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

Lecture - 37
Design of Computer Instruction Set and the CPU (Contd.)

We are discussing the design of CPU, all of you know the main part of CPU are actual 2, the 1 is the arithmetic logic unit ALU, and another is a control unit, these are the main two parts. Already we have read, the design of ALU and now we started the discussion, on the design of control unit. Last day, we have read the basic concepts of designing the control unit and we have seen there are two approach of designing CU, the control unit. One is the hardware control unit, another is the micro programmed control unit.

We have read last day, the different type of bus architecture, that single bus architecture, the two bus architecture, three bus architecture. And using the different type of bus architecture, actually that, how that how the performance varies, if we give the hardware, or actually the optimization between the hardware and the speed, that we have seen. Today, we will read the hardware control design.

(Refer Slide Time: 02:11)

First we see, that what are the steps involved, in hardware control design, now the step 1 is, derive a flow chart, from the problem definition and validate the algorithm, by using a trial data. So, given a problem, the first we derive the flow chart and flowchart means,

actually, if a flowchart is given, that algorithm we can get from the flowchart as it is. Now, validate the algorithm, by using a trial data, that means, whether it is correct algorithm or not, or the flowchart we have got from the problem definition, whether, actually it is given the right result or not, that we have to validate.

Step 2, obtain a register transfer description, of the algorithm from the flowchart, step 3, specify a processing hardware, along with various components, step 4 complete the design of the processing section, by establishing the necessary control inputs. So, from here we have the, we identify the processing hardware, with the components needed and then we complete, that design of the processing section. That means, what input is needed, or to interface between the components and the processing section.

(Refer Slide Time: 04:07)

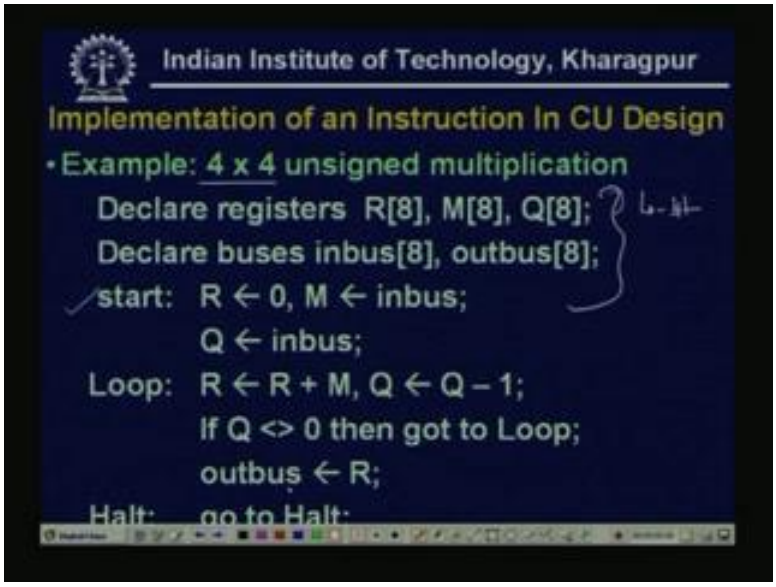
Then step 5, determine a block diagram of the controller, because we want a hardware design, so first we, from their now the controller, block diagram level controller design, we have to do. Then step 6, obtain the state diagram of the controller, once the block diagram is achieved, then we from there, we draw the state diagram, because it s nothing but, a machine. It has some input. then it will generate some output, step 7, specify the characteristic of the hardware, for generating the, require timing signal, used in the controller.

Because, the controller will, provide the sequences generated in different time instances that means, in which time instance, what control signals will be generated, that is the

function of the controller. So, that characteristics of the, hardware, for generating the required timing signal, because this is a very important thing in controller design, then draw the logic circuit of the controller. Now, we will do the, once the timing signal also we have got, hardware we have got then the logic circuit of the controller, it is d1.

Now, based on these steps, one by one, if we see the details, then we will be seeing that, how that actually the control unit will be designed, using the approach hardware control design.

(Refer Slide Time: 06:06)

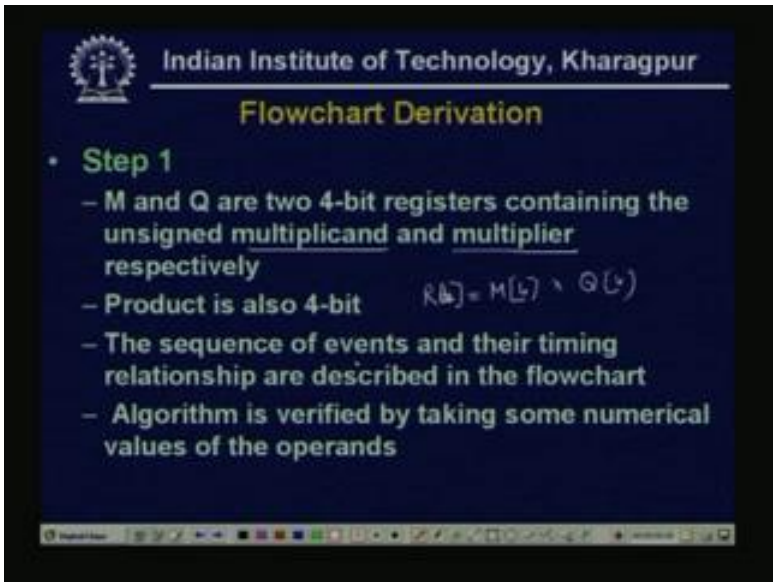


Now, we take one example, again we will see the day's example, that 1, 4 by 4 unsigned multiplication. So, already we have seen that, we have to declare some registers, and 3 registers are declared, here it should be, if it is a 4 by 4, this R 8 can be R 4, M 4, Q 4, last day we have seen that, this example. Now, declare buses, in bus 8 out bus 8 or in bus 4 out bus 4, so these are 4 bit, here it is 8 bit registers, here 8 bit in bus out bus, for 4 by 4, 4 by 4 we can 4 bit registers, and in bus, out bus can be defined declared.

Now, the actual algorithm, and these algorithm is nothing but, the repetitive multiplication, is nothing but, repetitive addition. So, initially, in the initialization position or the start loop, that register R is set to 0, and the data of in bus is given to registers M. In the next clock, the data of in bus is set to Q registers, that means, 1 is multiplier, another is multiplicand, now as it is a repetitive addition, so R is R plus M, that means, this multiplier, multiplicand is will be added Q times.

M will be added Q times, then it will give QM, so R is R plus M, what we are doing, every time after the, this addition Q is decremented by 1. And then, if Q not equal to 0 then go, to loop, out bus is R, and then, halt go to halt, that means, when Q equal to 0, that means, Q times it is loop, it is in loop then, it is go to halt.

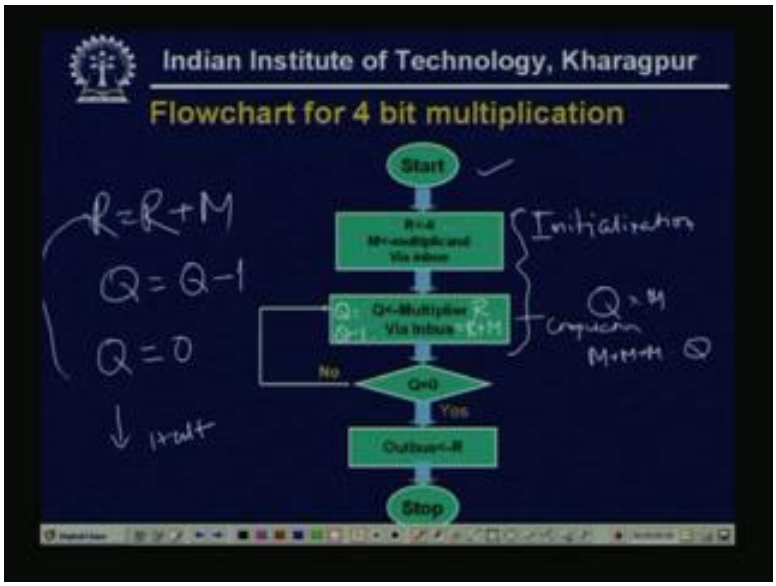
(Refer Slide Time: 08:46)



Now, the step 1, using this example the 4 by 4 multiplier, simple multiplication then we 1 by 1, we see that how the steps of the hardware control design is followed, and we can get a actual implementation of the control unit. Step 1 was, derive a flowchart from the problem definition and validate, the algorithm by using, that trial data, see M and Q are, 2 4 bit registers, obtaining M and Q are 2 4 bit registers, and they contain the multiplicand, and the multiplier respectively.

