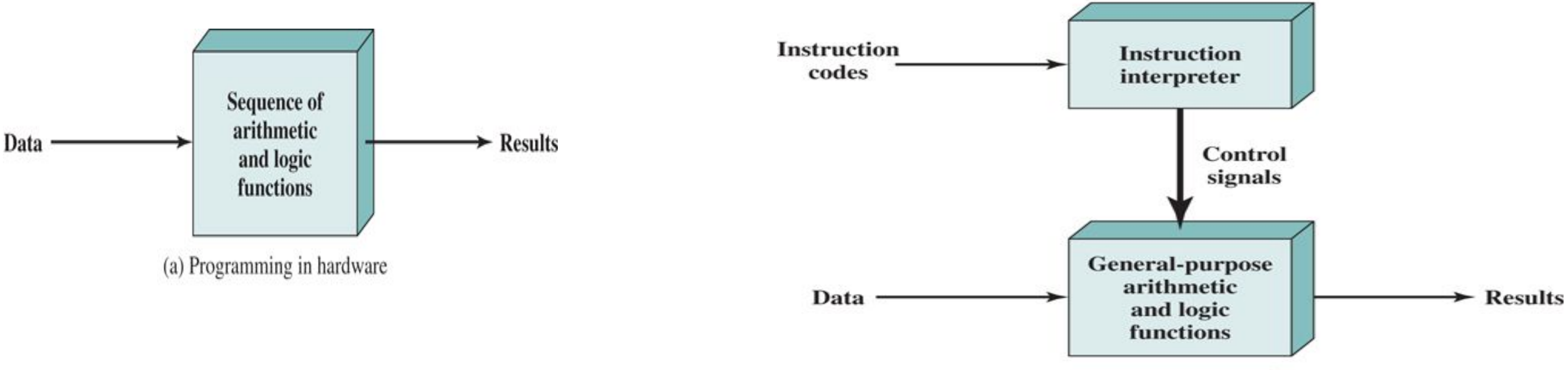
# 20CS2009
# Computer Architecture
# Lecture #2

