Interfacing the Avago HDSP-2xxx LED Alphanumeric Displays with the Intel 8751H Microcontroller

Application Brief D-005

Introduction

The HDSP-21xx/-25xx series of products is ideal for applications where displaying eight or more characters of dot matrix information is required. These devices are 8-digit, 5 x 7 dot matrix, alphanumeric displays. The on-board CMOS IC has the ability to decode 128 ASCII characters (HDSP-211x/25xx) or 128 Katakana characters (HDSP-212x), which are permanently stored in ROM. Up to 16 user defined symbols may be stored in on-board ROM, allowing flexibility for displaying additional symbols and icons. Seven brightness levels provide versatility in adjusting the display intensity and power consumption. These products are designed for standard microprocessor interface techniques. The display and special features are accessed through a bidirectional 8-bit data bus.

This application brief is a description of interfacing a "smart" HDSP-2xxx display with an INTEL 8751H microcontroller. This brief is to be used as a supplement to the data sheet on *8 Character 5 mm and 7 mm Smart Alphanumeric Displays*. Figures 1 and 2 show the circuit diagram and assembly source code that were implemented and will be the subject of this discussion. An instruction set description for the 8751H shown in Figure 3 has been included to assist in the understanding of the software used in this application. The approach taken here will be to view the hardware configuration and then step through the software that operates the circuit. By taking this approach, many important considerations that need to be made when doing this type of application will be identified.

Hardware

8751H

The circuit shown in Figure 1 shows the microcontroller and the display interfaced through Ports 1, 2, and 3 (P1.X, P2.X, and P3.X) of the controller. Each port is bit addressable, thus allowing for the manipulation of an individual pin voltage level while allowing the other pins to remain at their same values. This is a convenient characteristic when it is desired to switch only one of the display pin's logic level and not have to worry about latching other lines incorrectly. This characteristic makes writing code much simpler compared to a controller that can only address an external device in a byte format. During the power up, the port pins will be in a random state until the oscillator has started and the internal reset algorithm has written ones to them. The oscillator start up time will depend on the oscillator frequency. A 10 MHz crystal has about a 1 ms start up time, whereas a 1 MHz crystal's is about 10 ms. A 6.0 MHz crystal oscillator along with two capacitors, C1 = C2 = 18 pF, were used here. An external oscillator input to XTAL2 with XTAL 1 and V_{SS} grounded can be used instead of the crystal configuration shown here. The power up reset values of R = 8.2 kΩ and C3 = 10 μF were chosen to ensure a valid reset which is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods) while the oscillator is running. A decoupling capacitor of 0.5 μF, not shown here, was used between the power supply and ground to eliminate any high frequency noise from interfering with the controller's internal circuitry.

Display

As shown in Figure 1, the display data lines D0-D7 are connected to port 3 of the microcontroller, the display flash and address lines A0-A4 to port 2, and the display reset, chip enable, read, and write lines as well as the pass-fail self test indicator circuit to port 1.

Clock select is tied to V_{DD} to select the display's on-board internal oscillator. Having chosen the internal oscillator option, the display clock pin outputs the internal oscillator signal. Thus, this display's clock can be used as a master for other displays in the same system with each slave display having its clock externally sourced by tying its clock select to logic ground.

