Programming and Problem Solving

ELEC 330

Digital Systems Engineering

Dr. Ron Hayne

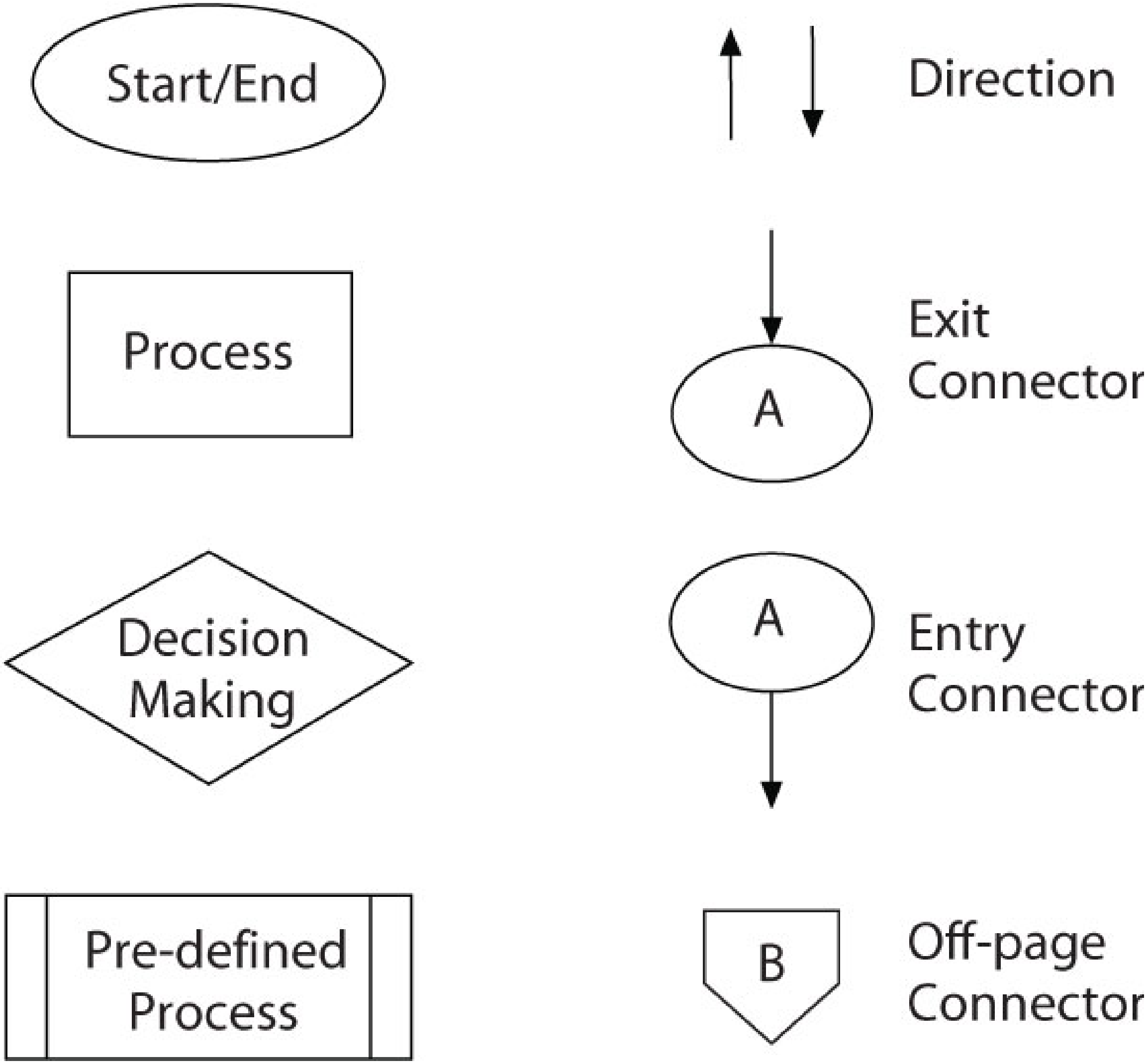
Images Courtesy of Ramesh Gaonkar and Delmar Learning

THE CITADEL
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Problem Solving

- Modular Design
  - Divide problem into various tasks
- Subroutines
  - Independent units that can be combined
- Flowchart
  - Graphical representation of processes (tasks)
  - Sequence to be followed

