

Call instructions

- Another control transfer instruction is the **CALL** instruction, which is used to call a subroutine.
- Subroutines are often used to perform tasks that need to be performed frequently.
- This makes a program more structured in addition to saving memory space.
- **ACALL-**
 - 2 byte instruction (target address of subroutines must be within 2K bytes because only 11 bits of 2 bytes are used for address)
- **LCALL-**
 - 3 byte instruction (first byte- opcode, second and third byte-address of target subroutine)
 - **LCALL** can be used to call subroutines located anywhere within 64K-bytes address space of 8051
 - After finishing execution of subroutine, instruction **RET** (return) transfers control back to the caller.

Example 3-8

Write a program to toggle all the bits of port 1 by sending to it the values 55H and AAH continuously. Put a time delay in between each issuing of data to port 1. This program will be used to test the ports of the 8051 in the next chapter.

Solution:

```
ORG      0
BACK:    MOV    A,#55H      ;load A with 55H
         MOV    P1,A        ;send 55H to port 1
         LCALL  DELAY       ;time delay
         MOV    A,#0AAH     ;load A with AA (in hex)
         MOV    P1,A        ;send AAH to port 1
         LCALL  DELAY
         SJMP   BACK        ;keep doing this indefinitely
;----- this is the delay subroutine
ORG      300H              ;put time delay at address 300H
DELAY:   MOV    R5,#0FFH    ;R5 = 255 (FF in hex), the counter
AGAIN:   DJNZ   R5,AGAIN    ;stay here until R5 becomes 0
         RET     ;return to caller (when R5 = 0)
         END      ;end of asm file
```

Note:

1. Upon executing "LCALL DELAY", address of instruction right below it, "MOV A, #0AAH", is pushed onto the stack and 8051 starts to execute instructions at address 300H

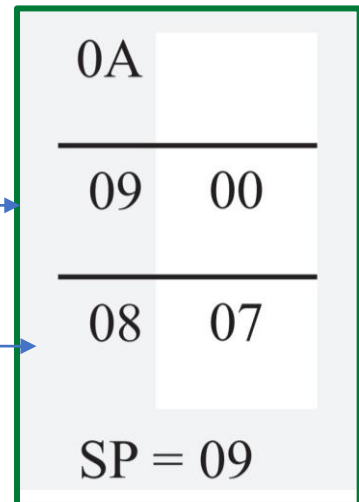
2. In DELAY subroutine, first R5= FFH; loop is repeated 256 times. When R5=0, control falls to RET instruction, which pops address from stack into PC and resumes executing the instruction after the CALL

Example 3-9

Analyze the stack contents after the execution of the first LCALL in the following.

Solution:

```
001      0000          ORG      0
002      0000      7455  BACK:   MOV      A,#55H          ;load A with 55H
003      0002      F590          MOV      P1,A            ;send 55H to port 1
004      0004      120300        LCALL  DELAY            ;time delay
005      0007      74AA          MOV      A,#0AAH         ;load A with AAH
006      0009      F590          MOV      P1,A            ;send AAH to port 1
007      000B      120300        LCALL  DELAY
008      000E      80F0          SJMP     BACK            ;keep doing this
009      0010
010      0010      ;—————this is the delay subroutine
011      0300          ORG      300H
012      0300          DELAY:
013      0300      7DFF          MOV      R5,#0FFH        ;R5=255
014      0302      DDFE  AGAIN:  DJNZ     R5,AGAIN        ;stay here
015      0304      22          RET                        ;return to caller
016      0305          END                                ;end of asm file
```



Note: RET: Directs CPU to POP the top bytes of the stack onto the PC and resume executing at address.

Example 3-10

Analyze the stack for the first LCALL instruction in the following program.

```
01 0000                                ORG      0
02 0000 7455 BACK:                     MOV      A,#55H          ;load A with 55H
03 0002 F590                           MOV      P1,A           ;send 55H to port 1
04 0004 7C99                           MOV      R4,#99H
05 0006 7D67                           MOV      R5,#67H
06 0008 120300                         LCALL    DELAY         ;time delay
07 000B 74AA                           MOV      A,#0AAH        ;load A with AA
08 000D F590                           MOV      P1,A           ;send AAH to port 1
09 000F 120300                         LCALL    DELAY
10 0012 80EC                           SJMP     BACK          ;keep doing this
11 0014                               ;-----this is the delay subroutine
12 0300                                ORG      300H
13 0300 C004 DELAY:                     PUSH     4             ;PUSH R4
14 0302 C005                           PUSH     5             ;PUSH R5
15 0304 7CFF                           MOV      R4,#0FFH       ;R4=FFH
16 0306 7DFF NEXT:                     MOV      R5,#0FFH       ;R5=255
17 0308 DDFF AGAIN:                     DJNZ     R5,AGAIN
18 030A DCFA                           DJNZ     R4,NEXT
19 030C D005                           POP      5             ;POP into R5
20 030E D004                           POP      4             ;POP into R4
21 0310 22                             RET                  ;return to caller
22 0311                                END                      ;end of asm file
```

