MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 01

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Introduction

Introduction:

Functional units are a part of a CPU that performs the operations and calculations called for by the computer program. Functional units of a computer system are parts of the CPU (Central Processing Unit) that performs the operations and calculations called for by the computer program.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Evolution of Computer Systems
- Basic Operation of a Computer

Detailed content of the Lecture:

FUNCTIONAL UNITS

A computer consists of 5 functionally independent main parts:

1) Input
2) Memory
3) ALU
4) Output &
5) Control units.

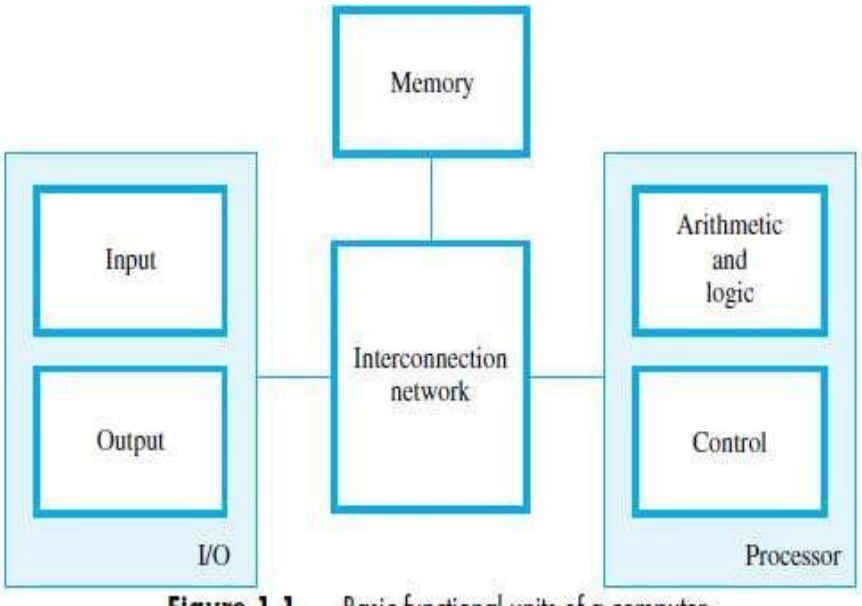


Figure 1.1 Basic functional units of a computer.

Input Units are used by the computer to read the data. The most commonly used input devices are keyboards, mouse, joysticks, trackballs, microphones, etc.

Memory unit can be referred to as the storage area in which programs are kept which are running, and that contains data needed by the running programs. The Memory unit can be categorized in two ways namely, primary memory and secondary memory.

ALU, most of all the arithmetic and logical operations of a computer are executed in the ALU (Arithmetic and Logical Unit) of the processor. It performs arithmetic operations like addition, subtraction, multiplication, division and also the logical operations like AND, OR, NOT operations.

Outputs are pieces of equipment that are used to generate information or any other response processed by the computer. These devices display information that has been held or generated within a computer. The most common example of an output device is a monitor.

Control Unit is a component of a computer's central processing unit that coordinates the operation of the processor. It tells the computer's memory, arithmetic/logic unit and input and output devices how to respond to a program's instructions. The control unit is also known as the nerve center of a computer system.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=bjFHDOecebI

https://www.youtube.com/watch?v=tAxSbdxwcw4

https://www.youtube.com/watch?v=Ot09heN1rVI

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 3-10

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 02

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Technologies for Building Processors – Performance – Power wall

Introduction:

- The computer generation means by step by step growth in the technology.
- Each phase of computer has been extended both incude hardware and software, together make up an entire computer system.
- In each phase development in the technology, the electronic device is utilised.
- Computer Generations are divided into five generations depending upon their Technologies.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Evolution of Computer Systems
- Basic Performance of a Computer

Detailed content of the Lecture:

- **Technology:**
  - Integrated circuit logic technology
  - Magnetic disc technology
  - Network technology
- **Integrated circuit logic technology:**

A growth in transistor count on chip of about 40 % to 55 % per year.

- **Magnetic disc technology:**

In 1990's disk density had been improving 60% to100% per year, while prior to 990 about30% per year. Since 2004, it dropped back to 30% per year.

- **Network technology:**

Latency and bandwidth are important. Internet infrastructure in the U.S. has been doubling in bandwidth every year. High performance Systems Area network delivering continuous reduced latency.

PERFORMANCE OF THE COMPUTER:

Speed

- Important measure of the performance of a computer.
- How quickly it can execute program.
- Speed Measure
  - Response time
    - Time between the start and completion of the task.
    - It is also referred as Execution time.
  - Throughput
    - Gives the total amount of work done in a given time.

Note: Reduction in response time increases throughput.

- Speed of the computer is increased in two ways:
  - Decreasing the response time.
  - Increasing the throughput.

MEASURING PERFORMANCE:

- Time-important measure of the performance of a computer.
  - Measured in terms of Seconds Per Program.
  - Defined as the total time to complete the task including
    - ✓ Disk access
    - ✓ Memory access
    - ✓ I/O activities
  - Also called as wall-clock time,response time or elapsed time.

POWER WALL

- For better performance, processor can run at high clock speed and it generate more heat and consume high power.
- If clock rate increases, power consume also increased.
- Power consumed by CPU is given by

$P=CV^2f$

where

- P is Power
- C is Capacitive loading
- V is voltage applied.

F is Frequency running

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=bjFHDOecebI

https://www.youtube.com/watch?v=tAxSbdxwcw4

https://www.youtube.com/watch?v=Ot09heN1rVI

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 24-42

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 03

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Operations of the Computer Hardware

Introduction:

An Instruction consists of 2 parts, 1) Operation code (Opcode) and 2) Operands. The data/operands are stored in memory. The individual instruction is brought from the memory to the processor.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Evolution of Computer Systems
- Basic Operation of a Computer

Detailed content of the Lecture:

Let us see a typical instruction

ADD LOCA, R0

This instruction is an **addition operation**. The following are the steps to execute the instruction:

Step 1: Fetch the instruction from main-memory into the processor.

Step 2: Fetch the operand at location LOCA from main-memory into the processor.

Step 3: Add the memory operand (i.e. fetched contents of LOCA) to the contents of register R0.

Step 4: Store the result (sum) in R0.

The same instruction can be realized using **2 instructions** as:

Load LOCA, R1

Add R1, R2

The following are **the steps to execute the instruction**:

Step 1: Fetch the instruction from main-memory into the processor.

Step 2: Fetch the operand at location LOCA from main-memory into the register R1.

Step 3: Add the content of Register R1 and the contents of register R0.

Step 4: Store the result (sum) in R0.

MAIN PARTS OF PROCESSOR

- The **processor** contains ALU, control-circuitry and many registers.
- The processor contains „n" general-purpose registers $\mathbf{R_0}$ through $\mathbf{R_{n-1}}$.
- The **IR** holds the instruction that is currently being executed.

- The **control-unit** generates the timing-signals that determine when a given action is to take place.
- The **PC** contains the memory-address of the next-instruction to be fetched & executed.
- During the execution of an instruction, the contents of PC are updated to point to next instruction.
- The **MAR** holds the address of the memory-location to be accessed.
- The **MDR** contains the data to be written into or read out of the addressed location.

STEPS TO EXECUTE AN INSTRUCTION

1) The address of first instruction (to be executed) gets loaded into PC.
2) The contents of PC (i.e. address) are transferred to the MAR & control-unit issues Read signal to memory.
3) After certain amount of elapsed time, the first instruction is read out of memory and placed into MDR.
4) Next, the contents of MDR are transferred to IR. At this point, the instruction can be decoded & executed.
5) To fetch an operand, it's address is placed into MAR & control-unit issues Read signal. As a result, the operand is transferred from memory into MDR, and then it is transferred from MDR to ALU.
6) Likewise required number of operands is fetched into processor.
7) Finally, ALU performs the desired operation.
8) If the result of this operation is to be stored in the memory, then the result is sent to the MDR.
9) The address of the location where the result is to be stored is sent to the MAR and a Write cycle is initiated.
10) At some point during execution, contents of PC are incremented to point to next instruction in the program.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=bjFHDOecebI
https://www.youtube.com/watch?v=tAxSbdxwcw4
https://www.youtube.com/watch?v=Ot09heN1rVI

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 63-67

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 04,05

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Operands, Signed and Unsigned Numbers

Introduction:

An Instruction consists of 2 parts, 1) Operation code (Opcode) and 2) Operands. The data/operands are stored in memory. The individual instruction is brought from the memory to the processor.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Evolution of Computer Systems
- Basic Operation of a Computer

Signed and Unsigned Numbers

- First, let's quickly review how a computer represents numbers.
- Humans are taught to think in base 10, but numbers may be represented in any base.
- For example,

123 base 10 = 1111011 base 2.

i.e.,(123)10 = (1111011)2

- Numbers are kept in computer hardware as a series of high and low electronic signals, and so they are considered base 2 numbers. (Just as base 10 numbers are called decimal numbers, base 2 numbers are called binary numbers).

Decimal number Representation

- **Unsigned numbers : Magnitude or +with Magnitude (value)**
  Ex: 746 or +746
- **Signed numbers : Sign with Magnitude**
  Ex: -46.25

Binary Numbers Representation

- Computer must represented everything with binary digits.
- Unsigned numbers – Magnitude Only
  Ex: 10010
- Signed numbers Sign bit with Magnitude bits
  0 mean positive (0 Positive)
  1 mean negative (1 Negative)

- Ex: 9 = 01001 MSB 0 is sign bit and 1001 is the magnitude bits
- Ex: -9 = 11001 MSB 1 is sign bit and 1001 is the magnitude bits

❖ **Representation of Signed Binary Numbers**

- There are three types of representations for signed binary numbers
    1. Sign-Magnitude form
    2. 1's complement form
    3. 2's complement form

1.Sign-Magnitude form

The convention is to make the sign bit

0 + ve

1 - ve

Positive decimal number +108

Ex: +10810 = 011011002

Negative decimal number -108.

Ex: -10810 = 111011002

2. 1's Complement form

- 1's complement of positive number gives a negative number. Similarly, 1's complement of negative number gives a positive number.
- The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's.

3. 2's Complement form

- The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.
  2's complement = 1's complement + 1

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=bjFHDOecebI

https://www.youtube.com/watch?v=tAxSbdxwcw4

https://www.youtube.com/watch?v=Ot09heN1rVI

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 75-82

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 06,07

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Representing Instructions

Introduction:

Instruction sequencing, the order in which the instructions in a program are carried out. Normally the sequence proceeds in a linear fashion through the program, and the address of the instructions is obtained from the program counter in the control unit.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory Locations & Addresses
- Memory Operations

Detailed content of the Lecture:

A computer must have instructions capable of performing 4 types of operations:

1) Data transfers between the memory and the registers (MOV, PUSH, POP, XCHG).
2) Arithmetic and logic operations on data (ADD, SUB, MUL, DIV, AND, OR, NOT).
3) Program sequencing and control (CALL.RET, LOOP, INT).
4) I/0 transfers (IN, OUT).

REGISTER TRANSFER NOTATION (RTN)

Location	Hardware Binary Address	Example
Memory	LOC, PLACE, NUM	R1 ← [LOC]
Processor	R0, R1, R2	[R3] ← [R1] +[R2]
I/O Registers	DATAIN, DATAOUT	R1 ← DATAIN

ASSEMBLY LANGUAGE NOTATION

Assembly Language Format	Description
Move LOC, R1	Transfer data from memory-location LOC to register R1. The contents of LOC are unchanged by the execution of this instruction, but the old contents of register R1 are overwritten.
Add R1, R2, R3	Add the contents of registers R1 and R2, and places their sum

	into register R3.

BASIC INSTRUCTION TYPES

Instruction Type	Syntax	Example
Three Address	Opcode Source1, Source2, Destination	Add A, B, C
Two Address	Opcode Source, Destination	Add A, B
One Address	Opcode Source/Destination	Load A
		Add B
		Store C
Zero Address	Opcode [no Source/Destination]	Push

INSTRUCTION EXECUTION & STRAIGHT-LINE SEQUENCING

The program is executed as follows:

1. Initially, the address of the first instruction is loaded into PC.
2. Then, the processor control circuits use the information in the PC to fetch and execute instructions, one at a time, in the order of increasing addresses. This is called *Straight-Line sequencing*.
3. During the execution of each instruction, PC is incremented by 4 to point to next instruction.

There are 2 phases for Instruction Execution:

1. **Fetch Phase:** The instruction is fetched from the memory-location and placed in the IR.
2. **Execute Phase:** The contents of IR are examined to determine which operation is to be performed. The specified-operation is then performed by the processor.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=dPa4T_EZ7Gs
https://www.youtube.com/watch?v=SFsnysyVhzA
https://www.youtube.com/watch?v=gfFgPvEDNYU

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 82-89

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 08

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Logical operations

Introduction:

- Instructions that perform logical operations which manipulate Boolean values.

