Computer Organization and Software Systems

CONTACT SESSION 1

Mr. Vaibhav Jain

WILP – BITS Pilani

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Computer Organization and Software Systems (SS ZG516)

Books for reference

- Stallings William, Computer Organization & Architecture, Pearson Education, 9th Ed., 2013.

- Computer Organization & Design, Hennenssy & D.A. Patterson, Morgan Kaufmann ,4th Ed.2009.

- Silberschatz Abraham and others, Operating Systems Concepts, Wiley Student, 8th Edition

Computer Organization and Software Systems (SS ZG516)

Computer System Structure

- **Hardware**

  Provide basic computing resource .e.g .CPU, memory, I/O

- **Operating system**

  -Controls and coordinates the use of hardware among various applications and users

  -OS is a program that acts as a an interface(intermediary) between user of a computer(application programs) and computer hardware

- **Application program**

  Define the ways in which system resources are used to solve computing problems of user.

- **Users**

  People ,machine ,other computers

Computer Organization and Software Systems (SS ZG516)

COA Introduction

- To understand computer system's functional components, their characteristics, their performance, and their interactions with each other.
- Covers a broad range of design principles and implementation issues and concepts.
- To understand how various peripheral devices interact with and how they are interfaced to a CPU.
- To understand the trade off among various components e.g CPU clock speed vs. Memory size.
- Factor responsible for the great increase in processor speed
  - Pipelining and parallel execution techniques

Computer Organization and Software Systems (SS ZG516)

Session _1

- Organization and Architecture
- Structure and Functions of a computer system
- Top-Level View of Computer Function
- Basic Instruction cycle state diagram
- Interrupts
- Instruction cycle state diagram with interrupts

Architecture & Organization

- Architecture refers to those attributes visible to the programmer.
- Examples of architectural attributes include
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
- Organization refers to operational units and their interconnections that realize the architectural specifications.
  - Control signals, interfaces, memory technology.

Architecture & Organization

- The architecture was first introduced in 1970 and included a number of models.
- The customer with modest requirements could buy a cheaper, slower model and, if demand increased, later upgrade to a more expensive, faster model without having to abandon software that had already been developed.
- Over the years, IBM has introduced many new models with improved technology to replace older models, offering the customer greater speed, lower cost, or both.

Architecture & Organization

- These newer models retained the same architecture so that the customer's software investment was protected.
- All Intel x86 family share the same basic architecture
- The IBM System/370 family share the same basic architecture.

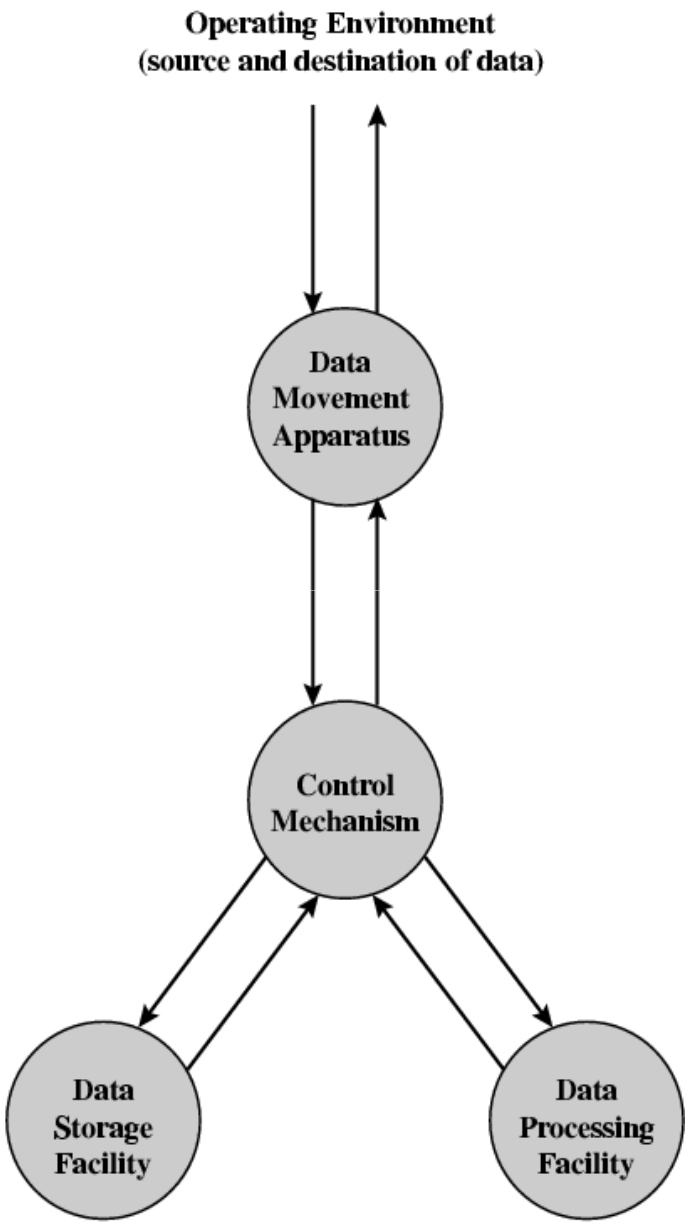
Structure & Function of a Computer System

- Structure is the way in which components relate to each other.

- Function is the operation of individual components as part of the structure.

Function

- All computer functions are:
  - Data processing
  - Data storage
    - if the computer is processing data on the fly (i.e., data come in and get processed, and the results go out immediately), the computer must temporarily store at least those pieces of data that are being worked on at any given moment. Thus, there is at least a short-term data storage function.
  - Data movement
  - Control



Function

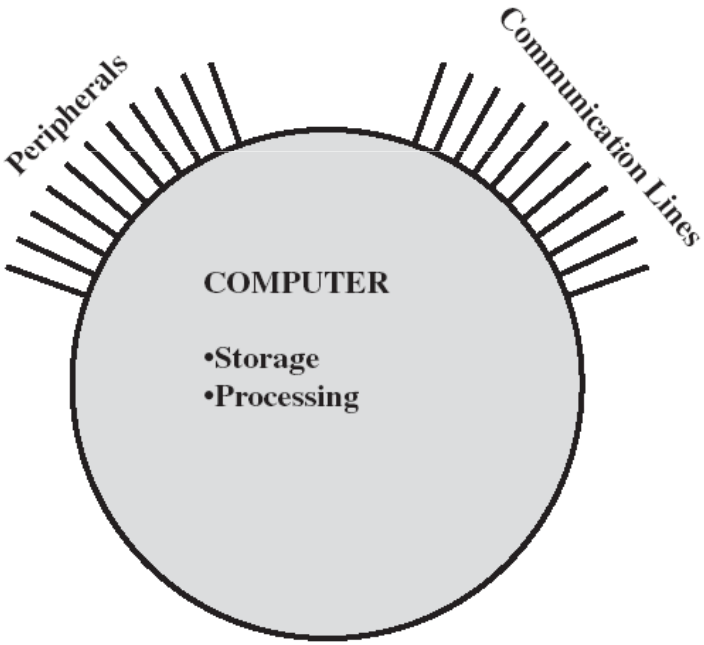
- Data movement
  - The computer must be able to move data between itself and the outside world.
  - When data are received from or delivered to a device that is directly connected to the computer, the process is known as input–output (I/O), and the device is referred to as a peripheral.
  - When data are moved over longer distances, to or from a remote device, the process is known as *data communications*.

