

АРИФМЕТИЧЕСКИЕ КОМАНДЫ

КЛАССИФИКАЦИЯ ПО ВИДАМ ОПЕРАНДОВ

1. Двоичные целые без знака 8 бит (0..255) или 16 бит (0..65535).
2. Двоичные целые со знаком 8 бит (-128..+127) или 16 бит (-32768..+32767).
3. Неупакованные десятичные: ЦБЗ от 0000 до 1001, т.е. одна десятичная цифра от 0 до 9, в одном байте.
Пр. 00000111b десятичная 7.
4. Упакованные десятичные: ЦБЗ от 0000 до 1001, т.е. одна десятичная цифра от 0 до 9, в старшем полубайте и ЦБЗ от 0000 до 1001 в младшем полубайте.
Пр. 10010111 десятичное 97.

! Хранение чисел в ОЗУ - справа налево:

"младший байт - младший адрес, старший байт - старший адрес".

Пр. 1234H -> 00010010 00110100

N+1 N <- номер байта в ОЗУ.

! МП обрабатывает все виды операндов в основном одними командами, но для десятичных есть дополнительные команды.

! Арифметические команды могут влиять только на следующие флаги: OF, SF, ZF, AF, PF, CF. *

```
SC1      SEGMENT   PARA PUBLIC 'CODE'
          ASSUME CS:SC1,DS:SC1,SS:SSEG
BEGIN    PROC      FAR
          MOV AX,SC1
          MOV DS,AX
          NOP
```

comment *

КОМАНДЫ СЛОЖЕНИЯ

1. Сложение байтов или слов:

ADD Приемник, Источник

Функция: Приемник := Приемник + Источник

изменяет флаги: OF, SF, ZF, AF, PF, CF.

Пример. ADD AH,AL 10101001 ў AH
 +01101110 ў AL

Перенос CF:=1 -> 1 00010111 ў AH

Знак + SF:=0

PF:=1 - четное число единиц

AF:=1 - есть дополнительный перенос

OF:=0 - нет переполнения *

```
MOV AH,10101001B
MOV AL,01101110B
ADD AH,AL
NOP
```

comment *

2. Сложение байтов или слов с переносом:

ADC Приемник, Источник

Функция: Приемник := Приемник + Источник + CF
изменяет флаги: OF, SF, ZF, AF, PF, CF.

Пример. Сложение 32-разрядных ЦБЗ

```
CX 0000 F000 AX
    +      +
DX 0000 8000 BX
    +CF
-----
CX 0001 7000 AX
```

*

```
MOV AX,0F000H
MOV BX,8000H
MOV CX,0
MOV DX,0
ADD AX,BX
ADC CX,DX
NOP
```

comment *

3. Увеличение байта или слова на 1:

INC Приемник

Функция: Приемник := Приемник + 1
изменяет флаги: OF, SF, ZF, AF, PF.
! не влияет на CF !

Пример.

```
MOV AH,11111111B
INC AH ;CF не изменился!
NOP
```

*

comment *

4. AAA - коррекция в AL неупакованного кода при сложении

5. DAA - коррекция в AL упакованного кода после сложения

КОМАНДЫ ВЫЧИТАНИЯ

1. Вычитание байтов или слов:

SUB Приемник, Источник

Функция: Приемник := Приемник - Источник
изменяет флаги: OF, SF, ZF, AF, PF, CF.

Пример. SUB AH,AL 00101001 ÿ AH
 ЦБЗ прям.код -> -01101110 ÿ AL

Есть заем CF:=1 -> 1 10111011 ÿ AH

Знак -, SF:=1

PF:=1 - четное число единиц

AF:=1 - есть дополнительный заем

OF:=0 - нет переполнения

*

```
MOV AH,00101001B
MOV AL,01101110B
SUB AH,AL
NOP
```

comment *

2. Вычитание байтов или слов с заемом:

SBB Приемник,Источник

Функция: Приемник:=Приемник-Источник-CF
изменяет флаги:OF,SF,ZF,AF,PF,CF.

Пример. Вычитание 32-разрядных ЦБЗ

CX 0020 7000 AX

- -

DX 0000 8000 BX

-CF

CX 001F F000 AX

*

MOV AX,7000H

MOV BX,8000H

MOV CX,020H

MOV DX,0

SUB AX,BX

SBB CX,DX

NOP

comment *

3. Уменьшение байта или слова на 1:

DEC Приемник

Функция: Приемник:=Приемник-1
изменяет флаги:OF,SF,ZF,AF,PF.
! не влияет на CF !

Пример.

MOV AH,0

DEC AH ;CF не изменился!

NOP

*

comment *

4. Изменение знака байта или слова:

NEG Приемник

Функция: Приемник:=-Приемник
изменяет флаги:OF,SF,ZF,AF,PF.
! Если Приемник <> 0, то CF:=1,
иначе CF:=0 и Приемник сохранит значение 0!
! Попытка изменить знак для -128 или -32768
не изменит операнд, но установит OF:=1!

Пример.

MOV AH,0

NEG AH ;CF:=0

MOV AH,-128

NEG AH ;OF:=1

MOV AH,128 ;== -128

NEG AH ;CF:=1

MOV AH,1

NEG AH ;CF:=1

NOP

*

comment *

5. Сравнение байтов или слов:

CMR Приемник,Источник

Функция: По значению разности Приемник-Источник

устанавливаются флаги: OF, SF, ZF, AF, PF, CF
для следующих команд условных переходов.

