Lecture 3+4

A Top-Level View of Computer Function and Interconnection

- At a top level, a computer consists of CPU (central processing unit), memory, and I/O components, with one or more modules of each type.
- These components are interconnected in some fashion to **achieve the basic function of the computer**, which is to execute programs.
- Thus, at a top level, we can characterize a computer system by describing:

(1) The external behavior of each component, that is, the data and control signals that it exchanges with other components.

(2) The interconnection structure and the **controls required** to manage the use of the interconnection structure.

top-level view of structure and function is important to understanding:

1- The nature of a computer.

2- The increasingly complex issues of performance evaluation of a computer.

3- alternate pathways, the magnitude of system failures if a component fails, and the ease of adding performance enhancements.

virtually all contemporary computer designs are based on concepts developed by John von Neumann at the Institute for Advanced Studies, Princeton. (IAS) Such a design is referred to as the von Neumann.

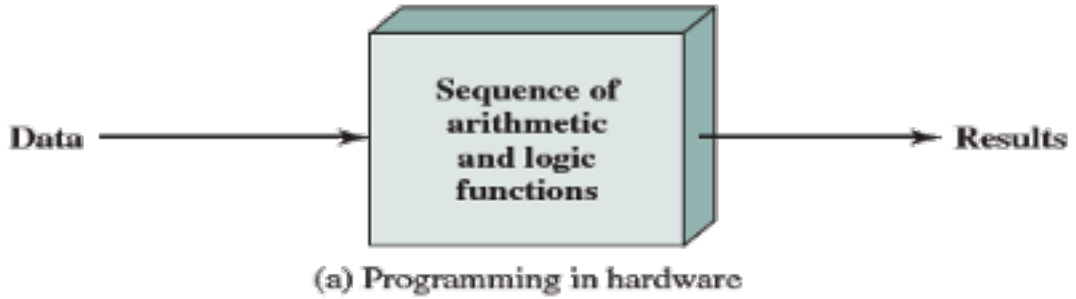
Architecture is based on three key concepts:

- 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.

- There is a small set of basic logic components that can be combined in various ways to store binary data and perform arithmetic and logical operations .
- If there is a particular computation to be performed, a configuration of logic components designed specifically for that computation could be constructed.
- We can think of the process of connecting the various components in the desired configuration as a form of programming. The resulting "program" is in the form of hardware and is termed a *hardwired program*.

- In the original case of customized hardware, the system accepts data and produces results
- (Figure 3.a).

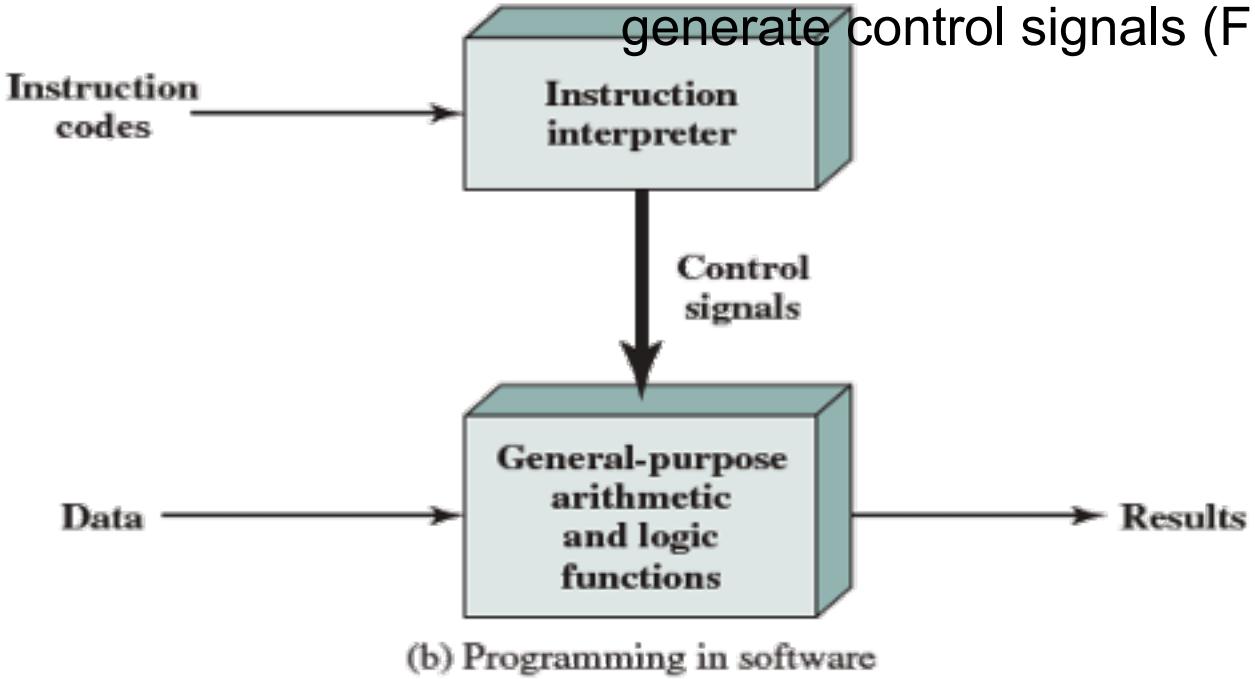
- With general-purpose hardware, the system accepts data and control signals and produces results. Thus, instead of rewiring the hardware for each new program, the **programmer** merely needs to supply a new set of control signals.



