Computer Architecture
M.Sc. Namar A taha

3rd class

Lecture one

Von Neumann models

This is the summary of the core attributes of the Von Neumann computer Architecture which is illustrated in fig -1. In modern computer the control unit and ALU are part of the CPU

The Von Neumann_Architecture is a design model for a stored program digital computer. Its main characteristic is a single sperate storage structure (memory) that holds both program and data.

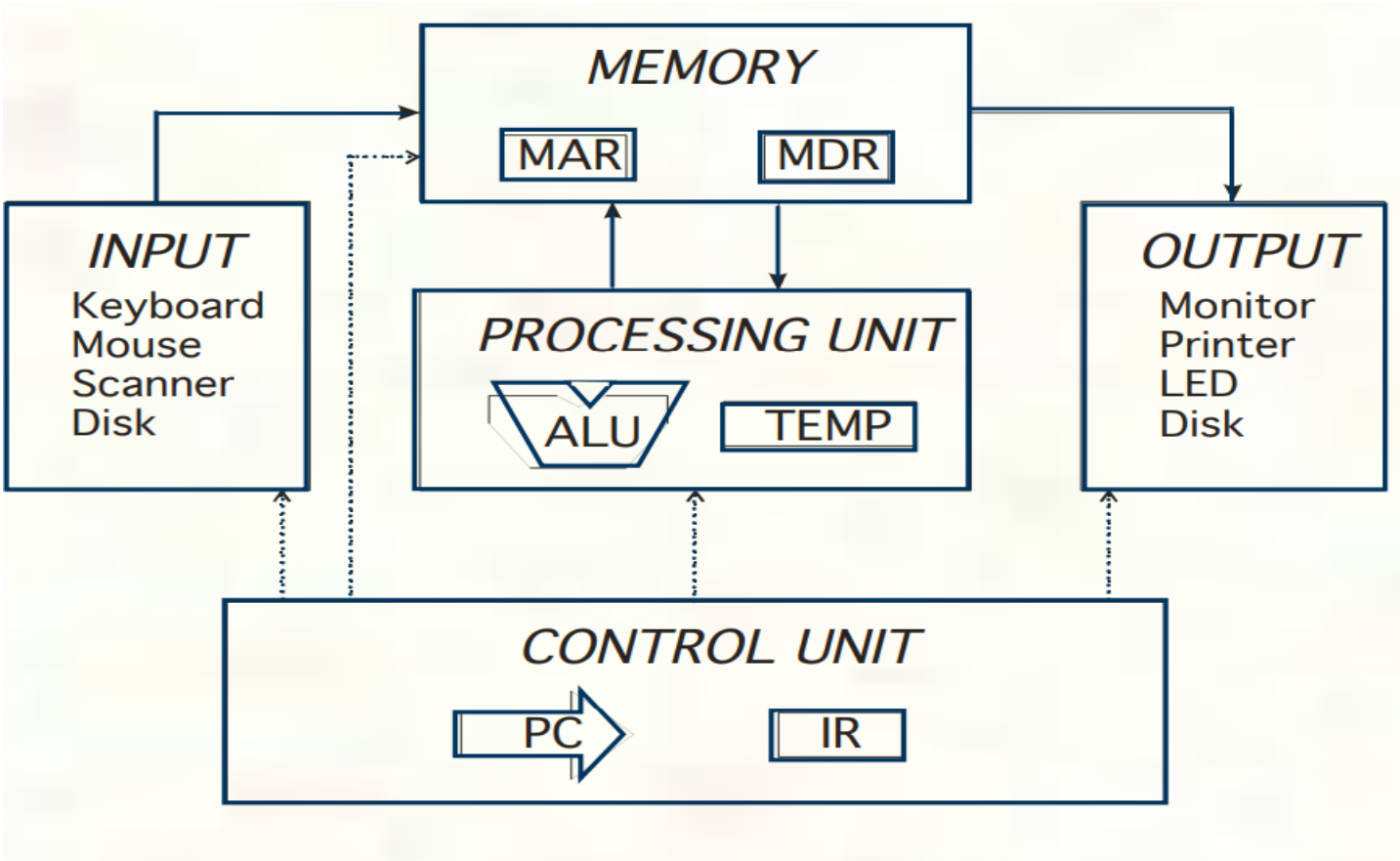


Figure 1. The von Neumann architecture model

Some important feature of the von Neumann Architecture are:

1- Both instruction (code) and data (variable and input/ output) are stored in memory
2- Memory is a collection of binary digits (bits) that have been organized in to bytes, words, and region with addresses
3- The code instruction and all data have memory addresses
4- To execute each instruction, it has to be moved to registers
5- Only registers have the "smarts" to do anything with instruction; memory location have no smarts
6- To save a result computed in the registers; has to be moved back to memory
7- Operating systems and compilers keep the instruction and data in memory organized so it doesn't get mixed up together

Von Neumann model component

1- Memory -store data and program
2- Processing unit- performs the data processing
3- Input -means to enter data and program
4- Output – mean to extract result
5- Control unit -controls the order of the instruction execution

1- Memory

- 2^k x m array of stored bits

- Address
  unique (k-bit) identifier of location
- Contents m-bit value stored in location

Basic Operations:

- LOAD

read a value from a memory location

- STORE

write a value to a memory location

How does processing unit get data to/from memory?

