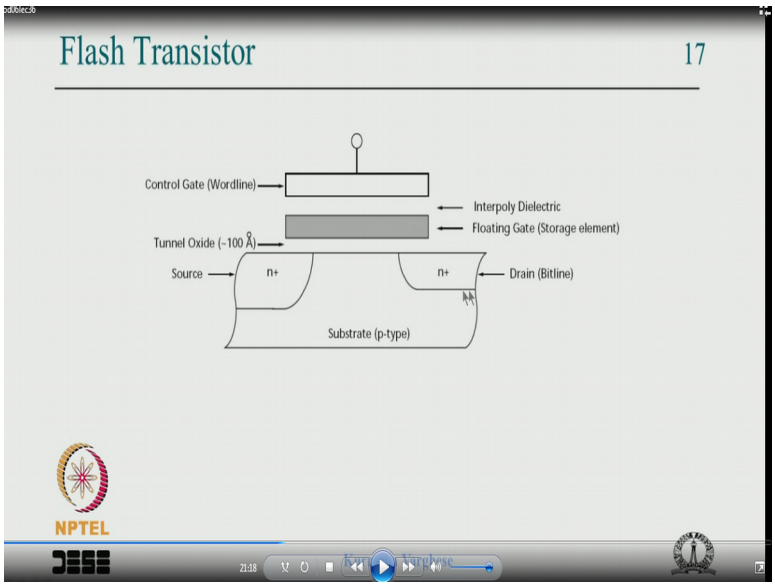
So it introduces each switch introduce a certain delay, so when you cascade multiple switches from source to destination attached to the delay and it occupies a lot of area and these as I said this whole thing is organised as a slump sale sign of certain word hence it is called just having technology.

(Refer Slide Time: 20:46)



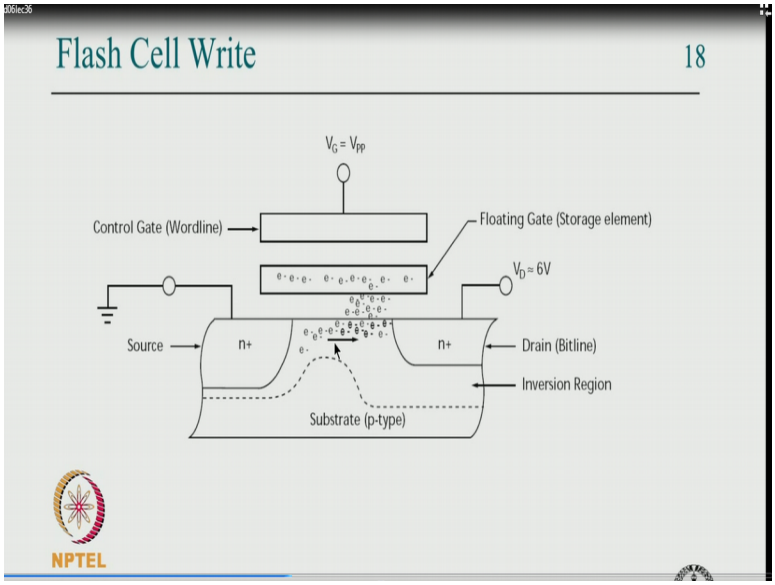
And when it comes to MOS transistor we have seen that with respect to the PLDS, CPLDs and MOS transistor has a it is a normal MOS transistor with the floating gate it conducts when it is not programmed of and can be electrically program off or on.

(Refer Slide Time: 21:07)



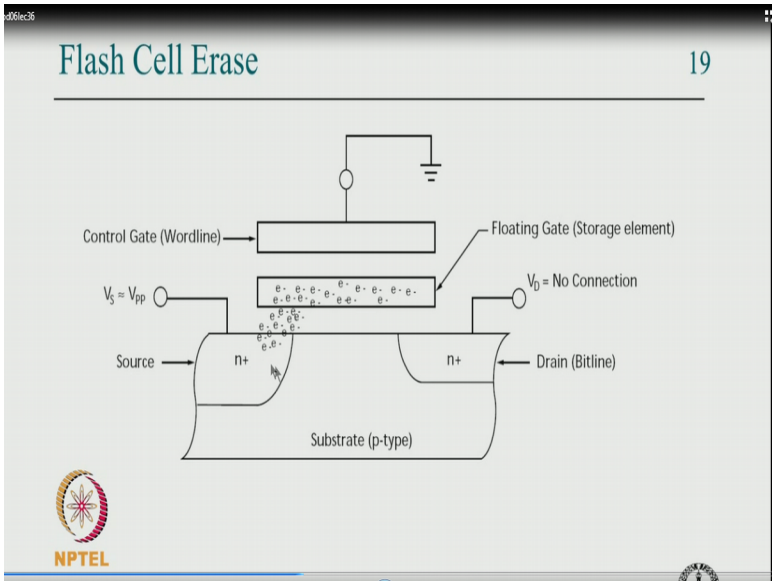
We have seen the S basic working of it, so you have an you know n channel transistor NMOS transistor with Nsource and drain P type substrate we have seen silicon oxide a poly silicon gate then again Silicon oxide the controlling Gate. Now normal case apply the check voltage channel forms and conduct but the trick is to trap some electrons to make the threshold voltage or so if you apply normal threshold voltage because electrons are sitting here and kind of you need a higher feel to form the channel ok.

