

Лекция 9. Микропроцессоры второго поколения

9.1. Общая характеристика шестнадцатиразрядных микропроцессоров второго поколения

16-разрядные МП второго поколения, такие как MC68010 (Motorola), 80286 (Intel), Z8003 (Zilog) обладают более широкими возможностями, по сравнению с процессорами первого поколения. Архитектура процессоров различных фирм в основном одинакова, а проблема состоит в разработке совместимого ПО.

Основная особенность в этих МП – использование поддержки режима виртуальной памяти (ВП) и аппаратная поддержка мультипрограммного режима. С целью применения режима виртуальной памяти в ПЭВМ используется устройство управления памятью (УУП).

Процессоры MC68010 и Z8003 дополняются дополнительными кристаллами УУП, а 80286 содержит встроенное УУП. Устройство управления памятью МП 80286 оказывает поддержку ВП и мультипрограммного режима. За счет аппаратной реализации УУП удалось существенно повысить быстродействие ПК.

9.2. Устройство управления памятью

9.2.1. Защита памяти

Важнейшей функцией устройства управления памятью (УУП) является *защита* памяти. В составленной программе, как правило, имеются ошибки. Ошибочный переход к области данных может привести их к уничтожению. Еще более опасна запись в область программы. При разрушении области ОС обычно возникает системный отказ. Для предотвращения таких ситуаций необходима защита памяти. Для ее осуществления кроме базового адреса вводится *указатель размера* и *атрибут сегмента*. В этом случае ситуация, когда величина смещения превышает размер сегмента, считается аварийной, т.к. возможно проникновение в область соседнего сегмента.

Атрибуты сегментов могут быть самыми разными. Наиболее широко применяются следующие:

- разделение на **системную область** и **область пользователя**;
- разделение на **область программ** и **данных**;
- в случае работы с областью программы производят разделение ее на участки, допускающие только считывание данных или также и запись.

Процессор в каждом цикле вырабатывает сигналы состояния, показывающие, является ли данный цикл считыванием или записью. УУП при каждом обращении к памяти сравнивает атрибуты сегментов с этими сигналами. Если в размере сегмента или в его атрибуте обнаруживается ошибка, УУП выдает в ЦП сигнал сегментной ошибки, вызывающий специальное прерывание.

Для хранения базового адреса, размера и атрибута сегмента в УУП имеются регистры – **дескрипторы сегментов**. Число таких регистров определяется количеством используемых сегментов. При этом если номер сегмента является n – разрядным, то возможно использовать 2^n регистров.

Для ускорения поиска регистра они строятся на принципах ассоциативной памяти. Для этого в составе каждого регистра имеется схема сравнения искомого номера регистра с хранящимся в регистре его номером.

Для защиты памяти, начиная с процессоров x286 используется система привилегий (PL), с помощью которой осуществляется защита сегментов. Система привилегий регулирует доступ к тому или иному сегменту, в зависимости от уровня его защищенности и от степени важности (привилегированности) запроса. В МП x286 и всех последующих

моделях установлены четыре уровня привилегий PL, которые задаются номерами от 0 до 3. Наиболее привилегированным является уровень с меньшим номером. Степень защищенности сегмента также имеет четыре уровня, которые схематически представляются в виде вложенных колец защиты.

В процессорах MC68000 и Z8001 предусмотрено **два** уровня привилегий: в системном и пользовательском режимах.

В соответствии с уровнями привилегий и защищенности установлены следующие правила доступа для сегментов программ и данных:

