Homework 1: Number Systems
CPE221

Instructor: Rahul Bhadani

Due: January 17, 2025, 11:59 PM
100 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

Upload a .pdf on Canvas with the format {firstname.lastname}_CPE221_hw01.pdf. For example, if your name is Sam Wells, your file name should be sam.wells_CPE221_hw01.pdf. If there is a programming assignment, then you should include your source code along with your PDF files in a zip file {firstname.lastname}_CPE221_hw01.zip. Your submission must contain your name, and UAH Charger ID or the UAH email address. Please number your pages as well.

---

1 Register Transfer Language (10 points)

What is the effect of the following sequence of RTL instructions? Describe each one individually and state the overall effect of these operations.

(1) $[8] \leftarrow 10$

(2) $[9] \leftarrow 15$

(3) [10] $\leftarrow$ [8] $\times$ [9]

(4) [8] $\leftarrow$ [10] - [9]

(5) [9] $\leftarrow$ [[8]] + [[10]]

Answer

(1) Places value 10 in memory address 8.

(2) Places value 15 in memory address 9.

(3) Places $10 \times 9 = 90$ in the memory address 10.

(4) Places $90 - 15 = 75$ in memory address 8.

(5) [[8]] means address value at memory address 75 (the previous step stored 75 at memory address 8). [[10]] means value at memory address 90(step 3). Hence [[8]] + [[10]] adds values at memory address 75 with values at memory address 90 and stores in memory address 9.

2 Range (5 points)

What range of numbers can be represented in 7 bits? Provide a range for signed and unsigned numbers.

Answer

In unsigned numbers, the range is from 0 to $2^7 - 1$ or from 0 to 127.
In signed numbers, the range is from -2^{7-1} to $2^{7-1} - 1$ or from -64 to 63.

3 2s Complement (5 points)

Represent 65000 as signed 17-bits and -4473 as signed 16-bit numbers (2s complement representation).

(**Note to TA:** Originally, I had asked for 16-bit representation, but students discussed in the class that 65000 cannot be represented using signed 16-bits, so I asked them to use 17 bits).

Answer

Since 65000 is a positive number, it can be represented using 16 bits. However, if we consider signed bit representation when a variable storing 65000 also represents a negative number, then we require 17-bit.

(1) 17-bit representation of 65000:

65000 = 0 1111 1101 1110 1000

(2) 16-bit representation of 65000 (this is a case of unsigned representation):

65000 = 1111 1101 1110 1000

(3) (This is not required by students) 17-bit representation of -4473:

4473 = 1 0001 0111 1001
= 0 0001 0001 0111 1001
Flip the bits
= 1 1110 1110 1000 0110
Add 1, and it gives 2's complement representation of -4473
-4473 = 1 1110 1110 1000 0111

(4) 16-bit representation of -4473:

4473 = 1 0001 0111 1001
= 0001 0001 0111 1001
Flip the bits
= 1110 1110 1000 0110
Add 1, and it gives 2's complement representation of -4473
-4473 = 1110 1110 1000 0111

4 Decimal to Unsigned Binary (10 points)

Convert the decimal number 87433 to unsigned binary in two ways: (a) convert directly to binary; (b) convert first to hexadecimal and then from hexadecimal to binary. Which method is faster?

Answer

To convert the decimal number 87433 to binary, we repeatedly divide the number by 2 and keep track of the remainder:

Step	**Division**	**Remainder**
1	87433 / 2 = 43716	1
2	43716 / 2 = 21858	0
3	21858 / 2 = 10929	0
4	10929 / 2 = 5464	1
5	5464 / 2 = 2732	0
6	2732 / 2 = 1366	0
7	1366 / 2 = 683	0
8	683 / 2 = 341	1
9	341 / 2 = 170	1
10	170 / 2 = 85	0
11	85 / 2 = 42	1
12	42 / 2 = 21	0
13	21 / 2 = 10	1
14	10 / 2 = 5	0
15	5 / 2 = 2	1
16	2 / 2 = 1	0
17	1 / 2 = 0	1

Reading the remainders from bottom to top,
Binary: 1 0101 0101 1000 1001

(b) Convert to Hexadecimal and Then to Binary

First, convert 87433 to hexadecimal:

Step	**Division**	**Remainder**
1	87433 / 16 = 5464	9
2	5464 / 16 = 341	8
3	341 / 16 = 21	5
4	21 / 16 = 1	5
5	1 / 16 = 0	1

Hexadecimal: 0x15589

Now, convert each hexadecimal digit to binary (that only requires a lookup table):

- $1 \rightarrow 0001$
- $5 \rightarrow 0101$
- $5 \rightarrow 0101$
- $8 \rightarrow 1000$
- $9 \rightarrow 1001$

Thus, the binary representation of 87433 is:
0001 0101 0101 1000 1001 or 1 0101 0101 1000 1001 (after removing leading three 0s).

Number of operations required:

- **Direct Binary Conversion** requires 17 divisions. The division is a complex operation.

- **Hexadecimal Conversion** is quicker because it requires 5 divisions and 5 lookups. Lookups are very fast compared to divisions.

5 Decimal to Binary (20 points)

Convert decimal +1943 and +12437 to binary, using signed-2's complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+1943) + (-12437), (-1943) + (+12437), and (-1943) + (-12437). Convert the answers back to decimals and verify that they are correct.

Answer

$$\begin{aligned} 2^{n-1} - 1 &\geq 12437 \\ 2^{n-1} &\geq 12438 \\ n - 1 &\geq \log_2(12438) \\ n - 1 &\geq 13.6 \\ n &\geq 14.6 \\ n &= 15 \end{aligned} \tag{1}$$

12437 can be represented using 15 bits in signed 2's complement notation. Hence, we need at least 15 bits.

