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

Lecture – 27
Multiplier Architecture

Hello everyone, welcome back to the course on Architectural Design of ICs. So, in the last class we have seen the techniques of pipelining and parallel processing, and how these 2 techniques helps to increase the sampling speed or the clock speed of the circuit whenever we have achieved. That means, after all the optimization that is algorithmic optimization or architectural optimization whatever logical expression we have got; that means, the critical path we cannot cut down by any further optimization. So, at that time there are; that means; 2 techniques by which you can increase the speed of the operation of the particular circuit.

So today, and then just prior to that we have learnt different adder architecture, so different adder architecture for different-different purposes right. So, here today we will start with multiplier architecture. So, this is ad addition operation, multiplication operation, division, squaring. So, these are the basic operators which are basically used in digital signal processing ok. So, then digital signal processing is basically applicable to everywhere; that means, whether there is image processing or this wireless communication or any biomedical signal processing; that means, everywhere this GSP blocks they are very much useful.

So, that is why those blocks can be built based on these particular operators. So, now, at that is why if we as I said that earlier also I said that if the basic building blocks whenever we are designing so, at that time we if can save. So, whenever multiple copies of that have been used in the particular. That means, in the upper blocks at that time, I can save that times of savings in terms of the whether that is power or area ok. So, that is why we have to learn this different architecture of this addition operation or this multiplication operation or division operation.

So, today we will see there are different architectures. So, that is why people have tried rigorously; that means, to find out the architecture for multiplier or this adder for so, that will be a high performance design; that means, how we can increase the performance of

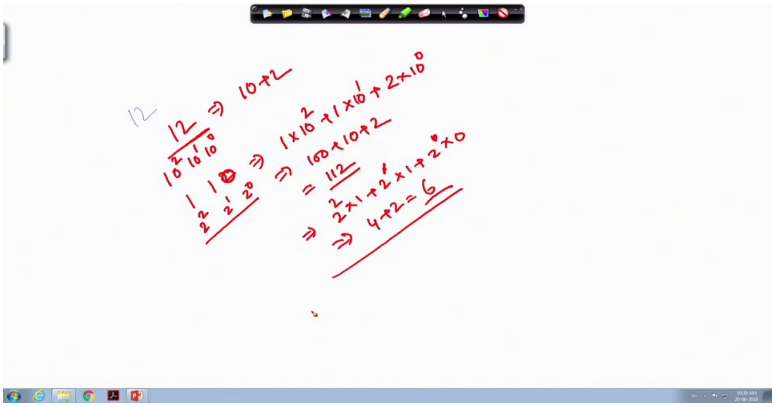
that particular circuit, that was the major focus in research or that means in useful for any application.

So, now today we will see this multiplier architecture. So, this is the road map multiplication whenever we will do multiplication at that time there will be I think in the basic digital course we have learnt, there are 2 types of number system where it is 1 is signed representation of the binary number system or any other number system and there is also unsigned number representation ok. So, whenever we will do multiplication at that time suppose if I want to multiply 2 in 2 when both are positive at that time. So, I will get 4, but if I want to multiply 2 with minus 2, so at the time the results will be minus 2 sorry minus 4.

So; that means, in this case whenever I have to do this 2 into minus 2 so, at that time I have to mention the circuit or I have to design the circuit in such a way so, that it can represent or it can give me the corresponding minus 4 value ok. So, that is why this is the multiplication there being; that means, divided into 2, whether there is signed multiplication or unsigned multiplication ok. So, in adder also it is useful; that means, signed adder and unsigned adders are there ok. So, I have not discussed on that particular time, but whenever we will do this multiplication operation so, at that time I have to consider this.

What is multiplication operation? Multiplication operation is basically repetitive addition operation. what you do in multiplication? In generic multiplication what we have learnt in our; that means, during our school time. So, that is nothing but that shift and then add then shift and then add. So, whenever we are doing this decimal multiplication. So, at that time the corresponding position the weight of that that is in terms of 10; that means, suppose 12. So, 12 how we define 12? 1 and 2 so that means the position for the if I just come down to this.

