Computer Architecture and Organization

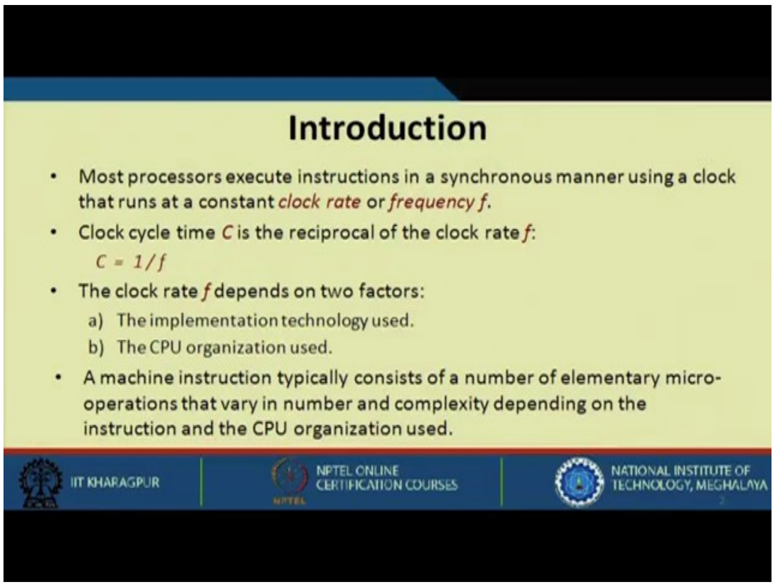
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture - 12
Measuring CPU Performance

Welcome to week 3 lecture. In the last couple of weeks what we have studied is how we can execute an instruction, what the various architectures that are existing, what are addressing modes, instruction format and various other things that are necessary for this particular course. Apart from that, we have talked about a simulator that is SPIM, and we have seen that how we can write programs using low level assembly language. In this particular week, we will be looking into how we can measure the performance of a CPU. We know that for any program, you need some instructions to execute that particular program.

Now how you can actually measure the CPU performance. By that I mean, that you can run the same program in 2 different CPU, and then how you can tell that which one is better. So, in this particular week we will be looking into the various aspects of CPU performance. And then we can say at the end of this week lecture, that how can you say that my CPU is better than the other CPU.

(Refer Slide Time: 01:57)



Introduction

- Most processors execute instructions in a synchronous manner using a clock that runs at a constant *clock rate* or *frequency f*.
- Clock cycle time *C* is the reciprocal of the clock rate *f*:
  $C = 1/f$
- The clock rate *f* depends on two factors:
  a) The implementation technology used.
  b) The CPU organization used.
- A machine instruction typically consists of a number of elementary micro-operations that vary in number and complexity depending on the instruction and the CPU organization used.

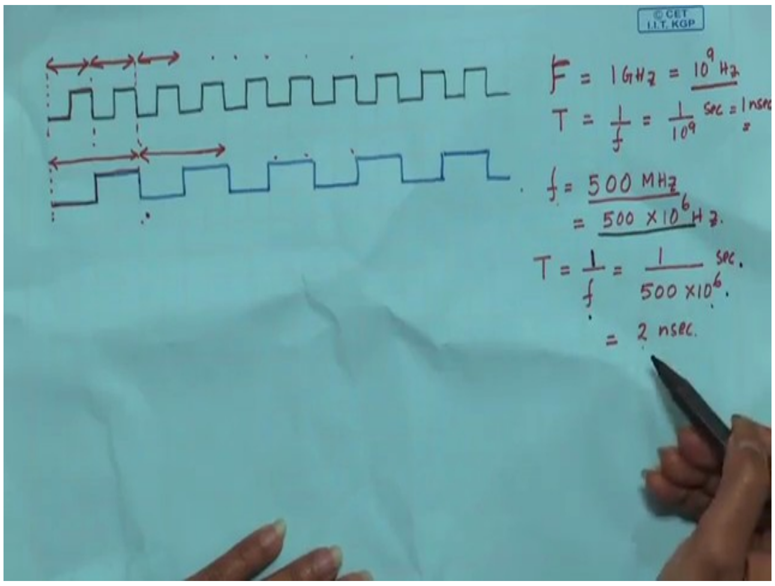
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So, coming to the introduction, we know that most processors execute instructions in a synchronous manner using a clock that runs at a constant rate or frequency. So, what do we mean by that.

Now, how instructions get executed through a clock; that means, a clock is coming? And at the positive edge of the clock or within that clock period, we can say certain task is performed.

Now what is clock cycle time? Clock cycle time is the reciprocal of clock rate. That is C is 1 / f. Clock cycle time is often termed as clock period, which is the reciprocal of frequency, that is 1 / f. First let us see these 2 factors f and C in some detail.

(Refer Slide Time: 03:29)



So, this is a clock, and this also a clock. Let us say in this clock, this is the off period, this is the on period. And this whole is one period. And let us say we have another clock whose period is little more.

So what we are doing? Here this is the off period and this is the on period. So, this is the total clock. Similarly for this is the off period, this is the on period. So, this is your total clock and so on. Now you see that, for this particular clock the time period is small, that is, your clock cycle time is small. And in this clock, the clock cycle time is more. And what we know that, in processors we perform a task with respect to these clocks. So, whenever this clock is coming a particular task is getting performed. And so what we can analyze from these 2 clocks. Let us see that for the first clock let us say the frequency is 1 GHz.

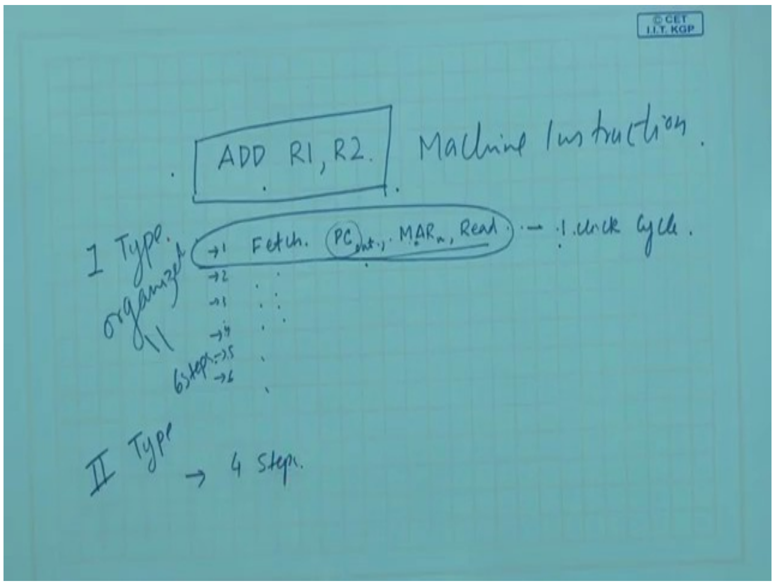
So what will be the time period? As I said time period will be 1 / f. So, time period will be 1 / 10^9 second, that is 1 nanosecond. Now let us see about this particular clock. In this clock the frequency is 500 MHz. So, as it is 500 megahertz, then f = 500 x 10^6 Hz. So, time period T = 1 / f = 2 nanoseconds. That means, for the previous clock the time period is 1 nanosecond. For this one, the time period is 2 nanosecond.