Normally the sequence proceeds in a linear fashion through the program, and the address of the instructions is obtained from the program counter in the control unit.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory Locations & Addresses
- Memory Operations

Detailed content of the Lecture:

❖ **AND instruction**

- Contains three register operands
- Perform Bitwise AND operation between two source registers and stores theresult in the destination register
- Syntax :

Operation destination, sourcel, source2

- Example

AND $1, $2, $3 is equivalent to $1= $2 & $3

❖ **OR instruction**

- Contains three register operands
- Perform Bitwise OR operation between two source registers and stores the result in the destination register
- Syntax

  Operation destination, sourcel, source2
- Example

  OR $1, $2, $3 is equivalent to $1 = $2 | $3

❖ **NOR instruction**

- Contains three register operands
- Perform Bitwise NOR operation between two source registers and stores the result in the destination register

- Syntax
  Operation destination, sourcel, source2..
- Example
  NOR $1, $2, $3 is equivalent to $1 = ~($2|$3)

❖ **AND Immediate (ANDI) instruction**

- Contains three register operands
- Perform Bitwise AND operation between a source registers and immediate values
  - Stores the result in the destination register
- Syntax
  - Operation destination, source1, Imme
- Example
  AND $1, $2, imme is equivalent to $1 = $2 & imme

❖ **OR Immediate (ORI) instruction**

- Contains three register operands
- Perform Bitwise OR operation between a source registers and immediate values
  - Stores the result in the destination register
- Syntax
  - Operation destination, source1, Imme
- Example
  OR $1, $2, imme is equivalent to $1 = $2 | imme

❖ **Shift Left Logical instruction**

- Contains three register operands
- Shifts a register value left by the shift amount listed in the instruction and places the result in a third register. Zeroes are shifted in
- Syntax
  - Operation destination, source1, Constant
- Example
  sll $1, $2, 10 is equivalent to $1 = $2 <<10

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=XNZVxWLzWmk
https://www.youtube.com/watch?v=be9q_9Hcips
https://www.youtube.com/watch?v=fMgeZiWkPLA

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 90-93

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 09

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : I

Date of Lecture:

Topic of Lecture: Instructions for making Decisions

Introduction:

➢ Instructions that perform control operations which manipulate Boolean values.

Normally the sequence proceeds in a linear fashion through the program, and the address of the instructions is obtained from the program counter in the control unit.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory Locations & Addresses
- Memory Operations

Detailed content of the Lecture:

Conditional Branch

BEQ Instruction

- ➢ BEQ - Branch on-equal
- ➢ Branches if the two registers are equal
- ➢ Syntax
  - Operation sourcel, source2, offset
- ➢ Example

beq $s, $t, offset

is equivalent to

if ($-$t) goto LI;

a=btc;

LI: a=b-c;

BNE Instruction

- ➢ BNE - Branch on Not Equal
- ➢ Brunches if the two registers are not equal
- ➢ Syntax
  - Operation sourcel, sourcc2, offset
- ➢ Example

bne $s $t, offset

is equivalent to

if (Ss -81) goto Ll;

a=btc;

Ll: a-b-c;

UNCONDITIONAL BRANCH

J Instruction

- J- Jump
- Jumps to the calculated address
- Syntax
  - Operation offiset
- Example

j terget;

JAL Instruction

- JAL – Jump and link
- Jumps to the calculated address and stores the return address
- Syntax
  - Operation offset
- Example

jal target;

JR Instruction

- JR-Jump register
- Jump to the address contained in register $s
- Syntax
  - Operation source
- Example

jal $s;

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=XNZVxWLzWmk

https://www.youtube.com/watch?v=be9q_9Hcips

https://www.youtube.com/watch?v=fMgeZiWkPLA

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017

Page no: 97-99

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 10

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: MIPS Addressing for 32-Bit Immediate and Addresses

Introduction:

- LEGv8 instructions 32 bits long simplifies the hardware, there are times where it would be convenient to have 32-bit or larger constants or addresses.
- The LEGv8 instruction set includes
  1. move wide with zeros (MOVZ)
  2. move wide with keep (MOVK)
- It specifically to set any 16 bits of a constant in a register.

Learning objective:

To understand the arithmetic and logic unit and implementation of fixed point and floating- Point arithmetic operations

Prerequisite knowledge for Complete understanding and learning of Topic:

- LEGv8 instructions 32 bits
- arithmetic and logic unit

Detailed content of the Lecture:

Loading a 32-Bit Constant

What is the LEGv8 assembly code to load this 64-bit constant into register X19?

```
00000000 00000000 00000000 00000000 00000000 00111101 00001001 00000000
```

First, we would load bits 16 to 31 with that bit pattern, which is 61 in decimal, using MOVZ:

```
MOVZ     X19, 61, LSL 16 // 61 decimal = 0000 0000 0011 1101 binary
```

The value of register X19 afterward is:

```
00000000 00000000 00000000 00000000 00000000 00111101 00000000 00000000
```

The next step is to insert the lowest 16 bits, whose decimal value is 2304:

```
MOVK     X19, 2304, LSL 0 // 2304 decimal = 00001001 00000000
```

The final value in register X19 is the desired value:

```
00000000 00000000 00000000 00000000 00000000 00111101 00001001 00000000
```

Addressing in Branches & Jumps

- It have the simplest addressing
- It is also called the B-type, which consists of 6 bits for the operation field and the rest of the bits for the address field

General syntax

B 10000 // go to location 10000_{ten}

- Unlike the branch instruction, a conditional branch instruction can specify one operand in addition to the branch address

CBNZ X19, Exit // go to Exit if X19 ≠ 0

- a branch instruction would calculate the following:

Program counter =Register +Branch offset

MIPS Addressing modes:

1.Immediate addressing
2.Register addressing
3.Base addressing
4.PC-relative addressing
5.Pseudo-direct addressing

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=o-WXqnagg0c
https://www.youtube.com/watch?v=bQOOsrE11gc&list=TLPQMTgwMzIwMjCJ7I1Z1z9q8w&index=2
https://www.youtube.com/watch?v=RsoiKAI_ndM

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 115-125

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 11

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Parallelism and Instructions: Synchronization

Introduction:

- Parallel execution is easier when tasks are independent, but often they need to cooperate. Cooperation usually means some tasks are writing new values that others must read.

Learning objective:

To understand the arithmetic and logic unit and implementation of fixed point and floating- Point arithmetic operations

Prerequisite knowledge for Complete understanding and learning of Topic:

- Parallelism and Instructions
- Synchronization

Detailed content of the Lecture:

- Parallel execution is easier when tasks are independent, but often they need to cooperate. Cooperation usually means some tasks are writing new values that others must read.
- Two operations

  1.lock operation

  2.unlock operation
- It is also called a mutual exclusion
- **atomic exchange or atomic swap**

  It interchanges a value in a register for a value in memory.

Ex,(ll,sc)

Where,

ll-load link

sc-store conditional

store conditional is to achieve this pneumonic code

- The exchange is indivisible, and two simultaneous exchanges will be ordered by the hardware. It is impossible for two processors trying to set the synchronization variable in this manner to both think they have simultaneously set the variable.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=o-WXqnagg0c

https://www.youtube.com/watch?v=bQOOsrE11gc&list=TLPQMTgwMzIwMjCJ7I1Z1z9q8w&index=2

https://www.youtube.com/watch?v=RsoiKAI_ndM
Important Books/Journals for further learning including the page nos.: David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017 Page no: 125-128

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 12

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Translating and Starting a Program

Introduction:

A compiler is a program that reads a program written in one lang(source lang) and translate it into an equialent program in anotherlang(target lang).

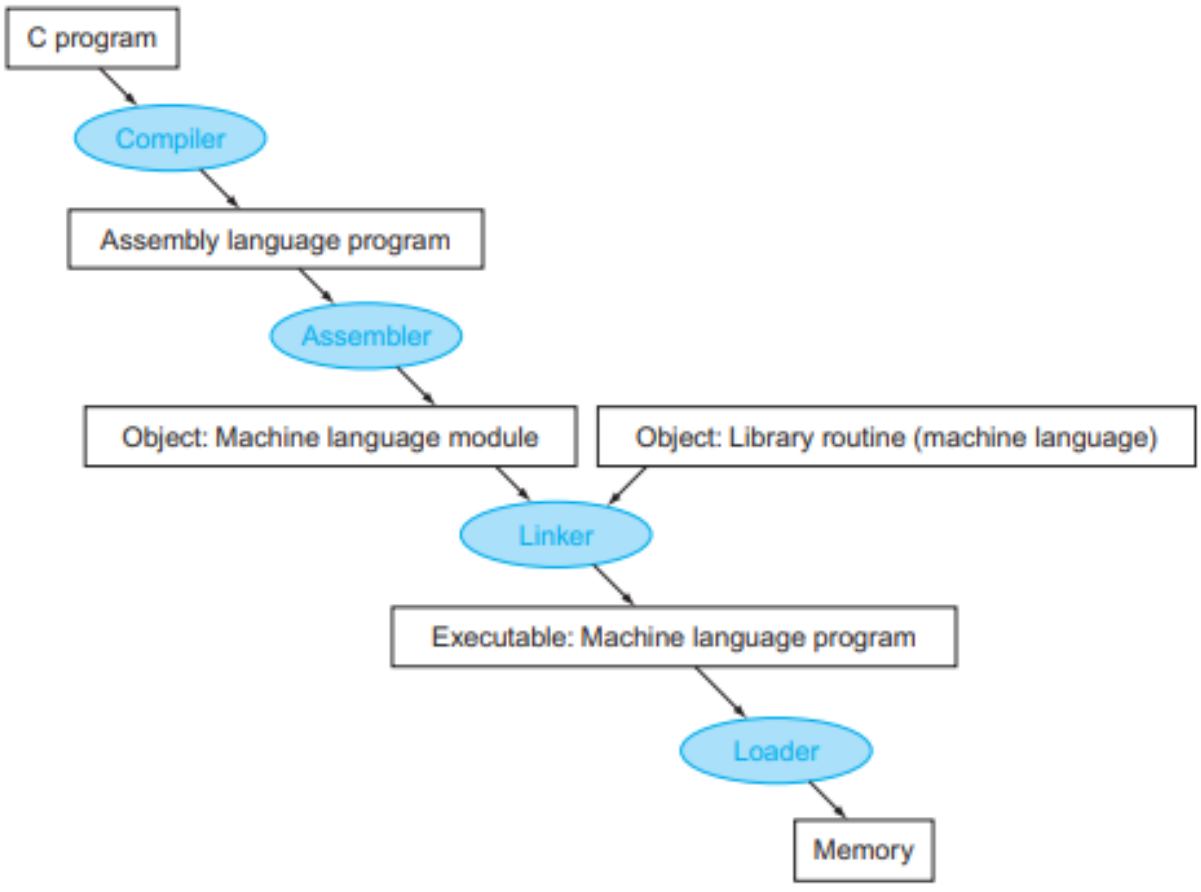
Learning objective:

To understand the arithmetic and logic unit and implementation of fixed point and floating- Point arithmetic operations

Prerequisite knowledge for Complete understanding and learning of Topic:

- Compiler , Linker
- Loader , Assembler

Detailed content of the Lecture:



There are

1.object file header –Position,size

2.text segment –binary transform

3.static data segment –contains static data

4.relocation information-identifies instructions and data words

5.symbol table-such as external references

6.debugging information-error find out

linker Also called link editor.

- A systems program that combines independently assembled machine language programs and resolves all undefined labels into an executable file.
- Link-editor - The linker resolves external memory addresses, where the code in one file may refer to a location in another file, so the relocatable machine code may have to be linked together with other relocatable object files and library files.It is done by linker

There are three steps for the linker:

- **1. Place code and data modules symbolically in memory.**
- **2. Determine the addresses of data and instruction labels.**
- **3. Patch both the internal and external references.**

Loader

- **Loader: A systems program that places an object program in main memory so that it is ready to execute.**

The loader follows these steps in UNIX systems:

- **1. Reads the executable file header to determine size of the text and data segments.**
- **2. Creates an address space large enough for the text and data.**
- **3. Copies the instructions and data from the executable file into memory.**
- **4. Copies the parameters (if any) to the main program onto the stack.**
- **5. Initializes the processor registers and sets the stack pointer to the first free location.**
- **6. Branches to a start-up routine that copies the parameters into the argument registers and calls the main routine of the program. When the main routine returns, the start-up routine terminates the program with an exit system call.**

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=o-WXqnagg0c

https://www.youtube.com/watch?v=bQOOsrE11gc&list=TLPQMTgwMzIwMjCJ7I1Z1z9q8w&index=2

https://www.youtube.com/watch?v=RsoiKAI_ndM

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 125-128

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 13

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Addition and subtraction

Introduction:

Numbers can be represented in 3 formats:

1) Sign and magnitude
2) 1's complement
3) 2's complement

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Addition and Subtraction

Detailed content of the Lecture:

In **sign-and-magnitude system**, negative value is obtained by changing the MSB from 0 to 1 of the corresponding positive value.

For ex, +5 is represented by 0101 & -5 is represented by 1101.

In **1's complement system,** negative values are obtained by complementing each bit of the corresponding positive number.

For ex, -5 is obtained by complementing each bit in 0101 to yield 1010.

In **2's complement system**, forming the 2's complement of a number is done by subtracting that number from 2^n. 2's complement system yields the most efficient way to carry out addition/subtraction operations.

For ex, -5 is obtained by complementing each bit in 0101 & then adding 1 to yield 1011.

ADDITION OF POSITIVE NUMBERS

- Consider adding two 1-bit numbers.
- The sum of 1 & 1 requires the 2-bit vector 10 to represent the value 2. We say that sum is 0 and the carry-out is 1.