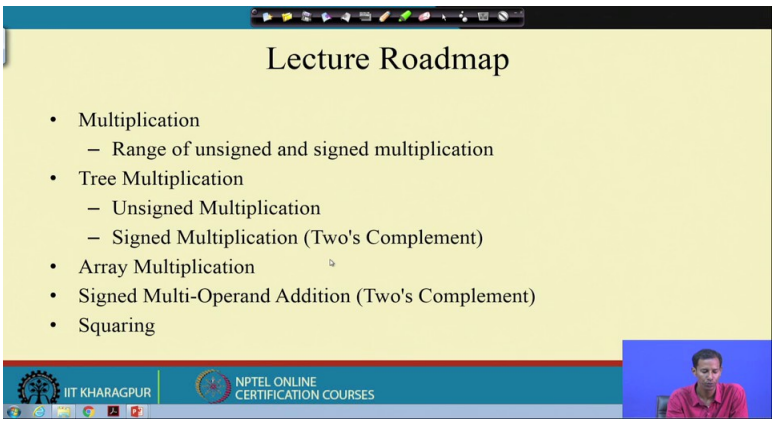
(Refer Slide Time: 05:34)



So, how we define 12 how we define 12; so that is 10 plus 2. So, basically this 12 means this is 10 to the power 0th position and this is 10 to the power 1 position and if I say this is 2. So, 112 means what; so 1 into 10 to the power 2 plus 1 into 10 to the power 1 plus 2, 2 into 10 to the power 0. So, this is 100 this is 10 this is 2. So, total 1 1 2.

So, the same things in binary if I represent. So, at that time this will be 0 or 1; if I put 0. So, at the time the weight becomes this is 2 give a 2 this is 2 to the power 1 this is 2 to the power 0. So, at that time this value becomes what; 2 to the power 2 into 1 plus 2 to the power 1 into 1 plus 2 to the power 0 into 0. So, that is nothing but 6 ok. This is unsigned number representation, but in signed number representation there are 2 format what is that? One is signed magnitude and another one is twos complement format ok. So, we will come to that point later, but there are 2 type of number representation in negative numbers in binary that is signed magnitude representation or twos complement number implementation ok.

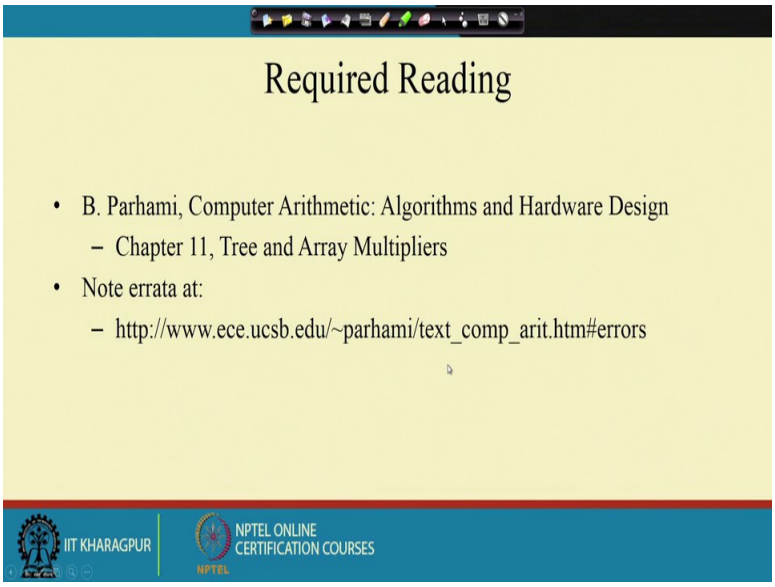
(Refer Slide Time: 07:23)



So, again if I just; so here if you see that in this lecture we will follow this multiplication, then type of multiplication or tree multiplication, then array multiplication then signed multi operand of addition or this twos complement multiplication and then squaring also squaring is also nothing but one multiplication operation.

