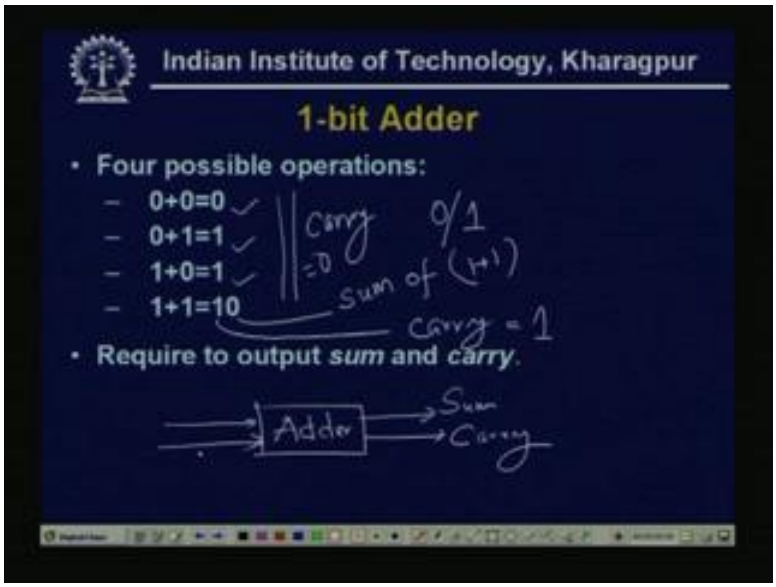
Digital Systems Design
Prof. D. Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 29
Design of Arithmetic Circuits

This lecture is on Design of Arithmetic Circuits, in this class we will read, how we can design the different type of adders, multipliers and other arithmetic units.

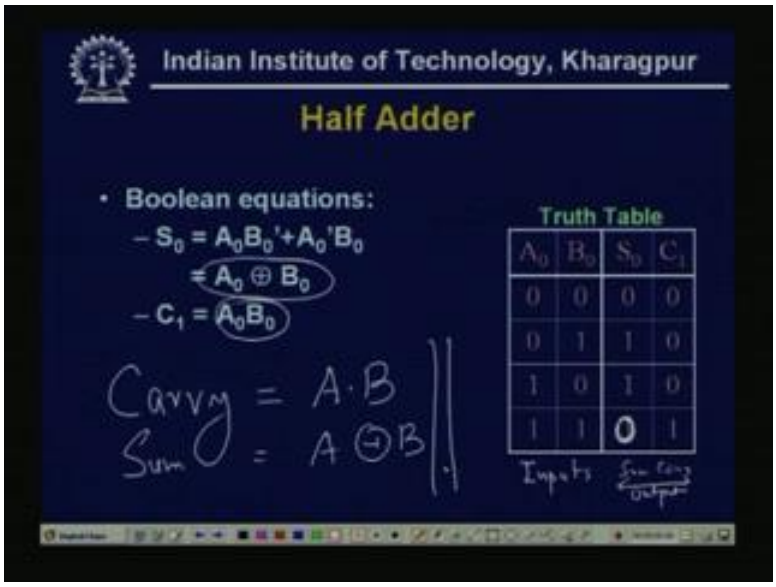
(Refer Slide Time: 01:12)



As, the adder is the most important and the mostly used digital circuits in the digital system, so first we will see, that the different type of adder design and it is performances and which one is suited for what type of design etcetera. So, first we start discussion on the adder design, as all we know, that what is the truth table of 1 bit error, so if we know that if it is a 1 bit mean the bit can be either 0 or 1. So, there are different, four possibilities that either 0 is can be added with 0 0 plus 0 is 0, 0 plus 1 is 1, 1 plus 0 is 1 in all this three cases the carry is 0, carry is 0.

Now, 1 plus 1 is 1 0, so if we see that binary this is 1 1 1 and actually 1 0 means this is 2 represented in binary. So, this is 1 0, so 0 is the result or the sum of 1 1 plus 1, sum of 1 plus 1 and this 1 is the carry. So, in this case carry is 1, so these are the 1 bit adder functions, so from here, only one thing we can tell that we require two output of a adder, that means if I draw a black box, that if it is adder circuitry, then actually I have two output one is sum, one is carry and say I have two inputs to be added.

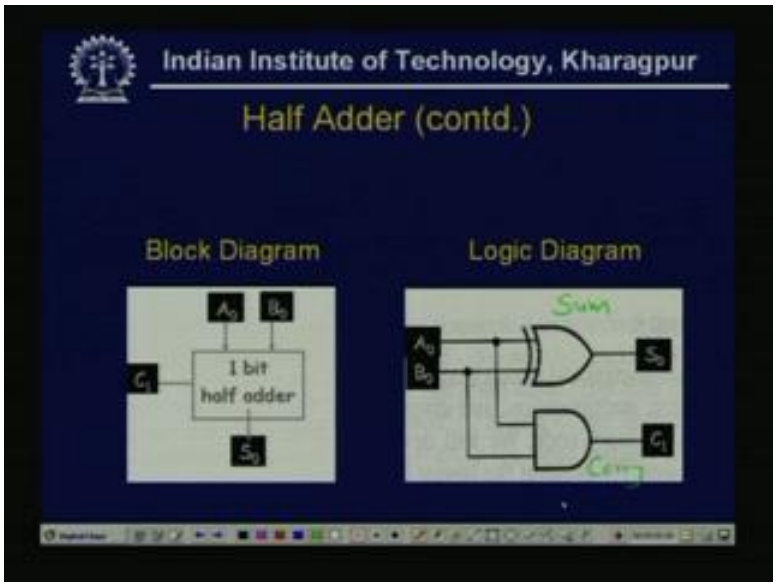
(Refer Slide Time: 03:39)



So, now we see what we call that half adder, see that just now the equations I got that what is the thing, that if I represent this as a truth table. So, I have two input functions these are my in the truth table that A 0 B 0 is the these are my these are the inputs, two inputs and S 0 and C are the sum and carry these are my two output. So, these two operands AB that can be 0 0, 0 1, 1 0 or 1 1.

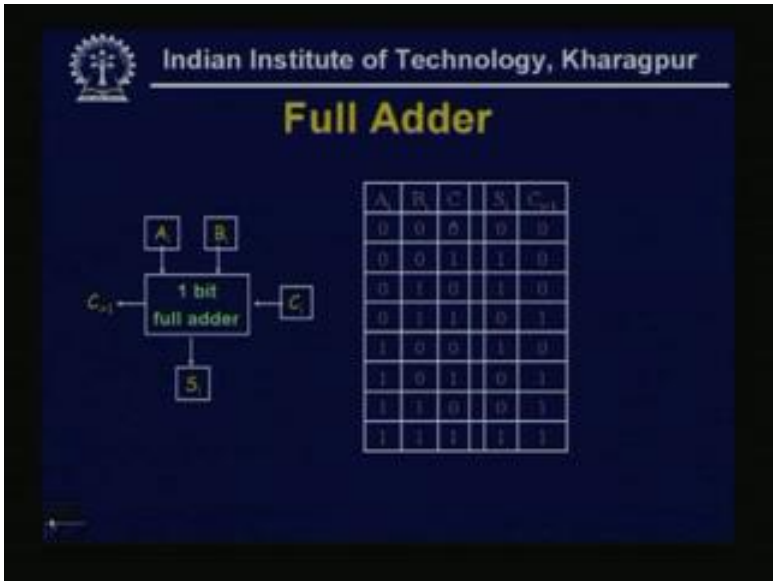
So, all ready we have seen that 0 0 sum is 0 carry is also 0, 0 1 sum is 1 carry is 0, 1 0 means 1 plus 0 sum is 1 carry is 0, now 1 plus 1 sum is 0, sum is 0 carry is 1, so this is my truth table. Now, if we realize this truth table or if we minimize this one, we will get the functions. The sum functions is A XOR B or that A 0 XOR B 0, so the two XOR of two operands this is my sum, and the carry will be simply the AND function means C is AB, that means, C 1 is A 0 B 0. So, I get the carry, I get the carry is AB if A and B I want to add and the sum is A XOR B, so these are the two things I need, now if I draw the diagram the half adder.