Flowcharting



Start/End

Process

Decision Making

Pre-defined Process

Direction

A

Exit Connector

A

Entry Connector

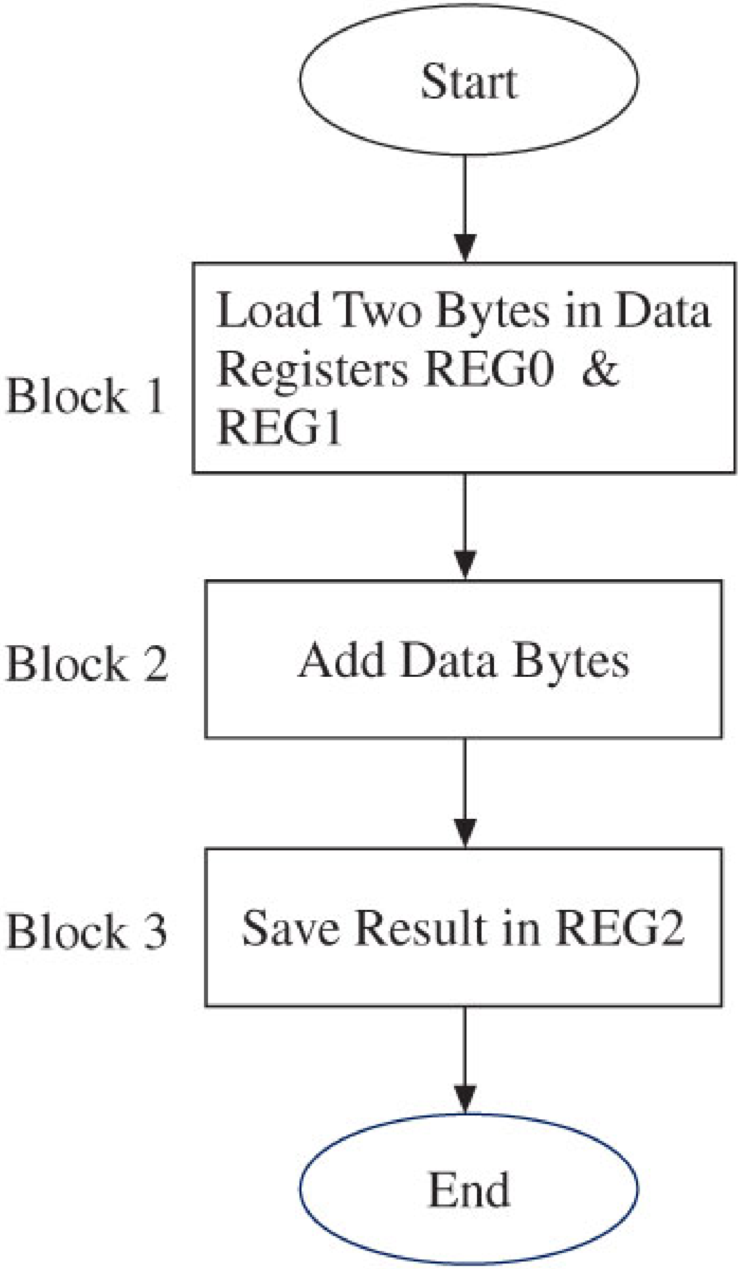
B

Off-page Connector

Example 4.1

- Write instructions to load two bytes (37_H and 92_H) in data registers REG0 and REG1. Add the bytes and store the sum in REG2.
- Steps
  - Load two bytes in data registers REG0 and REG1
  - Add the bytes
  - Save the sum in data register REG2

Example 4.1


Start
Block 1
Load Two Bytes in Data Registers REG0 & REG1
Block 2
Add Data Bytes
Block 3
Save Result in REG2
End

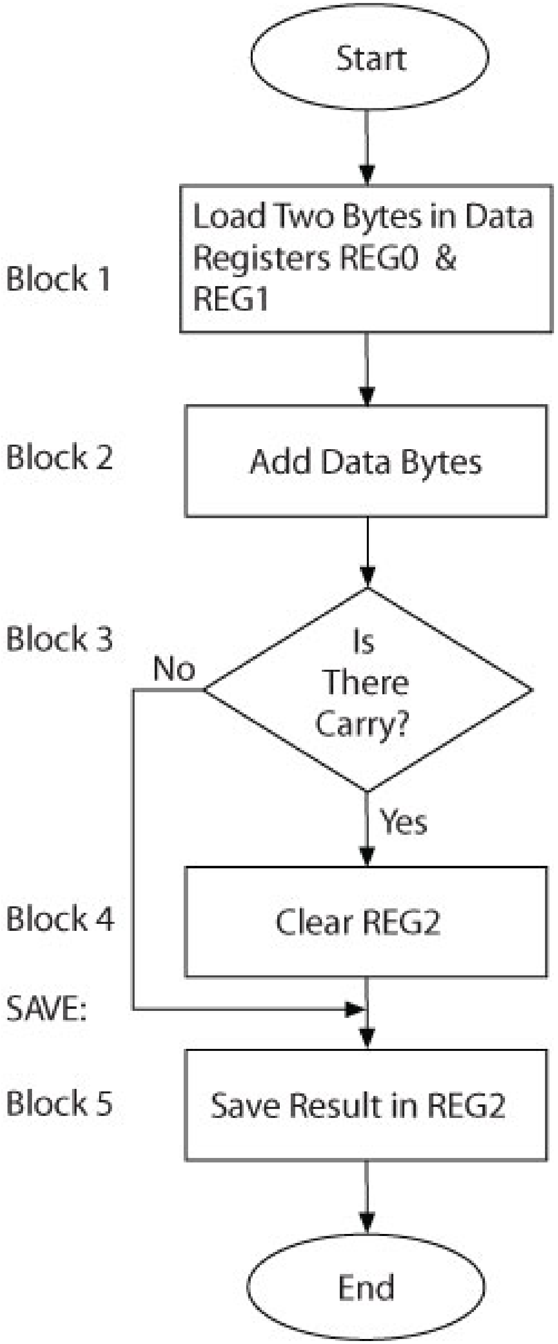
Assembly Language Programming

- Analyze the problem
- Draw a flowchart
- Convert the flowchart into mnemonics
- Look up Hex code and assign memory addresses
- Enter the Hex code into memory
- Execute the program
- Debug the program if necessary

Illustrative Program

- Addition With Carry Check
  - Write instructions to load two bytes ($F2_H$ and 32_H) in data registers REG0 and REG1 and add the bytes.
  - If the sum generates a carry, clear the data register REG2; otherwise, save the sum in REG2.