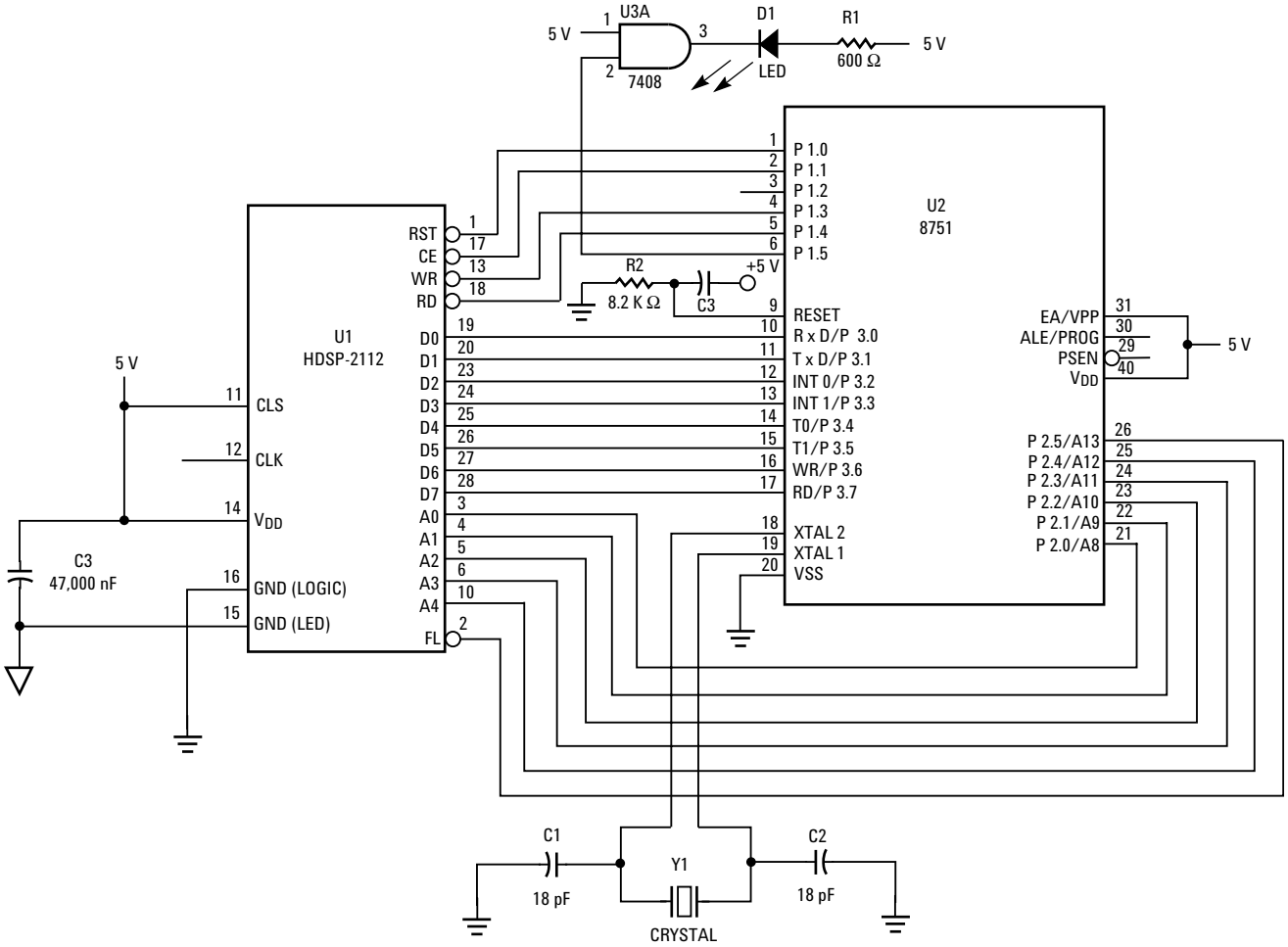

Figure 1. Circuit Diagram of HDSP-2112 Display interfaced with the Intel 8751H

When concerning the power supplied to the display and maintaining the voltage levels as specified by the data sheet, the LED supply ground and the logic ground pins should have separate traces returning to the power supply. This is done to isolate the logic from variations in the LED currents required to light the display. The V_{DD} power supply should be able to handle large current surges. The peak current surge value will be a function of the characters being displayed, the power supply value, and the junction temperature of the LEDs. Using a decoupling capacitor between the power supply and LED ground pin will help prevent any supply noise from interfering with the display's internal circuitry. The value of the capacitor depends on the series resistance from the ground back to the power supply and the range of frequencies that need to be suppressed. A 47 μF capacitor is recommended. It is also advantageous to use the largest ground plane possible. A large ground plane will reduce the ground path resistance (reluctance) and thus reduce any undesirable voltage drops (magnetomotive forces) which can act as noise sources along the ground to supply path. Here, V_{DD} of the display was connected to the same power supply and decoupling capacitor of 0.5 μF as the 8751H. Please refer to the data sheet for a description on thermal, electrical, and electrostatic discharge considerations concerning the display.

Software

The software shown in Figure 2 is a list of the source code (assembly) that was written to the 8751H. This code was the instruction set upon which the microcontroller would electrically operate the display.

