

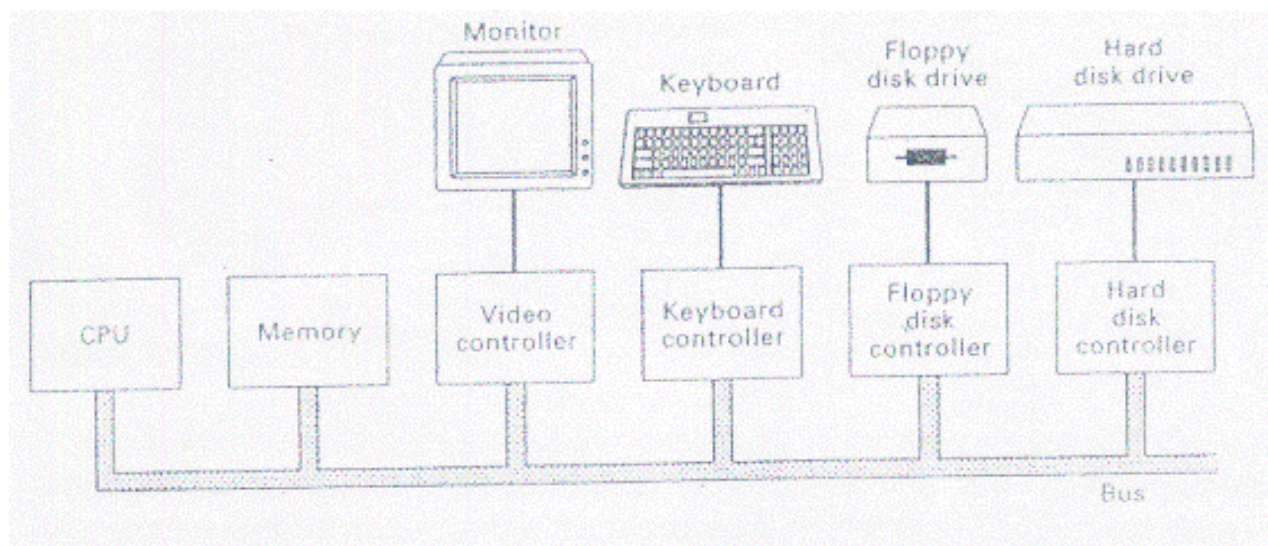
## MICROPROCESSOR AND MICROCOMPUTER BASICS

### Introduction

At present there are many types and sizes of computers available. These computers are designed and constructed based on **digital** and **Integrated Circuit (IC) fabrication technology**.

A digital computer is a machine that can be used to **solve problems** for people by carrying out the task by following the instructions given to it. A sequence of instructions describing how to perform a certain task or job is called a **program**.

A personal computer (PC) uses its memory to store the instructions, typically memory size is 32Mb of RAM. PCs were used for word processing, spreadsheets, and numerous highly interactive applications where larger computers could not handle well.



Structure of a PC

All types of computers, regardless of their size, type and use, consist of at least the following devices:-

- a. Microprocessor or Central Processing Unit (CPU)
- b. Storage device (Memory)
- c. Input/Output devices (I/O)
- d. Instructions or programs

When these devices are connected and make to work together as a system, it is called a **microcomputer system**.

The CPU, memory and I/O are called **hardware** since they are physical devices, whereas the instructions or programs (non-physical component) is called **software**.

### Microprocessor or Central Processing Unit (CPU)

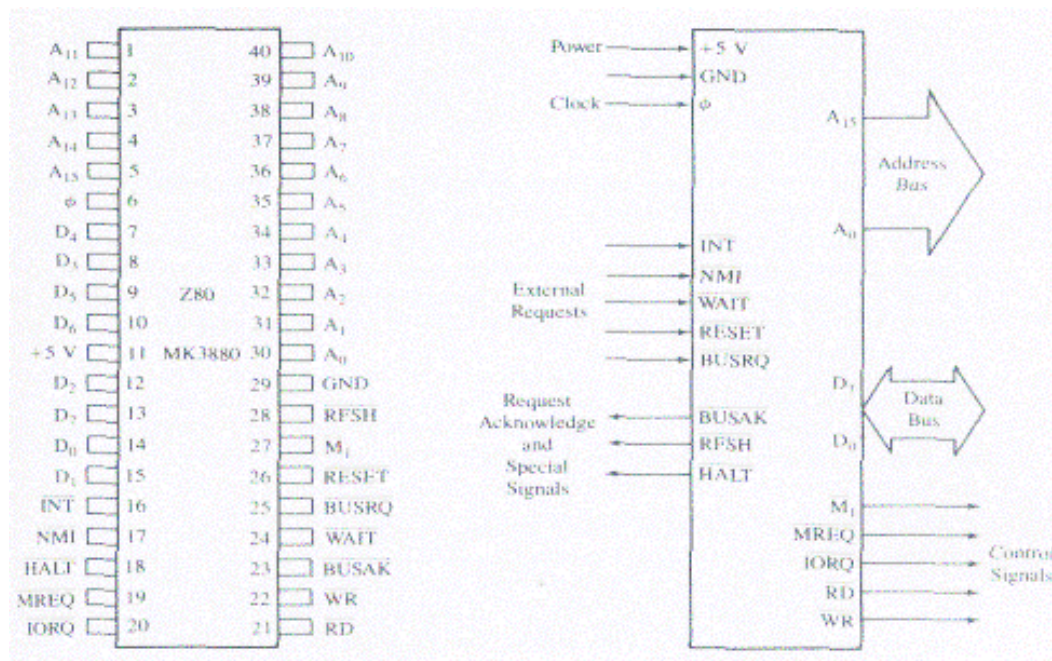
A microprocessor is a **multipurpose, programmable logic device (IC)** that **reads binary instructions** from a storage device called **memory**, accepts binary data as input and **processes data** according to those instructions, and **provide results** as output.

