Содержание

1.Введение

2.Режимы процессора

2.1.Реальный режим

2.2.Защищенный режим

3.Типы данных

4.Регистры

4.1.Регистры общего назначения

4.2.Регистр системных флагов

4.3.Регистры сегментов

4.4.Регистры управления сегментированной памятью

4.5.Указатель команд

4.6.Регистры управления

4.7.Регистры отладки

4.8.Буфер ассоциативной трансляции

5.Система команд

5.1.Формат команд

5.2.Описание обозначения

5.3.Список команд

1. Введение

МП 80386 вышел на рынок с уникальным преимуществом. Он является единственным 32-разрядным МП, для которого пригодно существующее прикладное програмное обеспечение, написанное для МП предыдущих моделей от 8086/88 до 80286. Любые программы, написанные для этих МП могут выполняться на 80386 без всяких исправлений и дополнений, лишь только с увеличением скорости их выполнения. Это свойство МП называется совместимостью снизу вверх. Также, дополнительными преимуществами этого МП являются многозадачность, встроенное управление памятью, виртуальная память с разделением на страницы, защита программ и большое адресное пространство. Аппаратная совместимость с предыдущими моделями сохранена посредством динамического изменения разряд­ности магистрали.

МП 80386 выполнен на основе технологии CHMOS III фирмы Intel, которая вобрала в себя быстродействие технологии HMOS (МДП высокой плотности) и малое потребление мощности техноло­гии CMOS (КМДП). МП 80386 предусматривает переключение прог­рамм, выполняемых под управлением различных операционных систем, таких как MS-DOS и UNIX. Это свойство позволяет разра­ботчикам программ включать стандартное прикладное программное обеспечение для 16-разрядных МП непосредственно в 32-разрядную систему. Процессор определяет адресное пространство как один или несколько сегментов памяти любого размера в диапазоне от 1 байт до 4 Гбайт. Эти сегменты могут быть индивидуально защище­ны уровнями привилегий и таким образом избирательно разде­ляться различными задачами.

2. Режимы процессора

Для более полного понятия системы команд МП 80386 необхо­димо предварительно описать общую схему его работы и архитек­туру.

В данном реферате не раскрывается более подробно значения некоторых специфических слов и понятий, считая, что читатель предварительно ознакомился с МП 8086 и МП 80286 и имеет представление о их работе и архитектуре. Описываются только те функции МП 80386, которые отсутствуют или изменены в предыду­щих моделях МП.

МП 80386 имеет два режима работы: режим реальных адресов, называемый реальным режимом, и защищенный режим.

2.1. Реальный режим

При подаче сигнала сброса или при включении питания уста­навливается реальный режим, причем МП 80386 работает как очень быстрый МП 8086, но, по желанию программиста, с 32-разрядным расширением. В реальном режиме МП 80386 имеет такую же базовую архитектуру, что и МП 8086, но обеспечивает доступ к 32-раз­рядным регистрам. Механизм адресации, размеры памяти и обра­ботка прерываний МП 8086 полностью совпадают с аналогичными функциями МП 80386 в реальном режиме.

Единственным способом выхода из реального режима является явное переключение в защищенный режим. В защищенный режим МП 80386 входит при установке бита включения защиты (РЕ) в нуле­вом регистре управления (CR0) с помощью команды пересылки (MOV to CR0). Для совместимости с МП 80286 с целью установки бита РЕ может быть также использована команда загрузки слова состо­яния машины LMSW. Процессор повторно входит в реальный режим в том случае, если программа командой пересылки сбрасывает бит РЕ регистра CR0.

2.2. Защищенный режим

Полные возможности МП 80386 раскрываются в защищенном ре­жиме. Программы могут исполнять переключение между процессами с целью входа в задачи, предназначенные для режима виртуально­го МП 8086. Каждая такая задача проявляет себя в семантике МП 8086 (т.е. в отношениях между символами и приписываемыми им значениями независимо от интерпретирующего их оборудования). Это позволяет выполнять на МП 80386 програмное обеспечение для МП 8086 - прикладную программу или целую операционную систему. В то же время задачи для виртуального МП 8086 изолированы и защищены как друг от друга, так и от главной операционной системы МП 80386.

3. Типы данных

МП 80386 подразделяет память на 8-разрядные байты, 16-разрядные слова и 32-разрядные двойные слова. Дополнительно МП 80386 поддерживает также следующие дополнительные типы дан­ных.

Неупакованный двоично-десятичный тип - распакованное бай­товое представление десятичной цифры от 0 до 9. Распакованные десятичные числа хранятся как беззнаковые байтовые значения по одной цифре в каждом байте. Значение цифры определяется млад­шим полубайтом. Старший полубайт должен быть равным нулю при делении и умножении, но может иметь любое значение при вычита­нии или сложении.

Упакованный двоично-десятичный тип - упакованное байтовое представление двух десятичных цифр от 0 до 9. Каждая цифра хранится в своем полубайте. Цифра в старшем полубайте является более значимой. Диапазон упакованного десятичного байта составляет от 0 до 99.

Битовое поле - непрерывная последовательность битов, в которой каждый бит рассматривается как независимая переменная. Битовое поле может начинаться с любого бита любого байта и мо­жет быть длиной до 32 бит.

Битовая строка - подобно битовому полю, битовая строка является непрерывной последовательностью битов. Битовая строка может начинаться с любого бита любого байта и иметь длину до (2Е32-1) бит.

Ближний указатель - 32-разрядный логический адрес, кото­рый представляет собой относительный адрес внутри сегмента. Ближние указатели используются как в сплошной, так и в сегмен­тированной модели памяти.