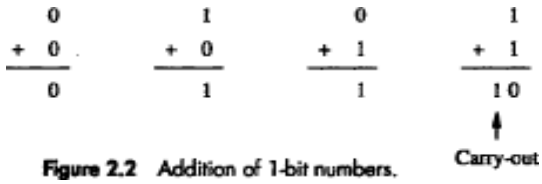


Figure 2.2 Addition of 1-bit numbers.

B	Values represented		
$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+7	+7	+7
0 1 1 0	+6	+6	+6
0 1 0 1	+5	+5	+5
0 1 0 0	+4	+4	+4
0 0 1 1	+3	+3	+3
0 0 1 0	+2	+2	+2
0 0 0 1	+1	+1	+1
0 0 0 0	+0	+0	+0
1 0 0 0	−0	−7	−8
1 0 0 1	−1	−6	−7
1 0 1 0	−2	−5	−6
1 0 1 1	−3	−4	−5
1 1 0 0	−4	−3	−4
1 1 0 1	−5	−2	−3
1 1 1 0	−6	−1	−2
1 1 1 1	−7	−0	−1

Figure 1.3 Binary, signed-integer representations.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=o-WXqnagg0c

https://www.youtube.com/watch?v=bQOOsrE11gc&list=TLPQMTgwMzIwMjCJ7I1Z1z9q8w&index=2

https://www.youtube.com/watch?v=RsoiKAI_ndM

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 188-191.

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 14

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Multiplication

Introduction:

The multiplication is a complex operation than addition and subtraction. It can be performed in hardware or software. A wide variety of algorithms have been used in various computers. For simplicity we will first see the multiplication algorithms for unsigned integers, and then we will see the multiplication algorithm for signed numbers.

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Addition and Subtraction

Detailed content of the Lecture: Detailed content of the Lecture:

Multiplication operation steps:

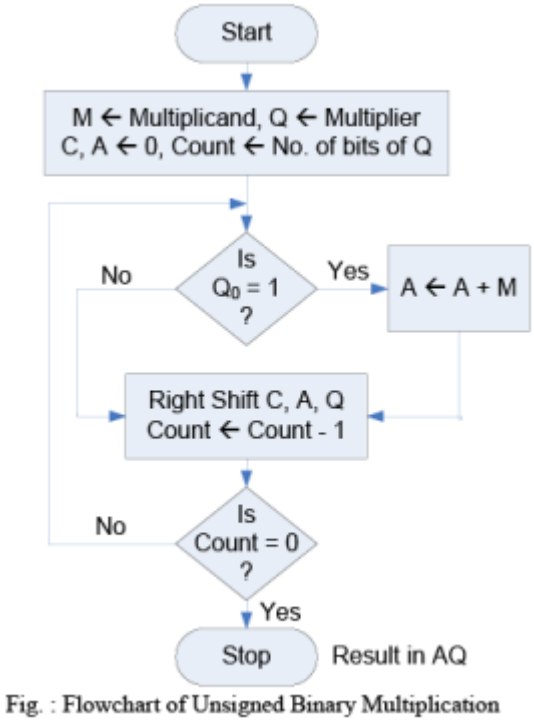
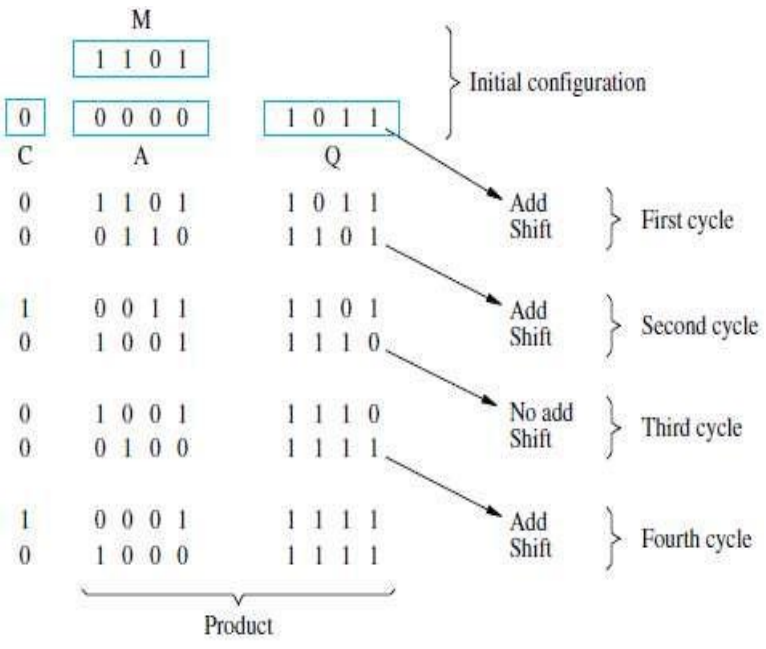


Fig. : Flowchart of Unsigned Binary Multiplication



(b) Multiplication example

Figure 9.7 Sequential circuit binary multiplier.

- Multiplication process involves generation of partial products, one for each digit in the multiplier. These partial products are then summed to produce the final product.
- In the binary system the partial products are easily defined. When the multiplier bit is 0, the partial product is 0, and when the multiplier is 1, the partial product is the multiplicand.
- The final product is produced by summing the partial products. Before summing operation, each successive partial product is shifted one position to left relative to the preceding partial product.
- The product of two n-digit numbers can be accommodated in 2n digits, so the product of the two 4-bit numbers in fits into 8-bits.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=B2bKdGf1Qoc
https://www.youtube.com/watch?v=b_azyJ4ZgVo
https://www.youtube.com/watch?v=XWnm1PyMgaY

Important Books/Journals for further learning including the page nos.:

https://nptel.ac.in/courses/106105163/
David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 188-191.

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 15,16

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Division

Introduction:

The division is more complex than multiplication. A **division** algorithm is an algorithm which, given two **integers** N and D, computes their quotient and/or remainder, the result of Euclidean **division**. Some are applied by hand, while others are employed by digital circuit designs and software.

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Addition and Subtraction
- Multiplication

Detailed content of the Lecture: Detailed content of the Lecture:

INTEGER DIVISION

- An n-bit positive-divisor is loaded into register M.
- An n-bit positive-dividend is loaded into register Q at the start of the operation.
- Register A is set to 0.
- After division operation, the n-bit quotient is in register Q, and the remainder is in register A.

```
      21                    10101
  ------                ---------
13 ) 274            1101 ) 100010010
     26                     1101
    ---                    -----
     14                     10000
     13                     1101
    ---                    -----
      1                      1110
                             1101
                            -----
                                1
```

Figure 9.22 Longhand division examples.

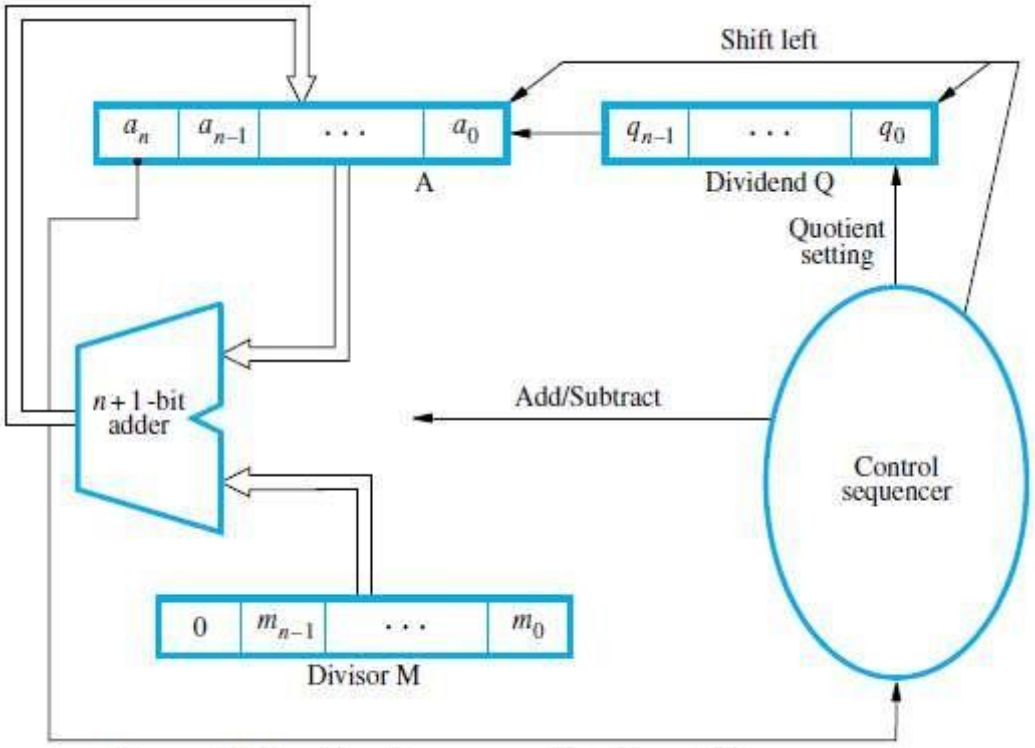


Figure 9.23 Circuit arrangement for binary division.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=F2ZS09T7ofE

https://www.youtube.com/watch?v=VKemv9u40gc

https://www.youtube.com/watch?v=ubCCemtuZH8

Important Books/Journals for further learning including the page nos.:

https://nptel.ac.in/courses/106105163/

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017

Page no: 197-205

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L - 17

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Floating Point

Introduction:

In this section we are going to see general procedures for addition, subtraction, multiplication and division of floating-point numbers.

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Addition and Subtraction

Detailed content of the Lecture:

ARITHMETIC OPERATIONS ON FLOATING-POINT NUMBERS

Multiply Rule

1) Add the exponents & subtract 127.
2) Multiply the mantissas & determine sign of the result.
3) Normalize the resulting value if necessary.

Divide Rule

1) Subtract the exponents & add 127.
2) Divide the mantissas & determine sign of the result.
3) Normalize the resulting value if necessary.

Add/Subtract Rule

1) Choose the number with the smaller exponent & shift its mantissa right a number of steps equal to the difference in exponents(n).
2) Set exponent of the result equal to larger exponent.
3) Perform addition/subtraction on the mantissas & determine sign of the result.
4) Normalize the resulting value if necessary.

IMPLEMENTING FLOATING-POINT OPERATIONS

• First compare exponents to determine how far to shift the mantissa of the number with the smaller exponent.

• The shift-count value n

→ is determined by 8-bit subtractor &

→ is sent to SHIFTER unit.

• In step 1, sign is sent to SWAP network (Figure 9.26).

If sign=0, then $E_A > E_B$ and mantissas M_A & M_B are sent straight through SWAP network. If sign=1, then $E_A < E_B$ and the mantissas are swapped before they are sent to SHIFTER.

• In step 2, 2:! MUX is used. The exponent of result E is tentatively determined as E_A if $E_A > E_B$ or

E_B if $E_A < E_B$

• In step 3, CONTROL logic

→ determines whether mantissas are to be added or subtracted.

→ determines sign of the result.

• In step 4, result of step 3 is normalized. The number of leading zeros in M determines number of bit shifts(X) to be applied to M.

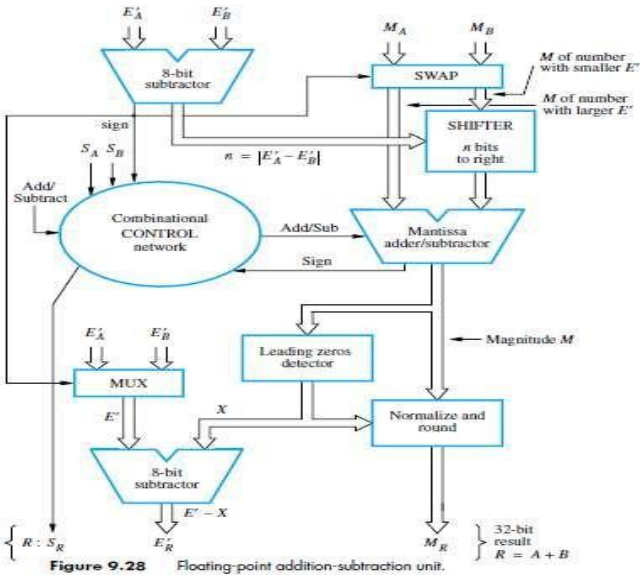


Figure 9.28 Floating-point addition-subtraction unit.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=-y90W_BEp-0

https://www.youtube.com/watch?v=0HiGruw9VcQ

https://www.youtube.com/watch?v=B3Sggj1HmR4

Important Books/Journals for further learning including the page nos.:

https://nptel.ac.in/courses/106105163/

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017 Page no: 205-230

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L - 18

IT

II/III

Course Name with Code : COMPUTER ORGANIZATION AND ARCHITECTURE-19ITC06

Course Faculty : T.MANIVEL

Unit : II

Date of Lecture:

Topic of Lecture: Parallelism and computer Arithmetic:Subword parallelism Streaming SIMD Extensions

Introduction:

In subword parallelism, multiple subwords are packed into a word and then process whole words. A subword is a lower precision unit of data contained within a word. Since the same instruction is applied to all subwords within the word, This is a form of SIMD(Single Instruction Multiple Data) processing.

Learning objective:

- To discuss the operation of subword parallelism of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts

Detailed content of the Lecture:

SUB WORD PARALLELISM

A subword is a lower precision unit of data contained within a word. In subword parallelism, multiple subwords are packed into a word and then process whole words. With the appropriate subword boundaries this technique results in parallel processing of subwords. Since the same instruction is applied to all subwords within the word, This is a form of SIMD(Single Instruction Multiple Data) processing.

