Lab 2

13 February, 2013

Andrew O'Neil-Smith

Objective-

This lab's objective is to write and 8051 assembly program to calculate (A+B)^2 in two different ways.

Equipment used-

Software:	a text editor and an 8051 ASM assembler	
	A step debugger that can be used to execute a program one step at	
	a time	
	Register, code memory, data memory, and input/output port	
	contents are displayed to aid debugging.	
Flow Chart-		

Test Results-

This screenshot shows that the first part of the given code was computed

correctly, getting C949h.

💰 EdSim51DI - Version 2.1.5				
System Clock (MHz) 12.0 1 Vpdat	Freq. RST Step Run New Load Save Copy Paste X Executed 0x003D: MOV 21H, 0F0H 1 Time: 61us U	P0.7 1 Display-select Decoder CS DAC WR P0.6 1 Keypad Column 2 P0.5 1 Keypad Column 1		
R/O W/O THO TLO R7 0x00 B 0x00 0x00 0x00 0x00 R6 0x00 ACC 0 RXD TXD R5 0x00 PSW 0	C9 () + + 49 org 3ØH 45	P0.4 1 Keypad Column 0 P0.3 1 Keypad Row 3 P0.2 1 Keypad Row 2		
1 1 TMOD 0x00 R4 0x00 IP 0 SCON 0x00 TCON 0x00 R3 0x00 IE 0 P0 0x00 PCON 0x00 P0 0x00 P0 0x00	00 ØØ3Ø mov B, #125 ØØ33 CLR A	P0.1 Keypad Row 1 P0.0 Keypad Row 0 P1.7 LED 678eg. dp1DAC DB71LCD DB7 P1.6 LED 678eg. g1DAC DB61LCD DB6		
pins bits TH1 TL1 R1 0x00 DPH 0xFF 0xFF 0x0F 0x00 0x00 DPL	Øø34 ADD A, #1ø2 Øø36 ADD A, B Øø38 MOV B, A	P1.5 1 LED 5 Seg. f DAC DB5 LCD DB5 P1.4 1 LED 4 Seg. e DAC DB4 LCD DB4 P1.3 1 LED 3 dl. DB3 DB3 R5 P1.4 1 LED 3 dl. DB3 DB3 R5		
0xFF 0xFFP2 pc 8051 sp 0xFF 0xFFP1 0x0040 ACC 0 1 0 1 0	07 ØØ3A MUL AB	P1.2 1 LED 2[e[DE2].DE2[LED E P1.1 1 LED 1[Seg. b]DAC DB1 LCD DB1 P1.0 1 LED 0[Seg. s]DAC DB0 LCD DB0 P2.7 1 SW 7[ADC DB7		
Modify RAM Data Memory addr 0x00 0x00	posil MOV 2011, A	P2.6 1 SW 6[ADC DB6 P2.5 1 SW 5]ADC DB5 P2.4 1 SW 4[ADC DB4 P2.4 4 SW 4]ADC DB4		
00 00 00 00 00 00 00 00 00 00 00 00 00	00 END 00 00	P2.3 SW 2 ADC DB2 P2.1 SW 2 ADC DB2 P2.1 SW 1 ADC DB1 P2.0 SW 0 ADC DB0		
30 00<		P3.7 1 ADC RD Comparator Output P3.6 1 ADC WR P3.5 1 Motor Sensor		
60 00 00 00 00 00 00 00 00 00 00 00 00 0	00 00	P3.4 1 Display-select Input 1 P3.3 1 AND Gate Output[Display-set 0 P3.2 1 ADC INTR P3.1 1 Motor Control Bit 11Ext UABT Rx		
Copyright @2005-2012 James Rogers Remove All Break	oints	P3.0 1 Motor Control Bit 0 Ext. UART Tx		
1 2 3 AND Gate Disabled U No Parity 8-bit UART @ 4800 Baud 1 2 3 AND Gate Disabled U No Parity 8-bit UART @ 4800 Baud 1 2 3 AND Gate Disabled Key Bounce Disabled Rx Rx Reset 0.0 V input				
0.0 V output Scope DAC BF 0 AC 0x00 IR 0x00 / BF 0 AC 0x00 IR 0x00 /				

This screen shows that our code computes C949h correctly, located in the bottom

right corner of Data Memory.



Conclusion-

This lab required us to think in a non-conventional way to perform the proper order of operations. It helped me to understand how the 8051 does simple math, with the help of the ADDC function.

Program-

org 0h ;;A*A mov a, #102 mov b, #102 mul ab mov 20h, a mov 21h, b

clr a ;;A*B mov a, #102 mov b, #125 mul ab mov 30h, a mov 31h, b clr a ;;A*B mov a, #102 mov b, #125 mul ab mov 40h, a mov 41h, b clr a ;;B*B mov a, #125 mov b, #125 mul ab mov 50h, a mov 51h, b clr a ;starting to calculate with the carry function ;;Lower of Asquared + lowerAB mov A, 20h add A, 30h mov 22h, a clr a ;;Upper of Asquared + upperAB mov a, 21h addc a, 31h mov 32h, a clr a

;;Lower of AB + lower Bsquared mov a, 40h add a, 50h mov 42h, a clr a ;;uppoer of AB + Lower Bsquared mov a, 41h addc a, 51h mov 52h, a clr a ;;Lower of final result mov a, 22h add a, 42h mov 7Eh, a clr a ;; Upper of final result mov a, 32h addc a, 52h mov 7Fh, a end