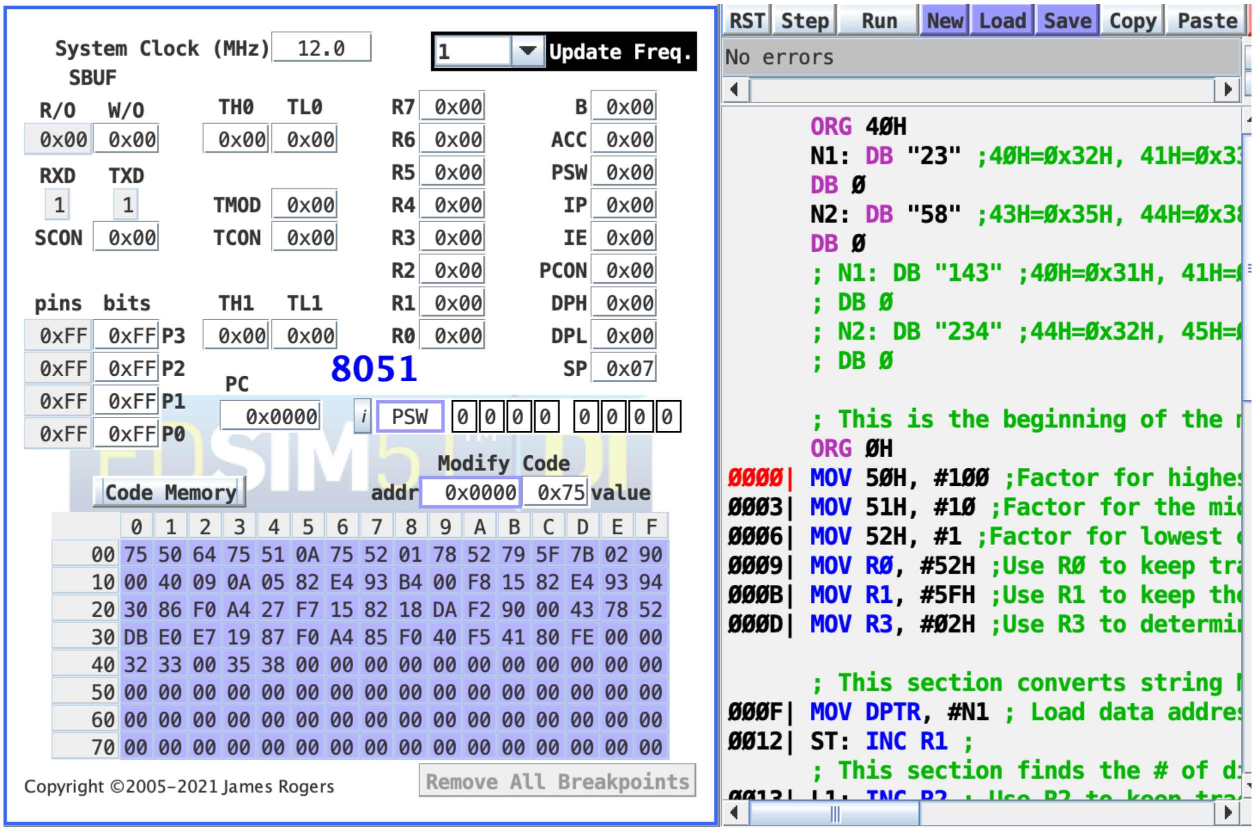
Processor Hardware/Software Interface
EECS 113

