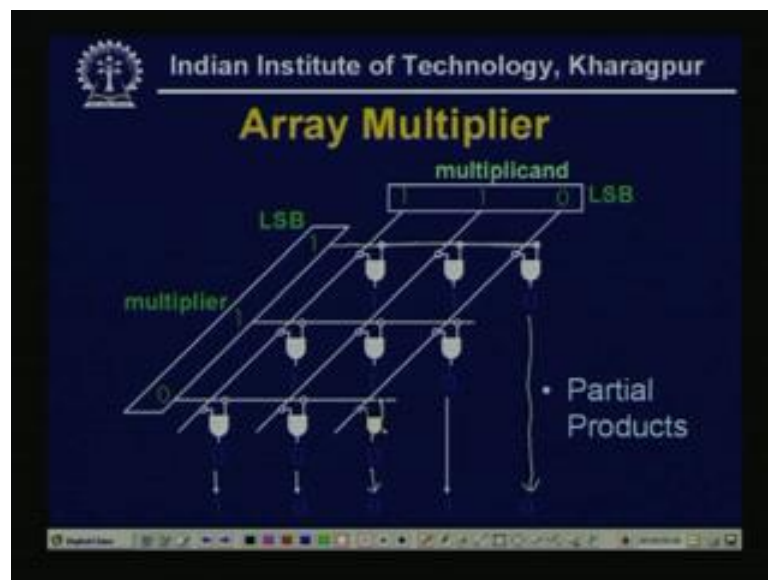


Digital Systems Design
Prof. D. Roychoudhury
Department Of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
Design of Arithmetic Circuits (Contd.)

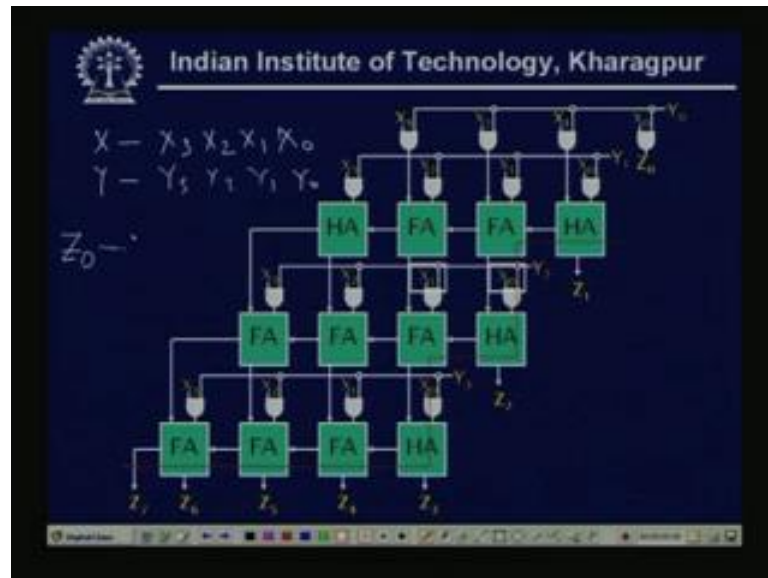
In the last class, we started discussion on Design of Arithmetic Circuits, there we have discussed the different type of design of adders. And we started the design of multipliers, also some on the multipliers we have seen.

(Refer Slide Time: 01:11)



Now, today will continue the design of multipliers and other arithmetic circuits. In the last class, we have seen the array multiplier, that where the multiplier and multiplicand. As if they are treated as two arrays and we have seen the partial products have been calculated.

(Refer Slide Time: 01:33)



And, then the architecture is of some of the arrays of half adders and full adders. So, here the structure we have seen that if it is a 4 bit multiplicand X is X_3, X_2, X_1, X_0 , and the multiplier Y again is 4 bit Y_3, Y_2, Y_1, Y_0 . Then last class we have seen that how the product Z_0 is calculated. As a set of half adders and full adders depending on the position of the addition of bits, say LSB when we are adding the 2 LSB bits it is half adder. And then for the LSB multiplier bit, the MSB is also a half adder, in that way we have calculated.

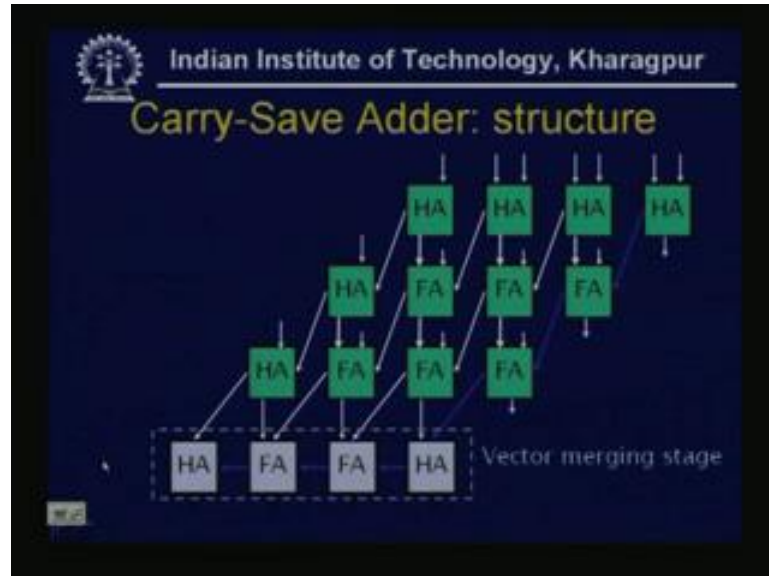
(Refer Slide Time: 03:01)

Carry-Save Adder

- **Concept:** Result does not change when carry is passed diagonally
- Carry bits are not immediately added but "saved" for the next adder stage

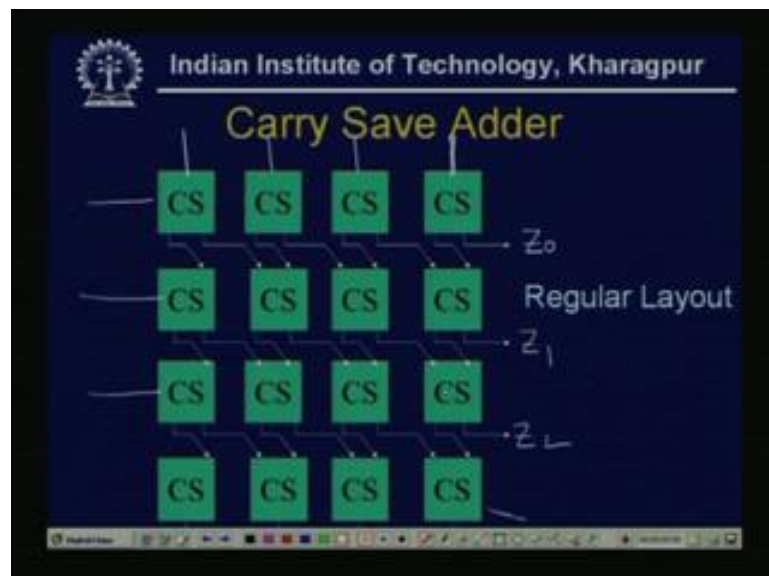
Now, carry save adder already we have discussed and the main concept was that carry bits are not immediately added, but saved for the next adder stage.

