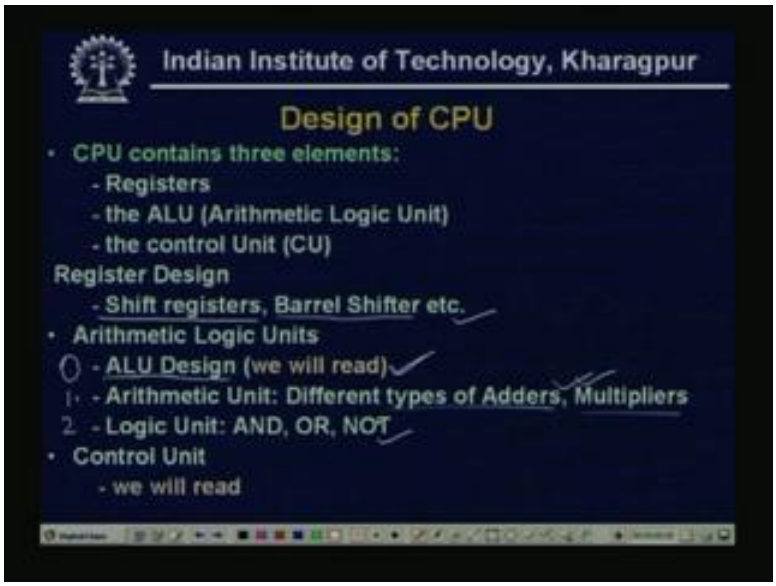
Digital Systems Design
Prof. D.Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
Design of Computer Instruction Set and the CPU (Contd.)

We are reading the Design of Computer Instruction Set and the CPU. In the last class, we have seen that how the computer instructions set can be design and the reduced instruction set machines, then the current state of the art actually the all the machines now a days are reduced instruction set.

So, how the instructions set are are selected? How it can be designed? How is the hardware implementation of it, that already we have read.

(Refer Slide Time: 01:33)



Today, we will start discussion on the design of CPU and first we will see the design of ALU in this class. So, CPU contains three elements, the resisters, the ALU the Arithmetic Logic Unit and the Control Unit or CU. Now already, when we have discussed we read the different type of register designs, there we have seen the how shift registers can be designed using flip flops. The barrel shifter or what we can tell that, this is actually a more than one bit transfer that called the barrel shifter. The 4 by 4 or 8 by 8 barrel shifter 16 by 16 how they can be design already we have read that part.

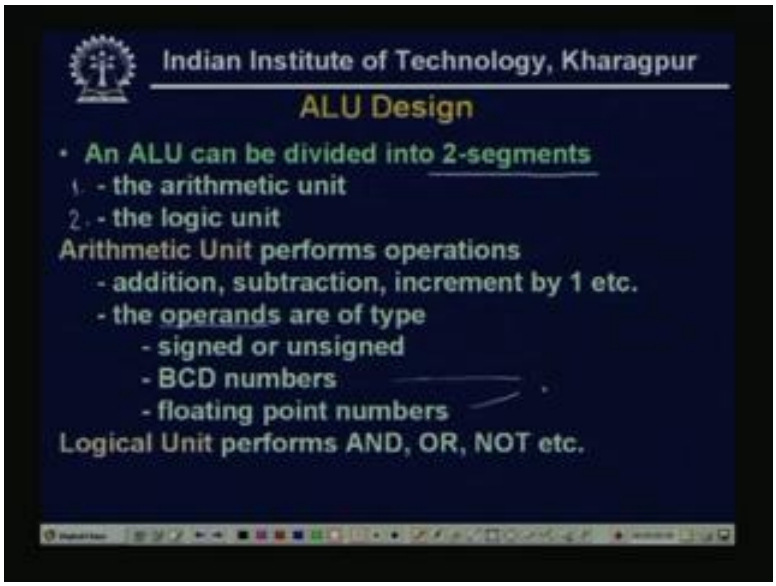
Now, the arithmetic logic units as it name implies the arithmetic units, so it has actually two parts, one is the ALU design the arithmetic unit, and arithmetic unit and the logic unit. Now, the ALU design means the combination of two the arithmetic unit and the logic unit, how they can be combined to form a ALU, whether it is a low scale, whether it is a upper scale ALU or larger ALU can be design that we will read in this class.

Now, the arithmetic unit means the different types of adder's, multipliers, sub tractors actually all type of arithmetic operation the unit does. And earlier classes, we have read different types of adders, the ripple carry adder normal carry look ahead adder carry save adder the one adder. Similarly, for multipliers we have seen the normal multipliers the booth's multiplier, the first multiplier, the array multiplier different type of multipliers we have already read. So, actually arithmetic unit, how they it can be design, already we have read.

Now, the logic units are nothing,, but our logic gates, it can be a simply one gate AND OR NOT NAND or the combination of AND OR NOT gates or in one word we can tell we have read the Boolean equations or how the logic outputs or the outputs can be generated from the combination of logic gates. So, already we have read this part also.

So, basically the different units of ALU means the arithmetic unit the logic unit, how they can be design? Already we have read, today we will read that how the using these different units, the arithmetic and logic units, how the ALU can be designed? And then, the other part of the CPU is the control unit that also we will read later.

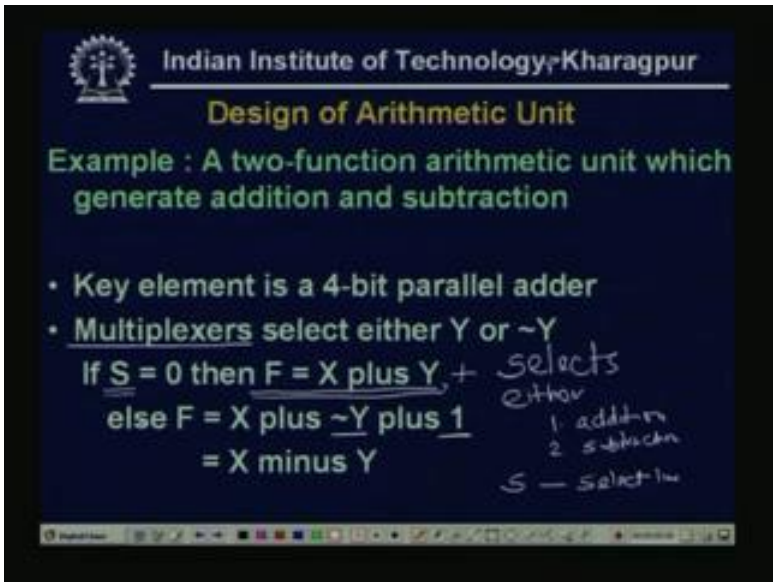
(Refer Slide Time: 05:36)



So, we start the ALU design, an ALU can be divided into two segments, two segments just now we have seen the arithmetic unit and the logic unit. Now, arithmetic unit performs operations addition, subtraction, multiplication, increment by one decrement by one etcetera, the operations. Obviously, it operates on some operands and these operands can be of type signed or unsigned. Say, binary coded decimal number or BCD numbers we read early floating point numbers. So, the different arithmetic units can be designed based on the type of operands.

So, some floating point unit or some if operands are BCD numbers, so simple signed or unsigned binary numbers. And logical unit performs AND OR NOT all these logical operations.

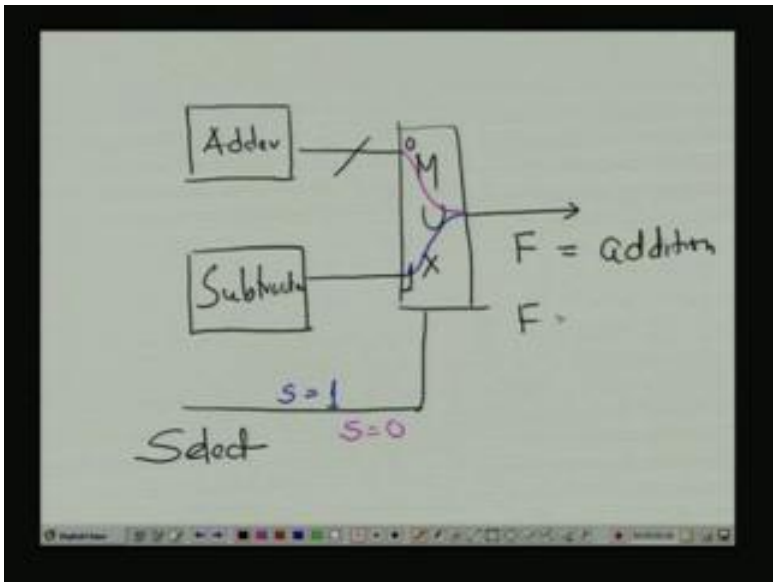
(Refer Slide Time: 07:10)



Now, first we see the design of arithmetic unit, we know the all the units or the different type of adders multipliers, sub tractors that all of we already we read. Now, I am considering a two function arithmetic unit, which generates addition and subtraction. See here the key element is a four bit parallel adder and as the module does the addition or subtraction, so for that we need one multiplexer. So, it selects either selects either addition or subtraction, so one multiplexer is needed for that.

