

12. Двоично-десятичная арифметика.

BСD. Двоично-десятичные числа

Упакованный BCD-формат (*Packed BCD*) — это упакованное двоично-десятичное число-байт от 00h до 99h. Например, 71 = 71h. Код старшей цифры числа занимает старшие четыре бита.

Двоично-десятичный код (*binary-coded decimal*) - это числа, в которых каждая цифра записывается в виде своего двоичного представления.

Десятичное число	Двоично-десятичный код
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Неупакованное BCD число (*Unpacked BCD*) - это двоично-десятичное число-байт от 00h до 09h. В неупакованном формате каждый байт содержит одну десятичную цифру в четырех младших битах. Старшие четыре бита имеют нулевое значение.

Команды

DAA

Команда DAA (Decimal Adjust AL after Addition) позволяет получать результат сложения упакованных двоично-десятичных данных в таком же упакованном BCD-формате. То есть она корректирует после сложения, пример:

```
mov AL, 71h      ; AL = 0x71h
add AL, 44h      ; AL = 0x71h + 0x44h = 0xB5h
daa              ; AL = 0x15h
                  ; CF = 1 - перенос является частью результата 71 + 44 = 115
```

DAS

Команда DAS (Decimal Adjust AL after Subtraction) позволяет получать результат вычитания упакованных двоично-десятичных данных в таком же упакованном BCD-формате. То есть она корректирует после вычитания, пример:

```
mov AL, 71h      ; AL = 0x71h
sub AL, 44h      ; AL = 0x71h - 0x44h = 0x2Dh
das              ; AL = 0x27h
                  ; CF = 0 - заем (перенос) является частью результата
```

AAA

Команда AAA (ASCII Adjust After Addition) - позволяет преобразовать результат сложения BCD-чисел в ASCII формат. Для того чтобы преобразовать содержимое регистра AL к ASCII формату, необходимо после команды AAA выполнить команду OR AL, 0x30h. Команда AAA должна выполняться после команды ADD, которая помещает результат сложения в AL. Если будет перенос, он записывается в AH.

```
xor ah, ah
add al, '6'      ; al = 0x36h
add al, '8'      ; al = 0x36h + 0x38h = 0x6Eh
aaa              ; ax = 0x0104h
or al, 0x30h     ; al = 0x34h = '4'
```

AAS

Команда AAS (ASCII Adjust After Subtraction) - позволяет преобразовать результат вычитания BCD-чисел в ASCII формат. Для этого команда AAS должна выполняться после операнды SUB, которая помещает 1 байтный результат вычитания в AL. Если был заем, то 1 вычитается из AH.

```
; положительный результат
xor ah, ah
mov al, '9'      ; al = 0x39h
sub al, '3'      ; al = 0x39h - 0x33h = 0x06h
aas              ; ax = 0x0006h
or al, 0x30h     ; al = 0x36h = '6'

; отрицательный результат
xor ah, ah
mov al, '3'      ; al = 0x33h
sub al, '9'      ; al = 0x33h - 0x39h = 0xFAh
aas              ; AX = 0xFF04h
or al, 0x30h     ; al = 0x34h = '4'
```

AAM

Команда AAM (ASCII Adjust AX After Multiply) позволяет преобразовать результат умножения неупакованных двоично-десятичных данных в ASCII-формат. Для этого команда AAM должна выполняться после команды беззнакового умножения MUL (но не после команды умножения со знаком IMUL), которая помещает двухбайтный результат в регистр AX.

Команда AAM распаковывает результат умножения, содержащийся в регистре AL, деля его на второй байт кода операции ib (равный 0x0Ah для безоперандной мнемоники AAM). Частное от деления (наиболее значащая цифра результата) помещается в регистр AH, а остаток (наименее значащая цифра результата) — в регистр AL.

Для того чтобы преобразовать содержимое регистра AX к ASCII-формату, необходимо после команды AAM выполнить команду OR AX, 0x3030h.

```
mov AL, 3      ; множимое в формате неупакованного BCD помещается в регистр A
mov BL, 9      ; множитель в формате неупакованного BCD помещается в регистр B
mul BL         ; AX = 0x03 * 0x09 = 0x001Bh
aam           ; AX = 0x0207h
or AX, 3030h   ; AX = 0x3237h, т.е. AH = '2', AL = '7'
```

AAD

Команда AAD (ASCII Adjust AX Before Division) используется для **подготовки** двух разрядов неупакованных BCD-цифр (наименее значащая цифра в регистре AL, наиболее значащая цифра в регистре AH) для операции деления DIV, которая возвращает неупакованный BCD-результат.

Команда AAD устанавливает регистр AL в значение $AL = AL + (imm8 * AH)$, где imm8 – это второй байт кода операции ib (равный 0x0Ah для безоперандной мнемоники AAD), с последующей очисткой регистра AH. После команды AAD регистр AX будет равен двоичному эквиваленту оригинального неупакованного двухзначного числа.

```
mov AX, 0207h  ; делимое в формате неупакованного BCD помещается в регистр AX
mov BL, 05h    ; делитель в формате неупакованного BCD помещается в регистр B
aad           ; AX = 0x001Bh
div BL        ; AX = 0x0205h
or AL, 30h     ; AL = 0x35h = '5'
```