MaAn

Lecture #2
7. Sep
Theory.

I/O Ports:

→ I/O device are conneted to the compiler through I/O circuits.
→ I/O circuits contains several registers called I/O ports.
→ I/O ports are used for data and control commands.
→ I/O ports have addresses known as I/O addresses.
→ I/O ports are connected to the bus system and can only be used in input or output instructions.
→ I/O ports functions as transfer points between the CPU and I/O devices.
→ Data to be input from an I/O device and sent to a port where they can be read by the CPU.
→ On output, the CPU writes data to an I/O port. The I/O circuit then transmits the data to the I/O device.

- There are two types of Ports:
  - → Serial
  - → Parallel

Serial and Parallel Ports:

- → A port that transfer 1 bit at a time between an I/O port and an I/O device is serial port.
- → Slow devices like the keyboard, always connect to a serial port.
- → A port that transfer 8 or 16 bit at a time between an I/O port and an I/O device is parallel port.
- → A parallel port requires more wiring connections
- → Fast devices, like the disk drive, always connect to a parallel port.
- → Some devices, like the printer can connect to either a serial or a parallel port.

Instruction Execution:

→ A machine instruction has two parts: An opcode and operands.
→ The **opcode** specifies the type of operation.
→ The **operands** are often memory addresses to the data to be operated on.
→ The CPU goes through the following steps to execute a machine instruction (the fetch-execute cycle).

Add ax [0]

Fetch:

→ Fetch an instruction from memory
→ Decode the instruction to determine the operation
→ Fetch data from memory if necessary

Execute:

→ Perform the operation on the data
→ Store the result in memory if needed.

Fetch the instruction:

→ To start the cycle, the BIU places a memory read request on the control bus and the address of the instruction on the address bus.

→ Memory responds by sending the contents of the location specified, namely, the instruction code just given over the data bus.

→ Because the instruction code is four bytes and the 8086 can only read a word at a time, this involves two read operations.

→ The CPU accepts the data and adds four to the IP so that the IP will contain the address of the next instruction.

Decode the instruction and Fetch Data from Memory:

→ On receiving the instruction, a decoder circuit in the EU decodes the instruction and determines that it is an ADD operation involving the word at address 0.

→ The EU informs the BIU to get the contents of memory word 0.
→ The BIU sends address 0 over the address bus and a memory read request is again sent over the control bus.
→ The contents of memory word 0 are sent back over the data bus to the EU and are placed in a holding register.

Perform the Operation and Store the Result:

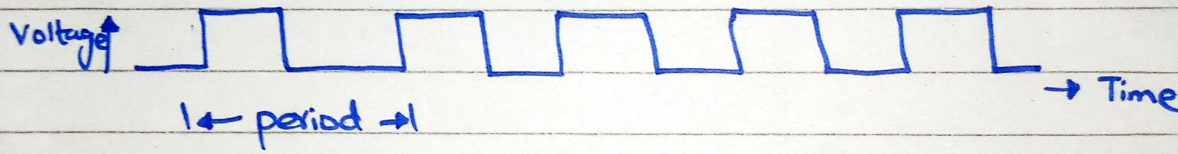
→ The contents of the holding register and the AX register are sent to the ALU circuit, which performs the required addition and holds the sum.
→ The EU directs the BIU to store the sum at address 0.
→ BIU sends out a memory write request over the control bus, the address 0 over the address bus, and the sum to be stored over the data bus.
→ The previous contents of memory word 0 are overwritten by the sum.

Timing:

→ A clock circuit controls the processor by generating a train of clock pulses.

→ Time interval between two pulses is known as a clock period.

→ Number of pulses per second is called the clock rate or clock speed, measured in megahertz (MHz)



→ Compiler circuits are activated by the clock pulses.

→ Circuits perform an operation only when a clock pulse is present.

→ Each step in the instruction fetch execution cycle requires one or more clock periods.

→ 8086 takes four clock periods to do a memory read

→ Multiplication operation may take more than seventy clock periods.