Function

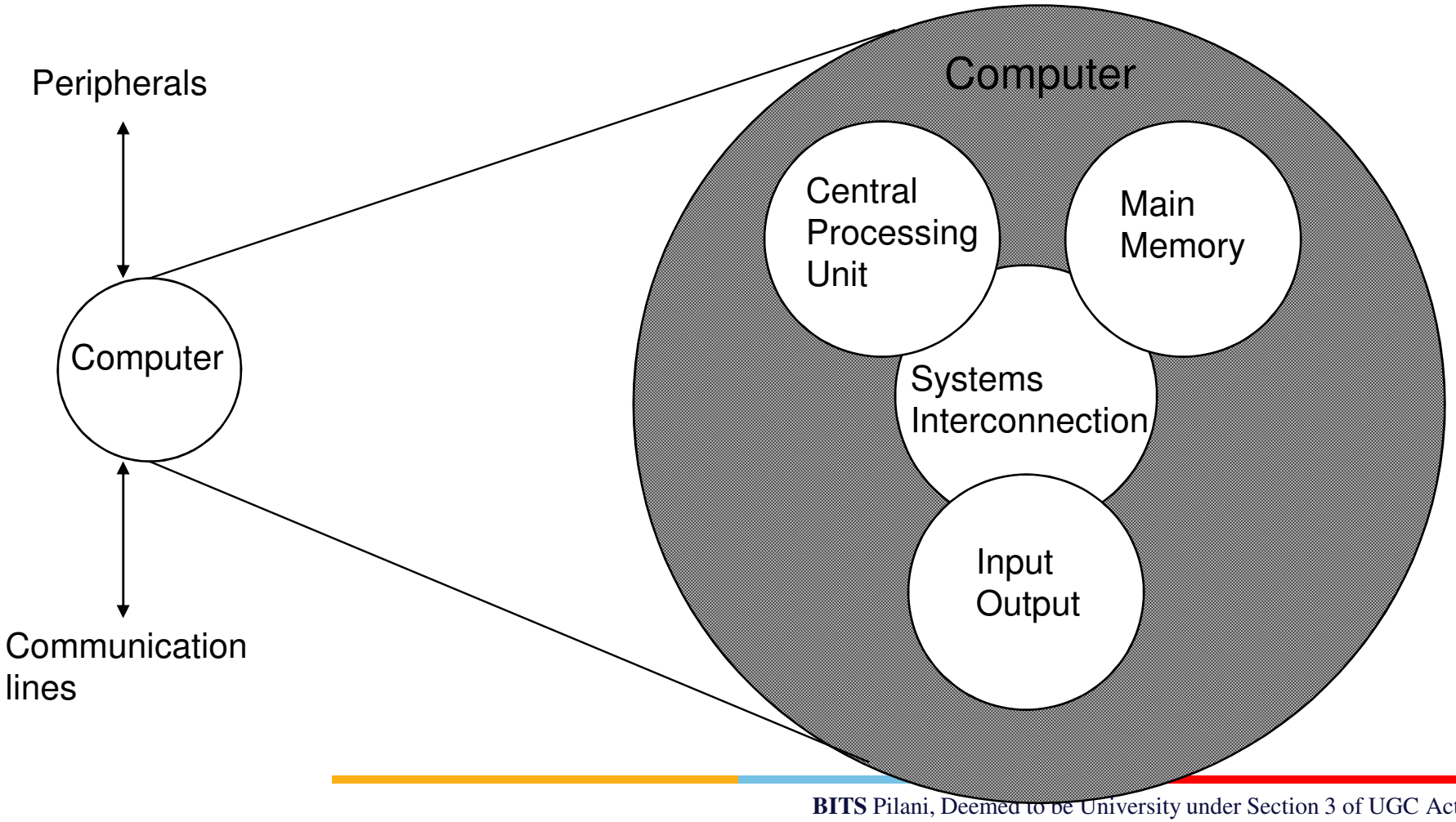
- Control
  - Finally, there must be control of these three functions.
  - Ultimately, this control is exercised by the individual(s) who provides the computer with instructions.
  - Within the computer, a control unit manages all the performance of its functional parts in response to those instructions.

Structure

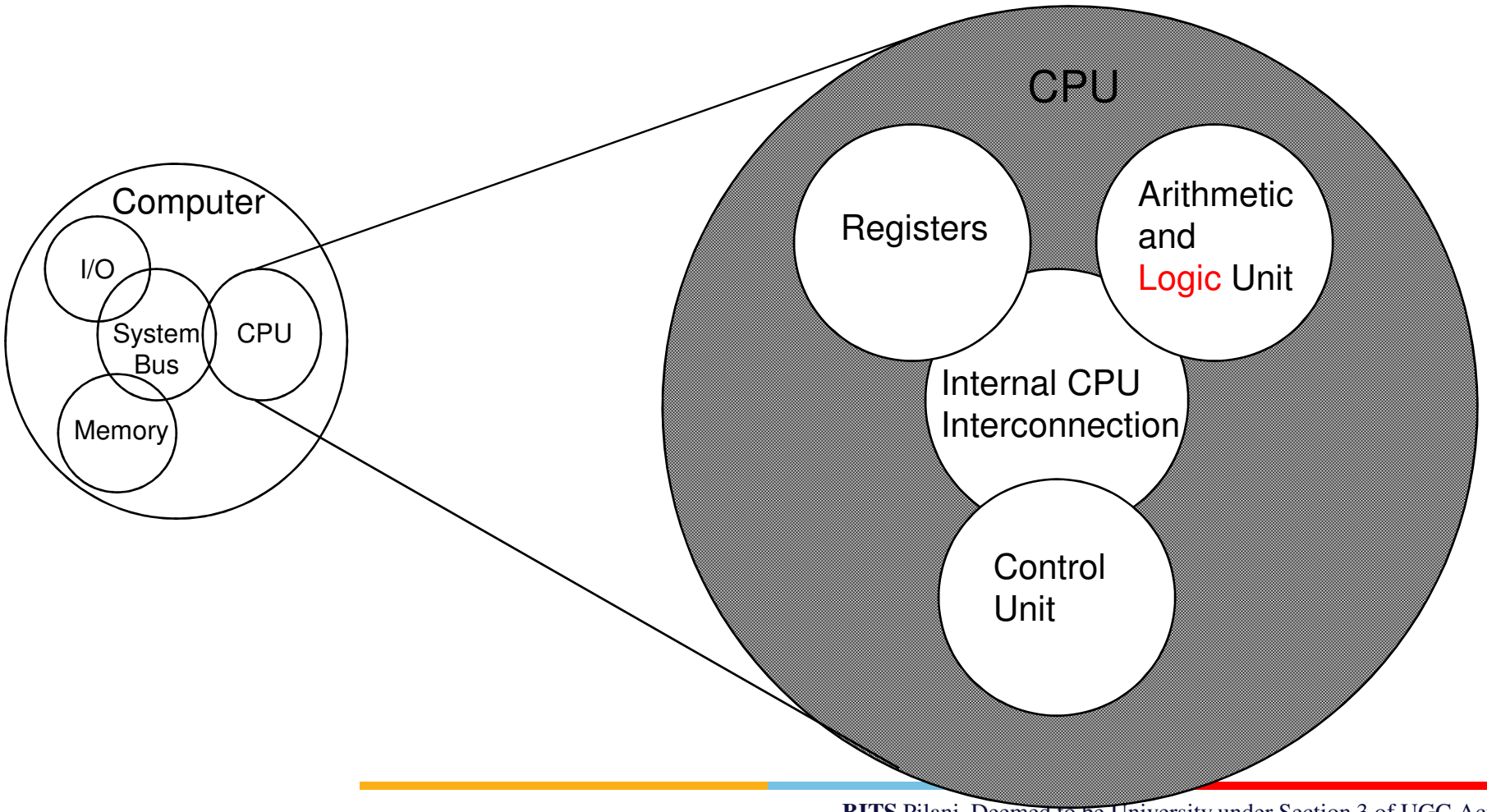
- The Computer
  - CPU (also known as processor)
    - Controls the operation of the computer and performs its data processing functions.
  - Main memory
    - Stores data
  - I/O
    - Moves data between the computer and its external environment
  - System interconnection
    - Provides for communication among CPU, main memory, and I/O. (System Bus as e.g.)



