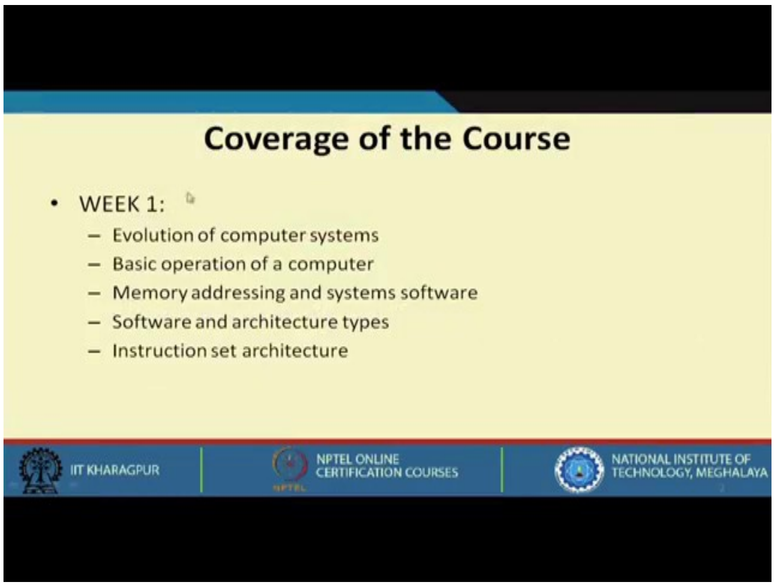
Computer Architecture and Organization

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 64
Summarization of the Course

So, we have at last come to the end of this course on computer architecture and organization. Over the last 12 weeks we have discussed the various concepts that are pertaining to computer organization and architecture. We have looked at some of the basic concepts. We have also looked at some of the advanced concepts that are there in many of the modern day processors. Before we say good bye to you, we would like to summarize the coverage of this course as we have carried out over the different weeks. I will be requesting Dr. Datta to talk about the first 6 weeks of the course that she had handled.

(Refer Slide Time: 01:13)



Coverage of the Course

- WEEK 1:
  - Evolution of computer systems
  - Basic operation of a computer
  - Memory addressing and systems software
  - Software and architecture types
  - Instruction set architecture

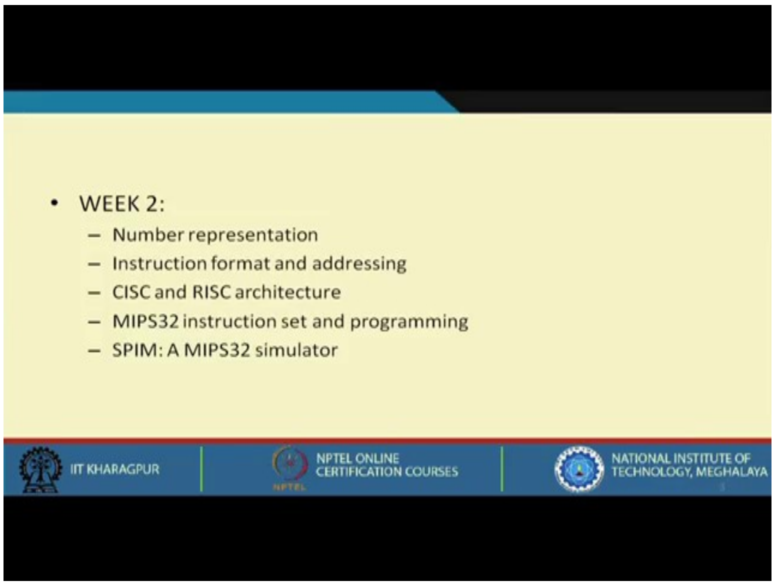
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So we have covered 12 weeks in total. Initially we started with the evolution of computer systems where we looked into how computers have evolved over the years, and of course where we stand today. In fact, the last lecture that you have heard that would have given you a picture of where we are now. We have also discussed about the basic operations of a computer, how a particular instruction is executed, what are the operations a computer performs, various memory addressing and system softwares. Like when we say that we are executing an instruction, we have shown that we store the instructions in memory. One by one we bring those instruction through some registers to our processor and then we execute it. In this process we require two important registers that is memory address register and memory buffer register or memory data register.

