

Языки программирования (Asm)

Занятие №6. Работа с двоично-десятичными числами.
Команды для работы с отдельными битами.

МИРЭА - РТУ, БК №252

Москва 2020

BCD-числа в ассемблере

Двоично-десятичный код (Binary-Coded Decimal) — форма записи рациональных чисел, при которой каждый десятичный разряд числа записывается в виде его четырёхбитного двоичного кода.¹

Например, десятичное число 35 может храниться в регистре AX следующим образом

$AX = 0x0305 = 0b0000001100000101$, т.е.

$AL = 0x05 = 0b00000101$

$AH = 0x03 = 0b00000011$

Таким образом, двузначное число будет занимать два байта памяти.

Этот случай называется неупакованным BCD-числом.

Если же обе цифры, составляющие число 35 будут храниться в одном байте таким образом: $0x35 = 0b00110101$, то данное число называют упакованным BCD-числом.

¹<https://ru.wikipedia.org/>

Список команд для обработки BCD-чисел

Следующие четыре команды используются для преобразования неупакованных BCD-чисел.

- ① AAA - (Ascii Adjust after Addition)
- ② AAD - (Ascii Adjust before Division)
- ③ AAM - (Ascii Adjust after Multiply)
- ④ AAS - (Ascii Adjust after Substraction)

Также существует две команды для преобразования упакованных BCD-чисел.

- ① DAA - (Decimal Adjust for Addition)
- ② DAS - (Decimal Adjust for Subtraction)

AAA ASCII-коррекция после сложения

Обычно команда AAA используется после сложения каждого десятичного разряда упакованных BCD-чисел командой ADD. Каждая цифра упакованного BCD-числа занимает младший полубайт байта. При этом старшие полубайты всех байт равны нулю. Если в результате поразрядного сложения двух BCD-чисел получается результат больше 9, то это число перестает быть BCD-числом. Его нужно преобразовывать к BCD-формату. Для корректировки используется команда AAA. Если необходимо, она увеличивает число десятков на 1, а также формирует правильное BCD-число в младшем полубайте. Подробнее рассмотрим на примере.

Пример работы команды AAA

Пусть в регистрах AX и BX находятся два неупакованных BCD-числа 19 и 6. В шестнадцатичном представлении это задается так:
AX = 0x0109, BX = 0x0006.

```
1 mov al, 9
2 mov ah, 1
3 mov bl, 6
4 mov bh, 0
5 add al, bl ; Сложили 0x09 и 0x06, получили 0xF
6 aaa
```

Команда сложения ADD применяется к младшим десятичным разрядам двух чисел. В результате, в регистре AL окажется число $15=9+6$. Команда AAA отнимает от AL лишний десяток, и прибавляет к AH единицу. Таким образом формируется верный результат в виде неупакованного BCD-числа. AX = 0x0205.

Команда AAD. ASCII-коррекция перед делением

Команду AAD используют до выполнения операции деления для подготовки двух неупакованных BCD-чисел. Команда AAD выполняет преобразование двузначного неупакованного BCD-числа, хранящегося в регистре AX в двоичное представление.

```
1 mov ah, 1
2 mov al, 8
3 mov bl, 9
4 aad
5 div bl
6 add al, 0x30
```

В данном примере неупакованное BCD-число $18 = 0x0108$ было преобразовано в число $0x12$, равное 18 в десятичной системе счисления. Число $0x12$ после выполнения команды AAD хранилось в регистре AL.