Structure - Top Level



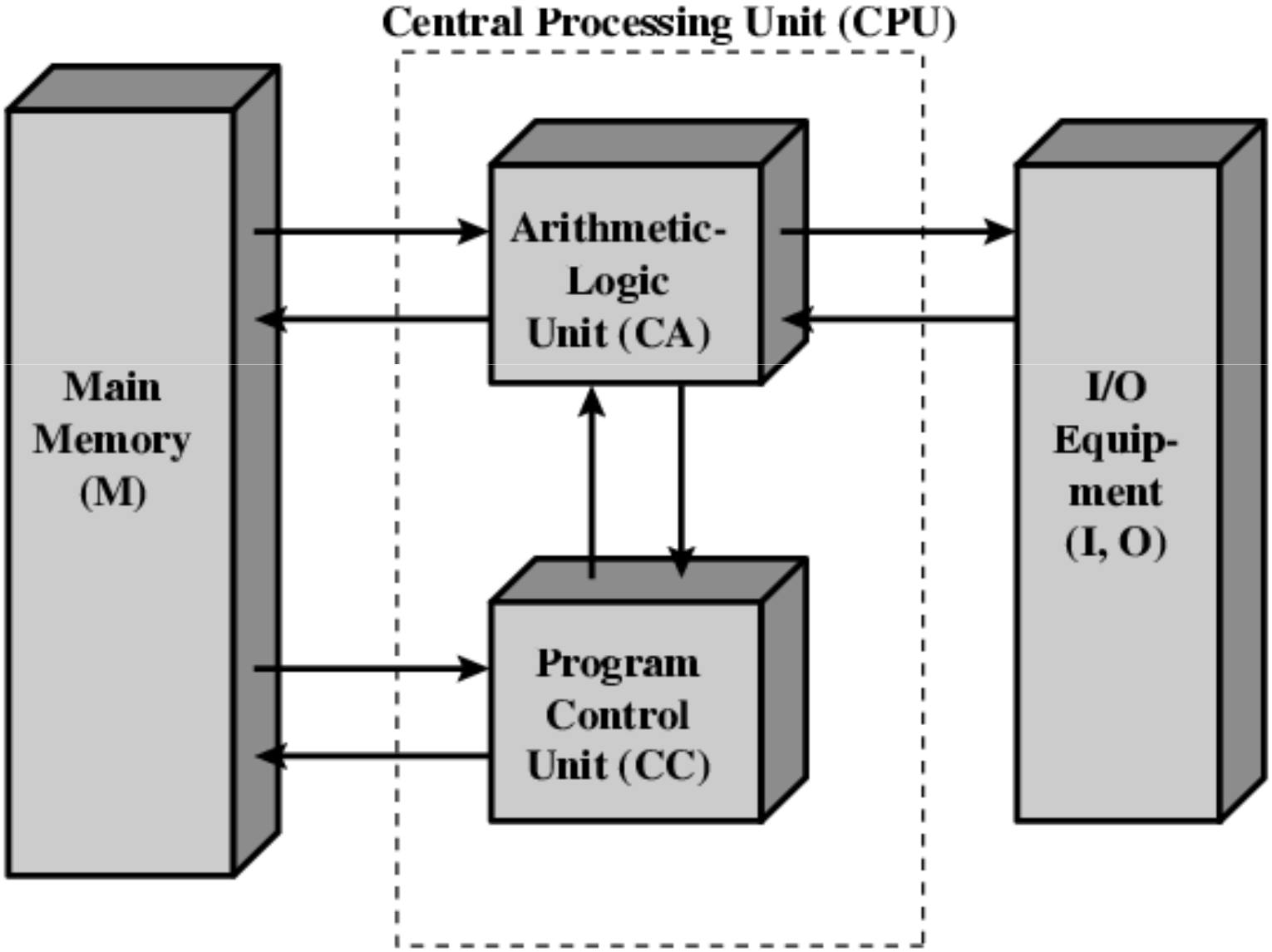
Structure - The CPU



Structure - The CPU

- Central Processing Unit (CPU)
  - Control Unit
    - Controls the operation of the CPU and hence the computer.
  - Arithmetic and logic unit (ALU)
    - Performs the computer's data processing functions
  - Registers
    - Provides internal storage for different operations.
  - CPU interconnection
    - Some mechanism that provides for communication among the control unit, ALU, and registers

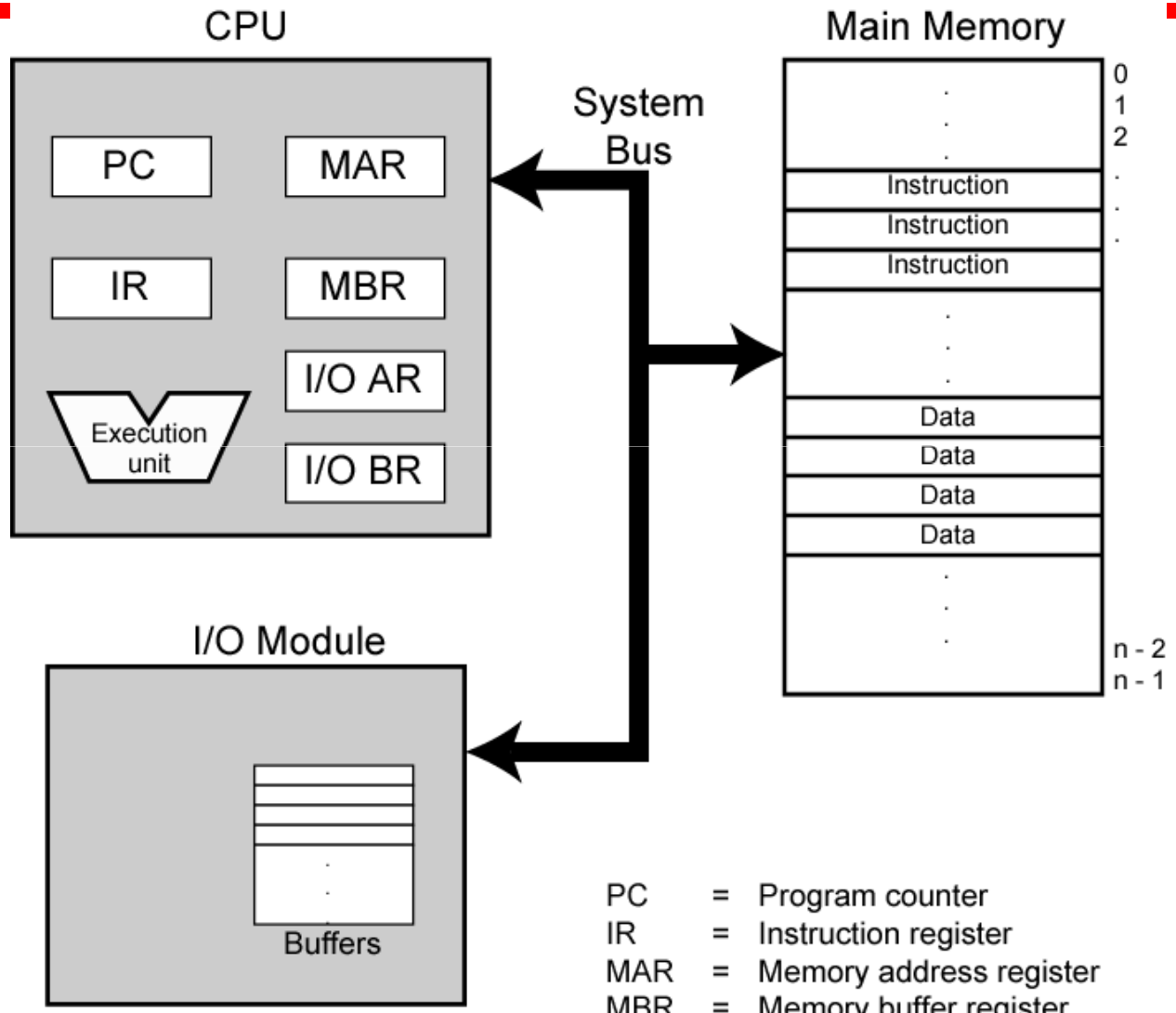
Structure of von Neumann machine (IAS Computer)



Structure of von Neumann machine (IAS Computer)

- Data and instructions are stored in a single read–write memory.
- The contents of this memory are addressable by location.
- Execution occurs in a sequential fashion from one instruction to the next.

Computer Components: Top Level View



PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Computer Components

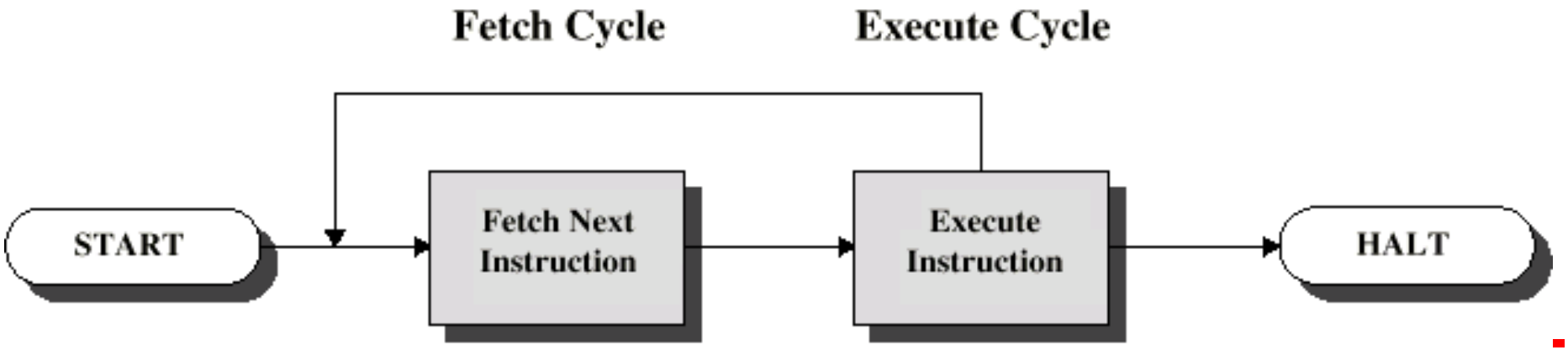
- The Control Unit and the Arithmetic and Logic Unit constitute the Central Processing Unit
- Data and instructions need to get into the system and results out
  - Input/output
- Temporary storage of code and results is needed
  - Main memory

Computer Components

- Here, The CPU exchanges data with memory or I/O Devices. Thus uses the following registers:
- Memory address register (MAR), which specifies the address in memory for the next read or write
- Memory buffer register (MBR), which contains the data to be written into memory or receives the data read from memory.
- I/O address register (I/OAR) specifies address of a particular I/O device.
- I/O buffer (I/OBR) used for exchange of data between an I/O module and the CPU.

Computer Function

- The basic function performed by a computer is execution of a program, which consists of a set of instructions stored in memory.
- The processing required for a single instruction is called an instruction cycle.
- Two steps:- 1) Fetch 2) Execute



IAS Computer – Instruction Cycle

- The IAS operates by repetitively performing an instruction cycle.