Then we talked about various architecture types, like von Neumann architecture and Harvard architecture. If you in von Neumann architecture we discussed about that how both program and data are stored alongside, and each time processor needs to access a particular data or instruction it fetches from memory, bring it to the processor and then it executes. But if you want to access either the instruction or the data you have to hit the same memory, because we are storing both the data and the program in the same memory. This kind of architecture is called von Neumann architecture. And we also discussed about Harvard architecture. We have seen that in Harvard architecture we store the data and instruction in two separate memories.

We have seen the advantages of using both Harvard architecture as well as von Neumann architecture. And if you think of that, going from the first lecture to the last lecture, some way or the other you can relate these two basic types of architectures. We then discussed about instruction set architecture or ISA, which is the programmer's view of the computer. How the programmer actually sees the computer. Like when we say a programming model, we can actually bring out how we can execute an instruction step by step, what steps are required. So, ISA is the programmer view of the computer.

(Refer Slide Time: 04:53)



- WEEK 2:
  - Number representation
  - Instruction format and addressing
  - CISC and RISC architecture
  - MIPS32 instruction set and programming
  - SPIM: A MIPS32 simulator

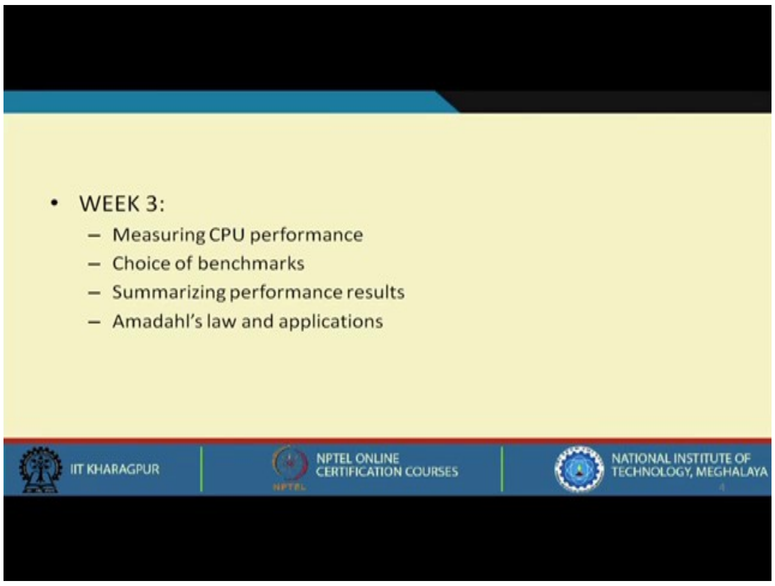
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Then we moved on with week 2, where we have briefly discussed about number representation that was used in later weeks on computer arithmetic.

We discussed the very important aspect of instruction format and addressing modes. What do you mean by instruction format?s An instruction consist of 2 parts, opcode and operand. And we have seen specifically for MIPS architecture how the instruction format is. The various addressing modes that are used today and that exist we also discussed about that. And we discussed about CISC and RISC architectures, and as a case study we have taken the MIPS32 instruction set architecture.

Then we also discussed about SPIM that is a MIPS32 simulator. Through examples we have shown how we can write an assembly language code, and execute it on SPIM.