A multipurpose device means it can be used to perform various sophisticated computing tasks or functions, as well as simple tasks.

A programmable device means that it can be instructed to perform given tasks within its capability. Today's microprocessor is designed to understand and execute many binary instructions.

Microprocessor is also called **Central Processing Unit (CPU)** since it is the functional centre of the computer system and it is used to process data.

The first microprocessor was **Intel 4004** (early 1970's) used in calculators. It was designed by Intel Corporation and become known as the 4-bit microprocessor. It was quickly replaced by the 8-bit microprocessor (Intel 8008), which in turn superseded by the Intel 8080. In the mid 1970's, the Intel 8080 was widely used in control applications, and small computers also were designed using the 8080 as the CPU.



The Zilog Z80 Integrated Circuit

Within a few years after the emergence of the Intel 8080 CPU; the Motorola 6800, the **Zilog Z80**, and the Intel 8085 microprocessors were developed as improvements over the 8080.

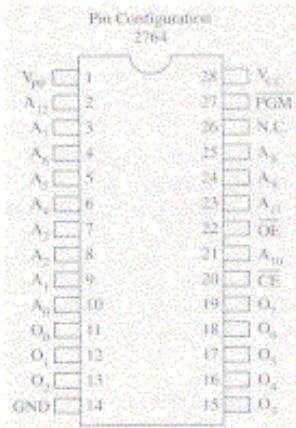
The Zilog Z80 microprocessor came into the market in middle of 1976, made up of 8200 transistors on a single IC.

## Memory

Memory is the term used to the various **storage devices** in which are used to store the programs and data for the microprocessor. These storage devices are made of semiconductor devices, and also known as Primary Storage Devices.

The semiconductor memory is of 2 types that is Read Only Memory (ROM) and Read Write Memory (RWM). RWM is popularly known as Random Access Memory (RAM) :-

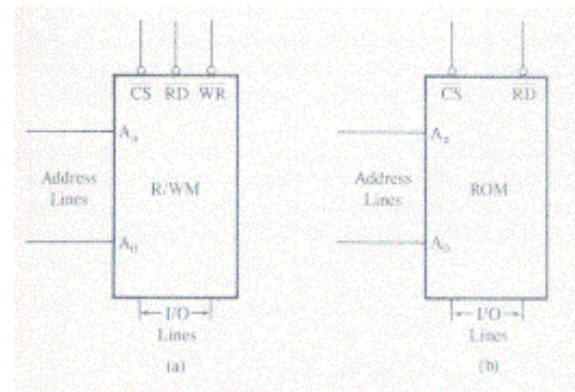
## 1. Read Only Memory (ROM)



It is used to store programs and data that need not to be altered, i.e. permanent storage. Programs and data stored in ROMs can only be read by the CPU.

Special equipment is used to write programs and data into the ROMs and is called EPROM Programmer.

The **monitor program** is normally stored in the ROM. Monitor program is actually the 'resource manager' of the microcomputer system, similar to DOS or Windows in a personal computer. An example of a EPROM chip is the 2764 (8K X 8).



## 2. Read Write Memory (RWM)

It is used to store user programs and data, and can be altered at any time, i.e. temporary storage. The information stored in RAM or RWM can be easily read and altered by the CPU. The contents (data or programs) stored is lost if power supply to this chip is turned off. An example of a RWM chip is the CMOS 6116 (2K X 8).

## Input/Output

The input/output unit allows the microprocessor to **communicate** with the outside world, either to receive or to send data. Most of the time, the input/output unit will also act as an **interface** for the microprocessor, that is to convert the data into a suitable format for the microprocessor. Data can be in the form of parallel (8 bit) or serial format (single line).

Input devices are devices that input data or send data to the computer. Input devices are such as keyboard, punched card readers, sensors, switches, etc.

Output devices are devices that output data or perform various operations under the control of the CPU. Output devices are LEDs, 7-segment display unit, speaker, CRT, printer, digital speedometer, fuel injectors, etc.

## Exercises :

1. The Z80 address bus contains how many lines? What is the address range accessible by Z80 in hexadecimal notation?
2. The Z80 data bus contains how many lines? In hexadecimal notation, what range of values that can be output or input by the Z80?
3. List the four main Z80 control signals which is used to control the memory and I/O unit.

