Microprocessors

Tuba Ayhan
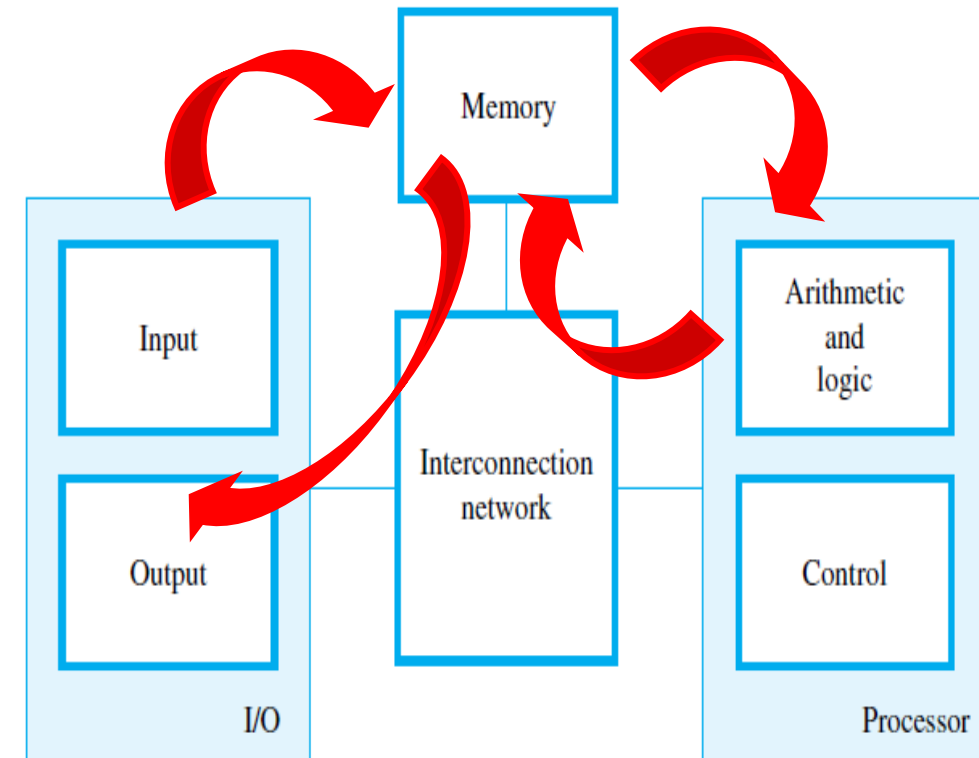
MEF University

Basic operational concepts

Computer Organization and Embedded Systems, Hamacher et. al

Operation of a computer – simplified

- Information is accepted: in the form of **programs** and **data** through an input unit and stores it in the memory.

- Information stored in the memory is fetched under program control into an arithmetic and logic unit, where it is processed.

- Processed information leaves the computer through an output unit.

