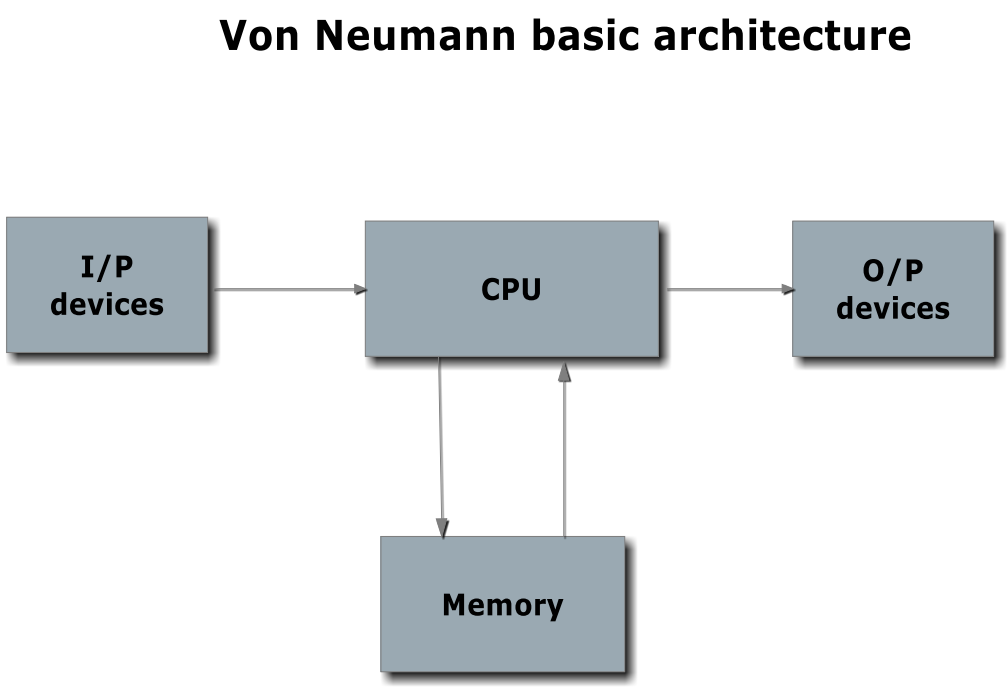
Unit-1, Module-5
Structural and Functional view of Computer

Question 1: Show the structural and functional view of a computer based on the Von Neumann Stored Programmed Principle.

Solution 1:
Von Neumann architecture is also referred to as the IAS computer and was developed at the Princeton Institute for Advanced Studies.

The basic structural and functional view of a computer according to Von Neumann stored program architecture is shown in the figure below:



The functionalities of the individual blocks are as follows

Central Processor Unit (CPU):

CPU consists of two basic blocks:

- The ***control unit*** has a set of registers and it generates control signals for the Arithmetic and Logic Unit (ALU). These signals control flow of data to (from) the input (output) ports of the ALU and initiate the required computation.
- The ***execution unit or data processing unit*** (i.e. ALU) contains a set of registers for storing data and hardware like adder, multiplier etc. for execution of arithmetic and logical operations.

In addition, CPU may have some additional registers for temporary storage of data.

Input Devices:

Input devices feed data from outside to the CPU. The data can be program or data which is to be processed. Input data is read into Memory unit from input device or under the control of CPU input instructions.

Example of input devices: Keyboard, Mouse, Hard disk, Floppy disk, CD-ROM drive etc.

Output Device:

With the help of output device computational results of the CPU can be provided to the user. Output data from main storage go to output device under the control of CPU output instructions.

Example of output devices: Printer, Monitor, Plotter, Hard Disk, Floppy Disk, CD ROM, Hard disk etc.

Memory Unit:

Memory unit is used to store the data and program. CPU can work with the information stored in memory unit.

Example of memory devices: Hard Disk, Floppy Disk, Magnetic Tape, CD-ROM, Pen drive etc.

Question 2: Explain the basic purposes of the special purpose registers that are available in the ALU proposed in the ISA architecture.

