CSC 230 Assignment 1

Fall 2015

This assignment has two parts:

1) **Written Part:** Complete the Assignment 1 problem set on connex (in the Tests & Quizzes section). The problem set is to be completed online and only requires you to supply the final answer. You are required to answer these questions without electronic tools. Treat these questions as if they were conducted during an exam. **Complete it on conneX by September 24, 2015.**

2) **Programming Part: Submit only your code (no executables) file on conneX. Be sure to include your name and student number. Submit it by October 1, 2015 on conneX**

Assignment 1: Part 1 - Written [23]

Question 1 [4]

a) A processor uses 24 bits for its memory addressing. How many possible distinct locations the computer can address?

2^{24} = 16777216 addresses

b) Let us consider the location $(7234)_{10}$. How do you represent this address in binary (i.e., base 2)?

$(7234)_{10}$ /2 = 3617 remainder 0
3617 / 2 = 1808 remainder 1
1808 / 2 = 904 remainder 0
904 / 2 = 452 remainder 0
452 / 2 = 226 remainder 0
226 / 2 = 113 remainder 0
113 / 2 = 56 remainder 1
56 / 2 = 28 remainder 0
28 / 2 = 14 remainder 0
14 / 2 = 7 remainder 0
7 / 2 = 3 remainder 1
3 / 2 = 1 remainder 1
1 / 2 = 0 remainder 1

Thus, $7234_{10} = 1110001000010_2$

c) Group the above address in bits of four (Hex) and represent the same address in Hexadecimal notation.

$1110001100010_2 = 1\ 1100\ 0110\ 0010_2 = 1C42_{16}$

d) If each memory location stores 16 bits of data, what is the maximum range of 2's complement integers that can be used?

Smallest value: $-2^{15} = -32768$. Largest value: $2^{15}-1 = 32767$

Question 2 [9]

a) Convert the following base 10 numerals to base 2 by repeated division algorithm:
   I. 104
   II. 514

104 / 2 = 52 remainder 0
52 / 2 = 26 remainder 0
26 / 2 = 13 remainder 0
13 / 2 = 6 remainder 1
6 / 2 = 3 remainder 0
3 / 2 = 1 remainder 1
1 / 2 = 0 remainder 1 Thus, $104_{10} = 1101000_2$

514 / 2 = 257 remainder 0
257 / 2 = 128 remainder 1
128 / 2 = 64 remainder 0
64 / 2 = 32 remainder 0
32 / 2 = 16 remainder 0
16 / 2 = 8 remainder 0
8 / 2 = 4 remainder 0
4 / 2 = 2 remainder 0
2 / 2 = 1 remainder 0
1 / 2 = 0 remainder 1 Thus, $514_{10} = 10\ 0000\ 0010_2$

b) Convert the following base 10 numerals to base 16 by repeated division algorithm:
   I. 16500
   II. 65536

16500 / 16 = 1031 remainder 4
1031 / 16 = 64 remainder 7
64 / 16 = 4 remainder 0
4/ 16 = 0 remainder 4 Thus, $16500_{10} = 4074_{16}$

65536 / 16 = 4096 remainder 0
4096 / 16 = 256 remainder 0
256 / 16 = 16 remainder 0
16/ 16 = 1 remainder 0
1/ 16 = 0 remainder 1 Thus, $65536_{10} = 10000_{16}$

c) Convert the numeral 0x4C32CB directly to binary without first converting to decimal.

0x4C32CB = 0b 0100 1100 0011 0010 1100 1011

d) Convert 1520_{10} and -352_{10} to their 2's complement representation, using 16 bits for each.

1520_{10} = 0000 0101 1111 0000 (2's complement)

352_{10} = 0000 0001 0110 0000_2
invert: 1111 1110 1001 1111
add 1: 1111 1110 1010 0000 (2's complement)

e) Convert these octal values to binary and then to hexadecimal without converting them to decimal: 377_8, 1037_8.

377_8 = 011 111 111_2 = 0000 1111 1111_2 = $0FF_{16}$

1037_8 = 001 000 011 111_2 = 0010 0001 1111_2 = $21F_{16}$

Question 3 [5]

a) Using Horner's rule, convert the hexadecimal number 9A2 to base 10.

$9A2_{16} = 9 \times 16^2 + 10 \times 16^1 + 2$
$= 16\,(16(9) + 10) + 2$
$= 2466_{10}$

b) Given two binary numbers 0b10111010 and 0b01001001, perform a bit wise AND operation and show the result.

```
        0b10111010
AND     0b01001001
        0b00001000
```

c) Given two binary numbers 0b10111010 and 0b01001001, perform a bit wise OR operation and show the result.

```
        0b10111010
OR      0b01001001
        0b11111011
```

d) Given two binary numbers 0b10111010 and 0b01001001, perform a bit wise XOR operation and show the result.

```
        0b10111010
OR      0b01001001
        0b11110011
```

e) Given a Hex number 0xA9BC, perform a bit wise NOT operation (complement) and show the result in Hex.

0xA9BC = 0b 1010 1001 1011 1100

$\neq$ 0b 0101 0110 0100 0011 = 0x 5643

Question 4 [5]

a) Given a byte, what mask and operation would you use to keep the upper nibble?

00001111

b) Given a byte, what mask and operation would you use to clear the lower nibble?

11110000

c) One can perform shift operations on binary data, i.e., binary data can be shifted right or left. A '0' is brought to fill vacated positions and the bit that is shifted out is discarded. Let us take a four bit example 0111. If we shift it left once, the result will be 1110 and if we shift right once the result will be 0011. Based on this example, shift the following hex number 0x18 to left and show the result.

0x18 = 0b00011000 $\Leftarrow$ 0b00110000 = ox30

d) Take your answer from (c) and shift it left once more. What is the result?

0b00110000 ⇐ 0b01100000 = 0x60

e) What is the effect of shifting left and right on the resultant value?

Shifting left multiplies by 2; shifting right divides by 2

Assignment 1: Part 2 Programming [30]

Please note for all programing assignments, ***you must include your name and student number.*** Make sure the submitted work is yours and not someone else.

Question 5 [7]

(Page 68, #3) Write an assembly language program to add three numbers together. The program must begin by loading the three numbers into three distinct registers. The sum is to be calculated in register zero using ADD instructions. The CLR instruction may need to be used. Assemble and test your program in the AVR Studio. Try several different numbers keeping mind that the sum cannot exceed 255 (if it is to be correct.)

Please submit your code for this question as “a1_question5.asm” on conneX.

Question 6 [20]

To be added in the near future. . . .