Then product is also 4 bit, that M is of 4 bit, Q is of 4 bit, and the product we this is a R is also 4 bit, this is the thing. The sequence of events, and the timing relationships are described in the flowchart.

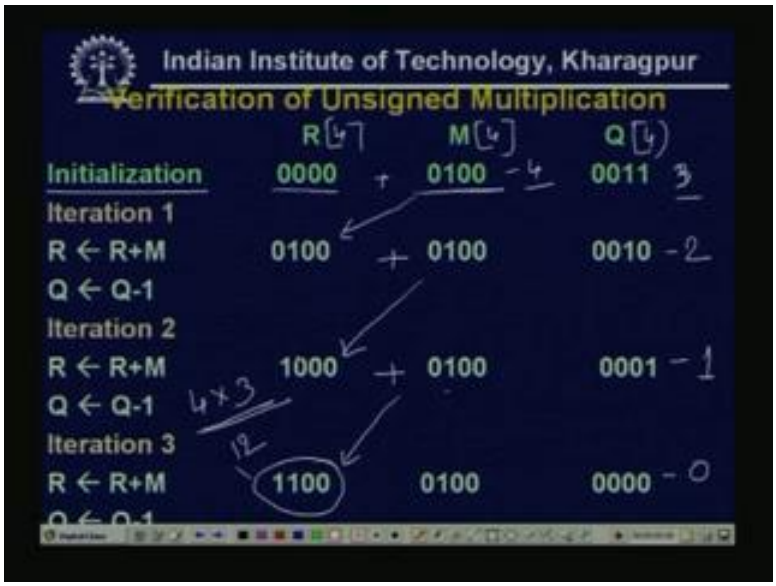
(Refer Slide Time: 10:18)



See we see the flowchart, first we start, then in the initialization part, this is the initialization portion, we have see, that register R is set to 0, and M is the multiplicand via in bus, Q is also multiplier via in bus, so these 2 are actually the initialization portion. Now, the actual operating portion, if, say if Q equal to 0, so if Q equal to 0, then ((Refer time: 11: 13)) it is, it program should be stopped, that means, M M is M should be, this Q M is nothing but, M plus M plus M Q times.

So, what we have to do again here, so we have to do that, R is we have to do here, R equal to Q plus M, R is R plus M, we can do that, R is, R is R plus, R is R plus M. So, this is a flowchart, 1 initialization part 1, computation part, computation part R is R plus, computation is R is R plus M. Then, if whether Q is 0 or not, Q equal to, Q equal to Q minus 1, and then we have to see whether Q equal to 0 or not, if 0 then halt. Q equal if not, then again we have to do, the R equal to R plus N again, so what we have to do here another step, Q equal to Q minus 1.

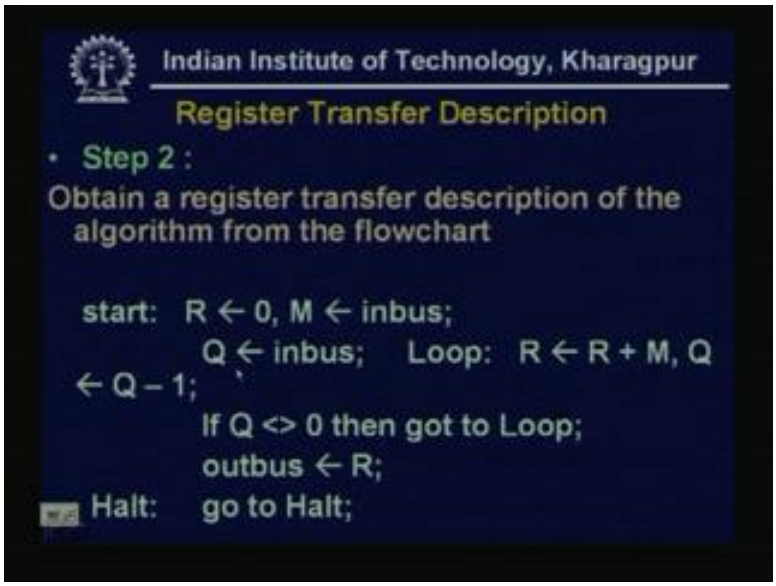
(Refer Slide Time: 13:01)



Now, how do we verify, second thing was validate the, the flow algorithm, a verification of unsigned multiplication. So, we have 3 registers R M Q say these are 4 bit registers, R M and Q, then initialization we take some actual data, say R is initially set to 0, so all 0, M is 0 1 0 0 means, decimal value 4. And Q is 0 0 1 1 means, the decimal value 3, now in iteration 1, R is R plus M, so it was, earlier these 2 will be added, and result is here, so all 0 plus 0 1 0 0, this is 0 1 0 0.

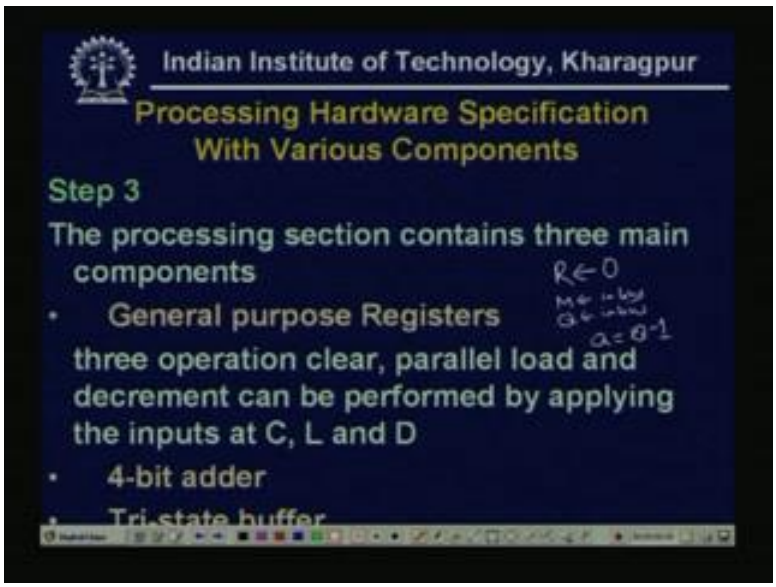
M is same no operation, then Q is Q minus 1, so it was 3, so Q minus 1, now it becomes 2. Now, iteration 2, again R is R plus M, so these two are added, and it becomes 1 0 0 0, M no change, Q becomes Q minus 1, 1, 0 0 0 1. Similarly, in iteration 3, 1 0 0 0 plus 0 1 0 0, this becomes 1 1 0 0, and Q becomes, Q become 0, Q minus 1 is 0, then it will stop. So, the result will be 1 1 0 0, which is decimal value is 12 and it was 4 into 3, so that is equal to 12, so it is validated, so the algorithm is validated.

(Refer Slide Time: 15:24)



Now, step 2; step 2 was obtain to register transfer description of the algorithm, from the flowchart. So, again using the same example of multiplication, R is set to 0, M is in bus, Q is assign to the data of in bus, and then, this loop is R is R plus M, Q is Q minus 1, if Q not equal to 0, then go to loop out bus is R halt go to halt. So, actually what, algorithm we, we have decided, the algorithm we have read, it is there, in the register transfer logic.