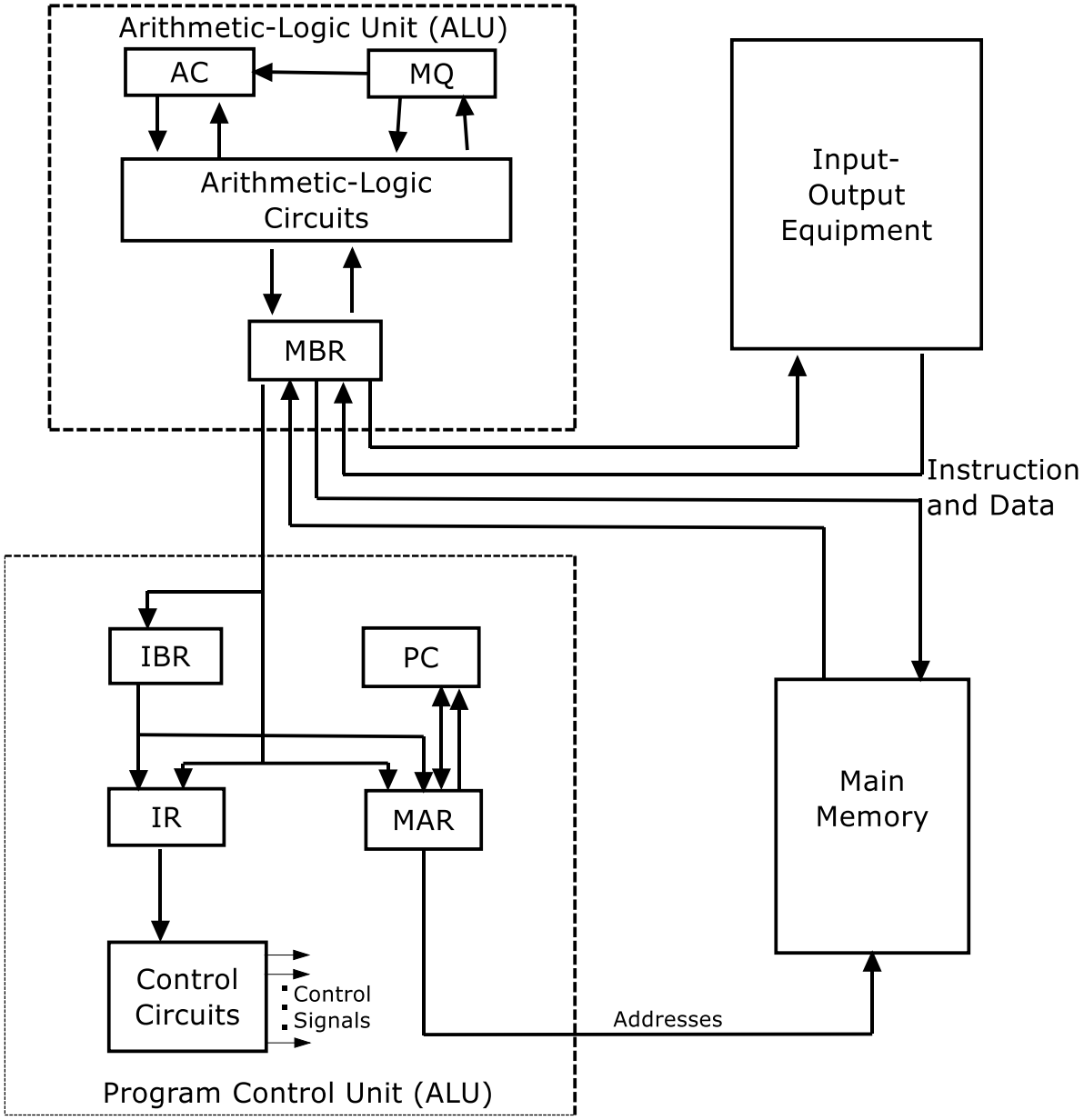
Solution 2: Control unit and the ALU contain storage locations, called registers, which can be classified as special purpose and general purpose. The general purpose registers are used for temporary storage by a programmer. The special purpose registers are the following:

- Memory buffer register (MBR).
- Memory address register (MAR).
- Instruction register (IR).
- Instruction buffer register (IBR).
- Program counter (PC).
- Accumulator (AC) and multiplier quotient (MQ)

The usage of these registers are discussed below

- Memory buffer register (MBR) contains the data that is to be stored in memory (from I/O unit) or to be read from the memory (and sent to I/O unit). Its size is 40 bits.
- Memory address register (MAR) contains the address of memory location that is to be written from or read into the MBR. Its size is 12 bits.
- Instruction register (IR) contains the instruction that is being executed. Its size is 20 bits to accommodate one instruction (whose size is also 20 bits) .
- Instruction buffer register (IBR) holds temporarily the instruction to be executed next. In ISA architecture, instruction comprises 8-bit operation code and 12-bit address. Also, the basic unit of information is 40-bit. So that two instructions could be fetched together and stored in each 40-bit memory location IBR. Hence the IBR is used to temporarily hold Right hand instruction for the next use.
- Program counter (PC) contains the address of the next instruction to be fetched from the memory. Its size is 12 bits.
- Accumulator (AC) and multiplier quotient (MQ) are used for temporary storage of operands and results of ALU. Sizes of both these registers are 40 bits.