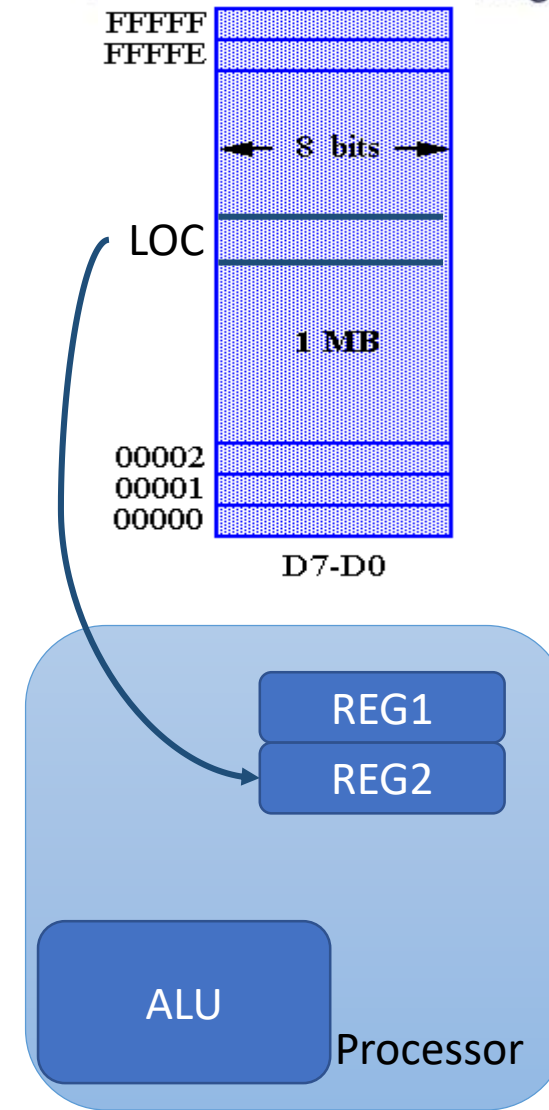- All activities in the computer are directed by the control unit.

Basic Operational Concepts

- To perform a given task, an appropriate program consisting of a list of instructions is stored in the memory.

- Individual instructions are brought from the memory into the processor, which executes the specified operations. Data to be used as instruction operands are also stored in the memory.
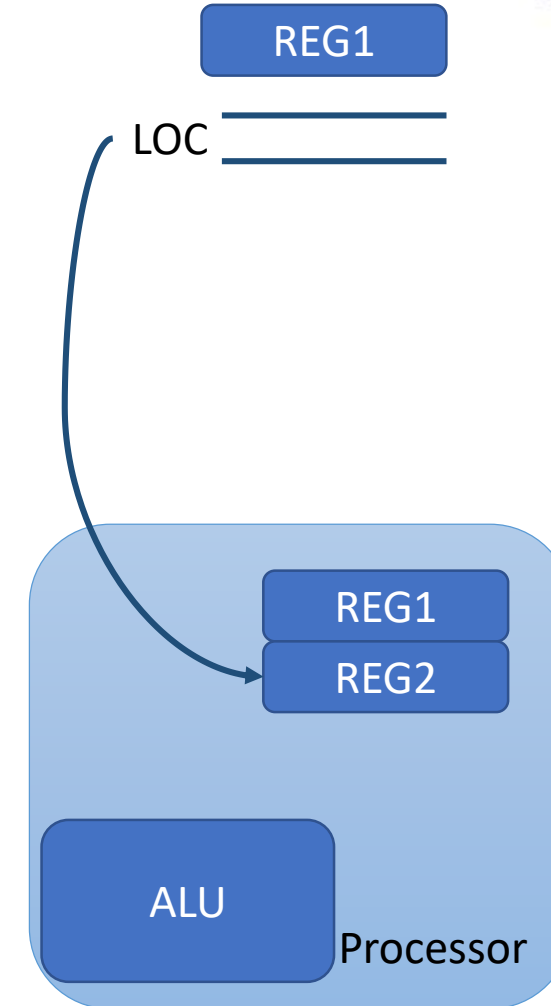

Load R2, LOC

1. read the contents of a memory location LOC

2. load them into processor register R2.

3. The original contents of location LOC are preserved, whereas those of register R2 are overwritten.
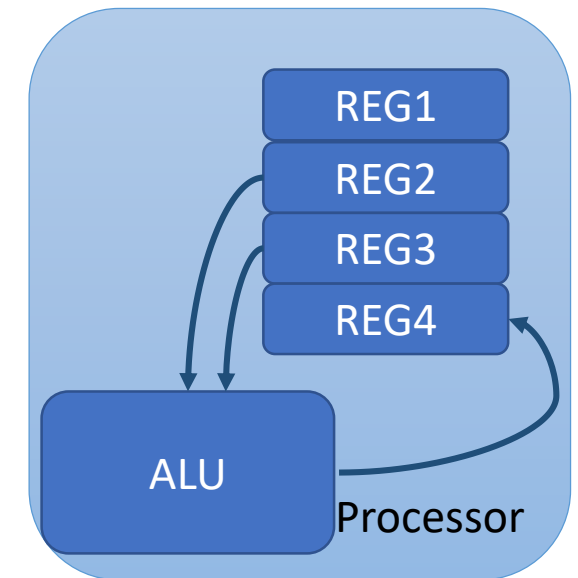
Instruction execution