(Refer Slide Time: 21:59)



So that makes it off you know you apply the control gate with logic high it does not conduct, so that is how it is turned off so that is shown here, the source is grounded drain is given the supply voltage pulse is applied a higher voltage power supply electrons and getting here and then the normal application of the gate voltage would not make the channel and it would not connect.

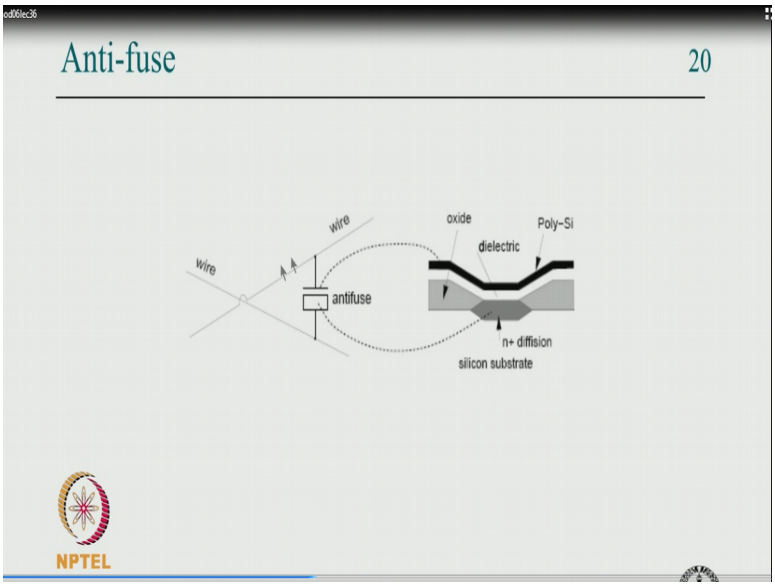
(Refer Slide Time: 22:16)



And when you want to erase it you do the opposite ground at the gate you ground apply false here gets out. So one advantage of the flash is that it can be programmed on or off electrically and so much more than that is the chip has a port for doing that within the circuit you can program it. So somebody makes an FPGA or CPLD with that technology on the board on the other destination it can be kind of it can be updated.

Nowadays more and more kind of over the network update or sometime is called over the air update wireless channel is updated and maybe even the hardware many times can be you know the updated like if you have a Programmable hardware not only the flash memory is a Programmable hardware also updated remotely through the network wireless and all that. So these technologies enable that.

(Refer Slide Time: 23:23)

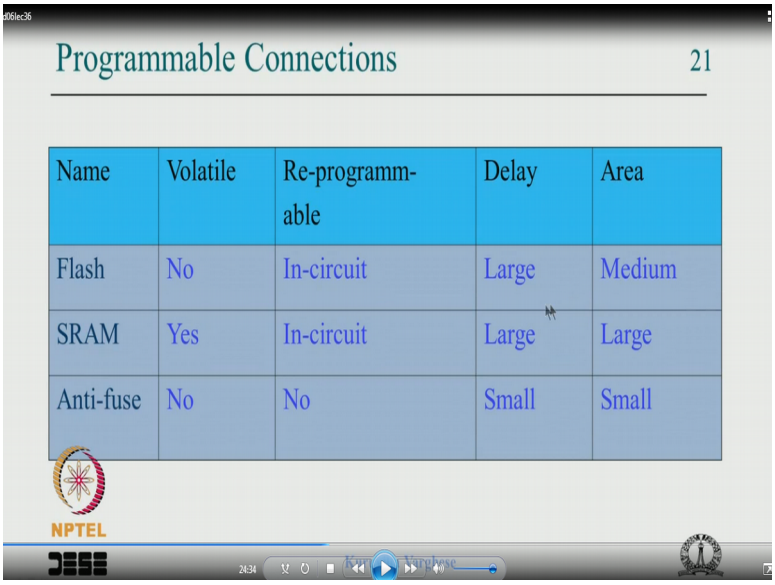


And the third one is call anti-fuse or anti-fuse and you know that in a chip design you have layers which implements transistor and your have layers with interconnecting wires ok. So you cannot have the wires in the same layer of the transistor. So there on a different layer and the layer to layer connection is by drilling a hole and normally on one layer the wire run horizontal in the next layer.

The wires will run vertically ok because you know it is easy to interconnect also the kind of the like the coupling between the two tracks will be less capacitance between the facts will be less, so essentially this is assume that is a horizontal why does the vertical via normally there at the cross coil will be a hold drill and then be conducting it is made conducting ok.

So in an anti-fuse FPGAs is the connection is made through this whole initially the whole would not be conducting there will be a fuse material as special fuse material deposited.

