Nathan Owen

Part 2 -- Written Assignment

1. Convert the following two decimal numbers into 8-bit 2's complement codes.

125

125/2 = 62.5 1
62/2 = 31 0
31/2 = 15.5 1
15/2 = 7.5 1
7/2 = 3.5 1
3/2 = 1.5 1
1/2 = .5 1

01111101

-125

01111101 -> 10000010 -> +1 -> 10000011

2. Identify the decimal numbers for the following two 8-bit 2's complement codes.

0101 1101
64 16 8 4 1

Signed and unsigned

64 + 16 + 8 + 4 + 1 = 93

1100 1101
128 64 8 4 1

11001101 -> -1 -> 11001100 -> 00110011 -> 32 16 2 1

Signed: -51

Unsigned: 241

3. Do 8-bit unsigned binary addition for 45 + 242. Do not worry that you have no space for the topmost carry bit (simply discard it as if it did not occur -- we will learn how CPU signal this later).

Show your work.

```
  1 1
  00101101
+ 11110010
-----------
  00011111
```

00011111 = 31

45/2 = 22.5 1
22/2 = 11 0
11/2 = 5.5 1
5/2 = 2.5 1
2/2 = 1 0
1/2 = .5 1

45 = 00101101

242/2 = 121 0
121/2 = 60.5 1
60/2 = 30 0
30/2 = 15 0
15/2 = 7.5 1
7/2 = 3.5 1
3/2 = 1.5 1
1/2 = .5 1

242 = 11110010

4. Do 8-bit 2's complement addition for 45 + (-14). Note 2's complement addition is done as if both numbers were unsigned. So there is no real difference between 2's complement addition and unsigned addition.

Show your work and confirm the result is correct. Briefly state what you learn here.

45 = 00101101

14/2 = 7 0
7/2 = 3.5 1
3/2 = 1.5 1
1/2 = .5 1

14 = 00001110 -14 = 11110010

00101101
+ 11110010
00011111

00011111 = 31

This problem demonstrates that 2's complement eloquently handles positive and negative integer addition. Regular addition between a positive and negative value in two's complement will always be correct.

5. Do the following math the C way:

5 * 4 =

5 = 00000101

5 * 4 → 5 * 2^2 → 5 << 2 → 00000101 << 2 = 00010100 → 00010100 = 20

18 / 4 =

18 = 00010010

18/4 → 18 * 2^{-2} → 18 >> 2 → 00010010 >> 2 = 00000100 = 4

Then show how the following C operations are done in binary and the results in decimal?

5 << 2 :

00000101 << 2 = 00010100 = 20

18 >> 2 :

00010010 >> 2 = 00000100 = 4

What do you learn from this exercise?

I Learned that C is capable of multiplication and division by powers of two using only bit shifting to the left and to the right.