Дальний указатель - 48-разрядный логический адрес из двух компонентов: 16-разрядного сегмента и 32-разрядного относи­тельного адреса. Дальние указатели используются программистами тольком в том случае, когда конструкторы системы выбирают сег­ментированную организацию памяти.

Целый тип - знаковое двоичное значение, содержащееся в 32-разрядном двойном слове, 16-разрядном слове или 8-разрядном байте. Все операции предполагают представление чисел в допол­нительном коде. Знаковый бит расположен в бите 7 в байте, в бите 15 в слове и в бите 31 в двойном слове. Он равен нулю для положительных чисел и единице для отрицательных. Поскольку этот старший бит используется как знаковый, то 8-разрядное (байт) целое число может изменяться в диапазоне от -128 до +127, 16-разрядное (слово) целое число в диапазоне от -32768 до +32767, а 32-разрядное (двойное слово) целое число в диапа­зоне от -2Е31 до +2Е31-1. Нулевое значение имеет положительный знак.

Обычный тип - беззнаковое двоичное значение, содержащееся в 32-разрядном двойном слове, 16-разрядном слове или 8-разряд­ном байте. Все биты определяют величину числа.

Строка - непрерывная последовательность байтов, слов или двойных слов. Строка может содержать от 0 до 2Е32 -1 байтов или 4 Гбайтов.

4. Регистры

Регистр является устройством временного хранения данных и используется с целью облегчения арифметических, логических и пересылочных операций. Регистры МП 80386 являются расширением регистров прежних МП 8086, 80186, 80286. Все 16-разрядные ре­гистры МП предыдущих поколений содержатся внутри 32-разрядной архитектуры.

Микропроцессор 80386 включает шесть непосредственно доступных программисту регистров селекторов сегментов, которые содержат указатели сегментов. Значения этих селекторов могут быть загружены при исполнении программы и являются специфичны­ми для задачи. Это значит, что регистры сегментов перезагружа­ются автоматически при переключении МП 80386 на другую задачу. За регистрами селекторов сегментов стоят реальные регистры кэш-памяти сегментов, которые содержат описания сегментов, указываемых селектором. Это сделано на аппаратном уровне для того, чтобы избежать дополнительной выборки из памяти в слу­чае, когда требуется описание сегмента.

4.1. Регистры общего назначения

Восемь регистров общего назначения имеют длину в 32 бит и содержат адреса или данные. Они поддерживают операнды-данные длиной 1, 8, 16, 32 и 64 бит; битовые поля от 1 до 32 бит: операнды-адреса длиной 16 и 32 бит. Эти регистры называются EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP.

Доступ к младшим 16 бит этих регистров выполняется неза­висимо. Это делается в большинстве ассемблеров при использова­нии 16-разрядных имен регистров: AX, BX, CX, DX, SI, DI, BP, SP.

4.2. Регистр системных флагов

Регистр EFLAGS управляет вводом-выводом, маскируемыми прерываниями, отладкой, переключением задач и включением исполнения в режиме виртуального МП 8086 в защищенной многоза­дачной среде - все это в дополнение к флагам состояния, кото­рые отражают результат исполнения команды. Младшие 16 бит его представляют собой 16-разрядный регистр флагов и состояния МП 80286, называемый FLAGS, который наиболее полезен при исполне­нии программ для МП 8086 и 80286.

4.2. Регистры сегментов

Шесть 16-разрядных регистров содержат значения селекторов сегментов, которые указывают на текущие адресуемые сегменты памяти. Ниже перечислены эти регистры.

Регистр сегмента программы (CS) - указывает на сегмент, который содержит текущую последовательность исполняемых ко­манд. Процессор выбирает все команды из этого сегмента, используя содержимое счетчика команд как относительный адрес. Содержимое CS изменяется в результате выполнения внутрисег­ментных команд управления потоком, прерываний и исключений. Он не может быть загружен явным способом.

Регистр сегмента стека (SS). Вызовы подпрограмм, записи параметров и активизация процедур обычно требуют области памя­ти, резервируемой под стек. Все операции со стеком используют регистр SS при обращении к стеку. В отличие от регистра CS ре­гистр SS может быть загружен явно с помощью команды программы.

Остальные четыре регистра являются регистрами сегментов данных (DS, ES, FS, GS), каждый из которых адресуется текущей исполняемой программой. Доступ к четырем раздельным областям данных имеет целью повысить эффективность программ, позволяя им обращаться к различным типам структур данных. Содержимое этих регистров может быть заменено под управлением программы.

При использовании регистров сегментов МП 80386 с каждым выбранным сегментом связывает базовый адрес. При адресации единицы данных внутри сегмента к базовому адресу сегмента до­бавляется 32-разрядный относительный адрес. Если сегмент выб­ран загрузкой селектора сегмента в регистр сегмента, то коман­дам манипуляции данными нужен только этот относительный адрес.

4.3. Регистры управления сегментированной памятью

Регистр таблицы глобальных дескрипторов (GDTR). Содержит 32-разрядный линейный адрес и 16-разрядную границу таблицы глобальных дескрипторов.

Регистр таблицы локальных дескрипторов (LDTR). Содержит 16-разрядный селектор для таблицы локальных дескрипторов. Так как эта таблица является специфичным для задачи сегментом, то она определяется значением селектора, хранимым в регистрах системного сегмента. Регистр дескриптора сегмента, связанный с этой таблицей, програмно недоступен.

Регистр таблицы дескрипторов прерываний (IDTR). Указывает на таблицу точек входа в программы обработки прерываний. Ре­гистр содержит 32-разрядный линейный базовый адрес и 16-раз­рядную границу таблицы дескрипторов прерываний (IDT).

