Digital VLSI System Design

Prof. Dr. S. Ramachandran

Department of Electrical Engineering

Indian Institute of Technology, Madras

Lecture - 30

Synthesis Tool

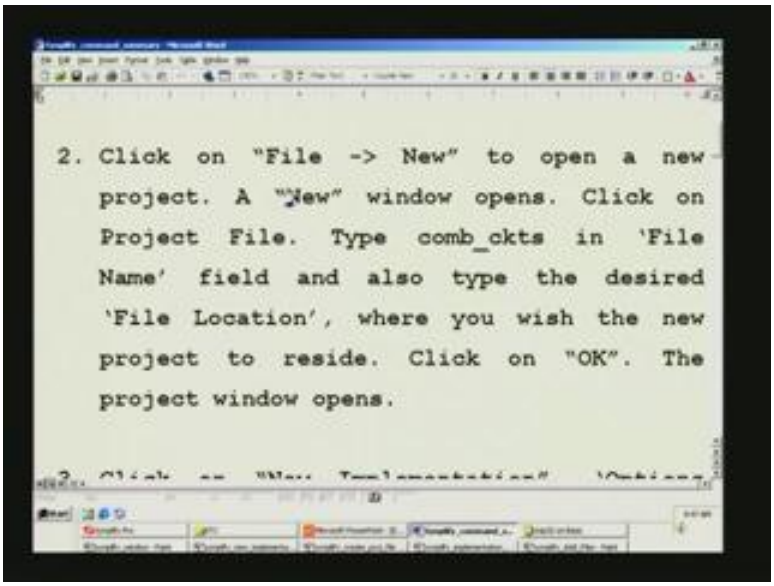
(Continued…)

Contents of Lecture - 30

(Refer Slide Time: 01:46)

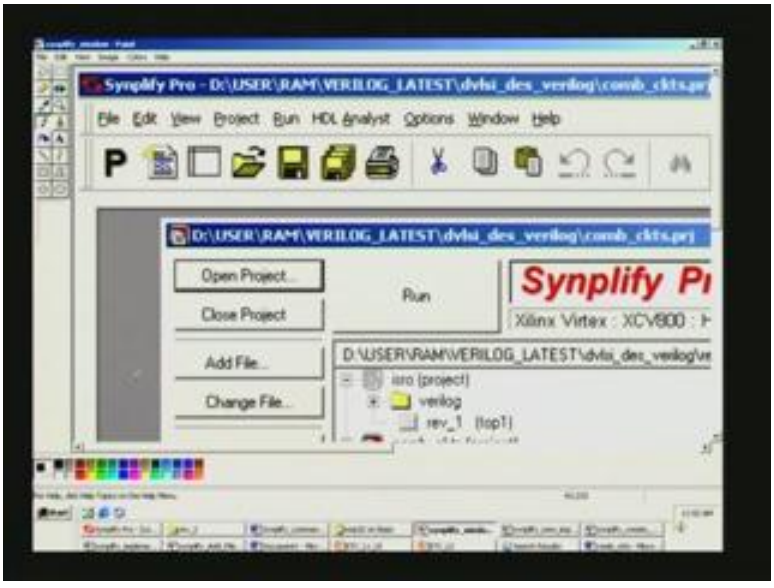
We were looking at synplify tool so far. We started with the following commands. We started the synplify window, by opening an icon here called synplify pro, by double clicking that. A window has opened and now we have seen how to open a new file.

(Refer Slide Time: 02:21)



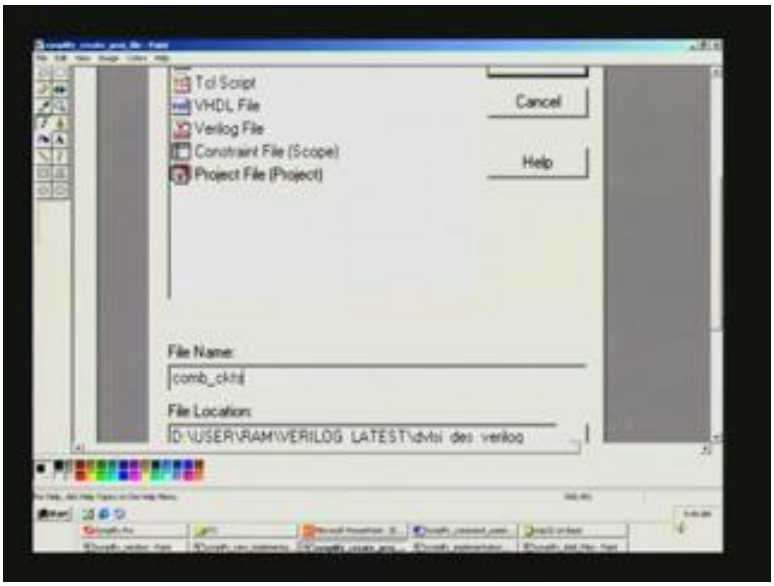
Click on file, New to open a new project. A new window opens. Click on project file, type combinational circuits. This is the design file that we had typed in the File Name field and also type the desired file location, where you wish the new project to reside. This is the place where your new files are residing and then you assert OK then the project window opens. This is what we had already seen here. This was the main window that we had.

(Refer Slide Time: 03:01)



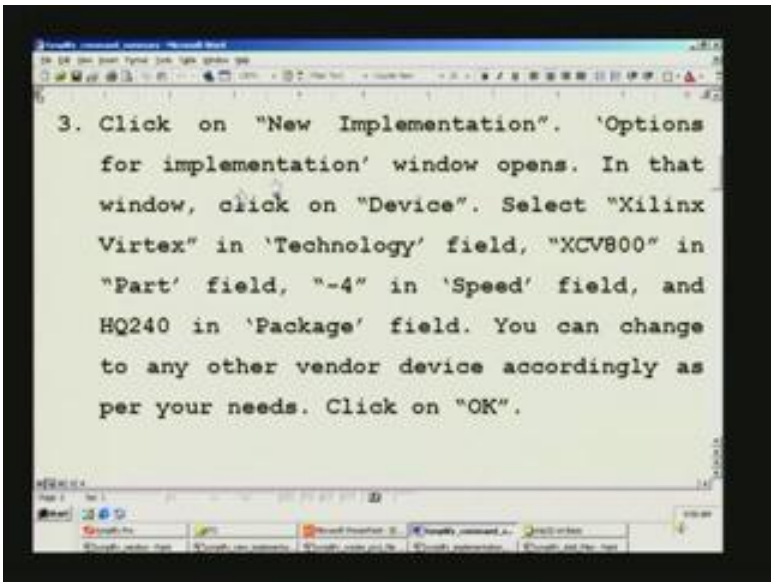
The main window looks like this with different options available, open project, then add file and once again, you have File Edit, View Project and so on. P also stands for project and so you can open the project from that also. The window that we had already seen is, seen just now here and the new project was also created by this window. We used this project file and then gave the combinational circuits identifying the design file.

(Refer Slide Time: 03:56)



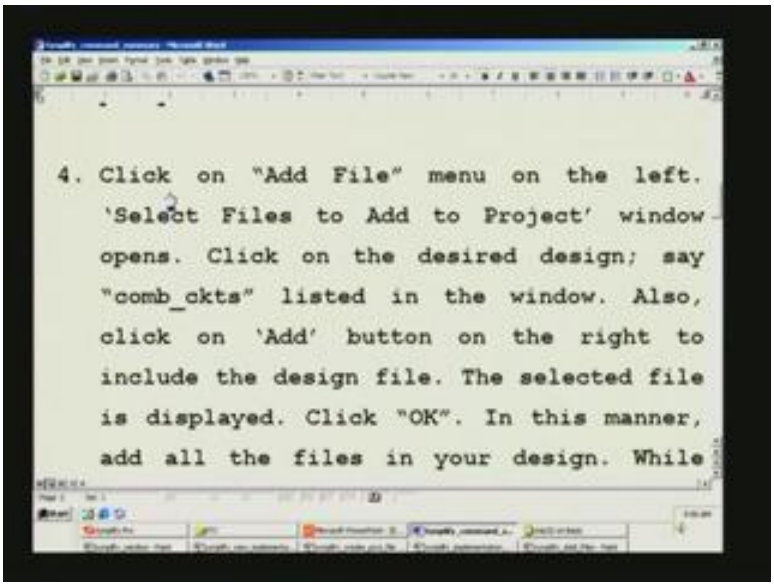
The entire path is shown here. When finally said ok, the new project was opened. Once we had opened the project, we had to identify the actual vendor, then the device and the package, desiring a particular number of pins and type of package.

(Refer Slide Time: 04:27)



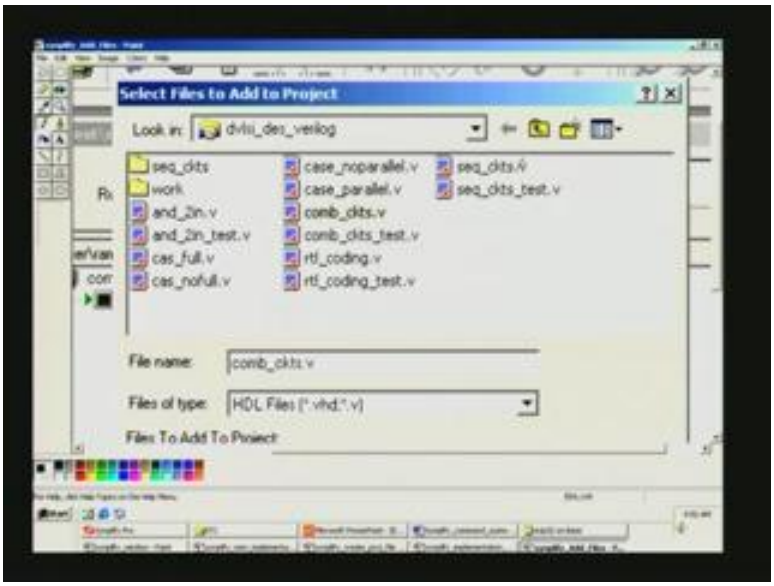
Click on, New Implementation options. For implementation, window opens click on Device. Select Xilinx Virtex in Technology field. XCV 800 is one of the devices that we had selected them in part field, - 4 stands for the speed and there are other versions - 5, 6 and so on. The higher the number higher the speed that you can get in Xilinx. In ultra, the reverse of that is true. You can change to any other vendor device accordingly as per your needs and then click OK. This also we have seen earlier and once you are done with this, just say click and it is up to that we had covered earlier now will carry on from that.

