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.)

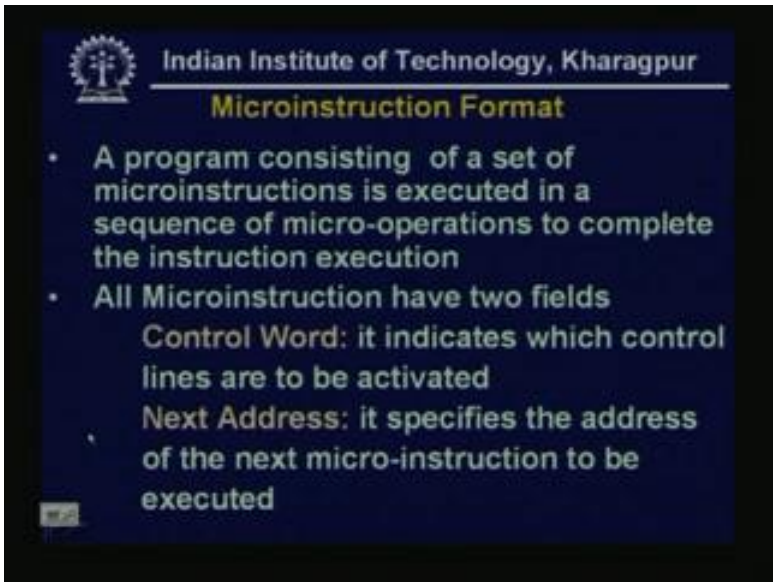
We are discussing the design of CPU and in the last class, we have read the design of one type of control unit, call the hardware control unit. Today, we will read the, other one the micro programmed control unit.

(Refer Slide Time: 01:06)

So, first we see, what 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, in to a instruction registers, from external memory.

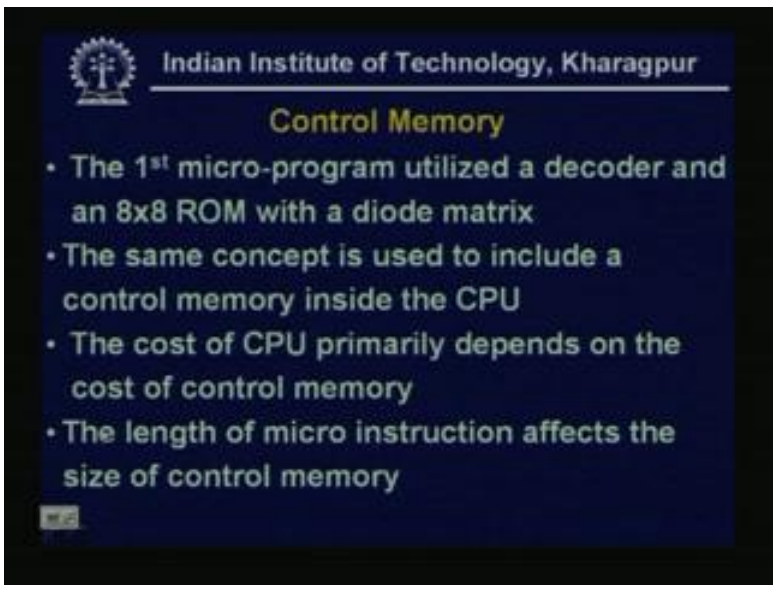
Then, the control unit translates the instruction, each control word, contains signals to activate one or more micro instruction. After translating this instruction, then the control unit decides, or activates the control signal, that what particular operations, to be executed.

(Refer Slide Time: 02:18)



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 microinstructions, so it is very much important, that what will be the format of this, microinstruction. So, that the sequence of micro-operations or microinstructions to be read, to control the micro-operations, now all microinstructions have to fields, one is control word, and other is next address. Now, this control word indicates, which control lines are to be activated, and next address is specifies, the address of the next microinstruction, to be executed.

(Refer Slide Time: 03:30)

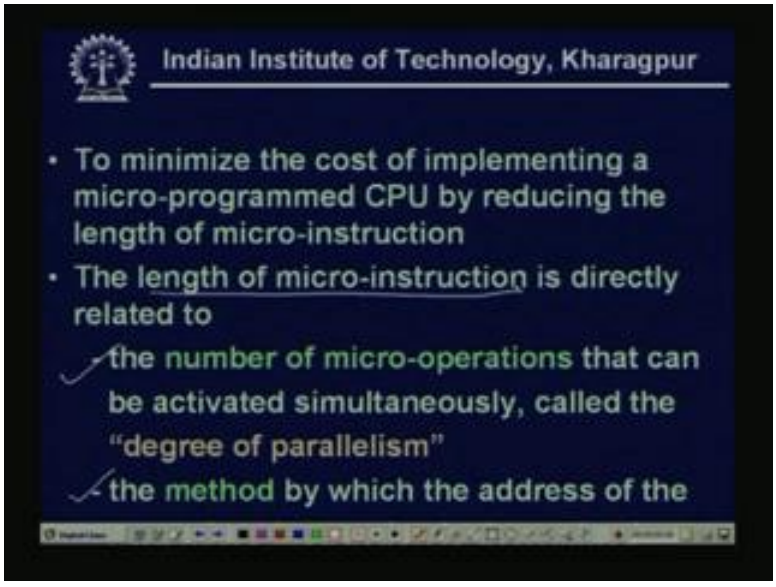


Now, this, the memory, which contains this program, micro-program, we call this is control memory. Now, we see, what will be the features, of this control memory or how actually the control memory is being accessed, or how it is written, written data on to it, now if the concept of micro-programming was first proposed by, W V Wilkes in 1951. So, when first it was introduced, the micro-program, utilized a decoder, and an 8 by 8 ROM with a diode matrix.

So, that was the first implementation, of a control memory, the 2 components are a decoder, one decoder, and a, and an 8 by 8 ROM with a diode matrix. Now, this same concept is extended, to include a control memory inside the CPU. So, now, the control memory stays inside the CPU, so the cost of CPU, primarily depends on the cost of the control memory. Now, the length of a micro-instruction, affects the size of the control memory.

Therefore, a major design a op-code is to minimize the cost of implementing a micro-program CPU, by reducing the length micro-instruction, as the length of micro-instruction affects the size of control memory. And the control memory, cost control memory, decides are cost of CPU, so first op-code, is to reduce the length of the micro-instruction. So, it will reduce the cost of memory, and finally, it results the reduction of the cost of CPU.

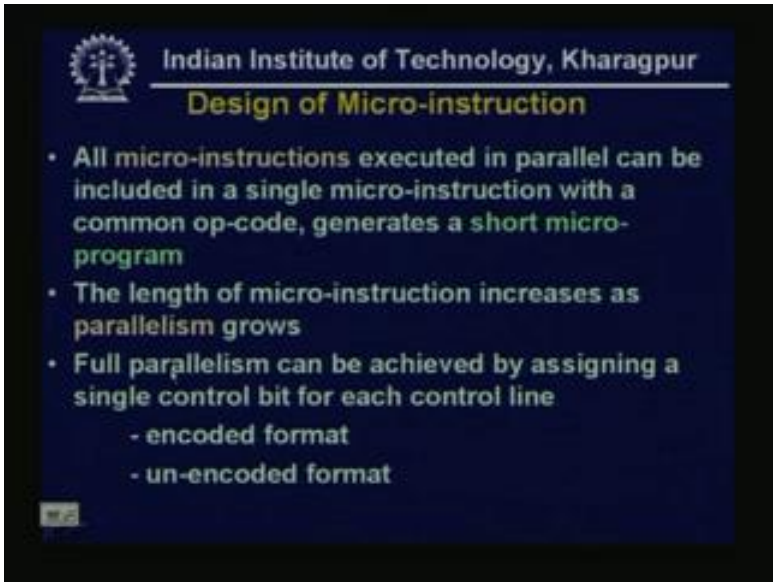
(Refer Slide Time: 05:49)



Now, to minimize the cost of implementing a micro-programmed CPU, reducing the length of micro-instruction is needed. Now, the length of micro-instruction, is directly related, to the number of micro-operations that can be activated simultaneously, call the degree of parallelism. As the program is a set of instruction or micro-operations, so in every program, there are some, a, some of the micro-operations can be performed, simultaneously.

And, as it is perform simultaneously, we call that these are some, parallel of micro-operations, this is some degree of parallelism. So, the length of micro-instruction, directly related to this degree of parallelism, and another thing is the method, by which the address of the next micro-instruction, is determined. So, mainly these two features, the number of micro-operations, and method by which the address is decoded, these two determines the, the length of micro-instruction.

