CSE2312 (Fall 2022)
Homework #2

Notes:
With this homework, we move from abstract processor concepts to the ARM processor used on the RPi3b/3b+/4b when running 32-bit Raspberry Pi operating system.

We also start writing assembly functions for the RPi 3b/3b+/4b.

All numbers are in base-10 unless otherwise noted.

If part of a problem is not solvable, explain why in the answer area.

The target date to complete this homework set is September 13, 2022.

This homework set will not be graded, but please solve all of the problems to prepare for the quizzes and exams.

1. For each of the following operations, show the value of R0 in <u>base-10 unsigned representation</u> (e.g., If R0 = 16384, then R0 LSR #1 = 8192).

In the questions below, it using arithmetic operations on unsigned numbers is noted as abnormal.

For these questions, assume that the register R0 contains an unsigned 32-bit integer (e.g., uint32_t) with a value of 2048 (0x00000800).

    a. R0 LSR #8:   8 _____

    b. R0 LSR #9:   4 _____

    c. R0 LSR #10:   2 _____

    d. R0 LSR #11:   1 _____

    e. R0 LSR #12:   0 _____

    f. R0 LSL #20:   2147483648 _____

    g. R0 LSL #21:   0 _____

    h. R0 LSL #22:   0 _____

    i. R0 ASR #8:   8 _____ (abnormal ASR usage)

For these questions, assume that the register R0 contains an unsigned 32-bit integer (e.g., uint32_t) with a value of 3758096384 (0xE0000000).

    j. R0 LSL #1:   7516192768 _____

    k. R0 LSR #1:   1879048192 _____

    l. R0 ASR #1:   4026531840 _____ (abnormal ASR usage)
                                                 abnormal to use ASR for UNsigned numbers

2. For each of the following operations, show the value of R0 in <u>base-10 signed representation</u> (e.g., If R0 = -64, then R0 ASR #1 = -32).

In the questions below, it using logical operations on signed numbers is noted as abnormal.

For these questions, assume that the register R0 contains a signed 32-bit integer (e.g., int32_t) with a value of -4 (0xFFFFFFFC)

R0 ASR #2: ___-1_____

R0 ASR #3: ___-1_____

R0 ASL #2: ___-16_____

R0 ASL #3: ___-32_____

R0 ASL #28: ___-1073741824_____

R0 ASL #29: ___-2147483648_____

R0 ASL #30: ___0_____

R0 LSR #2: ___1073741823_____ (abnormal LSR usage)

Assume that the register R0 contains a signed 32-bit integer (e.g., int32_t) with a value of 8 (0x00000008).

R0 ASR #3: ___1_____

R0 ASR #4: ___0_____

R0 ASL #3: ___64_____

R0 ASL #4: ___128_____

R0 LSR #3: ___1_____ (abnormal LSR usage)

abnormal to use LSR for SIGNED numbers

3. For each of these C functions, specify the ARM7 register(s) in which each argument is passed and result is returned.

a. uint32_t fn4(uint16_t a, uint32_t b, int8_t c, uint32_t d)

a is passed in: ___R0___

b is passed in: ___R1___

c is passed in: ___R2___

d is passed in: ___R3___

the result is returned in: ___R0___

b. uint64_t fn2(uint64_t a, uint64_t b)

a is passed in: ___R1:R0___

b is passed in: ___R3:R2___

the result is returned in: ___R1:R0___

4. Write 1-5 line assembly functions that implement the following C functions:

a. uint32_t subU32(uint32_t x, uint32_t y) // returns x-y

b. int32_t subS32(int32_t x, int32_t y) // returns x-y

c. uint64_t addU32_U64(uint32_t x, uint32_t y) // returns x+y

d. uint64_t addU64(uint64_t x, uint64_t y) // returns x+y

e. int32_t shiftRightS32 (int32_t x, uint8_t p) // returns x >> p = x*2^(-p) for p = 0..31

f. uint32_t shiftRightU32 (uint32_t x, uint8_t p) // returns x >> p = x*2^(-p) for p = 0..31

g. int32_t shiftLeftS32 (int32_t x, uint8_t p) // returns x << p = x*2^p for p = 0..31