Illustrative Program



Assembly Language

Label	Opcode	Operand	Comments
START:	MOVLW	0xF2	;Load first byte in W
	MOVWF	REG0	;Save first byte in REG0
	MOVLW	0x32	;Load second byte in W
	MOVWF	REG1	;Save second byte in REG1
	ADDWF	REG0,W	;Add bytes and save sum in W
	BNC	SAVE	;Branch if no carry
	MOVLW	0	;Clear W
SAVE:	MOVWF	REG2	;Save sum in REG2
	SLEEP		;Power Down

Code Look-up?

Address	Hex	Label	Opcode	Operand
000020	0EF2	START:	MOVLW	0xF2
000022	6E00		MOVWF	REG0
000024	0E32		MOVLW	0x32
000026	6E01		MOVWF	REG1
000028	2400		ADDWF	REG0,W
00002A	E301		BNC	SAVE
00002C	0E00		MOVLW	0
00002E	6E02	SAVE:	MOVWF	REG2
000030	0003		SLEEP	

Execute the Program

- Hardware
  - MPLAB IDE
  - PICDEM™ 2 Plus Demo Board
  - Microchip PIC18F4520 MPU
  - MPLAB ICD2
- Simulation
  - PIC18 Simulator IDE
  - Microchip PIC18F452 MPU

PIC18 Simulator IDE

- Integrated Development Environment
  - Editing
  - Assembling
  - Linking
  - Downloading
  - Executing
  - Simulation
  - Debugging

Using an Assembler

- Assembly Language Programs
  - Program in Mnemonics
  - Assembler Directives
  - Comments
- Assembly Language Statements (4 fields)
  - Label
  - Opcode
  - Operand (data, register, or memory address)
  - Comment

Assembly Language Format

Label	**Opcode**	**Operand**	**Comment**
START:	MOVLW	0xF2	;Load F2H in W

↑ ↑ ↑ ↑

Tab Tab Tab Semicolon

Assembler Directives

#INCLUDE	Include resources from library
ORG	Origin
EQU	Equate
DB	Data byte
END	End of assembly