AAM ASCII-коррекция после умножения

Команду AAM используют для коррекции результата умножения двух неупакованных BCD-чисел. Специальной команды умножения BCD-чисел нет. Поэтому BCD-числа умножаются поразрядно, как обычные двоичные числа, командой MUL. Максимальное число, которое получается при таком умножении — это $9 * 9 = 81_{10} = 51_{16}$. Отсюда понятно, что значения, для которых командой AAM можно получить их двузначный BCD-эквивалент в регистре AX, находятся в диапазоне от 0 до 51_{16} . Эту команду можно применять и для преобразования двоичного числа из регистра AX (в диапазоне от 0 до 63_{16}) к формату BCD-числа (от 0 до 99_{10}).

```
1 mov ax, 0x62
2 aam
3 add al, 0x30
4 add ah, 0x30
5 mov [y], al
6 mov [x], ah
```

AAS ASCII-коррекция после вычитания

Команду AAS используют для коррекции результата вычитания двух неупакованных одноразрядных BCD-чисел после команд вычитания в случае, если результат отрицательный. Операндами в команде SUB должны быть правильные одноразрядные BCD-числа.

```
1 mov al, 3 ; Нужно вычислить 3 - 8
2 mov bl, 8
3 sub al, bl
4 aas
5 mov bl, 10
6 sub bl, al
7 add bl, 0x30
```

Для получения результата нужно отнять содержимое AL от 10. Команда может изменять значения флагов CF и AF.

DAA Десятичная коррекция после сложения

Эту команду следует применять после сложения двух упакованных BCD-чисел с целью корректировки получающегося двоичного результата сложения в правильное двузначное десятичное число. После команды DAA следует анализировать состояние флага CF. Если он равен 1, то это говорит о том, что был перенос единицы в старший разряд и это нужно учесть для сложения старших десятичных цифр BCD-числа.

```
1 mov     al,69h
2 mov     bl,74h
3 adc     al,bl
4 daa
5 jc      m1
```

В случае возникновения переноса, его нужно обработать в одной из веток кода.

DAS Десятичная коррекция после вычитания

Команду DAS следует применять после вычитания двух упакованных BCD-чисел с целью корректировки получающегося двоичного результата вычитания в правильное двузначное десятичное число. После команды DAS следует анализировать состояние флага CF. Если он равен 1, то это говорит о том, что был заем единицы в старший разряд и это нужно учесть в дальнейших действиях. Если у вычитаемого нет больше старших разрядов, то результат следует трактовать как отрицательное двоичное дополнение. Для определения его абсолютного значения нужно вычесть 100 из результата в AL. Если у вычитаемого еще есть старшие разряды, то факт заема нужно просто учесть уменьшением младшего из этих оставшихся старших разрядов на единицу.

Работа с битами. BT проверка битов

Команду BT используют для определения значения конкретного бита в операнде источник. Номер проверяемого бита задается содержимым второго операнда (значение числом из диапазона 0...63). После выполнения команды, флаг CF устанавливается в соответствии со значением проверяемого бита.

```
1 mov     ebx, 0b01001100
2 bt      ebx, 5
3 jc      m1
```

В данном примере 5-й бит равен 0, переход к метке m1 выполнен не будет.

BTC, BTR

BTC (Bit Test and Complement) - Проверка бита с инверсией (дополнением)

Команда BTC используется для определения и инвертирования значения конкретного бита в первом операнде. Номер проверяемого бита задается содержимым второго операнда (значение из диапазона 0...31). После выполнения команды флаг CF устанавливается в соответствии с исходным значением бита, то есть тем, которое было до выполнения команды.

BTR - (Bit Test and Reset) - Проверка бита с его сбросом в 0

```
1 mov     ebx, 0b01001100
2 btc     ebx, 8
```

BTS, BSF, BSR

BTS - (Bit Test and Set) - Проверка бита с его установкой в 1.

BSF - (Bit Scan Forward) - Побитовое сканирование вперед.

Команду BSF используют при работе на битовом уровне для определения позиции в операнде крайнего правого единичного бита.

BSR - (Bit Scan Reverse) - Побитовое сканирование назад.

Команду BSR используют при работе на битовом уровне для определения позиции крайнего левого единичного бита.

```
1 mov     bx, 41h
2 bsr     ax, bx      ; 06h
3 jz      m1
4 shr     bx, ax      ; bx=0b0001
```