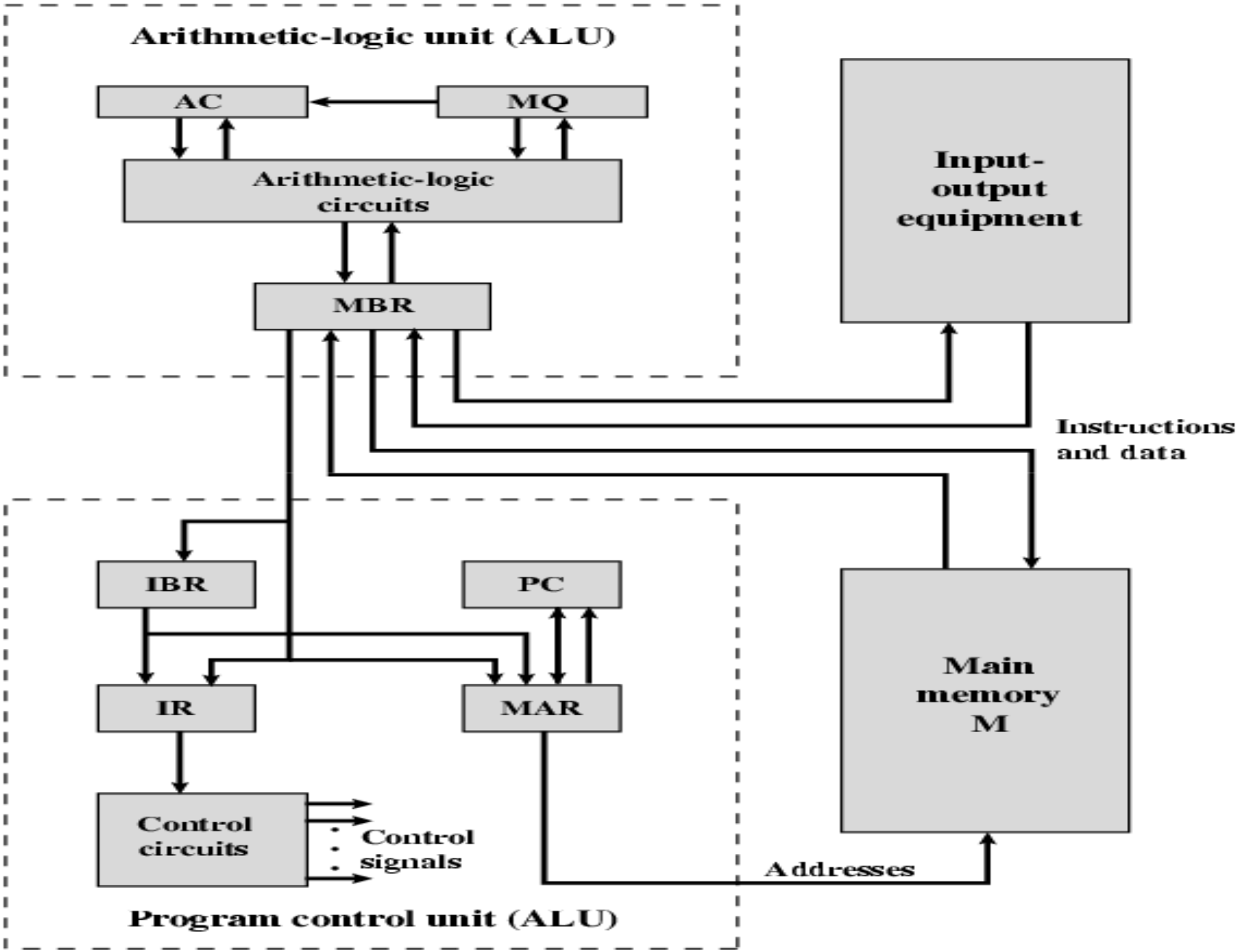
- Each instruction cycle consists of two subcycles.
  - Fetch Cycle
  - Execute Cycle

IAS Computer – Fetch Cycle

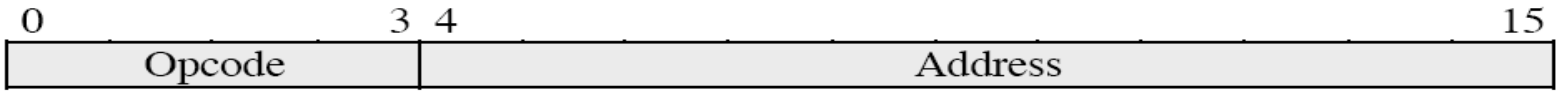
- During the fetch cycle,
  - The opcode of the next instruction is loaded into the IR and the address portion is loaded into the MAR.
  - This instruction may be taken from the IBR, or it can be obtained from memory by loading a word into the MBR, and then down to the IBR, IR, and MAR.

IAS Computer – Execute Cycle

- During the Execute cycle,
  - Once the opcode is in the IR, the execute cycle is performed.
  - Control circuitry interprets the opcode and executes the instruction by sending out the appropriate control signals to cause data to be moved or an operation to be performed by the ALU.


Arithmetic-logic unit (ALU)
AC
MQ
Arithmetic-logic circuits
MBR
Input-output equipment
Instructions and data
IBR
PC
IR
MAR
Main memory M
Control circuits
Control signals
Addresses
Program control unit (ALU)

Hypothetical Machine



(a) Instruction format

(b) Integer format

Program Counter (PC) = Address of instruction
Instruction Register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory
0010 = Store AC to Memory
0101 = Add to AC from Memory

(d) Partial list of opcodes

Figure 3.4 Characteristics of a Hypothetical Machine

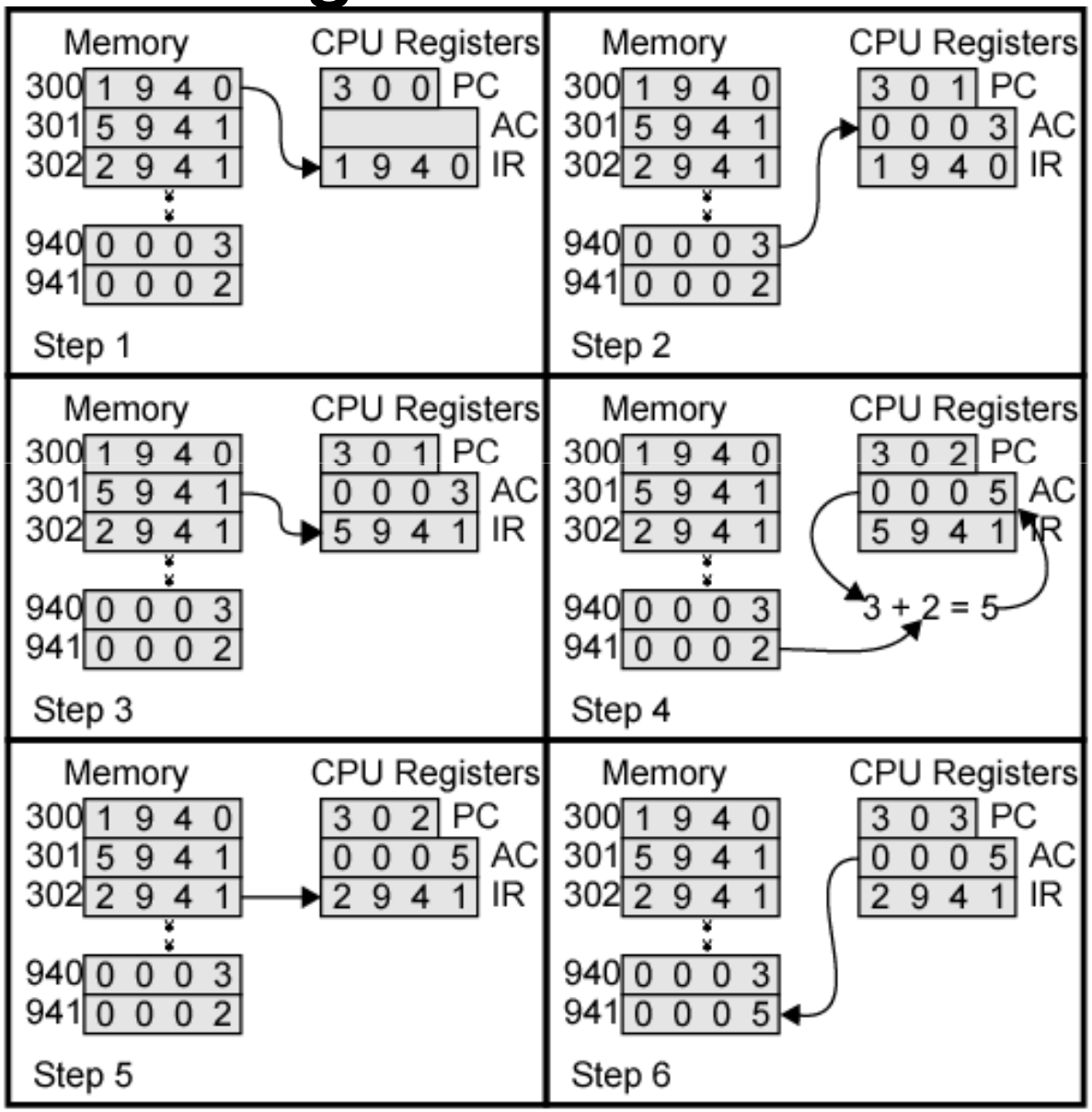
Hypothetical Machine

- ✓ Here, The processor contains a single data register, called an accumulator (AC).
- ✓ Both instructions and data are 16 bits long.
- ✓ The instruction format provides 4 bits for the opcode, so that there can be as many as 2^4 = 16 different opcodes.
- ✓ Up to 2^12 = 4096 (4K) words of memory can be directly addressed.

