CS 271Computer Architecture and Assembly Language

Self-Check for Lecture#7

Solutions

1. Add 8-bit binary. Show your work (carry bits, etc.) Check your work by converting all three numbers to decimal.

```
  11111
  00010111 =  23
+ 01011101 =  93
  01110100 = 116   ✓
```

2. Subtract 8-bit binary. Show your work (borrow bits, etc.) Check your work by converting all three numbers to decimal.

```
   111
  01110011 = 115
- 01011101 =  93
  00010110 =  22   ✓
```

3. Given the following decimal multiplication problem:

```
  2013
x  512
```

   Suppose that we are using 32-bit integers. Will the result cause overflow?
   (Note: You should be able to answer the question without doing the multiplication)

   2013 is less than 211, so it can be represented in less than 12 bits.512 is equal to 29, so multiplying by 9 would shift 9 places to the left, so the product can be represented in less than 21 bits. Therefore, no overflow.

4. Show the hexadecimal "endian" form of the 32-bit representation of 24685(decimal).

   Convert to hexadecimal 0x606D or 606Dh

   A. Big-endian: 00 00 60 6D

   B. Little-endian: 6D 60 00 00

5. Show the IEEE Standard 754 single-precision binary (32-bit) representation of the floating-point number 23.45. Indicate the three parts of the representation.

23.45 is approximately 10111.01110011001100 ...
or 1.011101110011001100 ... x 2^4

0	10000011	01110111001100110011001
sign(+)	exponent (4 + 127)	mantissa (normalized)

6. Convert single-precision floating-point hexadecimal 42E48000 to decimal floating-point.

0x42E48000 = 0100 0010 1110 0100 1000 0000 0000 0000 binary

0	10000101	11001001000000000000000
Sign	biased exponent (133)	mantissa (normalized)
0	133-127 = 6	1.11001001000000000000000
+	un-biased exponent	mantissa (un-normalized)

Move radix point 6 places to the right: +1110010.01 binary = 114.25 decimal