MAR: Memory Address Register

MDR: Memory Data Register

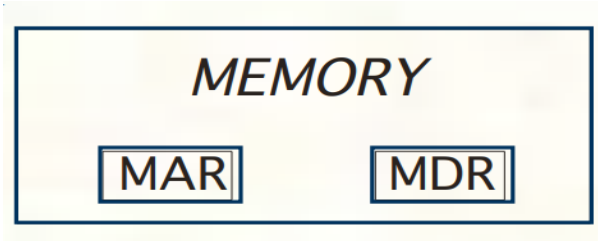
To LOAD a location (A):

1. Write the address (A) into the MAR.

2. Send a “read” signal to the memory.

3. Read the data from MDR.

To STORE a value (X) to a location (A):



1. Write the data (X) to the MDR.

2. Write the address (A) into the MAR.

3. a “write” signal to the memory

2- Processing Unit

In the simple form, it consist of two main parts

- ALU = Arithmetic and Logic Unit

could have many functional units some of them special-purpose (multiply, square root, ADD, …)

- A temporary storage, typically a set of few registers for storing few words of data.

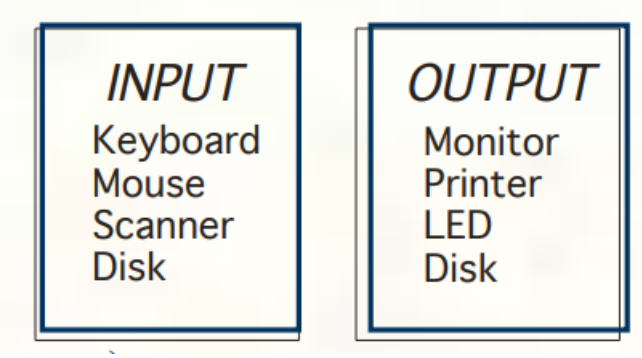
The size of data item processed by the ALU is referred to as the word length, and each data item is referred to as word

3- Input & output

In order for computer to process the information, the information it self needs to be entered into the memory

In order for us to Know the results of processing the information, the results need to be output in such a way that we can see them

In the simplest form, input and output device work with memory directly, that is, an input device places a value into some memory location and output device reads and displays a value from some memory location



4- Control unit

Directs the works of all other unit

Keeps track of which instruction is being execute and which instruction will be processed next; for this it uses two spatial register

- Instruction register (IR) contains current instruction being executed
- Program counter (PC) register Keeps a pointer (address) to the next instruction to be executed

Stored program concepts

- Program is stored in some part of computer memory as a sequence of instruction
- Instruction are represented and stored in memory as binary words
- Control unit reads an instruction form the memory

The instruction

The most basic unit of computer processing

In the simplest form, consists of two parts

- Opcode (operation code)- a portion of a machine language instruction that specifies the operation to be performed
- Operand – a part of machine language instruction that specifies the data to be operated on