Hypothetical Machine – Program Example

- Consider an example –
- Add the contents of the memory word at address 940 to the contents of the memory word at address 941 and stores the result in the latter location.

Example of Program Execution



Hypothetical Machine – Program Example

- ✓ The PC contains 300, the address of the first instruction. This instruction (the value 1940 in hexadecimal) is loaded into the instruction register IR and the PC is incremented.
- ✓ The first 4 bits (first hexadecimal digit) in the IR indicate that the AC is to be loaded. The remaining 12 bits (three hexadecimal digits) specify the address (940) from which data are to be loaded.
- ✓ The next instruction (5941) is fetched from location 301 and the PC is incremented.

Hypothetical Machine – Program Example

- ✓ The old contents of the AC and the contents of location 941 are added and the result is stored in the AC.
- ✓ The next instruction (2941) is fetched from location 302 and the PC is incremented.
- ✓ The contents of the AC are stored in location 941.

Hypothetical Machine – Question -

- The hypothetical machine has two I/O instructions:

  0011 Load AC from I/O

  0101 Add to AC from memory

  0111 Store AC to I/O

  In these cases, the 12-bit address identifies a particular I/O device. Show the program execution for the following program:

  1. Load AC from device 5.
  2. Add contents of memory location 940.
  3. Store AC to device 6.

  Assume that the value retrieved from device 5 is 3 and that location 940 contains a value of 2.

Hypothetical Machine – Question -

- Memory (contents in hex):

  300: 3005; 301: 5940; 302: 7006

  Step 1: 3005 → IR;

  Step 2: 3 → AC

  Step 3: 5940 → IR;

  Step 4: 3 + 2 = 5 → AC

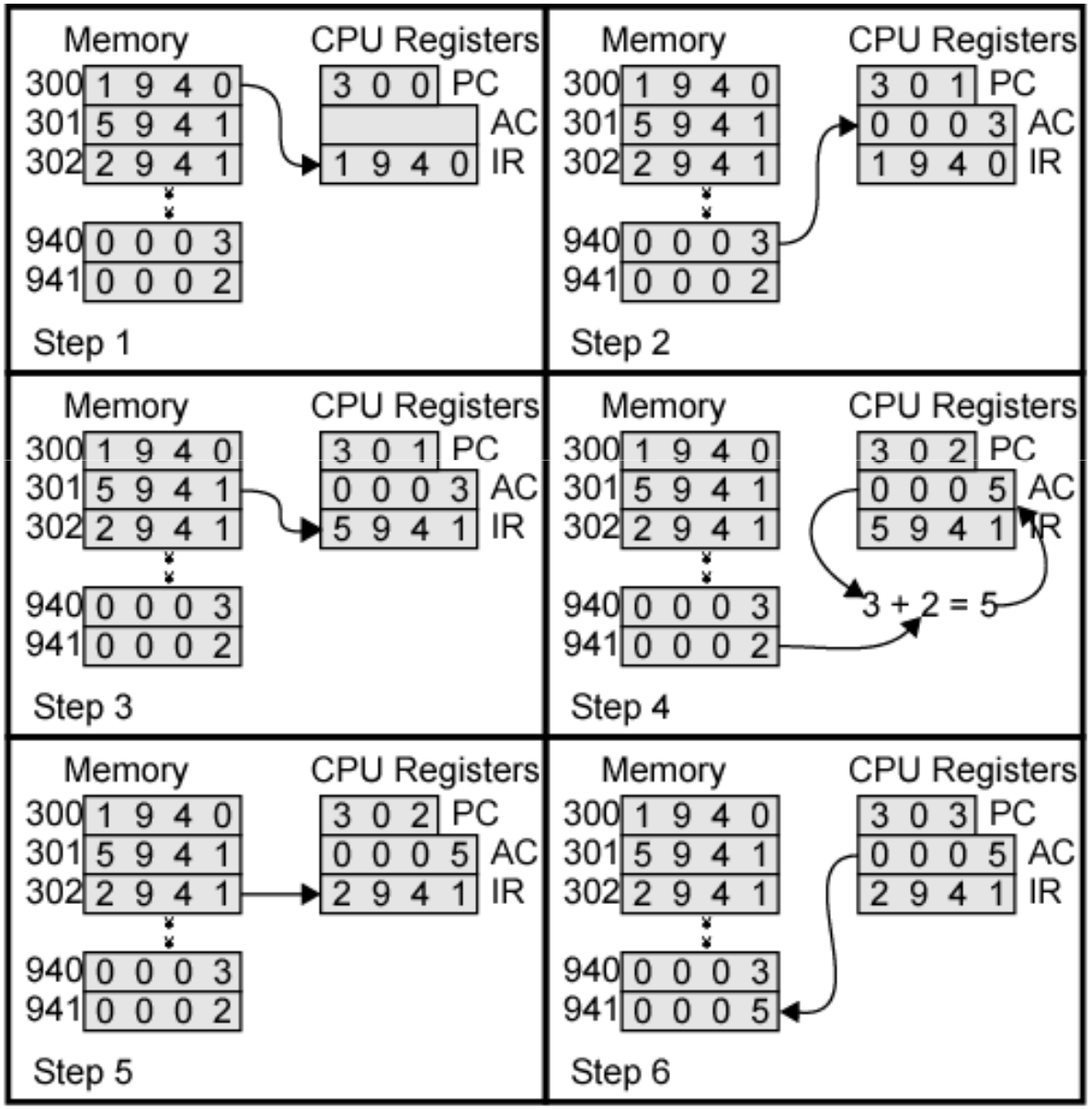
  Step 5: 7006 → IR;

  Step 6: AC → Device 6

Example of Program Execution - Question

innovate achieve lead

Can you Expand this description to show the use of the MAR and MBR?



Answer:

1. a. The PC contains 300, the address of the first instruction. This value is loaded in to the MAR.

   b. The value in location 300 (which is the instruction with the value 1940 in hexadecimal) is loaded into the MBR, and the PC is incremented. These two steps can be done in parallel.

   c. The value in the MBR is loaded into the IR.

2. a. The address portion of the IR (940) is loaded into the MAR.

   b. The value in location 940 is loaded into the MBR.

   c. The value in the MBR is loaded into the AC.

Answer:

3. a. The value in the PC (301) is loaded in to the MAR.

b. The value in location 301 (which is the instruction with the value 5941) is loaded into the MBR, and the PC is incremented.

c. The value in the MBR is loaded into the IR.

4. a. The address portion of the IR (941) is loaded into the MAR.

b. The value in location 941 is loaded into the MBR.

c. The old value of the AC and the value of location MBR are added and the result is stored in the AC.

Answer:

5. a. The value in the PC (302) is loaded in to the MAR.

   b. The value in location 302 (which is the instruction with the value 2941) is loaded into the MBR, and the PC is incremented.