Регистр задачи (TR). Указывает на информацию, необходимую процессору для определения текущей задачи. Регистр TR содержит 16-разрядный селектор дескриптора сегмента состояния задачи. Поскольку этот сегмент специфичен для задачи, то он определя­ется значениями селекторов, хранящихся в регистрах системного сегмента. Заметим, что с каждым регистром системных сегментов связан программно недоступный регистр дескриптора сегмента.

4.5. Указатель команд

Расширенный указатель команд (EIP) является 32-разрядным регистром. Он содержит относительный адрес следующей команды, подлежащей выполнению. Относительный адрес отсчитывается от начала сегмента текущей программы. Указатель команд не­посредственно не доступен программисту, но он управляется явно командами управления потоком, прерываниями и исключениями.

Младшие 16 бит регистра EIP называются IP и могут быть использованы процессором независимо. Это свойство полезно при исполнении команд МП 8086 и 80286, которые имеют только ре­гистр IP.

4.6. Регистры управления

МП 80386 имеет три 32-разрядных регистра управления (CR0, CR2 и CR3, а CR1 зарезервирован фирмой Intel), в которых хра­нятся состояния машины или глобальные состояния. Глобальное состояние - это такое состояние, к которому может получить доступ любой из логических блоков системы или которое управля­ет этими блоками. Вместе с регистрами системных адресов эти регистры хранят информацию о состоянии машины, которая влияет на все задачи в системе. Для доступа к регистрам управления определены команды их загрузки и сохранности содержимого.

Системным программистам регистры управления доступны только через варианты команды MOV, которые позволяют их загру­жать или сохранять в регистрах общего назначения.

4.7. Регистры отладки

Шесть доступных программисту регистров отладки (DR0-DR3, DR6 и DR7) расширяют возможности отладки в МП 80386, они уста­навливают точки останова по данным и позволяют устанавливать точки останова по командам без модификации сегментов программ. Регистры DR0-DR3 предназначены для четырех линейных точек останова. Регистры DR4 и DR5 зарезервированы фирмой Intel для будущих разработок. Регистр DR6 показывает текущее состояние точек останова, а регистр DR7 используется для установки точек останова.

4.8. Буфер ассоциативной трансляции

Буфер ассоциативной трансляции (TLB) - это кэш-память, используемая для трансляции линейных адресов в физические.

Механизм проверки TLB является уникальным для МП 80386 и может быть не реализован в том же виде в будущих процессорах. Программы, которые используют этот механизм в его нынешнем ви­де, могут оказаться несовместимыми с будущими процессорами.

5. Система команд

5.1. Формат команд

Команды МП 80386 состоят из отдельных элементов и могут иметь различные форматы. Из всех описанных ниже элементов только один (код операции, Коп) обязательно присутствует в лю­бой команде. Остальные элементы могут отсутствовать, что опре­деляется характером операции, а также местоположением и типом операндов.

Команды состоят из следующих элементов: необязательных префиксов; одного или двух байтов кодов операции; возможно - описателя адреса, который включает байт Mod R/M и байт масшта­ба, индекса и базы; смещения - если требуется; поля не­посредственных данных - если требуется.

Префиксы - один или несколько байтов, предшествующих ко­манде и модифицирующих операцию этой команды. Имеется 4 типа префиксов.

1. Повторение - используется с командами обработки строк; заставляет команду воздействовать на каждый элемент строки.

2. Размер операнда - переключает разрядность операндов, устанавливая их 32-разрядными или 16-разрядными.

3. Размер адреса - переключает разрядность адреса, опре­деляя образование 32-разрядных или 16-разрядных адресов.

4. Замена сегмента - в явной форме указывает, какой сег­ментный регистр должна использовать команда. Префикс отменяет действующий по умолчанию выбор сегментного регистра, обычно осуществляемый МП 80386 при выполнении этой команды.

Код операции (Коп) - описывает операцию, выполняемую ко­мандой. Некоторым командам присущи несколько кодов операций, каждый из которых описывает определенный вариант операции.

Описатель регистра - в команде могут быть описаны один или два регистра в качестве операндов. Описатель регистра мо­жет присутствовать как в байте кода операции, так и в байте описателя режима адресации.

Описатель режима адресации. Этот элемент, если он присутствует, описывает, является ли операнд содержимым ре­гистра или ячейки памяти. Если операнд находится в памяти, описатель режима указывает, надо ли использовать смещение, ин­дексный регистр, регистр базы и масштабирование.

Байты MOD R/M и SIB. Большинство команд, ссылающихся на операнд, находящийся в памяти, содержат после байта основного кода операции еще байт формы адресации. Этот байт описывает используемую форму адреса. Определенные значения кода поля MOD R/M указывают на наличие второго адресного байта SIB.

Байты MOD R/M и SIB содержат следующую информацию:

- тип индексации или номер регистра, используемого в ко­манде;

- используемый регистр или дополнительную информацию о выборе команды;

- информацию о базе, индексе и масштабе;

Смещение. Если описатель режима адресации указывает, что при вычислении адреса операнда будет использовано смещение, в состав кода команды включается поле смещения. Смещение представляет собой 8-, 16- или 32-разрядное целое число со знаком. 8-разрядная форма используется в тех случаях, когда значение смещения невелико.

Непосредственный операнд (данное). Если этот элемент присутствует, он представляет значение операнда команды. Не­посредственные операнды могут быть 8-, 16- или 32-разрядными. В случаях когда 8-разрядный непосредственный операнд использу­ется в команде вместе с 16- или 32-разрядным операндом, про­цессор автоматически увеличивает размер 8-разрядного операнда путем расширения его знакового разряда.

