Architectural Design of Digital Integrated Circuits

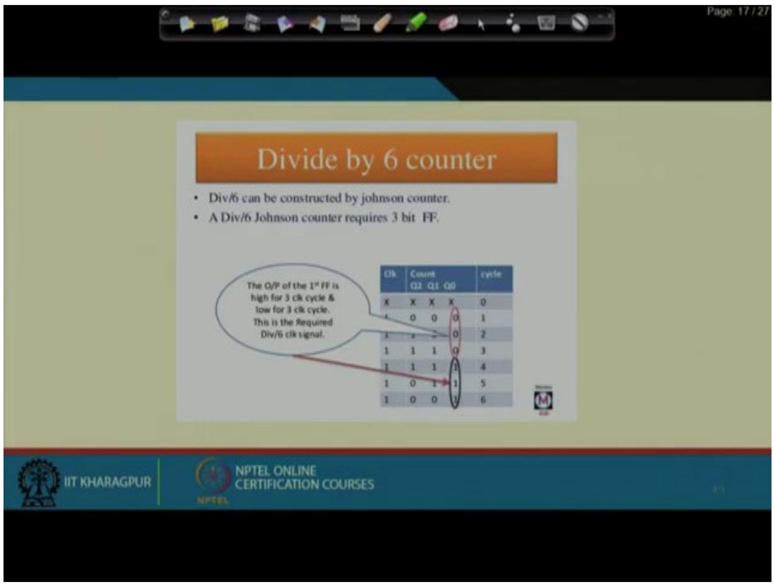
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 61
Design Tips for Basic Circuits Design (Contd.)

Hello everyone, welcome to the course on Architectural Design of ICs. So, we are seeing various clock divider circuits. So, we have already seen the architecture for clock divider where the deviation; that means, the division factor is in the power of 2 and if it is not in the power of 2 then also with the 50 percent duty cycle that also we have seen; that means, the divide division by 3, division by 5; so, that we have already seen. Then we will see some other actually clock divider circuit which will be done using the hardware.

And whenever we are seeing at the time we have also seen that at the very last to make the 50 percent duty cycle we have to put some extra OR gate to maintain, otherwise if we do not use that particular logic the generated this waveform that will not be of 50 percent duty cycle. So, to make it 50 percent duty cycle, we have to use this extra logic circuitry to maintain the 50 percent duty cycle though it should be divide by 3 or divide by 5. Now today we will see then again another example which is divide by 6 counter.

(Refer Slide Time: 01:43)



Divide by 6 counter

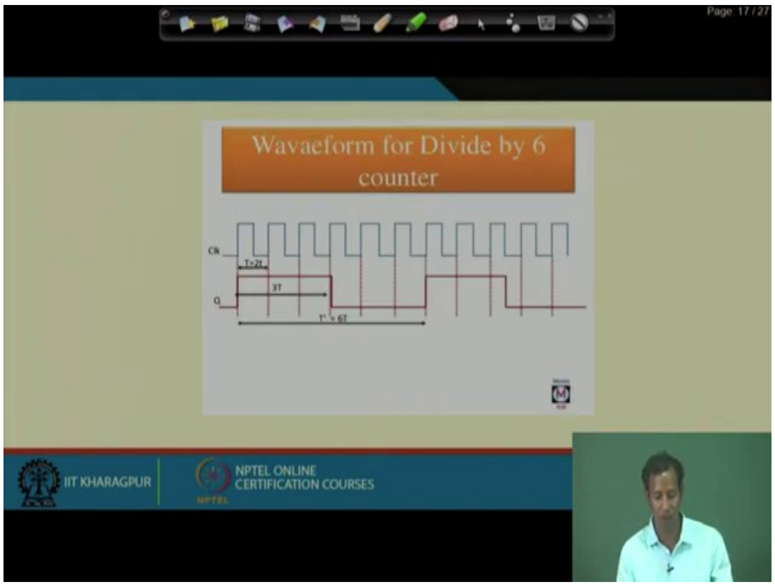
- Div/6 can be constructed by johnson counter.
- A Div/6 Johnson counter requires 3 bit FF.

Clk	Count Q2	Q1	Q0	cycle
X	X	X	X	0
1	0	0	0	1
1	1	0	0	2
1	1	1	0	3
1	1	1	1	4
1	0	1	1	5
1	0	0	1	6

The O/P of the 1st FF is high for 3 clk cycle & low for 3 clk cycle. This is the Required Div/6 clk signal.