(a) Programming in hardware

Figure 3 Hardware and Software Approaches

The entire program is actually a sequence of steps. At each step, some arithmetic or logical operation is performed on some data. For each step, a new set of control signals is needed. Let us provide a unique code for each possible set of control signals, and let us add to the general-purpose hardware a segment that can accept a code and generate control signals (Figure 3.1b).



(b) Programming in software

Instead of rewiring the hardware for each new program, all we need to do is provide **a new sequence of codes. Each code is**, in effect, an instruction, and part of the hardware interprets each instruction and generates control signals. To distinguish this new method of programming, a sequence of codes or instructions is called software

- Figure 3.1b indicates two major components of the system: an instruction interpreter and a module of general-purpose arithmetic and logic functions. These two constitute the CPU.
- Several other components are needed to yield a functioning computer:

1- Data and instructions must be put into the system by

- some sort of input module: contains basic components for accepting data and instructions in some form and converting them into an internal form of signals

2- A means of reporting results is needed, and this is in the form of an output module.

- Taken together, these are referred to as *I/O*

3- Memory component – for place to temporarily store both instructions and data.

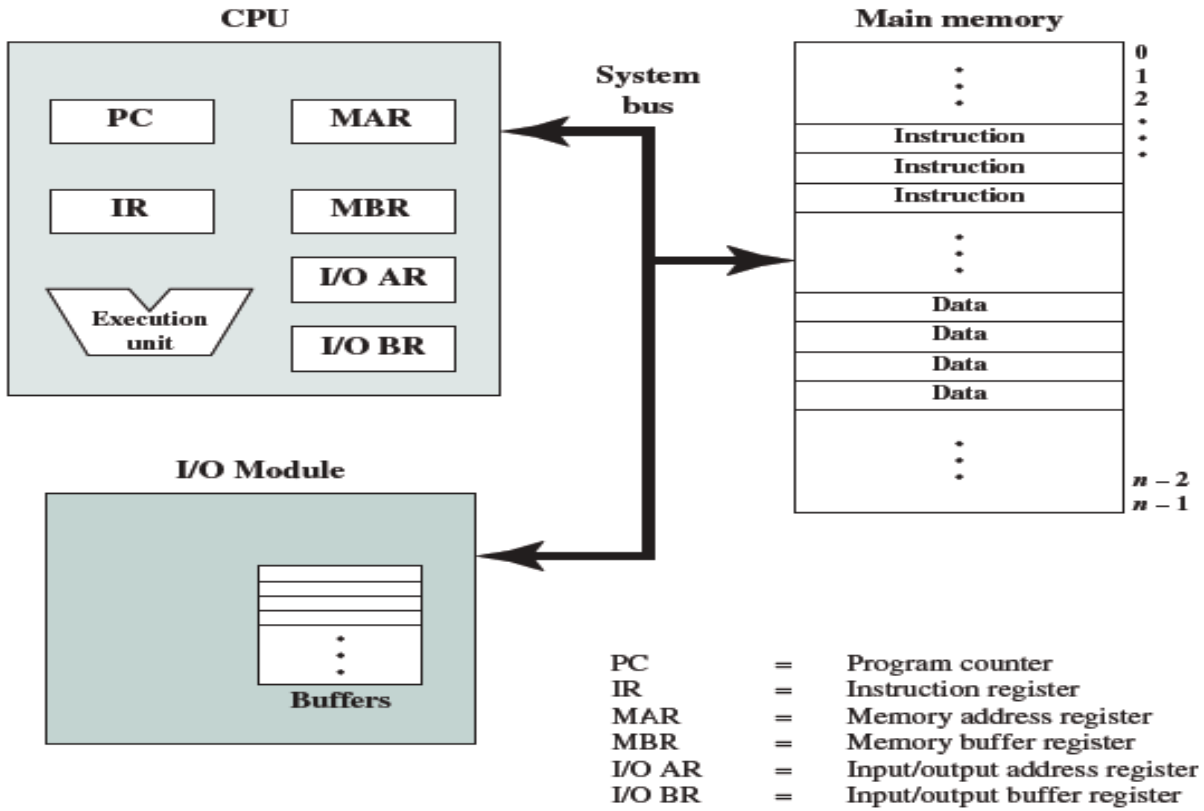


Figure 3.2 Computer Components: Top-Level View

Computer Function :