(Refer Slide Time: 05:16)



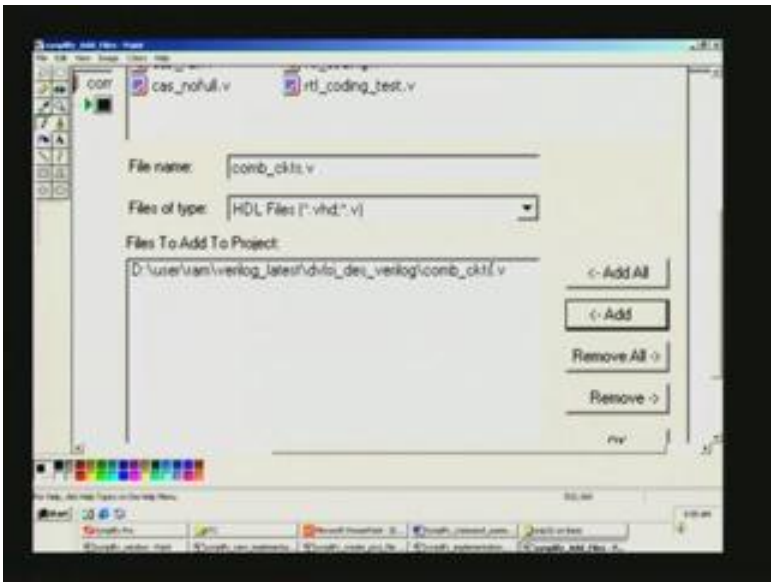
What we have to do is we have a design. The design may consist of several files and it may be hierarchical as in the flowchart seen earlier, the top down approach and bottom up approach and going in through several hierarchical files. If you have several such files, you had to add all these files and in this case of combinational circuits or sequential circuits that we are going to see will have only one file for the design. You need just add one file so to do that click on, Add File menu on the left. This is the file that you have here for add files. This is the window that opens when you click on Add. Now you have an add file here and so click on this, a window opens in which you have quite a choice to choose from it. Clicking this, you can go up folder by folder and if you click here you can go down if you wish and you can just by clicking at appropriate points you would land up in the desired folder where in all your design files are residing. For example, we had used combinational circuits that are also highlighted here and there are other designs as well and one such is sequential circuits.

(Refer Slide Time: 07:02)



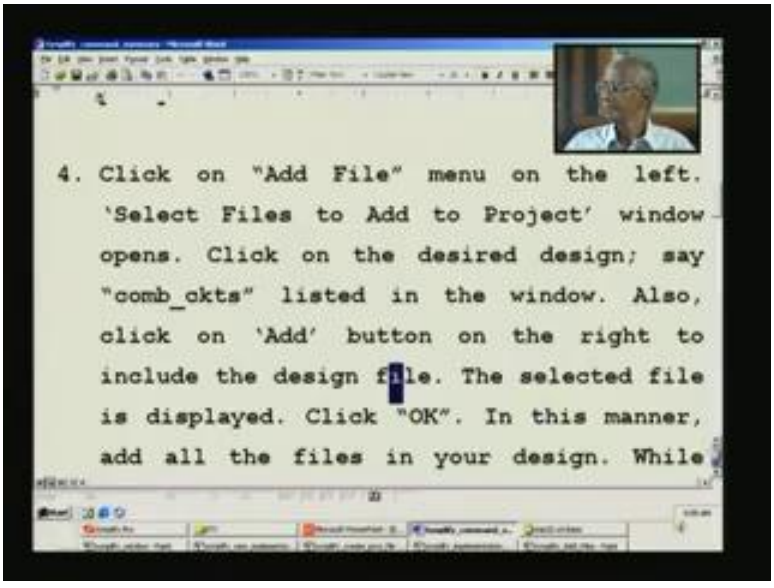
There is also a test bench, but you should not use test bench in synthesis. As far as this window is concerned, once you have selected the design, file one after another you will have to click. First, since this design has only one file, you just click on this here and the same name will be replicated here. It just says whether your file is dot v file or dot vhd. Dot vhd stands for the vhdl. As we had already seen another hdl and whether it is dot v or dot vhd, it is all same here. This dot v or dot vhd will discriminate what they are and the entire path is also mentioned here. You can see that it is dvlsi design verilog be in the last folder and in which this design file is located and if this does not appear right, now you have to add this file into the synthesis. Click Add. Upon clicking, you get this here. So that means this particular design is included in your list of files and like this you can go on listing as many number of files as there are in your design. And once you have done this just say and if suppose by mistake, you clicked on wrong design file, you can go on highlighting them by clicking on that and then say, remove. Suppose, you have made almost all of them incorrect, then you just say, remove all, it will remove all the things. Once again you can add, another approach is you can click on using control or use shift, you can select more than one design file. Then finally, add all. What you had to do is, you had to follow some hierarchic here also in the design files.

(Refer Slide Time: 09:24)



You start with lowest level and then go up gradually one after another. That is the way, it should be done. It is preferable that you add these files one after another by just using add and click one file and keep add. In this fashion you can keep adding more and right now, this particular design has only one file.

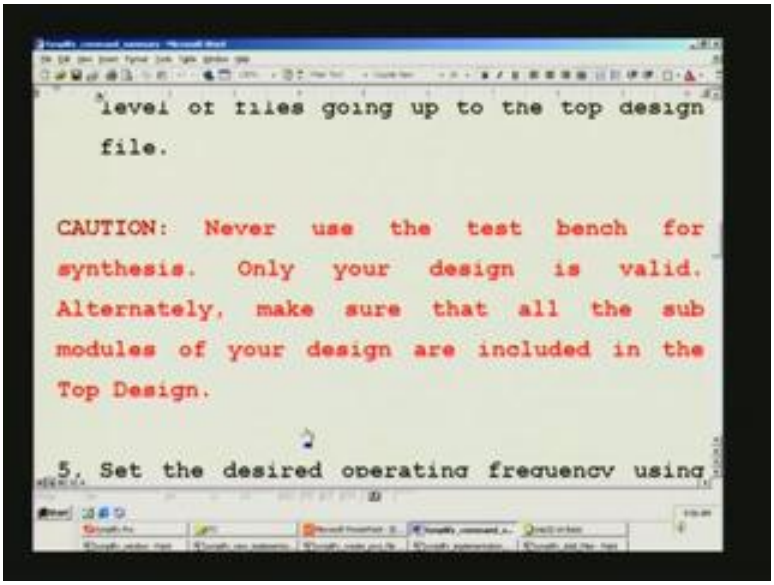
(Refer Slide Time: 10:03)



To summarise, Click on Add File menu on the left, select files to add to. Project window opens, click on the desired design. Combinational circuits are listed in the window. Also click on the

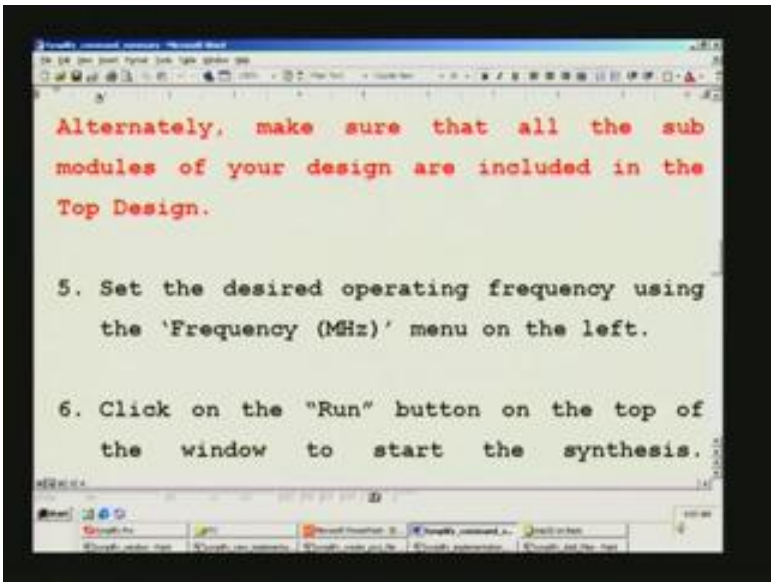
Add button, on the right to include the design file and the selected file is displayed. Click ok and in this manner, add all the files in your design. While adding files, add starting from the lowest level of files going up to the top design file.

(Refer Slide Time: 10:41)



We have also said you should not use a test bench in this because it is meant only to test your design and it is not part of the design. So there is no meaning in synthesizing the test bench. Many beginners do the same mistake of including the test bench and then land up in trouble. It may give lots of errors or warnings and therefore be on the caution. That is why, here it is said never use the test bench for synthesis. Only the design is valid. Alternately, make sure that all the sub modules of your design are included in the top design. This also we have already seen, I mean sub module in the sense the hierarchal files that you have just now seen.

