Notes of CSE-3523 (Microprocessor, Microcontroller and Embedded Systems)

Prepared by- Sorowar Mahabub, C201032, Cell: 01521564157

Microprocessor is a multipurpose, programmable register based electronic device which read binary instructions from memory, processes the input data as per instructions and provides output.

Microcontroller is a device that includes microprocessor, memory and input/output devices on a single chip. Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip. At times it is also termed as a mini computer or a computer on a single chip.

Seg-1-Introductory Concept

Types of Microprocessors

Microprocessors generally is categorized in terms of the maximum number of binary bits in the data they process (i.e. number of bits that their ALU can work with at a time) – that is, their word length, **Data width**. *Over time, five standard data widths have evolved for microprocessors: 4-bit, 8-bit, 16- bit, 32-bit, 64-bit.* So, 16-bit microprocessor means that ALU can work with 16-bit number at a time or, data width of this microprocessor is 16-bit.
There are so many manufacturers of Microprocessors, but only two companies have been produces popular microprocessors: ***Intel*** and Motorola.

Confused bet^n Tpes & Evaluation

Evaluation of microprocessor

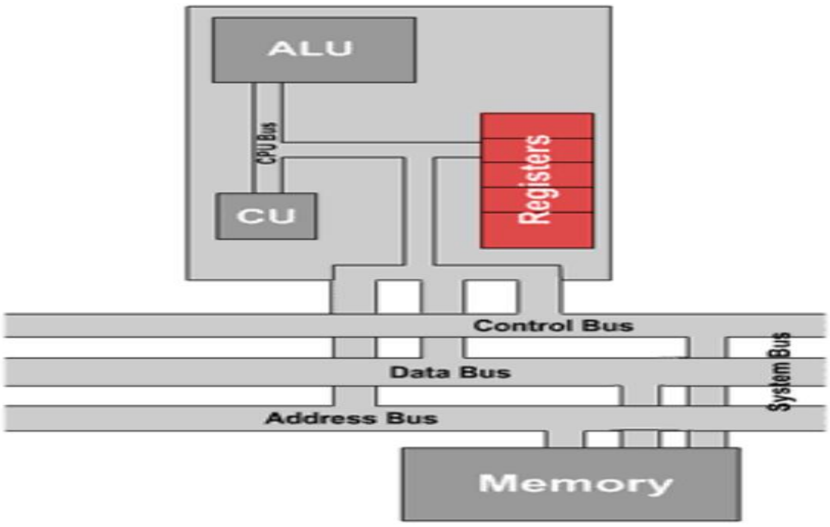
See Online / (Five Generation in brief)

System Bus

It is used for connections between the processor, memory and peripherals, and transferal of data between the various parts. The system bus consists of three different groups of wiring, called the data bus, control bus and address bus. These all have separate responsibilities and characteristics, which can be outlined as follows:

Control Bus: The control bus contains lines that select the memory or I/O and cause them to perform a read or write operation. In most computer system, there are four control bus connections: **MRDC** (Memory read control), **MWTC** (Memory Write control), **IORC** (I/O read control), **IOWC** (I/O write control).
Data Bus: This is used for the exchange of data between the processor, memory and peripherals, and is bi-directional so that it allows data flow in both directions along the wires. Data bus may be 8 bits, 16 bits, 32 bits or 64 bits.
Address Bus: This is a unidirectional bus. This bus is usually 8 to 32 bits wide. Information transfer takes place from the microprocessor to the memory or I/O elements. For a 16-bit address bus, microprocessor can generate 2^{16} = 65,536 different possible address. Each one of these addresses represents a definite memory location or an I/O element.

Hardware of a microprocessor



Memory-addressing technique

Seg-2- 8086 Microprocessor

Properties

Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
5 MHz	16 bits	16 bits	20 bits	2^20 =1 Mega byte

► Intel 8086 microprocessor is a 16-bit microprocessor
► Can process data and memory addresses that are represented by 16 bits at a time

Architecture
Read from my **MP Lab-Report- 02** >> Click Here

Registers
8086 has four 16 bit general purpose registers **AX, BX, CX and DX** to store intermediate values during execution. Each of these have two 8 bit parts (higher and lower).
• **AX register:** It holds operands and results during multiplication and division operations. Also an accumulator during String operations.
• **BX register:** It holds the memory address (offset address) in indirect addressing modes.
• **CX register:** It holds count for instructions like loop, rotate, shift and string operations.
• **DX register:** It is used with AX to hold 32 bit values during multiplication and division.
Arithmetic Logic Unit (16 bit): Performs 8 and 16 bit arithmetic and logic operations.
Special purpose registers (16-bit):
• **Stack Pointer:** Points to Stack top. Stack is in Stack Segment, used during instructions like PUSH, POP, CALL, RET etc. • **Base Pointer:** BP can hold offset address of any location in the stack segment. It is used to access random locations of the stack.
• **Source Index:** It holds offset address in Data Segment during string operations.
• **Destination Index:** It holds offset address in Extra Segment during string operations
• **Instruction Register and Instruction Decoder:** The EU fetches an opcode from the queue into the instruction register. The instruction decoder decodes it and sends the information to the control circuit for execution.

