BACHELOR OF COMPUTER APPLICATION (BCA)

SECOND SEMESTER

MICROPROCESSOR AND COMPUTER ARCHITECTURE

Dear Students, Don't limit your knowledge horizon, it's only a reference, and you can use other resources for more knowledge.

Unit 1: Fundamental of Microprocessor (5 Hrs.)

Introduction to Microprocessors, Microprocessor system with bus organization, Microprocessor architecture and operation, 8085 Microprocessor and its operation, 8085 instruction cycle, machine cycle, T states, Addressing modes in 8085, Introduction to 8086.

[Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 5th Edition,

Reference Manual Books:

1 Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 5th Edition
2 M. Morris Mano, Computer System Architecture, 3rd Edition
3 William Stallings, Computer Organization and Architecture, 8th Edition (Optional)
4 William Stallings, Computer Organization and Architecture, 9th Edition

Questions:

1. Define Microprocessor. Differentiate between Microprocessor and Microcontroller with example.

Ans:- ***A microprocessor*** is a multipurpose, programmable, clock-driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to these instruction, and provides result as output.

In other words, a **microprocessor** is the chip containing some control and logic circuits that is capable of making arithmetic and logical decisions based on input data and produces the corresponding arithmetic or logical output.

Microprocessor		***Microcontroller***	
1.	It is the Heart of computer system	1.	It is the Heart of embedded system
2	It is just processor, Memory and I/O components have to be connected externally.	2.	It has a external processor, along with Memory and I/O components.
3	Since memory and I/O has to be connected externally the circuit become large.	3.	Since memory and I/O are present internally, the small is circuit .
4	Cannot be used in compact systems and hence inefficient	4.	Can be used in compact systems and hence it is inefficient technique.
5	It have less number of registers, hence more operation are memory based.	5.	It have more number of registers, hence the programs are easier to write.
6	Mainly used in person computer.	6.	Used Mainly used in washing machine. MP_3

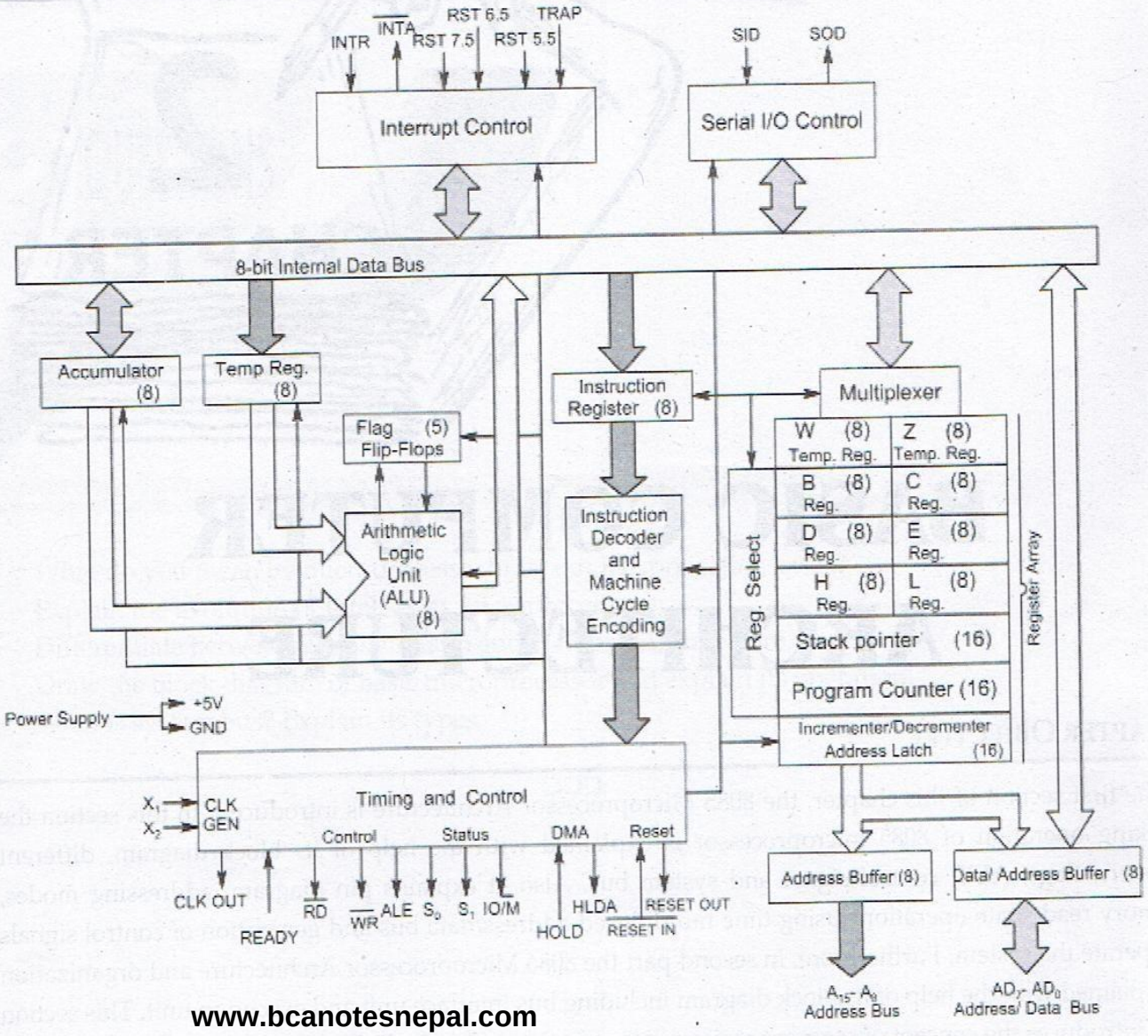
2. Explain microprocessor as a CPU.

1. Explain the organization of microprocessor based system with block diagram.

www.bcanotesnepal.com

2. Explain the microprocessor architecture and its operations. OR Explain the 8085 microprocessor with its functional diagram. ***(Page no 16 and 17)***

The Intel 8085 A is a complete 8 bit parallel central processing unit. The main components of 8085A are array of registers, the arithmetic logic unit, the encoder/decoder, and timing and control circuits linked by an internal data bus. The block diagram is shown below:



www.bcanotesnepal.com

Figure: The 8085 A microprocessor Functional Block Diagram.

1: **ALU:** The arithmetic logic unit performs the computing functions, it includes the accumulator, the temporary register, the arithmetic and logic circuits and five flags. The temporary register is used to hold data during an arithmetic/logic operation. The result is stored in the accumulator; the flags (flip-flops) are set or reset according to the result of the operation.

2. **Accumulator (register A):** It is an 8 bit register that is the part of ALU. This register is used to store the 8-bit data and to perform arithmetic and logic operations and 8085 microprocessor is called accumulator based microprocessor. When data is read from input port, it is first moved to accumulator and when data is sent to output port, it must be first placed in accumulator.

3. **Temporary registers (W & Z):** They are 8 bit registers not accessible to the programmer. During program execution, 8085A places the data into it for a brief period.

4. **Instruction register (IR):** It is a 8 bit register not accessible to the programmer. It receives the operation codes of instruction from internal data bus and passes to the instruction decoder which decodes so that microprocessor knows which type of operation is to be performed.

5. **Register Array:** (Scratch pad registers B, C, D, E): It is a 8 bit register accessible to the programmers. Data can be stored upon it during program execution. These can be used individually as 8-bit registers or in pair BC, DE as 16 bit registers. The data can be directly added or transferred from one to another. Their contents may be incremented or decremented and combined logically with the content of the accumulator.

   - **Register H & L**: They are 8 bit registers that can be used in same manner as scratch pad registers.

www.bcanotesnepal.com

   - **Stack Pointer (SP):** It is a 16 bit register used as a memory pointer. It points to a memory location in R/W memory, called the stack. The beginning of the stack is defined by loading a 16- bit address in the stack pointer.
   - **Program Counter (PC):** Microprocessor uses the PC register to sequence the execution of the instructions. The function of PC is to point to the memory address from which the next byte is to be fetched. When a byte is being fetched, the PC is incremented by one to point to the next memory location.

6. **Flag register:** Register consists of five flip-flops, each holding the status of different states separately is known as flag register and each flip-flop are called flags. 8085A can set or reset one or more of the flags and are sign(S), Zero (Z), Auxiliary Carry (AC) and Parity (P) and Carry (CY). The state of flags indicates the result of arithmetic and logical operations, which in turn can be used for decision making processes. The different flags are described as:

   - **Carry (CY)**: If the last operation generates a carry its status will 1 otherwise 0. It can handle the carry or borrow from one word to another.
   - **Zero (Z):** If the result of last operation is zero, its status will be 1 otherwise 0. It is often used in loop control and in searching for particular data value.
   - **Sign (S)**: If the most significant bit (MSB) of the result of the last operation is 1 (negative), then its status will be 1 otherwise 0.
   - **Parity (P)**: If the result of the last operation has even number of 1's (even parity), its status will be 1 otherwise 0.

www.bcanotesnepal.com

   - **Auxiliary carry (AC)**: If the last operation generates a carry from the lower half word (lower nibble), its status will be 1 otherwise 0. Used for performing BCD arithmetic.

