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 - 01

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.VINUPRIYA**

Unit : **I - BASIC STRUCTURE OF COMPUTERS**

Date of Lecture:

Topic of Lecture: Functional Units

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.

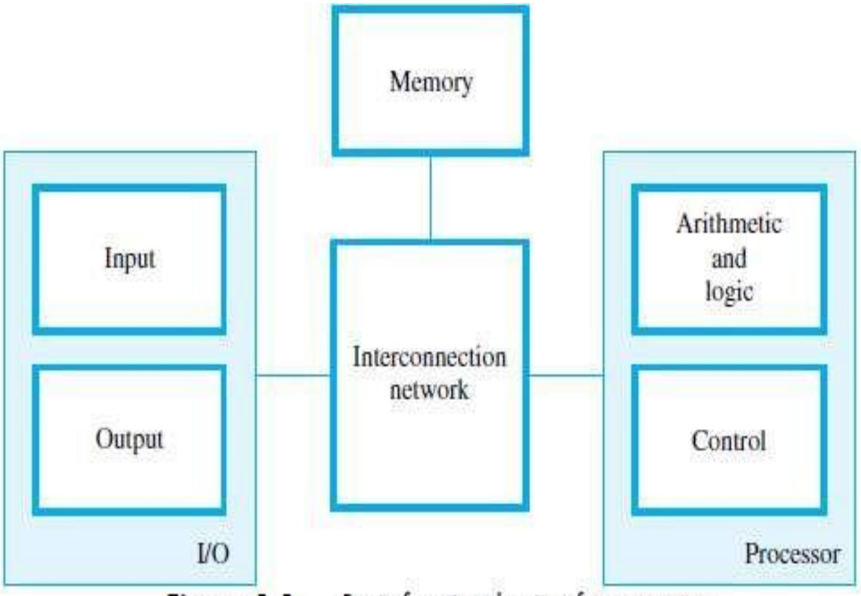


Figure 1.1 Basic functional units of a computer.

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=Ev1TQ6qEfzU
https://www.youtube.com/watch?v=U7elB4HpY3M
https://www.youtube.com/watch?v=UDvpNeTRBs0

Important Books/Journals for further learning including the page nos.:
https://nptel.ac.in/courses/106105163/
Computer Architecture: By Bester field ,Page no: 3
Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 1-6

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 02

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.VINUPRIYA

Unit : I - BASIC STRUCTURE OF COMPUTERS

Date of Lecture:

Topic of Lecture: Basic Operational concepts

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.:

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

Computer Architecture By Bester field , page no: 7

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no:6-9

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 03

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.VINUPRIYA

Unit : I - BASIC STRUCTURE OF COMPUTERS

Date of Lecture:

Topic of Lecture: Bus Structures

Introduction:

A bus is a group of lines that serves as a connecting path for several devices. A bus may be lines or wires. The lines carry data or address or control signal. There are 2 types of Bus structures: 1) Single Bus Structure and 2) Multiple Bus Structure.

Learning objective:

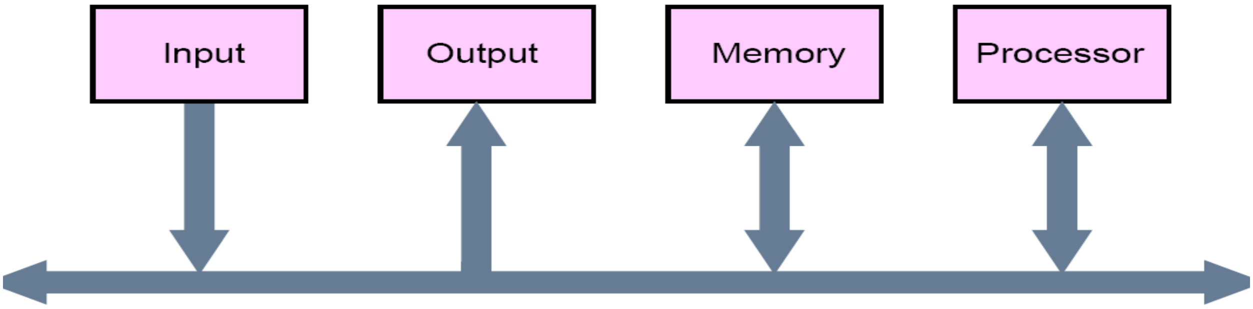
- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Evolution of Computer Systems
- Basic Operation of a Computer

Detailed content of the Lecture:

A group of lines that serves a connecting path for several devices is called a bus. In addition to the lines that carry the data, the bus must have lines for address and control purposes. The simplest way to interconnect functional units is to use a single bus, as shown below



Single Bus Structure

Because the bus can be used for only one transfer at a time, only 2 units can actively use the bus at any given time. Bus control lines are used to arbitrate multiple requests for use of the bus.

Drawbacks: The devices connected to a bus vary widely in their speed of operation.
Some devices are relatively slow, such as printer and keyboard.
Some devices are considerably fast, such as optical disks.
Memory and processor units are the fastest parts of a computer.

Efficient transfer mechanism thus is needed to cope with this problem. A common approach is to include buffer registers with the devices to hold the information during transfers. An approach is to use two-bus structure and an additional transfer mechanism.

Advantages: Low cost & Flexibility for attaching peripheral devices.

Multiple Bus Structure

Systems that contain multiple buses achieve more concurrency in operations. Two or more transfers can be carried out at the same time.

Advantage: Better performance.

Disadvantage: Increased cost.

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

https://www.youtube.com/watch?v=TgYAj7mlRT8

https://www.youtube.com/watch?v=4w_5Wp9IrJM

https://www.youtube.com/watch?v=uHqbE3JiDPY

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

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

Computer Architecture By Bester field, page no: 9

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 9-10

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 - 04

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.VINUPRIYA**

Unit : **I - BASIC STRUCTURE OF COMPUTERS**

Date of Lecture:

Topic of Lecture: Software Performance

Introduction:

The most important measure of the performance of a computer is how quickly it can execute programs. The speed with which a computer executes program is affected by the design of its hardware.

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:

PERFORMANCE

- The most important measure of performance of a computer is how quickly it can execute programs.
- The speed of a computer is affected by the design of
  1) Instruction-set.
  2) Hardware & the technology in which the hardware is implemented.
  3) Software including the operating system.
- Because programs are usually written in a HLL, performance is also affected by the compiler that translates programs into machine language. (HLL à High Level Language).
- For best performance, it is necessary to design the compiler, machine instruction set and hardware in a coordinated way.

PROCESSOR CLOCK

- Processor circuits are controlled by a timing signal called a **Clock**.
- The clock defines regular time intervals called **Clock Cycles**.
- Relation between P and R is given by

$$R = \frac{1}{P}$$

- R is measured in cycles per second.

BASIC PERFORMANCE EQUATION

- Let T = Processor time required to executed a program. N = Actual number of instruction executions. S = Average number of basic steps needed to execute one machine instruction. R = Clock rate in cycles per second.
- The program execution time is given by

$$T = \frac{N \times S}{R}$$

PIPELINING AND SUPER SCALAR OPERATIONS

- A substantial improvement in performance can be achieved by overlapping the execution of successive instructions using a technique called pipelining
- A higher degree of concurrency can be achieved if multiple instructions pipelines are implemented in the processor.
- Multiple functional units are used creating parallel paths through which different instructions can be executed in parallel
- Hence, it becomes possible to start the execution of several instructions in every clock cycle. This mode of operation is called superscalar execution

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

https://www.youtube.com/watch?v=TgYAj7mlRT8

https://www.youtube.com/watch?v=4w_5Wp9IrJM

https://www.youtube.com/watch?v=uHqbE3JiDPY

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

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

Computer Architecture By Bester field, page no: 9

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 12-16

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 05

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.VINUPRIYA

Unit : I - BASIC STRUCTURE OF COMPUTERS

Date of Lecture:

Topic of Lecture: Memory Location & Addresses

Introduction:

Memory consists of many millions of storage cells (flip-flops). Each cell can store a bit of information i.e. 0 or 1. Each group of n bits is referred to as a word of information, and n is called the **word length**. The word length can vary from 8 to 64 bits. A unit of 8 bits is called a **byte**.

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:

Accessing the memory to store or retrieve a single item of information (word/byte) requires distinct addresses for each item location. (It is customary to use numbers from 0 through 2^k-1 as the addresses of successive-locations in the memory). If 2^k = no. of addressable locations; then 2^k addresses constitute the address-space of the computer. For example, a 24-bit address generates an address-space of 2^{24} locations (16 MB).

BYTE-ADDRESSABILITY

- In byte-addressable memory, successive addresses refer to successive byte locations in the memory.
- Byte locations have addresses 0, 1, 2. . .. .
- If the word-length is 32 bits, successive words are located at addresses 0, 4, 8. . with each word having 4 bytes.
-

BIG-ENDIAN & LITTLE-ENDIAN ASSIGNMENTS

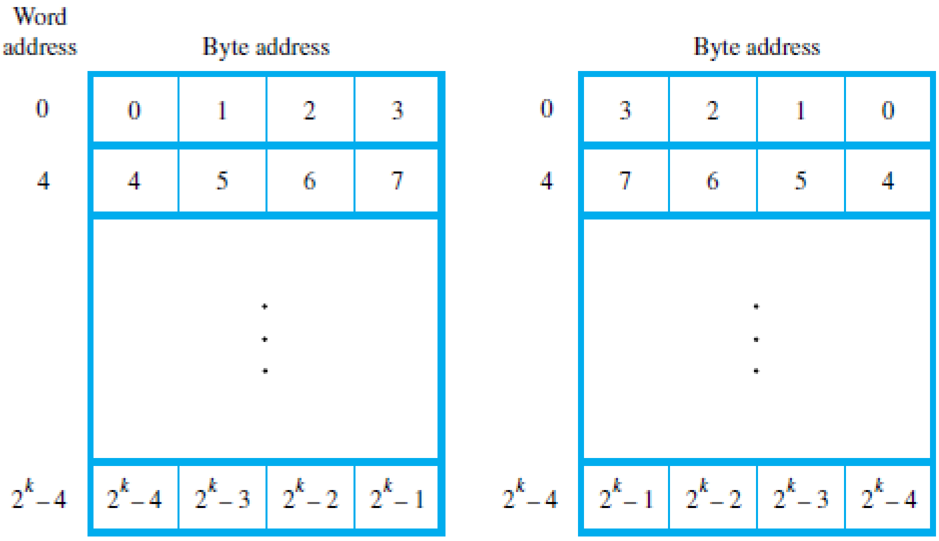
- There are two ways in which byte-addresses are arranged (Figure 2.3).

1) **Big-Endian:** Lower byte-addresses are used for the more significant bytes of the word.

2) **Little-Endian:** Lower byte-addresses are used for the less significant bytes of the

word

- In both cases, byte-addresses 0, 4, 8. . .. . are taken as the addresses of successive words in the memory.



(a) Big-endian assignment

(b) Little-endian assignment

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

https://www.youtube.com/watch?v=E1vUQEjNZr0

https://www.youtube.com/watch?v=0kvAkJ8vx9c

https://www.youtube.com/watch?v=F0Ri2TpRBBg

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

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

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 22-26

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 - 06

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.VINUPRIYA**

Unit : **I - BASIC STRUCTURE OF COMPUTERS**

Date of Lecture:

Topic of Lecture: Memory Operations

Introduction:

Memory Operations. There are two key **operations** on **memory**: fetch (address) returns value without changing the value stored at that address. store (address, value) writes new value into the cell at the given address.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory Locations & Addresses

Detailed content of the Lecture:

The **Load operation** transfers a copy of the contents of a specific memory-location to the processor. The memory contents remain unchanged.

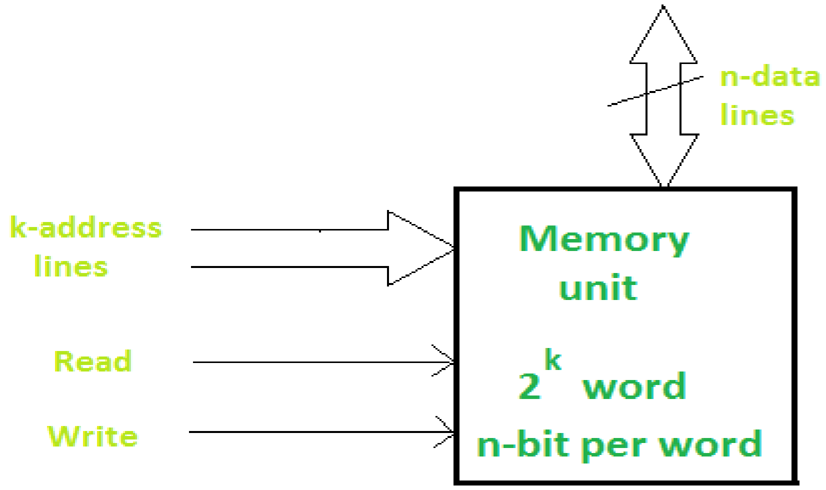
Steps for Load operation:

1) Processor sends the address of the desired location to the memory.
2) Processor issues „read‟ signal to memory to fetch the data.
3) Memory reads the data stored at that address.
4) Memory sends the read data to the processor.

The **Store operation** transfers the information from the register to the specified memory-location. This will destroy the original contents of that memory-location.

Steps for Store operation are:

1) Processor sends the address of the memory-location where it wants to store data.
2) Processor issues „write‟ signal to memory to store the data.
3) Content of register (MDR) is written into the specified memory-location.



LOAD and STORE OPERATION

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

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

https://www.youtube.com/watch?v=xgBeiLNKQ78

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

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

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

Computer Architecture By Bester field , page no: 36

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 26

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 - 07

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.VINUPRIYA**

Unit : **I - BASIC STRUCTURE OF COMPUTERS**

Date of Lecture:

Topic of Lecture: Instructions and Instruction Sequencing

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.:

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

Computer Architecture By Bester field , page no: 37

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 26-34.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L -08

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.VINUPRIYA

Unit : I - BASIC STRUCTURE OF COMPUTERS

Date of Lecture:

Topic of Lecture: Addressing modes, Assembly Language

Introduction:

Part of the programming flexibility for each processor is the number and different kind of ways the programmer can refer to data stored in the memory or I/O device. The different ways that a processor can access date are referred to as addressing schemes or addressing modes.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory Operations
- Memory Locations & Addresses

Detailed content of the Lecture:

IMPLEMENTATION OF VARIABLE AND CONSTANTS

- **Variable** is represented by allocating a memory-location to hold its value.
- Thus, the value can be changed as needed using appropriate instructions.
- There are 2 accessing modes to access the variables:
  1) Register Mode
  2) Absolute Mode

Register Mode

- The operand is the contents of a register.
- The name (or address) of the register is given in the instruction.
- Registers are used as temporary storage locations where the data in a register are accessed.
- For example, the instruction

Move R1, R2; Copy content of register R1 into register R2.