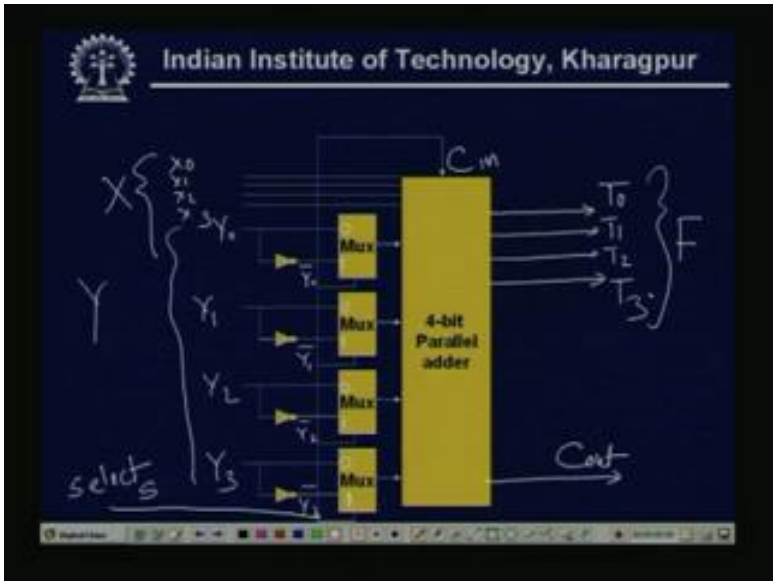
Now, say that S is the select line of the multiplier, S is the select line. So, my design is such that if S equal to 0, then the it the output function is F equal to X plus Y; that means, it is a addition else F equal to X plus Y bar comp Y complement plus 1; that means this is the 2's complement subtraction; that means X minus Y. So, if it is if S equal to 1, if S equal to else means if S equal to 1 select is 1, then, it is a minus 1. So, this is a simple logic.

(Refer Slide Time: 09:29)



Say I have a adder already we have seen, how the adder can be design, say I have a sub tractor. Now, this is a multiplexer, it has some select line say adder goes to zero line sub tractor goes to one line, so if S equal to 0, it will the adder is passed as a output, so this is a F is a addition. If S equal to 1, then this value will be passed and then F is a subtraction, so this is a one simple design which does the either addition or subtraction together. So, we are calling this is a two function arithmetic unit.

(Refer Slide Time: 11:29)

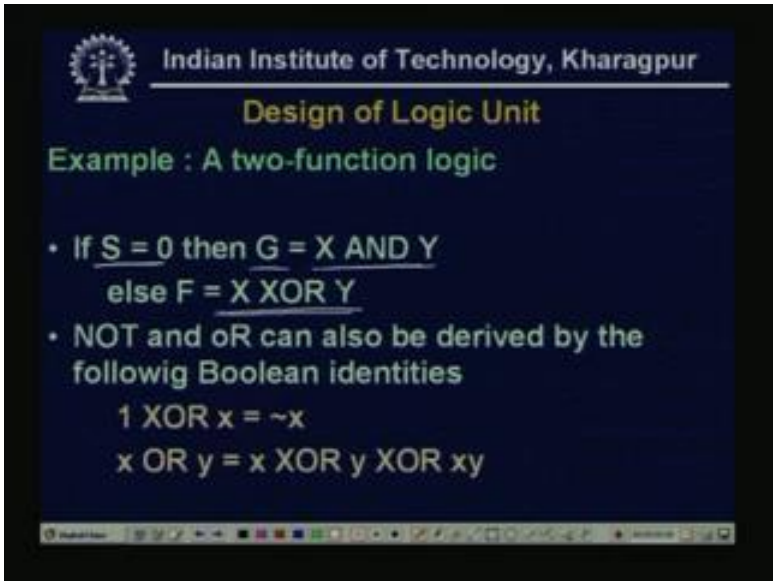


Now, if we see the design, see this is a four bit parallel adder, say it has 4 bit, so these are X 0 these are my X. This is X 0, X 1, X 2, X 3 and these are my Y lines, so this is Y 0, Y 1 Y 2, Y 3. Now notice here, we have given four multiplexers here and the inputs are either Y 0 or Y 0 complement. Similarly, either Y 1, Y 1 complements Y 2, Y 2 complement Y 3 Y 3 complement. Now, these select lines, say select line S, now if S is 0, then Y 0, Y 1, Y 2, Y 3 uncomplemented values of Y's are selected.

And, as if these are generating the output the four output bits the T 0, T 1, T 2 and T 3. So, these are my function F, F is a addition. Now, if S equal to 1, then thus if select line is 1, then actually these all the complemented values of Y, Y 0 bar Y 1 bar Y 2 bar and Y 3 bar will be selected. And then, the function will be that X 0 X 1 X 2 X 3 are it would be added with Y 0 bar Y 1 bar Y 2 bar Y 3 bar, so actually this is a subtraction.

So, what we can tell that, this is my C and this is a C out, so using four multiplexers with one select line. That means, 2 to 1 MUX, 4 to 1 MUX, that 1 4 bit parallel adder as well as a 1 sub tractor 4 bit sub tractor can be designed in this way.

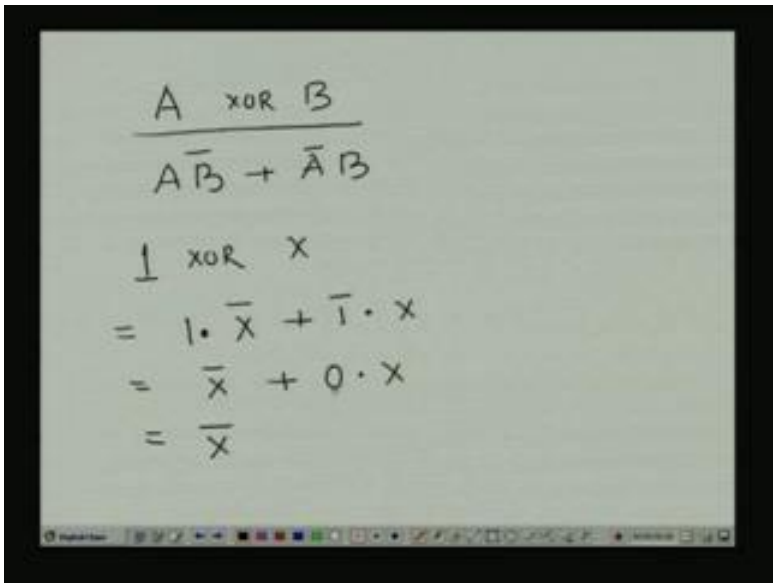
(Refer Slide Time: 16:05)



Now, we see two function logic, earlier we have seen two function arithmetic unit. Now, we see a two function logic logical unit, see in the example last example if S equal to 0, then it was a AND, if S equal to 1, then it was a sub tractor. Now, we are checking we are designing AND logic that if S equal to 0. That means, select line of the multiplexer is 0, then the function G is X and Y else the function is X XOR Y either it is doing AND or

which is doing a XOR. NOT and OR can also be derived by the following Boolean identities means using these XOR and AND, because if it is 1 XOR x, we know the XOR function

