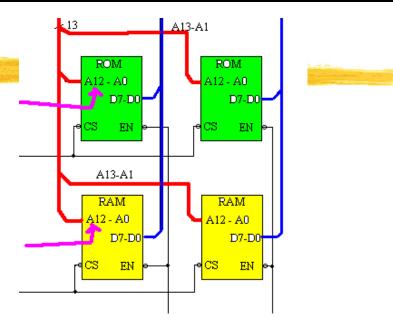
#### EECE416 : Microcomputer Fundamentals and Design

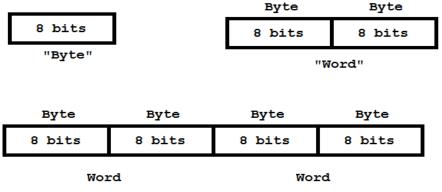
#### X86 Assembly Programming Part 1 – MASM

www.MWFTR.com

# Multiple Address Access Issues

- Reminder: a single address locates only a byte of memory
- 8-bit processor
  - Accesses one address with a byte data
- 16-bit processor
  - Accesses two address spaces (Even (or Low Byte) and Odd(or High Byte)) at a single execution with 2-byte (or "Word") data
  - Where do we store each of the 2 bytes to each of the 2 address spaces?





"Long Word"

#### **Big-Endian vs. Little-Endian**

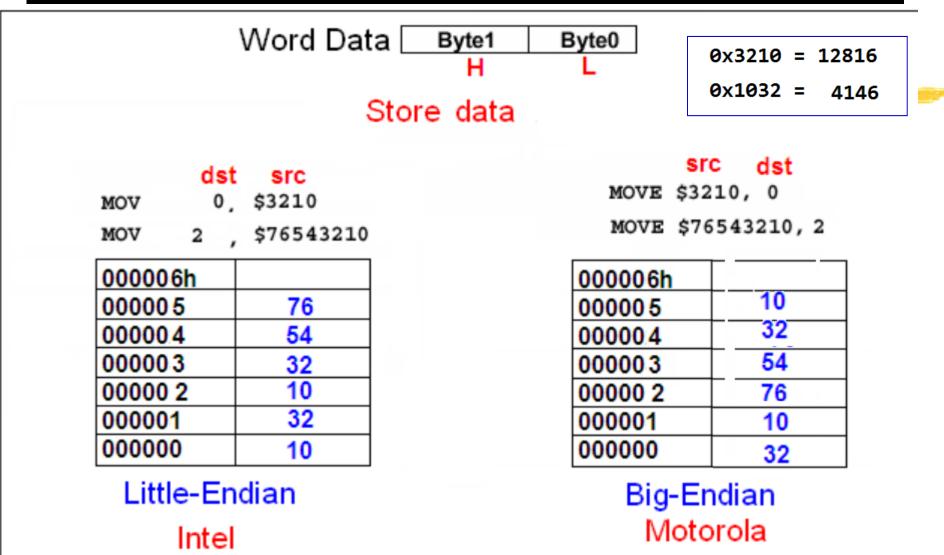
**Big-Endian**: Words are stored with the lower 8- bits ("Lower Byte") in the higher of the two storage locations ("Addresses"): Motorola

### <u>Big</u> guy ("Upper Byte") <u>ends</u> at lower address"

#Little- Endian: Lower-order byte stored in the lowest address: Intel 80x86 family

Little guys ("Lower Byte") ends at lower address"

#### **Big-Endian vs. Little-Endian**



## "Endianness"

#### **#** Endian or Endian-Architecture

- how multi-byte data is represented by a computer system and is dictated by the CPU architecture of the system
- Not all computer systems are designed with the same endian architecture
- ☐ Issues with software and interface

Computer System Endianness

#### Common file formats

| Platform              | Endian Architecture | Little Endion Format     | Die Endian Correct      | Variable or Di Endian Format          |  |  |
|-----------------------|---------------------|--------------------------|-------------------------|---------------------------------------|--|--|
|                       |                     | Little-Endian Format     | Big-Endian Format       | Variable or Bi-Endian Format          |  |  |
| ARM*                  | Bi-Endian           | BMP (Windows* & OS/2)    | PSD (Adobe Photoshop*)  | DXF (AutoCAD*)                        |  |  |
| DEC Alpha*            | Little-Endian       | GIF                      | IMG (GEM Raster*)       | PS (Postscript*, 8 bit                |  |  |
| HP PA-RISC 8000*      | Bi-Endian           | FLI (Autodesk Animator*) | JPEG, JPG               | interpreted text, no<br>Endian issue) |  |  |
| IBM PowerPC*          | Bi-Endian           | PCX (PC Paintbrush*)     | MacPaint                | POV (Persistence of                   |  |  |
| Intel® 80x86          | Little-Endian       | QTM (MAC Quicktime*)     | SGI (Silicon Graphics*) | Visionraytracer*)                     |  |  |
| Intel® IXP network    | Bi-Endian           | RTF (Rich Text Format)   | Sun Raster              | RIFF (WAV & AVI*)                     |  |  |
| processors            |                     |                          | WPG (WordPerfect*)      | TIFF                                  |  |  |
| Intel® Itanium®       | Bi-Endian           |                          |                         | XWD (X Window Dump*)                  |  |  |
| processor family      |                     | Bus Protocols            | Network Protocols       | Bus Protocols                         |  |  |
| Java Virtual Machine* | Big-Endian          | Infiniband               | TCP/IP                  | GMII (8 bit wide bus, no              |  |  |
| MIPS*                 | Bi-Endian           | PCI Express              | UDP                     | Endian issue)                         |  |  |
| Motorola 68k*         | Big-Endian          | PCI-32/PCI-64            |                         |                                       |  |  |
| Sun SPARC*            | Big-Endian          | USB                      |                         |                                       |  |  |

## **Bi-Endian & Endian-Neutral Approaches**

jakob.engbloms.se/archives/1336

#### **#**Conversion Convert to BE or LE byte sequence when sent to Byte Swap memory Seque Network I/O Macro nce of byte Internal register represention as word Processor model Memory model A pipe moving a sequence of bytes from one place to another The bus system

#### **∺** "Endian Neutral":

## HOMEWORK

**#**Technical Report on "Endian-Neutral Approaches"

△What are they?

⊡ How do they work?

○Who are involved and leading the works

Submission Due: Hardcopy + Softcopy by 5:30pm Thursday Oct 13

How to write well for busy technical people ?

#### **Compare this**

듣) 🕙 www.cbsnews.com/8301-202\_162-57600384/syria-strike-seems-inevitable-as-u.n-warns-against-unilateral-military-action-hunt-1

#### Updated at 6:48 a.m. Eastern

**DAMASCUS, SYRIA** U.N. chemical weapons experts investigating an alleged poison gas attack near Damascus left their hotel again Wednesday hoping to carry out their second field trip, which was delayed Tuesday for security reasons.

The team of about 20 inspectors left their hotel in the Syrian capital in a convoy of cars to visit the eastern Ghouta suburbs, where the Obama administration says President Bashar Assad's forces unleashed a chemical weapons attack on Aug. 21 that killed hundreds of people.

Local opposition activists told CBS News that the convoy had reached the town of Mleiha, in the sprawling Ghouta area, and videos posted online by the activists showed the U.N. inspectors interviewing patients at clinics in Mleiha and the nearby town of Zamalka.



Play VIDEO

Intercepted communications, tissue samples prove Syrian regime responsible for gas attack



On Tuesday, Vice President Joe Biden made it clear that regardless of what the U.N. inspectors find, the White House is now convinced the attack was carried out by Assad's forces.

The American government's assessment is based on the circumstantial evidence from videos posted on the internet, and, as CBS News correspondent David Martin reported Tuesday, intelligence -much of it still classified -- ranging from intercepted Syrian communications to tests of tissue samples taken from victims.

Another key piece of circumstantial evidence which has been cited by both officials and analysts for days is the simple fact that the regime is the only entity in Syria known to have chemical weapons and the means to disperse them.



# With this

By Oliver Holmes and Erika Solomon BEIRUT | Wed Aug 28, 2013 7:59am EDT

(Reuters) - The United Nations Security Council was set for a showdown over Syria on Wednesday after Britain sought authorization for Western military action that seems certain to be vetoed by Russia and probably China.

U.N. chemical weapons experts investigating an apparent gas attack that killed hundreds of civilians in rebel-held suburbs of Damascus made a second trip across the front line to take samples. Secretary-General Ban Ki-moon pleaded for them to be given the time they need to complete their mission.

But the United States and European and Middle East allies have already pinned the blame on Assad and, even without full U.N. authorization, U.S.led air or missile strikes on Syria look all but certain, though the timing is far from clear.

That has set Western leaders on a collision course with Moscow, Assad's main arms supplier, as well as with China, which also has a veto in the Security Council and disapproves of what it sees as a push for Iraq-style "regime change" - despite U.S. denials that President Barack Obama aims to overthrow Assad.

