Управление образования

Администрации Сергиево-Посадского района

Муниципальное бюджетное общеобразовательное учреждение

«Физико-математический лицей»

**«Арифметические и логические (битовые) операции. Маски»**

Урок- семинар, в 10 классе

Учитель: Перлова Н..В.

2017-2018 уч.г.

1. Общие сведения:

ФМЛ, 10 класс;

раздел «Компьютерная арифметика»

номер урока – 3.

учебные часы – 2.

1. Вводно-мотивационный этап:

**цели**:

образовательная: дать представление об основных понятиях поразрядных операций: инверсия, поразрядное и, позазрядное или, логический сдвиг;

развивающая: совершенствование умственной и познавательной деятельности учащихся, развитие мышления учащихся;

воспитательная: сознательное усвоение материала учащимися;

ядро содержания обучения: приобретение учащимися навыков работы с побитовыми операциями;

предварительная подготовка учащихся: знакомство с логическим устройством компьютера.

1. Предварительная подготовка учителя: изучение материала урока, написание конспекта, разработка методических пособий.
2. Дидактические основания урока:

методы обучения: объяснительно-иллюстративный;

тип урока: комбинированный урок объяснения нового материала и решения задач;

формы учебной работы учащихся: фронтальная работа

План урока*:*

1. Орг. Момент (1-2 мин)
2. Объяснение нового материала (30 мин)
3. Выполнение задания (10)
4. Итог урока (2 мин).

На прошлом уроке мы рассмотрели работу 16–ти разрядного сумматора. Мы говорили, что возможна ситуация возникновения ошибки при сложении двух чисел. В это случае ошибка фиксируется и в регистре состояний. Регистр обычно рассматривается состоящим из отдельных бит (флагов), каждый из которых несет в себе определенную информацию о каком либо одном признаке результата. Например флаг переноса *(Carry) -*устанавливается при возникновении переноса из старшего разряда результата. Встает вопрос как извлечь значение этого бита чтобы обработать ошибку. Для решения этой задачи и многих других используются побитовые операции. **Поразрядные (побитовые) операции** выполняются с отдельными битами числа и не влияют на остальные.

Сложение – это поразрядная операция? (нет)

**Рассмотрим, какие бывают побитовые операции**

**Побитовое отрицание (NOT)**

**Побитовое отрицание** (или **побитовое НЕ**, или **дополнение**) — это унарная операция, действие которой эквивалентно применению логического отрицания к каждому биту двоичного представления операнда. Другими словами, на той позиции, где в двоичном представлении операнда был 0, в результате будет 1, и, наоборот, где была 1, там будет 0.

**Например:** значение R 10011010 тогда not R 01100101

**Побитовое «И» (AND)**

**Побитовое «И»** — это бинарная операция, действие которой эквивалентно применению логического «И» к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 1, результирующий двоичный разряд равен 1; если же хотя бы один бит из пары равен 0, результирующий двоичный разряд равен 0.

**Пример.** 0101 and 0011 получим 0001

Второй операнд в данном случае позволит нам выделить или обработать только нужные биты. В данном примере мы получим значение двух младших бит. Остальные сбрасываются в 0. Второй операнд по другому называется маска. **Маска** – константа, которая определяет область применения логической операции к битам многоразрядного числа. Побитовые операции позволяют манипулировать непосредственно двоичным содержимым памяти компьютера.

**Пример(задание)**. Что получится если выполнить

**AA16 and 6C16= ? (2816)**

**С помощью операции «И» можно сбросить (установить в ноль) биты, для которых маска равна 0!**

**Побитовое «ИЛИ» (OR)**

**Побитовое «ИЛИ»** — это бинарная операция, действие которой эквивалентно применению логического «ИЛИ» к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 0, двоичный разряд результата равен 0; если же хотя бы один бит из пары равен 1, двоичный разряд результата равен 1.