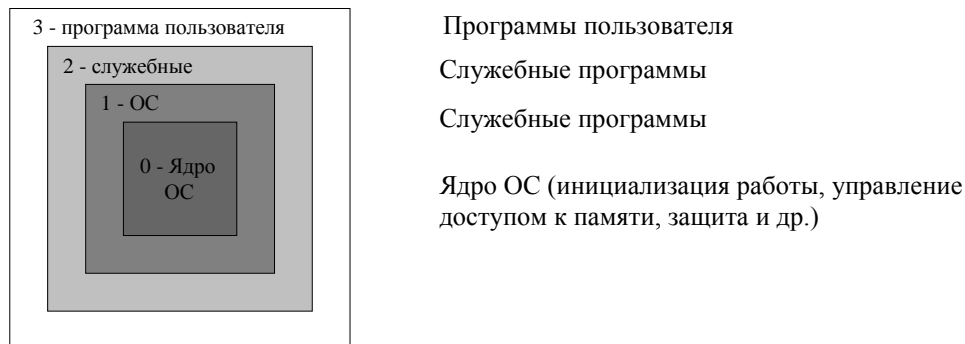


Рисунок 9.1 — Уровни привилегий и защиты

- 1) Данные из сегмента, имеющий уровень привилегий могут быть выбраны программой такого или более высокого уровня привилегий.
- 2) Сегмент программ, имеющий уровень защиты PL, может быть вызван программой, имеющий такой или более низкий уровень привилегий.
- 3) Уровень защиты и привилегий определяется двумя битами, значение которых указывает номер кольца защиты или уровня. Эти биты размещаются в байте доступа дескриптора (для привилегий дескриптора) или в селекторе (уровня привилегий запроса).
- 4) Выполнение команды ветвления ограничивается сегментами в пределах одной программы. В командах вызова и перехода допускается переход в сегменты другой программы при условии, что уровень ее привилегий равен текущему уровню.

Следует заметить, что процессор разрешает менее привилегированной процедуре вызывать более привилегированную, но ограничивает доступ разрешенными точками входа. Допустимые точки входа идентифицируются специальными дескрипторами, называемыми шлюзами вызова. Привилегия в шлюзе задается достаточно низкой, что позволяет обращаться в точку входа операционной системы. Размер смещения в шлюзе не учитывается.

9.2.2. Мультипрограммный режим

Обычно в памяти микро ЭВМ находится одновременно несколько программ: прикладная, ОС, библиотечные подпрограммы и др. При запуске прикладной программы выполняются не только ее команды, но и команды ОС и библиотечных прикладных программ. В процессе выполнения работы до ее окончания осуществляется переход от одной программы к другой. Такое выполнение последовательности команд называется *процессом*, а выполнение команд одной программы – *задачей*.



Рисунок 9.2 – Структура блока управления задач

В обычном компьютере имеется только один процессор, поэтому в каждый момент времени он может выполнять команды только одной задачи. Т.е. команды всех задач выполняются в режиме разделения времени.

Выполнение задачи может быть приостановлено на любой команде, поэтому необходимо обеспечить перезапуск соответствующей программы. Каждой задаче отводится отдельная область памяти, в том числе и стек для хранения всех данных, используемых в задаче (рис.9.2). Кроме того, в памяти предусмотрены ячейки для хранения содержимого программного счетчика ПС,

регистра флагов ФР и регистров общего назначения РОН.

Когда выполнение задачи приостанавливается, в них помещаются начальные значения указанных регистров. При возобновлении задачи содержимое этих ячеек пересылается в ПС, ФР и РОН процессора, после чего начинается выполнение задачи. Совокупность этих ячеек называется *блоком управления задачей*.

Возможные состояния задачи можно разделить на два типа *пассивные* и *активные*.

Активное состояние подразделяется на состояния *выполнения*, *готовности* и *ожидания*. Указанные состояния характеризуются следующими свойствами:

- *Пассивное состояние* – запуск еще не производился, или работа уже завершена;
- *Состояние выполнения (прогона)* – выполняются команды;
- *Состояние готовности* – возможно выполнение любой команды (в ходе выполнения процессором команды другой задачи);
- *Состояние ожидания* – ожидается возникновение прерывания.

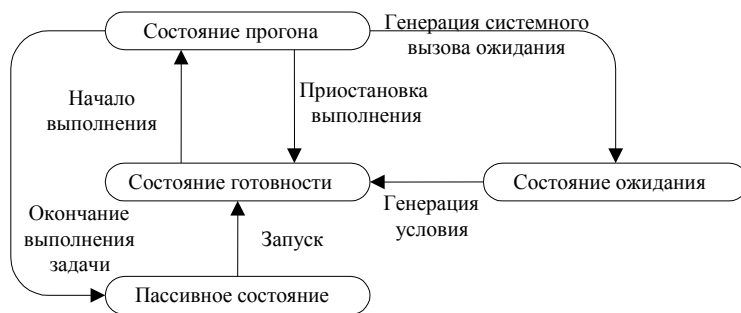


Рисунок 9.3 – Граф изменения состояний задачи

Для реализации мультипрограммного режима в ОС должны быть предусмотрены программы, выполняющие специальные функции. Одной из них является *программа управления задачами*, которая управляет переходом задачи из одного состояния в другое. Переход осуществляется в соответствии с диаграммой (рис.9.3).

При запуске какой-либо задачи устанавливается состояние готовности. В этом состоянии ожидается начало выполнения команд. Если в состоянии прогона происходит обращение к команде, выполнение которой возможно после генерации некоторых условий, выполнение приостанавливается и задача переходит в состояние *ожидания*. После окончания выполнения задача переходит в пассивное состояние.