Absolute (Direct) Mode

- The operand is in a memory-location.
- The address of memory-location is given explicitly in the instruction.
- The absolute mode can represent global variables in the program.
- For example, the instruction

Move LOC, R2; Copy content of memory-location LOC into register R2.

Immediate Mode

- The operand is given explicitly in the instruction.
- For example, the instruction

  Move #200, R0; Place the value 200 in register R0.
- Clearly, the immediate mode is only used to specify the value of a source-operand.

Indirect Mode

- The EA of the operand is the contents of a register (or memory-location).
- The register (or memory-location) that contains the address of an operand is called a **Pointer**.
- We denote the indirection by

  → name of the register or

  → new address given in the instruction.

INDEXING AND ARRAYS

- A different kind of flexibility for accessing operands is useful in dealing with lists and arrays.

Index mode

- The operation is indicated as X(Ri)

  where X=the constant value which defines an offset (also called a displacement).

  Ri=the name of the index registers which contains address of a new location.
- The effective-address of the operand is given by EA=X+[Ri]

RELATIVE MODE

- This is similar to index-mode with one difference:

  The effective-address is determined using the PC in place of the general-purpose register Ri.
- The operation is indicated as X(PC).

ADDITIONAL ADDRESSING MODES

1. Auto Increment Mode
2. Auto Decrement Mode

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

https://www.youtube.com/watch?v=3QAdd1ZiIVo

https://www.youtube.com/watch?v=1VnVLJR7EZI

https://www.youtube.com/watch?v=KQ72sSKy8B8

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

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

Computer Architecture By Bester field, page no: 48

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 34-41.

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

DESIGNING YOUR FUTURE
Estd. 2000

IQAC

LECTURE HANDOUTS

L - 09

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.VINUPRIYA

Unit : I - BASIC STRUCTURE OF COMPUTERS

Date of Lecture:

Topic of Lecture: Basic I/O Operations, Stacks and Queues

Introduction:

The problem of moving a character-code from the keyboard to the processor. For this transfer, buffer-register DATAIN & a status control flags (SIN) are used.

Learning objective:

- To understand the basic structure of a digital computer

Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Instructions and instruction sequencing

Detailed content of the Lecture:

BASIC INPUT/OUTPUT OPERATIONS

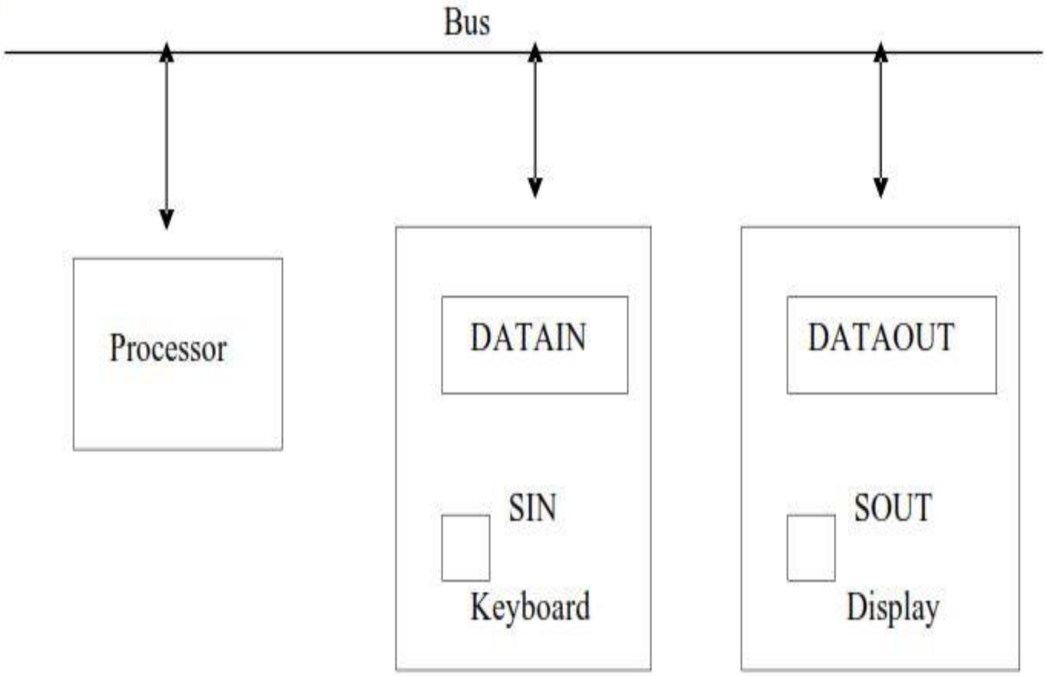
- A simple way of performing I/O tasks is to use a method known as program-controlled I/O
- The rate of data transfer from the keyboard to a computer is limited by the typing speed of the user, which is unlikely to exceed a few characters per second.
- The rate of output transfers from the computer to the display is much higher.
- It is determined by the rate at which characters can be transmitted over the link between the computer and the display device, typically several thousand characters per second.
- However, this is still much slower than the speed of a processor that can execute many millions of instructions per second.
- The difference in speed between the processor and I/O devices creates the need for mechanisms to synchronize the transfer of data between them

- When a key is pressed, the corresponding ASCII code is stored in a **DATAIN** register associated with the keyboard.
  - ➢ **SIN=1,** When a character is typed in the keyboard. This informs the processor that a valid character is in DATAIN.
  - ➢ **SIN=0,** When the character is transferred to the processor.
- An analogous process takes place when characters are transferred from the processor to the display. For this transfer, buffer-register DATAOUT & a status control flag

SOUT are used.

- **SOUT=1,** When the display is ready to receive a character.
- **SOUT=0,** When the character is being transferred to DATAOUT.

• The buffer registers DATAIN and DATAOUT and the status flags SIN and SOUT are part of circuitry commonly known as a **device interface**.



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.:

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

Computer Architecture By bester field ,page no: 64

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 47-50.

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

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: Addition and Subtraction of Signed Numbers

Introduction:

The addition, Subtraction, multiplication and division are the four basic arithmetic instructions and performed generally on binary or decimal data. Fixed-point numbers are used to represent integers or fractions. We can have signed or unsigned negative 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:

ADDITION & SUBTRACTION OF SIGNED NUMBERS

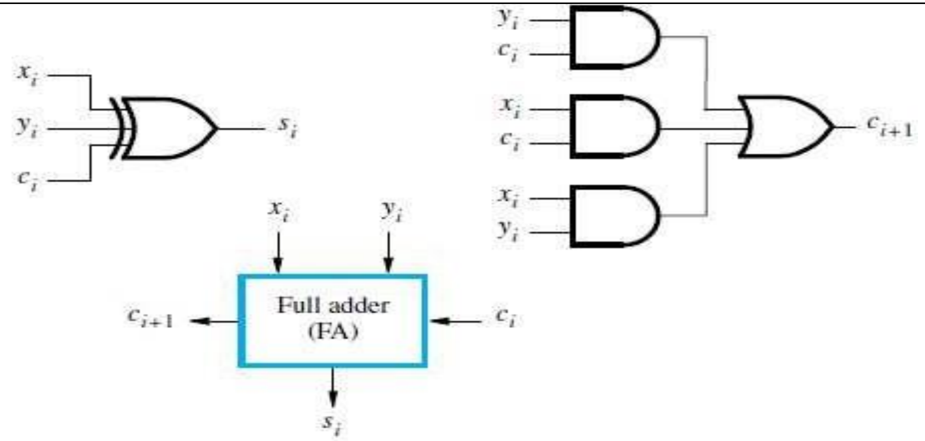
- Following are the two rules for addition and subtraction of n-bit signed numbers using the 2's complement representation system.

Rule 1:

- **To Add** two numbers, add their n-bits and ignore the carry-out signal from the MSB position.
- Result will be algebraically correct, if it lies in the range -2^{n-1} to $+2^{n-1}-1$.

Rule 2:

- **To Subtract** two numbers X and Y (that is to perform X-Y), take the 2's complement of Y and then add it to X as in rule 1.
- Result will be algebraically correct, if it lies in the range (2^{n-1}) to $+(2^{n-1}-1)$.



(a) Logic for a single stage

ADDITION/SUBTRACTION LOGIC UNIT

- The n-bit adder can be used to add 2's complement numbers X and Y.
- **Overflow** can only occur when the signs of the 2 operands are the same.

  Control-line=0 for addition, applying the Y vector unchanged to one of the adder inputs. Control-line=1 for subtraction, the Y vector is 2's complemented.

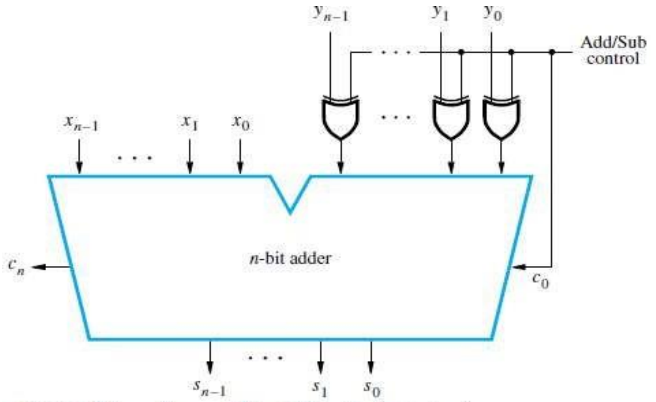


Figure 9.3 Binary addition/subtraction logic circuit.

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

https://www.youtube.com/watch?v=bQOOsrE11gc&t=67s

https://www.youtube.com/watch?v=o-WXqnagg0c&t=23s

https://www.youtube.com/watch?v=qk1rrfEWyGU

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

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

Computer Architecture By bester field: page no: 368

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 260-264.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 11

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: DESIGN OF FAST ADDERS

Introduction:

Two approaches can be used to reduce delay in adders:

1. Use the fastest possible electronic-technology in implementing the ripple-carry design.
2. Use an augmented logic-gate network structure.

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:

CARRY-LOOKAHEAD ADDITIONS

- The logic expression for s_i(sum) and c_{i+1}(carry-out) of stage i are

  $s_i=x_i+y_i+c_i$ ------(1) $c_{i+1}=x_iy_i+x_ic_i+y_ic_i$---------(2)

- Factoring (2) into

  $c_{i+1}=x_i\ \ y_i+(x_i+y_i)c_i$ we can write

  $c_{i+1}=G_i+P_iC_i$ where $G_i=x_iy_i$ and $P_i=x_i+y_i$

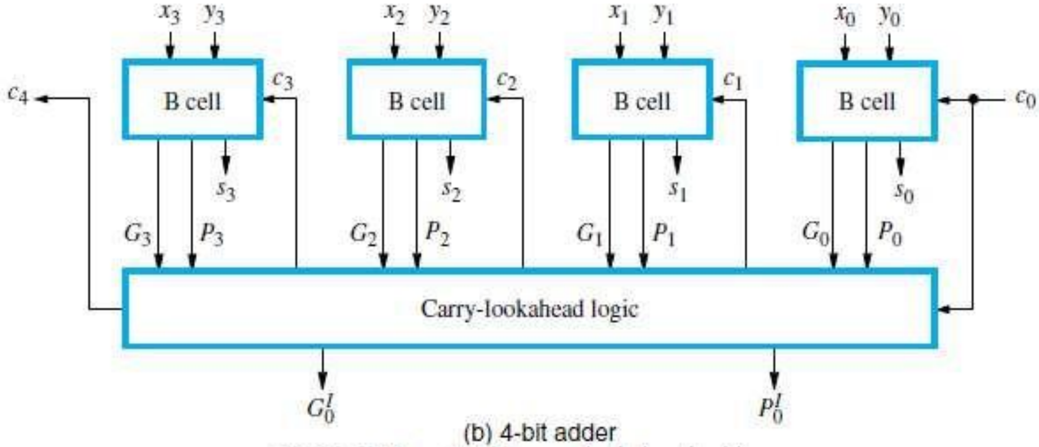
- The expressions G_i and P_i are called generate and propagate functions (Figure 9.4).
- If G_i=1, then c_{i+1}=1, independent of the input carry c_i. This occurs when both x_i and y_i are 1. Propagate function means that an input-carry will produce an output-carry when either x_i=1 or y_i=1.
- All G_i and P_i functions can be formed independently and in parallel in one logic-gate delay.
- Expanding c_i terms of i-1 subscripted variables and substituting into the c_{i+1} expression, we obtain $c_{i+1}=G_i+P_iG_{i-1}+P_iP_{i-1}G_{i-2.}\ldots\ldots+P_1G_0+P_iP_{i-1}\ \ P_0c_0$
- Conclusion: Delay through the adder is 3 gate delays for all carry-bits & 4 gate delays for all sum-bits.
- Consider the design of a 4-bit adder. The carries can be implemented as $c_1=G_0+P_0c_0$

  $c_2=G_1+P_1G_0+P_1P_0c_0$

  $c_3=G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0c_0$

  $c_4=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3$
  $P_2P_1P_0c_0$

- The carries are implemented in the block labeled carry-lookahead logic. An adder implemented in this form is called a **Carry-Lookahead Adder**.



(b) 4-bit adder

Figure 9.4 A 4-bit carry-lookahead adder.

HIGHER-LEVEL GENERATE & PROPAGATE FUNCTIONS

- 16-bit adder can be built from four 4-bit adder blocks (Figure 9.5).
- These blocks provide new output functions defined as G_k and P_k, where k=0 for the first 4-bit block, k=1 for the second 4-bit block and so on.
- In the first block,

$P_0=P_3P_2P_1P_0$ & $G_0=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0$

- The first-level G_i and P_i functions determine whether bit stage i generates or propagates a carry, and the second level G_k and P_k functions determine whether block k generates or propagates a carry.
- Carry c_{16} is formed by one of the carry-lookahead circuits as

$c_{16}=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0c_0$

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

https://www.youtube.com/watch?v=_bJ53XErKY8

https://www.youtube.com/watch?v=lQGyLRM8cXA

https://www.youtube.com/watch?v=Vp9J61ag5n4

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

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

Computer Architecture By bester field, page no: 371

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 260.

