

Модели адресации ОЗУ

Д. В. Луцив

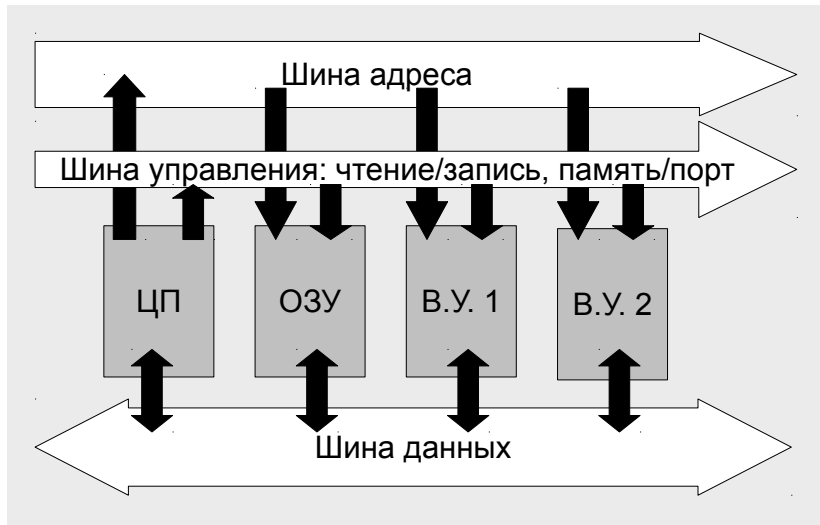
Кафедра системного программирования СПбГУ



CS220 (231000)

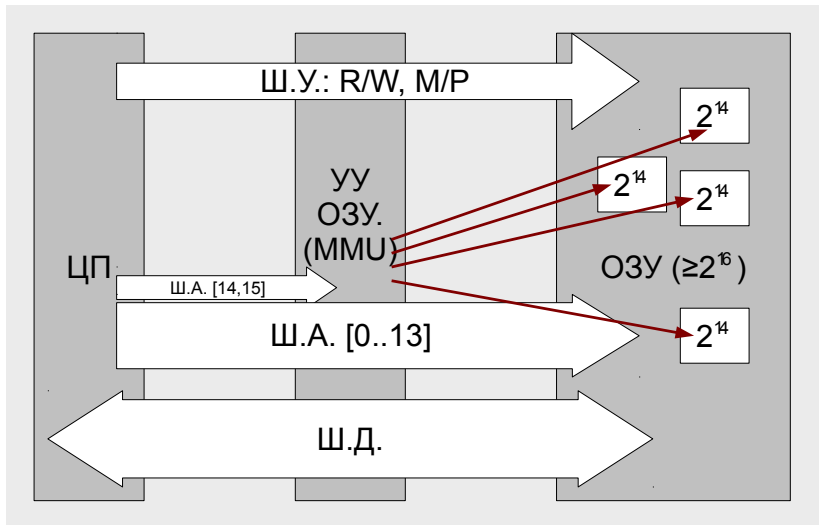
Содержание

- 1 Прямая адресация
 - Сущность
 - Свойства
- 2 Банковые расширения
 - Сущность
 - Свойства
 - Пример
- 3 Совместимые расширения ЦП
 - Расширение адресных регистров
 - Сегментная и страничная модели
- 4 Пример: Intel i80386



- Типично для восьмибитной шины данных — 16-битные внутренняя арифметика и шина адреса.
- Даже возможные 64KiB не использовались.

Для офисных приложений и игр — нужна графика и больше памяти.



Хорошо

- Программируется через порты
- Процессор тот же
- Дешево стоит
- Память можно расширять до бесконечности
- Вполне удобно работать с данными

Плохо

- Работает небыстро
- Тяжело менять банки кода, особенно для:
 - Вызова процедур
 - Прерываний
 - Переключения задач
- Разные реализации часто несовместимы

Хорошо

- Программируется через порты
- Процессор тот же
- Дешево стоит
- Память можно расширять до бесконечности
- Вполне удобно работать с данными

Плохо

- Работает небыстро
- Тяжело менять банки кода, особенно для:
 - Вызова процедур
 - Прерываний
 - Переключения задач
- Разные реализации часто несовместимы