1. Fetch the instruction from the memory into the processor.

2. The operation to be performed is determined by the control unit.

3. The operand at LOC is then fetched from the memory into the processor.

4. The operand is stored in register R2.

REG1

LOC

REG1

REG2

ALU

Processor

- After operands have been loaded from memory into processor registers, arithmetic or logic operations can be performed on them.

Add R4, R2, R3

- adds the contents of registers R2 and R3, then places their sum into register R4.

- Operands in R2 and R3 are not altered, but the previous value in R4 is overwritten by the sum.

REG1
REG2
REG3
REG4
ALU
Processor

- After completing the desired operations, the results are in processor registers. They can be transferred to the memory using instructions

Store R4, LOC

- The original contents of location LOC are overwritten, but those of R4 are preserved.

- NOTE: interconnects will be discussed later.

- The *instruction register* (IR) holds the instruction that is currently being executed.

- The *program counter* (PC) *points* to the next instruction that is to be fetched from the memory.

- *General-purpose registers* R0 through Rn−1, (processor registers) serve a variety of functions, including holding operands that have been loaded from the memory for processing.
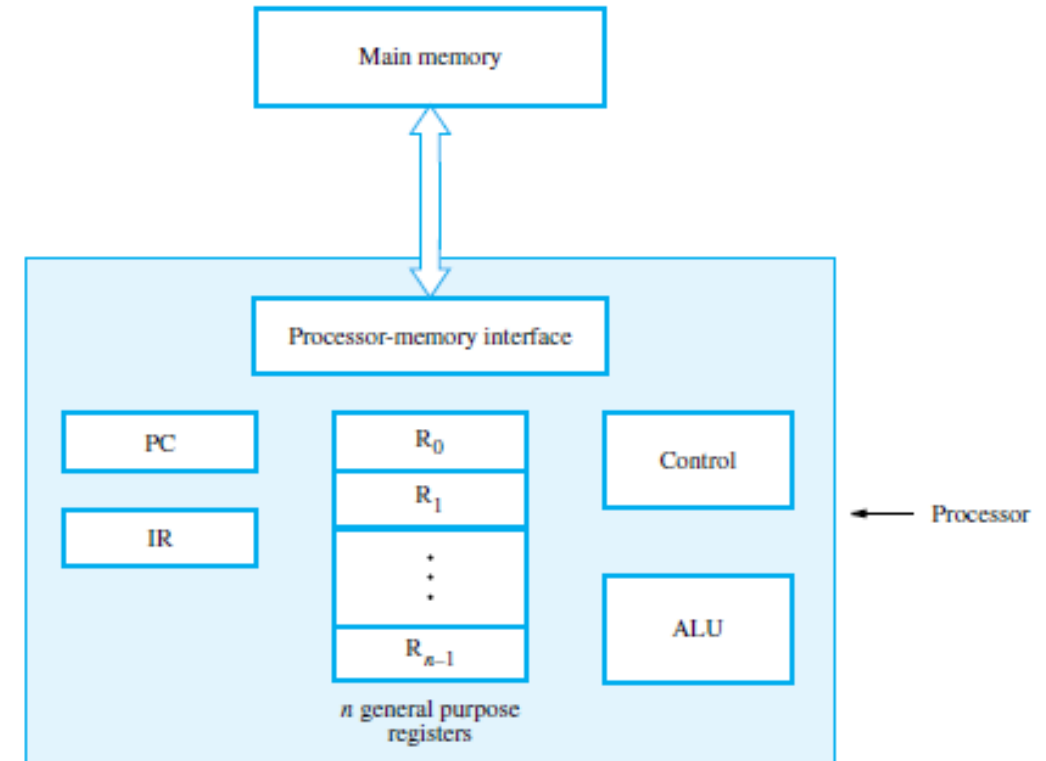

Figure 1.2   Connection between the processor and the main memory.

