

А. В. Карпов

**ИЗУЧЕНИЕ ГРУППЫ КОМАНД ПЕРЕДАЧИ УПРАВЛЕНИЯ И
УПРАВЛЕНИЯ СОСТОЯНИЕМ МИКРОПРОЦЕССОРА i8086**

Методические указания к лабораторной работе

Волгоград 2006

УДК 681.325

Изучение группы команд передачи управления и управления состоянием микропроцессора i8086: Методические указания к лабораторной работе/ Сост. А. В. Карпов. -- Волгоград, 2006.-- 17 с.

Приведено описание лабораторной работы “Изучение группы команд передачи управления и управления состоянием микропроцессора (МП) i8086” по курсу “Вычислительные машины, системы и сети”, в которой изучаются команды передачи управления и управления состоянием 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 передачи управления и управления состоянием МП i8086.

1 Основные сведения

Команды передачи управления

CALL Вызов подпрограммы

Признаки не меняются.

Команда: **CALL имя_подпрограммы**

Логика: Если FAR CALL то PUSH CS и CS=dest_seg

PUSH IP

IP=dest_offset.

CALL передает управление подпрограмме, которая может находиться как внутри текущего сегмента (NEAR-proc), так и вне его (FAR-proc). Этим двум типам CALL соответствуют две различные машинные команды, и команда RET возврата из подпрограммы должна соответствовать типу команды CALL (потенциальная возможность несоответствия возникает, когда подпрограмма и команда CALL ассемблируются раздельно).

Примечания. 1. Если подпрограмма находится в другом сегменте, то МП засылает в стек сначала текущее значение CS, затем текущее значение IP (IP указывает на команду, следующую за командой CALL), а затем передает управление в подпрограмму.

2. Если же подпрограмма находится в том же сегменте, то МП засылает в стек сначала текущее значение IP (которое опять же указывает на команду, следующую за командой CALL), а затем передает управление в подпрограмму. CALL также может считать адрес подпрограммы из регистра или из памяти. Эта форма команды CALL называется косвенным вызовом.

JMP Безусловный переход

Признаки не меняются.

Команда: **JMP метка (адрес)**

Условие перехода: переход осуществляется всегда

Команда JMP всегда передает управление в место, определяемое операндом *метка (адрес)*. Переход внутри сегмента может быть задан как операндом типа память, так и через 16-битный регистр. Переход во внешний сегмент может быть задан только через операнд типа "память".

RET Возврат из подпрограммы

Признаки не меняются.

Команда: **RET [значение]**

Логика: POP IP

Если FAR RETURN, то POP CS
и SP=SP+значение (если оно имеется).

Команда RET передает управление из вызванной подпрограммы команде, следовавшей непосредственно за CALL, и делает это следующим образом:

пересылает слово из вершины стека в IP;

если возврат осуществляется во внешний сегмент, то пересылает слово из новой вершины стека в CS;

увеличивает SP на значение операнда *значение*, если оно задано.

Ассемблер сгенерирует возврат во внутренний сегмент, если подпрограмма, содержащая RET, будет обозначена программистом как NEAR, и возврат во внешний сегмент, если как FAR. Операнд *значение* определяет значение, которое надо прибавить к SP, что имеет смысл "освобождения" стека от "лишних" байтов (например, от входных параметров, когда они передаются подпрограмме через стек).

Команды условного перехода

JA Переход если выше

Признаки не меняются.

Команда: JA *метка*

Условие перехода: Если CF=0 и ZF=0, то JMP *метка*

Команда JA используется после команд CMP и SUB и передает управление по *метке*, если первый операнд (который должен быть числом без знака) был больше, чем второй операнд (также без знака).

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNBE -- это та же команда, что и JA. Команду JA следует использовать при сравнении чисел без знака.

JAE Переход если выше или равно

Признаки не меняются.

Команда: JAE *метка*

Условие перехода: Если CF=0, то JMP *метка*

Команда JAE используется после команд CMP или SUB и передает управление по *метке*, если первый операнд был больше или равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNB -- это та же команда, что и JAE. Команду JAE следует использовать при сравнении чисел без знака.

JB Переход если ниже

Признаки не меняются.

Команда: JB *метка*

Условие перехода: Если CF=1, то JMP *метка*

Команда JB используется после команд CMP и SUB и передает управление по *метке*, если первый операнд был меньше, чем второй.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команды JB, JC и JNAE -- являются синонимами одной и той же команды. Команду JB следует использовать при сравнении чисел без знака.

