ЛАБОРАТОРНАЯ РАБОТА №4

Изучение системы команд и режимов адресации МП Intel 80x86 в реальном режиме

СПРАВОЧНЫЕ СВЕДЕНИЯ

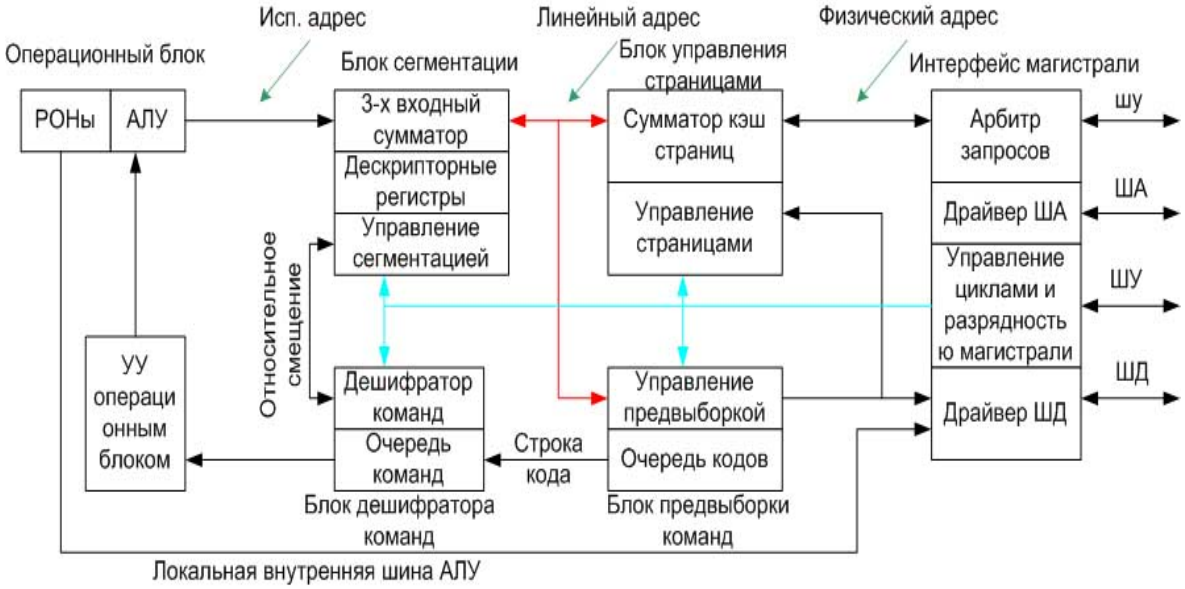
Архитектура микропроцессора 80x86

Высокопроизводительный 32-х разрядный микропроцессор 80x86 ориентирован на эффективное выполнение программ в среде многозадачных ОС (типа Windows). Микропроцессор имеет 32-разрядные регистры и 32-х разрядные раздельные шины адреса и данных.

Физическое адресное пространство равно 4 Гб (2^{32})

Виртуальное адресное пространство равно 64Тб (2^{46}).

Структурная схема микропроцессора



В состав микропроцессора входят:

-операционный блок;

-блок выборки команд;

-диспетчер памяти;

-устройство управления;

-интерфейс магистрали.

Операционный блок вместе с устройством управления составляют центральный процессор микропроцессора, который предназначен для выполнения всех логических и математических операций. Операционный блок включает в свой состав АЛУ и восемь 32-хразрядных регистров (РОНов). Подсистема выборки команд реализует двухступенчатый алгоритм конвейеризации и состоит из блоков предвыборки команд и дешифрации

команд. Блок предвыборки команд принимает команды из интерфейса магистрали, выстраивая их в очередь кодов.

Блок дешифрации команд производит преддешифрацию, то есть, определяет тип и формат команд, определяет номера используемых регистров, выделяет поле относительного смещения и передает его в блок сегментации для вычисления линейного адреса.

Диспетчер памяти состоит из блока сегментации и блока управления страницами, и осуществляет двухступенчатое формирование физического адреса ячейки памяти.

Имеется два режима работы микропроцессора:

а) режим реальных адресов (реальный режим);

б) режим защищенных виртуальных адресов (виртуальный режим).

В реальном режиме микропроцессор 80i86 работает как быстрый микропроцессор 8086. В реальном режиме страничная организация памяти не используется.

В защищенном режиме используются все возможности микропроцессора. При этом возможен многозадачный режим работы микропроцессора, причем каждая задача защищена и изолирована от других задач и от ОС. В защищенном режиме может быть реализована страничная организация виртуальной памяти объемом до 64 Тб для каждой задачи.

Оперативная память состоит из сегментов, каждый из которых может быть разбит на страницы. Каждая страница имеет фиксирующий размер по 4кб каждая страница, причем разбиение памяти на страницы возможно только в защищенном режиме.