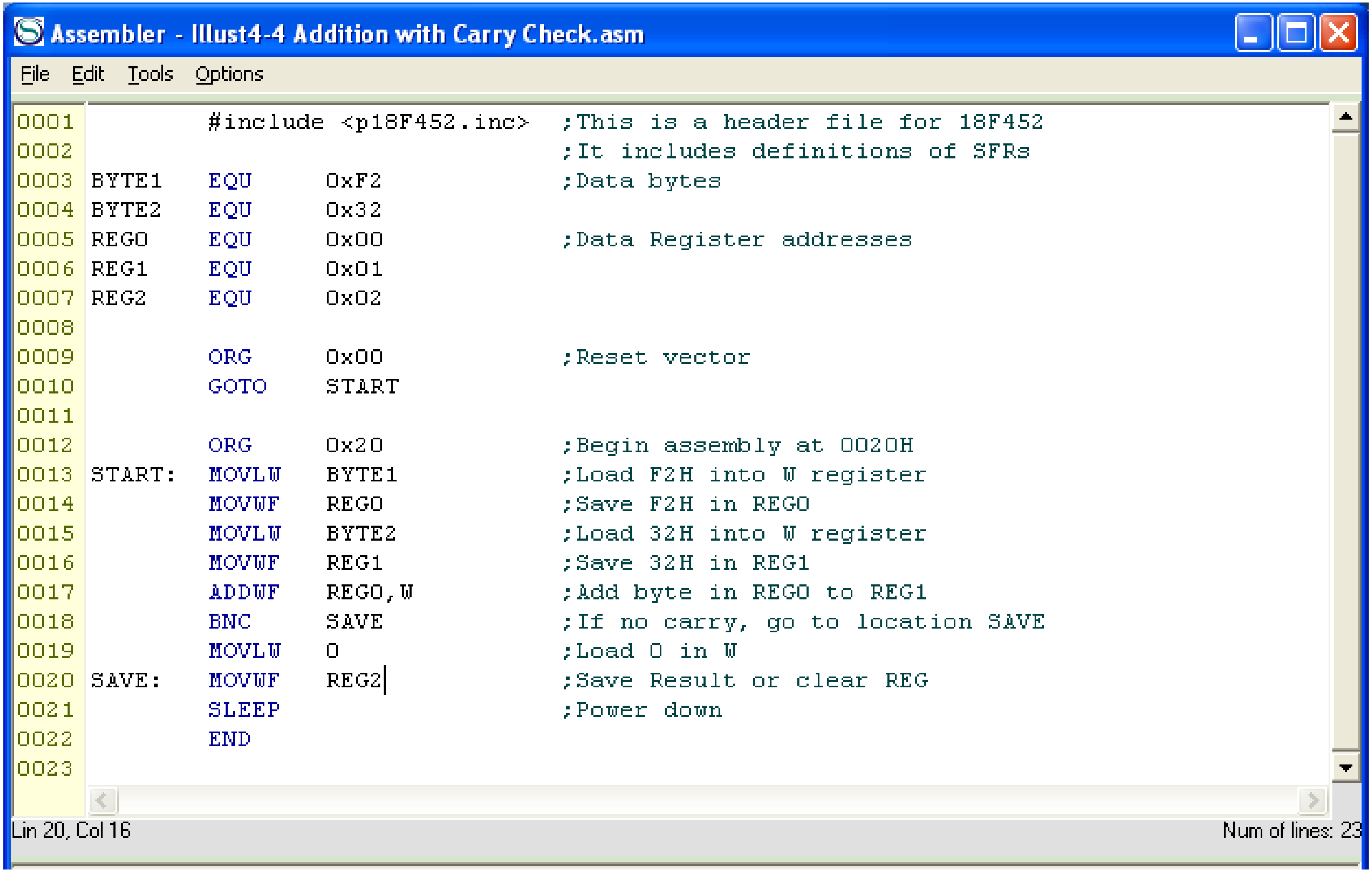
Radix Formats

Type	Format
Hexadecimal	0x4F H’4F’ 4F 4FH
Decimal	D’100’
Binary	B’10011000’
ASCII	‘Susan’ A’Susan’

PIC18 Simulator IDE

- Editor
  - Assembler
- Debugger
  - Breakpoint Manager
  - Watch Variables
- Simulator
  - Run and Step
- I/O Modules
  - LED Board
  - 7-Segment Display
  - Keypad Matrix

Assembler (Editor)

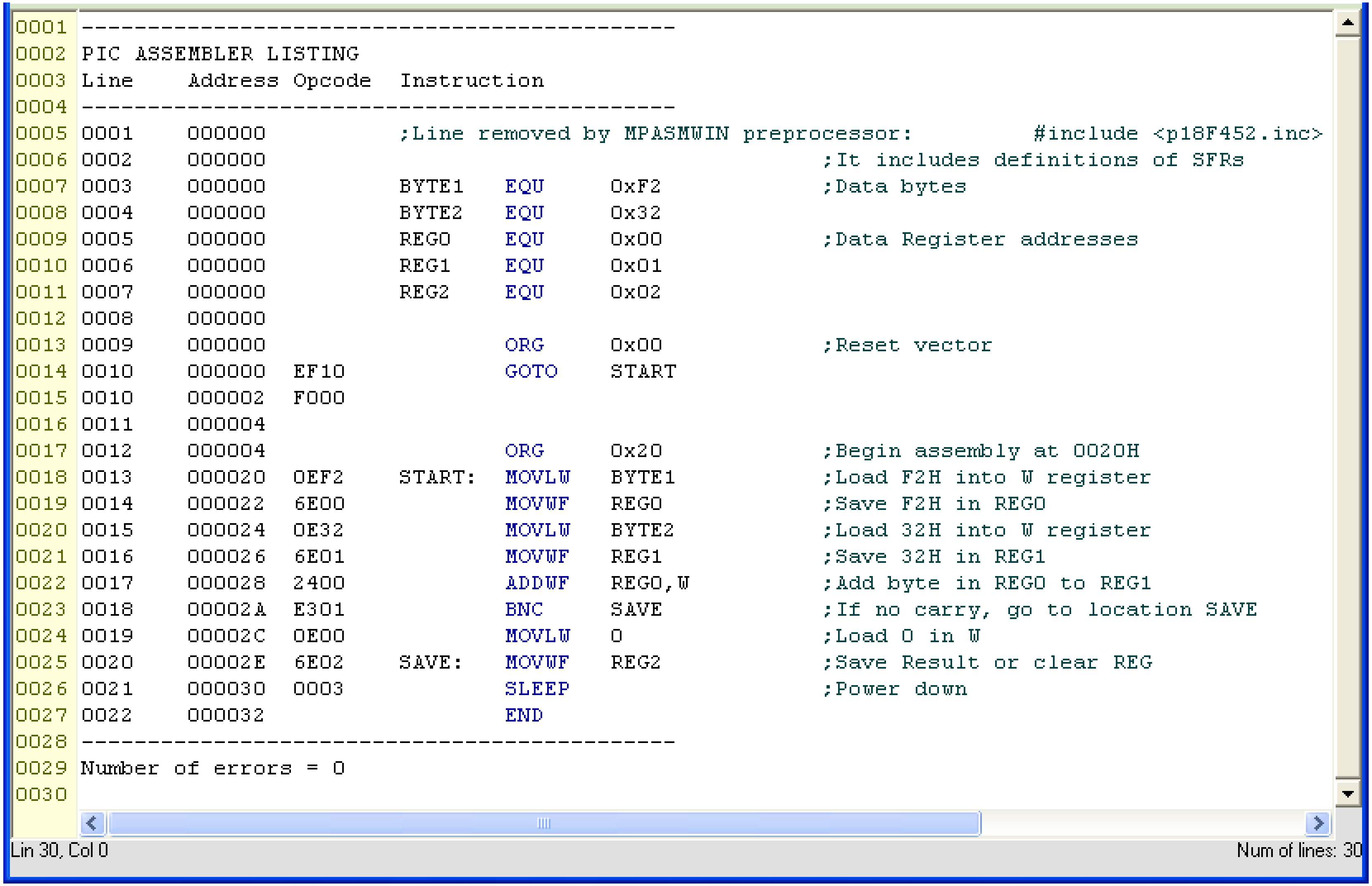


```
        #include <p18F452.inc>  ;This is a header file for 18F452
                                ;It includes definitions of SFRs
BYTE1   EQU     0xF2            ;Data bytes
BYTE2   EQU     0x32
REG0    EQU     0x00            ;Data Register addresses
REG1    EQU     0x01
REG2    EQU     0x02

        ORG     0x00            ;Reset vector
        GOTO    START

        ORG     0x20            ;Begin assembly at 0020H
START:  MOVLW   BYTE1           ;Load F2H into W register
        MOVWF   REG0            ;Save F2H in REG0
        MOVLW   BYTE2           ;Load 32H into W register
        MOVWF   REG1            ;Save 32H in REG1
        ADDWF   REG0,W          ;Add byte in REG0 to REG1
        BNC     SAVE            ;If no carry, go to location SAVE
        MOVLW   0               ;Load 0 in W
SAVE:   MOVWF   REG2            ;Save Result or clear REG
        SLEEP                   ;Power down
        END
```