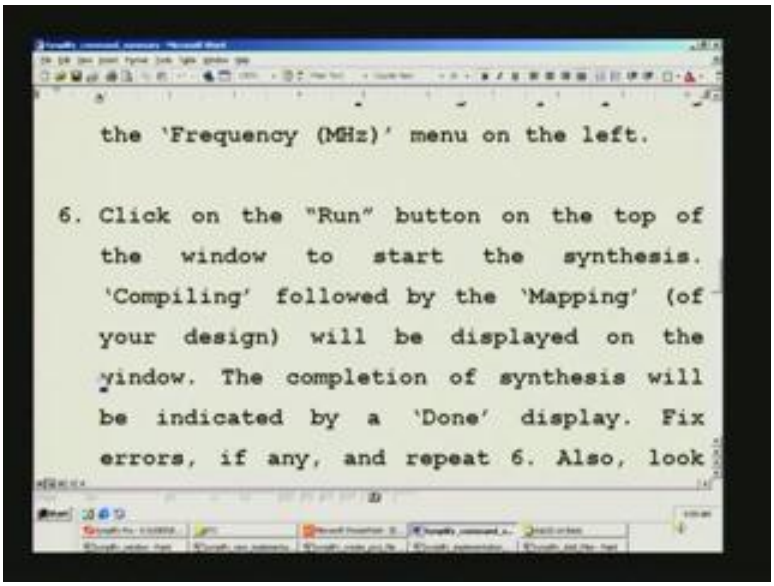
(Refer Slide Time: 11:37)



We have already seen sub module in the hierarchal files. Next step is to go to the actual synplify window. This is the window that we have already opened and we have to open one of these revisions also. The next step is to mention what frequency we want to synthesise in combinational circuits which is a hardware design. At what frequency it should function? This is only a desire that you are going to express. If I say 100 here, it means your desire of executing your codes at 100 megahertz and this is only a simulated situation. In synthesis, what you are going to get is analysis of your design. To find out the suitability for this frequency that you have chosen, Choose a high value and what the synthesis may report may be much lower because of the internal gate delays.

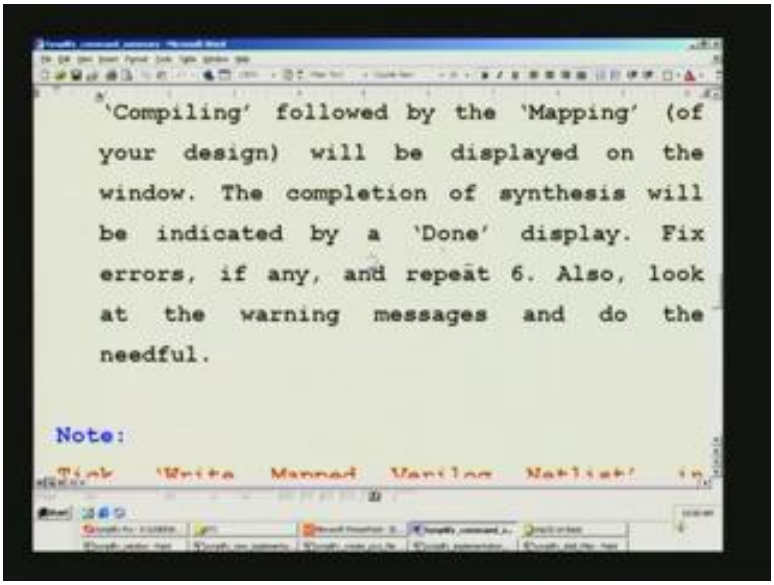
It may not be in a position to offer you this much high frequency of operation. You may have to tone it down to what it reports and this is not final. And only in place and route you get the actual frequency of operation. This is only a rough estimate and in some cases, if you have this FSM compiler resource which is ticked you may land up with peculiar warnings or errors. In that case, you can switch off this by removing or clicking on that and what we have seen at this step is we have just added in the menu, the frequency that we want and that is what is in step 5 which says, set the desired operating frequency using the frequency megahertz and it is the reporting menu on the left.

(Refer Slide Time: 13:56)



Next step is that you are ready for running the synthesis. So this is done by clicking on the run button on the top of the window to start the synthesis, compiling followed by the mapping of your design which will be displayed on the window.

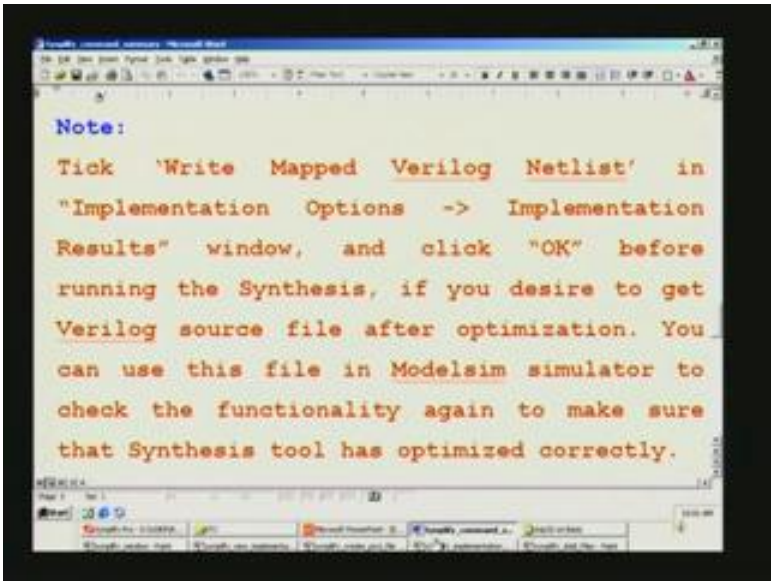
(Refer Slide Time: 15:18)



Once you say run, it takes your combinational circuit's dot v which is the design file and runs the compiler, in order to find whether there are errors and if there are errors, it will report. Once this is successful, it will automatically go on to the next step called mapping. You have already

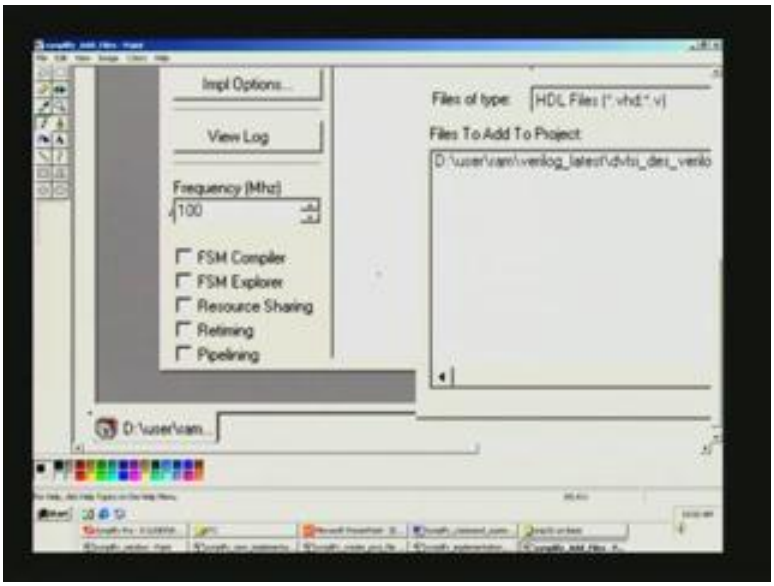
selected the device in the earlier few steps. And as per the mapping that you have done, it will actually take the primitive cells available in that particular FPGA. We will go in to the cover more details as to what these cells are and that is what is meant by mapping and so after compiling is over, mapping will be started. And once this is also over, it will report that synthesis is done. Followed by the mapping of your design, on the window the completion of the synthesis will be indicated by a done display and errors are fixed if any and then repeat 6. You have to keep on going around till you are free of all errors and take care to heat the warning messages that the synthesis tool gives.

(Refer Slide Time: 16:09)



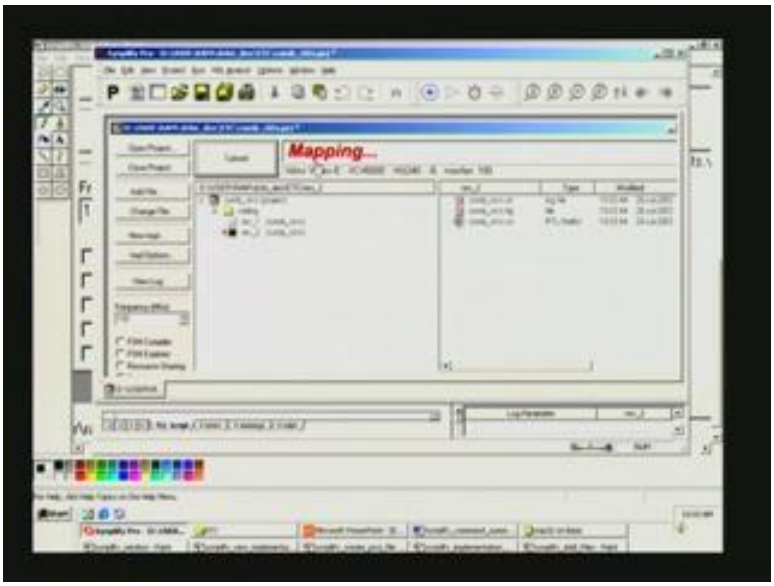
This is the synplify window. This is the live window and as we have already mentioned it is written as 100 megahertz here, you can see one of the windows in the zoomed version.