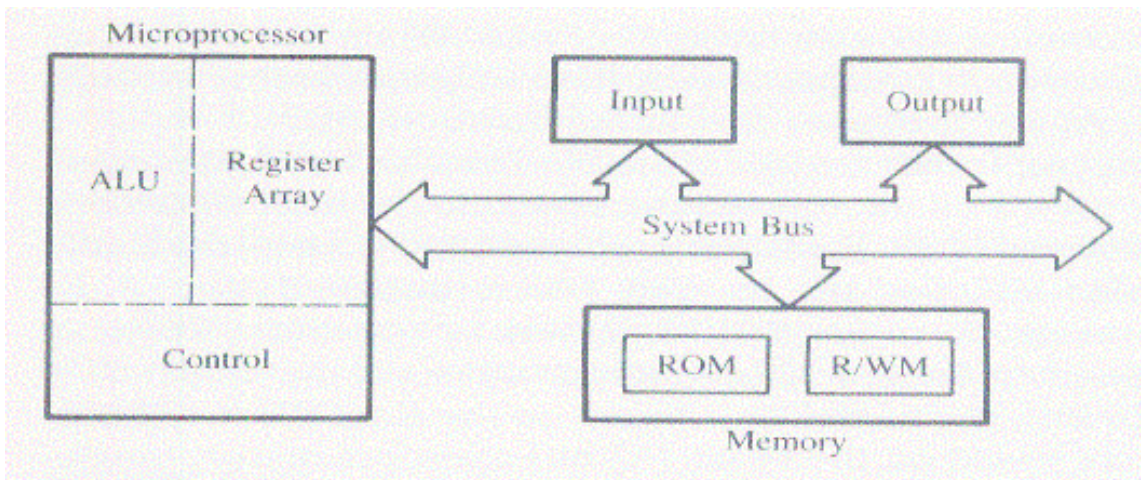




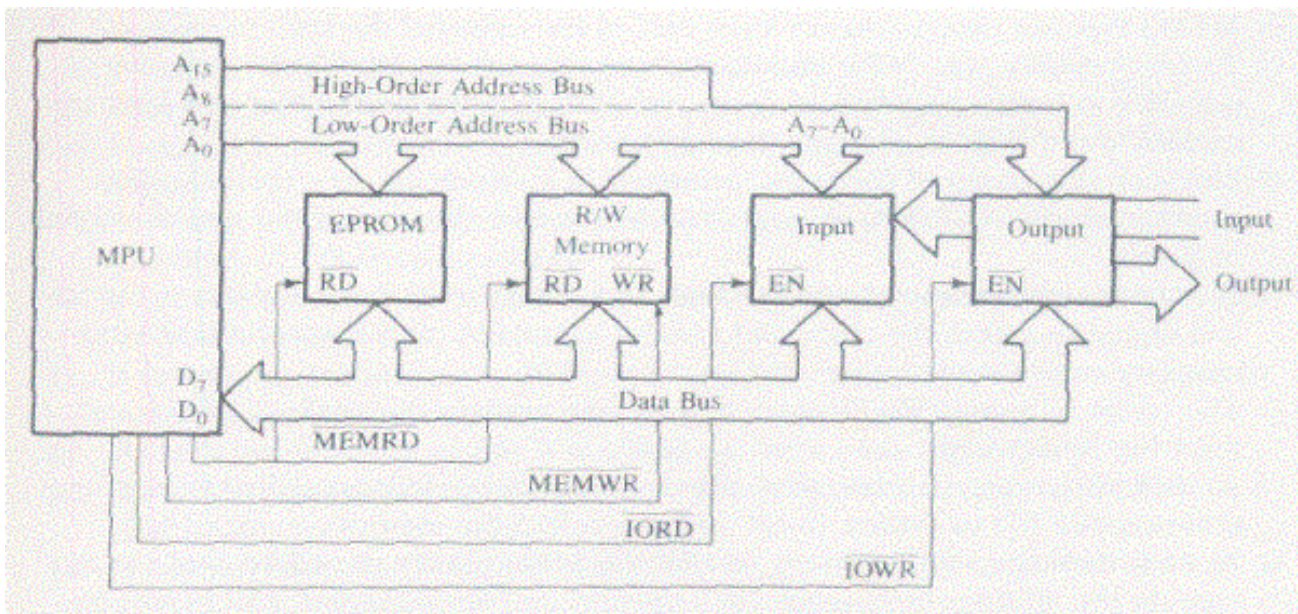
4. Describe the memory of the Z80 and hence draw the internal structure of a memory showing the address and data parts.
5. For the following memory units, calculate its address range in hexadecimal notation and the number of data bits per memory address :-
  - a) (1K x 8)      b) (2k x 8)      c) (4k x 8)
6. What is an Input/Output port?

