CS 219: Homework #5

Due on October 5th, 2016 at 4:00pm

Dr. Egbert

Matthew J. Berger

Problem 4

Review Questions

4.1.) a) When ADD r3, r5, r12 is executed what happens to the value that was in r3?

b) What is the effect (result) of executing ADD r3, r5, r5?

c) What is the effect (result) of executing ADD r3, r3, r3?

Answer:

a) The value that was in r3 becomes the sum of the values stored in r5 and r12.

b) The value stored in r5 is doubled and stored in r3. The original value of r5 remains unchanged.

c) The value stored in r3 is doubled in place. r3 now contains the doubled value.

4.2.) a) When MOV r11, r2 is executed what happens to the value that was in r11?

b) What is the effect (result) of executing MOV r4, #28?

c) What is the effect (result) of executing MOV r3, r3?

Answer:

a) The value that was in r11 is lost. The value in r11 is now the same as the value in r2.

b) The binary pattern equivalent to the decimal number 28 is copied into register r4.

c) The r3 register is updated with the value it already contains, thus the r3 registered is not changed and contains the same value after this operation.

4.3.) Prepare a program to add together the numbers 127 decimal, 0xe45ad hexadecimal and 2_10101110010 binary (ARM assembly language uses prefix 2_ to indicate a number in base 2). Using a development system, assemble your program repeatedly correct it and assemble again until there are no errors. Test that the program behaves correctly. What is the total obtained?

Answer:

```
        AREA Add_Example, CODE, READONLY ; code module named Add_Example
        ENTRY                            ; Indicate which instruction to run first

mystart MOV r3, #127      ; load first value
        MOV r7, #0xe45ad    ; load second value
        MOV r8, #2_10101110010 ; load third value
        ADD r2, r3, r7 ; Form the sum of the first two values
        ADD r2, r2, r8 ; Add the third value
stop

; The next 3 lines are used by ARM systems (e.g. Armulator, Evaluator-7T) to
; automatically return to the IDE when the program finishes.
        MOV r0, #0x18
        LDR r1, =0x20026
        SWI 0x123456
        END
```

The total obtained is: **0xE4B9E** in hex, **93682** in decimal, and **1110 0100 1011 1001 1110** in binary.

4.4.) a) Assume that register r2 holds the value 0x0f45, what is the value in register r5 after the execution of the instruction SUB r5, r2, #209?

b) Assume that the register r2 holds the value 0x045, what is the value in register r5 after the execution of the instruction RSB r5, r2, #209?

c) What problems arise when register r2 holds 0x0f45 and the instruction RSB r5, r2, #209 is executed? What will be the value in register r5 after the execution of this instruction?

Answer:

a) 0x0f45 is equivalent to 3909 in decimal. $3909 - 209 = 3700$ which is equivalent to the bit pattern 1110 0111 0100. This is the new value of r5, left padded to 32 bits with leading zeroes.

b) RSB stands for reverse substraction. 0x045 is equivalent to 69 in decimal. $209 - 69 = 140$ which is equivalent to the bit pattern 1000 1100. This is the new value of r5, left padded to 32 bits with leading zeroes.

c) 0x0f45 is equivalent to 3909 in decimal. The instruction attempts to do the following: $209 - 3909 = -3700$. The decimal number -3700 takes 16 bytes (64 bits) to represent. The ARM registers are only 32-bit, so there would be overflow and the resulting value in r5 would be the bit pattern 0001 1000 1100, left padded to 32 bits by leading 1's.

4.5.) a) What is the value in register r8 after the execution of the instruction MVN r8, #0xf4?

b) Assume that the register r3 holds the value 0x045 and r10 holds 0xffff. What is the value in register r3 after the execution of the instruction ADD r3, r3, r10?

c) Describe the form of the value that will be in register r6 after the execution of the instruction ADD r6, r6, #1?

Answer:

a) 0xf4 is equivalent to 1111 0100 in binary. When negated, it is equivalent to 0000 1011. So the value in r8 after this operation is 0000 1011, left padded to 32 bits by leading 1's (because of the negation).

b) 0x045 is equivalent to 69 in decimal. 0xFFFF is equivalent to 65535 in decimal. $65535 + 69 = 65604$, which is equivalent to the binary pattern 0001 0000 0000 0100 0100, left padded to 32 bits by leading zeroes. This value is the new value in register r3.

c) The form of the value that will be in register r6 after this instruction would be the current value in r6 plus one. r6 would be incremented by one after this instruction.