(Refer Slide Time: 16:42)



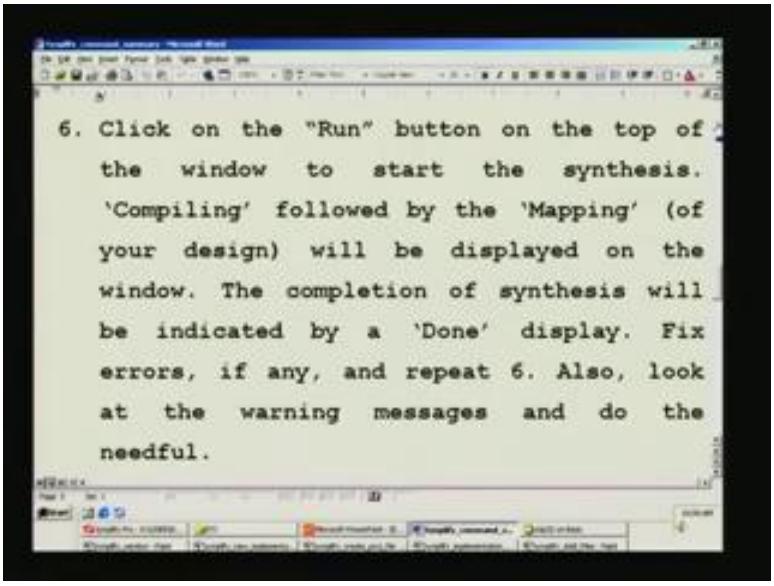
So the frequency mega is in hertz and if you want to change this one megahertz after another, you just need to just click this up arrow. If you click once it will go to 101 and every time it advances by 1. Similarly it will advance down by 1 if you click this arrow. This is what the tick marks here are. It will be safer to remove these features for the time being at least. Coming to the synplify window, what we need to do is to run the file as seen in step 6. Let us run it. This is revision to combinational circuits. Let us run this. What it says is there are no hdl files to compile in project combinational circuits. We have not added the design file. We should just say Add file and so if you say add file it lists all the design files here and we need to add the design file combinational circuits. You can either double click or single click and then say add, and as mentioned earlier, the same selected file along with the entire path is displayed here and once this is done say ok.

(Refer Slide Time: 18:07)



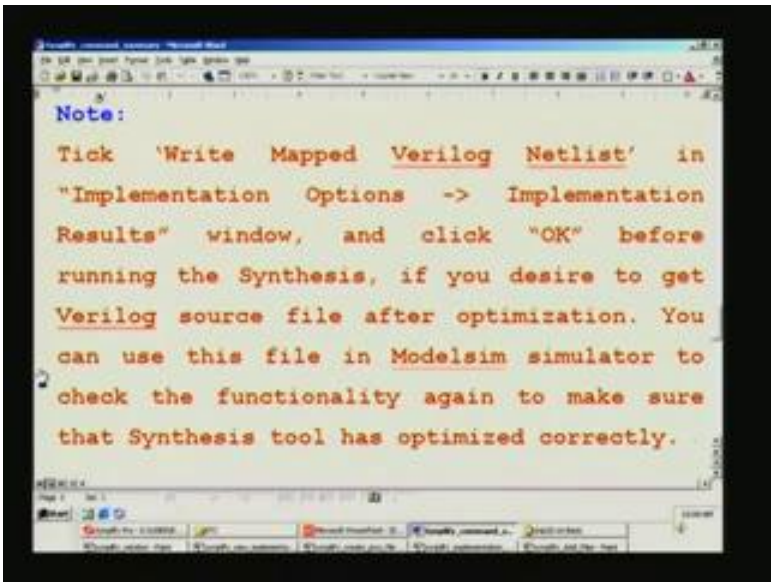
We have added in this combinational circuits design file and now let us see whether it does the job. Now you can see that it was doing the compiling that was successful and then went on to the mapping. That is what is happening in the window there. Once it is done, it reports as done and if there are warnings, it would have given them and that it means it is error free. How do you see look into all the details. You have what is called a log file and if you click on this, a log file will be opened.

(Refer Slide Time: 18:43)



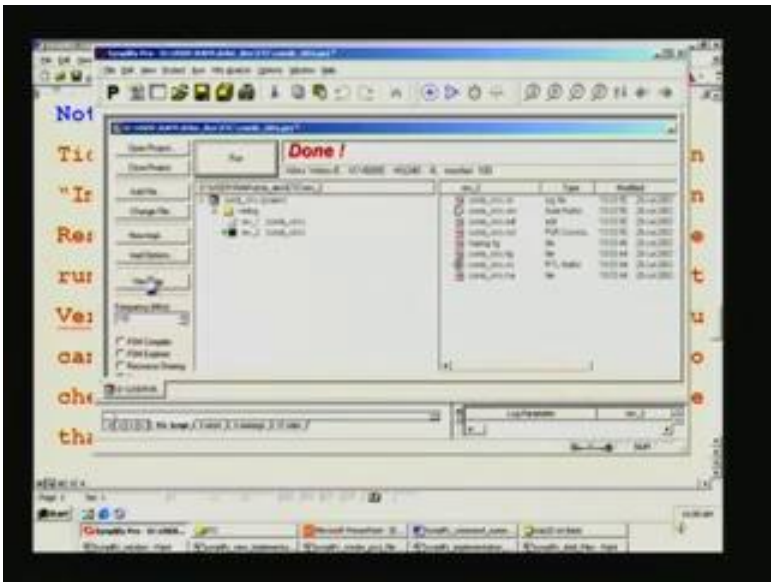
Click on the run button on the top of the window to start the synthesis compiling followed by the mapping of your design which will be displayed on the window. The completion of synthesis will be indicated by a done display and fix errors if any and repeat 6. Also look at the warning messages and do the needful.

(Refer Slide Time: 19:08)



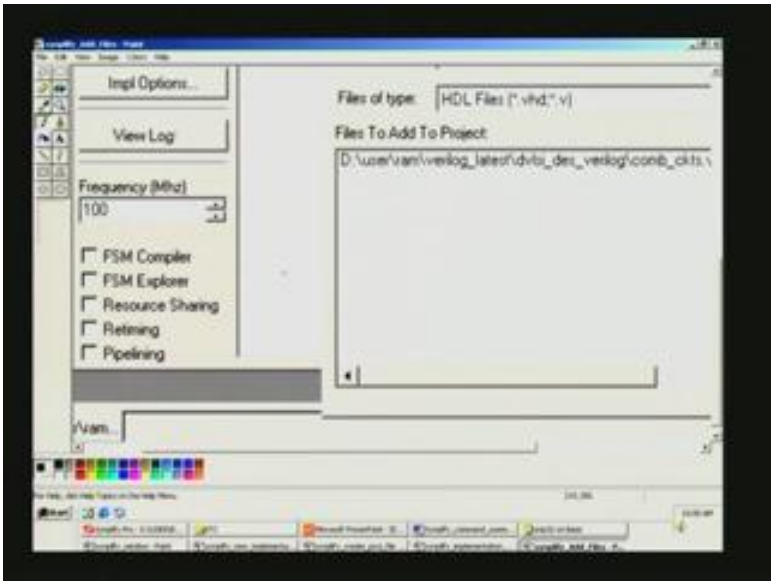
We have given a note here which says, tick write mapped verilog netlist in implementation. Before we come to this step, we will go in to the log file.

(Refer Slide Time: 19:26)



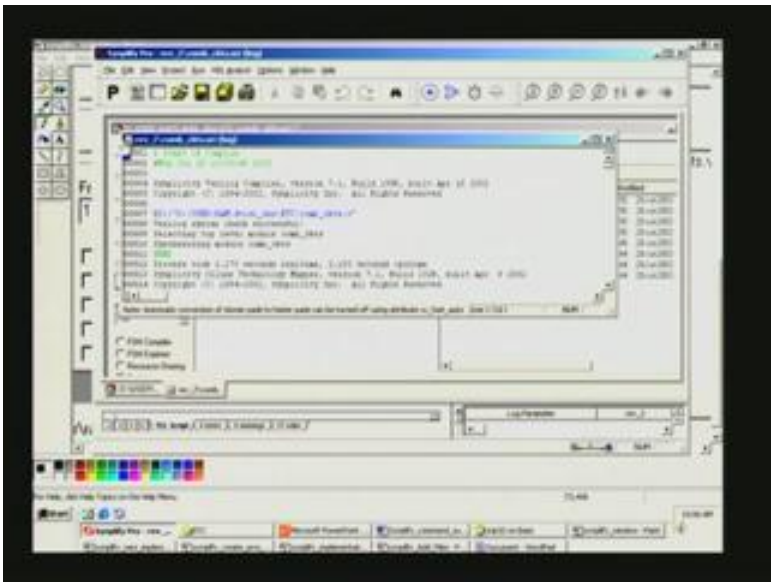
Here you see a view log. I will zoom it.

(Refer Slide Time: 19:37)



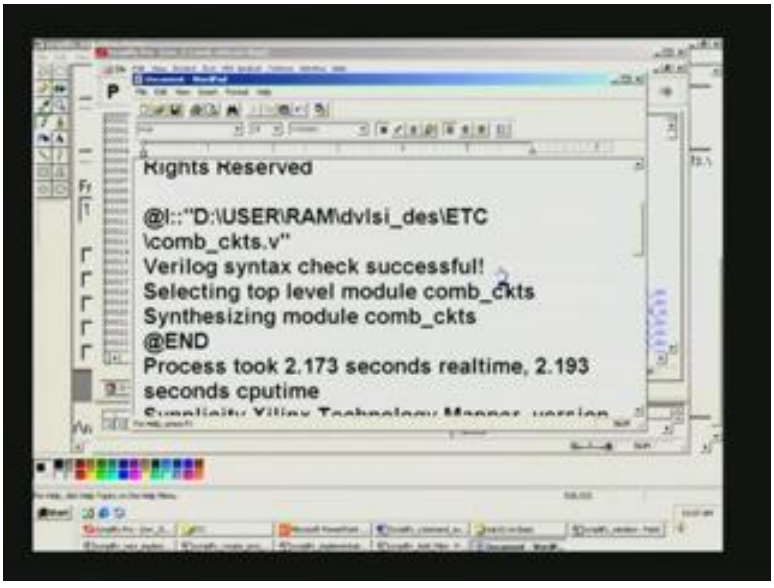
In the zoomed version you have a button for view log. If you press on this View log, this log file will open. Because the font size is not enough we will open one word pad and can just copy the log file to it with a bigger font, with 24 font and perhaps a bold so that we can see with ease. Now go back to the synplify window and then will just click on, View log and so now it opens a log file here.