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:



```
System Clock (MHz) 12.0        1  Update Freq.
SBUF
R/O   W/O     TH0   TL0     R7 0x00      B 0x00
0x00  0x00    0x00  0x00    R6 0x00    ACC 0x00
RXD   TXD                   R5 0x00    PSW 0x00
 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 0x00    DPH 0x00
0xFF  0xFF P3 0x00  0x00    R0 0x00    DPL 0x00
0xFF  0xFF P2            8051           SP 0x07
0xFF  0xFF P1  PC
0xFF  0xFF P0  0x0000   i  PSW 0 0 0 0  0 0 0 0
                         Modify Code
Code Memory         addr 0x0000  0x75 value
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

Copyright ©2005-2021 James Rogers

No errors

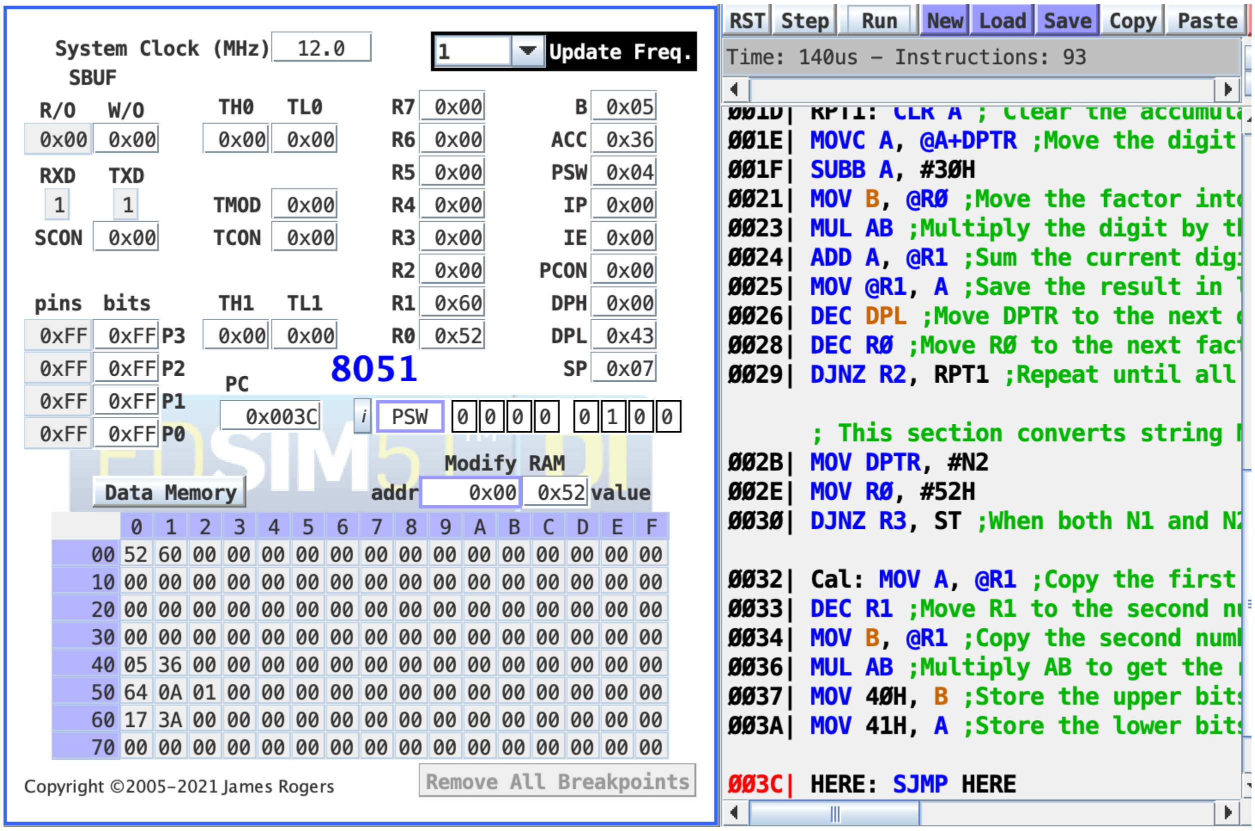
```
       ORG 4ØH
       N1: DB "23" ;4ØH=Øx32H, 41H=Øx3
       DB Ø
       N2: DB "58" ;43H=Øx35H, 44H=Øx3
       DB Ø
       ; N1: DB "143" ;4ØH=Øx31H, 41H=
       ; DB Ø
       ; N2: DB "234" ;44H=Øx32H, 45H=
       ; DB Ø