   Its bit position is shown in the following diagram:

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	-	AC	-	P	-	CY

7. **Timing and Control Unit:** This unit synchronizes all the microprocessor operations with the clock and generates the control signals necessary for communication between the microprocessor and peripherals. The control signals are similar to the sync pulse in an oscilloscope. The $\overline{RD}$ and $\overline{WR}$ signals are sync pulses indicating the availability of data on the data bus.
8. **Interrupt controls:** The various interrupt controls signals (INTR, RST 5. 5, RST 6. 5, RST 7. 5 and TRAP) are used to interrupt a microprocessor.
9. **Serial I/O controls:** Two serial I/O control signals (SID and SOD) are used to implement the serial data transmission.

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

3. Explain the bus architecture of 8085 microprocessor.(Page no 11 & 12)

Bus is a group of conducting wires which carries information. All the peripherals are connected to microprocessor through bus. Diagram given below represents the bus organization system of 8085 Microprocessor.

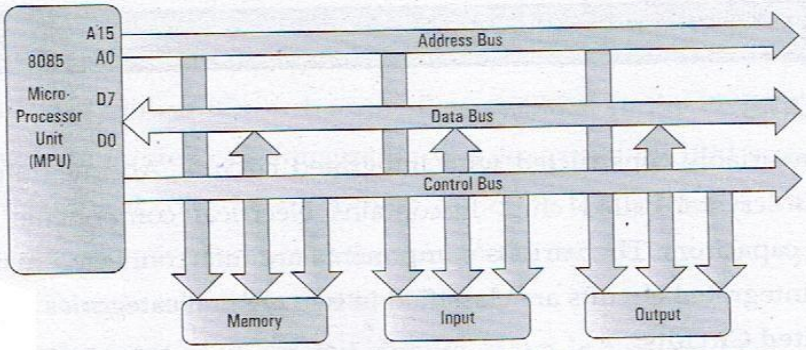


Figure : Bus organization of 8085 Microprocessor

There are three types of buses.

www.bcanotesnepal.com

Address Bus

It is a group of conducting wires which carries address only. Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor).

Length of Address Bus of 8085 microprocessor is 16 Bit (That is, Four Hexadecimal Digits), ranging from 0000 H to FFFF H, (H denotes Hexadecimal). The microprocessor 8085 can transfer maximum 16 bit address which means it can address 65, 536 different memory location.

The Length of the address bus determines the amount of memory a system can address. Such as a system with a 32-bit address bus can address 2^{32} memory locations. If each memory location holds one byte, the addressable memory space is 4 GB. However, the actual amount of memory that can be accessed is usually much less than this theoretical limit due to chipset and motherboard limitations.

Data Bus

It is a group of conducting wires which carries data only. Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/Output devices and from memory or Input/Output devices to microprocessor.

www.bcanotesnepal.com

Length of Data Bus of 8085 microprocessor is 8 Bit (That is, two Hexadecimal Digits), ranging from 00 H to FF H. (H denotes Hexadecimal).

When it is write operation, the processor will put the data (to be written) on the data bus, when it is read operation, the memory controller will get the data from specific memory block and put it into the data bus.

The width of the data bus is directly related to the largest number that the bus can carry, such as an 8 bit bus can represent 2^8 unique values, this equates to the number 0 to 255. A 16 bit bus can carry 0 to 65535.

Control Bus

It is a group of conducting wires, which is used to generate timing and control signals to control all the associated peripherals. Microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:

- Memory read
- I/O read
- Opcode fetch
- Memory write
- I/O Write

4. Explain the 8085 microprocessor signals with block diagram.

www.bcanotesnepal.com

5. Explain the opcode fetch and memory read machine cycles for MVI A, 48H with timing for execution diagram.

www.bcanotesnepal.com

6. Define Computer Architecture? Different between Computer architecture and Computer Organization.

Ans: - ***Computer architecture*** is a design principle that describes the functionality organization and implementation of computer system.

Computer Architecture	Computer organization
i) It abstract model and are those attributes that are visible to programmer like instructions sets, no of bits used for data, addressing techniques. ii) For eg:- a company that manufactures cars, design and all low-level details of the car come under computer architecture (abstract, programmers view)	i) It expresses the realization of the architecture or how features are implemented like these registers ,those data paths or this connection to memory. contents of CO are ALU, CPU and memory and memory organizations. ii) while making it's parts piece by piece and connecting together the different components of that car by keeping the basic design in mind comes under computer organization (physical and visible).

www.bcanotesnepal.com

www.bcanotesnepal.com

7. Explain the 8085 microprocessor addressing modes with example. ***(page no 22,23,24)***

Ans :- The different ways in which a processor can access data are referred as its addressing mode. In assembly language statements , The addressing mode is indicated in the instruction itself.

The various addressing mode are :-…….

i) Register Addressing Mode
ii) Immediate Addressing Mode.
iii) Direct Addressing Mode.
iv. Register Indirect Addressing Mode
V. Implied Addressing Mode

1. Register Addressing Mode

- ⌘ It is the most common form of data addressing.
- ⌘ Transfers a copy of a byte/word from source register to destination register.

Instruction	Source	Destination
MOV A,B	Register B	Register A

- ⌘ It is carried out with 8 bit registers A,B,C,D,E,H & L
- ⌘ It is important to use registers of same size.
- ⌘ Never mix an 8 bit register with a 16 bit register i. e. MOV A,SP

EXAMPLES : MOV A, B : Copys B into A
MOV SP, H : Copys H pair into SP

2. Immediate Addressing Mode

- ⌘ The term immediate implies that the data immediately follow the hexadecimal opcode in the memory.

Note *that immediate data are constant data.*

- ⌘ It transfers the source immediate byte/word of data in destination register or memory location.

Instruction	Source	Destination
MVI C,3AH	Data 3AH	Register C

EXAMPLES MOV A,90 : Copys 90 into A
LXI H,1234H : Copys 1234H into H

3. Direct Addressing Mode

- ⌘ In this scheme, the address of the data is defined in the instruction itself.

Instruction	Source	Destination
LDA 2000H	Memory Location 2000H	Register A

EXAMPLES LHLD 1000H : Copies the content of 1000H address memory to L and 10001H memory to H.
LDA 2000H : Copies the content of 2000H memory to accumulator
JMP 4000H
Call 5000H

www.bcanotesnepal.com

4. Register Indirect Addressing Mode

- ⌘ Register indirect addressing allows data to be addressed at any memory location through an address held in any of the H pair, B pair and D pair registers.
- ⌘ It transfers byte/word between a register and a memory location addressed.

Instruction	Source	Destination
MOV C,M	Assume HL = 1000H and M is the content of 1000 H address	Register C

EXAMPLES MOV C, M : Copies the word contents of the memory location addressed by HL pair into C
STAX B : Copies A into the memory location addressed by B pair
JMP 4000H
Call 5000H

5. Implied Addressing Mode

- ⌘ The addressing mode of certain instructions is implied by the instruction's function.

Instruction	Source	Destination
STC		Carry Flag

EXAMPLES STC : Set carry flag
CMC : Complementary carry flag
DAA : Decimal adjust accumulator content

8. Define Interrupt. Explain the types of Interrupt.

Ans:- Interrupt are the signal generated by the external device to request the microprocessor to perform a task. There are 5 ***types of interrupt signal*** i.e. TRAP,RST 7.5,RST6.5, RST5.5 and INTR. Interrupt are classified into following group based on their parameter.

(i) ***vector interrupt::-*** In this type of interrupt the interrupt address is known to processor. Eg. RST7.5, RST6.5, RST5.5 & TRAP.

(ii) ***Non-Vector interrupt:*** In this type of interrupt the interrupt address is not known to processor. Eg. INRT

(iii) ***Maskable Interrupt:-*** In this types of interrupt we can disable the interrupt by writing some instruction into the program Eg. RST 7.5,RST6.5, RST5.5

(iv) ***Non-Maskable Interrupt:-*** In this types of interrupt we can not be disable the interrupt by writing some instruction into the program Eg. TRAP.

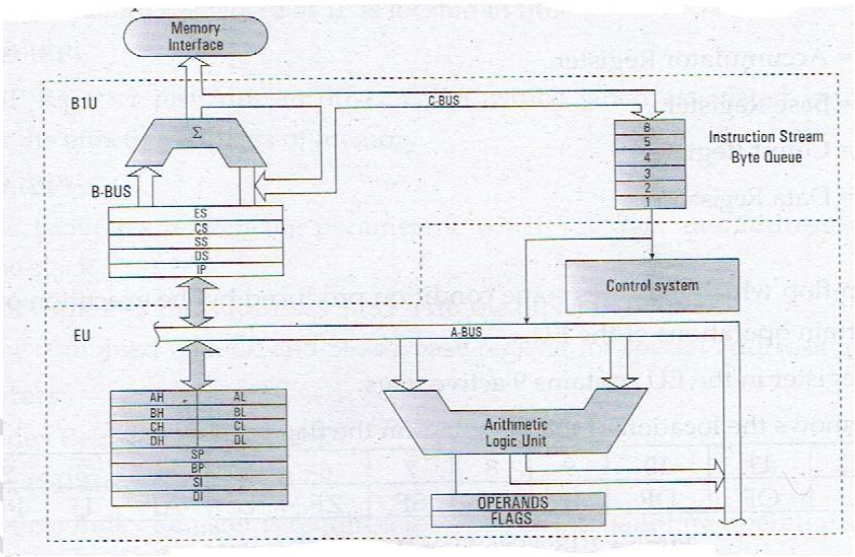
(v) ***Software interrupt:*** In this type of interrupt the programmable has to add the instruction into the program to executed the interrupt. There are 8 software interrupt in 8085,i.e. RST0,RST1, RST2, RST3, RST4, RST5, RST6, & RST7

(vi) ***Hardware interrupt:*** There are 5 interrupt pin in 8085 used as hardware interrupt ie. TRAP,RST7.5, RST6.5, RST5.5, INTA

9. List the features of 8086 microprocessor with its block diagram.

Ans:- The Feature of 8086 Microprocessors are …

(i) 16 bit microprocessor
(ii) 20 bit address bus and 16 bit internal data bus.
(iii) 4 general purpose registers AX, BX, CX and DX that store 16 bit each.
(iv) segmented memory with each segment having 64KB store capacity
(v) 4 segments register,
(vi) 9 flag.



www.bcanotesnepal.com

10. 8086 Microprocessor Block Diagram

11. Write short notes on:

Control and Status Signals, :-

These signals include two control signals (RD & WR) three status signals (IO/M, S1 and S0) to identify the nature of the operation and one special signal (ALE) to indicate the beginning of the operations.

