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

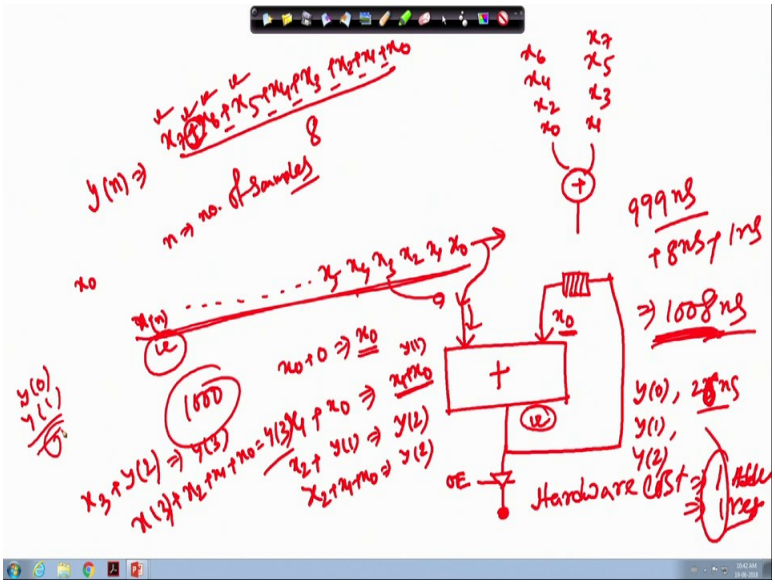
Lecture - 23
Efficient Adder Architecture (Contd.)

Hello everyone, welcome come back to the course on Architectural Design of ICS. So, in the last class we have seen the architecture for this running average circuit. So, we have discussed what is running average circuit ok.

First, we have started with what is the average circuit, then what is the meaning of running average circuit, then what is the parallel implementation of running average circuit, then what will be the computation time if I do the hardware for that, then what will be the computation time it will requires considering a; that means, fixed rate of number of samples. And then what will be the hardware requirement for that particular circuit also we have seen or we have discussed.

So, and at the end of that lecture we have said that now next we will follow or we will try to; that means, find out is there any alternate architecture. Then if I find any alternate architecture say at the time what will be the; that means, advantage or disadvantage of that. So, if I just again go back to the corresponding; that means circuit. So, in the last class what we have seen we have to find out.

(Refer Slide Time: 01:24)



This running average of this x 7, x 6, x 5, x 4, x 3, x 2, x 1 plus x 0 divided by 8 and what I said that the number of whenever running average means the numbers are coming like x 0, then x 1, x 2, x 3, x 4, x 5. So, dot dot dot up to xn values are coming, where n is the number of samples I consider and they are coming in this manner.

So, now actually that is in the in the last class we have discussed the; that means, this parallel architecture so; that means, has I am having this 8 numbers I have to add. So, we need require 7 number of adders then I have done that, so the serial implementation of that whenever we will do serial implementation. So, at that time means what at this particular operation there being common to each of this.

So; that means, at this operators is common. So, can I use these operators for performing the whole operation? So; that means, only 1 operator will be there and this will be; that means, the corresponding number of samples they will be time multiplexed ok. So, here what we are doing parallel implementation means all the samples I am putting to the corresponding particular circuit.

So, whenever I am putting 8 of these particular numbers to the circuit. So, at the time I will get after sometimes I will get the output right but, if I just put this because this operations are similar for each of this case. So, in time multiplex if I just put change the corresponding values of this; that means, I am having this particular operation. So, at the time initially it will be x 0 x 1 then at the next time it will be x 2, x 3, then x 4, x 5, x 6, x 7. So, something like this if I just add. So, at the time can I get that? Yes, I can get that right. So, that architecture is basically nothing but, the serial operation.

So, at that time how can I do? That means, now what will happen; that means, what I need is that I need one addition operation, where each one of the input they are off this particular input will come to this. And then one then again what will be the other input to these particular adders? They will be the, this is register.

So, this is the; that means, sum which is again coming to this particular block; that means, the output is again in a feedback they are connected to the adder. And what I require I need one additional; that means, output for getting the final output because what we are doing? We are doing this in a loop. And at the final whenever I will this circuit will says that the operation is over or we have seen in; that means, in the last few classes that whenever we are doing any computation in a loop.

