

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

Команда AAA

0025 Команда AAA



Инструкция AAA в Ассемблере делает необходимые поправки в [регистрах](#) AH и AL после сложения при работе с [BCD-значениями](#). У этой команды нет операндов.

Команда AAA работает согласно следующему алгоритму:

Если младшие (правые) четыре бита регистра AL > 9 или флаг AF = 1, то:

```
AL = AL + 6
AH = AH + 1
AF = 1
CF = 1
```

иначе

```
AF = 0
CF = 0
```

в любом случае команда очищает старшие четыре бита регистра AL.

[Флаги](#) AF и CF изменяются в зависимости от результата выполнения команды (см. алгоритм работы выше). Остальные флаги имеют неопределённое состояние и анализировать их нет смысла.

А теперь ещё немного, почти то же самое, что было сказано выше, но другими словами. Возможно, кому то это будет более понятно.

Команда AAA исправляет сумму двух неупакованных [двоично-десятичных чисел](#), которая находится в регистре AL.

Если исправление приводит к десятичному переносу, то значение регистра AH увеличивается на единицу.

Соответственно, перед вызовом команды AAA надо поместить сумму двух неупакованных двоично-десятичных чисел в регистр AL. То есть инструкцию AAA лучше использовать сразу после сложения двух таких чисел.

Например, если при сложении чисел 06 и 09 в регистре AX окажется число 000Fh, то команда AAA исправит его, и в регистре AX будет число 0105 (неупакованное десятичное число 15):

```
MOV AX, 15    ; AH = 00, AL = 0Fh = 15
AAA           ; AH = 01, AL = 05
```

Команда AAA применяется после выполнения команд [ADD](#) или ADC. В итоге сумма, которая находится в регистре AL, всегда будет соответствовать представлению чисел в формате ASCII.

Ниже приведён пример, где показано, как можно сложить два BCD-значения 6 и 9, а с помощью команды AAA получить правильную сумму в BCD-формате (в десятичном упакованном формате).

Обратите внимание, что перед сложением нужно обнулить содержимое регистра AH.

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

start:

```
MOV AH, 0      ;Обнулить AH
MOV AL, 6      ;AL = 6
ADD AL, 9      ;AL = 6 + 9 = 15 = 0Fh
AAA            ;Выполнить поправку
OR  AX, 3030h  ;Преобразовать в коды ASCII
MOV DL, AH     ;DL = 31h (ASCII-код числа 1)
```

```
MOV BX, 676h      ;Позиция первого символа на экране
MOV CX, 0B800h    ;Установить CX = B800h (память VGA)
MOV DS, CX        ;Копировать значение из CX в DS
MOV DH, 01001110b ;DH = атрибуты цвета
MOV [BX], DX      ;Записать символ в видеопамять
MOV DL, AL        ;AL = 35h (ASCII-код числа 5)
ADD BX, 2         ;Сместить позицию на экране
MOV [BX], DX      ;Записать символ в видеопамять

END      start
```

Способ вывода на экран с помощью команды MOV мы уже рассматривали [здесь](#) и [здесь](#). Остальное, думаю, понятно из описания.

С помощью команды OR AX, 3030h мы преобразуем содержимое регистров AH и AL в ASCII-коды символов, которые соответствуют находящимся в этих регистрах числам в BCD-формате. В итоге получаем в AX число 3135h (31h - ASCII-код символа 1, а 35h - ASCII-код символа 5).

Ну и напоследок о происхождении названия команды.

AAA - это ASCII Adjust After Addition - ASCII-коррекция после сложения.

[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](#)