Analysis & Opinion

Western powers could strike Syria within days



West mustn't rush into Syrian conflict

#### Related Topics

World » Russia » United Nations » Syria »

#### Related Video

#### U.N. resumes Syria chemical attack probe

4:20am EDT

Rebels gain ground in Northern Syria

Israel will respond with force to any attack from Syria

Biden: No doubt Syrian regime used chemical weapons

Uncertainty over how the escalation of the conflict at the heart of the oilexporting Middle East will affect trade, and the world economy sent oil prices, and gold, to their highest levels in months while stocks fell. Fears over the economy of Syria's hostile neighbor Turkey pushed its lira to a record low.

#### Writing well for the Class

- # People are more likely to read subjects/writings/emails that create curiosity or provide utility.
- When they are busy
  - Curiosity fades in importance
  - △ They read only the ones with practical importance ["utility"]
- So, write as if you are a staff writer (<u>targeting for busy</u> <u>people</u>) for a newspaper, and remember that you have an editor whose job is to cut your article to fit into a limited space, maybe just 1 inch in a column.
  - Important things [Conclusions and summary] in the first paragraph
    - Summary of the event/thing first so that it delivers message even though only that summary survives the "cutting"
  - △ Then expand your story after the <u>First Paragraph</u>
  - ⊡ Use your own words





## **HOMEWORK - Recap**

Herein Content an "Endian-Neutral Approaches"

**Subject** 

⊠What are they?

⊠How do they work?

⊠Who are involved and leading the works

☐ Format:

🗵 2 - 3 pages

First paragraph must summarize the entire report (the importance of the first paragraph – your own words --- this determines the grade): you may want to write this first paragraph after completing the report.

No figure, no photo, <u>text only</u>.

Submission Due: Hardcopy + Softcopy by <u>5:30pm</u> Thursday Oct <u>13</u>

### x86 Architecture

First x86 Family member:
 8086 (→ 8088). 1978
 △ Cf. 4004 → 8080 → 8085

#### **8086 ∺**

- 16-bit registers, external data bus
- $\sim$  20-bit addressing ( $\rightarrow$ 
  - 1MB address space)
- Segmentation : 64KB
  - Why? Internal 16-bit register cannot hold 20-bit address

| GND U<br>AD14 U<br>AD13 U<br>AD12 U<br>AD11 U<br>AD10 U<br>AD2 U<br>AD3 U<br>AD4 U<br>AD3 U<br>AD3 U<br>AD1 U<br>AD3 U<br>AD1 U<br>AD3 U<br>AD1 U<br>AD3 U<br>AD1 U<br>AD3 U<br>AD1 U<br>AD1 U<br>AD3 U<br>AD1 U<br>A | 17 | 8086<br>CPU | 40<br>39<br>38<br>37<br>36<br>35<br>34<br>33<br>32<br>31<br>30<br>29<br>28<br>27<br>26<br>27<br>26<br>25<br>24 | MAX<br>MODE<br>V <sub>CC</sub><br>AD15<br>A16/S3<br>A17/S4<br>A18/S5<br>A19/S6<br>BHE/S7<br>BHE/S7<br>MN/MX<br>RD<br>RQ/GT0<br>RQ/GT0<br>RQ/GT0<br>RQ/GT1<br>LOCK<br>S2<br>S1<br>S0<br>QS0<br>QS1 | <pre>{ MIN<br/>MODE }<br/>(HOLD)<br/>(HLDA)<br/>(WR)<br/>(M/IO)<br/>(DT/R)<br/>(DEN)<br/>(ALE)<br/>(INTA)</pre> |  |
|--|----|-------------|--|---|---|--|
| ADO 🗖  | 16 |             | 25   | <b>Q</b> QS0  | (ALE)   |  |

#### X86 Modes of Operation

#### **Real** Address Mode (16-bit registers)

- 1 MB Memory can be accessed 0x00000 0xFFFFF
- Processor runs one program at a time
- Application programs are allowed to access any memory location including that is linked directly to system hardware
- MS-DOS and Windows95 and 98
- A few extra features available for this mode by which direct access to system memory and hardware devices is possible
- Programs running in real-address mode can cause the OS crash

#### x86 Memory Management – Real Addr Mode

0xFFFFF 0xF0000 **Hemory Segmentization 0xEFFFF** (for 8086 with 20 Addr lines) 1 0xE0000 MB Memory can be accessed 0xD0000  $0 \times 00000 - 0 \times FFFFF$ 0xC0000 But 16-bit register cannot hold 0xB0000 the 20-bit address 0xA0000 Solution: Segmentize the 1MB memory space into multiple (16 0x90000 exactly) areas (segments) offset 0x80000 Accessing by Segment 0x70000 0x800250 (expressed by first 4 hex digits 0x80000:0x0250 0x60000 only) + Offset (max. offset = **OxFFFF**) 0x50000 8000:0250 0x40000

0x30000

0x20000

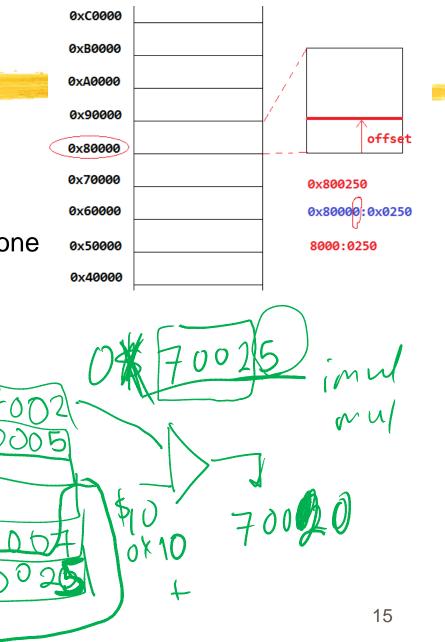
0x10000

0x00000

#### x86 Memory Management – Real Addr Mode

#### ∺20-bit address calculation

- 16-bit segment value
- 16-bit offset value
- Memory address = (16-bit Seg value) \*0x10 + (16-bit Offset)
- 16-bit segment value is placed in one of the segment registers:
  - ☑CS: Code seg register☑DS: data seg register
  - ⊠SS: Stack seg register
  - ⊠ES, FS, GS, etc

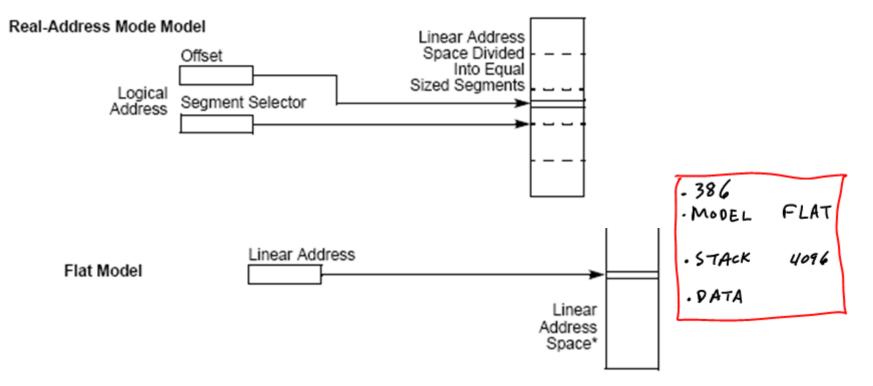


## x86 Mode – Protected Mode

- - Native State of the Processor in which all instructions and features are available
  - Programs are given separate memory areas and the processor prevents programs from referencing memory outside their assigned segments.
  - Access to a total of 4GB Memory (for i386): 16MB for i286
  - Multiple programs run at the same time own reserved memory area
  - ○Windows and Linux

#### x86 Memory Management – Protected Mode

- In Microsoft Assembler (MASM)
  - "Flat (Segment) Mode" is appropriate for protected mode programming
  - Just one(1) 32-bit address register is enough (for up to 4 GB Memory space)
  - Actual address location calculation is done in the background

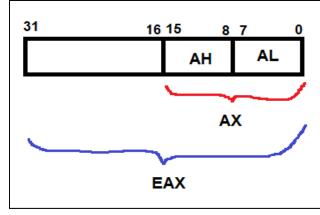


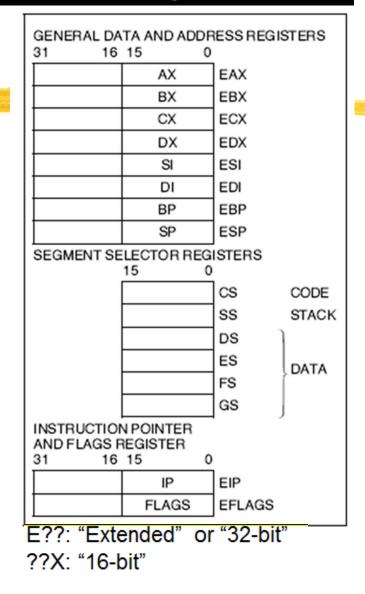
## x86 Architecture

| ₩ 80286 <b>7</b> 8 4  |
|---|
| "Protected Mode" first introduced   |
| Segment register contents as selector or pointer  |
| $\boxtimes$ 24-bit base address $\rightarrow$ 16MB memory size 32   |
| ₩ 80386   |
| $\simeq$ 32-bit registers for operands and addressing( $\rightarrow$ 4GB space)   |
| Lower half of 32 bits is equivalent to 16 bits of earlier<br>generations [Backward (upward) compatibility with 16-bit<br>registers]                         |
| Some new instructions was added (like <u>bit manipulation</u> )   |
| Max 4GB segmentation of physical space  |
| New Parallel Processing Stages introduced: Bus Interface Unit,<br>Code Prefetch Unit, Instruction Decode Unit, Execution Unit,<br>Segment Unit, Paging Unit |