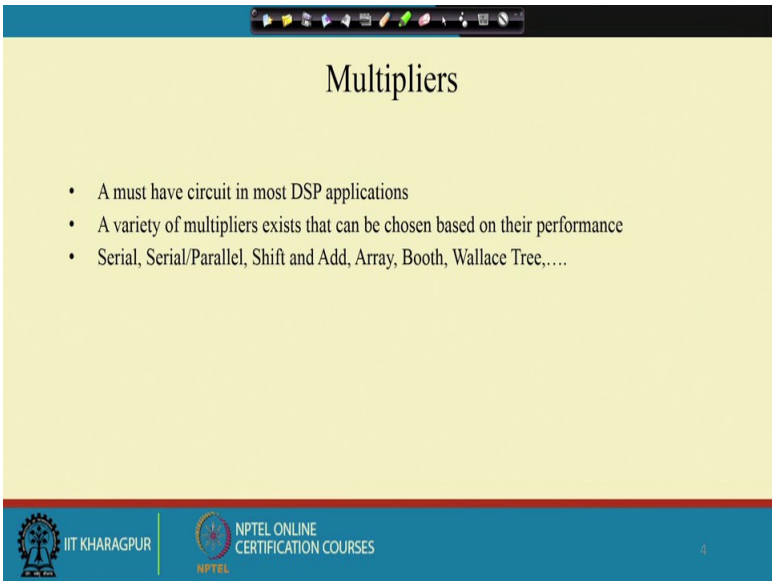
And here whenever we are considering so, at the time all the multiplication operation we are considering that, this is what this is what fixed number; that means, no floating point number we are considering ok. So, if we have to; that means, in some of the DSP algorithms, we have to use or we have to; that means, perform this we have to consider the floating point too. So, at the time how we can do; so that means, at that time multiplier architecture will be different for those types of operations if it is required ok. So, that type of multiplication operation or the multiplier architecture also we will see.

(Refer Slide Time: 08:39)



So, then this you have to follow I think I have already mentioned this is one of the; that means, best book for computer architecture that is written by B Parhami ok. So, the book name is computer architecture algorithms and hardware design. So, here you will find all these things; that means all the adder architecture multiplier architecture all this architectural details are there. So, you can follow this book.

(Refer Slide Time: 09:12)



Then what I said the multipliers they are mostly used for DSP application ok. And a variety of multiplier exists that can be chosen based on their performance; that means, So, different architecture has been evolved because of different need whether that is high speed or that is for low power or for that is low area and that is why there are the architectures are like serial, then serial parallel both, then shift and add then array then booths multiplication then Wallace tree multiplication, dadda multiplication, then hybrid multiplication then recently that means, people have developed this the vedic multiplication ok. So, so much of multiplication operation or architectures are there.

So, do we; that means, in this particular course I cannot; that means, as there are plenty of that, I cannot cover all the multiplier architecture it is better if you just want to; that means, the basic or the if you know or if you want to; that means, for Your research purpose if you want to consider this multiplier architecture, what will be the numeral architecture for or how you can define.

So, we will take 1 or 2 case studies, where I will show you that how; that means, what is the starting point for designing and then implementation of that particular design. That means, my point is that suppose I have to the multiplication operation everyone knows how we do; the same multiplication operation whenever we are implementing it hardware or whenever we are finding the VLSI architecture for that multiplication

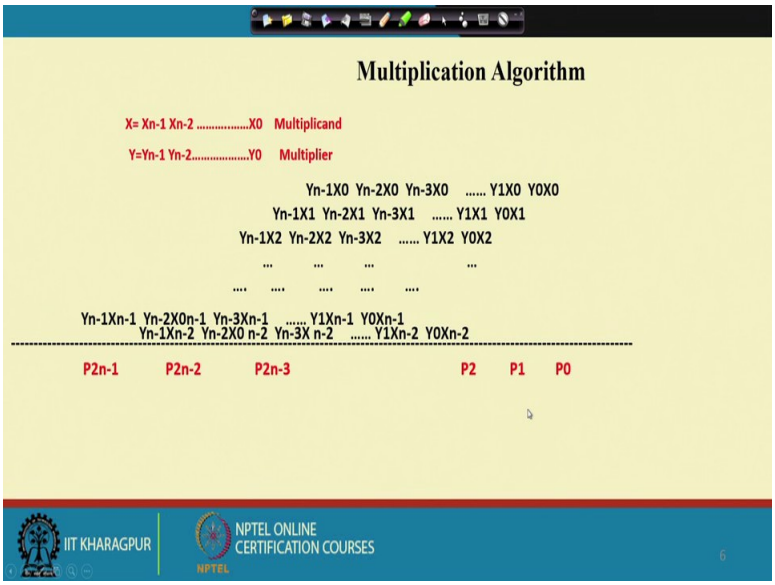
operation. So, at that time at what point I have to look and from looking or from observing that point how we have to proceed.

