Microcomputer Systems

Outline

- Components of microcomputer system
- Executing an instruction
- I/O devices
- Programming languages

Components of a Microcomputer System

- Memory
  - Information processed by the computer is stored in its memory
- The CPU
  - In a microcomputer the CPU is a single chip processor known as microprocessor
- I/O Ports
  - System board or mother board contains expansion slots which are connectors for additional circuits boards called add in boards. I/O circuits are usually on these boards

Memory

- A memory circuit element can store one bit of data
- Organized into groups that can store eight bits of data
- String of eight bits called a byte
- Each memory byte is identified by address
- The first memory byte has address 0
- The data stored in a memory byte called its contents or values
- The address of a memory byte is fixed and different from any other addresses whereas contents are not fixed
- The contents of memory byte are always eight bits but address depends on the processor. For example some assign 20 bits address whereas some assign 24 bit address

Memory

- Bit Position
  - The positions are numbered from right to left starting with 0
  - Low byte comes from memory byte with lower address and high byte comes from memory byte with higher address

Byte

Word

- Operations
  - Processor can perform two operations on memory: read(fetch) and write(store)
  - In read processor gets a copy of data and the contents of that location is unchanged whereas in write the data becomes the new content of the location

Memory

- RAM and ROM
  - Two kinds of memory circuits: RAM(Random Access Memory) and ROM(Read Only Memory)
  - RAM locations can be read and write but ROM locations only can be read
  - Program instructions and data normally loaded into RAM
  - System programs are stored in ROM
  - RAM memory lost when the power is off but ROM circuits retain their values when the power is off
- Buses
  - Processor communicates with memory and I/O devices by using signals that travel along a set of wires called buses
  - Three kinds of buses: address bus, data bus and control bus

Memory Devices

- Random-Access Memory (RAM)
  - DRAM = Dynamic RAM
    - 1-Transistor cell + trench capacitor
    - Dense but slow, must be refreshed
    - Typical choice for main memory
  - SRAM: Static RAM
    - 6-Transistor cell, faster but less dense than DRAM
    - Typical choice for cache memory
- Read-Only-Memory(ROM)
  - Many types: ROM, EPROM, EEPROM, and FLASH memory can be erased electrically in blocks
- Cache
  - A very fast type of RAM that is used to store information that is most frequently or recently used by the computer
  - Recent computers have 2-levels of cache; the first level is faster but smaller in size (usually called internal cache), and the second level is slower but larger in size (external cache).

Memory

- Suppose a processor uses 20 bits for an address. How many memory bytes can be accessed?
- The number of memory bytes will be $2^{20} = 1{,}048{,}576 = 1\text{MB}$

CPU

- It controls computer by executing programs stored in the memory
- The instructions performed by a CPU is known as instruction set
- Instruction set for each CPU is unique
- 8086 microprocessor has two main components: Execution Unit and Bus Interface Unit
- EU and BIU is connected by an internal bus
- When EU is executing an instruction the BIU fetches up to six bytes of the next instruction and places them in the instruction queue known as instruction prefetch

CPU

- Execution Unit
  - Used to execute instructions
  - Contains a circuit called Arithmetic and Logic Unit(ALU)
  - The data for the operations are stored in circuits called registers
  - Eight registers for storing data: AX, BX, CX, DX, SI, DI, BP and SP
  - It also contains FLAGS register whose individual bits reflect the result of a computation
- Bus Interface Unit
  - Facilitates communication between the EU and the memory or I/O circuits
  - Transmits addresses, data and control signals on the buses
  - Registers are CS, DS, ES, SS and IP holding address of memory locations
  - IP contains the address of the next instruction to be executed by the EU

I/O Ports

- I/O devices are connected to the computer through I/O circuits
- Each of these circuits contains several registers known as I/O ports
- I/O ports have addresses and connected to the bus system
- These addresses are known as I/O addresses and can only be used in input or output instructions
- Data to be input from an I/O device are sent to a port where they can be read by the CPU
- On output CPU writes data to an I/O port
- Two types of I/O ports: Serial and Parallel

I/O Ports

- Serial port
  - Transfers one bit at a time
  - Used for slower transfer such as keyboard
- Parallel port
  - Transfers 8 or 16 bits at a time
  - Requires more wiring connections
  - Used for faster data transfer such as disk drives

I/O Devices

- Magnetic Disk
  - Magnetic disks are used for permanent storage of programs and data
  - The device that reads and writes data on a disk is called disk drive
  - Floppy Disk
    - Light weight and portable
    - Easy to put away for safekeeping and use it on different computers.
    - Amount of data depends on type, ranging from 360KB-1.44MB (1KB)
  - Hard Disk
    - Enclosed in a hermetically sealed container that is non removable from computer called a fixed disk.
    - Can store more data than floppy disk. Typically 20, 40 to over 100MB.
    - A program can access information in a hard disk much faster than a floppy disk.

