Page 568 Exercises 3.1, 3.2, 3.4, 3.6, 3.7, 3.8, 3.14, 3.15, 3.16, 3.20, 3.21

3.1 [5] <§3.2> What is 5ED4 2 07A4 when these values represent unsigned 16-bit hexadecimal numbers? The result should be written in hexadecimal. Show your work.

- Step 1: convert both hexadecimal numbers to decimal.
  - 5ED4 (hex) = 24276 (decimal)
  - 07A4 (hex) = 1956 (decimal)
- Step 2: perform the subtraction in decimal.
  - 24276 (decimal) - 1956 (decimal) = 22320 (decimal)
- Step 3: convert the result back to hexadecimal.
  - 22320 (decimal) = 57D0 (hexadecimal).

3.2 [5] <§3.2> What is 5ED4 2 07A4 when these values represent signed 16-bit hexadecimal numbers stored in sign-magnitude format? The result should be written in hexadecimal. Show your work.

5ED4 2 07A4

E = 14

A = 10

D = 13

2 = SUB Instruction

Therefore,

5ED4 - 07A4 = (5-0) (E-7) (D-A) (4-4)

= 5 7 3 0

= 5730 (HEXADECIMAL).

3.4 [5] <§3.2> What is 4365 − 3412 when these values represent unsigned 12-bit octal numbers? The result should be written in octal. Show your work.

First we convert both values from OCTAL to DECIMAL.

4365 = 100 011 110 101

Sub

3412 = 011 100 001 010

0753 = 000 111 101 011 (OCT)

3.6 [5] <§3.2> Assume 185 and 122 are unsigned 8-bit

decimal integers. Calculate 185 – 122. Is there overflow, underflow, or neither?

185 = 1011 1001

Sub

122 = 0111 1010

63 = 1111 11

Since, 63 is within range of (0 to 256).

Therefore, there's no overflow or underflow.

3.7 [5] <§3.2> Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 + 122. Is there overflow, underflow, or neither?

+185 = 0 1011 1001

+

+122 = 0 0111 1010

=

307 = 0000 0001 0011 0011 (16 bit)

51   =   0011 0011 (8 bit)

Therefore, there was an overflow.

3.8 [5] <§3.2> Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 2 122. Is there overflow, underflow, or neither?

185 = 0 1011 1001

2   Means subtract

122 = 0 0111 1010

We get 2's complement of 122:

122 = 1 1000 0101

     +   1

    = 1 1000 0110

 185 = 10111001

-122 = 110000110

       ____

1 011110111

3.14 [10] <§3.3> Calculate the time necessary to perform a multiply using the approach given in Figures 3.3 and 3.5 if an integer is 8 bits wide and each step of the operation takes 4 time units. Assume that in step 1a an addition is always performed —either the multiplicand will be added, or a zero will be. Also assume that the registers have already been initialized (you are just counting how long it takes to do the multiplication loop itself). If this is being done in hardware, the shifts of the multiplicand and multiplier can be done simultaneously. If this is being done in software, they will have to be done one after the other. Solve for each case.

Case 1 (Hardware):

- Integer is 8-bits
- And 2 steps are needed per operation
- Each step = 4 TU
- 8 x 2 x 4 = 64 TU

Case 2 (software):

- Integer is 8-bits
- We have 3 steps
- Each step is 4 TU
- 8 x 3 x 4 = 96 TU

3.15 [10] <§3.3> Calculate the time necessary to perform a multiply using the approach described in the text (31 adders stacked vertically) if an integer is 8 bits wide and an

adder takes 4 time units.

- Integer = 8-bits
- 8 operations (32/4)
- 4 TU
- 8 x 8 x 4 = 256 TU

3.16 [20] <§3.3> Calculate the time necessary to perform a multiply using the approach given in Figure 3.7 if an integer is 8 bits wide and an adder takes 4 time units.

- Integer is 8-bits.
- Each operation takes 4 TU.
- Since the operations are done in series of 3: 8 x 3 x 4 = 96 TU

3.20 [5] <§3.5> What decimal number does the bit pattern 0×0C000000 represent if it is a two's complement integer? An unsigned integer?

0x0C000000 = 0000 1100 0000 0000 0000 0000 0000 0000

= 2^27 + 2^26 = 201326592

3.21 [10] <§3.5> If the bit pattern 0×0C000000 is placed into the Instruction Register, what MIPS instruction will be executed?

opcode 000011

target 00000000000000000000000000

This operation is jump and link (JAL)