So, in divide by 6 counter; actually, we need 3 different flip flops which will count and here I need one this 3 bit Johnson counter so, something like this.

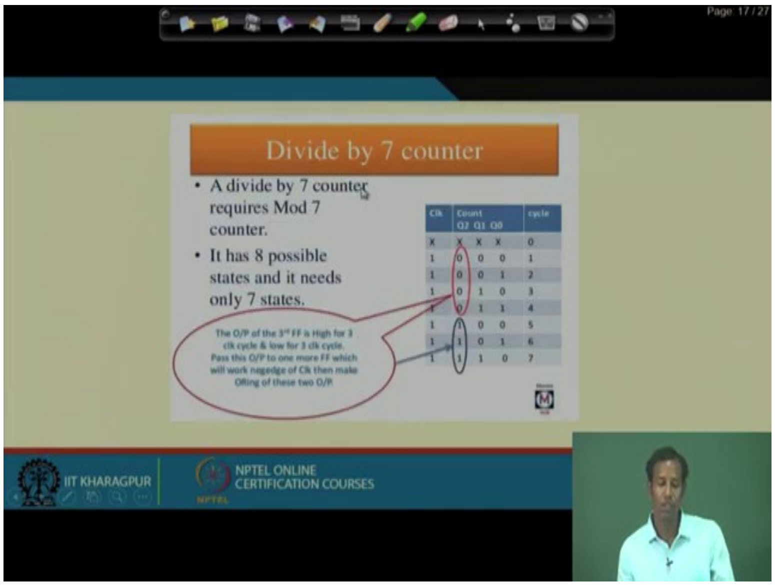
(Refer Slide Time: 01:51)



Wavaeform for Divide by 6 counter

So, in Johnson counter if you are having so, at the time; that means this if you design one this 3 bit Johnson counter that will be act as perfectly divide by 6 clock divider circuit ok.

(Refer Slide Time: 02:14)



Divide by 7 counter

- A divide by 7 counter requires Mod 7 counter.
- It has 8 possible states and it needs only 7 states.

Clk	Count Q2	Q1	Q0	cycle
X	X	X	X	0
1	0	0	0	1
1	0	0	1	2
1	0	1	0	3
1	0	1	1	4
1	1	0	0	5
1	1	0	1	6
1	1	1	0	7

The O/P of the 3rd FF is High for 3 clk cycle & low for 3 clk cycle. Pass this O/P to one more FF which will work negedge of Clk then make ORing of these two O/P.

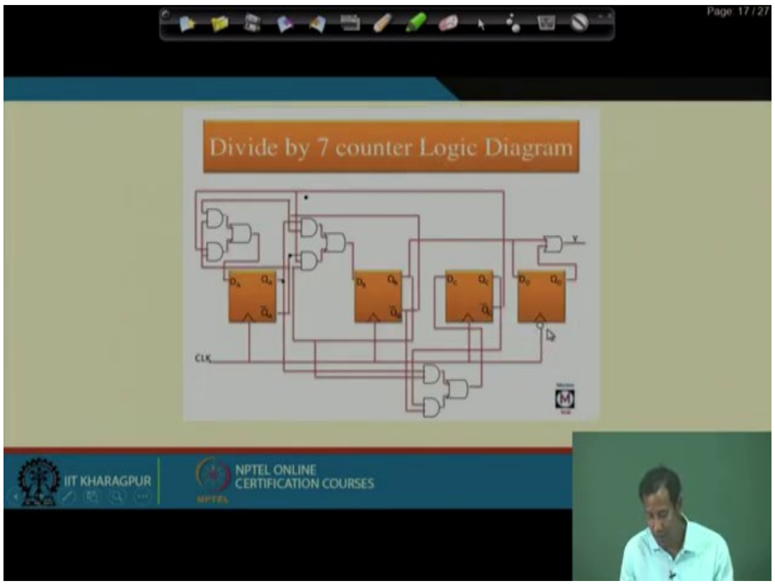
Now actually, if we divide by 7 counter, so, divide by 7 counter, it requires one Mod 7 counter and it has possible 8 possible states which by which it needs only it; that means, the 7 state is basically useful ok.