Step	**Division**	**Remainder**
1	12437 / 2 = 6218	1
2	6218/2 = 3109	0
3	3109/2 = 1554	1
4	1554/2 = 777	0
5	777/2 = 388	1
6	388/2 = 194	0
7	194/2 = 97	0
8	97/2 = 48	1
9	48/2 = 24	0
10	24/2 = 12	0
11	12/2 = 6	0
12	6/2 = 3	0
13	3/2 = 1	1
14	1/2 = 0	1

Thus $(12437)_{10} = (011\ 0000\ 1001\ 0101)_2$ in 15 bits.
Flip bits $(100\ 1111\ 0110\ 1010)$ and add 1. Thus,
$(-12437)_{10} = (100\ 1111\ 0110\ 1011)_2$

Similarly, 1943_{10} = 000 0111 1001 0111_2.
and
-1943_{10} = 111 1000 0110 1001_2.
From decimal calculation, we know that
(+1943) + (-12437) = -10494
(-1943) + (+12437) = 10494
and
(-1943) + (-124337) = -14380
Verifying in 2s complement calculation

+1943	000	0111	1001	0111
-12437	100	1111	0110	1011
-10494	101	0111	0000	0010

Here, 101 0111 0000 0010 is a 2s complement representation. Flip bits:
010 1000 1111 1101, add 1: 010 1000 1111 1110 which is binary representation of 10494.
Similarly,

-1943	111	1000	0110	1001
+12437	011	0000	1001	0101
+10494	1010	1000	1111	1110

and

-1943	111	1000	0110	1001
-12437	100	1111	0110	1011
-14380	1100	0111	1101	0100

6 Signed 2's Complement (10 points)

The following numbers are represented in a signed 2-s complement. What is their value in decimal?

(1) 1101 1110 0010

(2) 0101 1101 1010

(3) 0011 1110 1111

(4) 1111 1111 1111

Answer

(1) 1101 1110 0010

Flip bits: 0010 0001 1101

Add 1: 0010 0001 1110 which is binary for 542.

So, 1101 1110 0010 is binary for -542.

Alternatively, you can use the following method

1101 1110 0010 = $1 \times (-2)^{11} + 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = -542$

Notice that the highest power is of (-2).

The first option is good for implementation in logic circuits, while the second is a good option for implementation in computer programs.

(2) 0101 1101 1010

Its most significant bit (MSB) is 0, hence it is a positive number. We just need to convert that directly to decimal:

$0 \times (-2)^{11} + 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1498$

(3) 0011 1110 1111

Its most significant bit (MSB) is 0, hence it is a positive number. We just need to convert that directly to decimal:

$0 \times (-2)^{11} + 0 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1007$

(4) 1111 1111 1111

Its most significant bit (MSB) is 1, hence it is a negative number.

Flip bits: 0000 0000 0000, Add 1 = 0000 0000 0001, which is binary for 1. Hence 1111 1111 1111 is binary for -1 in 2's complement.

Alternatively,

$1 \times (-2)^{11} + 1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = -1$

7 Conversion (20 points)

Convert from one base to another:

(1) $(527)_{10} \rightarrow (?)_{16}$

(2) $(34F)_{16} \rightarrow (?)_{2}$

(3) $(11001001)_{2} \rightarrow (?)_{16}$

(4) $(66)_{10} \rightarrow (?)_{2}$

(5) $(33)_{8} \rightarrow (?)_{10}$

Show your work. For hexadecimal, you can assume that $A = 10, B = 11, C = 12, D = 13, E = 14, F = 15$.

Answer

(1) $(20F)_{16}$

(2) $(001101001111)_{2}$

(3) $(C9)_{16}$

(4) $(1000010)_{2}$

(5) $(27)_{10}$

8 Complements (20 points)

(1) If $X = 00110$, $Y = 11100$ are represented in 5-bit signed 2's complement form, then what's their sum $X + Y$ in 6-bit signed 2's complement representation? Verify your answer as well.

(2) What's the smallest integer that can be represented by an 8-bit number in 2's complement form? Show your work.

(3) What's 16-bit 2's complement representation for the decimal number -28?

Answer

(1) As X, and Y are 5-bit numbers, we first need to write them as 6-bit 2's complement. If written using 6 bits, their representation is $X = 000110, Y = 111100$.

Note that adding an extra 1 at MSB to a 2's complement representation of a negative number doesn't change its value.

Thus, adding X and Y =

$$\begin{aligned} X &= 000110 \\ Y &= 111100 \\ X + Y &= (1)000010 \end{aligned} \tag{2}$$

Since the answer must be in 6-bit, we must discard the carry-out 7th bit. Thus, the answer is 000010.

Verification

X = 00110 which is 6 in decimal. Y = 11100 which is -4 in decimal.

X + Y = 6 - 4 = 2.

Earlier, we got the answer as 000010 which is 2 in decimal, we arrived at the correct answer.

(2) In 2's complement, the range of the number represented by 8 bits is from -2^{n-1} to $2^{n-1} - 1$.

Hence, the smallest number is $-2^{8-1} = -2^7 = -128$.

(3)

Step	**Division**	**Remainder**
1	28 / 2 = 14	0
2	14/2 = 7	0
3	7/2 = 3	1
4	3/2 = 1	1
5	1/2 = 0	1

28 in binary is 1 1100.

In 16 bits, it would be 0000 0000 0001 1100.

Flip the bits: 1111 1111 1110 0011.

Add 1: 1111 1111 1110 0100

Hence, 16-bit 2's complement representation for the decimal number -28 is 1111 1111 1110 0100.

§ – § – §