(Refer Slide Time: 06:58)

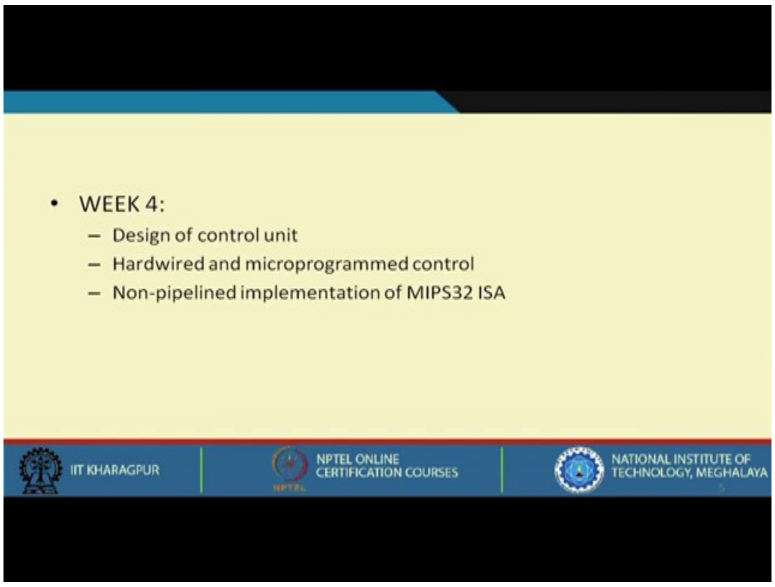

- WEEK 3:
  - Measuring CPU performance
  - Choice of benchmarks
  - Summarizing performance results
  - Amadahl's law and applications

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

Then we moved on with week 3, where we actually discussed about the quantitative approaches in evaluating the design alternatives.

By this what we mean is that these are very important concepts in modern-day technology. And here we actually show that how we can say that a computer A is better than computer Ba. ALso, what are the design principles that are followed.

(Refer Slide Time: 07:43)


- WEEK 4:
  - Design of control unit
  - Hardwired and microprogrammed control
  - Non-pipelined implementation of MIPS32 ISA

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

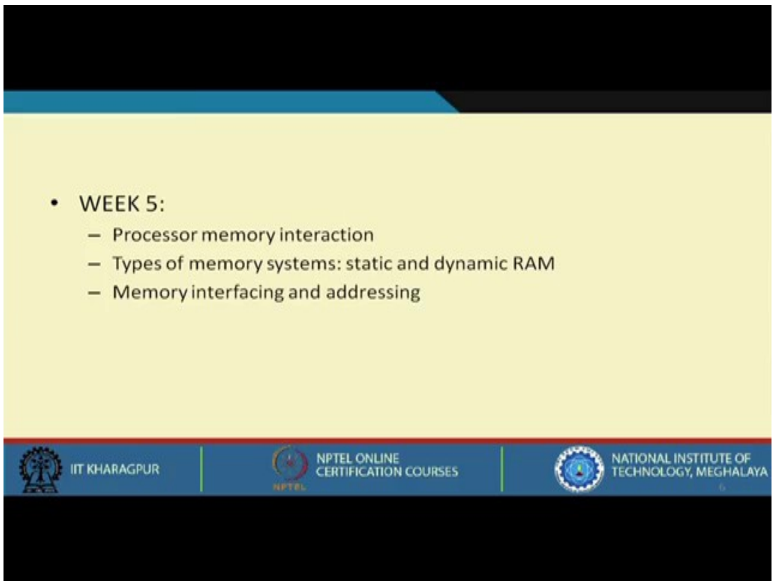
And then we have taken many examples to evaluate it, and the famous Amdahl’s law and its applications we have discussed.

Then we moved on to week 4 where we discussed about the design of control unit. How the instructions get executed inside a processor. The concept of micro-instructions and micro-programs have been discussed.

And then we also looked into two different ways of designing control units, hardwired and micro-programmed. In hardwired control unit design we have seen that it is much faster and how the instructions that needs to be executed must be simpler in nature. And in micro-programmed control unit design complex instructions can be executed in some way or other. We have a processor inside a processor. The micro-programmed control unit fetches some codes or micro-instructions from a small memory, and then executes them one by one.

And then we have discussed the non-pipelined implementation of MIPS32 instruction set architecture.