Диспетчер памяти (и сегментный, и страничный) служат для вычисления физических адресов при обращении микропроцессора к памяти.

Интерфейс магистрали позволяет осуществить обмен микропроцессора информацией с ОЗУ и ПУ с помощью 32-хразрядной двунаправленной ШД, 34-хразрядной ША и 16-разрядной ШУ. Особенностью ШД является возможность динамического изменения ее разрядности. За один цикл шины может быть переданы 1,2 или 4 байта. По ША передаются 32-х адреса. ША состоит из 30 адресных линий, обозначенных как A31-A2 и 4-х линий выбора байт, обозначенных как BE3-BE0. Сигналы выбора байт определяют какие байты 32-хразрядной шины данных участвуют в текущем цикле обмена. Это позволяет легко согласовать ШД с байтной организацией памяти:

При BE0=0	Адресуется младший байт	(т.е.D0-D7 ШД)
При BE1=0	Адресуется следующий байт	(т.е.D8-D15 ШД)

При BE2=0	Адресуется следующий байт	(т.е.D16-D23 ШД)
При BE3=0	Адресуется старший байт	(т.е.D24-D31 ШД)

Регистры

Набор регистров микропроцессора включает:

-РОНы;

-сегментные регистры;

-указатели команд и регистр флагов;

-регистры управления;

-регистры адреса системы;

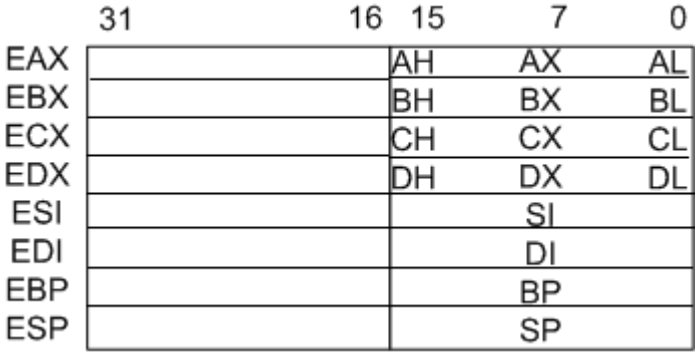
-регистры отладки;

-регистры тестирования.

Всего микропроцессор содержит 32 регистра, из которых 15 регистров могут адресоваться пользователем, кроме этого имеется указатель команд и 16 системных регистров недоступных пользователю. Все 16-тиразрядные регистры микропроцессоров 8086, 80186, 80286 содержатся в 32-разрядных регистрах микропроцессора 80i386.

РОНы

Восемь 32-хразрядных РОНов предназначены для хранения операндов и адресов, и располагаются в операционном блоке.



Младшие разряды РОНов (с 0 по 15) доступны отдельно при использовании имен AX, BX, CX, DX, SI, DI, BP, SP. При операциях с байтами для 4-х верхних регистров можно отдельно обращаться к младшему байту по, именам AL, BL, CL, DL и к старшим байтам по именам AH, BH, CH, DH.

Верхние 4 регистра используются для хранения операнда и позволяют выполнять операции над отдельными байтами, 16-разрядными словами и 32-разрядными двойными словами. Остальные (4 нижних) регистра используются как адресные регистры для хранения 16-разрядных и 32-разрядных адресов.

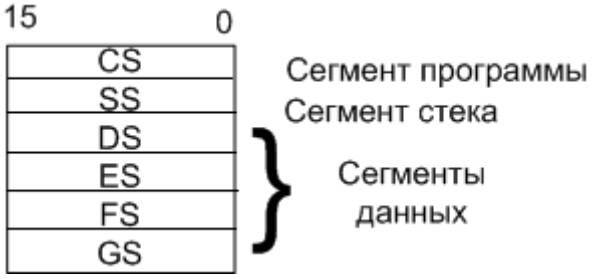
Регистр ESP используется как указатель вершины стека.

EBP – используется как базовый регистр.

А регистры EDI и ESI используются как индексные регистры (для хранения индексов). В качестве базового и индексного регистров в некоторых случаях может использоваться и регистр данных EBX.

Регистры сегментов

Архитектура микропроцессора поддерживает организацию памяти в виде сегментов. Всего имеется 6 сегментных регистров.



В реальном режиме: для хранения адресов начала соответствующих сегментов используется 16-хразрядные регистры CS, SS, DS, ES, FS, GS. Они выполняют те же функции, что и в микропроцессоре 8086. то есть, используются при вычислении физических адресов путем суммирования базового адреса, находящегося в сегментном регистре, со смещением. Такое суммирование выполняет блок сегментации, где располагаются и сами сегментные регистры. Максимальный размер сегмента в реальном режиме 64 Кбайт.

