Computer Organization and Architecture

Basic Operational Concepts

Veena Thenkanidiyoor

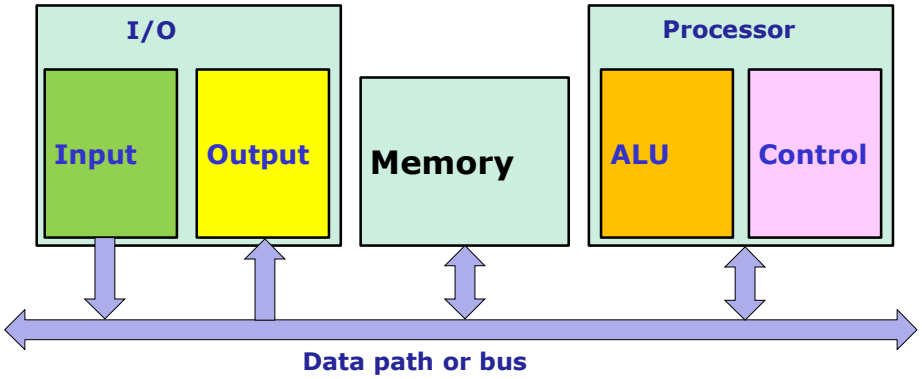
National Institute of Technology Goa

Recap

- Hierarchical nature of a computer
- Structure of a computer
- Function of a computer
- Internal of a single processor system
- Contemporary computers with multiple processors
- History of computers
- Computer organization- computer architecture
- Computer types
- Layered view of a computer system
- Computer as a multilevel machine
- Microarchitecture level
- Course content

Functional Units

- Input Unit
- Memory Unit
- Arithmetic and Logic Unit (ALU)
- Output Unit
- Control Unit



3

Information Processed by Computer

LOAD LocA, R0
ADD LocB, R0

- **Instructions:**
  - Instructions are commands that
    - Govern the transfer of information with in computer as well as between the computer and its I/O devices
    - Specify the arithmetic and logic operations to be performed
  - A set of instructions that perform a task is called program
  - Usually a program is stored in memory
  - Processor fetches the instructions that make up the program from memory, one after another, perform desired operation
- **Data:**
  - They are numbers or encoded characters that are used as operands by the instructions
- Information handled by a computer is encoded in a suitable format (string of binary digits called bits – 0/1)

4

Basic Operational Concepts

- High-level program segment

```
scanf("%d, %d", a,b);
c=a+b;
printf("&d", c);
```

COMPILER

ASSEMBLER

- Assembly level program segment

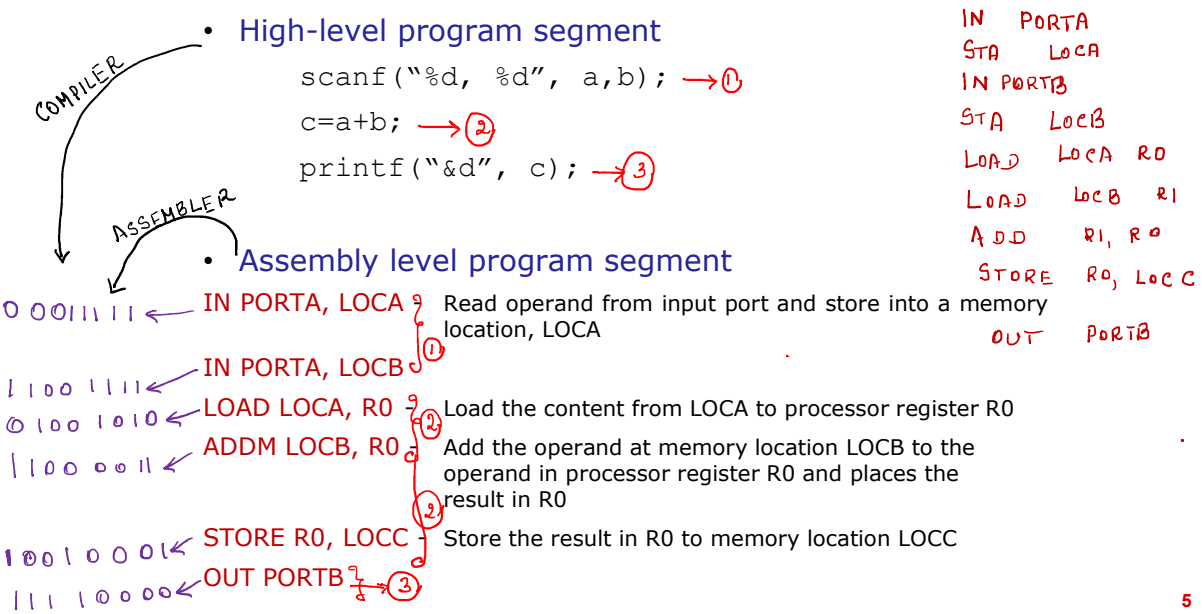
```
IN PORTA, LOCA
IN PORTA, LOCB
LOAD LOCA, R0
ADDM LOCB, R0
STORE R0, LOCC
OUT PORTB
```