(Refer Slide Time: 03:12)



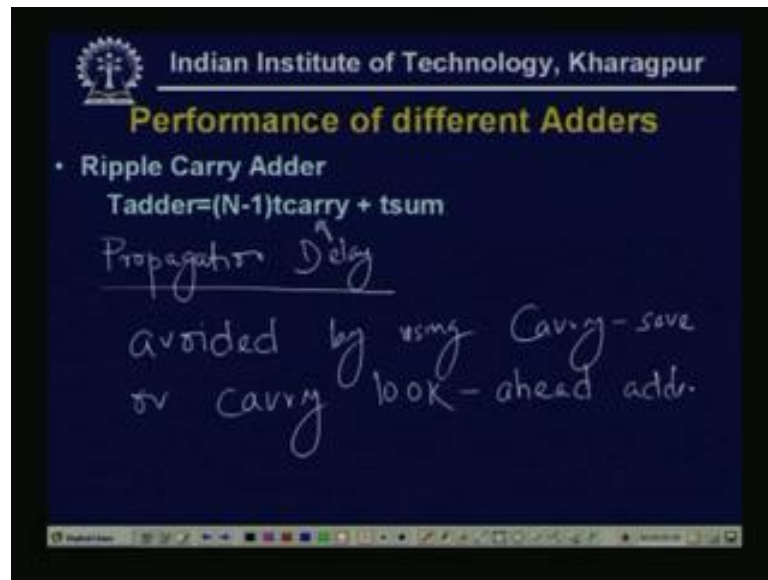
Now, using this carry save adder structure, some multiplier has been designed, where again that the first array is a set of half adders and the second array, third array and the fourth array is a set of half adders and full adders etcetera. So, this is carry save using carry save adder structure, where the last array is actually the vector, it is treated as a vector merging stage.

(Refer Slide Time: 03:56)



Now, if we consider the multiplier or the some regular layout of structure using the carry save adders, then we can this can be treated as if that, this is a, this is a carry save. So, as if inputs are coming here and there also and as if this Z_0 Z_1 Z_2 and this output Z_3 is also calculated, seen this way, this is regular layout structure.

(Refer Slide Time: 05:06)



Now, we see the performance of multiplier, so performance of adder the triple carry adder we have seen, that it is a actually that N minus 1 into t_{carry} plus t_{sum} . So, actually this is the propagation delay and S the ripple carry adder the carry is propagated. So, the main delay is for the carry itself, and that has been avoided by using the carry save or carry look ahead adder. So, now our approach is that, when we are designing the multiplier using the half adder and full adder circuit. Then, we want to use a carry save ahead adder in that full adder circuit, so what will be the performance of multiplier.

(Refer Slide Time: 06:20)

Indian Institute of Technology, Kharagpur

Performance of Multiplier

- **MXN Array multiplier**
Delay = $\frac{[(M-1)+(N-2)]t_{carry} + (N-1)t_{sum} + (N-1)t_{and}}$
Annotations: "Partial Product" points to the $(N-1)t_{and}$ term; "Carry Propagation" points to the $[(M-1)+(N-2)]t_{carry}$ term; "Adder" points to the $(N-1)t_{sum}$ term.
- **Multiplier Using Carry Save Adder Structure**
Delay = $(N-1)t_{carry} + (N-1)t_{and} + t_{merge}$

Already we have seen a M by N array multiplier, so if we remember that M by N array multiplier this was the array multiplier M was the multiplier, M number of bits the multiplier and N is the number of multiplicands. So, in this case, see the main concept is, here also the carry has been propagated from the first array to the next array, previous array to the next array. So, the multiplier performance is the delay is M minus 1 plus N minus 2 into t_{carry}.

So, mainly just like the ripple carry adder, as if here this is the delay part, the first part of the delay is coming for the carry propagation. Now the next term, second term is for the N minus 1 into t_{sum}. So, here the summation or the sum circuits that is the some adder circuits that is also coming here, and third term is what N minus 1 t. And so, this is for the AND means the partial products because the partial products are nothing, but the AB term A dot B so, X 0 Y 1 like that.

So, here delay causes of 3 terms, one is for carry propagation, one is for that addition, another is for that partial product. Now, when we use the multiplier or we implement a multiplier using carry save adder structure, then delay becomes N minus 1 into t_{carry} plus N minus 1 t_{and} plus t_{merge}. So, if we again see the structure of the vector margin...

(Refer Slide Time: 09:06)

Indian Institute of Technology, Kharagpur

Performance of Multiplier

- MXN Array multiplier
 $Delay = [(M-1)+(N-2)]t_{carry} + (N-1) t_{sum} + (N-1) t_{and}$
- Multiplier Using Carry Save Adder Structure
 $Delay = \underbrace{(N-1)t_{carry}}_{\text{propagation}} + \underbrace{(N-1) t_{and}}_{\text{partial product}} + t_{merge}$

So, this is the first part, is again for the carry propagation, the first part is N minus 1 into t_{carry} is for the propagation, carry propagation. Second term is for the partial product and as if the t_{merge} is the array of half adder and full adders of the last row, that array of full address and as if this is a vector margin steps.

(Refer Slide Time: 10:20)

Indian Institute of Technology, Kharagpur

Wallace Tree Multiplier

6 5 4 3 2 1 0

6 5 4 3 2 1 0

①

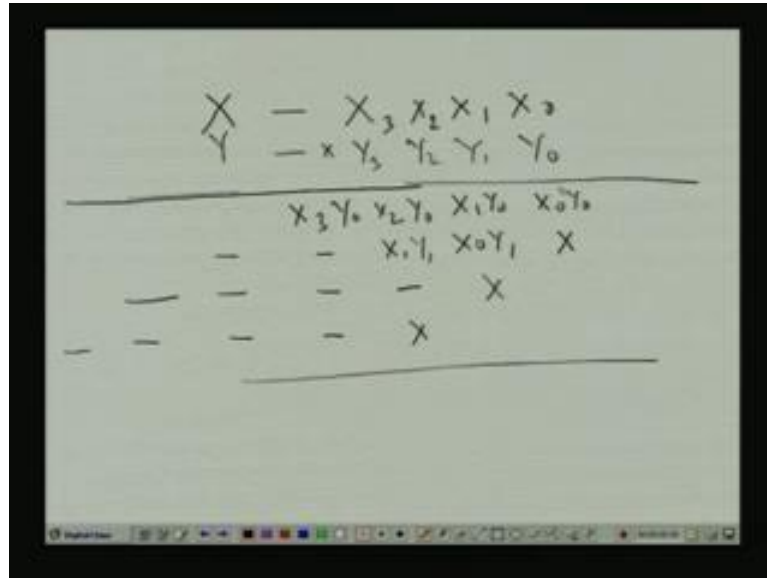
②

partial product

Now, we see a different type of multiplier structure and that is being used in recent times that is the state of the... or technique is called the Wallace tree multiplier. Now before I start explaining Wallace tree multiplier, first we see that, how actually what is the basic

principle behind this multiplier. See if these are the bit positions, say again this is 4 bit multiplier, now if I remember, if we remember that a 4 bit multiplier our (Refer Time: 11:08)

(Refer Slide Time: 11:11)