The figure below illustrates the expanded structure of a Von Neumann computer with the registers.



Question 3: What is a BUS? How the bus is used to connect different components of a computer.

Solution 3: Broadly speaking, Bus is a wire that connects two electrical ports so that signals can be transferred between them. The term Bus has been derived from the Latin word *omnibus*, meaning "for all". In terms of computer architecture, a bus is a set of electrical connections (cables, connections in printed circuit board or fabricated on-chip etc.) which can be shared by multiple hardware modules in order to communicate with one another. One key difference between the term “wire” and “Bus” is that, more than one parallel wire can be present in a single Bus between the points it connects.

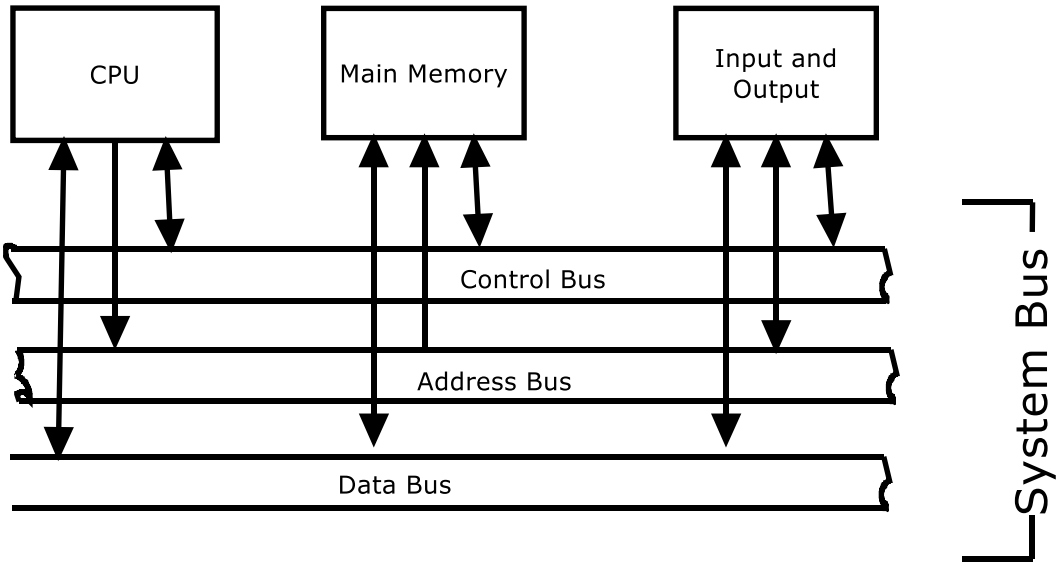
In a CPU is the following are the main three Buses

Address Bus: Sends the address of the memory location (to be read/written) to the memory from CPU. It is a unidirectional bus.

Data Bus: Sends/receives data in between the CPU, Memory and I/O devices. It is a bidirectional bus. For example, if a memory location is to be read then data bus sends the data from the respective memory location (that is given in address bus) to the CPU.

Control Bus: The control bus is used to control and co-ordination the data transfers between the hardware modules interconnected by the address and data bus. It is also a bidirectional bus. For example, whether a memory location will be read or written is determined by a control line.

The broad architecture showing the interconnections of CPU, memory and I/O devices using the three Buses is shown below.



Question 4: What are the basic phases of executing an instruction of a program in a Von Neumann computer?

Solution 4:

Broadly speaking, executing an instruction passes through the following three phases.

1. **Instruction Fetch**
   In Von Neumann computer both the instructions and data are stored in the memory. So, in the first phase the instruction to be executed is read from the memory to CPU.

2. **Instruction Decoding and Fetching the Operands:**
   Next, the instruction that is fetched is analyzed to determine the type of operation to be performed and operand(s) to be used. The type of operation is encoded in the OP code part of the instruction. However, the operands are to be fetched depending on the type of instruction; it involves the following sub-steps:

   - **Operand Address Calculation**
     If the operation involves reference to an operand in memory or available via I/O, then address of the operand is extracted from the instruction. Sometimes the operand is available in the instruction itself (called immediate address). In such a case operands need not be fetched explicitly.
   - **Operand Fetch:**
     Once the address in known, the operand from memory or I/O is read into the CPU.

