**Министерство образования и наук РФ**

**Федеральное государственное образовательное учреждение**

**Высшего образования**

**Рязанский государственный радиотехнический университет**

**Кафедра ЭВМ**

Отчет по лабораторной работе №1:

“Работа с двоичными масками”

Вариант 11

Выполнила: ст. гр 640

Тограева К. О.

Проверила:

Доц. к.т.н. Елесина С.И.

Рязань 2017

**Цель работы:**  
 получение навыков формирования и применения двоичных масок

при работе с целыми числами.

**Вариант №12:** 2. Из четырех однобайтовых переменных собрать значение для 4-байтовой переменной целого типа. Сборку выполнить так:

б) чтобы значение первой переменной попало в старший (4-й) байт, значение

второй переменной - в 2-й байт, третьей - во третий и четвертой переменной — в младший (1-й) байт переменной целого типа (4 байта).

5. В двухбайтовом числе \*(unsigned short, например) все нули в старшей тетраде поменять на единицы, а в младшей тетраде поменять все единицы на нули.

6. В младшей тетраде двухбайтового числа побитно поменять все нули на единицы, а единицы - на нули.

7. Обычно при циклическом сдвиге «выдвигаемые» (за границу числа) разряды «задвигаются» в сдвигаемое значение со стороны, противоположной направлению сдвига:

е) реализовать циклический сдвиг двухбайтового числа влево на 5 разрядов.

8. В двухбайтовом числе

а) в 1-ю тетраду записать код, равный сумме значений 1-й и 2-й тетрад;

9. Подсчитать число единиц в однобайтовом числе.

Математическая модель

2)2. Из четырех однобайтовых переменных собрать значение для 4-байтовой переменной целого типа. Сборку выполнить так:

б) чтобы значение первой переменной попало в старший (4-й) байт, значение

второй переменной - в 2-й байт, третьей - во третий и четвертой переменной — в младший (1-й) байт переменной целого типа (4 байта).

для сборки четырёхбайтовой переменной взяли ещё дополнительные четырёхабйтовые переменные. В них воодятся значения, и затем они сдвигаются на 28, 20 и 12 байт соотвественно. Затем все эти значения прибавляются к результату (логичиское или).

5)\*5\*. В двухбайтовом числе \*(unsigned short, например) все нули в старшей тетраде поменять на единицы, а в младшей тетраде поменять все единицы на нули.

Выделяем младшую тетраду и делаем сложение по модулю 2. Выделяем старшую тетраду и прибавляем (логическое или) единицы.

6) В младшей тетраде двухбайтового числа побитно поменять все нули на единицы, а единицы - на нули.

Выделяем младшую тетраду и меняем в ней единицы на нули, применяя маску со всеми единицами и используя сложение по модулю два.

7) Обычно при циклическом сдвиге «выдвигаемые» (за границу числа) разряды «задвигаются» в сдвигаемое значение со стороны, противоположной направлению сдвига:

е) реализовать циклический сдвиг двухбайтового числа влево на 5 разрядов.

С помощью масок выделяются первые пять символов и заносятся в переменные. Делается реальный сдвиг на 5 разрядов, затем запомненные разряды прибавляются в конце.

8)В двухбайтовом числе

а) в 1-ю тетраду записать код, равный сумме значений 1-й и 2-й тетрад;

Выделяются первая и вторая тетрады, сдвигаются так, чтобы их можно было сложить и выполняется сложение. Заностися в первую тетраду

9) Подсчитать число единиц в однобайтовом числе.

Из введённой двухбайтовой переменной в начале делается однобайтовая. Заводится переменная counter для подсчёта. Из введённого числа берётся первая цифра, сдвигается вправо и прибавляется к counter. Если там стояла 1, то она прибавляется. Елси 0, то количество единиц не увеличивается.

**Программирование**

**#include <stdio.h>**

**// Вариант 12**

**// задания 2б, 5, 6 7е, 8а, 9**

**unsigned int countFirstNumber(){**

**unsigned int counter = 0;**

**signed int input;**

**scanf( "%d \n", &input );**

**unsigned char c = (char)input;**

**unsigned char mask = 0x80;**

**int umnozh = 7;**

**printf("counter before interaction %x \n", counter);**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**umnozh -= 1;**