So, at the time I need one breaking point; that means, this is the condition where this loop will be the execution of the loop will be over and the corresponding results will be gone to the output bus. So; that means, now I need one output enable signal whenever this will indicate that this is the number of. So, 1000 number of samples it consumes now you just produce the results to the output. So, now, how can I do that? Or that means, what is that; that means, things were that how this working of this particular circuit?

So, initially x 0 is coming so, as this is registered is cleared initially. So, y x 0 will be added with 0 at the very beginning. So, I am getting this is s at this particular point. So, this is at x 0. So, then; that means, now this is x 0 is coming over here.

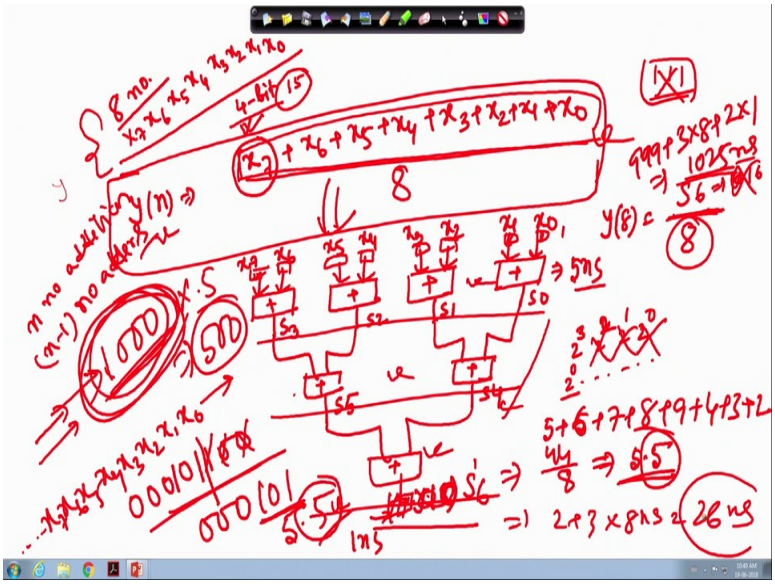
Now, next sample is I will here so, now, what we I am doing x 0 plus x 1 I am doing. So, then what is the output? x 1 plus x 0. So, now, this is the available at this particular register the way corresponding value for this x 1 plus x 0.

So, now in the next this x 2 is available over here so, now, x 2 will be added with what if this is y 1 then it is with y 1 that is y 2 which is nothing but, x 2 plus x 1 plus x 0 equals to y 2 ok. Then again in the next what will happen? Next in clock cycle what will happen? Now, x 3 will be available to this. So, now, x 3 will be added with y 2 ok, then x 3 plus that is equals to y 3 correct. So, now x 3 plus x 2 plus x 1 plus x 0 that is equals to y 3. So, something like this in this manner this that means loop will start counting or start adding in each of the clock.

So, in this particular case if I consider so at that time what will happen? So; that means, again to all computing all this particular samples what I require? I need to wait for 999 second for this. And what I require? I require this one addition operation and then again what is this 1 nanosecond right.

So, what is the time; that means, 1008 nanosecond I require so, I think.

(Refer Slide Time: 08:47)



So, here it will not be this will be the total computation time for this that will be 999 plus 3 into 8 plus 2 into 1 ok. So; that means, this will be 1000 and then this is 25 so, 1025. So, in the last I think I have mentioned a 1016. So, this is not the correct number so, 1025 nanosecond is the correct number for that particular cycle.

