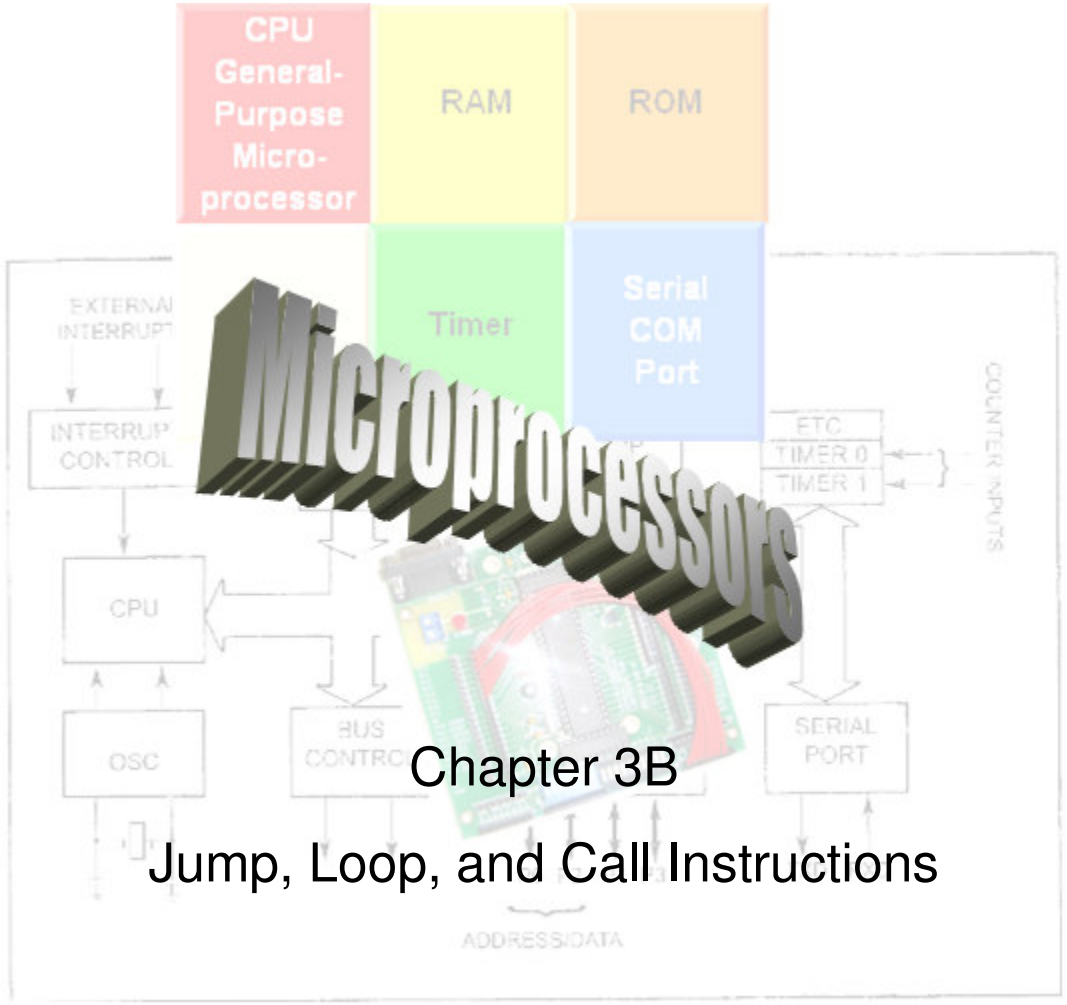

Chapter 3B

Jump, Loop, and Call Instructions

Jump, Loop, and Call Instructions

Chapter Objectives

- Code 8051 Assembly language instructions using loops
- Code 8051 Assembly language conditional jump instructions
- Explain conditions that determine each conditional jump instructions
- Code long & short instructions for unconditional long and short jumps
- Calculate target addresses for jump instructions
- Code 8051 subroutines