Адресация в микропроцессорах INTEL

Размещение байт и слов в памяти

Память логически организована как одномерный массив байт, каждый из которых имеет 20-тибитовый физический адрес в диапазоне от 00000 до FFFFF. Любые два смежных байта в памяти могут рассматриваться как одно 16-тиразрядное слово. Младший байт слова всегда имеет меньший адрес, а старший – больший адрес. Адресом слова считается адрес его младшего байта.

Пространство памяти емкостью в 1 Мб представлен как набор сегментов, определяемых программным путем.

Сегмент состоит из смежных ячеек памяти и является независимой и отдельно адресуемой единицей памяти объемом 64кб. Каждому сегменту программой назначается начальный (базовый) адрес – адрес 1-го байта сегмента. Начальные адреса 4-х сегментов, выбранных в качестве текущих записываются в сегментные регистры CS, DS, SS, ES.

15 0

00	ES
01	CS
10	SS
11	DS

В регистрах находятся базовые адреса для обращения к командам программы (CS), к данным (DS), стеку (SS), дополнительным данным (ES). Для обращения к командам и данным, находящимся в других сегментах, необходимо изменить содержимое сегментных регистров. Сегментные регистры инициализируются в начале программы засылкой в них соответствующих базовых адресов. В каждом сегментном регистре – 16 старших бит 20-разрядного начального адреса сегмента. 4 младших разряда адреса считаются равными 0 и автоматически приписываются справа к содержимому сегментного регистра при вычислении физических адресов.

Физический адрес ячейки памяти – это 20 - битовое число в диапазоне от 0 до FFFFF, которое однозначно определяет положение каждого байта в пространстве памяти емкостью до 1 Мб.

Логический адрес ячейки состоит из двух 16-битовых беззнаковых значений:

- начального адреса сегмента (база)
- внутрисегментного смещения, которое определяет расстояние от начала сегмента до адресуемой ячейки.
- Для вычисления физического адреса база сегмента сдвигается влево на 4 разряда и суммируется со смещением EA (это эффективный адрес, вычисляемый в соответствии с заданным способом адресации). Перенос из старшего бита при сложении игнорируется, что приводит к циклической организации памяти, когда за ячейкой с максимальным адресом следует ячейка с 0 адресом. Аналогичную кольцевую организацию имеет и каждый сегмент.

Смещение EA – эффективный адрес, вычисляемый в соответствии с заданным способом адресации.

Сегментная структура памяти обеспечивает возможность создания позиционно – независимых или динамически перемещаемых программ, что необходимо в мульти программной среде. Для этого все смещения в программе должны задаваться относительно фиксированных значений,

находящихся в сегментных регистрах. Это позволяет произвольно перемещать программу в адресном пространстве памяти, изменяя только содержимое сегментных регистров. Стек организуется в ОЗУ по принципу скользящей вершины и его положение в ОЗУ определяется содержанием регистров SS(база) и SP(смещение). При этом SS регистр хранит базовый адрес текущего сегмента стека, а регистр SP указывает на вершину стека. При каждом обращении к стеку пересылается 1 слово, причем содержимое SP изменяется автоматически: при записи в стек слова содержимое SP уменьшается на два, а при чтении из стека – увеличивается на два.

Форматы команд микропроцессора INTEL

Регистры общего назначения (РОНы) разбиты на две группы:

1. группа HL, состоящая из регистров AX, BX, CX, DX, которые предназначены для хранения данных и допускают раздельную адресацию их старших H и младших L половин.
2. группа PI, содержащая указательные регистры BP, SP и индексные регистры SI, DI, в которых обычно храниться адресная информация.

РОНы

	15 8	7 0	
000	AH	AL	AX - аккумулятор
001	CH	CL	CX
010	DH	DL	DX
011	DH	BL	BX
100	SP		Адресный регистр
101	BP		Адресный регистр
110	SI		Адресный регистр
111	DI		Адресный регистр

Команды могут адресовать один или два операнда. В двухоперандных командах один из операндов должен обязательно располагаться в регистре, поскольку имеются команды типа регистр-регистр, регистр-память, память регистр, но команда типа память-память отсутствует, за исключением команд пересылки цепочки байт ил слов.

Формат двухоперандной команды имеет следующий вид:

Cop dw	md reg r/m	Disp L	Disp H
1	2	3	4

Первый байт команды содержит Cop – код операции и два однобитовых поля: d- бит направления передачи и w- длина операнда.

При d=1, то осуществляется передача операнда или результата операции в регистр, номер которого задается полем reg второго байта команды.

При d=0, то осуществляется передача операнда или результата из адресуемого полем reg регистра.

Поле W идентифицирует тип(разрядность) операндов:

Если w=1, то команда оперирует с 2-хбайтным словом.

w=0, команда оперирует с 1 байтом.