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 - 12

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: MULTIPLICATION OF POSITIVE NUMBERS

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:

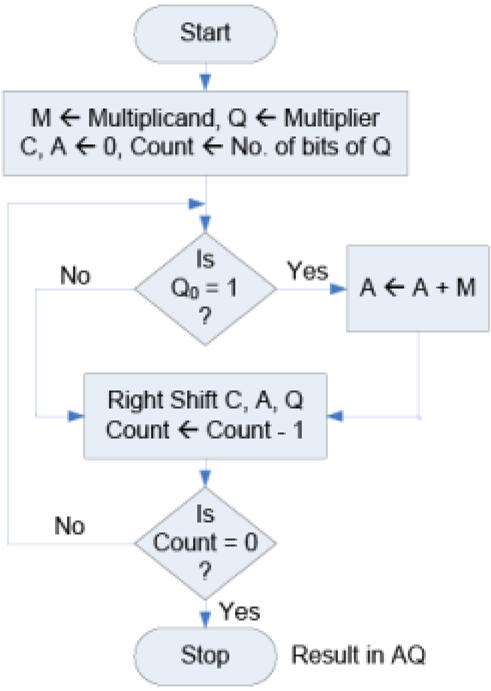
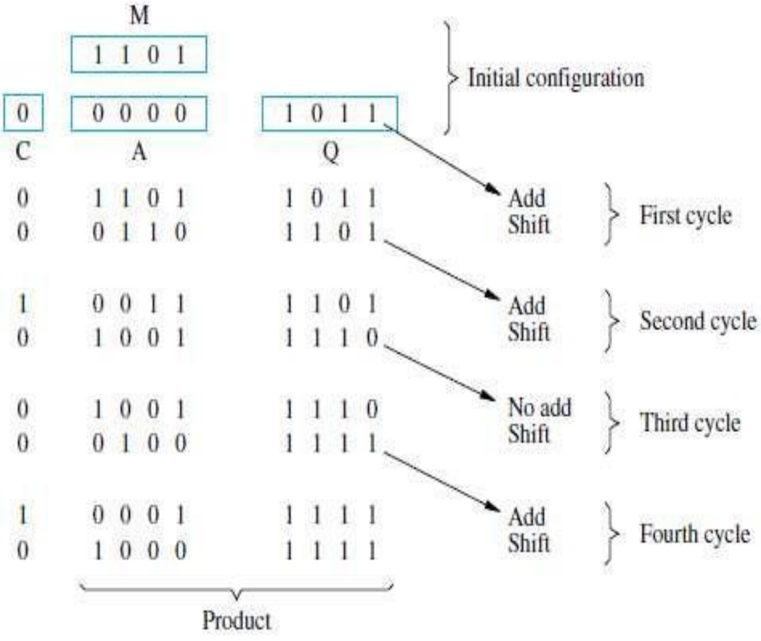


Fig. : Flowchart of Unsigned Binary Multiplication



C	A	Q		
	1 1 0 1 (M)			
0	0 0 0 0	1 0 1 1		Initial configuration
0	1 1 0 1	1 0 1 1	Add	First cycle
0	0 1 1 0	1 1 0 1	Shift	
1	0 0 1 1	1 1 0 1	Add	Second cycle
0	1 0 0 1	1 1 1 0	Shift	
0	1 0 0 1	1 1 1 0	No add	Third cycle
0	0 1 0 0	1 1 1 1	Shift	
1	0 0 0 1	1 1 1 1	Add	Fourth cycle
0	1 0 0 0	1 1 1 1	Shift	

(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/

Computer Architecture By bester field, page no: 376

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 271-273.

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

LECTURE HANDOUTS

L - 13

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: SIGNED OPERAND MULTIPLICATION

Introduction:

A technique which works equally well for both negative and positive multiplier, called Booth Algorithm. There are two techniques for speeding up the multiplication process. In first technique the maximum number of summands are reduced to n/2 for n-bit operands. The second technique, called the carry save addition reduces the time needed to add the summand.

Learning objective:

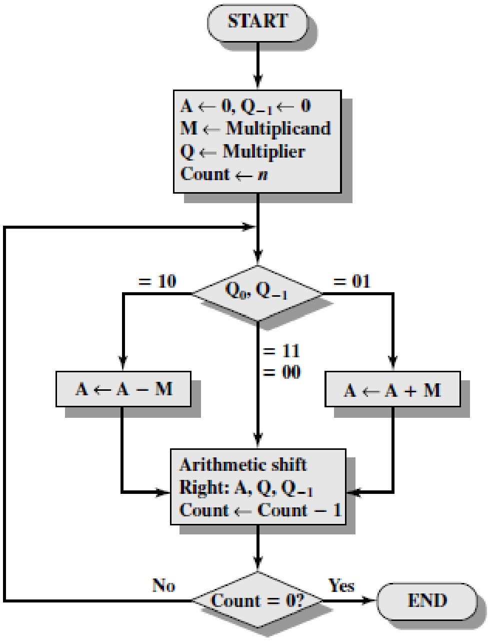
- To discuss the operation of various components of computing systems

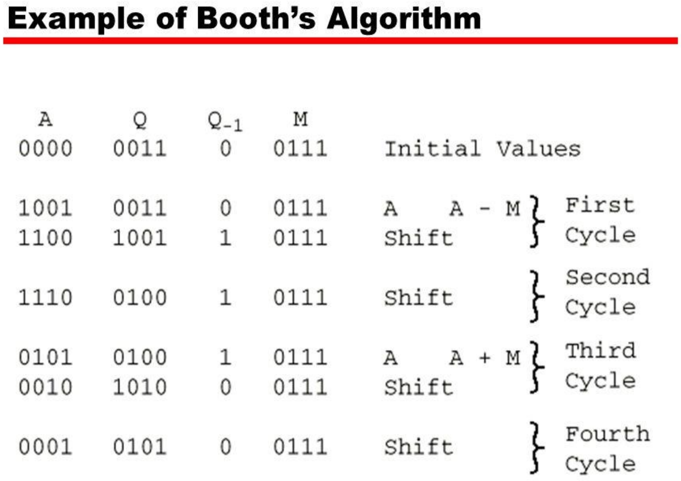
Prerequisite knowledge for Complete understanding and learning of Topic:

- Basic operational concepts
- Addition and Subtraction

Detailed content of the Lecture:

BOOTH'S ALGORITHM





Example of Booth's Algorithm

A	Q	Q_{-1}	M		
0000	0011	0	0111	Initial Values	
1001	0011	0	0111	A A - M	First Cycle
1100	1001	1	0111	Shift	
1110	0100	1	0111	Shift	Second Cycle
0101	0100	1	0111	A A + M	Third Cycle
0010	1010	0	0111	Shift	
0001	0101	0	0111	Shift	Fourth Cycle

FAST MULTIPLICATION

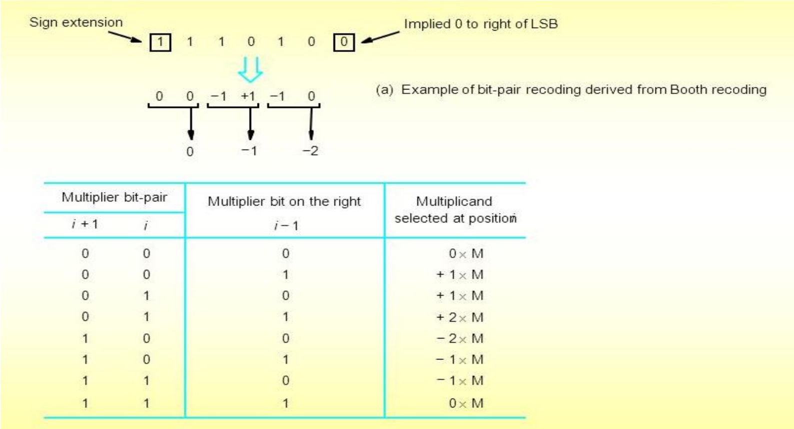
BIT-PAIR RECODING OF MULTIPLIERS

• This method

→ derived from the booth algorithm

→ reduces the number of summands by a factor of 2

- Group the Booth-recoded multiplier bits in pairs.
- The pair (+1 -1) is equivalent to the pair (0 +1).



(a) Example of bit-pair recoding derived from Booth recoding

Multiplier bit-pair		Multiplier bit on the right	Multiplicand selected at position
i + 1	*i*	*i* − 1	
0	0	0	0 × M
0	0	1	+ 1 × M
0	1	0	+ 1 × M
0	1	1	+ 2 × M
1	0	0	− 2 × M
1	0	1	− 1 × M
1	1	0	− 1 × M
1	1	1	0 × M

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

https://www.youtube.com/watch?v=DIp4GqSCZho

https://www.youtube.com/watch?v=cWfaw7b3jKY

https://www.youtube.com/watch?v=gdV-uaPkmxk

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

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

Computer Architecture By bester field Text Book 1: page no: 380

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 273-277.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 14

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.Vinupriya

Unit : II - Arithmetic and Logic Unit Date of Lecture:

Topic of Lecture: Fast Multiplication

Introduction:

The addition, Subtraction, multiplication and division are the four basic arithmetic instructions and performed generally on binary or decimal data. Fixed-point numbers are used to represent integers or fractions. We can have signed or unsigned negative numbers.

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

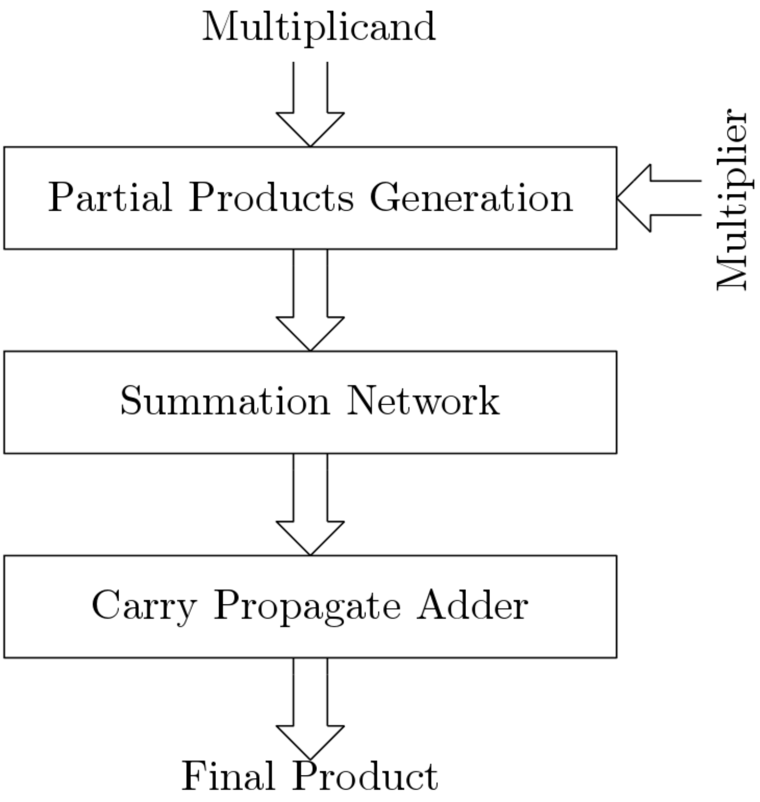
- Basic operational concepts
- Addition and Subtraction

Detailed content of the Lecture:

The fast multiplication can be achieved in three general ways.

1. The sequential multipliers sequentially generates the partial products and adds them with the previously stored partial products.
2. In the second method, high speed parallel multipliers generate the partial products in parallel and adds them by a fast multi-operand adder.
3. The third method corresponds to use of array of identical blocks that generates and adds the partial products simultaneously.

- The sequential multipliers have to work at high frequency and thus consumes high power.
- The performance of the parallel multipliers depends on the performance of multi-operand adder and also on the optimal number of partial products.
- The third method is actually the array multiplier having high hardware complexity. An obvious method to design a fast multiplier is to reduce the partial products and then applying fast addition methods.
- Here we will first discuss the methods to reduce the partial products and then fast multi-operand addition techniques will be discussed.
- All the steps are described below in details



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

https://digitalsystemdesign.in/fast-multiplication/

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

Computer Architecture By bester field: page no: 368

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 260-264.

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

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: INTEGER 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.

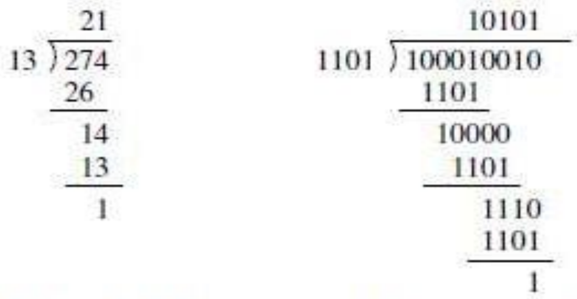


Figure 9.22 Longhand division examples.

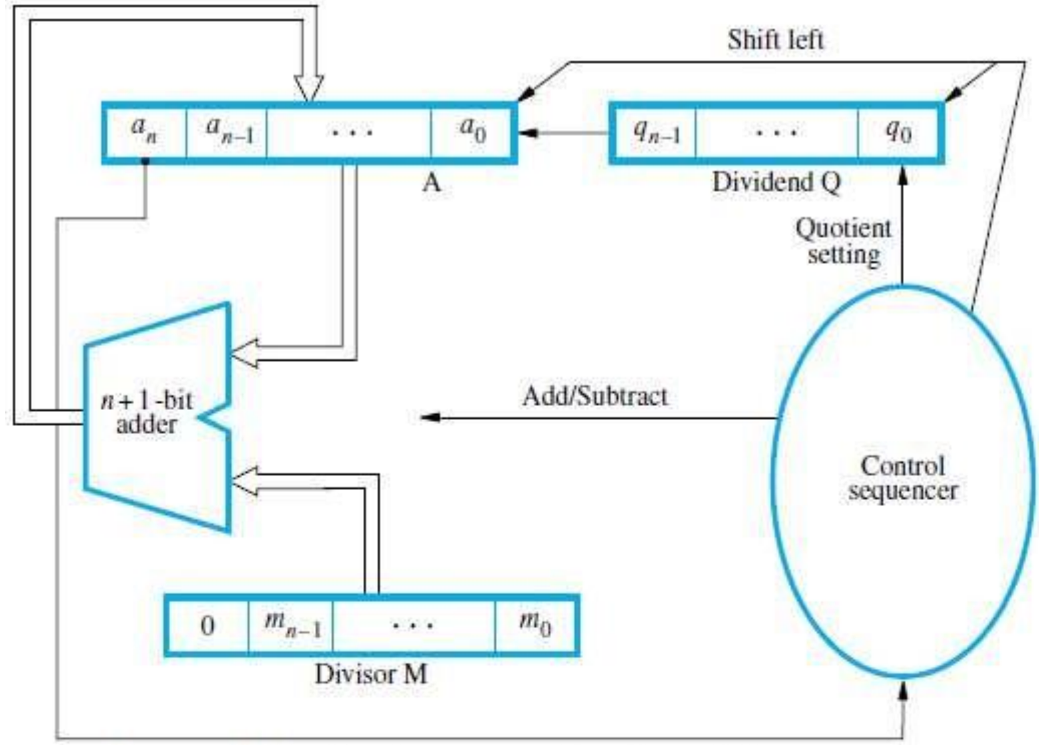


Figure 9.23 Circuit arrangement for binary division.

