

The University of Texas at Arlington

Lecture 3 PIC Assembly Basics



CSE 3442/5442
Embedded Systems I

Based heavily on slides by Dr. Roger Walker



From Last Lecture

- Download MPLAB and C18
- Link for MPLAB IDE

[http://www.microchip.com/stellent/idcplg?
IdcService=SS_GET_PAGE&nodeId=140
6&dDocName=en019469&part=SW00700
2](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002)

- Link for C18:

[http://www.microchip.com/stellent/idcplg?
IdcService=SS_GET_PAGE&nodeId=140
6&dDocName=en010014](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014)



Installation of Assembler/C Compiler

- http://www.microdigitaled.com/PIC/PIC_books.htm
- See:
<http://www.microdigitaled.com/pic/tutorials/MPLABInstall.pdf> for installing
- <http://www.microdigitaled.com/pic/tutorials/C18Install.pdf> for installing C18
- <http://www.microdigitaled.com/pic/tutorials/MPLAB.pdf> for MPLAB Tutorial



Compiler Data Formats

- Data Types - hex, decimal, binary, ASCII
- Hex:
 - Use h (or H) right after the number: `MOVLW 99H`
 - Put 0x (or 0X) in front of the number: `MOVLW 0x99H`
 - Put nothing in front or back of the number: `MOVLW 99`
 - Put h in front of the number, with single quotes around the number: `MOVLW h'99'`



Compiler Data Formats

- Binary:
 - Put B in front of the number in single quotes:
 - `MOVLW B'10011001'`
- Decimal:
 - Put D in front of the number in single quotes:
 - `MOVLW D'12'`
 - Use `.value`:
 - `MOVLW .12`



Compiler Directives

- Compiler directives: give directions to the compiler.
 - EQU (defining constants), (SET is similar but can be reset)
 - DATA1 EQU 39H - Data assignment
 - PORTB EQU 0xFF6 – SFR address assignment
 - MYREG EQU 0x12 – RAM address assignment
 - ORG (origin - explicit address offset operand must be hex)
 - END (tells assembler that this is end of code)

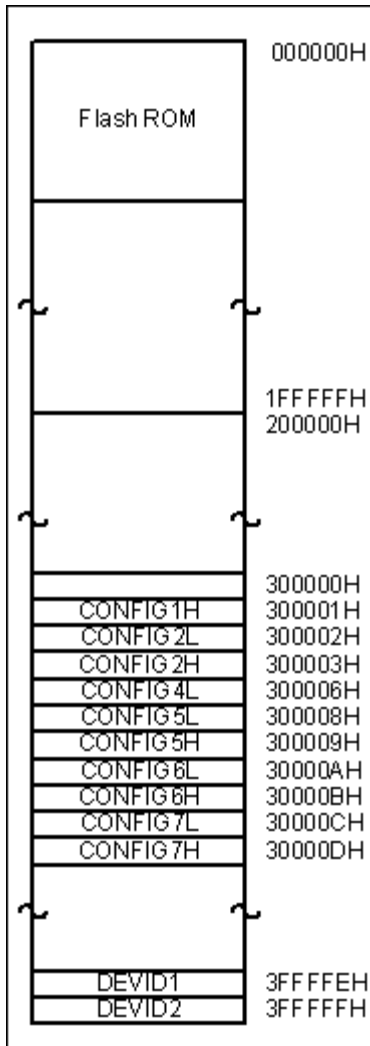


Compiler Directives

- LIST (indicates specific controller, e.g., LIST P=18F452) unique to PIC assembler.
- #include (to include libraries associated)
- _config directives – tell assembler what the configuration (stored at 300000H) bits of the target PIC should be
- radix (e.g., radix dec will change to decimal notation; default is hex)



Configuration Registers



File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h CONFIG1H	—	—	OSCSEN	—	—	FOSC2	FOSC1	FOSC0	--1- -111
300002h CONFIG2L	—	—	—	—	BORV1	BORV0	BOREN	PWRTEN	---- 1111
300003h CONFIG2H	—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN	---- 1111
300005h CONFIG3H	—	—	—	—	—	—	—	CCP2MX	---- -1
300006h CONFIG4L	DEBUG	—	—	—	—	LVP	—	STVREN	1---- -1-1
300008h CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0	---- 1111
300009h CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0	---- 1111
30000Bh CONFIG6H	WRD	WRB	WRTC	—	—	—	—	—	111- ----
30000Ch CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFFFh DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	(1)
3FFFFFFh DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 0100

Table 19-1 from Data Sheet



Assembly Language Instruction

- Assembly Language instruction contains four fields:
 - Label
 - Mnemonic
 - Operands
 - Comments

[label] mnemonics [operands] [;comments]



Assembly Programming sample

```
SUM EQU 10H ; RAM loc 10H for SUM
ORG 0H ; start at address 0
MOVLW 25H ;25H → WREG
ADDLW 0x34 ;+ 34H
ADDLW 11H ;+ 11H
ADDLW 25 ;+25H
ADDLW D'18 ' ;+ 18 decimal
ADDLW .18 ;+18 decimal
ADDLW A'0' ; add ascii of 0 (32)
ADDLW B'00000110 ' ;+6
MOVWF SUM
HERE GOTO HERE
END
```