- **ALE (output)** - Address Latch Enable. This signal helps to capture the lower order address presented on the multiplexed address / data bus. When it is the pulse, 8085 begins an operation. It generates AD0 - AD7 as the separate set of address lines A0 –A7.
- **RD (active low)** - Read memory or IO device. This indicates that the selected memory location or I/O device is to be read and that the data bus is ready for accepting data from the memory or I/O device.

www.bcanotesnepal.com

- **WR (active low)** - Write memory or IO device. This indicates that the data on the data bus is to be written into the selected memory location or I/O device.
- **IO/M (output)** - Select memory or an IO device. This status signal indicates that the read / write operation relates to whether the memory or I/O device. It goes high to indicate an I/O operation. It goes low for memory operations.

$IO/\overline{M}$	S_1	S_0	States
0	0	1	Memory Write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch

Flags,

www.bcanotesnepal.com

Instruction Cycle

The necessary steps that the CPU carries out to fetch an instruction and necessary data from the memory and to execute it constitute an instruction cycle. Moreover, it is defined as the time required to complete the execution of an instruction.

An instruction cycle consists of fetch cycle and execute cycle. In fetch cycle CPU fetches opcode from the memory. The necessary steps which are carried out to fetch an opcode from memory constitute a fetch cycle. The necessary steps which are carried out to get data if any from the memory and to perform the specific operation specified in an instruction constitute an execute cycle. The total time required to execute an instruction is given by IC = Fc + Ec. The 8085 consists of 1-6 machine cycles or operations.

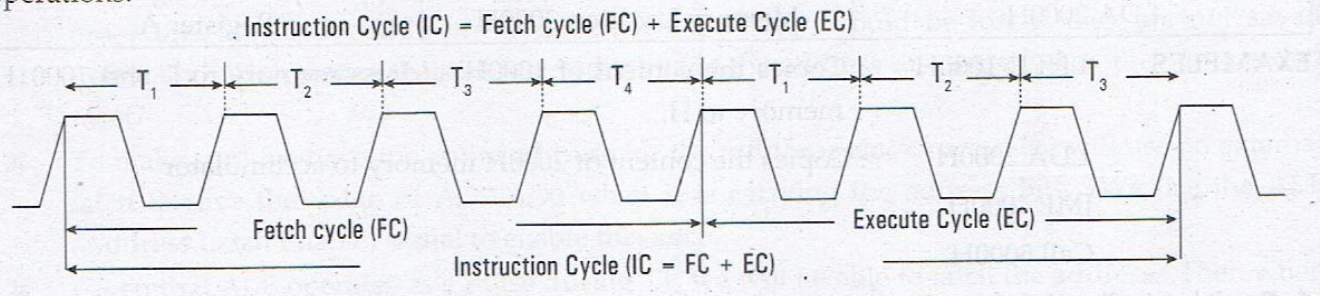


Figure : Timing diagram of instruction cycle

Machine Cycle:- ***Machine cycle*** is define as the time required to complete one operation of accessing memory i/p ,o/p or acknowledging and external request. This cycle may consists of 3 to 6 T state.

T-States :- ***T-state*** is define as one sub division of the operation performed in one clock period. These sub division are internal states synchronized with system clock and each state precisely equal to one clock

www.bcanotesnepal.com

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

period. Each of these operation performs a specific function some machine cycle are opcode fetch M.C. (4T),Memory Read MC. (3T). I/O read MC(T3),I/O write MC(T3), Interrupt MC (TC).

Unit 2: Introduction to Assembly Language Programming (10 Hrs.)

Assembly Language Programming Basics, Classifications of Instructions and Addressing Mode, 8085 Instruction Sets, Assembling, Executing and debugging the Programs, Developing Counters and Time Delay Routines, Interfacing Concepts.

[Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 5th Edition,

Questions solving :

www.bcanotesnepal.com

1. Define Assembling. Explain the merits and demerits of Assembly Language Programming*(Page no 52&53)*

- Assembly Language uses two, three or 4 letter mnemonic store present each instruction type.
- Low level Assembly Language is designed for a specific family of Processors: the symbolic instruction directly relate to Machine Language instructions one for one and are assembled into machine language
- To make programming easier, many programmers write programs in assembly language
- They then translate Assembly Language program to machine language so that it can be loaded into memory and run.

Advantages of Assembly Language

- A Program written in Assembly Language requires considerably less Memory and execution time than that of High Level Language.
- Assembly Language gives a programmer the ability to perform highly technical tasks.
- Resident Programs (that resides in memory while other programs execute) and Interrupt Service Routine (that handles I/P and O/P) are almost always developed in Assembly Language.
- Provides more control over handling particular H/W requirements.
- Generates smaller and compact executable modules.
- Results in faster execution.

2. Explain the 8085 Programming model.

www.bcanotesnepal.com

3. Classify the 8085 Instruction Set with example.

Instruction Set of 8085

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function.
- The entire group of instructions that a microprocessor supports is called ***Instruction Set***.
- 8085 has **246** instructions.
- Each instruction is represented by an 8-bit binary value.
- These 8-bits of binary value is called ***Op-Code*** or ***Instruction Byte***.

Classification of Instruction Set

•Data Transfer Instruction

•Arithmetic Instructions
•Logical Instructions
•Branching Instructions
•Control Instructions

Explain in 8085 details Pdf file ma xa

4. Write short notes on: Instruction word size, Data format

www.bcanotesnepal.com

Unit 3: Basic Computer Architecture (4 Hrs.)

Introduction: History of computer architecture, **[William Stallings, Computer Organization and Architecture, 9th Edition, Part – One, Chapter 2: "Computer Evolution and Performance"]**
Overview of computer organization, **[William Stallings, Computer Organization and Architecture, 9th Edition, Part – One, Chapter 1: "Introduction"]**
Memory hierarchy and Cache, **[William Stallings, Computer Organization and Architecture, 9th Edition, Part – Two, Chapter 4: "Cache Memory"]**
Organization of hard disk, **[William Stallings, Computer Organization and Architecture, 9th Edition, Part – Two, Chapter 6: "External Memory", 6.1: Magnetic Disk]**
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Twelve: "Memory Organization"].
Instruction Codes: Stored Program Organization, Indirect address, Computer Registers, Common Bus systems, Instruction set, Timing and Control, Instruction Cycle.
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Five: "Basic Computer Organization and Design" from 5-1 to 5-5].

Questions solving :

1. Explain the role of John Von Neumann in development of computer.(Page no 5& 6)

Ans:-There are two types of digital computer architures that describe the functional and implemention of computer system. One is the Jhon Von Neumann Arctecture that was designed by the renowned physicst and mathematician Jhon Von Neumann in the 1940s. and the other one is the Harvard architure which was based on the original Harvared Marl – I realy- based computer which employed separate memory system to store data and instructions.

Jhon Von Neumann was a theoretical designed based on the concept of stored- program computer where program data and instruction data are stored in the same memory.

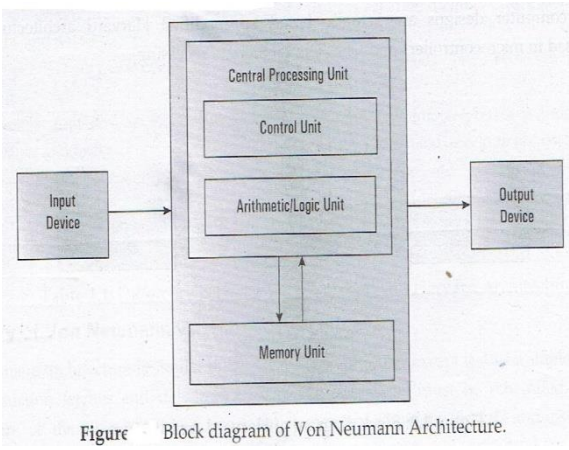


Figure Block diagram of Von Neumann Architecture.

The architecture was designed by the renowned mathematician and physicist John Von Neumann in 1945. Until the Von Neumann concept of computer design, computing machines were designed for a single predetermined purpose that would lack sophistication because of the manual rewiring of circuitry.

www.bcanotesnepal.com

The idea behind the Von Neumann architectures is the ability to store instructions in the memory along with the data on which the instructions operate. In short, the Von Neumann architecture refers to a general framework that a computer's hardware, programming, and data should follow.

