

**А. В. Карпов**

**ИЗУЧЕНИЕ ГРУППЫ КОМАНД АРИФМЕТИЧЕСКИХ,  
ЛОГИЧЕСКИХ ОПЕРАЦИЙ И ОПЕРАЦИЙ СДВИГА  
МИКРОПРОЦЕССОРА i8086**

**Методические указания к лабораторной работе**

**Волгоград 2006**

УДК 681.325

Изучение группы команд арифметических, логических операций и операций сдвига микропроцессора i8086: Методические указания к лабораторной работе/ Сост. А. В. Карпов. -- Волгоград, 2006.-- 15 с.

Приведено описание лабораторной работы “Изучение группы команд арифметических, логических операций и операций сдвига микропроцессора (МП) i8086” по курсу “Вычислительные машины, системы и сети”, в которой изучаются команды арифметических, логических операций и операций сдвига МП i8086. Издание предназначено для студентов дневной, вечерней и заочной форм обучения специальности 2102.

© А. В. Карпов, 2006

## Цель работы

Целью настоящей работы является изучение группы команд арифметических, логических операций и операций сдвига микропроцессора (МП) i8086.

### 1 Основные сведения

#### Арифметические операции

Команды арифметических операций, выполняемые в МП i8086, представляют собой команды сложения, вычитания, умножения и деления одно- и двухбайтных чисел со знаком и без знака. В операциях над целыми числами со знаком фиксируется переполнение OF. Числа со знаком представляются в дополнительном коде. Все команды арифметических операций изменяют содержимое регистра признаков.

#### **ADD Сложение**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*   \*

Команда: *ADD получатель, источник*

Логика: *получатель=получатель+источник*

ADD складывает операнды *получатель* и *источник* и засылает сумму в операнд *получатель*. Оба операнда могут быть двоичными числами (байтами или словами) со знаком или без знака.

#### **ADC Сложение с переносом**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*   \*

Команда: *ADC получатель, источник*

Логика: *получатель=получатель+источник+CF*

ADC складывает операнды *получатель* и *источник*, прибавляет единицу, если признак переноса CF установлен (CF=1), и засылает сумму в операнд *получатель*. Оба операнда могут быть двоичными числами (байтами или словами) со знаком или без знака.

**Примечание.** Команда ADC полезна при сложении чисел больше 16 бит, т. к. она прибавляет перенос от предыдущей операции.

#### **INC Инкремент**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*   \*

Команда: *INC получатель*

Логика: *получатель=получатель+1*

Эта команда прибавляет к операнду *получатель* единицу. Операнд *получатель*, который может быть словом или байтом, интерпретируется как двоичное число без знака.

### **SUB Вычитание**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*   \*

Команда: SUB *получатель, источник*

Логика: *получатель=получатель-источник*

Команда SUB вычитает операнд *источник* из операнда *получатель* и засылает результат в операнд *получатель*. Оба операнда могут быть двоичными числами (байтами или словами) со знаком или без знака.

**Примечания.** 1. Можно вычитать непосредственный операнд размером в байт из операнда *получатель*, даже если он размером в слово; в этом случае перед вычитанием байт растягивается до 16 битов, заносся в новые биты значение бита знака.

### **SBB Вычитание с заемом**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*   \*

Команда: SBB *получатель, источник*

Логика: *получатель=получатель-источник-CF*

Команда SBB вычитает операнд *источник* из операнда *получатель*, вычитает 1 из результата, если признак переноса установлен (т. е. если CF=1), и засылает результат в операнд *получатель*. Оба операнда могут быть двоичными числами (байтами или словами) со знаком или без знака.

**Примечания.** 1. Команда SBB полезна для вычитания чисел больше 16 битов, поскольку она вычитает заем (находящийся в CF) от предыдущего вычитания.

2. Можно вычитать непосредственный операнд размером в байт из операнда *получатель*, даже если он размером в слово; в этом случае перед вычитанием байт растягивается до 16 битов, заносся в новые биты значение бита знака.