(Refer Slide Time: 09:05)

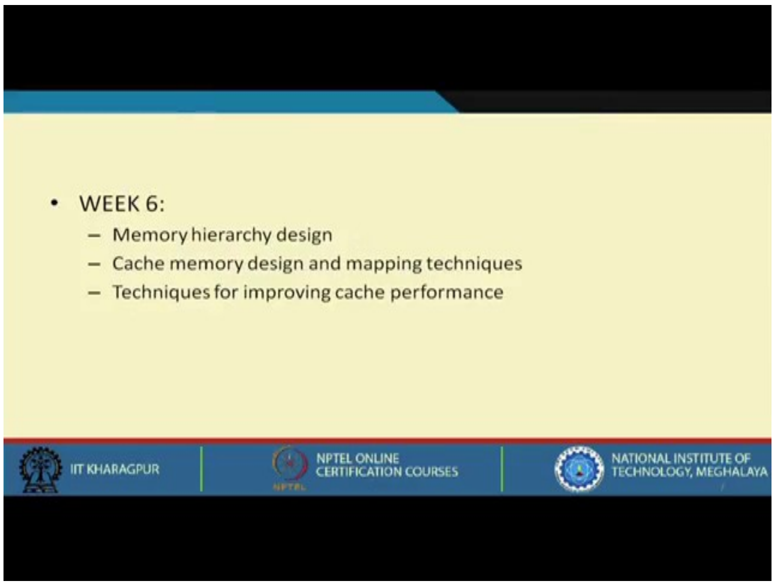


- WEEK 5:
  - Processor memory interaction
  - Types of memory systems: static and dynamic RAM
  - Memory interfacing and addressing

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

Then we moved on with week 5 where we discussed about processor memory interaction. We discussed about the various memory technologies that are used today. And what are the types of memory that are there, we discussed thoroughly about memory interfacing. We cannot use a very large memory module, how a large number of smaller memory modules can be combined together to form a large memory. We also discussed about memory interleaving.

(Refer Slide Time: 09:57)


- WEEK 6:
  - Memory hierarchy design
  - Cache memory design and mapping techniques
  - Techniques for improving cache performance

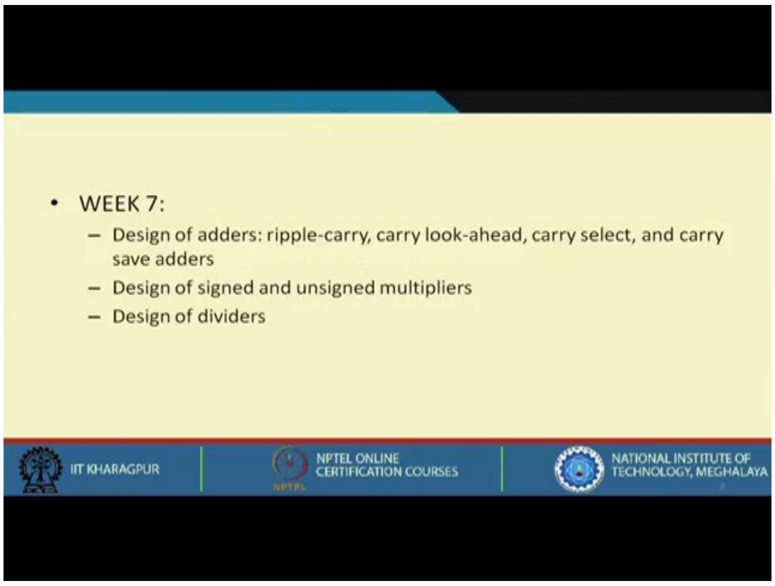
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Then we moved on with week 6 where we discussed about memory hierarchy design. A memory hierarchy design means how we are moving from one level of the memory to the next level, and so on.