# 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

## Hardware and Software Approaches

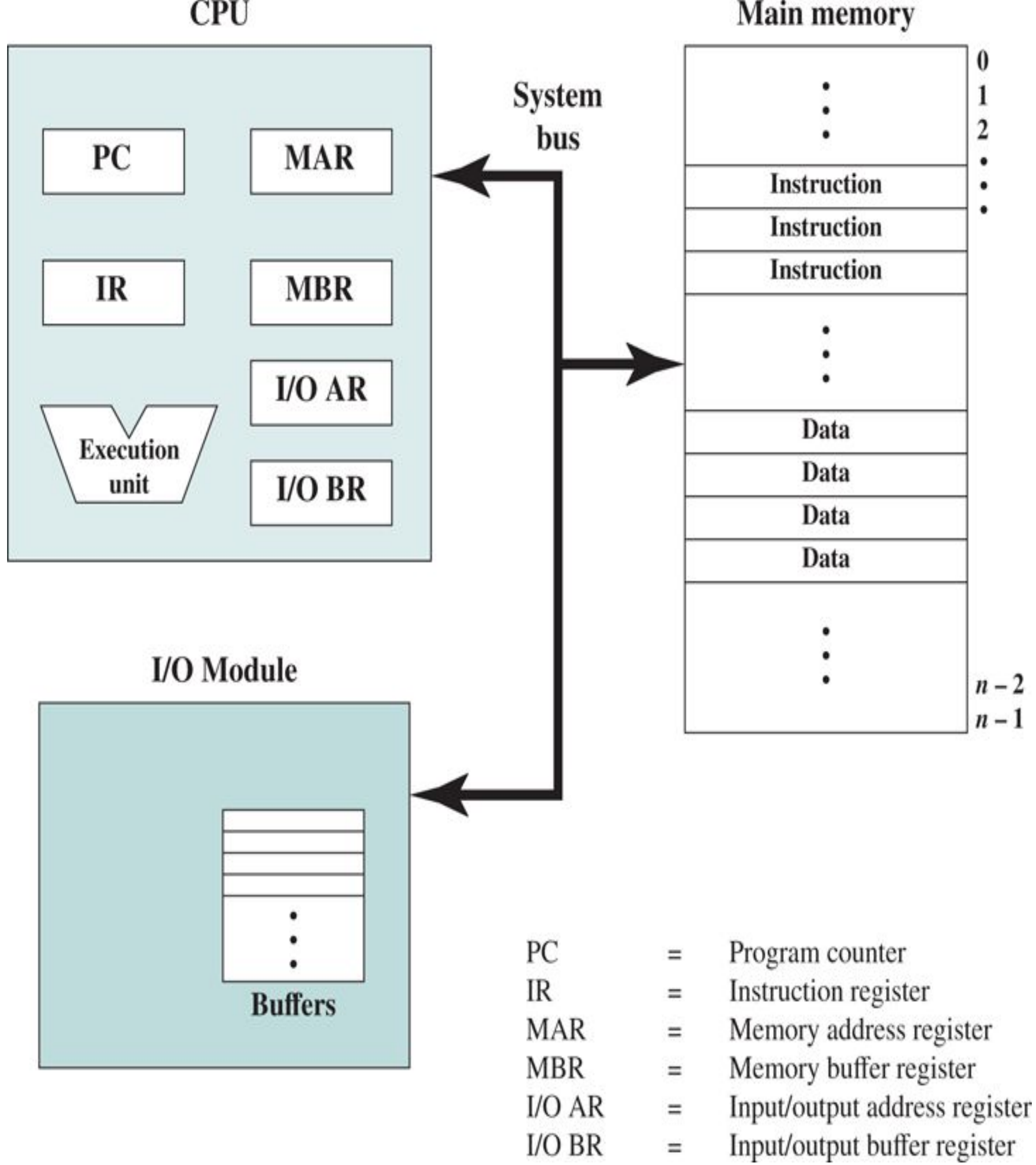

Data
Sequence of arithmetic and logic functions
Results
Instruction codes
Instruction interpreter
Control signals
Data
General-purpose arithmetic and logic functions
Results

(a) Programming in hardware

(b) Programming in software

## Top-level components and interactions among them

- A memory address register (MAR) , which specifies the address in memory for the **next read or write**, and a memory buffer register (MBR) , which contains the **data to be written into memory** or receives the data read from memory
- I/O **address register** (I/OAR) specifies a particular I/O device. An I/O buffer register (I/OBR) is used for the **exchange of data between** an I/O module and the CPU.

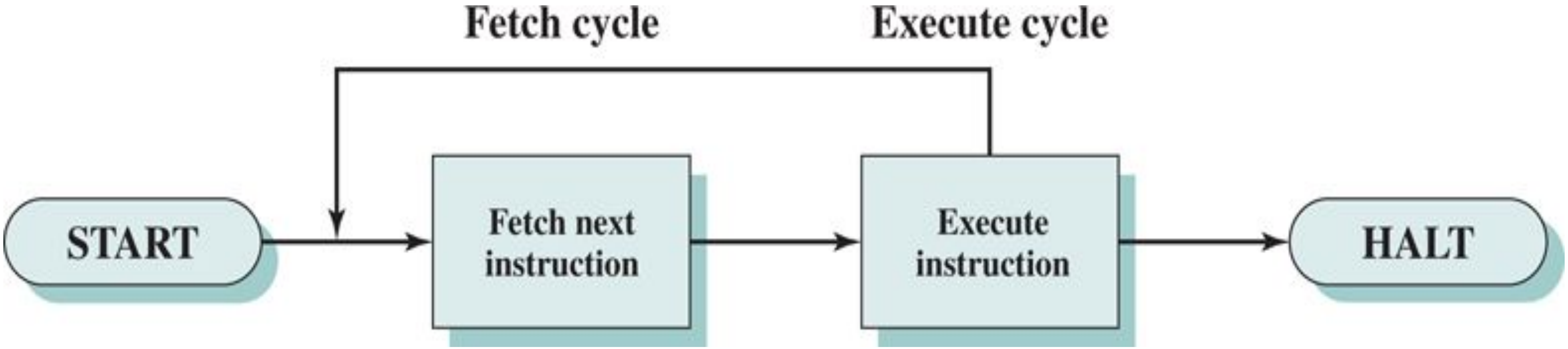


# Computer Function

- The basic function performed by a computer is **execution of a program**, which consists of a **set of instructions** stored in memory.
- The processor does the actual work by **executing instructions specified in the program**
- The processor **reads (fetches) instructions** from memory one at a time, then executes each instruction.
- Program execution consists of repeating the process of **instruction fetch and instruction execution.**
- The processing required for a **single instruction** is **called an instruction cycle**
- The two steps are referred
    - **Fetch Cycle**
    - **Execute Cycle**

## Basic Instruction Cycle:

- At the beginning of each instruction cycle, the processor fetches an instruction from memory
- A register called the program counter (PC) holds the address of the instruction to be fetched next.