So, once we have proceed; so at that time that means finally how will I map to that particular things on the; that means, the algorithms suppose we have find out one newer algorithm for implementing the multiplication operation right. So, after finding the algorithm, the how then this algorithm can be mapped to the architecture so, that I can get the benefit of speed power and area. It is not necessarily; that means, true that all the time I will get all 3 altogether. So, it is not; that means, all the time it is true, but I will try my level best or as a designer or as a researcher, I will try my level best to optimize the circuit in terms of this 3 particular constraint ok.

And apart from the Yesterday's class I have discussed that, we are not considering for this accuracy purpose or the resolution purpose; that means, what will be the word length as am I am considering; that means, I am working on the fixed point domain. So, at that time if I consider more number of word lengths for considering the number systems. So, at that time I can get more accurate result, but whenever we are; that means, an whenever suppose we are considering more number of word length. So, more number of word length means we are basically consuming more power and more area. So, then what will be my choice optimal choice so, that within that particular word length, my results will be satisfactory as well as my corresponding hardware will also be ok or that will be also within the limit ok.

So, that is why we have to do some analysis for that; fixed point analysis or error analysis we have to do whenever we map to the architecture that is also 1 part of this; that means, algorithm to architecture mapping ok. So, that we will see later of this course; that means we will see that as a case studies. So, there we will see explain or we will discuss more onto that particular perspective.

(Refer Slide Time: 13:45)



So, here this is the basic multiplication operation; suppose I have to multiply X and Y where X is the multiplicand and Y is the multiplier. So, in binary if I represent this X and Y. So, they both of X will be represented as X 0 to Xn-1; if I consider X of n bit Y of n bit. So, then X will be represented as X 0 Xn minus 1 and Y will be represented at Y 0 to Yn minus 1. So, then how we do this multiplication of this particular two? X dot Y so; that means, the bit wise is multiplication operation is performed here in binary multiplication so; that means, now what will happen this Y 0 bit of Y or this multiplier, the bit of Y 0 that will be multiplied with each bit of X and that will be written in this particular this particular in a first row.

So, here you see this Y 0 you has been ANDed here multi 1 bit operation 1 bit multiplication means nothing but the ANDing operation right so; that means, this Y 0 is basically ANDed with X 0 then Y 0 is; that means, sorry here just the opposite thing that is X 0 is the multiplicand and Y 0 is the multiplier. So, X 0 will be multiplied with all the bits of Y ok.

So, then that is why X 0 is now multiplied with Y 0, then X 0 is multiplied with Y 1 up to Yn minus 1 X 0 has been multiplied and that has been re; that means, reported in the first row. Then what will happen? In the next this X 1 will be multiplied; that means, the next bit now it will multiplied with Y, but whenever we are as in the beginning what I said, this number is numbers are basically what? They are basically weighted so; that

means, here this is 2 to the power 0 here it is not 2 to the power 0 it is 2 to the power 1. That means, now whenever I am considering X 1 bit that means in binary I have already considering the weight of 2.

So, that is why there is 1 bit shift over here so; that means, the weight will start from at this particular stage, it will not start from this particular stage ok. So, that is why because I am doing this 2 to the power 1 by; that means position now. X 1 is coming at 2 to the power 1 position that is why this X 1 Y 0 that will come at the 2 to the power 1 position only ok. In the same manner now X 1 will be multiplied with; that means, ANDed with Y 0 dot each bit of Y, then X 2 will be ANDed with each bit of Y and that position will be what? At this case the position will be 2 to the power 2. So, in this same manner I can write all the values of this X up to Xn minus 1 I have to write actually this row will be in the upper 1.