Выбор, какая задача будет выполняться следующей, решается *планированием задач*. Замена задач в состоянии выполнения называется *переключением задач*. В этом случае в блок управления задачей необходимо передать содержимое ПС, ФР и РОН, а из блока управления новой задачей считать содержимое этих регистров. Однако для реализации этих действий программно требуется много времени. В целях сокращения времени, основные операции целесообразно выполнять аппаратными средствами.

Очень часто при мультипрограммном режиме происходит обращение задач к *общим переменным*. Однако если в момент обращения задачи к этим переменным выполнение задачи приостанавливается, то результат такой операции может быть ошибочным, т.к. данные могут быть изменены (часть старых, а часть новых). Поэтому до окончания обращения одной задачи к общим переменным следует исключить возможность обращения к ним другой задачи. Эта ситуация называется *принципом взаимного исключения*, а область программы, в которой должен соблюдаться этот принцип – *критической областью*.

Наиболее простым способом реализации принципа взаимного исключения является *запрещение прерываний* непосредственно перед входом в критическую область и разрешение его сразу после выхода из этой области.

Для сокращения времени запрета прерываний применяется специальный флаг, называемый *семафором*, и для каждого взаимного исключения используется свой семафор. Перед входом в критическую область опрашивается флаг семафора, и если его сигнал (0) запрещающий, то вход задачи в область запрещен. При открытом семафоре (1) задача входит в критическую область. При выходе из критической области флагу семафора возвращается разрешающее значение.

Семафор также является общей переменной и при обращении к нему тоже необходимо соблюдать принцип взаимного исключения. Это можно реализовать методом запрета прерываний, т.к. время обращения к семафору мало.

9.3. Устройство управления памятью

В процессорах 80286, MC68010 и Z8003 для расширения их возможностей используется поддержка средствами виртуальной памяти (ВП). Предпосылками для реализации ВП служит применение устройств управления памятью (УУП), которое в МП 80286 интегрировано в кристалл ЦП. В других МП используются УУП, выполненные на отдельных кристаллах. Кроме этого УУП выполняет функции защиты памяти и управление мультипрограммным режимом.

9.3.1. Виртуальная память

Виртуальная память – это способ организации основной памяти большой емкости с помощью внешней памяти. Она позволяет при составлении программы распоряжаться всем пространством адресов, зарезервированных в процессоре. Такие адреса называются *виртуальными*.

При выполнении программы УУП преобразует виртуальные адреса в физические. Обычно в основной памяти размещается небольшое количество сегментов, к которым в данной части программы осуществляется обращение, т.е. размещаются только необходимые в данное время сегменты. Поэтому при ссылке на несуществующие сегменты выполнение программы приостанавливается и производится замена содержимого основной памяти.

При этом сегменты, которые не предполагается использовать, посылают во внешнюю память, а на их место размещают требуемые сегменты. Затем возобновляется выполнение приостановленной программы. Функция замены осуществляется ОС.

В случае реализации ВП, кроме базового адреса и размера смещения, дополнительно задается *поле атрибута сегмента* и *поле управляющих данных*. Для хранения такого слова в УУП имеются специальные регистры – **дескрипторы сегментов**. Формат дескриптора виртуальной памяти показан на рис.9.4.

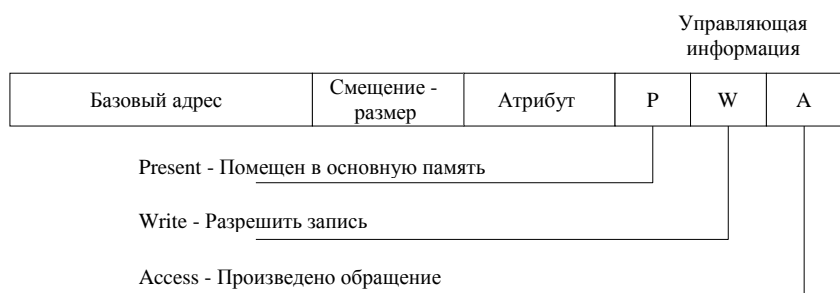


Рисунок 9.4 – Структура дескриптора сегмента виртуальной памяти

Размер смещения указывается потому, что в 286 процессорах сегменты имели произвольный размер ≤ 64 К. Это повышало эффективность использования памяти.

Управляющие биты имеют следующие значения. Бит Р – указатель нахождения сегмента в основной памяти. Если Р=1, сегмент в ВП и УУП осуществляет преобразование виртуального адреса в физический. Если Р=0, то УУП передает в ЦП сигнал отсутствия сегмента. Эта ситуация называется *ошибкой памяти* (memory fault).