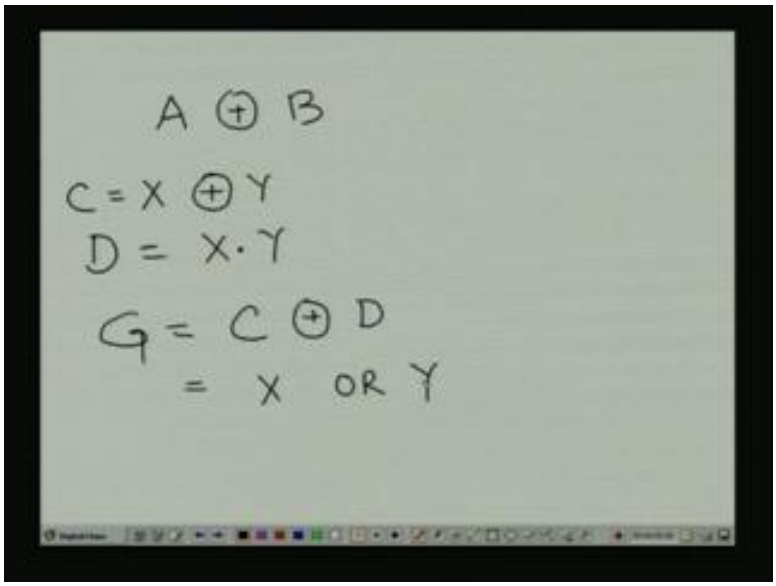
(Refer Slide Time: 17:24)



Say if it is A XOR B, we know the truth table of this thing is actually function is AB bar plus A bar B. So, 1 XOR X means 1 and X bar plus 1 bar dot X. Now this is nothing but X bar plus 1 bar is 0, so this portion will this is ended with 0 means nothing will be there. So, this is a X bar

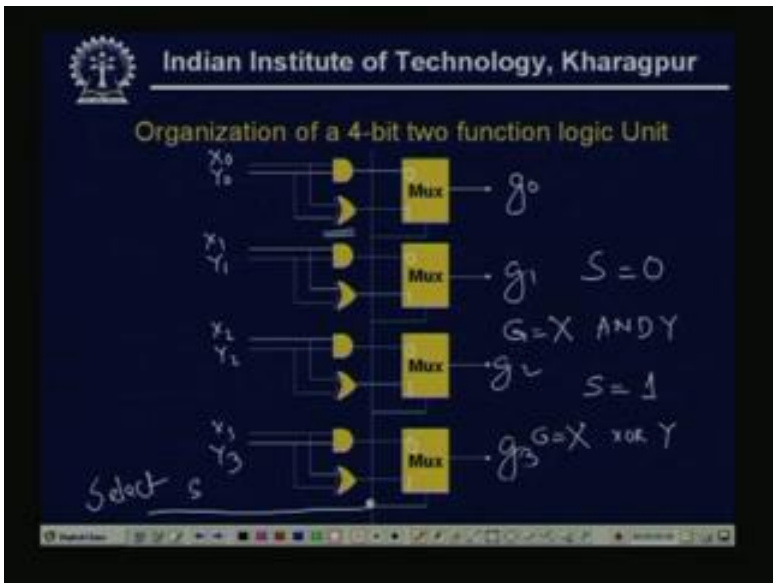
So, that is written here, (Refer Slide Time: 16:05) that 1 XOR x is nothing but a complemented value that is a NOT, so NOT can be designed using XOR. Similarly, X OR Y is nothing but X XOR Y XOR XY, so again using the XOR logic another level of XOR.

(Refer Slide Time: 19:06)



So, again what we can tell that if I have I know or I can design a say this is A XOR B. Then, we can do that X XOR Y and then it can be say it is a C equal X XOR Y and D equal to X AND Y. Then, G equal to C XOR D will give the C OR XOR Y that will be a X OR Y, (Refer Slide Time: 16:05) so that X OR Y is X XOR Y XOR xy. So, actually X XOR Y that we have written as a we have written as a C and then C XOR and the XY is written as a D, then this is C XOR Y is nothing but x OR y. So, actually if we can realize AND and XOR, then using these two logic NOT and OR can also be designed.

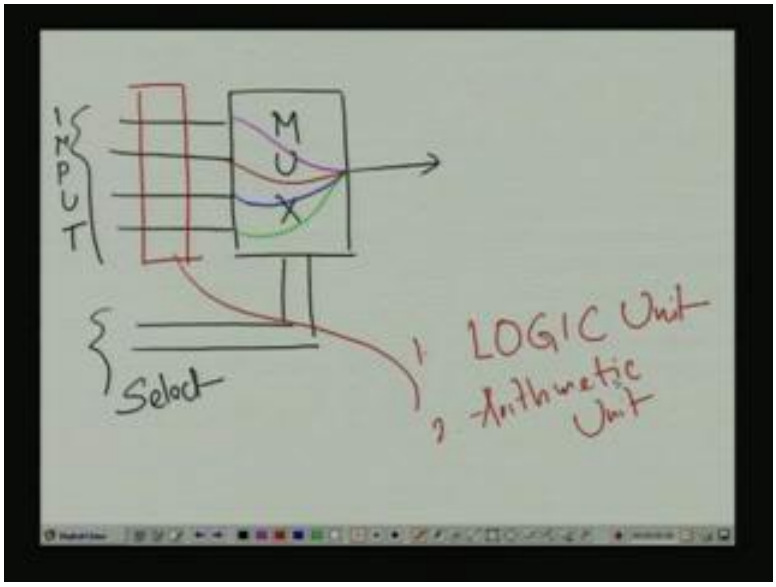
(Refer Slide Time: 20:52)



So, this 4 bit two function logic unit that we can give that if it again this is a 4 bit, so outputs are X 0 Y 0, then it is the AND is fed to the zero input. The multiplexer and the XOR goes to 1, X 1 Y 1, X 2 Y 2, X 3 Y 3, the 4 bit variable. Now, it has one similarly the MUX has that select lines S, so if S equal to 0, the 0 bit of the zero input of the MUX will be selected and it will be X 0 Y 0 X 1 Y 1 X 2 Y 2 X 3 Y 3. So, actually this is X and Y, say G equal to X AND Y, so we are getting g 0 g 1 g 2 g 3.

Now, if it is if S equal to 1, then the XOR will be selected output of XOR will be selected; that means, it will be X XOR Y. Then, g will be X XOR Y, so again it is a two function logic units. In this way using the multiplexers, we can always design a two function logical unit. So, we know that it is the multiplexer is a 2 to 1 multiplexer, the select line is 1, then it is 2. Similarly, we can the already we have read, how the bigger multiplexer can be design.

(Refer Slide Time: 23:38)

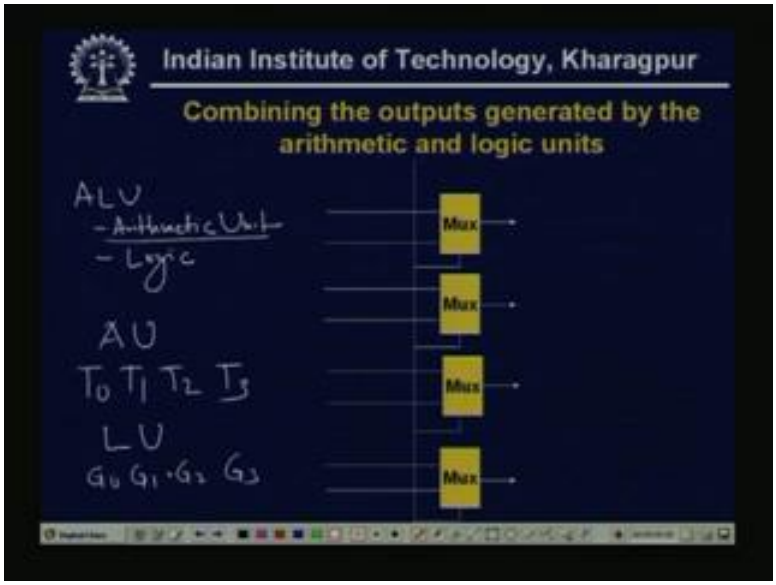


And, if we have a four function Logic units or four function arithmetic units we need, then actually the way. What we can do, we can take a multiplexers which have four input lines two select lines and the accordingly that particular output can be selected. These are my input lines and here these, what we can do that here, this can be this red box, this can be a LOGIC unit or this can be an arithmetic unit.

Now, as already we have read, that instead of larger MUX, what we can do, we can design a larger MUX multiplexers using smaller number of small MUX multiplexers or

or in more number of small multiplexers. So, in that way also we can do this or we can design this logic unit and arithmetic units using the multiplexers.