So, after that what I will do? I have to. So, these are called as the partial products. So, these particular rows are named as partial products. So, now, finally, what I have to do, whatever is the; that means, the numbers which am getting after this ANDing operation.

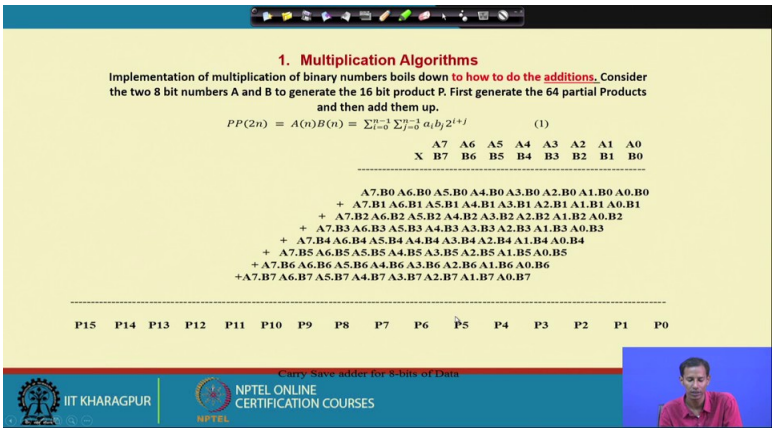
So, that I have to add and finally, I will get the product over here; and another thing is that whenever I am considering suppose n is 4 here. So, at the time 4 bit I have multiplied with X 4 bit with Y of 4 bit. So, at that time the product will be n plus n which is 2 n; that means, in this case the product will be 8. So, that is why you see there is 2 n minus from P 0 to that is P 2 n minus 1 because this is already. So, here if am considering 4 bit over here, and 4 bit over here so; that means, these is basically now at this particular case what is happening? This is 2 to the power 3 so; that means, this 2 to the power 3 for up to 2 to the power 3 it will be shifted so; that means, already I am multiplying with this particular number with what values. So, here this is 2 to the power 0, then 2 then 4 then 8 so; that means, already 2 to the power 7 sorry up to the 7.

So, initially am considering 4 bit over here, and then 3 bit are multiplying with 8 bits means what? I am left shifting by 3 so; that means, initially there will be 4 bit for X, now I am again I am just shifting that 2 additional 3 bit left shift. So that means, now the length becomes what? 7 and where from another bit will come as I am doing this bit wise addition operation for finding these products final products. So, at the final stage there is

a carry out so, that will be the 8th number of bit ok. So, that is why whenever we will multiply with n number of with n. So, at that time it will be n plus and which is 2 n.

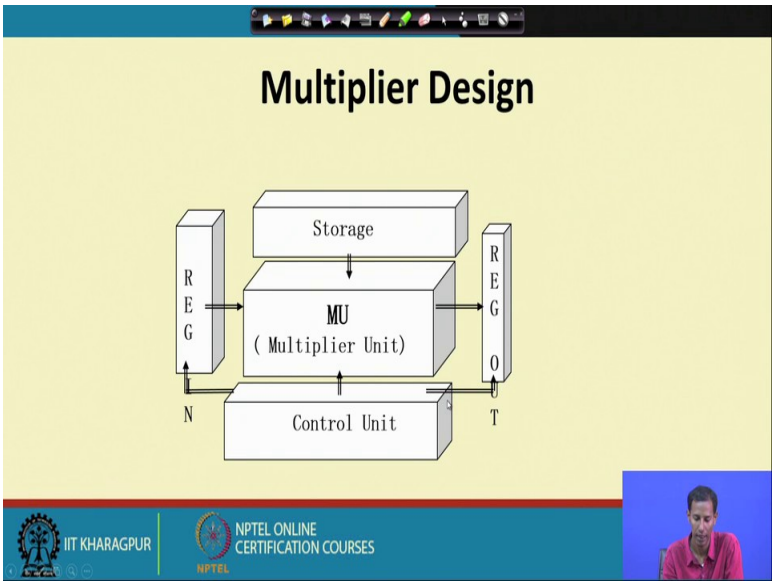
So, suppose if I am having X of m and Y of n. So, at that time what will be the product length? M plus n.