The basic function performed by a computer is program execution. The central processing unit (CPU) does the actual work by performing the instructions specified by the program. A program consisting of a set of instructions is stored in memory.

- The basic function performed by a computer is program execution.
- The central processing unit (CPU) does the actual work by performing the instructions specified by the program.
- A program consisting of a set of instructions is stored in memory.
- The processing required for a single instruction is called an instruction cycle (Figure 10).
- Each instruction cycle consists of two parts, called the *fetch cycle* and the *execute cycle*.
- The process halts only if the machine is turned off, some sort of error occurs, or a program instruction that halts the computer is encountered.

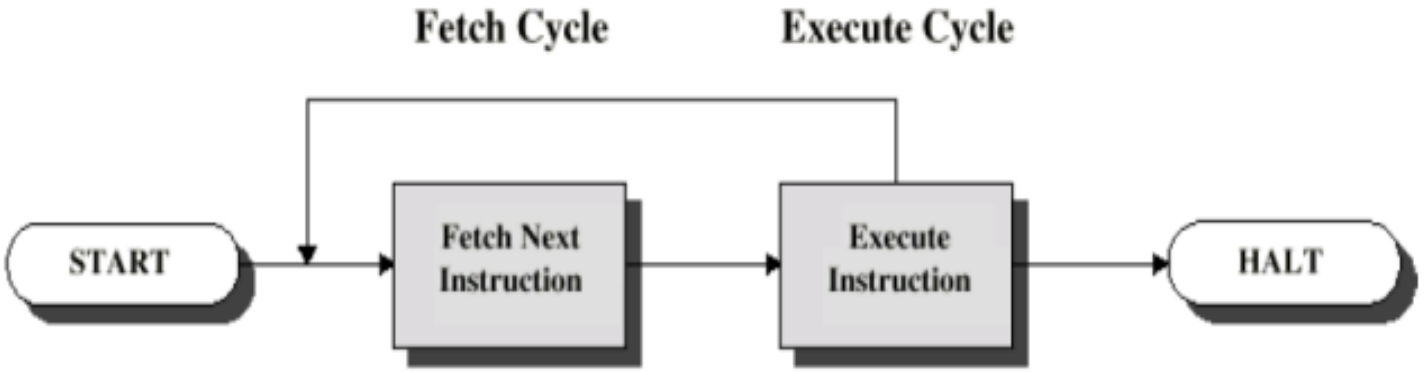


Figure 10 Basic instruction cycle

The fetch and Execute Cycle.

Fetch cycle consist of these operations:

- **Program Counter (PC) holds address of next instruction to fetch.**
- **Processor fetches instruction from memory location pointed to by PC.**
- **Increment PC.**
- **Instruction loaded into Instruction Register (IR).**
- **Processor interprets instruction and performs required actions.**

Execute cycle consist of these operations:

- **Processor-memory: Data transfer between CPU and main memory.**
- **Processor I/O: Data transfer between CPU and I/O module.**
- **Data processing: Some arithmetic or logical operation on data.**
- **Control: Alteration of sequence of operations, e.g. jump.**
- **Combination of above.**

- Consider a simple example using a hypothetical machine that includes the characteristics listed in Figure 3.4. The processor contains a single data register, called an accumulator (AC) , Op-code (operational code).