       ; This is the beginning of the
       ORG ØH
ØØØØ|  MOV 5ØH, #1ØØ ;Factor for highe
ØØØ3|  MOV 51H, #1Ø ;Factor for the mi
ØØØ6|  MOV 52H, #1 ;Factor for lowest
ØØØ9|  MOV RØ, #52H ;Use RØ to keep tr
ØØØB|  MOV R1, #5FH ;Use R1 to keep th
ØØØD|  MOV R3, #Ø2H ;Use R3 to determi

       ; This section converts string
ØØØF|  MOV DPTR, #N1 ; Load data addre
ØØ12|  ST: INC R1 ;
       ; This section finds the # of d
ØØ13|  L1: INC R2 ; Use R2 to keep tra
```

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) 12.0        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  0x00    R0  0x52      DPL  0x43
0xFF  0xFF P2    PC         8051          SP   0x07
0xFF  0xFF P1    0x003C     i  PSW  0 0 0 0  0 1 0 0
0xFF  0xFF P0
                            Modify RAM
Data Memory            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

Time: 140us – Instructions: 93

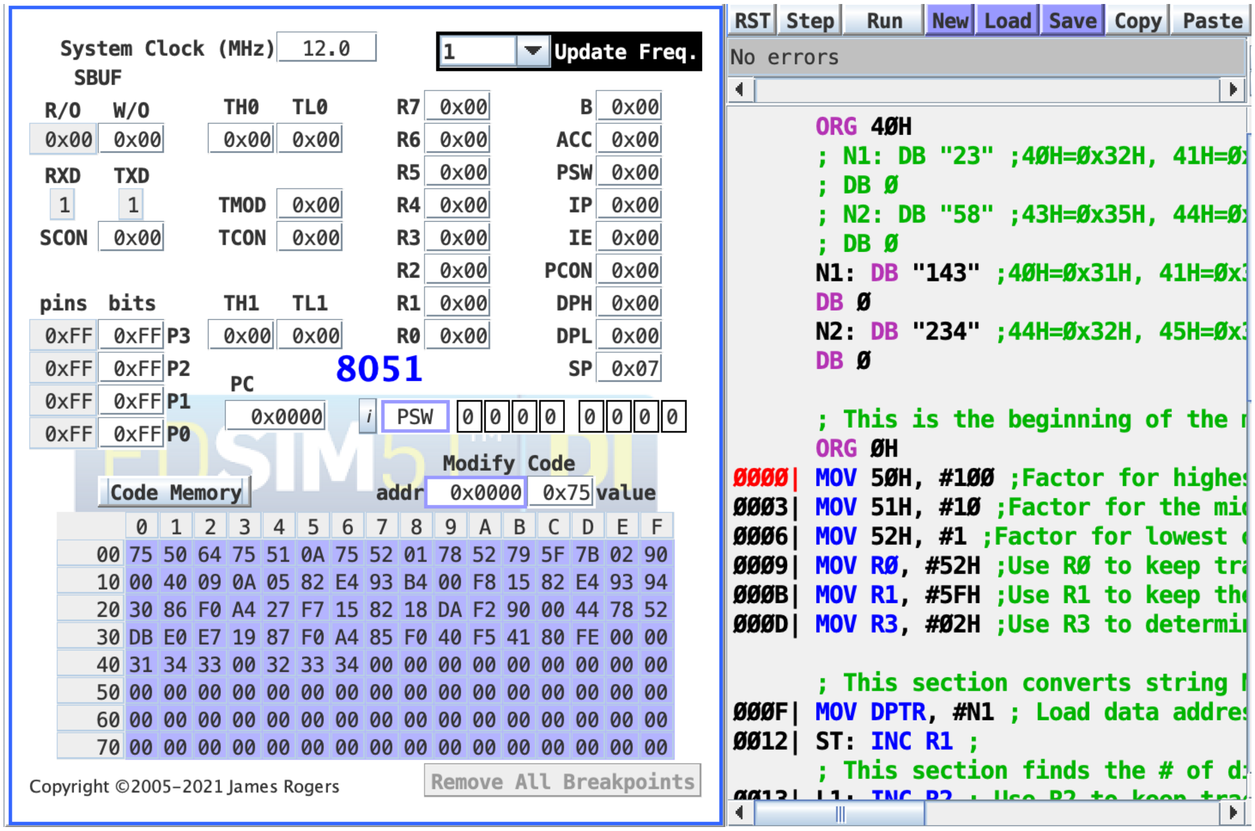
```
ØØ1D| RPT1: CLR A ; Clear the accumula
ØØ1E| MOVC A, @A+DPTR ;Move the digit
ØØ1F| SUBB A, #3ØH
ØØ21| MOV B, @RØ ;Move the factor int
ØØ23| MUL AB ;Multiply the digit by t
ØØ24| ADD A, @R1 ;Sum the current dig
ØØ25| MOV @R1, A ;Save the result in
ØØ26| DEC DPL ;Move DPTR to the next
ØØ28| DEC RØ ;Move RØ to the next fac
ØØ29| DJNZ R2, RPT1 ;Repeat until all

