Homework 3
CPE221

Instructor: Rahul Bhadani

Due: Feb 07, 2025, 11:59 PM
100 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

Upload a .pdf on Canvas with the format {firstname.lastname}_CPE221_hw03.pdf. For example, if your name is Sam Wells, your file name should be sam.wells_CPE221_hw03.pdf. If there is a programming assignment, then you should include your source code along with your PDF files in a zip file {firstname.lastname}_CPE221_hw03.zip. Your submission must contain your name, and UAH Charger ID or the UAH email address. Please number your pages as well.

---

1 IEEE 754 Floating-point Standard(10 points)

Convert the following decimal numbers to the binary representation using the IEEE 754 single precision format. Show sign, exponent, and mantissa separately. For example, -19 has sign of -1, exponent 2^4, and mantissa 1.1875.

1. -1609.5
2. -938.8125

3. 130.59375

4. -37.3125

5. 2.375

2 Flag it! (10 points)

For each of the following 8-bit operations, calculate the values of the C, Z, V, and N flags.

1. 1010 1101 + 1010 1101

2. 1111 1111 + 0100 0001

3. 0110 0100 - 1110 1101

4. 0010 1111 + 0110 1111

5. 0100 1000 − 1110 1001

3 On the way to be a hacker! (10 points)

If R1 = 0xA536 2C89 what is the value of R0 after each of the following instructions has been executed (assume that each instruction uses the same data)?

1. ADD R0, R1, R1, LSL #4

2. ADD R0, R1, R1, ASR #8

3. ADD R0, R1, R1, ROR #9

4. If R2 = 0xB903 8471 and R3 = -160, Write down effective address is generated by the instruction

   LDR R0, [R2, R3, LSL #1].

   Express your answer in hexadecimal.

4 Decipher (10 points)

Explain the effect of each of the following six instructions. Use register transfer notation.

1. LDR R1, [R4]
2. LDR R1, [R4, #12]
3. STR R1, [R2, #16]
4. STR R1, [R2, R1, ASR #4]
5. LDR R1, [R2, R1, LSL #9]

5 No Shortcut (10 points)

Without using the ARM's multiplication instruction, write one or more instructions (using ADD, SUB, and shifting) to multiply the contents of R1 by 145 and put the result in R0

6 Closer to being a Robot! (20 points)

Write ARM assembly code to implement the following C statements, assuming all variables are 32-bit integers and a declaration of $x[10]$, $y[10]$ and size $= 10$. Make sure to add comments in your assembly code documenting what your assembly code means and what they are trying to do. Comments in assembly code start '@'. You can assume about the pointer to the first element of a register storing x, y, and other constants.

```c
int x[10] = {8, 2, 9, 6, 7, 0, 1, 3, 5, 4};
for i = 0; i < 10; i++)
{
    y[i] = x[i] + i;
}
```

7 Am I a Hacker Now! (20 points)

Write an equivalent C code for the following: (Note: MVN is the bitwise-NOT operation in ARM assembly.)

1.
   ```
   MOV R6, R4
   LSL R6, R6, #2
   SUB R6, R4, R6
   MOV R8, R6
   ```

2.
   ```
   MOV R3, R4
   MUL R5, R4, #4
   ASR R3, R3, #3
   ADD R3, R3, R5
   MOV R8, R3
   ```

3.
   ```
   MOV R10, R4
   LSL R2,        R4, #2
   LSL R2, R2, R4
   MOV R8, R2
   ```

4.
   ```
   MOV R9, R4
   MVN R9, R9
   AND R9, R9, R4
   MOV R8, R9
   ```

5.
   ```
   MOV R12, R4
   AND R12, R12, #15
   ASR R1, R4, #2
   ADD R12, R12, R1
   MOV R8, R12
   ```

8 Being a Polyglot in Computing. (10 points)

Interpret the following Assembly code and write the value of each register at every line.

```
    MOV R5, #0
    MOV R6, #1
    MOV R1, #2
LOOP:
    CMP R1, #15
    BGE EXIT_LOOP

    ADD R7, R5, R6
    MOV R5, R6
    MOV R6, R7
    ADD R1, R1, #1
    B LOOP
EXIT_LOOP

```