It is possible to apply subword parallelism to noncontiguous subwords of different sizes within a word. In practical implementation is simple if subwords are same size and they are contiguous within a word. The data parallel programs that benefit from subword parallelism tend to process data that are of the same size.

For example if word size is 64bits and subwords sizes are 8,16 and 32 bits. Hence an instruction operates on eight 8bit subwords, four 16bit subwords, two 32bit subwords or one 64bit subword in parallel.

- Subword parallelism is an efficient and flexible solution for media processing because algorithm exhibit a great deal of data parallelism on lower precision data.

- It is also useful for computations unrelated to multimedia that exhibit data parallelism on lower precision data.

· Graphics and audio applications can take advantage of performing simultaneous operations on short vectors

ALU

Most computer operations are executed in the arithmetic and logic unit (ALU) of the processor. Consider a typical example: Suppose two numbers located in the memory are to be added. They are brought into the processor, and the actual addition is carried out by the ALU. The sum may then be stored in the memory or retained in the processor for immediate use.

Any other arithmetic or logic operation, for example, multiplication, division, or comparison of numbers, is initiated by bringing the required operands into the processor, where the operation is performed by the ALU. When operands are brought into the processor, they are stored in high-speed storage elements called registers. Each register can store one word of data. Access times to registers are somewhat faster than access times to the fastest cache unit in the memory hierarchy.

The control and the arithmetic and logic units are many times faster than other devices connected to a computer system.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=-y90W_BEp-0

https://www.youtube.com/watch?v=0HiGruw9VcQ

https://www.youtube.com/watch?v=B3Sggj1HmR4

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 230-233

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L – 19,L20

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE--19ITC06**

Course Faculty : **T.MANIVEL**

Unit : **III**

Date of Lecture:

Topic of Lecture: Building a data path- A simple Implementation scheme

Introduction:

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing. Pipelining is a technique where multiple instructions are overlapped during execution.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

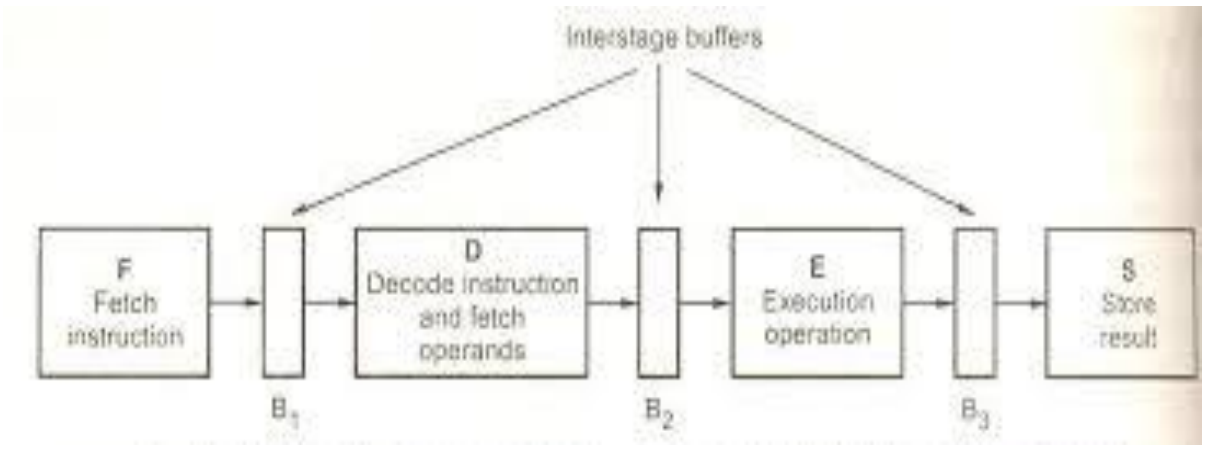
- Fundamental Concepts
- Execution of a Complete Instruction

Detailed content of the Lecture: Detailed content of the Lecture:

Most of the digital computers with complex instructions require instruction pipeline to carry out operations like fetch, decode and execute instructions.

New information is loaded into this buffer at the end of each clock cycle.

- **F Fetch:** read the instruction from the memory
- **D Decode:** decode the instruction and fetch the source operand(s)
- **E Execute:** perform the operation specified by the instruction
- **W Write:** store the result in the destination location



Pipelined execution

- In the first clock cycle, the fetch unit fetches an instruction I1 (step F1) and stores it in buffer B1 at the end of the clock cycle.
- In the second clock cycle the instruction fetch unit proceeds with the fetch operation for instruction I2 (step F2).
- Meanwhile, the execution unit performs the operation specified by instruction I1, which is available to it in buffer B1 (step E1).
- By the end of the second clock cycle, the execution of instruction I1 is completed and instruction I2 is available.
- Instruction I2 is stored in B1, replacing I1, which is no longer needed.
- Step E2 is performed by the execution unit during the third clock cycle, while instruction I3 is being fetched by the fetch unit.

Pipeline performance

- The pipeline may also be stalled because of a delay in the availability of an instruction.
- For example, this may be a result of a miss in the cache, requiring the instruction to be fetched from the main memory.
- Such hazards are often called control hazards or instruction hazards.

Control Implementation scheme

Instruction execution steps in successive clock cycles:

Function performed by each process stage in successive clock cycles

Clock Cycle	1	2	3	4	5	6	7	8	9
Stage F: Fetch	F1	F2	F2	F2	F2				
D: Decode		D1	idle	idle	idle	D2	D3		
E: Execute			E1	idle	idle	idle	E2	E3	
W: Write				W1	idle	idle	idle	W2	W3

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=q4fwx3h3mdg
https://www.youtube.com/watch?v=th2wcy0zJ-o
https://www.youtube.com/watch?v=apz1qL7jDeQ

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 263-271,271-283

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L – 21,L22

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE--19ITC06**

Course Faculty : **T.MANIVEL**

Unit : **III**

Date of Lecture:

Topic of Lecture: Overview of Pipelining – Pipelined datapath

Introduction:

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing. Pipelining is a technique where multiple instructions are overlapped during execution.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

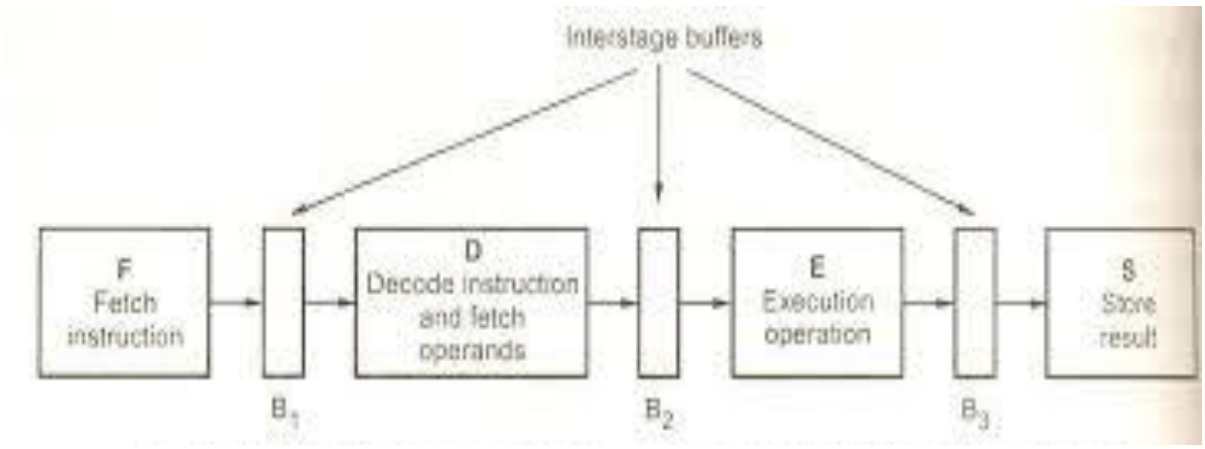
- Fundamental Concepts
- Execution of a Complete Instruction

Detailed content of the Lecture: Detailed content of the Lecture:

Most of the digital computers with complex instructions require instruction pipeline to carry out operations like fetch, decode and execute instructions.

New information is loaded into this buffer at the end of each clock cycle.

- **F Fetch:** read the instruction from the memory
- **D Decode:** decode the instruction and fetch the source operand(s)
- **E Execute:** perform the operation specified by the instruction
- **W Write:** store the result in the destination location



Pipelined execution

- In the first clock cycle, the fetch unit fetches an instruction I1 (step F1) and stores it in buffer B1 at the end of the clock cycle.
- In the second clock cycle the instruction fetch unit proceeds with the fetch operation for instruction I2 (step F2).
- Meanwhile, the execution unit performs the operation specified by instruction I1, which is available to it in buffer B1 (step E1).
- By the end of the second clock cycle, the execution of instruction I1 is completed and instruction I2 is available.
- Instruction I2 is stored in B1, replacing I1, which is no longer needed.
- Step E2 is performed by the execution unit during the third clock cycle, while instruction I3 is being fetched by the fetch unit.

Pipeline performance

- The pipeline may also be stalled because of a delay in the availability of an instruction.
- For example, this may be a result of a miss in the cache, requiring the instruction to be fetched from the main memory.
- Such hazards are often called control hazards or instruction hazards.

Pipelining –Pipelined datapath

Instruction execution steps in successive clock cycles:

Function performed by each process stage in successive clock cycles

Clock Cycle	1	2	3	4	5	6	7	8	9
Stage F: Fetch	F1	F2	F2	F2	F2				
D: Decode		D1	idle	idle	idle	D2	D3		
E: Execute			E1	idle	idle	idle	E2	E3	
W: Write				W1	idle	idle	idle	W2	W3

The operation specified by an instruction can be carried out by performing one or more of the following actions:

1) Read the contents of a given memory-location and load them into a register.
2) Read data from one or more registers.
3) Perform an arithmetic or logic operation and place the result into a register.
4) Store data from a register into a given memory-location.

SINGLE BUS ORGANIZATION

An instruction is executed by performing one or more of the following operations:

1) Transfer a word of data from one register to another or to the ALU.
2) Perform arithmetic or a logic operation and store the result in a register.
3) Fetch the contents of a given memory-location and load them into a register.
4) Store a word of data from a register into a given memory-location.

REGISTER TRANSFERS

• Instruction execution involves a sequence of steps in which data are transferred from one register to another. For each register, two control-signals are used: Ri_{in} & Ri_{out}. These are called **Gating Signals.**

Input & Output Gating for one Register Bit

• A 2-input multiplexer is used to select the data applied to the input of an edge-triggered D flip-flop.

• Ri_{in}=1 à mux selects data on bus. This data will be loaded into flip-flop at

rising-edge of clock. Ri_{in}=0 à mux feeds back the value currently stored in flip-flop.

• Q output of flip-flop is connected to bus via a tri-state gate. Ri_{out}=0 à gate's output is in the high impedancestate.

Ri_{out}=1 à the gate drives the bus to 0 or 1, depending on the value of Q.

PERFORMING AN ARITHMETIC OR LOGIC OPERATION

• The ALU performs arithmetic operations on the 2 operands applied to its A and B inputs.

• Instruction execution proceeds as follows:

Step 1 --> Contents from register R1 are loaded into register Y.

Step2 --> Contents from Y and from register R2 are applied to the A and inputs of ALU; Addition is performed & Result is stored in the Z register.

Step 3 --> The contents of Z register is stored in the R3 register.

FETCHING A WORD FROM MEMORY

- To fetch instruction/data from memory, processor transfers required address to MAR. At the same time, processor issues Read signal on control-lines of memory-bus.
- **Consider the instruction Move (R1), R2. The sequence of steps is:**
  1) $R1_{out}$, MAR_{in}, Read; desired address is loaded into MAR & Read command is issued.
  2) MDR_{inE}, WMFC; load MDR from memory-bus & Wait for MFC response from memory.
  3) MDR_{out}, $R2_{in}$; load R2 from MDR. where WMFC=control-signal that causes processor's control. circuitry to wait for arrival of MFC signal.

STORING A WORD IN MEMORY

• Consider the instruction *Move R2, (R1).* This requires the following sequence:

1) $R1_{out}$, MAR_{in} ; desired address is loaded into MAR.
2) $R2_{out}$, MDR_{in}, Write ; data to be written are loaded into MDR & Write command is issued.

MDR_{outE}, WMFC ; load data into memory-location pointed by R1 from MDR.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=q4fwx3h3mdg
https://www.youtube.com/watch?v=th2wcy0zJ-o
https://www.youtube.com/watch?v=apz1qL7jDeQ

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 283-316

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

LECTURE HANDOUTS

L – 25

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE--19ITC06**

Course Faculty : **T.MANIVEL**

Unit : **III**

Date of Lecture:

Topic of Lecture: Exceptions, Parallelism via Instructions

Introduction:

A goal of compiler and processor designers is to identify and take advantage of as much ILP as possible. Ordinary programs are typically written under a sequential execution model where instructions execute one after the other and in the order specified by the programmer. ILP allows the compiler and the processor to overlap the execution of multiple instructions or even to change the order in which instructions are executed.

How much ILP exists in programs is very application specific. In certain fields, such as graphics and scientific computing the amount can be very large. However, workloads such as cryptography may exhibit much less parallelism.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Instruction-level parallelism and basic concepts
- VLIW and control considerations
- Influence on Instruction Sets

Detailed content of the Lecture: Detailed content of the Lecture:

Instruction-level parallelism (**ILP**) is a measure of how many of the instructions in a computer program can be executed simultaneously.