JBE Переход если ниже или равно

Признаки не меняются.

Команда: JBE *метка*

Условие перехода: Если CF=1 или ZF=1, то JMP *метка*

Команда JBE используется после команд CMP и SUB и передает управление по *метке*, если первый операнд был меньше или равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNA -- это та же команда, что и JBE. Команду JBE следует использовать при сравнении чисел без знака.

JC Переход если перенос

Признаки не меняются.

Команда: JC *метка*

Условие перехода: Если CF=1, то JMP *метка*

Команда JC передает управление по *метке*, если признак переноса CF установлен (т. е. равен 1).

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команды JC, JB и JNAE -- являются синонимами одной и той же команды.

JCXZ Переход если CX=0

Признаки не меняются.

Команда: JCXZ *метка*

Условие перехода: Если CX=0, то JMP *метка*

Команда JCXZ передает управление по *метке*, если регистр CX равен 0.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Эта команда обычно применяется в начале цикла, чтобы пропустить тело цикла, когда счетчик (регистр CX) равен нулю.

JE Переход если равно

Признаки не меняются.

Команда: JE *метка*

Условие перехода: Если ZF=1, то JMP *метка*

Команда JE используется после команд CMP и SUB и передает управление по *метке*, если первый операнд был равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JZ -- это та же команда, что и JE.

JG Переход если больше

Признаки не меняются.

Команда: JG *метка*

Условие перехода: Если ZF=0 и SF=OF, то JMP *метка*.

Команда JG используется после команд CMP или SUB и передает управление по *метке*, если первый операнд был больше, чем второй.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNLE -- это та же команда, что и JG. Команду JG следует использовать при сравнении чисел со знаком.

JGE Переход если больше или равно

Признаки не меняются.

Команда: JGE *метка*

Условие перехода: Если SF=OF, то JMP *метка*

Команда JGE используется после команд CMP или SUB и передает управление по *метке*, если первый операнд был больше или равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNL -- это та же команда, что и JGE. Команду JGE следует использовать при сравнении чисел со знаком.

JL Переход если меньше

Признаки не меняются.

Команда: JL *метка*

Условие перехода: Если SF<>OF то JMP *метка*

Команда JL используется после команд CMP или SUB и передает управление по *метке*, если первый операнд был меньше, чем второй.

Примечание. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNGE -- это та же команда, что и JL. Команду JL следует использовать при сравнении чисел со знаком.

JLE Переход если меньше или равно

Признаки не меняются.

Команда: JLE *метка*

Условие перехода: Если SF<>OF или ZF=1, то JMP *метка*

Команда JLE используется после команд CMP или SUB и передает управление по *метке*, если первый операнд был меньше или равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNG -- это та же команда, что и JLE. Команду JLE следует использовать при сравнении чисел со знаком.

JNA Переход если не выше

Признаки не меняются.

Команда: JNA *метка*

JNA синоним JBE. См. описание JBE.

JNAE Переход если не выше и не равно

Признаки не меняются.

Команда: JNAE *метка*

JNAE синоним JB. См. описание JB.

JNB Переход если не ниже

Признаки не меняются.

Команда: JNB *метка*

JNB синоним JAE. См. описание JAE.

JNBE Переход если не ниже и не равно

Признаки не меняются.

Команда: JNBE *метка*

JNBE синоним JA. См. описание JA.

JNC Переход если нет переноса

Признаки не меняются.

Команда: JNC *метка*

Условие перехода: Если CF=0, то JMP *метка*

Команда JNC передает управление по *метке*, если признак переноса CF сброшен (т. е. равен 0).

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

JNE Переход если не равно

Признаки не меняются.

Команда: JNE *метка*

Условие перехода: Если ZF=0, то JMP *метка*

Команда JNE используется после команд CMP и SUB и передает управление по *метке*, если первый операнд не был равен второму.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JNZ -- это та же команда, что и JNE.

JNG Переход если не больше

Признаки не меняются.

Команда: JNG *метка*

JNG синоним JLE. См. описание JLE.

***JNGE* Переход если не больше и не равно**

Признаки не меняются.

Команда: *JNGE* метка

JNGE синоним *JL*. См. описание *JL*.

***JNL* Переход если не меньше**

Признаки не меняются.