So, for this particular actually divide by 7. So, divide by 6 means; only you need 3 bit Johnson counter, 3 bit Johnson counter will give you the divide by 6 counter or divide by 6 clock divider circuit with 50 percent duty cycles. You do not need the additional adder or; that means, you do not need to design 1 Mod 6 counter and then other logic you do not have to do using 1 3 bit Johnson counter you can do; that means make the circuit of divide by 6.

So, to make this divide by 7 counter where you this 7 particular state will be used and here you see that this for 0 0 0 the output of the third flip flop is high for 3 clock cycles and low for 3 clock cycles.

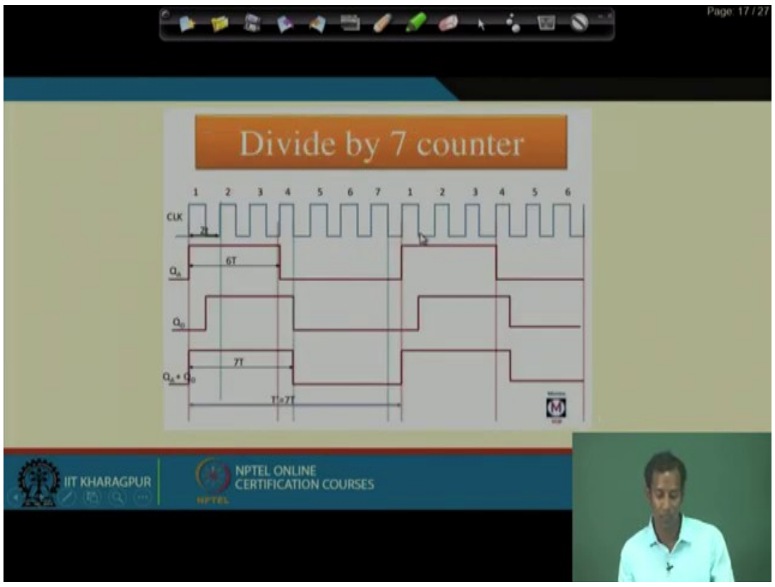
Pass this output one more flip flops which will work negedge of the clock then make ORing of these two outputs.

(Refer Slide Time: 03:44)


Divide by 7 counter Logic Diagram

Indicates what? Actually, this is the logic for this Mod; that means this Mod 7 counter and then again just like this the corresponding outputs from this Q B they are basically connected with the Y to make 50 percent duty cycles.

(Refer Slide Time: 04:09)



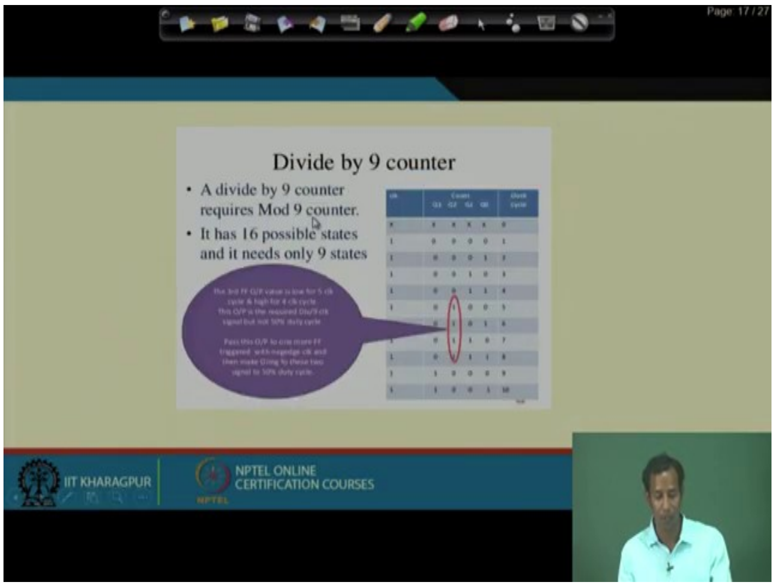
So, here you see that Q A, Q A is this one. So, this is this Q A the corresponding logic for; that means, the waveform for this Q A is this, then we need what? We need this Q D, Q D is nothing but, this Q B. So, Q B and then again we will get something like this. If we just add these two, then we will get the corresponding Y output which is nothing but division by 7 factor ok.

So, now here you see how you can measure that 1, 2, 3, 4, 5, 6, 7. So, up to 7 clock pulse, now your output is basically extended with the duty cycle of 50 percent. And everywhere every of this particular this on the whether you are designing this divide by 7 or divide by 5 or divide by 3, any of this circuit, here you consider that the final flip flop they are of negative edge triggered.