2-ой байт – постбайт, определяет участвующие в операции регистры или регистр и ячейку памяти. постбайт состоит из 3-х полей:

md – режим, показывающий как интерпретируется поле r/m для нахождения первого операнда.

Reg – регистр, используется в 2 –х операндных командах.

R/m – регистр/память

Поле reg определяет операнд, который обязательно находится в регистре микропроцессора и считается вторым операндом. Поле r/m определяет операнд, который может находится в регистре или памяти и условно считается первым. Поле reg используется только для указания регистра в двухоперандных командах. Если в команде один операнд, то он идентифицируется полем R/m, а поле reg отсутствует. Вместо поля reg в этом случае используется расширение кода операции.

Поле md(модальность) показывает, как интерпретируется поле R/m для нахождения первого операнда:

Если md=11, то операнд находится в регистре,номер которого задан полем R/m. При других значениях md операнд находится в памяти.

Когда адресуется память, то поле md определяет вариант использования смещения disp, находящегося в 3 и 4 байте.

Disp – смещение в команде, интерпретируемое как целое число со знаком.

md=00 смещение disp отсутствует.

md=01 disp =disp L, команда содержит 8 бит , смещение D8.

md=10 disp= dispH dispL, команда содержит 16 бит, смещение D16.

Режимы адресации (вычисление эффективного адреса EA)

Поле md					
R/m	md=00 disp=0	md=01 dispH=dispL=D8	md=10 disp=dispH dispL	11	
				W=1	W=0
000	BX+SI	BX+SI+D8	BX+SI+D16	AX	AL
001	BX+DI	BX+DI+D8	BX+DI+D16	CX	CL
010	BP+SI	BP+SI+D8	BP+SI+D16	DX	DL
011	BP+DI	BP+DI+D8	BP+DI+D16	BX	BL
100	SI	SI+D8	SI+D16	SP	AH
101	DI	DI+D8	DI+D16	BP	CH
110	DI6	BP+D8	BP+D16	SI	DH
111	BX	BX+D8	BX+D16	DI	BH

Приведенные в таблице правила имеют одно исключение, позволяющее реализовать прямую (абсолютную) адресацию: если md=D16=dispH dispL.

Таким образом, имеется три варианта интерпретации поля md и восемь вариантов интерпретации поля r/m, что дает 24 варианта вычисления эффективного адреса EA.

Команды микропроцессора реализуют разные способы адресации, что упрощает организацию и использование сложных структур данных и расширяет возможности отдельных команд, и повышает гибкость их применения.

- регистровая адресация. Операнд находится в одном из РОНов или сегментном регистре. Регистр может быть определен в байте кода операции или постбайте (3-битными полями) reg и r/m при условии md=11. Команды, оперирующие содержимым регистров, короткие и быстрые, так как не требуют вычисления эффективного адреса EA, и обращения к памяти.
- Непосредственная адресация. Непосредственные операнды – это const длиной 8 или 16 бит, которые размещаются в последних байтах команды.

Cop sw	md cop r/m	Disp L	Disp H	Data L	Data H

Так как 2-ой операнд размещается в команде, то поле reg отсутствует, но вместо него используется расширение кода операции cop. Отсутствует бит направления d, так как результат операции можно поместить только на место первого операнда. Место этого бита d занимает бит S, который является

признаком использования одного байта для задания непосредственного операнда при работе со словами.

Поля S и W интерпретируются следующим образом:

SW=X0, один байт данных Data L

SW=01, один байт данных Data H Data L

SW=11, один байт данных, который автоматически расширяется со знаком до 16 бит.

- прямая адресация. Эффективным адресом EA является содержание байта в смещении disp команды. Реализуется при использовании постбайта с полями md=00 и r/m=10.
- Косвенно-регистровая адресация. Эффективный адрес EA равен содержимому одного из регистров SI, DI, BX при md=00 и r/m=100, 101, 111.
- Базовая адресация EA вычисляется суммированием содержимого регистров BX и BP со смещением disp при md=01 и 10, r/m=100 и 111.
- Индексная адресация EA вычисляется суммированием индексных регистров SI и DI и смещения disp при md=01 и 10 при r/m=100, 101.
- Базовая индексная адресация. EA равно сумме содержимого базовых регистров BX или BP, индексного регистра SI или DI и смещения disp. Реализуется при md не равного 11 и r/m=000, 001, 010, 011.

Описание лабораторной работы

Целью выполнения лабораторной работы является закрепление знаний о программной модели, структуре команд и адресации МП Intel в реальном режиме.

В основе лабораторной работы заложена программная модель процессора по дешифрации команд этих МП. Модель предлагает студентам ряд команд на ассемблере, отвечающих различным режимам адресации.

Обучаемый должен перевести команду ассемблера в формат машинной команды. Команды выдаются случайным образом. Работа заканчивается по окончанию выполнения (правильной дешифрации) команд с использованием всего спектра возможных адресаций.