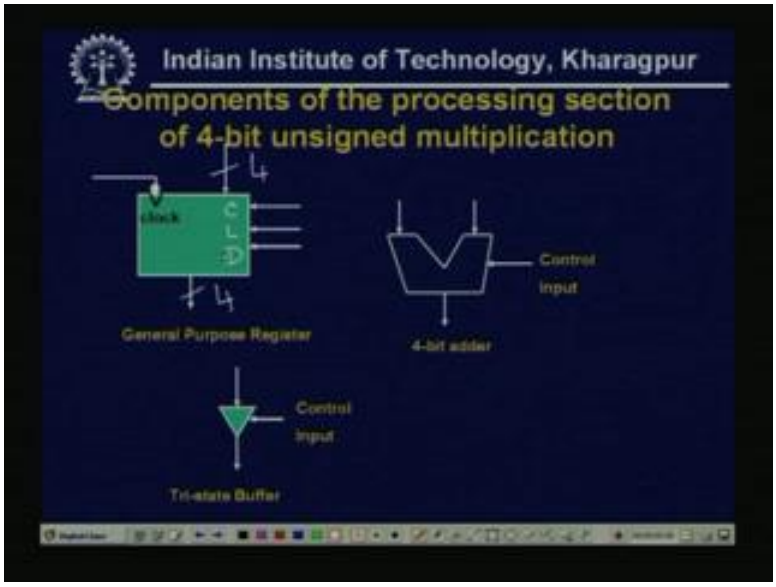
(Refer Slide Time: 16:50)



Now, the step 3, the processing section, step 3 was specify a processing hardware along with various components. So, the processing section contains, three main components,

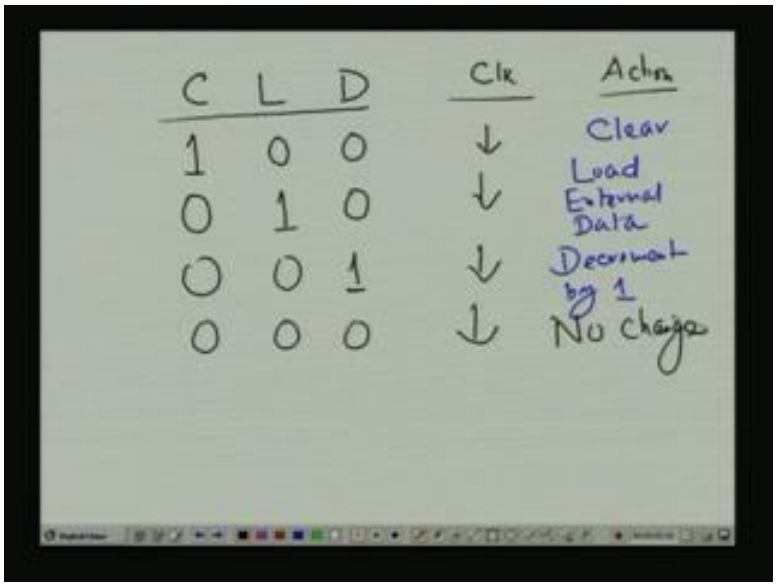
one is general purpose registers, so three operation clear parallel load, and decrement can be performed, by applying the input sets C L and D. If we remember the operations one is R is set reset, the register R is set to 0, 1 is R is set to 0, that is clear parallel load, is M is in bus and Q is in bus, that is the operations, the multiplicand and multiplier should be loaded. Another, is decrement, Q is Q minus 1, so these are my operations to be done, so these three operations are performed by applying the inputs at C L and D, now another component is a 4 bit adder, and tri state buffer.

(Refer Slide Time: 18:09)



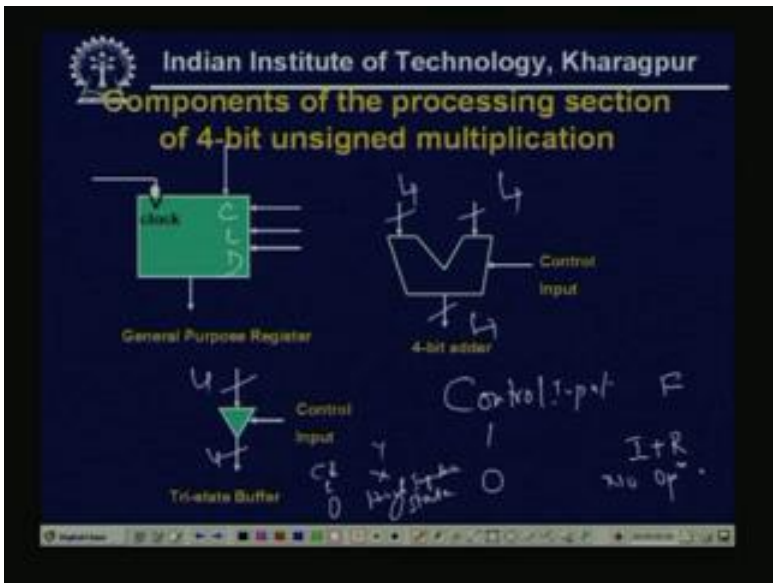
So, this, components are like that, one is the general purpose register, so this is the general purpose register. One clock is there, this is a 4 bit input, 4 bit output, and these are the three inputs C L and D.

(Refer Slide Time: 19:04)



So, C L D, the if we see the, the operations of these three inputs, so these are the values of C L D, when C value is 1, L D 0 0 then, the clock, say if it is a trailing edge triggered, then action is, the action is the clear, action is clear. Now, C is 0, L is 1, D is 0 clock is trailing edge triggered, this is load external operation will be load external data, load external data. Now, 0 0 1, so again it will be, the decrement operation is a, decrement by 1, decrement by 1. If it is 0 0 0 C L D clock is trailing edge, operation is no change, there is a no change, so these are the three inputs.

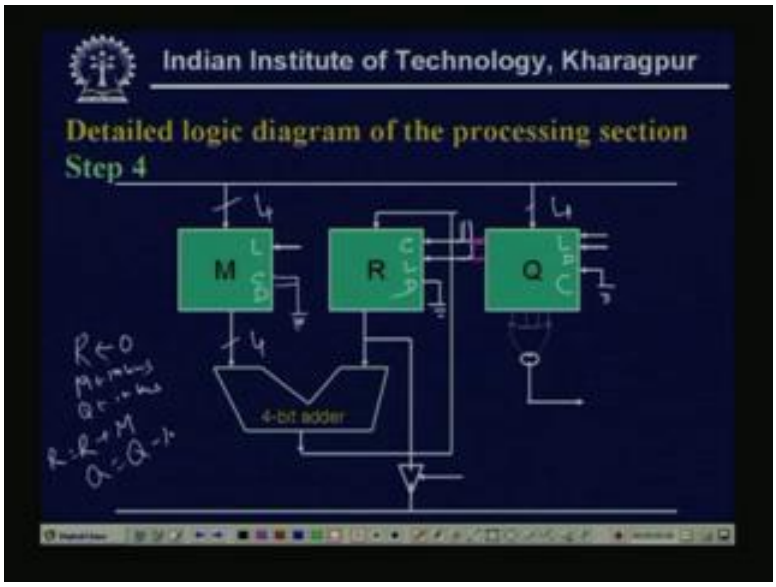
(Refer Slide Time: 20:28)



Now, another component is a 4 bit adder, so already we have read, this is a 4 bit, 2 4 bit operands, and the result is 4 bit, so if the control input comes, say the control input is, for these thing, the control input, either 1 or 0, if it is 1, the ((Refer Time: 21:04)) function is, say 2 things are added, I plus R, if it is 0, no operation, no operation. Now, another is a tri state buffer, so again if it is a 4 bit, 4 bit tri state, one control input, and the control input, either, it will 1 or 0, and the output will be either 1 is, when control input is 1.

Then it is, whatever input it is coming or high impedance, when it is 0, that high impedance state, when it is a 0, it is a high impedance state. So, this is the design of the three main components.

(Refer Slide Time: 21:58)



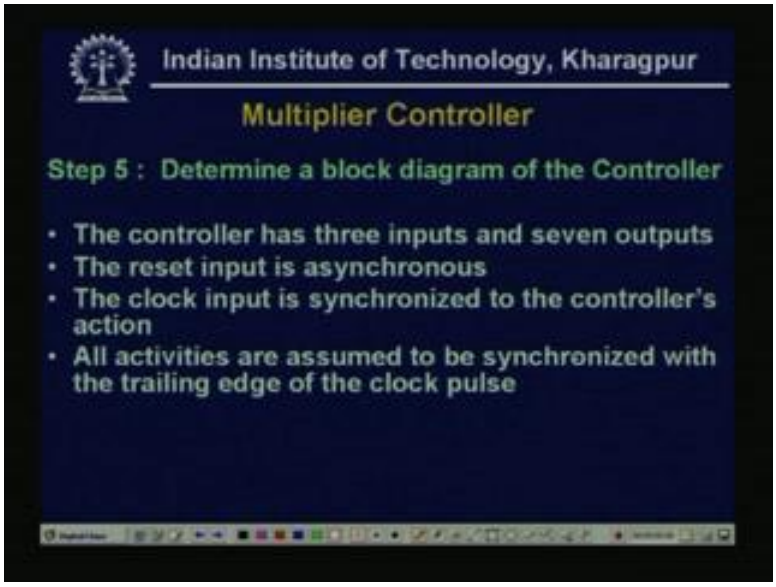
Now, step 4, the step 4 was, the complete the design of the processing section, by establishing the necessary control inputs. Now, we see the, detail logic diagram of the processing section, along with control inputs, we have seen, the three registers M R Q, the R is register, M and Q are the multi containing two registers, containing the multiplicand and the multiplier. Now, R is R plus M, initially R is set to 0, and then R is R plus M.