(Refer Slide Time: 24:33)



It show the cross section of that and normal it is not conducting this shows that kind of the horizontal wire on the top layer and this shows a vertical wire which is kind of coming out of the screen towards you. So there is a kind of hole with a proper deposit and you apply a high voltage when I say high voltage like it kind of nearing 10 volt and then its uses and make a connection okay which is permanent which cannot be kind of erase ok.

So that are certain advantage, so it does not kind of a noise would not reverse the status of connection when this program. So these kind of technologies are used in the space application mostly, so that anything send outer spaces exposed lot of radiation and this can kind of flip flops and that can cause problems so that and if use and it is not very kind of little bit you say resistant to the reverse engineering.

If somebody tried to kind of feel that you up and take the photograph of the mask, then one cannot make out the hold is kind of programmed on or off so that is an additional protection from kind of piracy the hardware piracy as far as concerned but in SRAM kind of FPGAs you know that these status of this flip flop has to be stored somewhere not the power on it has program and if somebody can read those memories then this design can be pirated.

But nowadays there are in description technologies which is available to protect the configuration file which is stored in a memory ok. But this as an additional advantage of that the summary is shown here, so in a flash kind of technology it is non-volatile. Once you program it till you erase it retails good that is way it is reprogrammable in circuit and the delay is quite large area is medium.

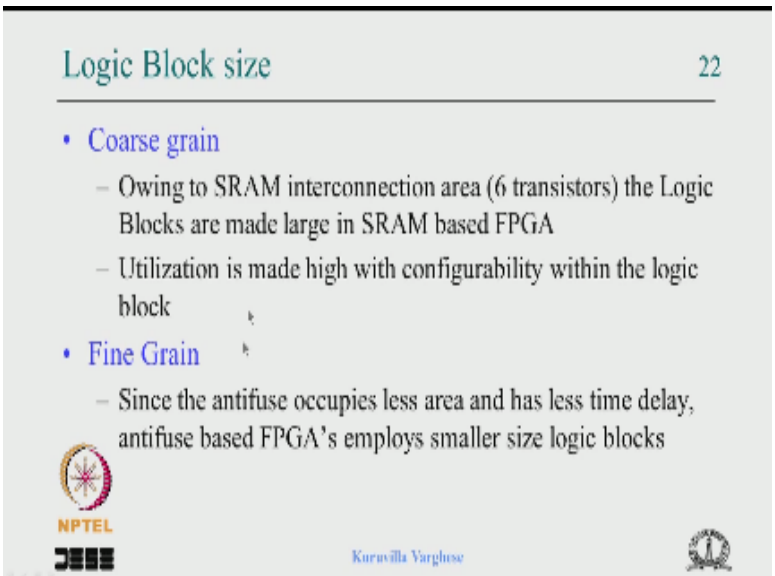
But when you come to SRAM it is volatile that means that the power on, i just reprogram, but then you know that the flash has some kind of limitation on the number of times you program it is not that kind of it is a serious limitation of a FPGAs concerned is not that when you design with FPGA program also for day because it take time to design.

And the design is you do lot of simulation and verify everything before going to the to the chip programmer chip, but still that SRAM has no such a great limitation that number of times it can be program. So this is very good for prototyping and particularly I would say coming from academic institution is very good to give to the student you know. They can program reprogram and it run for quite a long.

You know you can use it for 3 years, 4 years without trouble with this is also in insecure reprogrammable size large area is large we have seen the large area when you say anti-fuse it is non-volatile one time it is not reprogram but you see the delay small because it is not to a you know active circuit like transistor connection which is refused connection and the area is no expiry area like there any way there interconnection of the wires require .

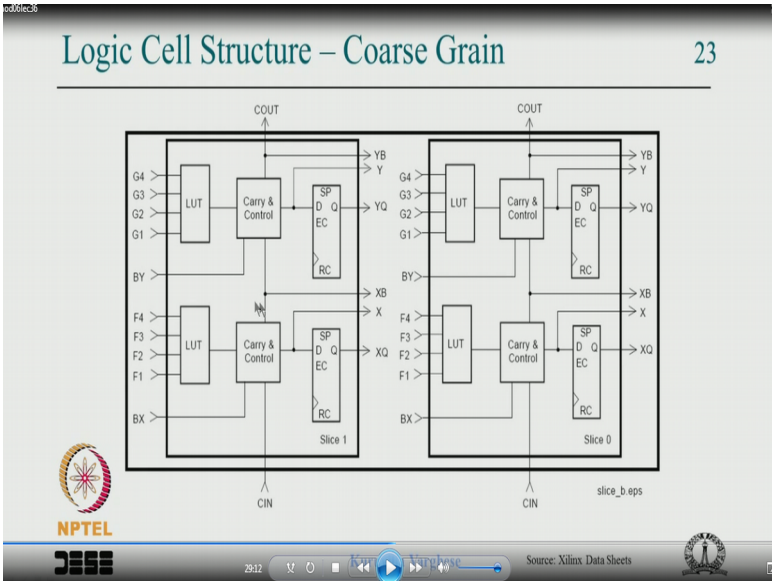
The wires which is called wire via so whole and so this deposit in that ways, so I have no idea but definitely to apply the voltage you need a transistor to isolate that all types of definitely there has to be right transistor at that place but still it is not much area. Now this has some effect, this program technology has some effect on the size of the logic clock ok.