(Refer Slide Time: 20:50)



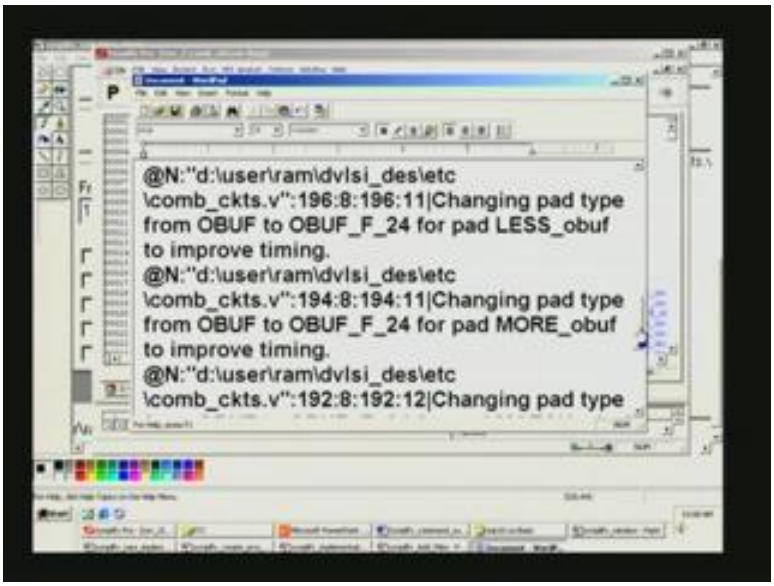
It will be difficult for you to read, and so we will just copy some of this and paste it here in word pad and now you can view it more clearly. The starting of compilation such as what is the time are all reported here. This is from synplicity corporation and so the entire path where your design file is located is also reported here and I is for information and it will throw some light that is what it has to communicate with you.

(Refer Slide Time: 21:47)



Here it reports, verilog syntax check is successful and your top module was combinational circuits. It reported that it was synthesizing these combinational circuits. How much time it took for processing and since we had selected xilinx vendor, all these are also reported here. Mapper means it is mapping on to the actual FPGA device. Since it is vendor specific that vendor is also reported here.

(Refer Slide Time: 22:37)



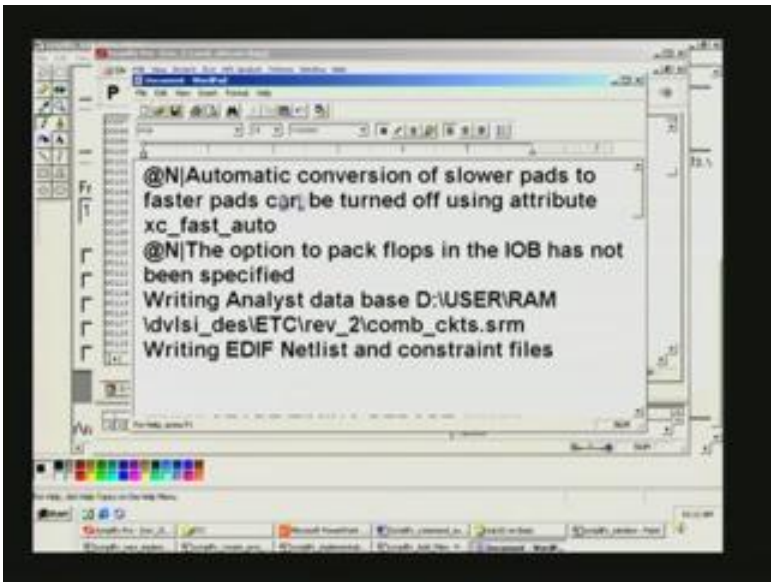
Here it reports that there are primitive cells in the fpga such as input buffer, output buffer and even with output buffer, there are high speed buffers available. If the timing constraint that you had mentioned is not meant, it automatically maps on to a buffer which gives a better response. Your frequency of operation improves by changing over from one of this buffer. Let us say one output buffer to output buffer with this extension. It means here by the statement to improve timing, this has been changed and this is automatically done by the synthesis tool, because it is part of the synthesis tool to do so.

Once again, it is all same to repeat it for different signals that you have in your design. Just remember we had more match in one of the example there and all of them are primarily the same. See there one N, N stands for note and there may not be much of a difference between information. Note it does not really matter what they have nomenclature, just look into what it reports and it is quite easy to follow. And once again, it says output buffer is reconfigured to another. Now we do not need this portion and we will remove this and will look into the other

aspects of the report file. All of this are primarily same and so I will by pass those and then tell what is relevant to you for analysis and so this will throw light into the working of your own design in a sense you can analyze using this report file. And what it says here is automatic conversion of slower pads to faster pads, and you can have even the option of turning off this conversion from lower buff to the higher speed. It is normally preferable to just leave it as we are anxious to have extra mileage or other timing closure. That is to say that we may be aiming at the highest possible frequency of operation for most of the projects and so we can agree with what synthesis tool is doing and the option to pack flops is flip flops, if the IOB has not been specified. There is also a provision for specifying this. As a default, we do not specify this and so it may not be that important.

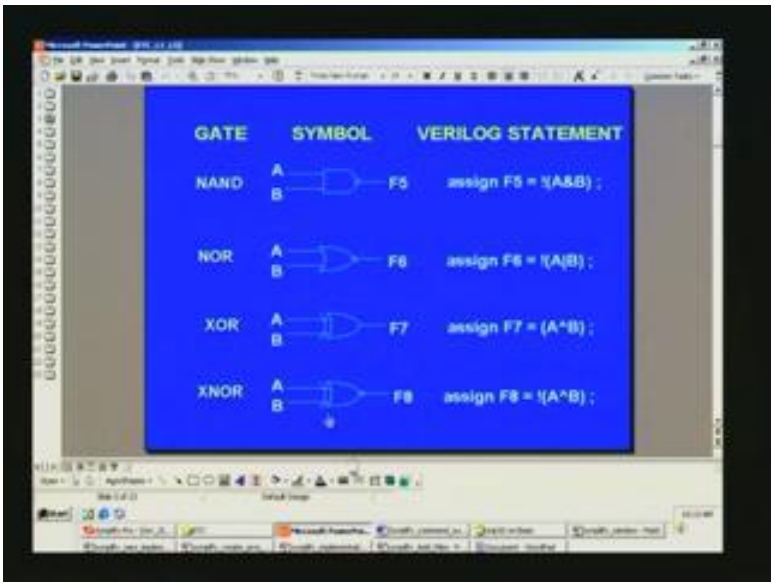
Probably later on you can experiment what they are and finally it says writing analysed at least database this is the path and the extension file is for this. The data that is being analysed is saved in this file and it also has one important file called EDIF and this is very essential if you want to migrate from synthesis to the place and route which is the next tool that we need to have. This will be a dot EDIF file and EDIF means electronic data information format. This comes as a standard format and normally the third party vendors would be writing in this format rather and that will be taken as input by the vendor specific platform for place and route. This is an important thing which we can generate and we will be having a look just as to how to do that. I will just copy this portion here up to this.

(Refer Slide Time: 27:01)



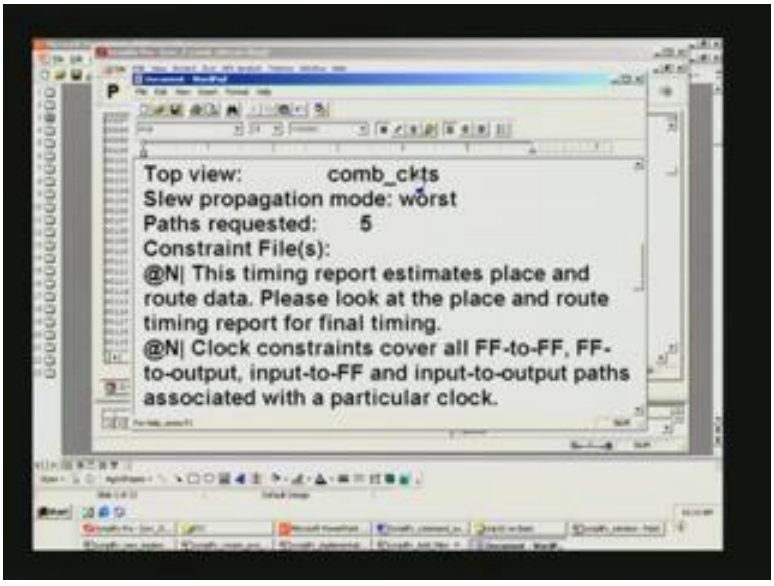
N is for note and automatic conversion of slower pads to faster pads can be turned off using attributing. By giving this attribute it can be turned off that is what we have already seen the option to pack flops in the IOB has not been specified. Writing EDIF netlist and constraint files, constraint in sense, the frequency that had been asked for is basically a constraint and associated timings are also available It creates these files as well and so then what you have is the starting of the timing report. Now we have already seen that one more I want to show you it is already here. These are all the combinational circuits we have seen earlier. We can also view the schematic and when we come to that we will have a comparison of that.

