EE321

Computer Architecture

Chap. 02 : Top Level View of Computer

Dr. Abdelhakim Khouas

Email : akhouas@hotmail.com

ab.khouas@univ-boumerdes.dz

IGEE (ex. INELEC)

University M'hamed Bougara of Boumerdes

Course chapters

1. Review of Digital Design
2. Top Level View of Computer
3. Central Processing Unit (CPU)
4. Control Unit
5. Memory
6. Instruction Set and Addressing Modes

Readings

Textbook

Computer Organization and Structure, Designing for Performance, By William Stallings, 8th edition

Sections

- Chapter 1, sections: 1.1 and 1.2
- Chapter 2, sections: 2.1 and 2.5
- Chapter 3, sections: 3.1 and 3.2

Lecture Objectives

This lecture focuses on the basic structures used for computer component interconnection

- A brief examination of the basic components and their interface requirements
- A functional overview
- Issue of performance

Lecture Outline

1. Definitions
2. Computer Evolution
3. Computer Components
    1. Central Processing Unit (CPU)
    2. Memory
    3. Input/Output (IO) Peripherals
    4. System Buses
4. Computer Function
5. Performance Assessment

1. Definitions

Computer Architecture

- Refers to attributes of a computer visible to the programmer and that have a direct impact to on the logical execution of a program
  - Examples of architecture attributes: Instruction set, number of bits, memory addressing modes

Computer Organization

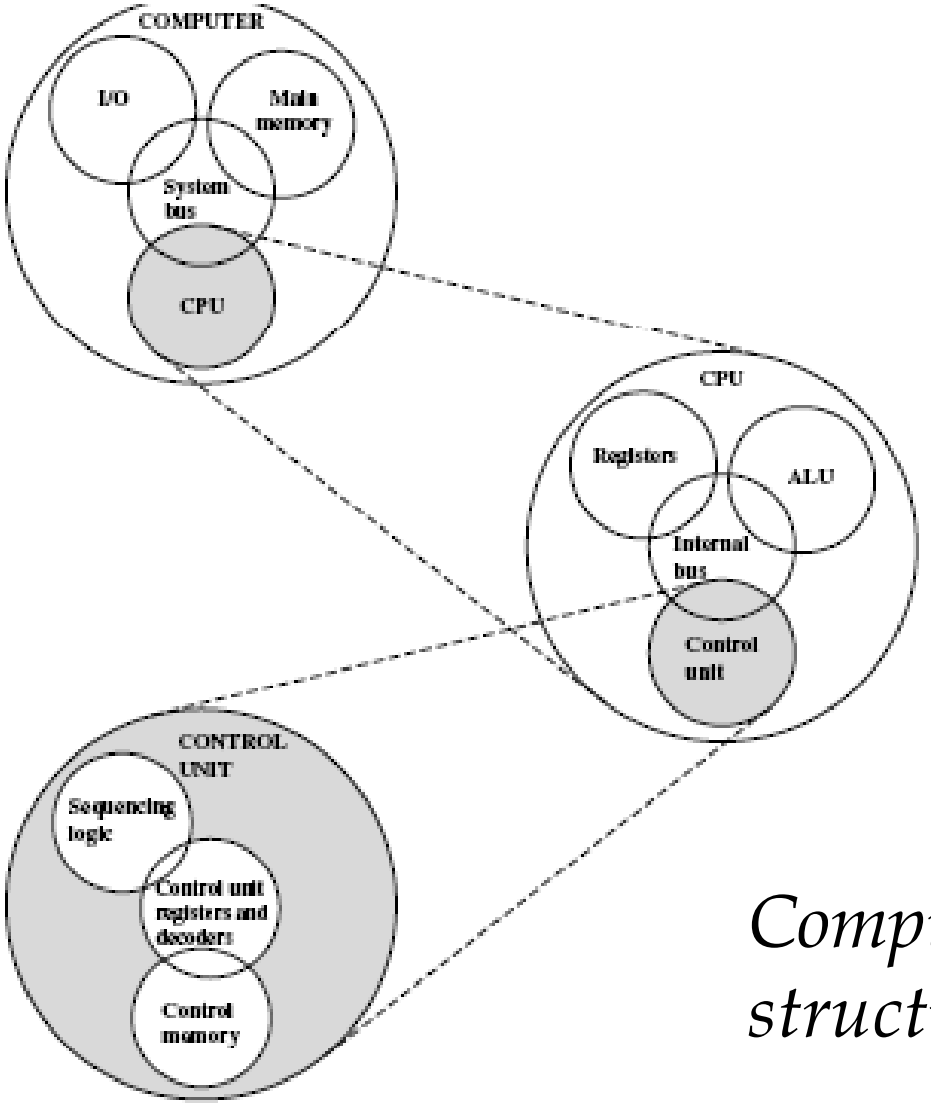
- Refers to the operational units and their interconnections that realize the architectural specifications
  - Example: Multiply Unit