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/

Computer Architecture By bester field , page no: 390

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 281-285.

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 - 16

CSE **II/III/A**

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: INTEGER DIVISION

Introduction:

A division algorithm provides a quotient and a remainder when we divide two number. They are generally of two type slow algorithm and fast algorithm. Slow division algorithm is restoring, non-restoring, non-performing restoring, SRT algorithm and under fast comes Newton–Raphson and Goldschmidt.

Learning objective:

- To discuss the operation of various components of computing systems

Prerequisite knowledge for Complete understanding and learning of Topic:

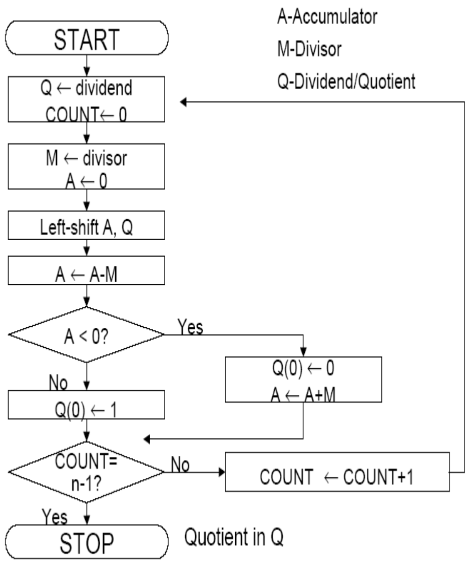
- Basic operational concepts
- Addition and Subtraction
- Multiplication

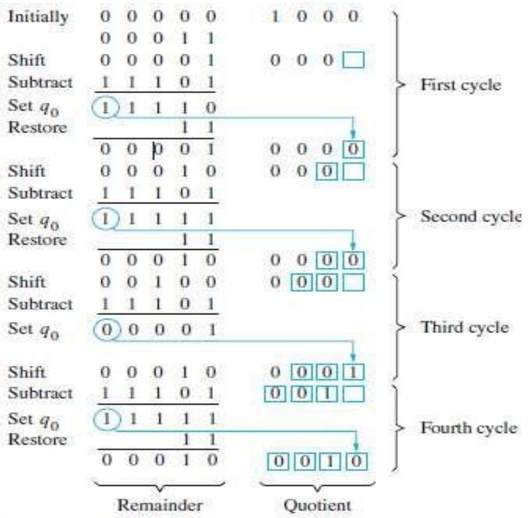
Detailed content of the Lecture: Detailed content of the Lecture:

RESTORING DIVISION

Procedure: Do the following n times

1. Shift A and Q left one binary position.
2. Subtract M from A, and place the answer back in A.
3. If the sign of A is 1, set q_0 to 0 and add M back to A (restore A).
4. If the sign of A is 0, set q_0 to 1 and no restoring done.





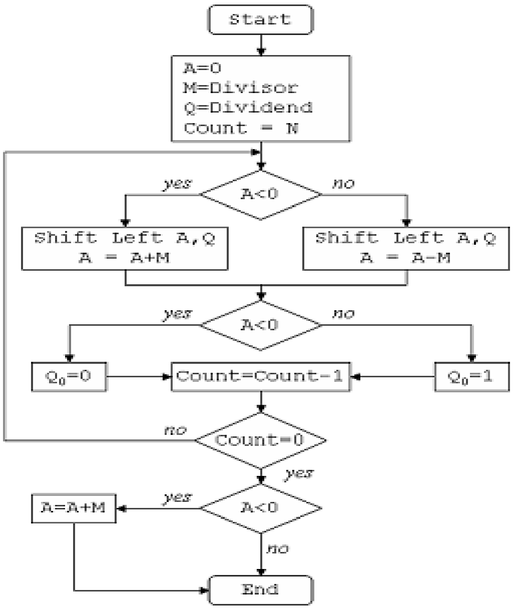
```
Initially   0 0 0 0 0     1 0 0 0
            0 0 0 1 1
Shift       0 0 0 0 1     0 0 0 □
Subtract    1 1 1 0 1                   First cycle
Set q0      1 1 1 1 0
Restore           1 1
            0 0 p 0 1     0 0 0 0
Shift       0 0 0 1 0     0 0 0 □
Subtract    1 1 1 0 1
Set q0      1 1 1 1 1                   Second cycle
Restore           1 1
            0 0 0 1 0     0 0 0 0
Shift       0 0 1 0 0     0 0 0 □
Subtract    1 1 1 0 1                   Third cycle
Set q0      0 0 0 0 1

Shift       0 0 0 1 0     0 0 0 1
Subtract    1 1 1 0 1     0 0 1 □
Set q0      1 1 1 1 1                   Fourth cycle
Restore           1 1
            0 0 0 1 0     0 0 1 0
            Remainder     Quotient
```

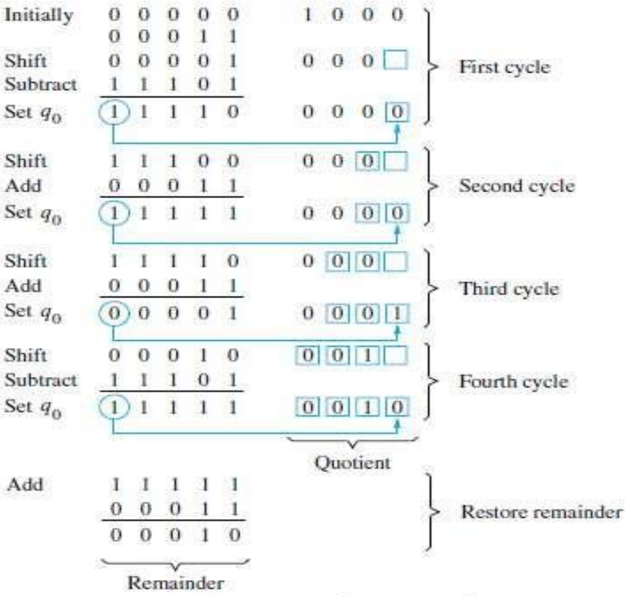
Figure 9.24 A restoring division example.

NON-RESTORING DIVISION

Procedure: Do the following n times

1. If the sign of A is 0, shift A and Q left one-bit position and subtract M from A; otherwise, shift A and Q left and add M to A.
2. Now, if the sign of A is 0, set q_0 to 1; otherwise set q_0 to 0.
3. If the sign of A is 1, add M to A (restore).





```
Initially   0 0 0 0 0     1 0 0 0
            0 0 0 1 1
Shift       0 0 0 0 1     0 0 0 □      First cycle
Subtract    1 1 1 0 1
Set q0      1 1 1 1 0     0 0 0 0

Shift       1 1 1 0 0     0 0 0 □      Second cycle
Add         0 0 0 1 1
Set q0      1 1 1 1 1     0 0 0 0

Shift       1 1 1 1 0     0 0 0 □      Third cycle
Add         0 0 0 1 1
Set q0      0 0 0 0 1     0 0 0 1

Shift       0 0 0 1 0     0 0 1 □      Fourth cycle
Subtract    1 1 1 0 1
Set q0      1 1 1 1 1     0 0 1 0
                          Quotient

Add         1 1 1 1 1                  Restore remainder
            0 0 0 1 1
            0 0 0 1 0
            Remainder
```

Figure 9.25 A non-restoring division example.

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

https://www.youtube.com/watch?v=6ToR6vuRb3M

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

https://www.youtube.com/watch?v=KoWT05pfd4k

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

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

Computer Architecture By Bester field ,page no: 390

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 281-285.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 17

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.Vinupriya

Unit : II - Arithmetic and Logic Unit Date of Lecture:

Topic of Lecture: FLOATING POINT NUMBERS AND OPERATIONS

Introduction:

2 parts

- Mantissa = signed fixed point number
- Exponent = the position of the decimal point

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:

IEEE STANDARD FOR FLOATING POINT NUMBERS

- Single precision representation occupies a single 32-bit word.

  The scale factor has a range of 2^{-126} to 2^{+127} (which is approximately equal to 10^{+38}).
- The 32-bit word is divided into 3 fields: sign (1 bit), exponent (8 bits) and mantissa (23 bits).
- Signed exponent=E.

  Unsigned exponent E'=E+127. Thus, E' is in the range 0<E'<255.
- The last 23 bits represent the mantissa. Since binary normalization is used, the MSB of the mantissa is always equal to 1. (M represents fractional-part).
- The 24-bit mantissa provides a precision equivalent to about 7 decimal-digits (Figure 9.24).
- Double precision representation occupies a single 64-bit word. And E' is in the range 1<E'<2046.
- The 53-bit mantissa provides a precision equivalent to about 16 decimal-digits.

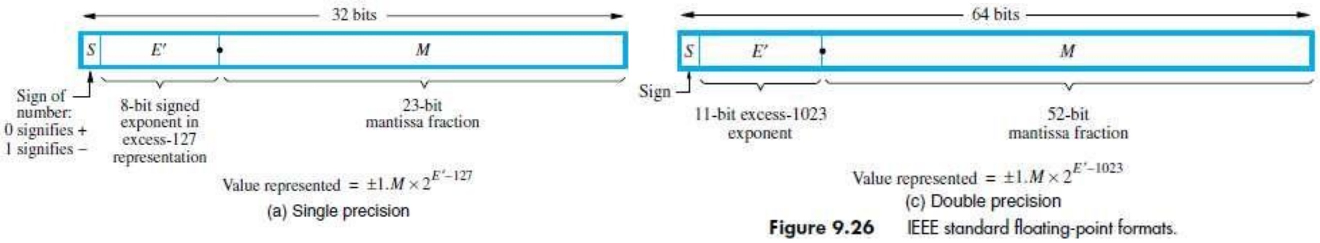

32 bits
S
E'
M
Sign of number: 0 signifies + 1 signifies −
8-bit signed exponent in excess-127 representation
23-bit mantissa fraction
Value represented = ±1.M × 2^{E'−127}
(a) Single precision
64 bits
S
E'
M
Sign
11-bit excess-1023 exponent
52-bit mantissa fraction
Value represented = ±1.M × 2^{E'−1023}
(c) Double precision

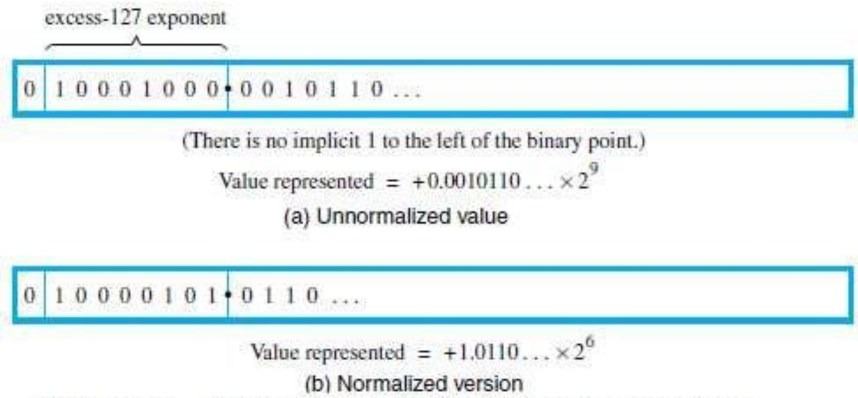
Figure 9.26 IEEE standard floating-point formats.

SPECIAL VALUES

- The end values 0 and 255 of the excess-127 exponent E′ are used to represent special values.
- When E′=0 and the mantissa fraction m is zero, the value exact 0 is represented.
- When E′=255 and M=0, the value ∞ is represented, where ∞ is the result of dividing a normal number by zero.
- when E′=0 and M! = -, denormal numbers are represented.
- When E′=255 and M! =0, the value represented is called not a number (NaN). A NaN is the result of performing an invalided operation such as 0/0 or $\sqrt{0}$.

NORMALIZATION

- When the decimal point is placed to the right of the first (non zero) significant digit, the number is said to be normalized.
- If a number is not normalized, it can always be put in normalized form by shifting the fraction and adjusting the exponent. As computations proceed, a number that does not fall in the representable range of normal numbers might be generated.
- In single precision, it requires an exponent less than -126 (underflow) or greater than +127 (overflow). Both are exceptions that need to be considered.



(There is no implicit 1 to the left of the binary point.)

Value represented = $+0.0010110\ldots \times 2^9$

(a) Unnormalized value

Value represented = $+1.0110\ldots \times 2^6$

(b) Normalized version

Figure 9.27 Floating-point normalization in IEEE single-precision format.

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

https://www.youtube.com/watch?v=iqMG00tGht0

https://www.youtube.com/watch?v=8afbTaA-gOQ

https://www.youtube.com/watch?v=ILKjj30pb9E

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

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

Computer Architecture By bester field , page no: 393

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 285-294.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 18

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: FLOATING POINT NUMBERS AND OPERATIONS

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.

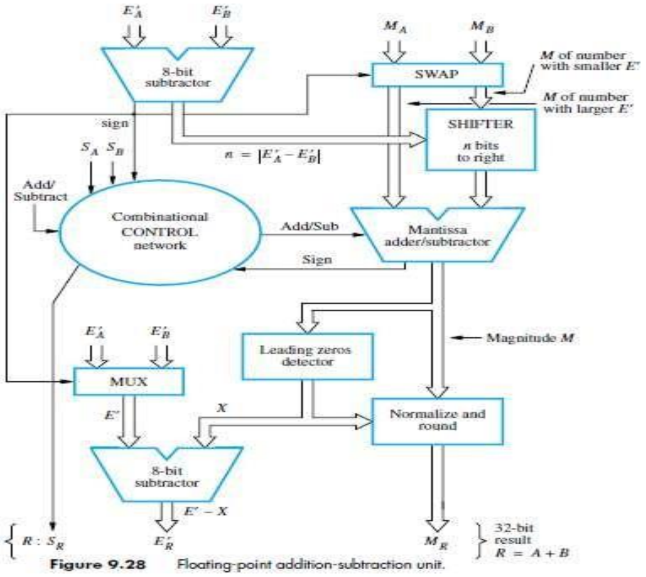


Figure 9.28 Floating-point addition-subtraction unit.

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/

Computer Architecture By bester field , page no: 393

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 285-294.

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 - 19

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: FUNDAMENTAL CONCEPTS

Introduction:

- Fetch contents of memory-location pointed to by PC. Content of this location is an instruction to be executed. The instructions are loaded into IR, Symbolically, this operation is written as:

  IR← [[PC]] Increment PC by 4 PC←[PC] +4. **[FETCH PHASE]**
- Carry out the actions specified by instruction (in the IR). **[EXECUTION PHASE]**

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Functional Units
- Basic Operational Concepts

Detailed content of the Lecture:

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

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

https://slideplayer.com/slide/9377818/

https://www.youtube.com/watch?v=-fUAdOt8iLI

https://slideplayer.com/slide/5116911/

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

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

Computer Architecture By bester field , page no: 412

Carl Hamacher, Zvonko Vranesic and safwat Zaky, " Computer Organization" Fourth edition , TMH 2002, Page no: 111-118.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 20

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: EXECUTION OF A COMPLETE INSTRUCTION

Introduction:

Let us find the complete control sequence for execution of the instruction Add $R_{1,}$ (R_2) for the single bus processor. The branch instruction loads the branch target address in PC so that PC will fetch the next instruction from the branch target address.