(Refer Slide Time: 29:06)



So this is one important things though nowadays most the complex FPGAs are SRAM.

(Refer Slide Time: 29:11)



So this kind are forgotten but then one should realise that when you take to say anti-fuse based FPGA since the area of the interconnection is very less they are able to make the logic clocks more ok, so this something to do with the kind of fragmentation like you would have seen in a hard disk say that are sector okay. Nowadays you have I know that terabytes or gigabytes of you know the space on hard disk.

But there are basic unit you know that hard disk can accommodate but why the storage unit storage and earlier the sector was 512 bytes but now it is kind of 488 depending on the size of the total size of the hard disk. Now advantage of using small size is that less space is wasted supposed the sector size is a 4K ok, suppose nowadays it the other files have less than 4K size ok supposed 60% of the file stored in hard disk is less than 4K.

In lot of area is wasted, so that we can say there is fragmentation or unused area in each sector. So this choosing the size of the sector is on the average size of the file. So it is ideal if you can choose it very small enough, so that nothing is wasted you know like if it is made 512 bytes are the maximum weight should be 512 bytes maybe there is a file which is very small.

So only 512 is pasted but then in the case of in the 4K can be wasted, you know very small file or there is a a file which is just little about the 4K, so that that might effects that I am talking about the fragmentation same are coming back to the slide same can apply here, you

know in a fine grained the fragmentation will be less but that is possible because the interconnection area is small.

But when using FPGA with SRAM which occupies lot of area if you make the logic block very small to avoid the kind of wastage within the logic block then the switchers will be much bigger than the logic clock ok. So what this kind of FPGA vendor do is that they make the logic block quite big because interconnection is costly. So they make sure that quite a lot of things pack into the FPGA.

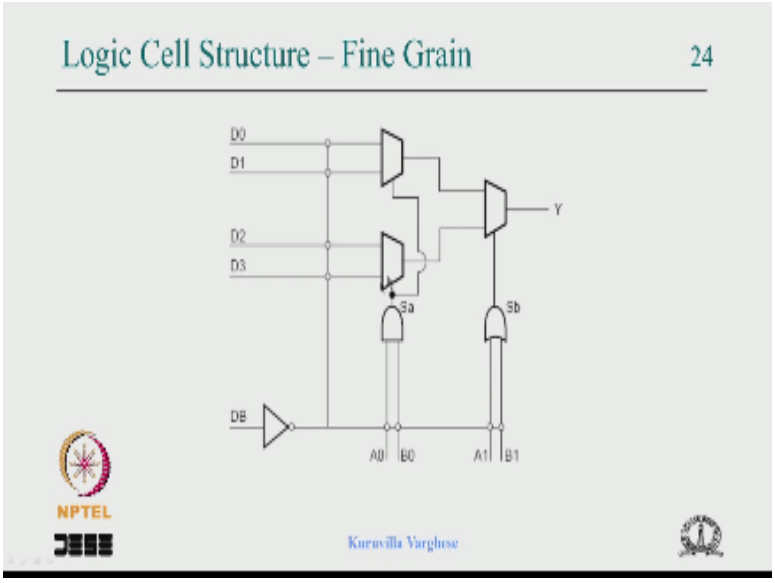
But then there is a great danger that only part of it is used, so the this kind of FPGA which is called coarse grain FPGA and the second one is fine grain. The coarse grain FPGA the vendor make sure that the logic blocks are very flexible that means that they make sure that everything can be used, suppose you are using a part of logic block other part can be used say we have seen the CPLD .

In a CPLD the AND and OR section output goes to a flip flop ok or in PLD like 22 metre, you can bypass flip flops you know no problem ok. So but if you bypass a flip flop that flip flop cannot be used separately ok. So that is a waste, but we will we will see soon see that in an FPGA the FPGA vendors make sure that like if the combination logic alone is used in a particular logic block the flip flop can be separately uses something else.

You know these kind of things are available that is how they exploit the kind of disadvantage of the large size of the static RAM and most I can say their successful in doing that they not bow down by that should do there is a division does not great disadvantage of the whole screen, kind of architecture ok that is what I want to highlight. I just show a picture this is a coarse grain FPGA.

Basically Xilinx FPGA you can see that it is a logic block configurable logic block there are two identical slices, they called slice and within a slice there are identical blocks, 2 blocks and there is a lookup table for input and flip flop. So a logic block has 4 lookup table and 4 flip flop ok and this look up table itself is 4 input which allows you to implement 4 input variable implementation. Please quit this huge drain and we will see this in detail later.