So, one 4 bit adder is needed, and Q is Q minus 1, that is to be done, and then, it is to be checked, that how many times, it will be added, so we need this, this is the detail logic diagram of the processing section. So, the all these buses will be 4 bit, because this we are this, illustrating the 4 by 4 multiplier, now these are 4 in bit, these are L and then C

D, these are grounded. This should be C L, this should be grounded, again these, these values should be coming from, here form, outside CL not from, not from here, now these are L D and C.

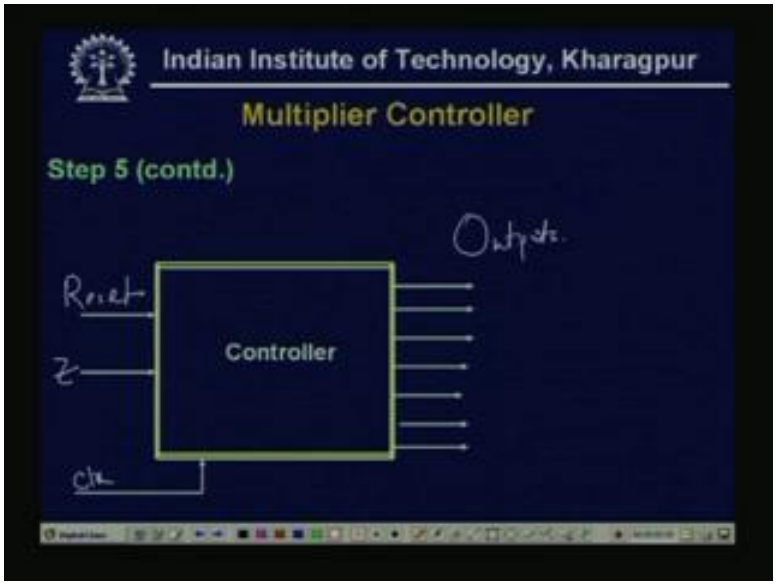
So, actually, that three operations it will be doing, one is R is set to 0, then, M is in bus, Q is in bus, assign to in bus means data, multiplicand and multiplier are loaded, then R is R plus M, R is R plus M, and Q becomes Q minus 1, and then it will be, checked whether Q equal to 0 or not, so this is the processing section.

(Refer Slide Time: 25:25)



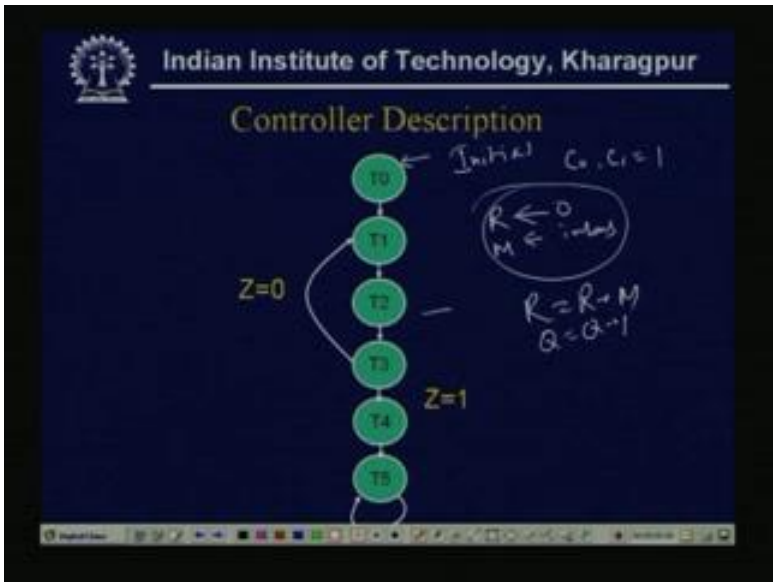
Now, step 5, determine a block diagram of the controller, now, the controller has three inputs and seven outputs, the reset input is asynchronous, the clock input is synchronized to the controller section. All activities are assumed to be synchronized, with the trailing edge of the clock pulse, so this is my step 5.

(Refer Slide Time: 25:59)



So, what will be the controller design, so controller has one input is reset, controller one input is reset, this is a Z input, this is a clock. And seven outputs, these are seven, these are the outputs.

(Refer Slide Time: 26:53)

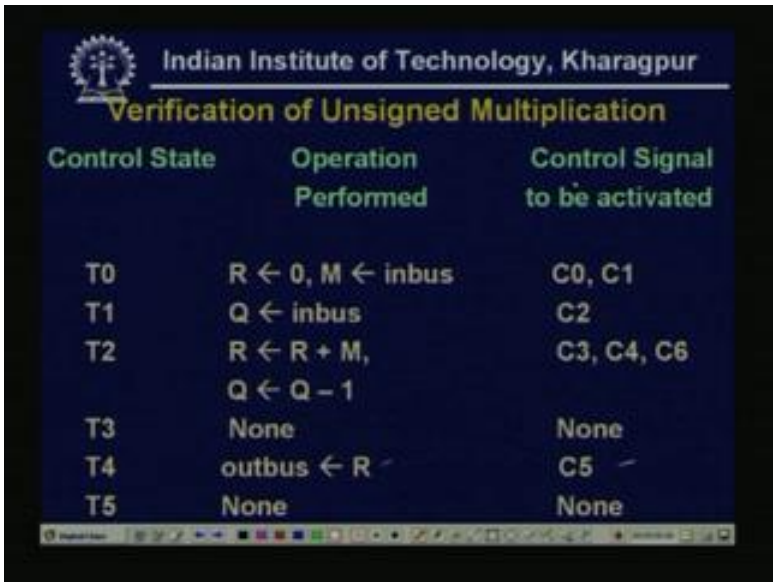


Now, we have to, see that, step six will be the, step diagram of the controller now controller must initiate a set of operations, in a specified sequence. Therefore, it is model as a sequential circuit, so the state diagram of the unsigned multiplier controller, we can draw, in this way. Say, these, these are the different states T 0, T 1, T 2, T 3, T 4, T 5. now initially the controller, is in state T 0, is a initial state, now at this point, the control signal C 0 and C 1 are high, and C 0, C 1 equal to 1.

Now, operations R is reset, R is clear and M assign to in bus, or carried out with trailing edge of the next clock pulse. So, the controllers needs to, state T 1 with this clock pulse, that means, after initialization after, one clock pulse, these two operations, that R set to 0 and M set to, in bus these two operations will be followed in this. Now, this say, this state is a controller moves to state T 1, then the controller is in state T 2, then R is R plus M, and Q equal to Q minus 1, these two operations will be performed.

Now, all these operations are takes place, at the trailing edge of the next clock pulse, and the controller moves to state T 5, only when the unsigned multiplication, is complete. So, this is a last state, the controller then stays in this state, and a, hardware is at input, causes the controller to move, to state T 0, and a new computation will start. Now, we see the, in this state diagram, the selection of states, how the selection of states, is made.

(Refer Slide Time: 29:12)

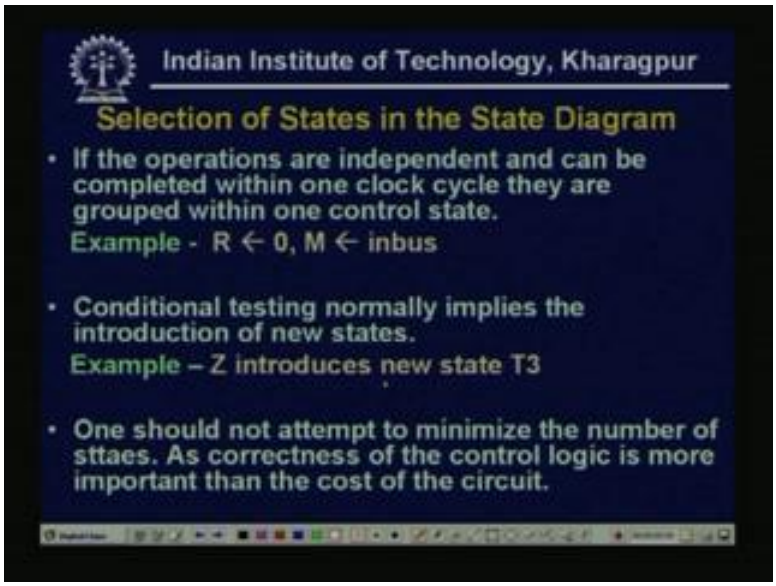


See, there are, we take the control state, operation perform and control signals to be activated, just now, I mentioned, that at T 0 state, T 0 state, R this is a initialization, so R is reset to 0, M is in bus, and then C 0, C 1 should be high. At T 1 state, Q is in bus, Q is in bus, and then C 2 is high, T 2, R is R plus M, Q is Q minus 1, say C 3, C 4, C 6 this three will be high, then T 3 none, T 4 is, R is go, R goes to out bus means, the result goes as a output.