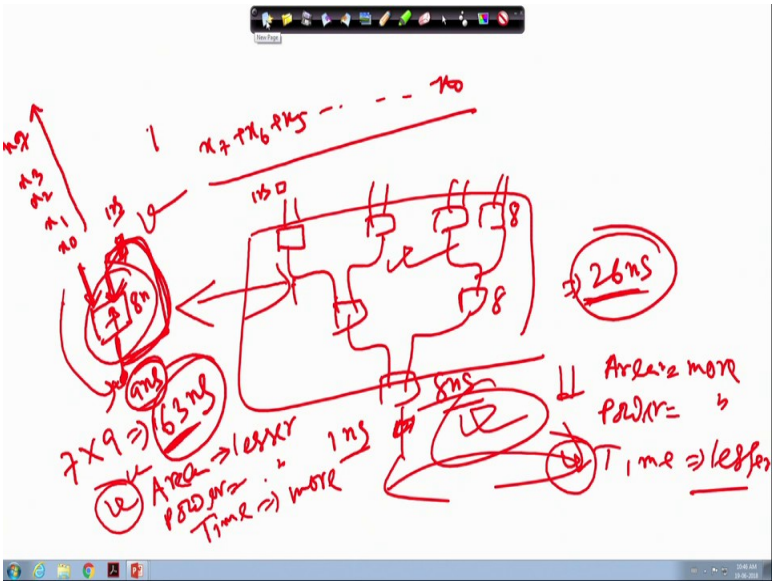
So, here what I am doing? I am I require only 1008 nanoseconds for computation of 1000 number of samples in a serial fashion. So, there what I am getting? There I am getting this y 0 and y 1 then y 2 after 25 nanosecond sorry 26 nanosecond. Why? Because, if I am considering this. So, at the time what I require? One for this and 24 for this particular step after what this is also one. So, total 2 plus 3 into 8 nanosecond there of 20 nanoseconds.

So, y 0, y 1, y 2, y 3, y 4 these samples are basically I have got after this 20 nano 26 nanosecond in each this case. So, here after; that means, running all this process I am getting the corresponding this 1008 nanosecond and then I am getting the corresponding value ok. But, if I consider this so, at that time the hardware requirement is that hardware cost what is that 1 adder and 1 register.

So, in the previous case how many I require? There of 7 adders and total 9 register so; that means, that has been reduced to 1 and 1 over here. But, here what is the problem with this I have to the where I am doing this serially so; that means, the wait time for getting the output that will be more ok. For getting this y 0 and y 1 so, something that

will be more, why more? Because, on that particular case so, suppose for this parallel implementation case.

(Refer Slide Time: 11:58)



Whenever I am doing this so, at the time what is there? That means, in each of these blocks whenever I am doing these. So, at the time only for this as the this is 8, this is 8 some sorry there I have consider 5 here I have consider 8 so, there is a problem, so 8, 8, 8 if I consider ok.

So there, then for computing of this in and this register delay as are 1 nanosecond at this stage, so total 26 nanosecond I require for computation of this particular stage right. But, the thing is that whenever I will just do this serial implementation of that at the time what how many times I require? I have to require here also 1 register right. So, now, x 0 x 1, x 2, x 3 something like this it is coming.

So; that means, for each of this stage I require total a 8 over here and 1 for this particular register. So, 9 nanosecond I require the time requirement for computation of one stage; that means, one of this addition operation it require total 9 nanosecond.

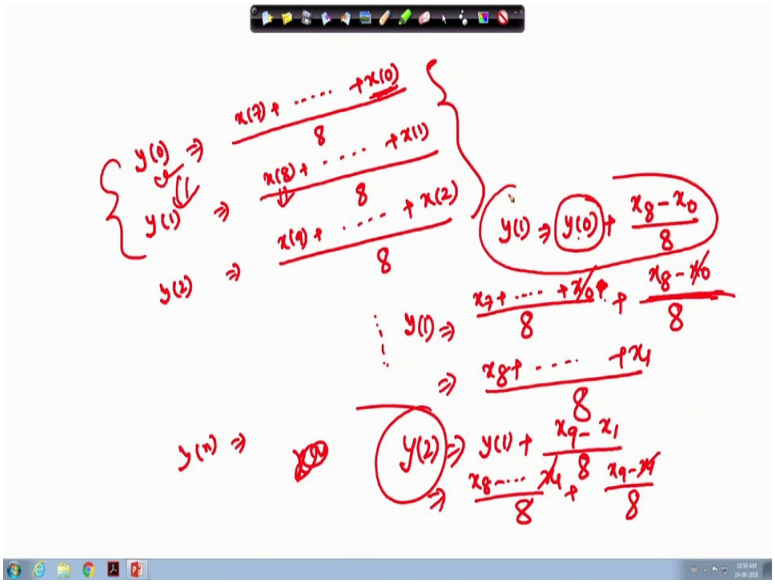
So, here how many times; how; that means, what are the numbers I require? That is up to x 7 right. So, that means, 7 after 7-th cycle I will get the first output which is enabled at this particular point so; that means, total 7 into 9. So, 63 nanoseconds I have to weight for getting the first results.