#### Overview of Basic Execution – 386 or higher

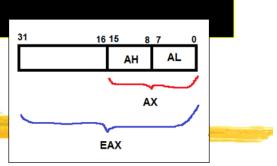
- Set of resources for Executing instructions and for Storing code, data, and state information
- **#** Resources:
  - 8 General data registers
  - △ 6 Segment registers
  - △ Status and control registers
- **Holding the following items (for all):** 
  - Operands for logical and arithmetic operations
  - Operands for address calculations
  - Memory pointers





#### General-Purpose Data Registers

- Primaries
  - EAX (accumulator for operands and results data)
  - EBX (Pointer to data in Segment)
  - ECX (Counter)
  - EDX (for I/O pointer)
- Secondaries
  - EBP (base pointer to data on the stack in DS segment)
  - ESI (Source pointer)
  - EDI (data pointer) for string instructions
  - ESP (Stack pointer)holds the stack pointer (restricted use)
  - ESP points to the top item on the stack and the EBP points to the "previous" top of the stack before the function was called.



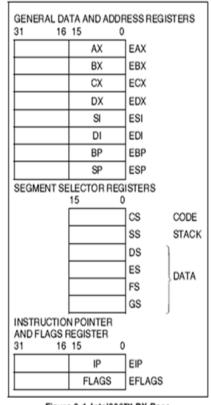


Figure 2-1. Intel386™ DX Base Architecture Registers

#### **EFLAG Register**

- ∺ 32-bit register
  - ☐ Initial state: 0x0000002
  - Contains a group of status flags (S), a control flag (C), and a group of system flags (X) 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| GENERAL DATA AND ADDRESS REGISTERS                      |            |        |       |  |  |  |  |
|---|------------|--------|-------|--|--|--|--|
| 31 16   | 15 0       |        |       |  |  |  |  |
|   | AX         | EAX    |       |  |  |  |  |
|   | BX         | EBX    |       |  |  |  |  |
|   | CX         | ECX    |       |  |  |  |  |
|   | DX         | EDX    |       |  |  |  |  |
|   | SI         | ESI    |       |  |  |  |  |
|   | DI         | EDI    |       |  |  |  |  |
|   | BP         | EBP    |       |  |  |  |  |
|   | SP         | ESP    |       |  |  |  |  |
| SEGMENT SE  | LECTOR REG |        |       |  |  |  |  |
|   |            | cs     | CODE  |  |  |  |  |
|   |            | ss     | STACK |  |  |  |  |
|   |            | DS     | )     |  |  |  |  |
|   |            | ES     | DATA  |  |  |  |  |
|   |            | FS     |       |  |  |  |  |
|   |            | GS     |       |  |  |  |  |
| INSTRUCTION POINTER<br>AND FLAGS REGISTER<br>31 16 15 0 |            |        |       |  |  |  |  |
|   | IP         | EIP    |       |  |  |  |  |
|   | FLAGS      | EFLAGS |       |  |  |  |  |
| 31 16   | IP         | EIP    |       |  |  |  |  |

Figure 2-1. Intel386™ DX Base Architecture Registers

| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   | C<br>F |
|---|--------|
| X ID Flag (ID)<br>X Virtual Interrupt Pending (VIP)<br>X Virtual Interrupt Flag (VIF)<br>X Alignment Check (AC)<br>X Virtual-8086 Mode (VM)<br>X Resume Flag (RF)<br>X Nested Task (NT)<br>X I/O Privilege Level (IOPL)<br>S Overflow Flag (OF)<br>C Direction Flag (DF)<br>X Interrupt Enable Flag (IF)<br>X Trap Flag (TF)<br>S Sign Flag (SF)<br>S Zero Flag (ZF)<br>S Auxiliary Carry Flag (AF)<br>S Parity Flag (PF) |        |

S Carry Flag (CF)

S Indicates a Status Flag

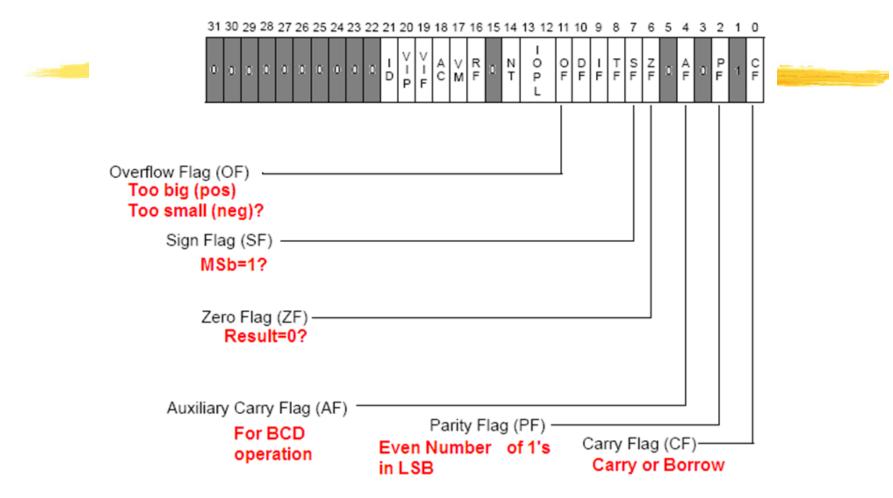
C Indicates a Control Flag

X Indicates a System Flag



Reserved bit positions. DO NOT USE. Always set to values previously read.

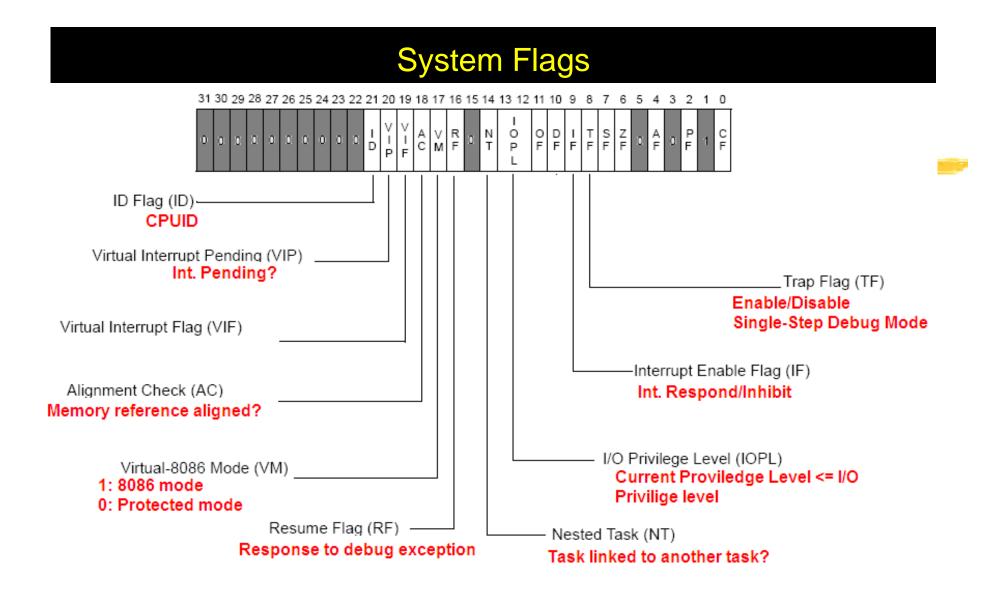
#### Status Flags



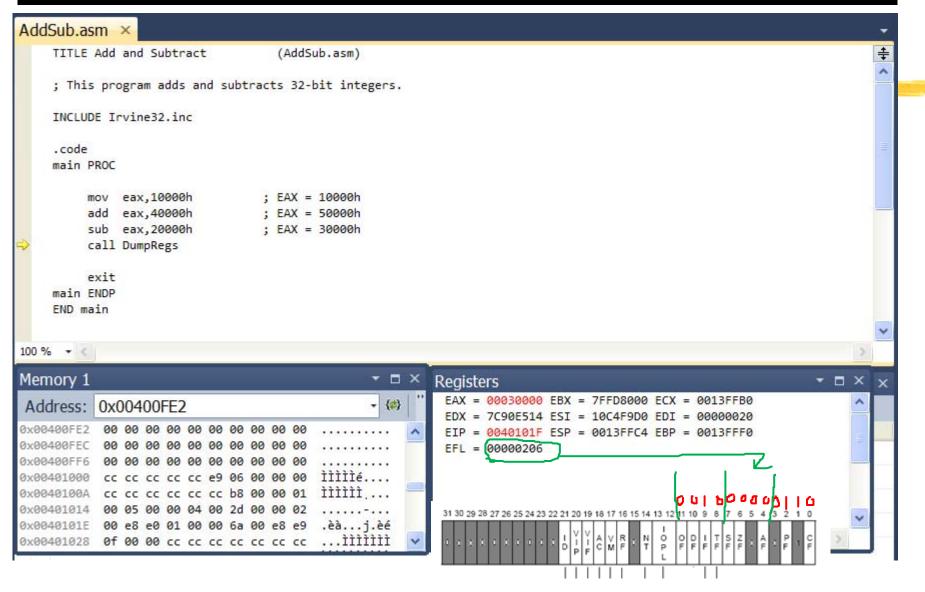
## Control Flag (DF)