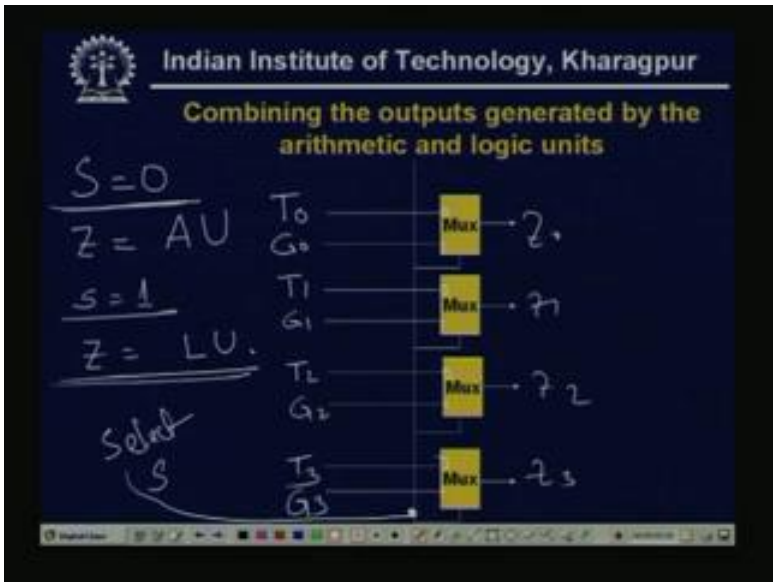
(Refer Slide Time: 25:44)



Now, we can combine the outputs generated by the arithmetic and logic unit as the ALU consist the, or ALU is has two parts, one is arithmetic unit and the logic unit. And just now, what we have done, the two way or two function arithmetic units and add and subtract we have designed two function logic unit AND and XOR we have designed.

If the output of the arithmetic unit that we can define as T 0 T 1T 2 T 3 our AU outputs and the logic units are G 0 G 1G 2 G 3. Then again another steps, what we can do, we can combine the logic units outputs and arithmetic unit outputs together by using some multiplexers.

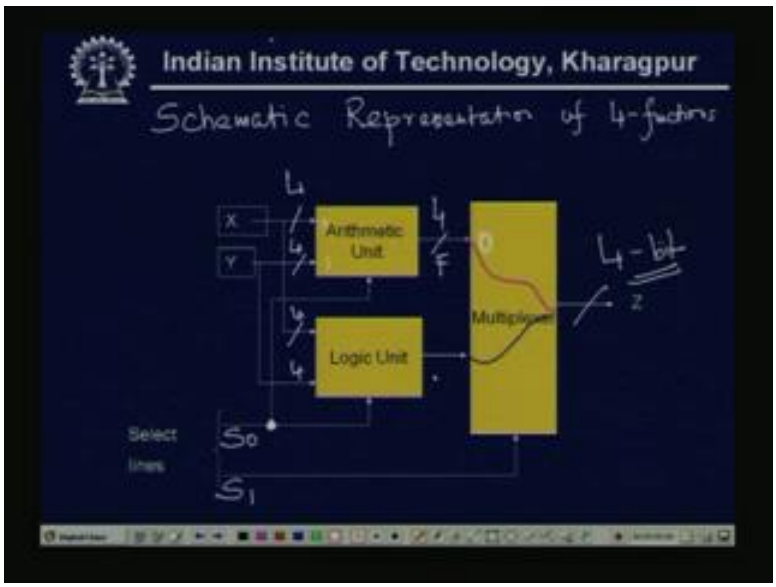
(Refer Slide Time: 27:46)



So, what we can do, say the T 0 T 1 T 2 T 3, the output of the arithmetic units come to that first line the zeroth line. Similarly, the G 0 G 1 G 2 G 3 comes to the multi other multiplexer inputs and as usually it has some select lines, it has the select lines S.

Now, if S equal to 0, then see these are my Z 0 Z 1 Z 2 Z 3, so if S equal to 0, the Z is actually the arithmetic units. If S equal to 1, the Z is the output of logic units, so in this way we can combine the arithmetic and logic units again using another level of multiplexers.

(Refer Slide Time: 29:26)



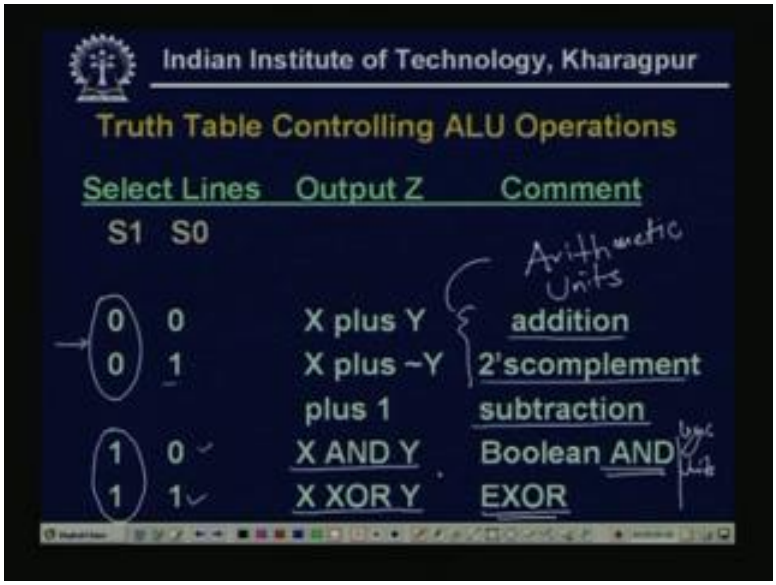
So, if we see the schematic representation, whatever be the unit, just as an example we have seen very small two function arithmetic units, addition and subtraction and two functional logic units the AND and the XOR.

Now, it can be any complex arithmetic functions, it can be any complex logic unit functions. Now, we are combine we want to combine these together, so this is a schematic representation of 4 functions. So this X is a 4 bit bus, Y is also 4 bit and this is arithmetic unit, again the output is four bit bus this is a F it goes to the multiplexers zero line.

Then, the logic unit again this is a four bit and it has two select lines because it has a two levels of multiplexers, so this is S 0 select lines S 0 and S 1. So, actually the S 0 lines it will select whether arithmetic units we need we need arithmetic functions or we need logic functions and then S 1 is selected, then which output will be we want say arithmetic unit or we want that logic unit as the output functions, again this will be the 4 bit. So, this is the overall schematic representation.

See here the select lines are this is a two bit and the multi two levels of multiplexers are there. Here, is actually S 0 selects which type of arithmetic unit, because it is all are two function arithmetic unit or two function logic unit. So, S 0 selects which function will select and S 1 selects that which unit to be output it arithmetic or logic. So, this is the schematic representation.

(Refer Slide Time: 33:23)

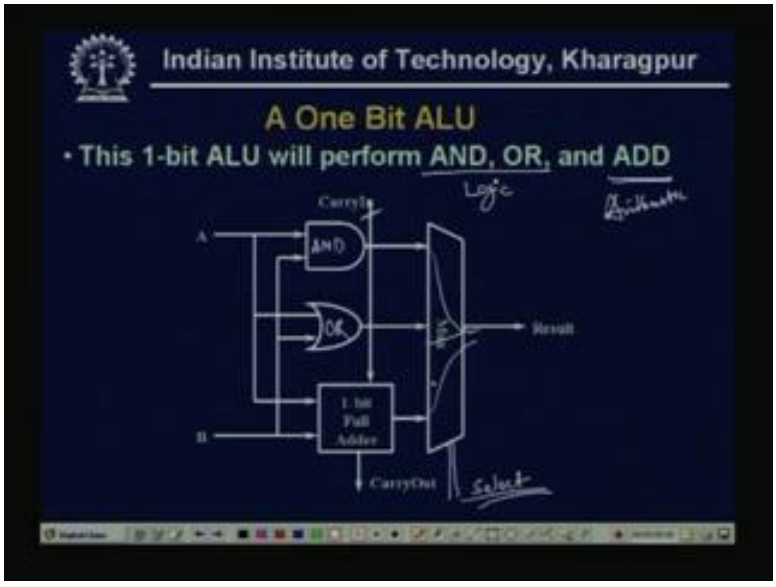


So, if we see the truth table controlling the operation if this ALU, so this is a truth table. So, I have two arithmetic or my arithmetic unit consists two functions, one is addition another is subtraction the 2's complemented subtraction.

So, if see my S 1 is 0, because this is a second level MUX selection, so this becomes 0. Now, if S 0 is 0 addition is selected if S 1 is 1, the 2's complements subtraction is selected; that means, that these two are my arithmetic units. So, if S 1 is 0 always arithmetic units is selected as the output; that means, Z is the output of arithmetic units and if S 0 is 0, then it will be addition if S 0 is 1, then it will be a 2's complement subtraction.