### BASIC LAYOUT OF A MICROCOMPUTER SYSTEM

In order for the microprocessor to function as a programmable device, it must work in a complete system comprising of three components : **microprocessor**, **memory** and **input/output**. This system is called microprocessor-based system or microcomputer system. These three components will work together or interact with each other to perform a given task.



Basic structure of a microprocessor-based system or microcomputer system.



Block diagram showing the connection details of a microcomputer system. The Input and Output units may be combined as one unit.

## Microprocessor or CPU

The CPU is the functional center of the microcomputer system. Its internal construction can be broadly divided into three sections :- **Control section, Arithmetic and Logic Unit** and **Register section**.

### Control Section

The control section/unit is the part of the microcomputer that controls its basic operations. It is made up of the control signal generating circuitry (clock) and the command (instruction) decoder.

The control section fetches pre-programmed instructions from memory (**op-code fetch cycle**) as needed and temporarily stores them in the command register (also known as **Instruction Register IR**).

These instructions are then decoded by the operation decoder (**decode cycle**), which sends control signals to the relevant parts of the microcomputer system (via the system busses) to cause them to carry out the required operation (**execute cycle**)

The timing with which these control signals are generated is determined by the **clock**. The number of **T-states** tells the time taken for the CPU to execute that particular instruction.

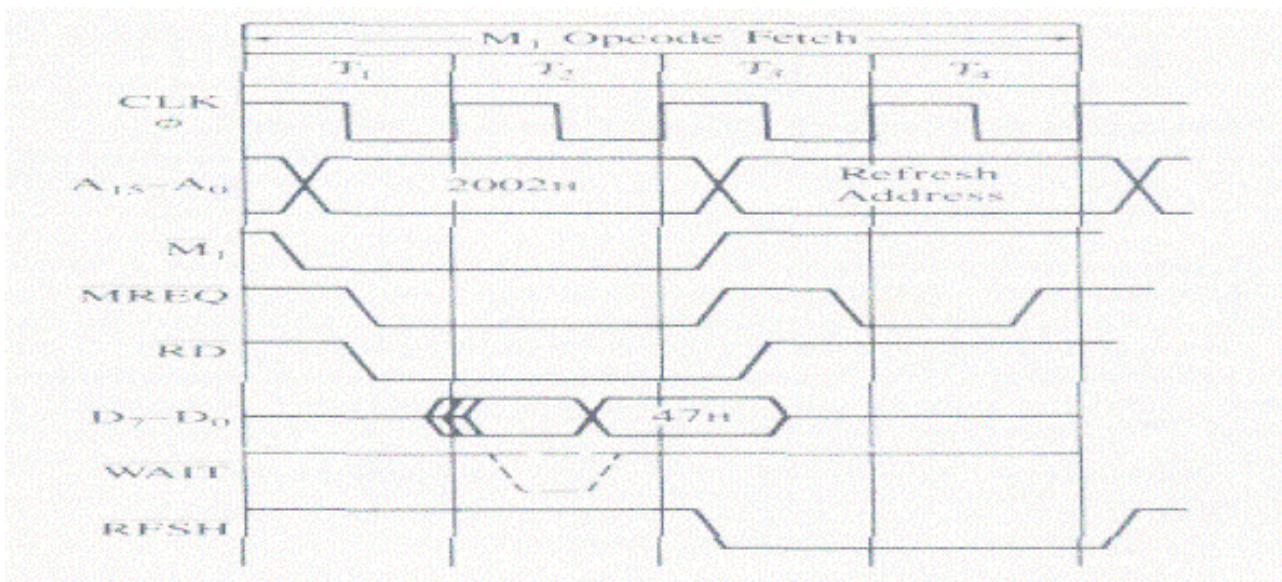
The major types of operations controlled by the control signals are :-

- a. sending of data from one part of the microcomputer to another (read or write cycle)
- b. inputting and outputting of data to/from the microcomputer (I/O read or write cycle)
- c. Arithmetic and Logic calculations.
- d. Halting of computer instructions.
- e. Jumping to another instruction during running (execution) of a program.

To visualize these operations clearly, the timing diagrams below shows the information on the busses.