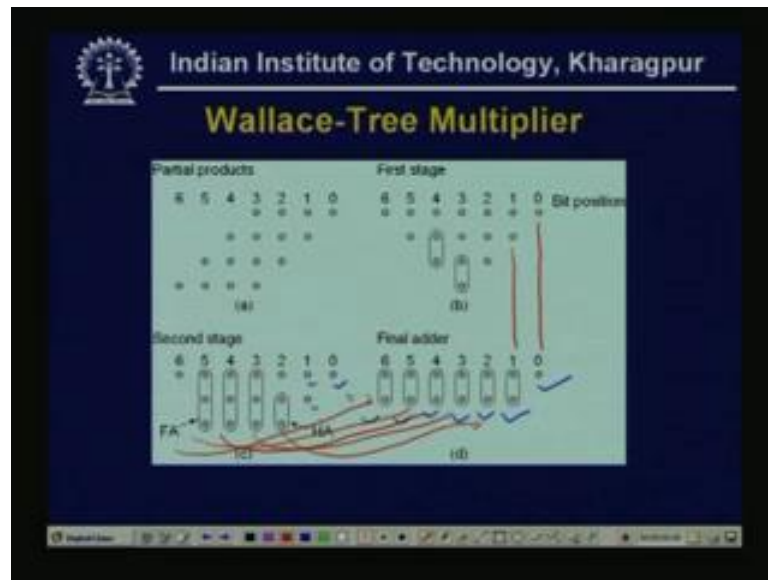
Say I have I have X is $X_3 X_2 X_1 X_0$, and Y is $Y_3 Y_2 Y_1 Y_0$, see if we multiply this two, I want to multiply normally, what we this is a term $X_0 Y_0$, then this is $X_1 Y_0$ then this is $X_2 Y_0$ then it is $X_3 Y_0$. Similarly, that then this will be $Y_1 X_0$, so $X_0 Y_1$ $X_1 Y_1$ in that way it will come, so I will getting a partial product for the each term and I have this type of structure. So, if I represent this term as a dot, so actually each row consists of 4 partial products.

So, see these dots or these bullets are nothing, but some of this bullets are some partial products, so if it is a 4 by 4 multiplier, so this type of structure will get. See this is 4 again shift, again the partial products, again shift, 4 partial products, again shift, 4 partial products now, we have to add this thing. Now the second step, so this is my first step first step, now in the second step, see that how we add or see here this is one the first partial product will be kept. This two will be added, so I need a adder for this then this three, now if I add two at a time. Then see this will be a, added and, so the this becomes 3.

So, here it is a, if I consider this one this is 1 then the 1th position this is 2, then this is 3 and this becomes... Now it is added and becomes 2, so this becomes 3 again another one

will come from here this becomes 3, this becomes 3 and this becomes 1. So, now if we again see then actually it will be a, it is nothing, but the first one is partial product and again some additions and needed for this. So, this is a tree type of structure that is, actually being used when the multiplier design is given using these basic concepts.

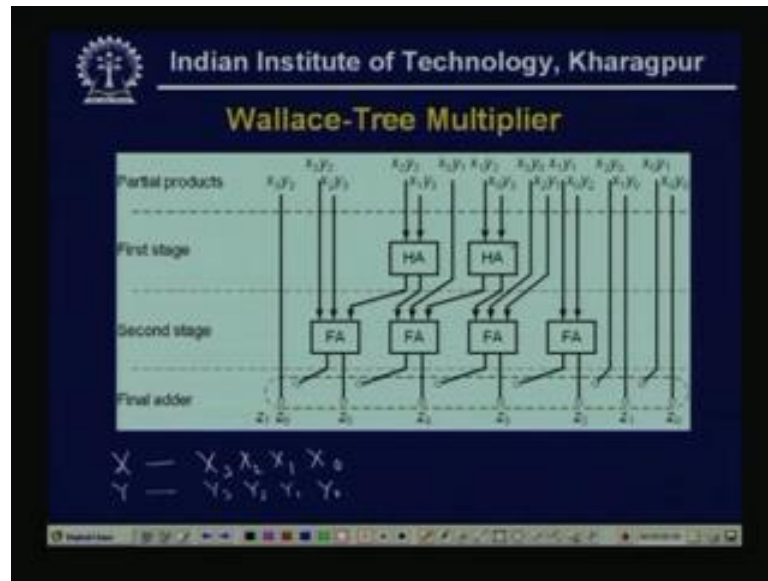
(Refer Slide Time: 15:15)



So, we see that thing again it is cleared here. These are the 4 partial products and the first stage these are the 0th to 6 bit positions and this is added, so this too added. Now in the second stage, see that this is as it is kept, this is as it is kept, one partial product. These two are, this is a half adder because 2, this full adder, this is full adder, this full adder. So, when in the fourth step or the final step, this is kept as it is, this two becomes is, this two already added becomes one sum.

So, these becomes 2, this two already added, this three is added giving one and one coming from here, so it becomes 2, this three in similar fashion becomes 2 and this is 6 position it is 2. So, in this way it will be added, this is the basic principle.

(Refer Slide Time: 16:37)

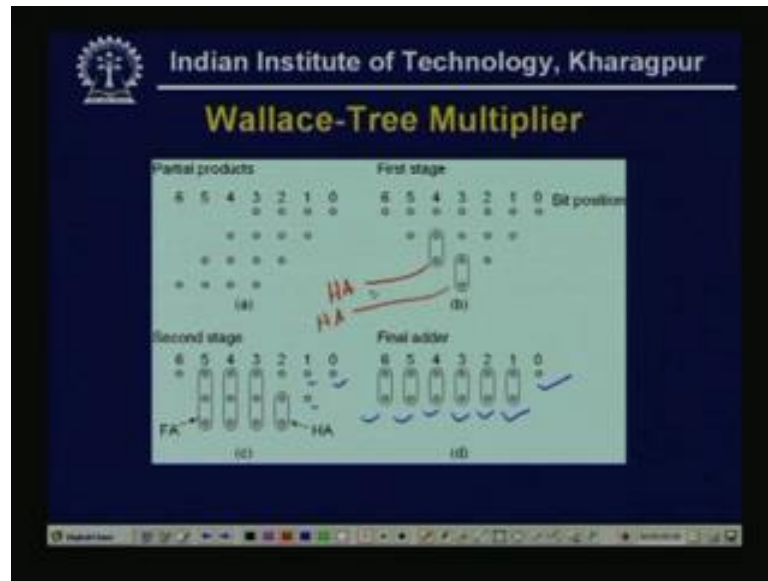


Now, if we see the step by step procedure, so for the Wallace tree, the these are the partial products available, so if it is $X_3 X_2 X_1 X_0$ is a 4 bit X is a 4 bit thing $X_3 X_2 X_1 X_0$, and Y is a $Y_3 Y_2 Y_1 Y_0$. Then, these are the partial products, generated when it is multiplied, now in the first stage as we have seen we need to half adders. So this partial products $X_0 Y_0$ is as it is coming to Z_0 , and these half adders are one is $X_0 Y_3$ and $X_3 Y_1 X_1 Y_2$.

So, if we see that previous picture, so bit positions is it is going here this two as if, so see if these bit position is as it is, it propagates this two we have to add and this is a half adder circuit in it goes here. Now, here it will be full adder and it will come from the second position after, the second stage, so it will go here. Similarly, the third step will go here, fourth step come from here, fifth also come here, sixth also it will be come from here. So, if from there where what we have seen, that $X_0 Y_0$ as it is come here.

So, if it is final adder circuitry and this is a partial product come here, similarly the $X_0 Y_1$ and this is the Z_1 these comes here. And the full adder circuit that 4 full adders, so this is the second stage a first we consider the first stage, first stage.