Now, the second level MUX if it is 1, then; obviously, logic units are selected, so these are my logic units and these are a Boolean AND and the EXOR. So, if it is 0, again if S 0 is 0, this is X AND Y, if S 0 is 1, then this is a X XOR Y. So, in this way we can the truth table, this is the truth table for controlling the ALU operations.

(Refer Slide Time: 35:47)

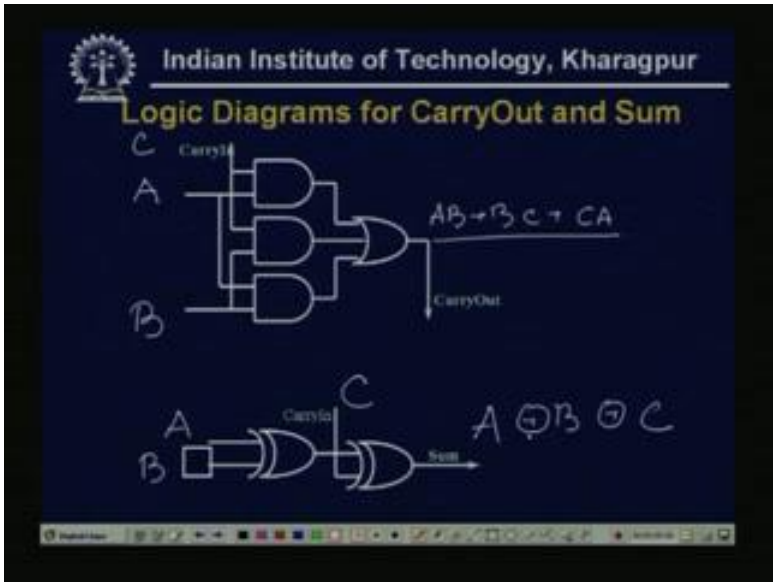


So far, what we have seen, now if we a draw the schematic diagram, say 1 bit ALU which perform AND OR and ADD. That means I am taking the logic operations as AND and OR. Logic operations as AND and OR and always this is a one function only, this is my arithmetic the arithmetic addition.

So, what it has two input, if it is a two input functions the A and B, then one AND is there this is my AND, this is my OR and the output goes to this as the input of the MUX. And, this is a one bit full adder, so it has two inputs AB and another input is carry in and the output goes to the another input to the MUX.

So, it has some MUX has the some select lines and accordingly the output will be or that particular input will be selected as the output of the multiplexer. So, these we can tell the schematic of a one bit ALU.

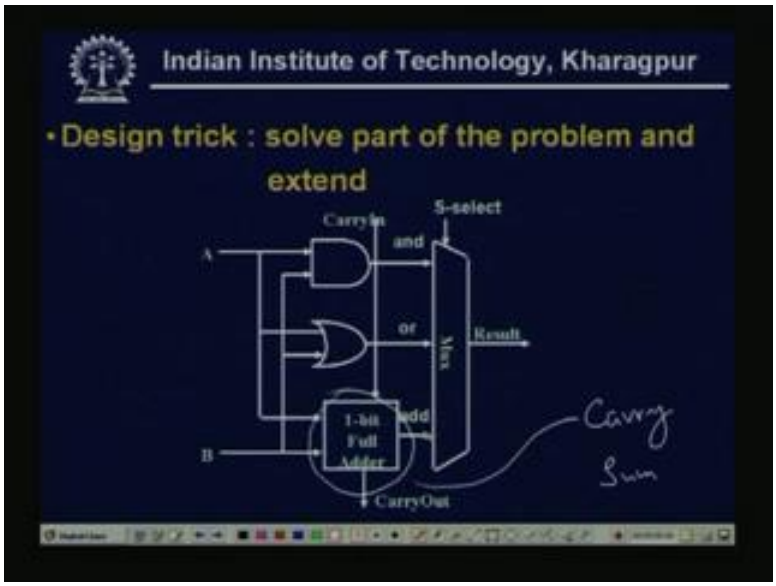
(Refer Slide Time: 37:53)



Now, say the one full bit for the full adder this is the logic diagrams for carry out and sum already we have several times we have read. We have the design of full adder and we know that this is the A if AB are the two inputs and is another is the carry in say a C, then we know that AB plus BC plus CA that is my AB plus BC plus CA is my carry out. So, this is the simple logic diagram

Similarly, the sum is nothing but A XOR B XOR C for the full adder, so if this is my A and this is my B 2 inputs this is my carry in C. Then, this is sum is A XOR B XOR C, so these are the a logic diagrams.

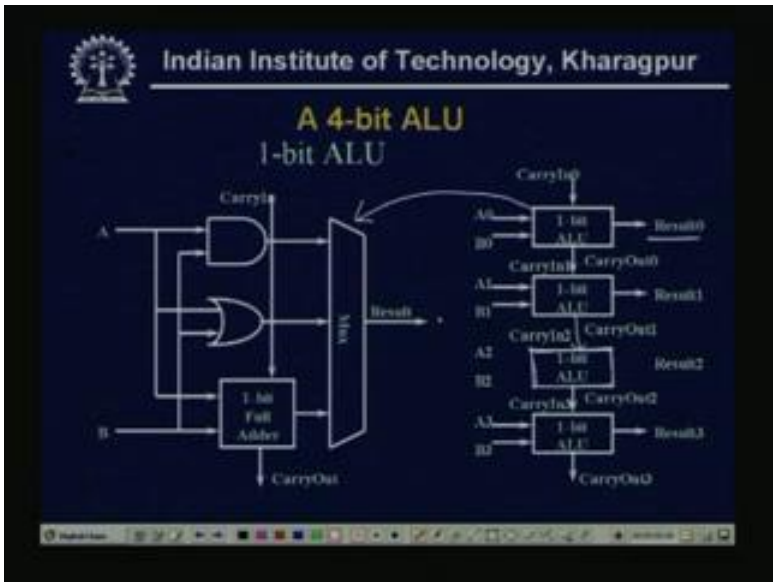
(Refer Slide Time: 39:09)



So, design trick is solver part of the problem and extended. That means first 1 bit full adder is designed, it has two parts the carry and sum. Just now we have seen we have design and then the other logics the one AND gate and one OR gate that we can do.

And then, the multiplexers it can it is fed to the AND OR ADD and it is fed to the multiplexers and depending on the select lines that particular function will be selected, whether it is arithmetic whether it is a particular type of logic unit and OR.

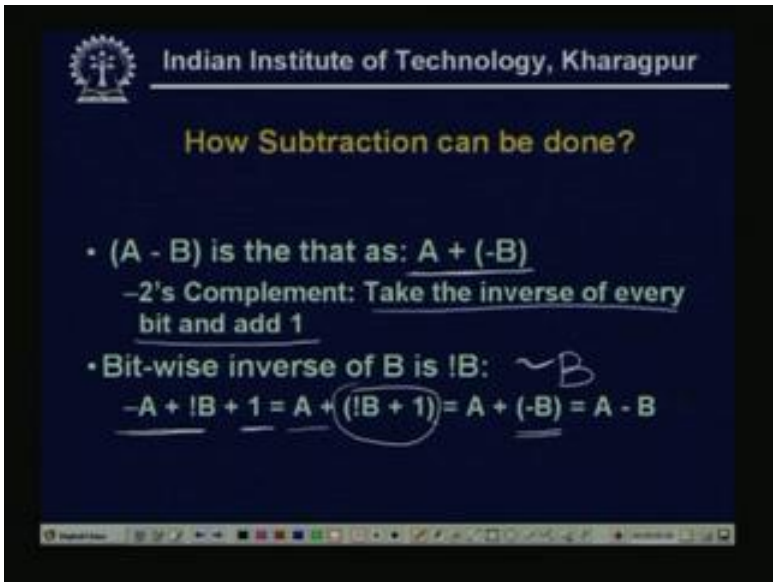
(Refer Slide Time: 40:04)



So, 1 bit ALU with the logic functions AND and OR and the arithmetic function as 1bit full adder can be designed, this is my one bit for ALU. Now, if I want a 4 bit ALU, so we have to add 4 such 1 bit ALU and this is the diagram for that. Now, it is A 0 B 0, if this is 1 bit ALU the block diagram is 1 bit ALU. So, it has two inputs A 0 B 0 carry in another input and the output is say I am giving result 0, because I do not know whether it is a logic functions whether it is an arithmetic function.