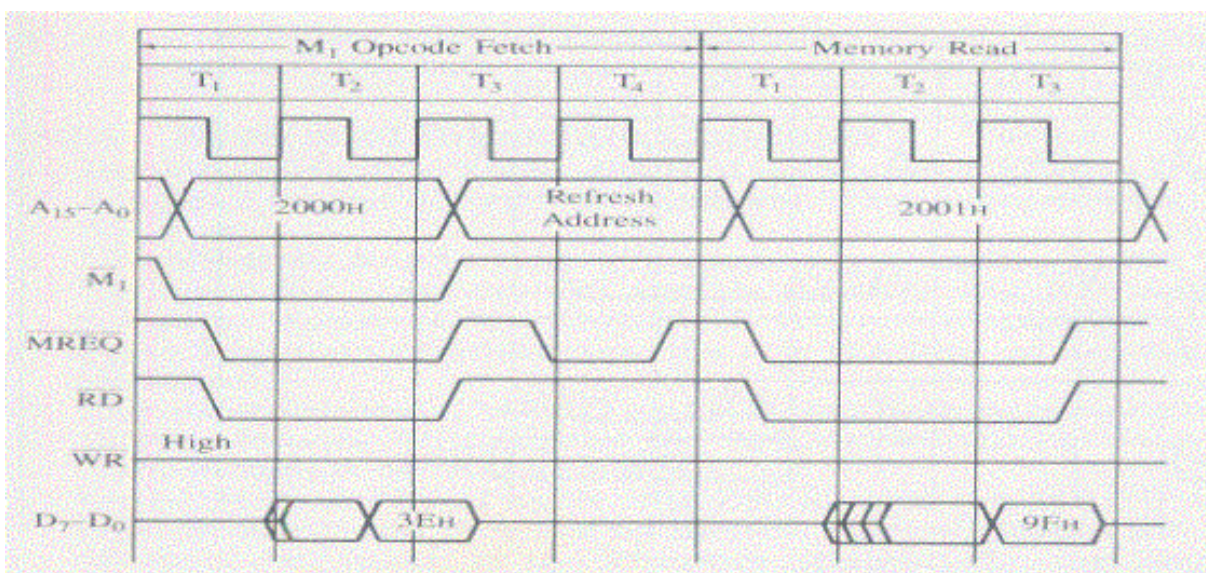
Understanding of these timing diagrams will lead to understanding the various pins functions of the Z80, thus enabling to design suitable circuitry with memory and I/O (memory and I/O decoding).





Z80 Op-code Fetch cycle

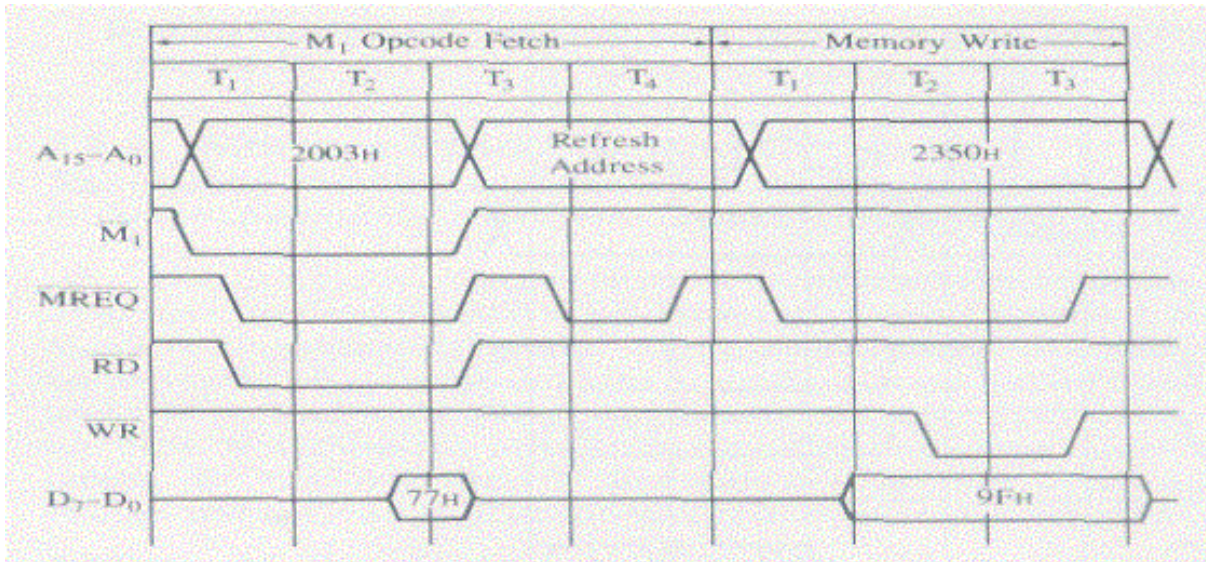
1. The op-code fetch cycle is completed in 4 clock periods or T-states. This machine cycle is also identified as the M1 cycle.
2. At the beginning of T<sub>1</sub>, the control signal M<sub>1</sub> goes low and the contents of Program Counter (PC) ie. 2002H is placed on the address bus.
3. After the falling edge of T<sub>1</sub>, the Z80 asserts MREQ\* and RD\* low. (ie. Read from memory)
4. The memory will identify the address on the address bus (ie 2002H) and after the falling edge of T<sub>2</sub>, places the data found in address 2002H (ie. 47H) on the data bus.
5. At the leading edge of T<sub>3</sub>, the Z80 will read data on the data bus.
6. During T<sub>3</sub> and T<sub>4</sub>, the instruction decoder decodes and executes the op-code. These are internal operations and cannot be observed on the data bus.



Z80 memory read cycle



1. After the completion of op-code fetch cycle, the Z80 places the address 2001H on the address bus and increment PC to the next address 2002H. Observe M1 is inactive.
2. After the falling edge of T1 of the memory read cycle, the signals MREQ\* and RD\* are asserted. This indicate read operation for the memory.
3. After the leading edge of T3, the Z80 reads the data bus and store the byte (ie. 9FH) into one of its register.
4. After the falling edge of T3, both signals MREQ\* and RD\* are inactive, and at the end of T3, the next machine cycle begins.