Understanding the List File

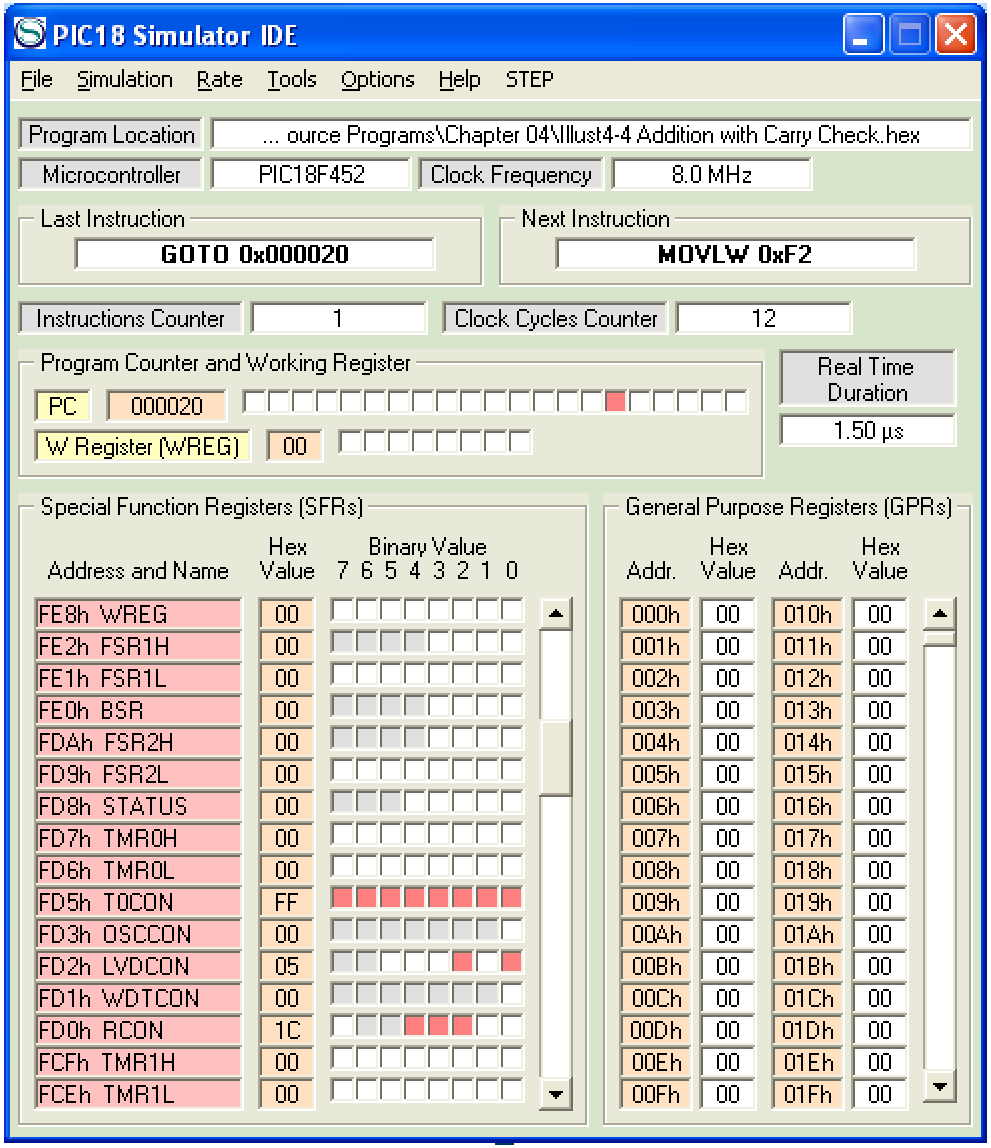
- List file generated primarily for documentation
- Includes seven columns
  - Line numbers
  - Memory addresses where binary code is stored
  - Hex code
  - Contents of source file
    - Labels
    - Opcode
    - Operands
    - Comments