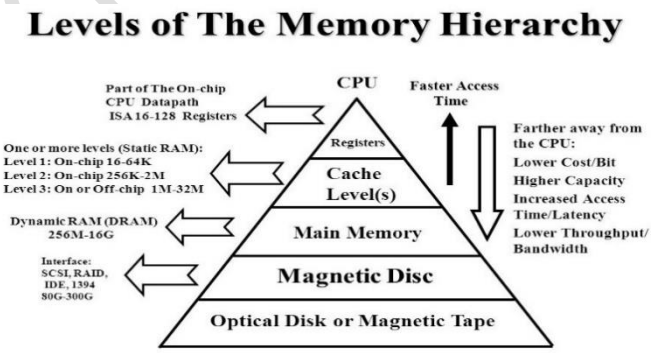
The Von Neumann architecture consists of three distinct components: a central processing unit (CPU), memory unit, and input/output (I/O) interfaces. The CPU is the heart of the computer system that consists of three main components: the Arithmetic and Logic Unit (ALU), the control unit (CU), and registers.

The ALU is responsible for carrying out all arithmetic and logic operations on data, whereas the control unit determines the order of flow of instructions that need to be executed in programs by issuing control signals to the hardware.

The registers are basically temporary storage locations that store addresses of the instructions that need to be executed. The memory unit consist of RAM, which is the main memory used to store program data and instructions. The I/O interfaces allows the users to communicate with the outside world such as storage devices.

2. Explain the Memory Hierarchy with diagram.

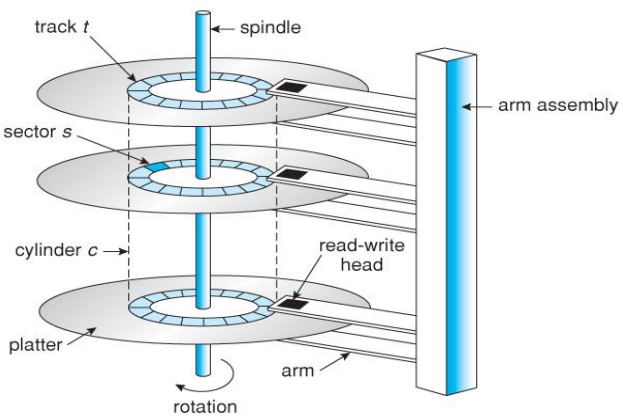
Ans:- In computer architecture, the *memory hierarchy* separates computer storage into a *hierarchy* based on response time. Since response time, complexity, and capacity are related, the levels may also be distinguished by their performance and controlling technologies.



3. Explain the elements of cache design.

4. Explain the internal structure of Hard Disk.

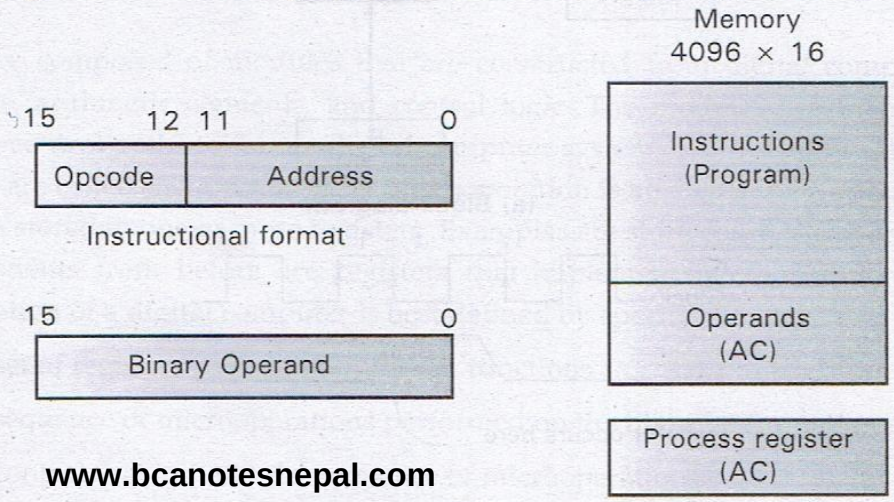
- Hard drives are usually devices that can be used for storing data and to retrieve it. They are composed of one or more than one rigid disc that rapidly rotates.
- Hard disks in them contain various components such as Platter, Spindle, Actuator, power connector, Jumper block, IDE connector etc.
- In the inside of a hard disk there is the spindle which is used to hold circular magnetic disks also known as platters. They are usually made up of a non magnetic material,
- These platters are spun at speeds in excess of 4000 RPM, nowadays 7200 RPM based hard disks are common enough.
- The read and write heads work on these spinning platters to read and write data. They are controlled by an actuator arm that prevents any error.



5. Explain the concept of Stored Program Organization. (Pg.154)

Stored Program Organization

The simplest way to organize a computer is to have one processor register and an instruction code format with two parts. The first part specifies the operation to be performed and the second specifies an address and the memory address tells the control where to find an operand in memory. This operand is read from memory and used as the data to be operated on together with the data stored in the processor register. The following figure 4.3 shows this type of organization.



www.bcanotesnepal.com

Figure : *Stored Program Organization*

Instructions are stored in one section of memory and data in another. For a memory unit with 4096 words, we need 12 bits to specify an address since $2^{12} = 4096$. If we store each instruction code in one 16-bit memory word, we have available four bits for operation code (abbreviated op code) to specify one out of 16 possible operations, and 12 bits to specify the address of an operand. The control reads a 16-bit instruction from the program portion of memory.

It uses the 12-bit address part of the instruction to read a 16-bit operand from the data portion of memory. It then executes the operation specified by the operation code.

Computers that have a single-processor register usually assign to it the name accumulator and label it AC. If an operation in an instruction code does not need an operand from memory, the rest of the bits in the instruction can be used for other purposes. For example, operations such as clear AC, complement AC, and increment AC operate on data stored in the AC register. They do not need an operand from memory. For these types of operations, the second part of the instruction code (bits 0 through 11) is not needed for specifying a memory address and can be used to specify other operations for the computer.

6. Demonstrate the theory of direct and indirect address with instruction format diagrams. (Pg.154,155)

Ans:- These parts of an instruction formate specifies the address of an operand, the instruction is said to have a direct address .In Indirect address, bits in the second parts of the instruction designate an

www.bcanotesnepal.com

address of a memory word in which the address of the operand is found. One bit of the instruction code can be used to distinguish between a direct and an indirect address. It consists of a 3-bit operation code, a 12-bit address, and an indirect address mode bit designated by I where, the mode bit is 0 for a direct address and 1 for an indirect address. A direct address instruction is shown in Figure It is placed in address 22 in memory. The I bit is 0, so the instruction is recognized as a direct address instruction. The op code specifies an ADD instruction, and the address part is the binary equivalent of 457. The control finds the operand in memory at address 457 and adds it to the content of AC.

The instruction in address 35 shown in Figure has a mode bit I = 1, recognized as an indirect address instruction. The address part is the binary equivalent of300. The control goes to address 300 to find the address of the operand. The address of the operand in this case is 1350. The operand found in address 1350 is then added to the content of AC. The indirect address instruction needs two references to memory to fetch an operand.

- The first reference is needed to read the address of the operand
- Second reference is for the operand itself.

The memory word that holds the address of the operand in an indirect address instruction is used as a pointer to an array of data.

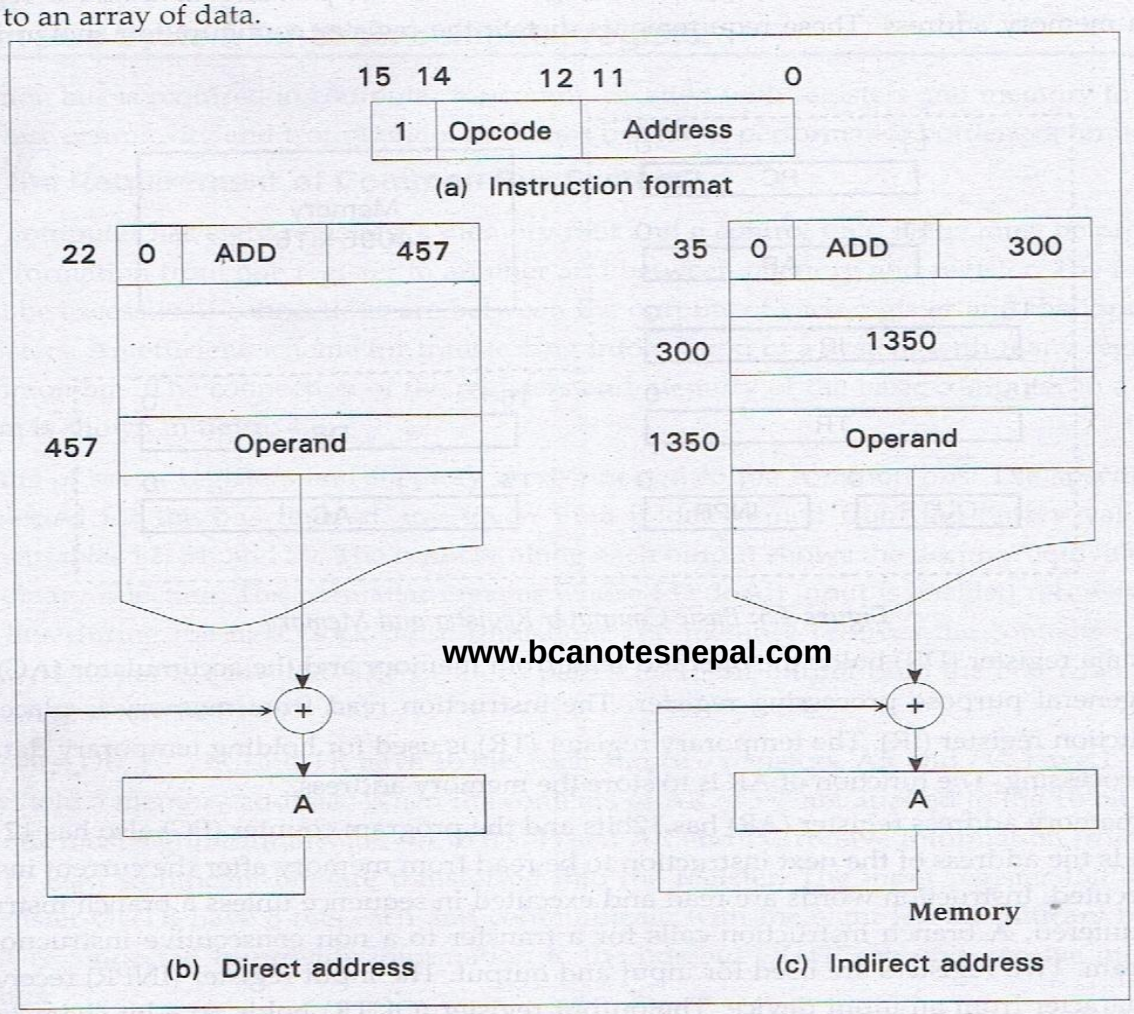


Figure *Direct and indirect advertising of a basic computer*

7. Define registers. Explain the basic computer registers with their uses. (Pg. 157)

Ans:- Register is the high speed memory device used for storing data during instruction execution .