### **DEC Декремент**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   \*   \*   \*   \*   \*

Команда: DEC *получатель*

Логика: *получатель=получатель-1*

Эта команда отнимает от операнда *получатель* единицу. Операнд *получатель*, который может быть словом или байтом, интерпретируется как двоичное число без знака.

### **MUL Умножение без учета знака**

Признаки: O    D    I    T    S    Z    A    P    C  
              \*                   ?   ?   ?   ?   \*

Команда: MUL *множитель*

Логика:  $AH=AL * \text{множитель}$ , если операнд *множитель* - байт  
или

$DH:AH=AH * \text{множитель}$ , если операнд *множитель* - слово.

Эта команда выполняет умножение без учета знака. Если операнд *множитель* является байтом, то MUL умножает операнд *множитель* на регистр AL, засылая произведение в регистр AX. Если же операнд *множитель* является словом, то MUL умножает операнд *множитель* на регистр AX, засылая произведение в регистровую пару DX:AX. Признаки переноса и переполнения CF и OF устанавливаются (=1), если старшая половина результата (т. е. регистр AH для случая, когда операнд *множитель* - байт, и регистр DX, когда операнд *множитель* - слово) содержит какую-либо значащую цифру произведения, иначе они сбрасываются (=0).

### **IMUL Умножение с учетом знака**

Признаки: O D I T S Z A P C  
\* ? ? ? ? \*

Команда: IMUL *множитель*

Логика:  $AX=AL*\text{множитель}$ , если операнд *множитель* - байт  
или

$DX:AX=AX*\text{источник}$ , если операнд *множитель* - слово.

Эта команда выполняет умножение с учетом знака. Если операнд *множитель* является байтом, то IMUL умножает операнд *множитель* на регистр AL, засылая произведение в регистр AX. Если же операнд *множитель* является словом, то IMUL умножает операнд *множитель* на регистр AX, засылая произведение в регистровую пару DX:AX. Признаки переноса и переполнения CF и OF устанавливаются (=1), если старшая половина результата (т.е. регистр AH для случая, когда операнд *множитель* - байт, и регистр DX, когда операнд *множитель* - слово) содержит какую-либо значащую цифру произведения, иначе они сбрасываются (=0).

### **DIV Деление без учета знака**

Признаки: O D I T S Z A P C  
? ? ? ? ? ?

Команда: DIV *делитель*

Логика:  $AL=AX/\text{делитель}$

$AH=\text{остаток}$ , если операнд *делитель* - байт

или

$AX=DX:AX/\text{делитель}$

$DX=\text{остаток}$ , если операнд *делитель* - слово.

Эта команда выполняет деление без учета знака. Если операнд *делитель* является байтом, то DIV делит значение в регистре AX на операнд *делитель*, засылая *частное* в регистр AL и *остаток* в регистр AH. Если же операнд *делитель* является словом, то DIV делит значение в регистровой паре DX:AX на операнд *делитель*, засылая *частное* в регистр AX и *остаток* в регистр DX.

**Примечания.** 1. Если результат слишком велик и не умещается в регистре AL (AX), то генерируется прерывание INTO, и тогда *частное* с *остатком* не определены.

2. Когда генерируется прерывание INTO, то для МП i80286 и i80386 запоминаемое значение CS:IP указывает на неудавшуюся команду DIV. Для МП 8088/8086 CS:IP указывает на команду, следующую за неудавшейся командой DIV.

### **IDIV Деление с учетом знака**

Признаки: O    D    I    T    S    Z    A    P    C  
              ?                               ?   ?   ?   ?   ?

Команда: IDIV *делитель*

Логика: AL=AX/*делитель*

AH=*остаток*, если операнд *делитель* - байт

или

AX=DX:AX/*делитель*

DX=*остаток*, если операнд *делитель* - слово.

Эта команда выполняет деление с учетом знака. Если операнд *делитель* является байтом, то IDIV делит значение в регистре AX на операнд *делитель*, засылая *частное* в регистр AL и *остаток* в регистр AH. Если же операнд *делитель* является словом, то IDIV делит значение в регистровой паре DX:AX на операнд *делитель*, засылая *частное* в регистр AX и *остаток* в регистр DX.