Von Neumann instruction cycle

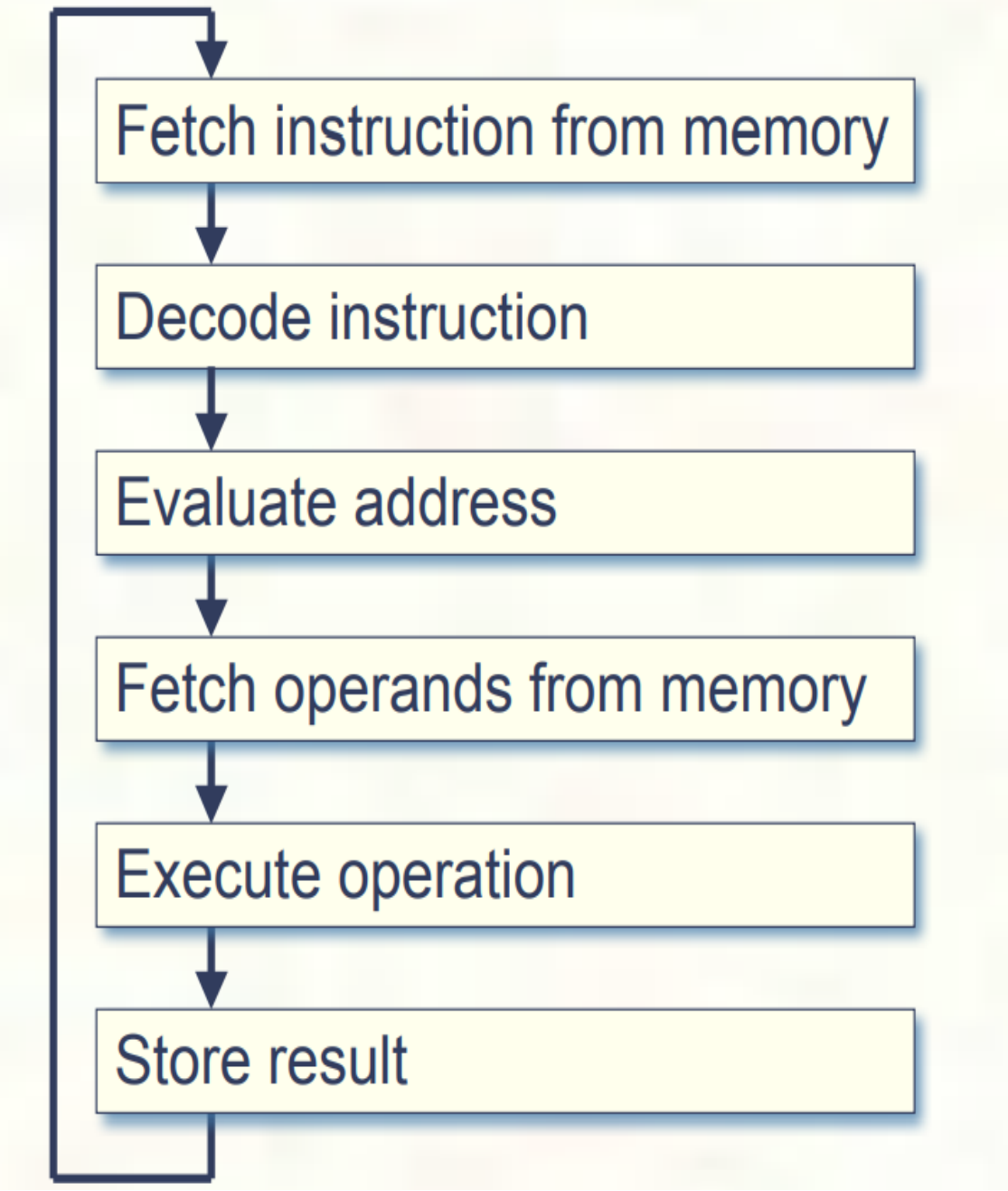


Figure 2. Von Neumann instruction cycle

1- FETCH INSTRUCTION

- Load next instruction (at address stored in PC) from memory into Instruction Register (IR).
- Copy contents of PC into MAR.
- Send “read” signal to memory.
- Copy contents of MDR into IR.
- Then increment PC, so that it points to the
- next instruction in sequence.
- PC becomes PC+1.

2- DECODE phase

The instruction stored in IR is examined in order to decide what portion of the microarchitecture needs to be involved in the execution of the instruction

3- EVALUATE ADDRESS phase

- For instructions that require memory access, compute address used for access
- Some instructions do not need this phase, e.g., instruction that work directly with the register

4- FETCH OPERAND phase

- In this phase, the source operand needed to carry out the instruction are obtained from memory
- For some instruction, this phase equal to loading values from the register file
- For other, this phase involves loading operand from memory

5- EXECUTE phase

- Instruction is carried out

- Some instruction may not require this phase e.g., data movement instructions for which all the work is actually done in the FETCH OPERAND phase

6- STORE RESULT phase

Write results to destination. (register or memory)

7- After the six phases of the instruction cycle are done, the control unit begins the next instruction cycle, starting with the new fetch (instruction) phase

- Since the PC was previous incremented by one, it contains the pointer to the next instruction to be fetched and executed

Advantage and Disadvantage of Von Neumann model

Disadvantage

Regarding disadvantages if we want to specify it is very less compared to the advantages. The processor takes more time to execute as it has to decide between the data and instruction as both are stored in same memory and also, we have two types of memory access, first to access data and next for instruction and vice versa. The above reason may also lead to system crash as there may be confusion between data and instruction

Advantages

Most of the modern disk-based operating system are based on von Neumann architecture which has made handling computer and working out computation easier. It also reduces the hardware requirements of the system as such by reducing the no. of the buses required to read the data and instruction separately from two different memories

We can also change the program (instruction) more easily with a single I/O peripheral which also increases the system's robustness.

Non-Von Neumann Architecture

Any computer architecture in which the underlying model of computation is radically different from the classical Von Neumann model. a non-Von Neumann machine may thus be without the concept of sequential flow of control (i.e., without any register corresponding to a program counter that indicate the current point that has been reached in execution of program) and /or without concept of a variable (i.e., without named "storage location in which a value may be stored and subsequently referenced or changed). Example of non-Von Neumann machines are the data flow machines and the reduction machines. In both of these cases there is a high degree of parallelism and instead of variable there are immutable bindings between name and constant values.

What is the Von Neumann architecture?

The Von Neumann model is as used in desktop computer executes instruction sequentially. Von Neumann computation are a class of computer program ideally suited to sequential processing. Turing machines are very similar

Non-Von Neumann architecture

One example is MIMD architecture (multiple instruction / multiple data), (multiple processors running in parallel)

Other example are analog computers, optical computers, quantum computers, and neural network