Introduction to Informatics

Piroska Biró

Course 06

MIDTERM

5th November 2014

Tuesday

10:00–12:00

F01

Revision

- conversion of the numbers (including repeating decimal)
- addition and subtraction of different base numbers
- representation of the numbers
  - fixed-point & floating point representation
- encoding characters, Unicode, UTF-8
- logical operations
- logical gates

Revision

- Convert the following different base representations to decimal form:
  - $3415.482_{(9}$
  - $AC34.F2_{(16}$

Revision

- Calculate the following arithmetical operations!

$$\begin{array}{rcccccccc} & 5 & 6 & 6 & 7 & 4 & 5 & 5 & (8 \\ + & 7 & 5 & 7 & 3 & 4 & 3 & 2 & (8 \\ \hline \end{array}$$

$$\begin{array}{rcccccccc} & 6 & F & 6 & 3 & D & B & 3 & (16 \\ + & 7 & E & 5 & 6 & 2 & A & 6 & (16 \\ \hline \end{array}$$

$$\begin{array}{rcccccc} & B & 4 & 7 & 6 & 9 & D & (16 \\ - & 9 & A & C & D & 3 & B & (16 \\ \hline \end{array}$$

$$\begin{array}{rcccccccccccc} & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & (2 \\ - & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & (2 \\ \hline \end{array}$$

Revision

- Convert the following ten base representation to **six** base form:
-
  - $13251.75_{(10}$
- Convert the following numbers to decimal system. (calculate separately the integer and the fraction part)
  - $2123.23'201'201'\ldots_{(5}$

Revision

- Convert the following binary representation to hexadecimal form, and the hexadecimal representation to binary form:
  - $11111101010110101010011_{(2}$
  - $75DA94C_{(16}$
- Convert the following binary representation to octal form, and the octal representation to binary form:
  - $11111001010110001001 1_{(2}$
  - $6234745_{(8}$

Revision

- Represent **-76** in **8 bits** with the following fixed-pointed methods.
  - sign-and-magnitude
  - 1's complement
  - 2's complement
  - excess 127
  - excess 128
- Convert the result to hexadecimal.

- Represent **+1895** in **16 bits** with the following fixed-pointed methods.
  - sign-and-magnitude
  - 1's complement
  - 2's complement
  - excess 2^{15} -1
  - excess 2^{15}
- Convert the result to hexadecimal.

Revision

- Which number was represented with the IEEE 754 floating point standard?
  - 0110 1111 0011 1010 1101 0000 0000 0000
- Represent the following decimal number in 32 bits using the IEEE 754 floating point standard. Convert the result to hexadecimal.
  - -10432.75

Revision

- Represent $+85.14_{(10}$ number in **octal** system.
  - starting with sign bit
  - the exponent will be 1 digit (3 bits), excess-4
  - the fraction part 4 digits

- Represent $-187.24_{(10}$ number in **hexadecimal** system.
  - starting with sign bit
  - the exponent will be 1 nibble (4 bits), excess-8
  - the fraction part 4 digits

Revision

- Give the Unicode value of the BMP plane's given character and the encoding of UTF-8 in the hexadecimal form.

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x2300	⌀	⌁	⌂	⌃	⌄	⌅	⌆	⌇	⌈	⌉	⌊	⌋	⌌	⌍	⌎	⌏
0x2310	⌐	⌑	⌒	⌓	⌔	⌕	⌖	⌗	⌘	⌙	⌚	⌛	⌜	⌝	⌞	⌟
0x2320	⌠	⌡	⌢	⌣	⌤	⌥	⌦	⌧	⌨	〈	〉	⌫	⌬	⌭	⌮	⌯
0x2330	⌰	⌱	⌲	⌳	⌴	⌵	⌶	⌷	⌸	⌹	⌺	⌻	⌼	⌽	⌾	⌿
0x2340	⍀	⍁	⍂	⍃	⍄	⍅	⍆	⍇	⍈	⍉	⍊	⍋	⍌	⍍	⍎	⍏
0x2350	⍐	⍑	⍒	⍓	⍔	⍕	⍖	⍗	⍘	⍙	⍚	⍛	⍜	⍝	⍞	⍟
0x2360	⍠	⍡	⍢	⍣	⍤	⍥	⍦	⍧	⍨	⍩	⍪	⍫	⍬	⍭	⍮	⍯

Revision

- Which characters were encoded in this hexadecimal UTF-8 form?