**Примечания.** 1. Если результат слишком велик и не умещается в регистре AL (AX), то генерируется прерывание INTO, и тогда *частное* с *остатком* не определены.

2. Когда генерируется прерывание INTO, то для МП i80286 и i80386 запоминаемое значение CS:IP указывает на неудавшуюся команду (команду IDIV). Для МП 8088/8086 CS:IP указывает на команду, следующую за неудавшейся командой IDIV.

### **CBW Преобразование байта в слово**

Признаки не меняются.

Команда: CBW

Логика: Если (AL<80h) то

AH=0

иначе

AH=FFh

CBW расширяет бит знака регистра AL в регистр AH. Эта команда переводит байтовую величину со знаком в эквивалентное ей слово со знаком.

**Примечание.** Эта команда положит AH равным 0FFh, если бит знака регистра AL (т. е. бит 7) установлен; если же бит 7 регистра AL не установлен, то в AH заносятся нули.

### **CWD Преобразование слова в двойное слово**

Признаки не меняются.

Команда: CWD

Логика: Если (AX<8000h) то

DX=0

иначе

DX=FFFFh



XOR выполняет операцию исключающего ИЛИ побитно над своими операндами и засылает результат в операнд *получатель*. Оба операнда могут быть словами или байтами.

Команда XOR устанавливает каждый бит результата в 1, если в точности один из соответствующих битов операндов равен 1.

### **OR Логическое сложение**

Признаки: O D I T S Z A P C  
0 \* \* ? \* 0

Команда: OR *получатель, источник*

Логика: *получатель=получатель OR источник*

OR выполняет логическое сложение побитно над своими операндами и засылает результат в операнд *получатель*. Оба операнда могут быть словами или байтами.

Команда OR устанавливает каждый бит результата в 1, если хотя бы один из соответствующих битов операндов равен 1.

### **Команды операций сдвига**

В состав команд данной группы входят восемь различных команд сдвига.

#### **RCL Циклический сдвиг влево через CF**

Признаки: O D I T S Z A P C  
\* \* \* \* \*

Команда: RCL *получатель, счетчик*

Команда RCL сдвигает слово или байт, находящийся в операнде *получатель*, влево на число битовых позиций, определяемое вторым операндом *счетчик*. Бит, который вытесняется за левый предел операнда *получатель*, заносится в признак переноса CF, а старое значение CF осуществляет ротацию, в том смысле, что оно заносится в освободившийся крайний правый бит операнда *получатель*. Число таких "битовых ротаций" определяется операндом *счетчик*. Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда в OF заносится результат выполнения операции исключающего ИЛИ, примененной к 2 старшим битам исходного значения операнда *получатель*.

#### **RCR Циклический сдвиг вправо через CF**

Признаки: O D I T S Z A P C  
\* \* \* \* \*

Команда: RCR *получатель, счетчик*

Команда RCR сдвигает слово или байт, находящийся в операнде *получатель*, вправо на число битовых позиций, определяемое вторым операндом *счетчик*. Бит, который вытесняется за правый предел операнда *получатель*, заносится в признак переноса CF, а старое значение CF осуществляет ротацию, в том смысле, что оно заносится в освободившийся крайний левый бит операнда *получатель*. Число таких "битовых ротаций" определяется операндом *счетчик*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда в OF заносится результат выполнения операции исключающего ИЛИ, примененной к 2 старшим битам результата.

### **ROL Циклический сдвиг влево**

Признаки: O D I T S Z A P C  
\* \* \* \* \*

Команда: ROL *получатель*, *счетчик*

Команда ROL сдвигает слово или байт, находящийся в операнде *получатель*, влево на число битовых позиций, определяемое вторым операндом *счетчик*. Бит, который вытесняется за левый предел операнда *получатель*, заносится в него снова с правого края. Значение CF совпадает со значением бита, который последним был вытеснен за левый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда в OF заносится результат выполнения операции исключающего ИЛИ, примененной к 2 старшим битам исходного значения операнда *получатель*.