The first command was to reset the display. The initial delay subroutines were called to allow the display to power up to the correct wake state. After resetting the display, another delay was called to allow for three display clock cycles to occur as required in the data sheet specification.

The Control Word was next addressed and written to. The self-test was enabled by writing a "1" to bit D6 of the Control Word utilizing subroutine WRITE. Then bit D5 was read in subroutine RESULT to see if the display passed the self-test. If the display passed, pin P1.5 of the microcontroller was switched to a logic low and caused the LED to light up.

```
;*****************************************************************************************
;FILENAME:  LEOPARD.ASM
;THIS PROGRAM INTERFACES THE INTEL 8751 8-BIT EMBEDDED CONTROLLER
;TO THE HDSP-21XX/25XX SMART ALPHANUMERIC DISPLAY
;
;PORT PIN ASSIGNMENTS
;
;P10        EQU     RESET
;P11        EQU     CHIP ENABLE
;P13        EQU     WRITE ENABLE
;P14        EQU     READ ENABLE
;P15        EQU     LED FOR DISPLAY PASSED TEST
;
;P20-P24    EQU     ADDRESS LINES AO-A4
;P25        EQU     FLASH ENABLE
;
;P30-P37    EQU     DATA LINES DO-D7
;
;NOTE:  THE OUTPUT PORTS AFTER CONTROLLER POWER UP ARE #FFH
;*****************************************************************************************
          .opd+f    EQU, .c+qu
          .opd+f    DB, .db
          .opd+f    CALL, 1call
          .opd+f    ORG, .org
          .opd+f    END, .+nd

;*****************************************************************************************
;MAIN PROGRAM
;SENDING A SET OF CHARACTERS TO THE LEOPARD
;*****************************************************************************************
          ORG       0000H

          CALL      DELAY
          CALL      DELAY
          CLR       P10             ;RESET THE DISPLAY
          SETB      P10
          CALL      DELAY           ;ALLOW FOR 3 DISPLAY CLK CYCLES
          MOV       P2, #30H        ;ADDRESS CONTROL WORD
          MOV       P3, #40H        ;SELF TEST, 100% BRIGHTNESS
          CALL      WRITE           ;TEST DISPLAY
          CALL      DELAY
          CALL      DELAY           ;WAIT FOR DISPLAY TO END TEST
          CALL      DELAY
          CALL      DELAY
          CALL      DELAY
          CALL      DELAY
          CALL      RESULT          ;DISPLAY RESULTS
          CALL      DELAY

          MOV       P3, #03H        ;SET DISPLAY TO 40% BRIGHTNESS
          CALL      WRITE           ;WRITE TO DISPLAY
          CALL      UDCHAR          ;CREATE A USER CHARACTER
AGAIN:    CALL      CHAR            ;DISPLAY USER AND ASCII CHARACTERS
          CALL      FLASH           ;FLASH AN INDIVIDUAL CHARACTER
          CALL      BLINK           ;BLINK THE WHOLE DISPLAY
          CALL      CLEAR           ;CLEAR THE DISPLAY
          AJMP      AGAIN
;*****************************************************************************************
;SUBROUTINE DELAY
;*****************************************************************************************
DELAY:    MOV       R3, #OFFH       ;255 TIMES
TIME:     MOV       R1, #OFFH       ;255 = 510 CYCLES
STALL1:   DJNZ      R1, STALL1
          MOV       R1, #OFFH       ;255 NEW
STALL2:   DJNZ      R1, STALL2
          MOV       R1, #OFFH       ;255
STALL3:   DJNZ      R1, STALL3
          DJNZ      R3, TIME
          RET                       ; RETURN TO MAIN PROGRAM
```

Figure 2. Assembly Source Code Used to Program the 8751H

The display brightness was set to 40% by writing the correct brightness octal 011 to Control Word bits D0-D2.

A user defined character utilizing subroutine UDCHAR was written to the display in eight write cycles. The first cycle is used to write to the UDC Address Register to assign the address space in the UDC RAM that will be written to during the next character write to the UDC RAM. The next seven cycles are used to define the character by addressing one row at a time and encoding which pixels will be turned on or off.

