**Ответы на контрольные вопросы.**

**1).** Основные блоки и их функциональное назначение в структуре IA-32:

- **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.

- **блок предварительной выборки команд:** функция опережающего просмотра программы, чтобы заранее получать команды или данные перед их фактическим использованием.

- **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).

- **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.

- **блок управления сегментами:** преобразует логические адреса в линейные по запросу

исполнительного блока.

- **блок страничной трансляции:** нужен для получения физических

адресов по линейным если механизм страничного преобразования включен. Позволяет прозрачно управлять пространством физических адресов независимо от управления сегментами.

**2). Какие регистры составляют программную модель IA-32?**

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.

**3). Регистры управления сегментированной памятью.**

Регистры управления сегментированной памятью, известные также как регистры

системных адресов, указывают на структуры данных, которые управляют механизмом

сегментированной памяти. Они определены для ссылок на таблицы или сегменты,

поддерживаемые моделью защиты МП.

∙ *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 программно-

недоступный 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 программно-

недоступный 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).

**4). В каких режимах может работать IA-32?**

*Непосредственный режим адресации* подразумевает включение операнда-источника в

код инструкции. Операнд может быть 8-битовым или 16-битовым, если значение

эффективного размера операнда - 16. Операнд может быть 8-битовым или 32-битовым,

если значение эффективного размера операнда - 32. Обычно непосредственные

операнды используются в арифметических инструкциях.

*Регистровый режим* *адресации* определят операнд-источник или операнд-приемник в

одном из регистров процессора или сопроцессора.

**5). Какие поля составляют команду IA-32?**



**6). Перечислите типы префиксов.**

- командные префиксы (префиксы повторения) REP, REPE/REPZ, REPNE/REPNZ;

- префикс блокировки шины LOCK;

- префиксы размера;

- префиксы замены сегмента.