### **ROR Циклический сдвиг вправо**

Признаки: O D I T S Z A P C  
\* \* \* \* \*

Команда: ROR *получатель*, *счетчик*

Команда ROR сдвигает слово или байт, находящийся в операнде *получатель*, вправо на число битовых позиций, определяемое вторым операндом *счетчик*. Бит, который вытесняется за правый предел операнда *получатель*, заносится в него снова с левого края. Значение CF совпадает со значением бита, который последним был вытеснен за правый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда в OF заносится результат выполнения операции исключающего ИЛИ, примененной к 2 старшим битам результата.

### **SAL Арифметический сдвиг влево**

Признаки: O D I T S Z A P C  
\* \* \* \* \* ? \* \*

Команда: SAL *получатель*, *счетчик*

Команда SAL сдвигает слово или байт, находящийся в операнде *получатель*, влево на число битовых позиций, определяемое вторым операндом *счетчик*. По мере вытеснения битов за левый край операнда *получатель*, справа в освободившиеся биты заносятся нули. Значение CF совпадает со значением бита, который последним был вытеснен за левый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда OF=0, если 2 старших бита исходного значения операнда *получатель* совпадали, иначе OF=1.

### **SAR Арифметический сдвиг вправо**

Признаки: O D I T S Z A P C  
\* \* \* ? \* \*

Команда: SAR *получатель*, *счетчик*

Команда SAR сдвигает слово или байт, находящийся в операнде *получатель*, вправо на число битовых позиций, определяемое вторым операндом *счетчик*. По мере вытеснения битов за правый край операнда *получатель*, слева в освободившиеся биты заносятся биты, совпадающие с битом знака исходного значения операнда *получатель*. Значение CF совпадает со значением бита, который последним был вытеснен за правый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, то OF=0.

### **SHL Логический сдвиг влево**

Признаки: O D I T S Z A P C  
\* \* \* ? \* \*

Команда: SHL *получатель*, *счетчик*

Команда SHL сдвигает слово или байт, находящийся в операнде *получатель*, влево на число битовых позиций, определяемое вторым операндом *счетчик*. По мере вытеснения битов за левый край операнда *получатель*, справа в освободившиеся биты заносятся нули. Значение CF совпадает со значением бита, который последним был вытеснен за левый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, тогда OF=0, если 2 старших бита исходного значения операнда *получатель* совпадали, иначе OF=1.

### **SHR Логический сдвиг вправо**

Признаки: O D I T S Z A P C  
\* \* \* ? \* \*

Команда: SHR *получатель*, *счетчик*

Команда SHR сдвигает слово или байт, находящийся в операнде *получатель*, вправо на число битовых позиций, определяемое вторым операндом *счетчик*. По мере вытеснения битов за правый край операнда *получатель*, слева в освободившиеся биты заносятся нули. Если бит знака сохраняет свое значение, то признак переполнения OF равен 0, иначе он равен 1. Значение CF совпадает со значением бита, который последним был вытеснен за правый край операнда *получатель*.

Если *счетчик* не равен 1, то признак переполнения OF не определен. Если же *счетчик* равен 1, то OF равен значению старшего бита исходного операнда.

**Примечания.** 1. В качестве операнда *счетчик* берется значение в регистре CL. Если необходимо осуществить сдвиг лишь на одну позицию, то можно заменить второй операнд (регистр CL) на число 1.

2. МП i80286 и i80386 ограничивают значение *счетчик* числом 31. Если *счетчик* больше, чем 31, то эти МП используют *счетчик* MOD 32, чтобы получить новый *счетчик* в пределах 0-31. Эта верхняя граница имеет своей целью сократить тот период времени, на который будет задерживаться ответ на прерывание из-за ожидания конца выполнения команды.

3. Несколько команд логических и циклических сдвигов, использующих 1 в качестве операнда *счетчик*, выполняются быстрее и требуют меньшего объема памяти, чем одна команда, использующая CL в качестве операнда *счетчик*.

### **CMP Сравнение**

Признаки: O D I T S Z A P C  
\* \* \* \* \*

Команда: CMP операнд1, операнд2

Логика: Установка признаков в соответствии с результатом операнд1-операнд2

CMP сравнивает два числа, вычитая операнд операнд2 из операнда операнд1, и изменяет значения признаков. CMP не изменяет сами операнды. Операндами могут быть байты или слова.