В случае разрешения записи информации в данный сегмент бит W устанавливается в 1. При обращении к сегменту биту А присваивается значение 1. ОС в фиксированный момент времени проверяет значение поля А, и если он равен 1, производит приращение содержимое строки справочной таблицы и сбрасывает А. Таким образом, с помощью справочной таблицы ОС отбирает сегменты, которые реже всего используются и поэтому в первую очередь подлежат замене.

Для упрощения замены содержимого памяти целесообразно заменять области одинакового размера. С этой целью вводится *страничная* организация памяти, согласно которой основная память разбивается на страницы объемом, например по 2 К байт (введена, начиная с процессоров x386).

Сегменты также делятся на страницы такого же размера, а в основную память помещаются только те из них, которые необходимы для выполнения программы. Таким образом, в основной памяти отсутствуют неиспользуемые промежутки, которые имели место при размещении сегментов (из-за их различной длины).

При обращении к странице некоторого сегмента, не существующей в основной памяти, генерируется сигнал ее отсутствия. Этот сигнал также называют *ошибкой памяти*, либо *страничной ошибкой*.

Для преобразования виртуальных адресов в физические требуется дескрипторы страниц, что приводит к усложнению УУП.

9.4 Архитектура процессоров второго поколения

9.4.1 Структурная схема процессора второго поколения (Intel 80286)

Структурная схема 16-разрядного процессора второго поколения, разработанного фирмой Intel, изображена на рис.9.5.

Процессор состоит из четырех блоков: *адресного AU*, *шинного BU*, *исполнительного EU* и *командного IU*, причем все блоки могут работать параллельно. Шинный блок осуществляет считывание памяти и портов ввода/вывода. Адресный блок вычисляет все

