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 back to the course on Architectural Design of IC’s. So, in the last few classes we have seen the adder architecture, different type of adder are there and they are used for different-different on perspective for different-different application considering the fact that we need. That means this speed of the constraint or area as a constraint or power as a constraint.

So, we had seen different type of adder like carry save adder, carry skip adder and we have started with the basic that is ripple carry adder then carry look ahead adder. And then we have seen that we have got or ultimately we have did; that means, carry select adder.

And then from that we have ultimately got to the point that conditional sum adder is one of the adder where it is by. That means, putting redundant the blocks how we can achieve; that means, the or how we can maximize the speed of operation of that addition operation. So, that we have seen in the last few classes.

So, now again suppose this adder why I need these type of adders? Because, I said that this adders are required or these type of architectures or different type of architecture has been evolved. Because, they need for different-different applications like signal processing application or wireless communication application or biomedical application everything for the signal processing application.

So; that means, my point is that suppose if I am doing for biomedical application or one particulars hardware I require for biomedical applications. So, at that time the power is the major constraint not the speed. On the same way if I am actually using the adder circuit for wireless communication application at that time speed is my major constraint because, that I have to that mean from the more the data rate or the more frequency I have to achieve there ok. So, that is why the 2 different applications require 2 different type of architecture because, whenever I will follow the architecture which gives me this high speed of operation.

So, at the time it cannot gives me this that it will consume low area low power. Whereas, the same thing; that means, whenever actually we have seen that type; that means, we have to lose somewhere to gain something. So, whenever we try to reduce the power so, at that time; obviously,; that means, lesser number of this the components I have to use or the basic gates I have to use.

So, at the time the lowering the number of components in the circuit; obviously, I have to follow this serial architecture not the parallel architecture. So, parallel architecture I will follow to maximize the frequency; that means, if it is for wireless communication application. I will try to reduce the power, whenever I will try to reduce the power. So, at the time I need to use lower number of components of the basic gates which; that means, I have to follow the serial architecture.

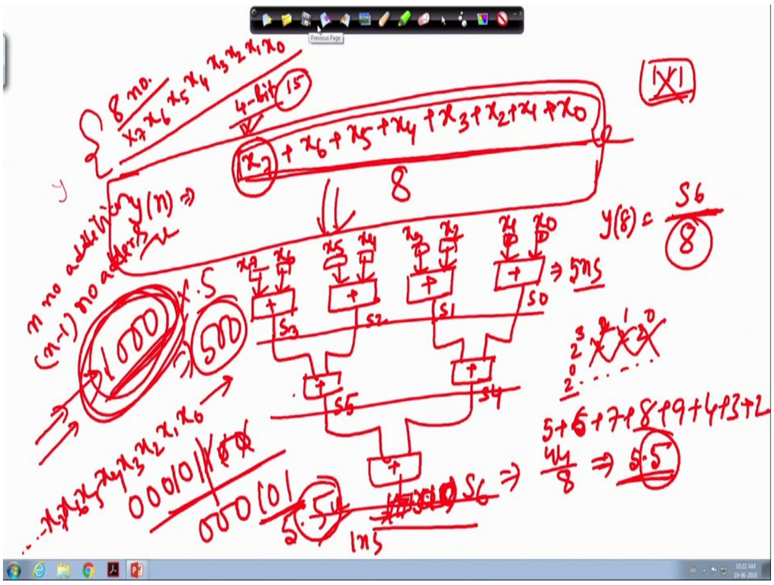
So, that is why I need different type of hardware and different type of; that means, hardware of adders we are discussing over here. So, in this whole; that means, course we will see different type of blocks basic blocks which are being used to design the function which are being used in real time processing or signal processing wireless communication; that means, or biomedical signal processing. So, any of this application it is used.

So, suppose we have seen those architectures, now suppose I have to design one running average circuit. Which is basic very much common in any of this; that means, signal processing blocks. So, running average blocks means I will just explain it.

So, I have to design one running average circuit. So, how can I do? Or what first the thing is that what is the meaning of running average circuit? And then how can I design one running average circuit in optimized way? Ok or what are the ways of implementing this running average circuit so that we will see now.

Suppose, I have to average running average means first let me explain what is averaging circuit? Averaging circuit means suppose I am having 8 numbers ok. So, the summation of these 8 numbers that will be divided by the numbers how many numbers I am adding, so divided by that that is the average circuit, basic averaging circuit ok.

