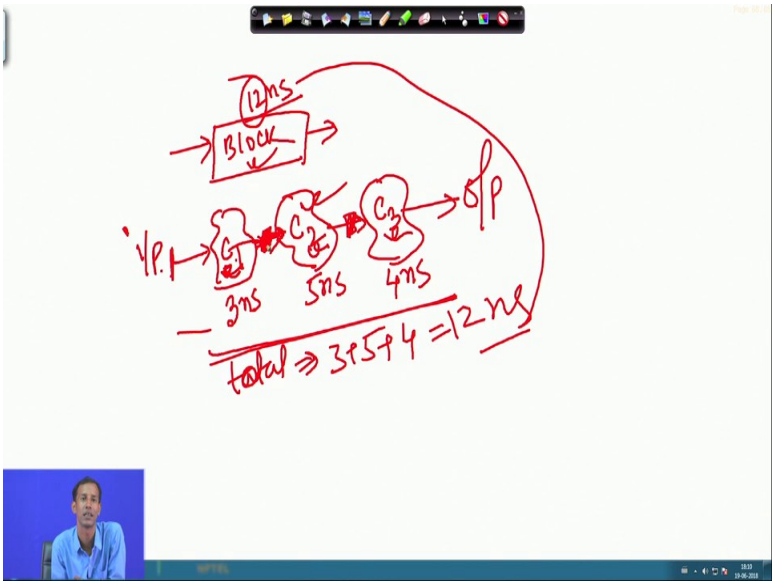
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 26
Pipelining and Parallel Processing (Contd.)

So, welcome back, again to the course on Architectural Design of IC's. So, we are discussing about the techniques for increasing the operating frequency of any circuit when we have done or we have completed or after completion of the all the architectural optimization or algorithmic optimization; then there is a again there is a another way to increase the operating speed frequency or the sampling frequency of that particular circuit. So, for that we have employed this technique which is named as pipelined and again I need more improve that means, increasing operating speed I have to employ the another technique which is parallel processing, ok.

So, we have learnt the basic ideas of that what is the meaning of this pipelining technique and the parallel processing. Here, today again we will take one of the example and we will see how basically this helps to reducing in the in real time how it is basically helping me to increase a corresponding speed of the system or the; that means, the sampling frequency of the operating speed of the system, ok.

(Refer Slide Time: 01:37)



So, now let us consider we will take one example, and consider one particular block, where I am having let us consider initially I am having one suppose this is one block which is basically taking 12 nanoseconds of delay, ok. So, this is the critical path of this particular block. So, this is black box. So, I do not know what functionality as of now. Now, how to, that means, after all this architectural optimization I have done inside this black box. So, now, how can I again improve the; that means, how can I reduce this particular 12 nanosecond, so that I can achieve the; that means, that performance on the higher side, ok.

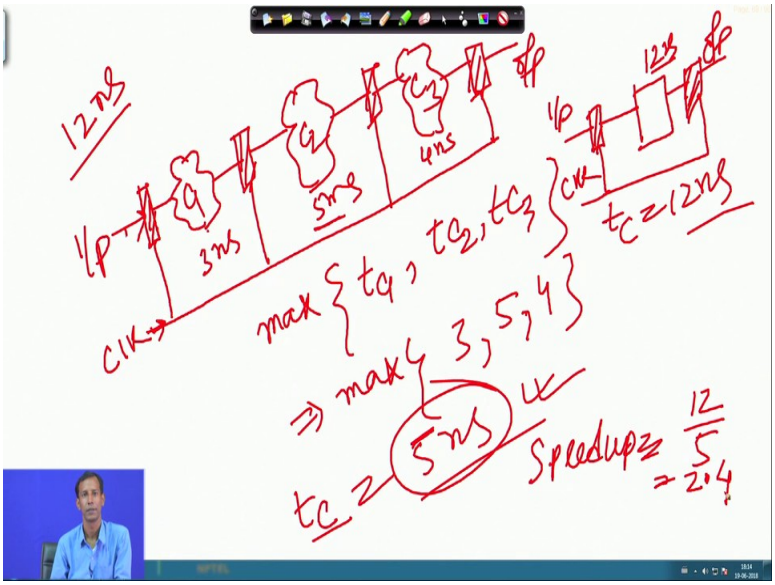
So, this after that means, analyzing or after observing this black box what I found that there is let us consider 3 combinational blocks they are connected. So, this is combinational block 1, this is combinational block 2, this is combinational block 3 and let us consider this takes 3 nanoseconds of time, this takes 5 nanoseconds of time and this takes 4 nanoseconds of time for the computation and as these are connected to each other so, that is why it is taking from the input this is the input and this is the output. So, from the input to the output the time total time is taken as sorry total time is taken as 3 plus 5 plus 4, which is equals to 12 nanoseconds which is the delay for these particular blocks.