(Refer Slide Time: 28:20)



Let us see what the timing report says. So the top view is again given here along with the number of paths. It goes through labyrinth of gates and those paths are also specified.

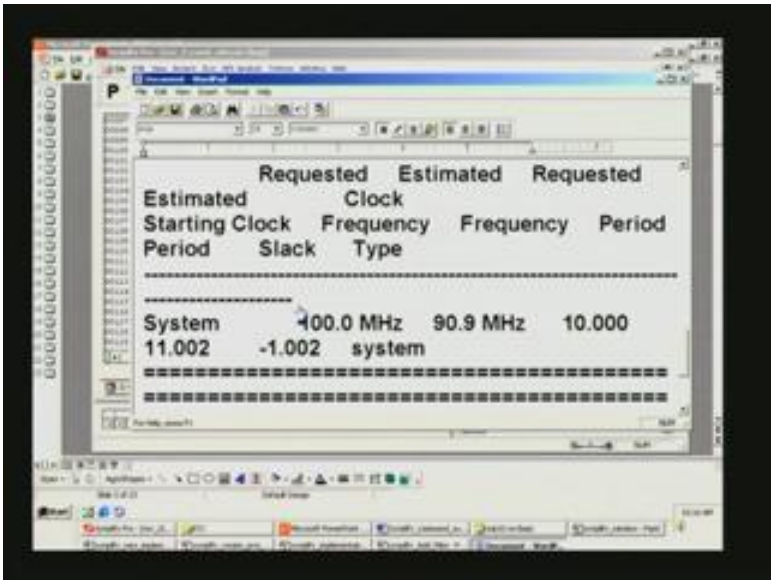
(Refer Slide Time: 28:36)



If you don't specify, a default will be put. In fact 5 is the default that has been put here and this timing report estimates place and route data. So even here it makes so to say a place and route. The actual place and route however happens in the next tool such as xilinx place and route. It says us to look at the place and route timing report for final timing. In fact, you will have to look

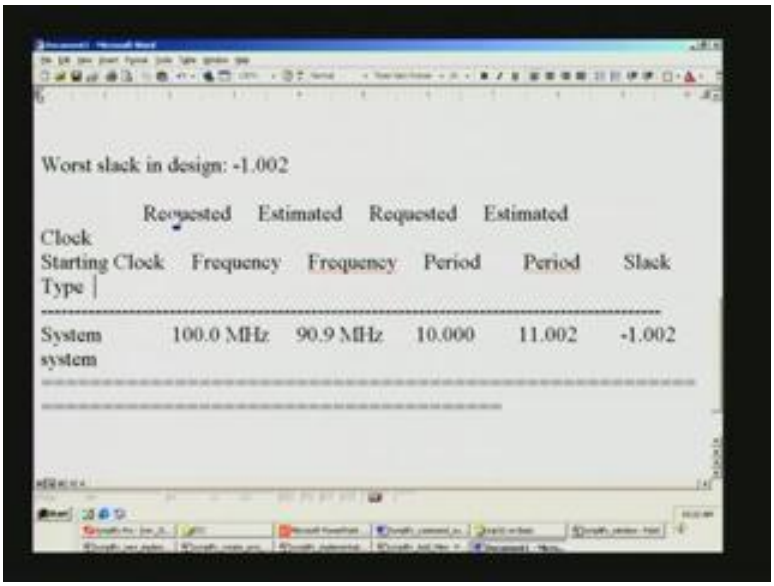
into the vendor specific thing. Clock constraints, Cover all flip flop to flip flop, flip flop to output, input to flip flop. All these timings and even synthesis tool will report here. Input to output paths associated with a particular clock is also reported and these are all very essential. If you want to really make a serious design. As such to have a better performance in terms of frequency operation, you need to know how much delay it has taken for a particular path and it may happen so that only some paths are very critical and they are the ones which are really slowing down your operation. You can see how to redesign so as to improve the timing. In future we will see how to improve the performance and now we will see the actual frequency that we are asked for 100 mega hertz.

(Refer Slide Time: 30:46)



It says clock frequency you have asked for is 100 mega hertz and what it is able to give is 90.9 megahertz and what was the device actually. All that will be listed later on here. So will go to a synplify window, if you see towards the end it is xcv 600 ehq 240-8 and this is the device that has been chosen. Looking at this window these are all clearly laid out. This 100 megahertz is requested frequency and estimated frequency is 90.9 mega hertz. It is all completely in disarray and this is the best we can do otherwise I will have to arrange it. It is a time consuming thing. We see that requested frequency is 100 mega hertz, and then estimated frequency is 90.9 mega hertz. We can see here in zoomed version and it is slightly in disarray but you can make it out that the requested frequency is 100 mega hertz and then estimated frequency is this.

Refer Slide Time: 32:02)



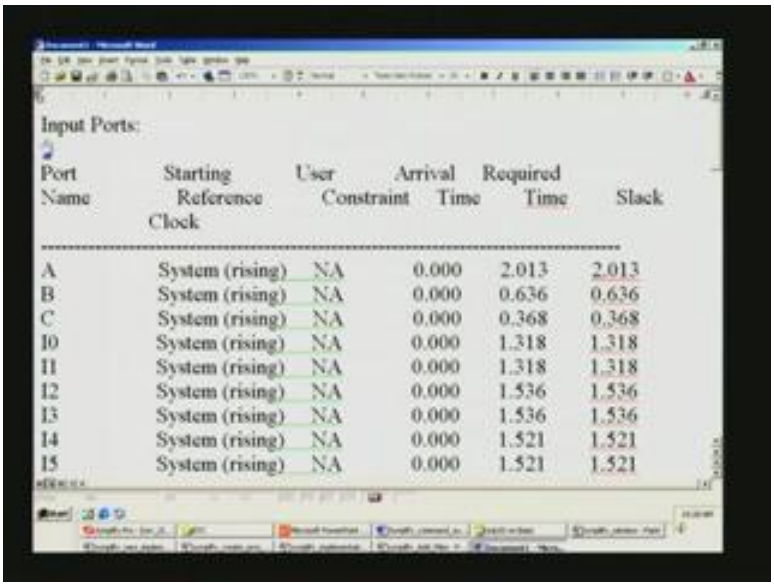
As I mentioned it is easy to divide 1000 by 100 mega hertz in order to get straight away in nanoseconds. What you get is 10. 10 nanoseconds are what appear here because frequency and period are inverse straight away in nanoseconds. The next one is what we have requested earlier is 100 mega hertz in the frequency in this window. If you see this window it could meet only 99.9 mega hertz because of the gate delays and so on. Again design to design, this will be varying and even the same design you can improvise on this timing if you have a fresh look into the entire paths which we will be seeing later on and you can even improve either marginally or even better. Whatever may be, why you go for a particular chip because you have the hardware available with you, you must have conducted complete survey earlier. Zoom into a particular vendor and particular device based up on your actual requirement.

It might have been dictated by users themselves. You do not probably have much choice and swapping from one to another. It may not be meaningful because you just want different timing and you cannot keep on changing device. The right device will have to be frozen, unless it is merely for learning. Say it has no meaning because your target is what you have to decide. At the design stage, you have to decide this and you have plenty of time to decide. I have mentioned earlier there will be a team of engineers working for the mechanical design and the hardware design that is the making the pcb and its layout and so on and the so called software team, who will be working on verilog and all these teams will have to be coordinated synchronize and

unless you had frozen the design. Unless you freeze those specifications, you cannot go forward. It is important that right at the beginning stage, you have to make a decision and unless something goes thoroughly wrong, you cannot afford to change the device from time to time. You may have to freeze at some point of time and we have already frozen on a particular device and therefore will have to stick on. The flexibility that you have here in this same category is the package is what is going to count so that you can change it to another type within the same vendor and retaining the same type of package. For example hq 240, you can stick on to that and even within this you have 3 or 4 speed ranges.

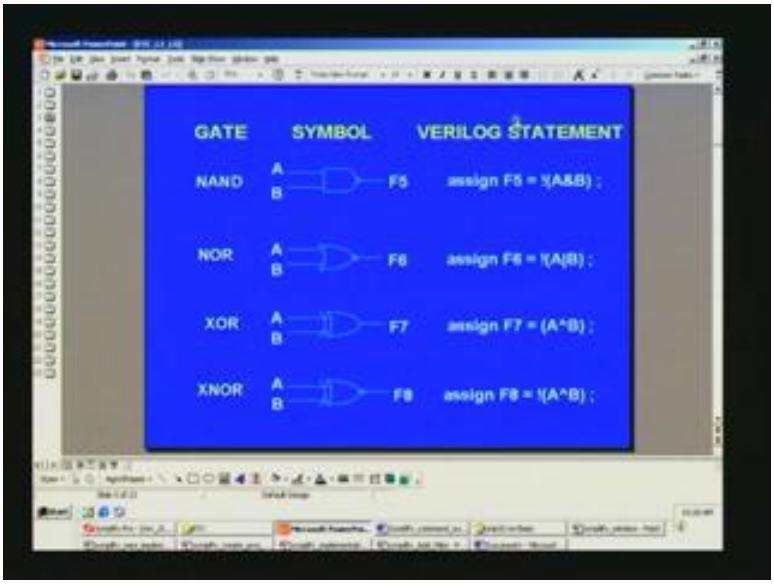
You can always select from that is corresponding to 90.9 mega hertz operation, which it has reported and even this is not the final thing and that will be reported by the place and route tool will be final. Xilinx tool for example for place and route or it may even go up and we cannot exactly say how it will behave. It all depends up on the complexity. From experience I can say large designs normally tend to flow down in place and route whereas small files which will not be the actual design which is final will tend to go up. This is not a clear indication as to this will be the final frequency and what will be reported by the xilinx say the place and route only will be final as far as the xilinx fpga is concerned. Corresponding to this, the time period is 11 nanoseconds. Note the difference here and this is referred to as slack time and it is a minus slack time. Beyond the guard, if you see a minus here, it only shows that it has not met the constraints that you had asked for and you had asked for 100 mega hertz whereas what it has to offer only this. So this is the price that you have to pay and at times you may have to pay much dearer price than this and the difference seen already is the slack time. All of them are in nanoseconds. Coming back to the window on progressing further, we have seen that there are multiple paths we are precisely going to see these paths and so I will just copy once again here.