And then, the carry in because it if it addition then this carry will go as the input carry of the second ALU, so this is again another bit ALU and similarly another carry in tool propagate. Another 1 bit ALU in that way in this case, if it is a 4 bit ALU 4 such 1 bit ALU are concatenated or appended with the previous carry out as the carry in of the next ALU and it would be design like that. So, this is a general schematic for N bit ALU.

(Refer Slide Time: 42:19)



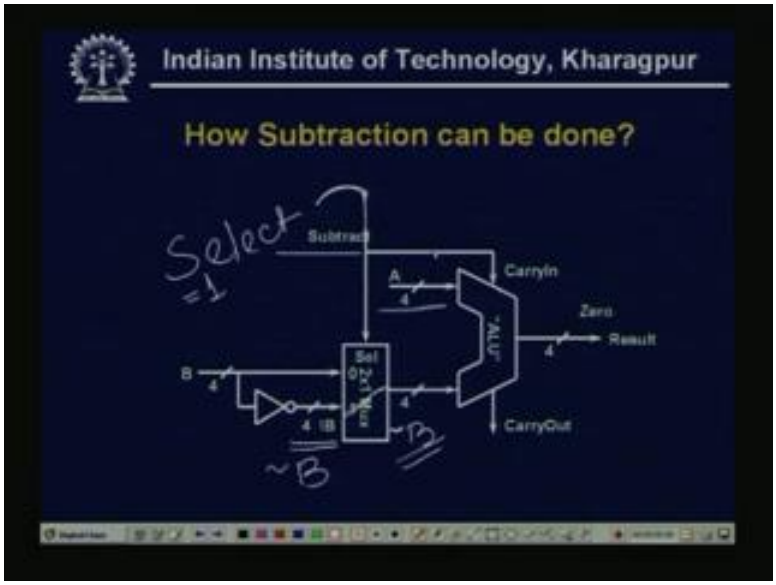
Now, we see how subtraction can be done, already we have seen that the adder the adder can be used as a sub tractor. So, A minus B is that nothing but A plus of minus B, we know 2's complement take the inverse every bit and add 1, this is you know that this is the 2's complement.

So, bit-wise inverse of B, that we have earlier example we have denoted as negation of B or B complemented and then minus A plus B means minus A plus negation of B plus 1, because inverse of every bit and add 1, this means that 2's complement. So, A plus negation of B plus 1 is A plus A plus if we consider this is as a negation of B plus 1

means B complemented plus and B complemented plus one is nothing but a minus B. So, this is A plus minus B means A minus B

So, in this way we can… that means, from the taking the inverse of every bit or complement of every bit we can do the subtraction, already we have done that way in this way we can do the subtraction.

(Refer Slide Time: 44:04)

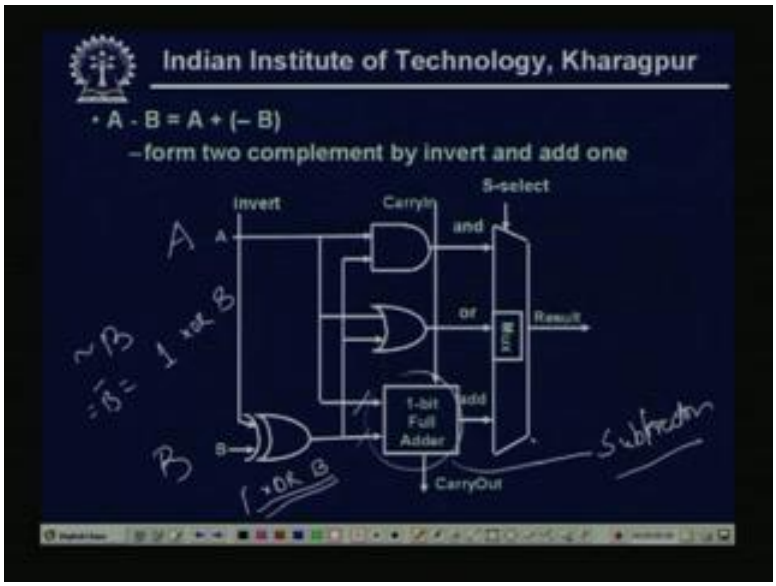


So, what will be the ALU for that, see that again we have taking a multiplexers and this is actually the select line, or we can tell this is the select line. So, if it is 0, then normal B and complemented B will go, but if it is select line is 1, then actually B complement is the implement; that means, every bit of B if it is a N bit of N bit B

Then, every bit of individual bit if complement and that will be given as the input of a multiplexers, so actually it will go to the output, so if it is a 4 bit, then the 4 bit inverse of every 4 bit will get here. And say, this is my arithmetic logic units, say it is a subtraction. So, here the 4 bit A values will go here.

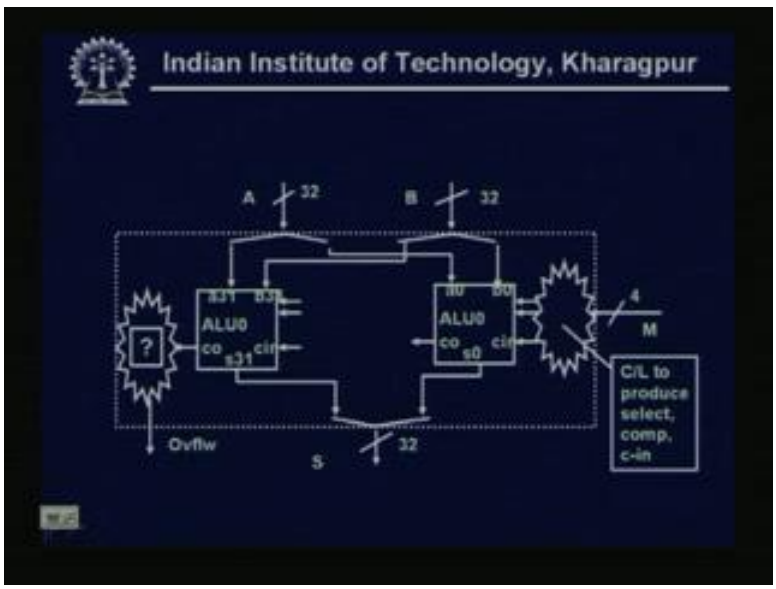
And, the 4 bit B complemented values and if the arithmetic unit is selected by these select line or that carry in. Then actually that the subtraction will be done at the output of this particular area and here is the carry out.

(Refer Slide Time: 46:04)



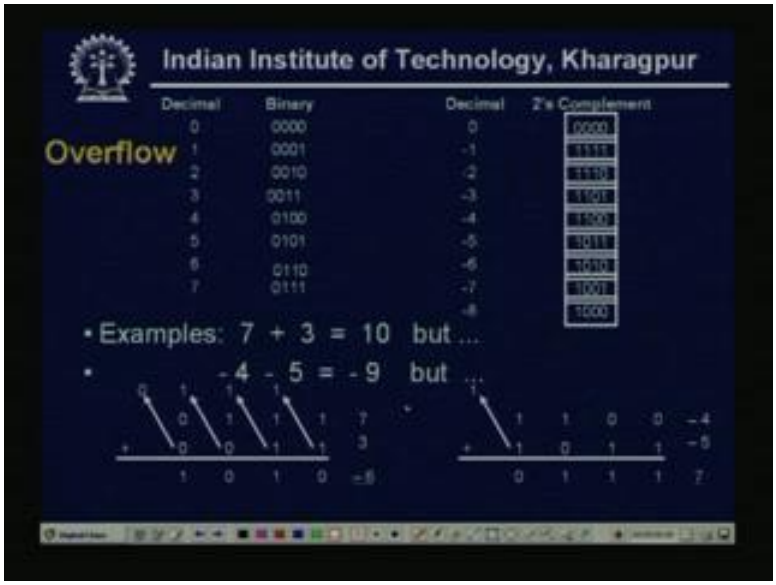
So, in this way individually the subtraction we can see, so the subtraction if the subtraction is an arithmetic unit, then using 1 bit full adder and we can do these thing. So, this is my two inputs A and B. Then say, A comes as it is to the full bit input of the full bit adder, but here B is inverted, B is not , this is nothing but 1 XOR 1 XOR B. So, this is 1 XOR B means B complement or NOT as already we have seen that B bar or B complement we can write as 1 XOR B. So, B complement is added to the full bit adder and if that is selected, then this particular it this bit adder will be actually a sub tractor, 2's complement sub tractor.