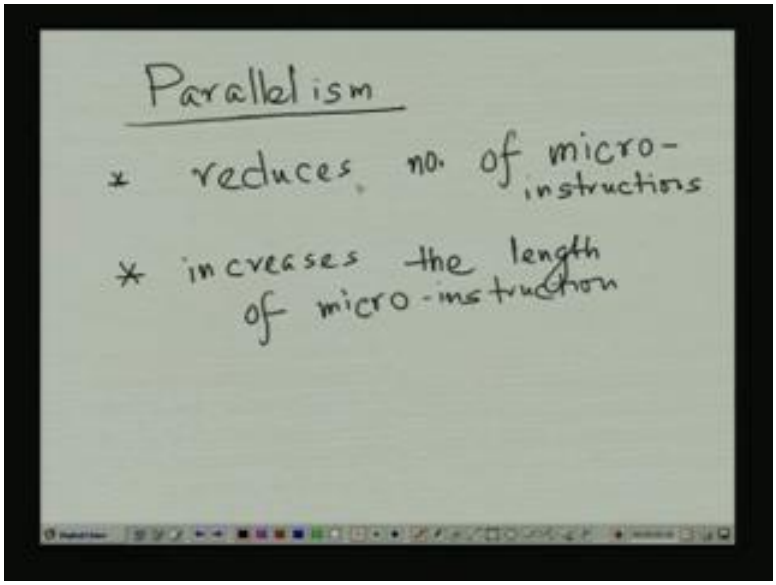
(Refer Slide Time: 07:11)



So, we see, what will be the design of micro-instruction first, because this is the primitive of the micro-program, set of micro-instruction, is the micro-program. Now, all micro-instructions, executed in parallel, can be included in a single micro instruction, this is a very important thing, that means, a parallelism is exploited, to reduce the number of micro instruction. So, the micro instructions, which executed in parallel, they are included in single micro instructions, with a common op-code, and this results a short

micro-program, this results a short micro program. But, the length micro-instruction increases, as the parallelism grows, so number decreases, but the length increases.

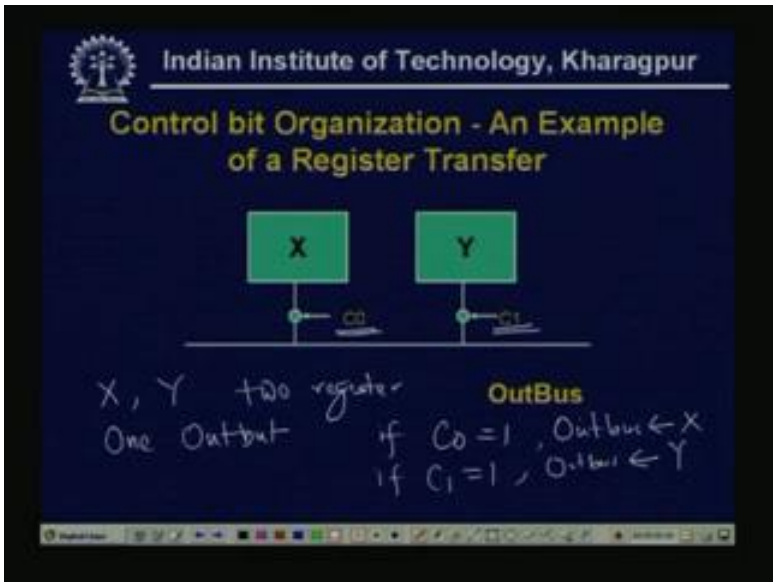
(Refer Slide Time: 08:24)



So, here, what we can tell, that for, the affect of parallelism is, it reduces the number of micro instructions, the all the micro instructions, that are executed parallelly, they can be combined, to as common op-code, and that means, it is one single micro-instruction. Obviously, that is much shorter, if we consider, the all the, micro instructions that are combined. But, parallelly, it increases the, the length obviously as, a number of operations are combined. So though it, now it gets the common op code, but the length micro-instruction is increased.

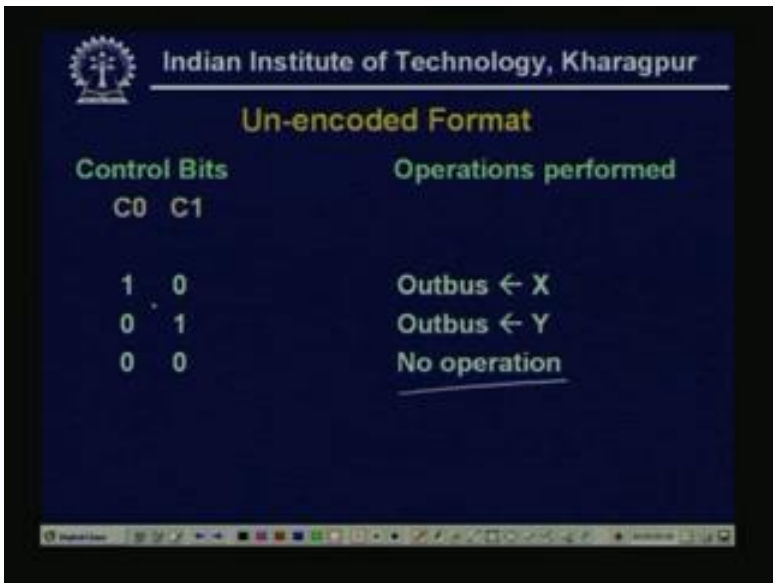
So, we need a, optimization, that what degree of parallelism, can be considered, so that, this it, the reduction of micro-instruction or the increase of length of micro-instruction, can be optimized ((Refer Time: 10:14)). Now, full parallelism can be achieved, by assigning a single control bit, for each control line, the control bits in a micro-instruction, can be organized in several ways; and one obvious way is to assign a single bit for, each control bit line. And these will provide, full parallelism, so no decoding of the control bit, is necessary. Normally these organization is a two type, one is call the encoded format, another is un-encoded format.

(Refer Slide Time: 11:06)



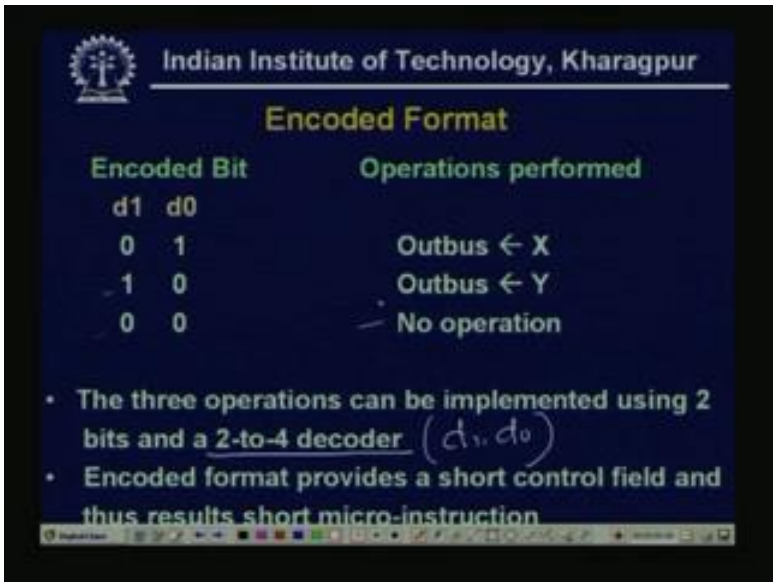
Now, we consider one example, of a register transfer, and we see, that how this control bit organization, can be achieved. See, here, there are 2 resistors, the 2 registers X and Y, X Y 2 registers, and 1 out bus, 1 out bus, now C 0 and C 1, C 0 and C 1 are the control bits. So, if C 0 equal to 1, X is, or out bus is X, out bus, out bus gets the content of register X, if C 1 equal to 1. Then out bus gets the contents of register Y, so this is the, register transfer. Now, here each operation can be performed, one at time, because there is only one out bus, so a single bit can be assigned, to perform, each transfer in this way.

(Refer Slide Time: 12:47)



See, first we consider the un-encoded format, un-encoded format, so this is the normal see a simple situation, that when each control bit is 1, 1 operation is performed. So, C 0is 1, C 0 is 1, X is assign to out bus of the content of this register, goes to the out bus, if C 1 is 1, content of Y register, goes to the out bus. If all are 0, C 0, C 1 is 0 their will be no operation. So, these three operations can be d1, by specifying the control bits C 0, C 1 as 1 0, 01, and 00.

