Санкт-Петербургский государственный университет информационных технологий,

механики и оптики

Факультет компьютерных технологий и управления

Кафедра информатики и прикладной математики

Лабораторная работа №3

По дисциплине «Организация ЭВМ и систем»

Выполнил студент группы P3317

Сорокин Юрий Борисович

Санкт-Петербург

2016

# Текст задания:

Совместить программы последовательного умножения и деления выполнить программы в С51 и в А51 с размещением данных в регистрах.

# Код программы на C51:

#include <reg51.h>

void div(){

unsigned int A;

long S;

char i;

S=P1;

A=P2<<8 ;

for (i=0 ; i<8; i++ ){

S = S<<1;

if ((S-A)>=0){

S = S-A+1;

}

}

P3 = S; //Write result of DIV to P3

}

void mul(){

unsigned short handl = P1; //Mult\_1

unsigned char val2 = P2; //Mult\_2

unsigned short result = 0; //Result

char ost;

int i = 0;

for (i = 0; i< 8; i++){

if (0x01&val2){ //Even or not (2)?

result += handl;

}

handl<<=1;

val2>>=1;

}

ost = result;

P0 = ost;

P3 = result>>8;

}

main()

{

// P1 INPUT

// P2 INPUT

// P3 OUTPUT

// P0 OUTPUT

mul();

// P2 OUTPUT;

div();

while(1);

}

# Код программы на А51:

XSEG AT 0

result: DS 2

CSEG AT 0

jmp START

input: DB 0x83,0xE7

START:

mov P0, #0

clr a

//---------MUL------------

// r0 - lowHandl || P1

// r1 - highHandl

// r2 - val2 || P2

// r3 - lowRes

// r4 - highRes

// r5 - i

mov r0, P1

mov r1, #0

mov r2, P2

mov r3, #0

mov r4, #0

mov r5, #8

MUL\_LOOP:

// check for loop end

mov a, r5

jz MUL\_LOOP\_END

// if (0x01&val2)

mov a, r2

anl a, #0x01

jz MUL\_ROL\_DATA

// result += handl;

mov a, r3

mov b, r0

add a, b

mov r3, a

mov a, r4

mov b, r1

addc a, b

mov r4, a

MUL\_ROL\_DATA:

// val2>>=1;

mov a, r2

rr a

mov r2, a

// handl<<=1;

mov a, r0

rlc a

mov r0, a

mov a, r1

rl a

mov b, #0

addc a, b

mov r1, a

// i++

dec r5

jmp MUL\_LOOP

MUL\_LOOP\_END:

mov P3, r4

mov P0, r3

//---------DIV---------------

clr a

// r0 - lowS || P1

// r1 - highS

// r2 - lowA || P2

// r3 - highA

// r5 - i

mov r0, P1

mov r1, #0

// A=P3<<8

mov r2, #0

mov r3, P2

// i = 8

mov r5, #8

DIV\_LOOP:

mov a, r5

//i=0?

jz DIV\_LOOP\_END

clr c

mov a, r0

rlc a

mov r0, a

mov a, r1

rl a

addc a, #0

mov r1, a

// if (S-A >= 0){

add a, #0

mov b, r3

subb a, b

jc DIV\_SKIP\_MATH

mov r1, a

mov a,r0

mov b,r2

subb a,b

inc a

mov r0, a

DIV\_SKIP\_MATH:

dec r5

jmp DIV\_LOOP

DIV\_LOOP\_END:

mov P3, r0

//mov P0, r0

mov P0, #0

jmp START

END