5.2. Описание обозначения

+rb, +rw, +rd Код регистра (от 0 до 7), который складыва­ется с указанным слева от знака плюс шестнадцетиричным числом (байтом) с лбразованием байта кода операции. Регистрам присво­ены следующие коды :

rb rw rd

AL=0 AX=0 EAX=0

CL=1 CX=1 ECX=1

DL=2 DX=2 EDX=2

BL=3 BX=3 EBX=3

AH=4 SP=4 ESP=4

CH=5 BP=5 EBP=5

DH=6 SI=6 ESI=6

BH=7 DI=7 EDI=7

/цифра Цифра, стоящая справа от косой черты, имеет, вооб­ще говоря, значение от 0 до 7. Она показывает, что в байте MOD R/M указывается только один операнд r/m (регистр/память). Сама цифра помещается в поле reg и образует расширение кода опера­ции.

/r Обозначение показывает, что байт MOD R/M содержит два операнда (reg и r/m).

cb, cw, cd, cp Величина размером 1 байт (cb), 2 байта (cw), 4 байта (cd) или 6 байт (cp) следует за кодом операции и определяет относительный адрес, а также, возможно новое значе­ние программного сегмента.

ib, iw, id Непосредственный операнд размером 1 байт (id), 2 байта (iw) или 4 байта (id) следует за байтами кода опера­ции, MOD R/M или SIB. Код операции указывает, является ли опе­ранд знаковой величиной.

imm8 Непосредственный операнд размером 1 байт. Величина imm8 является знаковой величиной между -128 и +127 включитель­но. В командах, где размер второго операнда составляет слово или двойное слово, величина imm8 расширяется до слова или двойного слова. Старшие байты расширенной величины заполняются старшим битом непосредственного операнда.

imm16 Непосредственный операнд размером 1 слово. Исполь­зуется в командах с атрибутом размера операнда 16 разрядов. Может иметь значение от -32768 до +32767 включительно.

imm32 Непосредственный операнд размером 1 двойное слово. Используется в командах с атрибутом размера операнда 32 разря­да. Может иметь значение от -2147483648 до +2147483647 включи­тельно.

m8 Байт памяти. Адресуется через регистры DS:SI или ES:DI.

m16 Слово памяти.

m32 Двойное слово памяти.

moffs8, moffs16, moffs32 Относительный адрес. Простая пе­ременная типа BYTE, WORD или DWORD, используемая некоторыми вариантами команды MOV. Фактический адрес записывается в виде простого смещения относительно базы сегмента. Число при аббре­виатуре "moffs" указывает разрядность смещения определяемую аттрибутом размера адреса в команде. Байт MOD R/M в команде не используется.

ptr16:16, ptr16:32 Указатель FAR (дальний). Обычно он на­ходится в другом программном сегменте по отношению к команде. Обозначение 16:16 говорит о том, что указатель состоит из двух частей. Величина слева от двоеточия - это смещение в сегменте приемнике. Величина справа от двоеточия - это 16-разрядный се­лектор или величина, предназначенная для регистра программного сегмента. Если атрибут размера операнда команды равен 16, используйте 16:16. Для 32-разрядного атрибута используйте 16:32.

r8 Один из байтовых регистров: AL, CL, DL, BL, AH, CH, DH, BH.

r16 Один из однословных регистров: AX, CX, DX, BX, SP, BP, SI, DI.

r32 Один из двухсловных регистров: EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI.

rel8 Смещение для перехода в диапазоне от 128 байтов пе­ред концом команды до 127 байтов после конца команды.

rel16, rel32 Смещение для перехода в том же программном сегменте, что и ассемблируемая команда, rel16 относится к ко­мандам с атрибутом размера операнда 16 разрядов. rel32 от­носится к командам с атрибутом размера операнда 32 разряда.

r/m8, r/m16, r/m32 Соответственно одно-, двух- и четырех­байтовый операнд. Представляет собой содержимое либо памяти, либо регистра.

rrr Когда обозначение rrr появляется в колонке двоичного эквивалента кода операции, оно указывает конкретный адресуемый регистр.

000 = AX/EAX 100 = SP/ESP

001 = CX/ECX 101 = BP/EBP

010 = DX/EDX 110 = SI/ESI

011 = BX/EBX 111 = DI/EDI

Sreg Регистр сегмента. Кодирование сегментных регистров в поле reg: ES=0, CS=1, DS=3, FS=4, GS=5.

5.3. Список команд

AAA ASCII-коррекция после сложения.

Команда AAA преобразует содержимое регистра AL в неупако­ванное десятичное число и сбрасывает его старшие четыре разря­да. Она должна всегда следовать за сложением двух неупакован­ных десятичных операндов в AL. Если возникает перенос, уста­навливается флаг CF и происходит инкремент в регистре AH.

AAD ASCII-коррекция регистра AX перед делением.

Команда AAD преобразует число в регистрах AH и AL, подго­тавливая деления двух неупакованных десятичных операндов так, чтобы полученное в результате деления частное было бы правиль­ным неупакованным десятичным числом. В регистре AH должна на­ходится старшая цифра, в AL - младшая. Команда AAD корректиру­ет число и помещает результат в AL. Регистр AH содержит 0.

AAM ASCII-коррекция в регистре AX после умножения. Команда AAM корректирует результат умножения двух неупа-

кованных десятичных чисел. Эта команда должна всегда следовать за умножением двух десятичных цифр с целью образования пра­вильного десятичного результата. Старшая цифра помещается в регистр AH, младшая - в AL.

AAS ASCII-коррекция регистра AL после вычитания.

Команда AAS преобразует содержимое регистра AL в неупако­ванную десятичную цифру и заполняет нулями старшие четыре раз­ряда. Эта команда должна всегда следовать за вычитанием одного неупакованного десятичного операнда из другого в AL. Флаг CF устанавливается, а при наличии займа выполняется декремент в регистре AH.