**Пример**. 0101 or 0011 получим 0111

**Пример(задание): AA16 xor 6C16 = ? (EE16)**

**С помощью операции «ИЛИ» можно записать единицу в биты, для которых маска равна 1!**

**Исключающее «ИЛИ» (XOR)**

**Исключающее« ИЛИ»** (или **сложение по модулю 2**) — это бинарная операция, результат действия которой равен 1, если число складываемых единичных битов нечётно и равен 0, если чётно. Другими словами, если оба соответствующих бита операндов равны между собой, двоичный разряд результата равен 0; в противном случае, двоичный разряд результата равен 1.

**Пример.** 0101 xor 0011 получим 0110

**Пример задание: AA16 xor 6C16 = ? (C616)**

Также данная операция может называться «инверсией по маске», то есть у исходного двоичного числа инвертируются биты, которые совпадают с 1 в маске.

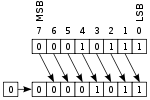
**Битовые логические операции (итог)**

1. **С помощью операции «И» можно сбросить (установить в ноль) биты, для которых маска равна 0!**
2. **С помощью операции «ИЛИ» можно записать единицу в биты, для которых маска равна 1!**
3. **C помощью исключающего «ИЛИ» можно изменить состояние разрядов (проинвертировать)**

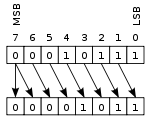
**Битовые сдвиги**

К битовым операциям также относят битовые сдвиги. При сдвиге значения битов копируются в соседние по направлению сдвига. Различают несколько видов сдвигов —мы рассмотрим два.

**Сдвиг влево** (в направлении от младшего бита к старшему)

[](https://commons.wikimedia.org/wiki/File:Rotate_right_logically.svg?uselang=ru) В Паскале **N := N shl 1;**

**Сдвиг вправо**  (в направлении от старшего бита к младшему)

[](https://commons.wikimedia.org/wiki/File:Rotate_right_arithmetically.svg?uselang=ru) В Паскале **N := N shr 1;**

**Рассмотрим работу с поразрядными операциями в Паскале.**

1. {Ввести целое число n и вывести 2𝑛, используя битовые операции}

**var** x:integer;

**begin**

writeln('Введите число');

readln(x);

x:=x **shl** 1;

writeln(x);

**end**.

1. {Вывести двоичное представление целого положительного числа,

используя битовые операции}

**var** x:integer; s:string;

**begin**

writeln('Введите число');

readln(x);s:='';

**while** x>0 **do**

**begin**

**if** x **and** 1 =0 **then** s:='0' +s **else** s:='1' +s;

x:=x **shr** 1;

**end**;

writeln(s);

**end**.

1. {Вывести двоичное представление целого положительного числа,

используя битовые операции}

**var** x:integer; s:string;

**begin**

writeln('Введите число');

readln(x);s:='';

**while** x>0 **do**

**begin**

**if** x **and** 1 =0 **then** s:='0' +s **else** s:='1' +s;

x:=x **shr** 1;

**end**;

writeln(s);

**end**.

Задачи для самостоятельного решения на битовые операции:

вариант 1

1.Найти количество единиц в двоичном представлении целого положительного числа.

2. Ввести целые числа n и i и вывести целое число, у которого i-й бит сброшен в 0, а все остальные биты совпадают с битами числа n на тех же позициях. Например, если введены 11 и 1, ответом будет 9.

вариант 2

1. Найти номер старшего бита в двоичном представлении целого положительного числа.

2. Ввести целые числа n и i и вывести целое число, у которого i -й бит отличается от i -го бита числа n , а все остальные биты совпадают с битами числа n на тех же позициях. Например, если введены 5 и 1, ответом будет

Используемые материалы:

1. <https://de.ifmo.ru/bk_netra/page.php?index=86&tutindex=5>
2. <https://yandex.ru/images/search?source=wiz&img_url=https%3A%2F%2Fcf.ppt-online.org%2Ffiles%2Fslide%2Fe%2FezQIBPJx3ukwClNRAYWEj97hMFnptX5Kc16GOo%2Fslide-71.jpg&p=2&text=%D1%81%D1%83%D0%BC%D0%BC%D0%B0%D1%82%D0%BE%D1%80&pos=79&lr=161980&rpt=simage>
3. <http://kpolyakov.spb.ru/school/probook/slides.htm>
4. <http://www.math.spbu.ru/user/nlebedin/tasks_bit.pdf>