Read operand from input port and store into a memory location, LOCA

Load the content from LOCA to processor register R0

Add the operand at memory location LOCB to the operand in processor register R0 and places the result in R0

Store the result in R0 to memory location LOCC



5

5

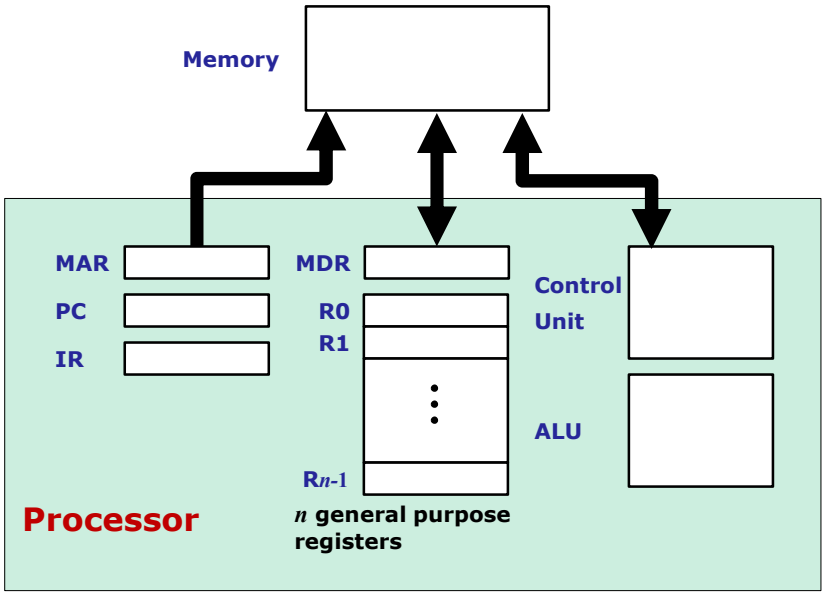
Execution of an Instruction

- Execution of an instruction requires to perform several steps
  - Instruction is fetched from memory into processor
  - If the instruction include operands, then the operands are fetched
  - If an instruction is for arithmetic operation, perform that operation on the fetched operands and store the results in destination location
- Transfers between memory and processor are started by sending the address of the memory location to be accessed to memory unit and issuing the appropriate control signals

6

6

Operational Details



7

7

Registers in Processor

- General purpose registers
  - Hold the operands or address of the operand
  - Typically 16 to 32
- IR (Instruction Register)
  - Holds the instruction currently being executed