So, that is why here and what is the number over here? What which I require that is 20 nanosecond but, what is the number here I require that is 63 nanosecond. So, that is why here though I can reduce the corresponding hardware but, I have to the for each of the samples to become computed that I require more time to be wait than this particular parallel implementation.

So, this is the advantage; that means, the advantage is that here advantage for the serial architecture is that I am getting this area that there is lesser; that means, the power will be also lesser but, the time is more. Here what is happening? Area is more power is also I can say that is more but, the time here less sorry lesser than this one.

Now, my point is that considering these 2 particular case this is the parallel, this is the serial implementation. Can I that means, design one particular circuit where I can get both the benefit of this with 2 particular circuit, can I do that? So, if I want to; that means, now what is my observation point here? That means, I will take the advantage of this serial and the advantage of this as that time in a particular circuit. So, for that if I just coming to this that what I said.

(Refer Slide Time: 16:24)



Y 0 equals to what that is x 7 to up to x 0 divided by 8 correct. Then y 1 what I said there of x 8 to x 1 divide by 8. Then what is y 2? That is x9 to x 2 divided by 8. So, something like this it is happening to y n.

Now, from this to particular equation from these particular equations so, can I; that means, make one or can I observe something. So, that I can do the modification and I can get one bit a circuit or what I need to do I mean I need to find out one particular circuit where I can use or where I can get both the benefit of that serial implementation as well as parallel implementation.

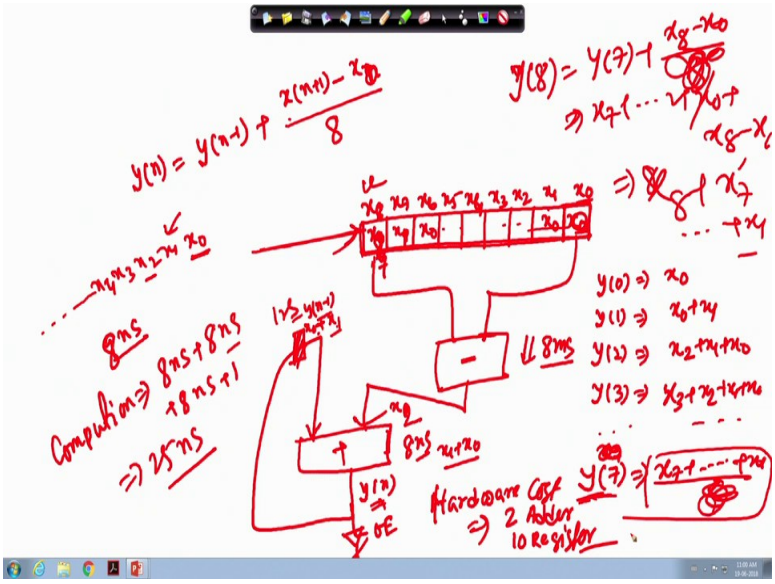
So, if I just; that means, look into these 2 particular y 0 and y 1 so, at that time you see only the thing is that; that means, it in this particular position x 0 is here and in this case the newer value is the x 8. That means, here it is x 0 which is not common in this particular 2 case and x 8 is another number which is not common in this particular case, rest of the all values are same in these 2 equations.

So, considering this can I change this y 1 value something like this can I write y 1 equals to y 0 plus x 8 minus x 0 divided by 8? Can I right that? Yes, I can. How I can? Because, what is y 0? y 0 is nothing but, x 7 to x 0 right divided by 8. Then here sorry here what I am doing? 8 right so, now, x 0 will be correct. So, it will become x 8 up to x 1 divide by 8.