- The **fetched instruction** is loaded into a register in the processor known as the **instruction register (IR).**
- The instruction **contains bits that specify the action the processor is to take**. The **processor interprets the instruction** and **performs the required action(CU)**.

**In general, these actions fall into four categories:**

- **Processor-memory**: Data may be transferred from **processor to memory** or from **memory to processor**.
- **Processor-I/O:** Data may be **transferred to or from a peripheral device** by transferring between the **processor and an I/O module**.
- **Data processing:** The processor may **perform some arithmetic or logic** operation on data.
- **Control: An instruction** may specify that the sequence of execution be altered.

## Understanding Memory Operations

- Bit - 0 / 1 – Bit ( stored in one Memory Cell)
- Byte - 8 Bits = 1 Byte ( 8 Memory Cell’s)
- Word - The word length of the computer has evolved from 8, 16, 24, 32 to 64 bits ( may depends on Computer System)

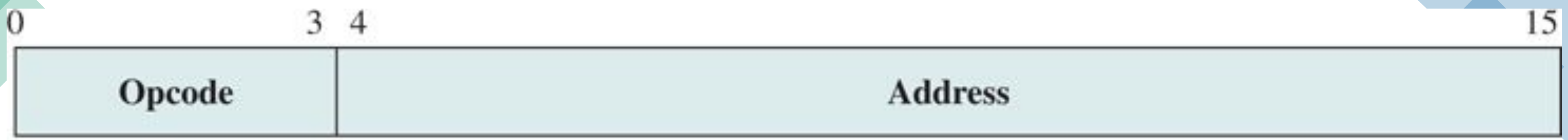
## General-purpose computers nowadays have 32 to 64 bits

- 1024 bytes = 1 KB can be represented by $2^{10}$
- $2^{10}$ = 1024 = 1K (Kilobyte)
  $2^{20}$ = 1,048,576 = 1M (Megabyte)
  $2^{30}$ = 1073741824 = 1G (Gigabyte)
  $2^{40}$ = 1.0995116e+12 = 1T (Terabyte)

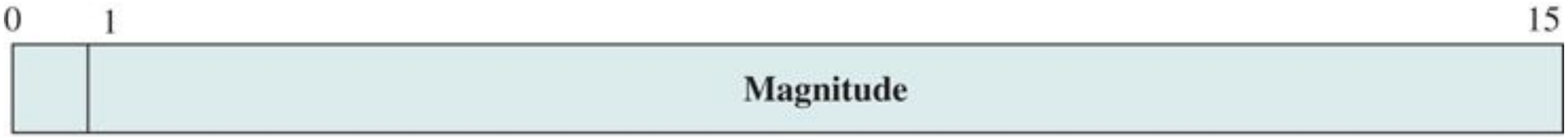
- The processor contains a single data register, called an accumulator (AC).

- 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 opcode
- so that there can be as many as $2^4$ = 16 different opcodes.
- $2^{12}$ = 4096
- (4K) words of memory can be directly addressed