Learning objective:

- To enhance the processor operation by employing pipelining

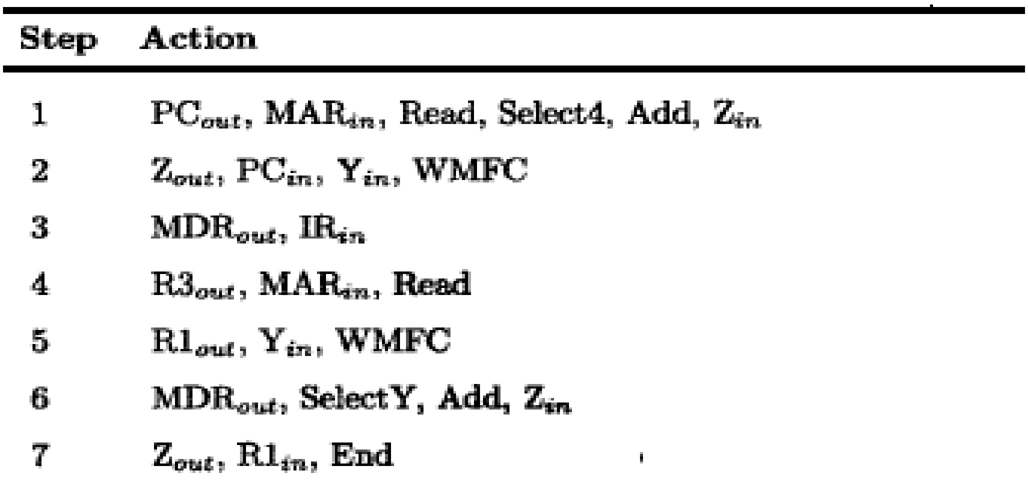
Prerequisite knowledge for Complete understanding and learning of Topic:

- Functional Units
- Fundamental concepts

Detailed content of the Lecture:

Executing this instruction requires the following actions:

1) Fetch the instruction.
2) Fetch the first operand.
3) Perform the addition &
4) Load the result into R1.



Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	$R3_{out}$, MAR_{in}, Read
5	$R1_{out}$, Y_{in}, WMFC
6	MDR_{out}, SelectY, Add, Z_{in}
7	Z_{out}, $R1_{in}$, End

Figure 7.6 Control sequence for execution of the instruction Add (R3),R1

- Instruction execution proceeds as follows:

Step1--> The instruction-fetch operation is initiated by

→ loading contents of PC into MAR &

→ sending a Read request to memory.

The Select signal is set to Select4, which causes the Mux to select constant 4. This value

is added to operand at input B (PC‟s content), and the result is stored in Z.

Step2--> Updated value in Z is moved to PC. This completes the PC increment operation and PC will now point to next instruction.

Step3--> Fetched instruction is moved into MDR and then to IR. The step 1 through 3 constitutes the **Fetch Phase**.

At the beginning of step 4, the instruction decoder interprets the contents of the IR. This enables the control circuitry to activate the control-signals for steps 4 through 7.

The step 4 through 7 constitutes the **Execution Phase**.

Step4--> Contents of R3 are loaded into MAR & a memory read signal is issued. Step5--> Contents of R1 are transferred to Y to prepare for addition.

Step6--> When Read operation is completed, memory-operand is available in MDR, and the addition is performed.

Step7--> Sum is stored in Z, then transferred to R1. The End signal causes a new instruction fetch cycle to begin by returning to step1.

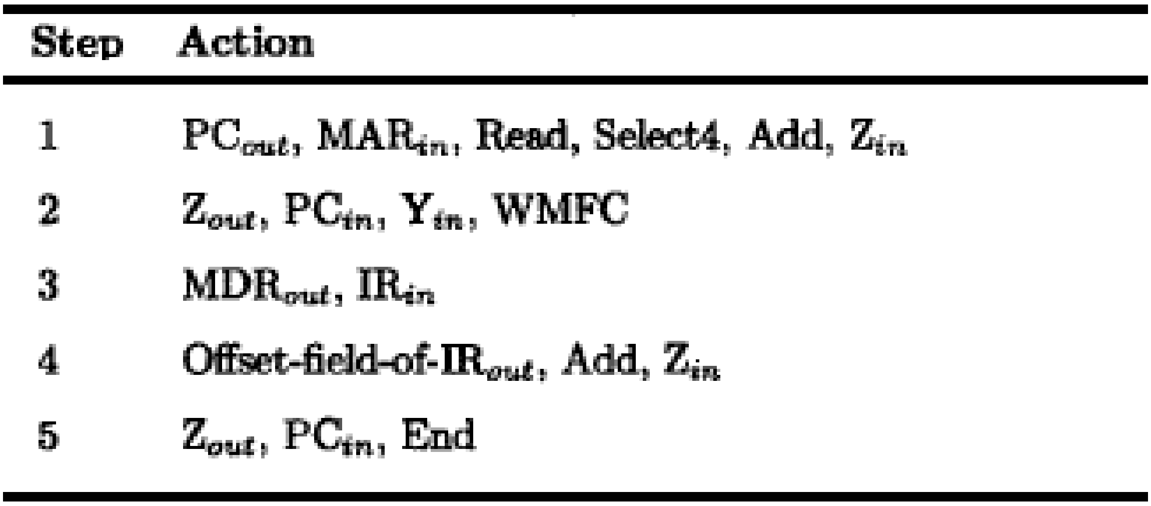
BRANCHING INSTRUCTIONS

- Control sequence for an **unconditional branch instruction** is as follows:
- Instruction execution proceeds as follows:

Step 1-3--> The processing starts & the fetch phase ends in step3.

Step 4--> The offset-value is extracted from IR by instruction-decoding circuit. Since the updated value of PC is already available in register Y, the offset X is gated onto the bus, and an addition operation is performed.

Step 5--> the result, which is the branch-address, is loaded into the PC.



Step	Action
1	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
2	Z_{out}, PC_{in}, Y_{in}, WMFC
3	MDR_{out}, IR_{in}
4	Offset-field-of-IR_{out}, Add, Z_{in}
5	Z_{out}, PC_{in}, End

Figure 7.7 Control sequence for an unconditional Branch instruction.

- In case of **conditional branch**,

we have to check the status of the condition-codes before loading a new value into the PC. e.g.: Offset-field-of-IR$_{out}$, Add, $Z_{in,}$ If N=0 then End

If N=0, processor returns to step 1 immediately after step 4. If N=1, step 5 is performed to load a new value into PC.

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

https://www.youtube.com/watch?v=zcxrrOvZW3s

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

https://www.youtube.com/watch?v=QNXEiURM2Fk

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

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

Computer Architecture By bester field Text Book 1: page no: 421

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 118-121.

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

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: MULTIPLE BUS ORGANIZATION

Introduction:

Because of single bus only one data word can be transferred over the bus in a clock cycle. This increases the steps required to complete the execution of the instruction. To reduce the number of steps needed to execute instructions, most commercial processors provide multiple internal paths that enable several transfers to take place in parallel.

Learning objective:

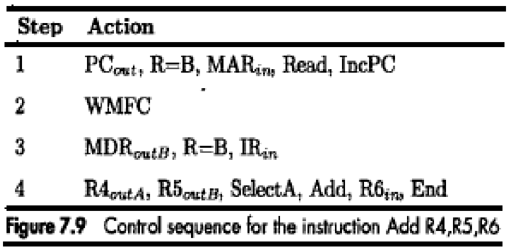
- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Fundamental Concepts
- Execution of a Complete Instruction

Detailed content of the Lecture:

- All general-purpose registers are grouped into a single block called the **Register File**.
- Register-file has 3 ports:
  1) Two output-ports allow the contents of 2 different registers to be simultaneously placed on buses A & B.
  2) Third input-port allows data on bus C to be loaded into a third register during the same clock-cycle.
- Buses A and B are used to transfer source-operands to A & B inputs of ALU.
- The result is transferred to destination over bus C.



Step	Action
1	PC_{out}, R=B, MAR_{in}, Read, IncPC
2	WMFC
3	MDR_{outB}, R=B, IR_{in}
4	$R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End

Figure 7.9 Control sequence for the instruction Add R4,R5,R6

- **Incrementor Unit** is used to increment PC by 4.
- Instruction execution proceeds as follows:

Step 1--> Contents of PC are

→ passed through ALU using R=B control-signal &

→ loaded into MAR to start memory Read operation. At the same time, PC is incremented by 4.

Step2--> Processor waits for MFC signal from memory.

Step3--> Processor loads requested-data into MDR, and then transfers them to IR. Step4--> The instruction is decoded and add operation takes place in a single step.

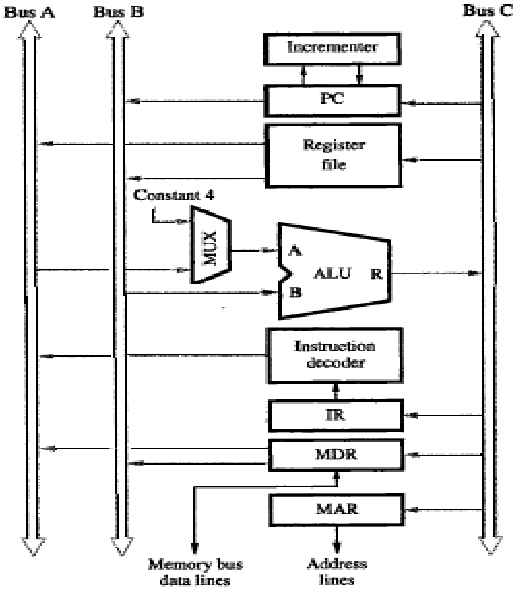


Figure 7.8 Three-bus organization of the datapath.

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

https://slideplayer.com/slide/7791731/

https://www.youtube.com/watch?v=TgYAj7mlRT8

https://www.youtube.com/watch?v=7TnL-BbnDP8

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

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

Computer Architecture By bester field , page no: 423

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 22

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.Vinupriya

Unit : II - Arithmetic and Logic Unit Date of Lecture:

Topic of Lecture: HARDWIRED CONTROL, MICROPROGRAMMED CONTROL

Introduction:

To execute instructions, the processor must have some means of generating the control-signals. There are two approaches for this purpose:

1) Hardwired control and 2) Microprogrammed control.

Learning objective:

- To enhance the processor operation by employing pipelining

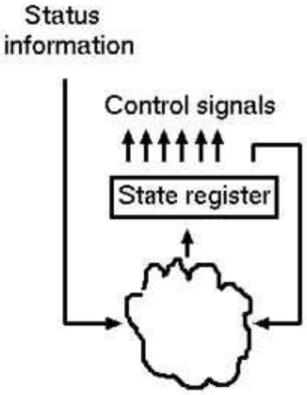
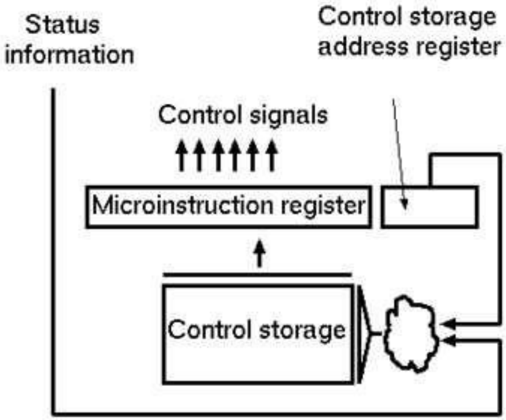
Prerequisite knowledge for Complete understanding and learning of Topic:

- Fundamental Concepts
- Execution of a Complete Instruction
- Hardwired Control, Microprogrammed Control

Detailed content of the Lecture:

HARDWIRED CONTROL VS MICROPROGRAMMED CONTROL

Attribute	Hardwired Control	Microprogrammed Control
Definition	Hardwired control is a control mechanism to generate control- signals by using gates, flip- flops, decoders, and other digital circuits.	Micro programmed control is a control mechanism to generate control-signals by using a memory called control store (CS), which contains the control-signals.
Speed	Fast	Slow
Control functions	Implemented in hardware.	Implemented in software.
Flexibility	Not flexible to accommodate new system specifications	More flexible, to accommodate new system specification or new instructions redesign is required.
Ability to handle large or complex instruction sets	Difficult.	Easier.
Ability to support operating systems & diagnostic features	Very difficult.	Easy.

Design process	Complicated.	Orderly and systematic.
Applications	Mostly RISC microprocessors.	Mainframes, some microprocessors.
Instruction set size	Usually under 100 instructions.	Usually over 100 instructions.
ROM size	-	2K to 10K by 20-400 bit microinstructions.
Chip area efficiency	Uses least area.	Uses more area.
Diagram		 Status information Control storage address register Control signals Microinstruction register Control storage

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

https://www.youtube.com/watch?v=7TnL-BbnDP8
https://www.youtube.com/watch?v=m9R8pEq75fU
https://www.youtube.com/watch?v=X6GbaLQUuz8

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

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

Computer Architecture By bester field , page no: 425

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 121-130.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 23

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: PIPELINING, BASIC CONCEPTS

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:

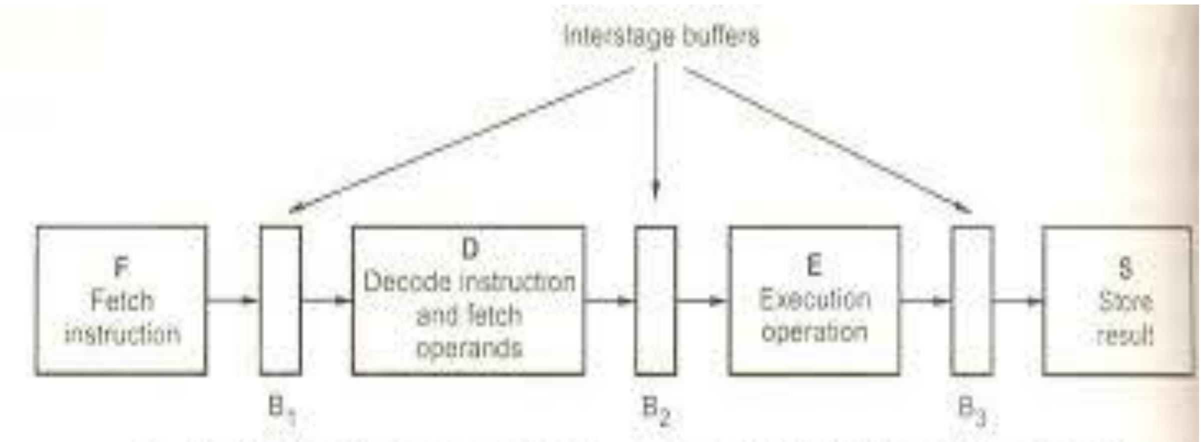
- Fundamental Concepts
- Execution of a Complete Instruction

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.