Команда: *JNL* метка

JNL синоним *JGE*. См. описание *JGE*.

***JNLE* переход если не меньше и не равно**

Признаки не меняются.

Команда: *JNLE* метка

JNLE синоним *JG*. См. описание *JG*.

***JNO* Переход если нет переполнения**

Признаки не меняются.

Команда: *JNO* метка

Условие перехода: Если $OF=0$, то *JMP* метка

Команда *JNO* передает управление по метке, если признак переполнения *OF* сброшен (т. е. равен 0).

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

***JNP* Переход если нечетно**

Признаки не меняются.

Команда: *JNP* метка

Условие перехода: Если $PF=0$, то *JMP* метка

Команда *JNP* передает управление по метке, если признак четности *PF* сброшен (т. е. равен 0).

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда *JPO* -- это та же команда, что и *JNP*.

***JNS* Переход если положительный результат**

Признаки не меняются.

Команда: *JNS* метка

Условие перехода: Если $SF=0$, то *JMP* метка

Команда *JNS* передает управление по метке, если признак знака *SF* сброшен (т. е. равен 0).

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

JNZ Переход если не ноль

Признаки не меняются.

Команда: JNZ метка

JNZ синоним JNE. См. описание JNE.

JO Переход если есть переполнение

Признаки не меняются.

Команда: JO метка

Условие перехода: Если OF=1, то JMP метка

Команда JO передает управление по метке, если признак переполнения OF установлен (т. е. равен 1).

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

JP Переход если четно

Признаки не меняются.

Команда: JP метка

Условие перехода: Если PF=1, то JMP метка

Команда JP передает управление по метке, если признак четности PF установлен (т. е. равен 1).

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда JPE -- это та же команда, что и JP.

JPE Переход если четно

Признаки не меняются.

Команда: JPE метка

JPE синоним JP. См. описание JP.

JPO Переход если нечетно

Признаки не меняются.

Команда: JPO метка

JPO синоним JNP. См. описание JNP.

JS Переход если отрицательный результат

Признаки не меняются.

Команда: JS метка

Условие перехода: Если SF=1, то JMP метка

Команда JS передает управление по метке, если признак знака SF установлен (т. е. равен 1).

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

JZ Переход если ноль

Признаки не меняются.

Команда: JZ *метка*

JZ синоним JE. См. описание JE.

LOOP Переход по счетчику

Признаки не меняются.

Команда: LOOP *метка*

Логика: CX=CX-1

Если (CX<>0), то JMP *метка*

Команда LOOP уменьшает содержимое регистра CX на 1, затем передает управление по *метке*, если регистр CX не равен 0.

Примечание. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

LOOPE Переход пока равно

Признаки не меняются.

Команда: LOOPE *метка*

Логика: CX=CX-1

Если (CX<>0) и (ZF=1), то JMP *метка*

Команда LOOPE используется после команд CMP или SUB. Она уменьшает содержимое регистра CX на 1, затем передает управление по *метке*, если регистр CX не равен 0 и если первый операнд команд CMP или SUB был равен второму операнду.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда LOOPZ -- это та же команда, что и LOOPE.

LOOPNE Переход пока не равно

Признаки не меняются.

Команда: LOOPNE *метка*

Логика: CX=CX-1

Если (CX<>0) и (ZF=0), то JMP *метка*

Команда LOOPNE используется после команд CMP или SUB. Она уменьшает содержимое регистра CX на 1, затем передает управление по *метке*, если регистр CX не равен 0 и если первый операнд команд CMP или SUB не равен второму операнду.

Примечания. 1. Цель перехода должна лежать в пределах от -128 до 127 байтов от следующей команды.

2. Команда LOOPNZ -- это та же команда, что и LOOPNE.

LOOPNZ Переход пока не ноль

Признаки не меняются.

Команда: LOOPNZ *метка*

LOOPNZ синоним LOOPNE. См. описание LOOPNE.

LOOPZ Переход пока ноль

Признаки не меняются.

Команда: LOOPZ *метка*

CLD Сброс признака

направления

Признаки: O D I T S Z A P C
0

Команда: CLD

Логика: DF=0 (Разрешает инкремент в командах обработки строк)

CLD сбрасывает (устанавливает равным 0) признак направления.

Другие признаки не меняются. Сброшенный признак направления влечет увеличение SI и DI на единицу в командах обработки строк.