1. Definitions

A computer is a complex system organized in a hierarchical system. At each system level:

- The **structure** describes the way in which the components are interrelated
- The **function** defines the operation of each individual component as part of the structure
  - Computer Structure: CPU, Memory, BUS, I/O
  - CPU Structure: ALU, Registers, Control Unit, Bus
  - Control Unit Structure: Sequencing logic, registers, decoders, memory control

1. Definitions



Computer top level structure

Source: Computer Organization and Structure, by W. Stallings

2. Computer Evolution

1947: ENIAC (Electronic Numerical Integrator And Computer)

- First computer developed at Pensylvania University, 30 tons, 150 m^2, 18 000 vacuum tubes, decimal machine, 20 10-bit accumulators, 5000 addition per second, programmed manually

1952: Von Newmann/Turing machine

- Developed at Princeton Institute for Advanced Studies (i.e. IAS computer)
- Stored-program concept
- Prototype of today general purpose computers

2. Computer Evolution

1947: Transistor invented at Bell Laboratory

- Made computers a lot smaller and more efficient

1971: Intel 4004 4-bit processor

- All of the components of a processor on a single chip (1st microprocessor)
- 108 Khz, 2300 transistors, 10μm technology

1972: Intel 8008

- First 8-bit microprocessor
- First commercially available processor

1965: Moore's Law

- The number of transistors and the frequency will double every 18 months

3. Computer Components

All contemporary computer design are based on concepts developed by Von Newman. Such a design is referred to as Von Newmann architecture or Turing machine and is based on 3 key concepts:

1. Data and instructions are stored on single read and write memory (RAM)
2. The contents of the RAM are addressable by location
3. Execution of instructions occurs in sequential (unless explicitly modified) from one instr. to the next one