If you think of a processor you have some registers inside. Register is some kind of memory or storage unit. Then you move down to the next level that is cache memory. Cache is a kind of memory that sits between the processor and the next level of memory, which is main memory. The frequently accessed instructions or data are brought from main memory to cache memory, and then they are sent to the processor. The next level after main memory is either disk or any other memory systems that are there.

This is how the memory hierarchy is there. And one important thing if you look into the memory hierarchy, when we move from registers then to cache memory then to the next level of main memory and then to disk, the size is always increasing. And size is increasing to a great extent, but the speed is also decreasing. So, the registers are the fastest then the cache memory then the main memory and then the next secondary memory. We also discussed about various techniques for improving cache performance, we discussed about various cache mapping techniques because if a memory is large we are bringing some data from a large memory to a small memory, we need to understand which data we are bringing where.

(Refer Slide Time: 12:06)


- WEEK 7:
  - Design of adders: ripple-carry, carry look-ahead, carry select, and carry save adders
  - Design of signed and unsigned multipliers
  - Design of dividers

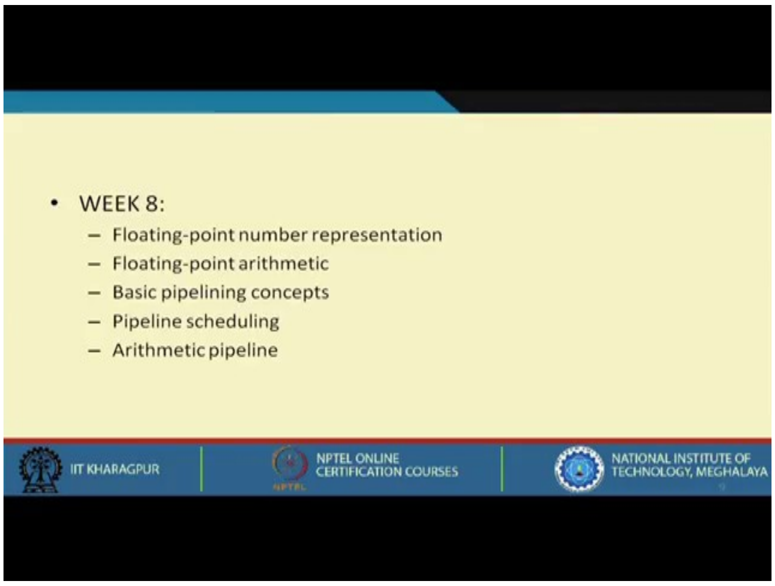
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

The mapping techniques are very important and we discussed in detail about all those techniques. And finally, we discussed about the various improvement strategies.

In week 7 we started our discussion on arithmetic circuits, because you know the computers are supposed to carry out some calculations, and in that sense the ALU can be considered as the heart of the system. The actual calculations are carried out there. In week 7 we started by discussing the various kinds of adders. We discussed the designs of ripple carry adder, carry lookahead adder, carry select adder and also carry save adders. Then we moved on to the design of signed and unsigned multipliers, we looked at the shift and add kind of multiplication, we looked at Booths multiplier then also we looked at some kind of faster multiplier using combinational logic blocks, for example, the carry save multiplier.

And lastly we looked at the design of various kinds of dividers. Specifically we discussed two algorithms, for restoring and non-restoring division. And also we discussed what kind of hardware is required to implement these kinds of division algorithms.

(Refer Slide Time: 13:50)