Assembler Listing



```
---------------------------------------------
PIC ASSEMBLER LISTING
Line    Address Opcode  Instruction
---------------------------------------------
0001    000000          ;Line removed by MPASMWIN preprocessor:         #include <p18F452.inc>
0002    000000                                          ;It includes definitions of SFRs
0003    000000          BYTE1   EQU     0xF2            ;Data bytes
0004    000000          BYTE2   EQU     0x32
0005    000000          REG0    EQU     0x00            ;Data Register addresses
0006    000000          REG1    EQU     0x01
0007    000000          REG2    EQU     0x02
0008    000000
0009    000000                  ORG     0x00            ;Reset vector
0010    000000  EF10            GOTO    START
0010    000002  F000
0011    000004
0012    000004                  ORG     0x20            ;Begin assembly at 0020H
0013    000020  0EF2    START:  MOVLW   BYTE1           ;Load F2H into W register
0014    000022  6E00            MOVWF   REG0            ;Save F2H in REG0
0015    000024  0E32            MOVLW   BYTE2           ;Load 32H into W register
0016    000026  6E01            MOVWF   REG1            ;Save 32H in REG1
0017    000028  2400            ADDWF   REG0,W          ;Add byte in REG0 to REG1
0018    00002A  E301            BNC     SAVE            ;If no carry, go to location SAVE
0019    00002C  0E00            MOVLW   0               ;Load 0 in W
0020    00002E  6E02    SAVE:   MOVWF   REG2            ;Save Result or clear REG
0021    000030  0003            SLEEP                   ;Power down
0022    000032                  END
---------------------------------------------
Number of errors = 0
```

Executing a Program



Program Location	... ource Programs\Chapter 04\Illust4-4 Addition with Carry Check.hex
Microcontroller	PIC18F452
Clock Frequency	8.0 MHz

Last Instruction: **GOTO 0x000020**

Next Instruction: **MOVLW 0xF2**

Instructions Counter	1	Clock Cycles Counter	12

Program Counter and Working Register

PC	000020	□□□□□□□□□□□□□□□■□□□□□
W Register (WREG)	00	□□□□□□□□

Real Time Duration: 1.50 µs

Special Function Registers (SFRs)

Address and Name	Hex Value	Binary Value 7 6 5 4 3 2 1 0
FE8h WREG	00	□□□□□□□□
FE2h FSR1H	00	□□□□□□□□
FE1h FSR1L	00	□□□□□□□□
FE0h BSR	00	□□□□□□□□
FDAh FSR2H	00	□□□□□□□□
FD9h FSR2L	00	□□□□□□□□
FD8h STATUS	00	□□□□□□□□
FD7h TMR0H	00	□□□□□□□□
FD6h TMR0L	00	□□□□□□□□
FD5h T0CON	FF	■■■■■■■■
FD3h OSCCON	00	□□□□□□□□
FD2h LVDCON	05	□□□□□■□■
FD1h WDTCON	00	□□□□□□□□
FD0h RCON	1C	□□□■■■□□
FCFh TMR1H	00	□□□□□□□□
FCEh TMR1L	00	□□□□□□□□

General Purpose Registers (GPRs)