(Refer Slide Time: 20:02)



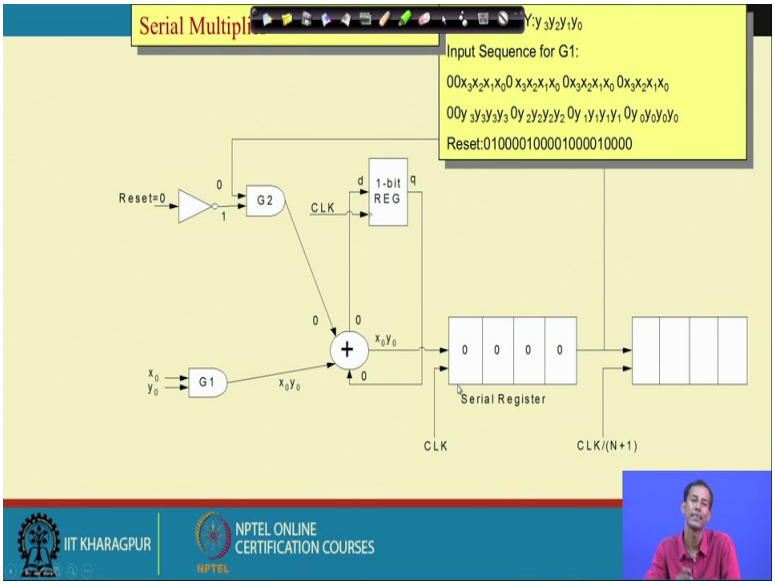
So, if I consider this. So, it says that implementation of multiplication of binary numbers Y is down to so, how to do the additions. Here if I consider 2 bit of A and B of 2 bit; that means, sorry 8 bit. So, at the time the multiplication operation will be something like this. So, 8 plus 8 means what? So, total there will be product will be of 16 bit right.

(Refer Slide Time: 20:47)



So, after that there are what I said that there are different type of multiplication operation ok.

(Refer Slide Time: 20:47)



So, the first thing is that, initially what we know that shift and then add. So, that is why this is the serial multiplication operation the basic multiplication operation multiplier architecture which is which over the algorithm or not the algorithm, which techniques we have learnt during our school times.

So, here you see there is one serial registers and this X 0 this is bit wise serial multiplication; means what serial multiplication how we can do? We can do in 2 way we can; that means, all the things all the bits we have we can consider together and then with Y 0 or Y 1, we can multiply and or and then we can add; so here also the same thing if you just see. Initially what is happening this X and Y there being ANDed and they are added with the corresponding output from this shift register or this is nothing but the shift register means that is nothing but the serial register so; that means, what is happening? So, initially if I go back to the previous slide so, initially what I have to do? This A 0 B 0, so that will come as P 0 ok.

So, then in the next A 1 B 0 that will be multiplied and then that will be added with A 0 and B 1; that means, whenever; that means, I will put this X 0 and not X and Y the corresponding bit of X and Y through a NAND gate sorry AND gate, and then that will pass through 1 OR gate if I do this bit wise operation, but here in this case we need

addition operation, because we have to generate each of these partial product using NAND gate sorry AND gate. And then those corresponding partial products they will be added over here and then they will be passed through 1 serial register ok.

So, this is the; that means, operation of thus or this is the serial multiplication architecture. So, this is the very basic of the multiplication. So, here what will happen? Because as am considering this; that means, serial operation, so to generate the corresponding partial products suppose 4 into 4 if I consider to generate, the partial products it needs 16 clock cycle. Then again addition operation of that then again that will take some time ok. So that means I have to wait if I consider more number of bits for multiplying.

