



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 СЦ.](#)

[19.06.2020 г.](#)

Добавлена статья [Выводы ATtiny13A.](#)

[19.05.2020 г.](#)

Добавлена статья [Регистр PRR.](#)

Команда NEG



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...](#)

0019 Команда NEG



Команда NEG в Ассемблере изменяет знак числа на противоположный, то есть положительное число делает отрицательным, а отрицательное - положительным. При этом число преобразуется в двоичный дополнительный код (иногда это называют “дополнение до двух”). Синтаксис:

NEG ЧИСЛО

Инструкция NEG изменяет флаги CF, ZF, SF, OF, AF, PF. Значения [флагов](#) зависят от результата операции отрицания.

ЧИСЛО может быть одним из следующих:

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

Операция отрицания

Алгоритм выполнения операции отрицания следующий:

1. Инвертировать все биты ЧИСЛА
2. Прибавить единицу к результату выполнения п.1

Надо понимать отличие инструкции NEG от [логического отрицания](#), которое выполняется командой NOT.

Команда NOT только инвертирует все биты числа, но не делает его отрицательным. А команда NEG именно делает число отрицательным. Как? См. дальше.

Дополнение до двух

В общем то алгоритм дополнения до двух уже описан выше. Суть его в том, что все разряды числа инвертируются, а затем к результату прибавляется единица. Таким образом меняется знак числа.

То есть аналогом команды NEG (с определёнными допущениями) можно считать такой код:

```
NOT AX  
ADD AX, 1
```

Более подробно о дополнении до двух см. [здесь](#).

Возможные неожиданности

Результат выполнения команды NEG может вас неприятно удивить. Он будет неправильным, если размер операнда этой команды слишком мал.

Например, если загрузить в [регистр](#) AL отрицательное число -128, а затем попытаться инвертировать значение в регистре AL, то по идее должно получиться число +128. Однако это число уже не поместится в регистр AL, так как в регистре AL может поместиться только один байт, а один байт в случае чисел со знаком может хранить значения только от -128 до +127.

В таком случае будет установлен флаг OF (переполнение), а значение в регистре AL (то есть результат выполнения команды NEG) будет неправильным:

```
MOV AL, -128 ; AL = 10000000b  
NEG AL ; AL = 10000000b, OF = 1
```

Ну а если в регистр AL загрузить число +127 и выполнить команду NEG, то результат будет правильным, а флаг OF не будет установлен:

```
MOV AL, 127 ; AL = 01111111b  
NEG AL ; AL = 10000000b, OF = 0
```

Пример программы, в которой используется команды NEG, приведён ниже.

```
.model tiny  
.code  
ORG 100h
```

start:

; Правильный результат

```
MOV AL, 127      ; AL = 01111111b = 7Fh
NEG AL           ; AL = 10000001b = 81h, OF = 0, SF = 1
```

; Правильный результат

```
MOV AL, -127     ; AL = 10000001b = 81h
NEG AL           ; AL = 01111111b = 7Fh, OF = 0, SF = 0
```

; Неправильный результат

```
MOV AL, -128     ; AL = 10000000b = 80h
NEG AL           ; AL = 10000000b = 80h, OF = 1, SF = 1
```

; Правильный результат

```
MOV AL, -1       ; AL = 11111111b = FFh
NEG AL           ; AL = 00000001b = 01h, OF = 0, SF = 0, CF = 1
```

; Пробуем обойтись без команды NEG

```
MOV AL, -1       ; AL = 11111111b = FFh
NOT AL           ; AL = 00000000b = 0h
ADD AL, 1        ; AL = 00000001b = 01h, OF = 0, SF = 0, CF = 1
```

```
MOV AL, -128     ; AL = 10000000b = 80h
NOT AL           ; AL = 01111111b = 7Fh
ADD AL, 1        ; AL = 10000000b = 80h, OF = 1, SF = 1
```

END start

Ну и напоследок о происхождении мнемоники **NEG**. Это сокращение от слова **NEGATIVE** - отрицать, отрицание.

[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](#)