(Refer Slide Time: 34:31)

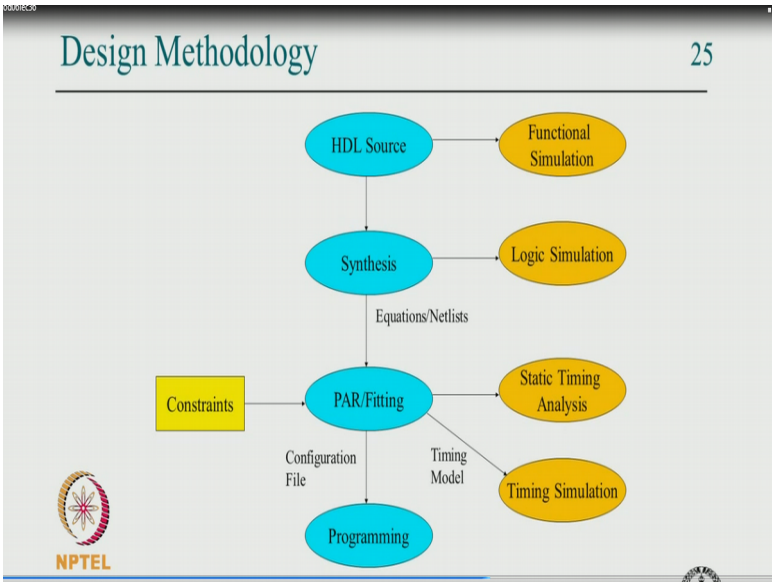


And if you take this is a taken from a Altera, so Actel FPGA, Actel 54X ok I have shown that also this is taken from the Actel data sheet and the 54x FPGA, so you see the essential combination circuit is a 4 to 1 multiplexer ok. You know that the multiplexer is nothing but and also cute and if you want implement Boolean function as possible say you connect a variable a here b here.

Then this line represent a bar, b bar, ab bar, a bar b, so a bar b ab bar and ab ok. So depending on like you want have an ab bar or a bar b it say you 2,0,1,1,0 when you get that implements. So this is a very simple but in this case there is an AND gate with two input OR gate with you know 2 input for normal if you want to then you connect want to the logic 1 other users variable, but in some cases like you give abcd.

Then you know some combination of product terms of 5 variables are possible and you also remember that if you connect and a b and instead of giving 1,0 here, if you connect a third variable here like ec bar and so on, you could make three variable elimination, so with you know like a one variable here and four here maybe some combination of 5 variable can be implemented, otherwise 3 variable implementation can be done. So that is fine grain you know logic cell of an FPGA.

(Refer Slide Time: 36:19)



So design methodology we have discussed this during the VHDL lecture but then if you have not gone through that lecture for some reason you are looking at FPGA part of the lectures then we will just, let us say that so the design methodology is currently you start with a hardware description language like VHDL for Verilog you describe your circuit in VHDL or Verilog after that you stimulate that is called functional simulation.

Sometime it call as behaviour simulation, these are just mean some time people say the behaviour simulation is nothing but functional simulation plus the timing details but never the less I mean by functional simulation, the basic code is simulator ok. Now that is not the circuit when you say, when you say y is assigned A and B we are not simulating the AND gate, just simulating that AND function ok.

So there is no circuit, this is a basic source code which is very fast because if you make Syntax errors we are you make logic error that can be quickly corrected at this point before you know generating the source, that is that functional simulation, then you go to a step call synthesis where in this the source code your return is synthesized in case of logic gates and flip flops maybe Boolean equation in the keys of the combination circuit.

But it is a logic circuit is generated out of your description, now if you are no receive the designers and or was he or she can go for a logic simulation that is different from this function simulation because we have your simulating the logic which is generated by the synthesis, here you know stimulating the code your return the code in human readable English that that syntax is made here, but here the logical stimulator.

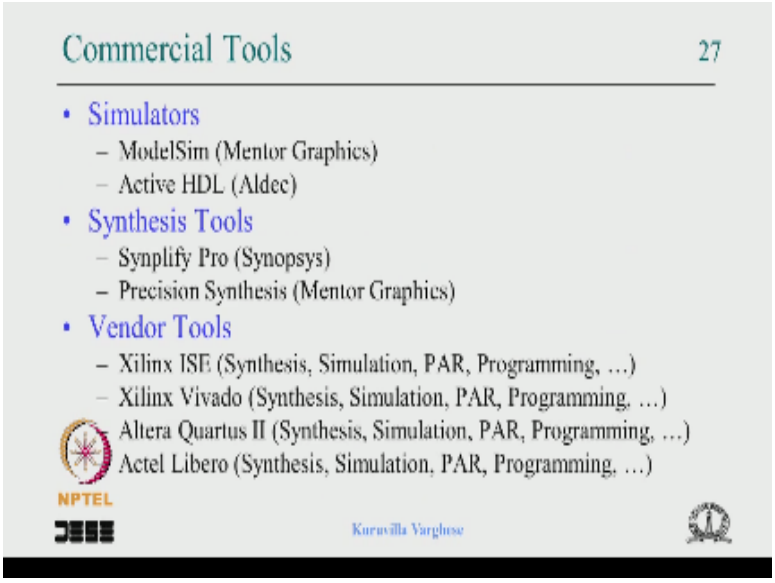
But mind you at this point here there are not delays because this is not yet implemented in a device, so there is no delay at this point, so you give to an AND gate A and B at 10 nanosecond at the y which is output come out at the 10 nanosecond ok no logic delay or it connect, so there is no delay here, but mind you here you simulating the code here you stimulating the circuit without delay.