From page 69



Assembly Assembled and Linked

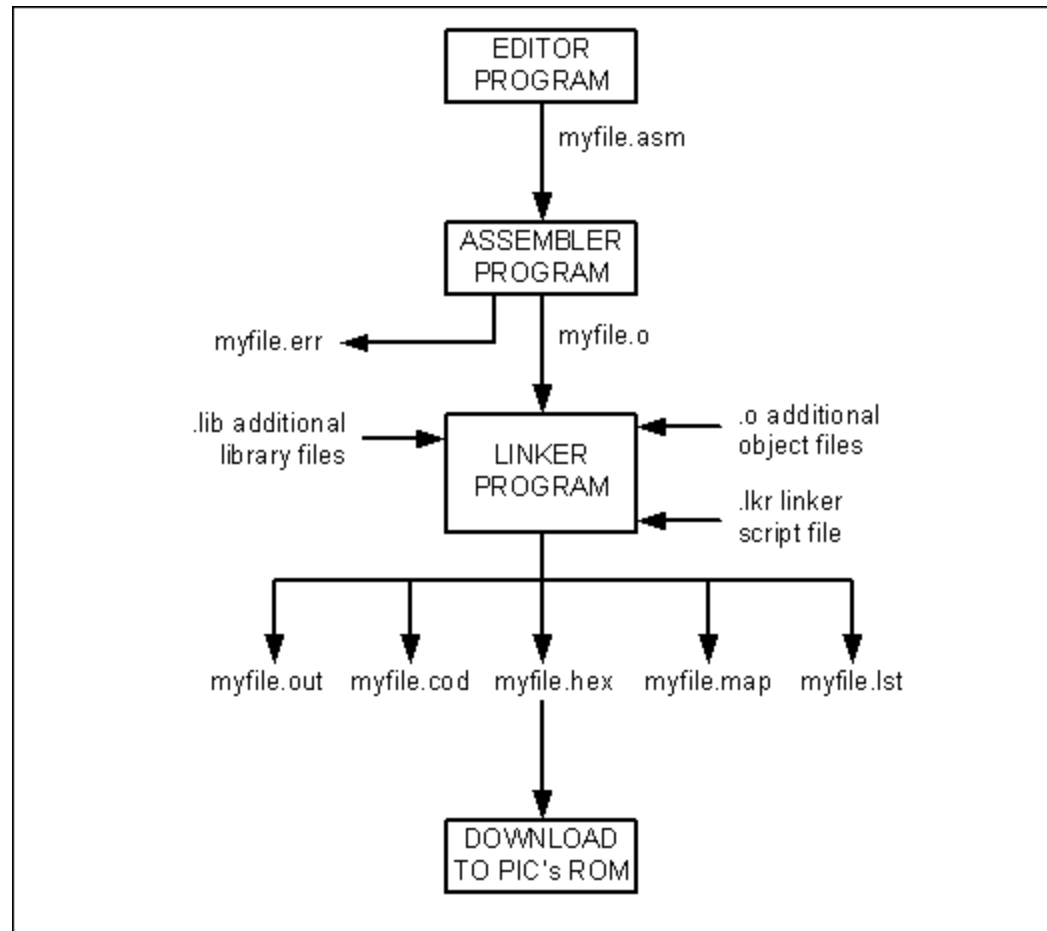


Figure 2-8.



Files Used and Created

- .hex → PIC ROM (loadable object)
- .mcp : MPLAB Project mcp
- .mcw: MPLAB Workspace
- .cod Code Listing
- .lst Program listing (machine code with assembly comments)
- .map a file containing memory layout of used and unused locations
- .o immediate object
- .err Debug information