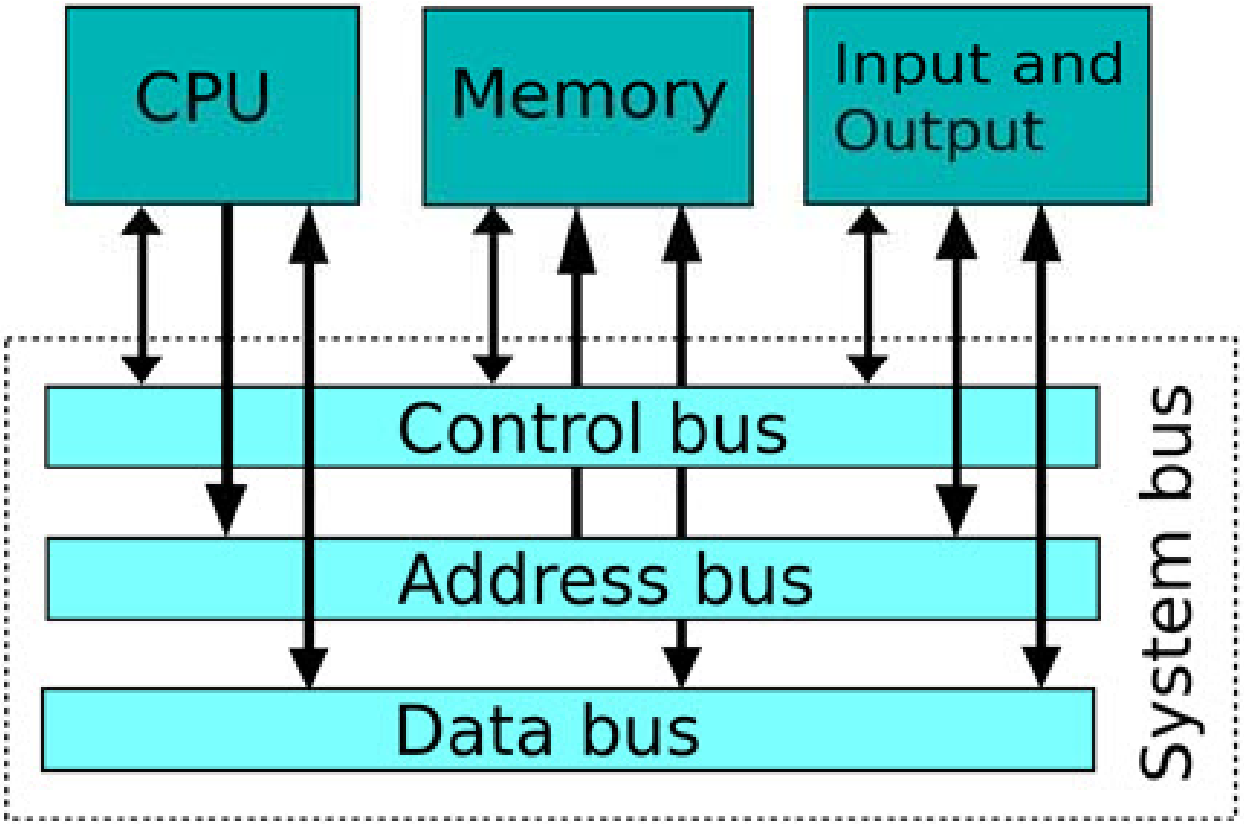
3. Computer Components

A program is a sequence of instructions called also a code

At top level, a computer consists of:

1. Central Processing Unit (CPU)
2. Memory
3. Input/Output (IO) peripherals
4. System buses for interconnections (data, address, and control buses)

3. Computer Components



Von Newmann architecture

Source: Wikimedia

3. Computer Components

Harvard architecture

- Uses two separate memories, data and program memory
- Harvard architecture is used in specific processors such as DSP (Digital Signal Processor), and in mobile communication and image processing systems

3.1. CPU

The basic function performed by CPU is execution of a program

Program is a set of sequential instructions stored in memory

Each instruction is represented by:

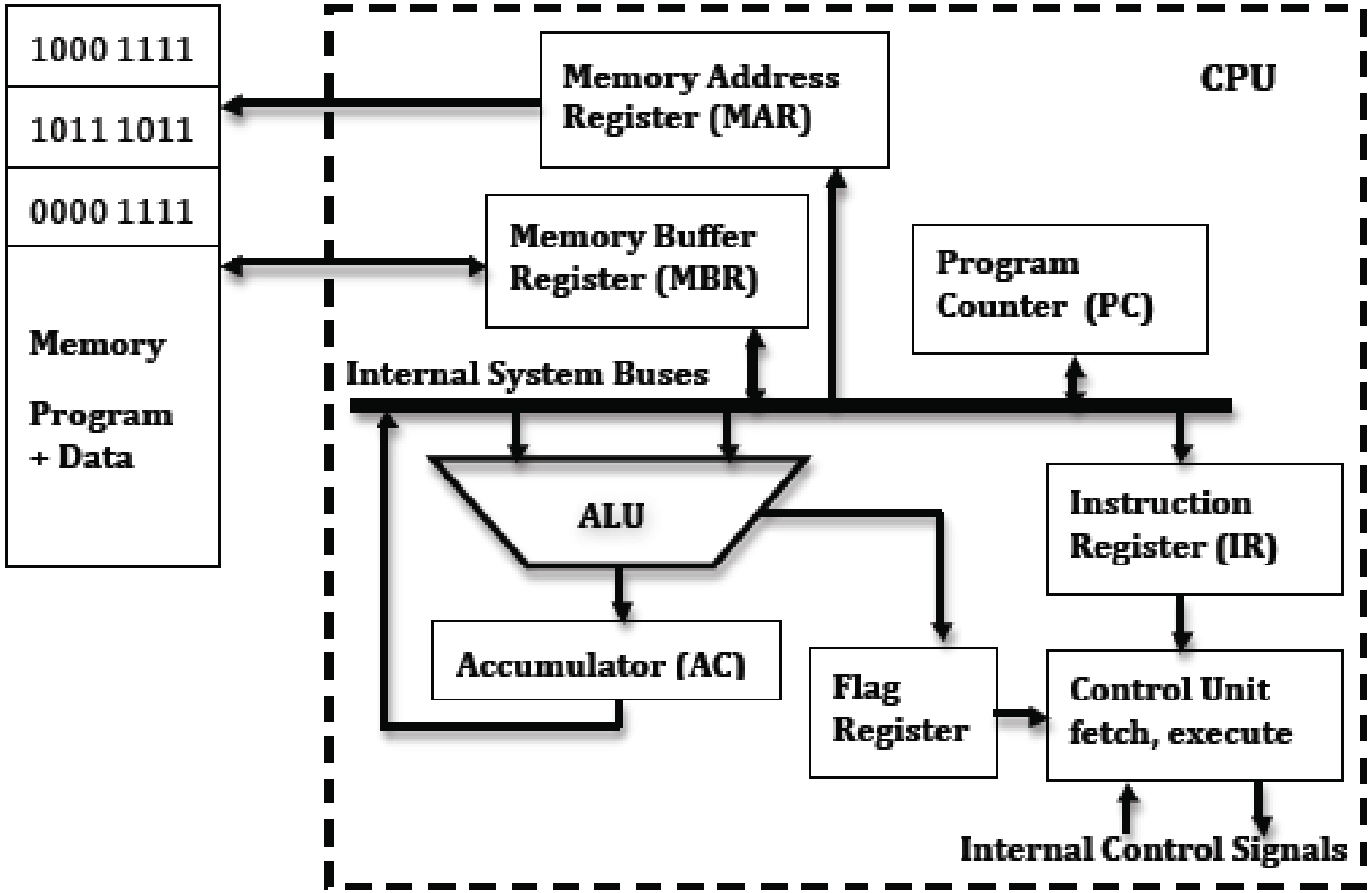
1. Operation code (opcode): specify the operation to be performed
2. Operand: specify the reference to the inputs and outputs of the operation

3.1. CPU

CPU is mainly composed of:

1. Registers
2. Arithmetic and Logic Unit (ALU)
3. Control Unit
4. Internal Bus

3.1. CPU



CPU components

Source:: Introduction to Computer Architecture, A. Maache

3.1. CPU

Registers

- Are used to temporarily store data, addresses, and codes. A generic basic CPU should have:
  1. General Purpose Registers (GPR)
  2. Program Counter (PC): it points to the address of the next instruction
  3. Memory Address Register (MAR): it contains the address for the next read and write to/from memory. PC is copied to this register to fetch next instruction (see also IOAR)
  4. Memory Buffer Register (MBR or MDR): it contains the data to be written or read into/from memory (see also IOBR)
  5. Instruction Registers (IR): it holds the current instruction which is being executed
  6. Accumulator (AC): it holds the result of the ALU

3.1. CPU

ALU

- This unit carries out the actual operation on data

Control Unit

- This units fetches program instruction from memory (fetch cycle) and stores it in IR. The instruction is decoded by the decoder (decode cycle), which then asserts the internal control signals to the relevant parts of the CPU via the internal control bus

3.1. CPU

Internal System Bus

- The 3 CPU components described above are connected together using three buses:

3.2. Memory

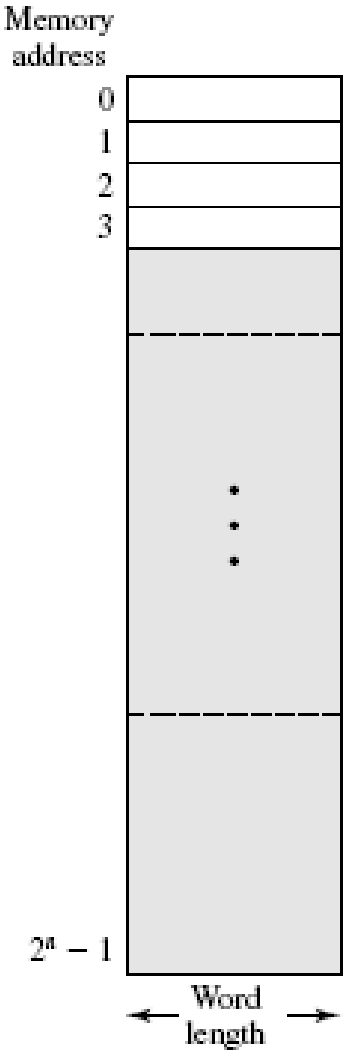
Memory module is

semiconductor devices that

store binary information

(instructions and data)

- Memory is organized as array of locations, each location holds n bits (N words of n bits)
- Depth= nb. of locations
- Width=nb. of bits per location
- Size (capacity) = depth x width



Source: Computer Organization and Structure, by W. Stallings

3.2. Memory

Each location in the memory has an address

(also called pointer or reference)

- The CPU can access to any location in the memory using its address

Two main types off memory

- Read Only Memory (ROM): The CPU can only read data from the memory
- Read/Write memory: The CPU can write and read any location of memory (also called RAM)

Source: Computer Organization and Structure, by W. Stallings

3.2. Memory

Stack memory

- Stack memory is R/W memory, but the CPU can only read the last saved data (also called Last-In First-Out memory)
- The CPU has a special register called Stack Pointer (SP) that contain the address of the last data saved in the stack
- The CPU uses PUSH instruction to save a data and POP instruction to read data from the stack
- SP is updated automatically after PUSH and POP instructions

Source: Computer Organization and Structure, by W. Stallings

3.2. Memory

Memory map

- Typically, in a computer with N-bit address lines, it is capable of addressing 2^N memory locations
- The memory range is divided by the programmer (or hardware designer) into sections suited to specific roles (e.g. program section, data section, stack section, ROM, EEPROM, RAM, etc.)
- This memory range division is called Memory Map

Source: Computer Organization and Structure, by W. Stallings

3.2. Memory

Example: Memory map of Z80 board

Location	Description
0000 3FFF	Monitor EEPROM U24 (?? KB)
4000 5DFF	User RAM U26, User Stack
5E00 5FFF	Monitor Stack
6000 7FFF	Expansion (already interfaced)
8000 FFFF	Unused

Source: Computer Organization and Structure, by W. Stallings

3.3. I/O Peripherals

A computer is useless unless it communicates with the external world (I/O peripherals)

- Input modules transfer data from the outside world to the CPU, it includes devices like: keyboard, switches, and ADC
- Output modules transfer data from CPU to the outside world, it includes devices like: display, printer, LEDs, DAC
- From internal view of CPU, I/O is functionally similar to memory, each I/O device has a port with unique address to read and write to/from it