      ; This section converts string
ØØ2B| MOV DPTR, #N2
ØØ2E| MOV RØ, #52H
ØØ3Ø| DJNZ R3, ST ;When both N1 and N

ØØ32| Cal: MOV A, @R1 ;Copy the first
ØØ33| DEC R1 ;Move R1 to the second n
ØØ34| MOV B, @R1 ;Copy the second num
ØØ36| MUL AB ;Multiply AB to get the
ØØ37| MOV 4ØH, B ;Store the upper bit
ØØ3A| MOV 41H, A ;Store the lower bit

ØØ3C| HERE: SJMP HERE
```

Test Case 2:
N1 = 143 (40H=0x31H, 41H=0x34H, 42H=0x33H),
N2 = 234 (44H=0x32H, 45H=0x33H, 46H=0x34H)
Code memory:



```
System Clock (MHz) 12.0        1
SBUF
R/O   W/O     TH0   TL0     R7 0x00    B    0x00
0x00  0x00    0x00  0x00    R6 0x00    ACC  0x00
RXD   TXD                   R5 0x00    PSW  0x00
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 0x00    DPH  0x00
0xFF  0xFF P3 0x00  0x00    R0 0x00    DPL  0x00
0xFF  0xFF P2                          SP   0x07
0xFF  0xFF P1   PC
0xFF  0xFF P0   0x0000   i  PSW  0 0 0 0  0 0 0 0
                                 Modify Code
                          addr 0x0000  0x75 value
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	44	78	52
30	DB	E0	E7	19	87	F0	A4	85	F0	40	F5	41	80	FE	00	00
40	31	34	33	00	32	33	34	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