Processor-memory interface

- The *processor-memory interface* manages the transfer of data between the main memory and the processor.

- Read from the memory:
  1. The interface sends the address of that word to the memory along with a Read control signal.
  2. The interface waits for the word to be retrieved, then transfers it to the appropriate processor register.

- Write into the memory:
  1. The interface transfers both the address and the word to the memory along with a Write control signal.
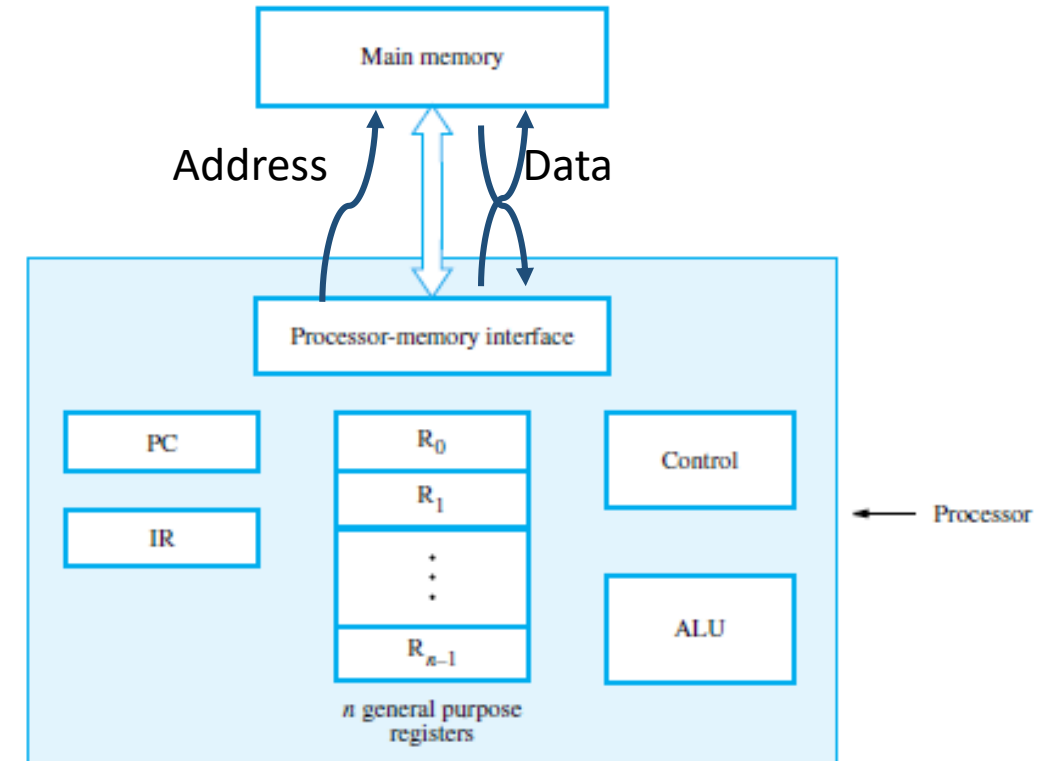
Main memory

Address   Data

Processor-memory interface

PC R_0 Control
            R_1
IR $\vdots$ Processor
            R_{n-1} ALU

n general purpose registers

Figure 1.2   Connection between the processor and the main memory.

Reading program

- Execution of the program begins when the PC is set to point to the first instruction of the program.

- The contents of the PC are transferred to the memory along with a Read control signal.