ILP must not be confused with concurrency:

- ILP is the parallel execution of a sequence of instructions belonging to a specific thread of execution of a process (a running program with its set of resources: address space, a set of registers, its identifiers, its state, program counter (aka instruction pointer), and more).
- On the other hand, concurrency involves the assignment of threads of one or different processes to a CPU's core in a strict alternation, or in true parallelism if there are enough CPU cores, ideally one core for each runnable thread.

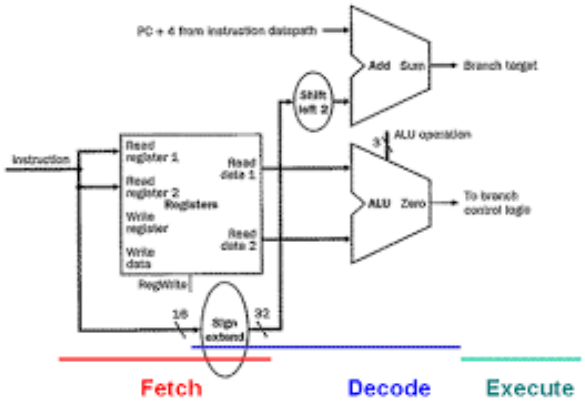
There are two approaches to instruction level parallelism: Hardware and Software.

Hardware level works upon dynamic parallelism, whereas the software level works on static parallelism. Dynamic parallelism means the processor decides at run time which instructions to execute in parallel, whereas static parallelism means the compiler decides which instructions to execute in parallel.[2][*clarification needed*] The Pentium processor works on the dynamic sequence of parallel execution, but

the Itanium processor works on the static level parallelism.

Micro-architectural techniques that are used to exploit ILP include:

- Instruction pipelining where the execution of multiple instructions can be partially overlapped.
- Superscalar execution, VLIW, and the closely related explicitly parallel instruction computing concepts, in which multiple execution units are used to execute multiple instructions in parallel.
- Out-of-order execution where instructions execute in any order that does not violate data dependencies. Note that this technique is independent of both pipelining and superscalar execution. Current implementations of out-of-order execution dynamically (i.e., while the program is executing and without any help from the compiler) extract ILP from ordinary programs.
- Register renaming which refers to a technique used to avoid unnecessary serialization of program operations imposed by the reuse of registers by those operations, used to enable out-of-order execution.
- Speculative execution which allows the execution of complete instructions or parts of instructions before being certain whether this execution should take place. A commonly used form of speculative execution is control flow speculation where instructions past a control flow instruction (e.g., a branch) are executed before the target of the control flow instruction is determined. Several other forms of speculative execution have been proposed and are in use including speculative execution driven by value prediction, memory dependence prediction and cache latency prediction.
- It is known that the ILP is exploited by both the compiler and hardware support but the compiler also provides inherent and implicit ILP in programs to hardware by compilation optimization. Some optimization techniques for extracting available ILP in programs would include scheduling, register allocation/renaming, and memory access optimization.



Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=q4fwx3h3mdg

https://www.youtube.com/watch?v=th2wcy0zJ-o

https://www.youtube.com/watch?v=apz1qL7jDeQ

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 336-342

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

LECTURE HANDOUTS

L – 26,L27

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE--19ITC06**

Course Faculty : **T.MANIVEL**

Unit : **III**

Date of Lecture:

Topic of Lecture: Instruction level Parallelism and Matrix Multiply Hardware Design language

Introduction:

A goal of compiler and processor designers is to identify and take advantage of as much ILP as possible. Ordinary programs are typically written under a sequential execution model where instructions execute one after the other and in the order specified by the programmer. ILP allows the compiler and the processor to overlap the execution of multiple instructions or even to change the order in which instructions are executed.

How much ILP exists in programs is very application specific. In certain fields, such as graphics and scientific computing the amount can be very large. However, workloads such as cryptography may exhibit much less parallelism.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Instruction-level parallelism and basic concepts
- VLIW and control considerations
- Influence on Instruction Sets

Detailed content of the Lecture: Detailed content of the Lecture:

Instruction-level parallelism (**ILP**) is a measure of how many of the instructions in a computer program can be executed simultaneously.

ILP must not be confused with concurrency:

- ILP is the parallel execution of a sequence of instructions belonging to a specific thread of execution of a process (a running program with its set of resources: address space, a set of registers, its identifiers, its state, program counter (aka instruction pointer), and more).
- On the other hand, concurrency involves the assignment of threads of one or different processes to a CPU's core in a strict alternation, or in true parallelism if there are enough CPU cores, ideally one core for each runnable thread.

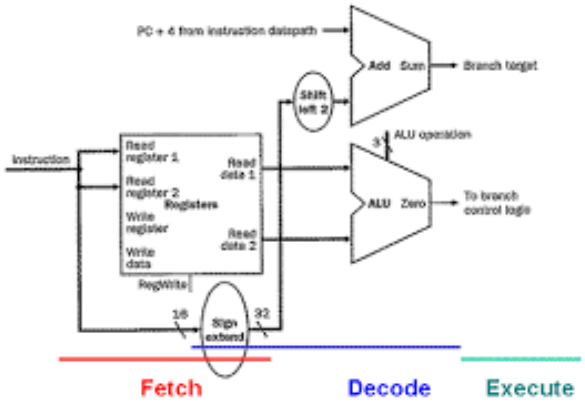
There are two approaches to instruction level parallelism: Hardware and Software.

Hardware level works upon dynamic parallelism, whereas the software level works on static parallelism. Dynamic parallelism means the processor decides at run time which instructions to execute in parallel, whereas static parallelism means the compiler decides which instructions to execute in parallel.[2][*clarification needed*] The Pentium processor works on the dynamic sequence of parallel execution, but

the Itanium processor works on the static level parallelism.

Micro-architectural techniques that are used to exploit ILP include:

- Instruction pipelining where the execution of multiple instructions can be partially overlapped.
- Superscalar execution, VLIW, and the closely related explicitly parallel instruction computing concepts, in which multiple execution units are used to execute multiple instructions in parallel.
- Out-of-order execution where instructions execute in any order that does not violate data dependencies. Note that this technique is independent of both pipelining and superscalar execution. Current implementations of out-of-order execution dynamically (i.e., while the program is executing and without any help from the compiler) extract ILP from ordinary programs.
- Register renaming which refers to a technique used to avoid unnecessary serialization of program operations imposed by the reuse of registers by those operations, used to enable out-of-order execution.
- Speculative execution which allows the execution of complete instructions or parts of instructions before being certain whether this execution should take place. A commonly used form of speculative execution is control flow speculation where instructions past a control flow instruction (e.g., a branch) are executed before the target of the control flow instruction is determined. Several other forms of speculative execution have been proposed and are in use including speculative execution driven by value prediction, memory dependence prediction and cache latency prediction.
- It is known that the ILP is exploited by both the compiler and hardware support but the compiler also provides inherent and implicit ILP in programs to hardware by compilation optimization. Some optimization techniques for extracting available ILP in programs would include scheduling, register allocation/renaming, and memory access optimization.



Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=q4fwx3h3mdg

https://www.youtube.com/watch?v=th2wcy0zJ-o

https://www.youtube.com/watch?v=apz1qL7jDeQ

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 342-365

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

L – 28,L29

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **IV** **Date of Lecture:**

Topic of Lecture: Memory technologies

Introduction : (Maximum 5 sentences) :

- The maximum size of the Main Memory (MM) that can be used in any computer is determined by its addressing scheme.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Storage devices
- Memory
- Primary storage
- Secondary storage

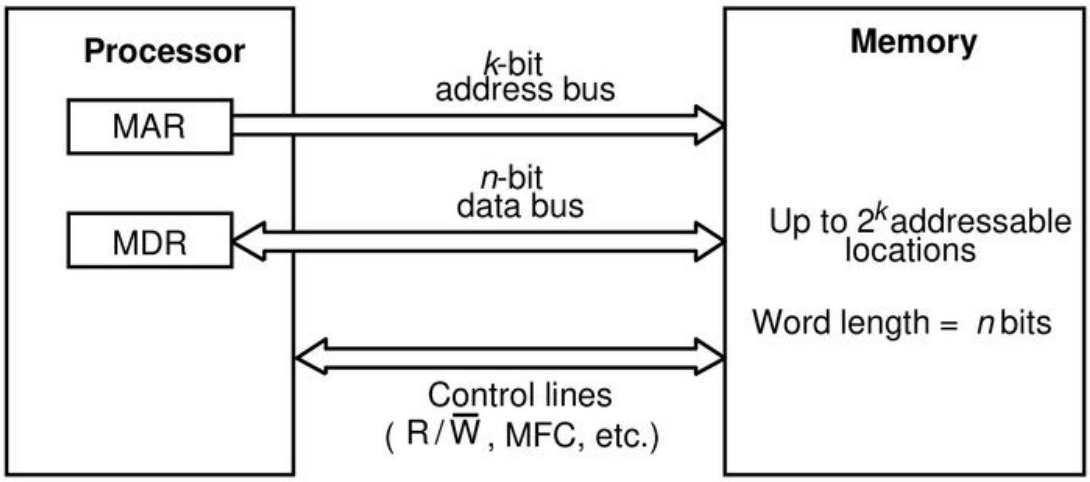
Detailed content of the Lecture:

- A 16-bit computer that generates 16-bit addresses is capable of addressing up to 216 =64K memory locations.
- If a machine generates 32-bit addresses, it can access up to 232 = 4G memory locations. This number represents the size of address space of the computer.
- If the smallest addressable unit of information is a memory word, the machine is called word-addressable.If individual memory bytes are assigned distinct addresses, the computer is called byte-addressable.
- Most of the commercial machines are byte addressable. For example in a byte-addressable 32-bit computer, each memory word contains 4 bytes.

Word Address	Byte Address
0	0 1 2 3
4	4 5 6 7
8	8 9 10 11
.	……
.	…….....

- With the above structure a READ or WRITE may involve an entire memory word or it may involve only a byte. In the case of byte read, other bytes can also be read but ignored by the CPU.

- However, during a write cycle, the control circuitry of the MM must ensure that only the specified byte is altered.
- In this case, the higher-order 30 bits can specify the word and the lower-order 2 bits can specify the byte within the word.



Memory Access Times: -

- It is a useful measure of the speed of the memory unit.
- It is the time that elapses between the initiation of an operation and the completion of that operation (for example, the time between READ and MFC).

Memory Cycle Time :-

- It is an important measure of the memory system.
- It is the minimum time delay required between the initiations of two successive memory operations (for example, the time between two successive READ operations).
- The cycle time is usually slightly longer than the access Time.

RAM is mostly volatile i.e. the data in RAM is dependent on the power and as lost when power is switched off. However, there are some non - volatile versions of RAM also available.

- There are mainly two types of RAM available i.e. Static RAM (SRAM) and Dynamic RAM (DRAM).

Static Random Access Memory (SRAM)

- SRAM is a type of semiconductor memory that uses flip flops to store the bits. SRAM is volatile even though it exhibits data eminence
- SRAM is more expensive than DRAM and consequently is used to create the cache while DRAM is used to create the main memory.

Characteristics of Dynamic RAM

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM

Advantages of SRAM

- It is relatively simple to handle a SRAM.

- SRAM results in a lower power consumption that DRAM.
- SRAM is quite reliable and so it is used as cache memory in computer systems.

Disadvantages of SRAM

- SRAM is quite expensive. So it is used to create a small cache memory and is not used for main memory.

Dynamic Random Access Memory (DRAM)

- DRAM is a type of semiconductor memory that uses capacitors to store the bits. The charging and discharging of the capacitor represents 0 and 1 i.e. the two possible values that can be stored in a bit.
- The DRAM is a volatile memory i.e. the data in memory is lost when power is switched off. However, it still displays some data eminence. DRAM is low cost compared to SRAM so it is primarily used in main memory.

Advantages of DRAM

- DRAM is much cheaper compared to SRAM so it is used as main memory.
- Its storage capacity is quite high.
- The structure of DRAM is simpler than SRAM.

Disadvantages of DRAM

- It is slower than SRAM. So it is not used for cache memory.
- DRAM has higher power consumption than SRAM.

Uses of RAM

- RAM is usually used as main memory i.e. temporary storage for computer applications and operations.

Virtual memory

- Operating systems use a part of RAM to implement paging. This leads to an illusion that memory is more than it actually. However, paging should only be used to a limit or it results in thrashing.

RAM disk

- The RAM disk is a part of the computer RAM that is treated as a hard drive by the system. However, the data in the RAM disk is lost if power is switched off, unless there is a backup power source.