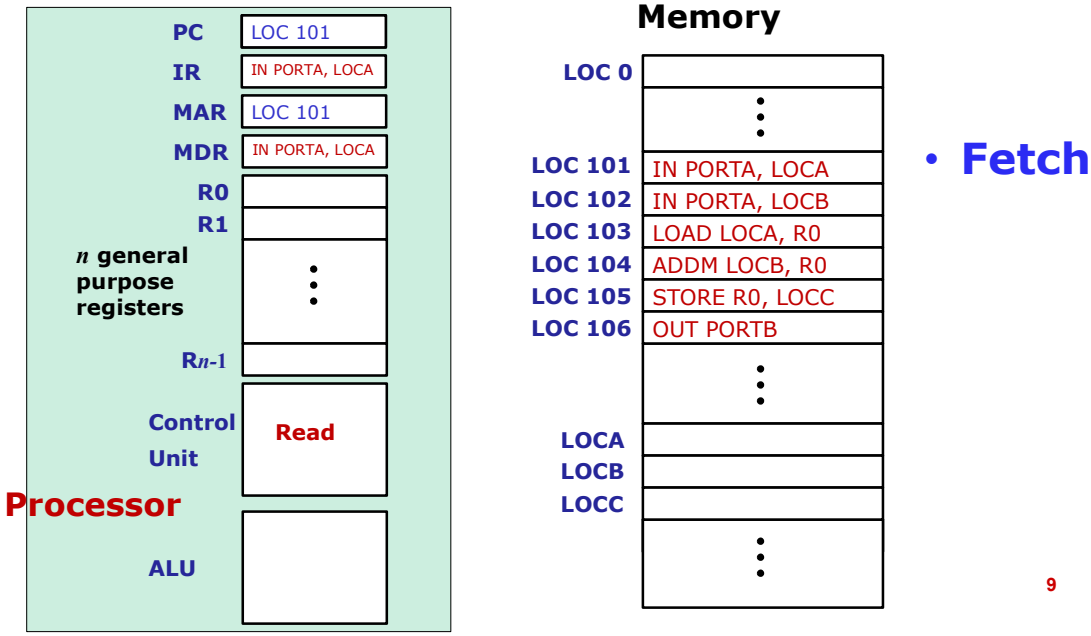
ADD R0, R1 11100000 Instruction word.

- PC (Program Counter)
  - Holds the memory address of the next instruction to be fetched and execured
- MAR (Memory Address Register)
  - Holds the address of the memory location to be accessed
- MDR (Memory Data Register)
  - Holds the data to be written into or read out of the addressed location