Now, what I said in the pipelining technique we will try to put the register in between this blocks ok. So, to make this particular here what is happening there is a data dependency because this C 2 is processing the data from C 1. This C 3 is basically processing the data from the C 2. Now, I will if I put the register in this path and this path at that time this will work as individual, this will work as individual, this will work as individual. How this will work as individual? There is a data dependency, but at that time the register will hold the value for one particular clock.

So, that means, once this is done then this will be stored at this particular level; that means, register the previous value of this C 1 that will be now processed in C 2 and then again this register will hold the value corresponding value and C 3 will consider the previous holded value of C 2. So, that means, that in if I have not put this register. So, at that time what is happening instead of C 2 is giving the that means, output producing the output immediately there is a change in the C 2, whenever C 2 is producing the data C 3 is immediately processed. So, that is why the total; that means, time is basically added to 12 nanoseconds.

But, as I have put register there it is not that though C 1 has processed the output or how much time it is taking to process that C 1 that means, for this C 1 process it is not at all dependent or it is not effecting the corresponding C 2 it is basically effecting what the previously holded value for C 1 which is stored in this particular register. So, that value is now basically coming to C 2 and that is changing the value of C 2, ok. So, that is why the data dependency is basically broken by putting this extra register in between this particular blocks.

(Refer Slide Time: 06:00)



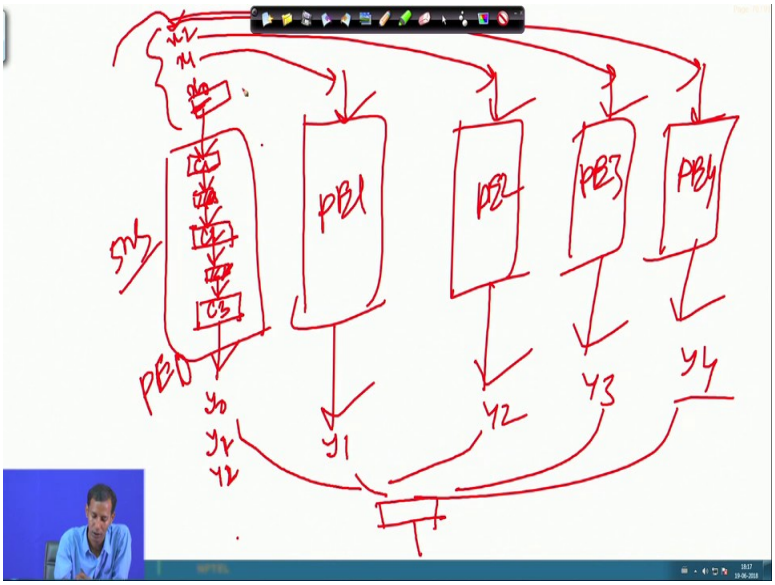
So, at the time now for if I just suppose then I am having the blocks something like this let us let me draw the block fast they can even we will discuss this will be connected to this particular register and this is nothing, but the clock this is the input and this is the output ok. So, this is C 1, this is C 2, this is C 3. So, this is taking 3 nanosecond, this is taking 5 nanosecond and this is taking 4 nanosecond, ok.

So, I have put two registers in between now what will be the corresponding critical path of this system now; that means, what will be the minimum clock period which I can consider for this particular case, what will be the clock period, what clock period I will set over here. So, the minimum clock period which I set that is this max of the delay of t c 1, t c 2 and t c 3. So, that is max of 3, 5, and 4 which is 5 nanoseconds. So, that means, initial requirement was 12 nanoseconds, but after putting these registers in between this

path now I have come down to; that means, after doing the pipelining now my clock period this t c becomes 5 nanoseconds.

