**P1: Addressing Data in Memory and Segments (16 bits)**

**Steps to configure MASM Project in Visual Studio 2017**

1. Create C++ empty project
2. Configure Linker for ‘General’, ‘Input’ to add libraries.

**General :** Additional directory libraries - c:\Irvine

**Input:** Additional dependencies: Irvine32.lib;

**System:** Subsystem - Console

1. Add Microsoft Macro Assembler into the project
   1. Right-click project file name - Build dependencies - build customization
   2. Check masm
2. Add item, C++ file, name it as main.asm/ main.cpp to embed assemble language in C/C++ files.
3. To Include Path c:\Irvine for ASM file

* Project Properties - Microsoft Macro Assembler -> General -> Include Paths - c:\Irvine

1. Add breakpoints in your code file (registers value can only be shown during debugging).
2. To enable the Memory windows, Enable address-level debugging must be selected in Tools > Options (or Debug > Options) > Debugging > General.
3. While debugging:

To show **register values:** Debug > Windows > Registers

To show **memory values**:Debug > Windows > Memory

Assembly language template:

.386

.model flat, stdcall

.stack 4096

ExitProcess PROTO, dwExitCode: DWORD

.data

;define variables here

.code

main PROC

;write assembly code here

INVOKE ExitProcess,0

main ENDP

END main

**Debug**

Objective : To understand the machine operation

Tools : Microsoft Visual Studio

# Part I

1. Write the program below and examine the register contents of **eax, ebx, ecx** and **edx** for each step (Press F10: Step Over).

a) mov cl, 42

mov dl, 29

add cl, dl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Command** | **EAX** | **EBX** | **ECX** | **EDX** |
| mov cl, 42 | 75733358 | 7EFDE000 | 00000000 | 000E1005 |
| mov dl, 29 | 75733358 | 7EFDE000 | 0000002A | 000E1005 |
| add cl, dl | 75733358 | 7EFDE000 | 0000002A | 000E101D |

b) EAX = 75733358 EBX = 7EFDE000 ECX = 00000000 EDX = 00991005

mov ax,0123

EAX = 7573007B EBX = 7EFDE000 ECX = 00000000 EDX = 00991005

add ax,0025

EAX = 75730094 EBX = 7EFDE000 ECX = 00000000 EDX = 00991005

mov bx,ax

EAX = 75730094 EBX = 7EFD0094 ECX = 00000000 EDX = 00991005

add bx,ax

EAX = 75730094 EBX = 7EFD0128 ECX = 00000000 EDX = 00991005

mov cx,bx

EAX = 75730094 EBX = 7EFD0128 ECX = 00000128 EDX = 00021005

sub cx,ax

EAX = 75730094 EBX = 7EFD0128 ECX = 00000094 EDX = 00021005

sub ax,ax

EAX = 75730000 EBX = 7EFD0128 ECX = 00000094 EDX = 00021005

1. To obtain the memory address that stores the executing command, examine the register contents of **EIP.**

Write the following program and list the memory addresses that stores each line of codes. (HINT: disassembly)

EAX = 75733358

mov ax, 56h

EAX = 75730056 EBX = 7EFDE000

mov bx, 02h

EAX = 75730056 EBX = 7EFD0002

mul bx

EAX = 757300AC EBX = 7EFD0002

1. Disassembly the following machine code to assembly code/symbolic code:

a) B8 54 01 05 25 00

mov eax,25050154h

add ah,cl

b) B8 05 1B 00 2C EB F8

mov eax,2C001B05h

jmp \_main@0+1Ah (01371042h)

4. Consider the machine language instructions

B0 1C D0 E0 B3 0C F6 E3 6A

Which instruction performs the following operations?

* 1. Move hex value 1C to the AL register.

mov al,1Ch

* 1. Shift the contents of AL one bit to the left.

shl al,1

* 1. Move the hex value 12 to BL.

mov bl,0Ch

* 1. Multiply AL by BL.

mul al,bl

Trace the program and find out the final product in AX? Confirm the result by manual calculation.

5. What is the output in AX?

EAX = 75733358 EBX = 7EFDE000

MOV AL, 5 ; AL = multiplicand

EAX = 75733305

MOV BL, 10 ; BL = multiplier (operand)

EAX = 75733305 EBX = 7EFDE00A

MUL BL

EAX = 75730032 EBX = 7EFDE00A

6. What is the output in AX and DX?

EAX = 75733358 EBX = 7EFDE000

MOV AX, 0083 ; dividend

EAX = 75730053 EBX = 7EFDE000

MOV BL, 2 ; divisor (8 bits)

EAX = 75730053 EBX = 7EFDE002

DIV BL

EAX = 75730129 EBX = 7EFDE002

# Part II

1. Enter the following instructions:

MOV AX, 0010

MOV BX, 0020

MOV CX, 0030

ADD AX, BX

INC BX

SUB CX, AX

DEC CX

What is the content of register AX, BX, CX and IP for each instruction?

Value of registers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **AX** | **BX** | **CX** |
| MOV | AX,010 | 3F86E5EC | 7F2DF000 | 00E61005 |
| MOV | BX,020 | 3F86000A | 7F2DF000 | 00E61005 |
| MOV | CX,030 | 3F86000A | 7F2D0014 | 00E61005 |
| ADD | AX,BX | 3F86000A | 7F2D0014 | 00E6001E |
| INC | BX | 3F86001E | 7F2D0014 | 00E6001E |
| SUB | CX,AX | 3F86001E | 7F2D0015 | 00E6001E |
| DEC | CX | 3F86001E | 7F2D0015 | 00E60000 |
|  |  | 3F86001E | 7F2D0015 | 00E6FFFF |

What is the value in decimal for CX register?

1. What is the final value of AX and BX?

MOV CX,3 ;Initialize for 3 loops

L1: MOV AX,00

MOV BX,00

ADD BX,AX

LOOP L1 ;Decrement CX ;Repeat if nonzero

|  |  |  |
| --- | --- | --- |
| **AX** | **BX** | **CX** |
| EF76505A | 7ED38000 | 01231005 |
| EF76505A | 7ED38000 | 01230003 |
| EF760000 | 7ED38000 | 01230003 |
| EF760000 | 7ED30000 | 01230003 |
| EF760000 | 7ED30000 | 01230003 |
| EF760000 | 7ED30000 | 01230002 |
| EF760000 | 7ED30000 | 01230002 |
| EF760000 | 7ED30000 | 01230002 |
| EF760000 | 7ED30000 | 01230002 |
| EF760000 | 7ED30000 | 01230001 |