So which will be faster; obviously, the first one will be faster than the next one. So, now we know how we can relate clock frequency with clock period.. Next we see that on what factor does this clock rate depend on. So, there are 2 important factors on which this clock rate depends. The first one is the implementation technology. So, by implementation technology what we mean is that with the advancement of technology, the size of transistors are becoming smaller and smaller. And with that the clock speed is becoming faster the clock is becoming faster basically.

So this is a factor on which the clock frequency depends. Another is the CPU organization. By CPU organization what we mean is that, suppose in a clock period we say, that some part of the instruction is executed. So, basically an instruction is divided into some cycles. I mean each of the work of that particular instruction is performed in those cycles. And by CPU organization we mean how we can organize the CPU such that, the number of tasks that can be performed within that clock period is maximized.

So, these are the 2 factors on which the clock rate actually depends. So, as I said just now that the machine instruction typically consists of number of elementary micro operations, that vary in number and complexity depending on the instruction, and the CPU organization used.

(Refer Slide Time: 09:32)



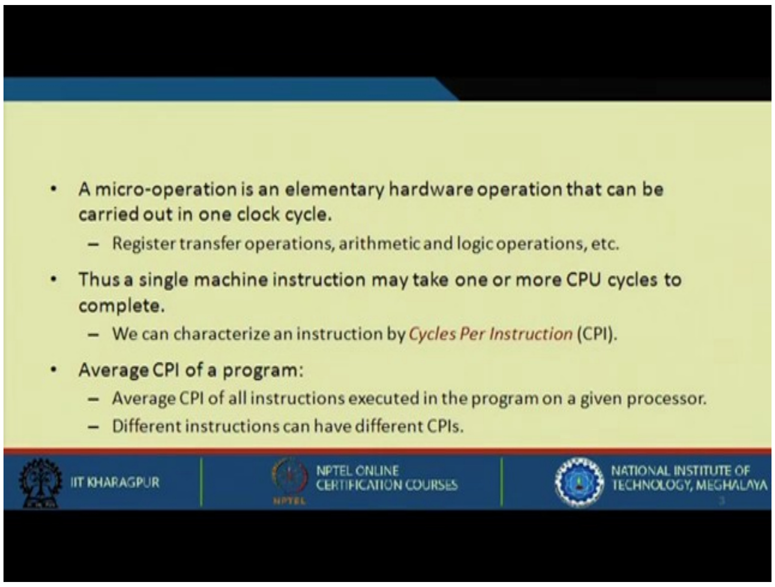
We will take an example here --- ADD R1, R2. To execute this instruction what we say that a machine instruction typically consists of number of elementary micro operations.

So to execute this particular instruction, we require certain steps. What are the steps? So, what we need to do first this particular instruction is stored in memory. You have to bring that from the memory. So, first is the fetch phase. Once you fetch, how will you fetch it? The content of PC should be made available. using some control signals. We will make the content of PC available, which is PCout we will see in details next. But in a simple word, I will say that once we do this PCout, the content of PC is irrelevant to some bus. And then we have to put this value in MAR memory address register.

So this we do: PCout, MARin, then we Read and then we do some other things. So, first we fetch the instruction. So, with just only these steps, we cannot fetch. We may require some more steps to fetch. Now what the point is one these particular steps can be performed, let us say in one clock cycle. And there can be many more steps to execute this machine instruction. What we require? Let us say we require 6 steps to execute this instruction. And we see that each of these steps require one clock cycle. So, this is what we mean by a machine instruction. Typically consists of number of elementary micro operations that vary in number and complexity depending on the instruction and the CPU organization used.

Now if you use a different CPU organization, these steps that I am saying might be different. Like say, for one type of organization, it requires 6 steps. For another type of organization, it may take 4 steps. So, we really cannot say that how you can differentiate. It depends on the organization used and it depends on the complexity of the instruction as well.

(Refer Slide Time: 12:54)



- A micro-operation is an elementary hardware operation that can be carried out in one clock cycle.
  - Register transfer operations, arithmetic and logic operations, etc.
- Thus a single machine instruction may take one or more CPU cycles to complete.
  - We can characterize an instruction by *Cycles Per Instruction* (CPI).
- Average CPI of a program:
  - Average CPI of all instructions executed in the program on a given processor.
  - Different instructions can have different CPIs.

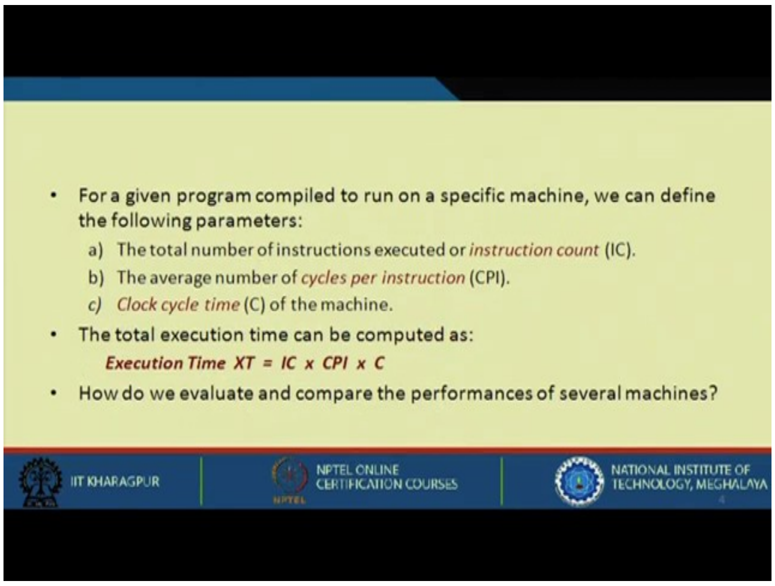
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Now, moving on, I said a micro-operation is an elementary hardware operation that can be carried out in one clock cycle. So, all the set of instruction as I said PCout, MARin, Read and so on can be executed in one clock cycle. And in one clock cycle what we can do basically is some register transfer instructions, some ALU operation instruction. Because all those are within the CPU and for that we do not have to bring it from the memory.

So whenever you have to bring it from the memory, we have to see that how much time will be required for that. CPU is much faster compared to memory. Transferring or getting a data from memory to CPU will take more time. Thus a single machine instruction may take one or more CPU cycles to complete. We can characterize an instruction by cycles per instruction. What do you mean by cycles per instruction? As I said an instruction is divided into some basic operations. Some micro-instructions are executed to execute that machine instruction. And all those those micro instruction that are getting executed requires some cycles.