I/O Devices

- Keyboard
  - Allows the user to enter information in a computer.
  - It has keys of typewriters and a number of control and function keys
  - Has own microprocessor that sends coded signal to computer when a key is pressed or released
  - No direct contact between keyboard and display
- Display Monitor
  - Standard output device of the computer
  - Displayed information on the screen is generated by video adapter
  - Most adapters can generate both text characters and graphics images.
  - Some even display in color

I/O Devices

- Printers
  - Printers are slower than monitors but provide more permanent output
  - Printer outputs are known as hardcopies
  - Daisey wheel
    - The output is similar to that of a typewriter
  - Dot matrix
    - Prints characters composed of dots
    - Some can generate near-letter-quality printing
    - Print characters with different fonts as well as graphics
  - Laser printers
    - Print characters composed of dots
    - The resolution is high (300 dots per inch)
    - It is expensive

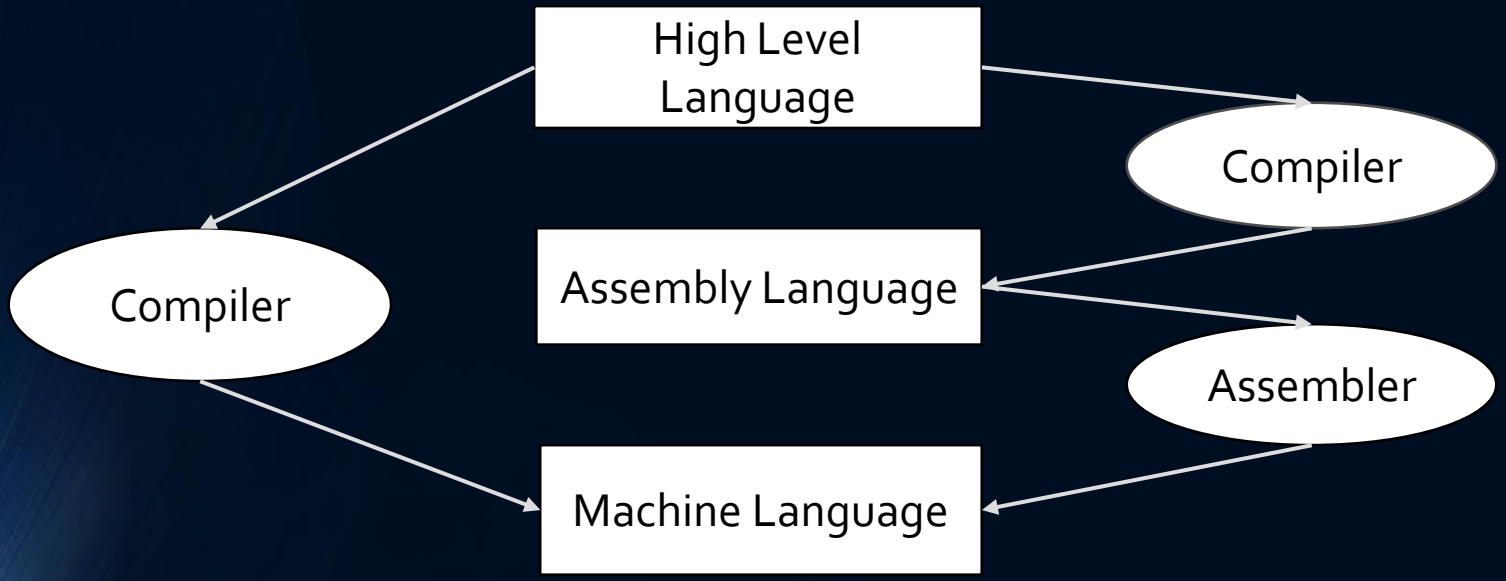
Instruction Execution

- Machine instructions have two parts: Opcode and Operands
- Opcode field which stands for operation code and it specifies the particular operation that is to be performed. Each operation has its unique opcode.
- Operands fields which specify where to get the source and destination operands for the operation specified by the opcode. The source/destination of operands can be a constant, the memory or one of the general-purpose registers.
- The steps of executing an instruction(the fetch-execution cycle) are:
  - Fetch
    1. Fetch an instruction from memory
    2. Decode the instruction to determine the operation
    3. Fetch data from memory if necessary
  - Execution
    1. Perform the operation on the data
    2. Store the result in memory if needed

Programming Languages

- Machine language
  - A CPU can only execute machine language instructions
  - Instructions consist of binary code: 1s and 0s
- Assembly language
  - A programming language that uses symbolic names to represent operations, registers and memory locations.
  - Readability of instructions is better than machine language
  - One-to-one correspondence with machine language instructions to machine code
  - Assemblers translates assembly code to machine code
- High Level Language
  - Compilers translate high-level programs to machine code directly or indirectly via an assembler

Compiler and Assembler



Advantages of High-Level Languages and Assembly Language

- High Level Language
  - Program development is faster as it is closer to natural language
  - Program maintenance is easier
  - Programs are portable as it can be used with little or no modifications on different machines
- Assembly Language
  - Assembly Language is close to machine language
  - It helps one to understand how the computer thinks

Mapping Between Assembly Language and High Level Language

- Translating High Level Language programs to machine language programs is not a one-to-one mapping
- A High Level Language instruction (usually called a statement) will be translated to one or more machine language instructions

Instruction Class	C	Assembly Language
Data Movement	A=5	MOV A,5

Arithmetic or Logic	B=A+5	MOV AX,A
		ADD AX,5
		MOV B,AX

Data Movement	goto LBL	JMP LBL

Why Learn Assembly Language?

- Accessibility to system hardware
  - Assembly Language is useful for implementing system software
  - Also useful for small embedded system applications
- Space and Time efficiency
  - Understanding sources of program inefficiency
  - Tuning program performance
  - Writing compact code
- Writing assembly programs gives the computer designer the needed deep understanding of the instruction set and how to design one
- To be able to write compilers for High Level Languages, we need to be expert with the machine language. Assembly programming provides this experience