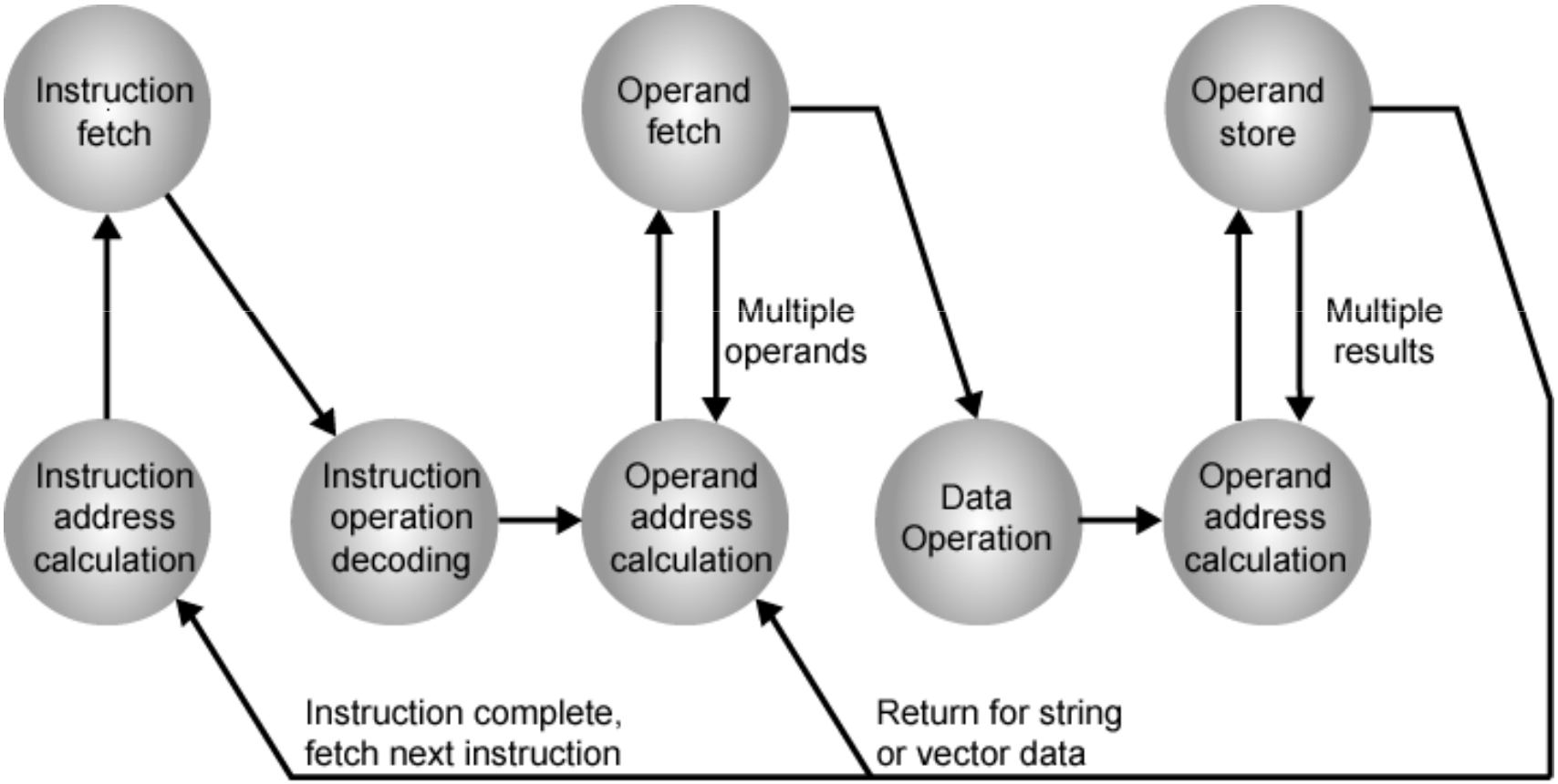
   c. The value in the MBR is loaded into the IR.

6. a. The address portion of the IR (941) is loaded into the MAR.

   b. The value in the AC is loaded into the MBR.

   c. The value in the MBR is stored in location 941.

Instruction Cycle State Diagram



Instruction Cycle State Diagram

- **Instruction address calculation (iac):**
  - Determine the address of the next instruction to be executed. Usually, this involves adding a fixed number to the address of the previous instruction. For example, if each instruction is 16 bits long and memory is organized into 16-bit words, then add 1 to the previous address. If, instead, memory is organized as individually addressable 8-bit bytes, then add 2 to the previous address.

- **Instruction fetch (if):**
  - Read instruction from its memory location into the processor.

Instruction Cycle State Diagram

- **Instruction operation decoding (iod):**
  - Analyze instruction to determine type of operation to be performed and operand(s) to be used.
- **Operand address calculation (oac):**
  - If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.
- **Operand fetch (of):**
  - Fetch the operand from memory or read it in from I/O.
- **Data operation (do):**
  - Perform the operation indicated in the instruction.
- **Operand store (os):**
  - Write the result into memory or out to I/O.

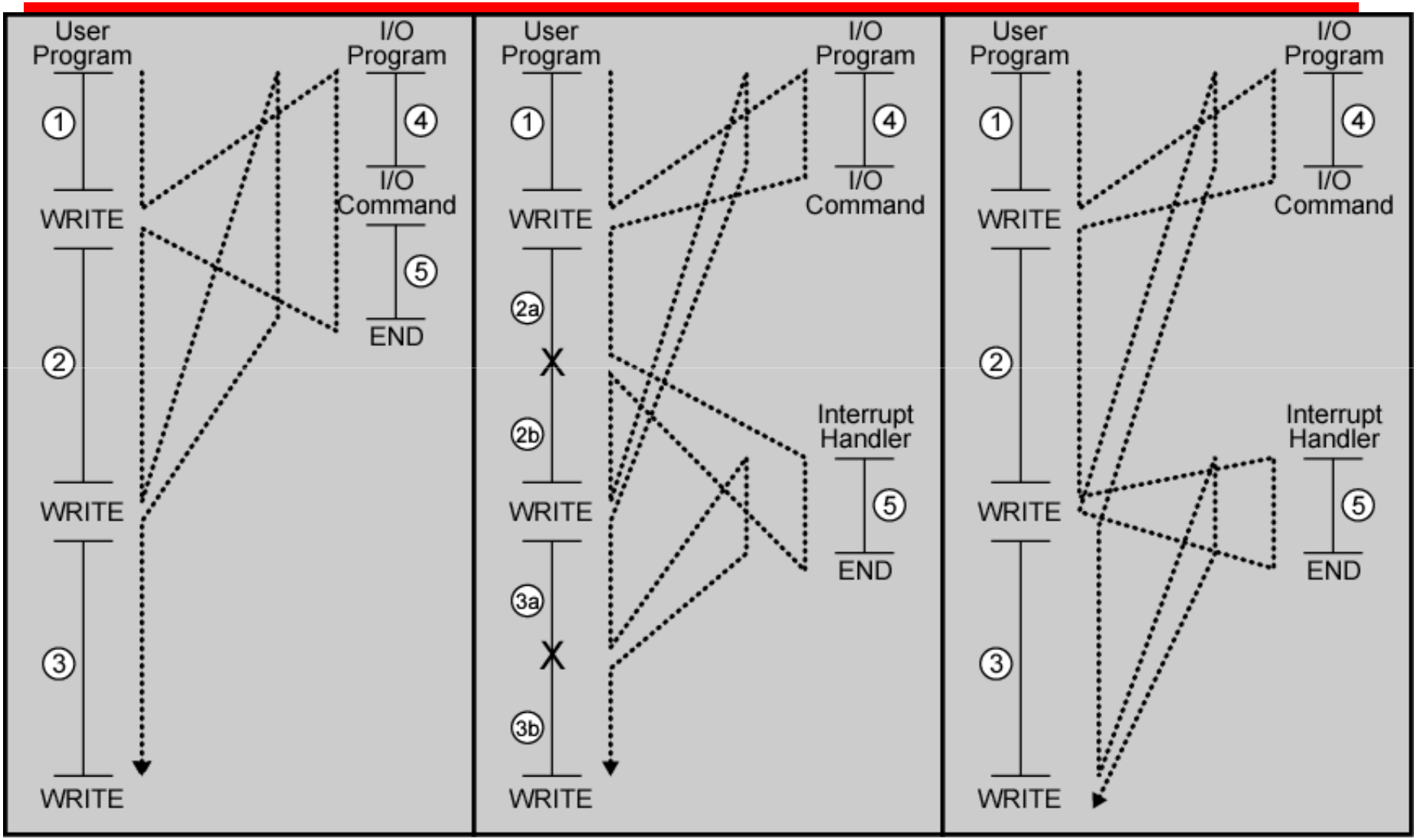
Interrupts

- Mechanism by which other modules (e.g. I/O, memory) may interrupt normal sequence of processing. Most Common interrupts-
- Program
  - e.g. overflow, division by zero
- Timer
  - Generated by internal processor timer
- I/O
  - from I/O controller
- Hardware failure
  - e.g. memory parity error, power failure

Interrupts – Improves Processing

- Primarily improve processing efficiency.
- For example, most external devices are much slower than the processor.
- Suppose that the processor is transferring data to a printer.
- After each write operation, the processor must pause and remain idle until the printer catches up.
- Clearly, this is a very wasteful use of the processor.