Computer Registers

It is necessary to provide a register in the control unit for storing the instruction code after it is read from memory. The computer needs processor registers for manipulating data and a register for holding a memory address. These requirements dictate the register configuration shown in Figure

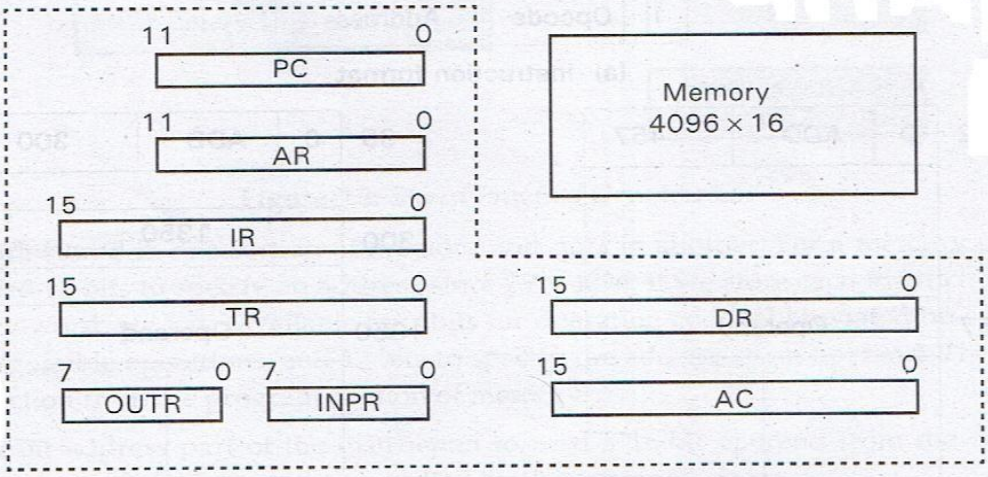


Figure : Basic Computer Register and Memory

The data register (DR) holds the operand read from memory and the accumulator (AC) register is a general purpose processing register. The instruction read from memory is placed in the instruction register (IR). The temporary register (TR) is used for holding temporary data during the processing. The function of AR is to store the memory address.

The memory address register (AR) has 12bits and the program counter (PC) also has 12 bits and it holds the address of the next instruction to be read from memory after the current instruction is executed. Instruction words are read and executed in sequence unless a branch instruction is encountered. A branch instruction calls for a transfer to a non consecutive instruction in the program. Two registers are used for input and output. The input register (INPR) receives an 8-bit character from an input device. The output register (OUTR) holds an 8-bit character for an output device.

8. Define Bus System. Explain the bus system construction mechanisms using Multiplexers and Three - state buffer gates.

www.bcanotesnepal.com

9. Explain the instruction set design issues.

10. Explain the structure of control unit of basic computer.

11. Define Instruction cycle. Explain the instruction cycle with state diagram. (Pg.162,163)

Ans :- ***Instruction cycle:-*** the necessary step that the CPU carries out to fetch an instruction and necessary data from the memory and to execute it constitute an instruction cycle . Moreover it is defined as the time required to complete the execution of an instruction . an instruction cycle consists of fetch cycle and execute cycle.

Instruction Cycle

- A program residing in the memory unit of the computer consists of a sequence of instructions. In the basic computer each instruction cycle consists of the following phases:
  - Fetch an instruction from memory.
  - Decode the instruction.
  - Read the effective address from memory if the instruction has an indirect address.
  - Execute the instruction.
- After step 4, the control goes back to step 1 to fetch, decode and execute the next instruction.
- This process continues unless a HALT instruction is encountered.

T_0 : $AR \leftarrow PC$

T_1 : $IR \leftarrow M[AR], \quad PC \leftarrow PC + 1$

T_2 : $D_0, \ldots, D_7 \leftarrow$ Decode IR (12 – 14), $AR \leftarrow IR(0 - 11), I \leftarrow IR(15)$

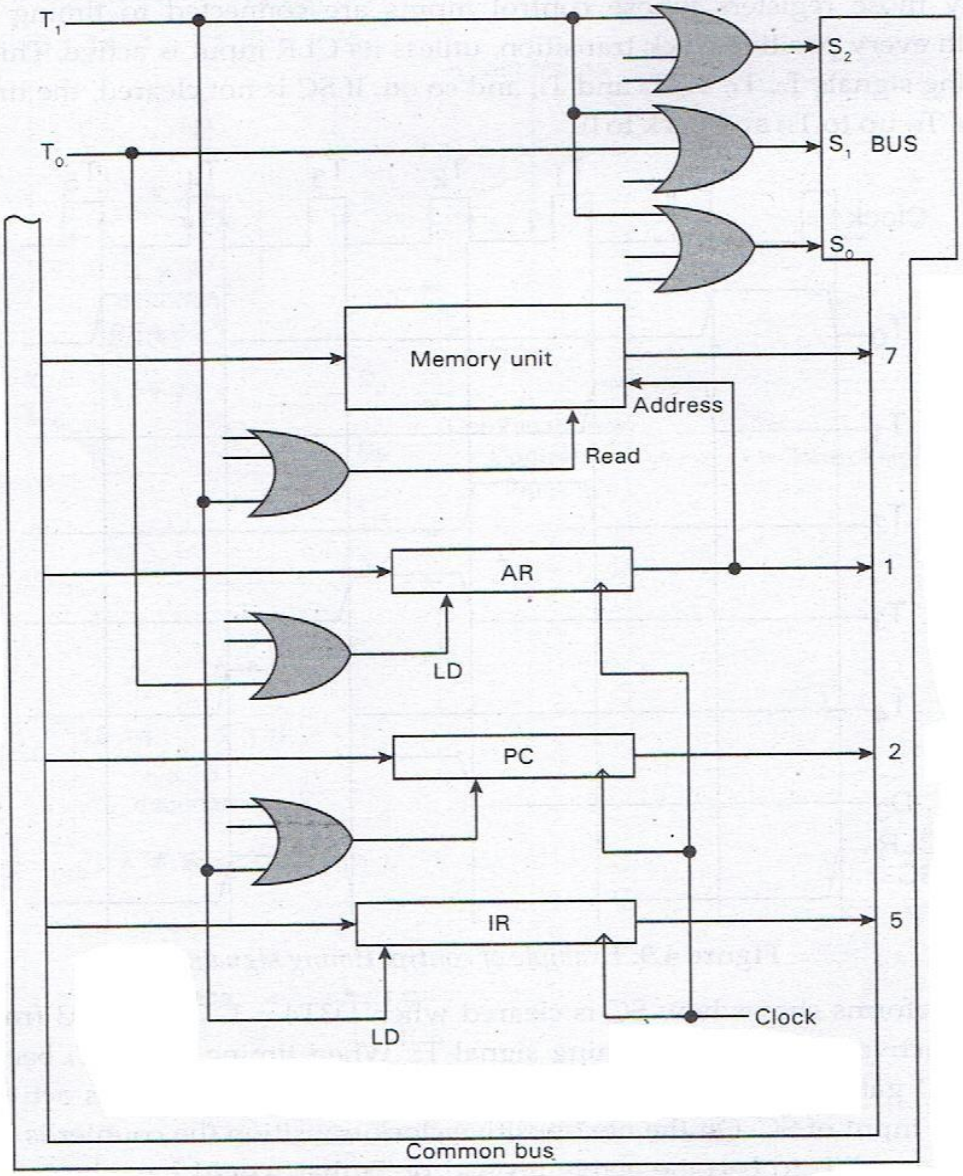


Figure : *Register transfer for the fetch phase*

www.bcanotesnepal.com

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

As shown in the above figure:

- *When T_0 =1, $s_2 s_1 s_0$ = 010: PC gets the bus, AR gets load signal and finally the contents of bus (PC) is loaded to AR.*
- *When T_1=1, $s_2 s_1 s_0$ = 111: Memory unit gets the bus, IR gets load signal and finally the content of memory is transferred to IR. Also PC gets INR signal and hence incremented by 1.*
- *When T_2 = 1: 12-14 bits of IR is decoded, 0-11 bits of IR are transferred to AR and 15 bit of IR is transferred to flag I.*