ADC Сложение с переносом целых чисел.

Команда ADC суммирует операнды, прибавляет 1, если уста­новлен флаг CF, и помещает результат по адресу назначения. Ес­ли флаг CF сброшен, команда ADC выполняет ту же операцию, что и ADD. Комбинация команды ADD и нескольких команд ADC позволя­ет складывать числа, содержащие более 32 разрядов.

ADD Сложение целых чисел.

В результате выполнения команды операнд-приемник заменя­ется суммой обоих операндов (источника и приемника). При нали­чии переполнения устанавливается флаг CF.

AND Логическое И.

Команда AND используется для сброса битов, указанных пользователем, например бита четности во входном потоке кодов ASCII от терминала. Будучи использована совместно с командой сравнения, команда AND позволяет убедиться, что указанные биты установлены.

ARPL Коррекция в селекторе уровня привилегий инициато­ра запроса.

Команда ARPL используется системными программами для обеспечения правильности передачи подпрограмме селекторов в качестве параметров. Селекторы не должны требовать больше при­вилегий, чем разрешено инициатору запроса.

В команде ARPL используются два операнда. первый предс­тавляет собой 16-разрядный регистр или слово памяти, содержа­щее значение селектора. В качестве второго операнда обычно выступает регистр, содержащий значение селектора программного сегмента CS инициатора запроса.

BOUND Проверка индекса массива на выход за границы. Команда BOUND проверяет, лежит ли знаковая величина, на-

ходящаяся в заданном регистре, внутри заданных границ. Если значение этой величины меньше нижней границы или больше верх­ней, возникает прерывание 5. Каждое из значений верхней и ниж­ней границ может быть словом или двойным словом.

BSF прямое сканирование битов.

Эта команда сканирует слово или двойное слово в поисках бита, равного 1, и заносит в регистр номер первого установлен­ного бита. Сканируемая строка может находиться как в регистре, так и в памяти. Если все слово равно 0, т.е. в нем нет единич­ных битов, устанавливается флаг ZF. Если единичный бит найден, флаг ZF сбрасывается.

BSR Обратное сканирование битов.

Эта команда действует также как и BSF, но начинает скани­рование со старшего бита. Это новая команда, специфичная для МП 80386.

BT Проверка бита.

Команда BT служит для определения того, установлен или нет определенный бит в битовом массиве. Значение проверяемого бита копируется во флаг CF. Это новая команда, специфичная для МП 80386.

BTC Проверка и инверсия бита.

Команда BTC проверяет указанный бит, копирует его в CF и инвертирует найденный бит. Это новая команда, специфичная для МП 80386.

BTR Проверка и сброс бита.

Команда BTR проверяет указанный бит, копирует его в CF и сбрасывает найденный бит. Это новая команда, специфичная для МП 80386.

BTS Проверка и установка бита.

Команда BTS проверяет указанный бит, копирует его в CF и устанавливает в найденном бите значение 1. Это новая команда, специфичная для МП 80386.

CALL Вызов процедуры.

Команда CALL передает управление из одной точки программ­ного сегмента в другую. Эти точки могут располагаться в одном и том же программном сегменте (ближний переход) или принадле­жать разным сегментам (дальний переход). Перед собственно пе­редачей управления команда CALL сохраняет в стеке адрес следу­ющей за CALL команды и текущее содержимое регистра EIP.

Команды CALL могут быть относительными, прямыми и косвен­ными. В косвенной команде абсолютный адрес перехода указывает­ся одним из двух способов: (1) МП извлекает адрес приемника из ячейки памяти, определенной в команде; (2) программа переходит в точку адрес которой указан в одном из регистров общего наз­начения.

CBW Преобразование байта в слово.

Эта команда расширяет значение бита знака в старшую часть более длинного регистра так, чтобы арифметические операции над содержимым этого регистра давали правильные результаты.

CWD Преобразование слова в двойное слово.

Действие команды аналогично команде CBW.

CLC Сброс флага переноса.

Команда сбрасывает флаг CF.

CLD Сброс флага направления.

Команда сбрасывает флаг DF. Если флаг DF сброшен, автома­тическая индексация будет выполняться с инкрементом. Автомати­ческая индексация используется командами обработки строк.

CLI Сброс флага прерываний.

Если текущий уровень привилегий по крайней мере столь же высок, как уровень привилегий ввода-вывода, команда сбрасывает флаг прерываний.

CLTS Сброс флага переключения задачи в управляющем ре­гистре 0.

Команда сбрасывает флаг TS в CR0. В МП 80386 флаг TS ус­танавливается каждый раз при переключении задачи. Команда ис­пользуется в системном программировании. Она представляет со­бой привилегированную команду, выполняемую только на нулевом уровне привилегий.

CMC Инвертирование флага переноса.

Команда инвертирует флаг переноса CF.

CMP Сравнение.

Команда выполняет вычитание операнда-источника из операн­да-приемника. В соответствии с результатом вычитания команда устанавливает состояние флагов, но не изменяет сами операнды.

CMPS Сравнение строковых операндов.

Ассемблер всегда транслирует команду CMPS как одну из CMPSB, CMPSW, CMPSD. Эти команды предназначены только для операций над строками. Они сравнивают по одному элементу каж­дой строки, причем элементами могут быть байт, слово или двой­ное слово. Элементы строк адресуются через регистры ESI и EDI. После каждой строковой операции ESI и/или EDI автоматически получают положительное или отрицательное приращение и указыва­ют на следующие элементы строк. Если DF=0, в индексных регист­рах осуществляется инкремент, если DF=1 - декремент. В зависи­мости от результата вычитания строкового элемента по адресу ES:EDI из строкового элемента по адресу DS:ESI устанавливаются флаги результата. Если команда модифицирована с помощью пре­фиксов, МП выполняет сравнение текущего элемента строки с со­держимым регистра EAX или его части (AL или AX).