(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

Figure 3.4 Characteristics of a Hypothetical Machine

- Both instructions and data are 16 bits long. Thus, it is convenient to organize memory using 16-bit words. The instruction format provides 4 bits for the op-code, so that there can be as many as 24 = 16 different op-codes, and up to 212 = 4096 (4K) words of memory can be directly addressed.

- The program fragment shown adds the contents of the memory word at address 940 to the contents of the memory word at address 941 and stores the result in the latter location.

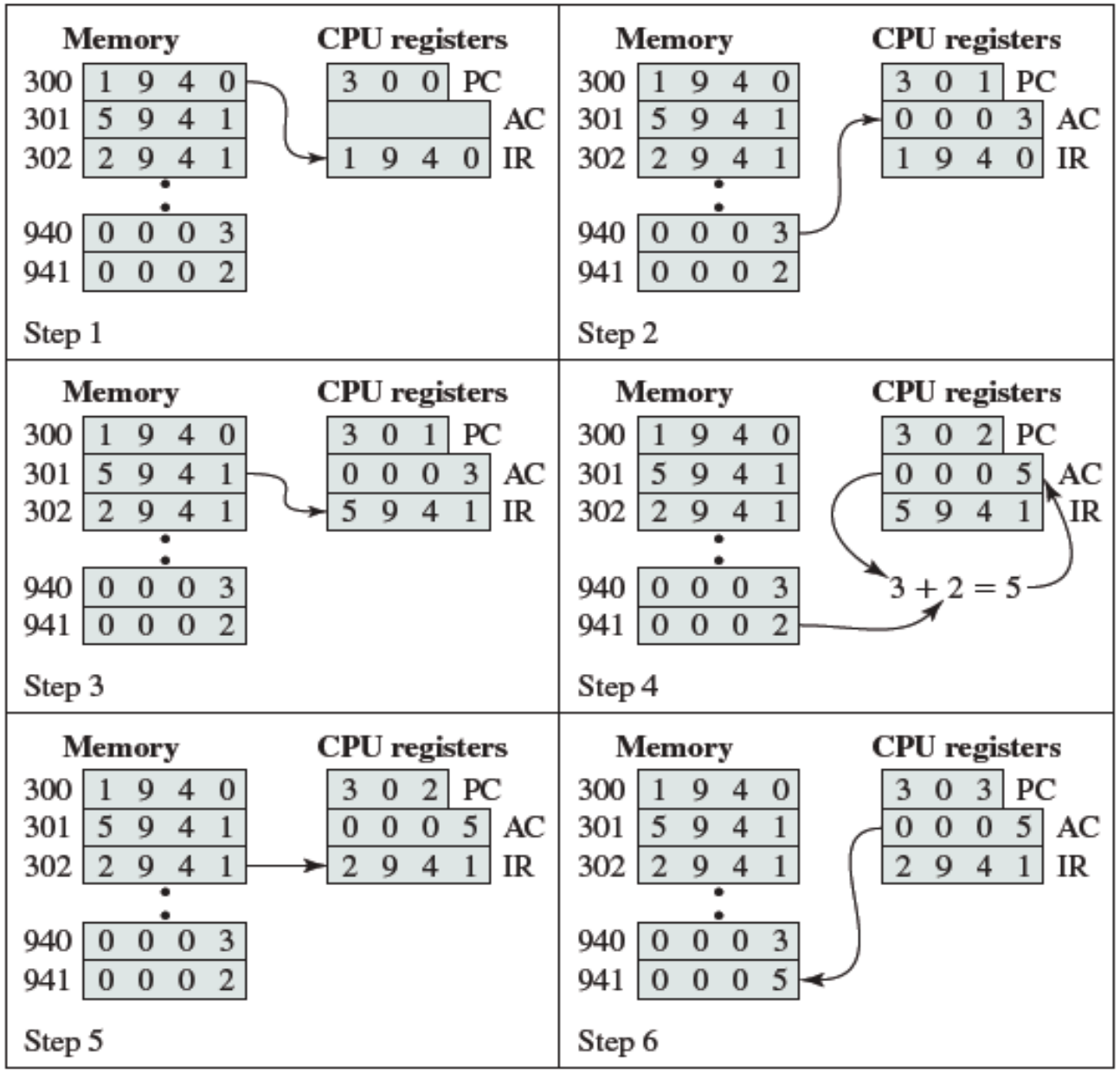


Figure 3.5 Example of Program Execution (contents of memory and registers in hexadecimal)

described as three fetch and three execute cycles, are required:

1. The PC contains 300, the address of the first instruction. This instruction (the value 1940 in hexadecimal) is loaded into the instruction register IR, and the PC is incremented. Note that this process involves the use of a memory address register and a memory buffer register. For simplicity, these intermediate registers are ignored.