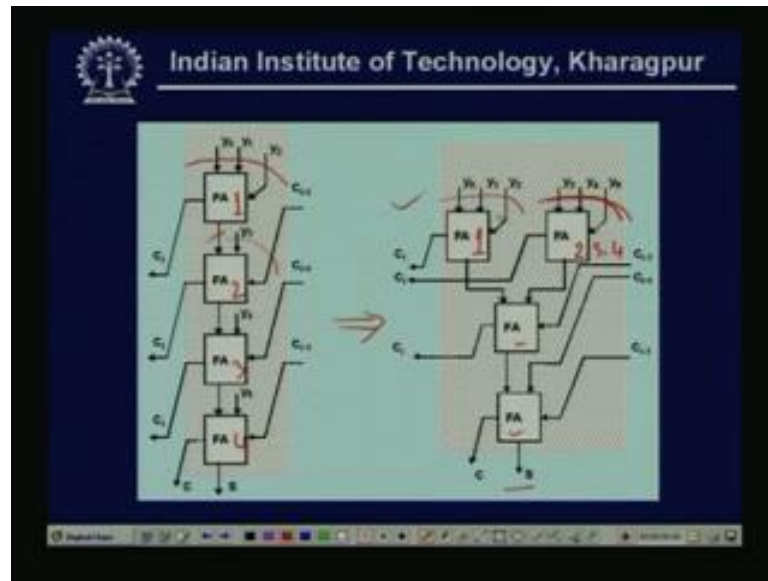
(Refer Slide Time: 19:56)



So, when the partial products are available, we need actually two half adders. We need here one half adder and this is another half adder in the third and fourth stage, so this is the two half adders in the third and fourth bit of the first stage. Now, if we consider the second stage, so second stage what we see that, one half adder and three full adders are needed. So, if you see that bit is three full adders, this full adder is needed, this is one half adder and one.

So, this is again a one full adder, so this is one again full adders are needed, so four, there are four full adders, here there are four full adders here, and the final adder that all these circuits, that all the as it these are added, as it is only the first partial product is coming and these are all added. So, as if this is the, final adder of the Wallace tree circuit, this is the final adder and from there that Z_0 to Z_7 , that product terms are coming.

(Refer Slide Time: 22:25)



So, now if we zoom the first stage, then these four stage, why it is tree structure? See that in the second stage we need four full adders, see the structure of the full adder or what are the inputs, what it is adding. So, see here it is coming $Y_0 Y_1 Y_2$ this three as if these are the inputs, these are the inputs of the full adder. Once carry is generated and this sum is propagated to the, next full adder, though this is the carry. Then, this is Y_3 similarly this two will be added and it is coming from the previous one.

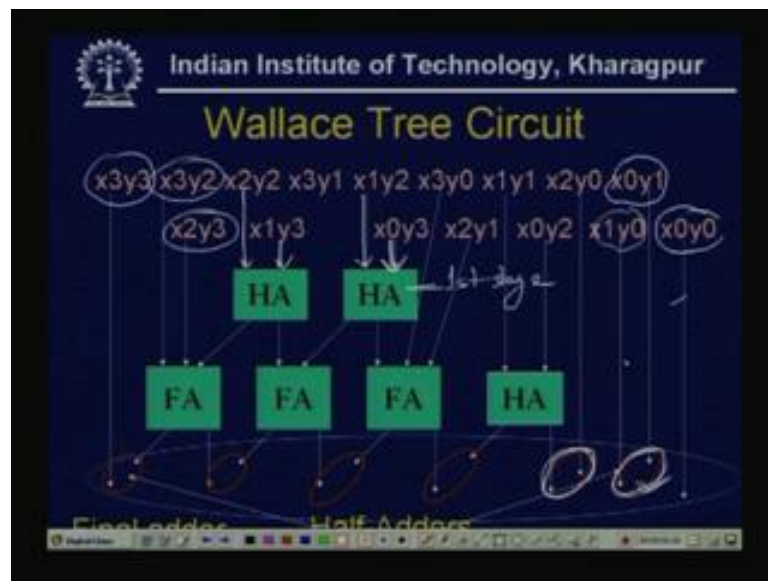
So, actually these are partial products again the similar way it is doing. So what we can do? We can now rearrange this, say this one full adder $Y_0 Y_1 Y_2$ is coming here, so first full adder, say if I now mark, this is as a 1 then this is my 1 here. Now, $Y_3 Y_4 Y_5$, so $Y_3 Y_4$ and Y_5 of the next one, these are the three, this three are the inputs of one full adder. And say the resultant here that after summing these two, this will be fed to the next full adder circuit and one coming from the previous stage the CI minus 1.

And, now this, so actually this becomes 2, this becomes 2 and this becomes the final one the CI minus 1, this is the, it is coming. So, this structure, this can be this can be treated as, as if $Y_0 Y_1 Y_2$ is fed to one full adder and $Y_3 Y_4 Y_5$ this can be fed to another full adder circuit and what, actually here sequentially it was adding. In this way if we draw the design of the full adders, then what we can redesign, that as if this one the number 1 is same, but as if 2 3 and 4 this full adder circuit.

We are redesigning as if $Y_3 Y_4 Y_5$ and they are adding into a full adder circuit. After these some circuit it is doing the addition. So, first we are doing, we are adding $Y_3 Y_4 Y_5$ I am taking the result and then this two are adding to by another full adder. So, these actually, this full adder is doing the one input of FA 2, FA 3 and FA 4. So, this is actually, the earlier circuit the 2 3 4 one inputs of 2 3 4, and then the result is another. It is taking new one say another FA circuit and this is another FA circuits, that will give the ultimate results.

So, here see that instead of doing the sequential thing as if here the partial products are fit parallel, so it will be a must much faster multiplier circuits because here we have to wait until the previous one comes. But here parallely we are feeding $Y_0 Y_1 Y_2 Y_3 Y_4 Y_5$ these are nothing, but the partial products available in the previous circuits. So, these are the thing available.

(Refer Slide Time: 28:07)



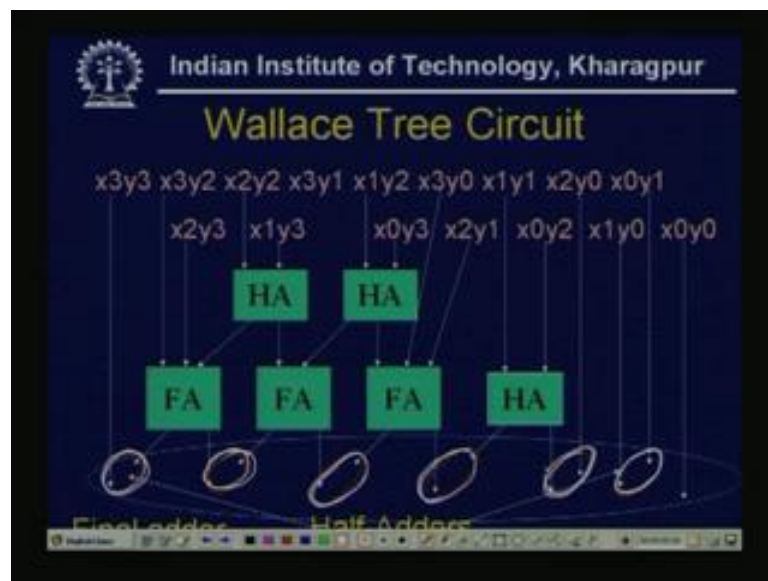
So, if we consider now the full tree say, these are the partial products, these are my these are my partial products say $X_3 Y_3 X_3 Y_2 X_2 Y_2$ like that. If we start from the LSB side it will be as it is one partial product will come here. And these are the two, the addition of as it addition of 2 bits where 1 bit is actually one partial product. So, if I treat these as a P 0 P 1, then actually this is the sum of this thing, these ring is that sum of this thing, this ring.

Similarly, the second one, the another two partial products and one coming from the another half adder and directly one partial product these will gives another addition. Similarly, it is coming, so in the first stage already we have seen that two half adders are needed. So, these are the two half adders which adding one is adding the partial product of $X_2 Y_2$ and $X_1 Y_3$ another is adding $X_1 Y_2$ and $X_0 Y_3$. One thing to be noticed, that here when the suffix if I take the symmetrical thing; that means, X is $X_3 X_2 X_1 X_0$ Y is $Y_3 Y_2 Y_1 Y_0$.