Program Flow Control



(a) No interrupts (b) Interrupts; short I/O wait (c) Interrupts; long I/O wait

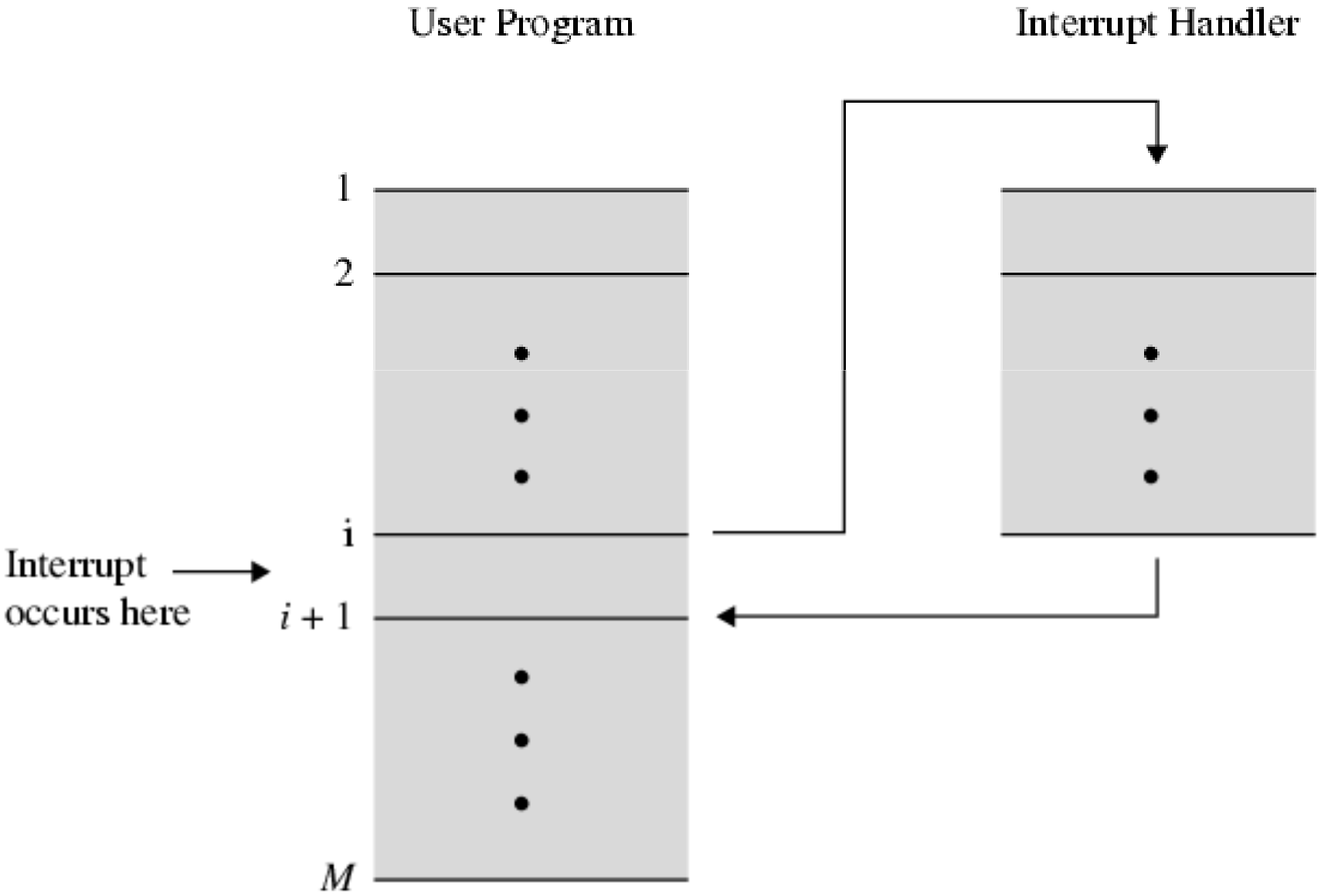
Program Flow Control

- Code segments 1, 2, and 3 refer to sequences of instructions that do not involve I/O.
- The WRITE calls are to an I/O program that is a system utility and that will perform the actual I/O operation.
- The I/O program consists of three sections:
  - A sequence of instructions, labeled 4 in the figure, to prepare for the actual I/O operation.
  - The actual I/O command. Without the use of interrupts, program might wait by simply repeatedly performing a test operation to determine if the I/O operation is done.
  - A sequence of instructions, labeled 5 in the figure, to complete the operation. This may include setting a flag indicating the success or failure of the operation.

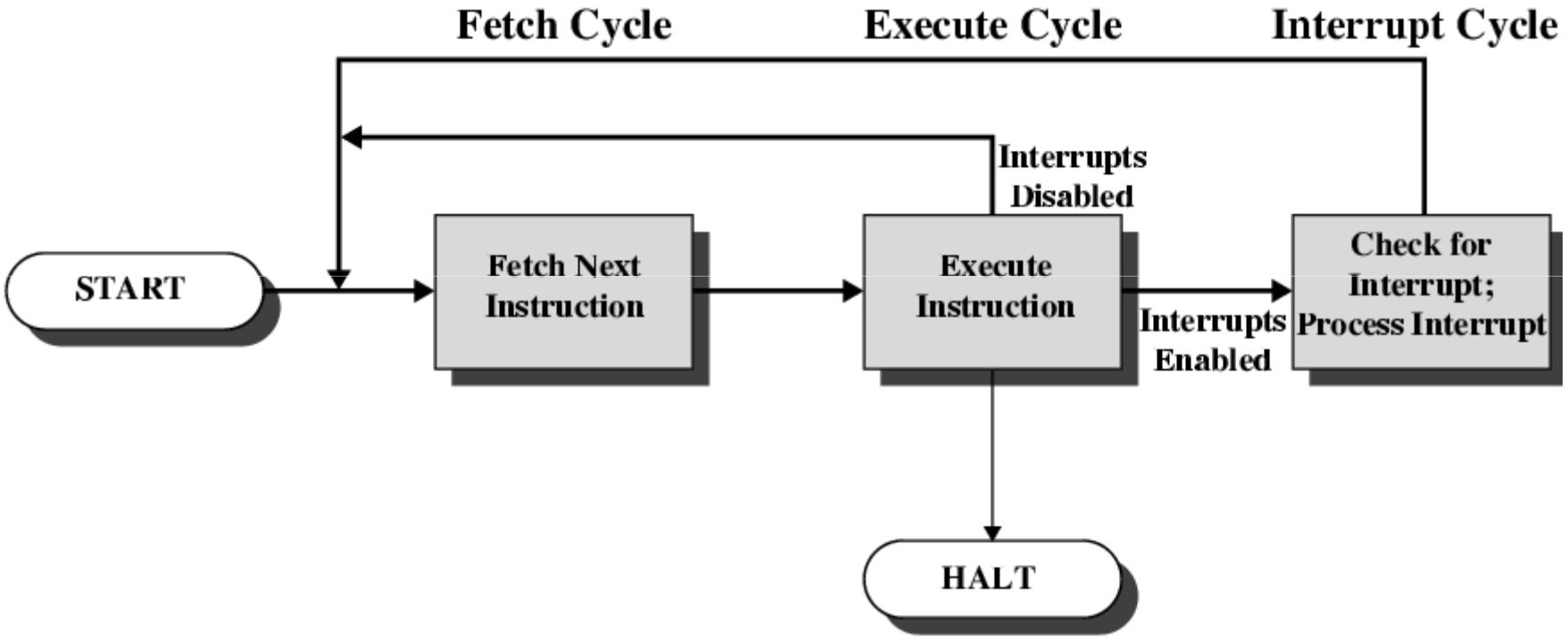
Interrupt Cycle

- Added to instruction cycle
- Processor checks for interrupt
  - Indicated by an interrupt signal
- If no interrupt, fetch next instruction
- If interrupt pending:
  - Suspend execution of current program
  - Save context
  - Set PC to start address of interrupt handler routine
  - Process interrupt
  - Restore context and continue interrupted program