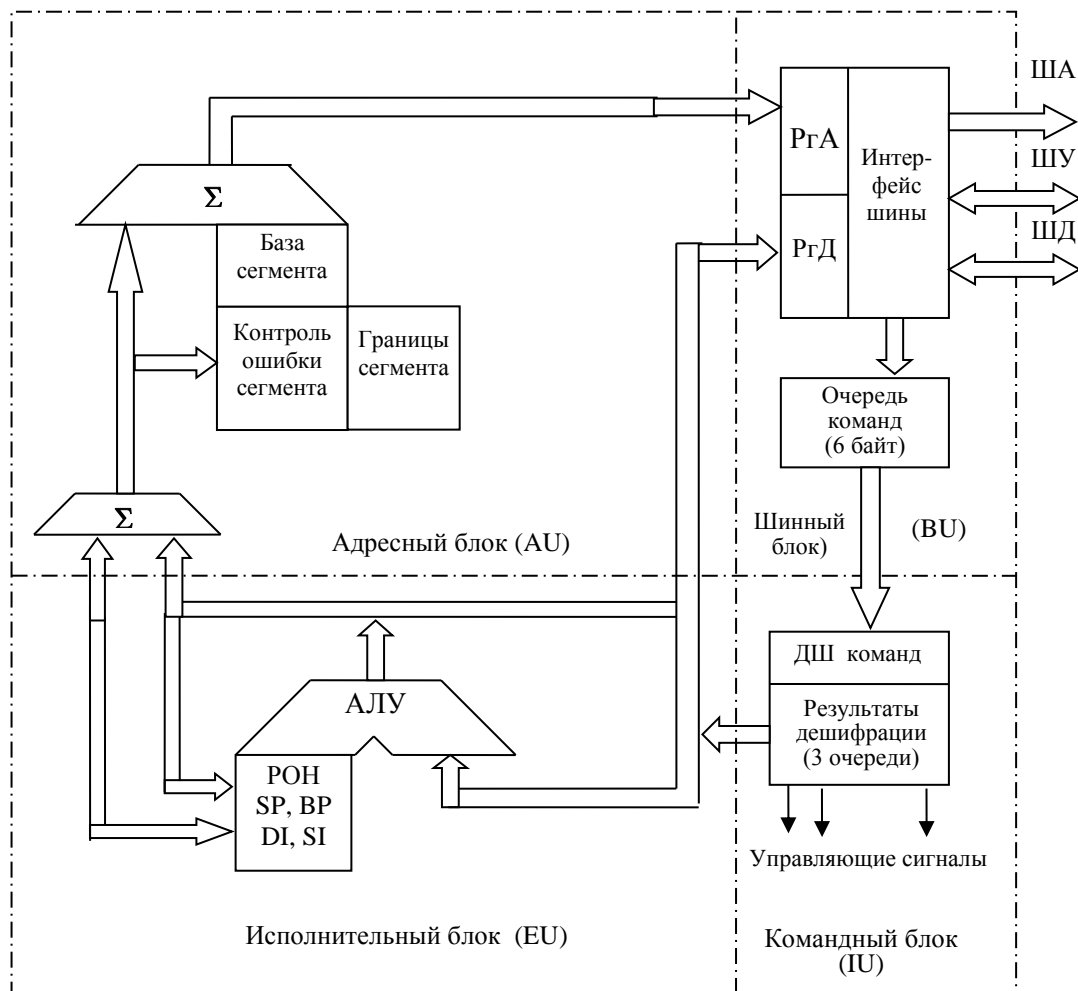


Рисунок 9.5 – Структурная схема 16-разрядного процессора второго поколения

адреса и формирует физические адреса.

За счет независимой параллельной работы блоков производительность возрастает в 2-3 раза по сравнению с процессором 8086 (при том, что тактовая частота у него возросла всего до 12,5 МГц по сравнению с 5 МГц в 8086). Для использования шин с максимальной эффективностью применяется 6-байтовая очередь команд. При этом осуществляется упреждающее считывание команд, время выполнения которых в АЛУ велико. Результаты дешифрации помещаются в очередь результата. Поэтому командный блок после выполнения одной команды сразу же переходит к выполнению следующей. При наличии команд переходов очередь сбрасывается.

Процессор может работать в двух режимах: **реальном** и **защищенном**.

Реальный режим используется для выполнения программ МП предыдущего поколения и инициализации регистров защищенного режима. Защищенный режим обеспечивает возможность поддержки виртуальной памяти, реализовать механизм привилегий с целью защиты памяти и многозадачную работу.

После подачи питания или сигнала сброса процессор устанавливается в реальный режим. В этом режиме процессор x286 имеет такую же базовую архитектуру, что и 8086, которая работает с большой скоростью. Различие двух режимов состоит в способе вычисления базового адреса.

9.4.2 Регистровая структура процессора

9.4.2.1. Управление виртуальной памятью МП 80286

БИС процессора 80286 имеет встроенное УУП. Это устройство оказывает поддержку ВП и мультипрограммного режима. Регистровая структура МП имеет вид, изображенный на рис.9.6.