PIC18 Program Counter

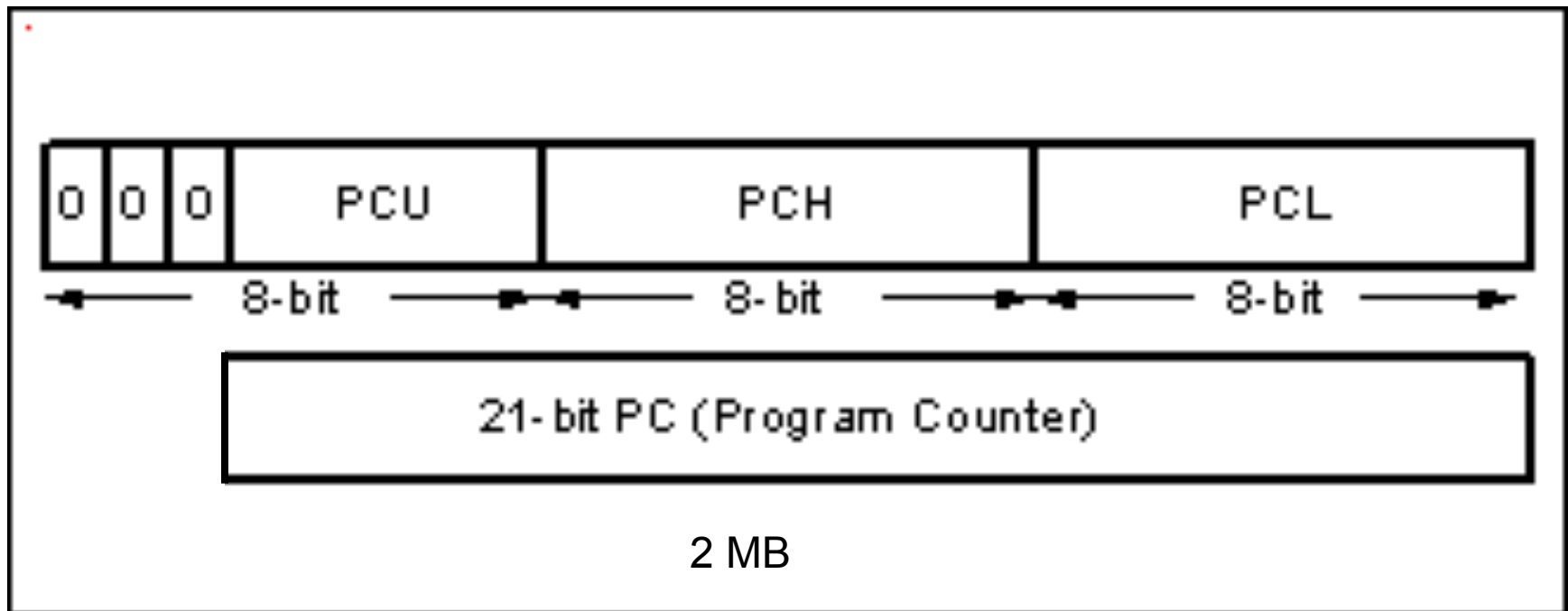


Figure 2-9.



PIC18 On-Chip Program