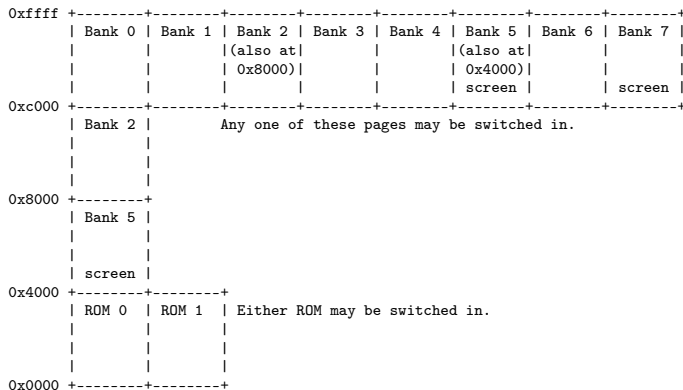
Окно в адресном пространстве

Особый вариант банковской системы.

- Некоторые внешние устройства, например видеокарты, поддерживают до сих пор
- Использовалась в EMS — сначала плата расширения, потом встроенная поддержка на материнской плате, потом эмуляция
- Использовалась в XMS — сразу чисто программное решение 286+

ZX Spectrum 128K

▶ Банки памяти



Расширение адресных регистров

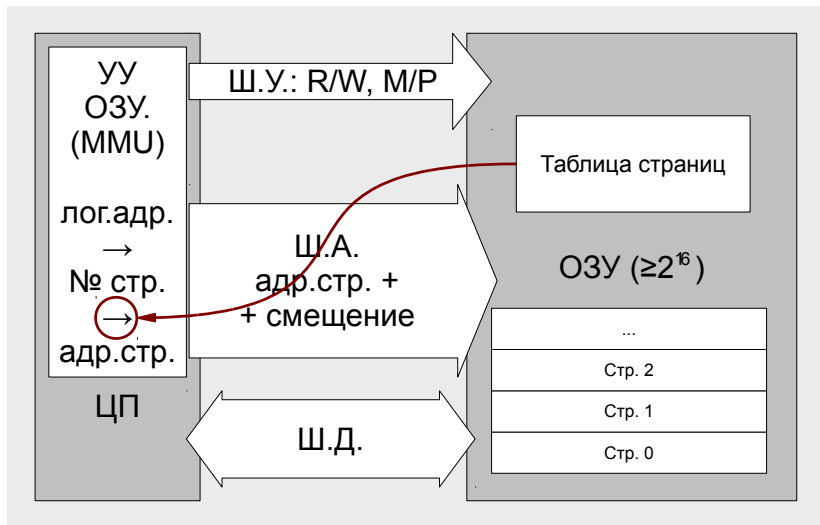
- Старый машинный код не знает, что у адресного регистра есть старшая половинка
- Все переходы и обращения к данным в старом коде «короткие»
- Т.к. старые приложения не используют всю память, появляется внутренняя фрагментация

Жёсткие сегменты

Дополнительный сегментный регистр для жестких сегментов

- Определяет, сколько надо прибавить к текущему адресу (с умножением на 16 (8086) или 256 (80286)). Сегменты пересекаются, это минимизирует внутреннюю фрагментацию.
- Инструкции передачи управления, вызова и прерывания с ним работают.
- Для данных тоже приятен, но отдельный.
- Хорошо бы еще и для стека (особенно, если много задач), и для данных сделать несколько.

Страничная адресация: Сущность



Страничная адресация: Свойства

- Страницы небольшие и не пересекаются
- В отличие от жестких сегментов, часто могут "гулять" по физическим адресам под управлением ОС.
- Для управляемых страниц нужны таблицы размещения в основной памяти, к которым обращается ЦП => нужен хороший (и спец-оптимизированный кэш)
- Удобны для реализации виртуальной памяти — их можно выгружать и загружать без ведома пользовательского процесса

Управляемые сегменты

- Как управляемые страницы, но произвольного размера
- Как и управляемые страницы, тоже требуют таблиц размещения
- Могут ассоциироваться с правами на хранение данных и кода и выполнение кода