So ultimately an instruction takes certain amount of cycles to execute it. So, that is called cycles per instructions. So, every instruction is taking some cycles to execute and that is termed as cycles per instruction. And what is this average CPI of a program? Average CPI of a program as we can say that see there can be many instructions and many instructions can have different CPIs. So, average CPI of all instruction executed in a program on a given processor. So, we average it; that means, some instruction, let us say takes 5 cycles, some instruction takes 7 cycles, some instruction takes 4 cycles. We average it out and then we say this is the average CPI. As different instructions can have different CPIs we will see in detail.

(Refer Slide Time: 15:38)



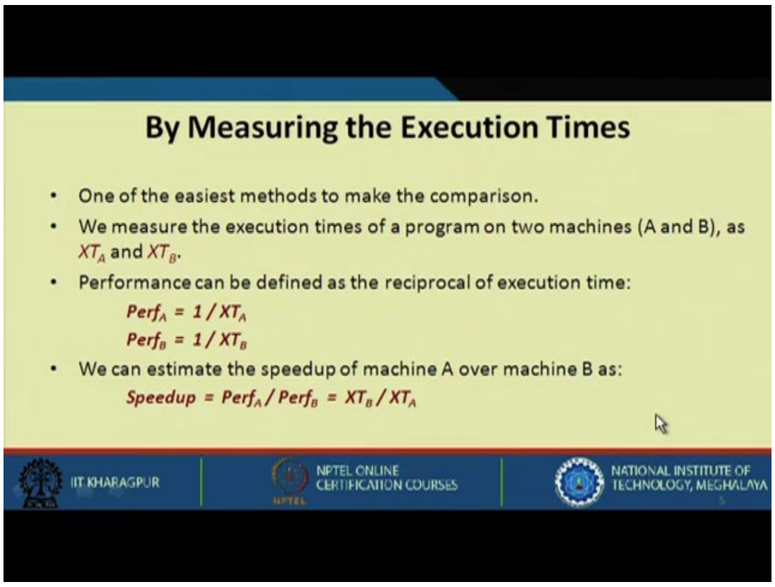
- For a given program compiled to run on a specific machine, we can define the following parameters:
  a) The total number of instructions executed or *instruction count* (IC).
  b) The average number of *cycles per instruction* (CPI).
  c) *Clock cycle time* (C) of the machine.
- The total execution time can be computed as:
  Execution Time XT = IC x CPI x C
- How do we evaluate and compare the performances of several machines?

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So for a given program, compiled to run on a specific machine, we define the following. Parameters for a given program, now we are talking about the program that is compiled on a machine, these are the parameters that are important. What are the parameters? The total number of instructions executed, we call it instruction count. So, for a program what is the total number of instruction that is executed? The average number of cycles per instruction as I have already discussed that is the CPI. So, if there are 20 instructions and each requires some cycles. So, CPI is the total average number of cycles per instruction. And finally, the clock cycle time or the period of the machine.

So now what will be the total execution time? So, the total execution time can be computed as XT = IC x CPI x C. So, how do we evaluate and compare the performances of several machines? Next we will see that for is the execution time. So, now, the execution time is the number of instructions multiplied by the CPI, multiplied by the clock cycle time. So, you multiply all these things you will get the execution time, we call it XT. Now we will see how we evaluate and come compare between the performances of several machines.

(Refer Slide Time: 17:57)



By Measuring the Execution Times

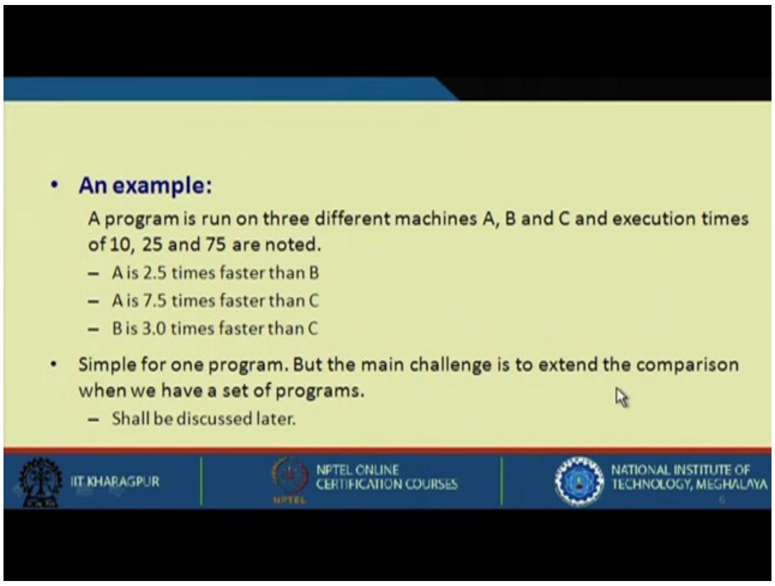
- One of the easiest methods to make the comparison.
- We measure the execution times of a program on two machines (A and B), as XT_A and XT_B.
- Performance can be defined as the reciprocal of execution time:
  - $Perf_A = 1 / XT_A$
  - $Perf_B = 1 / XT_B$
- We can estimate the speedup of machine A over machine B as:
  - $Speedup = Perf_A / Perf_B = XT_B / XT_A$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So one of the easiest method that can be used to compare is like we measure the execution time of a program on two machines, that is A and B. And we see that execution time of A is XT_A and execution time of B is XT_B. So, the performance of A is $1/XT_A$ and performance of B is $1/XT_B$. So, let us say the one processor performs a task in 10 seconds and another processor performs the task in 2 seconds. Which one is better? The processor that performs a task faster means, less time is better. So, the processor that performs within 2 seconds will be better.

Now we can estimate the speedup of machine A over machine B as performance of A divided by performance of B. And performance of A is $1/XT_A$ and performance of B is $1/XT_B$. So, if you just put on these two values in this place, you will get the speed up of machine A over machine B. Now let us take some examples. Let us say a program is run on 3 different machines.

