# Практическое задание 3

1. **В структуре IA-32 можно выделить шесть основных блоков, работающих параллельно:**

* Блок интерфейса с магистралью - содержит драйвер адреса, схемы управления размером адреса и конвейером, мультиплексор, приемопередатчики и др. Этот блок обеспечивает интерфейс между МП и его окружением. Он принимает внутренние запросы для выборки команд от блока предварительной выборки команд и для обмена данными с исполнительным блоком и устанавливает приоритет этих запросов. Одновременно он генерирует или обрабатывает сигналы для исполнения текущего цикла магистрали. К ним относятся сигналы адреса, данных и управления для обращения к внешней памяти и устройствам ввода-вывода. При помощи схемы арбитра запросов блок управляет интерфейсом с внешними задатчиками магистрали и сопроцессорами.
* Блок предварительной выборки команд – нужен для того, чтобы заранее получать команды или данные перед их фактическим использованием. Когда блок интерфейса с магистралью не занимает цикла магистрали для исполнения команды, блок предвыборки команд использует его для последовательной выборки из памяти байтов команд. Эти команды хранятся в 16-байтовой очереди команд в ожидании обработки блоком декодирования команд.
* Блок декодирования команд - преобразует байты команды из этой очереди в микрокод. Декодированные команды в ожидании обработки исполнительным блоком хранятся в очереди команд, работающей по принципу FIFO (First In First Out). В Intel-386 эта очередь имеет размер 3 команды, а в Intel-486 - уже 5 команд, что позволяет этому МП при некоторых условиях выполнять по одной команде за цикл. Непосредственные данные и относительные адреса в коде операции также берутся из очереди команд.
* Исполнительный блок – выполняет команды из очереди команд и взаимодействует со всеми другими блоками, нужными для завершения выполнения команды. Для ускорения выполнения команд с обращением к памяти исполнительный блок приступает к их исполнению до завершения выполнения предыдущей команды. Так как команды с обращением к памяти встречаются очень часто, то благодаря такому перекрытию по времени производительность повышается.
* Блок управления сегментами - преобразует логические адреса в линейные по запросу исполнительного блока. Для ускорения этого преобразования текущие дескрипторы сегментов помещаются во встроенную кэш-память. Во время трансляции адресов блок управления сегментами проверяет, нет ли нарушения сегментации. Эти проверки выполняются отдельно от проверок нарушений статической сегментации, осуществляемых механизмом проверки защиты. Блок сегментации обеспечивает четыре уровня защиты (от 0 до 3) с целью изоляции и защиты друг от друга прикладных программ и операционной системы. Этот компонент также позволяет легко создавать перемещаемые программы и данные и обеспечивает их совместное использование. Полученный линейный адрес направляется в блок страничной трансляции.
* Блок страничной трансляции - позволяет прозрачно управлять пространством физических адресов независимо от управления сегментами. Каждый сегмент отображается в пространство линейных адресов, которое, в свою очередь, отображается в одну или несколько страниц объемом 4 Кб

1. Программная модель включает восемь регистров общего назначения, шесть регистров сегментов, указатель команд, регистр системных флагов, регистры системных адресов, четыре регистра управления и шесть регистров отладки.
2. •Регистр глобальной дескрипторной таблицы (GDTR). Содержит 32-битный линейный адрес и 16-битную границу глобальной дескрипторной таблицы. Значение этого регистра можно загрузить/сохранить при помощи привилегированных инструкций LGDT/SGDT. В реальном режиме этот регистр не используется. Перед переходом в защищенный режим в этот регистр следует загрузить корректные значения.

• Регистр локальной дескрипторной таблицы (LDTR). Содержит 16-битный селектор локальной дескрипторной таблицы. С регистром связан программнонедоступный кэш дескриптора для хранения базового адреса, предела и атрибутов соответствующей дескрипторной таблицы. Значение этого регистра можно загрузить/сохранить при помощи привилегированных инструкций LLDT/SLDT. В реальном режиме этот регистр не используется, и попытка обращения к нему генерирует особый случай "недействительный код операции" (исключение #6). С каждой задачей в защищенном режиме может быть связана своя локальная дескрипторная таблица, поэтому селектор LDT хранится в TSS и автоматически загружается при переключении задач.

• Регистр таблицы дескрипторов прерываний (IDTR). Указывает на таблицу точек входа в программы обработки прерываний. Регистр содержит 32-битный линейный базовый адрес и 16-битный предел таблицы. Значение этого регистра можно загрузить/сохранить при помощи привилегированных инструкций LIDT/SIDT. При инициализации МП базовый адрес IDT устанавливается в 0, а предел - 0FFFFh. В реальном режиме эта таблица хранит 4-байтные векторы прерываний, а в защищенном - 8-байтные дескрипторы шлюзов обработчиков прерываний и исключений. Это единственный регистр среди перечисленных, который используется в реальном режиме.

• Регистр задачи (TR). Указывает на информацию, необходимую МП для определения текущей задачи. Регистр содержит 16-битный селектор дескриптора сегмента состояния задачи. С регистром связан программнонедоступный кэш дескриптора TSS для хранения базового адреса, предела и атрибутов соответствующего сегмента состояния задачи. Значение этого регистра можно загрузить/сохранить при помощи привилегированных инструкций LTR/STR. В реальном режиме этот регистр не используется, и попытка обращения к нему генерирует особый случай "недействительный код операции" (исключение #6).

1. IA-32 может работать в следующих режимах:

* Реальный режим (Real mode): 16-битный режим, совместимый с процессорами Intel 8086/8088.
* Защищенный режим (Protected mode): 32-битный режим с защитой памяти и виртуальной адресацией.
* Режим виртуального 8086 (Virtual 8086 mode): Режим, позволяющий запускать программы, предназначенные для процессоров 8086, в защищенном режиме.
* Режим системного управления (System Management mode): Режим, предназначенный для использования в системах управления питанием и мониторинга оборудовани

1. Поля:

* Префикс - необязательная часть инструкции, которая позволяет изменить некоторые особенности ее выполнения. В команде может быть использовано сразу несколько префиксов разного типа. Типы префиксов: командные префиксы (префиксы повторения) REP, REPE/REPZ, REPNE/REPNZ; префикс блокировки шины LOCK; префиксы размера; префиксы замены сегмента.
* КОП - код операции.
* Байт "Mod R/M" определяет режим адресации, а также иногда дополнительный код операции. Необходимость байта "Mod R/M" зависит от типа инструкции.
* Байт SIB (Scale-Index-Base) определяет способ адресации при обращении к памяти в 32-битном режиме. Необходимость байта SIB зависит от режима адресации, задаваемого полем "Mod R/M".
* Кроме того, инструкция может содержать непосредственный операнд и/или смещение операнда в сегменте данных.

1. Типы префиксов:

* командные префиксы (префиксы повторения) REP, REPE/REPZ, REPNE/REPNZ;
* префикс блокировки шины LOCK;
* префиксы размера;
* префиксы замены сегмента.