Примечание. Команды обработки строк увеличивают SI и DI на единицу, когда DF=0.

CLI Сброс признака разрешения прерывания

Признаки: O D I T S Z A P C
0

Команда: CLI

Логика: IF=0.

CLI сбрасывает (устанавливает равным 0) признак разрешения прерывания, вследствие чего МП не распознает замаскированные прерывания. Другие признаки не меняются.

CMC Инвертирование признака переноса

Признаки: O D I T S Z A P C
*

Команда: CMC

Логика: CF=NOT CF

CMC меняет текущее значение признака переноса на противоположное.

ESC Выборка кода операции и операнда

Признаки не меняются.

Команда: ESC код_операции, источник

Команда ESC используется для передачи управления от МП внешнему процессору (i8087). В ответ на ESC МП выбирает код_операции для внешнего процессора и код операнда источник и помещает их в шину. Внешний процессор поджидает команду ESC и выполняет команду, размещенную в шине, используя исполнительный адрес источник.

Примечание. В целях синхронизации с внешним процессором программист должен перед каждой командой ESC помещать команду WAIT. МП 80288 и 80386 обладают средствами автоматической синхронизации команд, поэтому для них команды WAIT можно опустить.

HLT Останов

Признаки не меняются.

Команда: HLT

Эта команда производит останов МП и переводит его в состояние ожидания сигнала сброса или сигнала немаскируемого прерывания.

LOCK Блокирование шины BUS

Признаки не меняются.

Команда: LOCK

LOCK это префикс, который может предшествовать любой команде. LOCK заставляет МП выработать сигнал блокировки шины на время выполнения последующей команды. Использование сигнала блокировки делает шину недоступной для любого внешнего устройства или события, включая прерывания и передачу данных.

Примечания. 1. Команда была предусмотрена для поддержки мультипроцессорных систем с разделенными ресурсами.

2. Эту команду следует использовать только с целью предотвращения прерывания операций по пересылке данных. Поэтому ее следует употреблять только перед командами XCHG, MOV и др.

NOP Нет операции

Признаки не меняются.

Команда: NOP

Логика: нет

Команда NOP является пустым оператором. Она используется в целях подгонки времени, для выравнивания памяти и как "держатель места".

STC Установка признака переноса

Признаки: O D I T S Z A P C
1

Команда: STC

Логика: CF=1

STC устанавливает признак переноса CF в единицу. Другие признаки не меняются.

STD Установка признака направления

Признаки: O D I T S Z A P C
1

Команда: STD

Логика: DF=1 (декремент в командах обработки строк)

STD устанавливает признак направления DF в единицу. Другие признаки не меняются.

Примечание. Команды обработки строк уменьшают SI и DI на единицу, когда DF=1.


```
@@2:
    MOV DX, 301h
    OUT DX, AL
    Программа 3:
    MOV AL, 3h
```

```
@@1:
    SUB AL, 2
    JS  @@2
    JMP @@1
```

```
@@2:
    MOV AL, 3h
```

```
@@3:
    SUB AL, 2
    JNS @@3
```

2.3 На примере программы 4 изучите работу команды организации цикла (переход по счетчику). Исследуйте выполнение программы по командам. Результаты занесите в табл. 1.

```
Программа 4:
    MOV DX, 300h
    IN  AL, DX
    MOV CX, 3
```

```
@@1:
    ADD AL, 1
    LOOP @@1
    MOV DX, 301h
    OUT DX, AL
```

2.4 На основании программы 5 изучите методы принудительной установки и сброса разных признаков. Исследуйте, как меняются при этом признаки, и результаты работы каждой команды занесите в табл. 2.

Программа 5:

```
STC
CLI
CLC
STI
CMC
```

Таблица 2

Состояние признаков до выполнения команды	Команда	Состояние признаков после выполнения команды
	STC CLI CLC STI CMC	

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 передачи управления.
2. Перечислите известные Вам команды (с указанием их формата и логики) МП i8086, относящиеся к группе команд управления состоянием МП.

Составитель Александр Викторович Карпов

ИЗУЧЕНИЕ ГРУППЫ КОМАНД ПЕРЕДАЧИ УПРАВЛЕНИЯ И
УПРАВЛЕНИЯ СОСТОЯНИЕМ МИКРОПРОЦЕССОРА i8086

Методические указания к лабораторной работе