(Refer Slide Time: 19:43)



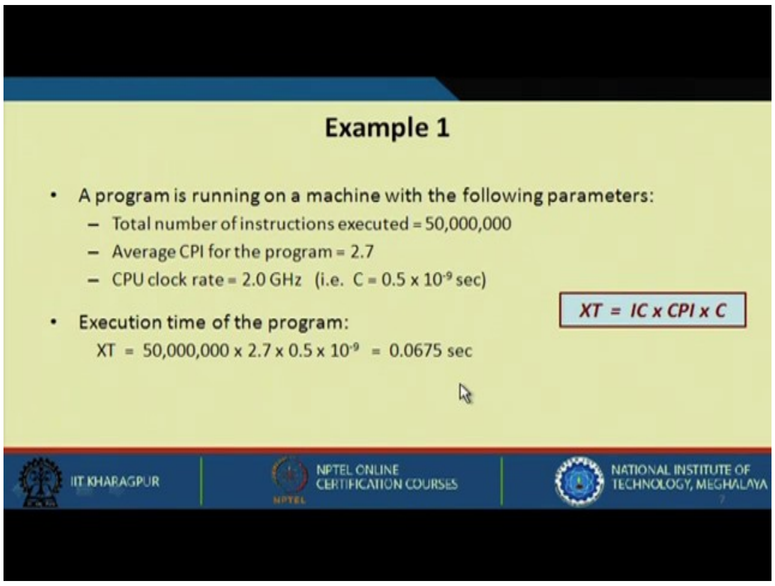
- **An example:**
  A program is run on three different machines A, B and C and execution times of 10, 25 and 75 are noted.
  - A is 2.5 times faster than B
  - A is 7.5 times faster than C
  - B is 3.0 times faster than C
- Simple for one program. But the main challenge is to extend the comparison when we have a set of programs.
  - Shall be discussed later.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So, you have machines A, B and C. And the execution times are 10, 25 and 75. So, what you can see from this what we can say that A is 2.5 times faster than B.

So B is taking 25. So, we divide 25 by 10 and we get 2.5. So, we can say that A is 2.5 times faster than B. Similarly let us compare A and C. We can say A is 7.5 times faster than C. And similarly B is 3 times faster than C. So, B is 25. So, 75 divided by 25 we get 3. So, B is 3 times faster than C. This is simple for one program, but what if we have different set of programs, how do we compare, this shall be discussed in course of time.

(Refer Slide Time: 21:17)



Example 1

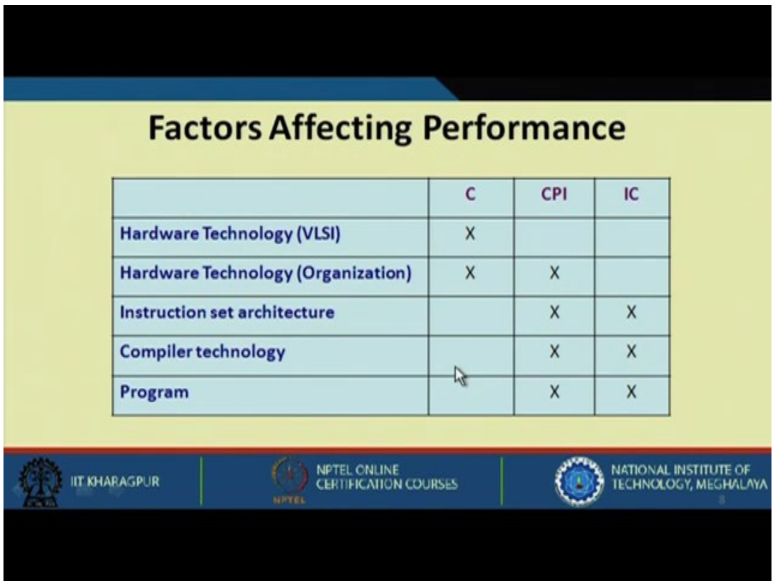
- A program is running on a machine with the following parameters:
  - Total number of instructions executed = 50,000,000
  - Average CPI for the program = 2.7
  - CPU clock rate = 2.0 GHz (i.e. $C = 0.5 \times 10^{-9}$ sec)
- Execution time of the program:
  $XT = 50{,}000{,}000 \times 2.7 \times 0.5 \times 10^{-9} = 0.0675$ sec

$\boxed{XT = IC \times CPI \times C}$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Now let us take an example. Say a program is running on a machine with the following parameters. So, what are the parameters, I give you the total number of instructions. So, what is your total number of instruction; this is your IC. So, what is the average CPI taking into account of different CPI is for these 50 million instructions, we get a CPI of 2.7. And what is the CPU clock rate? From frequency we can find out the time period by C will be your 1 / $2x10^9$ = 0.5 x 10^{-9} second. And as we know that the execution time is IC x CPI x C. And all these values are provided here. We get XT as 0.0675 second.

(Refer Slide Time: 22:38)



Factors Affecting Performance

	C	CPI	IC
Hardware Technology (VLSI)	X		
Hardware Technology (Organization)	X	X	
Instruction set architecture		X	X
Compiler technology		X	X
Program		X	X

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

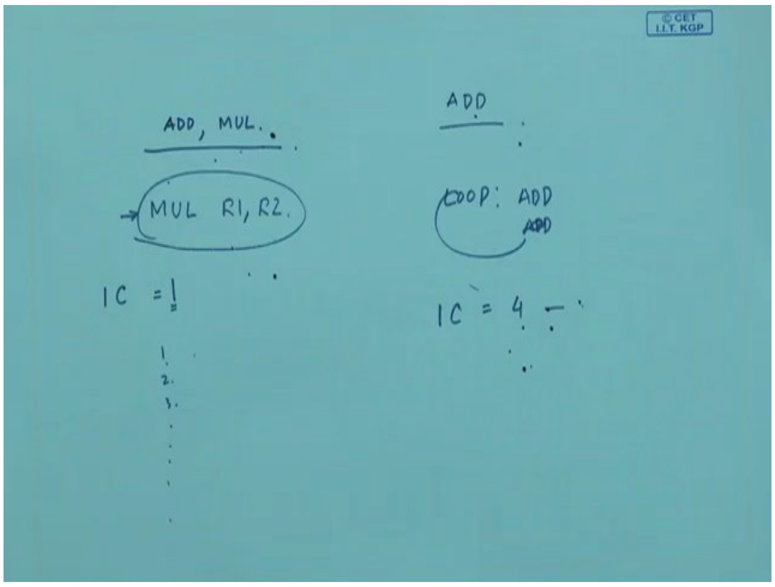
Now, see what are the factors that affect performance. First one is the hardware technology. Definitely if you make your clock cycle time smaller then it is pretty obvious, that you will be getting better result, but at the same time you have to cope up with the operations that can be performed in one clock period or one clock cycle. Next is the hardware technology, that is, the organization. So, what all factors that depends on this organization? First is the clock cycle time another is the cycles per instruction.

So let us say when we talk about organization, it depends on like a how your various hardwares are actually mapped. So, depending on that definitely your CPI that is cycles per instruction will vary you have an efficient organization, where is your cycles per instruction can be reduced. So, this particular factor that is hardware technology that is organization depends these are the factors that will govern, that is, your CPI as well as your clock cycle time. Now instruction set architecture, what all factors will depend it?

What is the instruction set architecture? We mean, that how various kinds of instruction you are putting it in your architecture. How many types how many varieties you are putting it there.

