Report For Lab 2

Connor McCullough

EEN 312

13 February 2013

**Objective:**

Implement the code to return the value for the function (A+B)^2 by first using the MUL function to calculate A\*A, A\*B, and B\*B, and then using the ADD function to sum them together.

**Equipment Used:**

Software:

* Text editor and 8051 ASM assembler
* Step debugger to execute program one instruction at a time
* Data memory, accumulator, code memory

**Flow Chart:**

**Add R5 to A**

Is carry bit set?t set?

**Set R2 to A**

**START**

**Add R7 to A**

**Set R3 to B**

**Set A to 102**

**Set A to 1**

**Set 23H to A**

**Set A to 125**

**Add R1 to A**

**Set B to 102**

**Set A to 20H**

**Add R3 to A**

**Set B to 102**

**Multiply A and B; Low byte stored in A, high byte in B**

**Add 22H to A**

**Set 21H to A**

**Set 24H to A**

**Multiply A and B; Low byte stored in A, high byte in B**

Is carry bit set?t set?

**Set A to R4**

**Set R0 to A**

**Add R6 to A**

**Set R1 to B**

**Set R6 to A**

**Set A to 1**

**Set 22H to A**

**Set R7 to B**

**Set A to 102**

**Add 21H to A**

Is carry bit set?t set?

**Set A to R0 plus R2**

**Set B to 125**

**Add 23H to A**

**END**

**Set 25H to A**

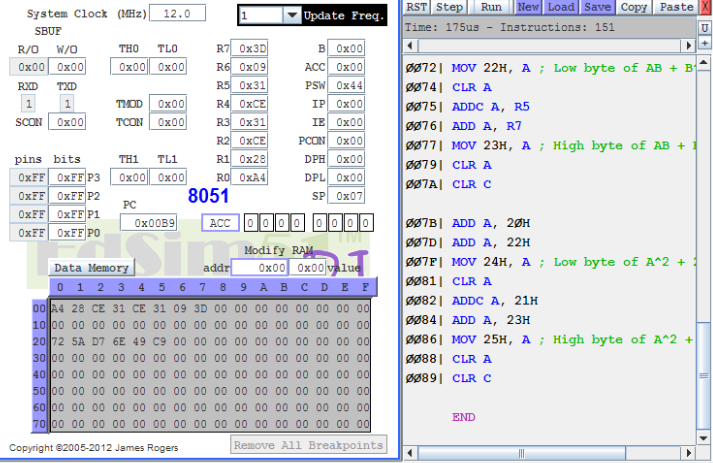
**Set A to 1**

**Set 20H to A**

**Multiply A and B; Low byte stored in A, high byte in B**

**Results:**

The memory after the entire program has run. 24H and 25H in the data memory contain 49 and C9 respectively, with 25H as the high byte and 24H as the low byte, which in decimal is 51529. This satisfies the function when A = 102 and B = 125.

****

**Conclusion:**

This lab taught the versatility of the MUL and ADD functions in calculating complex mathematical functions. Another important conclusion from this lab is that the same program can be written in multiple ways, differing significantly in lines of code and overall efficiency. One more lesson learned from this lab is that the ADDC function can be used for calculations of very large sizes, much larger than a single 8 bit register.

**Code:**

; “A” = 102, “B”=125

Org 30H

CLR A   
MOV A, #102 ; Move “A” to accumulator  
MOV B, #102 : Move “A” to register B  
MUL AB ; A^2 term now in A & B registers  
MOV R0, A ; Low byte A\*A  
MOV R1, B ; High byte A\*A

CLR A  
MOV B, #0H ; This is how the B register is cleared.

MOV A, #102 ; Move “A” to accumulator  
MOV B, #125 ; Move “B” to register B  
MUL AB ; A\*B term now in A & B registers  
MOV R2, A ; Low byte A\*B  
MOV R3, B ; High byte A\*B

CLR A  
MOV B, #0H

ADD A, #125  
MOV B, #102 ; To get 2AB term, A\*B must be added to itself  
MUL AB  
MOV R4, A ; Low byte A\*B  
MOV R5, B ; High byte A\*B

CLR A  
MOV B, #0H

ADD A, #125 ; Move “B” to accumulator  
MOV B, #125 ; Move “B” to register B  
MUL AB ; B^2 term now in A & B registers  
MOV R6, A ; Low byte B\*B  
MOV R7, B ; High byte B\*B

CLR A  
MOV B, #0H

ADD A, R0  
ADD A, R2 ; Only add two terms to the accumulator at once because only one bit can be carried.   
MOV 20H, A ; Low byte of A^2 + AB

CLR A  
ADDC A, R1 ; Adds 1 to the high byte if addition of the low bytes set the carry flag.  
ADD A, R3  
MOV 21H, A ; High Byte of A^2 + AB

CLR A  
CLR C ; Make sure carry is set to 0 before further calculations

ADD A, R4  
ADD A, R6  
MOV 22H, A ; Low byte of AB + B^2

CLR A  
ADDC A, R5  
ADD A, R7  
MOV 23H, A ; High byte of AB + B^2

CLR A  
CLR C

ADD A, 20H ; The two sets of terms are added together to get the complete function.  
ADD A, 22H  
MOV 24H, A ; Low byte of A^2 + 2AB + B^2

CLR A  
ADDC A, 21H  
ADD A, 23H  
MOV 25H, A ; High byte of A^2 + 2AB + B^2  
CLR A  
CLR C

END