(Refer Slide Time: 05:34)



So; that means, suppose if I consider if I consider suppose y equals to this y of n that is if I am considering of 8 numbers suppose. And this 8 numbers are x7, x6, x5, x4, x3, x2, x1 and x0. So, if I need to; that means, do the average of these particular 8 numbers at the time how it will be? That will be, it will be x7 for plus x6 plus x5 plus x4 plus x3 plus x2 plus x1 plus x0 divided by 8 ok.

So, in the hardware if I want to implement this particular circuit or the corresponding hardware for this particular circuit, so at the time how can I do? Now consider each of this particular; that means, bits; that means, each of this word length of these numbers let us consider that are of 4 bit ok. So; that means, 15 is the maximum number which I can do.

So, now if I want to implement these particular functions, so at the time how can I do? The parallel implementation is that; that means, for this total how many numbers I have to add that is 8 numbers. So, each 8 number if I want to add using; that means, adder means there of 2 input they can consider 2 input at a time. So, considering that fact how many number of adders I require to implement these two; that means, first initially what I have to do? I have to add all these numbers then I have to divide by 8.

So; that means, to add these 8 numbers I need 7 numbers of adder and this is the fact that if I want to add n number of adder sorry n number of addition operation then n minus 1 number of adders I require. So, here so, how can I implement these things? If I just map it to the; that means, corresponding architecture for this.

So, at the time this will be something like this. I need initially I need at the first stage I need 4 number of addition; that means, adder block correct. So, this will be x7, x6, x5, x4, x3, x2, x1 and x0.

So, I will get this sum 3 over here sum 2 over here sum 1 over here and sum 0 over here. Then again this particular 4 of this they need to add again; that means, again I need 2 adders for adding this 4 numbers. Then what I will get? Then again I will get s5 and s4 over here ok. Then again for this again I need this particular. So, this is my if I consider this n as; that means, 8. So, this is my final output. So, not ok, this is not the final output. What I left I have to; that means, do this divided by 8. So, how can I divide by 8? If that means, write this as the s6 0, 1, 2, 3, 4, 5, 6. So, if I; that means, mentioned that s6.

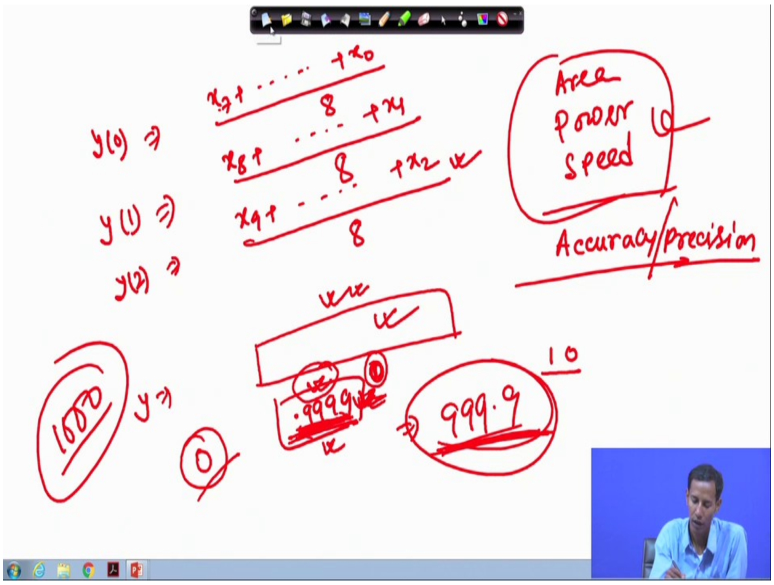
So, now, y y 8 equals to what? S 6 divided by 8. So, this divide by 8 means nothing, but the truncation or the of the last 3 LSB position. Why? Because, we know that what we know in binary numbers they are weighted off with the weight off 2; that means, there is for 2 to the power 3, 2 to the power 2, 2 to the power 1, 2 to the power 0.

So, the binary numbers are something like this they are being weighted something like this so; that means, divided by 8. So, if I just discard this particular 3 position. So, at that time so, this will be now coming to 2 to the power 0 so; that means, I am doing divide by 8 by truncating the last 3 bits. So, I do not need any extra hardware for doing this division operation. So, this is the corresponding; that means this parallel implementation of this particular circuit.

Now, coming to the point; that means, running average means what?. So, running average means these particular sets they are basically changing and running average circuit where do I require? That is very much required for one real time signal processing. So, in real time signal processing what does; that means, what happens? So that means, the samples are changing in real life; that means, they are dynamically changing there is a number of samples they are dynamically changing.

So; that means, now how does that mean samples are coming? Like this x1, x2 x3, x4, x5, x6, x7. So, something like this, it is coming ok. But, only what I have to do only; that means, these particular things I have to do. So that means, in the first set if I consider; that means, if I just go to there.