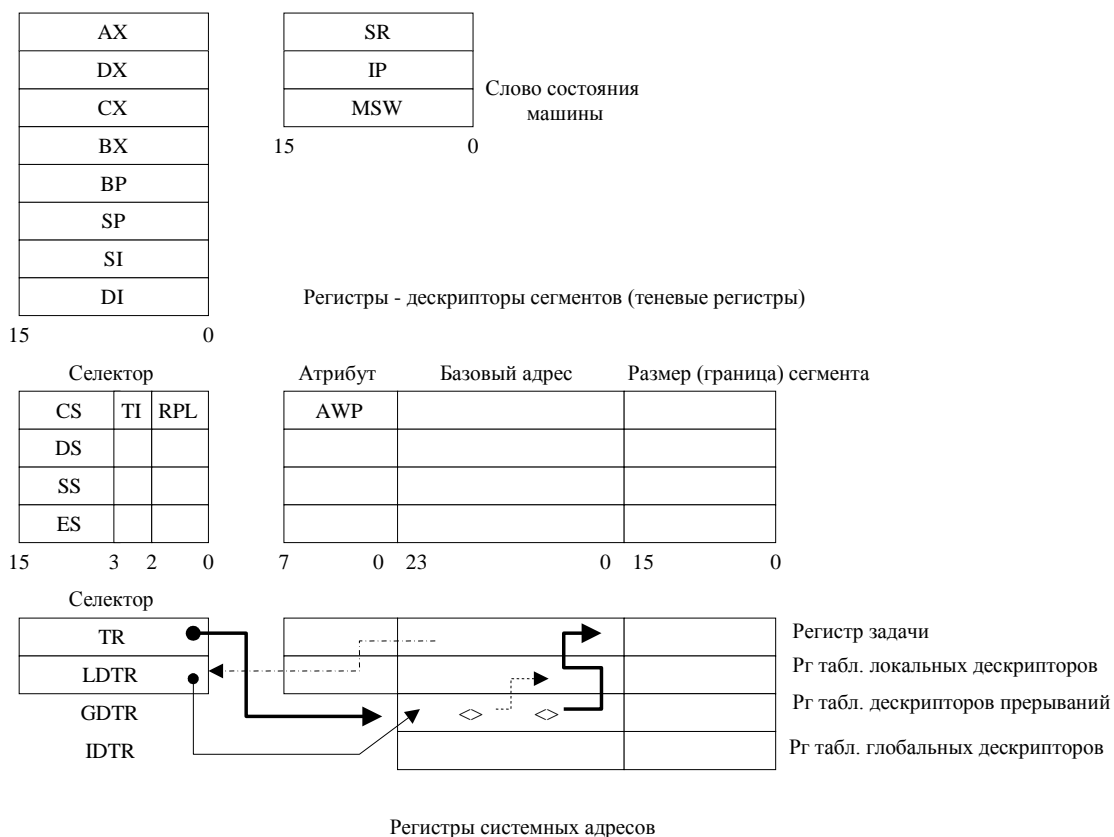


Рисунок 9.6 - Регистровая структура процессора Intel 80286

TI – Table Indicator; RPL – Requested Privilege Level.

В защищенном режиме в сегментных регистрах располагаются не информация о физическом адресе, а *селекторы*, указывающие на описатели сегментов. С помощью селекторов, расположенных в сегментных регистрах, центральный процессор получает из локальной или глобальной таблиц дескрипторов описатели, характеризующие размещение и длину используемых сегментов.

Если процедура получения дескрипторов будет осуществляться при выполнении каждой команды, то это приведет к существенному снижению быстродействия.

Дескрипторы находятся в ОЗУ и при поступлении нового селектора считываются в регистры дескрипторов. Таким образом, информация о физическом адресе сегмента и его длине при загрузке сегментного регистра загружается в так называемый *теневого регистр*, который имеется для каждого сегментного регистра. Т.е. каждый сегментный регистр содержит кроме видимой части – 16 битового селектора – другую невидимую часть, имеющую ширину 48 бит (для каждого из четырех селекторов). Аналогичные теневые регистры имеются для регистров глобальной и локальной таблиц дескрипторов.

Загрузка теневых регистров относительно медленная операция. Так для выполнения команды **MOV DS, AX** в реальном режиме требуется только два такта, то в защищенном режиме 18.

По сравнению с МП 8086 структура регистров CS, DS, SS и ES, предназначенных для управления памятью, расширена за счет дополнительных регистров, используемых в качестве *дескрипторов сегментов*. Содержимое дескрипторов не используется непосредственно в программе. Их содержимое автоматически записывается центральный процессор и используется для управления памятью.

Регистр задачи TR указывает задачу, выполняемую в данный момент.

Все пространство виртуальных адресов состоит из *локальных* адресных пространств, отдельных для каждой задачи, и *глобального* адресного пространства, общего для всех задач.

Системные программы (ОС и др.) хранятся в глобальном адресном пространстве. Пространство виртуальных адресов каждой задачи занимает объем 1 Гбайт $= 2^{13}$ сегментов $\times 2^{16}$ – объем сегмента. Для разделения пространства на глобальное и локальное в секторе сегмента выделяется один бит **ТИ** (индикатор таблицы: 0 – глобальный, 1 – локальный).

Физические адреса, выдаваемые МП во внешние блоки, являются 24-разрядными (т.е. объем основной памяти может достигать 16 Мбайт). С помощью 24-разрядной базы физического адреса, хранящейся в дескрипторе сегмента, вычисляется физический адрес, который равен базе физического адреса + смещение сегмента.

Одновременно на основании атрибута и размера сегмента, хранящегося в этом же регистре, обнаруживаются сегментные ошибки и ошибки памяти. При их возникновении инициируется обработка исключительной ситуации.

Селектор – это 16-разрядный указатель, с помощью которого выбираются дескрипторы. В логическом адресе селекторная часть определяет индивидуальный дескриптор.

Селектор имеет следующие поля:

- *Индекс* – биты 15-3. Индекс выбирает один из 8192 дескрипторов в таблице дескрипторов; 8192×8 (число байт в дескрипторе) = 64К – емкость сегмента;
- *Индикатор* таблицы ТИ – бит 3 определяет таблицу, на которую ссылается селектор: 0 – означает GDT, а 1 – текущую таблицу LDT;
- *Запрашиваемый уровень привилегий RPL* (2-0).

Регистры системных адресов GDTR, IDTR, LDTR и TR служат для обращения к элементам, которые управляют механизмом сегментации памяти.

Таковыми элементами являются таблицы и сегменты, входящие в структуру защиты памяти МП 286/386:

GDT	–	таблица глобальных дескрипторов
IDT	–	таблица дескрипторов прерываний
LDT	–	таблица локальных дескрипторов
TSS	–	сегмент состояния задачи.

