Printout

Saturday, March 20, 2021 8:12 PM

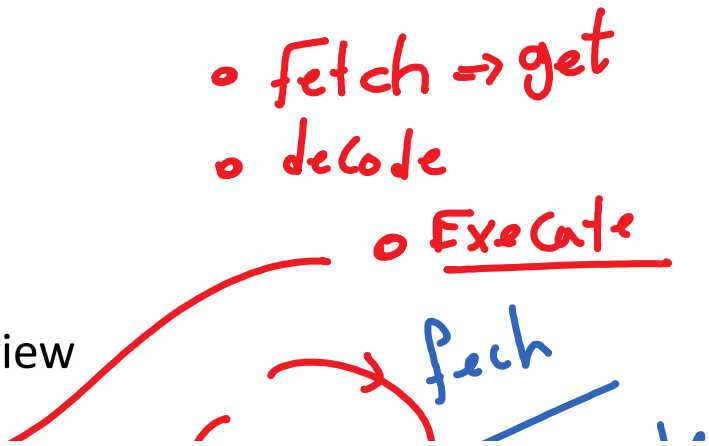
Chapter 3
A Top-Level View of Computer Function And Interconnection – Part 01

Dr. Mostafa Elhosseini

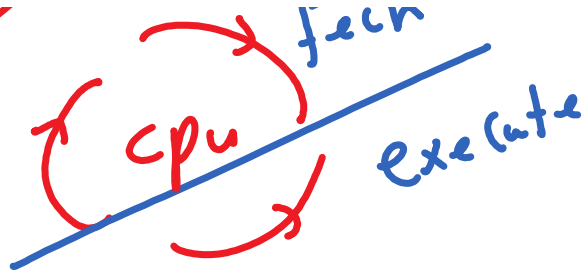
Instruction Fetch and Execute

Agenda

- Intro
- Hardwired Program
- Program Concept
  - Control unit – Instruction Decoder
- Computer component – Top-Level view
- Instruction Cycle



- Computer component – Top-Level View
- Instruction Cycle
  - Fetch Cycle
  - Execute Cycle
- Program Execution Example



Intro…

program - stored concept

- virtually all contemporary computer designs are based on concepts developed by John von Neumann
- Such a design is referred to as the von Neumann architecture

RAM

concepts developed by John von Neumann

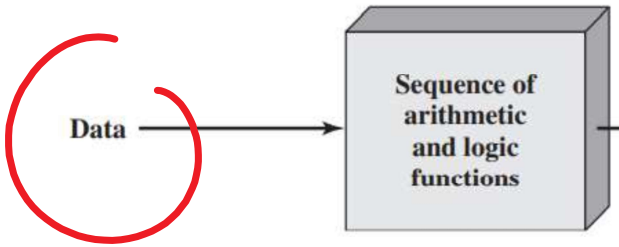
- Such a design is referred to as the von Neumann architecture and is based on three key concepts:
  - Data and instructions are stored in a single read–write memory.
  - The contents of this memory are addressable by location, without regard to the type of data contained there.
  - Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.