(Refer Slide Time: 12:23)



That means if I consider this is the first set. So, at the time this x7 up to x0 divide by 8 that will be done.

In the next, what will be done? That will be x8 plus x1 divide by 8. So, y 2 will be what? They will be x9 up to x2 divide by 8; that means, 1 by 1 this will be vanish ok. And the newer of the samples will be collected in the averaging circuit for doing this particular.

That means, average averaging of that where the number of samples they are coming serially to the circuit and when were the values of the corresponding incoming the samples their change changing dynamically ok. So; that means, now this is the corresponding; that means, circuit for and this is the; that means, concept of running average circuit.

So, now here actually one point is there actually 2 point I have to; that means, notice or I have to observe. That is till now what we have; that means, seen is that that area, power and speed these 3 are the major constraint while we are doing this VLSI that means, circuit design. But, there is another thing, which is this accuracy or this if I say that the precision, what is that because, what we are doing. Whenever we are talking about hardware so,; that means, we are dealing with fixed point number system. So, I think this fixed point and floating point number system that is the basics we have already learned in a digital electronic circuit.

So, this fixed point number system means; that means we have this, the word length they that that are limited ok. So; that means, we have to fix initially we have to fix or we have to; that means, final is that what will be the corresponding word length of the variables or the signals.

So, in; that means, in analog domain or in floating point what we do sorry what we do? We have something like; that means, infinite precision so; that means, I suppose this 0.999 value ok. Can I represent this 0.999 value in using this; that means, using binary numbers because, the binary number is taking only 0 or 1.

So, how can I represent then this 0.999 number? I cannot. Why I need to do this? Because, in signal processing sometimes what happens I am repeated; that means, the operation they are being used in a loop. For let us consider 1000 times or 2000 times suppose these numbers for this particular circuit suppose, this running average circuit I need I am doing right. So, each of these numbers they are lesser than 1 and thousands of this I am just adding. So, at the time if each of this number is let us considered fixed to 0.999.

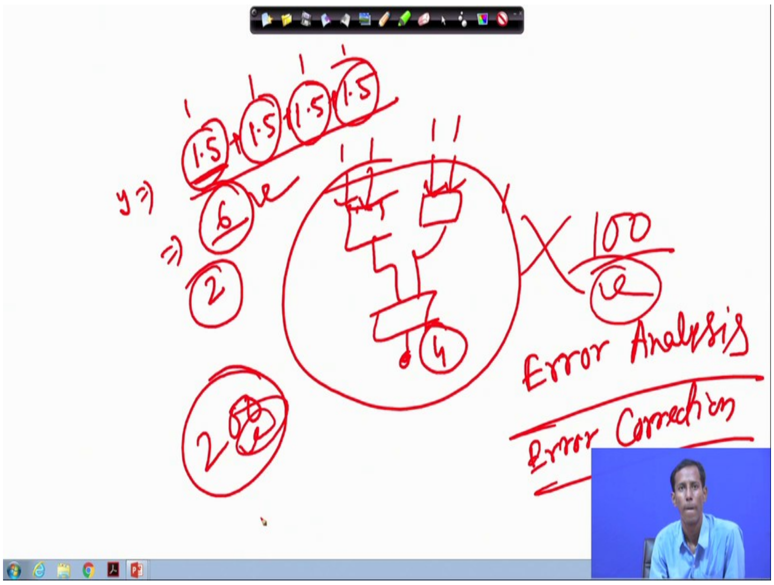
So, now, if I just add these particular 1000 numbers, so at the time what will be the output? This, suppose this particular numbers if I want to add 1000 times. So, then this will be my final output this should be the final output. But, in this case if I use this fixed point; that means, number representation.

So, at that time what I will get because this x0 to x7 they are not for any time they will not consider this as 1. They if that is considered because they are less than 1. So, that will be considered as 0. So, if I consider 0 so, 1000 times if I add this particular circuit it will give me 0 only.

So; that means, this is the error which I am accumulating whenever we are running in a loop. And this particular; that means, numbers is the numbers are there and they are being; that means, running in a loop; that means, what I am doing? In each of this stage I am just getting the error of this. And how many times I will run that particular loop that much of error I will accumulate finally, so that means, this is the error which I am accumulating finally.

The same thing if I take 1; that means, another newer example; that means one different example.

(Refer Slide Time: 18:27)



So, at the time let us consider 1.5, 1.5, 1.5 and 1.5 I have to add this 4 number I have to add. So, if I just add it just in decimal number considering decimal number if I want to add this particular. So, at the time what will be y? y will be 6. So, 6 will be my result.

