



Основы C++

Неважно, на каком языке вы программируете. Если вы не знаете C++, вы не можете считать себя программистом.

[Подробнее >>>](#)

Подписаться:



[Главная](#) [Ассемблер](#) [Микроконтроллеры](#) [Инструкции Intel](#) [Дневник](#)



Микроконтроллеры для ЧАЙНИКОВ

[Изучать БЕСПЛАТНО](#)

14.09.2020 г.

Добавлена статья [Уменьшение энергопотребления.](#)

[05.09.2020 г.](#)

Добавлены видео и статья [Самое простое устройство на микроконтроллере](#).

[21.08.2020 г.](#)

Добавлены видео и статья [Инструкция CLI](#).

[19.06.2020 г.](#)

Добавлена статья [Выводы ATtiny13A](#).

[19.05.2020 г.](#)

Добавлена статья [Регистр PRR](#).

Логическое ИЛИ



Что такое JavaScript

Если вы интересуетесь программированием вообще, и сайтостроением в частности, то вы наверняка слышали слово JavaScript. И, если вы до сих пор не узнали толком, что же это такое, то пришло время сделать это.

[Подробнее...](#)

0017 Логическое ИЛИ



Операция логического ИЛИ в Ассемблере выполняется с помощью команды OR. Эта команда выполняет логическое ИЛИ между всеми битами двух операндов. Результат записывается в первый операнд. Синтаксис:

OR ЧИСЛО1, ЧИСЛО2

В зависимости от результата могут быть изменены [флаги](#) ZF, SF, PF. Флаги OF и CF всегда сбрасываются.

ЧИСЛО1 может быть одним из следующих:

- Область памяти (MEM)
- Регистр общего назначения (REG)

ЧИСЛО2 может быть одним из следующих:

- Область памяти (MEM)
- Регистр общего назначения (REG)
- Непосредственное значение (IMM)

С учётом ограничений, которые были описаны выше, комбинации ЧИСЛО1-ЧИСЛО2 могут быть следующими:

REG,	MEM
MEM,	REG
REG,	REG
MEM,	IMM
REG,	IMM

Когда мы изучали [команду AND](#), то мы узнали, что с её помощью можно преобразовать маленькую букву в большую, используя [битовую маску](#).

Инструкция OR поступает наоборот - она может преобразовать большую букву в маленькую:

```
MOV AL, 'A'      ; AL = 01000001b
AND AL, 00100000b ; AL = 01100001b ('a')
```

Размер обоих операндов должен быть одинаковым: 8, 16 или 32 разряда.

Команда OR обычно используется для установки в единицу отдельных битов [двоичного числа](#) (например, флагов состояния процессора) по заданной [маске](#).

Если бит маски равен нулю, то значение соответствующего бита числа не изменяется, а если равен 1, то устанавливается в 1.

Ниже показано, как можно установить четыре младших бита числа:

```
0 0 1 1 1 0 1 1   - Исходное значение
0 0 0 0 1 1 1 1   - Битовая маска
0 0 1 1 1 1 1 1   - Результат
```

В результате четыре младших бита будут установлены в любом случае, независимо от того, какое состояние у них было в исходном значении (выделены красным цветом). Старшие четыре бита не изменятся, то есть будут такими же, как у исходного значения.

Ещё одно применение инструкции OR - это преобразование двоичной цифры от 0 до 9 в значение, которое соответствует ASCII-коду символа, обозначающему эту цифру. Для этого нужно установить в единицу биты 4 и 5.

Пример:

```
0 0 0 0 0 1 0 1   - Исходное значение: число 5
0 0 1 1 0 0 0 0   - Битовая маска (биты 4 и 5 установлены)
0 0 1 1 0 1 0 1   - Результат: 35h - ASCII-код символа 5
```

Пример программы:

```
.model tiny
.code
ORG 100h

start:

MOV AX, 0B800h ;установить AX = B800h (память VGA).
MOV DS, AX     ;копировать значение из AX в DS.
MOV CL, 'A'    ;CL = 41h (ASCII-код символа 'A').
MOV CH, 01001110b ;CH = атрибуты цвета (желтый текст на красном фоне).
MOV BX, 72eh   ;BX = позиция на экране = 2*(x + y*80) = (39, 11).
MOV [BX], CX   ;[0B800h:015Eh] = CX (записать символ в видеопамять).
OR CL, 00100000b ;Теперь CL = 61h (ASCII-код символа 'a').
MOV BX, 730h   ;BX = позиция на экране = 2*(x + y*80) = (40, 11).
MOV [BX], CX   ;[0B800h:015Eh] = CX (записать символ в видеопамять).

MOV CL, 5      ;Записать число в регистр CL
OR CL, 00110000b ;Преобразовать в ASCII-код
MOV AX, 0B800h ;установить AX = B800h (память VGA).
MOV DS, AX     ;копировать значение из AX в DS.
MOV CH, 01001110b ;CH = атрибуты цвета (желтый текст на красном фоне).
MOV BX, 732h   ;BX = позиция на экране = 2*(x + y*80) = (41, 11).
MOV [BX], CX   ;[0B800h:015Eh] = CX (записать символ в видеопамять).

END start
```

Таблица истинности логического ИЛИ

Как было сказано, команда OR выполняет операцию логического сложения (логического ИЛИ) между всеми битами двух чисел. Таблица истинности для операции логического сложения выглядит следующим образом:

0	И	0	=	0
0	И	1	=	1
1	И	0	=	1
1	И	1	=	1

Логическое ИЛИ - это операция логического сложения. Чтобы легче было запомнить таблицу истинности для логического ИЛИ, вспомните математику:

$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 1$

Последний случай вас, возможно, смутит. Но на самом деле всё в порядке))) Просто по правилам булевой алгебры при логическом сложении результат равен нулю, если все операнды равны нулю. Во всех остальных случаях результат равен 1.

[Подписаться на канал в YouTube](#)

[Вступить в группу "Основы программирования"](#)

[Подписаться на рассылки по программированию](#)



Первые шаги в программирование

Главный вопрос начинающего программиста – с чего начать? Вроде бы есть желание, но иногда «не знаешь, как начать думать, чтобы до такого додуматься». У человека, который никогда не имел дело с информационными технологиями, даже простые вопросы могут вызвать большие трудности и отнять много времени на решение. [Подробнее...](#)

Инфо-МАСТЕР®

Все права защищены ©

e-mail: mail@info-master.su

[Главная](#)

[Карта](#)

[Контакты](#)