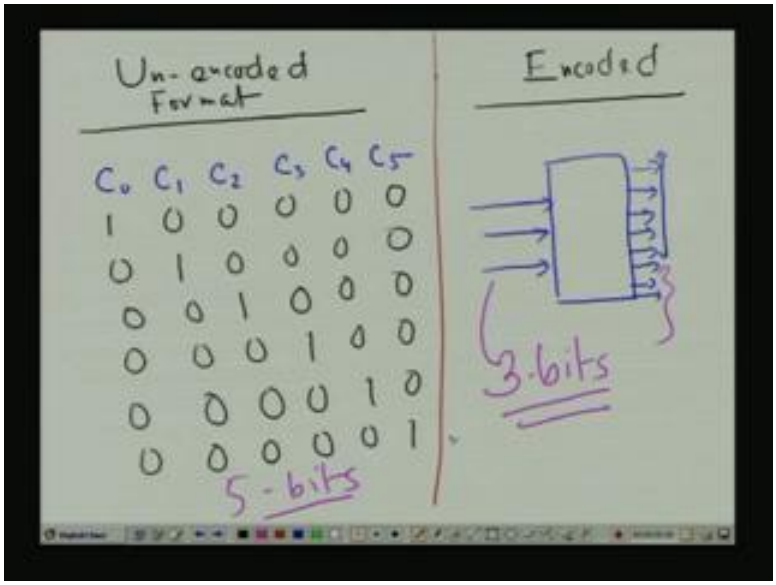
(Refer Slide Time: 13:44)



Now, the same thing can be d1, in different way, in a, encoded format, see as it is encoded format, so here, the relationship are like that, say this, these are some bits, this d 1, d 0. See, if we consider a, these 3 operation, we implement using 2 bits, and a 2 to 4 decoder, so as if d 1, d 0 of the inputs of this decoder. Now if d 1, d 0 is 0 1, X contents of X goes to out bus, if it is 1 0, contents of Y goes to out bus, if it is 0 0, then there is no operation

So, an encoded format, these are the control bits, and now we are these are the inputs of a decoder, so here, this encoded format can be implemented, using 2 bits and 2 to 4 decoder. So, encoded format provides, a short control field, why see, if we have, some five control lines, then we need a 3 to 8 decoder.

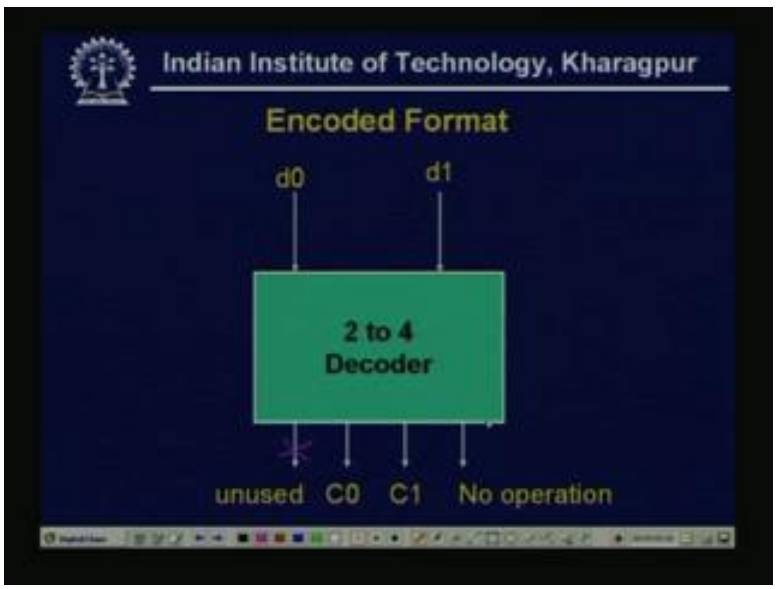
(Refer Slide Time: 15:26)



Five control lines, then for un-encoded format, say for un-encoded format, we need five bits, C 0, C 1, C 2, C 3, C 4, C 5, what we do, each time we this is 1 0 0 0 0 0, similarly this is 0 1 0 0 0 0, that one particular control bit is assigned one, this is un-encoded format. Now, if it is encoded 1, we use a decoder, so here, if we for five outputs, we need a 3 to 8 decoder, so we need a, 3 to 8, 3 to 8 decoder, here only five lines will be utilized, this five lines will be utilized, and this three will be unutilized.

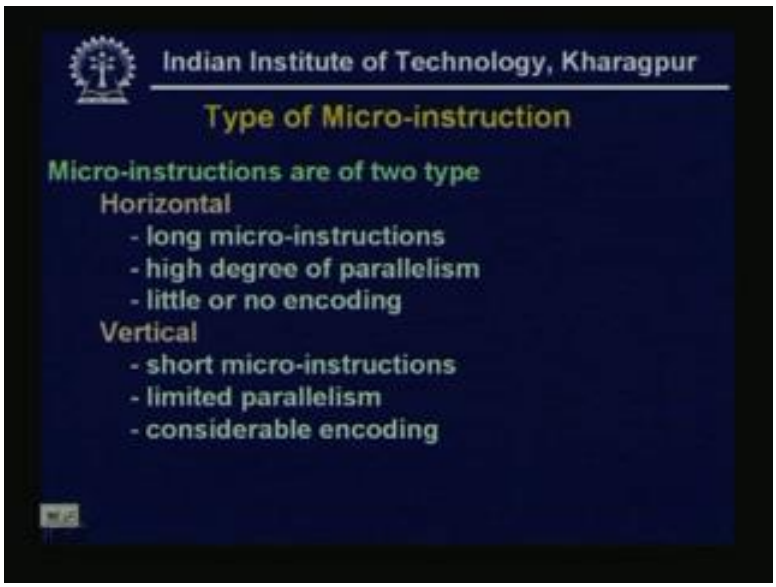
So, here, only 3 bits are necessary, 3 bits are necessary, here it is 5 bits. So, in this way ((Refer Time: 17:18)), encoded format, provides a short control field, on thus results, short micro instruction.

(Refer Slide Time: 17:28)



So, the implementation of encoded format is like that, only d 0, d 1 will be the input of decoder, and the outputs will be that, two will be the actual control signals, like in this example C 0, C 1, 1 is no operation. Actually, there are operations, if we include the no operation, itself is NO is one operation, and one line is unused, so this unused, so this, the implementation of encoded format, using decoder.

(Refer Slide Time: 18:07)



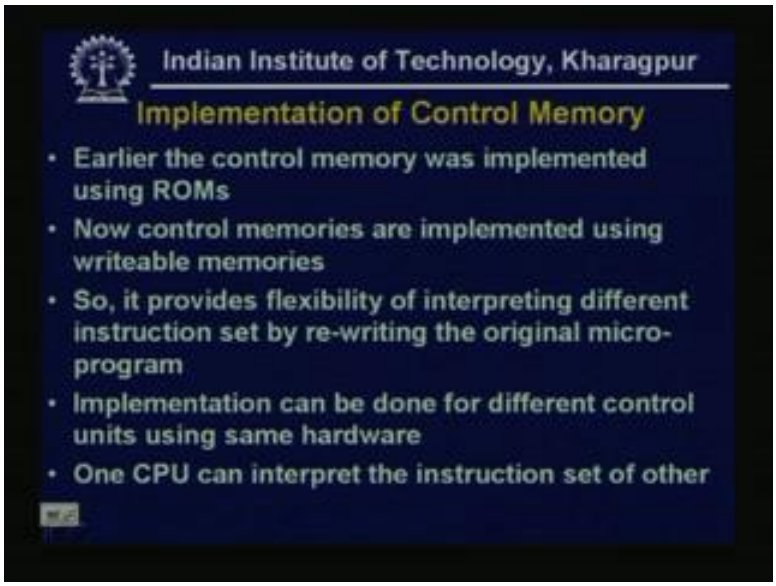
Now, we see that the different type of micro-instructions, so micro-instructions are of two types, one is called horizontal and another is vertical. Now, the horizontal micro-instruction mechanism provides, long micro-instruction, a high degree of parallelism and little or no encoding. Whereas, the vertical micro instruction it, offers short micro-instruction, it as limited parallelism, but it has considerable encoding.

(Refer Slide Time: 18:55)

Now, we see the microprogramming, so microprogramming is the technique, of writing micro-programs, in a micro-programmed control unit. Writing micro-program a, similar to writing, assembly language programming, micro programs are basically written in a symbolic language, called micro assembly language. Now, these programs are translated, by micro-assembled to generate micro-codes, which are then stored in the control memory.

So, just like our concept of software, again these are also the instructions are nothing but, software, so this micro-instructions are written, in some language called the micro-assembly language, and they are translated by assembler. And, then and actually, this programs are stored or these micro-codes, are stored in control memory.