PC

MAR

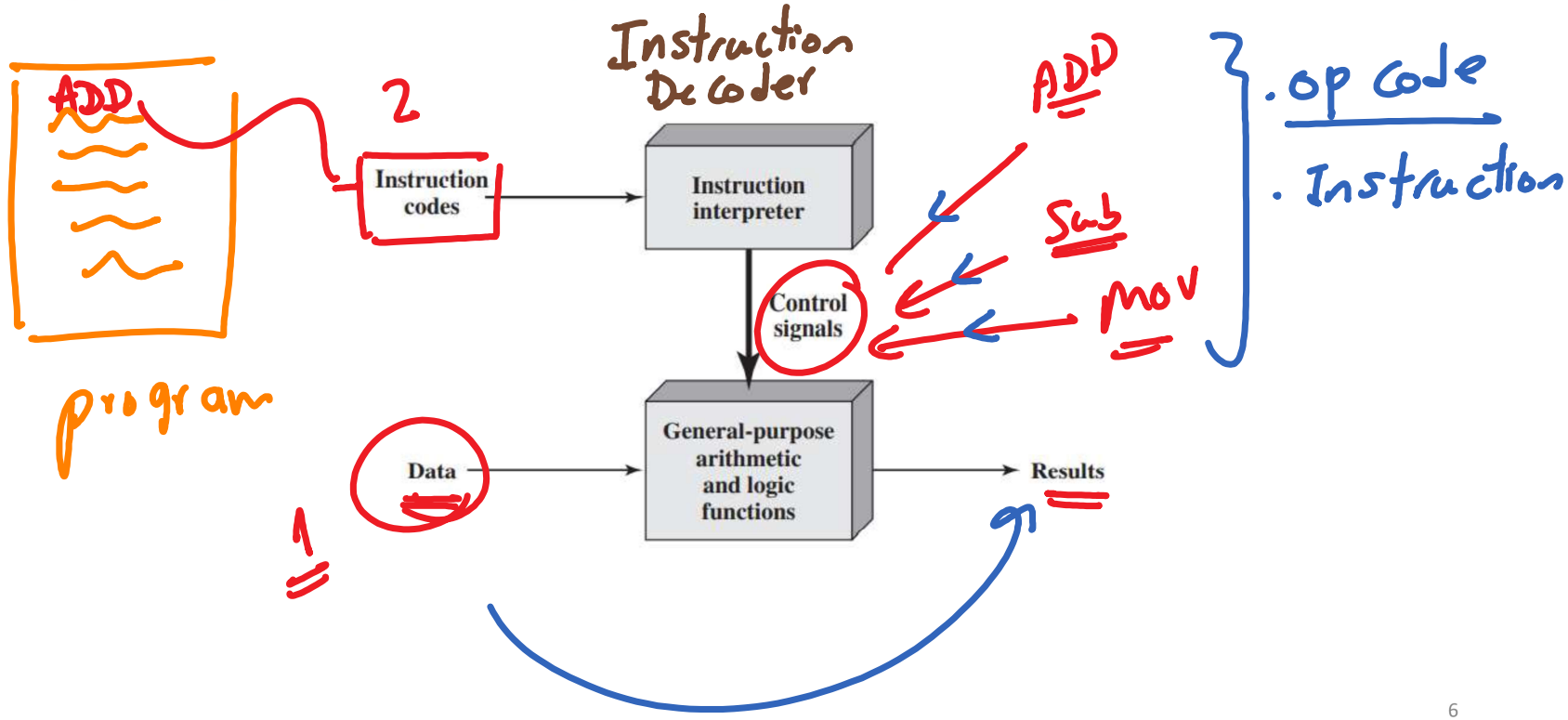
Hardwired Program

process of



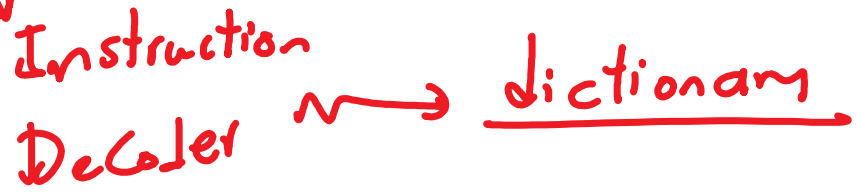
5

| Program Concept



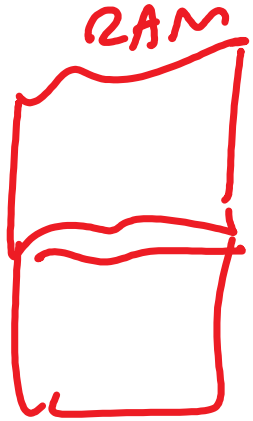
Control Unit – Instruction Decoder

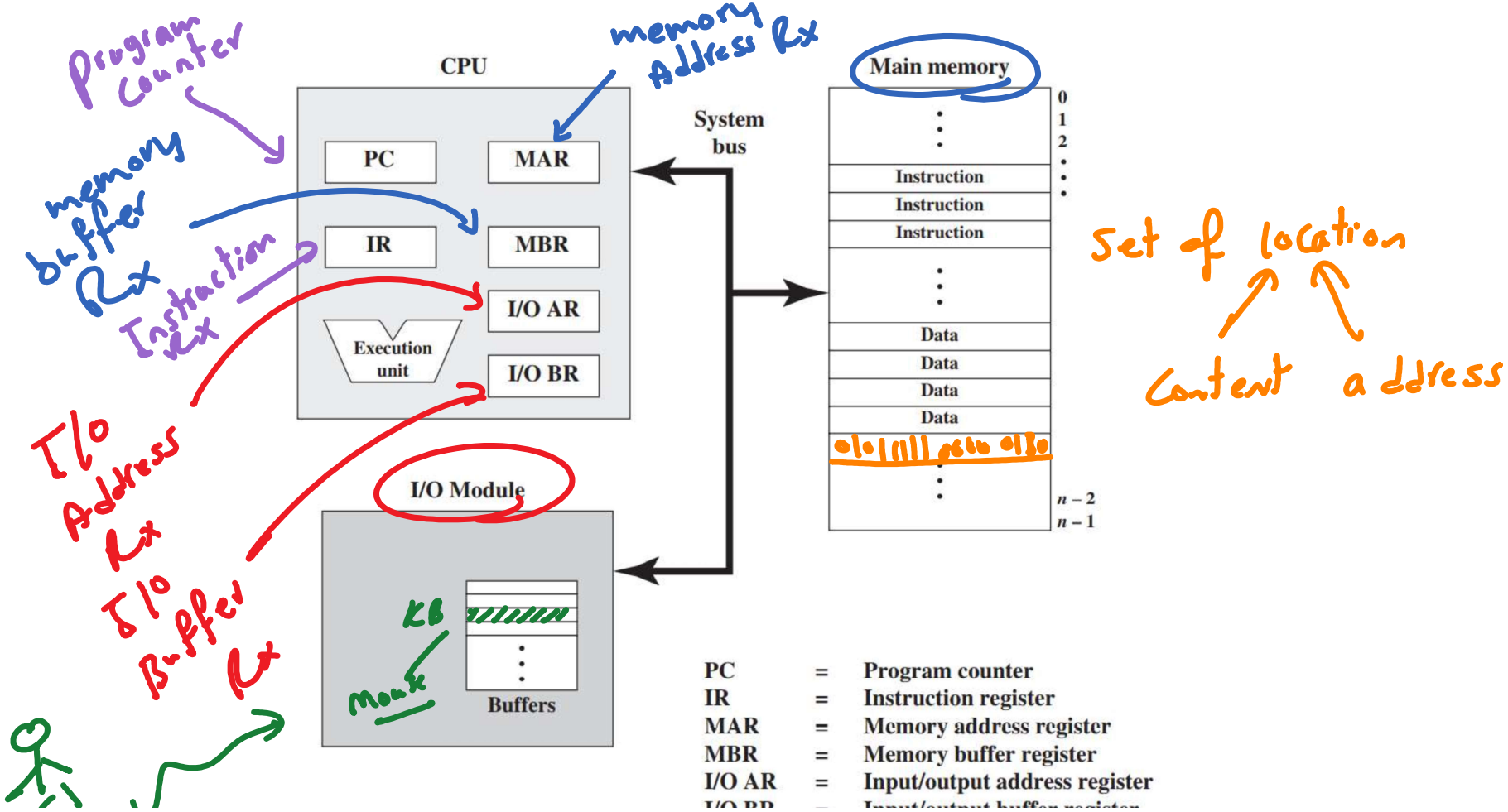
- For each operation, a unique code is provided
  - e.g., ADD, MOVE
- A hardware segment accepts the code and issues the control signals



Components

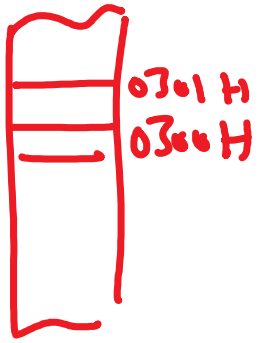
- The Control Unit and the Arithmetic and Logic Unit constitute the Central Processing Unit
- Data and instructions need to get into the system and results out
  - Input/output
- Temporary storage of code and results is needed
  - Main memory




CPU
PC
MAR
IR
MBR
I/O AR
Execution unit
I/O BR
System bus
Main memory
Instruction
Instruction
Instruction
Data
Data
Data
Data
I/O Module
Buffers
PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register

MBR	=	Memory buffer register
I/O AR	=	Input/output address register
I/O BR	=	Input/output buffer register

Instruction Cycle



- At the beginning of each instruction cycle, the processor fetches an instruction from memory.
- Program counter (PC) holds the address of the instruction to be fetched next
- Unless told otherwise, the processor always increments the PC after each instruction
- The fetched instruction is loaded into a register in the