Регистры GDTR и IDTR содержат 24-разрядные линейные адреса базы и 16-разрядные величины границ GDT и IDT, которые являются глобальными по отношению ко

всем задачам. Перед переходом в защищенный режим программа должна создать в ОЗУ таблицу GDT и загрузить регистр GDTR при помощи специальной команды LGDT.

Регистр таблицы локальных дескрипторов LDTR и регистр задачи TR содержит 16-разрядные селекторы сегментов LDT и TSS, которые определены для конкретной задачи. С каждым из них связан программно недоступный регистр дескрипторов сегмента.

В состав МП 80286 входит регистр *слова состояния машины* (это не регистр флагов! Он имеется само собой)

В его состав входят следующие поля:

- PE (Protected Mode Enable) – «защита разрешена». Устанавливается в 1 при работе в виртуальном режиме, 0 – в реальном;
- MP – бит присутствия сопроцессора. Если MP=1, то 286 будет учитывать команды WAIT, ESC;
- EM – режим эмуляции. Если EM=1, то МП при встрече команды ESC генерирует особый случай отсутствия сопроцессора. Процедура обработки особого случая в случае отсутствия сопроцессора может эмулировать команды сопроцессора программно;
- TS – бит “задача переключена”. Устанавливается автоматически, когда МП осуществляет переключение задач.

9.4.3 Дескрипторы сегментов и их таблицы

Каждый сегмент имеет свой дескриптор. Он имеет следующий формат (рис.9.7).



Рисунок 9.7 - Формат дескриптора сегмента процессора 80286

Дескрипторы сегментов располагаются в памяти ПЭВМ. Из четырех слов дескриптора одно не используется. Оно зарезервировано для последующего поколения процессоров (386 и т.д.).

В дескрипторе содержится 24-разрядный базовый адрес, 16 разрядов размера сегмента, 3 разряда типа, определяющие атрибут сегмента.

Кроме них имеются битовые поля:

- P – присутствие в основной памяти;
- A – обращение к сегменту;
- DPL – уровень привилегий дескриптора;
- S – бит системного сегмента. S=0 – системный; S=1 – сегмент программы или данных.

В поле "Тип" указывается, кодовый это сегмент или данных, разрешена запись в него или запрещена, расширение сегмента осуществляется вниз (для стека) или обычное.

Существуют 4 уровня привилегий сегмента: 0 – высший; 3 – низший. Процессор разрешает менее привилегированной процедуре вызвать более привилегированную, но ограничивает доступ разрешенными точками входа. Допустимые точки входа идентифицируются в процессоре специальными дескрипторами, называемыми шлюзами вызова. Привилегия в шлюзе задается достаточно низкой, что позволяет обращаться в точке входа к операционной системе.

Группа дескрипторов сегментов, расположенных в глобальном адресном пространстве называется *глобальной таблицей дескрипторов (GTD)*, а сегментов, расположенных в локальном адресном пространстве – *локальной таблицей дескрипторов (LDT)*.

Если число задач, выполняемых ЭВМ, равно N, то и число локальных адресов также равно N и соответственно количество LDT также равно N.

GTD и N-LTD располагаются в глобальном адресном пространстве (рис.9.8).

Следует помнить, что сама LDT также представляет собой отдельный сегмент и для его описания требуется дескриптор. Дескриптор LDT хранится в GTD.

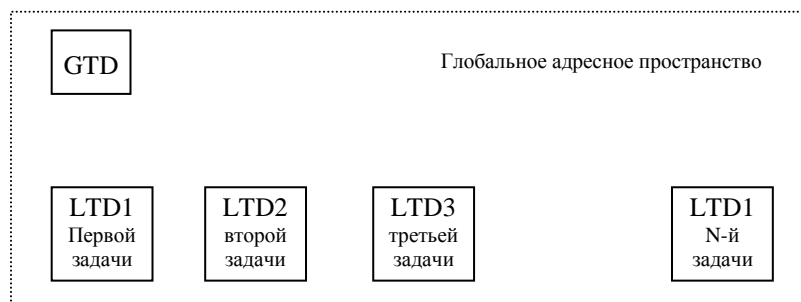


Рисунок 9.8 – Схема размещения таблиц дескрипторов в памяти ЭВМ

Глобальная таблица дескрипторов хранится в некоторой области основной памяти (внутри глобального адресного пространства). Ее базовый адрес и размер при запуске системы помещается в регистр GTD. После этого процессор на основании содержимого регистра GTD обращается к GTD.