No errors

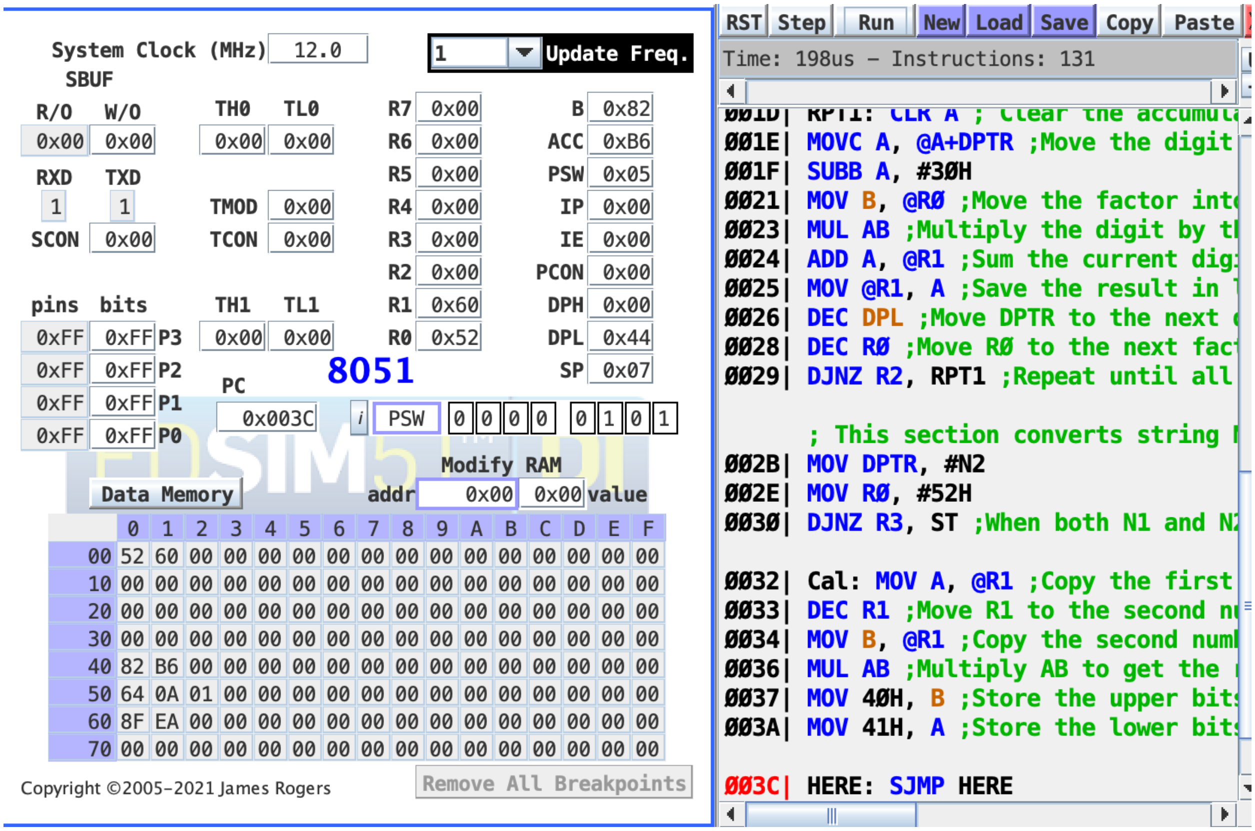
```
      ORG 40H
      ; N1: DB "23" ;40H=0x32H, 41H=0x
      ; DB 0
      ; N2: DB "58" ;43H=0x35H, 44H=0x
      ; DB 0
      N1: DB "143" ;40H=0x31H, 41H=0x3
      DB 0
      N2: DB "234" ;44H=0x32H, 45H=0x3
      DB 0

      ; This is the beginning of the r
      ORG 0H
0000| MOV 50H, #100 ;Factor for highes
0003| MOV 51H, #10 ;Factor for the mic
0006| MOV 52H, #1 ;Factor for lowest d
0009| MOV R0, #52H ;Use R0 to keep tra
000B| MOV R1, #5FH ;Use R1 to keep the
000D| MOV R3, #02H ;Use R3 to determir

      ; This section converts string N
000F| MOV DPTR, #N1 ; Load data addres
0012| ST: INC R1 ;
      ; This section finds the # of di
0013| L1: INC R2 ; Use R2 to keep tra
```

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) 12.0          1   Update Freq.
SBUF
R/O   W/O     TH0   TL0     R7 0x00      B 0x82
0x00  0x00    0x00  0x00    R6 0x00    ACC 0xB6
RXD   TXD                   R5 0x00    PSW 0x05
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  0x00    R0 0x52    DPL 0x44
0xFF  0xFF P2                           SP 0x07
0xFF  0xFF P1  PC
0xFF  0xFF P0  0x003C   PSW 0 0 0 0  0 1 0 1
                              addr 0x00   0x00 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 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

Time: 198us - Instructions: 131

```
001D| RPT1: CLR A ; Clear the accumula
001E| MOVC A, @A+DPTR ;Move the digit
001F| SUBB A, #30H
0021| MOV B, @R0 ;Move the factor int
0023| MUL AB ;Multiply the digit by t
0024| ADD A, @R1 ;Sum the current dig
0025| MOV @R1, A ;Save the result in
0026| DEC DPL ;Move DPTR to the next
0028| DEC R0 ;Move R0 to the next fac
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 N

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

003C| HERE: SJMP HERE
```