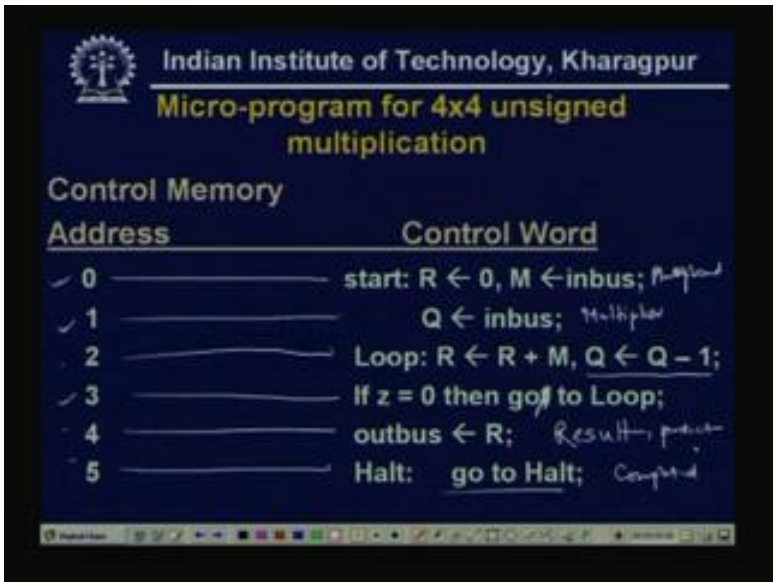
(Refer Slide Time: 20:16)



Now, how they are implemented, how this control memory is implemented, in earlier early days, the control memory was implemented using read only memories ROMS. So, it cannot be, changed, so now a days, people have change to the, type of memories, so that, it can be more flexible or the program can be changed easily. So, now the control memories are realized, by using some, writeable memories, not the read only memories, so write is also possible.

So, it provides the flexibility of interpreting different instruction set, by rewriting the original micro-program. And this allows implementation of different control unit, using same hardware, now using this, approach one CPU can interpret the instruction set of other CPU.

(Refer Slide Time: 21:37)

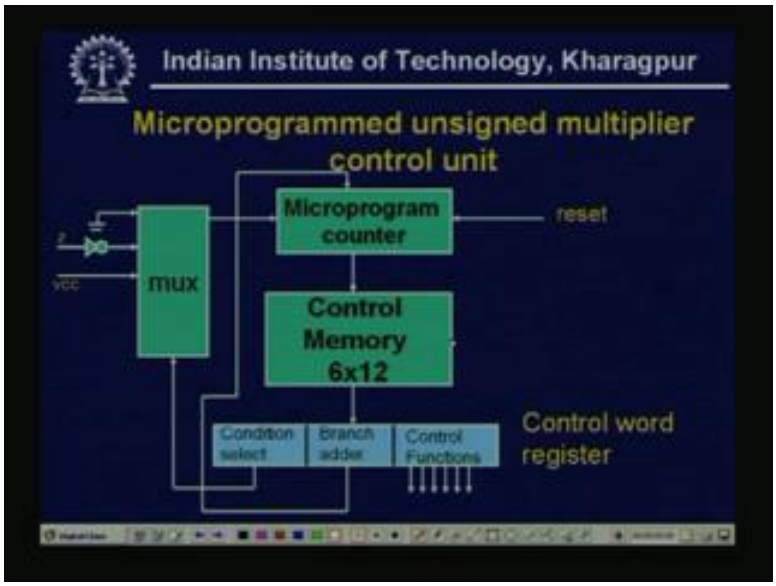


Now, the design of a programmed control unit, we consider, and the example we have taken, is the same example of that, hardware control unit, we have consider a 4 by 4, unsigned multiplication. So, this is 4 by 4 unsigned multiplication, the technique is repetitive addition, so here the, as it is a memory, so there will some address, and some content. So, the control memory address, say we give the control memory address are 0, 1, 2, 3, 4, 5 as there are six micro instructions.

And, what will be the control words or the content of the memory, so at, 0‘th address, it will be 0’th address, that register R is resets to 0, and the register M gates the in bus, say the multiplicand. And see, these are two independent operations, and they can be combined, to get a parallelism, so this is, these are in 0’th address. Now address 1, address 1 the register Q gets the in bus, say the multiplier, so Q gets the value of multiplier, where as M is multiplicand.

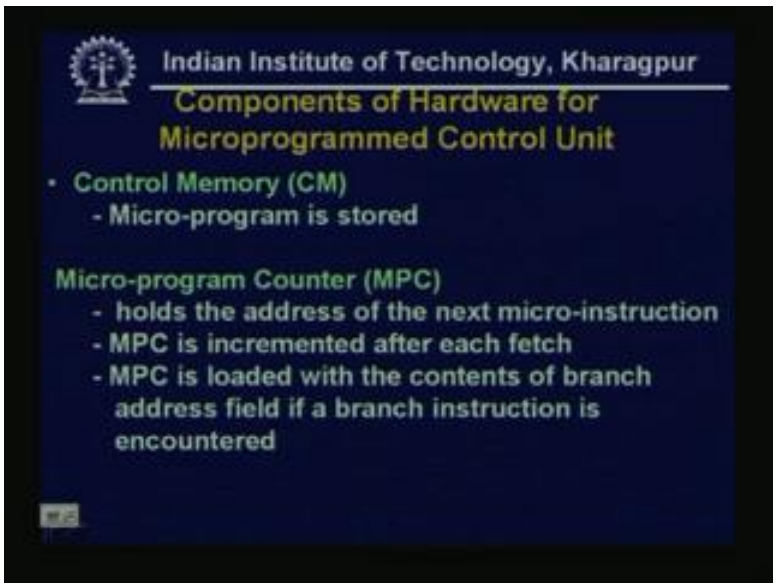
Now, at just to the operations, the register R is R plus M, as it is a repeated addition, so we have to check, that how many additions are there, because I want to Q number of additions, because Q is my multiplier. So, every time, after each addition, the multiplier is decremented by 1, that is Q is Q minus 1, at address 3, if z equal to 0, then go to loop, address 4, if out bus, the register goes to out bus. That means, this is the result the product, and then address 5, the halt go to halt means, it is the program is completed, so these are the control words, store in the control memory.

(Refer Slide Time: 24:35)



So, the unsigned 4 by 4 multiplication, uses is repeated addition, and the results is assume to be 4 bit what, now to implement is micro program, the hardware organization of the control unit, we can think like that. See, the what, are the various components of these micro-program control unit, of the multi micro-program multiplier control unit, unsigned multiplier. See, here, it has a one multiplexer, one micro program counter, the control memory, and the control word register, so we see the different components, we see that, different components.

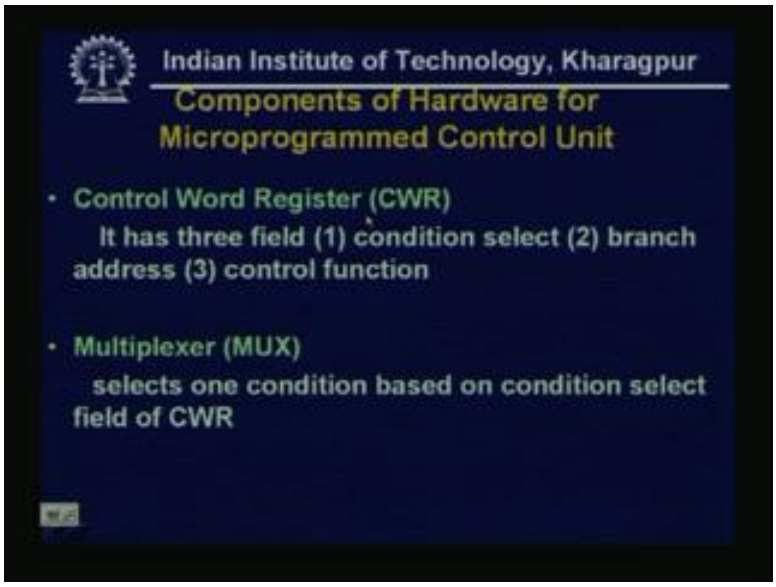
(Refer Slide Time: 25:55)



First is control memory, this is the main component of the design, and it stores, the micro it contains the micro-program, so the micro-program is stored here. Now, next one is the micro-program counter, it is called MPC, it holds the address of the next micro-instruction to be executed. Now, MPC is initially loaded, from an external source to point to the starting address of the micro program, the MPC similar to program counter, so MPC is implemented after each fetch.

If a branch instruction is encountered, the MPC is loaded, with the, contains of the branch address, field of the micro instruction. So, this is the function of a micro-program counter, in short we can think, that MPC is just like PC, the program counter of the CPU.