And then, C 5 is high, T 5 means, it is completed operation is multiplication is completed, and nothing, could be done, so none, none. So, these are the control signal to

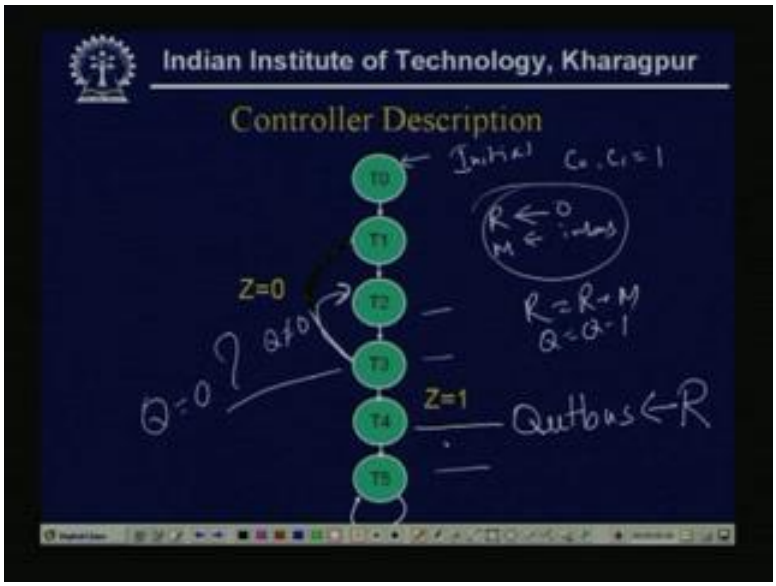
be activated, so from the state diagram, we get this information, that these are the control states, and then, I have to get this, control signal to be activate means, this control signal should be high. And when this control signal would be high, then these operations, should be performed, so we get this information.

(Refer Slide Time: 30:40)



Now, how the states are selected in this state diagram, now if the operation are independent, and can be completed within one clock cycle, then they are grouped within one control state. Example is, that R, if R register is reset to 0, and M is set to multiplicand that means, data, that time the data, contained in the in bus, then these two are actually independent, now one is effecting. So, these are grouped together, and they are in one state. Conditional testing, normally implies, introduction of new states, so Z introduces a new state T 3.

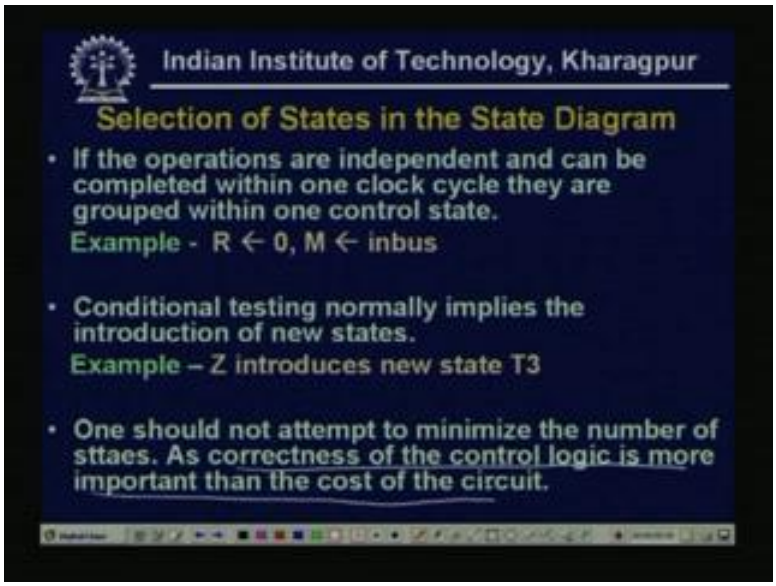
(Refer Slide Time: 31:35)



Now, we see the previous say this at T 3 state, so this, a, we are this is a, actually we are checking, whether, whether Q equal to 0 or not. Now, if Q not equal 0, then actually, we will go back to T 1 that means, again we have to perform R equal to R plus M, Q equal to Q minus 1. So, this should go, this should go here, not this T 1, because this is a reset state, this is a, this should go a T 2, then, if Z equal to 1 that means, that other condition is Q equal to 0, then will goes to T 4 state.

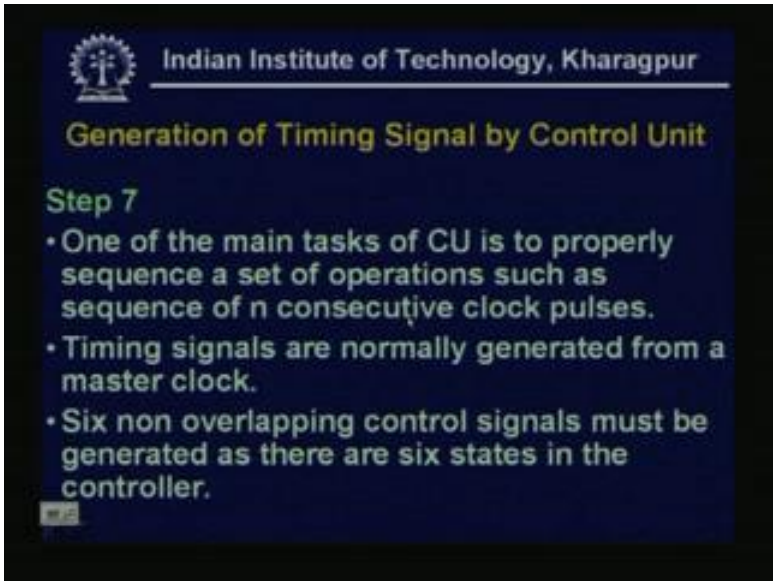
That means, it denotes that, that R is, in this T 4 state, this T 4 state, actually out bus, out bus becomes the R, out bus is assigned as R, and T 5 is nothing could be done. This already the operations of multiplication is completed.

(Refer Slide Time: 33:23)



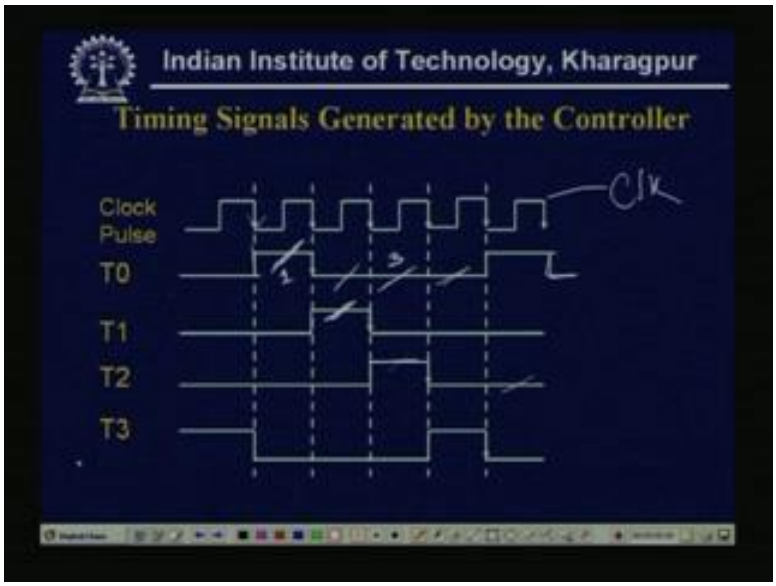
Now, one should not, attain to minimize the number of states, as correctness of the control logic is more important, than the cost of the circuit. So, for the first case for the initial design of control unit, we will not minimize the number of states, as it is a machine, so we can run, we can do the state minimization. But, as correctness of the control logic is more important, that is, that is here, our M is not the actually, the minimization, but to get the actual correct design of the CU.