(Refer Slide Time: 06:11)



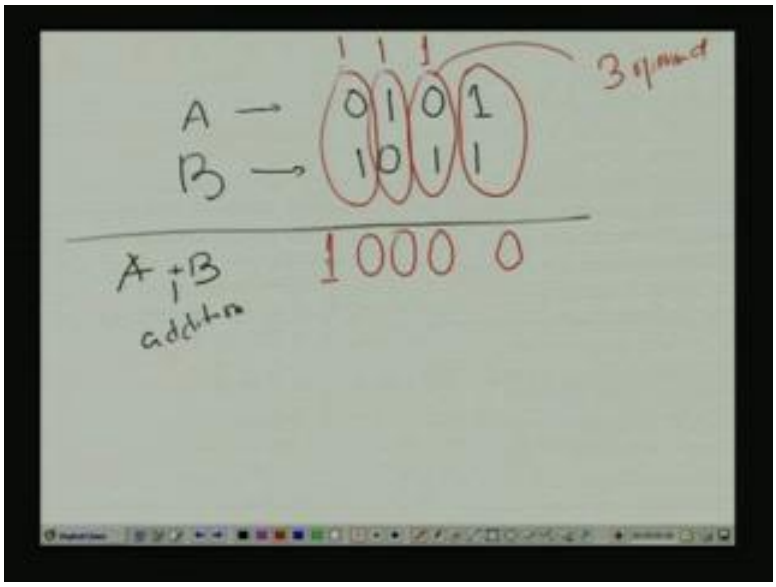
So, the block diagram is, this is a 1 bit half adder and these are the two inputs AB and two outputs are S and sum and carry, now what will the gate level diagram, as all ready we have seen the sum is nothing but A XOR B, so this is XOR function. So, this is my XOR the sum and this is my AB means this is my carry, so 1 XOR gate, 2 input AND gate.

(Refer Slide Time: 07:06)



Now, what is the full adder because, that now we know that, if it is a instead of 1 bit, I need a 2 bit.

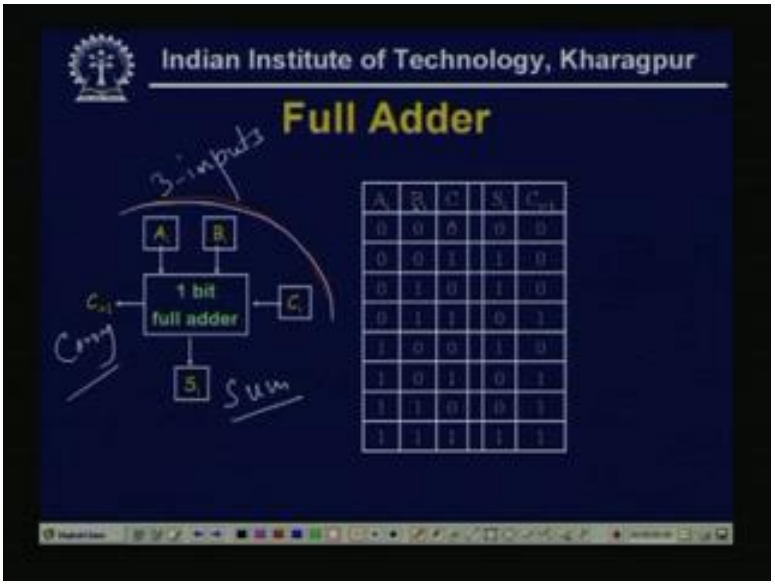
(Refer Slide Time: 07:36)



So, concept is that say, if I have say my A is a 4 bit say 0 1 0 1, B operand is also a 4 bit say 1 0 1 1, if I want to add A say plus B here it is addition, plus is addition then actually this one is a 1 bit adder, this 1 is a 1 bit adder, this is a 1 bit adder, this is a 1 bit adder. But see, here 1 plus 1 is 0 and I need a carry 1, because here carry is 1, so actually here it is not a two operand rather it is a, it is a three operand. So, this one is a three operand not two.

Again this is a 1 plus 0, 1 plus 1 again 0 carry 1, now again 1 plus 1 this is a three, this becomes a three operand adder, then or three input adder, then again it is 0 again it is 1, so 0 then carry 1, so this type of circuit we will get.

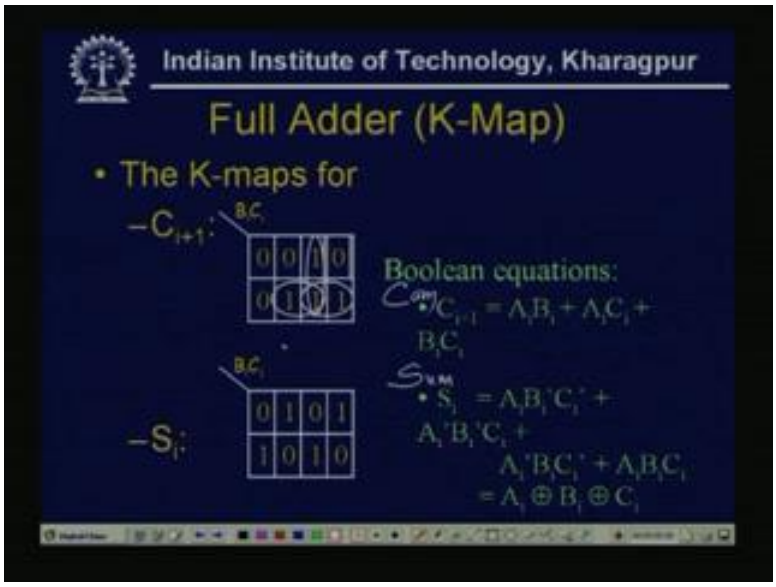
(Refer Slide Time: 09:18)



Now, we see, actually here this sometimes from the second bit onwards for the first bit it is a 2 bit adder, but the second bit onwards it is a 3 bit. So, instead of 2 bit now the full adder is become full 3 inputs that A B and Ci. See this is the 3 inputs and this is one sum is 1 output carry is the next output, now as it is a 3 input, so actually this is the carry bit coming from the previous bit sum, so as it is a 3 inputs, so Ai Bi plus Ci, if I write in this way.

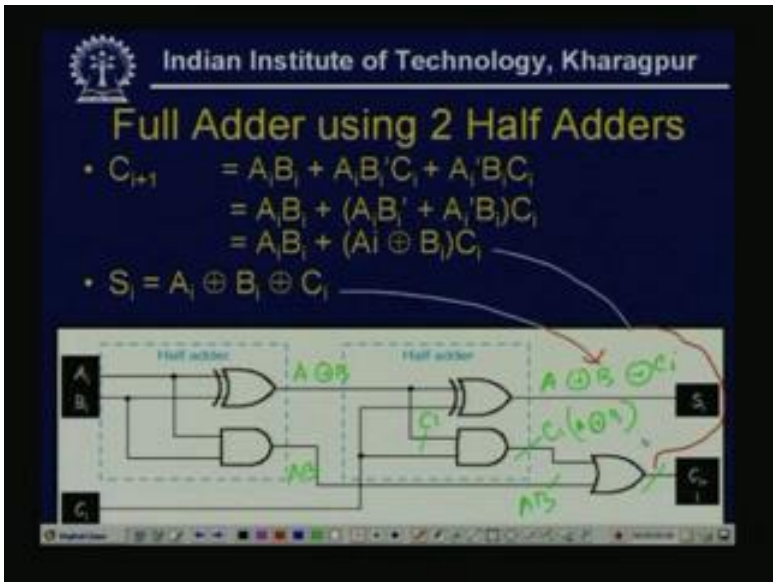
So, it has 2 to the power 3, eight such combinations and for this eight we can put the sum and carry in this way. And from there, we can realize the output functions the sum functions and the carry functions are the sum are A B dash C dash, A dash B dash C, A dash BC dash plus ABC.