So, initial circuit was something like this at this particular point I am having this register. So, here I have having a register. So, this is the input and this is the output, and this is the clock. So, this is requiring 12 nanoseconds, so, that means, the clock period requirement for this particular case that was 12 nanoseconds, but now it becomes to 5 nanoseconds. So, then I can say that the speedup is basically 12 by 5 that is 2.4 over here after doing the pipelining. Now, I want to increase the sampling speed or I just come down; that means, my requirement of the sampling speed is 1 nanosecond. So, at the time what I will do?.

(Refer Slide Time: 09:27)



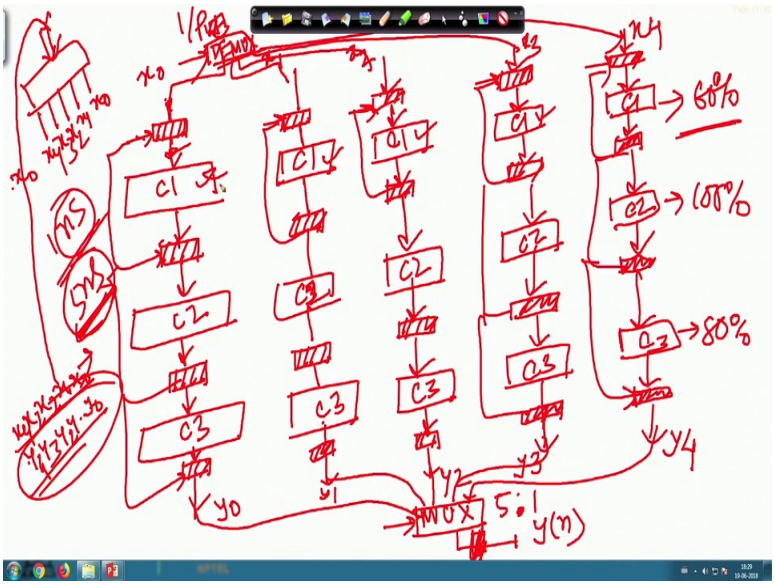
Then what I can do; that means, at that time if I that means, if I consider this is as C 1, this is as C 2 and this is as C 3 and there I am having this registers let me draw first. So, this is my C 1, this is my C 2 and this is my C 3, ok. So, there I have drawn it in; that means, horizontal, here I have drawn it in vertical.

So; that means, the input x 0, then x 1, x 2 it is basically coming in this manner to the corresponding pipelining things and I am getting the y 0, y 1, y 2 dot dot dot in this particular case. But, what I said my requirement is so, here that corresponding delay requirement is 5 nanosecond; correct. If I need the clock speed to come down to 1 nanosecond so, at the time what I have to do? I have to imply these parallel processing

levels of 5. That means, now if I process this if this is known as one of the processing element. So, this will be copied 5 times ok, this will be copied to 5 times, and then what will happen this x 0 will come over here, x 1 will come over here, x 2 will come over here, x 3 will come over here and x 5 will come over here and it will produce y 1, y 2, y 3, y 4, ok.

So, at that time; that means, this if; that means, if I need to process or if I have to channelize these particular samples to the corresponding processing element this is PE 1, this is PE 0, sorry this is 0, this is PE 1, this is PE 2, this is PE 3 and this is PE 4. And I need to channelize this proper that means, input to the corresponding processing element and again I need to combine this particular output to the single output.

(Refer Slide Time: 12:21)



So, that means, here at these particular things what will happen so, if I just draw this particular circuit. So, at the time how it will look like, ok so, it will be something like this. So, then again I need here this register then again I need to draw C 2 then again this registers I need and then again I need C 3 and here I need the registers, ok. So, this will be coming to this, this will be coming to this, this will be coming to this, this will be coming to this, this will be coming to this, this will be coming and this is the output, ok. So, this is the input is basically coming to this particular point.