PC = 0300H

PC = 0301 H

PC after each instruction

- The fetched instruction is loaded into a register in the processor known as the instruction register (IR)

IR

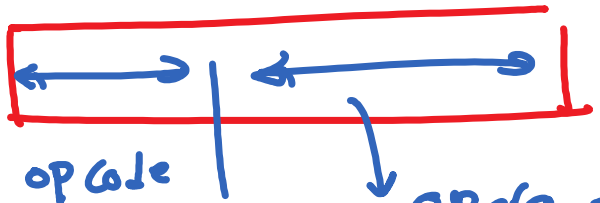
10

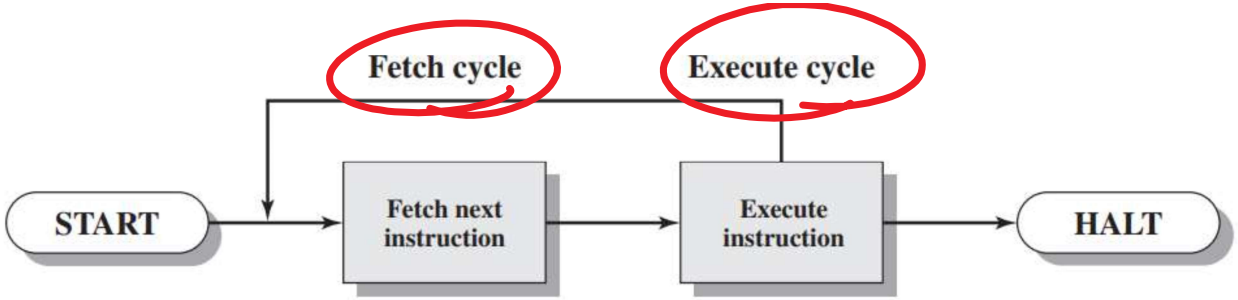
Instruction Cycle

- Two steps:
  - Fetch
  - Execute

fetch
Decode

Instruction format





Instruction Cycle

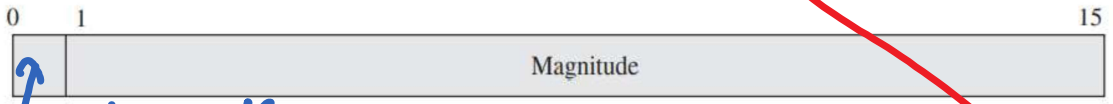
- The instruction contains bits that specify the action the processor is to take

- The instruction contains bits that specify the action the processor is to take.
- The processor interprets the instruction and performs the required action.
- In general, these actions fall into four categories:
  - Processor-memory ← Data transfer CPU, Memory
  - Processor-I/O ← Data transfer CPU I/O
  - Data processing → Logic, Arithmetic
  - Control → JMP

Program Execution Example

Program Execution Example

(a) Instruction format

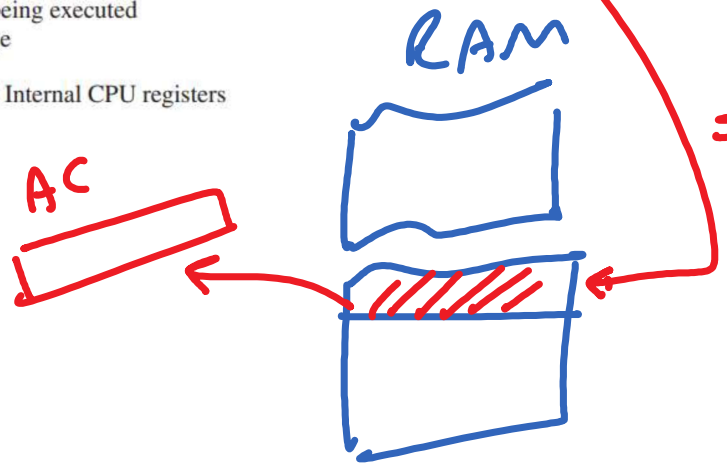


(b) Integer format

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



opcode = 4 bit 0000 ↓ 1111
# 16 opcode
# bit ⇒ 12 bit
memory = 2^{12}
= 4K location
= 4K byte memory

Sign → 1 ⇒ -ve, 0 ⇒ +ve

opcode

Program Execution Example

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