ROM Address Range

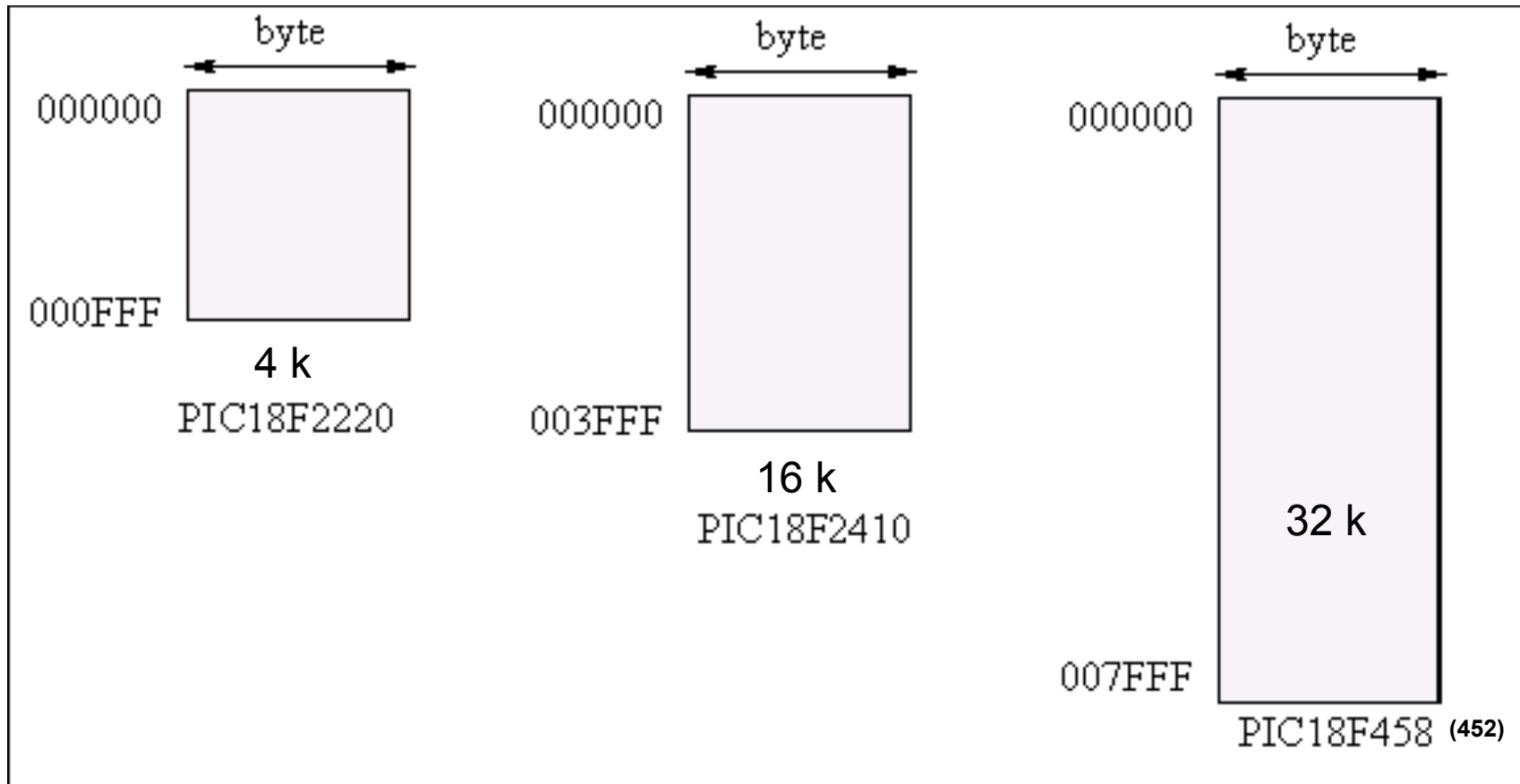
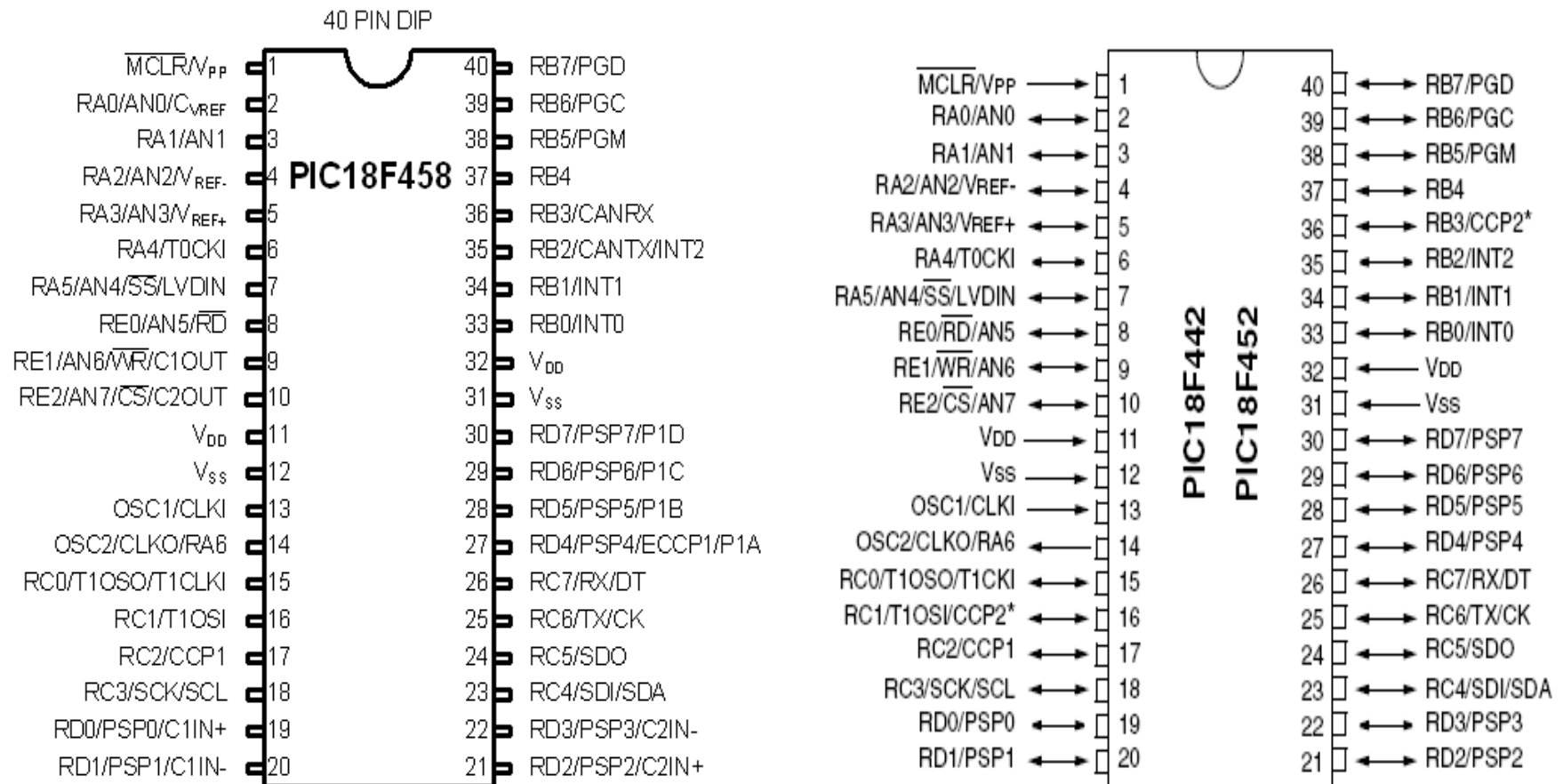