- When the addressed word (the first instruction of the program) has been fetched from the memory it is loaded into register IR. At this point, the instruction is ready to be interpreted and executed.
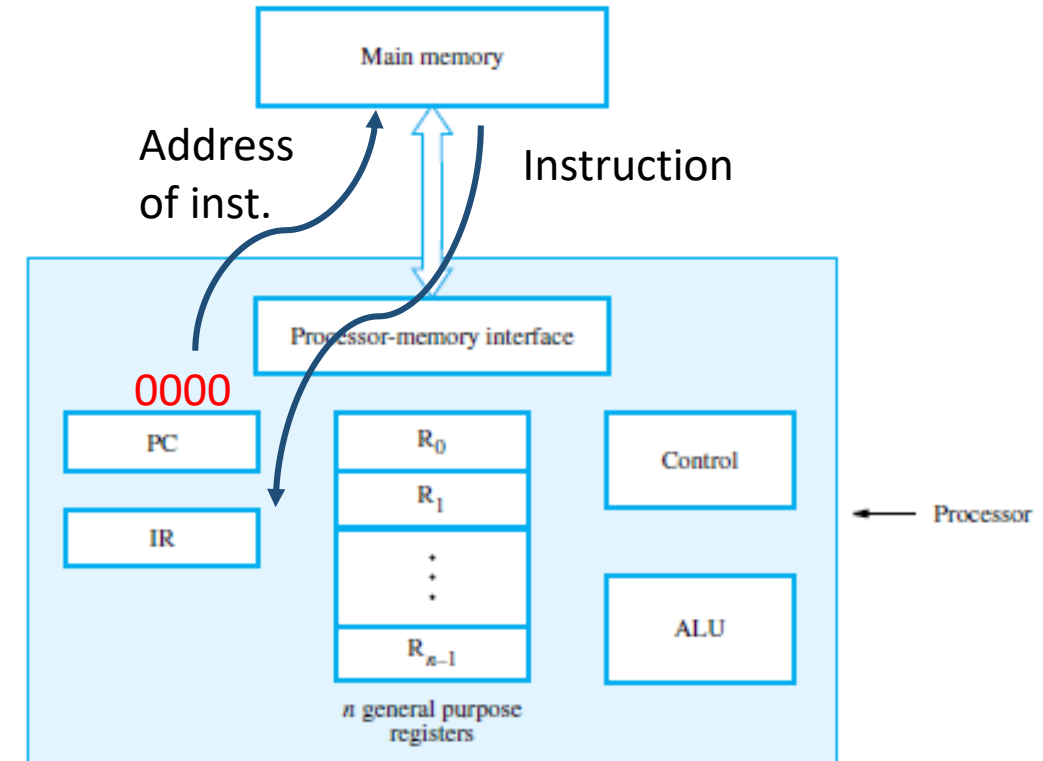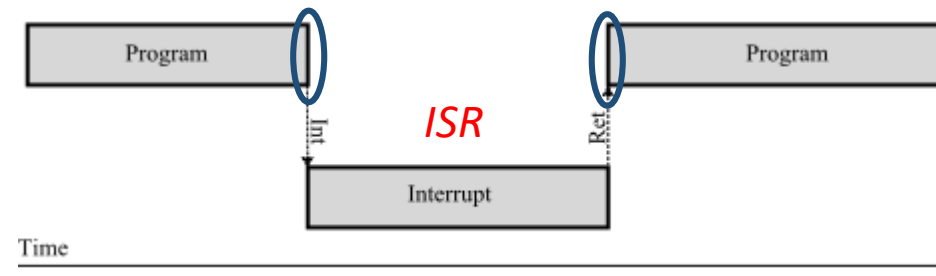
Main memory

Address of inst.

Instruction

0000

Processor-memory interface

PC

IR

R_0

R_1

$\vdots$

R_{n-1}

Control

ALU

Processor

n general purpose registers

Figure 1.2 Connection between the processor and the main memory.