#### 

- ∺ DF (Direction Flag)
  - The direction flag controls the string instructions (MOVS, CMPS, SCAS, LODS, and STOS).
  - $\square$  DF=1  $\rightarrow$  string instructions to auto-decrement (that is, to process strings from high addresses to low addresses).
  - $\square$  DF=0  $\rightarrow$  string instructions to auto-increment (process strings from low addresses to high addresses).
  - $\bigtriangleup$  STD  $\rightarrow$  Set DF flag
  - $\square$  CLD  $\rightarrow$  Clear DF flag



### **MASM Screen Capture**



25

#### Web Page for Instruction and Link Library

#### **H** Individual Assignment:

Install a Visual Studio version in to your computer

Read the Instruction very carefully

And install the Link Library

Bring your computer to the Tuesday (Oct 11 2016) class

EECE416 Microcomputers

Lecture 4: x86 Assembly Programming

Getting Started with MASM and Visual Studio:

- Visual Studio 2010: <u>Instruction</u> (Read this first) + Link Library (<u>.msi file</u>. Download and save this file in your computer)
- Visual Studio 2012: <u>Instruction</u> (Read this first) + Link Library (<u>.msi file</u>. Download and save this file in your computer)
- Visual Studio 2013: <u>Instruction</u> (Read this first) + Link Library (<u>.msi file</u>. Download and save this file in your computer)

**Microcomputer Project** 

₩Week 1 (due Oct 11)

# ISSUES – Driver Program and/or OS incompatibility

Mini Arduino

⊠MSP430

## SOLUTION1

☑ Try to solve by Wednesday

⊠ If not solved, receive new type of controllers ASAP

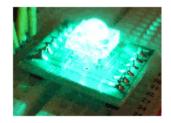
SOLUTION2

⊠Suggestion???

# **Microcomputer Project**

Week 2 (due Oct 18)
Connection of an RGB LED
Written Report – must include
Brief description of the project
Code
Connection Diagram
Screen captures or photo-shots of working system



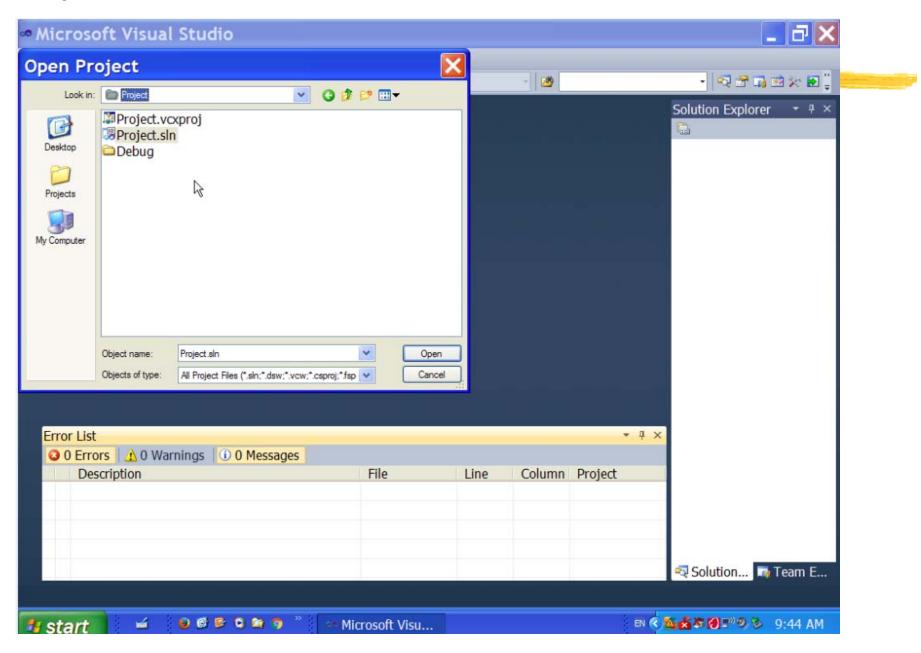




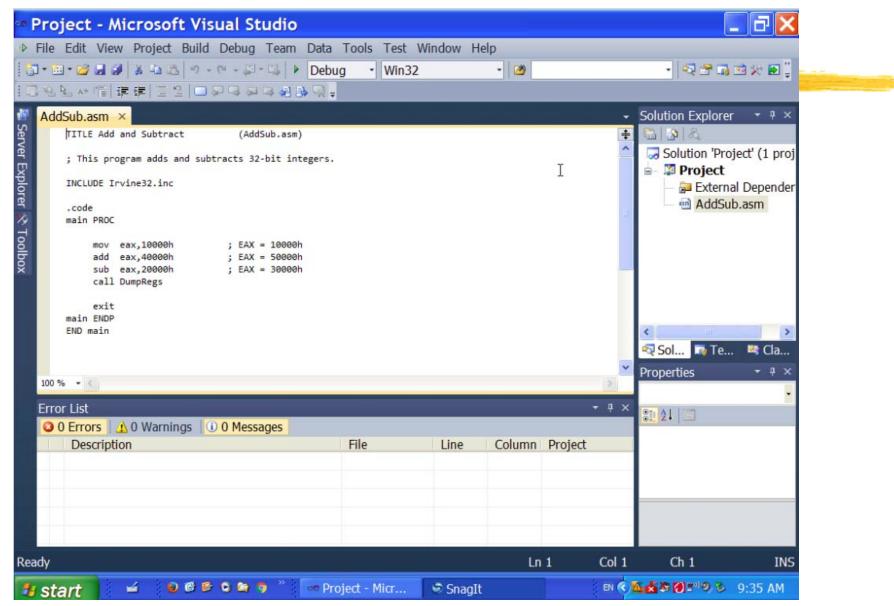
# **1** Sample Code and Run in Visual Studio – Open Project Solution

| Microsoft Visual Studio  |   |             |         |   |         |
|--|---|-------------|---------|---|---------|
| File Edit View Debug Team Data Tools   | Test Window Help  |             |         |   |         |
| New  | •   | - 2         |         | • 🔍 🕾 🖬 🖻                                     | i 🗶 🛃 📜 |
| Open<br>Close<br>Close Solution  | <ul> <li>Project/Solution</li> <li>Web Site</li> <li>Team Project.</li> </ul> | Shift+Alt+O |         | Solution Explorer                             | - 4 ×   |
| <ul> <li>Save Selected Items Ctrl+S</li> <li>Save Selected Items As</li> <li>Save All Ctrl+Shift+S</li> <li>Export Template</li> </ul> | File<br>Convert   | Ctrl+O      |         |   |         |
| Source Control   | •   |             |         |   |         |
| Page Setup Print Ctrl+P Ctrl+P   |   |             |         |   |         |
| Recent Files<br>Recent Projects and Solutions<br>Exit Alt+F4   | •   |             |         |   |         |
| Error List   |   |             | • ∓ ×   |   |         |
| O Errors ▲ 0 Warnings 0 0 Messages     Description   | File  | Line Column | Droject |   |         |
| Description  | 1110  | Column      | Tioject |   |         |
|  |   |             |         | Solution 🚮 T                                  | eam E   |
| Ready  |   |             |         |   |         |
| 💤 start 🧉 🧉 🖉 🖻 🖻 😭 🐐  | 🤲 Microsoft Visu  |             | en 🔍 🛙  | 1 <b>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 </b> | :43 AM  |

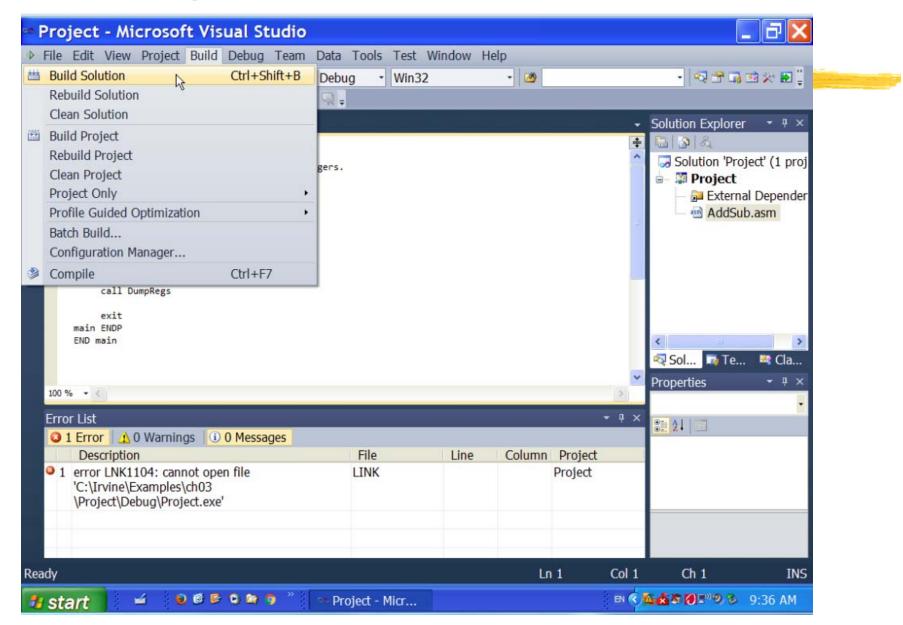
#### 2 Sample Code and Run in Visual Studio – Select Project.sln