Z80 memory write cycle

1. At the beginning of the memory write cycle, Z80 places the address on the address bus (ie. 2350H). At the falling edge of T1, MREQ\* goes low and the data from its register (ie. 9FH) is placed on the data bus.
2. After allowing one T-state (after MREQ\*) to stabilize the address, the Z80 asserts the signal write (WR\*), which is used to write the data byte at the address shown on the address bus.
3. After the falling edge of T3, both signals MREQ\* and WR\* become inactive.

### Arithmetic and Logic Unit (ALU)

This section, under the control of control section carries out the **actual processing of data**, normally describe as data manipulation. This consists largely of arithmetic operations (ADDition, SUBtraction, INCrementing, DECrementing etc) and logical operations (ANDing, ORing, XORing, NOTing etc).

The ALU carries out these operations in the following manner :-

1. stores data fetched from memory or I/O in the registers
2. fetches this data as needed from the registers and/or from relevant accumulators
3. send this data either to its arithmetic circuitry or logical circuitry, where necessary, where the necessary

arithmetic or logical operations are carried out

4. send results of its arithmetic or logical operation to relevant accumulator, to the memory, or to the I/O interfaces.

## Register Section

The register section/array consists completely of circuitry used to **temporarily store data** or **program codes** until they are sent to the ALU or to the control section or to memory. The number of registers are different for any particular CPU, and the more register a CPU have will result in easier programming tasks.

The registers in the Z80 microprocessor are of 2 types :-

### 1. General Purpose Registers

These registers are used to **store data temporarily**, either 8-bit data or 16-bit data according to their size. Registers are given names, normally an alphabet, such as A, B, C, D, E, H, L and each capable of storing an 8 bit data.

These registers can also work as a pair, such as BC, DE and HL and capable of storing 16 bit data.

For all 8 bit operations, register A is used as the **Accumulator**, where the result after an ALU operation will be stored automatically here. For 16 bit operations, register pair HL will be used to stored the result.

### 2. Special Purpose Registers

These registers have specific use to the human user and CPU. Normally they are used to keep the **address** or **status** for the use of human or CPU.

The types of registers are :- Flag Register (Register F), Index Register (IX, IY), Stack Pointer (SP) and Program Counter (PC).

Register F is an 8 bit register used to store the status of CPU, such as carry, zero, parity, overflow etc. The others are 16 bit registers which is used to store memory addresses.

The Index registers IX and IY are 16 bit wide registers, used to store 16 bit address, and is used in index addressing mode types of instructions. Registers IX and IY is normally used as pointers, as it points to memory locations, and normally used with a displacement d.

The Stack Pointer (PC) is a 16 bit register, used to store 16 bit address, and is used in special data transfer instructions (register to memory and memory to register transfer) by special instructions PUSH rr and POP rr, where rr are register pair.

The Program Counter (PC) is a 16 bit register, used to store the next address of the operation code to be fetched by the CPU. Not much use in programming, but as an indicator to user only.

## System Bus

The three components of the microcomputer system is connected by three busses, also known as system bus. These busses is used to transfer information (data) internally and externally to the microprocessor.

### 1. Address Bus



The address bus is '**unidirectional**', over which the microprocessor sends an address code to the memory or input/output. The size (width) of the address bus is specified by the number of bits it can handle.

The more bits there are in the address bus, the more memory locations a microprocessor can access. A 16 bit address bus is capable of addressing 65,536 (64K) addresses.

## 2. Data Bus

The data bus is '**bi-directional**', on which data or instruction codes are transferred into the microprocessor or on which the result of an operation or computation is sent out from the microprocessor to the memory or input/output.

Depending on the particular microprocessor, the data bus can handle 8 bit or 16 bit data.