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.:

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

Computer Architecture By bester field , page no: 454

Carl Hamacher, Zvonko Vranesic and safwat Zaky, “ Computer Organization” Fourth edition , TMH 2002, Page no: 302-310.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 24

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: DATA HAZARDS, INSTRUCTION HAZARDS

Introduction:

Any location that causes the pipeline to stall is called hazard. There are two types of hazards namely,

- Data Hazards
- Instruction Hazards

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Pipelining and basic concepts
- Execution of a Complete Instruction

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

Data Hazard

A data hazard is any conditions in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result, some operation has to be delayed and the pipeline stalls.

TYPES OF DATA HAZARDS

- Read After Write (RAW) – true dependency

  A hazard occurs if the read occurs before the write is complete.
- Write After Read (WAR) – anti-dependency

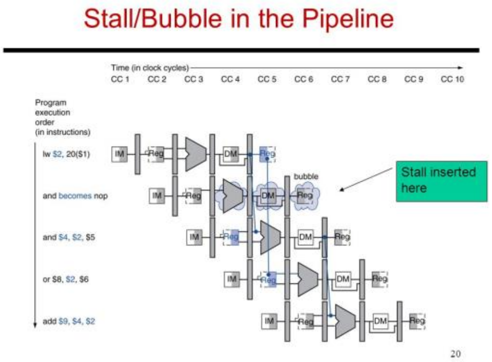
  A hazard occurs if the write occurs before the read happens.
- Write After Write (WAW) – output dependency

  A hazard occurs if the two write occurs in the reverse order than intended

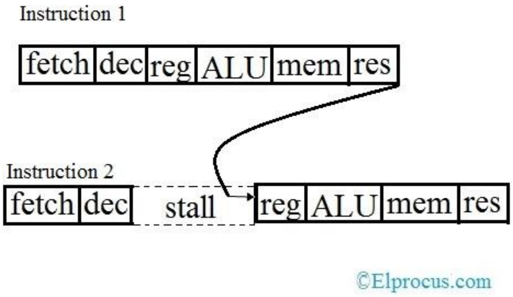
Solution for data hazards:

- Stalling
- Forwarding

  >> connect new value directly to the next stage.
- Reordering



Stall/Bubble in the Pipeline



Instruction Hazard

There are situations, called hazards, that prevent the next instruction in the instruction stream from being executing during its designated clock cycle. Hazards reduce the performance from the ideal speedup gained by pipelining. ... They arise from the pipelining of branches and other instructions that change the PC.

Control Hazards Solution

- Stall

  Stop loading instructions until result is available.

- Predict

  Assume an outcome and continue fetching (undo if prediction is wrong).

- Delayed branch

  Specify in architecture that following instructing is always executed.

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

https://www.youtube.com/watch?v=EL8uqzSsg_Q

https://www.youtube.com/watch?v=nC6csdXEkzU

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

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

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

Computer Architecture By bester field , page no: 461

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 - 25

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: INFLUENCE ON INSTRUCTION SETS

Introduction:

There is an influence of instructions on the pipelining. Some instructions are much better suited to pipelined execution than others. In this section we will discuss, how the features of the instruction affect the pipelined execution.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Pipelining and basic concepts
- Execution of a Complete Instruction
- Data Hazards, Instruction Hazards

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

Influence on Instruction

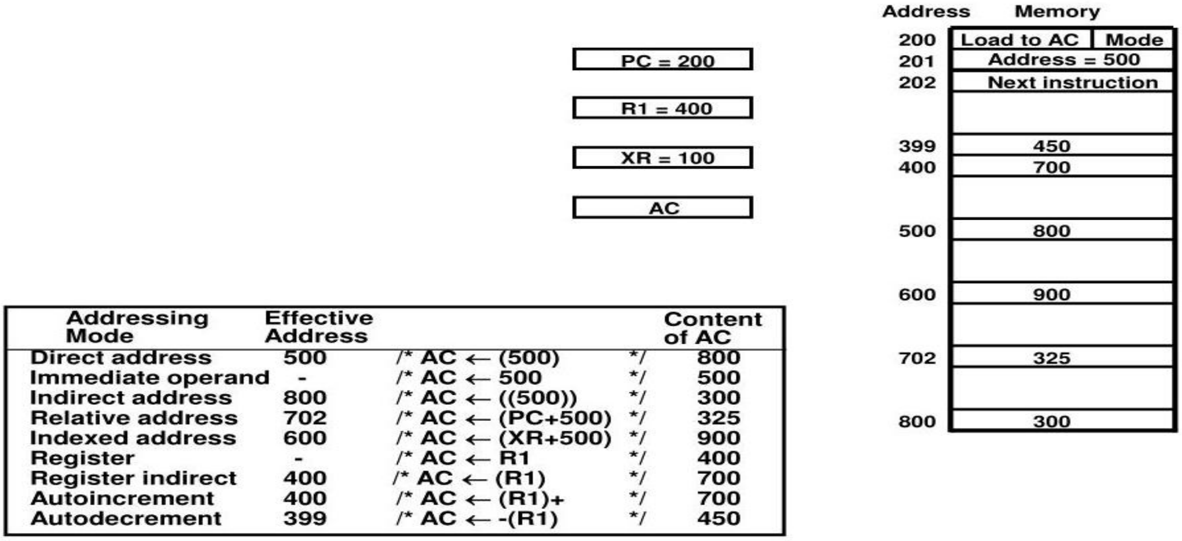
- **Addressing modes** Many processors provide various combinations of addressing modes such as index, indirect, auto increment, auto decrement and so on.
- We can classify these addressing modes as simple addressing modes and complex addressing modes.
- A complex address mode may require several accesses to the memory to reach the named operand while simple addressing mode requires only one access to the memory to reach the named operand.
- Consider the instruction Load R1, (X(R0)).
- This instruction has a complex addressing mode because this instruction requires two memory accesses one to read location X+[R0] and then to read location [X+[R0]].
- If R1 is a source operand in the next instruction that instruction has to be stalled for two cycles.
- If we want to perform the same operation using simple addressing mode instructions, we require to execute three instructions:

  ADD R1, R0, #X
  LOAD R1, (R2)
  LOAD R1, (R1)

  The ADD instruction performs the operation R1 [R0] +X.
- The two load instructions fetch the address and then the operand from the memory. The

two load instructions fetch the address and then the operand from the memory.

- This sequence of instructions takes exactly the same number of clock cycles as the single load instruction having a complex addressing mode.
- The above example indicates that in a pipeline processor, complex addressing mode do not necessarily lead to faster execution.
- But it reduces the number of instructions needed to perform particular task and hence reduce the program space needed in the main memory.



Addressing Mode	Effective Address		Content of AC
Direct address	500	/* AC ← (500) */	800
Immediate operand	-	/* AC ← 500 */	500
Indirect address	800	/* AC ← ((500)) */	300
Relative address	702	/* AC ← (PC+500) */	325
Indexed address	600	/* AC ← (XR+500) */	900
Register	-	/* AC ← R1 */	400
Register indirect	400	/* AC ← (R1) */	700
Autoincrement	400	/* AC ← (R1)+ */	700
Autodecrement	399	/* AC ← -(R1) */	450

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

https://www.youtube.com/watch?v=BrBZHteA3-0

https://www.youtube.com/watch?v=y7IR_aE8Sl0

https://www.youtube.com/watch?v=1KTW32xSs_k

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

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

Computer Architecture By bester field , page no: 465

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 26

CSE

II/III/A

Course Name with Code : 16CSD12& COMPUTER ARCHITECTURE

Course Faculty : R.Vinupriya

Unit : II - Arithmetic and Logic Unit Date of Lecture:

Topic of Lecture: DATAPATH AND CONTROL CONSIDERATIONS

Introduction:

When single bus is used in a processor only one data word can be transferred over the bus in a clock cycle. This increases the steps required to complete the execution of the instruction. To reduce the number of steps needed to execute instructions most commercial processors provide multiple internal paths that enable several transfers to take place in parallel.

Learning objective:

- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

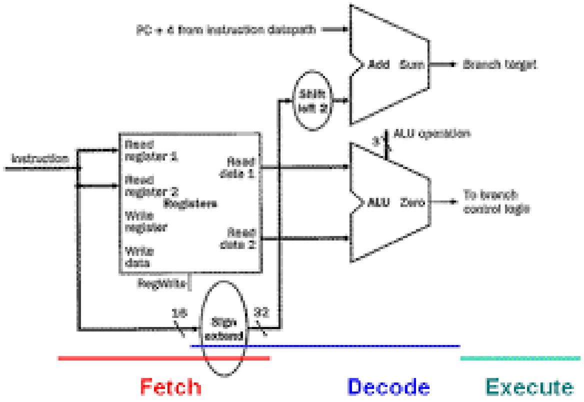
- Pipelining and basic concepts
- Datapath and control considerations
- Influence on Instruction Sets

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

Modified 3 bus structure of the processor for pipelined execution.

- 3 buses are used to connect registers and the ALU of the processor.
- All general-purpose registers are connected by a single block called register file.
- It has 3 ports:
  -- One input port
  -- Two output ports
- It is possible to access data of 3 register in one clock cycle, the value can be loaded in one register from bus C and data from two register can be accessed to bus A and bus B respectively.
- Buses A and B are used to transfer the source operands to the A and B inputs of the ALU.
- After performing arithmetic or logic operation result is transferred to the destination operand over bus C. To increment the contents of PC after execution of each instruction to fetch the next instruction separate unit is provided, it is known as incrementer.
- It is not required to pass this data through the ALU.
- Two buffer registers at the input and one at the output of the ALU are used.
- The instruction queue gets loaded from instruction cache. The output of the queue is connected to the instruction decoder input and output of the decoder is connected to the control signal pipeline.
- The processor can perform the following operations independently,

1. Reading an instruction from the instruction cache.
2. Incrementing the PC.
3. Decoding an instruction by instruction decoder.
4. Reading from and writing into the data cache.
5. Reading the contents of up to two registers from the register file.
6. Writing into one register in the register file.
7. Performing an ALU operation.



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

https://www.youtube.com/watch?v=0RMHamOq5FA

https://www.youtube.com/watch?v=94p9g0qhOSM

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

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

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

Computer Architecture By bester field , page no: 476

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 - 27

CSE

II/III/A

Course Name with Code : **16CSD12& COMPUTER ARCHITECTURE**

Course Faculty : **R.Vinupriya**

Unit : **II - Arithmetic and Logic Unit** **Date of Lecture:**

Topic of Lecture: SUPERSCALAR OPERATION

Introduction:

Several instructions can be executed concurrently because of pipelining. However, these instructions in the pipeline are in different stages of execution such as fetch, decode, ALU operation.

Learning objective:

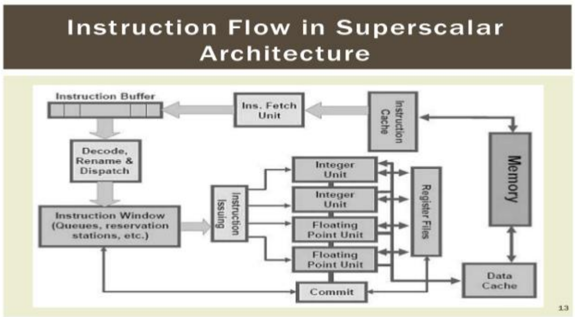
- To enhance the processor operation by employing pipelining

Prerequisite knowledge for Complete understanding and learning of Topic:

- Pipelining and basic concepts
- Execution of a Complete Instruction
- Influence on Instruction Sets

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

- Processors reach performance levels greater than one instruction per cycle by fetching, decoding and executing several instructions concurrently.
- This mode of operation is called superscalar.
- A processor has two execution units,
  - i) Integer unit (used for integer operations)
  - ii) Floating point unit (used for floating point operations)



Out-of-order Execution

The dispatch unit dispatches the instructions in the order in which they appear in the program. But their execution may be completed in the different order.

There are two causes of exceptions,

- a bus error (during an operand fetch)

- illegal operation (e.g. Divide by zero)

2 types of exceptions,

- Imprecise exceptions
- Precise exceptions

Imprecise exception Consider the pipeline timing, the result of operation of I2 is written into the register file in cycle 4. If instruction I1 causes an exception and succeeding instructions are permitted to complete execution, then the processor is said to have imprecise exceptions. Because of the exception by I1, program execution is in an inconsistent state.

Precise exception in the imprecise exception, consistent state is not guaranteed when an exception occurs. If the exception occurred then to guarantee a consistent state, the result of the execution of instructions must be written into the destination locations strictly in the program order.

Execution completion the decoding instructions are stored temporarily into the temporary registers and later they are transferred to the permanent registers in correct program order. Thus 2 write operations TW and W respectively are carried out.

Dispatch operation Each instruction requires the resources for its execution. The dispatch unit first checks the availability of the required resources and then only dispatches the instructions for execution. These resources include temporary register, a location in the order buffer, appropriate execution unit etc.

Deadlock Consider 2 units U1 and U2 are using shared resources. U2 needs completion of the task assign to unit U1 to complete its task. If unit U2 is using a resource which is also required to unit U1, both units cannot complete the tasks assigned to them. Both the units remain waiting for the need resource. Also, unit U2 is waiting for the completion of task by unit U1 before it can release that resource. Such a situation is called a deadlock.