And maybe you can which stimulate that if you find errors if there is synthesis errors then you can go back and if you look at here you quiet here but for export designer low coding which I have taught you in the keys of VHDL knowing a particular circuit how to write the proper VHDL code. So that you get what you want, you know then in that case you really do not required this logic simulation.

And then you go to a step call place and route that is basically like going back to this diagram maybe like you have maybe yes here, the synthesis tool will generate a logic circuit, now the FPGA has an array of logic blocks, now that circuit and these are identical circuit has to be placed in the logic block and has to be interconnected and that is not a very easy task.

Because we take a logic 1 logic block the first logic some circuit part of the circuit like if there are kind of 10000 or 100000 logic block you could place anywhere without any constrain, if there is no constrain the possibilities are too high so the place that is another placing the circuit in logic block and interconnecting it ok that what routing, so that is what it says.

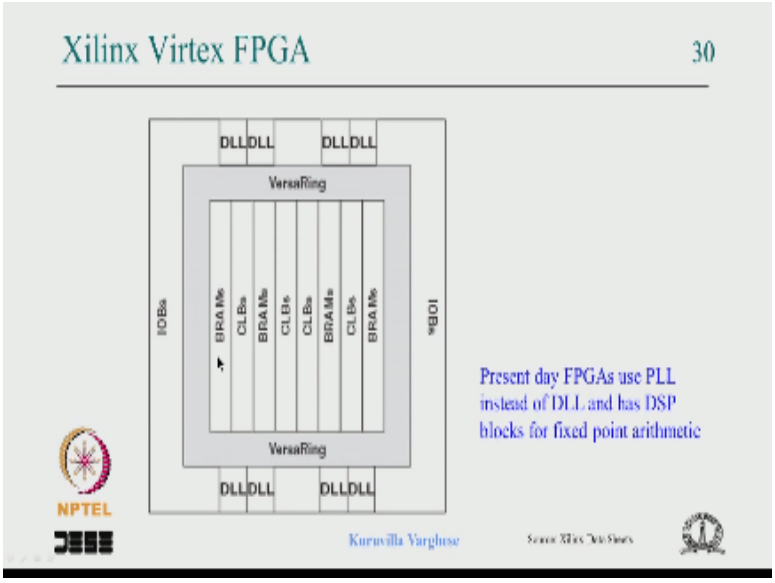
(Refer Slide Time: 40:46)



So this is what the place and route and at that point you know like the FPGA is a kind of general purpose circuit the I/O pins you know you are user input output signal can be assigned to any I/O pin, so you can specify the I/O constraint saying that you know please map you know map this particular signal to once pin because maybe one advantage of FPG A is that even before designing the chip you can start with the PC design like it you can assign the pin of the chip to a particular signal.

And you can go ahead with the PCB design while you are designing the chip okay ultimately some kind of printed circuit board has to be used to mount is a FPGA. So that can go parallel, so these I/O pin assign is specified in the constraint and another thing which can be specified is that as I said the placement is a very critical thing to do complex thing to do suppose you want performance you want to like you design a counter with the flip flops and next state logic.

(Refer Slide Time: 42:16)



And your planning to call the contract say 300 megahertz then if the counter flip flops are kind of place a part in the chip you know like you know the chip ok that she will come back to the chief diagram ok here. So suppose a part of the counter is here and other part is here, so for the counter it has to be interconnected, so it may like it incur lot of delay, it will be good if their place close together.

So that the interconnection would not suffer lot of delay, so that you will get a fast counter, so that kind of constraint call timing constraints can be specified at this point which say that for a combination circuit from an input and to a particular output the delay should be less than certain you know time like 5 nanosecond of 200 megahertz or you can say from my registered to through a combination circuit to that register which include all the delay ECQ, tcombinational, t-viabilit,y switch delay and the setup time memory.

It should be less than 5 nanosecond then place and tool will make sure that the circuits concerned a place together in close together and the number of interconnected switches are lost and so on ok. So it can quite over that, so when you say cannot find you know you have the timing constraints, the I/O constrain all that is there, then the place and route tool is called PAR tool you know place in routing.

Many a times in case of PLDs is call fitting ok, so basically it is just a term used you could in principle used a template out there also but then you know that there is only there is no variation, there is no grade, there is simple switch between logic and so it is basically you know how to fit the given circuit into the into the logic block is a main constrain, so that is a

call fitting and once you do that then at this point for the tool all the delays are not, in the all the combination circuit delay all the wires delay, all the register delay, all I/O delay everything is known.

So it timing model is generated, now you can simulate the you are synthesized and place and route circuit with time delays and given input output come with the delay you know the time simulation you should know that this are different, this is the original VHDL code you have written, you do the functions in this, this is the synthesise code which itself may be written in VHDL which is simulated.

