

# Assignment # 3: Problem Set 2, Problem 1

Brian Arnberg

1/29/2013

## Assignment

For the following, write an ARM assembly language program, and in the Keil MDK-ARM IDE, create a project, enter the program, and then execute and debug it in the Keil MDK-ARM debugger. You may run the program either in the simulator or in RAM on the STM32F4-DISCOVERY board. All program variables are to be 32-bit integers. You may choose your own test data values.

Compute:

`zz = aa*(bb+cc) (dd*35)`

Place aa,bb,cc,dd in the code area, so that you can provide initial values with "DCD" assembler directives. Place zz in the data area, so that you can write the result to it. The final debug window should show the final value of zz in memory.

## Debugging

The ARM assembly language program was written inside a Linux environment, with the expectation that it would be compiled against an "ARM" version of gcc (`arm-none-eabi-as`). The compiler did not accept the line labels, and would not compile the code. This led to the code being debugged on a school computer. The school computers do not have the proper drivers installed to communicate with the STM32F4, so the program was debugged inside the Simulator. Because the program was installed inside the simulator, the Target addresses for the ROM and RAM portions were left at their default values (0x8000000 and 0x20000000, respectively). The "Memory" section of the debugger is displayed in Figure 1, and the value of zz has been circled. The expected value for zz is -135, and this is the value stored in the memory location for zz. Therefore, the program worked as expected.

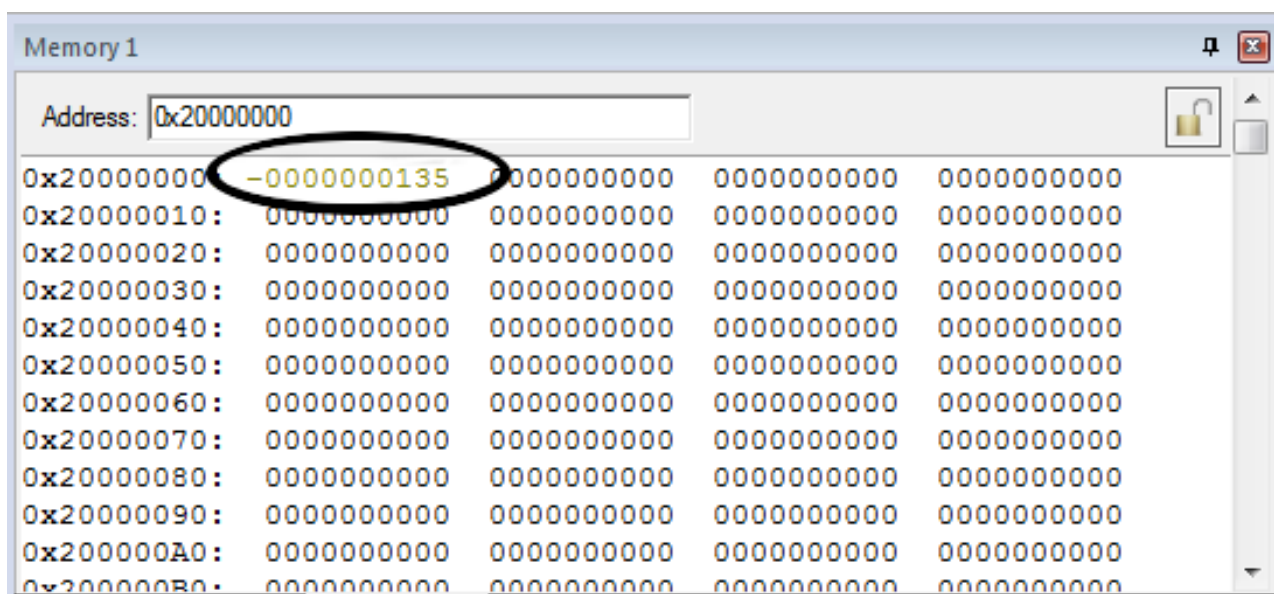


Figure 1: A screen capture of the memory box inside the debugger window. The expected value for zz is -135, and this is what is displayed. Therefore, the debugged program indicates that the program works as expected.

## Source Program

PS2-1.s

```

1  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2  ;;      Brian Arnberg - ELEC6260                      ;;
3  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4  ;; Problem Set 2, problem 1                          ;;
5  ;; Execute the following:                             ;;
6  ;;      zz = aa*(bb+cc) - (dd*35);                    ;;
7  ;;      define: aa,bb,cc,dd in code area              ;;
8  ;;      zz in data area                               ;;
9  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10
11      AREA      reset , CODE
12              ; Name this CODE block , reset
13      ENTRY      ; Mark first instruction to execute
14              ; aa*(bb+cc)
15      LDR r4 , =bb0      ; Get address of bb0
16      LDR r0 , [r4]      ; Store value of bb0 in r0
17      LDR r4 , =cc0      ; Get address of cc0
18      LDR r1 , [r4]      ; Store value of cc0 in r1
19      ADD r2 , r0 , r1    ; Compute bb0+cc0, store to r2 (r2 = bb0+cc0)
20      LDR r4 , =aa0      ; Get address of aa0
21      LDR r0 , [r4]      ; Store value of aa0 in r0
22      MUL r2 , r2 , r0    ; Compute aa0*(bb0+cc0), store to r2
23              ; (dd*35)
24      LDR r4 , =dd0      ; Get address of dd0
25      LDR r0 , [r4]      ; Store value of dd0 in r0
26      MOV r1 , #35       ; Store #35 to R1
27      MUL r3 , r0 , r1    ; Compute dd0*35, store to r3
28              ; aa*(bb+cc) - (dd*35)
29      SUB r2 , r2 , r3    ; Final Computation: r2 = r2 - r3
30              ; store to address of zz
31      LDR r4 , =zz0      ; Get address of zz0
32      STR r2 , [r4]      ; Store result to address of zz0 ( [zz0] <= r2 )
33
34              ; Declare variable values in code block
35 aa0      DCD 0x01      ; Declare value of aa0
36 bb0      DCD 0x02      ; Declare value of bb0
37 cc0      DCD 0x03      ; Declare value of cc0
38 dd0      DCD 0x04      ; Declare value of dd0
39
40      AREA      data1 , DATA
41              ; Name this DATA block , data1
42 zz0      SPACE 4        ; Set space aside for variable zz0
43      END

```

PS2-1.s is the ARM assembly language program written to compute  $zz = aa*(bb+cc) - (dd*35)$ . The values I selected for aa, bb, cc, and dd were 1, 2, 3, and 4, respectively. The expected value for zz is -135.