(Refer Slide Time: 27:29)



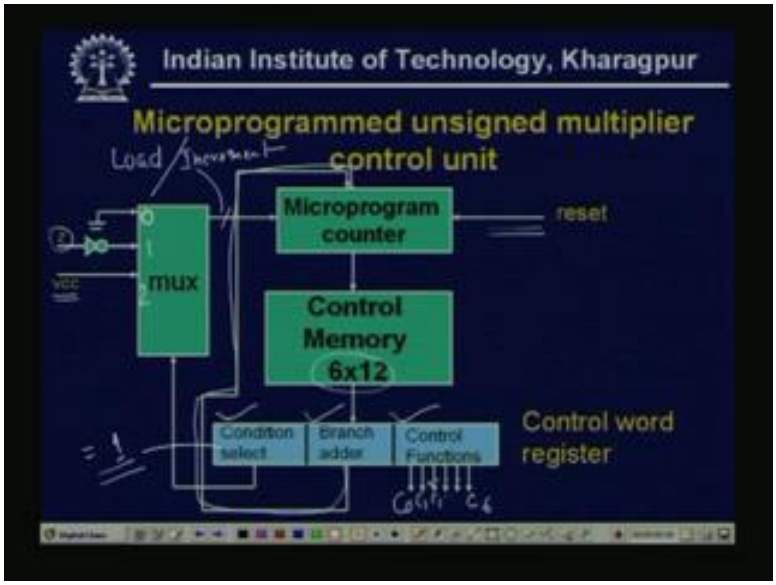
Now, the control word register or CWR, so control word register, each control word in the control memory, is assume to contain three fields, one is condition select, branch address and control function. Each micro-instruction, fetched from control memory is loaded in to CWR, the control word register, now the organization of the CWR is same. For each control word, and contain, the three fields, these three fields condition select branch address and control function.

And, in the case of a conditional branch micro-instruction, if the conditions specified, by the condition select field is true, the MPC is loaded with the branch address field of the CWR. Otherwise, the MPC is implemented, to point to the next micro instruction, so if the condition is satisfied, then this register, will be loaded to the content of the branch

address, otherwise it will be MPC is implemented just like PC, two point to the next micro-instruction, to be executed.

Now, the control function field contains the control signals, another is a multiplexer, which selects one condition. Based on condition select field of CWR, so input of multiplexer is the, the condition select fields.

(Refer Slide Time: 29:40)

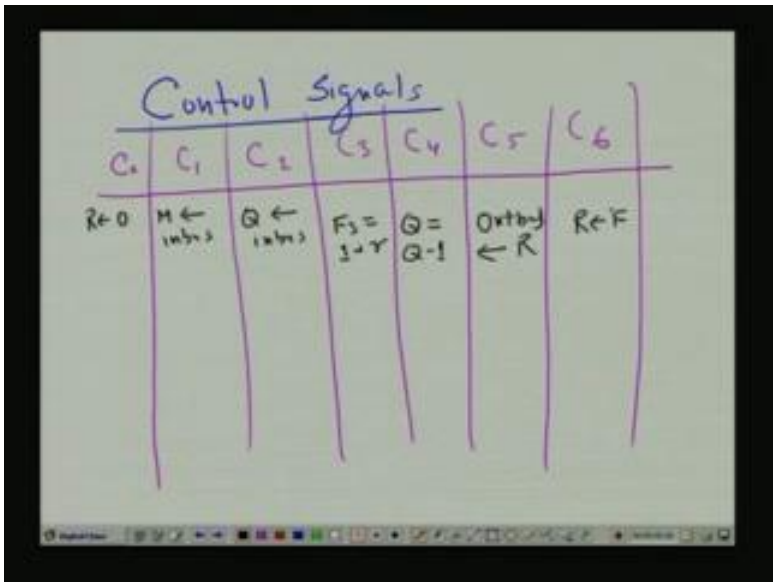


So, if we see, the design now, so first is multiplexer, so it has three line, the 0 ground line, 1, 2, this is actually the input Z, and this is some, this VCC is a 1 bit. So, this is a load, or load or increment, load or increment, according to the value, now micro program counter MPC, and it has some resent line. Because, always for the first time, it starts execution, this should be initialized that means, is should be reset. This is my control memory, one size is given here 6 in by 12, later we will see, that actually, the size of this control memory is determined.

Now, we are assuming one size, now the control word register, as already I mentioned, it has three fields, see this condition select, branch address and control functions. So, if the, conditions specified by the condition select field is true, so this condition select field, if this equal to this is 1. So, condition, specified by condition select field, if this equal to 1, then the content of this branch address, content of this branch address is loaded to the MPC.

So, this is the line, it goes to the MPC, otherwise MPC is incremented, by 1 just like PC, on the control function field, contains the control field, contains the control signals. So, here, how many control signals are there, we have seen that, control signal should be, say we defining C 0, C 1, C 2 up to C 6.

(Refer Slide Time: 33:10)

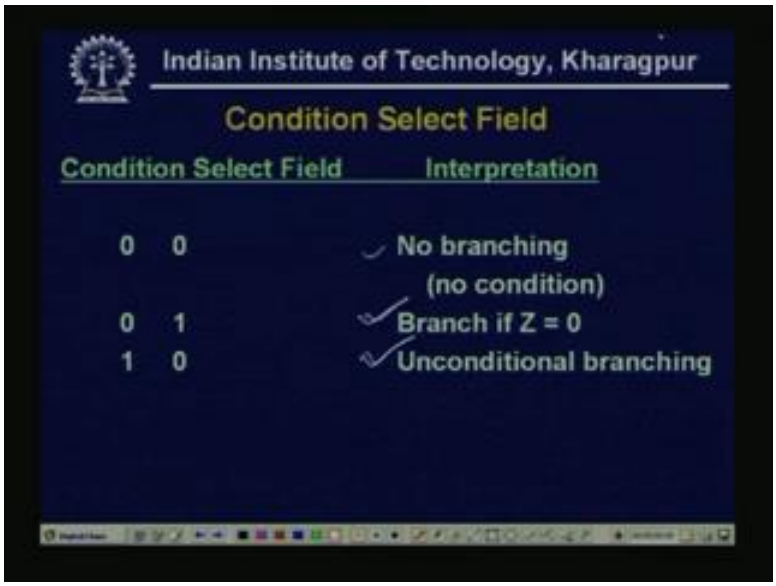


Now, see what are the, what are the control instruction, the control signals, so these are the seven control signals, and we can write, the control signal C 0 means, say the reset, C 1 is M is assign to multiplicand, C 2 is Q is a assign to multiplier. Then, the actual addition is in C 3, then it is a decrement, multiplier decrement, then the result goes to out bus that is activated by the control signal C 5. So, we can write out bus, out bus is assign to R, and then this is a completion, that R is F, so these are the seven control signals.

So, we need seven control signals, now what are control condition select field, because just now we have seen, that the control word registers have three field, condition select branch address and control functions. So, depending on the, number of micro instructions or number of micro-operations to be performed, the number of control signals to be determine. And that will directly, gives how many control signals will be generated, by the control functions.

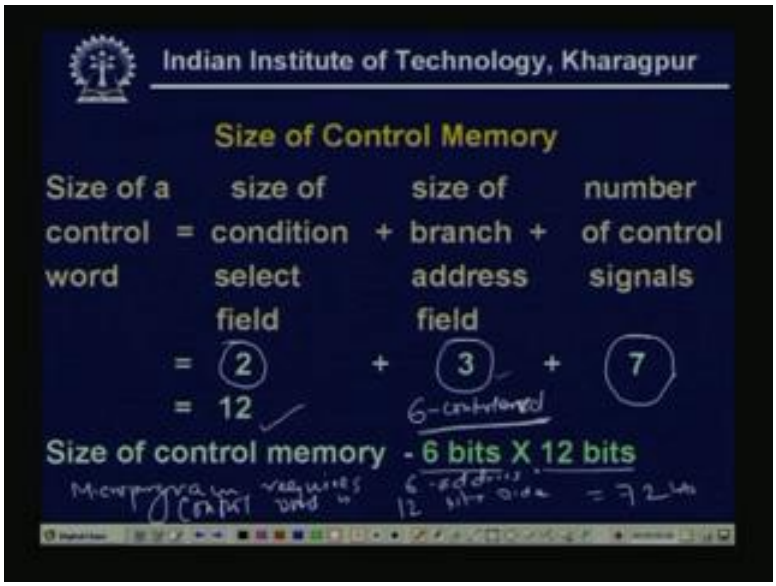
In this particular example, this see, C 7, C 0 to C 6 the seven control signals are generated, from the control functions. So, we have to design our control functions, control function field of CWR, the control word register accordingly.

(Refer Slide Time: 36:12min)



Now, we see the condition, select and branch address, so the first we see the condition. Select, if there is 0 0, condition select fields a two fields if we take, then there is no branching. That means, no condition. 0 1, branch if Z equal to 0, and 1 0 means this is, unconditional branching, so there are three situation one is 1 0 means, unconditional branching means no condition to be checked, so always it will be a branch. 0 1 means a condition is there, that means, branch if Z equal to 0, this type of condition to checked. 0 0 means, no branching is a sequential think, it will not jumped to, anywhere, so no, no branching is 0 0. So, as there are three, interpretation was three operations are necessary, so the condition select field, is sufficient to have 2 bits only.

