

Seminar 0x00

Cristian Rusu

1 Purpose of the seminar

In this seminar, we will solve some problems involving:

- number systems and conversion between them;
- arithmetic operations on numbers;
- logical operations on numbers.

2 Exercises

1. Complete the following table with the missing representations of natural numbers:

Base: $B = 2$	$B = 4$	$B = 8$	$B = 10$	$B = 16$
				0xBEEF
			42	
1001 1100 1111 0011				
	22331			
		777		
				0xDEAF
1100 1010 0101 0011				
	3210			
		7235		
			2020	
				0x1111
				0xFFFF
1111 1111 0000 0000				
0000 0000 1111 1111				

2. Complete the following table with the missing representations of integers (16 bits):

Base: $B = 2$	$B = 4$	$B = 8$	$B = 10$	$B = 16$
				0xFEED
			-42	
0101 1100 1111 0011				
	22331			
		777		
				0xA1F1
1100 1010 0101 0011				
	3210			
		7007		
			-1337	
				0xF111
				0xFFFF
1111 1111 0000 0000				
1111 0000 1111 0000				

3. Perform the following arithmetic operations on binary numbers in two's complement representation (check your calculations using the decimal system):

$$\begin{array}{r|l} 0101\ 1100\ 1111\ 0011 & \\ 1111\ 1111\ 0000\ 0000 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1111\ 1111\ 1111\ 1111 & \\ 0000\ 0000\ 0000\ 0001 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0111\ 1111\ 1111\ 1111 & \\ 0000\ 0000\ 0000\ 0001 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0101\ 1100\ 1111\ 0011 & \\ 0111\ 0000\ 1111\ 0000 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1111\ 1111\ 1111\ 1111 & \\ 1000\ 0000\ 0000\ 0000 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1000\ 0000\ 0000\ 0000 & \\ 0000\ 0000\ 0000\ 0001 & + \\ \hline & \end{array}$$

In the following cases, the number of bits shown is the total number of bits allocated to represent the numbers (14 and 8 bits on the left, 12 and 8 bits on the right):

$$\begin{array}{r|l} 11\ 1000\ 0010\ 0101 & \\ 1011\ 1100 & + \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1100\ 1001\ 0010 & \\ 0100\ 1010 & + \\ \hline & \end{array}$$

4. Perform the following bitwise logical operations:

$$\begin{array}{r|l} 0101\ 1100\ 1111\ 0011 & \\ 0101\ 1100\ 1111\ 0011 & \text{AND} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1101\ 1100\ 1111\ 0011 & \\ 1101\ 1100\ 1111\ 0011 & \text{XOR} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0111\ 1111\ 1111\ 1111 & \\ 0000\ 0000\ 0000\ 0001 & \text{AND} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0101\ 1100\ 1111\ 0011 & \\ 0111\ 0000\ 1111\ 0000 & \text{OR} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1101\ 1001\ 0110\ 0001 & \\ 1111\ 1111\ 0000\ 0000 & \text{AND} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0101\ 1100\ 1111\ 0011 & \\ 0000\ 0000\ 1111\ 1111 & \text{OR} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 0000\ 0000\ 1111\ 1111 & \\ 0000\ 0001\ 0000\ 0000 & \text{AND} \\ \hline & \end{array}$$

$$\begin{array}{r|l} 1100\ 0110\ 1001\ 1110 & \\ 1001\ 1111\ 0110\ 1100 & \text{XOR} \\ 1100\ 0110\ 1001\ 1110 & \text{XOR} \\ \hline & \end{array}$$

5. Using what you learned in the lectures, answer the following short questions:

- (a) what is the largest decimal number (natural number) that can be represented using N bits?
- (b) what is the largest integer that can be represented (in two's complement) using N bits? what about the smallest?
- (c) let x be a natural number; how many bits do we need to represent x in the binary system?
- (d) if a number x is represented using k digits in the hexadecimal system, how many bits do we need to represent x in the binary system?
- (e) if a number x is represented using k bits in the binary system, how many digits do we need to represent x in the hexadecimal system?
- (f) if a number x is represented using k digits in the decimal system, how many bits do we need to represent x in the binary system?

6. We discussed the binary representation of a number in the lectures. We can extend the representation to the right to include fractional values as follows (*binary fixed-point*):

...	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	...
-----	-------	-------	-------	-------	-------	-------	-------	-------	--	----------	----------	----------	----------	----------	----------	----------	-----

- calculate the decimal values of the following numbers represented in the format above (the point lies between 2^0 and 2^{-1}):
 - (a) 101.101
 - (b) 111.001
 - (c) 1110.00111
 - (d) 1010.0000011
 - (e) 1111.0010011
 - calculate the binary fixed-point representations of the following numbers given in decimal format:
 - (a) 3.75
 - (b) 12.3125
 - (c) 3.078125
 - (d) 17.671875
 - (e) $\frac{2}{3}$
 - when do numbers have a finite representation in this system (with no repeating fractional digits)?
 - what does a negative number look like in such a representation system? Represent $\frac{5}{8}$ and $-\frac{5}{8}$ in the binary fixed-point system (use a single bit to represent the integer part, and consider both using a sign bit and two's complement representation).
7. We discussed in the lectures how to find the value of a number written in two's complement format: invert the bits and add 1. Prove that this procedure is correct.
8. We discussed in the lectures how to convert a number from base B to base B^p , $p > 1, p$ being a natural number. Prove that the method (grouping p consecutive terms) is correct.
9. Prove that $\lfloor \log_2 x \rfloor = i_{\max}$ where x is a number represented using N bits and $i_{\max} = \max \{i \mid b_i = 1, \forall i = 0, \dots, N-1\}$ where b_i denotes the i -th bit in the binary representation of the number x . For example: if $x = 00101110$ (46 in decimal) then $\lfloor \log_2 x \rfloor = 5$.

10. In 2014, the music video for the song “Gangnam Style” was approaching 2,147,483,647 views. The team at Youtube had to make some changes to accommodate this large number of views. Explain what happened: what is the significance of the number above, what was the risk, and what is the solution to the potential problem.
11. Appendix 1 contains two implementations of the binary search algorithm. Which implementation would you use?

Appendix 1

```
int binarySearch1(int arr[], int start, int end, int x)
{
    if (end >= start)
    {
        int mid = start + (end - start) / 2;

        if (arr[mid] == x)
            return mid;

        if (arr[mid] > x)
            return binarySearch1(arr, start, mid - 1, x);

        return binarySearch1(arr, mid + 1, end, x);
    }

    return -1;
}

int binarySearch2(int arr[], int start, int end, int x)
{
    if (end >= start)
    {
        int mid = (start + end) / 2;

        if (arr[mid] == x)
            return mid;

        if (arr[mid] > x)
            return binarySearch2(arr, start, mid - 1, x);

        return binarySearch2(arr, mid + 1, end, x);
    }

    return -1;
}
```