8

8

Execution of an Instruction, IN

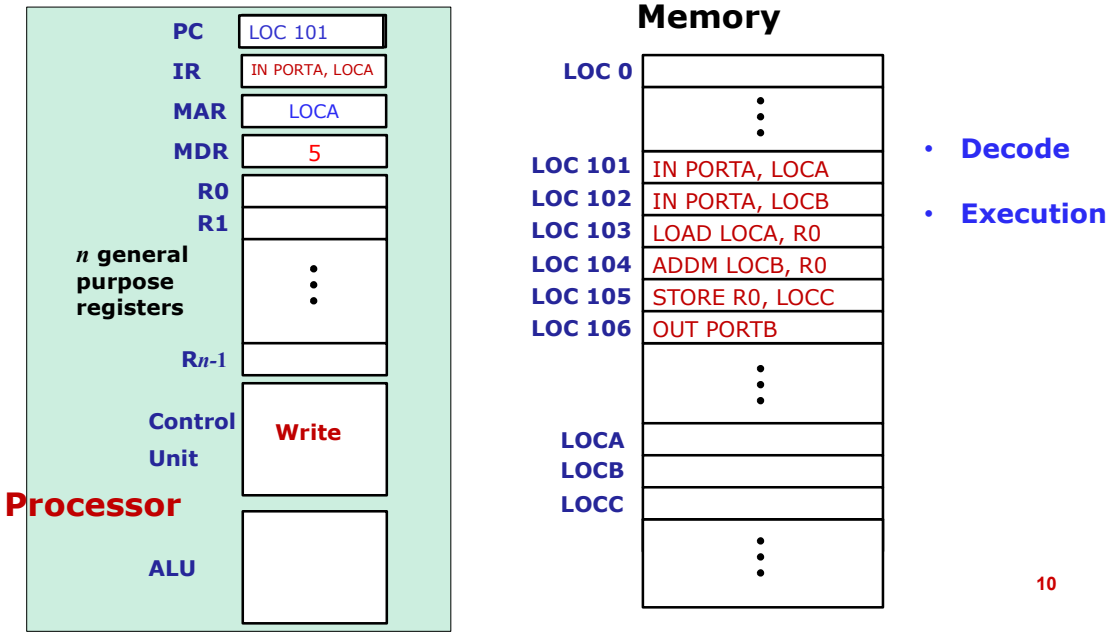


- **Fetch**

9

9

Execution of an Instruction, IN

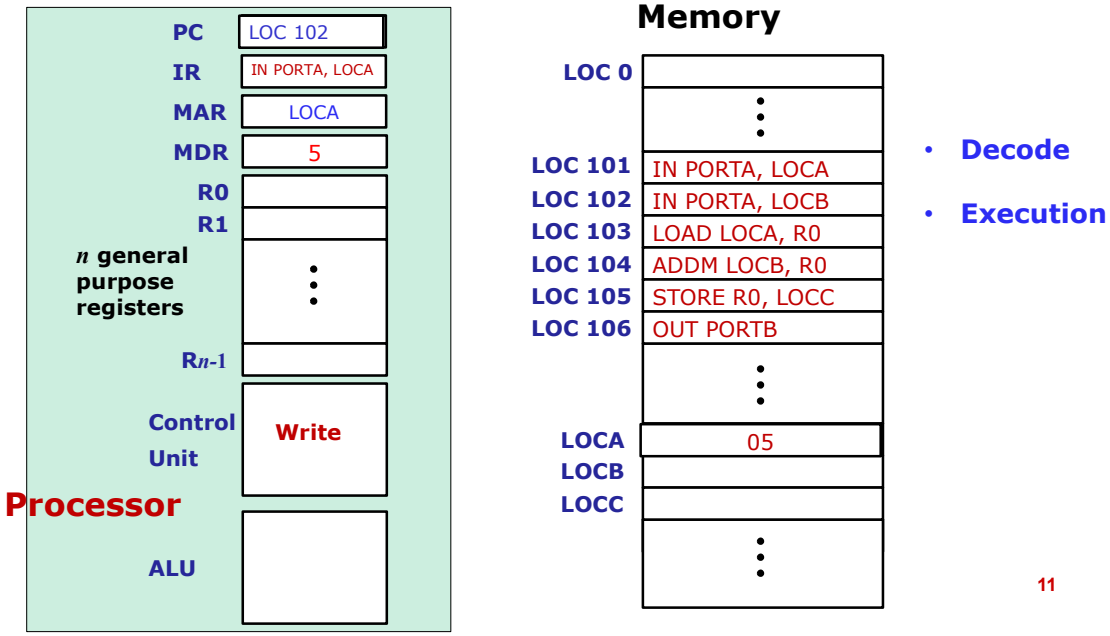


- **Decode**
- **Execution**

10

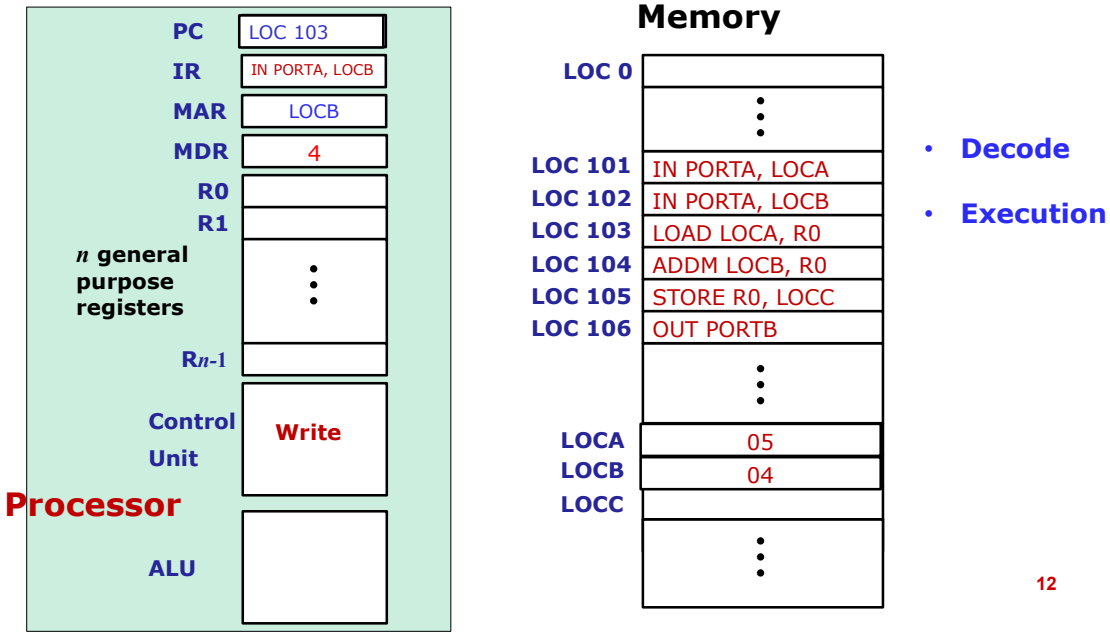
10