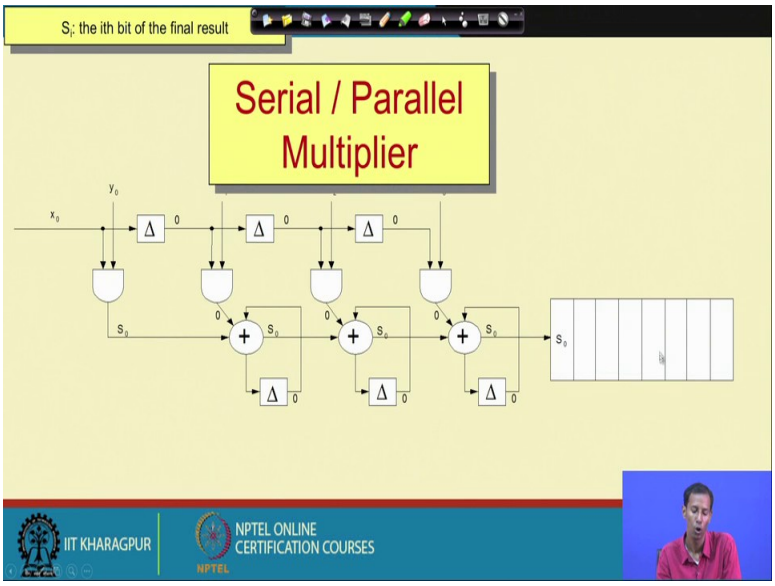
So, at that time my wait time or this if I say that is that has the total computation time for this multiplication operation that will be more ok. So, that is why this type of architecture is not that much advantageous, when you have this critical time requirement, but in this case what is the advantage? Here as am considering the sequential case or the serial multiplication so, at that time using only 1 adder I can do that. So that means, the hardware requirement for this particular this type of architecture, that will be lowered.

So, here you see this input sequence of this G 1 will be something like this; that means, what I said that? This is X and Y. So, it will be the sequence will be first X 0 Y 0, then X 1 Y 0, then X 2 Y 0, then X 3 Y 0 something like this for Y the sequence will be like this Y 0 4 then 0 then Y 1 n 0, then Y 2 n 0 then Y 3 then 0 0. For X what will be the multiplication; that means, input sequence that will be X 0 X 1 X 2 X 3 then X 0 X 1 X 2 X 3 then X 0 X 1 X 2 X 3 X 0 X 1 X 2 X 3 something like this. So, this is for what application? If I am multiplying X with Y so; that means, now what is happening? By putting this and gate; that means, whenever Y 0 then Y 0 X 0 Y 0 X 1 Y 0 X 2 Y 0 X 3 so; that means, the first partial product has been generated ok.

So, then that has been loaded to this particular serial register, then that will come over here and that will be added with this ok. So, then again that will be stored in somewhere in this particular register that will be stored ok. And then again that will come finally, that will come the final results that will come over here ok

So, this is the serial multiplication or the bit wise multiplication ok.

(Refer Slide Time: 26:49)

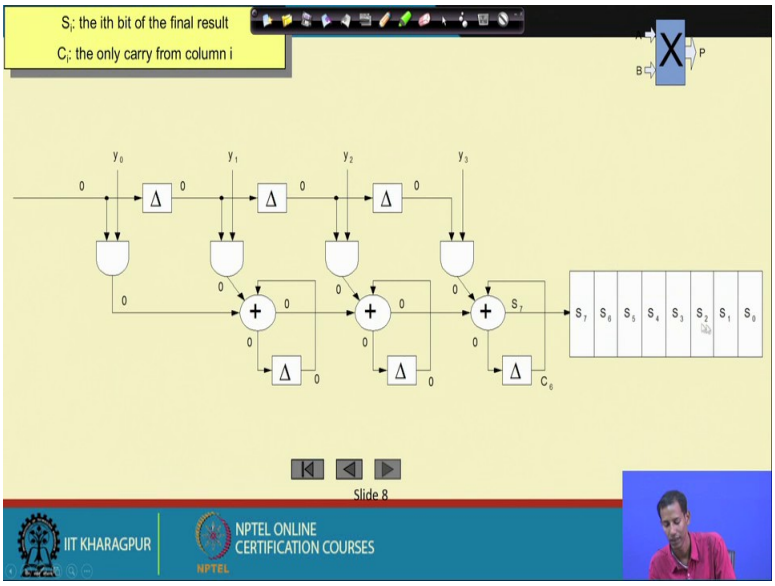


Then we have another architecture that is serial slash parallel architecture. So, serial slash parallel architecture is that; that means,. So, here this X 0 Y 0 X 0 Y 0 X 1 Y 1 sorry X 1 this X 0 Y 2 Y 1 X 0 Y 2 and X 0 Y 3 all together they are ANDed and they basically produce this initially they will produce this is basically this blocks are the delay blocks right. So, initially what will happen? X 0 that will be multiplied with Y 0 and that will come over here. So, in this particular case at that time X 0 is not passing over here, so; that means, it will be 0 that will be multiplied with one. So, it will be 0. So, it will produce S 0; so 0 over a 0 over here. So, this S 0 will come here.