3.4. System Buses

A bus is a communication path between CPU and other peripherals

All peripherals share same bus, with only one device using the bus at any time

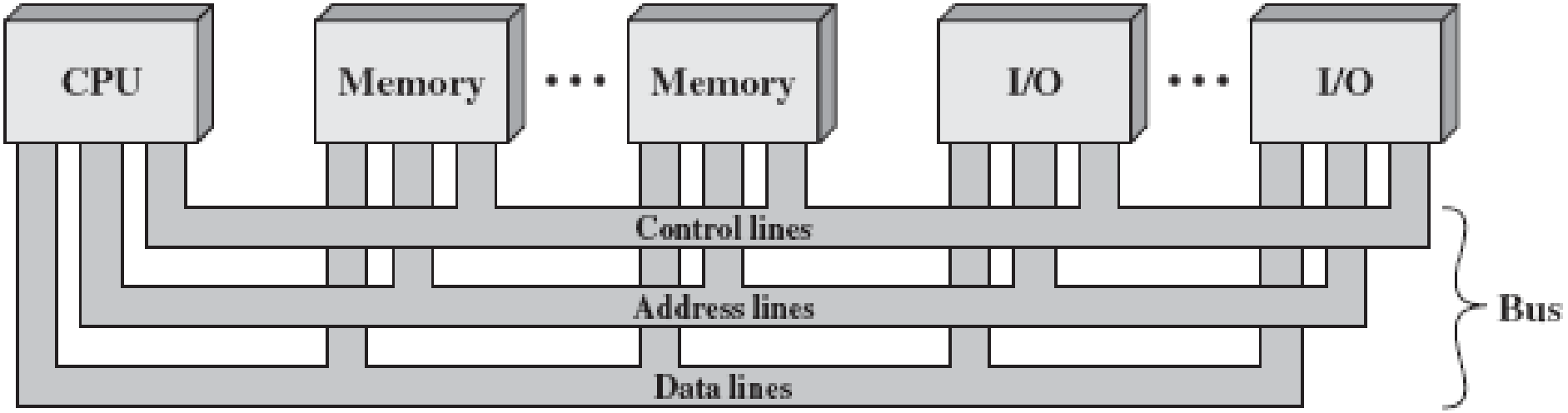
- Devices are connected to the bus through tri-state buffer

A system bus consists of about 50 to hundreds of separate lines, with particular function for each line

3.4. System Buses

Computers buses are divided into three main groups:

1. Data Bus
2. Address Bus
3. Control Bus



Source: Computer Organization and Structure, by W. Stallings

3.4. System Buses

Data bus

- Provide a bidirectional path for moving the data, it may consist of 32, 64, 128 or more bits (lines) referred as the data bus width

Address bus

- Provide a unidirectional path to send addresses to memory and I/O devices. The width of this bus specifies the maximum memory capacity of the system. A 16-bit address bus is capable of addressing 64K locations (Q: memory size ???)

Control bus

- Used to control the access to data and address buses

4. Computer Function

Basic function performed by a computer is execution of a program (code)

- A program consists of a set of sequential instructions stored in memory

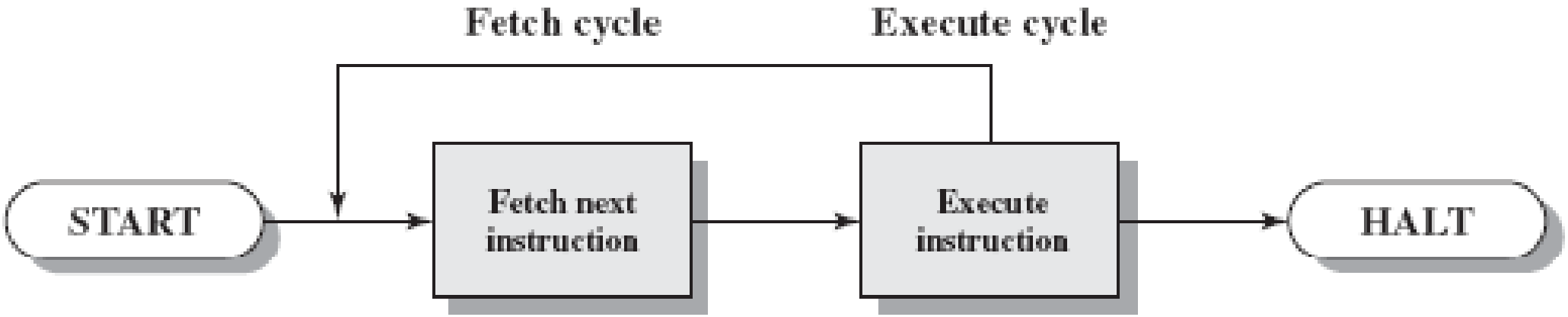
In its simplest form, instruction processing consists of two steps

1. CPU reads the instruction from memory one at time (fetch cycle)
2. CPU decodes and executes the instruction (decode/execute cycle)

4. Computer Function

Program execution

- It consists of repeating the process of fetch, and decode/execute cycles



Basic instruction cycle

Source: Computer Organization and Structure, by W. Stallings

4. Computer Function

Fetch cycle

- PC register holds the next instruction to fetch
- CPU fetches instruction from memory location pointed by PC (Q: How many memory reads are needed to fetch the instruction ?)
- PC is incremented to point to the next instruction (Q: PC = PC+??)
- Load current instruction into IR register (Q: What is the content of MAR and MBR registers at the end of fetch cycle?)