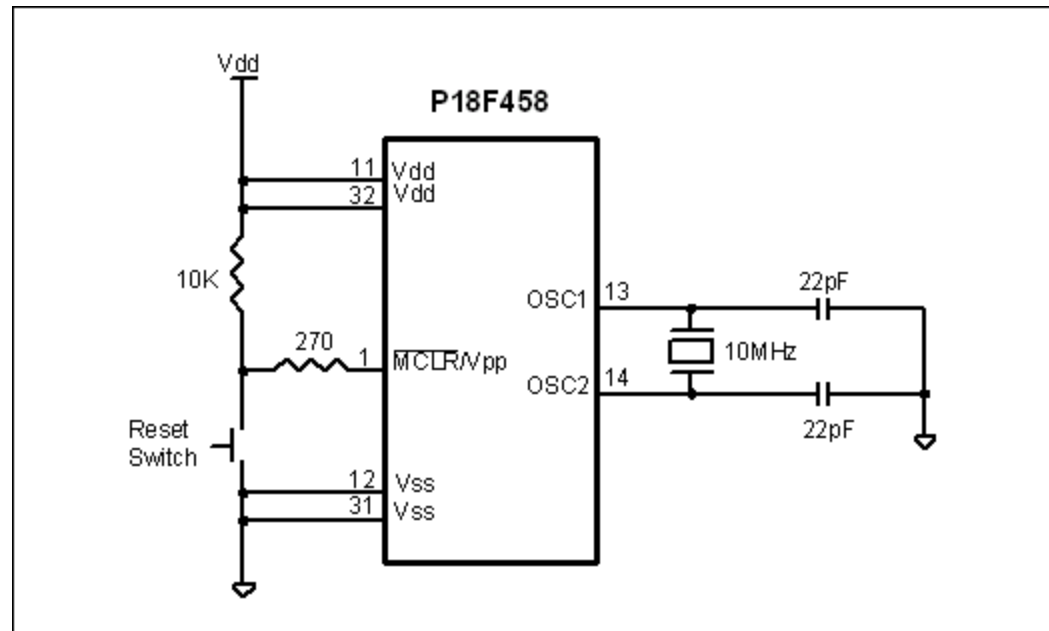
Figure 2-10



PIC18F458 Pin Diagram



Example - Powering Up PIC18F458

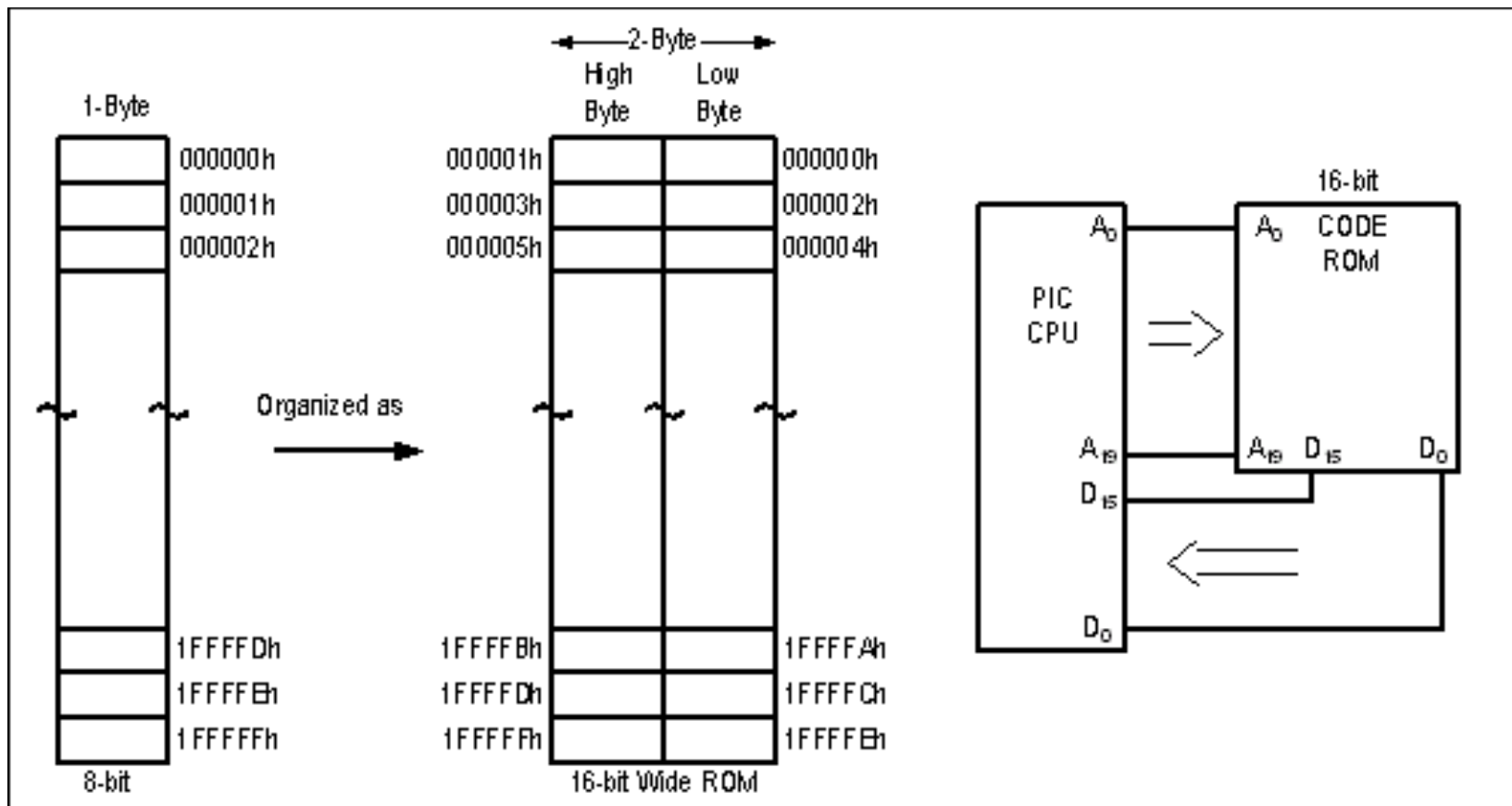




Programs in ROM

- When PIC is powered up (VCC applied to Reset Pin – Chapter 8), the microcontroller begins executing instruction at location 00000h (Reset Vector).
- Use ORG statement for this instruction in your code (if programming in assembly). C compiler takes care of creating assembly code having this.

PIC18 Program ROM Width





PIC18 Program ROM Contents for Program 2-1 List File

WORD ADDRESS	HIGH BYTE	LOW BYTE
000000h	0Eh	25h
000002h	0Fh	34h
000004h	0Fh	11h
000006h	0Fh	12h
000008h	0Fh	1Ch
00000Ah	0Fh	06h
00000Ch	6Eh	10h
00000Eh	EFh	07h
000010h	0Fh	00h

Memory is structured in *'little endian'* mode
(lowest value at lowest address)

