Санкт-Петербургский государственный университет информационных технологий,

механики и оптики

Факультет компьютерных технологий и управления

Кафедра информатики и прикладной математики

Лабораторная работа №2

По дисциплине «Организация ЭВМ и систем»

Выполнил студент группы P3317

Сорокин Юрий Борисович

Санкт-Петербург

2016

# Текст задания:

Разработать программу ввода и вывода целых чисел со знаком в С51 и А51 через порты и ввода дробных в С51 в формате

+\- а2 а1 а0

# Код программы на C51:

#include <reg51.h>

short TransformTo2D(short val)

{

char \* ptr;

char sotni, desyatki, edinici;

short result = 0;

char negative = 0;

if (val < 0){

negative = 1;

val \*= -1;

}

sotni = val / 100;

desyatki = (val / 10) % 10;

edinici = val % 10;

ptr = (char\*)&result;

ptr[0] = sotni;

if(negative != 0)

{

ptr[0] |= 0xf0;

}

ptr[1] = ( desyatki << 4 );

ptr[1] |= edinici;

return result;

}

short TransformFrom2D(unsigned char high, unsigned char low)

{

char negative = (high & 0xf0);

short result = 0;

result = (high & 0x0f) \* 100 +

((low & 0xf0) >> 4) \* 10 +

(low & 0x0f);

if(negative != 0) result \*= -1;

return result;

}

main()

{

unsigned char high = P1;

unsigned char low = P0;

short result = TransformFrom2D(high, low);

short result\_output = TransformTo2D(result);

char \* ptr = &result\_output;

P3 = \*ptr;

P2 = \*(ptr+1);

while(1);

}

# Код программы на A51:

XSEG AT 0

result: DS 2

CSEG AT 0

jmp START

input: DB 0xf3,0xE7

START:

clr a

// Transform from 2-d

// r0 - sign true/false

// r1 - low

// r2 - high

// r3 - result low

// r4 - result high

mov r2, P1 // high

mov r1, P0 // low

mov a, r2

anl a, #0x80

mov r0, a

//result = (high & 0x0f) \* 100 +

// ((low & 0xf0) >> 4) \* 10 +

// (low & 0x0f);

mov a, r2

anl a, #0x0F

mov b, #0x64

mul ab

mov r4, b // r4 = sotni

mov r3, a

mov a, r1

anl a, #0xF0

rr a

rr a

rr a

rr a

mov b, #0x0A

mul ab

add a, r3

mov r3, a // r3 = sotni + desyatki

mov a, r1

anl a, #0x0F

add a, r3

mov r3, a // r4 and r3 = number

// if(negative != 0) result \*= -1;

mov a, r4

orl a, r0

mov r4, a

// write to dataseg

mov DPTR, #result

mov a, r4

movx @DPTR, a

inc DPTR

mov a, r3

movx @DPTR, a

SECOND\_PART:

// Transform to 2-d

// r0 - sign

// r1 - low

// r2 - high

// r3 - result low

// r4 - result high

// r5 - sotney

// r6 - desyatkey

// r7 - edenecey

clr a

mov DPTR, #input

movc a, @a+DPTR

mov r2, a

mov a, #0x01

movc a, @a+DPTR

mov r1, a

// sign detection

mov a, r2

anl a, #0x80

mov r0, a // r0 = sign

// devisioning 1C8 XYZ

mov a, r2 // mov SX to A

anl a, #0x0F // 0X

rl a

rl a

rl a

rl a

mov r5, a // save X0

mov a, r1 // mov A YZ

anl a, #0xF0 // Y0

rr a

rr a

rr a

rr a

add a, r5 // 1C

mov b, #0x0A

div ab

mov r7, b // 8 to r7

mov r5, a // 2 to r5

mov a, b // A = 08

rl a

rl a

rl a

rl a

mov b, r1 // C8

anl b, #0x0F // 08

add a, b // 88

mov b, #0x0A

div ab // A = D B = 6

mov r7, b

mov b, a // save 0D to B

mov a, r5

rl a

rl a

rl a

rl a // 20

add a, b // 2D

mov b, #0x0A

div ab

mov r5, a

mov r6, b

// constructing the output

mov a, r5

orl a, r0

mov r4, a

mov a, r6

rl a

rl a

rl a

rl a

orl a, r7

mov r3, a

mov P2, r3

mov P3, r4

END