(Refer Slide Time: 10:52)



So, this is my sum, this is my, this is my sum functions, sum, similarly, the carry we get C is AB plus BC plus CA this is my carry. So, these are the two outputs of the full adder all ready earlier also we discussed this thing all of we know this. So, now this full adder can be implemented by using two half adders.

(Refer Slide Time: 11:23)

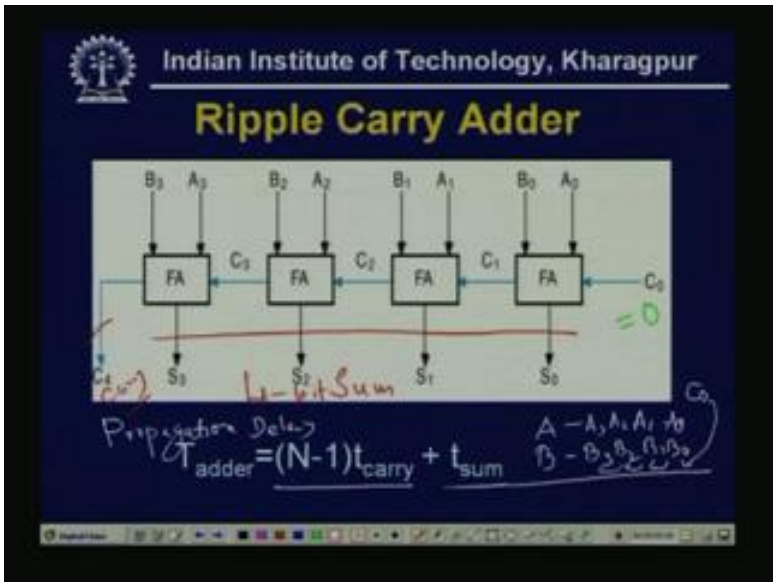


See that the carry is we can write AB plus, if it is 2 bit earlier if it is a 2 bit sum, so this is A XOR Bi, Ai EXOR Bi and this is with Ci, so this is nothing, but that Ai AB plus BC plus Ci and similarly that sum, this is the actually XOR of the 3 inputs. So, if we add two half adders, so where here Ai Bi is, so actually we are doing that bit wise addition, so Ai Bi, I get the sum of the first bits and this is the AB is the carry. Now, again this is the sum of the first 2 bits, LSB's that going to that input of the XOR, this is the carry of the previous 2 bits.

So, this is the carry of the previous 2 bits and I need another one, so for the carry itself, so actually if we put two half adder side by side and the output of the previous one is fed to the next one, then actually I will be getting the sum and the carry of the full adder. Say actually it realizes…so this is my, if I write in this way, this is my A XOR B, A XOR B and now Ci is comes here, so this is A XOR B XOR Ci, so this is my sum, this is the thing, this is the thing.

And the carry bit say this is my, this is my AB and this is AB plus this is my OR and Ai, so this output is Ci, this output is Ci and this output A XOR B. So, this is Ci into Ci dot A XOR B and this is my AB, so this is a plus, so this will give the carry of this realizes the carry here. So, by operating two half adders, I can get a one full adder, now we one by one we start reading different type of adder design, so first one we will see the Ripple carry adder.

(Refer Slide Time: 15:49)



So, this is one example of a 4 bit adder, that means the two operands A and B, A is, A is A 0 A 0 A 1 A 2 A 3 similarly B is B 3 B 2 B 1 B 0, so these are 4 bit operands, so what we want, so this is a, A 0 B 0 and the as if that sum carry is coming for the LSB this carry is 0, the initially the carry is 0. So, this is a full adder, so one carry is coming from the as if this is the initial carry, now the carry is propagating here again this is a full adder, now the carry comes here this is a three full adder, again this is a full adder. So, if we see now this full adder it has 3 inputs the LSB A 0 B 0 and the initial carry which is assumed as 0 initial.

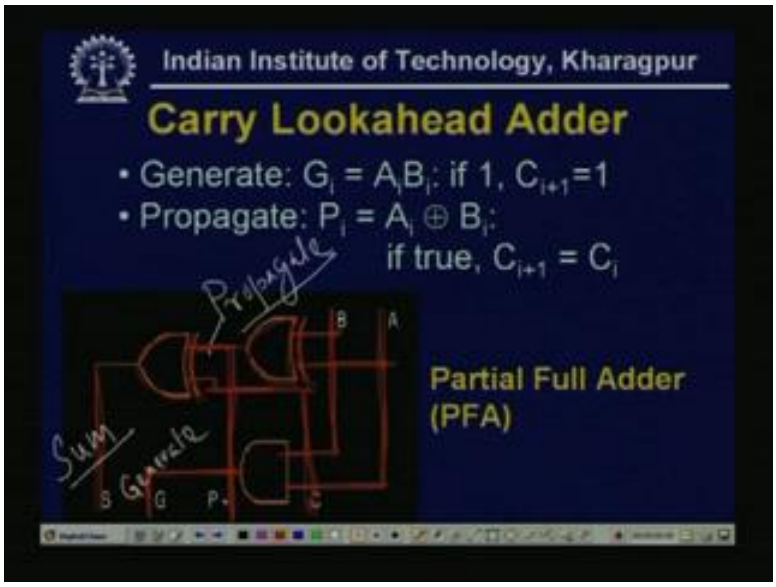
Then the sum is A 0 B 0 plus C 0 this is my the first bit or the LSB bit of the 4 bit sum S 0, now the carry is propagated. So, this is my C 1, then the second this is for the second LSB; that means, A 1 B 1 and C 1 is the 3 input S is the sum. Then the carry is propagated the C 2, similarly A 2 B 2 C 2 this is S 2 C 3 A 3 B 3 S 3 then C 4. So, these S 3 S 2 S 1 S 0 is my sum 4 bit sum, this is my 4 bit sum and this is my carry, carry of the the adder.

See why the name is as if the there are some rippling effect, that means carry is propagated from the input to output means from LSB side to MSB side, so this called the, that is why it is called the ripple carry adder, then what is the propagation delay. So, this is my propagation delay of the 4 bit or say M 1 general N bit ripple carry adder, see here if it is a N bit ripple carry adder, say for this case N equal to 4, so actually carry is ripple, rippled thrice C 1 C 2 C 3.

So, the delay is needed for this, in this case it is three; that means, for N, N bit adder it should be N minus 1, so N minus 1 into t carry, if the propagation delay for the carry bit is t carry then it is total is N minus 1 into t carry and this is the propagation delay for the sum. So, the total time or the processing time or the propagation delay of the t adder ripple carry adder is the N minus 1 into t carry plus t sum.

So, one thing is clear, that if N is very large, say I want a one twenty eight bit adder and then this should be a the delay is quite large, because this is one twenty seven into some carry propagation for 1 bit and this is for sum. So, actually for carry propagation delay is, is huge, now this is a carry look ahead adder.