Program 2-1 List

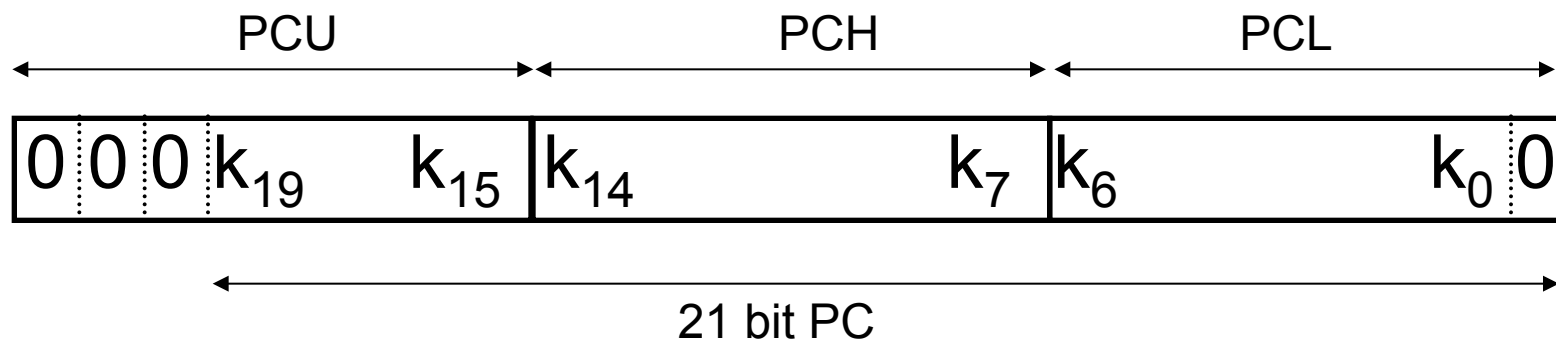


GOTO and the PC

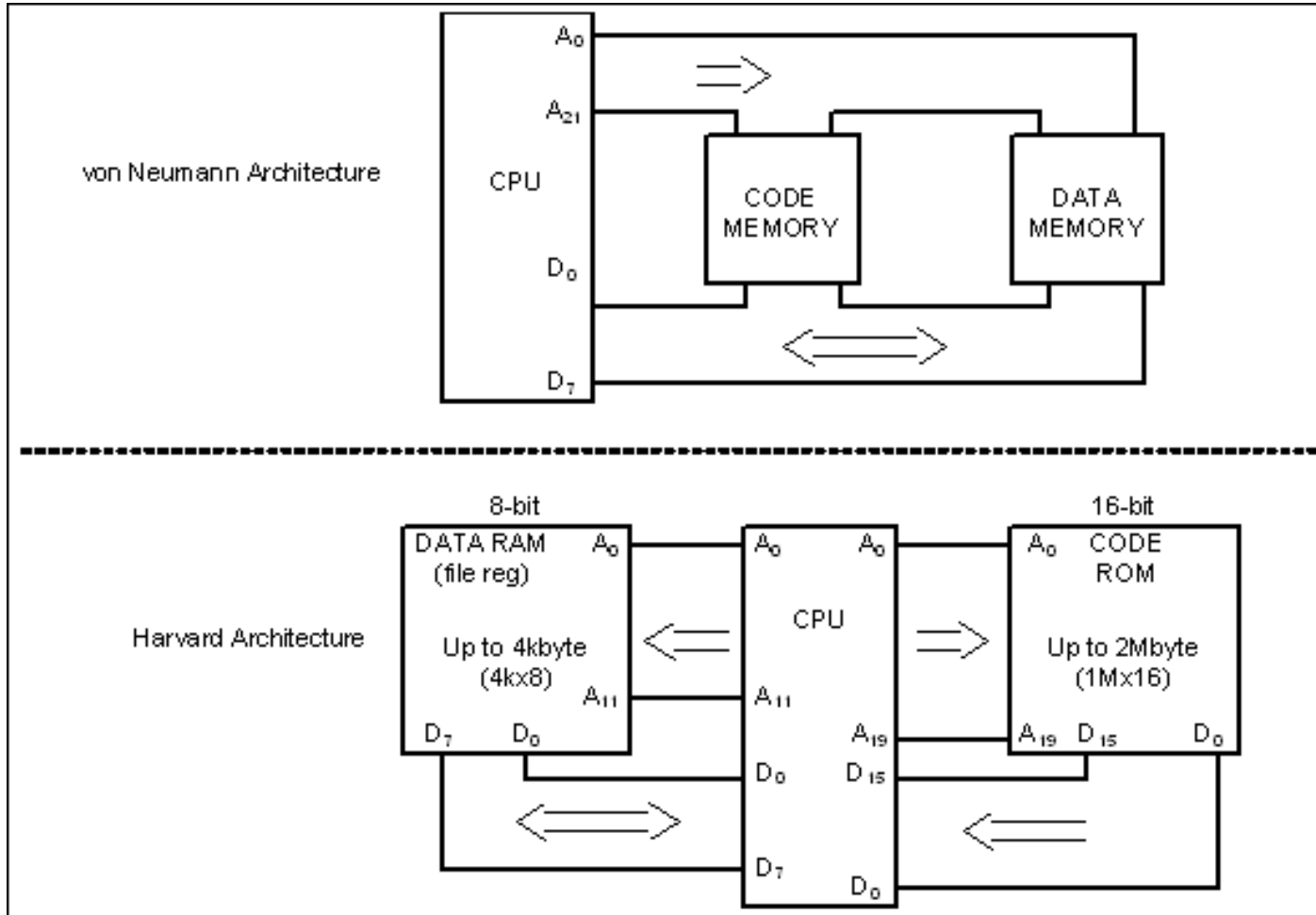
- GOTO, 4 byte instruction:

0E	07
0F	EF
10	00
11	F0

little endian!