Then, we will get the product term has $X_2 Y_2$ and say X subscript is reduced by 1 whereas Y subscript is increased by 1, say $X_1 Y_3$, this two are added this two are as if the two input, two product terms inputs as product terms to one of the half adder. Similarly, the other half adder, see this is $X_1 Y_2$, so when the X subscript is reduced by 1 it becomes $X_0 Y_3$, so it is $X_0 Y_3$ and then these two are added, so in that, this is my first stage.

Now, in the second stage we need three full adder, one half adder what sometimes we write the four adders, four full adders itself because one half adder always we can write as full adder. So, this can be a three full adder one half adder and then the outputs are taken care.

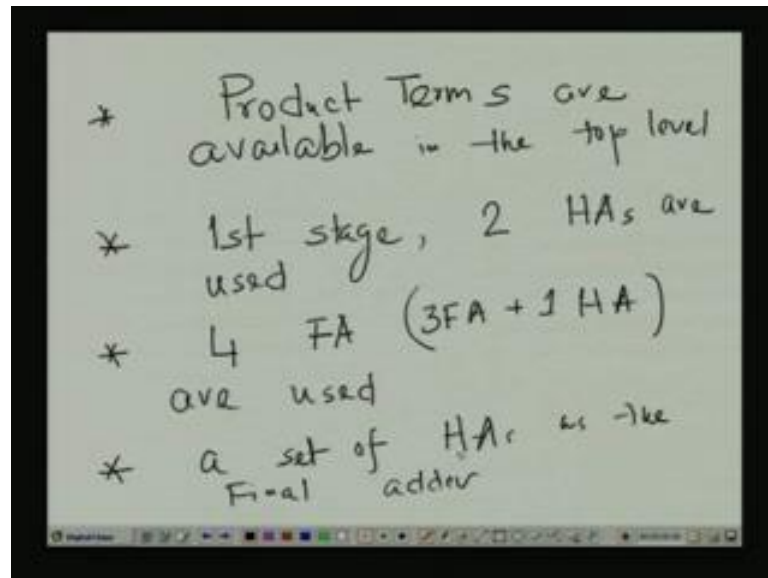
(Refer Slide Time: 31:07)



So, this is as if my final adder circuitry, this is my final adder and these are nothing but the a this rings of the half adders. So, as if all these rings, they are summing two inputs

as if two operand it is working, so these are the 3, 4, 5 and 6 as if these are the half adder. So, always these are functioning on two inputs and this is my final stage, so or this is my, if it is like tree structure and this is the root, so this is called the now Wallace tree circuit. So, the basic principle is again, if we summarize that basic principle is that.... As if product terms...

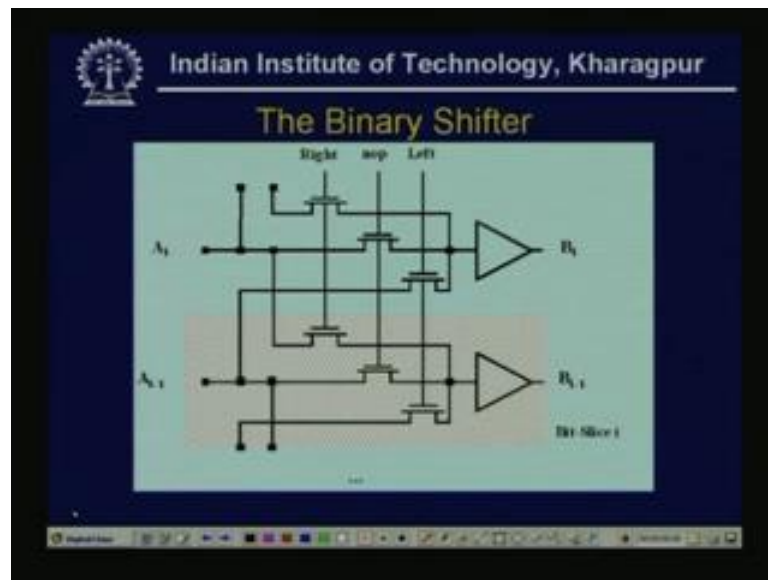
(Refer Slide Time: 32:17)



All product terms are available in the in one level, in the top level, am I right? This is one thing, then in the first stage, first stage two half adders are used, in the second stage we can tell four full adders or three full adders plus one half adder are used and then in the last phase, the set of half adders as the final adders. So, in this way we can design the circuit. So, this is a now there are many other multiplier circuit, we have just discussed 3 type of multiplier circuit.

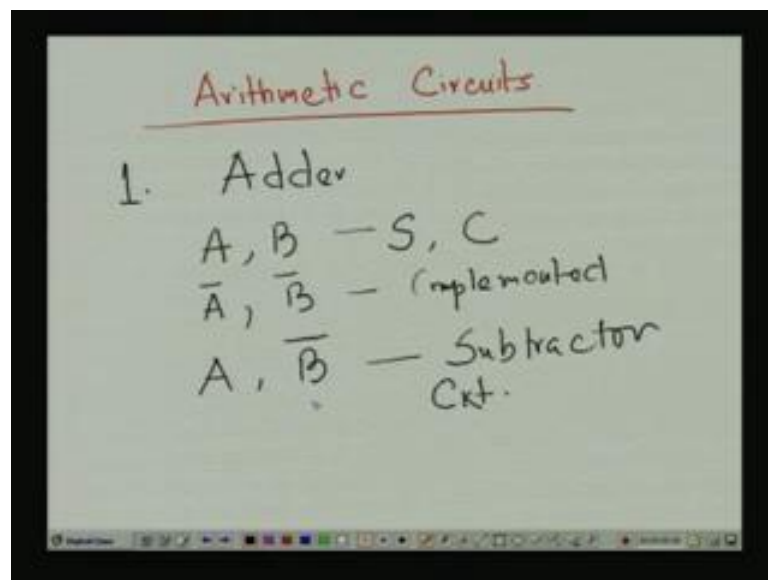
One, our basic AND and shift, one is our array multiplier, another is our carry save multiplier using carry save adders structure, and other is that Wallace tree circuit or sometimes called that using Wallace tree array that is the Wallace tree multiplier. Now, see there are several other multiplier circuits are available, but the basic principle are same, always we try to reduce the digital elements or like the half adders, full adders etc.

(Refer Slide Time: 34:36)



Now, another type of or there are many other, arithmetic circuits are also used in different type of digital designs.

(Refer Slide Time: 34:58)

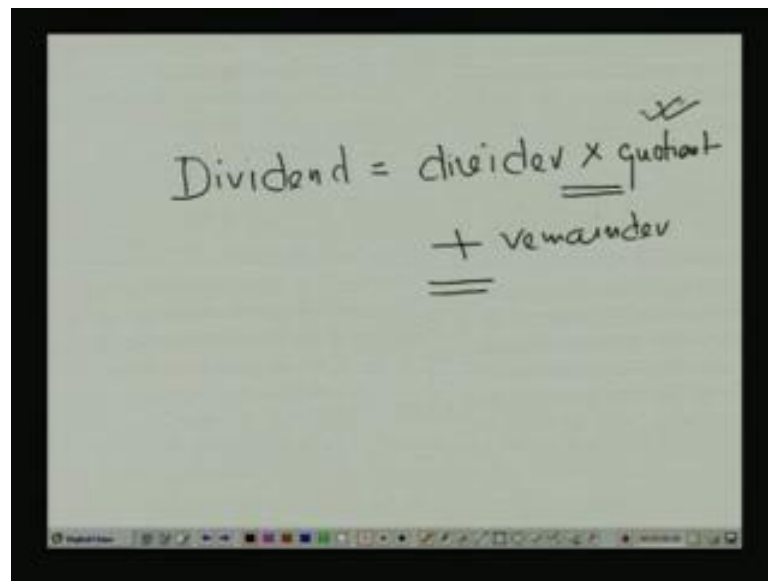


So, far we have discussed, so we are discussing the arithmetic circuits, and the first one is the adder. And all of you know, that if the input variable is complemented, that means, if we use A B as the two operands and it gives some S and carry C and obviously, we can give the A bar B bar and it will do the complement of the sum complement result is

complemented. Now, if one operand kept as it is and one is complemented then obviously, we will get a subtractor circuit.