(Refer Slide Time: 38:27)



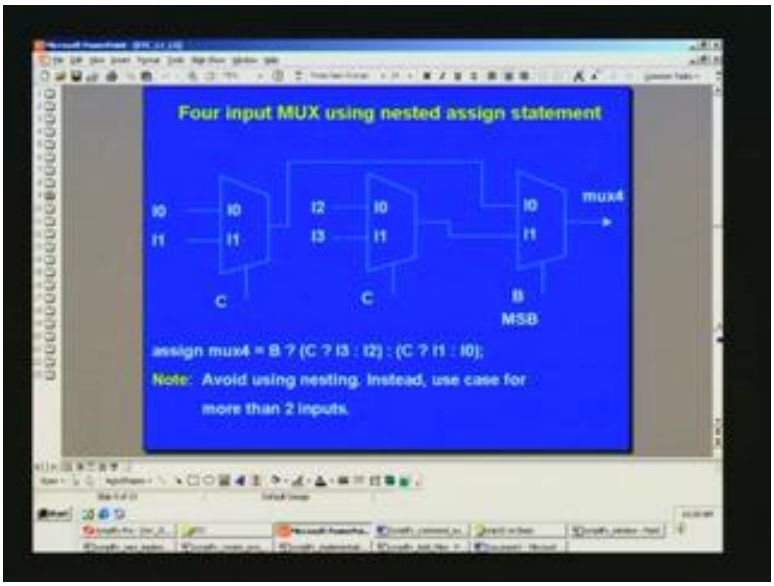
What you see here is the report for input ports and all input ports in the particular design have been used by us, such as A B C as input and then, I0 for MUX and so on.

(Refer Slide Time: 38:27)

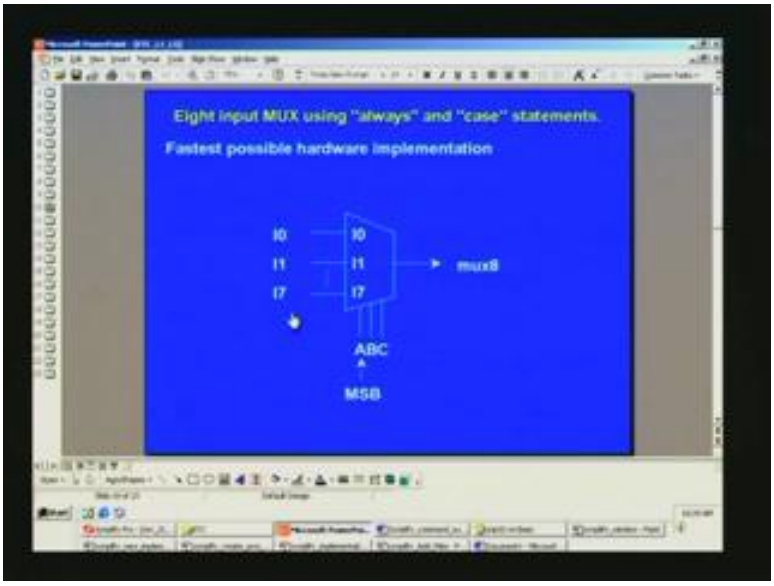


A B is all inputs and these are all the outputs. Then we had gone for MUX, all I0I1 as you can see and this is a 4 input MUX then 8 inputs MUX and so on.

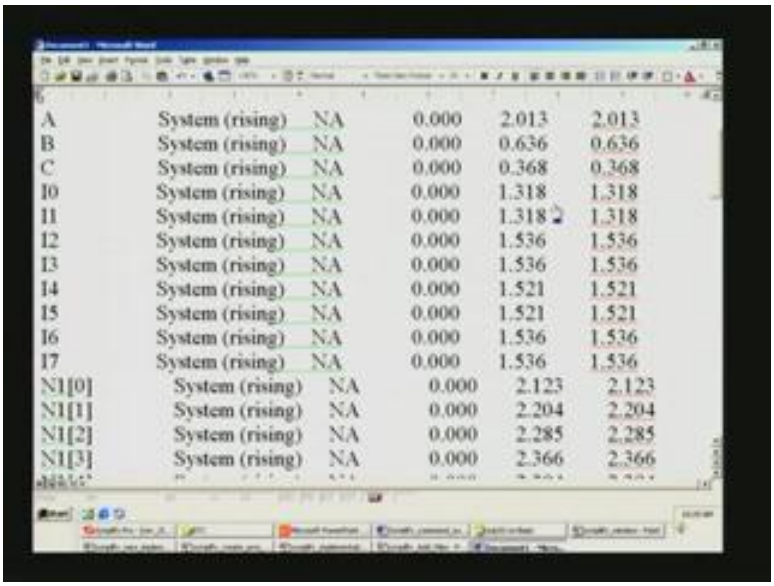
(Refer Slide Time: 38:57)



(Refer Slide Time: 38:59)

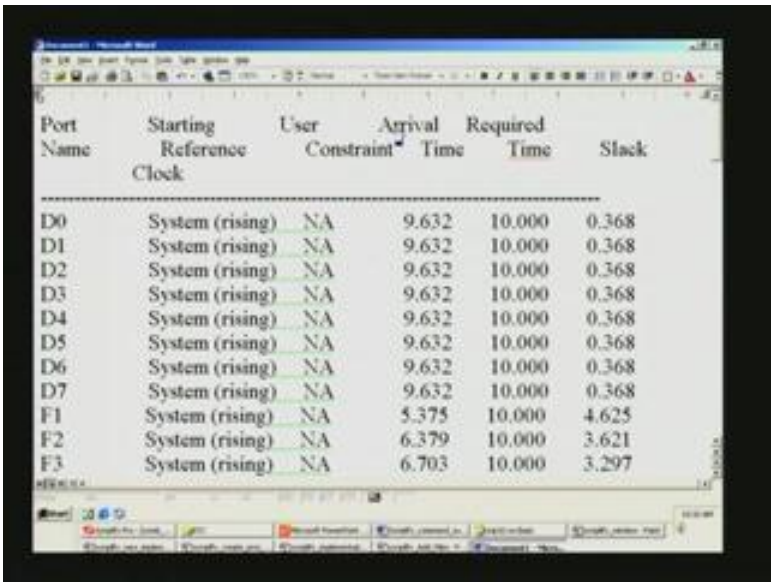


(Refer Slide Time: 39:19)



This is precisely reported here. First, Input ports and then followed by output ports. All the signals will be reported and will see one after another. The first thing that we were looking at is this input and you can see here, the starting point, and the path it travels a particular signal how it goes. And at every point of time some delay is caused by the cells or gates or whatever they call as cell in FPGAS. To start with, this is for A B C all these inputs are here. I mentioned them as 0 because, that is the start for this and how much time it takes to come out of the input buffer is not clear and this is input buffer which we will confirm later on. What is important is, it gives the actual delay in time taken place naturally and it might mean either the net or some buffers. These are all in nanoseconds. The slack time is also reported here. So you have asked for one thing and what has been achieved is another and so the difference is what is called slack and two columns you notice to be same. It merely lists all the inputs here and some of them are negative and so let us find out why they are. The details are not enough here to find out what it means, whether it is a net delay or the get delay. For all the signals it is reporting here and similarly outputs are also listed in this fashion and some of which are seen.

(Refer Slide Time: 41:59)

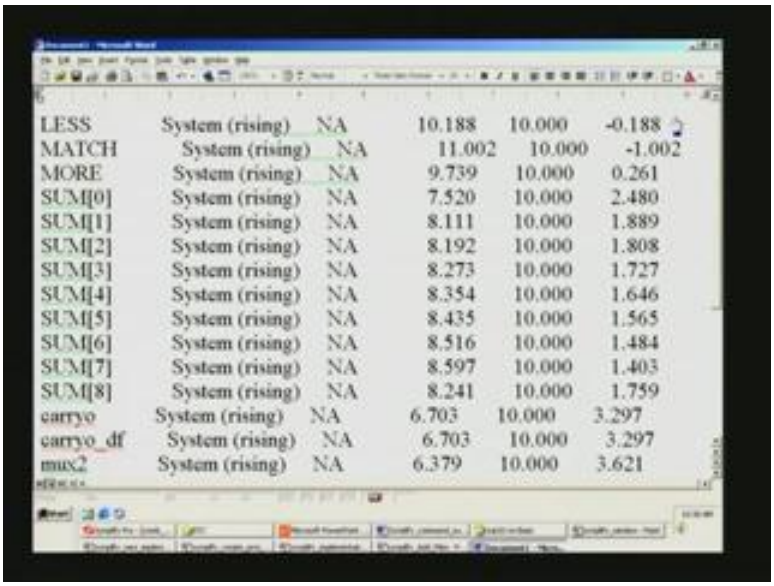


Port Name	Starting Reference Clock	User Constraint	Arrival Time	Required Time	Slack
D0	System (rising)	NA	9.632	10.000	0.368
D1	System (rising)	NA	9.632	10.000	0.368
D2	System (rising)	NA	9.632	10.000	0.368
D3	System (rising)	NA	9.632	10.000	0.368
D4	System (rising)	NA	9.632	10.000	0.368
D5	System (rising)	NA	9.632	10.000	0.368
D6	System (rising)	NA	9.632	10.000	0.368
D7	System (rising)	NA	9.632	10.000	0.368
F1	System (rising)	NA	5.375	10.000	4.625
F2	System (rising)	NA	6.379	10.000	3.621
F3	System (rising)	NA	6.703	10.000	3.297

The arrival time means the data delay is what is mentioned here. This is the output. It has started at a particular reference point and the starting reference point is given here with reference to a clock. The combinational circuits have no meaning with reference to the clock as such but since no reference was there, it has taken this as reference by default. And so what it takes is the d0 manifests only after this much time. You have requested 10 nanoseconds. The difference is slackened. Note that they are all positive slacks here and that means you have still cushions. So you can jack up your frequency of operation as far as these outputs are concerned.