So, definitely if you restrict the number of instructions to a minimum level for a program you might require more number of instruction counts.

(Refer Slide Time: 25:13)

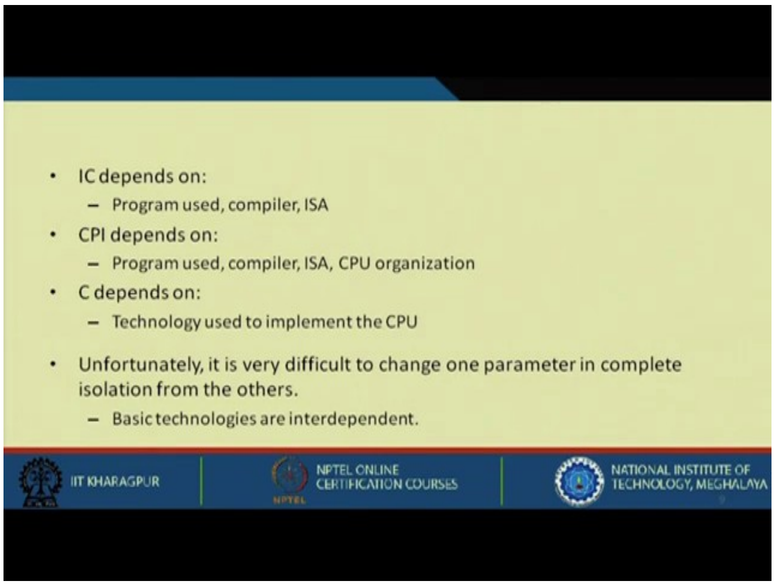


So, there are two things that can happen. Basically what I am trying to mean is that, let us say you have ADD and MUL instructions. And in one case you only have ADD instruction.

So in this particular architecture if you want to multiply two numbers you can simply use MUL R1,R2, but in the same case if you want to use repeated addition, then you have to use a loop. And that loop will perform that addition number of times. So, this particular ADD instruction may be used in that loop repeatedly. So, the idea is if you have more number of instructions, you might require less number of instructions in the end to execute a program. In a similar case you have less number of total instructions, but in that case you require more number of instructions to execute the same program. So, this is the difference.

So these instructions set architecture will affect two factors, one is CPI, another is instruction count. Now different compilers can generate different codes. Let us say for the same program one compiler is taking 10 instructions, another compiler is taking 20 instructions. So this IC varies greatly on the compiler technology that is used. So, nowadays the compiler is becoming more and more intelligent and they are going hand in hand with the hardware. The compiler must know what kind of hardware architecture you are using such that it will generate the code in a fashion that will be easy for execution. And so such that it will also require less number of instructions and of course, what are the factors will depend both the CPI and the instruction count.

(Refer Slide Time: 27:34)

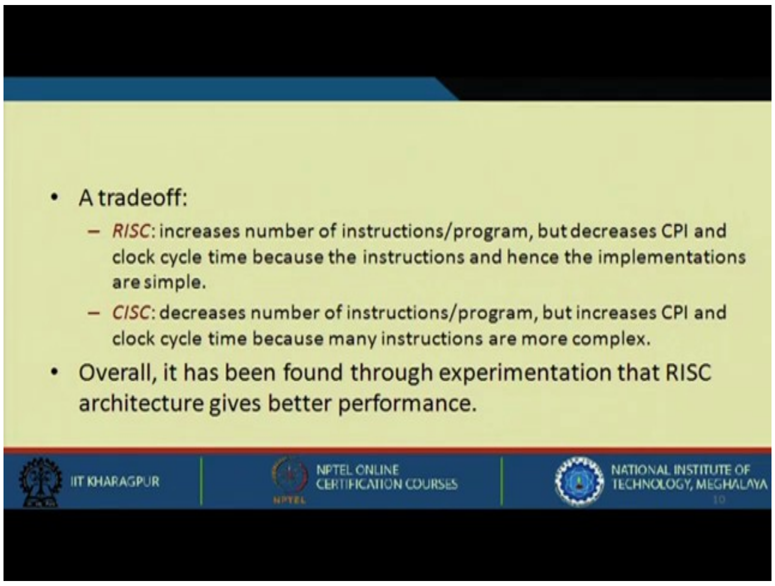


- IC depends on:
  - Program used, compiler, ISA
- CPI depends on:
  - Program used, compiler, ISA, CPU organization
- C depends on:
  - Technology used to implement the CPU
- Unfortunately, it is very difficult to change one parameter in complete isolation from the others.
  - Basic technologies are interdependent.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So as I discussed in the previous slide, IC depends on program used, the compiler, and the instruction set architecture. CPI also depends on program used, compiler, the instruction set architecture, as well as the CPU organization --- how you organize your CPU, how you organize your various hardware. So, final CPI will definitely depend on the CPU organization that you are using. And finally, C depends entirely on the technology used to implement the CPU. And this is very unfortunate that it is very difficult to change one parameter in complete isolation from the others. So, the basic technologies are very much interrelated to each other. So, we really cannot vary one parameter completely compared to the other.

(Refer Slide Time: 28:43)



- A tradeoff:
  - *RISC*: increases number of instructions/program, but decreases CPI and clock cycle time because the instructions and hence the implementations are simple.
  - *CISC*: decreases number of instructions/program, but increases CPI and clock cycle time because many instructions are more complex.
- Overall, it has been found through experimentation that RISC architecture gives better performance.

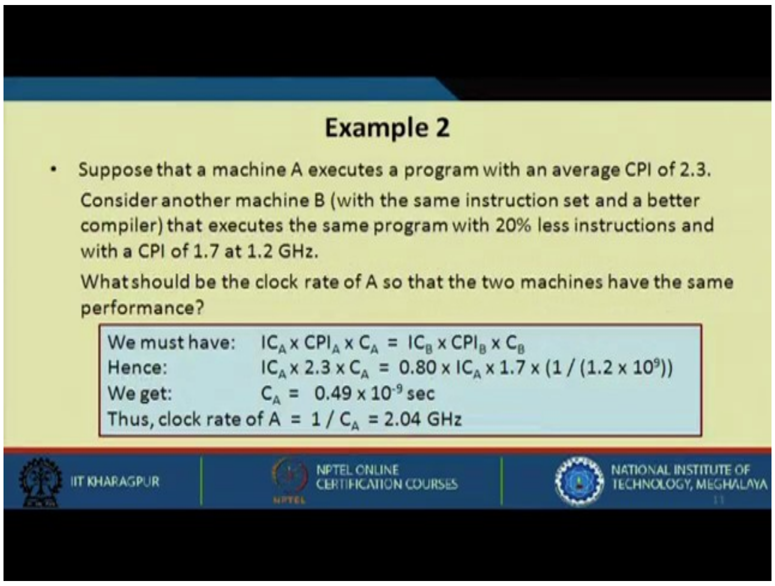
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA
10

So what is the tradeoff here? If you see a RISC machine, the number of instructions per program is more. So, increases the number of instructions per program, but at the same time it decreases the CPI and the clock cycle time. Because the instructions and hence the implementations are simple. But in CISC, decreases the number of instructions, but increases the CPI and the clock cycle time because many instructions are more complex. So, overall what has been found? It has been found that, RISC architecture gives better performance. So, let me tell you with the same example that we have taken earlier.