PIC uses Harvard Architecture





RISC

1. Fixed instruction size (2 and 4 bytes in PIC ; ADD, GOTO)
2. Many registers (no need for large stack)
3. Small instruction set – longer code
4. Small clock cycle/instruction
5. Usually Harvard architecture
6. No microcoding; instructions are internally hardwired – can result in 50% reduction in the number of transistors
7. No cross operations between GFR registers



MPLAB Simple Assembly Sample

- Create project
- Type in code
- Set debugger to simulator
- Add watch windows
- Look at what is happening

prog1.mcp

- prog1.mcp
 - Source Files
 - PROGRAM1.ASM

C:\VCSE3442_5342\PROGRAM1.ASM

```
SUM EQU 10H
ORG 0H
MOVLW 25H
ADDLW 0x34
ADDLW 11H
ADDLW D'18'
ADDLW 1CH
ADDLW B'00000110'
MOVWF SUM
HERE GOTO HERE
END
```


- Project Wizard...
- New...
- Open...
- Close
- Set Active Project
- Quickbuild (no .asm file)
- Clean
- Export Makefile
- Build All** Ctrl+F10
- Make F10
- Build Configuration
- Build Options...
- Save Project
- Save Project As...
- Add Files to Project...
- Add New File to Project...
- Remove File From Project
- Select Language Toolsuite...
- Set Language Tool Locations...
- Version Control...

Output window showing build logs:

```
Build Ver
Debug bu
Preproce
Mon Sep
Clean: De
Clean: Dc
Executing
Warning[
Loaded C
Debug bu
Preproce
Mon Sep
BUILD S
```

Editor window showing code:

```
3 - Lec
uite\W
11 : F
3 - Lec
```

Output

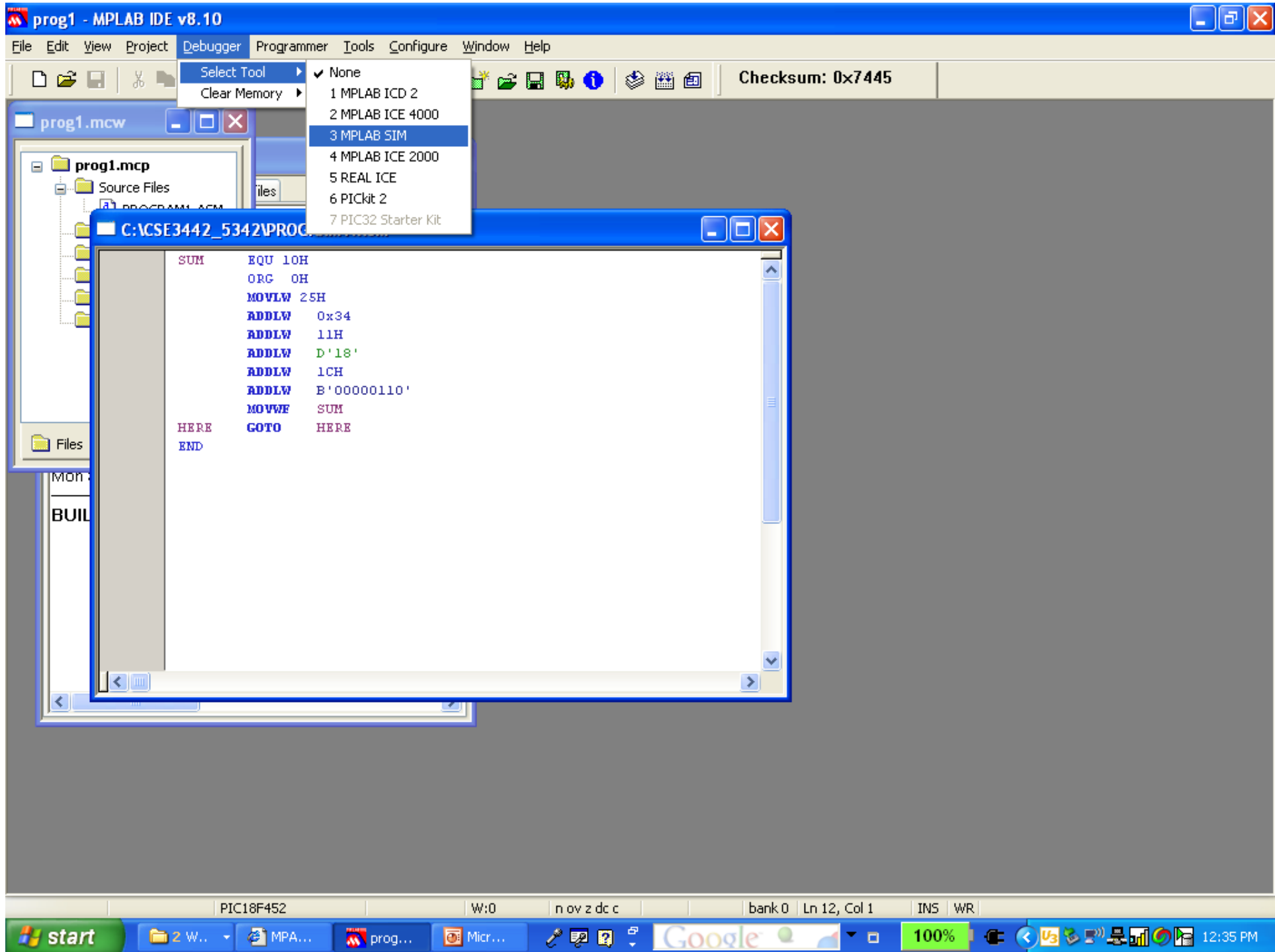
Build Version Control Find in Files

Debug build of project `C:\CSE3442_5342\Fall 2008 - Lec
Preprocessor symbol `__DEBUG` is defined.
Mon Sep 01 12:28:25 2008

Clean: Deleting intermediary and output files.
Clean: Done.
Executing: "C:\Program Files\Microchip\MPASM Suite\W
Warning[205] C:\CSE3442_5342\PROGRAM1.ASM 11 : F
Loaded C:\CSE3442_5342\PROGRAM1.cod.