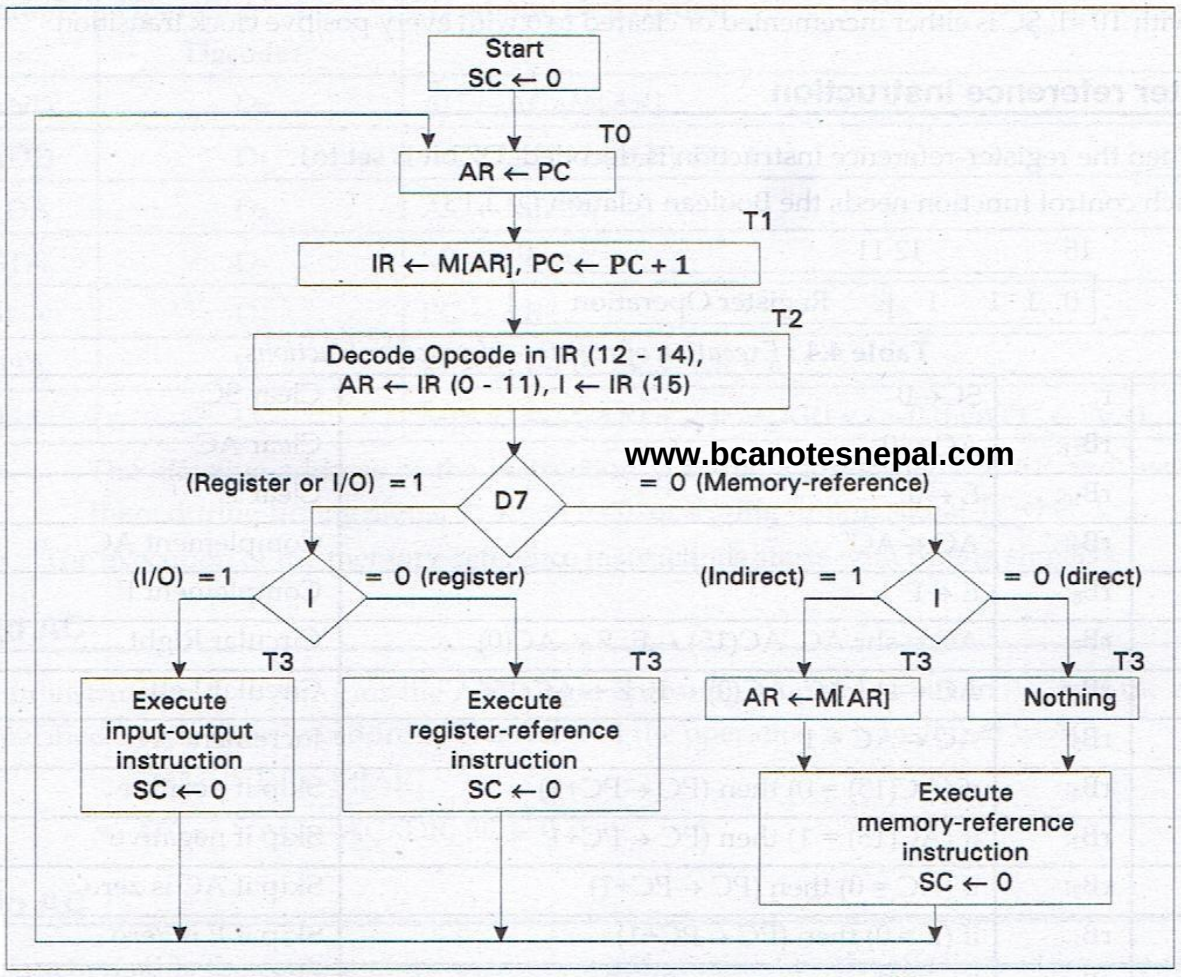


Figure : *Flowchart for instruction cycle (initial configuration)*

Unit 4: Micro programmed Control (10 Hrs.)

Basic Design of Accumulator: Control of AC register, ALU organization,
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Five: "Basic Computer Organization and Design" from 5-10].
Control Memory, Address Sequencing: *Conditional Branching, Mapping of Instruction, Subroutines*, Microprogram: *Symbolic Microprogram, Binary Microprogram*, Design of Control Unit, Basic requirement of Control Unit, Structure of Control Unit, Microprogram Sequencer.
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Seven: "Micro programmed Control" from 7-1 to 7-4].
[William Stallings, Computer Organization and Architecture, 8th Edition, Part – Four, Chapter 15 & 16].

Questions solving :

1. Explain the Design procedure of Accumulator Logic.

Ans:-

2. Explain the Gate structure for controlling the LD, INR, and CLR of Accumulator.

Ans:-

3. Explain the components of ALU with their functions.

Ans:-

www.bcanotesnepal.com

4. Define the Control Unit. Explain the organization of Micro programmed Control unit.

Ans:-

5. Differentiate between Hardwired and Micro programmed Control Design.

Ans:-

6. Explain the address sequencing procedure.

Ans:-

7. Explain the conditional branching mechanism.

Ans:-

8. What do you mean by mapping of instructions? Explain the procedure for mapping from instruction code to microinstruction address.

Ans:-

9. Define Micro program. (Pg 180)

Micro program

- A sequence of microinstructions constitutes a microprogram.
- Since alterations of the microprogram are not needed once the control unit is in operation, the control memory can be a read-only memory (ROM).
- ROM words are made permanent during the hardware production of the unit.
- The use of a micro program involves placing all control variables in words of ROM for use by the control unit through successive read operations.
- The content of the word in ROM at a given address specifies a microinstruction.

10. Differentiate between Symbolic and Binary micro program with example. (180)

Ans:-

11. Explain the basic requirements for designing control unit.

Ans:-

12. Explain the structure of Control Unit.

Ans:-

13. Write the role of micro program sequencer in micro programmed control unit.'

Ans:-

Unit 5: Central Processing Unit (10 Hrs.)

General Register Organization: Control Word, Stack Organization, Instruction formats, addressing modes. 5

Data Transfer and Manipulation: Data transfer instructions, Data manipulation instructions, Arithmetic instructions, Logical and Bit Manipulation Instructions, Shift Instructions.

Program Control: Status Bit Conditions, Conditional Branch Instructions, Subroutine Call and Return, Program Interrupt, Types of Interrupt.

[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Eight: "Central Processing Unit" from 8-1 to 8-8].

Questions solving :

www.bcanotesnepal.com

1. Explain the Register set with common ALU.

Ans:-

2. Explain the general register organization

Ans:-

3. Define control word. Explain the procedure for determining control word for specific operation.

Ans:-

4. Define stack.

Ans:- A stack is a storage device that be store information in such a manner that the item stored last is the first item retrieved . The stack in digital computer is essentially a memory unit with an address register that can count only. The register that hold the address for the stack is called a stack pointer. Because its value always points at the top item in the stack . The physical register of a stack are always available for reading or writing. It is the content of n the word that is inverted or deleted.

5. Explain the stack organization.

Ans:-

6. Explain the different instruction formats with examples.

Ans:-

7. Explain the different types of instruction addressing modes.

Ans:-

8. Explain the different Data Transfer and Manipulation instruction

www.bcanotesnepal.com

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

Data Transfer Instructions

Data transfer instructions move data from one place in the computer to another without changing the data content. The most common transfers are between memory and processor registers, between processor registers and input or output, and between the processor registers themselves. The *load* instruction has been used mostly to designate a transfer from memory to a processor register, usually an accumulator and the *store* instruction designates a transfer from a processor register into memory. The *move* instruction has been used in computers with multiple CPU registers to designate a transfer from one register to another. It has also been used for data transfers between CPU registers and memory or between two memory words. The *exchange* instruction swaps information between two registers or a register and a memory word. The *input and output* instructions transfer data among processor registers and input or output terminals and the *push and pop* instructions transfer data between processor registers and a memory stack.

Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

www.bcanotesnepal.com

Data Transfer Instructions

Typical data transfer instruction

Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Data transfer instructions with different addressing modes

Mode	Assembly Convention		Register Transfer
Direct address	LD	ADR	AC ← M[ADR]
Indirect address	LD	@ADR	AC ← M[M[ADR]]
Relative address	LD	$ADR	AC ← M[PC + ADR]
Immediate operand	LD	#NBR	AC ←NBR
Index addressing	LD	ADR(X)	AC ← M[ADR + XR]
Register	LD	R1	AC ← R1
Register indirect	LD	(R1)	AC ← M[R1]
Auto increment	LD	(R1) +	AC ← M[R1], R1 ← R1 + 1
Auto decrement	LD	(R1) –	R1 ← R1 - 1 AC ← M[R1]

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

Data Manipulation Instructions

Three Basic Types:

- Arithmetic instructions
- Logical and
- Bit manipulation instructions

Shift instructions

Name	Mnemonic
Logical shift right	SHR
Logical shift left	SHL
Arithmetic shift right	SHRA
Arithmetic shift left	SHLA
Rotate right	ROR
Rotate left	ROL
Rotate right thru carry	RORC
Rotate left thru carry	ROLC

Arithmetic Instructions

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with Carry	ADDC
Subtract with Borrow	SUBB
Negate(2's Complement)	NEG

www.bcanotesnepal.com

Logical and Bit Manipulation Instructions

Name	Mnemonic
Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC
Enable interrupt	EI