So we have MUL instruction and this is the instruction count, that is, IC is 1 here. And for this let us say the loop executes 4 times. So, in such cases depending on how many number 4 multiplied 5 times with 5 or 4 multiplies 10 times. So, it depends on that now IC is 1, but the number of cycles required to execute; that means, the micro operation that you are using steps in each steps we are performing something that might be more. So, CPI will be more. In this case may be IC is more, but overall it will take less number of cycles. So, CPI will be less here, but IC will be more. So, this is a tradeoff we can see.

So far in one case, we can have more IC where the CPI will be less; in some case we have less IC, but the CPI intern will be more.

(Refer Slide Time: 30:53)



Example 2

- Suppose that a machine A executes a program with an average CPI of 2.3.
  Consider another machine B (with the same instruction set and a better compiler) that executes the same program with 20% less instructions and with a CPI of 1.7 at 1.2 GHz.
  What should be the clock rate of A so that the two machines have the same performance?

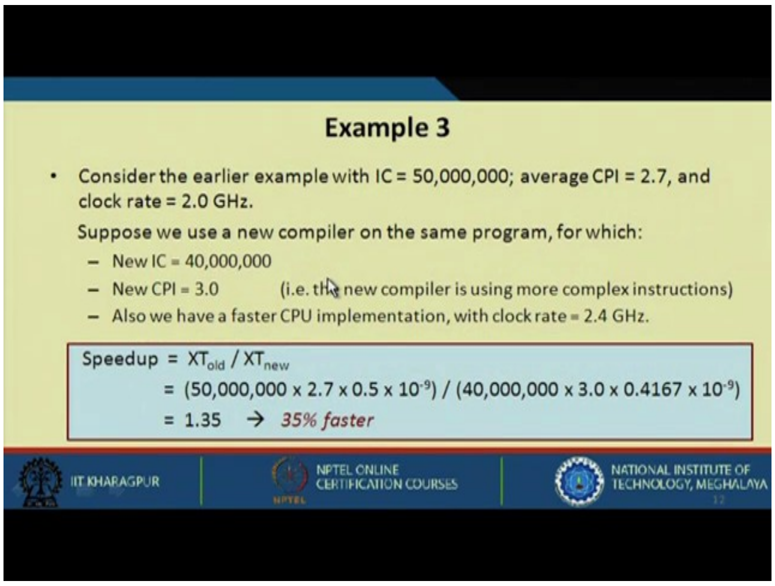
We must have: $IC_A \times CPI_A \times C_A = IC_B \times CPI_B \times C_B$
Hence: $IC_A \times 2.3 \times C_A = 0.80 \times IC_A \times 1.7 \times (1 / (1.2 \times 10^9))$
We get: $C_A = 0.49 \times 10^{-9}$ sec
Thus, clock rate of A = $1 / C_A$ = 2.04 GHz

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us take an example. Suppose that a machine a executes a program with an average CPI of 2.3; consider another machine B with the same instruction set and a better compiler that execute the same program with 20% less instructions. And with the CPI of 1.7 at 1.2 GHz what should be the clock rate of A so that the two machines have the same performance. So, for both the machines CPI is given. And one machine executes the same program with 10% less instruction.

So one takes 100% and another takes 20% less; that means, 80%. So, for A IC_A will remain same; this is 2.3; and this is C_A, that is what we have to find it out. Clock of A and for this IC is 20% less. So, it will become 80%. So, 0.80 x IC x 1.7 x this is the clock rate this is the period. So, the clock rate is 1.2 GHz, that is, 1.2×10^9; 1 divided by that will give you the clock period of this. You will get a clock period of 0.49×10^{-9} second that is coming to 2.04 GHz.

(Refer Slide Time: 32:35)



Example 3

- Consider the earlier example with IC = 50,000,000; average CPI = 2.7, and clock rate = 2.0 GHz.

  Suppose we use a new compiler on the same program, for which:
  - New IC = 40,000,000
  - New CPI = 3.0 (i.e. the new compiler is using more complex instructions)
  - Also we have a faster CPU implementation, with clock rate = 2.4 GHz.

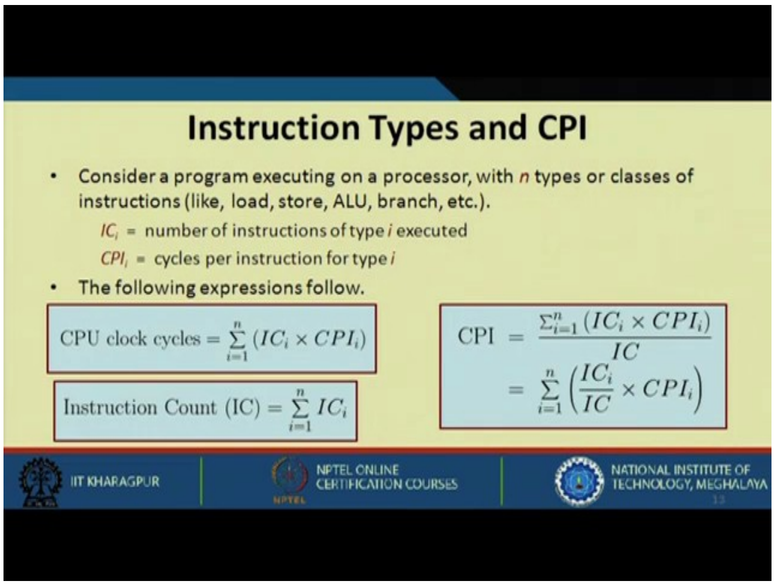
Speedup = XT_{old} / XT_{new}
= $(50{,}000{,}000 \times 2.7 \times 0.5 \times 10^{-9}) / (40{,}000{,}000 \times 3.0 \times 0.4167 \times 10^{-9})$
= 1.35 → *35% faster*

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So we need 2.04 GHz clock for a machine such that both the result should be same. Let us take another example where consider the earlier example with the instruction count of 50 million. Average CPI of 2.7 and clock rate of 2 GHz. Suppose we use a new compiler on the same program for which new IC is 40 million and the new CPI has also increased to 3. Also we have a faster CPU implementation with clock of clock rate of 2.4 GHz. So, these are the different. So, one is having 2 GHz, this is having 2.4, but the CPI of this is less CPI of this is more, but the instruction count of this is even more and this is less.