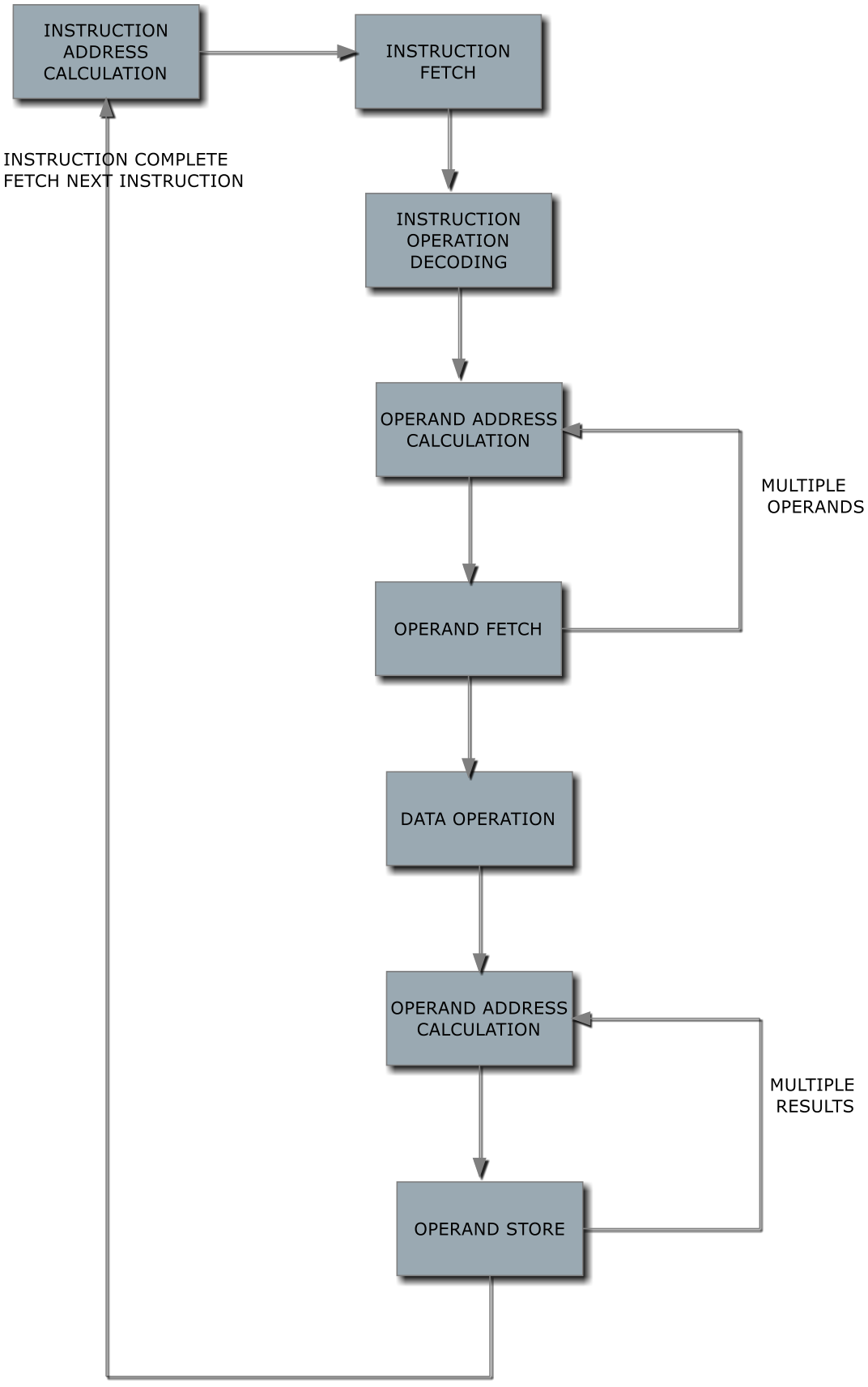
3. **Instruction Execution:**
   In this phase, operation is performed on the operands. The results of the execution can be classified as thee basic types

   - **Data Transfer:**
     Write the result into memory or to an I/O device. In this case also we need to obtain the address where the data has to be transferred.

   - **Arithmetic and Logic:**
     Examples are Addition, Subtraction. Complement etc.

   - **Control:**
     Examples are Jump to some memory location depending on the result of the computation.

The figure given below illustrates the flow of phases involved in executing an instruction.


INSTRUCTION ADDRESS CALCULATION
INSTRUCTION FETCH
INSTRUCTION COMPLETE
FETCH NEXT INSTRUCTION
INSTRUCTION OPERATION DECODING
OPERAND ADDRESS CALCULATION
MULTIPLE OPERANDS
OPERAND FETCH
DATA OPERATION
OPERAND ADDRESS CALCULATION
MULTIPLE RESULTS
OPERAND STORE