

~~AD~~

- `Max Proc`: this signifies the beginning / start of a p
- `Mov Ax, 47104`: this copies 47104 to the Ax register
- `ADD Eax, 1270`: this instruction adds 1270 to the eax register
- `MOV DS, Ax`: this instruction moves the number of the data segment ~~to~~ from the Ax to the DS
- `ENDP`: this signals the end of the procedure

~~TITLE Add and Subtract~~

- ~~Program Assignment 1 - subtracting three integers~~
- ~~using the~~

`INCLUDE Irvine32.inc`

• `data`

`val1 DWORD 10000h`

`val2 DWORD 40000h`

`val3 DWORD 20000h`

`finalVal DWORD ?`

• `code`

`main PROC`

`mov eax, val1` : start with 10000

`add eax, val2` : add 40000

`sub eax, val3` : subtract 20000

`mov finalVal, eax` : store the result (30000)

`call DumpRegs` ; display the registers

`exit`

`main ENDP`

`END main`

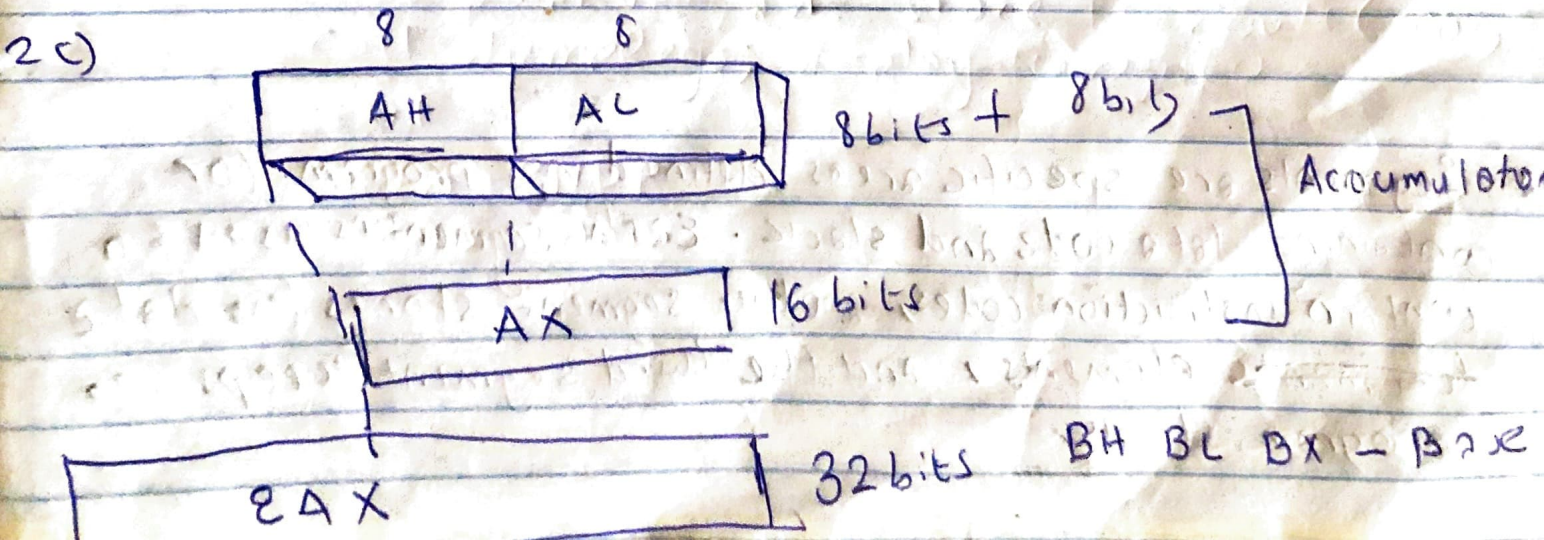


1a) Basically the coded addresses in the instructions would have to be updated whenever new variables were inserted before existing ones

b) Object files are produced by the assembler and also listing (LST) files.

2 Portability is a characteristic attributed to a computer program if it can be used in an operating system other than the one in which it was created without requiring major rework. Also the usage of the same software in different environments.

2b) the assembly language is highly processor dependent ~~that~~ so, no because they are different processors.



3b) ~~MOV AX, 47104 ; move 47104 to the ax register
 ADD EAX, 127Q ; add
 MOV DS, AX ; move AX to DS~~

3c)

- i) Value1 BYTE 6DH
 (value1) is the name given to the variable
 (BYTE) is the directive that defines the data that is being represented
 (6DH) this is the initializer of the hexadecimal
- ii) Value2 [DWORD]: the value name is "Value2". D word is the directive and the question mark means 2 value would be assigned to it later
- iii) Value3 SBYTE -10, -20, -30, -40, -50
 this is the defined statement that contains multiple signed bytes (SBYTES). the name given to it is Value3.

3a) Segments are specific areas defined in a program for containing data code and stack. Each segment is used contain instruction codes, another segment stores the ~~documents~~ elements, and the third segment keeps program stack.

~~AD~~

- `MAN PROC`: this signifies the beginning / start of a p
- `MOV AX, 47104`: this copies 47104 to the AX register
- `ADD EAX, 1270`: this instruction adds 1270 to the EAX register
- `MOV DS, AX`: this instruction moves the number of the data segment ~~to~~ from the AX to the DS
- `ENDP`: this signals the end of the procedure

~~TITLE Add and Subtract~~

- ~~Program Assignment 1 - subtracting three integers~~
- ~~using the~~

`INCLUDE Irvine32.inc`

• `data`

`val1 DWORD 10000h`

`val2 DWORD 40000h`

`val3 DWORD 20000h`

`finalVal DWORD ?`

• `code`

`main PROC`

`mov eax, val1` : start with 10000

`add eax, val2` : add 40000

`sub eax, val3` : subtract 20000

`mov finalVal, eax` : store the result (30000)

`call DumpRegs`; display the registers

`exit`

`main ENDP`

`END main`