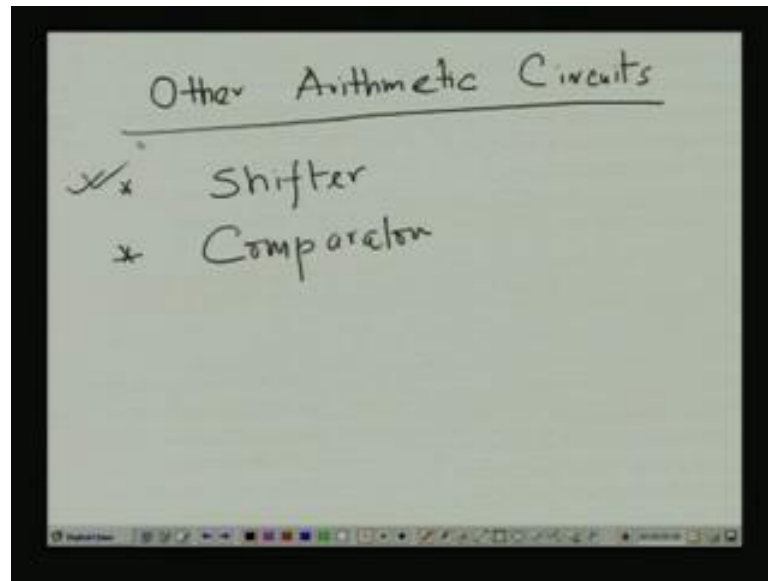
So, actually the adder circuit, we inverted or complemented input that can be used as a subtractor, so adder subtractor, normally we use the same type circuit. Now, another is the multiplier, multiplier we have used and if we can do that thing for division of circuit also because on division always we can, write like that.

(Refer Slide Time: 36:56)


$$\text{Dividend} = \text{divisor} \times \text{quotient} + \text{remainder}$$

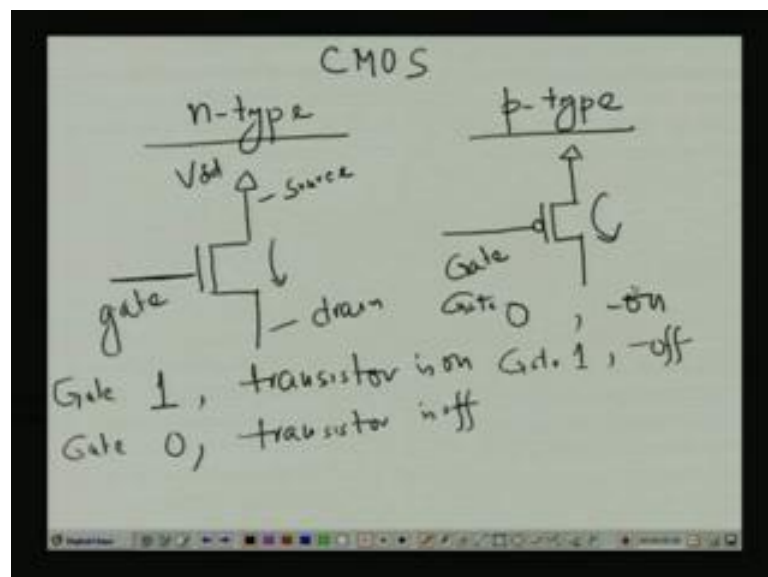
Our dividend is divided into the quotient plus remainder. So if I know the multiplier and the adder circuit, so we can divide that thing also, that means, we can always we can get the quotient from here. Now, there are many other arithmetic circuit one is the comparator.

(Refer Slide Time: 37:55)



The other arithmetic circuits, so this is, one is shifter another is comparator, so first we see the shifter circuits, see this is binary shifter and in this circuit actually we have used some pass transistors, pass transistors circuits. So, before we start explaining shifter, so briefly I am telling the function of transistor this type of pass transistor circuit. See normally there are two type of transistor we know.

(Refer Slide Time: 39:16)

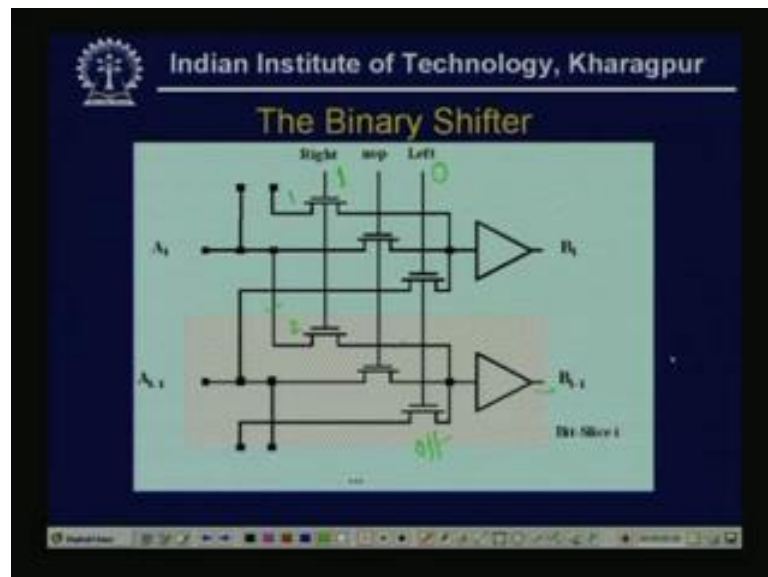


This is n type and a p type transistor, these we have calling CMOS transistors these are CMOS. Now, n type, so see this is there is one source, drain and this one gate, now if

some V_{del} this is attached with power supply. So, in the gate if we apply A 1 in the gate if it is 1, then the transistor on; that means, it will passing the current from source to drain. If it is 0 that means, if the gate, if the gate is and the gate will the 0, then actually it is off, transistor is.

Now, similar thing for the p type, normally the notation is like that and here it is totally reverse, that means, if the gate is 0, if it 0 is apply to the gate then it will be on, and if gate is 1 then it is off. So, using this concept we can normally... this n transistor are called the pass transistors because if it is 1 it will pass the current, so this is the concept.

