Университет ИТМО  
Кафедра вычислительной техники

Организация ЭВМ и систем

**Лабораторная работа №5**

*Вариант 9*

Выполнил:

Студент группы P3317

Сорокин Ю.Б.

Преподаватель

Скорубский В.И.

Санкт-Петербург

2016 г.

**Задание**

1). В С51 и A51 вывести вектор значений функции z=(x1y1 v /x2y2)(/x2 v y1) в порт.

2). Построить таблицу истинности из 8 наборов данных, где каждый из предикатов {(aa<bb), (cc!=dd), bb} true или false

3). Для заданного варианта функции выполнить ее преобразование в ОПЗФ и вывести таблицу истинности в порт. Значения False(0) и True(1) кодируются при вычислениях в байтах.

**1).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2 | y1 | y2 | z=(x1&y1 | !x2&y2)&(!x2 | y1) |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Код программы С51:**

#include <reg51.h>

char bdata mem;

bit r1;

sbit x1=mem^0;

sbit x2=mem^1;

sbit y1=mem^2;

sbit y2=mem^3;

sbit p1=P1^0;

sbit p2=P2^0;

main()

{

for(mem=0;mem<0x10;mem++)

{

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

if(mem<=0x07){

P1<<=1;

p1=r;

}else{

P2<<=1;

p2=r;

}

}

while(1);

}

**Код программы A51:**

bseg at 0x10

x1 bit ACC.0

x2 bit ACC.1

y1 bit ACC.2

y2 bit ACC.3

cseg at 0x00

jmp START

START:

// r0 - mem

// r1 - res

mov r0, #0

mov P1, #0

mov P2, #0

LOOP:

mov a, r0

mov b, #0x10

cjne a, b, LOOP\_CONTINUE

jmp LOOP\_END

LOOP\_CONTINUE:

// r = ((x1&y1)|((!x2)&y2))&((!x2)|y1)

mov c, x2

cpl c

orl c, y1

mov ACC.4, c // (!x2)|y1)

mov c, x1

anl c, y1

mov ACC.5, c // x1&y1

mov c, x2

cpl c

anl c, y2 //(!x2)&y2

orl c, ACC.5

anl c, ACC.4

mov ACC.6, c

mov b, a

mov a, r0

subb a,#0x07

jnc SECOND\_BAIT

mov a, b

mov c, ACC.6

mov a, P1

rl a

mov P1, a

mov P1.0, c

jmp INC\_R0

SECOND\_BAIT:

mov a, b

mov c, ACC.6

mov a, P2

rl a

mov P2, a

mov P2.0, c

INC\_R0:

inc r0

jmp LOOP

LOOP\_END:

END

**2).**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | a&c | b!=c | d>=c | (a&c)&(b!=c)||(d>=c) |
| 1 | 2 | 2 | 1 | 0 | 0 | 0 | 0 |
| 1 | 2 | 2 | 2 | 0 | 0 | 1 | 1 |
| 1 | 1 | 2 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 2 | 2 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 |
| 1 | 2 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |

**Код программы С51:**

#include <reg51.h>

char a,b,c,d,S;

main()

{

a = 1;

b = 2;

c = 2;

d = 2;

S = (a&c)&(b!=c)||(d>=c);

while(1);

}

**Код программы A51:**

Dseg at 0x10

aa: ds 1

bb: ds 1

cc: ds 1

dd: ds 1

S: ds 1

Cseg at 0

mov aa, #1

mov bb, #1

mov cc, #1

mov dd, #1

clr c

mov a, dd

xrl a, #0x80

mov r0, a

mov a, cc

xrl a, #0x80

subb a, r0 //cc-dd

jc RES\_1 //if (dd>=cc) goto res=1

//(a&c)&(b!=c)||(d>=c)

clr c

mov a, bb

xrl a, #0x80

mov r0, a

mov a, cc

xrl a, #0x80

subb a,r0

jz RES\_0 //if (bb != cc) goto res=0

clr c

mov a, aa

xrl a, #0x80

mov r0, a

mov a, cc

xrl a, #0x80

anl a, r0

jz RES\_0 //if (aa & cc) goto M0

jmp RES\_1

RES\_0:

mov S, #0x00

jmp EXIT

RES\_1:

mov S, #0x01

EXIT:

END

**3).**

**Код программы C51:**

z=(x1 & y1 | ¬x2 & y2) & (¬x2 | y1)= ¬ (¬ (x1 & y1) & ¬( ¬ x2 & y2) ) & ¬ (x2 & ¬y2)

**Вероятностная формула**

Z = ( 1 – R1R3 )( 1 – Q2R4 )( 1 - R2Q3 )

#include <reg51.h>

char bdata mem;

bit r;

sbit x1=mem^0;

sbit x2=mem^1;

sbit y1=mem^2;

sbit y2=mem^3;

sbit p1=P1^0;

sbit p2=P2^0;

main()

{

for(mem=0;mem<0x10;mem++)

{

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

if(mem<=0x07){

P1<<=1;

p1=r;

}else{

P2<<=1;

p2=r;

}

}

while(1);

}

**Код программы A51:**

bseg at 0x10

x1 bit ACC.0

x2 bit ACC.1

y1 bit ACC.2

y2 bit ACC.3

cseg at 0x00

jmp START

START:

mov r0, #0

mov P1, #0

mov P2, #0

LOOP:

mov a, r0

mov b, #0x10

cjne a, b, LOOP\_CONTINUE

jmp LOOP\_END

LOOP\_CONTINUE:

// r = !(!(x1&y1)&!(!x2&y2))&!(x2&!y1);

mov c, y1

cpl c

anl c, x2

cpl c

mov ACC.4, c // !(x2&!y1)

mov c, x1

anl c, y1

cpl c

mov ACC.5, c // !(x1&y1)

mov c, x2

cpl c

anl c, y2

cpl c //!(!x2&y2)

anl c, ACC.5 //!(x1&y1)&!(!x2&y2)

cpl c //!((x1&y1)&!(!x2&y2))

anl c, ACC.4 //!((x1&y1)&!(!x2&y2))&!(x2&!y1);

mov ACC.6, c

mov b, a

mov a, r0

subb a,#0x07

jnc SECOND\_BAIT

mov a, b

mov c, ACC.6

mov a, P1

rl a

mov P1, a

mov P1.0, c

jmp INC\_R0

SECOND\_BAIT:

mov a, b

mov c, ACC.6

mov a, P2

rl a

mov P2, a

mov P2.0, c

INC\_R0:

inc r0

jmp LOOP

LOOP\_END:

END