So if you compare this what will be the speedup? You have to find the execution time of old one compared to execution time of new. So, you just put those values you get the execution time old that is of this one. And you put all these values you get the execution time of this one, which is coming down to 1.35. So, we can say that it is 35% faster.

(Refer Slide Time: 33:48)



Instruction Types and CPI

- Consider a program executing on a processor, with *n* types or classes of instructions (like, load, store, ALU, branch, etc.).
  - IC_i = number of instructions of type *i* executed
  - CPI_i = cycles per instruction for type *i*
- The following expressions follow.

$$\text{CPU clock cycles} = \sum_{i=1}^{n} (IC_i \times CPI_i)$$

$$\text{Instruction Count (IC)} = \sum_{i=1}^{n} IC_i$$

$$\text{CPI} = \frac{\sum_{i=1}^{n} (IC_i \times CPI_i)}{IC} = \sum_{i=1}^{n} \left(\frac{IC_i}{IC} \times CPI_i \right)$$

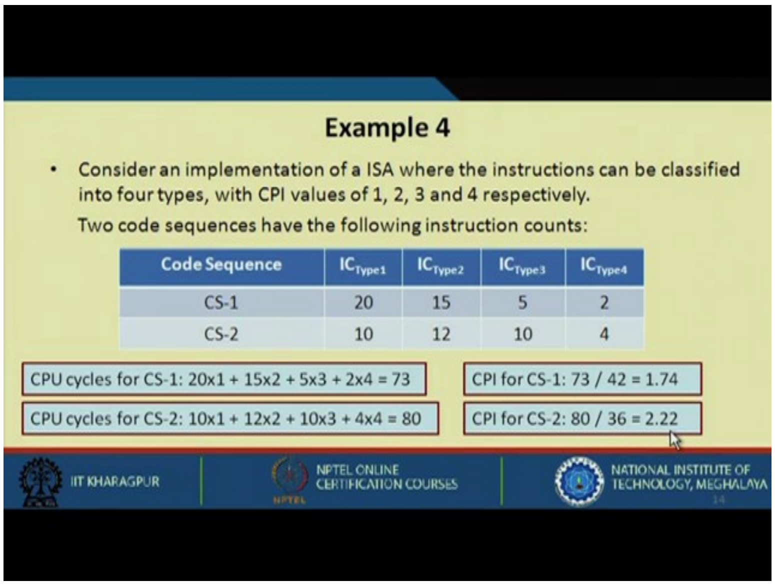
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us take another thing that is instruction types and CPI. Consider a program executing on a processor with n types or classes of instructions. So, generally we do not have one kind of instructions. We have load-store instruction; we have data transfer instruction within the CPU; we have variety of instructions basically.

So these classes are divided into let us say load-store, ALU, branch, etc. So, IC_i is number of instructions of type i executed, CPI_i is cycles per instruction of type i, the following expression follows from this. So, till now we were saying that this is the total IC, this is the total CPI now we divide it we say that there are various kinds of instructions. So, we can have various kinds of instructions and each of these instructions can have different CPIs. So, each instruction will have different CPI. So, CPU clock cycles will be IC_i x CPI_i summation of that. Similarly, instruction count will be considering all the instruction of all the types IC of type I, where i go from 1 to n.

So there are n type of instructions and what will be CPI now? CPI will be summation of instruction count and cycles per instruction divided by instruction count total instruction count. So, summation of IC_i divided by IC into CPI.

(Refer Slide Time: 35:51)



Example 4

- Consider an implementation of a ISA where the instructions can be classified into four types, with CPI values of 1, 2, 3 and 4 respectively.
  Two code sequences have the following instruction counts:

Code Sequence	IC_{Type1}	IC_{Type2}	IC_{Type3}	IC_{Type4}
CS-1	20	15	5	2
CS-2	10	12	10	4

CPU cycles for CS-1: 20x1 + 15x2 + 5x3 + 2x4 = 73

CPU cycles for CS-2: 10x1 + 12x2 + 10x3 + 4x4 = 80

CPI for CS-1: 73 / 42 = 1.74

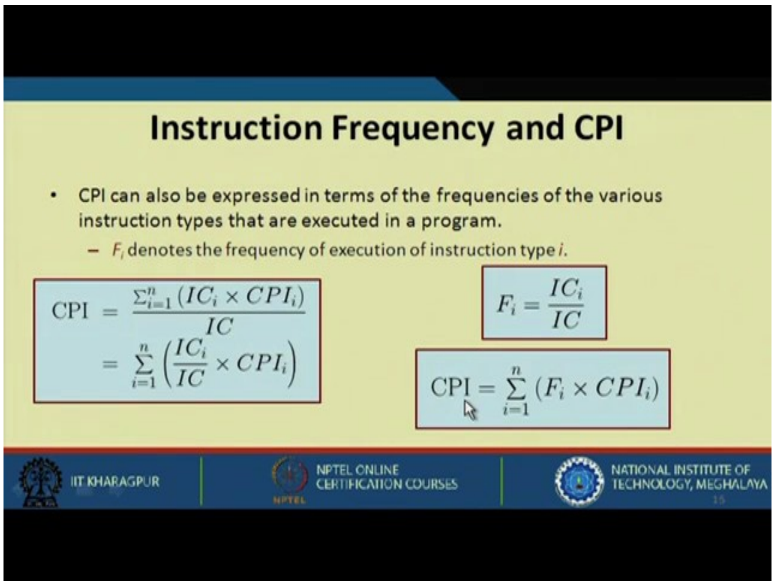
CPI for CS-2: 80 / 36 = 2.22

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us take one more example where we consider the implementation of an instruction set architecture, where the instructions can be classified into four types. So, the CPI values of these 4 types of instructions are 1, 2, 3 and 4 respectively. Two code sequences have the following instruction counts; that means, there are 2 code sequence and these are the various instruction count of type one instruction count of type 2 and so on.

So now you see the CPU cycles for CS-1: 20 x 1 because for type 1 the CPI is 1, type 2 multiplied by 2, 15 multiplied by 2, 5 multiplied by 3 and 2 multiplied by 4. So, this will give you the total CPU cycles that is 73. So, what will be the cycles per instruction, total instruction 20 + 15 + 5 + 2, which is coming down to 42 total instructions. So, CPI will be CPU cycle divided by 42, which is coming down to 1.74. Similarly, for the next one, a total CPU cycle is 80 and CPI is 2.22. So, we can see that it greatly depends on both the type of instruction and what is the CPI of that type of instruction. So, both varies.