(Refer Slide Time: 37:31)

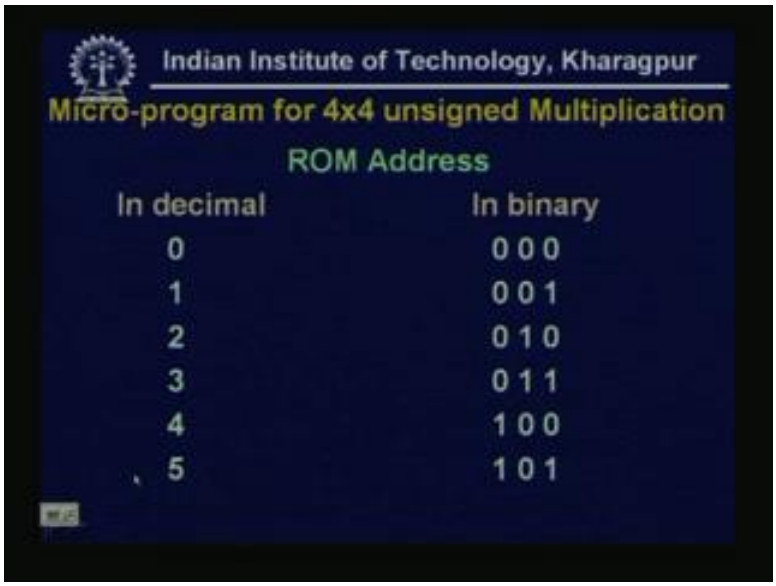


So, we see what will be the size of control memory, see size control word, should be size condition select field, size of branch address field, and number of control signals. So, just now we have seen, the number of control signals are, 7, and size of condition select field, size of condition select field is 2. Now, there are 6 control memory address see, just we have seen, there are 6 control memory address, 0 to 5 are required, for the control memory to store the micro-program.

Therefore a 3 bit address is necessary, for each micro-instruction, as there are 5 address, so in it 3 bit for encoding, this addresses in binary. So, if you take the binary value, size of branch addresses is 3, as there are 6 control word, as there are, there are 6 control word. So, for 6 memory address, control memory address is necessary, and for that, we need a 3 bit, so size of control word is the sum of these three fields. So, 2 the condition select, 3 the branch address, and 7 the number of control signals, so it will be 12.

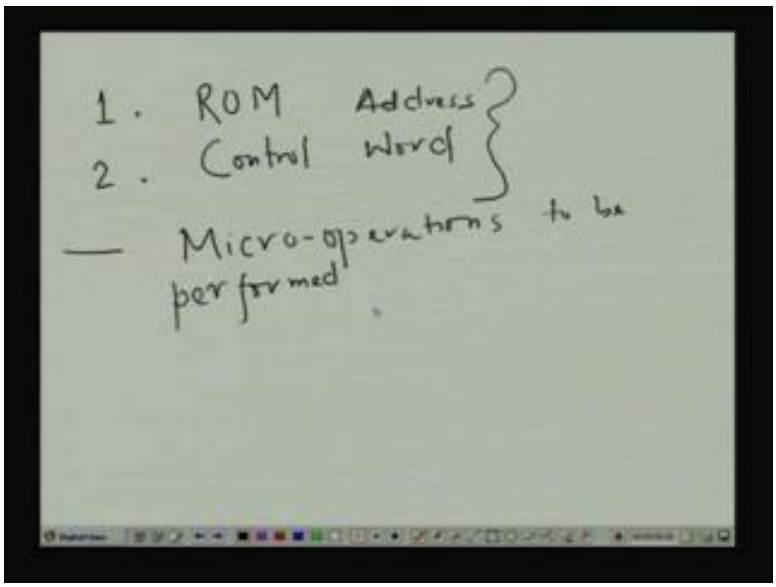
So, size of control memory is 6 bit by 12 bits, because the micro program requires 6 addresses, and the control word is 12 bits wide. So, the memory size will be, 6 by 12 bits or we can write this a 72 bits, so this way, the size of memory can be calculated, so we have the seen the size of control memory.

(Refer Slide Time: 41:14)



Now, we consider the micro-program the unsigned multiplication, the example we have considered,

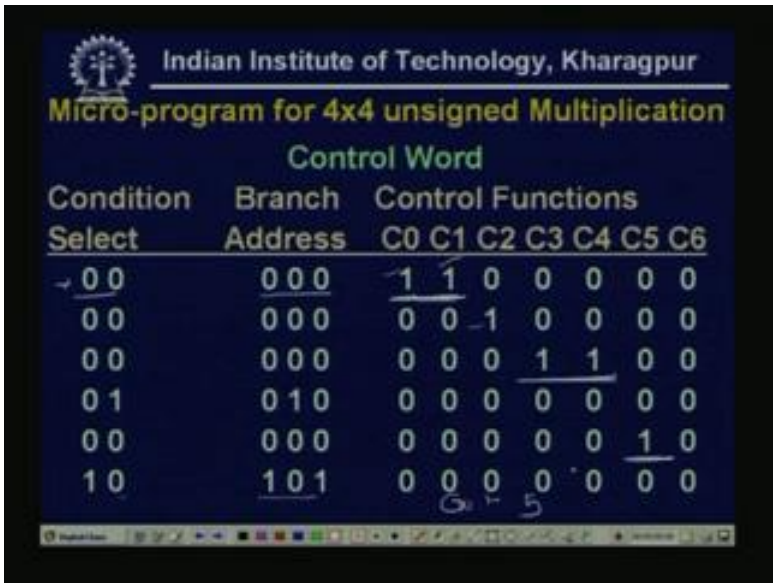
(Refer Slide Time: 41:41)



First we see the, actually we have to consider three things, what will be the, what will be the ROM address, Second is the control word, and for these two we will see, what are the micro operation, to be performed, and this gives me, the micro-program. Because, these are the set of micro instructions or micro programs, that generates the micro-program, so mainly, what will be the ROM address, and the control word, that we have to see ((Refer

Time: 42:38)). Now, ROM address, as there are 6 address, so in decimal 0 to 5, and binary 0 0 0 0 0 1, because we need, as there are 6 unit 3 bits to encoded, a simply 3, 3 bits are necessary for ROM.

(Refer Slide Time: 42:57)



Now, the control word, control word has three field condition select, branch address control function. Now, we see the condition select, condition select we have seen that, we need only 2 bits for condition select, branch address is 3 bits, and the control functions are C 0 2, C 6 means that they are seven control functions. So, if the condition select, condition select is 0 0, so condition select, we consider the first one, condition select is 0 0, then the branch address is 0 0 0.

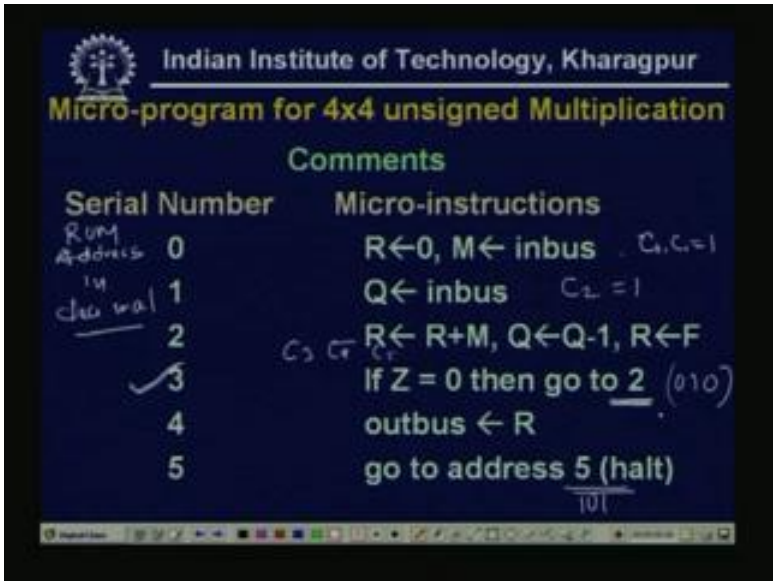
And, say control functions are, see both C 0 C 1, C 0 was a reset, and C 1 is the multiplicand is assign to register in, as these two are parallel, so that is why, both C 0 and C 1 together is R 1. So, 2 1, in one row of the control function, indicates a parallelism that means, two micro instructions are combine, and this is the select line for that. Similarly, the condition select is 0 0, branch address is 0 0 0, because we it is not, going anywhere, then C 2 control signal is generated, means the C 2 means, the multiplier is assign 2 register Q.