(Refer Slide Time: 20:15)



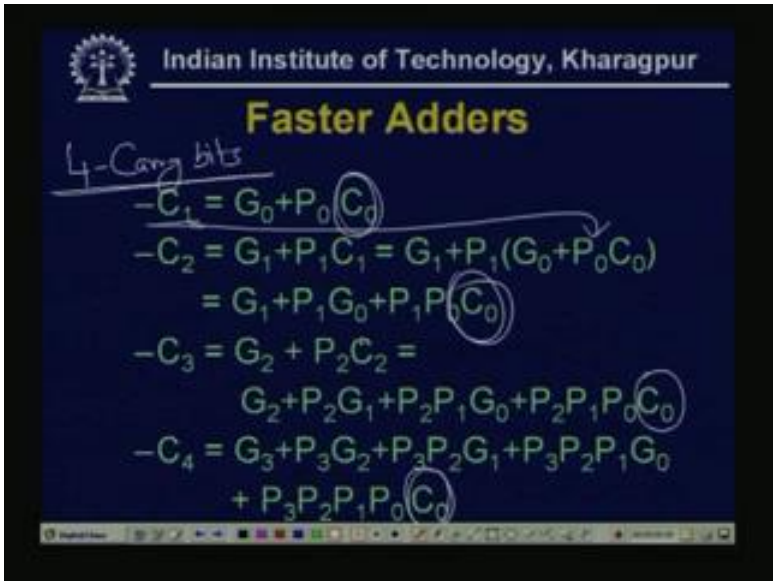
So, as the carry, if the carry is propagating bit by bit, then the, propagation delay is more the adder will slow, so a new type of adder, it is called the carry look ahead, that is again proposed the design is proposed. So, for this purposes as if we proposed 2 G functions, 1one is called generate another is called propagate, so if we remember the sum and carry functions of half adder. So, actually the two functions one is AND that is my for my carry C is AB and the sum is the XOR of 2; that means, S equal to A XOR B.

So, from there we proposed two functions generate, Gi is Ai Bi, if 1 then C Ci plus 1 equal to 1, and propagate is Ai XOR Bi, if true means 1 then Ci plus 1 equal to Ci. So, if it is partial full adder, so now this thing is utilized this concept is utilized generate and propagate, this two is utilized here that see that AB. See that these are the two operands

A and B and this is a AND gate, now this becomes this is my Ai Bi, this is XOR and this is my carry carry line.

So this is A B C, so this is my A XOR B XOR C and that is my sum, so this is my, this is my sum and AB, so propagate or carry and that we are telling that AB is actually Gi, the generate. So, this is my generate line, this is my generate line and this is my propagate Ai Bi. So, mainly two new functions as if generate and propagate has been proposed, so this is the propagate and this is generate. Now, using these concepts of generate and propagate, we introduce a new type of adder called the faster adders.

(Refer Slide Time: 23:48)



So, this is look ahead adders, so as if the we are seeing the carry earlier before it propagates we are predicting first, this is the concept. So, the carry bit C 1 are pre-designed, so if it is a 4 bit adder say C 0 is the initial carry and the 4 carry bits these are the 4 carry bits are generated. Though this 4 carry bits are generated during the addition of the 4 bits of A and B, but as if we know the addition function, the sum and carry functions. So, from there as if these are pre-determined, this is the basic concept of the faster adder.

So, as if these are not propagating, these are designed separately, so in this way the first carry bit C 1 is G 0 plus P 0 C 0. Similarly C 2 is G 1 plus P 1 C 1, so this is nothing, but G 1 plus P 1 and C 1, I can replace, so I can replace C 1 by G 0 plus P 0 C 0, so this is G 1 plus P 1 G 0 plus P 1 P 0 C 0. Similarly, when I am computing C 3, I know it is G 2 P

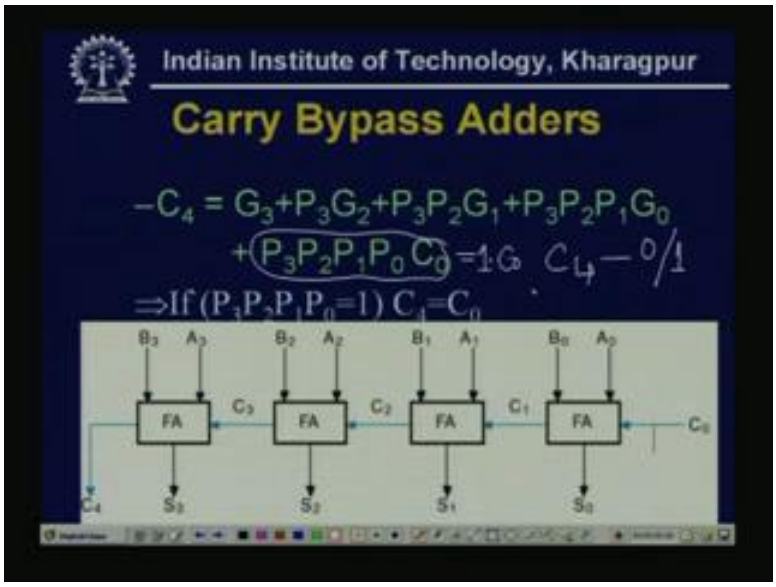
2 C 2, again C 2 is replaced by its expression G 1 plus P 1 G 0 plus P 1 P 0 C 0 and I am getting that G 2 plus P 2 G 1 P 2 P 1 G 0 P 2 P 1 G 0 C 0, similarly C 4.

So, from this 4 carry bit expression, so as if before we do the addition, though the carry is generated during the addition, but the carry bit functions, as if these are the carry bit functions, they are pre-computed using the two function propagate and generate that just. Now, we have proposed the GP and these are the, now see that C 1 is the function of C 1 is the function of the first generate bit and the carry bit and the propagate bit initial.

C 2, C 2 is again the function of see the first carry bit, first carry bit and only the propagate and generate function, C 3 is again the first carry bit, first carry bit and the function and the functions of the generate and propagate. So, one thing to be noted that here the, the next carry bits are not the function of the previous carry bits. They are only dependent on the initial carry bit C 0 only C 0 is here, so then as if, when I am computing C 3 that C 2 is not needed.

So, we have not to wait until that C 2 comes, but based on these functions the G and P the generate and propagate, I can I can compute C 1 C 2 C 3 C 4 parallel. So, this is obviously, this will be faster because I have not to wait for the propagation of the carries. Just now what we have seen the for ripple carry as the carries are propagating. So, there the adder becomes slow, it has a huge propagation delay, so here the main concept is we are not waiting for to generate the Nth carry bit until the N minus 1th carry bit is comes or feeds to the that particular addition bit, bit additions. So, this is the concept, so now based on this concepts we propose a new design called the carry bypass adders.

(Refer Slide Time: 28:39)

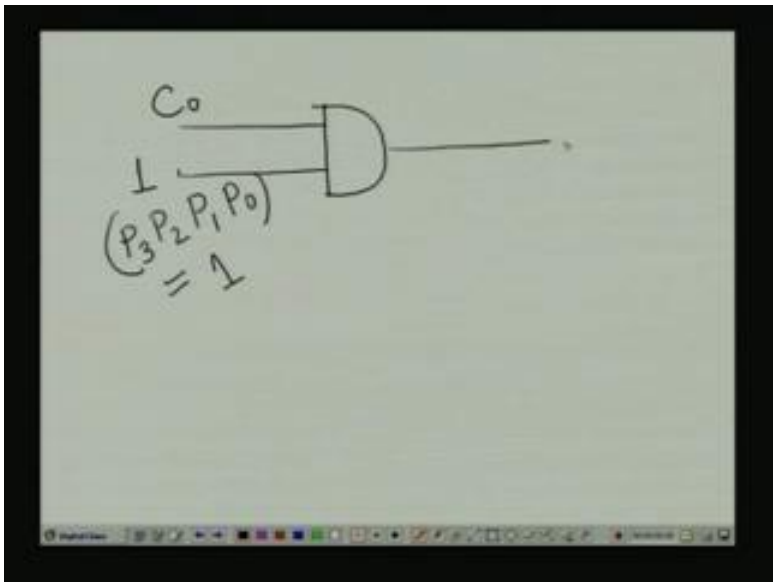


Carry bypass means, as if intermediate carries we are bypassing, we do not need to wait to generate the intermediate carries, because from the design or form the introduction of two generate and propagate function, we can design the all the carry bits parallel.