(Refer Slide Time: 48:22)



Now say, we have taken the two ALU and actually we are considering overflow, so if it is a 32 bit, A is the 32 bit, B is 32 bit and we are taking that 32 such ALUs that 1 bit ALU we have kept here. So, A 0 to A 31, so here some ALUs are there, A 0 to A 32, similarly B 0 to B 32 are fed. And these are the select lines, S 0 to S 31, and it produce that the combinational logics to produce select compare the carry in etcetera.

(Refer Slide Time: 49:29)

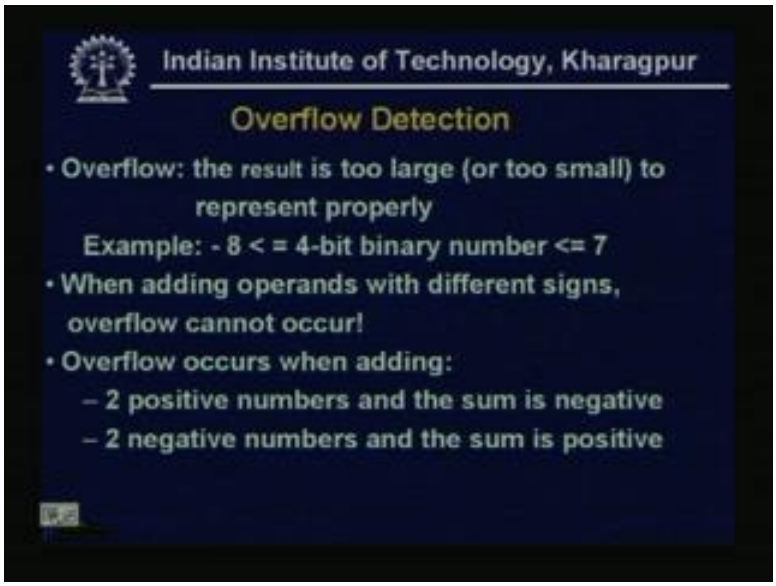


So, how the overflow can be detected, because whenever it is a addition or even subtraction the overflow can be there. See that, if we have the decimal values that 0, 1, 2, 3, 4, 5, 6, 7, these are my binary values, it is a 4 bit representation. Now, for these decimal values, this 2's complement representation, so these are for negative numbers. This is 4 bit 2's complement representation.

Now say, we want to add 7 plus 3 is 10, see if we see the 7 plus 3 is 10, but here if we add 7 plus 3 it gives that minus 6. That means, when two positive numbers sum of two positive numbers, the result is a negative number, then actually the overflow occurs. Similarly, here minus 4 and minus 5, so this is minus 4 is 1 1 0 0 2's complement representation minus is 1 0 1 1.

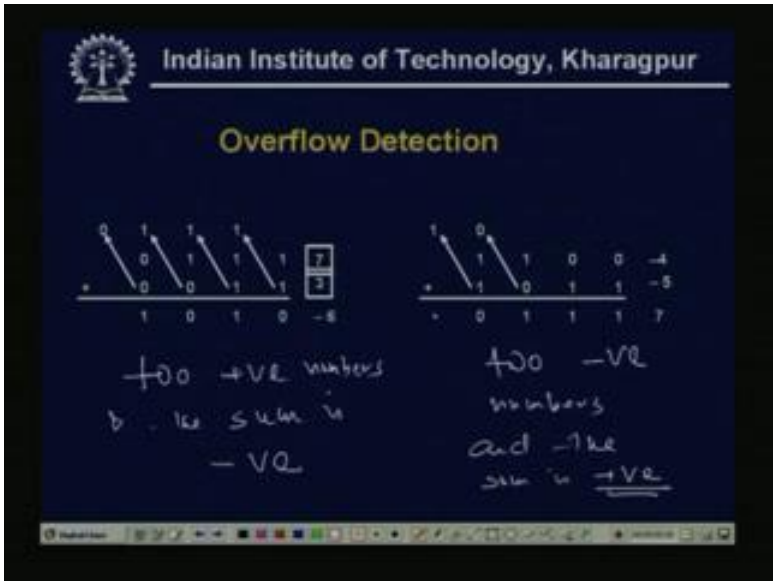
Now, if we add this 2 0 plus 1 is 1 0 plus 1 is 1 1 0 is 1 1 1 0, this is carry 1, see minus 4 minus 5, if we add we are getting 7 and that is a positive number. So, if the result is or the sum of two negative numbers is a positive numbers then overflow occurs. Similarly, here it is negative numbers minus 6.

(Refer Slide Time: 51:32)



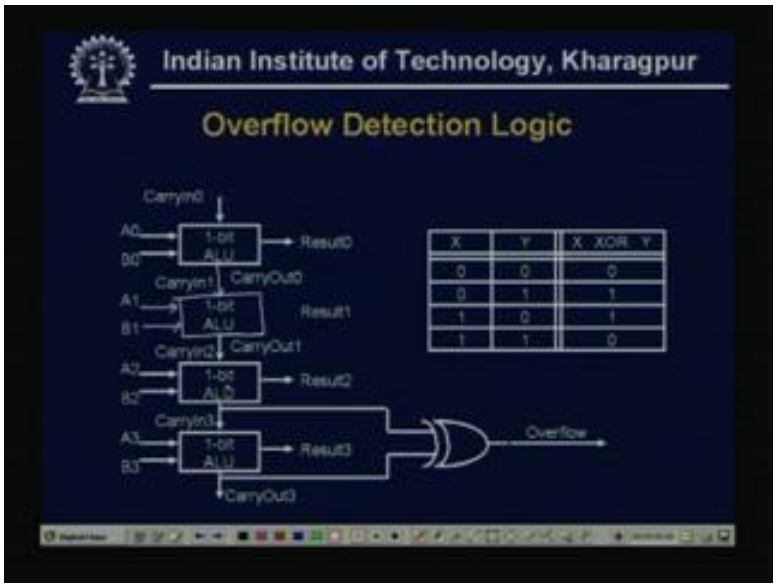
So, how we can detect the overflow, the result is too large or too small to represent properly. And example is just now we have seen when adding operands with different signs overflow cannot occur. Now, overflow occurs when the two positive numbers is a very important thing that two positive numbers and the sum is negative or other cases two negative numbers and the sum is positive. So, just now we have seen the two examples of each of the cases.

(Refer Slide Time: 52:22)



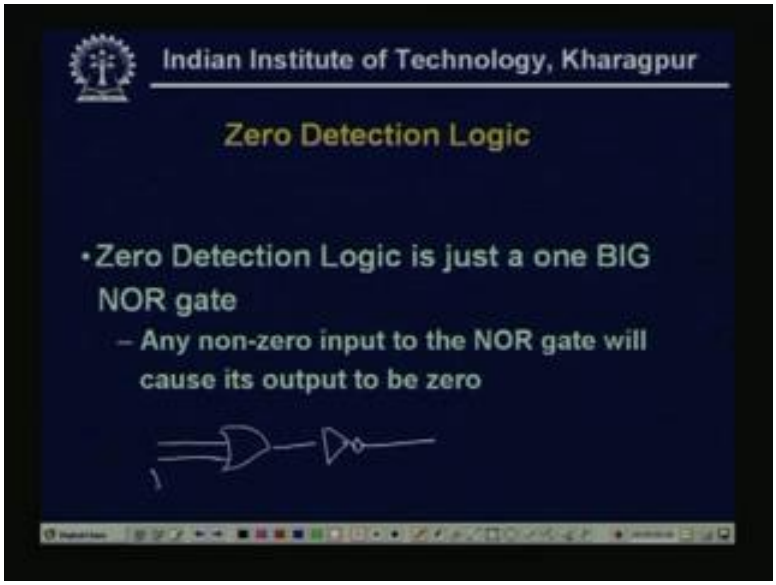
Each of the cases, how we can detect the overflow, these are the two cases that this is two positive numbers 7 and 3 two positive numbers and there is sum is negative. Here, two negative numbers and the sum is positive, so both the cases the overflow occurs.

