

# Processor Hardware/Software Interface

## EECS 113

### Assignment 2: ASCII-to-Decimal Conversion and Multiplication

Edward Ou  
15561434

Test Case 1:

N1 = 23 (40H=0x32H, 41H=0x33H),

N2 = 58 (43H=0x35H, 44H=0x38H)

Code memory:

The screenshot displays the Proteus IDE interface for an 8051 microcontroller project. The left pane shows the 8051 configuration window with various registers and pins set to default values. The right pane shows the assembly code for the program.

**8051 Configuration:**

- System Clock (MHz): 12.0
- SBUF: 0x00
- R7: 0x00, R6: 0x00, R5: 0x00, R4: 0x00, R3: 0x00, R2: 0x00, R1: 0x00, R0: 0x00
- B: 0x00, ACC: 0x00, PSW: 0x00, IP: 0x00, IE: 0x00, PCON: 0x00, DPH: 0x00, DPL: 0x00, SP: 0x07
- TH0: 0x00, TL0: 0x00, TMOD: 0x00, TCON: 0x00
- TH1: 0x00, TL1: 0x00
- PC: 0x0000
- PSW: 00000000

**Code Memory:**

addr	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E	0x000F
00	75	50	64	75	51	0A	75	52	01	78	52	79	5F	7B	02	90
10	00	40	09	0A	05	82	E4	93	B4	00	F8	15	82	E4	93	94
20	30	86	F0	A4	27	F7	15	82	18	DA	F2	90	00	43	78	52
30	DB	E0	E7	19	87	F0	A4	85	F0	40	F5	41	80	FE	00	00
40	32	33	00	35	38	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

**Assembly Code:**

```
ORG 40H
N1: DB "23" ;40H=0x32H, 41H=0x33H
DB 0
N2: DB "58" ;43H=0x35H, 44H=0x38H
DB 0
; N1: DB "143" ;40H=0x31H, 41H=0x32H, 42H=0x33H
; DB 0
; N2: DB "234" ;44H=0x32H, 45H=0x33H, 46H=0x34H
; DB 0

; This is the beginning of the program
ORG 0H
0000| MOV 50H, #100 ;Factor for highest digit
0003| MOV 51H, #10 ;Factor for the middle digit
0006| MOV 52H, #1 ;Factor for lowest digit
0009| MOV R0, #52H ;Use R0 to keep track of current digit
000B| MOV R1, #5FH ;Use R1 to keep track of current digit
000D| MOV R3, #02H ;Use R3 to determine the number of digits

; This section converts string to decimal
000F| MOV DPTR, #N1 ; Load data address
0012| ST: INC R1 ;
; This section finds the # of digits
0013| L1: INC R2 ; Use R2 to keep track of number of digits
```

The result of converting from ASCII characters to decimal numbers is stored at 60H for N1 and 61H for N2. After the conversion, N1 in hex is 17, N2 is 3A. They match with the screenshot of the data memory. The result of multiplication is 0536H, with 05 stored in 40H and 36 stored in 41H. This also matches with the screenshot of the data memory.

Data Memory

System Clock (MHz)  1 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0x05
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x36
RXD	TXD			R5	0x00	PSW	0x04
1	1	TMOD	0x00	R4	0x00	IP	0x00
SCON	0x00	TCON	0x00	R3	0x00	IE	0x00
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x60	DPH	0x00
0xFF	0xFF	P3	0x00	R0	0x52	DPL	0x43
0xFF	0xFF	P2				SP	0x07
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 0x003C PSW 00000100

8051

Modify RAM

addr 0x00 0x52 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	52	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	05	36	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	64	0A	01	00	00	00	00	00	00	00	00	00	00	00	00	00
60	17	3A	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005–2021 James Rogers Remove All Breakpoints

RST Step Run New Load Save Copy Paste

Time: 140us - Instructions: 93

```

001D| RPT1: CLR A ; Clear the accumulator
001E| MOV A, @A+DPTR ; Move the digit
001F| SUBB A, #30H
0021| MOV B, @R0 ; Move the factor into
0023| MUL AB ; Multiply the digit by the
0024| ADD A, @R1 ; Sum the current digit
0025| MOV @R1, A ; Save the result in
0026| DEC DPL ; Move DPTR to the next
0028| DEC R0 ; Move R0 to the next factor
0029| DJNZ R2, RPT1 ; Repeat until all

; This section converts string
002B| MOV DPTR, #N2
002E| MOV R0, #52H
0030| DJNZ R3, ST ; When both N1 and N2

0032| Cal: MOV A, @R1 ; Copy the first
0033| DEC R1 ; Move R1 to the second number
0034| MOV B, @R1 ; Copy the second number
0036| MUL AB ; Multiply AB to get the
0037| MOV 40H, B ; Store the upper bits
003A| MOV 41H, A ; Store the lower bits

003C| HERE: SJMP HERE

