Университет ИТМО

**Лабораторная работа №6**

Выполнил:

студент III курса группы 3125

Припадчев Артём

Проверит:

Тропченко А.А.

Санкт-Петербург

2014

**Битовые данные:** Алгоритмы логического управления, программные модели конечных автоматов используют битовое кодирование событий и состояний. возможность выполнения операций с битами – уникальная особенность архитектуры MCS51.

## Доступ к битам в С51

#include <reg51.h>

unsigned char bdata value = 0;

unsigned char bdata temp;

sbit x1 = value^0;

sbit x2 = value^1;

sbit y1 = value^2;

sbit y2 = value^3;

sbit z = temp^0;

void main()

{

unsigned char counter, counter2;

value = 0;

for(counter = 2; counter > 0; counter--)

{

temp = 0;

for(counter2 = 8; counter2 > 0; value++, counter2--)

{

temp <<= 1;

z= (!y1&x2)|(y2&(!x1|!x2));

}

if(counter == 2)P0 = temp;

else P1 = temp;

}

}

**Листинг программы**

0000 E4 CLR A

0001 F500 R MOV value,A

;---- Variable 'counter' assigned to Register 'R7' ----

0003 7F02 MOV R7,#02H

0005 E4 CLR A

0006 F500 R MOV temp,A

; SOURCE LINE # 18

;---- Variable 'counter2' assigned to Register 'R6' ----

0008 7E08 MOV R6,#08H

000A E500 R MOV A,temp

000C 25E0 ADD A,ACC

000E F500 R MOV temp,A

; SOURCE LINE # 21

0010 A200 R MOV C,x1

0012 B3 CPL C

0013 A000 R ORL C,/x2

0015 8200 R ANL C,y2

0017 92F7 MOV B.7,C

0019 A200 R MOV C,x2

001B B000 R ANL C,/y1

001D 72F7 ORL C,B.7

001F 9200 R MOV z,C

0021 0500 R INC value

0023 DEE5 DJNZ R6,?C0004

0025 BF0205 CJNE R7,#02H,?C0007

0028 850080 R MOV P0,temp

002B 8003 SJMP ?C0003

002D 850090 R MOV P1,temp

0030 DFD3 DJNZ R7,?C0001

0032 22 RET

CODE SIZE = 51

1. **Адресация к битам в Ассемблере**

cseg at 0

jmp start

codeseg segment code

rseg codeseg

value equ 20h

z bit acc.0

x1 bit value.0

x2 bit value.1

y1 bit value.2

y2 bit value.3

start:

mov r6, #2

mov value, #0

portCounterLoop:

clr a

mov r7, #8

mainLoop:

rl a

mov c,x1

cpl c

orl c,/x2

anl c,y2

mov b.7,c

mov c,x2

anl c,/y1

orl c,b.7

mov z,c

inc value

djnz r7, mainLoop

endMainLoop:

cjne r6, #2, storeP1

mov p0, a

djnz r6, portCounterLoop

storeP1:

mov p1, a

end

CODE SIZE = 40