(a) Instruction format



(b) Integer format

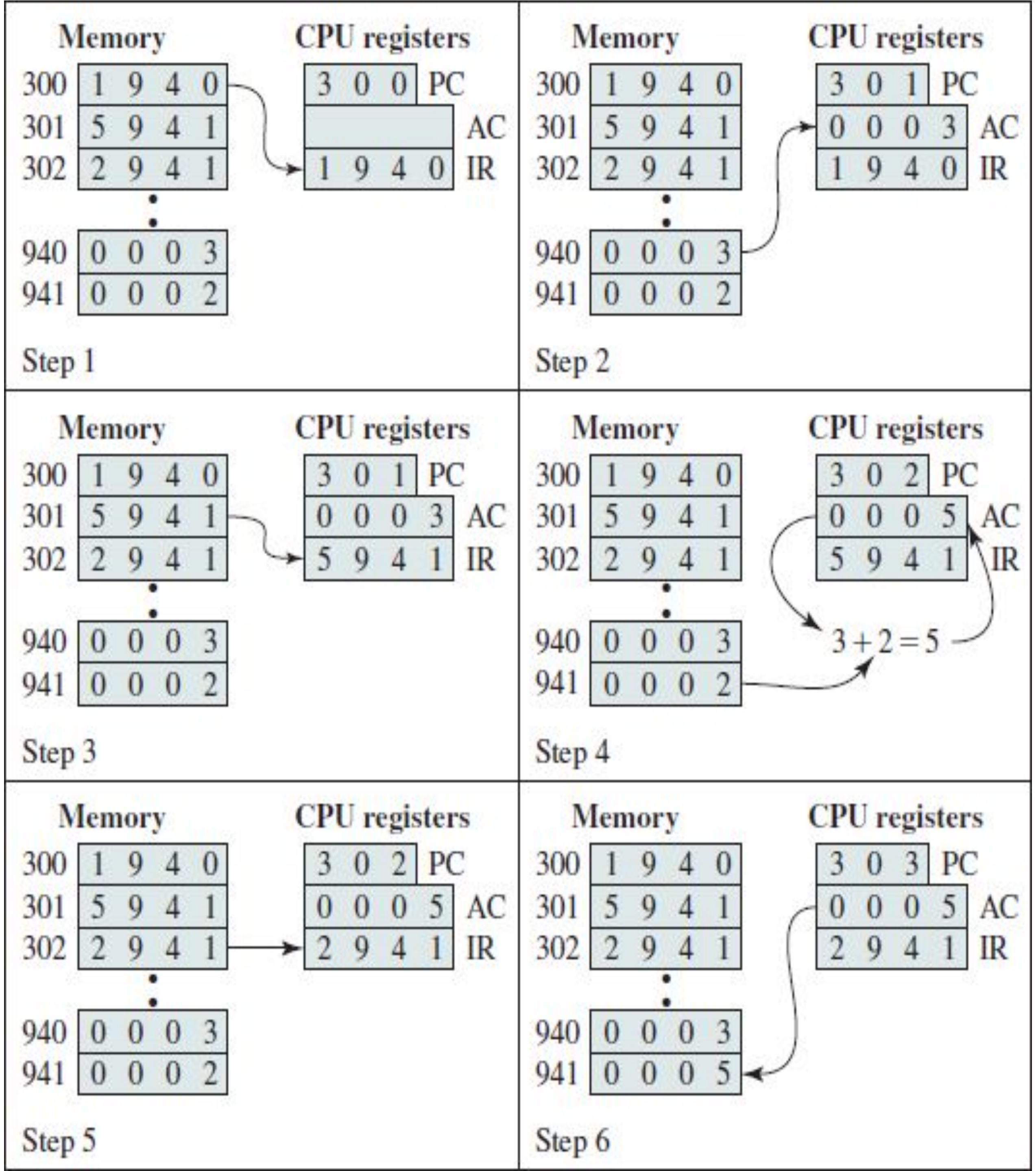
Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

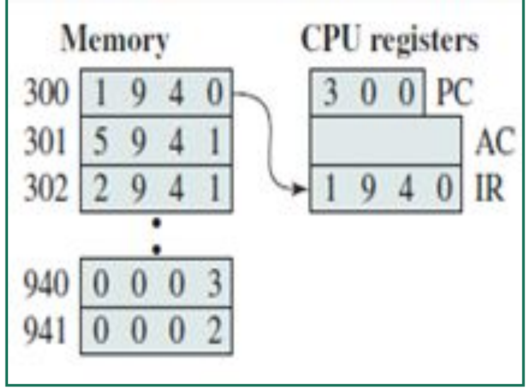
0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