9. What are control instructions? Explain the different types of program control instructions with their roles.

Ans:-

10. Define program interrupt. Explain the types of interrupt. (Pg. 209,210)

Program interrupt

The concept of program interrupt is used to handle a variety of problems that arise out of normal program sequence. Program interrupt refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request. Control returns to the original program after the service program is executed. After a program has been interrupted and the service routine been executed, the CPU must return to exactly the same state that it was when the interrupt occurred. Only if this happens will the interrupted program be able to resume exactly as if nothing had happened. The state of the CPU at the end of the execution cycle (when the interrupt is recognized) is determined from:

1. The content of the program counter
2. The content of all processor registers
3. The content of certain status conditions

The interrupt facility allows the running program to proceed until the input or output device sets its ready flag. Whenever a flag is set to 1, the computer completes the execution of the instruction in progress and then acknowledges the interrupt. The result of this action is that the return address is stored in location 0. The instruction in location 1 is then performed; this initiates a service routine for the input or output transfer. The service routine can be stored in location 1. The service routine must have instructions to perform the following tasks:

1. Save contents of processor registers.
2. Check which flag is set.
3. Service the device whose flag is set.
4. Restore contents of processor registers.
5. Turn the interrupt facility on.
6. Return to the running program.

www.bcanotesnepal.com

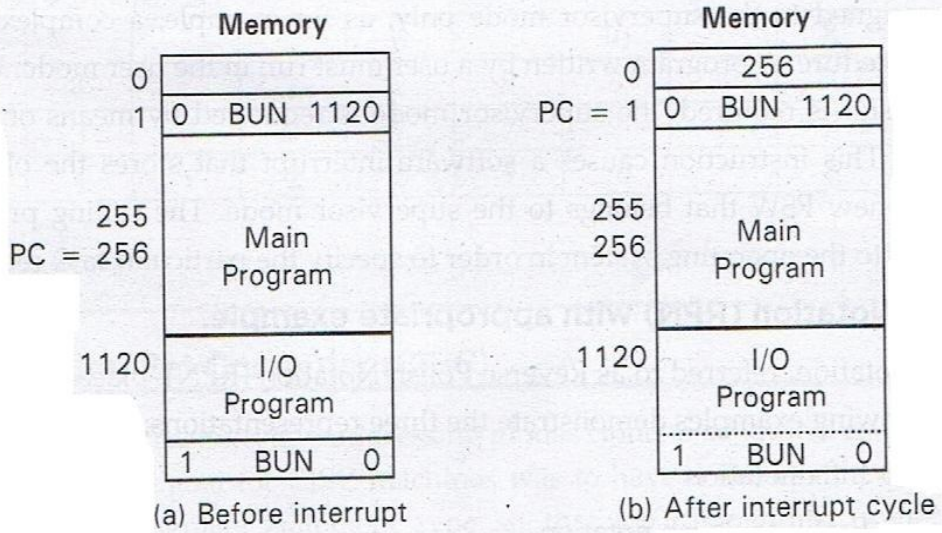


Figure : *Interrupt cycle*

Types of interrupts

There are three major types of interrupts that cause a break in the normal execution of a program. They can be classified as:

1. External interrupts
2. Internal interrupts
3. Software interrupts

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

1. **External interrupts:** External interrupts come from input-output (I/0) devices, from a timing device, from a circuit monitoring the power supply, or from any other external source. Examples that cause external interrupts are I/0 device requesting transfer of data, I/o device finished transfer of data, elapsed time of an event, or power failure. Timeout interrupt may result from a program that is in an endless loop and thus exceeds its time allocation. Power failure interrupt may have as its service routine a program that transfers the complete state of the CPU into a nondestructive memory in the few milliseconds before power ceases. External interrupts are asynchronous. External interrupts depend on external conditions that are independent of the program being executed at the time.

2. **Internal interrupts:** Internal interrupts arise from illegal or erroneous use of an instruction or data. Internal interrupts are also called traps. Examples of interrupts caused by internal error conditions are register overflow, attempt to divide by zero, an invalid operation code, stack overflow, and protection violation. These error conditions usually occur as a result of a premature termination of the instruction execution. The service program that processes the internal interrupt determines the corrective measure to be taken. Internal interrupts are synchronous with the program. If the program is rerun, the internal interrupts will occur in the same place each time.

3. **Software interrupts:** A software interrupt is a special call instruction that behaves like an interrupt rather than a subroutine call. It can be used by the programmer to initiate an interrupt procedure at any desired point in the program. The most common use of software interrupt is associated with a supervisor call instruction. This instruction provides means for switching from a CPU user mode to the supervisor mode. Certain operations in the computer may be assigned to the supervisor mode only, as for example, a complex input or output transfer procedure. A program written by a user must run in the user mode. When an input or output transfer is required, the supervisor mode is requested by means of a supervisor call instruction. This instruction causes a software interrupt that stores the old CPU state and brings in a new PSW that belongs to the supervisor mode. The calling program must pass information to the operating system in order to specify the particular task requested.

11. Write short notes on:
    Status bit conditions,

www.bcanotesnepal.com

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

Conditional branch Instructions

Mnemonic	Branch condition	Tested condition
BZ	Branch if zero	Z =1
BNZ	Branch if not zero	Z =0
BC	Branch if carry	C =1
BNC	Branch if no carry	C =0
BP	Branch if plus	S =0
BM	Branch if minus	S =1
BV	Branch if over flow	V =1
BNV	Branch if no over flow	V =0
***Unsigned* compare conditions (A - B)**		
BHI	Branch if higher	A >B
BHE	Branch if higher or equal	A ≥B
BLO	Branch if lower	A <B
BLOE	Branch if lower or equal	A ≤B
BE	Branch if equal	A =B
BNE	Branch if not equal	A ≠B
***Signed* compare conditions (A - B)**		
BGT	Branch if greater than	A >B
BGE	Branch if greater or equal	A ≥B
BLT	Branch if less than	A <B
BLE	Branch if less or equal	A ≤B
BE	Branch if equal	A =B
BNE	Branch if not equal	A ≠B

Subroutine Call and Return

www.bcanotesnepal.com

Subroutine Call

Call subroutine Jump to subroutine Branch to subroutine

Branch and save return address

Two Most Important Operations are implied;

- Branch to the beginning of the Subroutine
  - Same as the Branch or Conditional Branch
- Save the Return Address to get the address of the location in the Calling Program upon exit from the Subroutine
  - Locations for storing Return address:
    - Fixed Location in the subroutine(Memory)
    - Fixed Location in memory
    - In a process or Register
    - In a memory stack
    - Most efficient way

```
CALL
SP←SP -1, M[SP] ← PC
PC←EA

RTN
PC ←M[SP]
SP ←SP + 1
```

12. Differentiate between CISC and RISC architecture.'

CISC	RISC
The original microprocessor ISA	Redesigned ISA that emerged in the early 1980s
Instructions can take several clock cycles	Single-cycle instructions
Hardware-centric design — the ISA does as much as possible using hardware circuitry	Software-centric design — High-level compilers take on most of the burden of coding many software steps from the programmer
More efficient use of RAM than RISC	Heavy use of RAM (can cause bottlenecks if RAM is limited)
Complex and variable length instructions	Simple, standardized instructions
May support microcode (micro-programming where instructions are treated like small programs)	Only one layer of instructions
Large number of instructions	Small number of fixed-length instructions
Compound addressing modes	Limited addressing modes

www.bcanotesnepal.com

Unit 6: Pipeline, Vector Processing and Multiprocessors (6 Hrs.)

Parallel Processing, Pipeline Examples: Four segment instruction pipeline, Data dependency, Handling of branch instructions, vector processing, vector operations, matrix multiplication.
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Nine: "Pipelining and Vector Processing" from 9-1 to 9-7].
[M. Morris Mano, Computer System Architecture, 3rd Edition, Chapter Thirteen: "Multiprocessors" from 13-1 to 13-2].

Questions solving :

1. What is parallel processing? Explain the benefits of parallel processing.

Ans ;-

- It refers to techniques that are used to provide simultaneous data processing.
- The system may have two or more ALUs to be able to execute two or more instruction at the same time.
- The system may have two or more processors operating concurrently.
- It can be achieved by having multiple functional units that perform same or different operation simultaneously.