Interrupt

- Normal execution of a program may be preempted if some device requires urgent service.

- The device raises an *interrupt* signal, which is a request for service by the processor. The processor provides the requested service by executing a program called an *interrupt-service routine (ISR)*.

- Its state must be saved in the memory before servicing the interrupt request. The contents of the PC, the contents of the general-purpose registers, and some control information are saved,generally.

- When the ISR is completed, the state of the processor is restored.

Problem

- List the steps needed to execute the machine instruction

  Load R2, LOC

- In terms of transfers between the components shown in Figure and some simple control commands. Assume that the address of the memory location containing this instruction is initially in register PC.
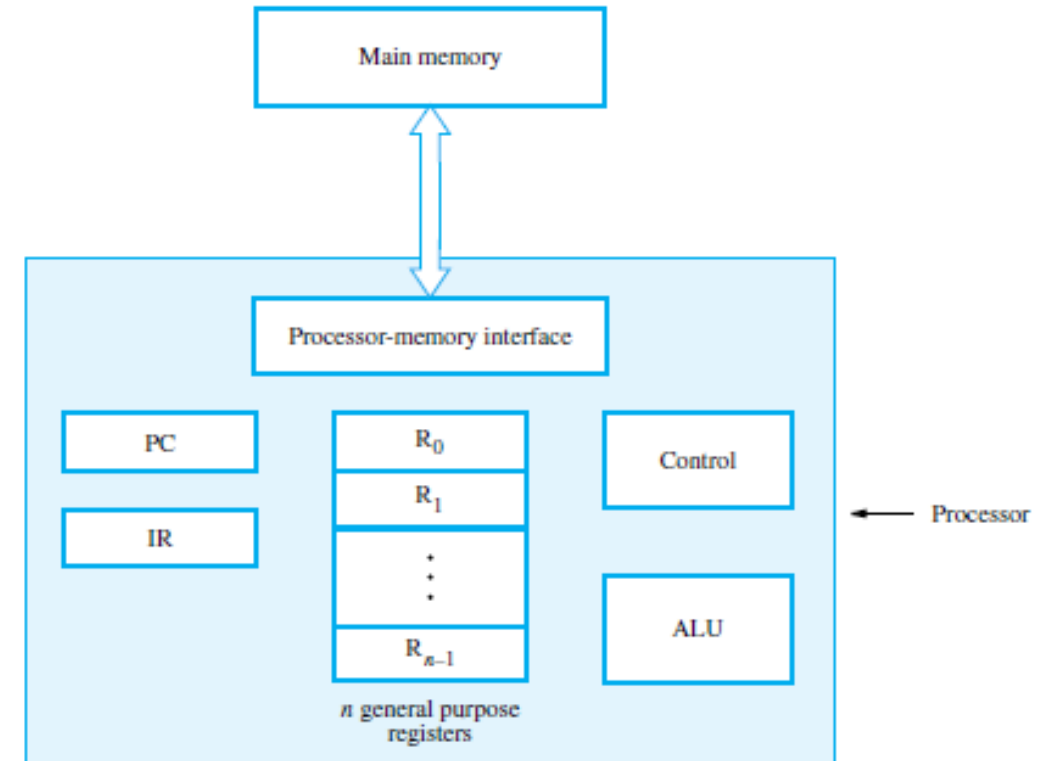

Figure 1.2   Connection between the processor and the main memory.

Solution

- Solution: The required steps are:
- Send the address of the instruction word from register PC to the memory and issue a Read control command.
- Wait until the requested word has been retrieved from the memory, then load it into register IR, where it is interpreted (decoded) by the control circuitry to determine the operation to be performed.
- Increment the contents of register PC to point to the next instruction in memory.
- Send the address value LOC from the instruction in register IR to the memory and issue a Read control command.
- Wait until the requested word has been retrieved from the memory, then load it into register R2.