(Refer Slide Time: 53:22)



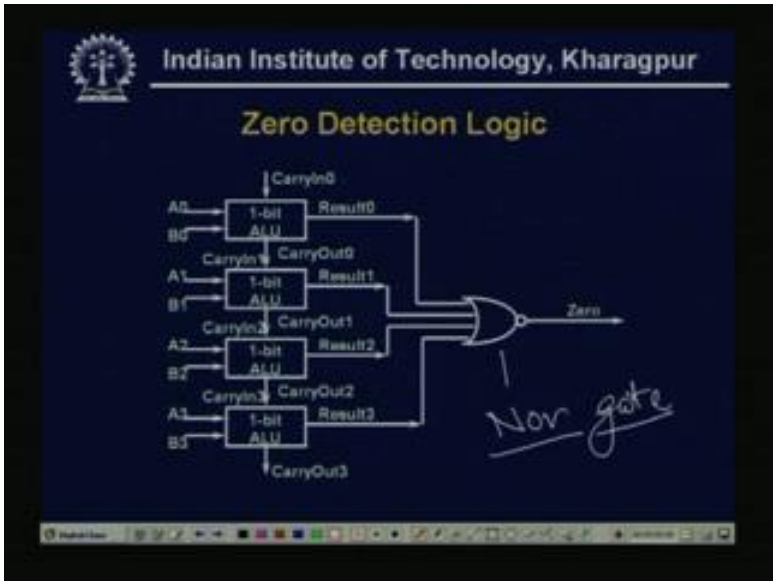
Now, how we can detect, this overflow detection, so if X Y 0 0, then X XOR Y is 0. This is my XOR truth table. Now, this is actually that 1 bit ALU, so I have taken a 4 bit. So, if it is a I have taken A 0 B 0 A 1 B 1 A 2 B 2 A 3 B 3 and these are my carry in 1 carry in 2 carry in 3. So, now the carry in 3 and carry out 3, if we take the XOR of these 2 carrying 3 and carry out 3 and we take the XOR, then this will detect the, it will detect the overflow. So, this is the overflow detection logic.

(Refer Slide Time: 54:31)



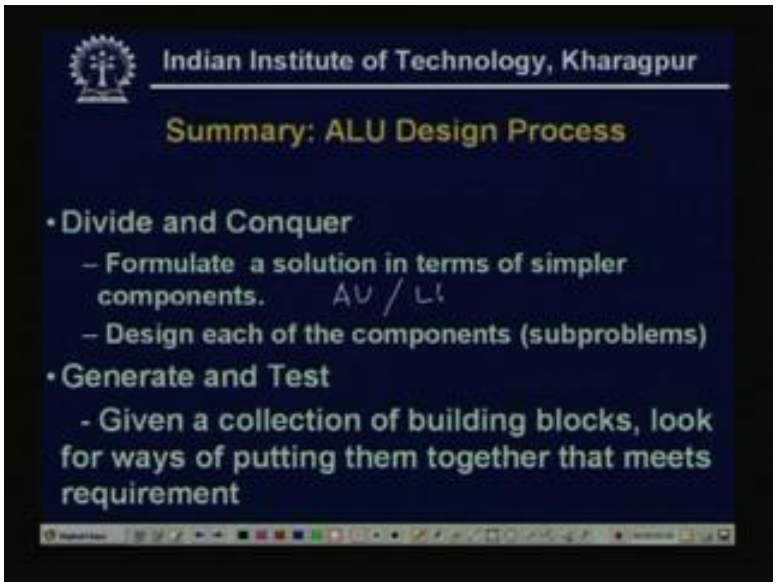
Now, zero detection logic, so zero detection is just a one BIG NOR gate. Any non-zero input to the NOR gate will cause its output to be 0, because NOR is complement of OR is that if anyone at least if 1 bit is one input is 1,then it will be output will be 1. So, NOR is that if 1bit is 1 is actually my NOR gate, so anyone is 1, then it will be a 0, so that logic we can use for a zero detection.

(Refer Slide Time: 55:25)



So, again if we use a 1 bit ALU just there are 4 bit ALU A 0 B 0 A 1 B 1 is given and the four outputs result are this is simply a NOR gate, So, you A 0 at the output gives a A zero detection.

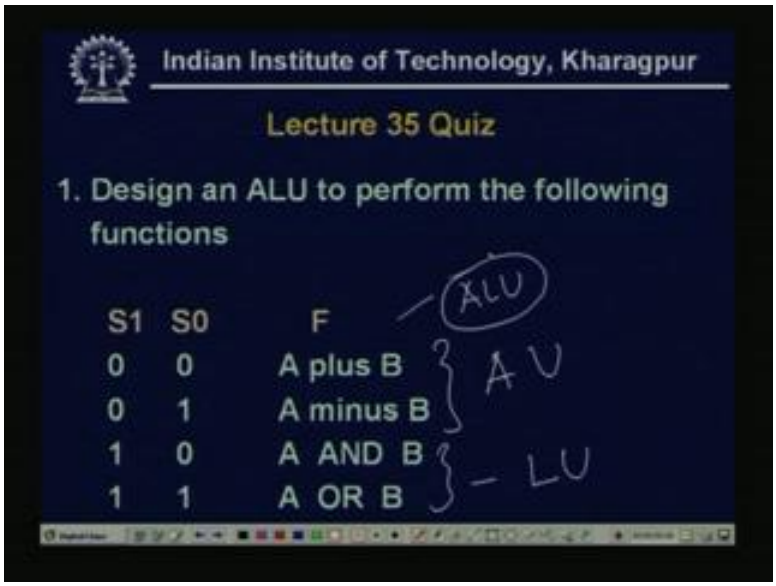
(Refer Slide Time: 56:06)



So, if we to the summary the of the ALU design process that there are two main concepts one is divide and conquer; that means formulate a solution in terms of simpler components. So, if it is a ALU means either arithmetic units or logic units that first we will be designed using gates or our simple gates.

Then, design each of the components or sub problems and generate and test, so given a collection of building blocks look for ways of putting them together that meets the requirement. So, these are the two fundamentals of the two basic concepts for the ALU design process.

(Refer Slide Time: 57:00)



Now, the today's quiz question is that one simple ALU design, the design ALU to perform the following functions. So, again here we have taken the two function arithmetic units and two function logic units, so if the select lines as 0 0, it will addition if it is 0 1, it will be a subtraction if it is 1 0, it will be a simple AND if it is 1 1 is a A OR B. So, design an ALU, so that it performs this arithmetic logic units, this is the quiz questions of this. So, today we have seen the arithmetic logic unit design and next day we will continue the control unit design.

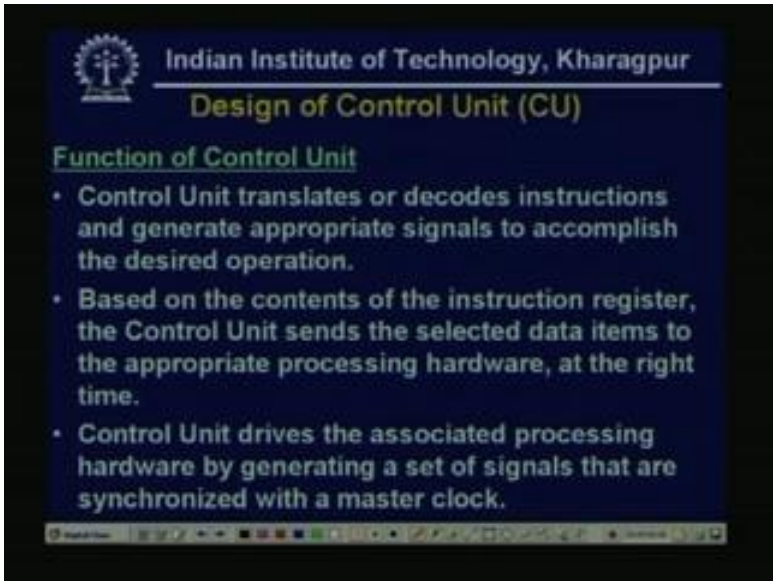
Digital Systems Design
Prof. D.Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
Design of Computer Instruction Set and the CPU (Contd.)

We are discussing the design of a CPU in the last class. We have read the design of arithmetic logic unit the ALU and today we will read the design of control unit.

Control unit as its names implies actually it is the unit of mastering the interfaces of all other units of the CPU. So, first we see that what is the function of control unit? So control unit translates or decodes instructions.

(Refer Slide Time: 58:48)



And generate appropriate signals to accomplish the desired operations, so we have read the design of instruction set there.