Shadow RAM

- This is created if the contents of a slow ROM are copied into a much faster read/write memory. The memory locations are then switched. This is known as shadowing.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=3YHAOib_7K8
https://www.youtube.com/watch?v=tas2eUavhRE

Important Books/Journals for further learning including the page nos.:
David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 392-397

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L – 30,L31

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **IV** **Date of Lecture:**

Topic of Lecture: Basics of Cache – Measuring and improving cache performance

Introduction : (Maximum 5 sentences) :

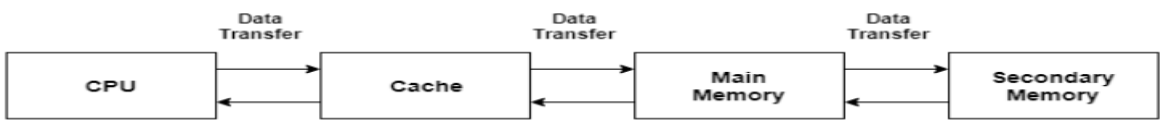
- It acts as a buffer between the CPU and the main memory.
- It is used to hold those parts of data and program which are most frequently used by the CPU.
- Performance Divides the memory system into a number of memory modules of a processor
- How fast machine instructions can be brought into the processor for execution.
- How fast the instructions can be executed.
- Interleaving

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Memory
- Storage
- Address
- Performance
- Machine instructions

Detailed content of the Lecture:

- Capacity in terms of storage increases.
- Cost per bit of storage decreases.
- Frequency of access of the memory by the CPU decreases.
- Access time by the CPU increases.
- Cache is a type of memory that is used to increase the speed of data access. Normally, the data required for any process resides in the main memory. However, it is transferred to the cache memory temporarily if it is used frequently enough.
- A diagram to better understand the data transfer in cache management is as follows:



Cache Performance

- If a process needs some data, it first searches in the cache memory. If the data is available in the cache, this is termed as a cache hit and the data is accessed as required.
- If the data is not in the cache then it is termed as a cache miss. Then the data is obtained from the main memory. After that the data is transferred to the cache memory under the assumption that it will be needed again.
- The performance of the cache is measured using the hit ratio.
- It is the number of cache hits divided by the total cache accesses. The formula for this is:

$$\text{Hit Ratio} = \frac{\text{Number of Cache Hits}}{\text{Number of Cache Hits + Number of Cache Misses}}$$

Types of Cache Memory

There are mainly two types of cache memory i.e. primary cache and secondary cache. These are explained in detail as follows:

Primary Cache

- Primary cache is very fast and its access time is similar to the processor registers. This is because it is built onto the processor chip.
- However because of this reason, its size is quite small. It is also known as a level 1 cache and is building using static RAM (SRAM).

Secondary Cache

- The secondary cache or external cache is cache memory that is external to the primary cache. It is located between the primary cache and the main memoryAdvantages of Cache Memory
- Cache memory is faster than main memory as it is located on the processor chip itself. Its speed is comparable to the processor registers and so frequently required data is stored in the cache memory.
- The memory access time is considerably less for cache memory as it is quite fast. This leads to faster execution of any process.
- The cache memory can store data temporarily as long as it is frequently required. After the use of any data has ended, it can be removed from the cache and replaced by new data from the main memory.

Disadvantages of Cache Memory

- Since the cache memory is quite fast, it is extremely useful in any computer system. However, it is also quite expensive and so is used judiciously.
- The cache is memory expensive as observed from the previous point. Also, it is located directly on the processor chip.
- Because of these reasons, it has a limited capacity and is much smaller than main memory.
- Performance Divides the memory system into a number of memory modules of a processor
- How fast machine instructions can be brought into the processor for execution.

- How fast the instructions can be executed.
- Interleaving
- Response time is the time from start to completion of a task.

This also includes:

- Operating system overhead.
- Waiting for I/O and other processes
- Accessing disk and memory
- Time spent executing on the CPU or execution time.

Throughput is the total amount of work done in a given tim

CPU execution time is the total time a CPU spends computing on a given task.

- It also excludes time for I/O or running other programs.
- This is also referred to as simply CPU time.
- Performance is determined by execution time as performance is inversely proportional to execution time.

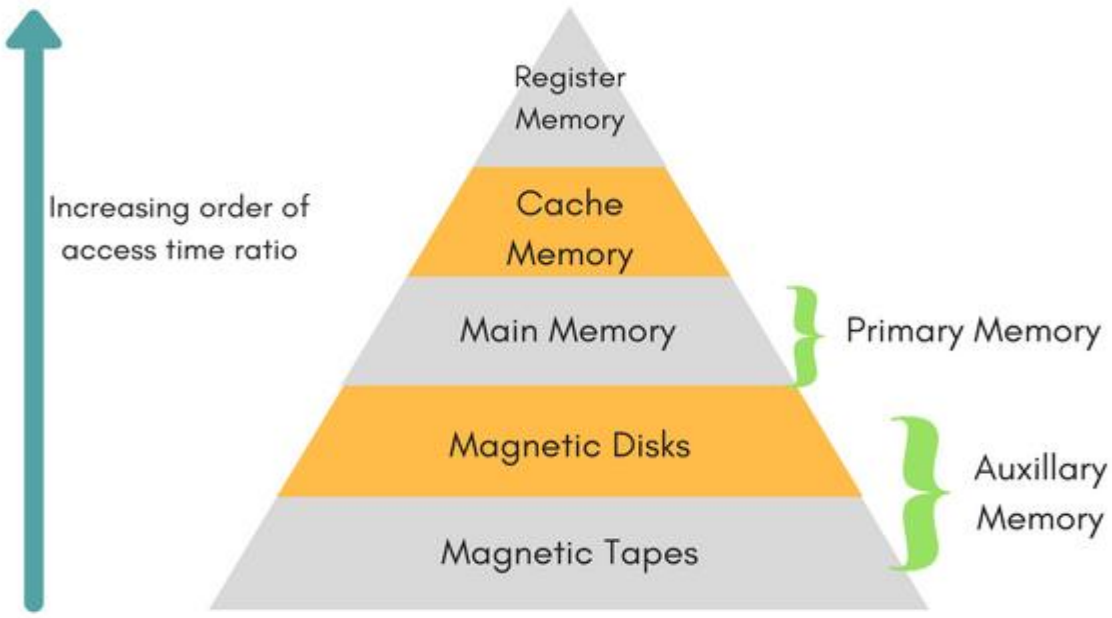
```
Performance = (1 / Execution time)
```

The units for CPU Execution time are:

And,

```
(Performance of A /  Performance of B)
= (Execution Time of B / Execution Time of A)
```

- If given that Processor A is faster than processor B, that means execution time of A is less than that of execution time of B.
- Therefore, performance of A is greater than that of performance of B.



How to Improve Performance?

- To improve performance you can either:
- Decrease the CPI (clock cycles per instruction) by using new Hardware.
- Decrease the clock time or Increase clock rate by reducing propagation delays or by use pipelining.Decrease the number of required cycles or improve ISA or Compiler.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=fn9Hn__x6dA
https://www.youtube.com/watch?v=QcAaP5V2Gpc
https://www.youtube.com/watch?v=iL01Qr0Tspc
https://www.youtube.com/watch?v=TyYeCpdJCb8

Important Books/Journals for further learning including the page nos.:
David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 397-412,412-432

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L – 32,L 33

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **IV** **Date of Lecture:**

Topic of Lecture: Virtual Memory

Introduction : (Maximum 5 sentences) :

- Virtual memory is a valuable concept in computer architecture that allows you to run large, sophisticated programs on a computer even if it has a relatively small amount of RAM.
- A computer with virtual memory artfully juggles the conflicting demands of multiple programs within a fixed amount of physical memory.

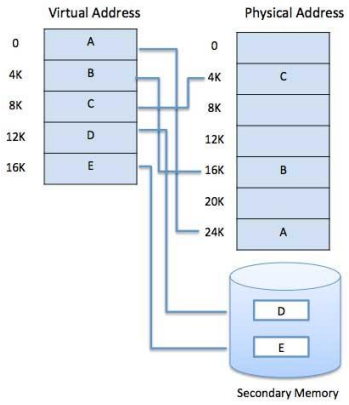
Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Virtual address
- Physical address
- Memory management unit

Detailed content of the Lecture:

- A computer can address more memory than the amount physically installed on the system.
- This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.
- The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes.
- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.
- The MMU's job is to translate virtual addresses into physical addresses.

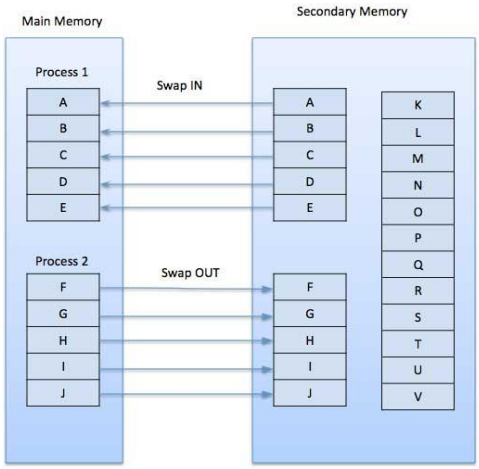
- A basic example is given below −



- Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system.
- Demand segmentation can also be used to provide virtual memory.

Demand Paging

- A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand.



Advantages

- Large virtual memory.
- More efficient use of memory.
- There is no limit on degree of multiprogramming.

Disadvantages

- Number of tables and the amount of processor overhead for handling page interrupts are greater than in the case of the simple paged management techniques.

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=ujoJ7J_l9cY
https://www.youtube.com/watch?v=6oQfvjd_T_E

Important Books/Journals for further learning including the page nos.:
David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 438-446

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L – 34,L 35

LECTURE HANDOUTS

IT

II/IV

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **IV** **Date of Lecture:**

Topic of Lecture: Secondary Storage-Redundant Array of Inexpensive Disks

Introduction : (Maximum 5 sentences) :

- ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is non-volatile.
- The information is stored permanently in such memories during manufacture.
- A ROM stores such instructions that are required to start a computer.
- This operation is referred to as bootstrap. ROM chips are not only used in the computer but also in other electronic items like washing machine and microwave oven.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Storage devices
- Primary storage
- Secondary storage

Detailed content of the Lecture:

MROM (Masked ROM)

- The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions.
- These kinds of ROMs are known as masked ROMs, which are inexpensive.

PROM (Programmable Read Only Memory)

- PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program.
- Inside the PROM chip, there are small fuses which are burnt open during programming.
- It can be programmed only once and is not erasable.

EPROM (Erasable and Programmable Read Only Memory)

- EPROM can be erased by exposing it to ultra-violet light for duration of up to 40 minutes. Usually, an EPROM eraser achieves this function.

- During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than 10 years because the charge has no leakage path.
- For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid).
- This exposure to ultra-violet light dissipates the charge. During normal use, the quartz lid is sealed with a sticker.

EEPROM (Electrically Erasable and Programmable Read Only Memory)

- EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times.
- Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

Advantages of ROM

The advantages of ROM are as follows −

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs
- Static and do not require refreshing

Some ROM is non-volatile but can be reprogrammed, this includes:

- Erasable Programmable Read-Only Memory (EPROM): This is programmed with the use of very high voltages and exposure to approximately 20 minutes of intense ultraviolet (UV) light.
- Ultraviolet-Erasable Programmable Read-Only Memory (UV-EPROM): This is read-only memory that can be erased by the use of ultraviolet light and then reprogrammed.
- ROM is also often used in optical storage media such as various types of compact discs, including read-only memory (CD-ROM), compact disc recordable (CD-R) and compact disc rewritable (CD-RW).

Video Content / Details of website for further learning (if any):
https://www.youtube.com/watch?v=9-ivunH8Aps
https://www.youtube.com/watch?v=VkGmnsWisBk

Important Books/Journals for further learning including the page nos.:
David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 481-482

Course Faculty

Verified by HOD

L - 36

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **IV** **Date of Lecture:**

Topic of Lecture: Implementing Cache coontrollers

Introduction : (Maximum 5 sentences) :

- Memory management keeps track of the status of each memory location, whether it is allocated or free.
- It allocates the memory dynamically to the programs at their request and frees it for reuse when it is no longer needed.
- Memory management meant to satisfy some requirements that we should keep in mind.

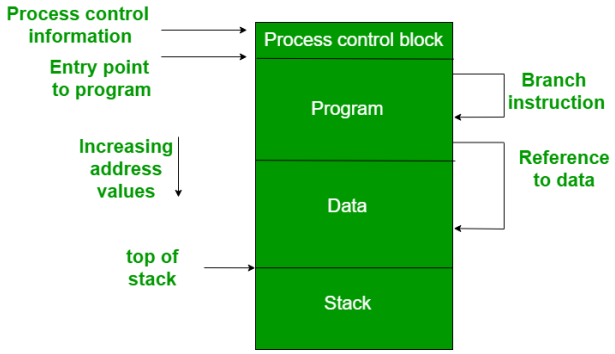
Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Memory management
- Memory location
- Relocation

Detailed content of the Lecture:

Relocation

- The available memory is generally shared among a number of processes in a multiprogramming system, so it is not possible to know in advance which other programs will be resident in main memory at the time of execution of his program.
- Swapping the active processes in and out of the main memory enables the operating system to have a larger pool of ready-to-execute process.



Protection

- There is always a danger when we have multiple programs at the same time as one program may write to the address space of another program.
- So every process must be protected against unwanted interference when other process tries to write in a process whether accidental or incidental.
- Between relocation and protection requirement a trade-off occurs as the satisfaction of

relocation requirement increases the difficulty of satisfying the protection requirement.

- Prediction of the location of a program in main memory is not possible, that's why it is impossible to check the absolute address at compile time to assure protection.

Sharing

- A protection mechanism must have to allow several processes to access the same portion of main memory. Allowing each processes access to the same copy of the program rather than have their own separate copy has an advantage.

Logical organization

- Modules are written and compiled independently and all the references from one module to another module are resolved by `the system at run time.
- Different modules are provided with different degrees of protection.
- There are mechanisms by which modules can be shared among processes. Sharing can be provided on a module level that lets the user specify the sharing that is desired.