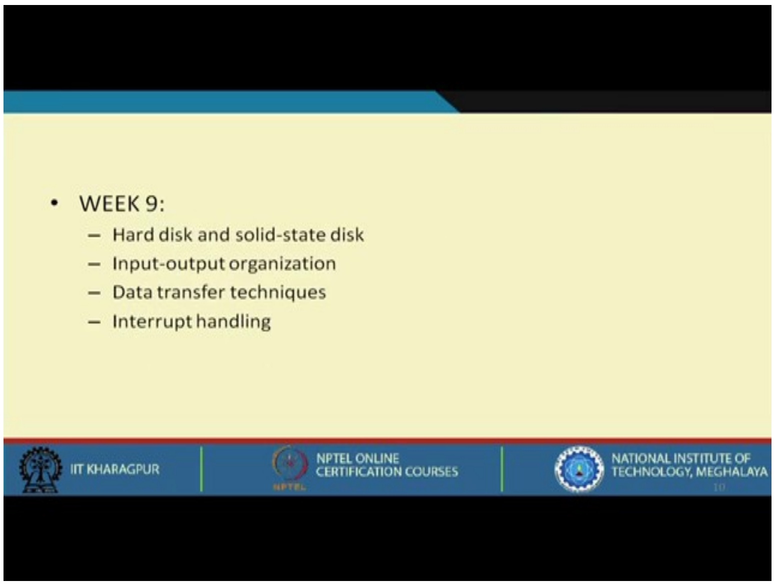
- WEEK 8:
  - Floating-point number representation
  - Floating-point arithmetic
  - Basic pipelining concepts
  - Pipeline scheduling
  - Arithmetic pipeline

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

Then we continued to week 8 where we moved on to the floating-point numbers. Since we did not discuss floating-point number representation earlier, we started by discussing the floating-point number format, in particular the IEEE standard. Then we talked about how we can carry out floating-point addition, subtraction, multiplication, division and what kind of hardware requirements are there, what kind of additional processing steps are there. Our objective later on was to explore how these arithmetic circuits can be implemented in an arithmetic pipeline so as to run them faster with lower latency and higher throughput.

We looked at the basic pipelining concepts first, what are the different kinds of pipelines, how we can schedule the pipeline so that collisions do not occur. Then we explored how we can map some of the arithmetic algorithms like floating-point addition, floating -point multiplication into a pipeline. One thing we also mentioned is that getting a pipeline implementation for division operation though not impossible is very difficult.

(Refer Slide Time: 15:47)



- WEEK 9:
  - Hard disk and solid-state disk
  - Input-output organization
  - Data transfer techniques
  - Interrupt handling

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

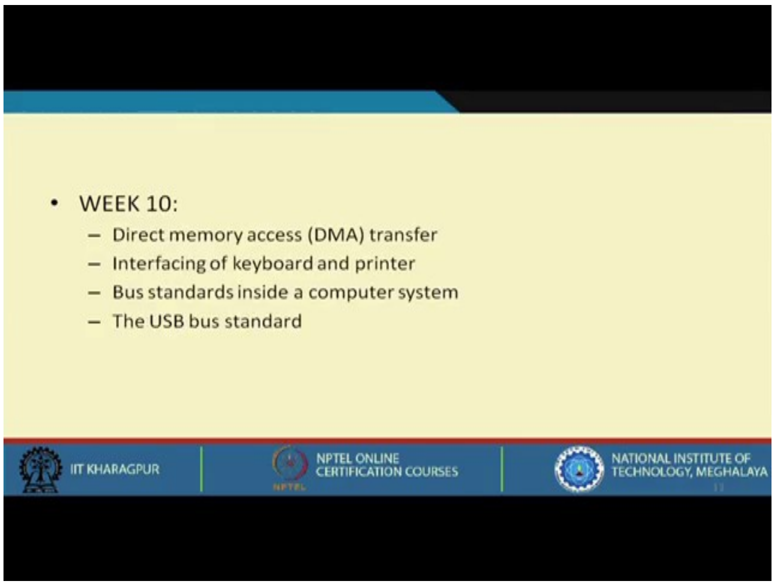
The pipeline implementation of dividers requires a lot of hardware. So, there are many processors where division is often not pipelined.

Then in week 9 we moved on to the input/output part of the processing. We started with the most important kinds of secondary memory devices that we use, namely the hard disk. And of course, the computing technology that is called solid-state disk, which is very fast replacing hard disks. Now-a-days if you buy laptops you often come with an option to get SSDs instead of hard disks. Because in SSDs there are no moving parts, while computing even if you shake your laptop nothing will happen, which was a danger for hard disk because there is a mechanical moving part. So, if there any jerks there is a chance of the hard disk getting damaged.