### **TEST Тест**

Признаки: O D I T S Z A P C  
0 \* \* ? \* 0

Команда: TEST операнд1, операнд2

Логика: Установка признаков в соответствии с результатом операнд1 AND операнд2

CF=0  
OF=0.

Команда TEST выполняет операцию AND над своими операндами и меняет значения признаков. Сами операнды не изменяются.

**Примечание.** Команда TEST полезна при проверке значения конкретного бита.

## **2 Порядок выполнения работы**

2.1 Введите в микроЭВМ программу 1:

*MOV BL, 2*

*@@1:*

*SUB BL, 1*

*JNS @@1* ; Если не минус переход на метку @@1

Исследуйте выполнение команды по шагам. После выполнения каждого шага программы анализируйте содержимое всех программно-доступных регистров. Результаты занесите в табл. 1.

Таблица 1

Мнемокод	Программно-доступные регистры			Регистр признаков FLAGS	Порт 300h (301h)
	AL (AX)	BL(BX)	DX		

Замените в программе строку *SUB BL, 1* на следующую:

*DEC BL*

Проверьте работу программы и результаты занесите в табл. 1.

2.2 Исследуйте работу программ 2--3. После пошагового выполнения программ записывайте содержимое регистров в табл. 1.

Программа 2:

*MOV AX, 10h*

*MOV DX, 10h*

*MOV BX, 2h*

*DIV BX*

*DIV BL*

Программа 3:

*MOV DX, 300h*

*@@1:*

*IN AL, DX*

*CMP AL, 62h*

*JNZ @@1* ; Если не 0, переход на метку *@@1*

2.3 На примере программ 4--7 изучите работу логических операторов NOT, AND, OR, XOR. Результаты работы программ занесите в табл. 2.

Программа 4:

*MOV DX, 300h*

*IN AL, DX*

*NOT AL*

*MOV DX, 301h*

*OUT DX, AL*

Программа 5:

*MOV DX, 300h*

*IN AL, DX*

*MOV BL, AL*

*IN AL, DX*

*AND AL, BL*

*MOV DX, 301h*

*OUT DX, AL*

Программа 6:

*MOV DX, 300h*

*IN AL, DX*

*MOV BL, AL*

*IN AL, DX*

*OR AL, BL*

*MOV DX, 301h*

*OUT DX, AL*

Программа 7:

*MOV DX, 300h*

*IN AL, DX*

*MOV BL, AL*

*IN AL, DX*

```

XOR AL, BL
MOV DX, 301h
OUT DX, AL

```

Таблица 2

Первое введенное число	Второе введенное число	Операция	Результат

Введите программу 8 в микроЭВМ и изучите работу операторов сдвига на примере логического сдвига влево *SHL*.

```

Программа 8:
MOV DX, 300h
IN AL, DX
SHL AL, 1
MOV DX, 301h
OUT DX, AL
MOV BL, AL
MOV CL, 3
SHL AL, CL
ADD AL, BL
OUT DX, AL

```

Исследуйте выполнение программы по командам и результаты занесите в табл. 1.

### 3 Содержание отчета

Отчет по лабораторной работе должен содержать следующие сведения: 1) цель работы; 2) текст программ; 3) протокол полученных результатов.

### 4 Контрольные вопросы

1. Перечислите известные Вам команды (с указанием их формата и логики) МП i8086, относящиеся к группе команд арифметических операций.
2. Перечислите известные Вам команды (с указанием их формата и логики) МП i8086, относящиеся к группе команд логических операций.
3. Перечислите известные Вам команды (с указанием их формата и логики) МП i8086, относящиеся к группе команд операций сдвига.

Составитель Александр Викторович Карпов

ИЗУЧЕНИЕ ГРУППЫ КОМАНД АРИФМЕТИЧЕСКИХ, ЛОГИЧЕСКИХ  
ОПЕРАЦИЙ И ОПЕРАЦИЙ СДВИГА МИКРОПРОЦЕССОРА i8086

Методические указания к лабораторной работе