Пример.

*

```
MOV AX, 0
MG:    CMP AX, -1
        JE  ME
        MOV AX, -1
        JG  MG
ME:    NOP
```

comment ^

4. AAS - коррекция в AL неупакованного кода после вычитания

5. DAS - коррекция в AL упакованного кода после вычитания

КОМАНДЫ УМНОЖЕНИЯ

1. Умножение байтов или слов без знака:

MUL Источник (байт|слово) {Рег|ОП}

Функция MUL для байтов: AX:=AL*Источник,
AH - старшая часть произведения,
AL - младшая часть произведения.
/ 0, если AH=0,
CF:=OF:=<
1, если AH<>0.

флаги SF, ZF, AF, PF после операции не определены .

Пример.

^

```
MOV AL, 255
MOV DL, 4
MUL DL
NOP
```

comment ^

Функция MUL для слов: (DX, AX):=AX*Источник,
DX - старшая часть произведения,
AX - младшая часть произведения.
/ 0, если DX=0,
CF:=OF:=<
1, если DX<>0.

флаги SF, ZF, AF, PF после операции не определены .

Пример.

^

```
MOV AX, 0FFFFH
MOV DX, 4
MUL DX
NOP
```

comment ^

2. Умножение байтов или слов со знаком:

IMUL Источник (байт|слово) {Рег|ОП}

Функция IMUL для байтов: AX:=AL*Источник,
AL - младшая часть произведения,
AH - старшая часть произведения.
/ 0, если AH=знаковое расширение AL
CF:=OF:=< (т.е. результат уместился в AL),
1, если иначе.

флаги SF, ZF, AF, PF после операции не определены .

Пример.

```
MOV AL,-16
MOV DL,4
IMUL DL
NOP
```

comment ^

Функция IMUL для слов: (DX,AX):=AX*Источник,

AX - младшая часть произведения,

DX - старшая часть произведения.

/ 0, если DX=знаковое расширение AX

CF:=OF:=< (т.е. результат уместился в AX),

\ 1, если иначе.

флаги SF,ZF,AF,PF после операции не определены .

Пример.

```
MOV AX,0FFH
MOV DX,4
IMUL DX
NOP
```

comment ^

3. AAM - коррекция в AH и AL неупакованных чисел после умножения

КОМАНДЫ ДЕЛЕНИЯ

1. Деление байтов или слов без знака:

DIV Источник (байт|слово) {Рег|ОП}

Функция DIV для байтов:

AL:= частное от деления AX:Источник,

AH:= остаток от деления AX:Источник.

флаги OF,SF,ZF,AF,PF,CF после операции не определены .

! Если частное не м.б. размещено в AL, т.е. если оно >0FFH,
то происходит прерывание типа 0 ("деление на 0").

Пример.

```
MOV AH,0
MOV AL,240
MOV DL,4
DIV DL ;"деление на 0" при AH>=4
JMP MDIV
```

comment ^

Функция DIV для слов:

AX:= частное от деления (DX,AX):Источник,

DX:= остаток от деления (DX,AX):Источник.

флаги OF,SF,ZF,AF,PF,CF после операции не определены .

! Если частное не м.б. размещено в AX, т.е. если оно >0FFFFH,
то происходит прерывание типа 0 ("деление на 0").

Пример.

```
XDIV DW 4
MDIV: MOV AX,8000H
MOV DX,3 ;"деление на 0" при DX>=4
DIV XDIV
NOP
```

comment ^

2. Деление байтов или слов со знаком:

IDIV Источник (байт|слово) {Рег|ОП}

Функция IDIV для байтов:

AL:= частное от деления AX:Источник,
AH:= остаток от деления AX:Источник.

флаги OF, SF, ZF, AF, PF, CF после операции не определены .

! Если частное не м.б. размещено в AL, т.е. если оно выходит за диапазон -127..+127 (81H..7FH), то происходит прерывание типа 0 ("деление на 0").

Пример.

```
MOV AH,0
MOV AL,240
MOV DL,4
IDIV DL ;"деление на 0" при AH>=2
;! Результат уменьшается до целого числа отбрасыванием
;! дробной части частного.
;! Остаток имеет тот же знак, что и делимое.
MOV AX,3
MOV BX,2
IDIV BL
MOV AX,-3
IDIV BL
JMP MIDIV
```

comment ^

Функция IDIV для слов:

AX:= частное от деления (DX,AX):Источник,
DX:= остаток от деления (DX,AX):Источник.

флаги OF, SF, ZF, AF, PF, CF после операции не определены .

! Если частное не м.б. размещено в AL, т.е. если оно выходит за диапазон -32767..+32767 (8001H..7FFFH), то происходит прерывание типа 0 ("деление на 0").

Пример.

```
XIDIV DW 4
MIDIV: MOV AX,8000H
MOV DX,1 ;"деление на 0" при DX>=2
IDIV XDIV
NOP
```

comment ^

3. AAD - готовит в AL неупакованное десятичное для последующего деления командой DIV.

```
MOV AH,4CH
INT 21H
BEGIN ENDP
SC1 ENDS

SSEG SEGMENT STACK
DB 5 DUP('A')
SSS DB 64 DUP('STACK---')
SSEG ENDS
END BEGIN
```