We moved on to the various input/output organizational issues. How devices can be connected to the processor, the concept of input/output port, the concept of memory mapped I/O interfacing, then the concept of I/O mapped I/O interfacing. We worked out a lot of examples to illustrate the basic ideas on these concepts. Then we started our discussion on various data transfer techniques that are possible. Broadly we said there are two methods, programmed and direct memory access. We discussed mostly the programmed I/O concepts, mainly the synchronous I/O transfer, then the asynchronous I/O transfer or hand shaking, and thirdly interrupt driven I/O transfer.

In particular we discussed interrupt processing in some detail where we talked about how multiple device interrupts can be handled, how interrupt priorities can be handled, the concept of interrupt nesting, the concept of identifying the interrupting device using the concept of interrupt vector or some other thing.

(Refer Slide Time: 18:16)



- WEEK 10:
  - Direct memory access (DMA) transfer
  - Interfacing of keyboard and printer
  - Bus standards inside a computer system
  - The USB bus standard

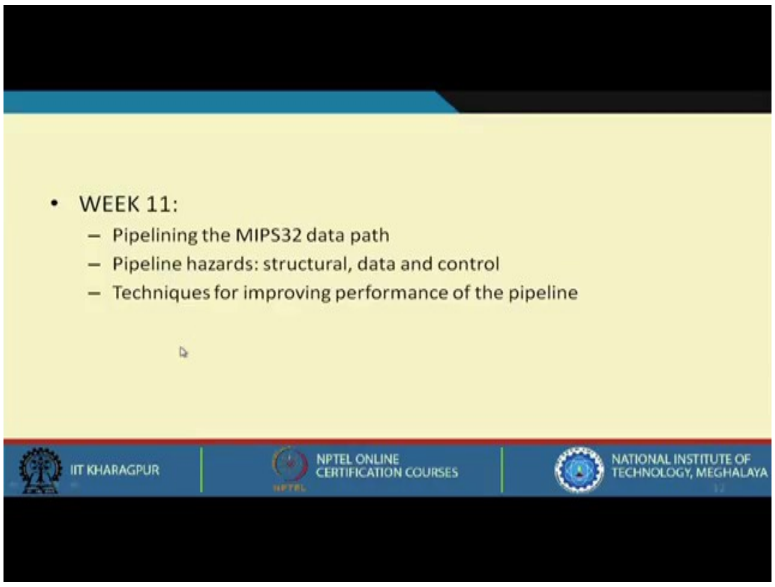
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

In week 10 we first discussed DMA transfer that is supposed to be very fast. Instead of the processor executing an instruction to transfer data between an I/O device and memory, here there is an external hardware controller called the DMA controller that takes hold of the memory bus and directly transfers data between I/O and memory without CPU intervention. Naturally all data transfer can be carried out at hardware speed without any instructions being executed.

Then we looked at some example interfacing, we looked at the examples of how we can interface a keyboard and a printer and we discussed how various methods like asynchronous interrupt driven methods can be used along with these kind of devices. For more complex devices very similar things can be used, but for the sake of illustrations we took two examples that are sufficiently simple and easy to understand.

Then we looked at the various bus standards that exist inside a computer system. Within a computer system if you look at an Intel motherboard, there are so many kinds of buses. These buses are often segregated using some bridges, there is a concept of north bridge south bridge; all the various internal buses are connected to one of these bridges. Then you have the external bus standard like USB on universal serial bus. Earlier we had many different standards, but today out of a conscious effort by the computer manufacturers we have zeroed down to a single standard, the USB, which can cater to the requirements of various categories of I/O devices, from very slow ones like keyboard and mouse to very fast ones like hard disk.