So, if we consider the C 4 the last carry bit, the which is the worst case situation. We have already seen C 4 in this type of function. So, from these expression just we see we notice one thing, that the function is see that G 3 P 3 G 2 P 3 P 2 G 1 P 3 P 2 P 1 G 0 P 3 P 2 P 1 P 0 C 0.

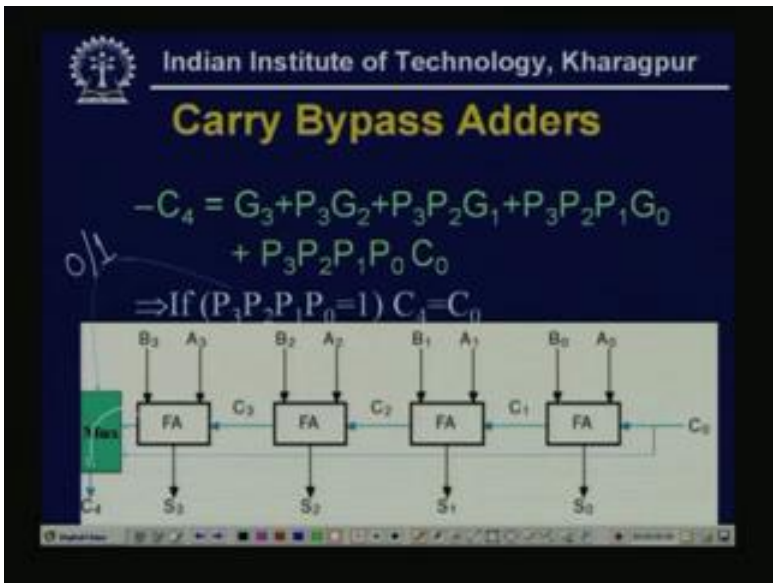
Now, it depends… here the factor main contributing factor is the this one, P 3 P 2 P 1 C 0, so my carry bit is either C 4 is either 0 or 1. Now see if P 3 P 2 P 1 P 0 is the term, only this term carry is there, the initial carry is there. So, if P 3 P 2 P 1 P 0 is 1 it is a AND function, so this will become, this will becomes 1 into 1 dot C 0. That means if C 0 is 0 then C 4 equal to 0, if C 0 is 1 then C 4 equal to 1 that means it is a AND function.

(Refer Slide Time: 30:54)



So, this is a, as if this is 1 AND and here it is a C 0 and here we are giving P 3 P 2 P 1 P 0. So if this is 1 that means, here 1 is coming. So, actually I will be getting here C 0 itself, so this is my C 4, so here C 4 equal to C 0. So that means, a P 3 P 2 P 1 P 0 is 1 C 4 is C 0.

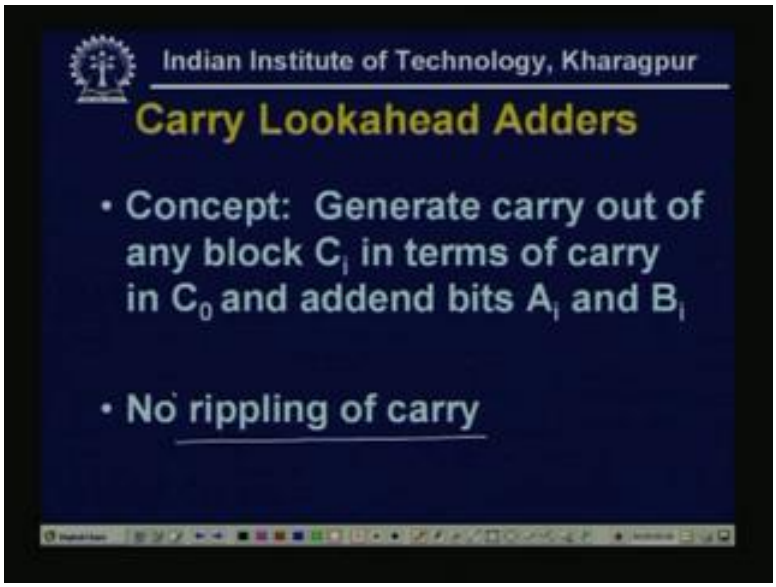
(Refer Slide Time: 31:48)



So, now say as if one multiplexer is added here and see how, I bypass why it is carry bypass adder, that if it is 1 then as if that C 0 becomes the C 4, so this is a multiplexer has two inputs if P 3 P 2 P 1 P 0 equal to 1. So, directly C 0, C 0 is the becomes the C 4, C 0 becomes the C 4. So, actually at least for this condition I am bypassing the a carry generation, intermediate carry generation.

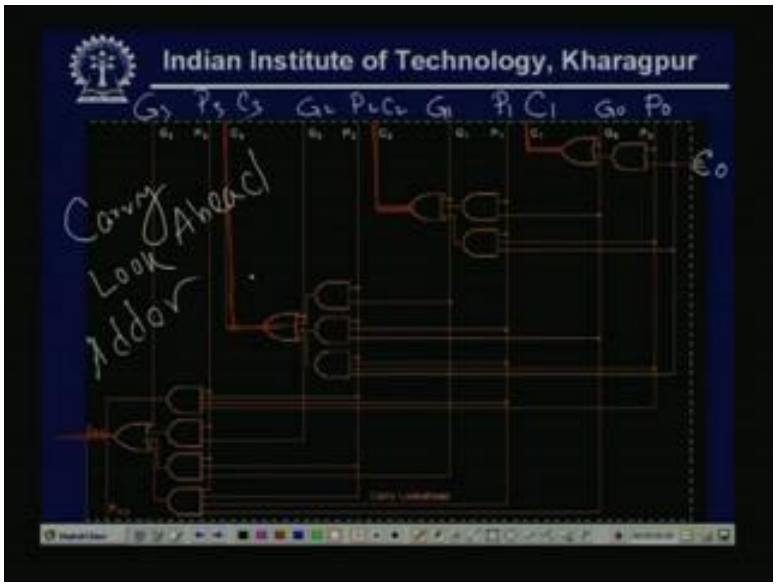
So, actually for this condition, that it will be a if P 3 P 2 P 1 P 0 equal to 1 for this condition, that it will be a very fast adder because I am not waiting to generate the carry. And similar thing happens for the other intermediate carries also, so this is the concept of the carry bypass adders, And obviously, this is my select line if it is 1 then only it is select line can be 0, 0 or 1. If it is 0 then the actual carry, that is going there and if it is 1 then this will be going there, so carry look ahead adders….

(Refer Slide Time: 33:58)



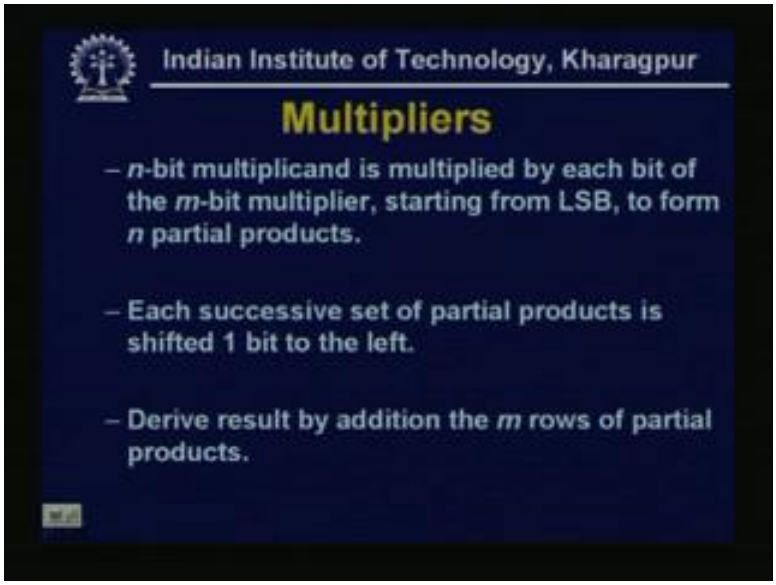
The concept is generate carry out of any block Ci in terms of carry in C 0 and addend bits Ai and Bi no rippling of carry. So, here the advantage is that propagation delay is very less, there as there is no rippling of there is no rippling effect of carry. So, here the propagation is propagation delay is very less. Now, these are the design that mainly the gate implementation of the 4 bit carry look ahead adder.