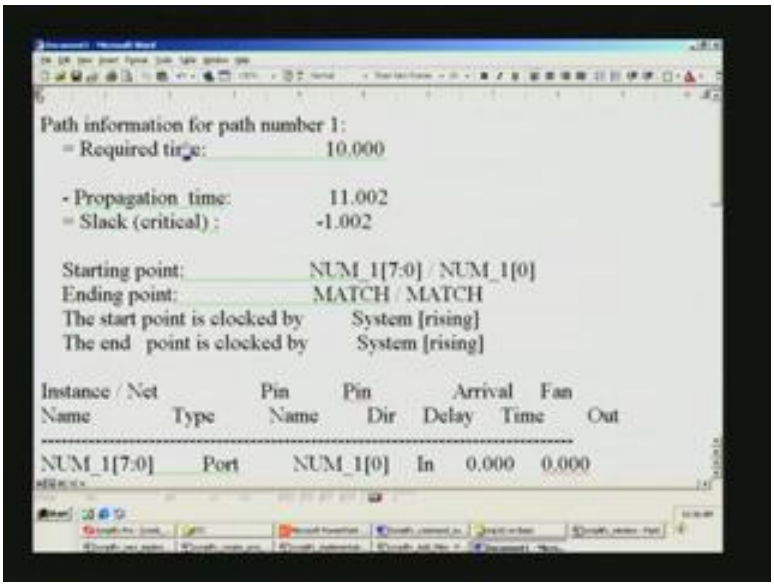
Like this if you analyse each and every output, you will know where negative slack happens. So they are the critical paths that you will have to reconstruct designing. These are all the inputs we have already seen. There are so many other outputs, f_5 f_9 and so on and all of them report more or less the same. Around 6.379 is the arrival time whereas 10 nanoseconds you had asked for and 3 or 4 nanoseconds are available. You can speed up the operation. You have lots of scope there and some of them it says timings are not available and you may have to look into namely f_{11} to $f_{12.}$ We have to see, why it is not available, and you can see a negative time somewhere it is here. If you see here, you can see negative timings for fewer matches. These are all critical paths and it is only loosing very marginally.

(Refer Slide Time: 45:02)



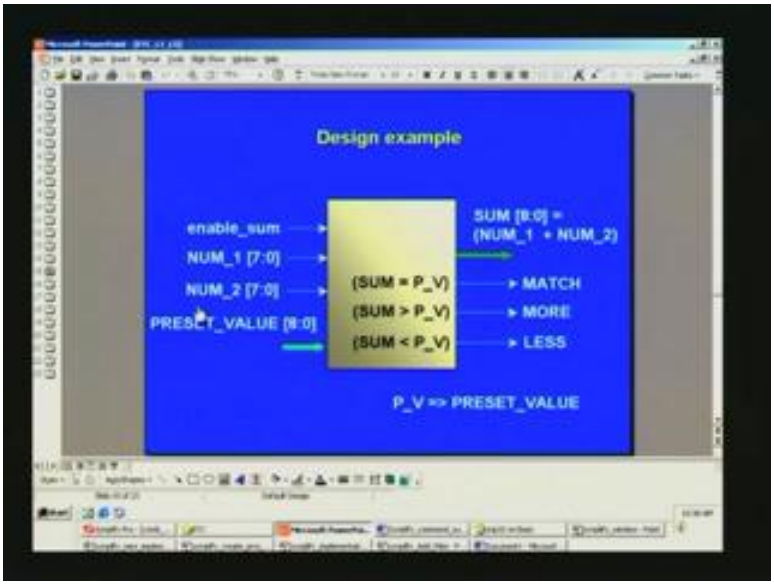
You can have a look at this and see whether you can improve the timings. If you cannot improve the timing, after doing your best probably you have to tone down your frequency of operation or you may have to go for higher speed version. Similarly, it reports for sum carry MUX 2 MUX 4 MUX 8 and some total and so on. All of them are safe and matches are the ones and it is minus 1 nanosecond for the match. If you have a look then you can into the schematic which you can view and then you can analyse further. Let us see what more the report files say. Again some more negative cells are reported here for number and we had done sum and comparison of two numbers. In that example, it is showing this here. Once again same match is reported here and let us have look at that.

(Refer Slide Time: 46:53)



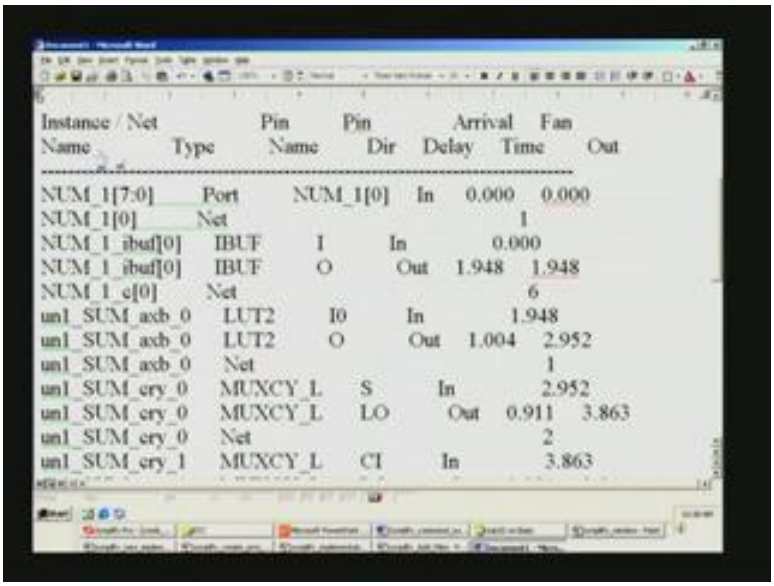
Here it says for path number 1it is mere number here to keep track of how many paths and the required time is 10 nanoseconds and slack is mentioned here. The total propagation time for this particular signal here say the starting point is for number 1 and the ending point is matched here. Even with this the problem that was encountered is not for all the bits, but only for a bit 0 and this is the start and this is the end match.

(Refer Slide Time: 48:00)



These are all the two numbers. If a match is found after addition, the sum is equal to the pre-set values, and then a match is reported. Otherwise if it is greater more and if it is less then goes high. That is the actual circuit that we had coded and for which we saw and the synthesis report is here for that particular thing. It has a 7 through 0 8 bits. That we had is number 1 and 0 bit. The path which is critical is for 0 bit.

(Refer Slide Time: 48:00)



It starts and goes through the labyrinth of different gates and primitive cells and it reports delay at every point of time and once again it is in slight disarray. This is number 1 and 0 bit in question and type is net. It means this is the 0 bit in question. Next one is pin number it reports whether that particular thing is an input or output or if it is net neither of this. That is what is implied here and so the number 1 0 is reported there and the number 1 0, whether it is input port or output port will be reported here. But in this case, it is an input net and it is basically a wire. So it is just a physical wire there. It is neither input nor an output here and the pin direction is in or out and it echoes the same thing. The next one is an important thing which says the delay associated with this. For example, in this number 1, the reference time is 0 here and then progressively the delay is at every point of time. For example, next one if you take it goes through an input buffer and that is the input it travels via the net and here it reports. There is a net delay and it has not reported here. This is for the net input of the buffer that goes to this number 1, 0 bit that goes in to an input buffer and the net delay is 0 there and the output of the buffer is

what is reported here. And it takes 1.948 nanoseconds propagation delay for this particular input buffer and that is what it means by delay.

At every point of time it goes through another. For example, it goes through 2 input look up table. FPGA has different primitive cells and they are all input buffers, output buffers, LUT 2 input, 3 input, 4 input and then MUX, 2 input MUX and so on. These are all the basic components an FPGA is composed of and vendor to vendor, it may change a little. What you see here is the number 1, that will finally manifest as some output such as match but it will have to go through a number of gates such as IBUF, then LUT 2, MUX and so on. It is at every gate or cell what you call, there will be a delay associated with that. That is what is reported here finally and it reports is the cumulative delay. For example, this is the arrival time and so you started with 0 then first gate contributed to 1.9 and so is the case with the cumulative delay. Next one is that the LUT 2 contributed to 1 nanosecond. This is cumulative and this gets added up and so on. It finally reaches the match and when it does, it has taken 11 nanoseconds. In this fashion you can analyse any of the circuit of your design and then make a decision as to whether you can rewrite your code and so that you can improve the speed. We will continue with this in the next class. Thank You.

Summary of Lecture 30

(Refer Slide Time: 53:34)

Next Lecture

Synplify Tool - Schematic Circuit Diagram View

(Refer Slide Time: 54:00)