2. The first 4 bits (first hexadecimal digit) in the IR indicate that the AC is to be loaded. The remaining 12 bits (three hexadecimal digits) specify the address (940) from which data are to be loaded.

3. The next instruction (5941) is fetched from location 301, and the PC is incremented.

4. The old contents of the AC and the contents of location 941 are added, and the result is stored in the AC.

5. The next instruction (2941) is fetched from location 302, and the PC is incremented.

6. The contents of the AC are stored in location 941.

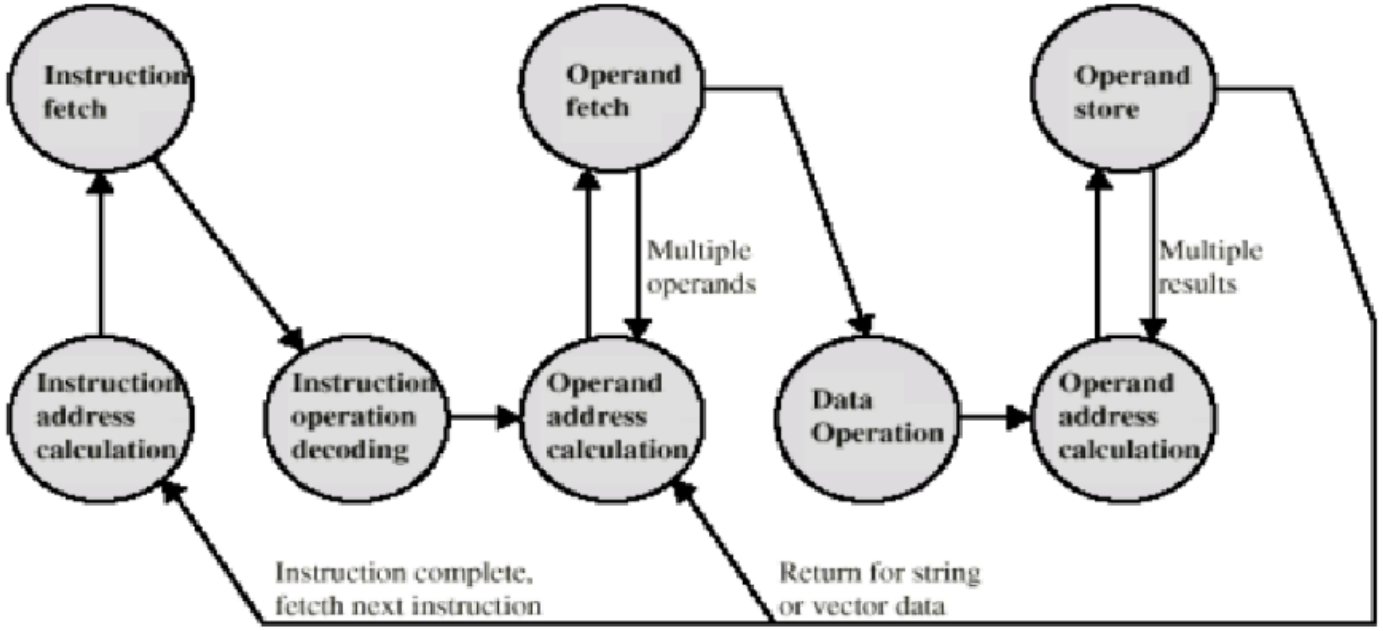


Figure 13 Instruction cycle state diagram.

A single instruction cycle with the following steps occurs:

- Fetch the ADD instruction.
- Read the contents of memory location A into the processor.
- Read the contents of memory location B into the processor. In order that the contents of A are not lost, the processor must have at least two registers for storing memory values, rather than a single accumulator.
- Add the two values.
- Write the result from the processor to memory location A.