```

Test Case 2:

N1 = 143 (40H=0x31H, 41H=0x34H, 42H=0x33H),

N2 = 234 (44H=0x32H, 45H=0x33H, 46H=0x34H)

Code memory:

The screenshot displays the Proteus 8051 simulator interface. On the left, the hardware configuration panel shows the system clock set to 12.0 MHz and the update frequency set to 1. The SBUF register is configured with R/O and W/O bits set to 0x00. The RXD and TXD pins are set to 1. The SCON register is set to 0x00. The pins and bits section shows P3, P2, P1, and P0 pins with their respective bits. The PC register is set to 0x0000. The PSW register is set to 0000. The code memory section shows a table of addresses and values, with the address 0x0000 highlighted. The assembly code section on the right shows the following code:

```
ORG 40H
; N1: DB "23" ;40H=0x32H, 41H=0x33H, 42H=0x34H
; DB 0
; N2: DB "58" ;43H=0x35H, 44H=0x36H, 45H=0x37H
; DB 0
N1: DB "143" ;40H=0x31H, 41H=0x34H, 42H=0x33H
DB 0
N2: DB "234" ;44H=0x32H, 45H=0x33H, 46H=0x34H
DB 0

; This is the beginning of the program
ORG 0H
0000| MOV 50H, #100 ;Factor for highest digit
0003| MOV 51H, #10 ;Factor for the middle digit
0006| MOV 52H, #1 ;Factor for lowest digit
0009| MOV R0, #52H ;Use R0 to keep track of current digit
000B| MOV R1, #5FH ;Use R1 to keep track of current digit
000D| MOV R3, #02H ;Use R3 to determine number of digits

; This section converts string to decimal
000F| MOV DPTR, #N1 ; Load data address
0012| ST: INC R1 ; Increment R1 to point to next digit
; This section finds the # of digits
0013| L1: TNC R3 ; Use R3 to keep track of number of digits
```

The result of converting from ASCII characters to decimal numbers is stored at 60H for N1 and 61H for N2. After the conversion, N1 in hex is 8F, N2 is EA. They match with the screenshot of the data memory. The result of multiplication is 82B6H, with 82 stored in 40H and B6 stored in 41H. This also matches with the screenshot of the data memory.

Data memory:

System Clock (MHz) 

1 Update Freq.

R/O W/O

0x00 0x00

RXD TXD

1 1

SCON

0x00

TH0 TL0

0x00 0x00

TMOD TCON

0x00 0x00

TH1 TL1

0x00 0x00

PC

0x003C

R7 R6 R5 R4 R3 R2 R1 R0

0x00 0x00 0x00 0x00 0x00 0x00 0x60 0x52

B ACC PSW IP IE PCON DPH DPL SP

0x82 0xB6 0x05 0x00 0x00 0x00 0x00 0x44 0x07

pins bits

0xFF 0xFF P3

0xFF 0xFF P2

0xFF 0xFF P1

0xFF 0xFF P0

Modify RAM

addr 0x00 0x00 value

Data Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	52	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	82	B6	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	64	0A	01	00	00	00	00	00	00	00	00	00	00	00	00	00
60	8F	EA	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005–2021 James Rogers

Remove All Breakpoints

RST Step Run New Load Save Copy Paste

Time: 198us - Instructions: 131

```

001D| RPT1: CLR A ; Clear the accumulator
001E| MOVC A, @A+DPTR ; Move the digit from memory to A
001F| SUBB A, #30H
0021| MOV B, @R0 ; Move the factor into B
0023| MUL AB ; Multiply the digit by the factor
0024| ADD A, @R1 ; Sum the current digit with the product
0025| MOV @R1, A ; Save the result in memory
0026| DEC DPL ; Move DPTR to the next memory location
0028| DEC R0 ; Move R0 to the next factor
0029| DJNZ R2, RPT1 ; Repeat until all factors are used

; This section converts string "1234" to decimal
002B| MOV DPTR, #N2
002E| MOV R0, #52H
0030| DJNZ R3, ST ; When both N1 and N2 are converted

0032| Cal: MOV A, @R1 ; Copy the first digit
0033| DEC R1 ; Move R1 to the second digit
0034| MOV B, @R1 ; Copy the second digit
0036| MUL AB ; Multiply AB to get the product
0037| MOV 40H, B ; Store the upper bits of the product
003A| MOV 41H, A ; Store the lower bits of the product

003C| HERE: SJMP HERE

```