**mask = mask >> 1;// сдвигаем единичку вправо**

**counter = counter + ((c & mask) >> umnozh);**

**return counter;**

**}**

**unsigned short makeStrangeEvents(){**

**unsigned short input;**

**scanf( "%hx \n", &input );**

**printf("input is %hx \n", input);**

**unsigned short mask = 0xf000;**

**input = input | mask;**

**mask = 0xfff0;**

**input = input & mask;**

**return input;**

**}**

**unsigned short makeStrangeEvents2(){**

**unsigned short input;**

**scanf( "%hx \n", &input );**

**printf("input is %hx \n", input);**

**unsigned short mask = 0x000f;**

**input = input ^ mask;**

**printf("inverted input is %hx \n", input);**

**return input;**

**}**

**unsigned short makeStrangeEvents3(){**

**unsigned short input;**

**scanf( "%hx \n", &input );**

**printf("input is %hx \n", input);**

**unsigned short mask1 = 0xf000;**

**unsigned short mask2 = 0x0f00;**

**unsigned short final1 = input & mask1;**

**unsigned short final2 = input & mask2;**

**final2= final2 << 4;**

**// это нужно записать в первую тетраду**

**unsigned short res = final1 + final2;**

**printf("1 tetrada %hx \n", res);**

**// и чтобы кроме первой тетрады везде были нолики, что не нолики - обрезаем**

**res = res & mask1;**

**printf("1 tetrada with 0000 0000 0000 is %hx \n", res);**

**// маска для обнуления первой тетрады**

**unsigned short mask3 = 0x0fff;**

**// обнуляем первую тетраду**

**input = mask3 & input;**

**printf("zero first is %hx \n", input);**

**// и праибавляем**

**input = input | res;**

**printf("final result is input is %hx \n", input);**

**return input;**

**}**

**long makeNumberFrom4(){**

**long result = 0;**

**long input1, input2, input3, input4;**

**scanf( "%lx \n", &input1 );scanf( "%lx \n", &input2 );scanf( "%lx \n", &input3 );scanf( "%lx \n", &input4 );**

**input1 = input1 << 28;/**

**input2= input2 << 20;**

**input3 = input3 << 12;**

**result = result | input1;**

**result = result | input2;**

**result = result | input3;**

**result = result | input4;**

**return result;**

**}**

**//c234**

**//**

**//1100 0010 0011 0100**

**//**

**//c234 << 5**

**//**

**//Без циклического сдвига**

**//0100 0110 1000 0000**

**//4680**

**//**

**//с циклическим сдвигомы**

**//1 1000 0100 0110 1001**

**//2 0000 1000 1101 0011**

**//3 0001 0001 1010 0110**

**//4 0010 0011 0100 1100**

**//5 0100 0110 1001 1000**

**//**

**//res 4698**

**unsigned short cycleSdvid(){**

**unsigned short input;**

**scanf( "%hx", &input );**

**printf("input is %hx \n", input);**

**unsigned short mask1 = 0x8000;// первый символ достаём**

**unsigned short mask2 = 0x4000;// второй символ достаём**

**unsigned short mask3 = 0x2000;// третий символ достаём**

**unsigned short mask4 = 0x1000;// четвертый символ достаём**

**unsigned short mask5 = 0x0800;// пятый символ достаём**

**unsigned short s1, s2, s3, s4, s5;// если число получиться неравное 0, то там 1 =)**

**s1 = mask1 & input;**

**s2 = mask2 & input;**

**s3 = mask3 & input;**

**s4 = mask4 & input;**

**s5 = mask5 & input;**

**input = input << 5;**

**// добавляем символы в конце**

**input = input | (s1 >> 11);**

**input = input | (s2 >> 11);**

**input = input | (s3 >> 11);**

**input = input | (s4 >> 11);**

**input = input | (s5 >> 11);**

**return input;**

**}**

**int main()**

**{**

**printf( "Write task number 2б, 5, 6 7е, 8а, 9 (only number, without letter)" );**

**int input;**

**scanf( "%d", &input );// это для ввода информации (d это signed int)**

**switch ( input ) {**

**case 2:**

**//2. Из четырех однобайтовых переменных собрать значение для 4-байтовой**

**//переменной целого типа. Сборку выполнить так:**

**//б) чтобы значение первой переменной попало в старший (4-й) байт, значение**

**//второй переменной - в 2-й байт, третьей - во третий и четвертой переменной - в**

**//младший (1-й) байт переменной целого типа (4 байта).**

**printf("2б\n");**

**printf("input:\n");**

**long res = makeNumberFrom4();**

**printf("result is %lx\n", res);//todo**

**break;**

**case 5:**

**//\*5\*. В двухбайтовом числе \*(unsigned short, например) все нули в старшей тетраде поменять на единицы, а в**

**//младшей тетраде поменять все единицы на нули.**

**printf("5\n");**

**printf("input:\n");**

**unsigned short res4 = makeStrangeEvents();**

**printf("result is %hx\n", res4);**

**break;**

**case 6:**

**//6. В младшей тетраде двухбайтового числа побитно поменять все нули на единицы, а**

**//единицы - на нули.**

**printf("6\n");**

**printf("input:\n");**

**unsigned short res2 = makeStrangeEvents2();**

**printf("result is %hx\n", res2);**

**break;**

**case 7:**

**//7. Обычно при циклическом сдвиге «выдвигаемые» (за границу числа) разряды**

**//«задвигаются» в сдвигаемое значение со стороны, противоположной направлению**

**//сдвига:**

**//е) реализовать циклический сдвиг двухбайтового числа влево на 5 разрядов.**

**printf( "7е\n" );**

**printf("input:\n");**

**unsigned short resu = cycleSdvid();**

**printf("result is %hx\n", resu);**

**break;**

**case 8:**

**//8. В двухбайтовом числе**

**//а) в 1-ю тетраду записать код, равный сумме значений 1-й и 2-й тетрад;**

**printf( "8а\n" );**

**printf("input:\n");**

**unsigned short res8 = makeStrangeEvents3();**

**printf("result is %hx\n", res8);**

**break;**

**case 9:**

**//9. Подсчитать число единиц в однобайтовом числе.**

**printf("9\n");**

**unsigned int counter = countFirstNumber();**

**printf("количество единиц = %d\n", counter);**

**break;**

**default:**

**printf( "Нет такого задания. Увы =()\n" );**

**}**

**getchar();**

**return 0;**

**}**

**Тестовый пример:**

**Пример 1**

Write task number 2б, 5, 6 7е, 8а, 9 (only number, without letter)6

6

input:

4

3

input is 4

inverted input is b

result is b

**Пример 2**

Write task number 2б, 5, 6 7е, 8а, 9 (only number, without letter)9

9

5

counter before interaction 0

количество единиц = 2

**Пример 3**

Write task number 2б, 5, 6 7е, 8а, 9 (only number, without letter)7

7е

input:

787

input is 787

result is f0e0

**Пример 4**

Write task number 2б, 5, 6 7е, 8а, 9 (only number, without letter)5

5

input:

7c23

input is 7c23

result is fc20

**Вывод:**

Мы научились работать с двоичными числами и масками, а также осуществлять побитовые операции