So the language maybe VHDL, but that is not the code you have written and when it comes here still can be a VHDL code many times Verilog is used but with for some reason the timing model with time delay is this can be you know in the same language totally different code that is timings related but then it takes a long time, one thing do understand is that you give the import then the stimulator has to now calculate all the time delays which is composed of lot of you know individual time delay and it is not a gross.

So it is the time of the I/O, I/O delay interconnect delay, wire delays, and look up table to combinational circuit delay various components circuit components, everything had to delay, there is quite a lot of the computation delay this can be very timing simulation can be very time consuming ok. So for a complex circuit in might take a day to stimulate that and so the beginning start when you do first equation of the place and route if you go for timing simulation and find you take 6 hours delay and find a small error.

Then come back and correct is a very hot thing to do, so there is something Timing Analysis which is Static Timing Analysis done. So this dynamic in the sense that you give the input you literally I know make the circuit work from the input output but here what is that is that this tool will look at the block delay interconnect delay and estimate the time from input to output of a combinational circuit or for register to register for a data path or sequence ok.

But the problem is that there are no inputs, ok no inputs are specified, so the state machine or the controllers are not active what is available is setup registers, combination of registers and it had support all possible paths and show of the delay ok. Now if my mistake you know it

might apart from a register to another register which is never used in the real circuit real life ok. So because you know that the controller is the one which is controlling the data path.

Maybe there are 4 source registers, some combination circuit and say 5 destination register, the real operation of the circuit only involve a like a kind of one to one mapping maybe one register result goes to the other register for of them and maybe one of the register give the result to the 2 register you know some time to 1 or something to other but if given to a Static Timing Analysis tool it does not know all that.

Because it has no idea of the signal kind of signal status ok, control signal status so what it assume is that 4 show's 5 destinations total possible paths are 20, but real life is only 5 and report all the timing from source to destination of all the to the path and you might as a designer you might find that I know 6 of them is violating the timing control you put ok, but the none of it may be used in the real life.

So it can be kind of MS LED in this gain of which is very important that when one do static analysis such information is given to the tools saying that what are the paths valid okay, like that could be another situation where there is a source register and unregister the destination is not clock you know kind of every path like the source get an input data but the result get latch of to the clock cycle.

But this that is enabled through the enable of the destination register through state machine or a controller which the Timing Analysis tool or it knows, so will assume that all the registers are working all the time now up on every clock edge, so will assume you know it over estimate the critical path delay and we assume this has to be told to the Timing Analysis tool.

Then it be very far so after the place and route one do the Static Timing Analysis if that is not met you it right back now maybe you go back in the constrain you do certain change in the synthesis options whatever is the correct option or even go back and modify the circuit and once everything is done you do the timing simulation again go back maybe this way up to here, up to here.

If you have you tried it when everything is done when the designer satisfy a configuration file is generated that you can program the chip okay. So that is what is in there can be more tools

more advanced tool but at least there minimum tools required for FPGA design is that you have a editor which can be internal external a synthesis tool the place and route tool are fitting., a programming, a constrain editor.

A simulator which can do the functional simulation logic simulation and timing simulation normally any simulator will do support all three and a Static Timing Analysis tool and so many times vendors give everything put together as a single kind of integrated design environment so I need the science call I integrated system and one and all that ISE and some time some of these tools like there are vendors give very good synthesis tool ok.

And the vendors allow you to use that in the tool along with their you know tools can be replace the synthesis tool which is supplied by the vendor and replace it with a third party is synthesis tool and so on, but normally the place and route is done by the vendor because the vendor knows the FPGA internal details which is very much required for properly doing that.

And many times did not voice to give out this detail because people to make out what is really inside though we have the data sheet all the details are not exposed for you know keep the intellectual property, so that is the design methodology and if you look at the commercial tool just give a favour and this is record in 2014 January I am not sure that after 2 years if you listen to this lecture these tools will remain order some company, some vendors are given in the bracket you know that maybe one day required by somebody else.

I have no guarantee but also now as on 2014 January you have the model sim simulator from mentor graphics which is very good simulator um you have active HDL simulator from Aldec, these are good simulator but vendors are their own simulators and synthesis tool we have simply Pro from synopsis very good synthesis tool precisions synthesis from mentor graphics.