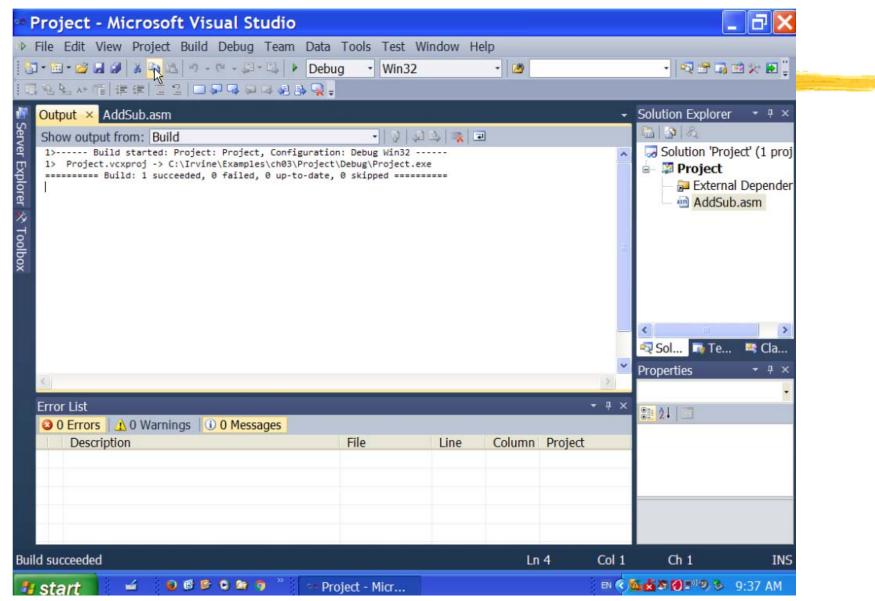
#### 3.Sample Code and Run in Visual Studio – Addsub.asm



# 4 Sample Code and Run in Visual Studio – Build Solution/Compile



#### 5. Sample Code and Run in Visual Studio – Build Success



#### 6 Sample Code and Run in Visual Studio --- Run/Debug (F10 key)

| Project (Debugging) - Microsoft Visual Studio   | _ 7 🗙                         |
|---|-------------------------------|
| File Edit View Project Build Debug Team Data Tools Test Window Help   |                               |
| □ - = - 😂 🚽 👙 🚵 👘 - ભ - 💭 - 🖏 ト Debug - Win32 - 🔯   | - 🔍 🕾 🖬 🖄 🗶 💽 🗍               |
| □ 12 22 A / 「 # # □ 2 □ P P P P P P P P I = 0 + 9 1 [ + 1 = 0 + 9 1 [ + 1 = 0 + 9 1] + ( + 1 = 0 + 9 1] + ( + 1 = 0 + 1 + 1 + 1 = 0 + 1 + 1 + 1 = 0 + 1 + 1 + 1 + 1 = 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 +   |                               |
| Process: [4044] Project.exe   | t.exe!main()Line 11 🔹 =       |
| AddSub.asm ×  | - Solution Explorer - + ×     |
| TITLE Add and Subtract (AddSub.asm)   | [♣] 월 월                       |
| ; This program adds and subtracts 32-bit integers.  | Solution 'Project' (1 project |
| .code<br>main PROC  | External Dependencie          |
| mov eax,10000h ; EAX = 10000h<br>add eax,40000h ; EAX = 50000h<br>sub eax,20000h ; EAX = 30000h   |                               |
| call DumpRegs<br>I<br>exit<br>main ENDP<br>END main   |                               |
| exit<br>main ENDP<br>END main   | ~                             |
| exit<br>main ENDP<br>END main   | >                             |
| exit<br>main ENDP<br>END main<br>100 % • <<br>Memory 1 • • × Registers •  |                               |
| intervent       intervent         intervent <td></td> |                               |
| exit<br>main ENDP<br>END main<br>100 % • <<br>Memory 1 • • × Registers •<br>Address: 0x00401000 • (#) '' EAX = 00010000 EBX = 7FFDA000 ECX = 0013FFB0   |                               |
| exit         main ENDP         END main         100 % • <   |                               |
| exit         main ENDP         END main         100 % • <   |                               |
| exit         main ENDP         END main         100 % • <   |                               |
| exit         main ENDP         END main         100 % * <   | × < ×                         |
| i       i         exit       main ENDP         END main       i         100 % • <   | Solution                      |
| exit         main ENDP         END main         100 % • <   | × < ×                         |

# 7. Sample Code and Run in Visual Studio --- LIST file open

| Project - Microsoft Visual Studio              |                      |                 |              | - 7 🔀                      |  |
|--|----------------------|-----------------|--------------|----------------------------|--|
| File Edit View Project Build Debug Team        | Data Tools Test V    | Window Help     |              |                            |  |
| New  | ebua - Win32         | - 20            |              | - 🔍 🕾 🖬 🖄 🗶 🛃 📜 👝          |  |
| Open   | 🖞 🛱 Project/Solution | on Ctrl+Shift+C |              |                            |  |
| Add  | " 🐸 Web Site         | Shift+Alt+O     |              | Solution Explorer • + ×    |  |
| Close  | Team Project.        |                 | +            |                            |  |
| Close Solution                                 | File                 | Ctrl+O          | ^            | Solution 'Project' (1 proj |  |
| Save AddSub.asm Ctrl+S                         | Convert              |                 |              | 🖮 🗊 Project                |  |
| Save AddSub.asm As                             |                      |                 |              | - 🔁 External Depender      |  |
| Advanced Save Options<br>Save All Ctrl+Shift+S |                      |                 |              | MddSub.asm                 |  |
| Export Template                                |                      |                 |              |                            |  |
| Source Control                                 |                      |                 |              |                            |  |
| Page Setup                                     | -                    |                 |              |                            |  |
| I Print Ctrl+P                                 |                      |                 |              |                            |  |
| Recent Files                                   |                      |                 |              |                            |  |
| Recent Projects and Solutions                  |                      |                 |              | < >                        |  |
| Exit Alt+F4                                    |                      |                 |              | 💀 Solution 🖬 Team E        |  |
| 100 % - <                                      |                      |                 | 2            | Properties • 7 ×           |  |
| Error List                                     |                      |                 | • <b>∓</b> × | - A1 (1995                 |  |
| 3 0 Errors 1 🗘 0 Warnings 1 0 Messages         |                      |                 |              | 21 21 I                    |  |
| Description                                    | File                 | Line Column     | Project      |                            |  |
|  |                      |                 |              |                            |  |
|  |                      |                 |              |                            |  |
|  |                      |                 |              |                            |  |
|  |                      |                 |              |                            |  |
| leady  |                      | Ln              | 11 Col 45    | Ch 32 INS                  |  |
| 🖞 start 🧉 🧉 🖻 🖻 🖻 🤋 🐩                          | 🖙 Project - Micr     |                 | EN 🔇         | 🏧 🛣 🔞 🕬 🔌 9:39 AM          |  |

# 8 Sample Code and Run in Visual Studio --- .LST file

| Project - Microsoft Visual Studio   |                  |               | 🔳 🗗 🔀  |  |
|---|------------------|---------------|--|--|
| Open File   | ×                | ow Help       |  |  |
| Look in: 🗁 ch03   | G 🗊 📁 🗔 🗸        | - 29          | • 🔍 🕾 🖬 🖄 💥 🖬 🍹  |  |
| Image: Computer       Image: Computer         Image: Computer       Image: Computer | ₽ <b>3</b>       |               | <ul> <li>Solution Explorer</li> <li>Solution 'Project' (1 proj</li> <li>Project</li> <li>External Depender</li> <li>AddSub.asm</li> </ul>  |  |
| Object name: AddSub.lst<br>Objects of type: All Files (*.*)   | Open     Cancel  |               | Solution Team E<br>Properties Team X   |  |
| Error List  |                  |               | ×  |  |
| O Errors 0 Warnings 0 Messages<br>Description   | File             | Line Column P | and a second sec |  |
| 📲 start 🧉 🧉 🖉 📽 🧿 🐣   | 🗢 Project - Micr |               | en <b>( ) 🔩 🖧 🏹 () E</b> <sup>11</sup> 🤊 🕉 9:40 AM   |  |