(Refer Slide Time: 34:34)



So, here it is a 4 bit, so we can what we can do again the inputs are G 0, so this line is G 0 P 0 and carry is C 1, here the another carry is C 0 this is G 1 P 1 G 2 P 2 is a C 2 and this is G 3 P 3 and C 3. So, if I consider C 0 this is G 0 P 0 C 0 the carry is C 1 is generated, similarly then for the next bit G 1 P 1 and C 1, so the next carry is C 2 is generated. Similarly here, so as if it has the 4 stages C 3 and this is my that next P 0 3 or the next carry bit is generated from here and the P bit comes here. So, this is the circuit of that carry look ahead adder, this is my circuit of carry. Now, so some of the adder circuits, common adder circuits we have just now seen.

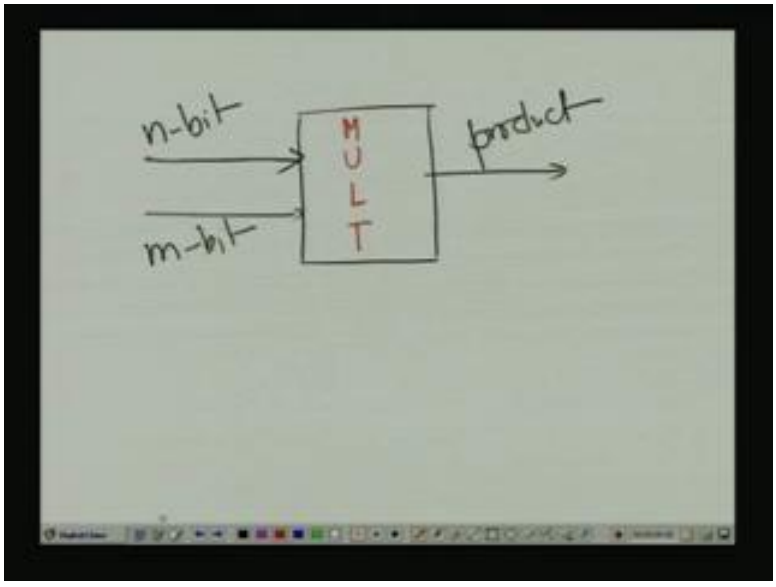
(Refer Slide Time: 36:27)



Main concept is that normally, a full adder can be realized by two half adders, if it is a 4 bit adder or N bit adder the actual, it is a ripple carry adder and the propagation delay is huge, just to avoid the propagation delay or to get the first as if the intermediate carries can be generated from the first bit itself, first carry only it depends on the first carry. So, that we can bypass type of bypass adder or carry look ahead adder, that means that carries are computed from the first bit itself not the intermediate sum bits lost by the propagation of the carry bit, so it is it becomes faster.

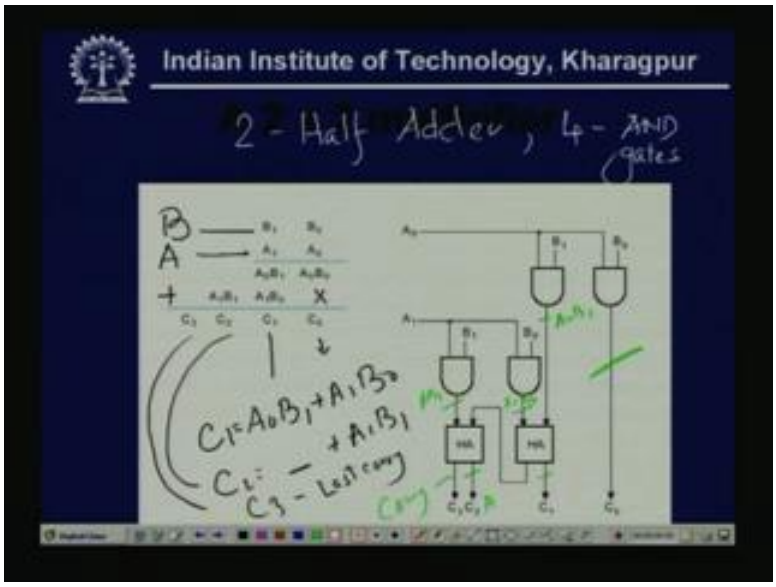
Now, all of you know that multiplier is nothing, but the add and shift the adder and shifter. So, mainly for the design circuits, we the arithmetic processing we need that adder and multiplier, so now we see that how multipliers are designed. So, multiplier has normally there are two operands one multiplicand and another is multiplier and the sum result is the product. So, n bit multiplicand is multiplied by each bit of the m bit multiplier. That means here we have taken a the block diagram, we draw…

(Refer Slide Time: 38:08)



Then, actually the block diagram, see this is a multiplier circuit, it is a n bit multiplicand, it is a m bit multiplier and we get a product. So, this is my, this is my multiplier MULT circuit, each successive set of partial product is shifted 1 bit to the left. And derive result by addition the m rows of partial products, so, if we remember the multiplication function, the normal way of multiplication we do, it is nothing but the shift and add, we see one example.

(Refer Slide Time: 39:04)

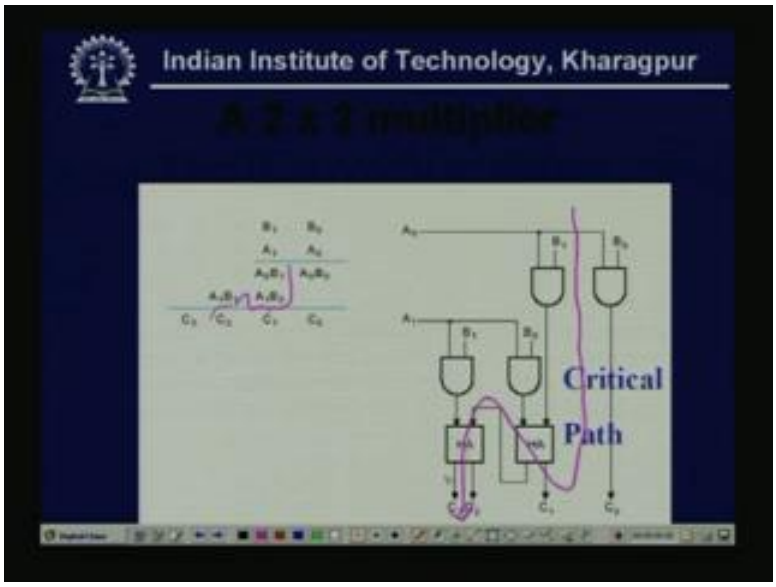


Earlier also we have seen the multiplier circuits, say here 2 bit multiplicand and 2 bit multiplier we have taken. So, this is my B, B is B 1 B 0 this is my A is A 1 A 0, 2 bit, so first is A 0 B 0 and A 0 B 1, then I shift, I shift and then A 1 B 0 and A 1 B 1 now we have to add. So, this is A 0 B 0, so as it is A 0 B 0 is kept here and then A 0 B 1, so it is A 0 B 1 plus A 1 B 0, so this is my C 1.

Similarly, C 2 is the carry of this sum and plus A 1 B 1 and C 3 is only the last carry. So now if we see this thing, so the first one is A 0 B 0, so we see this is only one AND, so this is A 0 and B 0, so this is my first bit of the product C 0. Now, C 1 is A 0 B 1 plus A 1 B 0, so we see that A 0 B 1 now the second bit is there this is A 0 B 1 then A 1 B 0. So, the second bit A 1 B 0 this is my A 1 B 0, so this is nothing, but a half adder, two there are 2 bit sum and this is a sum.