Now, considering each of these numbers on that particular circuit what we have just discussed now. So, if I just add at the time what will happen? So, at the time what will happen? So, this will not consider 1.5. So, this will consider 1, 1, 1, 1 so, this will be 1, 1, 1, 1. So that means, finally, I will collect 4. So; that means, 2 are the error which I am basically accumulating. Suppose this particular circuit running for 100 times so; that means, 200 will be the error which I will accumulate for running this particular circuit.

So, accuracy is also one of the constraints which we have to bother about whenever we are doing the corresponding architecture hardware architecture from the algorithm. So, when we are doing the algorithmic verification so, at the time if we consider only floating point number of the number representation.

So, at the time we will not be bother about that much bother about this particular error accumulation. But, as we are doing this fixed point operation in the hardware circuit so, at the time we have to bother where we have to be cautious more cautious about how much error I am accumulating whenever I am doing the corresponding particular operation or the. That means, this internal circuit of that hardware.

So, we will take; that means, many of the; that means, example. So, where we you can see that how much or how to tackle with or how to; that means, do this with. Or how to tackle this changing of this accuracy whenever we are doing this serial operation or we are performing the operation in a loop that we will see in the later of this course.

So, here actually; that means, for that what I have to do? I have to this do this error analysis part. Error analysis and after that what I need to error correction method. So, if I just go back to the corresponding circuit. So, at the time if I just at the time what I am doing? Here I am just why I have just discussed this particular accuracy point whenever I am just discarding this particular 8 bit finally, here.

So, let us consider these numbers suppose they are of 5 plus 5 plus not 5, then 6, then 7, then 8, then 9, then 4, 3, 1, 2, 3, 4, 5, 6 and then 2. So, what is the number? 11, 18, 26, 35, then 39, 42, 44, 44 divided by 8. So, what will be then; that means, number that what will be the corresponding output of this? That will be 5.5 ok.

So, what I will get if I just add this particular from this particular point if I just do this. So, I will not get so, 44 means what will be it is binary number representations; so 32 then 8, then 4. So, this will be the corresponding binary representation of 44.

So, now what I have to do I whenever what I said that I have to discard the last 3 LSB position for as I am doing this divided by 8. So, here in this number; that means, this is my s6 right this 44 is my s6. So, now, what I have to do? I have to discard this particular 3 bit.

So, now what value I am getting 0 1 0 1. So, 1 0 1 means 5, but what is my result that is 5.5. So; that means, this .5 is the error I am accumulating for truncation of this particular block ok.

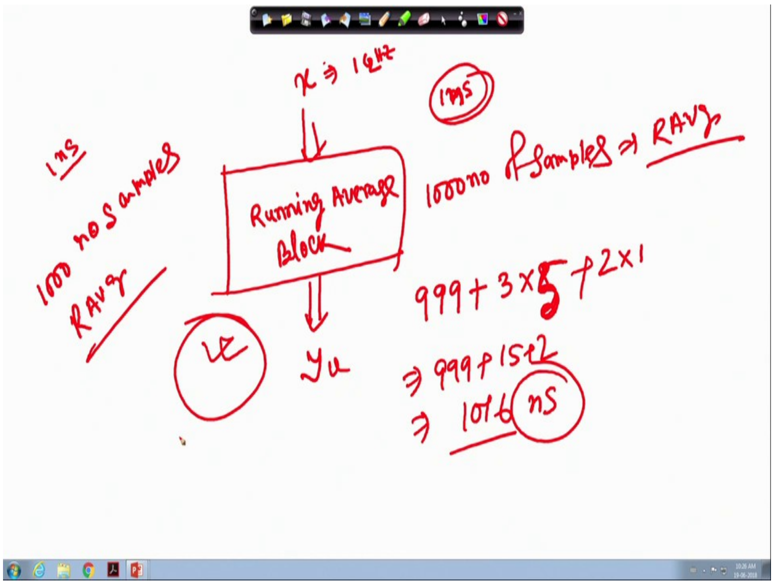
Now, if these circuits what I said now if this circuits if I run it for 1000 time in a loop. So, at the time in each of this stay or each of this loop I am accumulating 0.5 so; that means, now point 5 means 500 is the amount of error which I am accumulating if I just run this particular circuit in a loop ok. So, this is the running average circuit and they that is that this is the problem in corresponding running average circuit whenever I we are doing.

Now, this is the; that means, the corresponding parallel implementation right. Now, for these suppose what I said that I have to do this 1000 number of addition operation. That means, this is the number of samples which are coming at this flow there I have to do the running average of this 1 0 1000 1000 number of samples.