The Character RAM is then addressed and used to write the user defined and desired ASCII characters to the LED driver circuits. Note that the logic level of data bit D7 determines whether the Character RAM is addressing the UDC or ASCII RAM. This process is defined in subroutine CHAR.

The flashing, blinking, and clearing capabilities are implemented through subroutines FLASH, BLINK, and CLEAR. The delays are used to modify the amount of time each feature is being used.

```
;*****************************************************************************************
;SUBROUTINE WRITE
;WRITES IN THE ADDRESS AND DATA LINE INFO TO THE DISPLAY
;*****************************************************************************************
;
WRITE:    CLR     P11       ;SET CE1 LOW
          CLR     P13       ;SET WRITE LOW
          SETB    P13       ;SET WRITE HIGH
          SETB    P11       ;SET CE1 HIGH

          RET

;*****************************************************************************************
;SUBROUTINE RESULT
;CHECKS BIT D5 OF BOTH DISPLAYS CONTROL WORD FOR SELF TEST RESULT
;*****************************************************************************************
;
RESULT:   MOV     P3, #OFFH         ;SET UP 8751H TO READ DATA LINE BY
                                    ;WRITING ALL "1"s TO THE PORT
          CLR     P11               ;SET CE1 LOW
          CLR     P14               ;SET READ LOW
          JB      P35, PASS1        ;CHECK SELF TEST PASS/FAIL
          RET
PASS1:    CLR     P15               ;LED INDICATES # 1 PASS
          SETB    P14               ;SET READ HIGH
          SETB    P11               ;SET CE1 HIGH

          RET


;*****************************************************************************************
;SUBROUTINE UDCHAR
;CALLS UPON CHARACTERS WITHIN THE CHARACTER TABLE
;*****************************************************************************************
;
UDCHAR:   MOV     P2, #20H          ;ADDRESS UDC REGISTER
          MOV     P3, #00H          ;USER CHARACTER # 0
          CALL    WRITE             ;DEFINE UDC POSITION
          MOV     P2, #28H          ;UDC RAM ROW 1
          MOV     P3, #00H
          CALL    WRITE;                                  THIS IS THE
          MOV     P2, #29H          ;UDC RAM ROW 2        UDC CHARACTER
          MOV     P3, #0AH          ;                     THAT WILL APPEAR
          CALL    WRITE
          MOV     P2, #2AH          ;UDC RAM ROW 3
          MOV     P3, #00H          ;
          CALL    WRITE             ;
          MOV     P2, #2BH          ;UDC RAM ROW 4
          MOV     P3, #04H
          CALL    WRITE;                                    *   *
          MOV     P2, #2CH          ;UDC RAM ROW 5
          MOV     P3, #11H          ;                       *
          CALL    WRITE;                                *       *
          MOV     P2, #2DH          ;UDC RAM ROW 6          *   *   *
          MOV     P3, #0EH
          CALL    WRITE
          MOV     P2, #2EH          ;UDC RAM ROW 7
          MOV     P3, #00H
          CALL    WRITE

          RET
```

Figure 2. Assembly Source Code Used to Program the 8751H (continued)

```
;*****************************************************************************************
;SUBROUTINE CHAR
;DISPLAY THE DEFINED AND ASCII CHARACTERS TO THE DISPLAY
;*****************************************************************************************
;
CHAR:       MOV     P3, #80H              ;DATA LINES POINTING TO UDC RAM
            MOV     P2, #38H              ;ADDRESS CHARACTER RAM
SCROLL:     CALL    WRITE                 ;LATCH DISPLAY LED DRIVERS
            CALL    DELAY
            CALL    DELAY
            MOV     R2, P2
            INC     P2
            CJNE    R2, #40H, SCROLL      ;SCROLL SMILE FACE LEFT TO RIGHT
            MOV     P3, #48H              ;H
            MOV     P2, #38H
            CALL    WRITE
            MOV     P3, #45H              ;E
            INC     P2
            CALL    WRITE
            MOV     P3, #57H              ;W
            INC     P2
            CALL    WRITE
            MOV     P3, #4CH              ;L
            INC     P2
            CALL    WRITE
            MOV     P3, #45H              ;E
            INC     P2
            CALL    WRITE
            MOV     P3, #54H              ;T
            INC     P2
            CALL    WRITE
            MOV     P3, #54H              ;T
            INC     P2
            CALL    WRITE
            CALL    DELAY
            CALL    DELAY
            CALL    DELAY

            MOV     P3, #50H              ;P
            MOV     P2, #38H
            CALL    WRITE
            MOV     P3, #41H              ;A
            INC     P2
            CALL    WRITE
            MOV     P3, #43H              ;C
            INC     P2
            CALL    WRITE
            MOV     P3, #4BH              ;K
            INC     P2
            CALL    WRITE
            MOV     P3, #41H              ;A
            INC     P2
            CALL    WRITE
            MOV     P3, #52H              ;R
            INC     P2
            CALL    WRITE
            MOV     P3, #44H              ;D
            INC     P2
            CALL    WRITE

            RET
```