(Refer Slide Time: 34:04)



Now, step 7 the general of time signal by control unit, see one of the main tasks of CU is to properly sequence a set of operations, such as, sequences of n consecutive clock pulses. And timing signals are normally generated from a master clock, so six non overlapping control signals, must be generated, as there are six states in the controller.

(Refer Slide Time: 35:04)



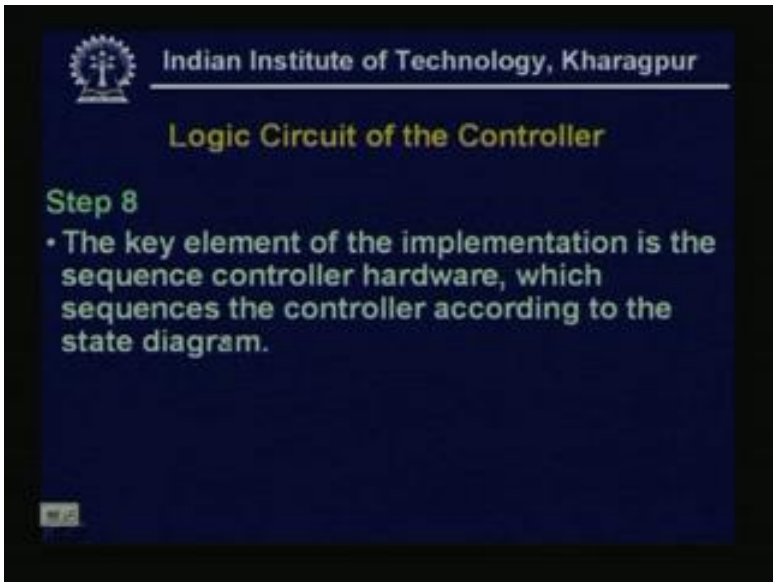
Now, we see one example that how the timing signals, are generated, see, we have, one we have 1 master clock pulse. So, there are six states in the controller state diagram, so six non overlapping timing signal T 0 through T 5, must be generated. So, that only one will be high, for a clock pulse. Now, here we have, shown that 4 timing signals T 0, T 1, T 2, T 3 say this is a master clock pulse, this is my master clock pulse.

And, say this T 0 clock pulse, this is a, this is a different clock and when it is high, say here, that, the this T 0 is high, at the trailing edge of the master clock pulse, after the first clock the trailing edge of the master clock, it becomes high. And, the again it s high, after the next trailing edge of the master pulse, so in this way, it will this is one high, and then 1, 2, 3 clock pulses, after 3 clocks of the master clock, it becomes low. So that means, in 1 clock pulse, of the master clock pulse it becomes high, and for 3 clock pulses it becomes low, in this way, it will be, if the T 0, will be of that type.

Now, T 1, T 1 is only in the second clock pulse, say in the, this is a 1, that means, T 0 becomes high, and just the T 0, at the trailing edge of the T 0, T 1 becomes high, and it is only for 1 clock pulse, of the master clock pulse, and then again it become low.

Similarly, at the trailing edge of T 1, T 2 becomes high, and it will be high, for 1 clock pulse of master clock, then again it becomes low. T 3 is similarly, at the trailing edge of, T 2 it becomes high, and again it becomes low, after 1 clock pulse of the master clock. So, in this way T 0, T 1, T 2, T 3 can be generated, so this is, how from the, 1 master clock, the other time signals, timing signals can be generated, in the control unit design part.

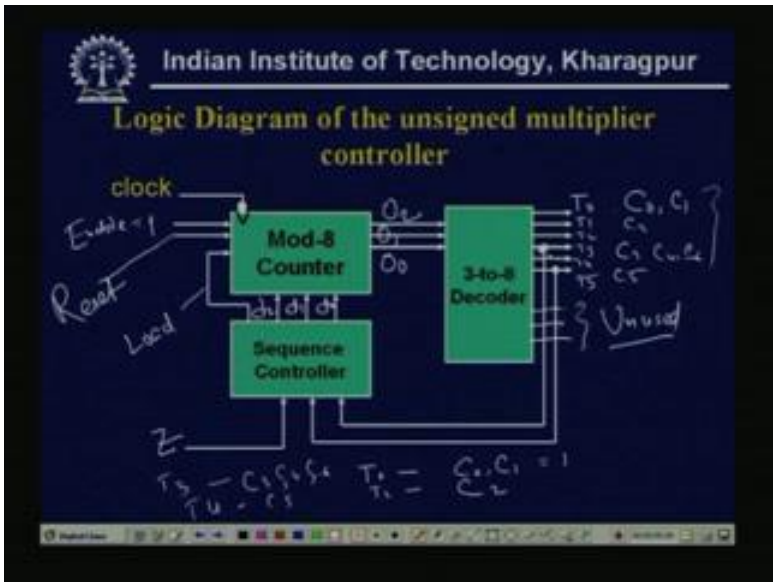
(Refer Slide Time: 38:22)



Now, the logic circuit of the controller the step 8, what will be the logic circuit of the, controller. So, first we see, the, the what will be the, so for we have discussed, the starting from the flowchart, then algorithm, then the hardware components identification, then the state diagram, and from the state diagram, that what will be the, timing signals can be generated. And, now we see, we can we have to combine the whole thing, and the, how what will be the logic circuits of the controller.

Now, the key element of the implementation is the sequence controller hardware, which sequences the controller, according to the state diagram, already the state diagram, we have drawn. We know that, what output should be generated, that according, then at that, clock what control signals will be, when what control signals will be high, at what time state, that also we have derived. Now, the all the things, will be all the steps, so for we have discussed, that must be combine in this step 8, to get the actual logic circuit of the controller.

(Refer Slide Time: 39:45)



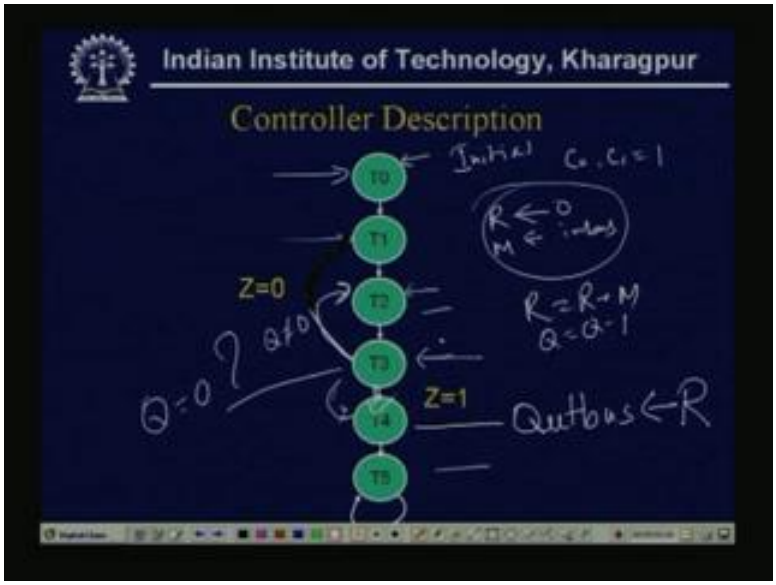
Now, see this is the key thing is that, the key element of the implementation is the sequence controller. So, the controller, so it is the, first is the one mod 8 counter, one sequence controller, one, 3 to 8 decoder, this is the whole thing of the logic circuit, so if we complete the whole design, so mod 8 counter, it has we have already seen that, enable clear unload. Say this is, this is enable when this line is high, then it is a reset line, it is a reset, this is a load line, so these are enable, this is a clear this is a load.

And the three outputs, say the three outputs are O 0, O 1, O 2, these goes to a 3 to 8 decoder, and the other inputs from the sequence controller, one is L, one is, and the other three are, say 3 d 2, d 1, d 0 and, 3 to 8 decoder, so this 8 from here. We have seen there, are 5 states, so these are actually, T 0, T 1, T 2, T 3, T 4 and T 5, and the rest three are actually unused, these three lines are unused. Now, already we have seen, we have derived, that at T 0 the control signals, should be high their C 0 and C 1.