www.bcanotesnepal.com

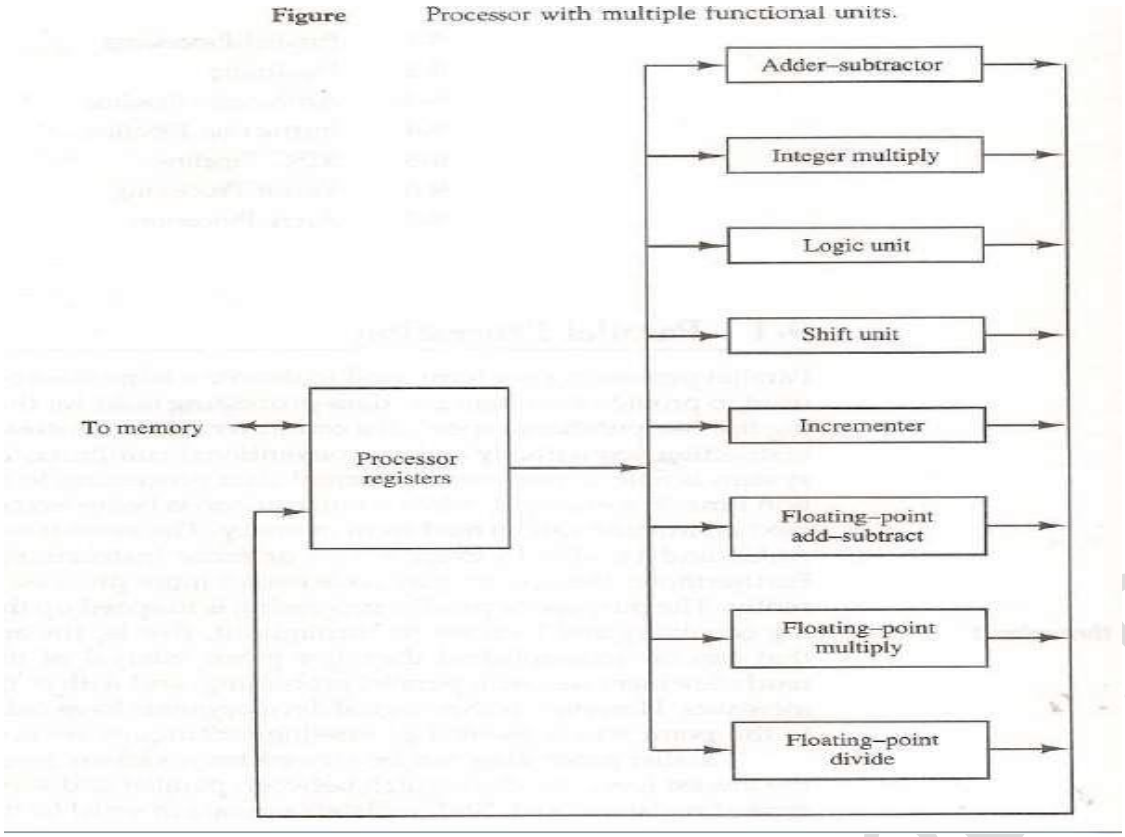


Figure Processor with multiple functional units.

2. Explain the classifications of parallel processing by M. J. Flynn.

M.J. Flynn classify the computer on the basis of number of instruction and data items processed simultaneously.

- Single Instruction Stream, Single Data Stream(SISD)
- Single Instruction Stream, Multiple Data Stream(SIMD)
- Multiple Instruction Stream, Single Data Stream(MISD)
- Multiple Instruction Stream, Multiple Data Stream(MIMD)

- **SISD** represents the organization containing single control unit, a processor unit and a memory unit. Instruction are executed sequentially and system may or may not have internal parallel processing capabilities.
- **SIMD** represents an organization that includes many processing units under the supervision of a common control unit.
- **MISD** structure is of only theoretical interest since no practical system has been constructed using this organization.
- **MIMD** organization refers to a computer system capable of processing several programs at the same time.

3. Explain the role of pipelining in computing.

Ans:-

4. What is pipelining?

Pipelining

- It is a technique of decomposing a sequential process into sub operations, with each sub process being executed in a special dedicated segments that operates concurrently with all other segments. Each segment performs partial processing dictated by the way task is partitioned. The result obtained from each segment is transferred to next segment. The final result is obtained when data have passed through all segments.

Example **www.bcanotesnepal.com**

- Suppose we have to perform the following task:

Once visit on here for Get Everything Related to BCA www.bcanotesnepal.com

- Each sub operation is to be performed in a segment within a pipeline. Each segment has one or two registers and a combinational circuit.
- The sub operations in each segment of the pipeline are as follows:

$R1 \leftarrow A_i,\quad R2 \leftarrow B_i$ — Input A_i and B_i

$R3 \leftarrow R1 * R2,\quad R4 \leftarrow C_i$ — Multiply and input C_i

$R5 \leftarrow R3 + R4$ — Add C_i to product

Figure Example of pipeline processing.

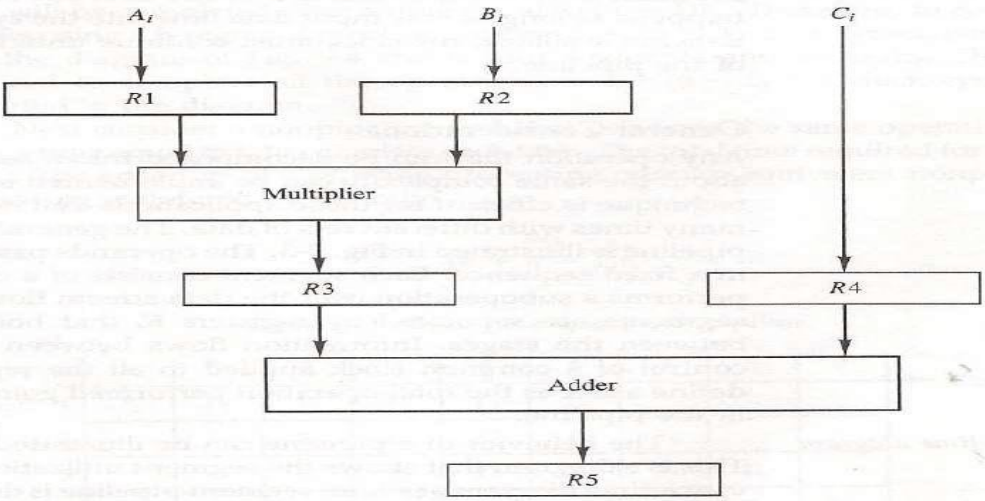


TABLE Content of Registers in Pipeline Example

Clock Pulse Number	Segment 1		Segment 2		Segment 3
	R1	R2	R3	R4	R5
1	A_1	B_1	—	—	—
2	A_2	B_2	$A_1 * B_1$	C_1	—
3	A_3	B_3	$A_2 * B_2$	C_2	$A_1 * B_1 + C_1$
4	A_4	B_4	$A_3 * B_3$	C_3	$A_2 * B_2 + C_2$
5	A_5	B_5	$A_4 * B_4$	C_4	$A_3 * B_3 + C_3$
6	A_6	B_6	$A_5 * B_5$	C_5	$A_4 * B_4 + C_4$
7	A_7	B_7	$A_6 * B_6$	C_6	$A_5 * B_5 + C_5$
8	—	—	$A_7 * B_7$	C_7	$A_6 * B_6 + C_6$
9	—	—	—	—	$A_7 * B_7 + C_7$

www.bcanotesnepal.com

5. Define instruction pipeline. Explain the four segment instruction pipeline.

Instruction Pipeline

- Pipeline processing can occur not only in the data stream but in the instruction stream as well.
- An instruction pipeline reads consecutive instruction from memory while previous instruction are being executed in other segments.
- This caused the instruction fetch and execute segments to overlap and perform simultaneous operation.

6. Explain the different pipeline hazards (conflicts)?

Ans:-

7. How to handle the branch instruction in pipeline? Explain.

Handling of Branch Instruction

- Pre fetch the target instruction.
- Branch target buffer(BTB) included in the fetch segment of the pipeline

 Branch Prediction
 Delayed Branch

RISC Pipeline

 Simplicity of instruction set is utilized to implement an instruction pipeline using small number of sub operation, with each being executed in single clock cycle.
 Since all operation are performed in the register, there is no need of effective address calculation.

Three Segment Instruction Pipeline

 I: Instruction Fetch
 A: ALU Operation
 E: Execute Instruction

Consider now the operation of the following four instructions

1. LOAD: $R1 \leftarrow M$[address 1]
2. LOAD: $R2 \leftarrow M$[address 2]
3. ADD: $R3 \leftarrow R1 + R2$
4. STORE: M[address 3] $\leftarrow R3$

Clock cycles:	1	2	3	4	5	6
1. Load $R1$	I	A	E			
2. Load $R2$		I	A	E		
3. Add $R1 + R2$			I	A	E	
4. Store $R3$				I	A	E

Pipeline timing with data conflict

Clock cycle:	1	2	3	4	5	6	7
1. Load $R1$	I	A	E				
2. Load $R2$		I	A	E			
3. No-operation			I	A	E		
4. Add $R1 + R2$				I	A	E	
5. Store $R3$					I	A	E

Pipeline timing with delayed load

Delayed Branch **www.bcanotesnepal.com**

 Let us consider the program having the following 5 Instructions

Load from memory to $R1$
Increment $R2$
Add $R3$ to $R4$
Subtract $R5$ from $R6$
Branch to address X

Clock cycles:	1	2	3	4	5	6	7	8	9	10
1. Load	I	A	E							
2. Increment		I	A	E						
3. Add			I	A	E					
4. Subtract				I	A	E				
5. Branch to *X*					I	A	E			
6. No-operation						I	A	E		
7. No-operation							I	A	E	
8. Instruction in *X*								I	A	E

Using no-operation instructions

Clock cycles:	1	2	3	4	5	6	7	8
1. Load	I	A	E					
2. Increment		I	A	E				
3. Branch to *X*			I	A	E			
4. Add				I	A	E		
5. Subtract					I	A	E	
6. Instruction in *X*						I	A	E

Rearranging the instructions

www.bcanotesnepal.com

8. Define vector processing. Explain the application areas of vector processing.

Ans:- ***Vector Processing***

- There is a class of computational problems that are beyond the capabilities of the conventional computer.
- These are characterized by the fact that they require vast number of computation and it take a conventional computer days or even weeks to complete.
- Computers with vector processing are able to handle such instruction and they have application in following fields:
  - Long range weather forecasting
  - Petroleum exploration
  - Seismic data analysis
  - Medical diagnosis
  - Aerodynamics and space simulation
  - Artificial Intelligence and expert system
  - Mapping the human genome
  - Image Processing

9. Explain the characteristics of multiprocessor system.

Ans:-

www.bcanotesnepal.com

10. Explain the interconnection structure of multiprocessor system.

Ans:-

11. Write short notes on: Arithmetic pipeline, vector operations, matrix multiplications.