

Информатика



Златопольский Дмитрий Михайлович

Кандидат технических наук, доцент кафедры информатики и прикладной математики Московского городского педагогического университета.

Логические операции над числами

Логические операции конъюнкции (или логического умножения), дизъюнкции (логического сложения) в компьютере используются не только при работе со сложными логическими выражениями при формировании запросов к базам данных, в условных операторах в программах, в функции ЕСЛИ в электронной таблице Microsoft Excel и т. п., но и применительно к числам. Выполняются эти операции в процессоре компьютера (поэтому их называют также логическими командами) над числами, представленными в двоичном виде.

Рассмотрим те логические команды, которые выполняются над двумя числами (говорят, что у них – два операнда):

- 1) **AND** (русский вариант – **И**);
- 2) **OR** (**ИЛИ**);
- 3) **XOR** (от английского *eXclusive OR* – **исключающее ИЛИ**).

В отличие от арифметических операций над двумя операндами, логические команды являются *по-разрядными*. Например, при сложении двух двоичных цифр возможен перенос в старший разряд,

а при логических операциях все разряды рассматриваются изолированно друг от друга. Разумеется, действия над всеми разрядами выполняются параллельно и одновременно. Описанные операции называют также «битовыми».

Чтобы было легче понять, в чём заключаются указанные логические операции в процессоре, условимся называть их первый операнд «данными», а второй – «маской». Правила выполнения логических операций в каждом разряде представлены в таблице 1.

Таблица 1

X (данные)	Y (маска)	XAND Y	XOR Y	XXOR Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Вам эта таблица ничего не напоминает? Да, конечно, она аналогична таблицам истинности для логических операций над величинами логического типа (с той разницей, что здесь операндами являются двоичные цифры).

Например, при $X = 101011$ и $Y = 1101$ имеем $X \text{ AND } Y = 1001$, $X \text{ OR } Y = 101111$, $X \text{ XOR } Y = 100110$.

Конечно, возможна и поразрядная операция **NOT (HE)**. У неё операнд один. В результате её выполнения в операнде двоичные цифры 0 и 1 меняются соответственно на 1 и 0 (число инвертируется).

Несмотря на то что, как отмечалось, логические операции выполняются в процессоре компьютера, в языке программирования Паскаль имеется воз-

можность дать команду на их выполнение в программе. Для этого в Паскале существуют операции с соответствующими именами: **and**, **or**, **xor**, **not**. Их можно применять к данным целого типа, и результат, который они возвращают, также является целым числом.

Так, результатом операции **and** над двумя операндами является десятичное число, которое соответствует двоичному представлению результата поразрядной операции **И** над исходными операндами. Например, в результате выполнения фрагмента программы на языке Паскаль:

```
x := 11 and 2;
writeln(x);
```

на экран будет выведено число 2. Обоснование приведено в табл. 2:

Таблица 2

	Двоичное представление			
11	1	0	1	1
2	0	0	1	0
11 and 2	0	0	1	0
Десятичное представление	2			

Результатом операции **or** над двумя операндами является десятичное число, которое соответствует двоичному представлению результата поразрядной операции **ИЛИ** над исходными операндами. Например, в

результате выполнения фрагмента программы:

```
x := 11 or 2;
writeln(x);
```

на экран будет выведено число 11. Обоснование:

Таблица 3

	Двоичное представление			
11	1	0	1	1
2	0	0	1	0
11 or 2	1	0	1	1
Десятичное представление	11			

Результатом операции **xor** над двумя операндами является десятичное число, которое соответствует двоичному представлению результата поразрядной операции исключающего **ИЛИ** над исходными опе-

рандами. Например, в результате выполнения фрагмента программы:

```
x := 11 xor 2;
writeln(x);
```

на экран будет выведено число 9 (см. табл. 4).

Таблица 4

	Двоичное представление			
11	1	0	1	1
2	0	0	1	0
11 xor 2	1	0	0	1
Десятичное представление	9			

Результатом операции **not** с одним операндом является десятичное число, которое соответствует двоичному представлению результата поразрядной операции **НЕ** над исходным операндом. Например, в результате выполнения фрагмента программы:

```
Var a: byte;
...
a := 11;
writeln(not a);
```

на экран будет выведено число 244. Обоснование:

Таблица 5

	Двоичное представление							
11	0	0	0	0	1	0	1	1
not 11	1	1	1	1	0	1	0	0
Десятичное представление	244							

Предлагаю читателям выполнить ряд интересных заданий, связанных с логическими операциями над числами.

1. Рассчитайте значения:

- 1101101 **AND** 10101;
- 1011011 **OR** 11001;
- 1001101 **XOR** 10111.

2. Определите:

1) чему равен результат операции **И** при нулевой маске?

2) чему равен результат операции **ИЛИ** при маске:

- из всех единиц?
- в виде нуля?

3) какая операция и с какой маской позволяет выделить (получить) младший разряд двоичного числа?

4) какой получится результат, если к какому-то числу дважды применить операцию исключающего **ИЛИ**?

5) какой получится результат, если операцию исключающего **ИЛИ** применить к двум одинаковым числам?

6) какой получится результат, если к какому-то числу применить операцию исключающего **ИЛИ** с маской из всех единиц?

3. Выясните, можно ли определить десятичный результат выполнения поразрядной логической операции **NOT** с целым десятичным числом типа *byte* «в уме»? Почему?

4. Установите, можно ли применять поразрядные логические опера-

ции к одному или двум отрицательным операндам?¹

5. Определите результат выполнения следующего фрагмента программы на языке Паскаль:

```
Var a: byte; b: shortint;  
BEGIN  
  a := 11;  
  writeln(not a);  
  b := 11;  
  writeln(not b)  
END.
```

Почему выводятся разные значения?

6. Объясните, почему $a \text{ and } -(a + 1) = 0$? (Например, $51 \text{ and } -52 = 0$).

7. Каким будет результат применения операции **NOT** к целому положительному числу – положительным или отрицательным?

8. Может ли результат применения операции **AND** к двум положительным числам быть отрицательным? А операции **OR**? А операции **XOR**?

9. Может ли результат применения операции **AND** к двум отрицательным числам быть положительным? А операции **OR**? А операции **XOR**?

10. Может ли результат применения операции **AND** к двум числам разного знака быть положительным?

А операции **OR**? А операции **XOR**?

11. Определите, каким (положительным или отрицательным) будет результат выражения **NOT (A AND B)**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

12. Определите, каким (положительным или отрицательным) будет результат выражения **NOT (A OR B)**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

13. Определите, каким (положительным или отрицательным) будет результат выражения **NOT (A XOR B)**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

14. Определите, каким (положительным или отрицательным) будет результат выражения **NOT A AND NOT B**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

15. Определите, каким (положительным или отрицательным) будет результат выражения **NOT A OR NOT B**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

16. Определите, каким (положительным или отрицательным) будет результат выражения **NOT A XOR NOT B**, если:

- 1) A и B – положительные числа;
- 2) A и B – отрицательные числа;
- 3) A и B – числа разного знака.

¹ Напомним, что в двоичном представлении целых чисел со знаком старший разряд является знаковым – если в нем записан 0, то число положительное, если 1 – отрицательное. При выполнении заданий, в которых фигурируют числа со знаком, следует принять некоторую разрядность операндов (например, 8).