If all these flip flops are positive edge triggered, this is of negative edge trigger and as this is of negative edge trigger; because of that we are basically getting this divide by 7, divide by 5, divide by 3, something like that otherwise we could not have get this divide by 7 and that is why you see in this waveform it is at the corresponding negative edge up to this negative edge it is getting extended.

So, this Q A and Q D; if you see there is this Q D it is getting the output because this particular flip flop is negative edge triggered. So, that is why whenever it; that means, finds this negative edge then only it is changes its value ok. So, here also by that by using this negative edge triggered and then using this extra OR gates we can make this happen that we can make this divide by 7 with 50 percent duty cycle ok.

(Refer Slide Time: 06:38)



Divide by 9 counter

- A divide by 9 counter requires Mod 9 counter.
- It has 16 possible states and it needs only 9 states

The 3rd FF O/P value is low for 5 clk cycle & high for 4 clk cycle
This O/P is the required Div/9 clk signal but not 50% duty cycle

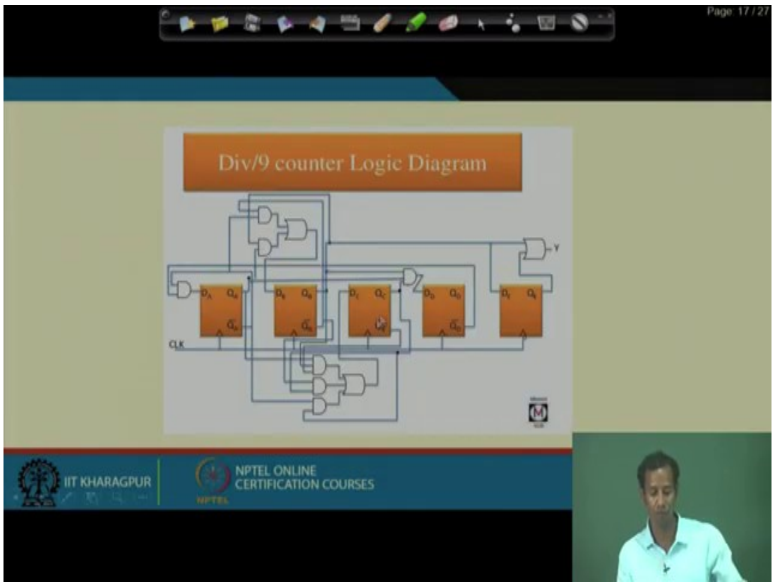
Pass this O/P to one more FF triggered with negedge clk and then make Oring to these two signal to 50% duty cycle.

clk	Count				Clock cycle
	Q3	Q2	Q1	Q0	
x	x	x	x	x	0
1	0	0	0	0	1
1	0	0	0	1	2
1	0	0	1	0	3
1	0	0	1	1	4
1	0	1	0	0	5
1	0	1	0	1	6
1	0	1	1	0	7
1	0	1	1	1	8
1	1	0	0	0	9
1	1	0	0	1	10

So, then again another example that I need one counter which will be divide the clock by 9 ok. So, at that 9 we will be having total 16 possible states among them only the 9 state will be used ok. So, here you can actually observe that the third flip flop output value is low for 5 clock cycles and high for 4 clock cycles. The output is required divide by 9 clock signals, but not 50 percent duty cycle.

So, to make this 50 percent duty cycle pass this output to one more flip flop triggered with negative clock edge and then make ORing to these two signals to make it 50 percent duty cycle ok. So, that is why how many to make this divide by 9, how many flip flops I require? 5. 4 for this and 1 extra to make that output of 50 percent duty cycle.