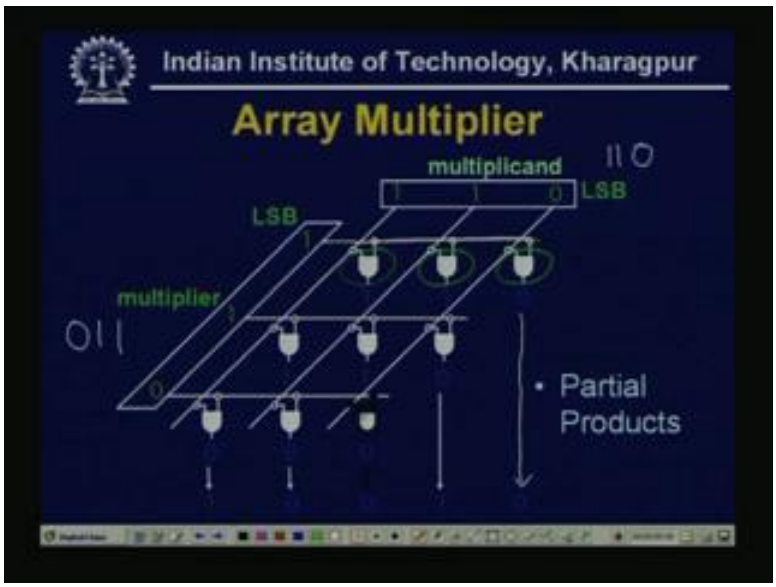
The carry is propagated to the next half adder and this is A 1 B 1, C 3 nothing, but the C 2 is nothing, but the A 1 B 1, so this is my A 1 B 1 and this is my last carry switching. So, this is the simple circuit we have needed, so for that 2 bit multiplier what we see, we need two half adder and 4 AND gates, so if we write we need two half adder and 4 AND gates.

(Refer Slide Time: 42:32)



So, this is the critical path, we see that A 0 B 1, A 0 B 1 then A 1 B 0, so why we are, see because this simply A 0 B 0, so this is, this one see if I draw this is, this one is added and then carry is go there. So, that means, A 0 B 1 then this half adder, carry is going to that half adder and this. So, the critical path is this, this path is the this is the critical path of the 2 bit multiplier. Now, we see a array multiplier this is only a bit by bit multiplier we have seen.

(Refer Slide Time: 43:39)



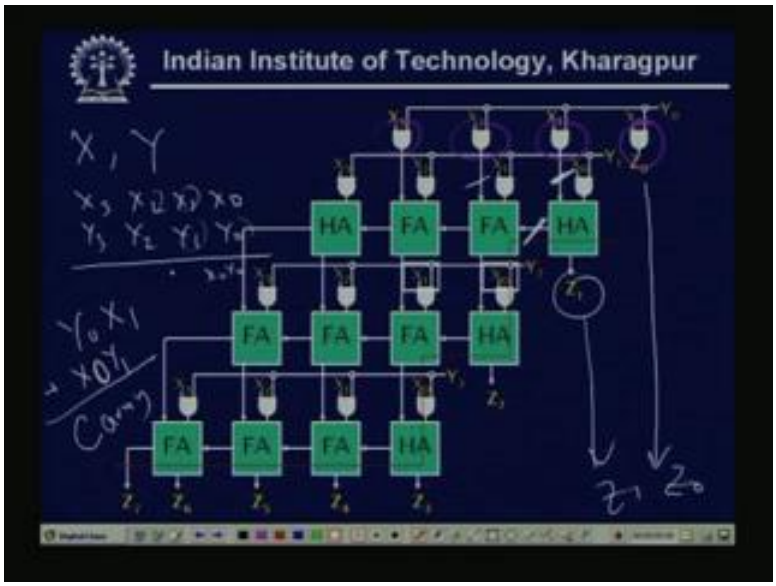
So, again there are multiplicand and multiplier, as if they are treated as a 2 bit, 2 arrays, so these are my multiplier, say this is actually multiplier is 0 0 1 1 this is 1 array of bits and multiplier multiplicand is 1 1 0. So, if I just from the previous picture what we have

seen that LSB, so this is the, this is the LSB is coming and this is a partial product, so actually I need another line here, so 0 this is 0 dot 1, 0 1 the first bit.

Now, here it will be 0 dot the second bit multiplier and this is, so these are the partial products, these are the these are the partial products in this case this will be 0, similarly this is the partial products. So, actually when LSB of the multiplier, these are the LSB of the multiplier is getting multiplied by all the bits of the multiplicand. So this line, this is these are the, these are the, the product partial products this three are the partial products that 1 1 0.

Now, the second multiplier bit is multiplied by the full multiplicand array, then these are the partial products. Similarly, these are the partial product, so now we need some adder, because this when once the products are available only we have to do the addition to get the product, the whole product of the multiplier. So, now as if the products are available and we give the adder circuit.

(Refer Slide Time: 46:48)



So, these are the partial products available, say if X and Y are the multiplicand and multiplier, if X, if X and X and Y are the multiplicand and multiplier. Then see these are these all these AND gates all these AND gates are giving the, the partial products, these are the these AND gates are giving the partial products. The first because Y 0 into this gives Y 0 X 0 Y 0 X 1 Y 0 X 2 Y 0 X 3, so as if the Y 0 is the first multiplier bit and

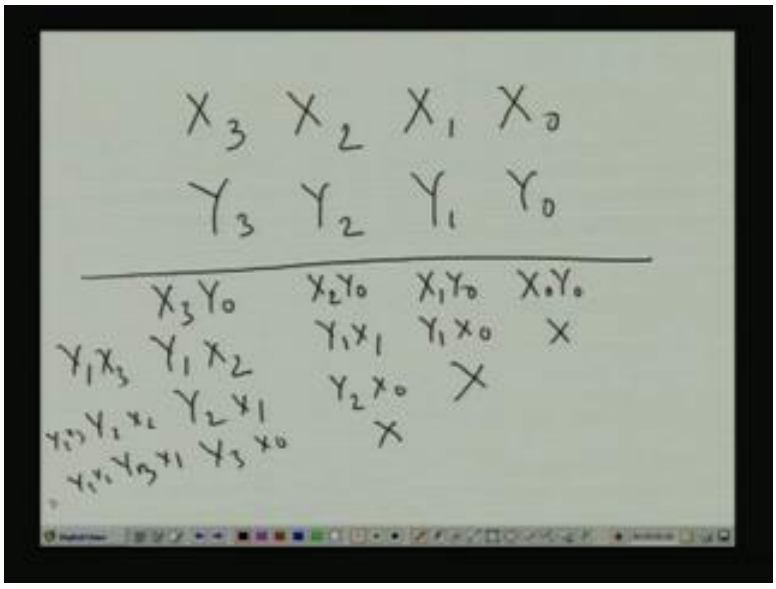
getting multiplied by all the bits of the first the multiplicand, so it gives the partial products.

Now, if I again if I draw, that thing; that means, that X, X 3 X 2 X 1 X 0 Y 3 Y 2 Y 1 Y 0, so Y 0 X 0 X 0 Y 0 will be as it is, so that is the Z 0 that is my Z 0 this is Z 0. Now, X 1 Y 0 X 1 second term will be Y 0 X 1, so if I draw the second terms Y 0 X 1 plus X 1 Y 1. So, this is the second half adder is doing Y 0 X 1, this is Y 0 X 1 and this is my Y 1 sorry Y 1 X 0 this is Y 1 this is Y 1 X 0. So, this is the half adder the Z 1 is giving, this is Z 1.