(Refer Slide Time: 41:36)

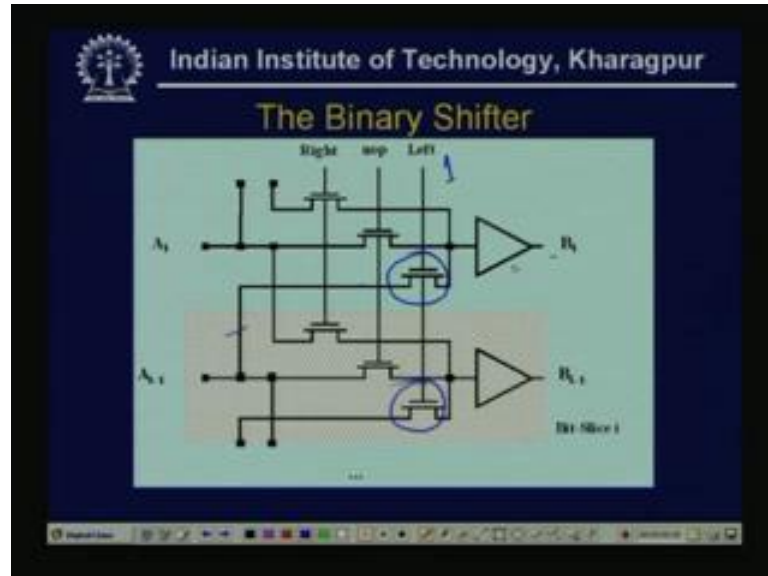


Now, if we take a 2 bit shifter say this is A_i and A_i minus one, now we see that what type of structure it is, see the shifter can be either a right shift or left shift. So, first we see the right shift, so this A_i this A_i , so and this is A_i minus 1. That means, this is LSB this is MSB. So, if it is right shift that means, say for this 2 transistor, see these are the this is transistor 1 and 2 the at the gate A right a selected means 1. So, it will pass the value, now see the this is, this value A_i and this pass to as a A_i is passed as a B_i minus 1, as it is right shift.

Now, this is A_i minus 1, A_i minus 1 and that is going here and as the left is zero, so it will not pass here. Similarly, see when the left is 0, so this is also off and this will not pass from here, so it will not pass from here. So, it will go here that means, A_i is B_i minus one, so this is a right shift because the MSB becomes the 1 lessor bit, bit positions;

that means, F_i , F_{i+1} equal to 5 F_i becomes B_4 . So, this is right shift, now we see the left shift, so if it is a left shift, now in this situation say....

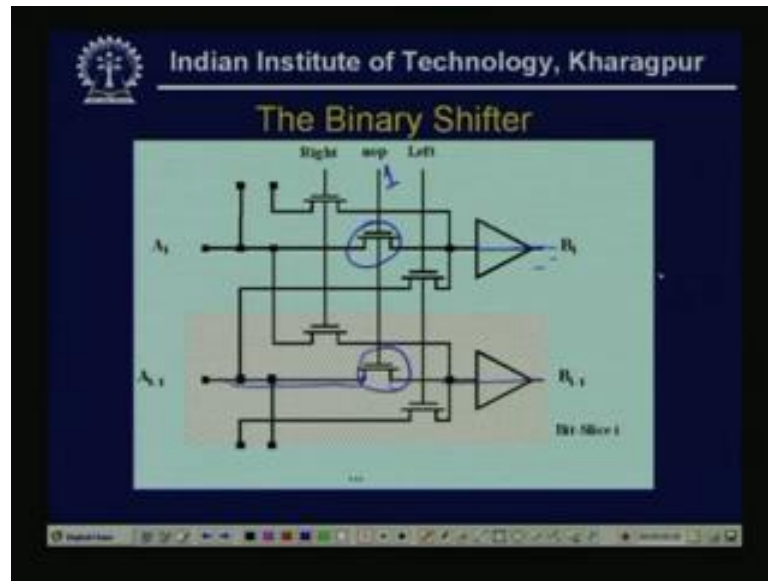
(Refer Slide Time: 44:34)



Say, if it is, now left shift, so the name this left bit becomes 1. So, this left bit becomes 1 now see, if this line is one that means, this transistor is on as well as this transistor is on. Now, if this transistor is on then actually A_i minus 1 means the lowest bit or the LSB side that is being passed as the B. That means, A_i minus 1 goes to B_i , so this is a left shift, and when this is on see here, this should this bit actually the A_i minus 2 this can be A_i minus 2, it goes to B_i minus 1 like that.

So, that means, A_i minus 1 goes to B_i , A_i minus 2 goes to B_i minus 1. So, this is a if, I is 5 again then is actually A_4 going to B_5 . That means, this is left shift, so using this simple circuitry that and taking that which or selecting the control lines, that actually whether we want a right shift or left shift or no operation. Later we will see this situation, so it can be accordingly shifted right or left. Now, if it is no operation say again for no operation that another 2 transistors are kept. So, if it is no operation then actually all things will go.

(Refer Slide Time: 46:44)



And here move up is this line is B_1 . Obviously, this 2 transistors are on, and see as it is A_1 will pass as B_1 and A_0 will pass as B_0 . So, this is the no operation, that means no shift is needed. So, this is for a shifter of bits lies 1, this is a shifter. Now, will consider a more complex shifter.

(Refer Slide Time: 47:19)



So, this is normally called a barrel shifter, say actually here it is multiple bits are shifted together or what we can tell, as if it is a, the multi bit structure and together it is shifting. So, earlier example what we have seen, that only 1 bit is shifted; that means, if it is left

the LSB shifted to MSB, if it is right shift then MSB is shifted to LSB. Now, for the barrel shifter say I have a, I want to give a shift or a 4 bit value, one 4 bit value I want to give a shift. Say I have a n bit bus and I want that the whole n bit bus should be shifted by some of the bits, then we will see that how the structure will be.

Sometimes these shifter have called the VLI shifter because this 4 can be 4 bit or 8 bit shifter or 16 bit shifter and it is very regulars structure will be seen that thing, See in first thing we see the design architecture. So, I want the bits 4 bits to be shifted or A₃ A₂ A₁ A₀ this 4 bits, now, if I want a shift 1, then say as actually it has the, it has 2 D type of structure. And here there are 4 input bits that to be shifted and there are 4 options than how many shifts we want. It is shift 0 this is shift 1 this is shift 2 and this is shift 3.

And see the structure it is being there, so this is say if we start seeing from here, say this is shift 0. So, this transistor is for shift 0 again diagonally this is for shift 0 again it is connected this is shift 0 and this is shift 0. Now, if it is shift 1, I give a different colour then this is a shift, these diagonal is for is for shift 1. Similarly, this circuit is for shift 2 and this circuit is for it is for shift 3.

Similarly, if we see actually here also, so it is diagonally symmetrical or across the diagonal it is symmetrical. So, if we consider this shifter as if this is a 2 D arrays of pass transistors then each diagonal represents a shift, a particular shift. Say here the main diagonal is the shift 0 and the off diagonal or that means, the next diagonal is shift 1 represents, then another diagonal is it is shift 2 and the last one is a shift 3.

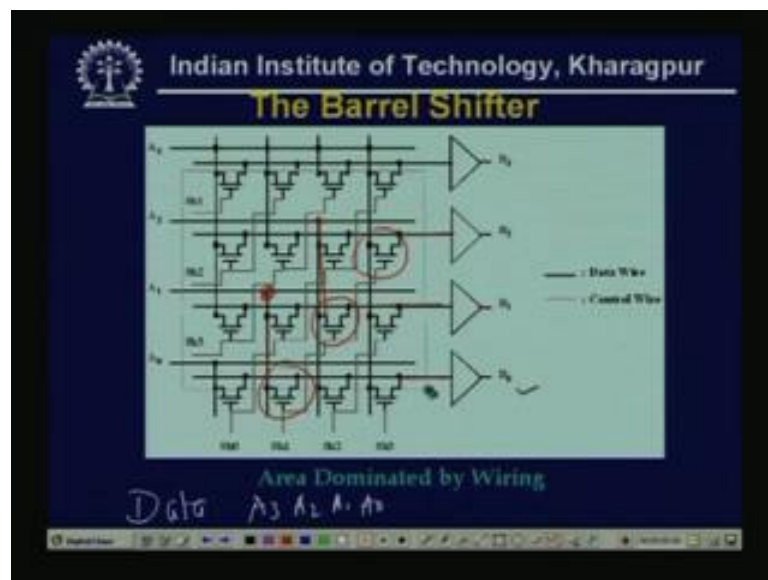
So, similarly here also we can do that thing, now, say I have 4 bits this is my data bits to be shifted, so these are my A₃ the my data bits are my data bits are, data bits are A₃ A₂ A₁ A₀, now see how the shift can be there. So, see it that this is my A₀ line the LSB, see this is my LSB line and when the shift 0 is selected. So, this see this is my A₀, so this is be this is nothing, but a pass transistor just now we have mentioned, so this is being passed.