So, if we write clearly, say actually at T 0 state, C 0 and C 1 control signals should be high, then at T 2, C 2 is high, similarly we have seen T 3 C 3, C 4 and C 6 and T 4, there is C 5. These we have seen already ((Refer Time: 42:51)), T 0, T 1, T 2, T 3, T 4, T 5 that, this control states, this control signals should be activated, so signal so C 0, C 1, C 2 then C 3, C 4 C 6 then C 5, and at T 5, it will be, this C 6 at T 4, and they, so this, this control signal should be high.

And, these are, this is my another Z input, and this should be, that these two inputs, are coming from the T 3 line, and the T 0, T 1, T 2, T 3, T 4 T 5 lines, so T 3 line and T 5 line, these will be these two are added, so these are the logic diagram of the unsigned multiplier. Now, the logic involved, in deriving the entries of the, the sequence control truth table, the mod eight counter is loaded or initialized, with the specified external data, just now we have seen, that the enable clear and load line. Now, the input L provides the counter enable control input E, already we have seen, now from the controllers state diagram, if we see again ((Refer Time: 45:09)).

(Refer Slide Time: 45:09)



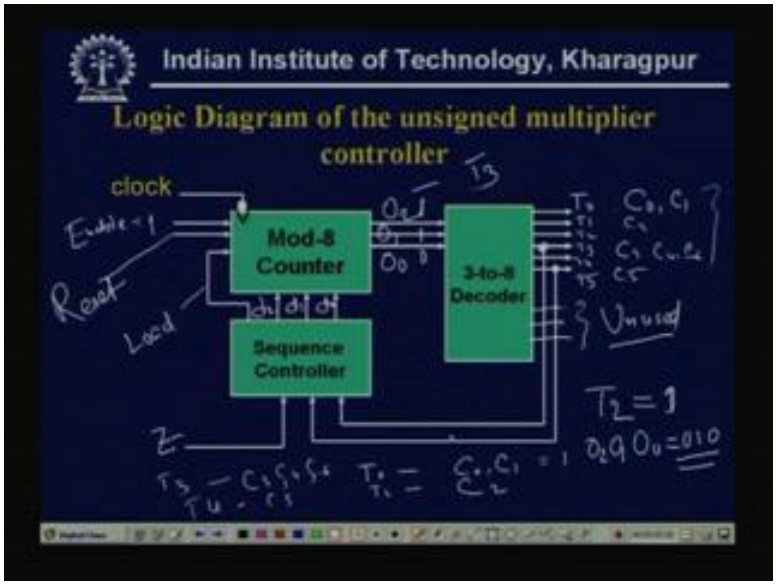
The controller state diagram, the controller counts up automatically, in response to the next clock pulse, every clock pulse it is in the next state. When the counter, load control input L equal to 0, because the enable input E is at, to high, such normal, non sequencing, activity is desirable. Say, the present control state is in T 0, then the present control state T 3, and Z equal to 1, the next state will be, say if the, if the present control state is T 0, T 0 then in the next clock pulse, it will be in T 1, no operations assigned are defined at this state will be performed.

Then, in the next again clock pulse will change, the next clock pulse it will be in T 2, operations will be performed at this state, define in this state, then it will be in T 3, and parallelly, the conditions to be checked that means, Z equal to 0 or Z equal to 1. If, that T 3 and Z equal to 1, present control state is T 3 and Z equal to 1, then only in the next

clock pulse, it will be in, it will be in T 4, this state. And, if the present control state is T 3, and the condition Z equal to 0, then the next state is T 2, instead of T 4.

So, these are my control signals, that means, present state and the input. So, here the input is, the conditions, the signal generated after the condition, and the machine will change.

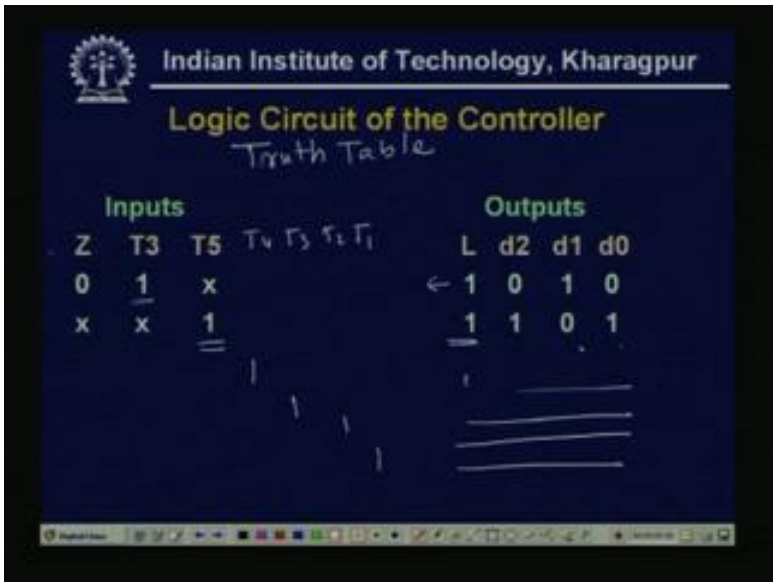
(Refer Slide Time: 47:17)



So from the controller state diagram, say, if the present control state is T 3, the counter output will be, that O 0, 1, O 2, so these are the O 0, 1, O 2 these will be 0 1 1, now in the state is, say when the state is T 3. Now, and if Z equal to 0, the next state is T 2, when these condition occur, the counter must be loaded, with external value 0 1 0, at the trailing edge of the next clock pulse. When T 2 equal to 1, only when O 2, O 1, O 0, is 0 1 0, that means, T 2 is 1, T 2 is 1, only when O 2, O 1, O 0 these are 0 1 0, it will take.

Therefore, the SC generates L equal to 1, the load value equal 1, and O 2 ,1, O 0 equal to 0 1 0 it has to generate. Now, similarly, from the controller state diagram, if the present state is T 5, if we again if we see, if it is, T 5 the next control state is also T 5, the SC, the sequence controller must generate the output L equal to 1, and d 2, d 1, d 0 is 1 0 1. So, the SC truth table, shows these out of sequence counts, for each show of the SC truth table, shows that what, output value, should be generated by the SC controller. And accordingly, the, the design should be like that, see the, if we see, see the logic circuits.

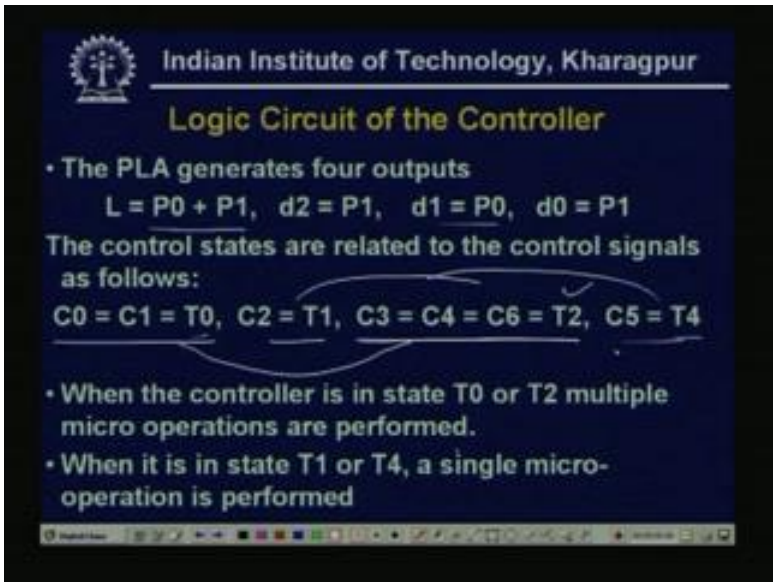
(Refer Slide Time: 50:04)



Now, first we see the, the truth table the see would, see the truth table, truth table of the logic circuit, this is the truth table. Say, this is Z, T 3 T 5 and outputs are L, d 2, d 1, d 0, now, Z is 0, say if T 3 equal to 1, and T 5 we do not know, T 3 we are seeing, then actually, output should be L equal to 1, and d 2, d 1, d 0 is 0 1 0, just now we have seen. Now, if T 5 equal to 1, then it should be, in the same state, so again L is equal to 1, and d 2, d 1, d 0 is 1 0 1.