Сегменты i80386

- Процесс получает сегмент (реже несколько) кода и сегмент (реже несколько) данных
- Процесс получает сегмент стека
- По прерыванию таймера происходит переход в другой сегмент кода (адрес полный обработчика в другом сегменте), при этом автоматом:
 - регистры сохраняются в стек (на то оно и прерывание)
 - меняются текущие сегменты стека и данных
- Сегмент логически до 2 ГиБ — адресные регистры расширены, для работы со всем регистром новые инструкции

Адресные регистры по 32 бита вместо 16

В реальном режиме старшие разряды игнорируются

Страницы i80386

- Память разбита на страницы по 4 КиБ, в отдельных таблицах указано их размещение в основной памяти. Независимо от сегментов. Это дает возможности:
 - выделять физическую память по мере использования, даже если процесс сразу запросил много
 - не перемещать данные физически, если надо перенастроить сегменты процесса
- Маленькие одинаковые страницы позволяют эффективно реализовать виртуальную память больше физической — с подкачкой. при отсутствии нужной страницы в физической памяти она загружается с диска. Тоже через прерывание.

При этом (I)

- Тормоза — сначала надо читать таблицы дескрипторов сегментов, потом страниц => нужен умный кэш
- Обратная совместимость — процесс может пользоваться младшими 16 битами адресного регистра, как и раньше
- Старым процессам, рассчитанным на жесткие сегменты, выделяются сегменты, пересекающиеся, как раньше (реальный режим)
- Про страницы пользовательский процесс не знает вообще ничего
- По прерываниям таймера переход идет на диспетчер процессов, а он уже дает переход на следующий процесс, выбирая его исходя из приоритета
- По прерываниям отсутствующих страниц переход идет на диспетчер виртуальной памяти

При этом (II)

- Виртуальная память позволяет эмулировать устройства, перехватывая отсутствие страниц, адрес которых задан в несуществующей физической памяти
 - Банковые расширения памяти — EMS, например — на месте окна EMS — страницы, физически расположенные в несуществующей памяти
 - Плоские адресные буферы внешних устройств, например, видеокарт, в действительности не поддерживающих их **UniVBE**. Для видеокарты задан сегмент в разделе памяти целиком из подложных страниц, а по прерыванию переключается её окно.
- 386 адресует до 4 ГиБ физической и до 64 ТиБ виртуальной, при этом виртуальная м.б. тоже реализована, как физическая, хоть и не адресуется напрямую (и вновь банки, ведь чем буфер НЖМД не банк?..)

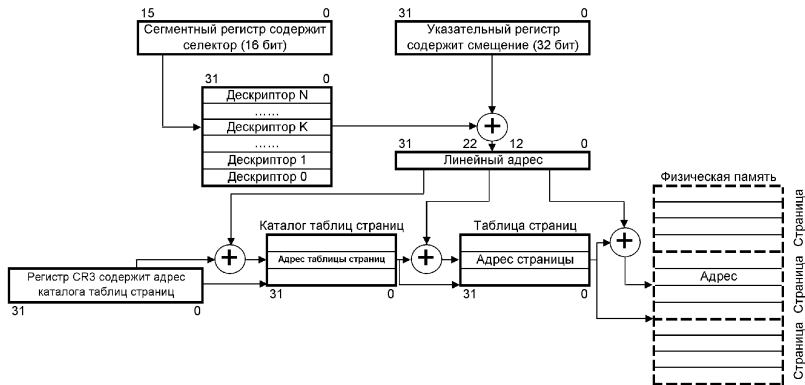
Модель памяти

▶ Страничное преобразование

Странично-сегментное преобразование.

▶ Ещё одно неплохое описание

▶ Забытая картинка из Wikipedia:



Дальнейшие расширения

- Устройства располагают свою память в 4-м ГиБ, так что, если установлено 4 ГиБ, часть памяти перекроется (будет недоступно).
- 4 ГиБ может быть тоже мало.

Проблема решается при помощи Physical Address Extension — доп. расширения, появившегося в Pentium Pro (на тот момент — для серверов). Расширение позволяет страницам памяти располагаться не в контексте 4 ГиБ ОЗУ, а в контексте 64 ТиБ ОЗУ, по объему совпадающим с виртуальной памятью. При этом изменяется формат дескрипторов сегментов и страниц. Пользовательские процессы обычно не работают с памятью такого объема, для них всё выглядит по-старому. Картина меняется для ядра. Программы, которым объем памяти критичен, можно пересобрать со специальными библиотеками.