So, this A₀ value this is this A₀ value is passed and this is going to the because, it is shift 0, say at a time only in the last row, only these value will be selected, so this is going to the B₀, so A₀ goes to B₀. Similarly, say if it A₁ and see that actually for A₁ this value will be this is the shift 0, because this diagonal represents the shift 0. So, A₁

will be as it is, it will be B 1 say as it is a shift 0 means no shift. So, A 3 A 2 A 1 A 0 will move from, as it is will as it is will be shifted to output.

Now, we see the say we see the shift one, so if it is shift 1, say the next diagonal this is. Now say A 0, A 0 has come here say 0. So, shift 1 is selected, so, now this will be passed to here, it will be see, this is shift 1 and this value is this values is A 1. So, this is A 1 say this is A 1, here it is A 1 is attached.

(Refer Slide Time: 54:32)

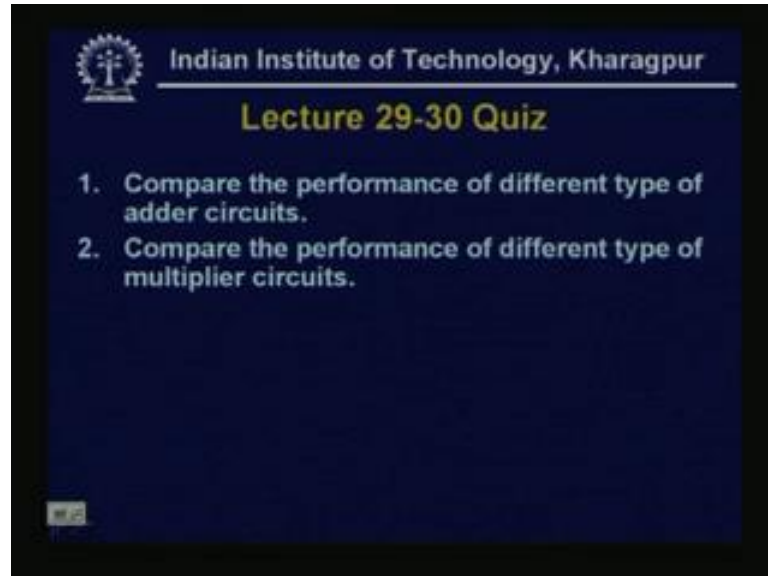


So, now it will be clear, so if I if I select the shift 1 that means, this is my this is my A 1 value this is my A 1 value, this is my A 1 value this comes here and it will be shifted this will go to B 0. Similarly, A 2 say shift 1, so for when A 2 this will be this is my A 2 and see this is my shift 1, so this transistor is selected this transistor is selected, it will go to A 1. Similarly, if it is A 3 see that this A 3 is A 1 will be going to be 0, A 2 will give going to be 3, A 3 will go a go to B 2 this is my shift B 2.

And what about the A 0, where the A 0 will go, as it is shift 1 see this shift 1, so this is my shift 1 and it will go. So, actually it will be 1 bit shifted A 3 will go to B 2, A 2 will go to B 1, A 1 will go to B 0 so, this is the... if when the shift 1 is selected. Now, if we consider the, shift 2, if we consider the shift 2 then will be seeing that accordingly the 2 bits will be shifted. That means, A 3 will be go to A 1, so these two will be my shift 2, in this way the A number of bits or the in this example it is 4 bit inputs will be shifted

accordingly. So, these are the different arithmetic circuits we have seen, already we have discussed the comparator circuits.

(Refer Slide Time: 0:57:01)

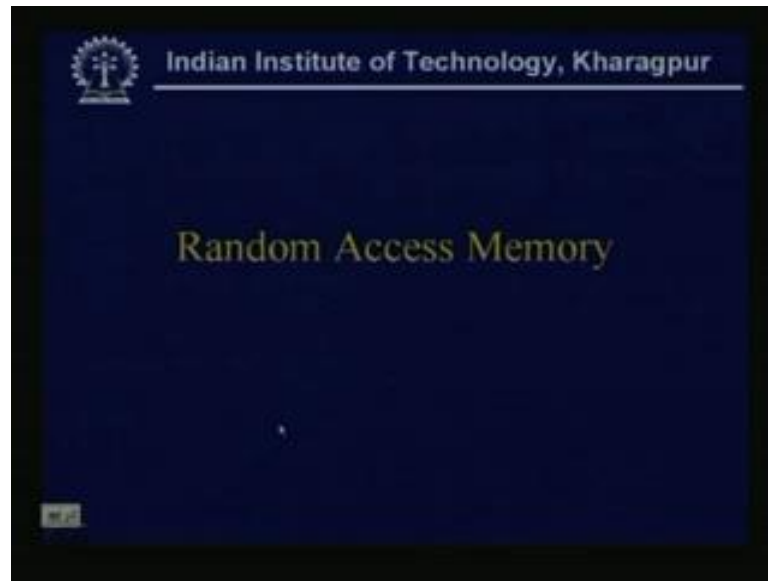


So, we can and here we end our lecture on arithmetic circuits, and these are the quizzes for the lecture twenty nine and thirty.

Thank you.

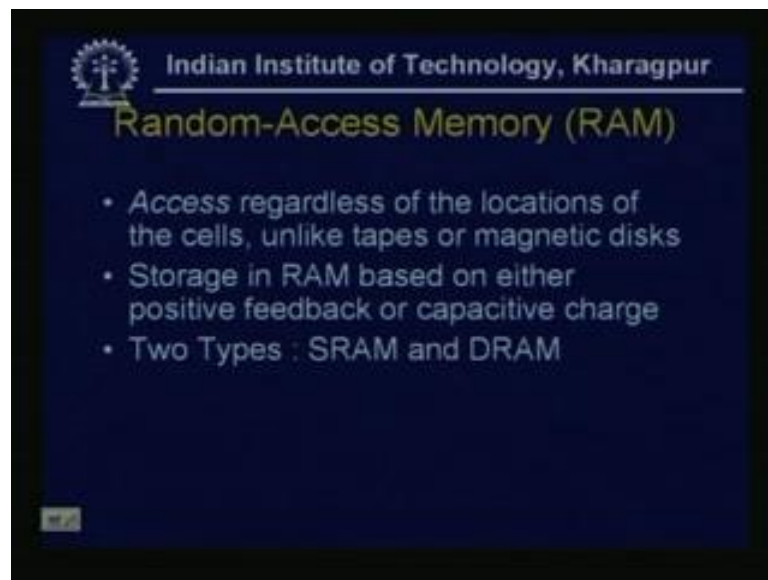
In the last few lectures we have seen the design of arithmetic circuits, mainly there we have studied the design of adder, the different type of adders the multipliers, the first multipliers etc. Now, today will see another very important design structure that is called the memory. And that is one of the important circuits that we need for any digital systems, today will read the design of memory circuits

(Refer Slide Time: 58:08)



Now, as already we have read the, read only memory, now today will see that random access memory or the RAM. So, first we define what to do you mean by the random access memory or the RAM.

(Refer Slide Time: 58:27)



Normally it access regardless of the locations of the cells and unlike tapes or magnetic disks. Now storage in RAM based on either, positive feedback or capacitive charge, capacitive charge. But before this the different types of memories, we see that, that what is the normal structure of that a memory.