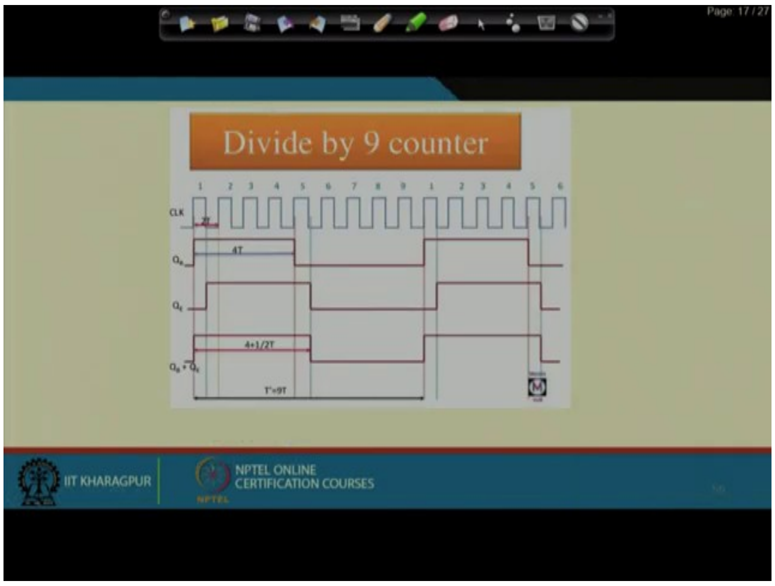
(Refer Slide Time: 07:51)



Div/9 counter Logic Diagram

So, if you see that 4 of this divide by 9 logic and then the corresponding output is now OR with this Q B which is coming over here and then if you just OR you will get the corresponding output sequence something like this.

(Refer Slide Time: 08:12)



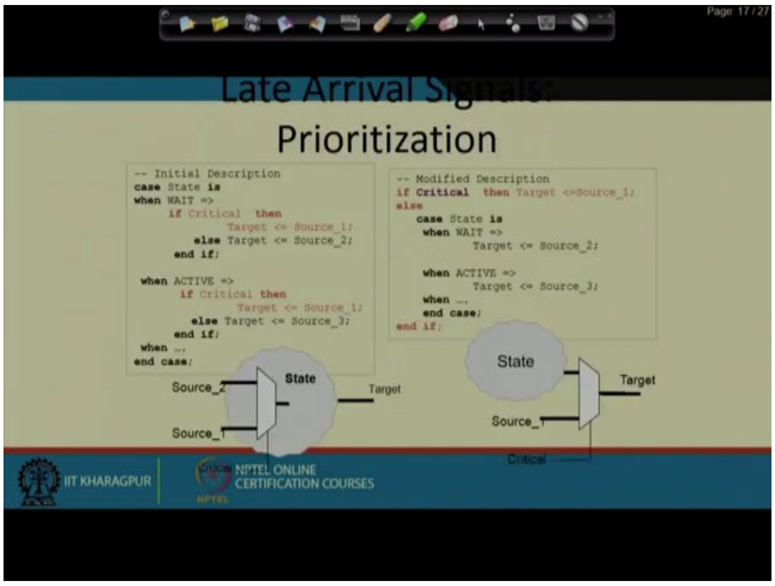
Divide by 9 counter

So, that means; here the Q B is this one, Q B you will get this one, then depending on that what will be the Q E sorry this D E or this Q E, Q E will be depending on the corresponding that negative edge triggered values. So, in negative edge it is find that 1. So, it will be extended up to this then again it is just extended.

So, here you will get that 1, 2, 3, 4, 5, 6, 7, 8, 9. So, up to 9, this particular Q B plus Q E is extended. So, now, if I ask you to design one divide by 10 counter. So, divide by 10 counter you require again the Johnson counter with the terminal count of this 9 and only of 4 bit Johnson counter with the terminal count of 9. So, whenever it will find that the it is counting from 0 to 9 after that again it will come down to 0. So, which will make you the divide by 10 counter.

So, now again divide by 11 counter if you want to make then you again you have to consider something like this. So, there will be sixteen possible state, but among them only 11 state will be used. So, based on that you find out these particular logic to make that Mod 11 counter and then you put this extra logic to make that; that means, 50 percent duty cycle. So, at this output this logic you can put to make 50 percent duty cycle. So, then by this way you can design or you can; that means, develop any clock division circuit or the multiplier circuit also we have already seen.

(Refer Slide Time: 10:34)



Late Arrival Signals: Prioritization

```
-- Initial Description
case State is
when WAIT =>
     if Critical  then
              Target <= Source_1;
          else Target <= Source_2;
       end if;

 when ACTIVE =>
       if Critical then
              Target <= Source_1;
          else Target <= Source_3;
       end if;
 when ...
end case;
```

```
-- Modified Description
if Critical  then Target <=Source_1;
else
   case State is
    when WAIT =>
           Target <= Source_2;

    when ACTIVE =>
           Target <= Source_3;
    when ...
    end case;
end if;
```

Now, we will see another; that means, different example is that this is one of the techniques which we apply whenever we use or this is a tips or that this is a tricks what we use if we find some problems whenever we declared the corresponding functions, declare the corresponding function means; we suppose this one is the code for one particular logic ok.