Execution of an Instruction, IN



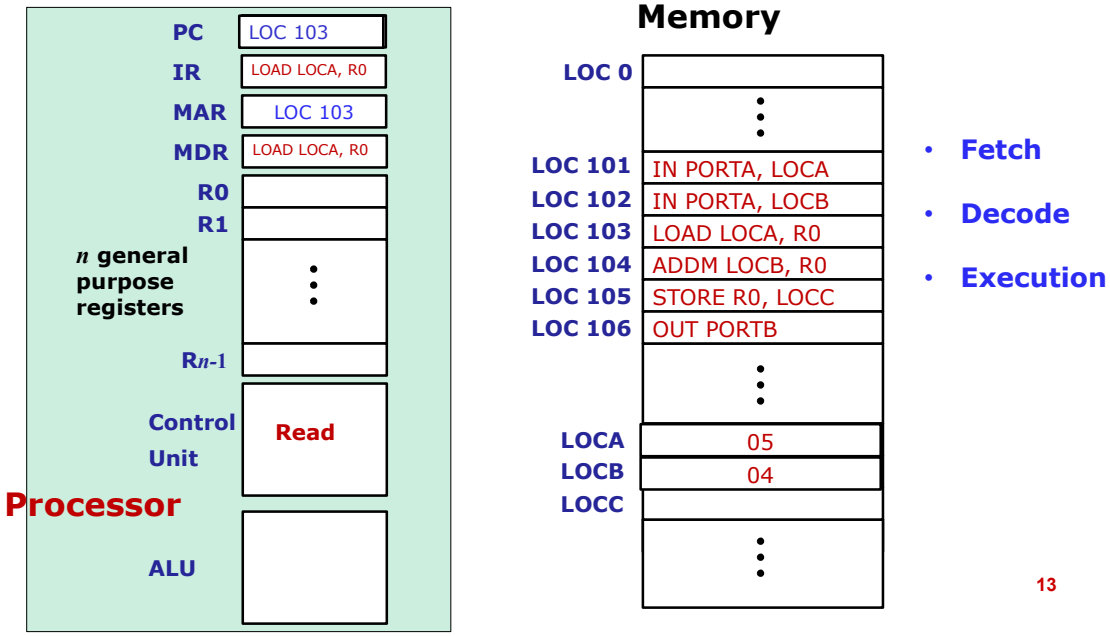
11

Execution of an Instruction, IN



12

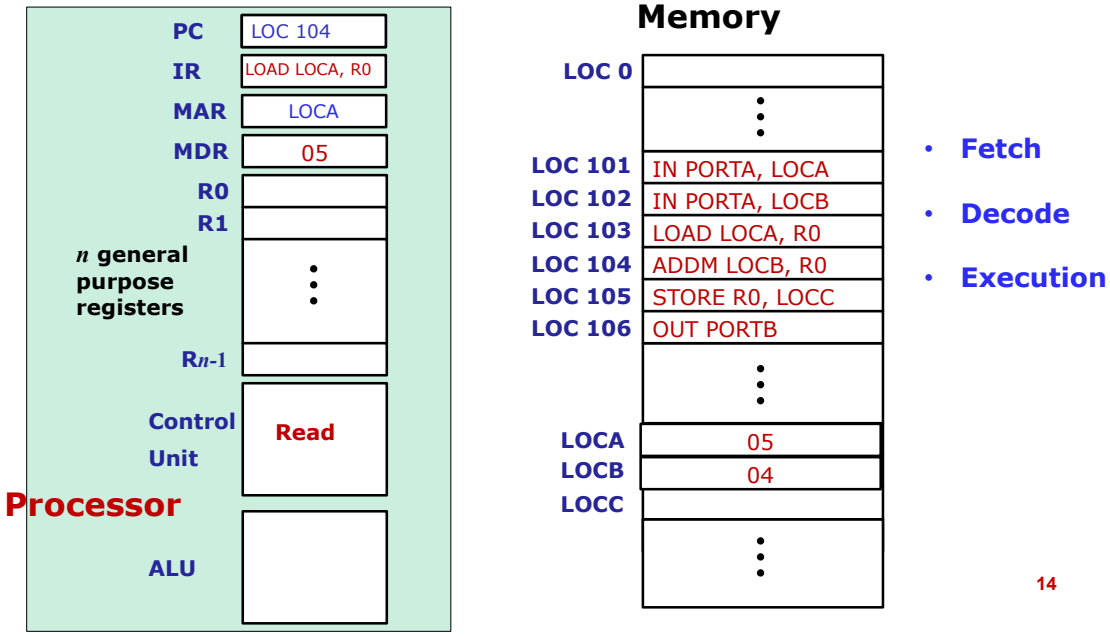
Execution of an Instruction, LOAD



- Fetch
- Decode
- Execution

13

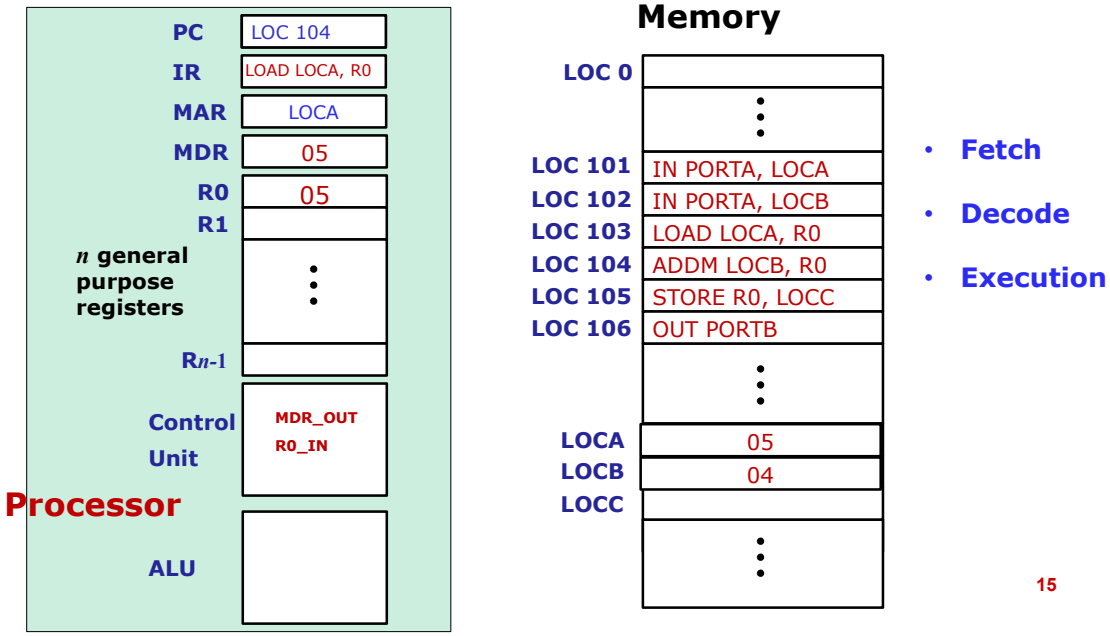