(Refer Slide Time: 37:40)



Instruction Frequency and CPI

- CPI can also be expressed in terms of the frequencies of the various instruction types that are executed in a program.
  - F_i denotes the frequency of execution of instruction type i.

$$\text{CPI} = \frac{\sum_{i=1}^{n}(IC_i \times CPI_i)}{IC} = \sum_{i=1}^{n}\left(\frac{IC_i}{IC} \times CPI_i\right)$$

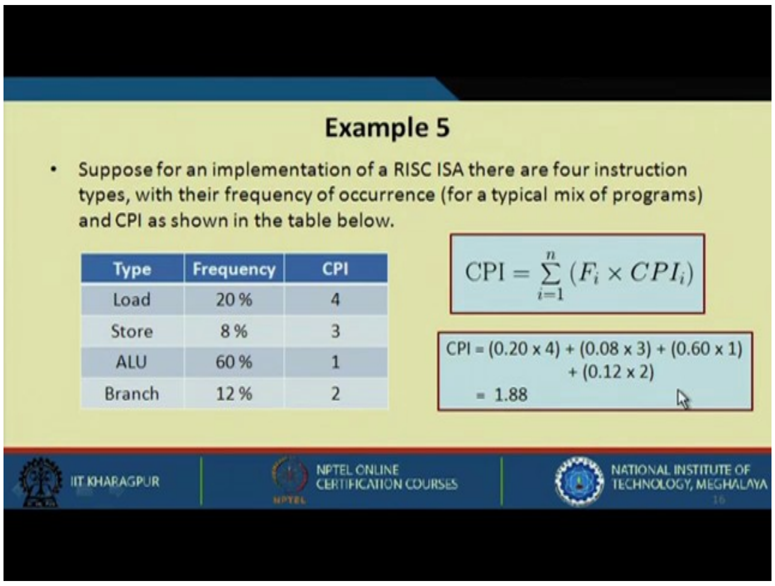
$$F_i = \frac{IC_i}{IC}$$

$$\text{CPI} = \sum_{i=1}^{n}(F_i \times CPI_i)$$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So this is instruction frequency and CPI, where CPI can be expressed in terms of frequencies of various instruction types that are executed in a program. So, F_i denotes the frequency of execution of type i. So, F_i is IC_i divided by total instruction count and CPI we have already shown it in the previous slide can be expressed in terms of frequency. So, we substitute this here. So, we get frequency multiplied by CPI_i.

(Refer Slide Time: 38:15)



Example 5

- Suppose for an implementation of a RISC ISA there are four instruction types, with their frequency of occurrence (for a typical mix of programs) and CPI as shown in the table below.

Type	Frequency	CPI
Load	20 %	4
Store	8 %	3
ALU	60 %	1
Branch	12 %	2

$$\text{CPI} = \sum_{i=1}^{n}(F_i \times CPI_i)$$

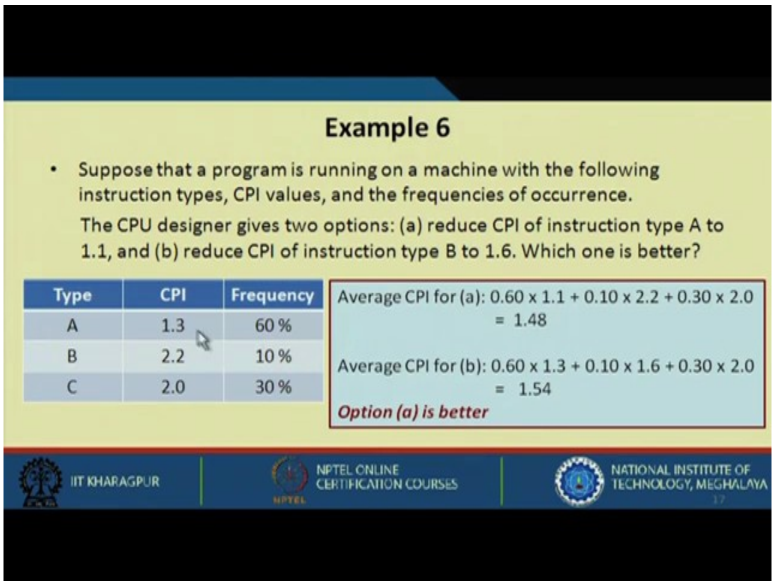
CPI = (0.20 x 4) + (0.08 x 3) + (0.60 x 1) + (0.12 x 2)
= 1.88

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Now, let us take another example. Where suppose for an implementation of a RISC, ISA there are 4 instruction types with their frequency of occurrence for typical mix of programs let us say and CPI as shown in table below.

So this is the frequency at which load instruction is executed. This is the frequency at which store instruction is executed and this is the frequency for ALU one branch. So, and the CPI is given cycles per instruction this is the frequency at which it is happening. So, what is the frequency that is 0.2 and the CPI is 4. So, if you want to find CPI, you can find F_i multiplied by CPI_i. So, 20 percent 0.20 multiplied by 4, 8 percent 0.08 multiplied by 3 and so on. And we get 1.88.

(Refer Slide Time: 39:15)



Example 6

- Suppose that a program is running on a machine with the following instruction types, CPI values, and the frequencies of occurrence.

  The CPU designer gives two options: (a) reduce CPI of instruction type A to 1.1, and (b) reduce CPI of instruction type B to 1.6. Which one is better?

Type	CPI	Frequency
A	1.3	60 %
B	2.2	10 %
C	2.0	30 %

Average CPI for (a): 0.60 x 1.1 + 0.10 x 2.2 + 0.30 x 2.0
= 1.48

Average CPI for (b): 0.60 x 1.3 + 0.10 x 1.6 + 0.30 x 2.0
= 1.54

Option (a) is better

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us take another example suppose that a program is running on a machine with the following instruction types. CPI values and frequency of occurrence the CPU designer gives 2 options, the first option is to reduce the CPI instruction of type A to 1.1 %. So, type A instruction we are reducing to 1.1 %. And reduce CPI instruction of type B to 1.5, CPI of type B is reduced to 1.6. So, this is type A is reduced from 1.3 to 1.1 and type B is reduced from 2.2 to 1.6. Now let us see.

So average CPI for a will be 60.60 %. So, 0.60 multiplied with 1.1 this is the new CPI and all the rest CPI remains the same. So, we get 1.4448 similarly the CPI for B is 0.60 into no change here, but here we change to 1.6 this. 2.2 becomes 1.6 and this remains same. So, we get this. So, from this what we clearly can say is that option A is better, but you see what we have done option A is we have reduced from 1.3 to 1.1, but you see the frequency of this instruction that is getting executed that is 60 %. So, it is much more.

So you must take into account the frequency. Some instructions that are frequently getting executed and you reduce the CPU even less amount, but you are using that particular instruction much more. So, in that case you will get a better result even if you are reducing certain CPI to a great extent, but that is not executed more. So, you see that type B is executing only the frequency of execution is 10%. So, in that case if you reduce it to 1.6, also you are not getting a better result compared to when you are reducing A to just 1.1. So, we came to the end of lecture 12 where we have seen that various things that affect a CPU performance. And next we will see in some more detail in the next lecture.