Then again this y th 2 can I write that as this y 2 again that will be written as what? y 1 plus x9 minus x 1 divided by 8, can I write that? Yes, I can. Why, because, this is x 8 to x 1 divided by 8 plus x9 minus x 1 divided by 8. So, x 1, x 1 cut so, x9 to x 2, x9, to x 2 addition divided by 8 so, that is y 2.

So, now; that means, now what I am doing? That means, the previous computed values of these particular numbers samples they are being used in the computation of the next sample; that means, next averaging of this particular circuit. So, then what will be the circuit or what will be the architecture for this particular equation? Now, if I just generalize this equation.

(Refer Slide Time: 20:44)



So, at the time y n will be what? Y n minus 1 plus x n plus 1 minus that will be x 0 or; that means, x 0 if I consider divide by 8. So, now what I require is that here I require 1, 2, 3, 4, 5, 6, 7, 8 ok. So; that means, what I require is that I need this x 0, x 1, x 2, x 3, x 4 sorry x 4, x 5, x 6, x 7, x 8 I need to do this minus of this particular to this is nothing but, one FIFO; that means, the first in first out.

Now, next what I have to do? I have to add with the corresponding this y n minus one. So, now, I need one adder circuit where one of the input will be this and the output will come from sorry here one register I need to put, so this is y and as I am putting registers. So, this will become y n minus 1. So, and this is the; that means, the inputs are coming like this x 0, x 1, x 2, x 3, x 4 something like this it is coming.

So, now consider I have one 9 bit of FIFO here over here and then I have put this ninth bit here and the zero th bit here for doing this subtraction. Then I am putting to a; that means, adder circuit where the output is; that means, feedback to with one reconnected with one register that are connected with this. And this means that we subtractors output they are connected to another input of this adder.

Now, that means, what is the operation or how this is basically operating for computation of running average circuit. Let us see that. So, initially what is happening? Initially x 0 is coming. So, then x 0 will be filled over here and rest will be all zeroes right so; that

means, x 0 minus 0. So, x 0 is coming over here and then this is basically coming to this particular output.

So, in the next clock cycle what is happening? So, x 0 is now coming over here x 1 is changed to this. So, what I said this is nothing but, one free 4. So, thing; that means, x 0 initially was coming to this in the next clock cycle what will happen? x 1 will be filled over here x 0 will come to this particular position. So, now, x 1 and here it is 0 so, now, x 1 this value will be updated with x 1.

So, now what will be the output of this? That will be x 1 plus x 0 because x 0 is the another input and x 1 is the other input: so x 1 and x 0 so, now, this value will be updated with x 1 ok. Then again in the next third; that means, clock cycle what will happen? x 2 will come over here, x 1 will be here, x 0 will be here so; that means, now this value will update again with x 2 and initially what was the value that was x 0 plus x1.

So, it will be x 0 plus x 1 plus x 2 now so; that means, if I consider these so, y 0 value is x 0 at the very beginning, y 1 value is x 0 plus x 1, then y 2 value is x 2 plus x 1 plus x 0. Then y 3 value is x 3 plus x 2 plus x 1 plus x 0.

So, something like this it will be run. So, whenever all these values are fixed; that means, all full whenever; that means, this value is x 8, this is x 0. So, now, at that time if I consider this; that means, whenever this is x 7 and let consider this is x 7, this will be 0 up to this is x 0 right. So, now, whenever I will consider this x, y 7, so at the time all will be there up to x 0 divided by 8.

So, in the next what will happen this will be filled with x 8 and this will be filled with x 0. Now, in the previous case what we have considered that is x 8 minus x 0 along with y is 0 at the time. So, it is not y 0 so, at the time this is this y 0 means is nothing but, this particular operation. So, y 0 is we are considering in this case y 0 we are considering this. So, that is why y 0 is written over here.

So, here y 7 is that particular case so, at that time what will happen? So; that means, it will be sorry y8 will be y 7 plus x 8 minus x 0 divided by 8. So, y 7 is nothing but, this corresponding value sorry 8 values we are not considering over here, we are considering only this. So, x 8 minus x 0 this is what x 7 dot dot dot plus x 0 plus x 8 minus x 0. So, x 0, x 0 will cut so, which will be x 8 plus x 7 up to x 1.