### 9 Sample Code and Run in Visual Studio --- .lst

| o <mark>ject - Micros</mark><br>Edit View Proje |  |                          | Data Tools    | Tost Wi              | ndow H   | oln                                  |         |       |   |                  |
|---|--|--------------------------|---------------|----------------------|----------|--------------------------------------|---------|-------|---|------------------|
| - 🚅 🛃 🥵 🕺 🕮                                     |  |                          |               | Win32                |          | •   🖄                                |         |       |   | a 🖻 🌫 🖬 🖞        |
|   | Alexand a second se | ent meetings             |               | WIII5Z               |          |                                      |         |       |   | -10 -22          |
| -   |  |                          |               |                      |          |                                      |         |       |   |                  |
| IdSub.lst × AddSu                               |  |                          | 707 40        | 01/20/05             |          |                                      |         |       | Solution Explo  | rer 🔹 म 🗙        |
| Microsoft (R) Mac<br>*Add and Subtract          |  | dSub.asm                 |               | 01/28/06<br>ge 1 - 1 | 17:52:13 |                                      |         | + ^   | and the second se | Project' (1 proj |
|   | TITLE Add and  | Subtract                 | ()            | AddSub.asm)          |          |                                      |         |       |   | rnal Depender    |
|   | ; This program<br>; Last update:   |                          | ubtracts 32-1 | oit integer          | 5.       |                                      |         |       | - 🌆 Add   | Sub.asm          |
|   | INCLUDE Irvine<br>C ; Include f<br>C<br>C ;OPTION CAS<br>C<br>C INCLUDE Sma<br>C .NOLIST                       | ile for Irv<br>EMAP:NONE | ; option      | nal: make i          |          | inc)<br>s case-sens:<br>ructures, an |         | nts   |   |                  |
|   | C .LIST<br>C<br>C INCLUDE Vir  |                          | c             |                      |          |                                      |         | G     | <<br>⊲⊋ Solution  |                  |
| )% - <  | C ; VirtualKe  | ys.inc                   |               |                      |          |                                      |         | >     | Properties  | <b>-</b> ₽ ×     |
| or List<br>0 Errors 🛛 🔬 0 Wa                    | minas 0 Ma   | 52005                    |               |                      |          |                                      | ł       | • ₽ × |   | •                |
| Description                                     |  | sayes                    | File          |                      | Line     | Column                               | Project |       |   |                  |
|   |  |                          |               |                      |          |                                      |         |       |   |                  |
|   |  |                          |               |                      |          |                                      |         |       |   |                  |
|   |  |                          |               |                      |          |                                      |         |       |   |                  |
|   |  |                          |               |                      |          |                                      |         |       |   |                  |
|   |  |                          |               |                      |          |                                      |         |       |   |                  |
|   |  |                          |               |                      |          | Lri                                  | 1       | Col 1 | Ch 1  | INS              |

#### 10 Sample Code and Run in Visual Studio --- .lst

| Project - Microsoft Visual Studio  |                              |            |        |           |       |  | - <b>-</b> 🗙 |
|--|------------------------------|------------|--------|-----------|-------|--|--------------|
| File Edit View Project Build Debug Team Data   | Tools Test V                 | /indow H   | elp    |           |       |  |              |
| 🛐 • 💷 - 🥔 🛃 🛃 👗 🐚 🖄 🕫 - 🔍 - 💭 • 🖳 🕨 Debu   | g • Win32                    |            | - 🙆    |           |       | - 🛛 🕾 🖬 🛛  | 🖻 🗶 💽 🚆      |
| 김 생님 ^ 個 孝孝 [김 일] 그 위 다 의 다 원 봐 있 .  |                              |            |        |           |       |  |              |
|  |                              |            |        |           | ÷     | Solution Explorer  | - ₽ ×        |
| AddSub.lst ×         AddSub.asm           C .NOLIST         C .LIST           C         .code           00000000         .code           00000000         main PROC           00000000         B8 00010000           00000000         B8 0000000           00000000         B8 0000000           00000000         B8 0000000 |                              |            |        |           | ÷     |  |              |
| C .LIST  |                              |            |        |           | ^     | Solution 'Proje  | ect' (1 proj |
| c  |                              |            |        |           |       | - I Project  | cor (1 proj  |
| 00000000 .code   |                              |            |        |           |       |  | Depender     |
| 00000000 main PROC   |                              |            |        |           |       | - 🔂 External   |              |
|  |                              |            |        |           |       | - 🚈 AddSub   | .asm         |
| 00000000 B8 00010000 mov eax,10000h  | ; EAX = 1000                 |            |        |           |       |  |              |
| 00000005 05 00040000 add eax,40000h<br>0000000A 2D 00020000 sub eax,20000h   | ; EAX = 5000<br>; EAX = 3000 |            |        |           |       |  |              |
| 0000000F E8 00000000 E call DumpRegs   | ; CAX = 5000                 | 2011       |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
| exit   |                              |            |        |           |       |  |              |
| 00000014 6A 00 * push +000000000h  |                              |            |        |           |       |  |              |
| 00000016 E8 00000000 E * call ExitProces<br>00000018 main ENDP   | s                            |            |        |           |       |  |              |
| END main   |                              |            |        |           |       | 4  | 2            |
| Microsoft (R) Macro Assembler Version 8.00.50727.4   | 2 01/28/06                   | 5 17:52:13 |        |           |       | Solution   | Team E       |
| *Add and Subtract (AddSub.asm  | Symbols 2 -                  | 1          |        |           |       | Contraction of the second seco | ream E       |
| 100 % - <  |                              |            |        |           | >     | Properties   | - 4 ×        |
|  |                              |            |        |           | -     |  | -            |
| Error List   |                              |            |        |           | ≁ ‡ × |  |              |
| 3 0 Errors 1 0 Warnings 0 Messages   | Eile.                        | 1.1        | Caluma | Duralizat |       |  |              |
| Description  | File                         | Line       | Column | Project   |       |  |              |
|  |                              |            |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
|  |                              |            |        |           |       |  |              |
| eady   |                              |            | Ln     | 1         | Col 1 | Ch 1   | INS          |
|  |                              |            |        |           |       |  |              |

## 11 Sample Code and Run in Visual Studio – When a new code is tested

| - 🖉 🖬 🖉   X 🖧 🖄   9 - 99 - 92 - 93   🕨 🛛  | Debug • Win32 |                        | - 29  | -   💀 🖻  | ' 🖬 🕺 🗶 🛃 🚆                                 |
|---|---------------|------------------------|---|--|---|
| & ^ 盾 # #   ] 2   □ ♥ ₽ ₽ ₽ <b>8</b> ₽  |               |                        |   |  |   |
| ISub.asm ×<br>TITLE Add and Subtract (AddSub.asm)<br>; This program adds and subtracts 32-bit integ<br>INCLUDE Irvine32.inc   | ers.          |                        |   | 📄 📓 Proje  | 'Project' (1 proj<br>ect<br>ternal Depender |
| .code<br>main PROC<br>mov eax,10000h ; EAX = 10000h<br>add eax,40000h ; EAX = 50000h<br>sub eax,20000h ; EAX = 30000h<br>call DumpRegs<br>exit<br>main ENDP<br>END main |               | 8)<br>8<br>8<br>1<br>1 | Open<br>Open With<br>View Class Diagram<br>Compile<br>Exclude From Project<br>Cut<br>Copy<br>Remove | Ctrl+F7  | dSub.asm<br>►<br>Team E<br>► ₽ ×            |
| Ye • <  |               | -                      | Rename<br>Properties  | Alt+Enter  | n File Propert -                            |
| or List<br>0 Errors 🛛 🛕 0 Warnings 🗐 0 Messages   |               |                        | Troperaes   |  |   |
| Description   | File          | Line                   | Column Project  | ■ Misc<br>(Name)<br>Content<br>File Type<br>(Name)<br>Names the fi | AddSub.asr<br>False<br>Document             |