CMPSB Сравнение строковых байтов.

CMPSW Сравнение строковых слов.

CMPSD Сравнение строковых двойных слов.

CDQ Преобразование двойного слова в четверное.

CWDE Преобразование слова в двойное слово с расширением.

DAA Десятичная коррекция в регистре AL после сложения. Команда корректирует результат сложения двух правильных

упакованных десятичных операндов в регистре AL. Эта команда должна всегда следовать за сложением двух пар упакованных де­сятичных чисел, чтобы получить в результате пару правильных упакованных десятичных цифр.

DAS Десятичная коррекция в регистре AL после вычитания. Команда DAS аналогична команде DAA за исключением, что

коррекция выполняется путем вычитания 6 из полубайтов регистра AL вместо прибавления 6.

DEC Декремент на 1.

Команда вычитает 1 из операнда приемника. Состояние флага CF не изменяется.

DIV Деление целых чисел без знака.

Команда выполняет деление числа без знака в аккумуляторе на операнд-источник. Размер делимого в битах в два раза больше размера делителя. Если делитель равен 0, или если частное не помещается в назначенный регистр, возбуждается прерывание 0.

ENTER Создание кадра стека для параметров процедуры Команда создает кадр стека, который можно использовать

для реализации правил языков высокого уровня с блочной струк­турой. Команда LEAVE в конце процедуры выполняет обратные действия.

Команда имеет два параметра. Первый определяет число бай­тов динамической памяти, выделяемых в стеке для вызванной программы. Второй параметр соответствует лексическому уровню вложенности программы (от 0 до 31). Этот уровень определяет, сколько наборов указателей кадра стека копируются центральным процессором в новый кадр стека из текущего кадра.

ESC Расширение процессора

Арифметический сопроцессор предоставляет расширение

системы команд МП 80386. Сопроцессор поддерживает высокоточные

вычисления как целочисленные, так и с плавающей точкой и, кро­ме того, содержит набор полезных констант, ускоряющих вычисле­ния. Сопроцессор работает параллельно с центральным процессо­ром, обеспечивая таким образом высокую производительность.

Команды сопроцессора включаются в общий поток команд, составляющих программу для МП 80386. Система выполняет команды сопроцессора в том порядке, в котором они появляются в потоке.

HLT Останов

Команда прекращает выполнение любых команд и переводит МП в состояние останова. Немаскируемые прерывания, операция за­пуска и разрешенные прерывания возобновляют работу процессора. Команда HLT обычно является последней командой в последова­тельности команд останова системы, например для сохранения состояния процесса после обнаружения сбоя питания.

IDIV Знаковое деление

Команда выполняет знаковое деление. Делимому, частному и остатку неявно назначаются определенные регистры, в то время как местонахождение делителя указывается явным образом. По форме делителя определяется, какие регистры должны использо­ваться. Если делитель равен нулю или частное слишком велико для регистра-приемника, возбуждается прерывание 0.

IMUL Знаковое целочисленное умножение

Команда выполняет операцию знакового целочисленного умно­жения. Команда имеет три варианта.

1. Однооперандная форма. Операнд может быть байтом, сло­вом или двойным словом в памяти или регистре общего назначе­ния. Команда использует содержимое регистров EAX и EDX в ка­честве операндов.

2. Двухоперандная форма. Один из операндов-источников мо­жет находиться в любом регистре общего назначения, в то время как другой может быть в регистре общего назначения или в памя­ти. Произведение размещается на месте операнда в регистре об­щего назначения.

3. Трехоперандная форма. Два операнда представляют собой источники и один приемник. Один из операндов-источников явля­ется непосредственным значением, записанным в команде. Второй может находиться в памяти или в любом регистре общего назначе­ния. Произведение может быть записано в любой регистр общего назначения. Непосредственный операнд считается знаковым. Если он является байтом, процессор перед выполнением умножения ав­томатически расширяет его знак до размера второго операнда.

IN Ввод из порта

Команда вводит байт или слово из порта и записывает его в регистре (AL, AH, EAX). Порт указывается вторым операндом. Для доступа к порту его номер следует поместить в регистр DX и использовать команду IN с обозначением DX в качестве второго параметра.

INC Инкремент на 1.

Команда прибавляет 1 к операнду-приемнику, но в отличие от ADD не влияет на флаг CF.

INT Вызов процедуры обработки прерывания

Команда передает управление от одной ячейки программного сегмента на другую. Эти ячейки могут принадлежать одному прог­раммному сегменту или разным программным сегментам. Команда возбуждает программное прерывание, позволяющее пользователю передть управление из своей программы программе обработки пре­рываний.

Команда INTn активизирует программу обработки прерывания, соответствующую номеру, указанному в команде. Команда может определять прерывание любого типа. Заметьте, что прерывания

0..31 зарезервированы фирмой Intel. Возврат управления из программы обработки прерывания осуществляется командой IRET.

INTO Прерывание по переполнению

Команда при установленном флаге OF возбуждает прерывание 4, которое специально зарезервировано для этой цели. Флаг OF устанавливается рядом арифметических, логических и строковых команд.

IRET Возврат из прерывания

Команда возвращает управление прерванной процедуре. В от­личие от команды RET IRET извлекает из стека значения флагов и помещает их в регистр флагов. Флаги записываются в стек в про­цессе реализации прерывания.

IRETD Возврат из прерывания в 32-разрядном режиме

JMP Переход

Команда передает управление из одной точки сегмента в другую. Точка перехода может быть в том же сегменте (ближний переход) или в другом сегменте (дальний переход). Команда бе­зусловно передает управление в точку перехода и является, та­ким образом, однонаправленной.