Debug build of project `C:\CSE3442_5342\Fall 2008 - Lec
Preprocessor symbol `__DEBUG` is defined.
Mon Sep 01 12:28:26 2008

BUILD SUCCEEDED



The screenshot shows the MPLAB IDE interface. A file explorer on the left displays the project structure for 'prog1.mcp', including 'Source Files' and 'PROGRAM1.ASM'. The main editor window, titled 'C:\VCSE3442_5342\PROGRAM1.ASM', contains the following assembly code:

```
SUM EQU 10H
ORG 0H
MOVLW 25H
ADDLW 0x34
ADDLW 11H
ADDLW D'18'
ADDLW 1CH
ADDLW B'00000110'
MOVWF SUM
GOTO HERE
HERE
END
```

A green arrow points to the 'HERE' label on the left side of the code editor.

prog1.mcp

- prog1.mcp
 - Source Files
 - PROGRAM1.ASM
 - Header Files
 - Object Files
 - Library Files
 - Linker Script
 - Other Files

Files Symbols

MPLAB SIM

mer event occurred. Break i
mer event occurred. Break i
mer event occurred. Break i

Watch

Add SFR WREG Add Symbol 18F452

Update	Address	Symbol Name	Value
	FE8	WREG	0x25

Watch 1 Watch 2 Watch 3 Watch 4

C:\CSE3442_5342\PROGRAM1

```
SUM EQU 10H
ORG OH
MOVLW 25H
ADDLW 0x34
ADDLW 11H
ADDLW D'18'
ADDLW 1CH
ADDLW B'00000110'
MOVWF SUM
GOTO HERE
HERE
END
```

prog1.mcp

- Source Files
 - PROGRAM1.ASM
- Header Files
- Object Files
- Library Files
- Linker Script
- Other Files

Files Symbols

MPLAB SIM

mer event occurred. Break i

mer event occurred. Break i

mer event occurred. Break i

Watch

Add SFR WREG Add Symbol 18F452

Update	Address	Symbol Name	Value
	FE8	WREG	0x59

Watch 1 Watch 2 Watch 3 Watch 4

```
SUM EQU 10H
ORG OH
MOVLW 25H
ADDLW 0x34
ADDLW 11H
ADDLW D'18'
ADDLW 1CH
ADDLW B'00000110'
MOVWF SUM
GOTO HERE
HERE
END
```

prog1.mcp

- Source Files
 - PROGRAM1.ASM
- Header Files
- Object Files
- Library Files
- Linker Script
- Other Files

Files Symbols

MPLAB SIM

mer event occurred. Break i
mer event occurred. Break i
mer event occurred. Break i

Watch

Add SFR WREG Add Symbol 18F452

Update	Address	Symbol Name	Value
	FES	WREG	0x9E

C:\CSE3442_5342\PROGRAM1.ASM

```
SUM EQU 10H
ORG 0H
MOV LW 25H
ADD LW 0x34
ADD LW 11H
ADD LW D'18'
ADD LW 1CH
ADD LW B'00000110'
MOV WF SUM
GOTO HERE
HERE
END
```



Special File Register Window in MPLAB Simulator

Address	SFR Name	Hex	Decimal	Binary	Char
0F80	PORTA	00	0	00000000	.
0F81	PORTB	00	0	00000000	.
0F82	PORTC	00	0	00000000	.
0F83	PORTD	00	0	00000000	.
0F84	PORTE	00	0	00000000	.
0F89	LATA	00	0	00000000	.
0F8A	LATB	00	0	00000000	.
0F8B	LATC	00	0	00000000	.
0F8C	LATD	00	0	00000000	.
0F8D	LATE	00	0	00000000	.
0F92	TRISA	00	0	00000000	.
0F93	TRISB	00	0	00000000	.
0F94	TRISC	00	0	00000000	.
0F95	TRISD	00	0	00000000	.
0F96	TRISE	00	0	00000000	.
0F9D	PIE1	00	0	00000000	.
0F9E	PIR1	00	0	00000000	.
0F9F	IPR1	00	0	00000000	.
0FA0	PIE2	00	0	00000000	.
0FA1	PIR2	00	0	00000000	.
0FA2	IPR2	00	0	00000000	.



File Register (Data RAM) Window in MPLAB Simulator

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010	0E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Hex Symbolic



Program (Code) ROM Window in MPLAB Simulator

The screenshot shows the 'Program Memory: 1' window in the MPLAB Simulator. It displays a table of assembly instructions with columns for Line, Address, Opcode, and Disassembly. The first instruction is highlighted with a green arrow pointing to the line number '1'.

Line	Address	Opcode	Disassembly
1	C000	0E0A	MOVLW 0xA
2	C002	6E25	MOVWF 0x25, ACCESS
3	C004	0E00	MOVLW 0
4	C006	0F03	ADDLW 0x3
5	C008	0625	DECF 0x25, F, ACCESS
6	C00A	E1FD	DNZ 0x6
7	C00C	6E21	MOVWF 0x21, ACCESS

At the bottom of the window, there are three tabs: 'Opcode Hex', 'Machine', and 'Symbolic'. The 'Symbolic' tab is currently selected.



Assignment for Next Class

- Review the material in Chapter 3.
- Pay close attention to the instruction word sizes and target address for the BRA, GOTO, and CALL instructions, and for the Stack implementation.
- Using the MPLAB assembler and simulator, code and run the example on page 115 (Example 3-11), adding appropriate break points for using the simulator to display the stack contents after one of the call examples.



PIC Data Sheets

- Note, the PIC18f452
<http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>
- PIC18f458
<http://ww1.microchip.com/downloads/en/DeviceDoc/41159e.pdf>