Oregon State University

CS 472 - Computer Architecture

Spring 2014

Lab 2 - The ARM ISA

Author:
Drake Bridgewater
Ryan Phillips

Professor:
Kevin McGrath

April 17, 2014

Getting started

Begin with reading the introduction the KEIL simulator on pages 16-29 of the Student Workbook which accompanies the textbook

Part 1

Please provide answers to the problems in Problem Set 5 in the Student Workbook. Here we provide an introduction to the Keil ARM processor development system.

1) Write a simple program to perform: $Z = A+B+B-(D\times E)$. The instructions you may use are ADD, SUB, and MUL. Assume that the data is in registers r0 to r4 (representing A to E) and the result is put in r5. Enter your program into the Keil simulator and run it. You can use move instruction to load data into registers. Do you get the expected answer?

lab2_part1/lab2_1.asm

```
AREA lab2_1, CODE, READONLY
ENTRY

MOV r0, #6      ;a
MOV r1, #3      ;b
MOV r2, #4      ;c; MOV r3, #3      ;d
MOV r4, #2      ;e


ADD r1,r1,r1       ;r1=r1+r1 || b=b+b
ADD r1,r1,r0       ;r1=r1+r0 || b=a+(b+b)

;for muls you have to have diff destination register
MULS r5,r3,r4      ;r5=r3+r4 || z=d*e
SUB r5,r1,r5       ;r5=r1-r3 || z=b-(d*e)

END
```

While watching the register reflect each line of code I was able to view each commands effects on the bits. This allowed me to realize that the add operation and subtraction operations are performed exactly as expected, but once the multiplication command came into play I noticed that when the numbers are smaller minus larger the result is a negative.

2) Now assume A, B, C, D and E are 16-bit values in memory. You can load them by using a DCD directive. Enter the program, compile (build) it and test it.

lab2_part1/lab2_2.asm

```
    AREA lab2_2, CODE, READONLY
    ENTRY

    ADR r0, A
    ADR r2, C
    ADR r4, D
    ADR r6, E
    ADR r8, F
    ADR r10, Z
A   DCD  4
C   DCD  -2
D   DCD  3
E   DCD  -12
F   DCD  5
Z   DCD  0


    ADD r2,r2,r2        ; b=b+b
    ADD r2,r2,r0        ; b=a+(b+b)

    MULS r10,r6,r8        ; z=d*e
    SUB r10,r2,r10        ; z=b-(d*e)

    END
```

3) Write a program that includes deliberate syntax errors. Enter it in the development system, assemble (build) it and then debug it.

lab2_part1/lab2_3_errors.asm

```
        AREA lab2_3_errors, CODE,
           READONLY
 2      ENTRY

 4    ADR r0, A
      ADR r2, C
 6  ADR r4, D
      ADR r6, E
 8    ADR r8
      ADR r10, Z
10  A     DCD  4
    C     DCD
12  D     DCD  3
    E     DCD  -12
14  F     DCD  5
    Z     DCD  0
16

18    ADD r2,r2
      ADD r2,r2,r0
20
      MULS r10,r6,r8
22    SUB r10,r2,r10
```

lab2_part1/lab2_3_fixed.asm

```
 1      AREA lab2_3_fixed, CODE,
           READONLY
        ENTRY
 3
        ADR r0, A
 5      ADR r2, C
        ADR r4, D
 7      ADR r6, E
        ADR r8, F
 9      ADR r10, Z
    A   DCD  4
11  C   DCD  -2
    D   DCD  3
13  E   DCD  -12
    F   DCD  5
15  Z   DCD  0

17
        ADD r2,r2,r2
19      ADD r2,r2,r0

21      MULS r10,r6,r8
        SUB r10,r2,r10
23
        END
```

```
assembling lab2_3_errors.asm...
lab2_3_errors.asm(6): error: A1163E: Unknown opcode r4, expecting opcode or Macro
lab2_3_errors.asm(8): error: A1106E: Missing comma
lab2_3_errors.asm(11): error: A1110E: Expected constant expression
lab2_3_errors.asm(24): warning: A1313W: Missing END directive at end of file
assembling lab2_3_fixed.asm...
".\lab2.axf" - 3 Error(s), 1 Warning(s).
```

Error on line 6: can be fixed by adding tabbing out the command out. The compiler thinks that ADR is a label and that r4 is the opcode; that is why it is saying unknown opcode r4. Error on line 8: can be fixed by adding comma but will still complain if you only add a comma. Therefore you will have to add an additional operand. Error on line 11: can be fixed by adding the expected expression, either numeric or PC-relative, as the Directive allocates one or more words of memory aligned on four-byte boundaries. Error on line 24: can be fixed by telling the compiler where the end of the program is or placing 'END' on the last line. Error on line 18: this error is not seen as an error but as a bug; the attempt is to add the two register but you must specify the Rd or destination, Rn or the first operand and operand2 or the second operand in order to sum two values.

Part 2: Examination of compiler output

In the homework, you were asked to write an assembly routine that checks for palindromes in odd length strings. For this portion of the lab, please write a simple C program that does the same thing, use the web based compiler and compile your C code to ARM assembly (use arm-linux-gnueabi-g++-4.6 with option -O0). Compare this assembly with what you wrote. Modify the optimization level from 0 to one of the values in the set 1,2,3,s. What changes? Why do you think these changes were made?

For this lab, you will need to create a write-up discussing the differences you see in Part 2. Do you see any interesting features being used? Is the hand assembly you wrote significantly different than what the compiler produced? Which uses fewer instructions? In other words, provide an analysis of Part 2.