Flag/Status register (16 bits)
8086 has 9 flags that help change or recognize the state of the microprocessor. 6 Status flags:
1. carry flag(CF) 2. parity flag(PF) 3. auxiliary carry flag(AF)

4. zero flag(Z)   5. sign flag(S)   6. overflow flag (O)

Status flags are updated after every arithmetic and logic operation. 3 Control flags:

1. trap flag(TF)   2. interrupt flag(IF)   3. direction flag(DF)

These flags can be set or reset using control instructions like CLC, STC, CLD, STD, CLI, STI, etc. The Control flags are used to control certain operations.

Why Register? Why not Memory?

► Even though the processor can operate on data stored in memory, the same instruction is faster if stored in registers, requires fewer clock cycles

Segmentation: Segmentation is the process in which the main memory of the computer is divided into different segments. A segment is a logical unit of memory that may be up to 2^16=64 kilobytes long.

Why memory is divided into segments?

► 8086 BIU sends out 20 bit address, so it can address any of 2^20 = 1 Megabytes in memory. But, at any time 8086 can address any of 2^16= 64 Kbyte memory. Hence, memory is divided into (4) four 2^16= 64 Kbyte segments. At any given time 8086 can work with only this 4 (four) 64 Kbyte segments within 1Mbyte memory. Four segment registers hold the upper 16 bit of starting address of these four memory segments that 8086 is working at a particular time.

Physical address calculation

Physical Address = Segment × 10h + Offset

Ex: A4FB:4872 = A4FB0h + 4872h = A9822h (Here, ***Segment:*** A4FB and ***Offset:*** 4872, ***h*** means Hexadecimal Number)

Addressing modes (*Read Details from any source or Click Here*)

Data addressing modes: Register addressing, Immediate addressing, Direct addressing, register indirect addressing, Base-plus-index addressing, register relative addressing, Base relative-plus-index addressing

Program memory addressing modes: Direct program memory addressing, Relative program memory addressing, Indirect program memory addressing

Stack memory addressing modes

Addressing Techniques of 8086 Microprocessor

Offset ► Within a segment, a memory location is specified by giving an offset. An offset is the number of bytes from the beginning of the segment to that particular memory location. With a 2^16=64 Kilobyte segment, the offset can be given as a 16-bit number. The first byte in a segment has offset 0000h. The last offset in a segment is FFFFh.

Seg-3-Instruction

Instruction set

An instruction set is a group of commands for a central processing unit (CPU) in machine language. The term can refer to all possible instructions for a CPU or a subset of instructions to enhance its performance in certain situations. The instructions tell the CPU to perform tasks. Some instructions are- ADD, READ, WRITE, MOVE, SUBSTRUCT, MUL, JUMP, LOAD, OUT, STORE, IN commands that direct data to different hardware elements.

1.Operation Field: Opcode

- Specifies the operation to be performed by the instruction. Eg: ADD,SUB,MOV,etc.
- It can be a value or register number on which the operation is performed.

- Mandatory part of every instruction.

2. Address Field

- Address of operand/Operand Reference
- Refers to a location (address) where the operand is stored.
- The address may be a memory address or a register address.

Instruction format

The way an instruction is written. TYPES OF INSTRUCTION FORMATS

1.Zero Address Instruction Format

There is no address field. Stack is used.

2. One Address Instruction Format

- This instruction format uses only one address field.
- The other operand is stored on Accumulator Register.

> Opcode >> Operand/ Address of Operand >> Mode
> Example: **ADD B** >> AC = AC + M[B]

3. Two Address Instruction Format

- It uses two address fields.
- Most commonly used instruction format.
- Example instructions are ADD, MUL, MOV

> Opcode >> Destination Address >> Source Address >> Mode
> Example**: MOV R1, A** >> R1 = M[A]
> **ADD R1, B** >> R1 = R1 + M[B]

4. Three Address Instruction Format

It uses three instruction fields.

> Opcode >> Destination Address >> Source Address >> Source Address >> Mode

Fetch-decode-Execution

To execute an instruction processor must follows minimum of four steps:

(1) Instruction Fetch
(2) Instruction Decode
(3) Instruction execute
(4) Store or, Store of Result

Instruction Fetch: Control unit collect the instructions from main memory and put them in CPU register. This is called instruction fetch.

Instruction Decode: When instruction reaches in processor register, CU decodes or interprets the instruction and sends necessary signals and data to ALU.

Instruction Execute: ALU process the data with arithmetic and logic operations and gives a result according to instructions.

Store Result: Finally, CU stores result in Accumulator or main memory.