9.4.4 Сегмент состояния задачи и переключение задач в процессоре 80286

Как уже упоминалось ранее, в случае приостановки выполнения команды задачи, информация, необходимая для ее перезапуска, должна сохраняться в *блоке управления задачей*, который размещается в *сегменте состояния задачи (CC3 – TSS)*. Для этого сегмента также используется свой дескриптор, который хранится в ГТД. Для выполнения N задач соответственно используется N сегментов состояния задачи. Таким образом, в ГТД хранятся N дескрипторов ЛТД и N дескрипторов CC3 (TSS).

Когда некоторая задача находится в состоянии прогона, в соответствующую часть регистра задачи (TR) помещается селектор, с помощью которого осуществляется считывание дескриптора CC3. Автоматически считанный из ГТД дескриптор CC3 помещается в регистр дескрипторов сегментов TR, в котором содержится базовый адрес CC3.

Сегмент состояния задачи состоит из 22-х слов (44 байта). В нем, помимо начальных значений всех регистров, хранятся также начальные значения указателей стеков 0 -, 1 – и 2-го уровней привилегий. В конце сегмента CC3 располагается *селектор ЛТД данной задачи*. При запуске задачи восстанавливается содержимое регистров процессора, а содержимое ЛТД автоматически помещается в селекторную часть РгЛТД.

В верхней части (с начала) сегмента состояния задачи располагаются данные (2 байта) о связи с задачей, которая выполнялась непосредственно перед этой задачей.

При переключении на другую задачу осуществляются следующие действия:

- 1) Состояние выполняемой задачи помещается в CC3;
- 2) Селектор новой задачи – в селекторную часть TR;
- 3) Дескриптор CC3 новой задачи в регистр дескрипторов TR;

4) Состояние новой задачи посылается в процессор. Кроме того, в процессор посылается селектор ЛТД, с помощью которого можно обратиться к новой ЛТД.

В 80286 эта процедура выполняется аппаратно, на что затрачивается ~180 тактовых импульсов (при $f=10$ МГц это составляет 18 нс).

В ССЗ хранится информация об обратной связи, на основании которой указывается задача, вызвавшая вложенную задачу. Т.е., если переключение происходит в результате прерывания, то автоматически устанавливается связь с задачей, выполняемой в момент возникновения прерывания. Таким образом, в случае возврата автоматически осуществляется возвращение к прежней задаче.

Планирование выполнения задач может быть осуществлено в программе управления задачами в ОС. Для этого при запуске программы управления задачами информацию о связи со следующей задачей следует поместить в поле обратной связи ССЗ. (Имеются команды загрузки LTR и сохранения STR регистра задачи.)

С помощью вентиля задачи системные программы могут ограничивать право переключаться на специфические задачи. Вентиль – логический элемент, пропускающий только определенные процессы.

9.4.5 Способы обращения процессора 80286 к памяти

Процессор имеет различные виды адресного пространства: физическое, логическое и линейное.

Физическое – это реальные адреса, используемые для выбора ячеек микросхем физической памяти, содержащих данные.

Логический адрес состоит из селектора и относительного адреса внутри сегмента.

Линейный адрес – это адрес, сформированный добавлением относительного адреса к базовому адресу сегмента.

Допустим, что процессор выполняет задачу, в которой используется некоторое локальное адресное пространство и ЛТД сегментов, расположенных в нем. Deskрипторы ЛТД-сегментов хранятся в ГТД. При инициировании этой задачи селектор ЛТД, служащий для выбора deskриптора сегмента, автоматически помещается в соответствующее поле регистра ЛТД (РгЛТД). Индекс, содержащийся в селекторе, дает смещение относительно базового адреса ГТД. Deskриптор сегмента имеет длину 4 слова или 8 байт. Индекс помещается в старшие 13 бит селектора. Поэтому смещение в памяти равно:

$$\text{Смещение} = \text{Числовое_значение_индекса} \times 8.$$

При таком способе обращения автоматически считываются deskрипторы сегментов ЛТД, хранящиеся в ГТД, и помещаются в свой РгЛТД. С помощью содержимого этого регистра можно обратиться в ЛТД. Схема преобразования виртуальных адресов в физические изображена на рис.9.9. В селекторной части регистра CS хранится селектор сегмента программы. Deskриптор сегмента программы, указываемый этим селектором, автоматически считывается из ЛТД и помещается в регистр deskрипторов сегментов CS. Содержимое его принимается за базу, а содержимое IP – за смещение, и на основании их значений осуществляется обращение к памяти. Процессор может одновременно обратиться к четырем сегментам. В этих сегментах уже задается смещение, поэтому формирование адреса может осуществляться с высокой скоростью.