- E2 8F B0

Revision

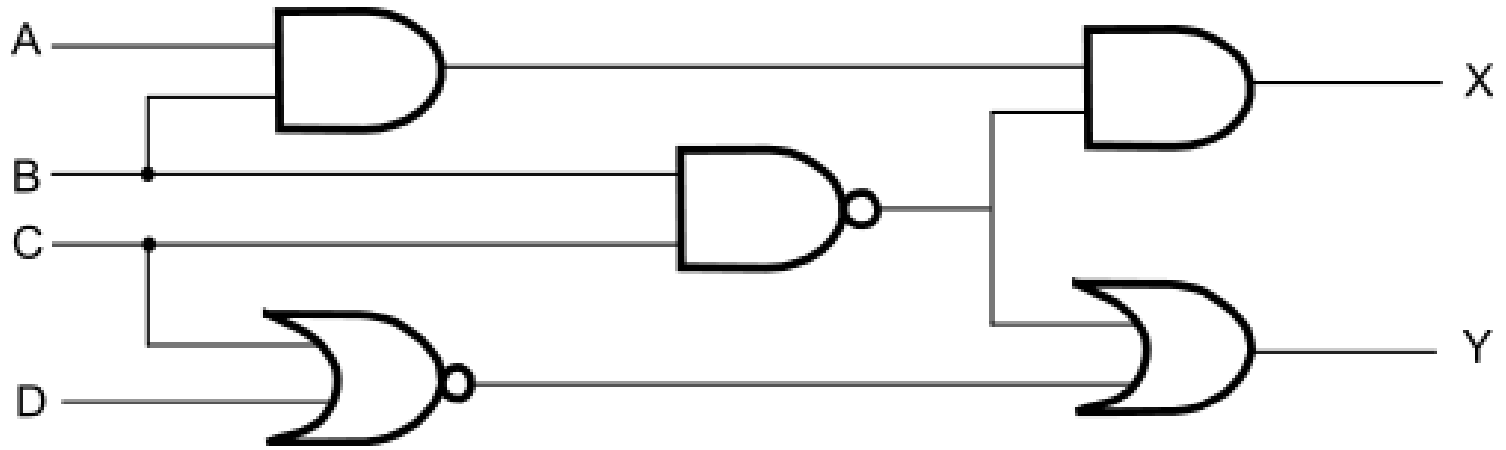
- Calculate the following logical operations.

1. NOT(10000110) NOR (10110111 AND 11001010)

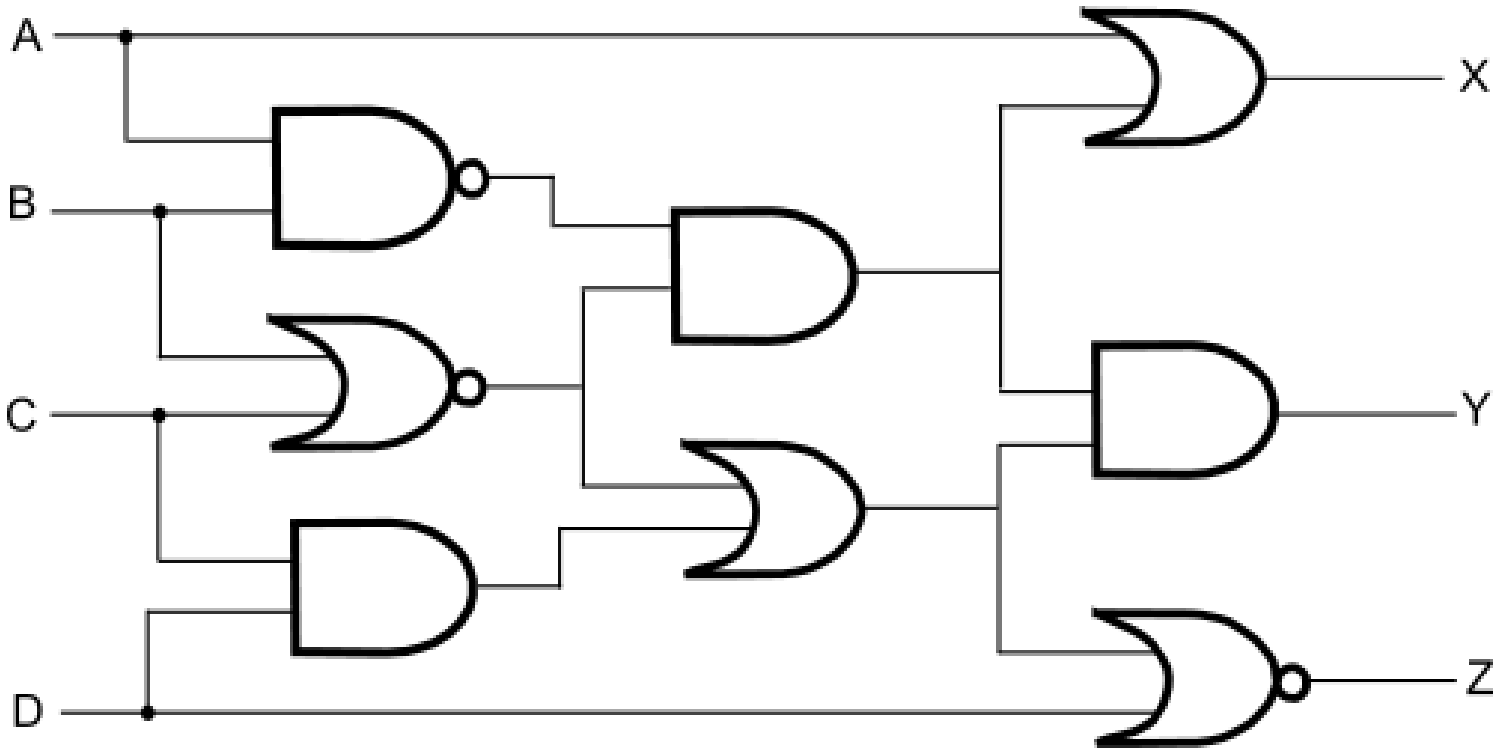
2. (10110101 XOR 11110011) OR (10001011 NAND 11001110)

Revision

- Describe the following circuit with logic expression. According to this give the mathematical equivalents. What is the value of expression, if A=B=1, C=D=0?



Logical gate



Inputs:

A=B=1

C=D=0

Outputs:

x, y, z