Physical organization

- The structure of computer memory has two levels referred to as main memory and secondary memory.
- Main memory is relatively very fast and costly as compared to the secondary memory. Main memory is volatile.
- Thus secondary memory is provided for storage of data on a long-term basis while the main memory holds currently used programs.
- The major system concern between main memory and secondary memory is the flow of information and it is impractical for programmers to understand this for two reasons:
- The programmer may engage in a practice known as overlaying when the main memory available for a program and its data may be insufficient. It allows different modules to be assigned to the same region of memory. One disadvantage is that it is time-consuming for the programmer.
- In a multiprogramming environment, the programmer does not know how much space will be available at the time of coding and where that space will be located inside the memory.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=UDPYpf-nsDY
https://www.youtube.com/watch?v=FrTttJLN7Kw

Important Books/Journals for further learning including the page nos.:

David A.Patterson John L.Hennessey, Computer Organization and design ARM Edition 2017
Page no: 482

Course Faculty

Verified by HOD

L - 37

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Accessing I/O Devices

Introduction : (Maximum 5 sentences) :

- A simple arrangement to connect I/O devices to a computer is to use a single bus arrangement.
- The bus enables all the devices connected to it to exchange information.
- Typically, it consists of three sets of lines used to carry address, data, and control signals. Each I/O device is assigned a unique set of addresses.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Input Device
- Output Device

Detailed content of the Lecture:

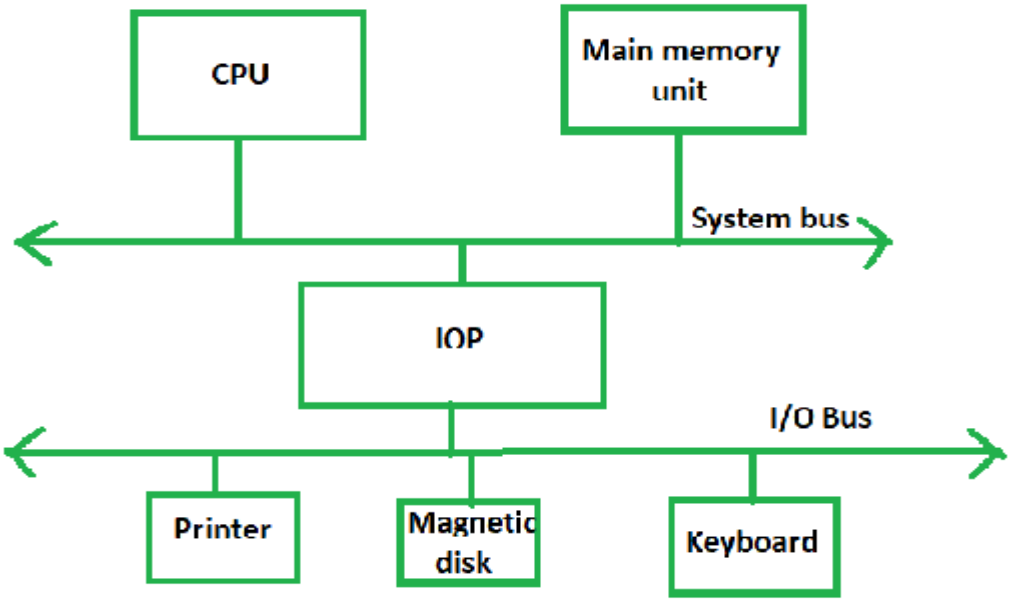
- A simple arrangement to connect I/O devices to a computer is to use a single bus arrangement. The bus enables all the devices connected to it to exchange information.
- Typically, it consists of three sets of lines used to carry address, data, and control signals. Each I/O device is assigned a unique set of addresses.
- When the processor places a particular address on the address line, the device that recognizes this address responds to the commands issued on the control lines.
- The processor requests either a read or a write operation, and the requested data are transferred over the data lines, when I/O devices and the memory share the same address space, the arrangement is called memory-mapped I/O.
- With memory-mapped I/O, any machine instruction that can access memory can be used to transfer data to or from an I/O device.
- For example, if DATAIN is the address of the input buffer associated with the keyboard, the instruction

Move DATAIN, R0

- Reads the data from DATAIN and stores them into processor register R0.

Move R0, DATAOUT

- Sends the contents of register R0 to location DATAOUT, which may be the output data buffer of a display unit or a printer. Most computer systems use memory-mapped I/O.
- some processors have special In and Out instructions to perform I/O transfers.
- When building a computer system based on these processors, the designer had the option of connecting I/O devices to use the special I/O address space or simply incorporating them as part of the memory address space.
- The I/O devices examine the low-order bits of the address bus to determine whether they should respond.
- The hardware required to connect an I/O device to the bus.
- The address decoder enables the device to recognize its address when this address appears on the address lines.
- The data register holds the data being transferred to or from the processor.
- The status register contains information relevant to the operation of the I/O device.
- Both the data and status registers are connected to the data bus and assigned unique addresses.
- The address decoder, the data and status registers, and the control circuitry required to coordinate I/O transfers constitute the device's interface circuit.



Input-Output Processor

Video Content / Details of website for further learning (if any):
https://slideplayer.com/slide/8420897/
https://www.youtube.com/watch?v=CBTv37EFqTo

Important Books/Journals for further learning including the page nos.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no: 151-155.

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L - 38

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Interrupts-Interrupt Hardware

Introduction : (Maximum 5 sentences) :

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Input signal
- Interrupt signal

Detailed content of the Lecture:

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.
- In I/O devices one of the bus control lines is dedicated for this purpose and is called the Interrupt Service Routine (ISR).
- In digital computers, an interrupt is an input signal to the processor indicating an event that needs immediate attention.
- An interrupt signal alerts the processor and serves as a request for the processor to interrupt the currently executing code, so that the event can be processed in a timely manner.

Hardware Interrupts:

- In a hardware interrupt, all the devices are connected to the Interrupt Request Line.
- A single request line is used for all the n devices.
- To request an interrupt, a device closes its associated switch.
- When a device requests an interrupts, the value of INTR is the logical OR of the requests from individual devices.

Sequence of events involved in handling an IRQ:

1. Devices raise an IRQ.
2. Processor interrupts the program currently being executed.

3. Device is informed that its request has been recognized and the device deactivates the request signal.
4. The requested action is performed.
5. Interrupt is enabled and the interrupted program is resumed.

Handling Multiple Devices:

When more than one device raises an interrupt request signal, then additional information is needed to decide which which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=lxBfuyOagS8
https://www.youtube.com/watch?v=NYTK65RrjnE

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:155-157.

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L - 39

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Enabling and Disabling Interrupts

Introduction : (Maximum 5 sentences) :

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Input signal
- Interrupt signal

Detailed content of the Lecture:

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.
- In I/O devices one of the bus control lines is dedicated for this purpose and is called the Interrupt Service Routine (ISR).
- In digital computers, an interrupt is an input signal to the processor indicating an event that needs immediate attention.
- An interrupt signal alerts the processor and serves as a request for the processor to interrupt the currently executing code, so that the event can be processed in a timely manner.

ENABLING AND DISABLING INTERRUPTS:

- The facilities provided in a computer must give the programmer complete control over the events that take place during program execution.
- The arrival of an interrupt request from an external device causes the processor to suspend the execution of one program and start the execution of another.
- Because interrupts can arrive at any time, they may alter the sequence of events from the envisaged by the programmer.
- Hence, the interruption of program execution must be carefully controlled.

Before proceeding to study more complex aspects of interrupts,
let us summarize the sequence of events involved in handling an interrupt request from a single device. Assuming that interrupts are enabled, the following is a typical scenario.

1. The device raises an interrupt request.
2. The processor interrupts the program currently being executed.
3. Interrupts are disabled by changing the control bits in the PS (except in the case of edge-triggered interrupts).
4. The device is informed that its request has been recognized, and in response, it deactivates the interrupt-request signal.
5. The action requested by the interrupt is performed by the interrupt-service routine.
6. Interrupts are enabled and execution of the interrupted program is resumed.

Handling Multiple Devices:

When more than one device raises an interrupt request signal, then additional information is needed to decide which which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=lxBfuyOagS8
https://www.youtube.com/watch?v=NYTK65RrjnE

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:155-157.

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

L - 40

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Handling Multiple Devices

Introduction : (Maximum 5 sentences) :

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Input signal
- Interrupt signal

Detailed content of the Lecture:

- Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high priority process requiring interruption of the current working process.
- In I/O devices one of the bus control lines is dedicated for this purpose and is called the Interrupt Service Routine (ISR).
- In digital computers, an interrupt is an input signal to the processor indicating an event that needs immediate attention.
- An interrupt signal alerts the processor and serves as a request for the processor to interrupt the currently executing code, so that the event can be processed in a timely manner.

Hardware Interrupts:

- In a hardware interrupt, all the devices are connected to the Interrupt Request Line.
- A single request line is used for all the n devices.
- To request an interrupt, a device closes its associated switch.
- When a device requests an interrupts, the value of INTR is the logical OR of the requests from individual devices.

Sequence of events involved in handling an IRQ:

1. Devices raise an IRQ.

2. Processor interrupts the program currently being executed.
3. Device is informed that its request has been recognized and the device deactivates the request signal.
4. The requested action is performed.
5. Interrupt is enabled and the interrupted program is resumed.

Before proceeding to study more complex aspects of interrupts,
let us summarize the sequence of events involved in handling an interrupt request from a single device. Assuming that interrupts are enabled, the following is a typical scenario.

1. The device raises an interrupt request.
2. The processor interrupts the program currently being executed.
3. Interrupts are disabled by changing the control bits in the PS (except in the case of edge-triggered interrupts).
4. The device is informed that its request has been recognized, and in response, it deactivates the interrupt-request signal.
5. The action requested by the interrupt is performed by the interrupt-service routine.
6. Interrupts are enabled and execution of the interrupted program is resumed.

Handling Multiple Devices:

When more than one device raises an interrupt request signal, then additional information is needed to decide which which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=lxBfuyOagS8
https://www.youtube.com/watch?v=NYTK65RrjnE

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:155-157.

Course Faculty

Verified by HOD

L - 38

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V**

Date of Lecture:

Topic of Lecture: Controlling Device Requests

Introduction : (Maximum 5 sentences) :

- One of the most important things concerning serial communication is the Protocol, which should be strictly observed.
- It is a set of rules, which must be applied such that the devices can correctly interpret data they mutually exchange.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Communication
- Serial communication
- Parallel communication

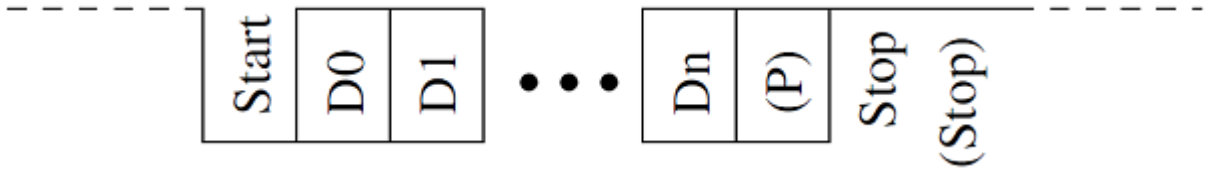
Detailed content of the Lecture:

Types of Serial Communications

- Serial communication can be further classified as −
- **Synchronous** − Devices that are synchronized use the same clock and their timing is in synchronization with each other.
- **Asynchronous** − Devices that are asynchronous have their own clocks and are triggered by the output of the previous state.
- It is easy to find out if a device is synchronous or not. If the same clock is given to all the connected devices, then they are synchronous. If there is no clock line, it is asynchronous.
- For example, UART (Universal Asynchronous Receiver Transmitter) module is asynchronous.
- The asynchronous serial protocol has a number of built-in rules. These rules are nothing but mechanisms that help ensure robust and error-free data transfers. These mechanisms, which we get for eschewing the external clock signal, are −
    - Synchronization bits
    - Data bits
    - Parity bits
    - Baud rate

Synchronization Bits

- The synchronization bits are two or three special bits transferred with each packet of data. They are the start bit and the stop bit(s). True to their name, these bits mark the beginning and the end of a packet respectively.
- There is always only one start bit, but the number of stop bits is configurable to either one or two (though it is normally left at one).
- The start bit is always indicated by an idle data line going from 1 to 0, while the stop bit(s) will transition back to the idle state by holding the line at 1.



Data Bits

- The amount of data in each packet can be set to any size from 5 to 9 bits. Certainly, the standard data size is your basic 8-bit byte, but other sizes have their uses.
- A 7-bit data packet can be more efficient than 8, especially if you are just transferring 7-bit ASCII characters.

Parity Bits

- The user can select whether there should be a parity bit or not, and if yes, whether the parity should be odd or even.
- The parity bit is 0 if the number of 1's among the data bits is even. Odd parity is just the opposite.