Команды условных переходов:

JA/JNBE Выше, не ниже и не равно

JAE/JNB Выше или равно, не ниже

JB/JNAE Ниже, не выше и не равно

JBE/JNA Ниже или равно, не выше

JC Перенос

JE/JZ Равно, нуль

JNC Отсутствие переноса

JNE/JNZ Не равно, не нуль

JNP/JPO Отсутствие четности, нечетность

JP/JPE Четность

JG/JNLE Больше, не меньше и не равно

JGE/JNL Больше или равно, не меньше

JL/JNGE Меньше, не больше и не равно

JLE/JNG Меньше или равно, не больше

JNO Отсутствие переполнения

JNS Отсутствие знака

(положительно, включая нуль)

JO Переполнение

JS Знак (отрицательно)

LAHF Загрузка флагов в регистр AH

Хотя для изменения флагов CF и DF предусмотрены специаль­ные команды, остальные флаги, используемые в прикладных прог­раммах, нельзя изменять непосредственно. Эта команда дает воз­можность программного изменения остальных битов флагов с по­мощью команд побитовых операций после пересылки флагов в стек или в регистр AH.

Команда копирует биты SF,ZF,AF,PF,CF соответственно в разряды 7,6,4,2,0 регистра AH.

LAR Загрузка байта прав доступа

Команда читает дескриптор сегмента и заносит биты дроб­ности(23), свободный (20), присутствия (15), DPL (14), типа (9-11) и доступа (8) в 32-разрядный регистр, бит дробности и свободный бит не пересылаются.

LEA Загрузка исполнительного адреса

Команда пересылает по адресу приемника относительный ад­рес операнда-источника. Операнд-источник должен находиться в памяти. Операнд-приемник должен быть регистром общего назначе­ния. Команда особенно полезна для инициализации регистров пе­ред выполнением действий над начальными данными или команды XLAT.

LEAVE Выход из процедуры высокого уровня

Команда выполняет действия противоположные действию ко­манды ENTER.

LGDT Загрузка регистра таблицы глобальных дескрипторов

LIDT Загрузка регистра таблицы

Команда оповещает аппратные средства о точке перехода в случае прерываний.

LGS Загрузка полного указателя

LSS Загрузка указателя с использованием регистра SS

LDS Загрузка указателя с использованием регистра DS

LES Загрузка указателя с использованием регистра ES

LFS Загрузка указателя с использованием регистра FS

Команды указателей данных загружают указатель, состоящий из селектора сегмента и относительного адреса, в регистр сег­мента и регистр общего назначения.

LLDT Загрузка регистра локальных дескрипторов

Таблица локальных дескрипторов загружается, когда задача или главная подсистема получает или восстанавливает контроль над системой.

LWSW Загрузка слова состояния машины

Команда загружает в регистр CR0 слово сотояния машины. Команду можно использовать для переключения в защищенный ре­жим.

LOCK Префикс установки сигнала LOCK#

Сигнал LOCK# захватывает общую память, так что МП получа­ет ее в свое исключительное пользование на время выполнения следующей за LOCK команды.

LODS Загрузка строкового операнда

LODSB Загрузка байта

LODSW Загрузка слова

LODSD Загрузка двойного слова

Эти команды обрабатывают не логические или арифметические переменные, а строки. Они воздействуют на один элемент строки, который может быть байтом, словом или двойным словом.

LOOP Циклическое выполнение, пока счетчик ECX не нуль

LOOPE Цикл, пока равно

LOOPZ Цикл, пока нуль

LOOPNE Цикл, пока не равно

LOOPNZ Цикл, пока не нуль

Команды обеспечивают условный переход для циклического выполнения участка программы. Число шагов определяется значе­нием, занесенным в регистр ECX. Все разновидности команды ав­томатически выполняют декремент ECX и останавливают цикл, если ECX=0.

LTR Загрузка регистра задачи

Первый операнд команды LTR определяет регистр-источник или ячейку памяти, содержащие информацию для регистра задачи. Команда загружает эту информацию в регистр задачи.

MOV Пересылка в/из специальных регистров

Команда используется для загрузки и выгрузки специальных регистров и регистров общего назначения.

MOVS Пересылка данных из строки в строку

MOVSB Пересылка байта строки

MOVSW Пересылка слова строки

MOVSD Пересылка двойного слова строки

Эти команды используются для работы со строками, а не ло­гическими или арифметическими величинами.

MOVZX Пересылка с расширением нуля

Команда расширяет 8-разрядную величину до 16-разрядной и 8- или 16-разрядную величину до 32-разрядной заполнением стар­ших разрядов нулями.

MUL Целочисленное беззнаковое умножение содержимого ре­гистров AL или AX

Команда перемножает операнд-источник и содержимое аккуму­лятора и возвращает результат удвоенной длины.

NEG Изменение знака, дополнение до 2

Команда выполняет вычитание знакового целочисленного опе­ранда из нуля.

NOP Холостая команда

Команда занимает в памяти только 1 байт. Она действует только на указатель команд EIP. Команда NOP полезна при "вы­равнивании" адресов переходов.

OR Логическое ВКЛЮЧАЮЩЕЕ ИЛИ

Команда сравнивает два операнда и вычисляет следующее: если соответствующие биты в операндах равны 0, результат 0; в противном случае результат 1.

OUT Вывод в порт

Команда пересылает данные из регистра в порт вывода. Опе­ранд-источник находится в регистре AL,AX,EAX. Номер порта оп­ределяется первым операндом. Для вывода данных в любой порт от 0 до 65535 номер порта помещается в регистр DX.