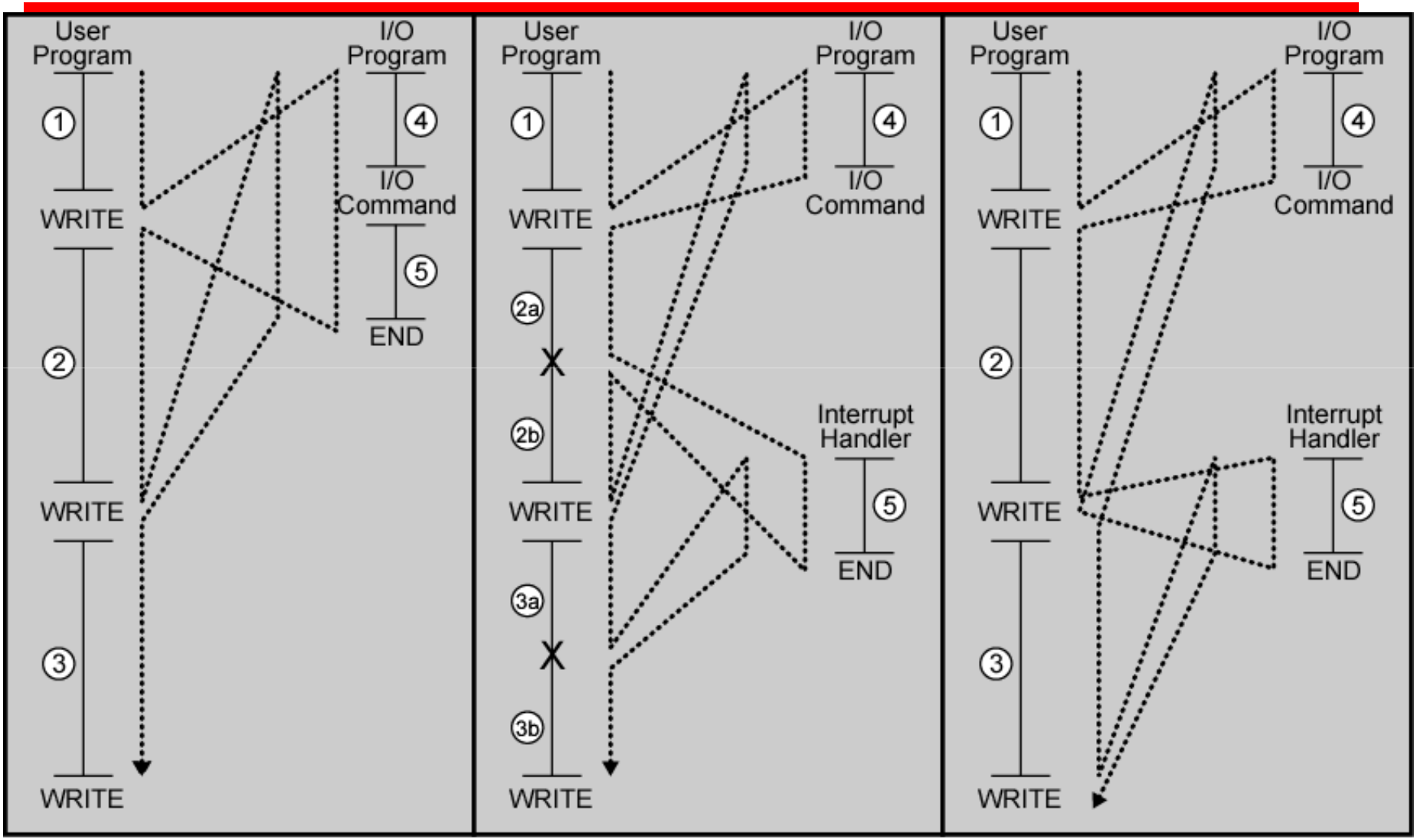
Transfer of Control via Interrupts



Instruction Cycle with Interrupts

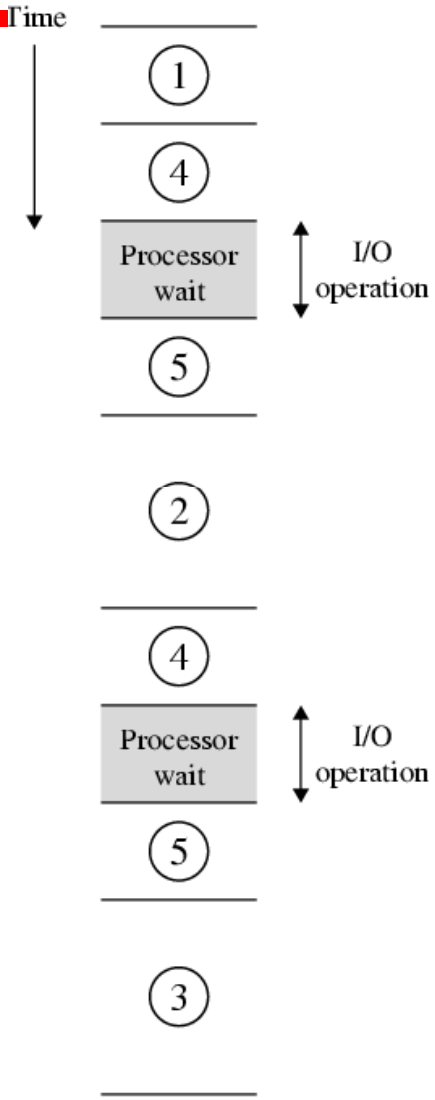


Program Flow Control

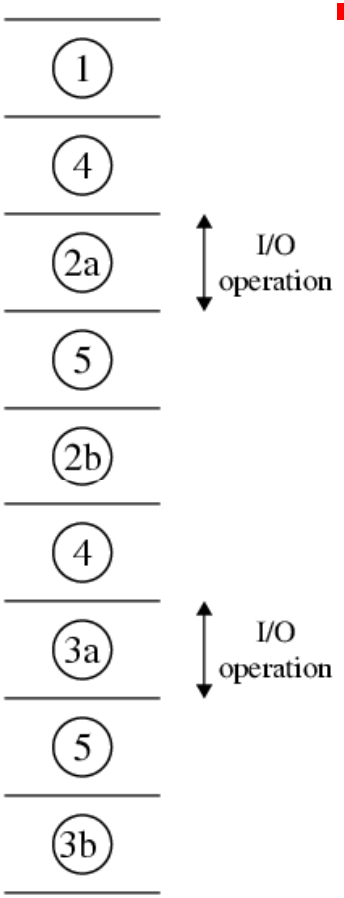


(a) No interrupts (b) Interrupts; short I/O wait (c) Interrupts; long I/O wait

Program Timing
Short I/O Wait



(a) Without interrupts

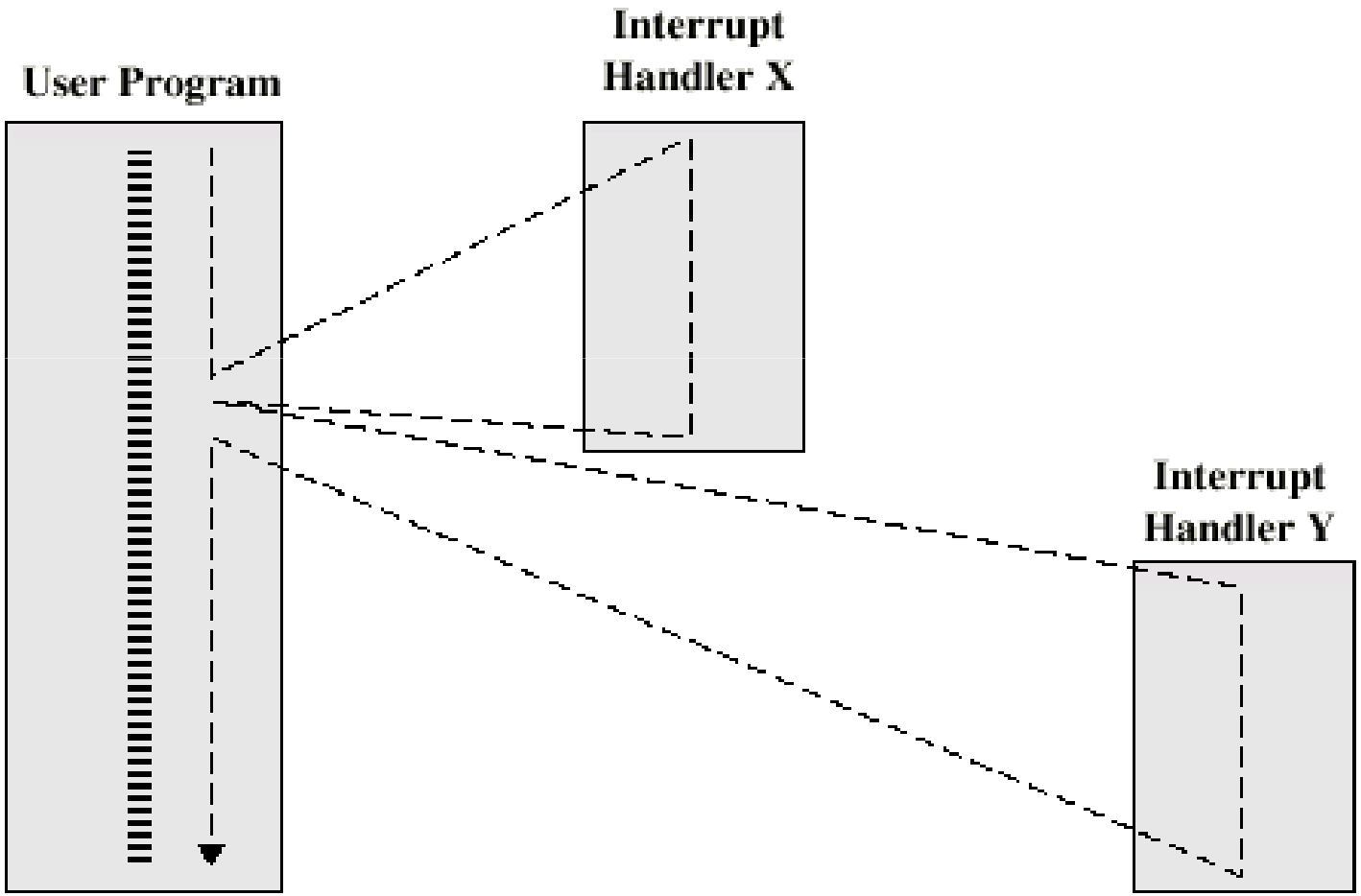


(b) With interrupts

Multiple Interrupts

- Disable interrupts
  - Processor will ignore further interrupts whilst processing one interrupt
  - Interrupts remain pending and are checked after first interrupt has been processed
  - Interrupts handled in sequence as they occur
- Define priorities
  - Low priority interrupts can be interrupted by higher priority interrupts
  - When higher priority interrupt has been processed, processor returns to previous interrupt

Multiple Interrupts - Sequential



Multiple Interrupts – Nested