Example 3-10

Solution:

First notice that for the PUSH and POP instructions we must specify the direct address of the register being pushed or popped. Here is the stack frame.

After the first LCALL

0B		
0A		
09	00	PCH
08	0B	PCL

After PUSH 4

0B		
0A	99	R4
09	00	PCH
08	0B	PCL

After PUSH 5

0B	67	R5
0A	99	R4
09	00	PCH
08	0B	PCL

**Figure 3-1. 8051
Assembly Main
Program That Calls
Subroutines**

```
;MAIN program calling subroutines
                                ORG      0
MAIN:    LCALL    SUBR_1
                                LCALL    SUBR_2
                                LCALL    SUBR_3

HERE:    SJMP     HERE
;-----end of MAIN
;
SUBR_1:   ....
          ....
          RET
;-----end of subroutine 1
;
SUBR_2:   ....
          ....
          RET
;-----end of subroutine 2

SUBR_3:   ....
          ....
          RET
;-----end of subroutine 3
          END                    ;end of the asm file
```

Example 3-11

A developer is using the Atmel AT89C1051 microcontroller chip for a product. This chip has only 1K byte of on-chip flash ROM. Which instruction, LCALL or ACALL, is more useful in programming this chip?

Solution:

The ACALL instruction is more useful since it is a 2-byte instruction. It saves one byte each time the call instruction is used.

Example 3-12

Rewrite Example 3-8 as efficiently as you can.

Solution:

```
                ORG      0
                MOV      A,#55H      ;load A with 55H
BACK:           MOV      P1,A        ;issue value in reg A to port 1
                ACALL    DELAY       ;time delay
                CPL      A           ;complement reg A
                SJMP     BACK        ;keep doing this indefinitely
```

;—————this is the delay subroutine

DELAY:

```
                MOV      R5,#0FFH    ;R5=255 (FF in hex), the counter
AGAIN:          DJNZ     R5,AGAIN    ;stay here until R5 becomes 0
                RET                          ;return to caller
                END                      ;end of asm file
```

Note: Complement of 55H is AAH and complement of AAH is 55H.

Example 3-13

The following shows crystal frequency for three different 8051-based systems. Find the period of the machine cycle in each case.

- (a) 11.0592 MHz (b) 16 MHz (c) 20 MHz

Solution:

(a) $11.0592 \text{ MHz}/12 = 921.6 \text{ kHz}$; machine cycle is $1/921.6 \text{ kHz} = 1.085 \mu\text{s}$
(microsecond)

(b) $16 \text{ MHz}/12 = 1.333 \text{ MHz}$; machine cycle (MC) = $1/1.333 \text{ MHz} = 0.75 \mu\text{s}$

(c) $20 \text{ MHz}/12 = 1.66 \text{ MHz}$; MC = $1/1.66 \text{ MHz} = 0.60 \mu\text{s}$

Example 3-14

For an 8051 system of 11.0592 MHz, find how long it takes to execute each of the following instructions.

- (a) MOV R3,#55 (b) DEC R3 (c) DJNZ R2,target
(d) LJMP (e) SJMP (f) NOP (no operation) (g) MUL AB

Solution:

The machine cycle for a system of 11.0592 MHz is $1.085 \mu\text{s}$ as shown in Example 3-13. Table A-1 in Appendix A shows machine cycles for each of the above instructions. Therefore, we have:

Instruction	Machine cycles	Time to execute
(a) MOV R3,#55	1	$1 \times 1.085 \mu\text{s} = 1.085 \mu\text{s}$
(b) DEC R3	1	$1 \times 1.085 \mu\text{s} = 1.085 \mu\text{s}$
(c) DJNZ R2,target	2	$2 \times 1.085 \mu\text{s} = 2.17 \mu\text{s}$
(d) LJMP	2	$2 \times 1.085 \mu\text{s} = 2.17 \mu\text{s}$
(e) SJMP	2	$2 \times 1.085 \mu\text{s} = 2.17 \mu\text{s}$
(f) NOP	1	$1 \times 1.085 \mu\text{s} = 1.085 \mu\text{s}$
(g) MUL AB	4	$4 \times 1.085 \mu\text{s} = 4.34 \mu\text{s}$

Example 3-15

Find the size of the delay in the following program, if the crystal frequency is 11.0592 MHz.

```
                MOV      A,#55H  ;load A with 55H
AGAIN:  MOV      P1,A          ;issue value in reg A to port 1
                ACALL  DELAY      ;time delay
                CPL      A        ;complement reg A
                SJMP    AGAIN     ;keep doing this indefinitely
;----Time delay
DELAY:  MOV      R3,#200        ;load R3 with 200
HERE:   DJNZ     R3,HERE        ;stay here until R3 become 0
                RET              ;return to caller
```

Solution:

From Table A-1 in Appendix A, we have the following machine cycles for each instruction of the DELAY subroutine.

		Machine Cycle
DELAY:	MOV R3,#200	1
HERE:	DJNZ R3,HERE	2
	RET	2

Therefore, we have a time delay of $[(200 \times 2) + 1 + 2] \times 1.085 \mu\text{s} = 436.255 \mu\text{s}$.

Example 3-16

For an 8051 system of 11.0592 MHz, find the time delay for the following subroutine:

	Machine Cycle
DELAY: MOV R3,#250	1
HERE: NOP	1
NOP	1
NOP	1
NOP	1
DJNZ R3,HERE	2
RET	2

Example 3-16

Solution:

The time delay inside the HERE loop is $[250(1 + 1 + 1 + 1 + 2)] \times 1.085 \mu\text{s} = 1500 \times 1.085 \mu\text{s} = 1627.5 \mu\text{s}$. Adding the two instructions outside the loop we have $1627.5 \mu\text{s} + 3 \times 1.085 \mu\text{s} = 1630.755 \mu\text{s}$.

If machine cycle timing is critical to your system design, make sure that you check the manufacture's data sheets for the device specification. For example, the DS89C430 has 3 machine cycles instead of 2 machine cycles for the RET instruction.

Example 3-17

For a machine cycle of 1.085 ms, find the time delay in the following subroutine.

DELAY:		Machine Cycle
	MOV R2,#200	1
AGAIN:	MOV R3,#250	1
HERE: NOP		1
	NOP	1
	DJNZ R3,HERE	2
	DJNZ R2,AGAIN	2
	RET	2

Example 3-17

Solution:

For the HERE loop, we have $(4 \times 250) \times 1.085 \mu\text{s} = 1085 \mu\text{s}$. The AGAIN loop repeats the HERE loop 200 times; therefore, we have $200 \times 1085 \mu\text{s} = 217000$, if we do not include the overhead. However, the instructions “MOV R3,#250” and “DJNZ R2,AGAIN” at the beginning and end of the AGAIN loop add $(3 \times 200 \times 1.085 \mu\text{s}) = 651 \mu\text{s}$ to the time delay. As a result we have $217000 + 651 = 217651 \mu\text{s} = 217.651$ milliseconds for total time delay associated with the above DELAY subroutine. Notice that in the case of a nested loop, as in all other time delay loops, the time is approximate since we have ignored the first and last instructions in the subroutine.

Table 3-2: Clocks per Machine Cycle (MC) for Various 8051 Versions

Chip/Maker	Clocks per Machine Cycle
AT89C51 Atmel	12
P89C54X2 Philips	6
DS5000 Dallas Semi	4
DS89C430/40/50 Dallas Semi	1

Example 3-18

From Table 3-2, find the period of the machine cycle (MC) in each case if XTAL = 11.0592 MHz, and discuss the impact on performance.

