

# Stacks and Subroutines

*ELEC 330*

*Digital Systems Engineering*

*Dr. Ron Hayne*

*Images Courtesy of Ramesh Gaonkar and Delmar Learning*

# Subroutine

- ◆ A group of instructions that performs a specified task
- ◆ Written independent of main program
- ◆ Can be called multiple times to perform task
  - By main program
  - By another subroutine

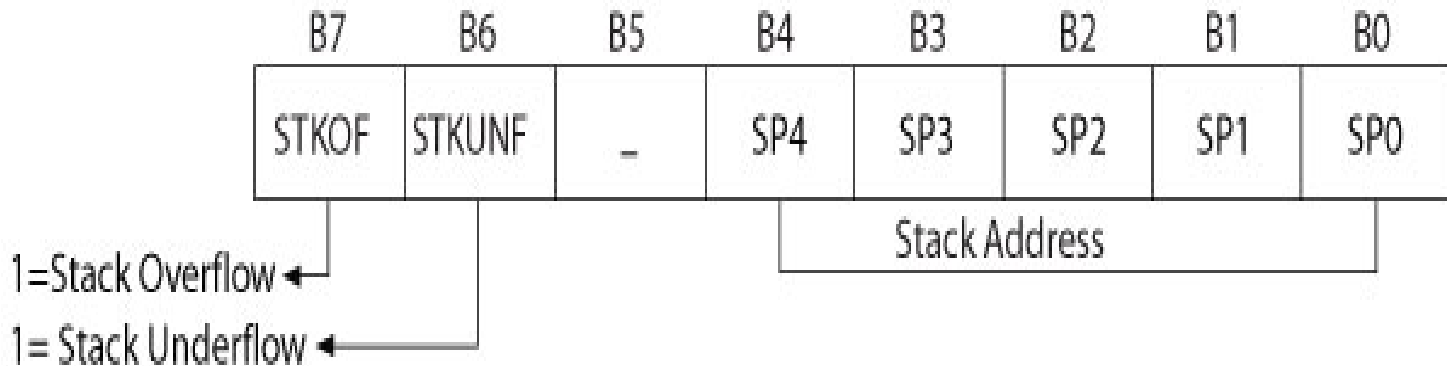
# Stack

- ◆ Temporary memory storage space used during the execution of a program
- ◆ Can be part of R/W memory or specially designed group of registers
- ◆ Stack Pointer (SP)
  - Keeps track of available stack locations

# PIC18 Microcontroller Stack

- ◆ Hardware Stack
  - 31 registers
  - 21-bits wide
  - Not part of program memory or data registers
- ◆ Stack Pointer (STKPTR)
  - 5-bit address
- ◆ Top of the Stack (TOS)
  - Pointed to by the stack pointer
  - Copied into three special function registers
  - TOSU (Upper), TOSH (High), and TOSL (Low)

# STKPTR Register



- ◆ SP4-SP0: Stack Address
- ◆ STKOF: Stack overflow
  - Attempt to use more than 31 registers to store information on the stack
- ◆ STKUNF: Stack underflow
  - Attempt to retrieve more information than was stored previously on the stack

# Stack Instructions

## ◆ PUSH

- Increment the memory address in the stack pointer and store the contents of the program counter (PC+2) on the top of the stack

## ◆ POP

- Discard the address of the top of the stack and decrement the stack pointer by one

# Subroutine

- ◆ A group of instructions that performs a specified task
- ◆ Written independent of a main program
- ◆ Can be called multiple times to perform task
  - By main program
  - By another subroutine
- ◆ Call and Return instructions
  - Call a subroutine
  - Return from the subroutine

# Call and Return Instructions

- ◆ CALL Label, s ;Call subroutine at Label
- ◆ CALL Label, FAST ;FAST equivalent to s = 1
  - If s = 0: Increment the stack pointer  
Store the return address (PC+4) on the TOS  
Go to the subroutine located at Label
  - If s = 1: Also copy W, STATUS, and BSR  
to their respective shadow registers
- ◆ RCALL, n ;Relative call to subroutine
  - Increment the stack pointer  
Store the return address (PC+2) on the TOS  
Branch to the location Label  $\pm$  512 words




# Call and Return Instructions

- ◆ RETURN, s ;Return from subroutine
- ◆ RETURN FAST :FAST equivalent to s = 1
  - If s = 0: Get the return address from the TOS (to PC)  
Decrement the stack pointer
  - If s = 1: Also retrieve W, STATUS, and BSR  
from their shadow registers
- ◆ RETLW 8-bit ;Return literal to WREG
  - Get the return address from the TOS (to PC)  
Decrement the stack pointer  
Return 8-bit literal to WREG

# Example 7.2

## ◆ Program Listing with Memory Addresses

Main Program				Subroutine			
0020	0EFE	START:	MOVLW B'11111110'	DELAY50MC:			
0022	6E94		MOVWF TRISC	0040	0EA6	MOVLW	D'166'
0024	6E01		MOVWF REG1	0042	6E10	MOVWF	REG10
0026	C001 FF82	ONOFF:	MOVFF REG1,PORTC	0044	0610	DECF	REG10,1
002A	EC20 F000		CALL DELAY50MC	0046	E1FE	BNZ	LOOP1
002E	1E01		COMP REG1,1	0048	0012	RETURN	
0030	D7FA		BRA ONOFF				



# Parameter Passing

- ◆ Parameter passing
  - Information exchanged between a calling program and a subroutine
- ◆ Subroutine documentation should include
  - Function of the subroutine
  - Input parameters
  - Output parameters
  - Registers modified
  - List of subroutines called

# Illustrative Program

## ◆ Problem Statement

- Write a program to copy a given number of unsigned 8-bit numbers from program memory into specified data registers
- Add the numbers and display the 16-bit sum at PORTB and PORTC
- The program should be subdivided into general purpose subroutines