OUTS Вывод строки в порт

OUTSB Вывод байта

OUTSW Вывод слова

OUTSD Вывод двойного слова

Разновидности команды OTS действуют аналогично команде OUT, но выводит в порт строку и после вывода данного содержи­мое регистра-источника получает приращение.

POP Извлечение слова из стека

Команда пересылает слово или двойное слово из текущей вершины стека (регистр ESP) по адресу приемника. Затем регистр ESP инкрементируется.

POPA Извлечение из стека содержимого всех регистров

POPAD Извлечение из стека содержимого всех регистров - 32-разрядный режим

POPF Восстановление из стека регистра FLAGS или EFLAGS

POPFD Восстановление из стека - 32-разрядный режим

PUSH Занести операнд в стек

Команда выполняет декремент указателя стека (ESP), затем заносит операнд-источник на вершину стека, куда указывает ESP.

PUSHA Занести в стек содержимое всех регистров общего назначения

Команда сохраняет в стеке содержимое восьми регистров об­щего назначения.

PUSHF Занесение в стек содержимого регистра флагов

RCL Циклический сдвиг влево через бит переноса с исполь­зованием бита CF для расширения

RCR Циклический сдвиг вправо через бит переноса с исполь­зованием бита CF для расширения

ROL Циклический сдвиг влево с циклическим возвратом битов

ROR Циклический сдвиг вправо с циклическим возвратом битов Команды циклического сдвига дают возможность циклически

сдвигать биты в байтах, словах и двойных словах.

REP Повторение последующей строковой операции

REPE Повторение, пока равно

REPZ Повторение, пока нуль

REPNE Повторение, пока не равно

REPNZ Повторение, пока не нуль

Префикс REP задает повторение строковой операции, что позволяет МП обрабатывать строки значительно быстрее, чем с помощью обычнного программного цикла.

RET Возврат из процедуры

Команда завершает выполнение вызванной процедуры и пере­дает управление посредством обратной ссылки, хранящейся в сте­ке. Обратная ссылка указывает на программу, первоначально выз­вавшую процедуру.

SAHF Запись содержимого регистра AH в регистр флагов

SAL/SAR/SHL/SHR Команда сдвига

Биты в байтах, словах и двойных словах могут сдвигаться логически или арифметически. Сдвиг осуществляется на заданное число разрядов вплоть до 31.

SBB Целочисленное вычитание с займом

Команда вычитает операнд-источник из операнда-приемника. Если флаг CF установлен, вычитается еще 1.

SCAS/SCASB/SCASW/SCASD Сравнение строковых данных

Эти команды используются для работы со строками. Они воз­действуют на один элемент строки. Адресация элементов строки осуществляется через регистры ESI и EDI. После каждой строко­вой операции эти регистры автоматически уменьшаются или увели­чиваются на 1 в зависимости от регистра DF.

SETcc Установка байта по условию

Команда записывает в байт 0 или 1 в зависимости от любого из 16 условий, определяемых флагами состояния. Байт может быть в памяти или в однобайтовом регистре общего назначения. Если условие cc истинно, команда записывает в байт 1; в противопо­ложном случае - 0.

SGDT Запись в память содержимого регистра таблицы гло­бальных дескрипторов

SIDT Запись в память содержимого регистра таблицы деск­рипторов прерываний

Эти команды копируют содержимое регистра в поле из 6 байт на которое указывает операнд.

SHLD Сдвиг влево с двойной точностью

SHRD Сдвиг вправо с двойной точностью

Команды предоставляют возможность реализации операций на данных невыровненных строках битов.

SLDT Запись содержимого регистра таблицы локальных деск­рипторов

Команда записывает содержимое LDTR в регистр или ячейку памяти с исполнительным адресом, указанным в операнде команды.

SMSW Запись слова состояния машины

Слово состояния машины является частью регистра управле­ния CR0. Команда записывает это слово в двухбайтовый регистр или ячейку памяти. Команда оставлена для совместимости с МП

80286. В МП 80386 следует использовать команду MOV ... CR0.

STC Установка флага переноса CF

STD Установка флага направления DF

STI Установка флага прерываний IF

STOS/STOSB/STOSW/STOSD Запись строки данных

Запись строки данных. Адресация через регистры ESI и EDI.

STR Запись регистра задачи

Команда копирует содержимое регистра задачи в двухбайтный регистр или ячейку памяти.

SUB Вычитание целых чисел

Команда вычитает операнд-источник из операнда-приемника и помещает результат на место операнда-приемника.

TEST Логическое сравнение

Команда выполняет логическую операцию И над двумя операн­дами. Затем команда сбрасывает флаги OF и CF, оставляет AF не­определенным и модифицирует SF,ZF,PF. Команда отличается от команды AND тем, что она не модифицирует операнд-приемник.

VERR Проверка сегмента на чтение

VERW Проверка сегмента на запись

Эти команды проверяют, доступен ли выбранный селектором сегмент при текущем уровне привилегий и разрешены ли в нем чтение или запись. Если сегмент доступен, флаг ZF устанавлива­ется в 1.

WAIT Ожидание пассивного состояния вывода BUSY#

Команда приостанавливает выполнение программы МП 80386 до тех пор пока ЦП 80386 не обнаружит пассивное состояние вывода BUSY. Это указывает на то, что сопроцессор завершил выполнение задачи и что ЦП может получить результат.

XCHG Обмен между регистрами или между памятью и регистром Команда заменяет три команды MOV. Она не нуждается во

вспомогательной ячейке для обмена операндами.

XLAT Табличное перекодирование

Команду удобно использовать при преобразовании из одной системы кодов в другую. Длина таблицы перекодирования от 1 до 256 байт.

XOR Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