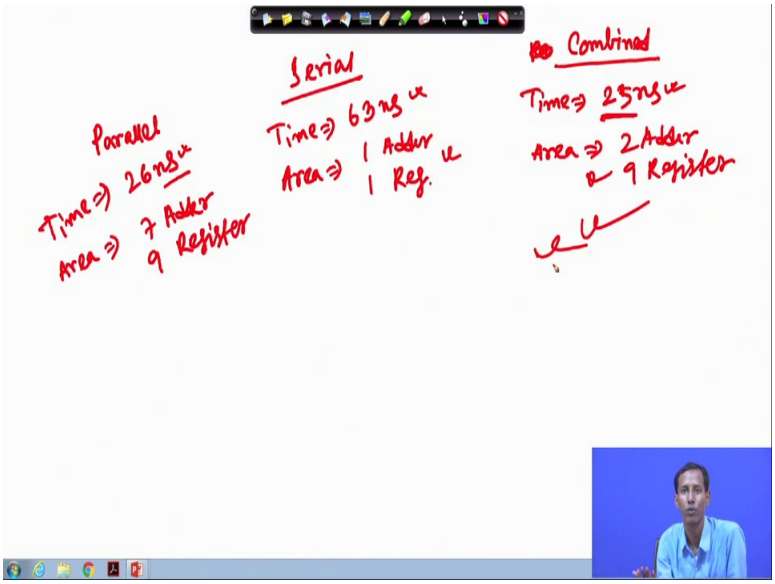
Now, finally, what I have to do? I have to pass through one this and then the LSB will be cut so, this is an output enable signal ok. So; that means, now you see only to fill this particular register I need to wait, after that in each of these cycles I will get the corresponding output. So, in this particular case what is happening I have to wait for 63 nanosecond. But, in this case I do not have to wait so, to fill this particular; that means, blocks how much time I require? All this value 9 nanosecond for 9 filling up these 9 values sorry 8 values 8 nanosecond to fill up all these particular cells.

So, then again if I consider this adder and subtractor delays are same. So, at the time 8 nanosecond delay I require for this and 8 nanosecond delay for this and this is for 1 nanosecond. So, the total computation time is now what is that the computation time which I require that are 8 nanosecond over here, plus 8 nanosecond they over here, then again 8 nanosecond for this particular case and a 1.

So; that means, total 25 nanosecond I have to wait and what is the number of adder so; that means, the hardware cost is 2 adder if I consider adder and subtractor same 2 adder and 1 register sorry total 10 register I think 10 if I consider each of the cell also while register. So, 10 register ok.

That mean, if I just again come back to this. Now let us conclude to this for the parallel implementation.

(Refer Slide Time: 30:23)



That the time requirement is how much? 26 nanosecond and the area is 7 adders plus total 9 register right for serial implementation time is 63 nanosecond and the area is 1 adder and 1 register. For this particular implementation; that means, this I if I that; that means, if I consider this as a combined mode. So, at the time the time determined is how much? 25 nanosecond area is 2 adders and 9 register.

So; that means, now you see in this particular case what I said here I am getting this requirement of the area is advantage over here but, the time requirement is very poor. In this case time requirement; that means, the computation time they are very good, but the area requirement they are very poor. But in this case, I can manage the time is also good the area is also good. That means, both of the advantage of this serial and parallel they are being used in this particular case to get the corresponding optimized hardware for running average circuit.

So, what we have been doing what; that means, now what I am trying to see is that you have to note down; that means, what are the advantage or what are the strength of this particular circuit and what are the weakness of this particular circuit or this particular architecture. Then based on that how we can come to a newer architecture so, that I can get more advantage in terms of speed power area. So, we have started from this; that means, basic then we have seen these architectures and from this architecture how we can build one newer architecture that we have seen.

So, it is not that this is the final architecture again people have done; that means, several works or several researchers on finding out the corresponding optimize hardware for running average circuit. So, this is only the one example I am consider, this is not the case that only one example is there or for this particular case this architecture. There are several types of; that means, blocks are available signal processing blocks. So, for each of this case these kinds of things happen so; that means, what will be the based architecture where I can get the performance on the higher side ok.

So, for today this is it, again we will start with newer thing on the upcoming classes.

Thank you.