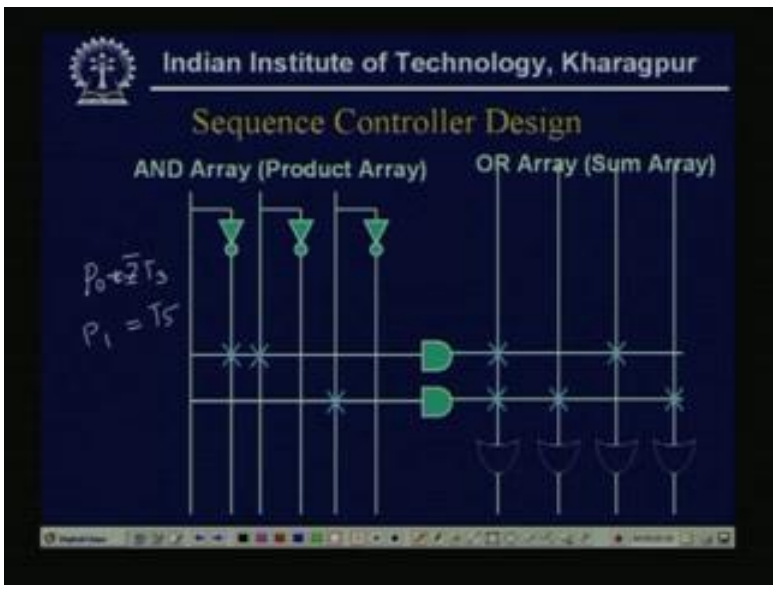
See, in this way, the output, input output, values that means the, truth table will be completed it, it can be completed, with the all other steps, that means, when T what will be the, T 3 T 5, then what will be T 4, T 3, T 2, T 1. And, when every 1 will be 1, say T 4 will be 1, then what output value, will their here, similarly, when T 3 equal to 1, what output value will be generated here. T 2 is 1, and T 1 is 1, then we have to what output values, so these are my, that if these are the inputs, then what output will be generated by my SC controller.

(Refer Slide Time: 52:20)



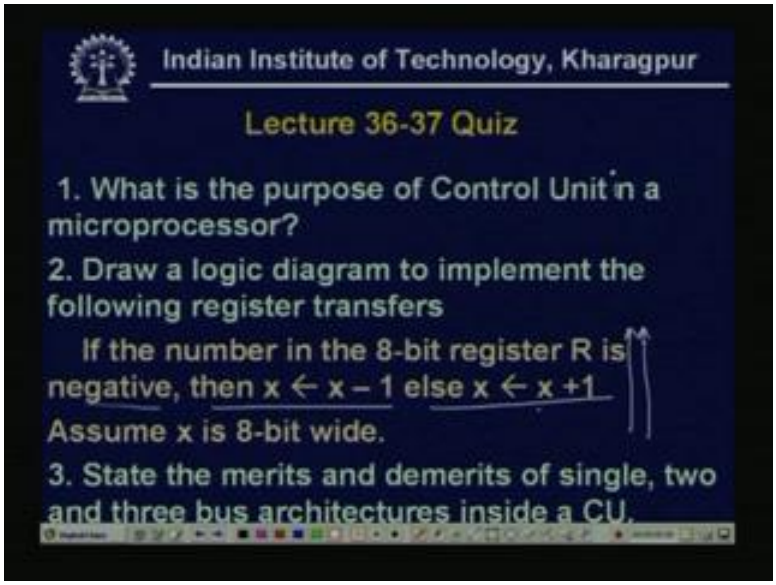
Now, if we, see the logic circuits, if it is a PLA design, then actually the L can be P 0 plus P 1, d 2 equal to P 1, d 2 the four outputs, L d 2, L d 2, d 1, d 0, so L is P 0 plus P 1, d 2 is P 1, d 1 is P 0, d 0. And the control states are related to the control signals, as follows, say C 0 equal to C 1 is to 0, C 2 is T 1, C 3 equal to equal C 4 equal to C 6 is T 2, because if T 2 is high, these three signals should be high, at T 4 states. If C 5 is high, so when the controller is in state T 0 or T 2, multiple micro operations are performed, this T 0 and T 2, T 0 and T 2 these two. Now, when it is in state T 1 or T 4 a, single micro operations will be performed, that means, at T 1 and T 4.

(Refer Slide Time: 53:26)



So, if it is a PLA design, then this type of AND array, and OR array circuits will be there. And the logic, we can write, that of the product term is generated in the PLA designs as P 0 plus Z bar T 3, and P 1 equal to T 5. P 0 equal to Z bar T 3 and this is P equal to T 5, and then, this the output signals will be generated. So, and we have seen, that in this way, we can design the control circuits.

(Refer Slide Time: 54:11)



Now, these last two classes the, we have mainly discussed about the control unit design, so the quiz on lecture 36, 37 is a first is, what is the purpose of control unit, in a micro processor. And, draw a logic diagram two implement the following registers transfers, if the number in the 8 bit register R is negative, then X is X minus 1, else X is plus 1, assume X is 8 bit wide. So these register transfer, what will be the logic, and states the merits, and demerits of single, 2 and 3 bus architecture, inside a control unit.

So, this is actually, the part of the register transfers, 1 steps of the ALU, that a logic diagram to implement the following register transfers. After the flowchart derivations, the step 2, step 2 was the obtain a registers transfer description algorithm from the flowchart.

So, this, as if step 2 is given, X is X minus 1, and X is X else X else X plus 1, if R is negative, then X is X minus 1. If R is positive this is, then what will the hardware for this, the logic diagram for this, now this control unit for this type of logic.

So, this is simple very primitive control unit design, and that is the quiz question of lecture, 36 and 37. Actually, we have, discuss the multiplier, 4 by 4 multiplier, instead of that, we want to design this CU. So, this is the end of this class, next day we will discuss the other thing.

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

Lecture - 38
Design of Computer Instruction Set and the CPU (Contd.)

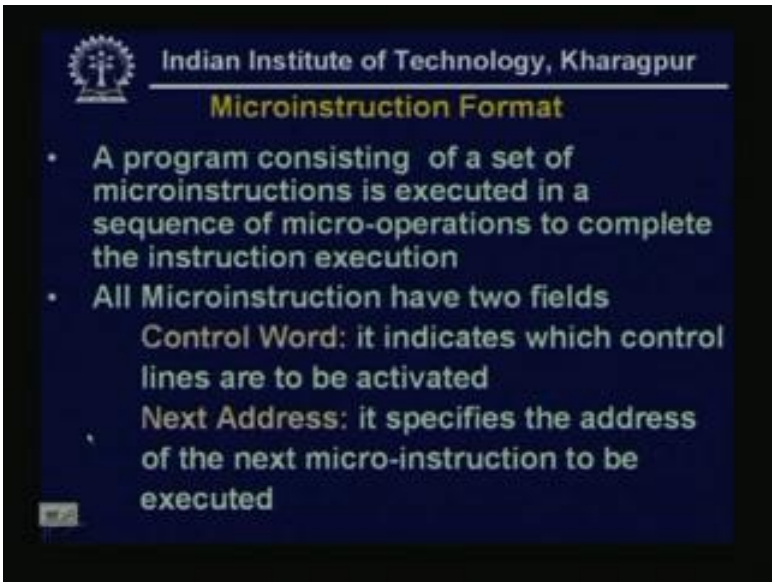
We are discussing the design of CPU, and in last class we have, read the design of one type of control unit, call the hardware control unit. Today, will read the other one, the micro programmed control unit.

(Refer Slide Time: 57:01)

So, first we see, what do we mean by, micro programmed control unit, and then, we will read the design of, that type, particular type of control unit, Now, micro programmed control unit, contains the programs, this program is nothing but, a set of micro instructions. Now, programs are stored in a control memory, normally in a ROM inside the CPU, now, to execute instructions, the microprocessor, first fetch each instruction into instruction registers, from external memory.

Then, the control unit translates, the instruction, each control word contains signals, to activate one or more micro instruction. After, translating this instruction, then the control unit decides or activates the control signal, that what particular operations to be executed.

(Refer Slide Time: 58:14)



Now, a program, consisting of a set of microinstructions is executed in a sequence of micro operations, to complete the instruction execution. As this, micro program is nothing but, a set of micro instructions, so it is very much important, that what will be the format of this micro instruction. So, that the sequence of micro operations or micro instructions to be read, to control the micro operations. Now, all micro instructions have two fields, one is control word, and other is next address, now this control word.