So, what is this code says that if, this is the function state the logic says that when wait is activated the signal wait is activated then, if critical is another signal then target will be goes to; that means, the source 1 will be goes to the target otherwise if the; that means, the net critical is not active. So, at the time source 2 will be assigned to the target.

So, this logic will happen when that state is wait state. When that is in the active state, so, at that time the critical if the critical is another actually signals when the corresponding signal critical is 1, then target will be source 1 otherwise this target will be from source 3 ok.

So, now depending on this actually what I need is that; so, the state and I need the signal which is this critical ok. So, when this wait state if the critical signal is 1 then it will select the source 1 if it is 0 so; that means, if the critical signal is not active so, at that time it will select this source 2 which will go to the target. But, when in active state this will be source 1 and source 3 and this will be the critical right.

Now, as we can see that from both of these particular things, both of these; that means, I need two of this; one for active state another for wait state, where this source 1 is fixed is common to each of this state and only the other terminal which is the source 2; that means, the here for this wait state that is different for active state which is this source 3.

So, to make this particular logic; that means, in a single code so, how we can modify is that if critical is 1 then all the time with irrespective of the wait state or active state whenever this critical signal is active the target basically is assigned to source 1 and if the critical net or the critical signal is not active at that time that in wait state the target will be in source 2, in active state the target will be come down to source 3.

So, what we can do? We can have only once multiplexer where this critical is the select line and the source 1 is both common for wait state and this active state. So, the state will be the other select it goes to the other select line in; that means other input when this critical net is 0. So, at the time it will select depending on the state whether that is wait or active it will select the source 2 or source 3 to the circuit.

So, this is one of the examples; that means, if we; that means, if this is the initial description of the code and based on that we can make the corresponding circuit. But if we modify the same code in the other way at that time that whenever this critical is, this indicates what? We have to wait to check which state it is.

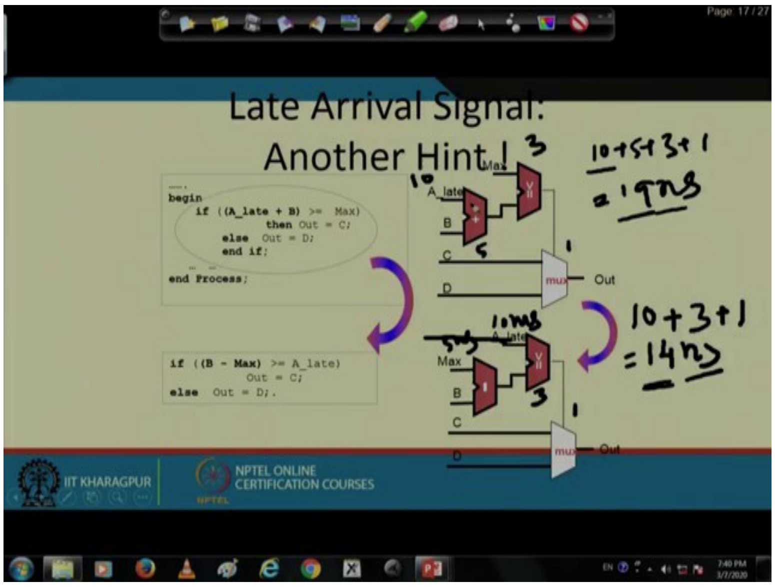
But here we do not have to check when that net signal is that critical is active immediately the target will be assign to source 1, where in these particular case we have to check the whether the critical is 1 or 0 depending on the state. First we have to check the state which state it is whether that is in active state or whether that is in wait state, then we have to calculate we have to; that means, check whether that critical net is active or the critical net is not active. If it is active then select source 1, if it is not active then select source 2.

In wait state; if sorry in active state if this critical is 1 then source 1 or else source 3, but here in this case we do not have to consider the state for the critical condition whenever the critical condition is on. So, automatically the target will be assigned to the value of source 1. And when the critical is not active then only I will check whether that state is 0 or whether; that means, the state is active or whether the state is wait.