4. Computer Function

Decode/Execute cycle

- CPU decodes the fetched instruction
- CPU performs the required action (e.g. Data transfer to fetch needed data and store result, data processing, and control operation)

Questions:

- How many memory reads are needed in execute cycle?
- How many memory writes?
- Give example of control operation instructions

4. Computer Function

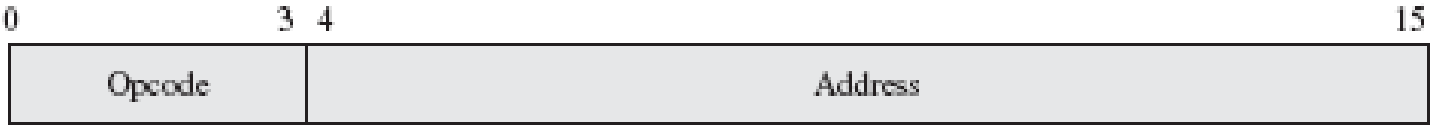
Example1: Hypothetical machine function

Consider 16-bit hypothetical machine with the following specs:

- Instructions are 16-bit long (2bytes)
- Address bus is 12-bit long
- Instruction format provides 4 bits for the opcode and 12 bits for addressable memory
- Single data register called AC
- Partial list of opcodes is:
  - '1H' = 0001 = Load AC from memory location (Q: which address?)
  - '2H' = 0010 = Store AC to memory
  - '5H' = 0101 = Add to AC the content of memory
  - '7H' = Halt execution

4. Computer Function

Example1: Hypothetical machine function



(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

Characteristics of hypothetical computer

Source: Computer Organization and Structure, by W. Stallings

4. Computer Function

Example1: Hypothetical machine function

1. What is the size of AC, IR, MAR, MBR and PC registers?
2. What is the maximum nb. of different opcodes?
3. What is the maximum directly addressable space?
4. Consider the memory content of the table. What is the size of the program at address 300H? Explain the execution of the program.

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0002H
942H	0001H
...	...

4. Computer Function

Example1: Hypothetical machine function

5. Give the content of registers PC, IR, MBR, MAR at the end of the execution of:
   a) Fetch cycle of the 1st instruction,
   b) Execute cycle of the 1st instruction,
   c) Instruction cycle of the last instruction.

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0005H
942H	0001H
...	...

4. Computer Function

Example1: Hypothetical machine function

- Step 1: fetch of 1st instruction PC=300H
  - IR <--- [PC] = [300] = 1940H
- Step 2: execute 1st instruction
  - Opcode = 1H = LOAD to AC
  - Address = 940
  - AC <--- [940] = 0003H

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0002H
942H	0001H
...	...

4. Computer Function

Example1: Hypothetical machine function

- Step 3: fetch of 2st instruction PC=301H
  - IR <--- [PC] = [301] = 5941H
- Step 4: execute 2nd instruction
  - Opcode = 5H = ADD to AC
  - Address = 941
  - AC <--- AC+[941] = 0002+0003H =0005H

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0002H
942H	0001H
...	...

4. Computer Function

Example1: Hypothetical machine function

- Step 5: fetch of 3rd instruction PC=302H
  - IR <--- [PC] = [302] = 2941H
- Step 6: execute 3rd instruction
  - Opcode = 2H = STORE AC
  - Address = 941
  - [941] <--- AC = 0005H

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0005H
942H	0001H
...	...

4. Computer Function

Example1: Hypothetical machine function

- Step 7: fetch of 4th instruction PC=303H
  - IR <--- [PC] = [303] = 7940H
- Step 8: execute 4th instruction
  - Opcode = 7H = HALT program

Adr.	Data
...	...
300H	1940H
301H	5941H
302H	2941H
303H	7940H
...	...
940H	0003H
941H	0005H
942H	0001H
...	...

