Yeasin Arafat Prantik

152072202

Homework : 01

Sub: CSE331L

Date: 07-08-20

microcontroller Vs microprocessor:

* A microcontroller, its associated support circuitry, memory, and peripheral I/O components are implemented on a single chip, it is a microcontroller.
* A microprocessor is a cpu on a single chip.

Features of 8086

* 8086 is a 16 bit processor, Its ALU, internal registers work with 16 bit binary word.
* 8086 has 16 bit data bus. It can read or write data to memory either 16 bits or 8 bits at a time.
* 8086 has a 20 bit address bus which means it can address up to 2^{20} = 1 mb memory location.

Scanned with CamScanner

Registrar - Register - resistor

* Both ALU & FPU have a very small amount of super fast private memory placed right next to them for their exclusive use. These are called registers.

* The ALU and FPU store intermediate and final results from their calculation in these registers.

* Processed data goes back to the data cache and then to the main memory from these registers.

Special purpose registers:

* IP - the instruction pointer. Points to the next location of instruction in the memory

* Flags register - Determine the current state of the microprocessor.

Scanned with CamScanner

Creating variables

Syntax for variable declaration.

name DB value
name DW value

DB - stands for define byte
DW - stands for define word

* name - can be any letter or digit combination though it should start with a letter. It's possible to declare unnamed variables by not specifying the name

* value - can be any numeric value in any supported numbering system or symbol for variables that are not initialized.

Creating constants.

constants are just like variables but they exist only until your program is compiled After definition a constant its value can not be changed. To define constants is used

The syntax for DUP:

number - numbers of duplication to make

value - expression that DUP will duplicate

for example

Q DB 5 DUP(9)

is an alternative way of declaring.

Q DB 9,9,9,9,9

OFFSET:

"offset" is an assembler directive in x86 assembly language. It actually means "address" and is a way of handling the overloading of the "mov" instruction. Allow me to illustrate the usage.

1. mov si, offset variable
2. mov si, variable.

Scanned with CamScanner

Print: Hello world in Assembly language.

```
DATA SEGMENT
    MESSAGE DB "HELLO WORLD!!!$"
ENDS
CODE SEGMENT
    ASSUME DS: DATA CS: CODE
START:
    MOV AX, DATA
    MOV DS, AX
    LEA DX, MESSAGE
    MOV AH, 9
    INT 21H
    MOV AH, 4CH
    INT 21H
ENDS
END START
```

Scanned with CamSca