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

https://www.youtube.com/watch?v=vMcMUIXDrz4

https://www.youtube.com/watch?v=ZUhJu84LMQo

https://www.youtube.com/watch?v=LWhQ6wnOceU

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

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

Computer Architecture By bester field , page no: 481

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

L - 28

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Basic Concepts

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)

- Memory locations.
- Word-addressable
- Addressing scheme

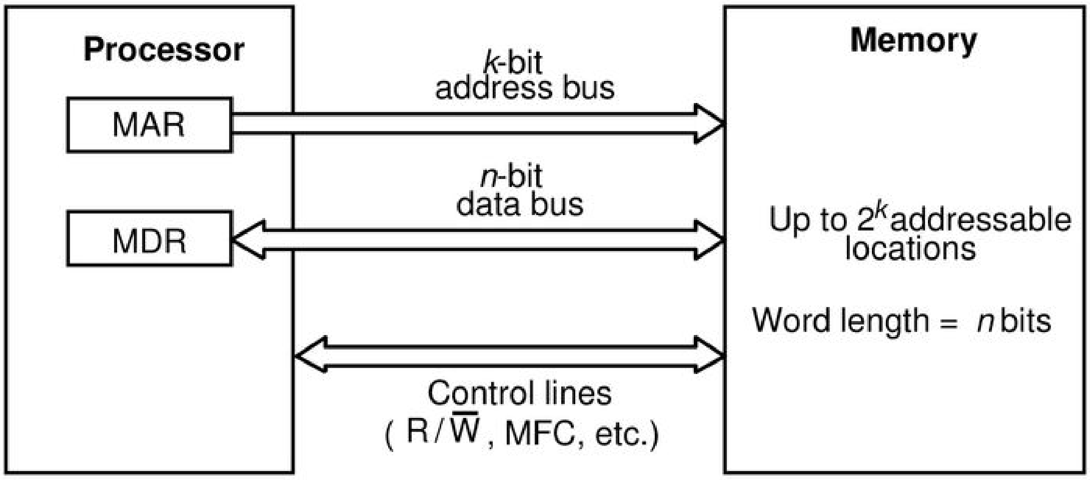
Detailed content of the Lecture:

- 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.

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.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,page no:207-210.

Course Faculty

Verified by HOD

L - 29

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Semiconductor RAM

Introduction : (Maximum 5 sentences) :

- Random access memory (RAM) is a type of primary storage.
- It allows the user to randomly access any part of the data regardless of its position in roughly the same time.
- This is not possible using other storage devices such as hard disks, CD's etc. because they have physical constraints such rotation speeds, arm movements etc.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Storage devices
- Memory
- Primary storage
- Secondary storage

Detailed content of the Lecture:

- 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

Some advantages of DRAM are:

- 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

Some disadvantages of DRAM are:

- 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=I9aqXyOZj8E
https://www.youtube.com/watch?v=c3V8w2Wk-D0

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

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002, page-no:210-221.

Course Faculty

Verified by HOD

L - 31

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Speed, Size and Cost

Introduction : (Maximum 5 sentences) :

- Memories store data and allow processors to read and to write the data.
- Given a memory address and some data, a memory write copies the data to that memory address, given a memory address, a memory read returns a copy of the data most recently written to that location

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

- Speed
- Size
- Cost

Detailed content of the Lecture:

Costs:

Typically costs are expressed as a ratio of dollars per byte.

- Costs can be accounted for in different ways, such as purchase price, total costs of ownership, and in between measures.
- For example, an enterprise that uses disks to store 1 Terabyte of data pays more than just the 20 disks needed to store the data.
- The disks need to be placed in a storage cabinet, and connected to the storage server. In addition to this hardware, there are also the costs of administration and maintenance.

Capacity:

- The amount of data that can be stored per unit.

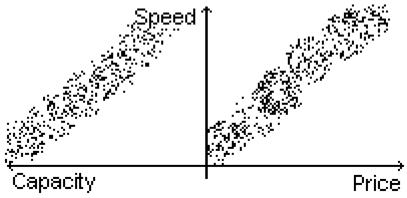
Speed:

- The time it takes to access data. When we assess the speed of devices, we need to distinguish between the raw speed of the device (e.g. 50 nsec for RAM) and the speed of data access using the technology embedded in the system (e.g. 200 nsec for accessing main memory because of paging support).
- A device might react differently to request depending on previous operations.
- For example, streaming from a disk drive yields higher data rate (and hence higher access speed per byte) than true random accesses.

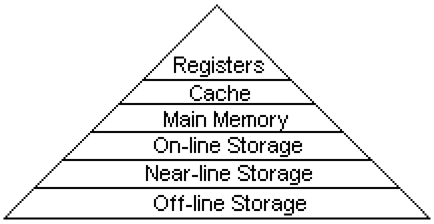
- Reading from a RAM bank repeatedly is slower than reading from different RAM banks, etc.

Reliability:

- Increasingly important, reliability in the strict sense measures the time from initialization to the first/next failure event.
- It is measured in mean time to failure (MTTF). If a module can be repaired, we can calculate or statistically determine the mean time to repair (MTTR).
- Availability in the strict sense is the proportion of time that a device is up and running, or - in an alternative interpretation - the probability of finding a device up and running.
- Availability equals MTTF / (MTTF + MTTR).
- Other characteristics such as **volatility** (does the device retain data after powering down), or **transportability**.
- If we plot the size, speed, and costs of memory and storage devices, we obtain the following, schematic picture:



- Faster devices cost more and have lower capacity. Currently the largest capacity storage devices are tapes, which still cost marginally less than hard drives



- We can arrange the different types of memory in a pyramid. At the top is the fastest, smallest, and most expensive memory, the registers.
- At the bottom, the largest, cheapest, and slowest memory, off-line archival storage (e.g. a tape library).

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

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

https://www.frontiersin.org/articles/10.3389/fnhum.2017.00615/full

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

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,page-no:223-224.

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

DESIGNING YOUR FUTURE
Estd. 2000

IQAC

LECTURE HANDOUTS

L - 32

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Cache Memories

Introduction : (Maximum 5 sentences) :

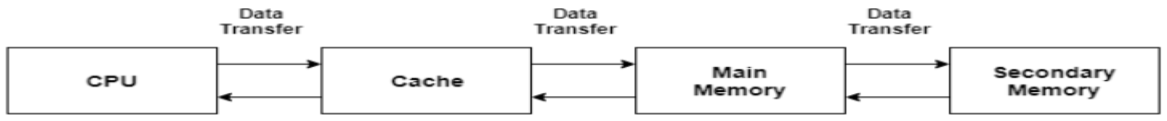
- It acts as a buffer between the CPU and the main memory.
- It is used to hold those parts of data and program which are most frequently used by the CPU.

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

- Memory
- Storage
- Address

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:

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

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.

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

Important Books/Journals for further learning including the page nos.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,page-no: 224-238.

Course Faculty

Verified by HOD

L - 33

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Performance Considerations

Introduction : (Maximum 5 sentences) :

- 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)

- Performance
- Machine instructions

Detailed content of the Lecture:

- 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.

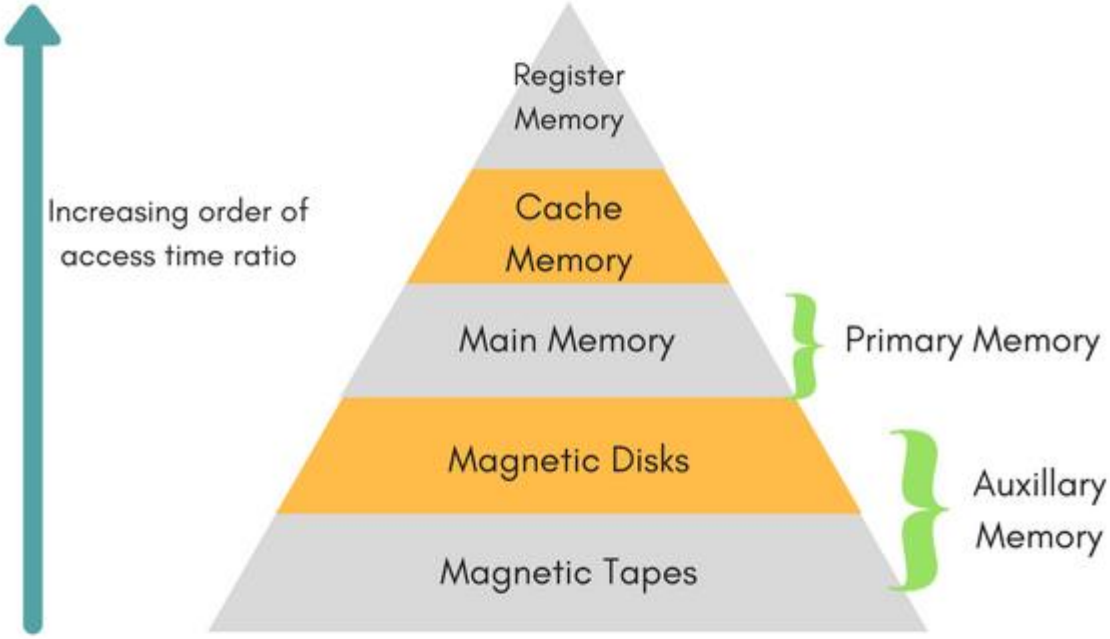
```
Performance = (1 / Execution time)
```

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=iL01Qr0Tspc
https://www.youtube.com/watch?v=TyYeCpdJCb8

Important Books/Journals for further learning including the page nos.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002, page no: 238.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 34

CSE

II/III/A

Course Name with Code	**: 16CSD12& Computer Architecture**
Course Teacher	**: R.Vinupriya**
Unit	**: IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Virtual Memory

Introduction : (Maximum 5 sentences) :

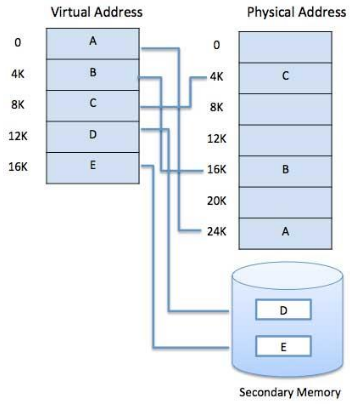
- Virtual memory is a valuable concept in computer architecture that allows you to run large, sophisticated programs on a computer even if it has a relatively small amount of RAM.
- A computer with virtual memory artfully juggles the conflicting demands of multiple programs within a fixed amount of physical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

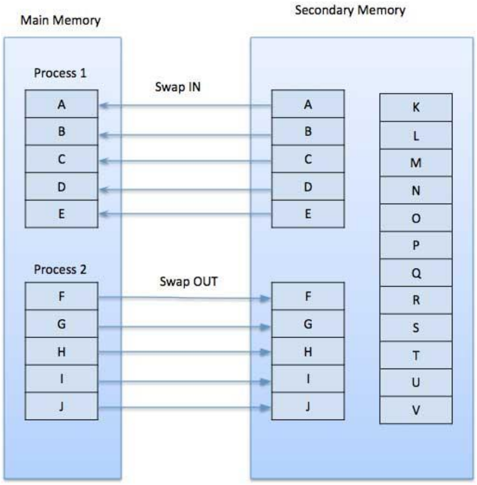
- A computer can address more memory than the amount physically installed on the system.
- This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.
- The main visible advantage of this scheme is that programs can be larger than 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.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,page no:245-250.

Course Faculty

Verified by HOD

L - 30

LECTURE HANDOUTS

CSE

II/IV/A

Course Name with Code : 16CSD12& Computer Architecture

Course Teacher : R.Vinupriya

Unit : IV - MEMORY SYSTEM Date of Lecture:

Topic of Lecture: ROM

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.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,Page no:245-250.

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 - 35

LECTURE HANDOUTS

CSE **II/III/A**

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Memory Management Requirements

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.

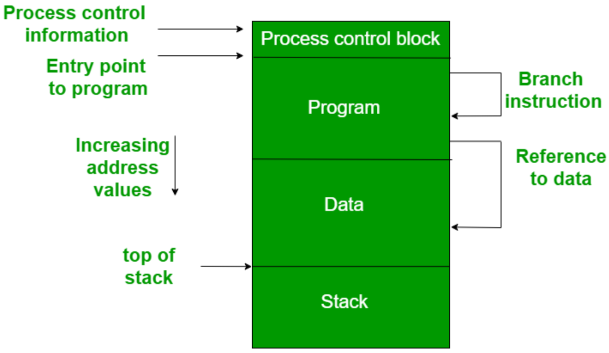
Prerequisite knowledge for Complete understanding and learning of Topic: (Max. Four important topics)

- 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.:
Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002, 251-252.

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

L - 36

LECTURE HANDOUTS

CSE **II/III/A**

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **IV - MEMORY SYSTEM** **Date of Lecture:**

Topic of Lecture: Secondary Storages

Introduction : (Maximum 5 sentences) :

- A secondary storage device refers to any non-volatile storage device that is internal or external to the computer.
- It can be any storage device beyond the primary storage that enables permanent data storage.
- A secondary storage device is also known as an auxiliary storage device or external storage.

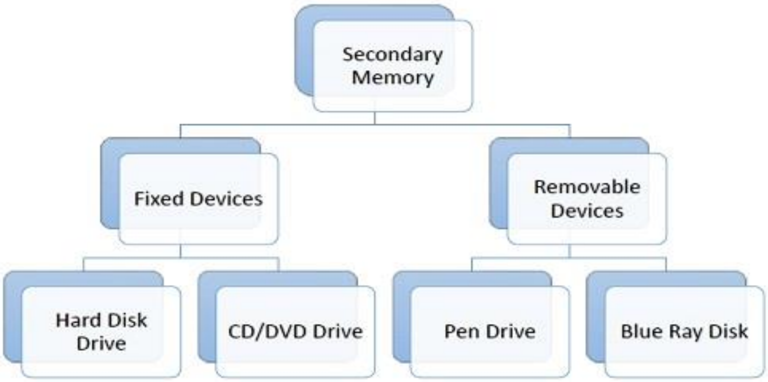
Prerequisite knowledge for Complete understanding and learning of Topic:
(Max. Four important topics)

- Non-volatile
- Volatile

Detailed content of the Lecture:

Characteristics of Secondary Memory

- It is non-volatile, i.e. it retains data when power is switched off
- It is large capacities to the tune of terabytes
- It is cheaper as compared to primary memory
- Depending on whether secondary memory device is part of CPU or not, there are two types of secondary memory – fixed and removable.



Hard Disk Drive

- Hard disk drive is made up of a series of circular disks called platters arranged one over the other almost ½ inches apart around a spindle. Disks are made of non-magnetic material like aluminum alloy and coated with 10-20 nm of magnetic material.

CD Drive

- CD stands for Compact Disk. CDs are circular disks that use optical rays, usually lasers, to read and write data. They are very cheap as you can get 700 MB of storage space for less than a dollar. CDs are inserted in CD drives built into CPU cabinet. They are portable as you can eject the drive, remove the CD and carry it with you. There are three types of CDs −

- CD-ROM (Compact Disk – Read Only Memory) − The data on these CDs are recorded by the manufacturer. Proprietary Software, audio or video are released on CD-ROMs.
- CD-R (Compact Disk – Recordable) − Data can be written by the user once on the CD-R. It cannot be deleted or modified later.
- CD-RW (Compact Disk – Rewritable) − Data can be written and deleted on these optical disks again and again.

DVD Drive

- DVD stands for Digital Video Display. DVD are optical devices that can store 15 times the data held by CDs.
- They are usually used to store rich multimedia files that need high storage capacity. DVDs also come in three varieties – read only, recordable and rewritable.

Pen Drive

- Pen drive is a portable memory device that uses solid state memory rather than magnetic fields or lasers to record data.
- It uses a technology similar to RAM, except that it is nonvolatile. It is also called USB drive, key drive or flash memory.