## 3. Control Bus

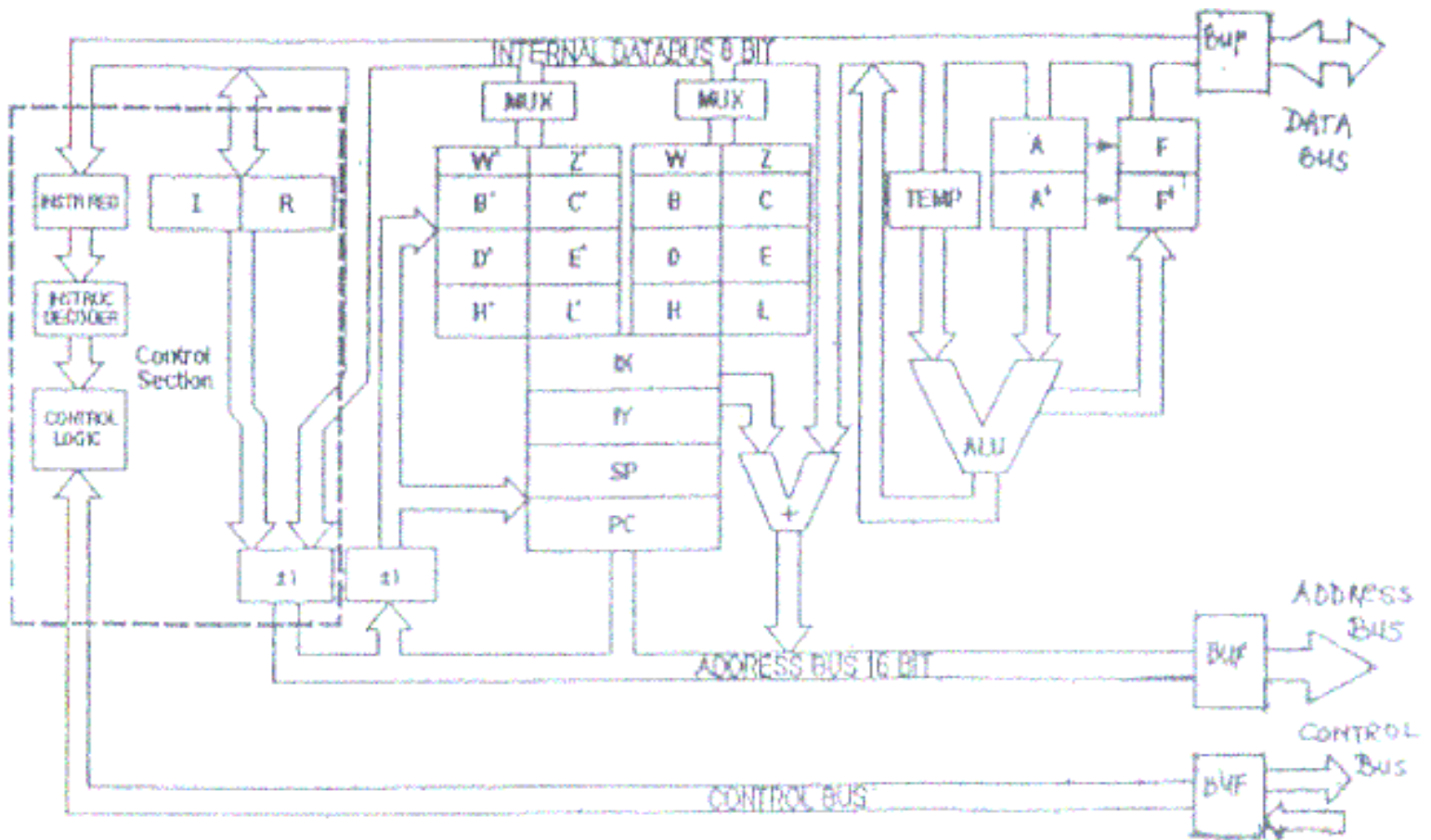
The control bus is used by the microprocessor to send out or receive timing and control signals in order to **co-ordinate and regulate** its operation and to communicate with other devices, i.e. memory or input/output.

The lines used to control memory and I/O devices are MEMRQ\*, IORQ\*, RD\* and WR\*. Others are general control signals to handle special external requests (interrupts), special I/O devices (DMA) and special kind of memory (DRAM).

## Z80 INTERNAL ARCHITECTURE

The internal architecture of Z80 shows a more detailed picture of what is inside the CPU. This processor has an instruction register (IR), an instruction decoder and control logic circuitry in its control unit; an ALU, and a register array (register set or programmer's model).

Registers W and Z are only used for internal operations in the processor, like jump to a new addresses. This is so since Z80 can only transfer 8-bits of data at a time, so to transfer a 16-bit address, it will first store it in reg. WZ.



The Internal Architecture of the Zilog Z80 Microprocessor

### Exercises :

1. Describe the operation of fetch, decode and execute cycle.
2. What is the purpose of the Instruction Register (IR)?
3. Draw the programming model of the Z80 microprocessor.
4. What are the differences between a general purpose with a special purpose register?
5. What registers that can be paired to form a 16 bit register?
6. What is the answer register?
7. List each of the flag bits and briefly describe their function.
8. Why is the Program Counter (PC) is called a counter?
9. What is the purpose of IX and IY registers?
10. The Stack Pointer (SP) uses the concept of 'Last-In First-Out' in storing and retrieving data from memory. Explain.

## PROGRAMMING AND ITS LANGUAGE



Instructing a computer to do a task is called **programming**. To program a computer, a programmer must feed into the computer a set of instructions that tell the computer **exactly** how to operate.

To overcome the problem of understanding binary codes, special 'programming languages' have been developed for the CPU that are easier for humans to understand, yet which can also be understood by the computer. This special programming language is called **assembly language programming**.

### An example of Z80-CPU instruction set

Here is a table of a portion of the instruction set for the Zilog Z80 microprocessor.