So, in the next case what will happen? X 0 will be coming to this, and X 1 will be coming to this so; that means, at that particular clock phase what will happen? X 0 into Y 1 plus X 1 into Y 0. So, you see in this particular case what is needed? That P 0 sorry P 0 is nothing but this X 0 and Y 0 or a 0 and b 0 in this case. In the next what will happen this A 1 B 0 that will be added with A 0 B 1 right. So, this has been generated using; that means, here what is happening? X 0 Y 1 plus X 1 into Y 0 that is basically happening over here and that is stored in this particular and then again all this are zeroes. So, then S 1 will come over here than in next clock cycle what will happen?

In the next clock what will happen? This X 0 will now come over here X 1 will be here and X 2 will be here so; that means, X 2 Y 0 then X 1 Y 1 and X 0 Y 2 ok. So, all together they will be added and then that will come as S 3 over here.

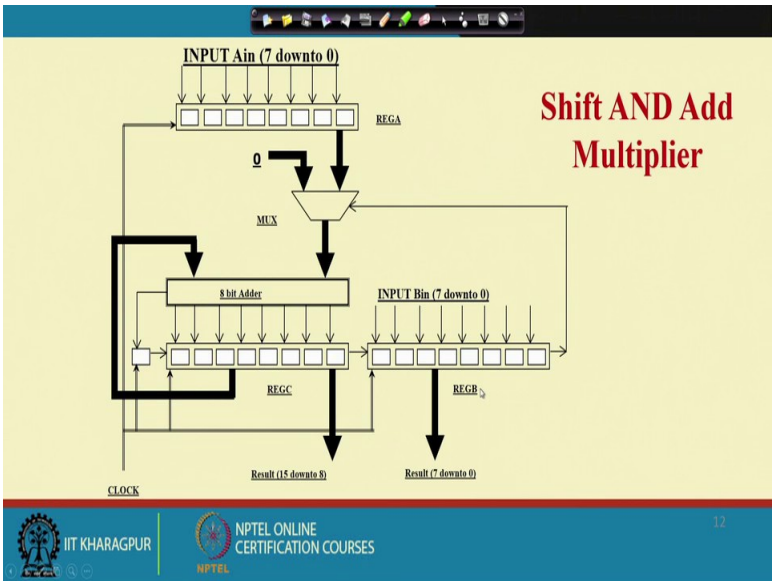
(Refer Slide Time: 29:24)



So, in this manner all this; that means, now S 0 sorry S 0 will be shifted, S 2 will come over here. So, if I just go back. So, S 0 will be basically this is nothing but the shift register ok. So, initially we have started with S 0. So, in the next S 1 comes, S 0 shifted over here S 1 fills with there. Then in the next S 2 S 2 over here S 1 over here S 0 over here; then in the next S 0 over here S 1 over here S 2 over here S 3 over here. Then again in the next S 0 will come S 1 S 2 S 3 S 4 in this manner. So, at the very last so, S 0 will come over here and S 7 will fill here so, after that I will get the corresponding output of 4 bit multiplication operation.

So, at that time what will happen? The corresponding X 0 what will be the input signals of X 0 that will be X 0 X 1 X 2 X 3 then 0 0 0 0. So, that is the input sequence for X 0 for this particular case. So, this is the combination of serial as well as parallel so; that means, the this parallely basically I am computing this partial product ok, then I am doing addition in serial manner. So, that is why this is serial and with combine of parallel operation ok. So, then what I said that, there is this shift and add multiplier these are the basic multiplier ok.

(Refer Slide Time: 31:00)



So, again we will see this shift and add multiplier along with other multiplier architecture in a next class.

Thank you for today.