Execution of an Instruction, LOAD



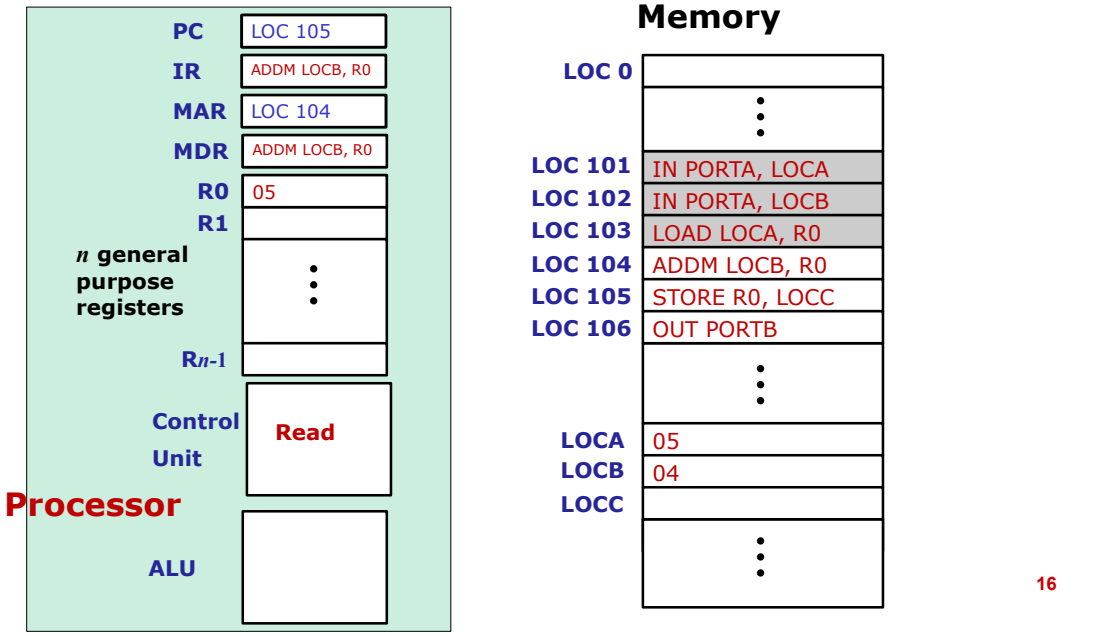
- Fetch
- Decode
- Execution

14

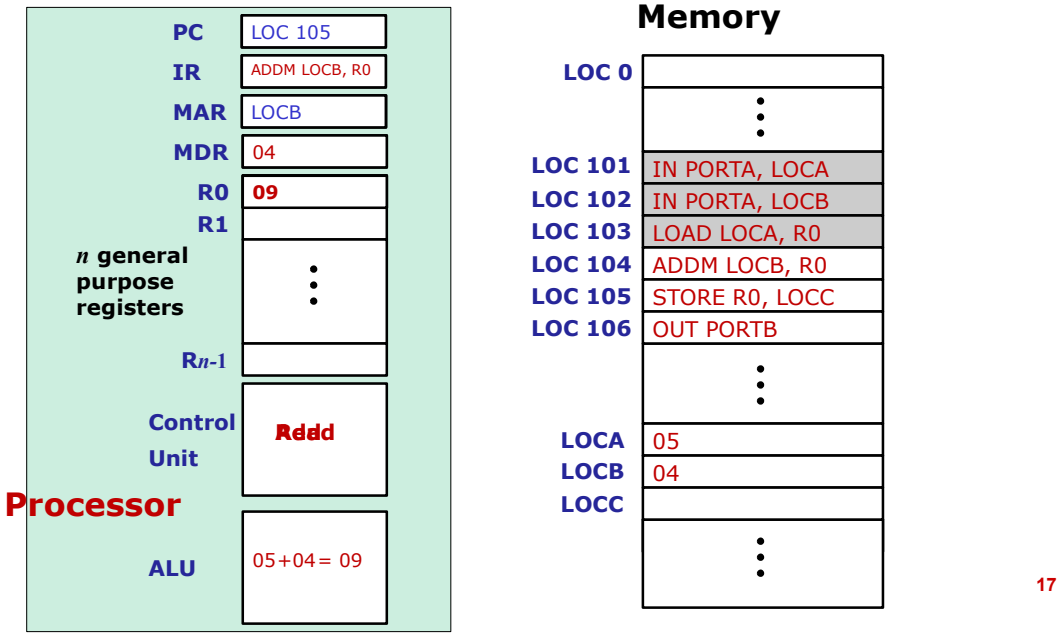
Execution of an Instruction, LOAD



Fetching and execution of an Instruction, ADDM



Fetching and execution of an Instruction, ADDM



17

17

To Summarize

- Functional units
- Basic operational concepts
  - Execution of an instruction
  - Important processor registers

18

18

References

- **Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002**

19

Thank You

20