

LIST OF SUBROUTINES CALLED

If a subroutine is calling other subroutines, the user should be provided with a list. The user can check what parameters need to be passed to various subroutines and what registers are modified in the process.

RESTART, CONDITIONAL CALL, AND RETURN INSTRUCTIONS 9.3

In addition to the unconditional CALL and RET instructions, the 8085 instruction set includes eight Restart instructions and eight conditional Call and Return instructions.

9.3.1 Restart (RST) Instructions

RST instructions are 1-byte Call instructions that transfer the program execution to a specific location on page 00H. They are executed the same way as Call instructions. When an RST instruction is executed, the 8085 stores the contents of the program counter (the address of the next instruction) on the top of the stack and transfers the program to the Restart location. These instructions are generally used in conjunction with the interrupt process discussed in Chapter 12. These instructions are listed here to emphasize that they are Call instructions and not necessarily always associated with the interrupts. The list of eight RST instructions is as follows:

RST 0	Call 0000H	RST 4	Call 0020H
RST 1	Call 0008H	RST 5	Call 0028H
RST 2	Call 0010H	RST 6	Call 0030H
RST 3	Call 0018H	RST 7	Call 0038H

9.3.2 Conditional Call and Return Instructions

The conditional Call and Return instructions are based on four data conditions (flags): Carry, Zero, Sign, and Parity. The conditions are tested by checking the respective flags. In case of a conditional Call instruction, the program is transferred to the subroutine if the condition is met; otherwise, the main program is continued. In case of a conditional Return instruction, the sequence returns to the main program if the condition is met; otherwise, the sequence in the subroutine is continued. If the Call instruction in the main program is conditional, the Return instruction in the subroutine can be conditional or unconditional. The conditional Call and Return instructions are listed for reference.

CONDITIONAL CALL

CC	Call subroutine if Carry flag is set (CY = 1)
CNC	Call subroutine if Carry flag is reset (CY = 0)
CZ	Call subroutine if Zero flag is set (Z = 1)
CNZ	Call subroutine if Zero flag is reset (Z = 0)
CM	Call subroutine if Sign flag is set (S = 1, negative number)
CP	Call subroutine if Sign flag is reset (S = 0, positive number)

- CPE Call subroutine if Parity flag is set ($P = 1$, even parity)
 CPO Call subroutine if Parity flag is reset ($P = 0$, odd parity)

CONDITIONAL RETURN

- RC Return if Carry flag is set ($CY = 1$)
 RNC Return if Carry flag is reset ($CY = 0$)
 RZ Return if Zero flag is set ($Z = 1$)
 RNZ Return if Zero flag is reset ($Z = 0$)
 RM Return if Sign flag is set ($S = 1$, negative number)
 RP Return if Sign flag is reset ($S = 0$, positive number)
 RPE Return if Parity flag is set ($P = 1$, even parity)
 RPO Return if Parity flag is reset ($P = 0$, odd parity)

9.4

ADVANCED SUBROUTINE CONCEPTS

In Section 9.2, one type of subroutine (multiple-calling of a subroutine by a main program) was illustrated. Other types of subroutine techniques, such as nesting and multiple-ending, are briefly illustrated below.

9.4.1 Nesting

The programming technique of a subroutine calling another subroutine is called **nesting**. This process is limited only by the number of available stack locations. When a subroutine calls another subroutine, all return addresses are stored on the stack. Nesting is illustrated in Figure 9.15.

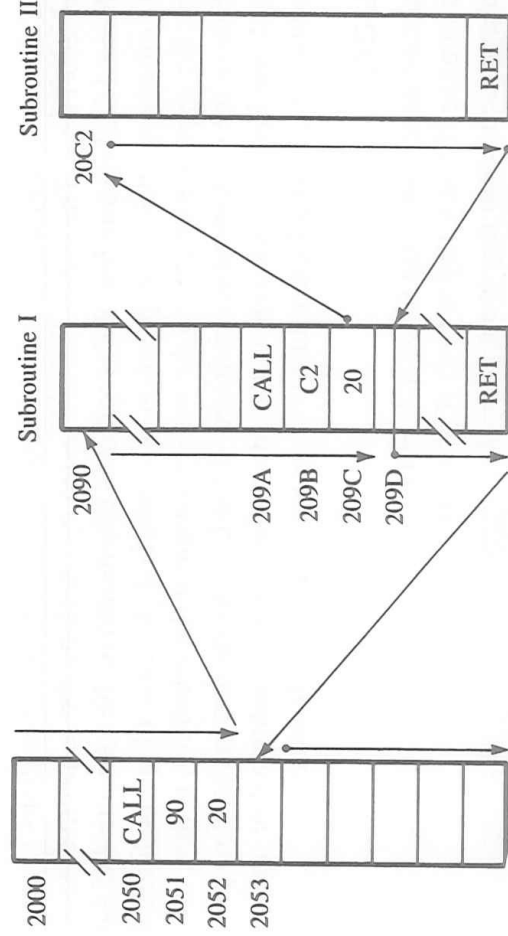


FIGURE 9.15
Nesting of Subroutines