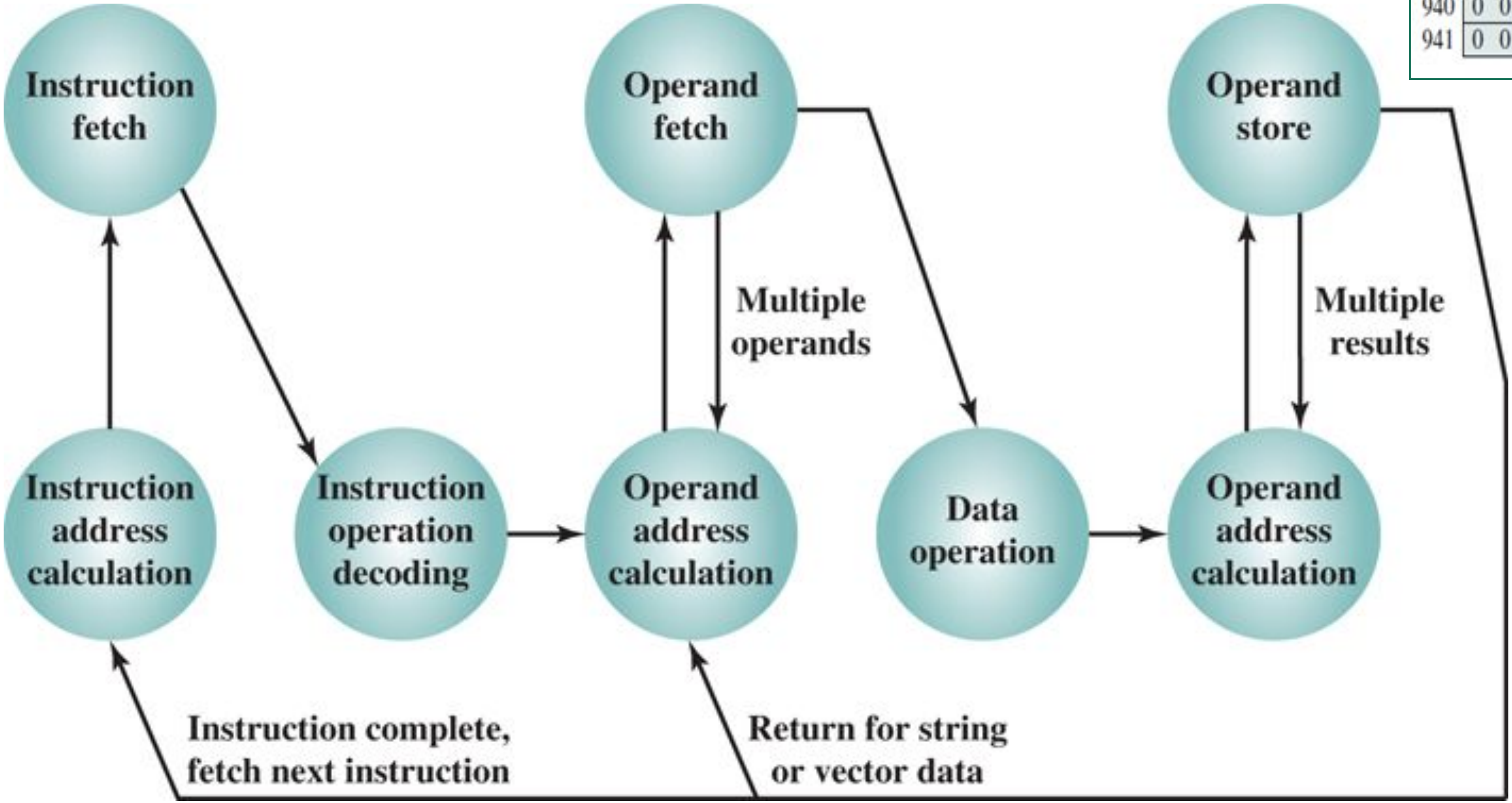
(d) Partial list of opcodes

- The Diagram shows the partial program execution, showing the relevant portions of memory and processor registers
- 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.
- Three instructions, which can be described as three fetch and three execute cycles, are required



- Instruction Cycle State Diagram




Instruction fetch
Operand fetch
Operand store
Multiple operands
Multiple results
Instruction address calculation
Instruction operation decoding
Operand address calculation
Data operation
Operand address calculation
Instruction complete, fetch next instruction
Return for string or vector data

- 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.
- **Instruction address calculation (IAC):** Determine the address of the next instruction to be executed.
- **Instruction fetch (IF):** Read instruction from its memory location into the processor.
- **Instruction operation decoding (ID):** Analyse instruction to determine type of operation to be performed and operand(s) to be used.

- **Operand address calculation (OAC):** If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.
- **Operand fetch (OF):** Fetch the operand from memory or read it in from I/O.
- **Data operation (DO):** Perform the operation indicated in the instruction.
- **Operand store (OS):** Write the result into memory or output  to I/O