Figure 2. Assembly Source Code Used to Program the 8751H (continued)

```
;*****************************************************************************
;SUBROUTINE FLASH
;IMPLEMENT INDIVIDUAL CHARACTER FLASH
;*****************************************************************************

FLASH:  MOV     P2, #04H        ;ADDRESS FLASH RAM, CHARACTER #4
        MOV     P3, #0IH        ;STORE # 4 FLASH IN FLASH RAM
        CALL    WRITE
        MOV     P2, #30H        ;ADDRESS CONTROL WORD
        MOV     P3, #0BH        ;ENABLE FLASH, 40% BRIGHTNESS
        CALL    WRITE
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        MOV     P3, #03H        ;DISENABLE FLASH, 40% BRIGHTNESS
        CALL    WRITE
        MOV     P2, #04H        ;ADDRESS FLASH RAM, CHARACTER #4
        MOV     P3, #00H        ;REMOVE # 4 FLASH IN FLASH RAM
        CALL    WRITE

        RET


;*****************************************************************************
;SUBROUTINE BLINK
;IMPLEMENT WHOLE DISPLAY BLINKING
;*****************************************************************************

BLINK:  MOV     P2, #30H        ;ADDRESS CONTROL WORD
        MOV     P3, #13H        ;ENABLE BLINKING, 40% BRIGHTNESS
        CALL    WRITE
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        MOV     P3, #03H        ;DISENABLE BLINKING, 40% BRIGHTNESS
        CALL    WRITE

        RET


;*****************************************************************************
;SUBROUTINE CLEAR
;CLEARS THE DISPLAYS
;*****************************************************************************

CLEAR:  MOV     P3, #80H        ;CLEAR THE DISPLAY
        CALL    WRITE

        RET
```

Figure 2. Assembly Source Code Used to Program the 8751H (continued)

Arithmetic Operations

	Mnemonic	Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w Borrow	1	1
SUBB	A,#data	Subtract immed. data from A w Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

Figure 3. MCS-51™ Instruction Set Description for 8751H

Logical Operations

	Mnemonic	Description	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive - OR register to Accumulator	1	1
XRL	A,direct	Exclusive - OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive - OR indirect RAM to Accumulator	1	1
XRL	A,#data	Exclusive - OR immediate data to Accumulator	2	1
XRL	direct,A	Exclusive - OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive - OR immediate data to direct byte	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Figure 3. MCS-51™ Instruction Set Description for 8751H (continued)

Data Transfer

	Mnemonic	Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM.	2	1
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order Digit ind. RAM w A	1	1

Figure 3. MCS-51™ Instruction Set Description for 8751H (continued)

Boolean Variable Manipulation

	Mnemonic	Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

Figure 3. MCS-51™ Instruction Set Description for 8751H (continued)

Program and Machine Control

	Mnemonic	Description	Byte	Cyc
ACALL	addr11	Absolute Subroutine Call	2	2
ICALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETL		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg. & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	Working register R0-R7
direct	128 internal RAM locations, any I/O port, control, or status register
@Ri	Indirect internal RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data16	16-bit constant included as bytes 2 & 3 of instruction
bit	128 software flags, any I/O pin, control, or status bit

Notes on program addressing modes:

addr16	Destination address for LCALL & LJMP may be anywhere within the 64-Kilobyte program memory address space.
addr11	Destination address for ACALL & AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.
rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 to -128 bytes relative to first byte of the following instruction

Figure 3. MCS-51™ Instruction Set Description for 8751H (continued)