Now, again condition select is 0 0, branch address is 0 0 0, now again C 3 C 4, C 3 means, this the addition, R is R plus M, and this the C 4 Q equal to Q minus M. Again two micro instructions are get together, and this is the condition select, because no

branching, now, when condition select is 0 1, this is 0 1 0. So, control functions are all 0, because if Z equal to 0, then it goes to address 2, now when select is 0 0, branch address is 0 0 0, then C 5, and C 5 means, the register R is signed to out bus, that is selected, if it is 1 0, then 1 0, branch address is 1 0 1, because go to address 5, so this is a this is halt.

So, this is a, this is go to 5 instructions, go to 5, and this is go to 2, go to 2, so actually, if we see the comments, are, are what a micro instructions are being performed. So, these are the, if we take these are the serial number, or the decimal number of ROM address, we can tell, these are the ROM address in decimal, ROM address in decimal.

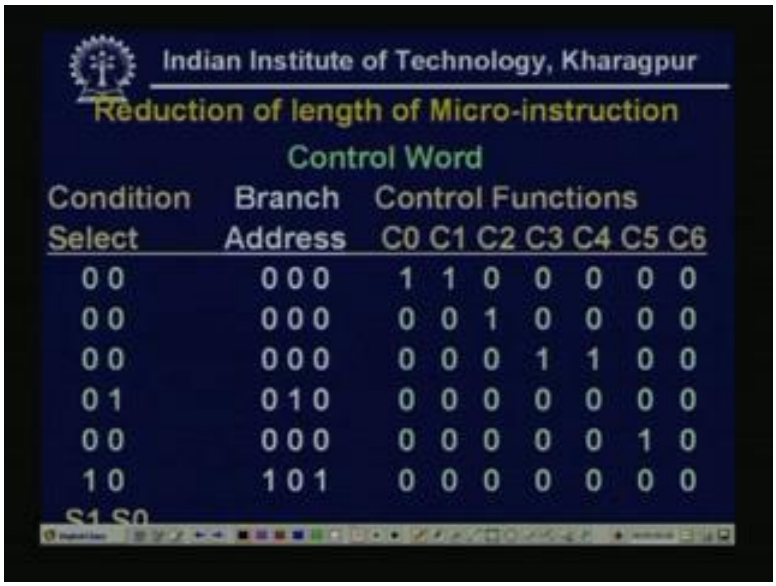
(Refer Slide Time: 47:44)



Then, at 0 th address, actually these two instructions C 0, C 1, so here C 0, C 1 are C 0, C equal to 1, no branch, here Q is in bus, that means, C 2 is 1 no branch, R is R plus M, Q is Q minus 1. So, again it here it is, here it is R is are plus M, Q is Q minus 1, and F is R means, the result, these are actually C 3, C 4, C 3 R is are plus M, is C 3, C 3, C 4 and C 5. Now, next is a, if Z equal to 0, then go to 2, go to 2 means in decimal 0 1 0 and go to address 5, this is unconditional 1 0 1.

So, that is why, here it is a branching, to address 0 1 0, here it is a branch, with the address 1 0 1. So, if we check now, if we check this thing, then it is go to 2, and go to 5, that is why, it is go to 2 and go to 5.

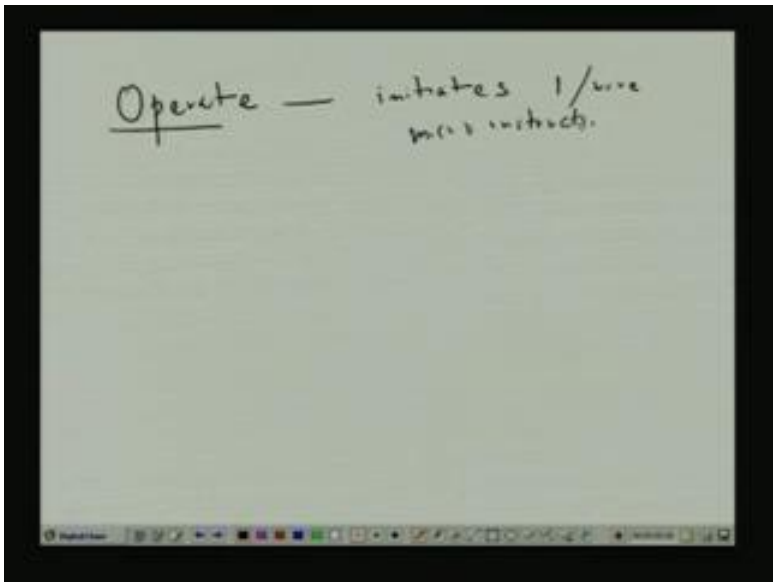
(Refer Slide Time: 50:01)



Now, we see the whether the length of micro-instruction can be reduced, see, the what we have seen, that it is obvious, the control function field, contains all 0's in, in case of branch instructions. So, in case of branch instruction, these are the two branch instruction, one is go to 2, and go to 5, so the branch address is 2 in decimal five, and then, this control functions or the control signals of the functions are 0. Now, in a typical micro-program, there may be several conditional and unconditional branch instructions.

So, therefore, a lot of memory space, inside the control unit, but now what we have seen, that if a, if it is a branch instruction, then actually, the control functions are our control signals are 0's, control functions generates, 0 control signals. So, in practice the format of the control word is organized in a different manner, to minimize the size, this reduces the implementation cost of control unit. Now, whenever, there are several branch instructions, the micro instructions can be formatted, by using a method called multiple micro-instruction format. Now, in these, approach the micro instructions are divided into two groups, Operate and Branch instructions.

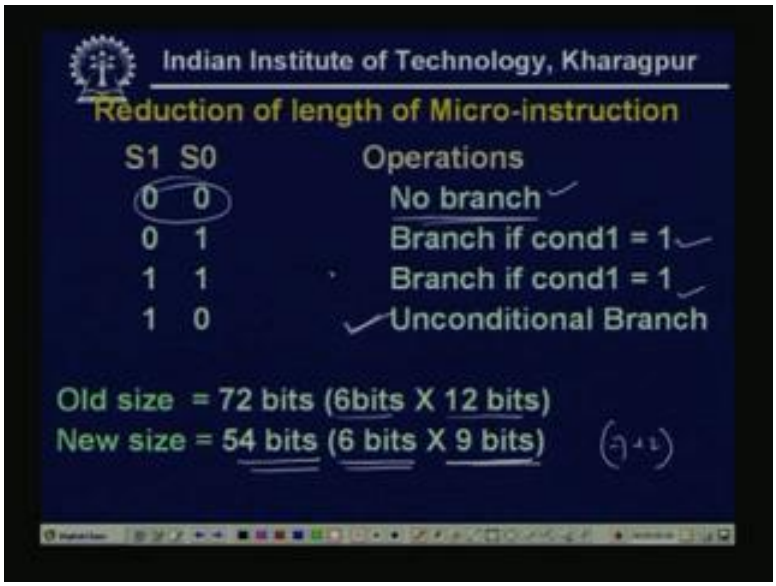
((Refer Slide Time: 52:18)



An operate instructions initiates, operate instructions initiates 1 or more micro-operations, it initiates 1 or more micro instructions. So, after the, of execution of operate instruction, MPC will be incremented by 1, and in case of a branch instruction, no micro-operation, we will usually be initiated, and the MPC, may be loaded with a new value. Now, this means, that the branch address field can be removed, from the micro-instruction format.

So, the control function field is used to specify ((Refer Time: 53:04)), the branch address itself. So now, if we see, the reduction, so these white marked column, branch address that can be, that can be detached, that can be removed from the control word. So, now it size will be, only the condition select, and the control functions.

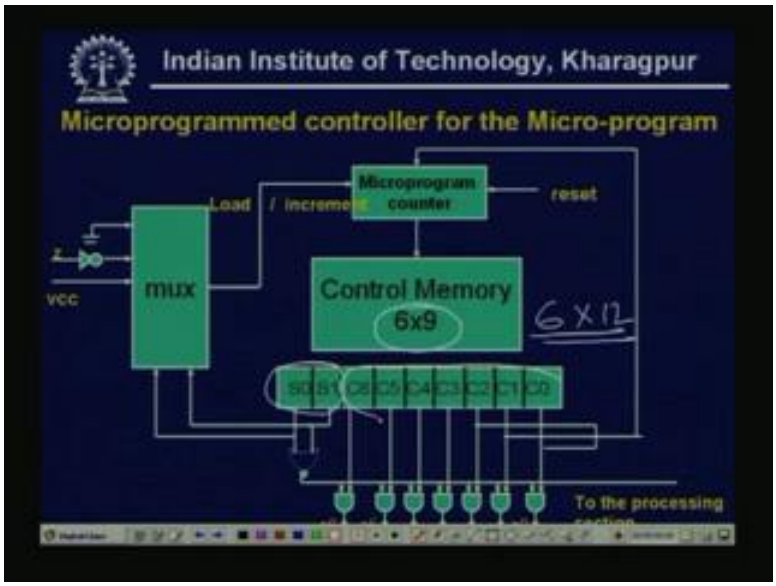
(Refer Slide Time: 53:37)



So, reduction will be, now this select will be 0 0, 0 1, 1 1, 1 0, and the operations are no branch, no branch for 0 0, branch if condition 1, branch if condition 2. Because here, there are only two branches, and this is a unconditional branch, this is a unconditional branch, say complete. And this is go to 2, and go to 5 for this particular example, and always it will be 0 0, whenever there is a no branch, so the old size was the old memory size was 72 bits, because 6 bits address, and 12 bits control words.