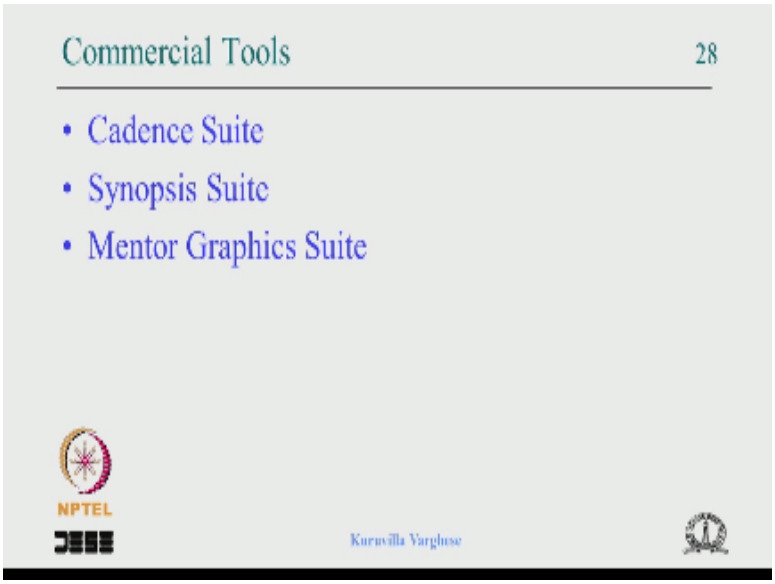
But when it comes to vendors you know the Xilinx Is or ISE it has everything, you know you have simulation PAR programming, Static Timing Analysis, constrained editor, Power Analysis, the floor planning you know that everything is there and these I tools work up to the chips of the family Spartan 6 and Vertex 6, but for 7 day have a different tool much more kind of better tool so we got very good tool a Quantum leap in the in the whole tool Technology.

The quiet old but this time you and good but it is only now currently supporting the vertex 7 series maybe as a yes come by this may disappear and this alone will there will be there and Altera has Quartus II once again that has everything what you mean whatever the Xilinx tools are there everything is there, Actel Libero used again saying they have everything required for kind of for FPGA design their family of FPGAs.

Now if you are a normal I know is a designer and you are starting with small kind of project then what you need is only this but if you are doing very complex thing and you are able to you want very high performance, you want very high tight fitting, kind of production power, I know the you want to refinement in the area of the implementation then you need to use these kind of things.

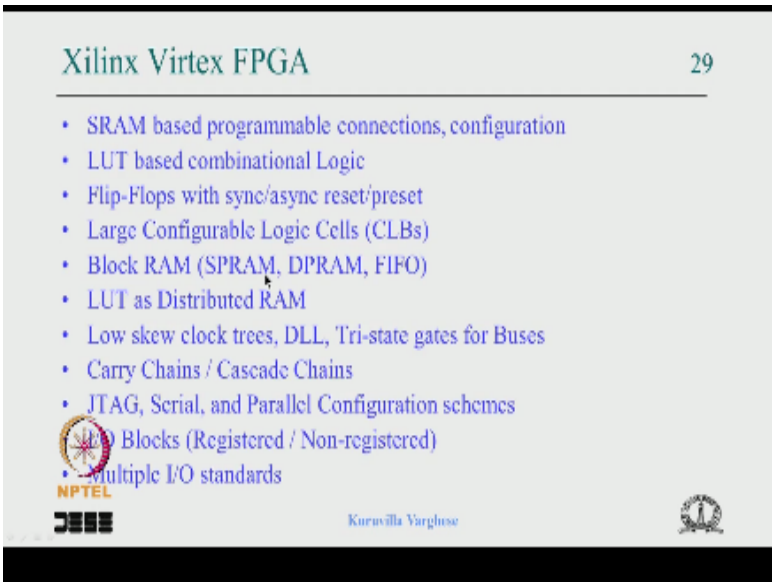
And these tools are this vendor tools are improving compared to 5 years back the current really very good you do not need to properly use, the third party two but that is up to you to make a judgement in a what you do is that you value it like if you have a really if you feel that you are not getting the performance you want that done everything under the earth weigh get the performance.

(Refer Slide Time: 55:54)



But you need lot of performance then you can probably move from the vendor without synthesis tool to get the performance out of it so of course you know when coming to the other tools we are the major VLSI design as you know you have the Cadence suite, synopsis and mentor are this has basically the synthesis tool and the simulator very good simulator.

(Refer Slide Time: 56:10)



But then you have to definitely do the place and route with the vendor tool, u know this also support so maybe I think I thought of going for the but then it is good that you know a good introduction gives you a good grip on the on the subject we will see the internal details of the Xilinx what is FPGA and I am taking that as an example. So what we have seen today is a Programmable interconnect Technologies SRam, the flash and anti-fuse a kind of features, what is the advantage, disadvantage.

And we have seen how this forces a coarse grain and fine-grained kind of architecture and how the coarse grain people at the vendors make sure that nothing is wasted and we have looked at the design methodology what are the steps the tool floor in design FPGAs, we have seen commercial tool. Now I think I have touched up on all the all the introductory material to get a very good grip.

And we will go details and will take an architecture vortex architecture which is not used at all, but it is a very good starting point once you understand that any other complex FPGA architecture can be understood very nicely. So we will go into details very very much in the details of the vortex FPGA architecture. So please now on your part you can go through my slide, you can also read the data sheets of the vendor, the application notes from the vendor.

There are lot of material which is available, maybe the textbook are not the ideal place to learn FPGA from very you do not waste your time on most of the text book I have in frankly

followed up but then you can use law of the data sheets and so on. So we will see in the next lecture the details of the FPGA, I wish you all the best and thank you.