4. Computer Function

Example 2: Program execution

- Consider a 16-bit CPU with 12-bit address bus, 16-bit data bus and single accumulator register named A. The CPU supports the opcodes given in table 1, the instruction format provides 4 bits for the opcode and 12 bits for the operand. Consider the memory contents of table 2.

4. Computer Function

Example 2: Program execution

Opcode	Mnemonic	Description
0000 (0H)	LDD	Load to the accumulator the content of the memory
1000 (8H)	LDI	Load to the accumulator the value of the operand
0001 (1H)	STD	Store the content of the accumulator
0010 (2H)	ADD	Add the content of the memory to the accumulator
1010 (AH)	ADI	Add the value of the operand to the accumulator
0011 (3H)	MLD	Multiply the content of the memory with the accumulator
1011(BH)	MLI	Multiply the value of the operand with the accumulator
1111 (FH)	HLT	Stop execution

Table 1: Opcodes partial list.

4. Computer Function

Example 2: Program execution

1. What is the range of 2's complement value that can be specified in the ADI instruction?
2. Write and briefly explain the corresponding assembly code of the program stored in memory location 800H.
3. What is the size of this program?
4. Give the contents of registers A, MAR, MBR, PC, and IR after the execution of the fetch cycle of the first instruction.

Adr.	Data
800H	080AH
801H	B010H
802H	280BH
803H	180CH
804H	A001H
805H	1809H
806H	F80DH
807H	080EH
808H	080EH
809H	0032H
80AH	000FH
80BH	0033H
80CH	0034H

Table 2

4. Computer Function

Example 2: Program execution

5. Give the contents of registers A, MAR, MBR, PC and IR after the execution of the execute cycle of the first instruction.
6. Give the contents of registers A, MAR, MBR, PC, and IR after the execution of the fetch cycle of the second instruction.
7. Give the contents of registers A, MAR, MBR, PC and IR after the execution of the second instruction.
8. Give the memory contents (only modified locations) after the execution of the program.

Adr.	Data
800H	080AH
801H	B010H
802H	280BH
803H	180CH
804H	A001H
805H	1809H
806H	F80DH
807H	080EH
808H	080EH
809H	0032H
80AH	000FH
80BH	0033H
80CH	0034H

Table 2

5. Performance Assessment

Key parameters to consider in evaluating computer system performances:

- Speed
- Cost
- Size
- Reliability
- Power consumption

5. Performance Assessment

Clock speed and instructions per seconds

- Operations executed by the processor are governed by a system clock
  - 1 GHz processor receives 1 billion pulses per second
  - Processor frequency (clock rate) is fixed by the design and physical layout of the processor
- The execution of a given instruction involves a certain number of clock cycles (from few cycles to a dozens cycles)
- When pipelining is used, multiple instructions are being executed simultaneously

5. Performance Assessment

Instruction execution rate

- A common measure of performance for a processor is the rate at which instructions are executed, expressed as millions of instructions per second (MIPS), referred to as the MIPS rate

$$\text{MIPS rate} = \frac{f}{CPI \times 10^6}$$

$$f : clock\ frequency$$

$$CPI : average\ clock\ cycles\ per\ instruction$$

5. Performance Assessment

Instruction execution rate

- Another common performance measure deals only with floating-point instructions. Floating point performance is expressed as millions of floating-point operations per second (MFLOPS), defined as follows:

$$\text{MFLOPS rate} = \frac{number\ of\ executed\ floating\text{-}point\ operation}{execution\ time \times 10^6}$$

Summary

Computer Components

- Central Processing Unit (CPU)
- Memory
- Input/Output (IO) Peripherals
- System Buses

CPU

- Registers: PC, AC, IR, MAR, MBR, and GPR
- ALU
- Control unit
- Internal buses

Computer Function

- Program execution
- Fetch Cycle
- Decode cycle

Performance Assessment

- MIPS and MFLOPS