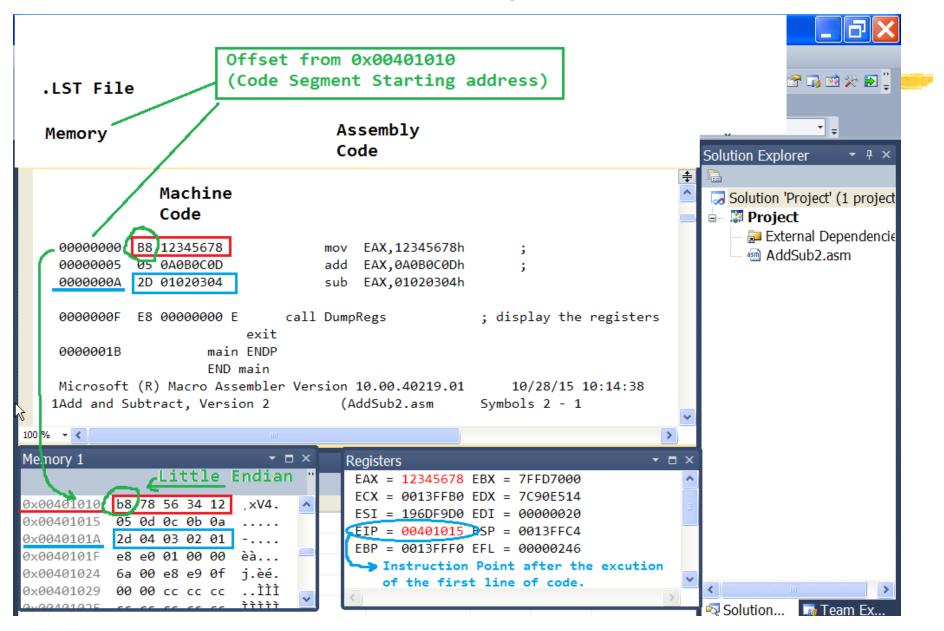
# 12 Sample Code and Run in Visual Studio --- Read in the new code

| Project - Microsoft Visual Studio   |        |  |             |                   |   |  |
|---|--------|--|-------------|-------------------|---|--|
| ✤ File Edit View Project Build Debug Team Data Too  | ols    |  |             |                   |   |  |
|   | -      | Win32 ·  | 2           | •                 | 💐 🖀 🖬 🖄 🗶 📜 📜   |  |
| I Server Explorer 🛠 Toolbox   |        | Build  |             |                   | n Explorer • • ×<br>ution 'Project' (1 proj<br><b>roject</b><br>• External Depender |  |
| rrer 🗞 Toolbox  |        | Rebuild<br>Clean<br>Project Only<br>Profile Guided Optimiz<br>Build Customizations |             | •                 | -   |  |
| <ul> <li>New Item Ctrl+Shift+A</li> <li>Existing Item Shift-t_Alt+A</li> </ul>            |        | Add<br>References<br>Class Wizard  |             | •<br>Ctrl+Shift+X |   |  |
| - New Hiter   | 2      | View Class Diagram<br>Set as StartUp Project<br>Debug                              |             | curromiterx       | ion 제 Team E<br>es 국무 ×   |  |
| Error List  | 2      | Add Solution to Source<br>Cut  | e Control   | Ctrl+X            | Project Propertie -   |  |
| <ul> <li>O Errors</li> <li>O Warnings</li> <li>O Messages</li> <li>Description</li> </ul> | ×      | Paste<br>Remove<br>Rename<br>Unload Project  |             | Ctrl+V<br>Del     | e) Project<br>t Dep<br>t File C:\Irvine\E   |  |
|   | ů<br>N | Rescan Solution<br>Open Folder in Window<br>Properties                             | ws Explorer | Alt+Enter         | )<br>s the project na   |  |
| ready<br><b>Start</b> 🖌 🖉 🖉 📽 🧿 🎽 ా Project   |        |  |             |                   | ) <b>F"® 3</b> 9:45 AM  |  |

#### 13 Sample Code and Run in Visual Studio ---New code

| Project (Debugging) - Microsoft Visual Studio  | _ 7   |
|--|---|
| File Edit View Project Build Debug Team Data Tools Test Window Help  |   |
|  | - 🔍 🖀 🖬 🕸 🎌 🗐 🚆 📥   |
| : 🖪 🗞 🐜 👘 (幸 幸) 🚍 😫 🔲 💭 🖓 💭 🥥 🧔 😓 🥋 📜 🕨 💷 🚇 🗇 📬 💭 🍲 🗐 🖓 📮  |   |
| Process: [3852] Project.exe - Thread: [4728] Main Thread - 🚩 🕅 Stack Frame: Project.exe!r  | main()Line 11 🔹 🚽   |
| AddSub2.asm ×  | Solution Explorer 🔹 म 🛪   |
| INCLUDE Irvine32.inc   | <b>陸 診 </b> 表   |
| .code<br>main PROC   | <ul> <li>Solution 'Project' (1 project)</li> <li>Project</li> <li>External Dependencie</li> </ul> |
| <pre>mov EAX,12345678h ; add EAX,0A0B0C0Dh ; sub EAX,01020304h</pre>   | MddSub2.asm   |
| call DumpRegs ; display the registers<br>exit<br>main ENDP<br>END main   |   |
| 100 % - <  |   |
| Memory 1   | 1 1   |
| Address: 0x00401010 - 🗇 Columns: Auto -  |   |
| 0x00401010 b8 78 56 34 12 05 0d 0c 0b 0a 2d 04 03 02 01 e8 e0 .xV4èà<br>0x00401021 01 00 00 6a 00 e8 e9 0f 00 00 cc cc cc cc cc ccj.èéÌÌÌÌÌÌÌ<br>0x00401032 cc 50 e8 d4 ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÈ |   |
| 0x00401043  0f 00 00 c3 55 8b ec 83 c4 e8 60 80 3d 00 50 40 00ÃU.ìfÄè`€=.P@.<br>0x00401054  00 75 05 e8 ff 04 00 00 8d 45 ea 50 ff 35 34 5b 40 .u.èÿEêPÿ54[@   |   |
| 0x00401065 00 e8 d3 0f 00 00 66 8b 45 ea 66 a3 11 56 40 00 81 .èÓf.Eêf£.V@   | Solution  |

#### 14.LST File and The Memory Contents of Code

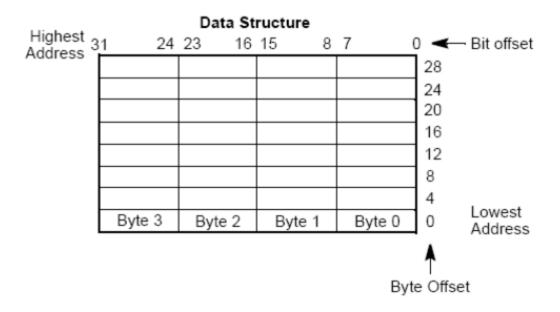


## **Notational Conventions**

🔀 Little Endian Machine

#### Hit and Byte Oder

- Smaller address at the bottom of figure
- Address increases toward top
- Bit positions numbered from right to left
- △ the bytes of a word are numbered starting from the least significant byte



| Integer ( | Constants        |      |                  |                   |      |                              |                               |
|-----------|------------------|------|------------------|-------------------|------|------------------------------|-------------------------------|
| [{+/-]    | }] digits [radix | ]    | main PROC<br>mov | eax,5             |      | ; move 5 to                  | the EAX register              |
| 🔼 Radix   |                  |      | add<br>call      | eax,6<br>WriteInt |      | ; add 6 to t<br>; display va | he EAX register<br>lue in EAX |
| ⊠H        | hexadecimal      |      | exit             |                   |      | ; quit                       | fue in DAA                    |
| ⊠q/       | o Octal          |      | main ENDP        |                   |      |                              |                               |
| ×D        | Decimal {Defau   | ılt} |                  |                   |      |                              |                               |
| ⊠b        | Binary           |      |                  |                   |      |                              |                               |
| ⊠ E>      | kample:          |      |                  |                   |      |                              |                               |
|           | 26               | Deci | mal              | 4                 | 120  | Octal                        |                               |
|           | 26d              | Deci | mal              | 1                 | LAh  | Hexadec:                     | imal                          |
|           | 11010011b        | Bina | ry               | 7                 | )A3h | Hexadec:                     | imal                          |
|           | 42q              | Octa | 1                |                   |      |                              |                               |
|           |                  |      |                  |                   |      |                              |                               |

×Note: Hex constant beginning with a letter must have a leading 0

### **#**Character Constants

A single character enclosed in single or double quotes × "A"

⊠"d″

MASM stores the value in memory as the character's binary ASCII code

### **String Constants**

△A sequence of characters (including spaces) enclosed in single or double quotes

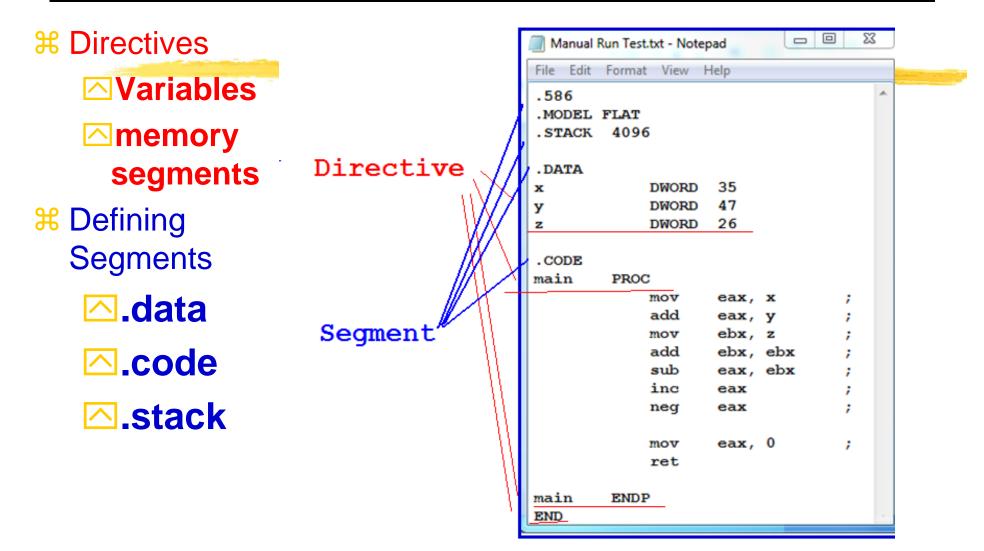
ĭ×'ABC′

⊠"Good night, Gracie"