USB Port

- USB stands for Universal Serial Bus. It is the industry standard for short distance digital data connection. USB port is a standardized port to connect a variety of devices like printer, camera, keyboard, speaker, etc.

Floppy Disk

- A floppy disk is a flexible disk with a magnetic coating on it. It is packaged inside a protective plastic envelope.
- These are one of the oldest types of portable storage devices that could store up to 1.44 MB of data but now they are not used due to very less memory storage.

Hard disk

- A hard disk consists of one or more circular disks called platters which are mounted on a common spindle. Each surface of a platter is coated with a magnetic material.
- Both surfaces of each disk are capable of storing data except the top and bottom disk where only the inner surface is used.
- The information is recorded on the surface of the rotating disk by magnetic read/write heads. These heads are joined to a common arm known as access arm.

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

https://www.youtube.com/watch?v=_iMxT9VmF2k
https://www.youtube.com/watch?v=CdNGX9kM-is

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

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002, page no:382.

Course Faculty

Verified by HOD

Estd. 2000

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

L - 37

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: Input and output devices

Introduction : (Maximum 5 sentences) :

- An input device sends information to a computer system for processing, and an output device reproduces or displays the results of that processing.
- Input devices only allow for input of data to a computer and output devices only receive the output of data from another device.

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

- Input devices
- Output devices

Detailed content of the Lecture:

Following are few of the important input devices which are used in a computer:

- Keyboard ,Mouse ,Joy Stick ,Light pen
- Track Ball ,Scanner Graphic ,Tablet ,Microphone

Keyboard :

- Keyboard is the most common and very popular input device which helps in inputting data to the computer.
- Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

Mouse:

- Mouse is most popular pointing device.
- It is a very famous cursor-control device having a small palm size box with a round ball at its base which senses the movement of mouse and sends corresponding signals to CPU when the mouse buttons are pressed.

Advantages: Easy to use Not very expensive

Joystick :

- Joystick is also a pointing device which is used to move cursor position on a monitor screen.

Light Pen :

- Light pen is a pointing device which is similar to a pen.

Scanner: Scanner is an input device which works more like a photocopy machine.

- It is used when some information is available on a paper and it is to be transferred to the hard disc of the computer for further manipulation.

Following are some of the important output devices used in a computer.

- Monitors
- Graphic Plotter
- Printer

Monitors

- Monitors, commonly called as Visual Display Unit (VDU), are the main output device of a computer.
- There are two kinds of viewing screen used for monitors.
  - Cathode-Ray Tube (CRT)
  - Flat-Panel Display

Printers

- Printer is an output device, which is used to print information on paper.
- There are two types of printers −
- **Impact Printers:** Impact printers print the characters by striking them on the ribbon, which is then pressed on the paper.
- **Non-Impact Printers**: Non-impact printers print the characters without using the ribbon.
- These printers are of two types −Laser Printers & Inkjet Printers

Character Printers

Character printers are the printers which print one character at a time.

- Dot Matrix Printer(DMP)
- Daisy Wheel

Advantages

- Very high speed
- Very high quality output
- Good graphics quality
- Supports many fonts and different character size

Disadvantages

- Expensive
- Cannot be used to produce multiple copies of a document in a single printing

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

https://www.youtube.com/watch?v=CTNtf-oGLgY

https://www.youtube.com/watch?v=Gg8yOzP8ENY

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

Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization, Fifth Edition, McGraw-Hill, 2002,page no:370-382.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 39

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

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

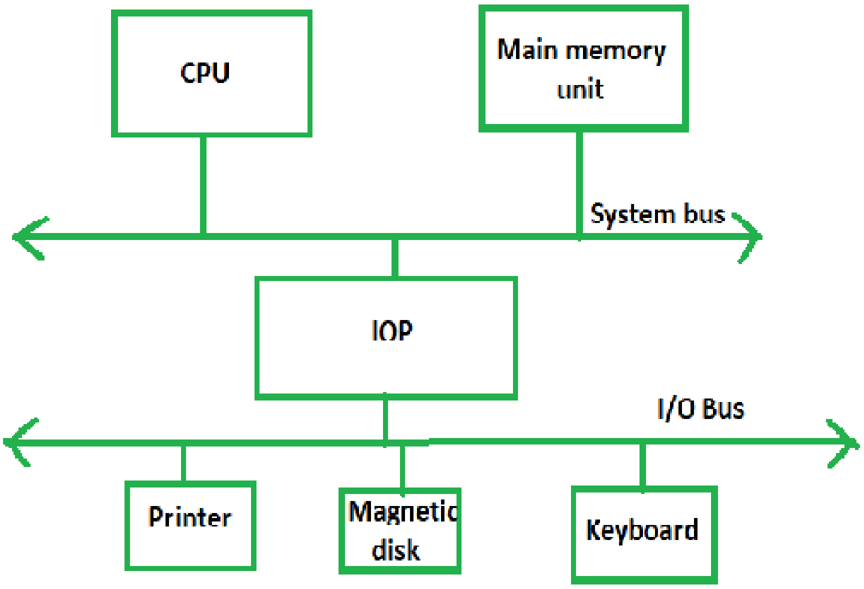
Move DATAIN, R0

- 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, 2002, page no: 152.

Course Faculty

Verified by HOD

LECTURE HANDOUTS

L - 38

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: Serial communication links

Introduction : (Maximum 5 sentences) :

- In computing, input/output, or I/O, refers to the communication between an information processing system (computer), and the outside world.
- Inputs are the signals or data received by the system, and outputs are the signals or data sent from it.
- I/O devices are used by a person (or other system) to communicate with a computer.

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

- Input signal
- Interrupt signal

Detailed content of the Lecture:

- Memory-mapped I/O: The arrangement of I/O devices and the memory share the same address space 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 reads the data from DATAIN and stores them into processor register RO.
- Similarly, the instruction 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.
- 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.
- I/O devices operate at speeds that are vastly different from that of the processor.
- When a human operator is entering characters at a keyboard, the processor is capable of executing millions of instructions between successive character entries.
- An instruction that reads a character from the keyboard should be executed only when a character is available in the input buffer of the keyboard interface. An input character is read only once.
- For an input device such as a keyboard, a status flag, SIN, is included in the interface circuit as part of the status register.

- This flag is set to 1 when a character is entered at the keyboard and cleared to 0 once this character is read by the processor.
- Hence, by checking the SIN flag, the software can ensure that it is always reading valid data.
- This is often accomplished in a program loop that repeatedly reads the status register and checks the state of SIN. When SIN becomes equal to 1, the program reads the input data register.
- A similar procedure can be used to control output operations using an output status flag, SOUT.
- Input–output (I/O) devices vary substantially in their characteristics.
- One distinguishing factor among input devices (and also among output devices) is their data processing rate, defined as the average number of characters that can be processed by a device per second.
- For example, while the data processing rate of an input device such as the keyboard is about 10 characters (bytes)/second, a scanner send data at a rate of about 200,000 characters/second.

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, 2002, page no: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

L - 40

LECTURE HANDOUTS

CSE **II/III/A**

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: 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)*.

When a device raises an interrupt at let's say process i, the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR. Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt the processor can continue with process i+1.

While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called Interrupt Latency.

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 interrupt, the value of INTR is the logical OR of the requests from individual devices.

The sequence of events involved in handling an IRQ:

1. Devices raise an IRQ.

2. The processor interrupts the program currently being executed.
3. The device is informed that its request has been recognized and the device deactivates the request signal.
4. The requested action is performed.
5. An 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 device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting. These are explained as following below.

Polling:

In polling, the first device encountered with the IRQ bit set is the device that is to be serviced first. Appropriate ISR is called to service the same. It is easy to implement but a lot of time is wasted by interrogating the IRQ bit of all devices.

Vectored Interrupts:

In vectored interrupts, a device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt. The special code can be the starting address of the ISR or where the ISR is located in memory and is called the interrupt vector.

Interrupt Nesting:

In this method, the I/O device is organized in a priority structure. Therefore, an interrupt request from a higher priority device is recognized whereas a request from a lower priority device is not. To implement this each process/device (even the processor). The processor accepts interrupts only from devices/processes having priority more than it.

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, 2002, page no: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

L - 42

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: Direct Memory Access

Introduction : (Maximum 5 sentences) :

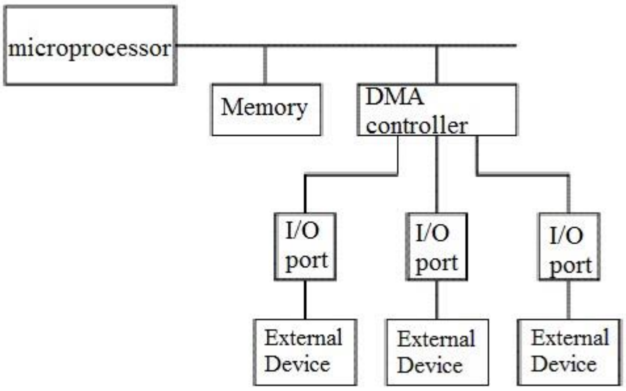
- Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations.
- The process is managed by a chip known as a DMA controller (DMAC)

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

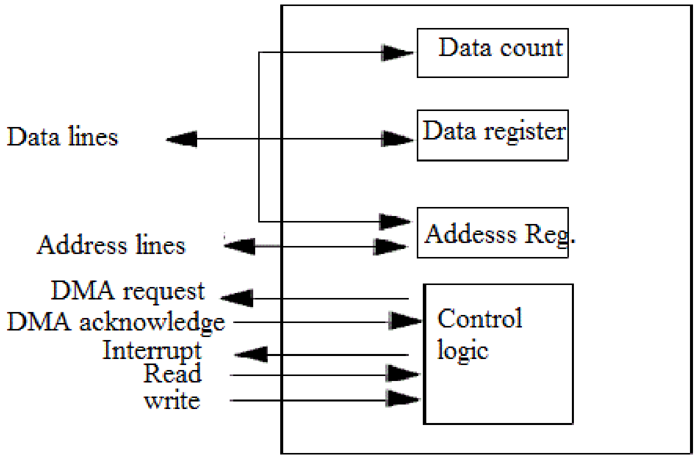
- Direct Memory Access
- Memory Access

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, 2002, page no:175.

Course Faculty

Verified by HOD

L - 41

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : 16CSD12& Computer Architecture

Course Teacher : R.Vinupriya

Unit : V - Peripherals and I/O Organization

Date of Lecture:

Topic of Lecture: Direct Memory Access

Introduction : (Maximum 5 sentences) :

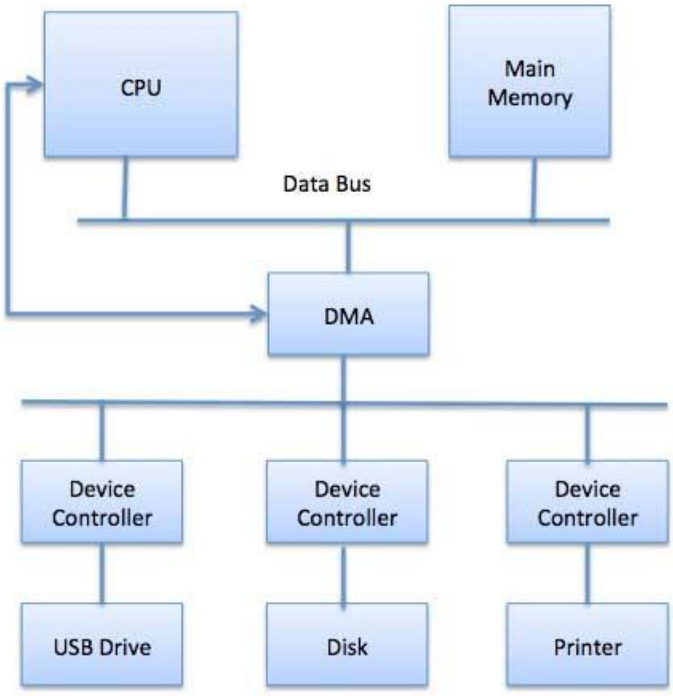
- Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main memory, bypassing the CPU to speed up memory operations.
- The process is managed by a chip known as a DMA controller (DMAC)

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

- Direct Memory Access
- Memory Access

Detailed content of the Lecture:

- Slow devices like keyboards will generate an interrupt to the main CPU after each byte is transferred.
- If a fast device such as a disk generated an interrupt for each byte, the operating system would spend most of its time handling these interrupts.
- So a typical computer uses direct memory access (DMA) hardware to reduce this overhead.
- Direct Memory Access (DMA) means CPU grants I/O module authority to read from or write to memory without involvement.
- DMA module itself controls exchange of data between main memory and the I/O device.
- CPU is only involved at the beginning and end of the transfer and interrupted only after entire block has been transferred.
- Direct Memory Access needs a special hardware called DMA controller (DMAC) that manages the data transfers and arbitrates access to the system bus.
- The controllers are programmed with source and destination pointers (where to read/write the data), counters to track the number of transferred bytes, and settings, which includes I/O and memory types, interrupts and states for the CPU cycles.



The operating system uses the DMA hardware as follows −

Step	Description
1	Device driver is instructed to transfer disk data to a buffer address X.
2	Device driver then instruct disk controller to transfer data to buffer.
3	Disk controller starts DMA transfer.
4	Disk controller sends each byte to DMA controller.
5	DMA controller transfers bytes to buffer, increases the memory address, decreases the counter C until C becomes zero.
6	When C becomes zero, DMA interrupts CPU to signal transfer completion.

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, 2002,page no: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

DESIGNING YOUR FUTURE
Estd. 2000

IQAC

L - 43

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: Buses

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.
- 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.

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, 2002, page no:180

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 - 44

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: Standard I/O Interfaces (PCI, SCSI, 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

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, 2002, page no:194.

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

L - 45

LECTURE HANDOUTS

CSE

II/III/A

Course Name with Code : **16CSD12& Computer Architecture**

Course Teacher : **R.Vinupriya**

Unit : **V - Peripherals and I/O Organization**

Date of Lecture:

Topic of Lecture: SCSI (Small Computer System Interface)& USB (Universal Serial Bus)

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)

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.

USB:

- Short for universal serial bus, USB (pronounced yoo-es-bee) is a plug and play interface that allows a computer to communicate with peripheral and other devices. USB-connected devices cover a broad range; anything from keyboards and mice, to music players and flash drives. For more information on these devices, see our USB devices section.
- USB may also be used to send power to certain devices, such as powering smartphones and tablets and charging their batteries. The first commercial release of the Universal Serial Bus (version 1.0) was in January 1996. This industry standard was then quickly adopted by Intel, Compaq, Microsoft, and other companies.

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, 2002, page no:194.

Course Faculty

Verified by HOD