Baud Rate

- The term baud rate is used to denote the number of bits transferred per second [bps]. Note that it refers to bits, not bytes.
- It is usually required by the protocol that each byte is transferred along with several control bits.
- It means that one byte in serial data stream may consist of 11 bits.
- For example, if the baud rate is 300 bps then maximum 37 and minimum 27 bytes may be transferred per second.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=reeV290ChGQ
https://www.youtube.com/watch?v=Ut2SjYuVBM0

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:158-175

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L – 42,L 43

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Exceptions - Direct Memory Access

Introduction : (Maximum 5 sentences) :

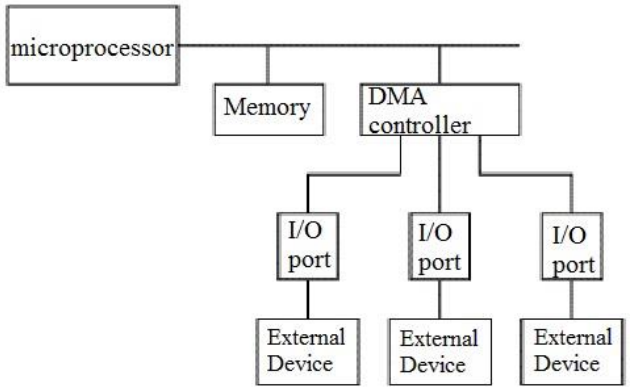
- Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations.
- The process is managed by a chip known as a DMA controller (DMAC)

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

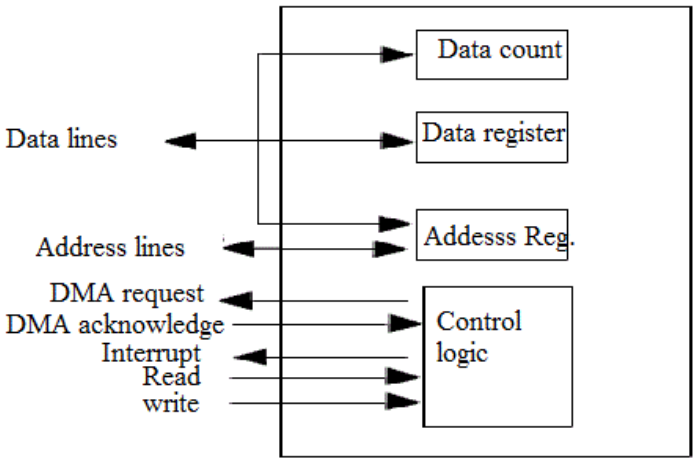
- Direct Memory Access
- Memory Access

Detailed content of the Lecture:

- The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA controller is a control unit, part of I/O device's interface circuit, which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.
- DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor. The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.



- DMA controller contains an address unit, for generating addresses and selecting I/O device for transfer. It also contains the control unit and data count for keeping counts of the number of blocks transferred and indicating the direction of transfer of data. When the transfer is completed, DMA informs the processor by raising an interrupt. The typical block diagram of the DMA controller is shown in the figure below



Working of DMA Controller

- DMA controller has to share the bus with the processor to make the data transfer. The device that holds the bus at a given time is called bus master. When a transfer from I/O device to the memory or vice verse has to be made, the processor stops the execution of the current program, increments the program counter, moves data over stack then sends a DMA select signal to DMA controller over the address bus.
- If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal. Processor grants the bus to the controller by raising the bus grant signal, now DMA controller is the bus master. The processor initiates the DMA controller by sending the memory addresses, number of blocks of data to be transferred and direction of data transfer. After assigning the data transfer task to the DMA controller, instead of waiting ideally till completion of data transfer, the processor resumes the execution of the program after retrieving instructions from the stack.

  The DMA transfers the data in three modes which include the following.

  a) **Burst Mode**: In this mode DMA handover the buses to CPU only after completion of whole data transfer. Meanwhile, if the CPU requires the bus it has to stay ideal and wait for data transfer.

  b) **Cycle Stealing Mode**: In this mode, DMA gives control of buses to CPU after transfer of every byte. It continuously issues a request for bus control, makes the transfer of one byte and returns the bus. By this CPU doesn't have to wait for a long time if it needs a bus for higher priority task.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=v58UFPKa8zs

https://www.youtube.com/watch?v=trO6gWz5l2k

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:158-175.

Course Teacher

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

L - 44

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: Buses -Standard I/O Interfaces

Introduction : (Maximum 5 sentences) :

- In computer architecture, a bus (a contraction of the Latin omnibus) is a communication system that transfers data between components inside a computer, or between computers.
- This expression covers all related hardware components (wire, optical fiber, etc.) and software, including communication protocols.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Communication
- Protocol

Detailed content of the Lecture:

- In most traditional computer architectures, the CPU and main memory tend to be tightly coupled.
- A microprocessor conventionally is a single chip which has a number of electrical connections on its pins that can be used to select an "address" in the main memory and another set of pins to read and write the data stored at that location.
- In most cases, the CPU and memory share signaling characteristics and operate in synchrony.
- The bus connecting the CPU and memory is one of the defining characteristics of the system, and often referred to simply as the system bus.
- It is possible to allow peripherals to communicate with memory in the same fashion, attaching adaptors in the form of expansion cards directly to the system bus.
- This is commonly accomplished through some sort of standardized electrical connector, several of these forming the expansion bus or local bus.
- However, as the performance differences between the CPU and peripherals vary widely, some solution is generally needed to ensure that peripherals do not slow overall system performance.

- Many CPUs feature a second set of pins similar to those for communicating with memory, but able to operate at very different speeds and using different protocols.
- Others use smart controllers to place the data directly in memory, a concept known as direct memory access.
- Most modern systems combine both solutions, where appropriate.

Internal bus

- The internal bus, also known as internal data bus, memory bus, system bus or Front-Side-Bus, connects all the internal components of a computer, such as CPU and memory, to the motherboard.
- Internal data buses are also referred to as local buses, because they are intended to connect to local devices.
- This bus is typically rather quick and is independent of the rest of the computer operations.

External buses

- The external bus, or expansion bus, is made up of the electronic pathways that connect the different external devices, such as printer etc., to the computer.

Address bus

- An address bus is a bus that is used to specify a physical address.
- When a processor or DMA-enabled device needs to read or write to a memory location, it specifies that memory location on the address bus (the value to be read or written is sent on the data bus).

Synchronous bus :

- All devices derive timing information from a common clock line.
- Equally spaced pulses on this line define equal time intervals.
- Each of these intervals constitutes a bus cycle during which one data transfer can take place.

Asynchronous bus :

- This is a scheme based on the use of a handshake between the master and the slave for controlling data transfers on the bus.
- The common clock is replaced by two timing control lines, master-ready and slave-ready.
- The first is asserted by the master to indicate that it is ready for a transaction and the second is a response from the slave.

Standard I/O Interfaces

- The master places the address and command information on the bus.
- It indicates to all devices that it has done so by activating the master-ready line.
  - The processor bus is the bus defied by the signals on the processor chip itself.
  - Devices that require a very high-speed connection to the processor, such as the main memory, may be connected directly to this bus.

For electrical reasons, only a few devices can be connected in this manner.

- Let us look in to Processor bus and Peripheral Component Interconnect (PCI) bus.
- These two buses are interconnected by a circuit called bridge.
- It is a bridge between processor bus and PCI bus.

-PCI (Peripheral Component Interconnect) – SCSI (Small Computer System Interface) – USB (Universal Serial Bus)
Video Content / Details of website for further learning (if any): https://www.youtube.com/watch?v=v58UFPKa8zs https://www.youtube.com/watch?v=trO6gWz5l2k
Important Books/Journals for further learning including the page nos.: Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:180-193

Course Faculty

Verified by HOD

MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

L - 45

LECTURE HANDOUTS

IT

II/III

Course Name with Code : **COMPUTER ORGANIZATION AND ARCHITECTURE -19ITC06**

Course Teacher : **T.MANIVEL**

Unit : **V** **Date of Lecture:**

Topic of Lecture: PCI Bus, SCSI Bus, USB

Introduction : (Maximum 5 sentences) :

- The processor bus is the bus defied by the signals on the processor chip itself.
- Devices that require a very high-speed connection to the processor, such as the main memory, may be connected directly to this bus.
- For electrical reasons, only a few devices can be connected in this manner.

Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Processor bus

Detailed content of the Lecture:

- Let us look in to Processor bus and Peripheral Component Interconnect (PCI) bus.
- These two buses are interconnected by a circuit called bridge.
- It is a bridge between processor bus and PCI bus.

-PCI (Peripheral Component Interconnect)
– SCSI (Small Computer System Interface)
– USB (Universal Serial Bus)

PCI (Peripheral Component Interconnect):

- The topics discussed under PCI are: Data Transfer, Use of a PCI bus in a computer system, A read operation on the PCI bus, Device configuration and Other electrical characteristics.
- Host, main memory and PCI Bridge are connected to disk, printer and Ethernet interface through PCI bus.
- At any given time, one device is the bus master. It has the right to initiate data transfers by issuing read and write commands.
- A master is called an initiator in PCI terminology.
- This is either processor or DMA controller. The addressed device that responds to read and write commands is called a target.
- A complete transfer operation on the bus, involving an address and a burst of data, is called a transaction.
- A PCI bus lets you change different peripherals that are attached to the computer system, so it allows the use of different sound cards and hard drives.
- Usually, there are three or four PCI slots on a motherboard. With PCI, you can unplug

the component you want to swap and plug in the new one in the PCI slot.

- Or, if you have an open slot, you can add another peripheral like a second hard drive to dual boot your computer or a special sound card if you deal with music a lot.
- Computers might have more than one type of bus handling different traffic types. The PCI bus used to come in both 32-bit and 64-bit versions. PCI runs at 33 MHz or 66 MHz

Some of its features include these:

- 32 data bits (64 bit option), 32 address bits (64-bit option)
- Up to 33 MHz, synchronous
- 132 M/s burst (sustained) (264 M/s with 64-bit option)
- Full bus master capability
- Good bus arbitration
- Slot limited to three or four cards typically
- Auto configurable
- Coexistence with ISA/EISA/MCA as well as another PCI bus
- Strong acceptance outside of the PC architecture
- Moderate cost
- Voltage: 3.3 V and 5 V

SCSI Bus:

- The SCSI controller contends for control of the bus (initiator).
- When the initiator wins the arbitration process, it selects the target controller and hands over control of the bus to it.
- The target starts an output operation. The initiator sends a command specifying the required read operation.
- The target sends a message to the initiator indicating that it will temporarily suspends the connection between them. Then it releases the bus. 10
- The target controller sends a command to the disk drive to move the read head to the first sector involved in the requested read operation.
- The target transfers the contents of the data buffer to the initiator and then suspends the connection again.
- The target controller sends a command to the disk drive to perform another seek operation.
- As the initiator controller receives the data, it stores them into the main memory using the DMA approach.
- The SCSI controller sends an interrupt to the processor to inform it that the requested operation has been completed.
- The bus signals, arbitration, selection, information.
- SCSI is a standard for parallel interfaces that transfers information at a rate of eight bits per second and faster, which is faster than the average parallel interface.
- SCSI-2 and above supports up to seven peripheral devices, such as a hard drive, CD-ROM, and scanner, that can attach to a single SCSI port on a system's bus.
- SCSI ports were designed for Apple Macintosh and Unix computers, but also can be used with PCs.
- Although SCSI was popular in the past, today it has largely been superseded by faster connection types, such as SATA.
- Computers that have a SCSI port (all Macs [except the ancient 128 and 512] and

some pcs) can have up to seven devices attached to the computer, such as an external hard disk, a scanner, a CD-ROM player, etc.

- Since information travels through the cables to these separate devices, each one must have a different **SCSI address** so the information gets to the right place.
- A SCSI address is also called a SCSIID

Because the computer always has a SCSI address of 7 and there cannot be more than one device with the same address, you cannot connect two computers to a common device at the same time, such as an external hard disk or a scanner or a CD-ROM player.

Universal Serial Bus (USB):

- The USB has been designed to meet several key objectives such as:
- Provide a simple, low-cost and easy to use interconnection system that overcomes the difficulties due to the limited number of I/O ports available on a computer
- Accommodate a wide range of data transfer characteristics for I/O devices, including telephone and Internet connections
- Enhance user convenience through a "plug-and-play" mode of operation

Port Limitation: The user may also need to know how to configure the device and the software.

Device Characteristics: A variety of simple devices attached to a computer generate data in different asynchronous mode.

Plug-and-play:

- The system should detect the existence of this new device automatically, identify the appropriate device-driver software and any other facilities needed to service that device, and establish the appropriate addresses and logical connections to enable them to communicate.

USB architecture:

- To accommodate a large number of devices that can be added or removed at any time, the USB has the tree structure. Each node has a device called a hub.
- Root hub, functions, split bus operations – high speed (HS) and Full/Low speed (F/LS).

Advantages:

- The Universal Serial Bus was designed to simplify and improve the interface between personal computers and peripheral devices, when compared with previously existing standard or ad-hoc proprietary interfaces.
- The USB interface is self-configuring.
- This means that the user need not adjust settings on the device and interface for speed or data format, or configure interrupts, input/output addresses, or direct memory access channels.
- USB connectors are standardized at the host, so any peripheral can use any available receptacle.
- USB takes full advantage of the additional processing power that can be economically put into peripheral devices so that they can manage themselves.
- USB devices mostly do not have user-adjustable interface settings.

Disadvantages

- USB cables are limited in length.
- USB has a strict "tree" topology and "master-slave" protocol for addressing peripheral

devices.

Peripheral devices cannot interact with one another except via the host, and two hosts cannot communicate over their USB ports directly.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=v58UFPKa8zs

https://www.youtube.com/watch?v=trO6gWz5l2k

Important Books/Journals for further learning including the page nos.:

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2012, page no:194-200

Course Faculty

Verified by HOD