So, at the time for the whole process, the total computation time for running average circuit; that means, doing the running average of these 1000 numbers of samples what will be the computation time for that? If I consider this each of this; that means, delay for this adder, they are let us consider of 5 nanosecond and I need to put the registers over here for each of this there will be. That means, each of that input side there will be register and the output side the register also we will be there right. So, this is the input register this is the output register.

So, if these registers may requires 1 nanosecond of delay, so at the time at the time sorry. So, at the time what will be my total computation time for doing the running average of 1000 number of samples? So; that means.

(Refer Slide Time: 26:45)



Let us consider each of the samples is coming after 1 nanosecond so; that means, the sampling speed. So, now, if I just draw the block of this is the running average blocks correct. Now x is coming x is the input which is coming and y is the output I am getting right.

So, now x is coming what? They will also have some clock frequency ok. So,; that means, now this is coming or this sampling speed of this they are if I; that means, consider that as 1 this Gigahertz. So, that means, each of the samples are coming at 1 nanosecond of time.

So, if I consider this 1000 number of samples 1000 number of samples for doing the running average. So, at the time how much how much time I have to wait for all the samples to come for to this particular circuit? They will be 999; that means, after 999 I will get the output of y.

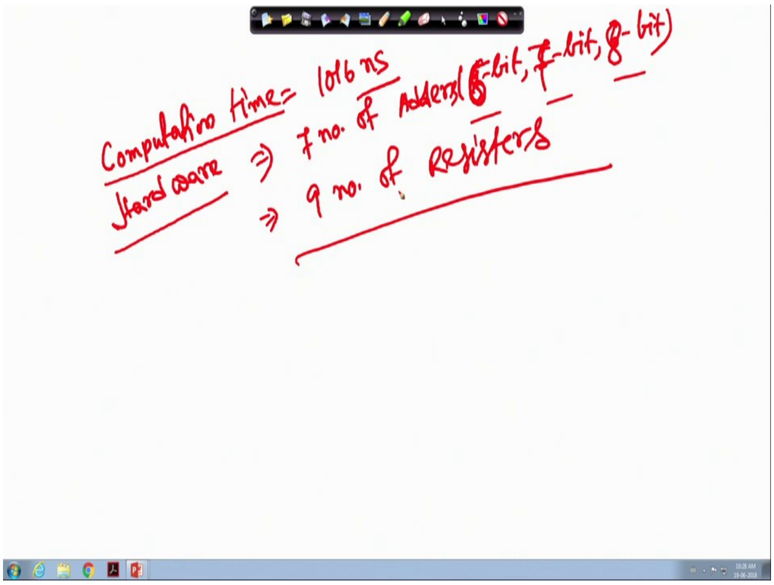
Now, apart from that what I need I have to consider the corresponding delay of this each of this stage; that means, the adder corresponding adder delay for this 3 stage that also I have to consider. So, what I require? That means, 3 number of stages are there; that means, 3 and each of the adder delay is 8 sorry 5. So, that will be the corresponding delay and what I require oh; that means, what else I will left there that are the corresponding delay for this particular registers input registers and output registers.

So, then 2 of them I am using why 2 of them; that means, I am using 8 at the input side and one at the output side, but 8 at a parallel. So, each of this is; that means, consuming 1 nanosecond delay so; that means, a maximum of that that will be the delay for that register. So, maximum is one. So, I am considering one for the input delay and one for the output delay so, into 1.

So; that means, now 999 plus 15 plus 2 that is equals to 1016. So, this time I require to complete the whole process of 1000 number of samples running average. So, this is the; that means, parallel implementation if I choose so, at the time this will give me this.

So, then again this is the numb; that means, if I consider.

(Refer Slide Time: 30:17)



Computation time = 1016 ns

Hardware ⇒ 7 no. of Adders (6-bit, 7-bit, 8-bit)

⇒ 9 no. of Registers

Then; that means, what I am getting? The computation time is 1016 nanosecond ok. Then the hardware requirement, hardware requirement is 7 number of adders each of are not 4 bit that will be 4 bit. In the next stage it require 5 bit and in the next stage it require 6 bit.

So, why I need to do that that I will also not here. So, it will consider 4 5 2 bit I will have 6, here 7, this will require 8 bit. So, why I require 8 bit? That I will come later not 6, 7, 8. So, what we will be because running bit size then again I will discuss on that and then totally plus 1 9 number of registers I require as the hardware requirement. So, if I consider this particular circuit; that means, that parallel architecture. So, at the time this is the number of. That means, this is the total computation time and this is the hardware requirement of that party this for this corresponding circuit.

So, for today this is it then again we will discuss or we will see different architecture for this particular operation; so that, we will see in the next class.

Thank you.