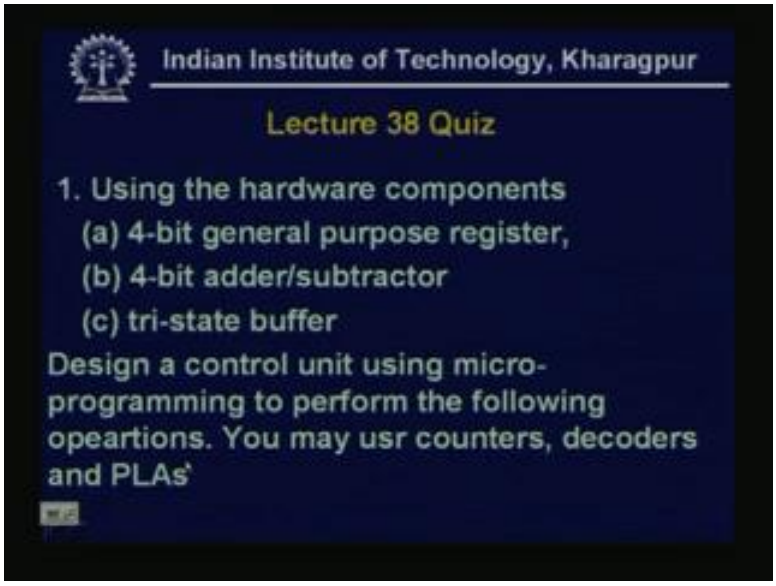
Now, for after, removing the branch address, branch address the, it becomes 7 plus 2, 9 bits 7counter signals and 2, select, condition select fields, so this is 7 plus 2. And, new size will be, 6 bits the address remains same, and by 9 bits, so these are 54 bits, so this is a reduction of length of micro instruction.

(Refer Slide Time: 54:50)



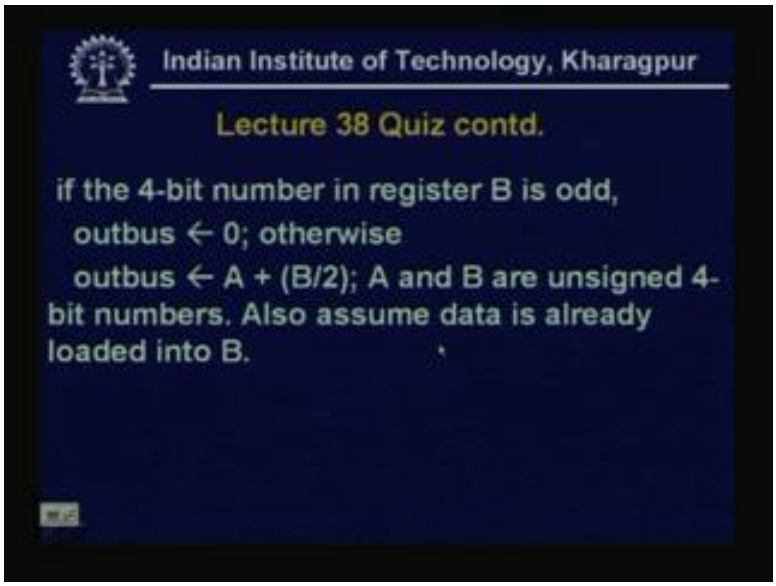
So, if we the modified one, design if we write, again this will be a multiplexer, the micro-program counter, control memory is now, becomes 6 into 9, instead of 6 into 6 by 12. So, the now this will be S 0, S 1 only the select lines, and C 0 to C 6 the seven controls feeds, and there will be no branch, there will be no branch, and these are the C 0 to C 6 lines to the processing section. So, in this way, the reduction can be possible for the Micro programmed memory, that design or length of micro-instruction we can tell.

(Refer Slide Time: 56:01)



Now, we see that, quiz of this lecture, so using these hardware components, say 4 bit general purpose register, 4 bit adder, subtractor, tri-state buffer, design a control unit, using microprogramming to perform the following operations. Now we can use, we can use, counters decoders and PLAs.

(Refer Slide Time: 56:28)



And, the operations are like this, if the 4 bit number in register B is odd, then 0 is assign to out bus, otherwise out bus is A plus B by 2, A and B are 4 bit numbers. And, we assume, data is already loaded into B, so this is the quiz of this particular lecture, and here, we finish this class.

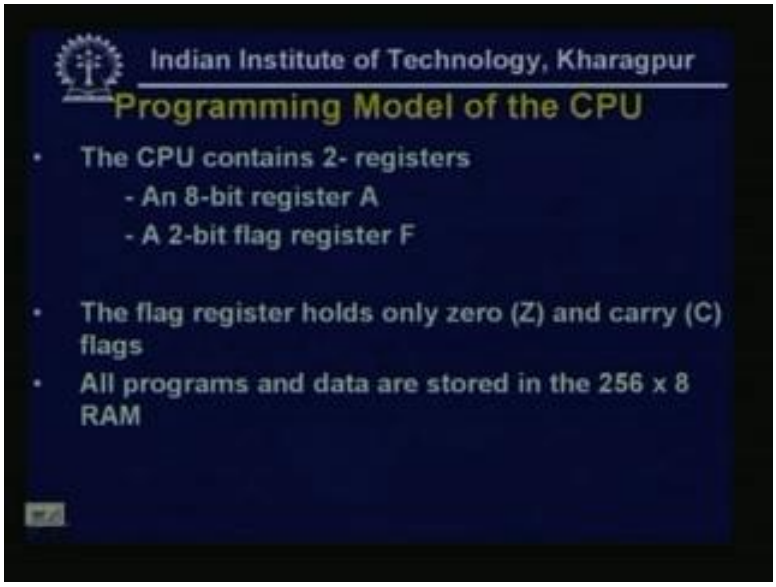
Thank you.

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

Lecture - 39
Design of a Micro-Programmed CPU

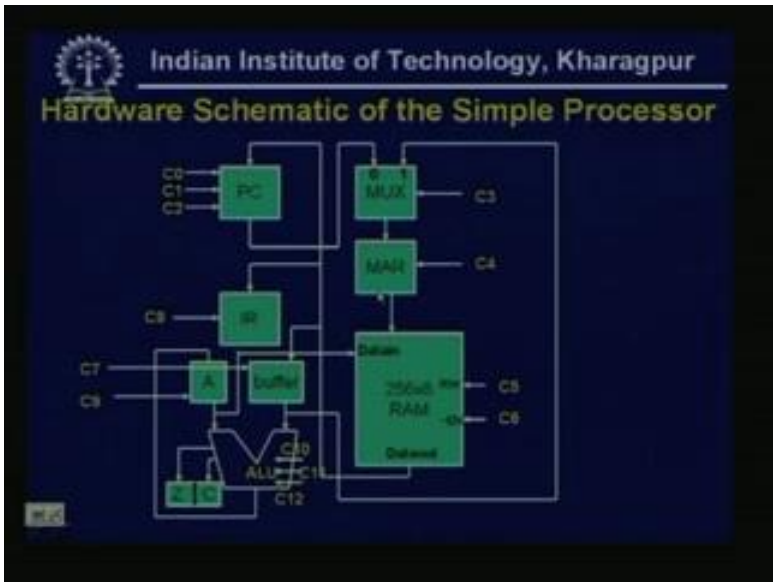
So, already we have, read the design of a CPU, mainly the two parts we have read, the design of ALU, and the design of control unit, there are two approaches, the hardware controlling unit, and the micro-programmed control unit. Now, this class, we will read a design of a micro-programmed CPU.

(Refer Slide Time: 57:42)



So, now first we assume, the programming module of the CPU, The CPU, that we want to design contains 2 registers. An 8 bit registers A, A 2 bit flag register F. The flag register holds only 0, and carry C, carry C flags, all programs and data are stored in a memory, and the memory size is a 256 by 8 bit RAM. Now, the first we see, the hardware schematic of the, data flow part of this processor.

(Refer Slide Time: 58:44)



So, it has the program counter, PC MAR the memory address register.