Now, I need a full adder, because, now I will the carry of this one the carry is propagated, so see here, now this is the this is the carry and second bit will be Y 0 X 2. So this is Y 0 X 2 and, and this will be Y 1 X 1, so Y 0 X 2 and Y 1 because this bit is Y 0 X 2 and then Y 1 X 1 plus the carry of the previous one. So, this is a full adder circuitry two product two product lines or product bits and this is carry, so this will give you the Z 2. Now, similarly the full adder and the last bit it will be a half adder.

And again that, when we will be adding because it is a 4 bit, so another some more array of products will be there. So, again this sum of this full adder will be added to the half adder and then it will be Z 2.

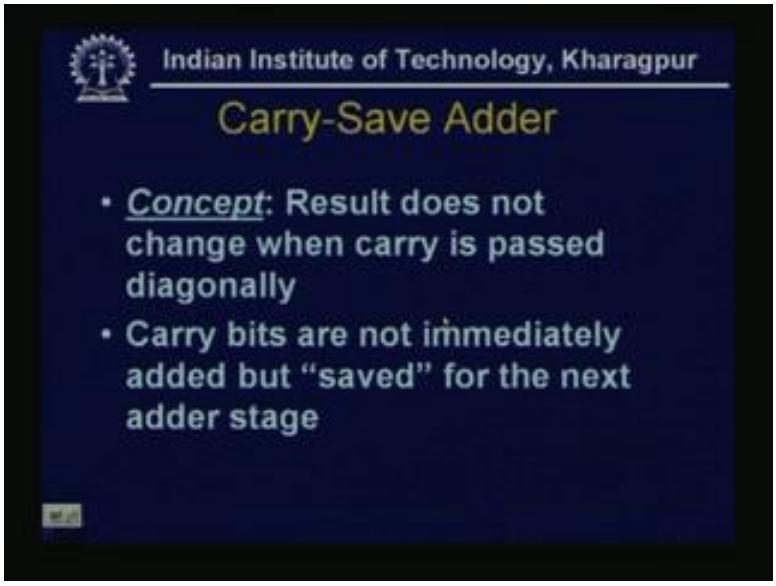
(Refer Slide Time: 50:59)



So, actually if we, if we try then to get the products, it will be say X 3 X 2 X 1 X 0 these are my multiplicands and Y 3 Y 2 Y 1 Y 0 these are the multipliers, so this will be X 0 Y 0 which is nothing, but Z 0. This term will be, see X 1 Y 0 this is say X 2 Y 0, this is X 3 Y 0, now again if we do Y 1 X 0 then Y 1 X 1 Y 1 X 2 Y 1 X 3, again it will be multiplied by Y 2 X 0 then Y 2 X 1 then Y 2 X 2 then Y 2 X 3.

Similarly, the last one will be Y 3 X 0 Y 3 X 1 Y 3 X 2 and Y 3. So, if we draw this product terms will be seeing that actually the, the we need a full adder and the half adder for the second bit, we need two full adders and half adder for the third bit in this way we will be getting. And again that the last, but second one is the two full adders and the last one is a one full adder, depending on what partial products we are getting. So, now we, we another type of adder we introduced, this is called a carry save adder.

(Refer Slide Time: 53:34)

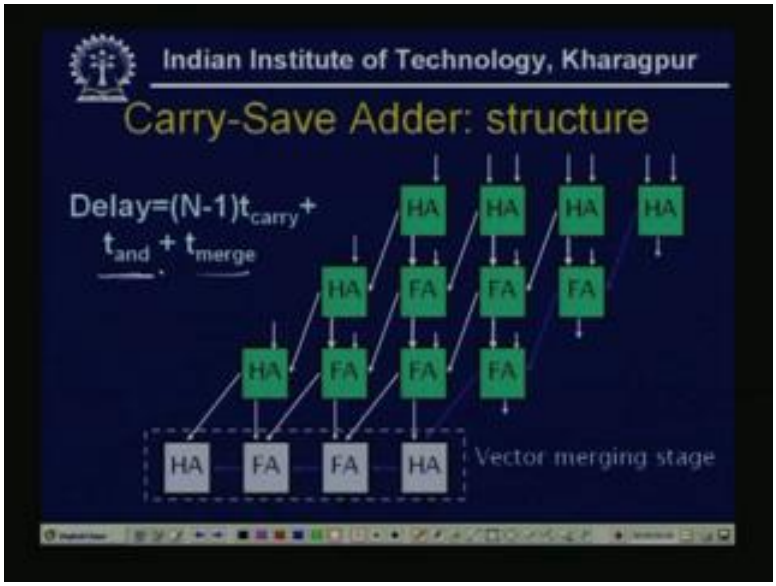


That result does not change when carry is passed diagonally, because for this multiplication circuit, what is clear that we have to propagate the carries of the different adders from the previous bits, so there must be some propagation delays for that to propagate the carry. Now, as if this is a carry save adder, so result does not change when carry is passed diagonally and carry bits are not immediately added, but saved for the next adder stage. So, this is the main concept behind the carry save adder and this is very much utilized for the design of a multiplier circuit. So, carry save adder structure is this is the thing….

(Refer Slide Time: 54:41)

Say here the input bits are coming, so the first array is a half adder, array of half adders, second array you see that one half adder and all the full adder, where that actually the carry is coming from the previous circuit. So, actually here from here the carry is coming, here again from this full adder the carry is coming, so here that carry is coming diagonally. See so, here the carry is coming, carry is coming diagonally here also the carry is coming diagonally, see here this carry is coming, here the carry is coming. So, this is the diagonally we are getting this thing….

(Refer Slide Time: 55:50)

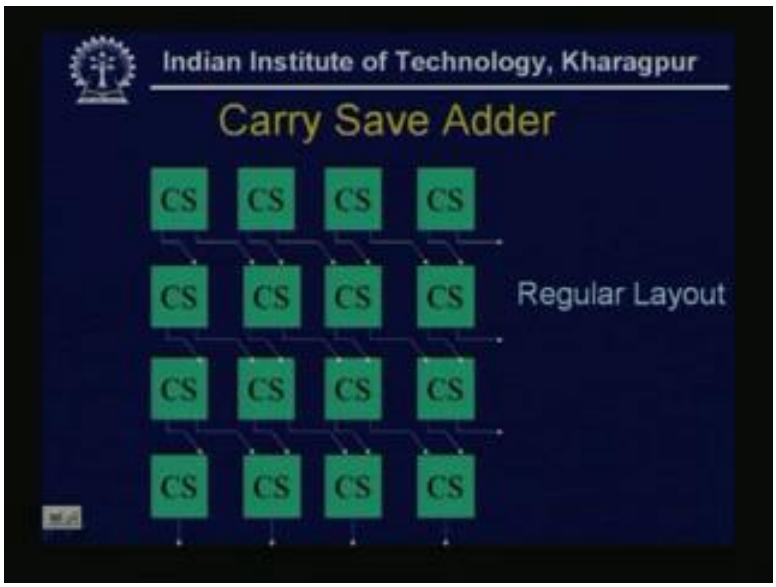


Now, this is called that, if we consider the last array this is a vector margin stage, because that as if, that using this one half adder full adder, full adder and two half adders

two full adders in the last array as if we are getting the actual product So, in this case the delay is N minus 1 just like the ripple carry adder, because the carries are propagating here also. That N minus 1 into t carry plus t AND and t merge, because here that just to get the, the partial products we have a number of AND gates, so which is actually the half adder circuit.

And we are merging this thing for the vector merge circuit, so the delay the carry propagation and the partial products and delay and for this merge in delay. So, this is the total delay of the carry save adder structure, so this is a regular layout the carry save adder.

(Refer Slide Time: 57:04)



These are the as if these are the adder circuits and we get the this as if 4 by 4 carry save adder circuits, so actually there are different type of adders which are being utilized in real life circuit. So, we will continue this lecture in the next class also as for the digital circuits, we need the efficient design of adders and multipliers. So, today's class we end here that how the multiplier can be realized by using the different type of or the carry save adders.

Thank you