(a) AT89C51 (b) P89C54X2 (c) DS5000 (d) DS89C4x0

Solution:

(a) $11.0592 \text{ MHz}/12 = 921.6 \text{ kHz}$; MC is $1/921.6 \text{ kHz} = 1.085 \text{ } \mu\text{s}$ (microsecond) = 1085 ns

(b) $11.0592 \text{ MHz}/6 = 1.8432 \text{ MHz}$; MC is $1/1.8432 \text{ MHz} = 0.5425 \text{ } \mu\text{s} = 542 \text{ ns}$

(c) $11.0592 \text{ MHz}/4 = 2.7648 \text{ MHz}$; MC is $1/2.7648 \text{ MHz} = 0.36 \text{ } \mu\text{s} = 360 \text{ ns}$

(d) $11.0592 \text{ MHz}/1 = 11.0592 \text{ MHz}$; MC is $1/11.0592 \text{ MHz} = 0.0904 \text{ } \mu\text{s} = 90 \text{ ns}$

Table 3-3: Comparison of 8051 and DS89C4x0 Machine Cycles

Instruction	8051	DS89C4x0
MOV R3,#value	1	2
DEC Rx	1	1
DJNZ	2	4
LJMP	2	3
SJMP	2	3
NOP	1	1
MUL AB	4	9

Example 3-19

For an AT8051 and DS89C430/40/50 system of 11.0592 MHz, find how long it takes to execute each of the following instructions.

- (a) MOV R3,#55 (b) DEC R3 (c) DJNZ R2,target (d) LJMP (e) SJMP
(f) NOP (no operation) (g) MUL AB

Solution:

Refer to example 3-18 and table 3-3 .

Instruction	AT8051	DS89C430/40/50
(a) MOV R3,#55	1x1085 ns = 1085 ns	2x90 ns = 180 ns
(b) DEC R3	1x1085 ns = 1085 ns	1x90 ns = 90 ns
(c) DJNZ R2,..	2x1085 ns = 2170 ns	4x90 ns = 360 ns
(d) LJMP	2x1085 ns = 2170 ns	3x90 ns = 270 ns
(e) SJMP	2x1085 ns = 2170 ns	3x90 ns = 270 ns
(f) NOP	1x1085 ns = 1085 ns	1x90 ns = 90 ns
(g) MUL AB	4x1085 ns = 4340 ns	9x90 ns = 810 ns

Example 3-20

Find the time delay for the loop section of the following subroutine if it is run on a DS89C430 chip, assuming a crystal frequency of 11.0592 MHz.

DS89C430 Machine Cycle

DELAY: MOV R3,#250

HERE: NOP	1
NOP	1
NOP	1
NOP	1
DJNZ R3,HERE	4
RET	

Solution:

The time delay inside the HERE loop is $[250(1 + 1 + 1 + 1 + 4)] \times 90 \text{ ns} = 2000 \times 90 \text{ ns} = 180 \mu\text{s}$. Comparing this with Example 3-16, we see DS89C4x0 is about 9 times faster. $(1627 \mu\text{s} / 180 \mu\text{s} = 9)$

Example 3-21

Write a program to toggle all the bits of P1 every 200 ms. Assume that the crystal frequency is 11.0592 MHz and that the system is using the AT89C51.

Solution:

;Tested for AT89C51 of 11.0592 MHz.

```
                MOV     A,#55H
AGAIN:          MOV     P1,A
                ACALL   DELAY
                CPL      A
                SJMP     AGAIN

;-----Time delay
DELAY:          MOV     R5,#2
HERE1:          MOV     R4,#180
HERE2:          MOV     R3,#255
HERE3:          DJNZ    R3,HERE3
                DJNZ    R4,HERE2
                DJNZ    R5,HERE1
                RET
```

$$2 \times 180 \times 255 \times 2 \text{ MC} \times 1.085 \mu\text{s} = 199,206 \mu\text{s}$$

Example 3-22

Write a program to toggle all the bits of P1 every 200 ms. Assume crystal frequency is 11.0592 MHz and the system is using DS89C430/40/50.

Solution:

```
;Tested for DS89C430 of 11.0592 MHz.
```

```
                MOV     A,#55H
AGAIN:  MOV     P1,A
                ACALL  DELAY_200m
                CPL     A
                SJMP   AGAIN
```

```
;----Time delay
```

```
DELAY_200m:
                MOV     R5,#9
HERE1:  MOV     R4,#242
HERE2:  MOV     R3,#255
HERE3:  DJNZ    R3,HERE3
                DJNZ    R4,HERE2
                DJNZ    R5,HERE1
                RET
```

Delay $9 \times 242 \times 255 \times 4 \text{ MC} \times 90 \text{ ns} = 199,940 \mu\text{s}$