Now, what I have to do I have to copy this for 5 times, right, these particular blocks as I have employed the parallel processing level of 5, ok. So, this is C 1, this is C 2, this is C

3, correct. Now, for this again I need to put the extra register over here and from here I will get y 0, from here I will get y 1, here this will be giving x 0, this will be giving x 1, then again for this again I need to draw the corresponding circuit which are nothing, but this; that means, I am just copying this particular circuit with the one processing element for multiple times, correct.

So, then again coming to this so, this is for y 2 and this is for x 2 then again for the fourth number of processing element again I need to do the same thing. So, what I have to do? I have to connect this. So, this is for x 3 and this is for y 3 and at the very end; that means this is the fifth element or the fifth processing element here what I have to do is that. So, this is for x 4, this is for y 4, and all are this is C 1, this is C 2, this is C 3 sorry, this is C 2 again C 1, C 2, C 3; C 1, C 2, C 3; C 1 C 2, C 3 and what I said actually my data they are coming like this x 0, x they are coming like this x 0, x 1, x 2, x 3, x 4 and I need the output will come like this y 0, then y 1, y 2, y 3, y 4 something like this; that means, in the serial manner the input the flows of the output and input are in a serial fashion, ok.

So, whenever I will put the serial input to the corresponding parallel processing block. So, I need one serial to parallel converter at the input of this. So, what circuit is serial to parallel converter? So, serial to parallel converter is a nothing, but the de-multiplexer where it; that means, takes the input one input and it produce several outputs and in the same that means, one input it takes and it produce several outputs. So, that that is why I need so, serial input is this one. So, now, this will come and it will produce x 0 over here, x 1 over here, x 2 over here, x 3 over here and x 4 over here, ok.

So, that means, now I need the de-multiplexer block from where all these will be connected and the input is basically which is the; that means, input which is coming this is the de-multiplexer, this is the de-multiplexer block at the input side which I require and in the same; that means, in the opposite manner what I require, here am producing the y 0, y 1, y 2, y 3, y 4 parallelly, but I need the output in a serial manner.

So, that means, in this case what circuit is basically parallel to serial converter I require. So, multiplexer is basically does the same. It takes several inputs and it produces one particular output, ok. So, that means, now it will take more; that means, at this particular case it will require one MUX at the output where each of this output will be connected to this blocks and it will produce one which is y were y n ok. So, that means, now it will be

5 is to one multiplexer, and the select lines are something like this, ok. So, this is the corresponding circuit for.

So, now, the clock period of this; that means, the this particular sampling speed of these the input sample speed or this output sample speed of that means, whenever I need to have produce this. So, at the time it will be of 1 nanosecond; that means, here actually after that this I require one register which will produce the y n. So, this the register not if I do not consider this then the register for this input and the registers which I require for the output case both are requiring 1 nanosecond of sampling speed. The internal of this that will be different, and this clock will be the they will be connected and this block is basically they are all connected to the same clock and this clock will be the 5 nanosecond which is the requirement of processing elements critical path the requirement for this clocks of this register so, that will be of 5 nanosecond.

So, if I that means, say; that means, the clock period of this is 5 these are 5 nanoseconds to each of these particular register so, at the time you see within 3 nanoseconds I have computed completed the job over here. But rest of the 2 nanosecond it is just remain this block is basically remain idle sitting idle where I require at these particular C 2 block. Basically, I need 5 nanosecond for computation of this C 2 block in each of this case and again if I that means, come down to C 3 block, it computes for 4 nanosecond, but it has to wait or it is just sitting idle for another extra one nanosecond of clocks, ok.

So, that means, there are redundant computation or the blocks which are basically sitting idle. So, that means, working ah; that means, percentage of this C 1 is 60 percent ok, this is 100 percent ; that means, I have all time at the; that means, the clock frequency is 5 so; that means, these blocks are running at 100 percent rate, but this is the working at 8 percent rate. Why, because this blocks are basically a sitting idle for 1 nanosecond delay where the clock period requirement is 5 nanosecond delay. This basically is; that means, is sitting idle for 2 nanosecond if delay as the clock period is 5 nanosecond.