Say "Good night," Gracie'

#### **#** Directives

- A command embedded in the source code that is recognized and acted upon by the assembler
- Directives can define variables, macros, and procedures
- Directives can assign names to memory segments
- Directives do not execute at runtime
- ➢ Directives are case insensitive in MASM
- **#** Defining Segments (or Program Sections)
  - △.data ; the area of program containing variables
  - **code** ; the area of a program containing executable instructions
  - .stack ; the area of a program holding the runtime stack (with its size)



#### **Histruction**:

"A statement that becomes executable when a program is assembled" – Instructions are translated by the assembler into machine language bytes, which are loaded and executed by the CPU at runtime

#### Instruction Format

#### Label, Instructional mnemonic, argument1, argument2, argument3

**Label:** Identifier (followed by a colon)

Mnemonic: a reserved name for a class of instruction op-codes which have the same function

#### **Operands (arguments)**:

 $\boxtimes$ 0 to 3 operands

☑2 types of form: literals (i.e., number) or identifiers for data items.

**#Instruction Format** 

When two operands are present in an arithmetic or logical instruction

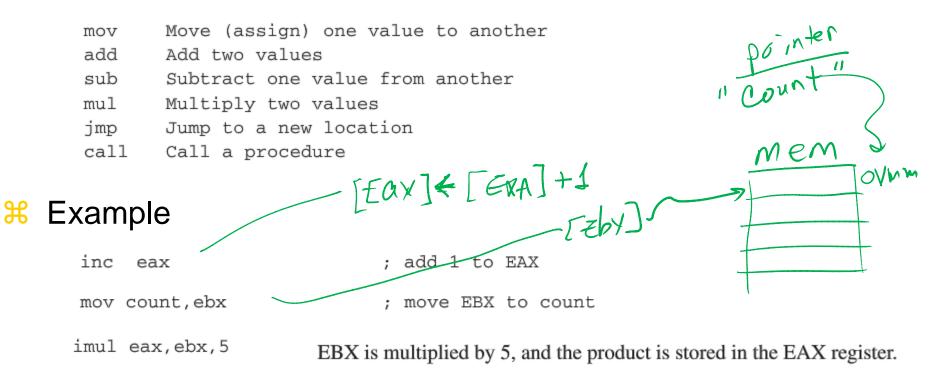
- ⊠ the right operand is the **source (src)** and
- ☑ the left operand, the destination (dst)

## Example:

| Count    | DWORD    | 100           | ; data label |
|----------|----------|---------------|--------------|
| LOADREG: | MOV      | EAX, SUBTOTAL |              |
| ;label   | mnemonic | dst, src      |              |

Instruction Mnemonic: a short word that identifies an instruction

Example



[eax] ~ [ebx] \* 5

**#** Comments

H

- **%** Single Line Comments (;)
- Block Comments: Begin with the COMMENT directive and a user-specified symbol

```
COMMENT !
This line is a comment.
This line is also a comment.
!
COMMENT &
This line is a comment.
This line is also a comment.
&
```

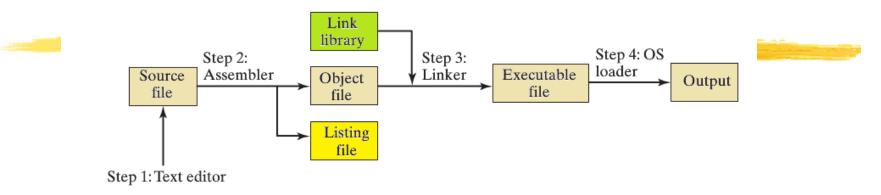
### **XNOP** (No Operation) Instruction

- Takes up 1 byte of program storage
- △Does not do any work
- Use when to align code to even-addressed boundaries (which, in x86, loads data more quickly.)

| 00000000 | 66 8B C3 | mov ax,bx   |   |       |      |             |
|----------|----------|-------------|---|-------|------|-------------|
| 00000003 | 90       | nop         | ; | align | next | instruction |
| 00000004 | 8B D1    | mov edx,ecx |   |       |      |             |

## Assembling and Running (Debugging) Programs

#### Assemble-Link-Execute Cycle



# 1: Create an ASCII text file named *source file* using a text editor

- NOTE: Starting from an existing project (and source file) and revising and saving as a new .asm file is most highly recommended
- 2: Assemble the source file. The assembler reads the source file and produces an *object file*, a machine language translation of the program. Optionally, it produced a *listing file*. {VS: BUILD SOLUTION}
- **3**: Execute the program (by Debug)

#### Assembling and Running Programs in Visual Studio

- Step 1: Open an existing project and Revise and Save as a new file name using
- Step 2: Assemble the source file (by *Build*).
- **Step 3: Debug (F10 key for line execution)**

|      | Aicrosoft Visual Studio        |            |    |                  |              |         |                         |
|------|--------------------------------|------------|----|------------------|--------------|---------|-------------------------|
| Þ    | File Edit View Debug Team Data | Tools Test | N  | /indow Help      |              |         |                         |
|      | New                            | •          |    |                  | - 10         |         | • 💀 🕾 🖬 🖄 🛠 🖬 📜         |
|      | Open                           | •          | â  | Project/Solution | Ctrl+Shift+C | )       | Solution Explorer - 7 × |
|      | Close                          |            | 8  | Web Site         | Shift+Alt+O  |         |                         |
| , B  | Close Solution                 |            |    | Team Project     |              |         |                         |
| iel. |                                | +S         | 2  | File             | Ctrl+O       |         |                         |
|      | Save Selected Items As         |            |    | Convert          |              |         |                         |
|      |                                | +Shift+S   |    |                  |              |         |                         |
|      | Export Template                |            |    |                  |              |         |                         |
|      | Source Control                 | •          |    |                  |              |         |                         |
|      | Page Setup                     |            |    |                  |              |         |                         |
| 9    |                                | +P         |    |                  |              |         |                         |
|      | Recent Files                   | •          |    |                  |              |         |                         |
|      | Recent Projects and Solutions  | •          |    |                  |              |         |                         |
|      | Exit Alt-                      | -F4        |    |                  |              |         |                         |
|      |                                |            |    |                  |              |         |                         |
|      |                                |            |    |                  |              |         |                         |
|      | Error List                     |            |    |                  |              | * å ×   |                         |
|      | O Errors 0 Warnings 0 Mes      | sages      |    |                  |              |         |                         |
|      | Description                    |            |    | File Lin         | e Column     | Project |                         |
|      |                                |            |    |                  |              |         |                         |
|      |                                |            |    |                  |              |         |                         |
|      |                                |            |    |                  |              |         |                         |
|      |                                |            |    |                  |              |         | Solution 📭 Team E       |
| Rea  | dv                             |            |    |                  |              |         |                         |
| _    |                                |            |    |                  |              |         |                         |
| -    | start 🧉 🔍 🕫 🖻 🖬                | 9 0        | Mi | crosoft Visu     |              | BN 🔇    | 🗛 🎝 🔞 🕬 🖉 🌛 9:43 AM     |

### Assembly Language Fundamentals - Summary

| Adding and Subtracting         | TITLE Add and Subtract  | (AddSub.asm)                 |
|--------------------------------|-------------------------|------------------------------|
| Integers                       | . This program adds and | aubtraata 32-bit intogora    |
| HITLE directive                | ; INIS program adds and | l subtracts 32-bit integers. |
| Hine comment                   | INCLUDE Irvine32.inc    |                              |
| INCLUDE directive              | .code                   |                              |
|                                | main PROC               |                              |
| 8 .code directive              |                         |                              |
| PROC directive – start         | marr                    | . EXY - 1000b                |
|                                | mov eax,10000h          | ; EAX = $10000h$             |
|                                | add eax,40000h          | ; EAX = $50000h$             |
| CALL a procedure               | sub eax,20000h          | ; EAX = 30000h               |
| 🔀 Exit halt to program         | call DumpRegs           | ; display registers          |
| (Not a MASM keyword,           | 1 5                     | ,                            |
| but of Irvine32.inc)           | exit                    |                              |
| ENDP directive – end           |                         |                              |
| # END directive – last line to | main ENDP               |                              |
| be assembled                   | END main                |                              |
|                                |                         |                              |
| Houtput (DumpReas)             |                         |                              |
| EAX=00030000 EBX:              | =7FFDF000 ECX=00000101  | EDX=FFFFFFFF                 |
| ESI=00000000 EDI:              | =00000000 EBP=0012FFF0  | ESP=0012FFC4                 |

CF=0

SF=0

ZF=0

EIP=00401024

EFL=00000206

PF=1

OF=0 AF=0