Jump, Loop, and Call Instructions

Chapter Objectives…

- Describe precautions in using the stack in subroutines
- Discuss crystal frequency versus machine cycle
- Code 8051 programs to generate a time delay

Jump, Loop, and Call Instructions

CALL Instructions

- Another control transfer instruction is the “CALL” instruction.
- Used to call a subroutine. **NOTE**: Subroutines are often used to perform tasks that need to be performed frequently.
- Subroutines make a program structured and saves memory space.
- 2 types of a CALL instructions:
  - a). LCALL
  - b). ACALL

Jump, Loop, and Call Instructions

LCALL (Long CALL)

- In this 3 byte instruction, the 1st byte is the op-code and the 2nd and 3rd bytes are used for the address of the target subroutine.
- LCALL, therefore can be used to call subroutines located anywhere within the 64K-byte address space of the 8051.
- To make sure that after the execution of the called subroutine the 8051 knows where to comeback to, the processor automatically saves on the stack the address of the instruction immediately below the LCALL.

Jump, Loop, and Call Instructions

LCALL (Long CALL). . .

- When a subroutine is called, control is transferred to that subroutine, and the processor saves the PC (program counter) on the stack and begins to fetch instructions form the new location..
- After finishing execution of the subroutine, the instruction RET (return) transfers control back to the caller.
- Every subroutine needs RET as the last instruction.

See Example 3-8, **page 76** in the textbook.

Jump, Loop, and Call Instructions

CALL Instruction and the role of the stack

Key points from Example 3-9.

•When the first LCALL is executed, the address of the instruction "*MOV A, #0AAH*" is saved on the stack.

•Notice the low byte goes 1st and high byte goes last.

•The last instruction of the called subroutine must be a RET instruction which directs the CPU to POP (place) the top bytes of the stack into the PC and resume executing at address 07.

Jump, Loop, and Call Instructions

Use of PUSH and POP Instructions in Subroutines

- The stack keeps track of where the CPU should return after completing the subroutine.
- Care must be used when manipulating the contents of the stack.
- The rule is that the number of PUSH and POP instructions must always match in any called subroutine.
- For every PUSH there must be a POP.

Jump, Loop, and Call Instructions

ACALL (Absolute Call)

- ACALL is a 2-byte instruction in contrast to LCALL which is 3bytes.
- ACALL is a 2-byte instruction, the target address of the subroutine must be within 2K bytes because only 11 bytes of the 2 bytes are used for the address.
- There is no difference between ACALL and LCAL in terms of saving the program counter on the stack or the function of the RET instruction.
- The only difference is that the target address for LCALL can be anywhere within the 64K-byte address space of the 8051 while the ACALL must be within a 2K-byte range.

Jump, Loop, and Call Instructions

ACALL (Absolute Call)…

- In many variations of the 8051 marketed by different companies, on-chip ROM is as low as 1K byte.
- In such cases, the use of ACALL instead of LCALL can save a number of bytes of program ROM space.

Jump, Loop, and Call Instructions

TIME DELAY FOR VARIOUS 8051 CHIPS:

Machine cycle for the 8051

- The CPU takes a certain number of clock cycles to execute an instruction.
- The 8051 family, these clocks cycles are referred to as machine cycles.
- See Table A-1(**pages 554-555** in textbook) for a list of 8051 instructions and their machine cycles.
- If machine cycle timing is critical to your system design, make sure that you check the manufacturer's data sheets for the device specifications.

Jump, Loop, and Call Instructions
Delay Calculations

In creating a time delay using Assembly Language instructions , 2 factors affect the accuracy of the delay.

a). The crystal frequency

b) The actual 8051 uC

Jump, Loop, and Call Instructions

Delay Calculations…

To determine the performance boost factor used in the Delay Timing Subroutine, with a target delay value (in us, ms, sec, etc.) the following equation may be used.

Equation 1:

Performance Boost factor (x) = Desired Time/0.02221

For additional examples of creating various time delays in 8051 Assembly Language, see pages **86-87** in the textbook.