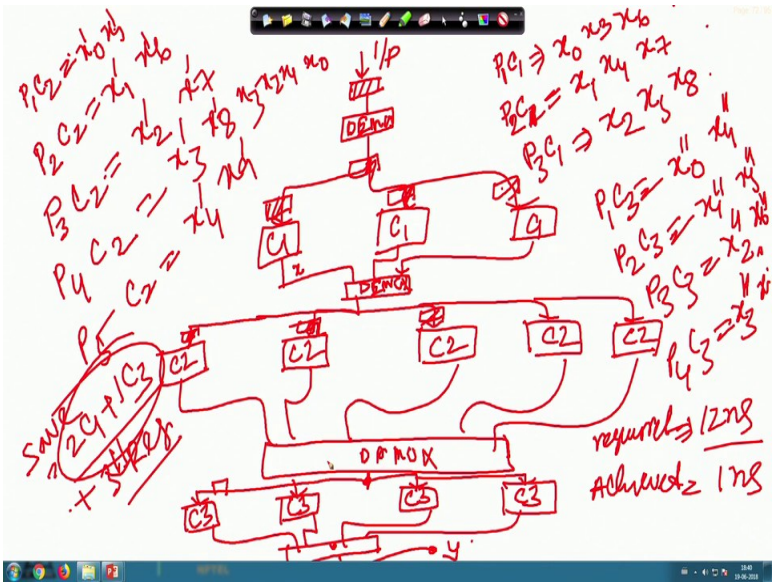
So, that means, the working the corresponding this efficiency of this C 1 block that is 60 percent efficiency for C 2 that is 100 percent and efficiency for this working efficiency there is C 3 that is 80 percent. So, from these observing these particular point so, can I reduce these particular circuit; that means, can I that means, whenever I am employing these particular elements; that means, it consumes the area if it is sitting idle I am ah; that

means, what I have employed that all the blocks are not utilized or the utilizing efficiency that are not hundred percent in each of this stages or each of this blocks that is C 1, C 2, C 3; that means, I have put redundant block to the circuit.

So, redundant I have increased the circuit or; that means, the in operating frequency by putting what redundant circuit to the system, but whenever I have put the redundant circuit as that is not utilization of that means, efficiency is not 100 percent so, that means, I am basically loosing unnecessarily I have put those blocks which is basically costing me more area and more power too. So, can I reduce that? So, that means, if I want to reduce that.

So, at the time what will happen I need to that means so, that these particular register will run at 3 nanosecond this particular ah; that means, this register will run at 5 nanosecond this particular register will run at 4 nanosecond, ok. So, that means, at the time I need this skewed clock which will be connected to this and at the time if I actually I have just I am just coming down to this that particular circuit.

(Refer Slide Time: 24:16)



So, at the time what is happening basically if I just consider withdraw the corresponding circuit. So, at the time after that I have one de-multiplexer. So, this is the input sorry this is the input then I need 3 of the corresponding C 1, C 1, C 1 block then this will be distributed to ok so, not this one this will not be directly connected sorry. So, this will be again that means, connected with one of the blocks which are basically again it needs

one de-multiplexer where it is taking this and it is producing the corresponding this particular path. And this is C 2, this is C 2, all are C 2 then again this will be connected to one block which is basically connected to the corresponding this is C 3, C 3, C 3, C 3. And this will be, if I just connect to this will be connected to this, this will be connected to this then again this I need to one multiplexer. So, this will produce one single output which is nothing, but your y, ok.

So, that means, what; actually I have not put those registers over here, ok. So, in between sorry in between this particular case the register will be placed over here. So, there also this registers will be placed. So, at the input side of these blocks the registers will be here; that means, placed. So, that means, x 0, x 1, x 2, x 3 they are coming something like this then what will be the; that means, corresponding processing table for this? If I consider the processing table for whole of this system so, at that time for C 1 for C 1, for suppose this is the PC 1 then it will be connected with x 0 then x 1 for PC 2 that will be connected with x 1 PC sorry PC this is this is P 1 C 1, this is P 2 C 2, this is P 3 C 1, it will be connected with x 2.