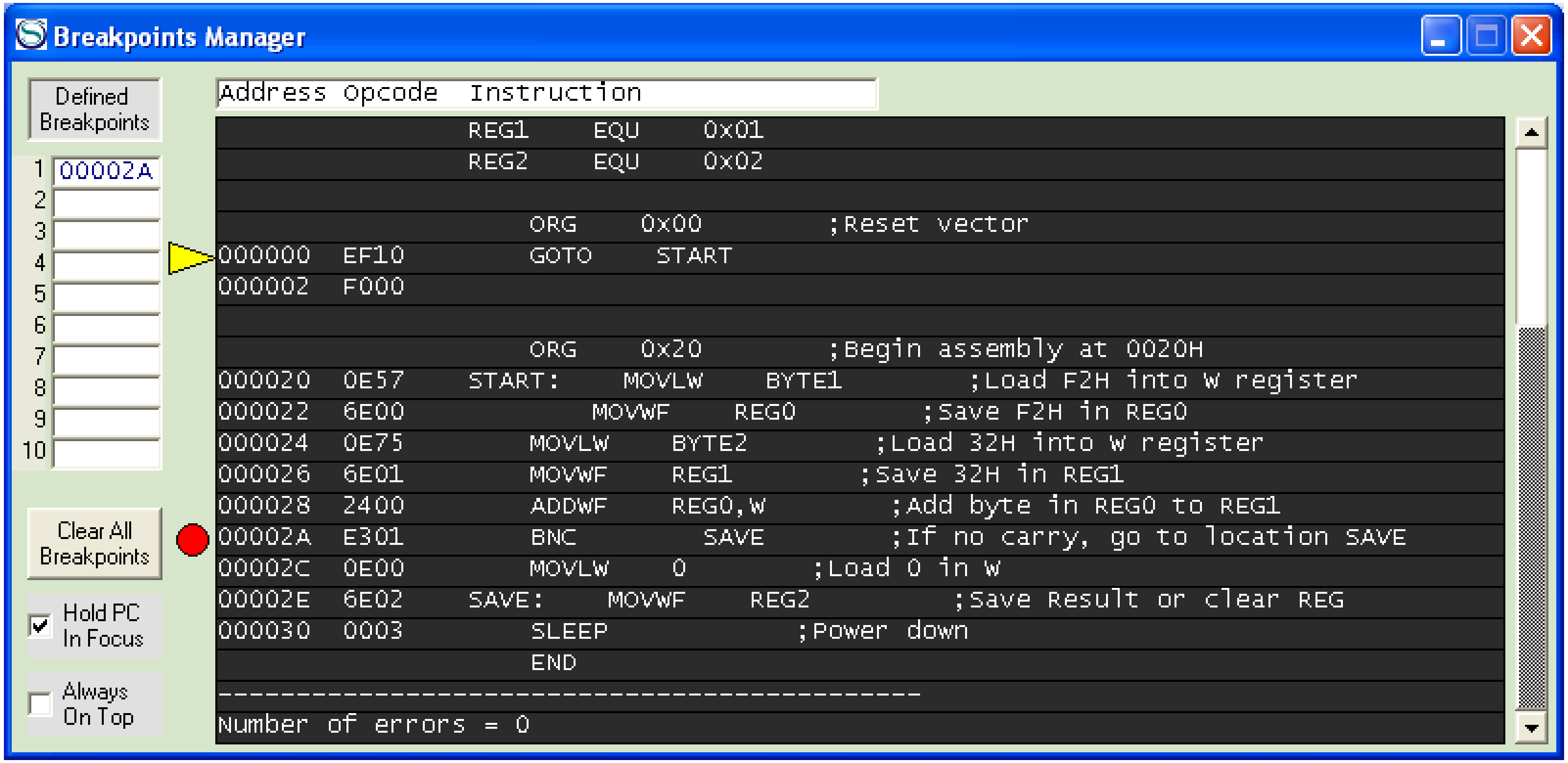
Addr.	Hex Value	Addr.	Hex Value
000h	00	010h	00
001h	00	011h	00
002h	00	012h	00
003h	00	013h	00
004h	00	014h	00
005h	00	015h	00
006h	00	016h	00
007h	00	017h	00
008h	00	018h	00
009h	00	019h	00
00Ah	00	01Ah	00
00Bh	00	01Bh	00
00Ch	00	01Ch	00
00Dh	00	01Dh	00
00Eh	00	01Eh	00
00Fh	00	01Fh	00

Debugging a Program

- Single-step technique
  - Execute one instruction at a time
  - Observe registers for expected results
- Breakpoint technique
  - Execute a group of instructions
  - Observe registers for expected results

Breakpoint Manager

- Simulation automatically switches to step-by-step mode when reaching a breakpoint