So, depending on that then I will assign that target value; that means, at the time I do not need immediate action. So, that is why here you actually does this trick is known as this late arriving signals prioritization ok. So, the priority is being given to the corresponding this particular critical net irrespective of what state it is at this the corresponding logic demands the critical the priority instruction to the critical net ok.

So, this whenever we will design the circuit, at that time this small modification will make your design so, much efficient ok.

(Refer Slide Time: 17:49)



Late Arrival Signal: Another Hint !

```
......
begin
    if ((A_late + B) >=  Max)
            then Out = C;
        else  Out = D;
        end if;
    ...  ...
end Process;
```

```
if  ((B - Max) >= A_late)
            Out = C;
else   Out = D;.
```

Then, another see another example we will see this late arrival signals. So, late arrival signal means; suppose, one logic is this that begin if A late plus B greater than equals to max then out equals to C else out equals to D. So, A late means; this is the signals which is coming from another logic and it is coming very lately to this particular port.

So, according to the logic what we have to do? A plus A, A plus B we have to check that is whether that is greater than equals to max, then the output of that will decide that we have to select C line or D line to the output. So, then what will happen? What is the problem in this particular circuit is that as already this A late signal is coming late.

Let us say this A late signal is coming late by 10 nanosecond ok. So, then again if this particular block is having 5 nanoseconds of delay and these block is having 3 nanoseconds of delay and this multiplexer again is having 1 nanoseconds of delay. So, then total after 10 plus 5, 15 plus 3, 18 plus 1. So, total 19 nanoseconds I will get the output at this particular out.

So, irrespective of doing that what I can make is that; I will just change this logic to B minus max greater than equals to A late then Out equals to C else Out equals to D; that means, keeping the logic same. I have just; that means, shift this max to this a late to this particular side. So, then what is happening? Initially we have to calculate with this B minus max and then we have to calculate with this A late signals whether that is greater than equals to and depending on that this will generate the corresponding signals which will decide that the output will go to C or D among this C and D what value will go to Out.

So, in this particular case; what we have considered? So, this signal is coming let us say 10 nanosecond so, this is having 5 nanosecond this is of 3 nanosecond this is taking 1. So, total is 10 plus 5 plus 3 plus 1 so, total 19 nanoseconds, it requires to complete the full operation. Why because unless and until this particular signal is arrived I cannot starts it is operation. So, once this operation is not yet complete I cannot start this operation. One this particular operation is not yet complete I cannot start this operation.

So, that is why the total summation of all these paths delay information will be the maximum computation time for this particular circuit whereas, by making this small changes in the logic. Now, we what if I say that the adder and the subtractor will be having the same delay let us say this is of 5 nanosecond and now this is coming at 10 nanosecond this is of 3 nanosecond this is of 1.

So, now, whenever this 5 is already coming 10 is also parallel which is coming from others other signals. So, at that time the max of these two path will be consider here what is happening? 10 plus 5, but here among of this 10 and 5, which one is the maximum one? So, 10. So, it we will consider 10 is the maximum delay for computing these two particular path plus then again to; that means, complete this task I need 3 plus 1 means total of 14 nanosecond in these particular case.

So; that means, this the logic remain same. In this case I need 19 nanoseconds to complete the full task, but in this case I need only 14 nanoseconds to complete the full task ok. As because I made this corresponding modification to make this two; that means, while these particular signal is not coming by that time I am making I am completing the corresponding task over here. I am not waiting for the signal which I am waiting here in this particular example ok.

So, this is 1 of another tricks which we what actually which we use in case of optimizing or to improve the performance of our circuit ok. So, it is not that only these are the tips and tricks which are there. So, it depends upon you while you are designing. So, at the time how; that means, depending on this whether you have to do the priority selections or if one signal is already coming late so, you have to do some kind of this modification. So, everything depends upon you how you; that means, how you can you tackle the problem to get one circuit or to design one circuit which will be much more; that means, the performance of the circuit will be much more on the enhance side ok.

So, this is some of the; that means, tips and tricks with what we have already which we use in case of digital circuit design or digital integrated circuit design ok. So, it is not that there this is enough. So, there are so many if you need more information then please let me know through discussion forum I will provide you more information or I will provide with the documents where you can find more on this. And people are basically doing research everyday to improve the performance day by day on this while they are doing this hardware design of digital integrated circuits. So, with that thank you for today again we will see and; that means, some other topic on this architectural design of ICs.