Then in the next cycle again it will repeat and x 3 will come over here, x 4 will come over here, x 5 will come over here, then again x 6 will be connected over here, x 7 will be connected, x 8 will be connected over here and it will run in this fashion; that means, within from this particular de-multiplexer block. So, all these x 0, x 3, x 6 that will be coming to this particular processing blocks at this particular point this x 1, x 4, x 7 that will be connected. And for this P 3 C 1 this x 2, x 5, x 8 that samples; will come for at these particular processing blocks, ok. Then again that will be this 3 input now again it will be connected to one de-multiplexer ok, which will be again connected to the 5 of the C 2 block ok. So, 5 of the C 2 block means then the corresponding if I consider P 1 C 2 ok.

So, P 1 C 2 that it will consider if this is known as sorry that means, the corresponding output from that then again that will be connected. So, how that will be? So, it will be x dash 0 in P 2 C 2, P 2 it will be x 1 dash x 1 dash is the output from this de-multiplexer, correct. So, then P 2 C P 3 C 2 that will be connected to x 2 dash then P 4 C 2 that will be x 3 dash and then P 1 5 C 2 that will be connect to x 3 x 4 dash. Then again x 5 will come over here, x 6 will come over here, x 7 at this particular point, x 8, x 9 something like this, and these will run this is for this particular case. Again, that will produce 5 then

that will be connected via this after passes through the de-multiplexer again the again that will be connected will all these four C 3 block, ok.

So, then for C 3 block that will be corresponding for P 1 C 3 it will be as x double dash 0 for P 2 C 3 that will be x 1 double dash for P 3 C 3 it will be x 2 double dash for P 4 C 3 it will be connected as x 3 double dash; double dash is the output from these de-multiplexer, ok. So, then x 4 will come over here, x 5 will come over here, x 6 will come over here and x 7 will come over here and in this manner for this C 2 this x 0 then x 5; then x 9 this will be connected and to this particular blocks.

That means, those samples will from this de-multiplexer that will be produced and this will come connected to this C 2. Then this will be connected x 1, x 6 then x 10 something like this then again this will be connected as x 2 sorry x 11 here. So, x 2, x 7 then x 12 for here this is x 3, x 8 then again with 5 that is x 13 so, x 4, x 9 then 14 x 15 x 14. So, again for these x 0, x 5, x 10, x 15 so, all these will be come all the samples will come for this; so, that means now if I just go back to these.

So, here what I require is that I may require 5C 1, 5C 2, 5C 3, but here in these particular case what I require I need 3C 1, 5C 2 and 4C 3 and here you see all the blocks are low there utilization efficiency is 100 percent; that means, all the time they are working. So, at that time I have said 2C 1 block plus 1C 3 block, ok. So, that means, I can reduce the area by this number reduce the area and I can also reduce the power too, ok.

So, initially we have started with what the requirement the clock period requirement for initial was black box was 12 nanosecond. Now, I have achieved that is equals to 1 nanosecond of clock period, and along with that this the blocks of 3C 1, 5C 2 and 4C 3 along with the registers are also I have just put extra registers also we have put; that means, 3 register for this 3 5 register for this 5 and 4 register for this; that means, here I have said 2C 1, 1C 3 plus total three registers too, ok.

So, I have; that means, this; that means, from this particular point what I can conclude is that I can using pipeline and I can; that means, using parallel processing I can improve the operating speed. And if actually this blocks the sub blocks there I all the delays of those sub blocks they are identical sat the time I do not have to follow this particular architecture I do not have to optimize. So, whatever circuit this circuit we have seen if each of this block is taking 5 nanosecond; there is no chance for further optimization. If

these blocks are different; that means, the delay for this C 1, C 2, C 3 they are different at the time there is a chance for optimization. And how we have done that we have done in this particular case considering these facts that means, what is the utilization efficiency of that we have find out and based on that we have just calculated those things, ok.

So, that means, by doing these two things together I can achieve the that means, operating frequency I can on the higher side as well as I can reduce the power to by putting the minimum number of redundant circuit to the system, ok. So, this is one of the techniques or tricks we use to increase the operating frequency of any circuit, where we have start; that means, after all the optimization where we have start still from that particular point I can increase or I can improve the frequency of that particular circuit, ok.

So, for today this is for today's class this is it, ok. So, in the next class again we will see actually we will start one new topic which is the that means, there are several multiplier architecture. So, we will start with multiplier architecture from the next class onwards.

So, thank you for today.