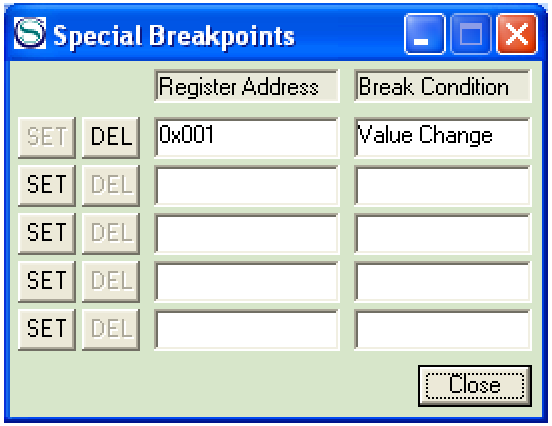
1 00002A
2
3
4
5
6
7
8
9
10

☑ Hold PC In Focus

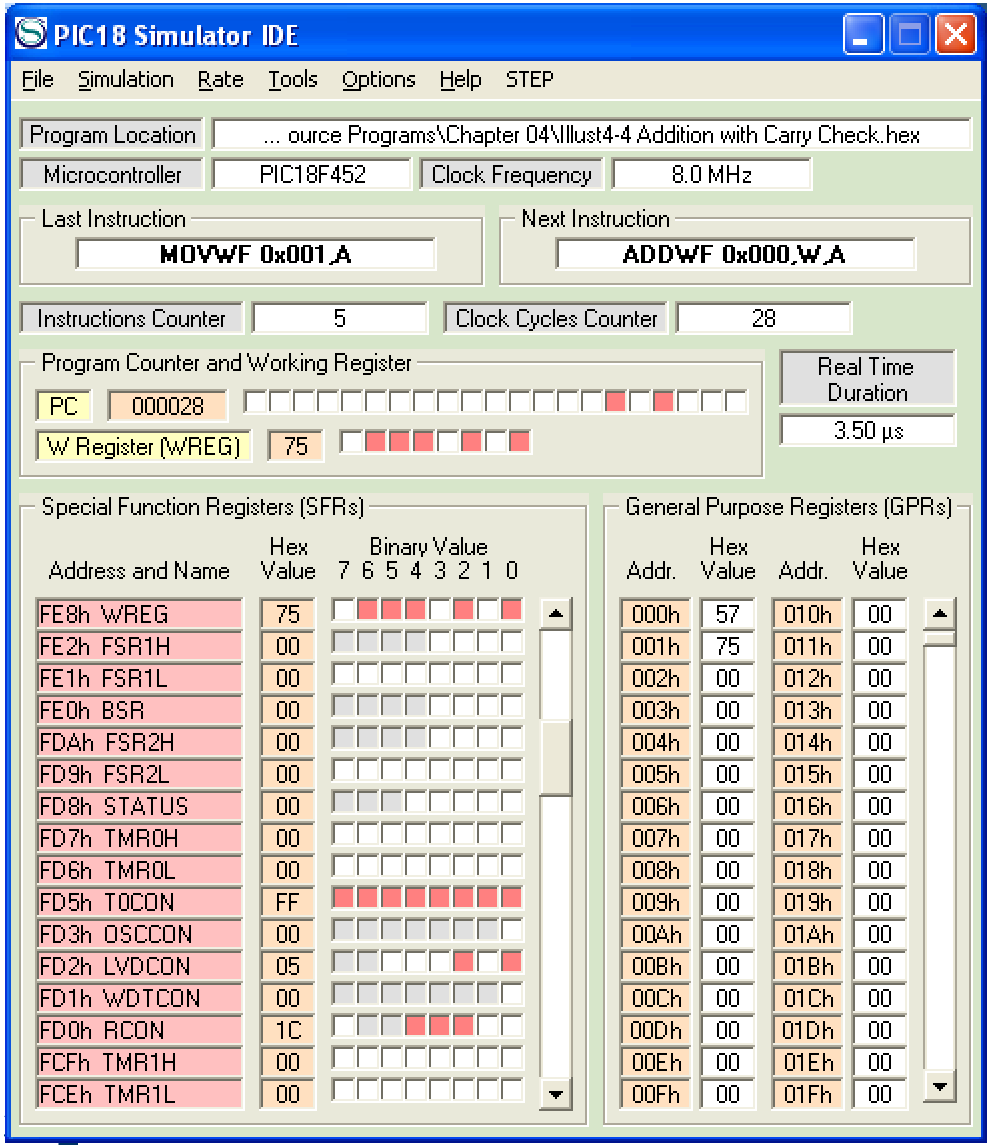
☐ Always On Top

```
Address Opcode  Instruction
        REG1    EQU    0x01
        REG2    EQU    0x02

          ORG    0x00         ;Reset vector
000000  EF10      GOTO    START
000002  F000

          ORG    0x20         ;Begin assembly at 0020H
000020  0E57    START:    MOVLW    BYTE1         ;Load F2H into W register
000022  6E00            MOVWF    REG0         ;Save F2H in REG0
000024  0E75      MOVLW    BYTE2         ;Load 32H into W register
000026  6E01      MOVWF    REG1         ;Save 32H in REG1
000028  2400      ADDWF    REG0,W         ;Add byte in REG0 to REG1
00002A  E301      BNC       SAVE         ;If no carry, go to location SAVE
00002C  0E00      MOVLW    0         ;Load 0 in W
00002E  6E02    SAVE:    MOVWF    REG2         ;Save Result or clear REG
000030  0003      SLEEP              ;Power down
                  END
---------------------------------------------
Number of errors = 0
```

Special Breakpoints



		Register Address	Break Condition
SET	DEL	0x001	Value Change
SET	DEL		
SET	DEL		
SET	DEL		
SET	DEL		



Program Location: ... ource Programs\Chapter 04\Illust4-4 Addition with Carry Check.hex

Microcontroller: PIC18F452

Clock Frequency: 8.0 MHz

Last Instruction: MOVWF 0x001,A

Next Instruction: ADDWF 0x000,W,A

Instructions Counter: 5

Clock Cycles Counter: 28

Program Counter and Working Register

PC: 000028 □□□□□□□□□□□□□□□🟥□🟥□□□

W Register (WREG): 75 □🟥🟥🟥□🟥□🟥

Real Time Duration: 3.50 µs

Special Function Registers (SFRs)

Address and Name	Hex Value	Binary Value 7 6 5 4 3 2 1 0
FE8h WREG	75	□🟥🟥🟥□🟥□🟥
FE2h FSR1H	00	□□□□□□□□
FE1h FSR1L	00	□□□□□□□□
FE0h BSR	00	□□□□□□□□
FDAh FSR2H	00	□□□□□□□□
FD9h FSR2L	00	□□□□□□□□
FD8h STATUS	00	□□□□□□□□
FD7h TMR0H	00	□□□□□□□□
FD6h TMR0L	00	□□□□□□□□
FD5h T0CON	FF	🟥🟥🟥🟥🟥🟥🟥🟥
FD3h OSCCON	00	□□□□□□□□
FD2h LVDCON	05	□□□□□🟥□🟥
FD1h WDTCON	00	□□□□□□□□
FD0h RCON	1C	□□□🟥🟥🟥□□
FCFh TMR1H	00	□□□□□□□□
FCEh TMR1L	00	□□□□□□□□

General Purpose Registers (GPRs)

Addr.	Hex Value	Addr.	Hex Value
000h	57	010h	00
001h	75	011h	00
002h	00	012h	00
003h	00	013h	00
004h	00	014h	00
005h	00	015h	00
006h	00	016h	00
007h	00	017h	00
008h	00	018h	00
009h	00	019h	00
00Ah	00	01Ah	00
00Bh	00	01Bh	00
00Ch	00	01Ch	00
00Dh	00	01Dh	00
00Eh	00	01Eh	00
00Fh	00	01Fh	00

- Switch to step-by-step when the value of a register has changed or equals a predefined value