Machine Code	Mnemonic	T-states	Description
80	ADD A,B	4	$A \leftarrow A + B$
A0	AND A,B	4	$A \leftarrow A \wedge B$
3D	DEC A	4	$A \leftarrow A - 1$
3C	INC A	4	$A \leftarrow A + 1$
C2 nn	JP NZ addr	10/7	if Z=0 then $PC \leftarrow nn$
CA nn	JP Z addr	10/7	if Z=1 then $PC \leftarrow nn$
78	LD A,B	4	$A \leftarrow B$
79	LD A,C	4	$A \leftarrow C$
32 nn	LD (addr),A	13	$(nn) \leftarrow A$

\*flag reg. is omitted for simplicity and nn is a 16 bit address.

Machine code is also known as operation code or in short op-code.

T-states describes how long the CPU takes to executes the instruction.

If the CPU has a clock of 1MHz, then a T-state is 1  $\mu$ sec. Hence, the instruction ADD A,B requires 4  $\mu$ sec. to be completed.

### An example Assembly Language Programming

The program below shows how two hexadecimal numbers, 0AH and 47H are added together. The result of the addition is in register A (accumulator).

Mnemonics	Machine codes
LD A,0AH	3E 0A
LD B,47H	06 47

ADD A,B	80	
LD (2000H),A	32 00 20	; observe that the address nn is 2000H written 00 20
HALT	76	

Programs written normally will not use all of the available instructions or mnemonics. The rate of occurrence of the instructions is as follows :-

Data transfers (LD, PUSH, POP)	45%
Program control (JP, JR, CALL)	29%
Arithmetic (ADD, SUB)	10%
Compare (CP)	6%
Logical (AND, OR)	4%
Shift & rotate	3%
Bit manipulation (BIT, SET, RES)	2%
Input/output & control (IN, OUT)	1%

### PROGRAMMING BASICS

The programmer, after understanding a little bit about the microprocessor, most of the time during program writing, must understand clearly the working and functions of the registers in the CPU, which is also known as Programmer's Model.

Then slowly, learns about the instruction set, and how and where to use it in order to achieve the objective.

Alternate Registers	
Accumulator A'	Flags F'
B'	C'
D'	E'
H'	L'

Accumulator A	Flags F
B	C
D	E
H	L
Index Register IX	
Index Register IY	
Stack Pointer SP	
Program Counter PC	
Interrupt Vector I	Memory Refresh R

### The Z80 Programming Model



**Exercises :**

1. For the following instructions, determine the number of T-states for fetch, decode and execute cycle, hence calculate the total time taken to execute the instruction for clock of 1MHz.
  - a) LD A, B      b) LD BC, 7634H      c) ADD A,B      d) IN (A), 80H
2. List four good programming practices.

**THE Z80 INSTRUCTION SET**

The instruction set of a microprocessor determines the capability of its operations, the power of its data manipulation, and the ease of programming it. The Z80 microprocessor has **158** instruction types, in which give rise to **694** machine codes in total.

Each instruction has two parts :-

1. Task to be performed (operation code or op-code)
2. Data to operated upon (operand)

**ADDRESSING MODES**

The addressing mode of an instruction describe how the data or operand for that particular instruction is accessed. This is to ensure correct conversion from mnemonics to its machine codes.

1. *Immediate* – the byte immediately following the op-code is the operand. e.g. LD B,23H
2. *Immediate Extended* – two bytes immediately following the op-code is the operand. e.g. LD BC,4587H
3. *Register* – a byte is copied from one register into another register. e.g. LD A,B
4. *Implied* – the op-code itself implies one or more register as containing the operand. e.g. AND B, SUB C
5. *Index* – involves the use of Index register, and the byte following the op-code is a displacement value which is added to the Index register to form an address. e.g. LD (IX+05),C
6. *Relative* – the byte following the op-code is the displacement value for the jump. e.g. JP NZ, 2000H, JR C, 2000H
7. *Bit manipulation* – op-codes that contain the information of the position of the bit to be manipulated. e.g. BIT 7,A
8. *Page zero* – op-codes that causes the microprocessor to executes the program at certain determined address. e.g. RST 38H

**CLASSES OF INSTRUCTIONS**

There are many instructions available for a CPU, but basically they are classified into only the following types :-

1. Data Transfer Instructions

2. Data Manipulation Instructions
3. Decision Making Instructions
4. CPU Control Instruction

## **WRITING PROGRAMS**

Flow chart ==> Mnemonics ==> Machine codes ==> To memory for storage ==> Program execution

The results of the program are then examined to ensure that it is correct.

The requirements of a good program are :-

1. The program written must do the same thing every time it is run. There should be no need to reenter data or instructions to run the program again. It must perform the same function on the same data producing the same results every time.
2. The program must not modify itself. That is, the instructions cannot be changed by the program.
3. The program must work correctly for all reasonable data values.
4. The instructions and data numbers should be separated in memory.
5. The program should start at the first instruction and stop at the last instruction.

The goals of a programmer are :-

1. Write the fastest program.
2. Write an easily understood program.
3. Write an easily modified program.

Third Edition, November 2004