(Refer Slide Time: 20:35)



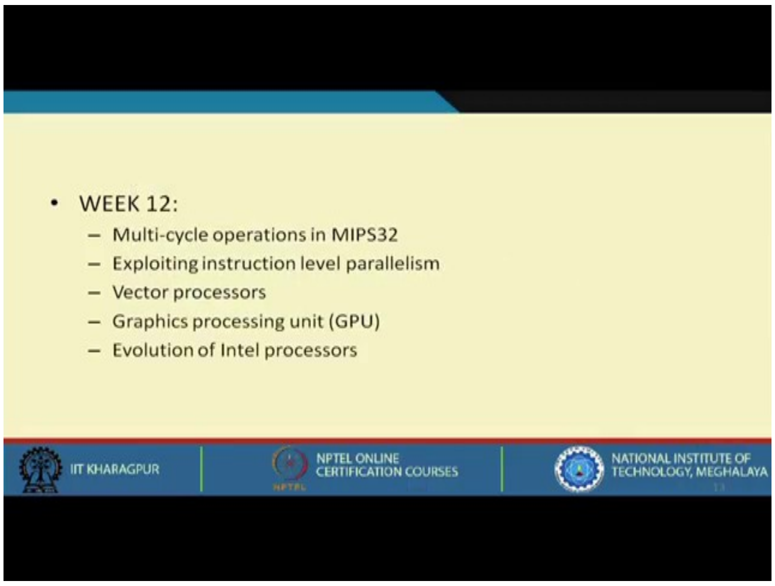
- WEEK 11:
  - Pipelining the MIPS32 data path
  - Pipeline hazards: structural, data and control
  - Techniques for improving performance of the pipeline

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

During week 11 when we explored the MIPS32 data path again, earlier Dr. Datta discussed the hardware implementation of the MIPS32 instruction set, but here we discussed how we can implement MIPS32 instruction execution in a pipeline. Specifically we talked about a 5 stage pipeline implementation. Because of the simplicity of the instruction set the pipeline implementation is also seen to be very simple. Then you looked at various different kinds of pipeline hazards, that limit is the maximum speedup that can be achieved in a pipeline. We talked about structural hazards, data hazards, control hazards, what kind of penalties they incur and we also suggested various techniques by which these penalties can be reduced.

Various predictive branch performance improvements were discussed to improve the performance of a pipeline in general. Some of these techniques are actually used in modern computer systems. Though we can call them as advanced topics, but any state-of-the-art processor we see use all these techniques in a very big way.

(Refer Slide Time: 22:19)


- WEEK 12:
  - Multi-cycle operations in MIPS32
  - Exploiting instruction level parallelism
  - Vector processors
  - Graphics processing unit (GPU)
  - Evolution of Intel processors

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

In the last week, we talked about the multi-cycle operations in MIPS32, how we can extend the MIPS32 pipeline to handle operations that may require more than 5 cycles. Particularly in the EX stage for floating-point operations, it can require more than one cycles to run. Similarly multiplication or division may require multiple cycles to run in the EX stage. These are something called multi-cycle operations. We specifically discussed how multi-cycle operations can be handled, how data hazards are tackled in such a scenario, how interrupts and exception processing can be done.

Then we looked at how we can exploit instruction level parallelism by using techniques like loop unrolling, which is a compiler technique. Given a program, the compiler will try hard to expose more parallelism by a concept called loop unrolling, where copies of the loop are made multiple times so that that more parallelism can be exploited.

We also discussed vector processors, and how we can extend the basic MIPS32 pipeline to incorporate processing of vectors of data. Here we able to avoiding a loop, translating it into a single instruction, that is of course much more efficient. Then we looked at some of the case studies. We looked at in particular the GPU and then looked at how Intel processors have evolved over the years.

This was roughly the total course coverage that we have gone through over the last 12 weeks. Let me wish you all the best.

So, I request Dr Datta to also say a few words.

As we told in the introductory video we will be starting with the very basics and then move on to some advanced concepts. I think we have done that in these 12 weeks. And I hope that you have enjoyed this course.

Thank you.