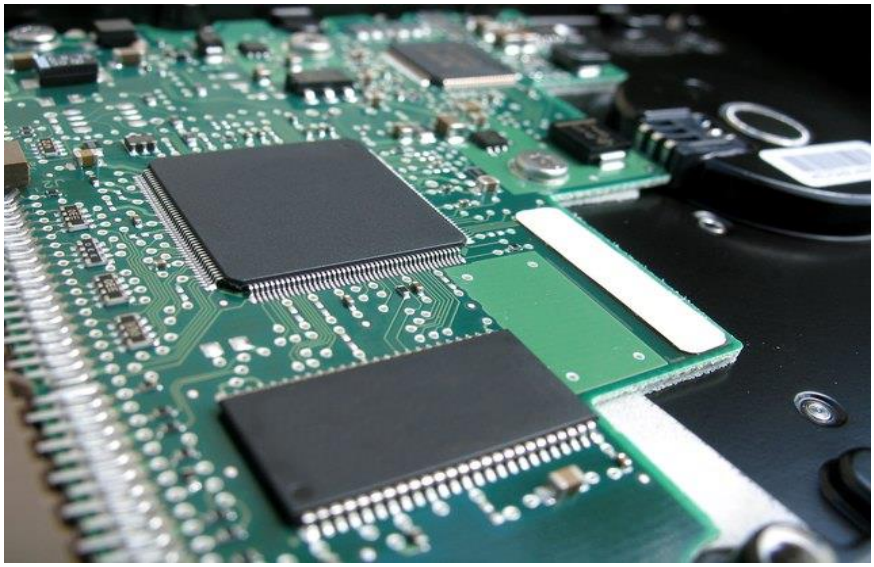
جامعة الحمدانية
كلية التربية

Lecture 20

Microprocessors



Prepared by:

Firas Abdulrahman Yosif

Brief Instructions to Machine Code

Machine Instructions are commands or programs written in machine code of a machine (computer) that it can recognize and execute.

- A machine instruction consists of several bytes in memory that tells the processor to perform one machine operation.
- The processor looks at machine instructions in main memory one after another, and performs one machine operation for each machine instruction.
- The collection of machine instructions in main memory is called a **machine language program.**

Machine code or machine language is a set of instructions executed directly by a computer's central processing unit (CPU). Each instruction performs a very specific task, such as a load, a jump, or an ALU operation on a unit of data in a CPU register or memory. Every program directly executed by a CPU is made up of a series of such instructions.

رمز الآلة أو لغة الآلة عبارة عن مجموعة من التعليمات التي يتم تنفيذها مباشرة بواسطة وحدة المعالجة المركزية للكمبيوتر .(CPU) تؤدي كل تعليمات مهمة محددة جدًا ، مثل عملية تحميل أو قفزة أو عملية ALUعلى وحدة من البيانات في سجل وحدة المعالجة المركزية أو الذاكرة. يتكون كل برنامج يتم تنفيذه مباشرة بواسطة وحدة المعالجة المركزية من سلسلة من هذه التعليمات.

The general format of a **machine instruction** is

[Label:] Mnemonic [Operand, Operand] [; Comments]

- Brackets indicate that a field is optional
- Label is an identifier that is assigned the address of the first byte of the instruction in which it appears. It must be followed by **":"**
- Inclusion of spaces is arbitrary, except that at least one space must be inserted; no space would lead to an ambiguity.
- Comment field begins with a semicolon **" ; "**

Machine instructions used in 8086 microprocessor

1. Data transfer instructions– move, load exchange, input, output.

- MOV: Move byte or word to register or memory.
- IN, OUT: Input byte or word from port, output word to port.
- LEA: Load effective address
- LDS, LES Load pointer using data segment, extra segment.
- PUSH, POP: Push word onto stack, pop word off stack.
- XCHG: Exchange byte or word.
- XLAT: Translate byte using look-up table.

2. Arithmetic instructions – add, subtract, increment, decrement, convert byte/word and compare.

- ADD, SUB: Add, subtract byte or word
- ADC, SBB: Add, subtract byte or word and carry (borrow).

- INC, DEC: Increment, decrement byte or word.
- NEG: Negate byte or word (two's complement).
- CMP: Compare byte or word (subtract without storing).
- MUL, DIV: Multiply, divide byte or word (unsigned).
- IMUL, IDIV: Integer multiply, divide byte or word (signed)
- CBW, CWD: Convert byte to word, word to double word
- AAA, AAS, AAM ,AAD: ASCII adjust for add, sub, mul, div .
- DAA, DAS: Decimal adjust for addition, subtraction (BCD numbers)

3. Logic instructions – AND, OR, exclusive OR, shift/rotate and test

- NOT: Logical NOT of byte or word (one's complement)
- AND: Logical AND of byte or word
- OR: Logical OR of byte or word.
- XOR: Logical exclusive-OR of byte or word
- TEST: Test byte or word (AND without storing).
- SHL, SHR: Logical Shift rotate instruction shift left, right byte or word? by 1or CL
- SAL, SAR: Arithmetic shift left, right byte or word? by 1 or CL
- ROL, ROR: Rotate left, right byte or word? by 1 or CL.
- RCL, RCR: Rotate left, right through carry byte or word? by 1 or CL.

4. **String manipulation instruction** – load, store, move, compare and scan for byte/word

- MOVS: Move byte or word string

- MOVSB, MOVSW: Move byte, word string.
- CMPS: Compare byte or word string.
- SCAS: Scan byte or word string (comparing to A or AX)
- LODS, STOS: Load, store byte or word string to AL.

5. Control transfer instructions – conditional, unconditional, call subroutine and return from subroutine.

- JMP: Unconditional jump .it includes loop transfer and subroutine and interrupt instructions.
- JNZ: jump till the counter value decreases to zero. It runs the loop till the value stored in CX becomes zero

6. Loop control instructions-

- LOOP: Loop unconditional, count in CX, short jump to target address.
- LOOPE (LOOPZ): Loop if equal (zero), count in CX, short jump to target address.
- LOOPNE (LOOPNZ): Loop if not equal (not zero), count in CX, short jump to target address.
- JCXZ: Jump if CX equals zero (used to skip code in loop).
- Subroutine and Interrupt instructions-
- CALL, RET: Call, return from procedure (inside or outside current segment).

- INT, INTO: Software interrupt, interrupt if overflow. IRET: Return from interrupt.

7. Processor control instructions-

Flag manipulation:

- STC, CLC, CMC: Set, clear, complement carry flag.
- STD, CLD: Set, clear direction flag. STI, CLI: Set, clear interrupt enable flag.
- PUSHF, POPF: Push flags onto stack, pop flags off stack.