|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Рубежный контроль №1**

Дисциплина: Архитектура ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-56Б |  | 06.11.2021 | В.В. Леонов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.Ю. Попов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

# Вопрос 1. Виртуальная память: назначение и преимущества

Идея виртуальной памяти заключается в том, чтобы абстрагироваться от физического «железа». Программы видят память в некотором искаженном виртуализованном виде.

Причиной появления виртуальной памяти является неопределенность необходимого количества памяти на этапе определения архитектуры системы ЭВМ и связанные с этим трудности написания ПО, а также проблемы дальнейшего развития, улучшения и масштабирования системы. Таким образом, создание программного обеспечения происходит независимо от использующей его вычислительной машины (например, 64-битное приложение может корректно работать с устройством, имеющем 16 Гб оперативной памяти, хотя теоретическое максимальное количество памяти на устройстве может быть гораздо больше).

Используя виртуальное адресное пространство, можно:

1. Увеличить объем адресуемой памяти.
2. Использовать физическую память различного объема.
3. Возложить на аппаратную составляющую механизмы доступа к ВЗУ.
4. Сгладить разрыв в производительности ОП и ВЗУ.
5. Ускорить доступ к данным по последовательным адресам.
6. Способствует реализации защиты памяти.

Технически выделяют три основных метода осуществления механизма виртуальной памяти:

1. Системы с блоками различного размера (сегментная реализация).
2. Системы с блоками одинакового размера (страничная организация).
3. Системы смешанного типа (сегментно-страничная организация).

# Вопрос 2. Диаграмма состояний УА DDR SDRAM

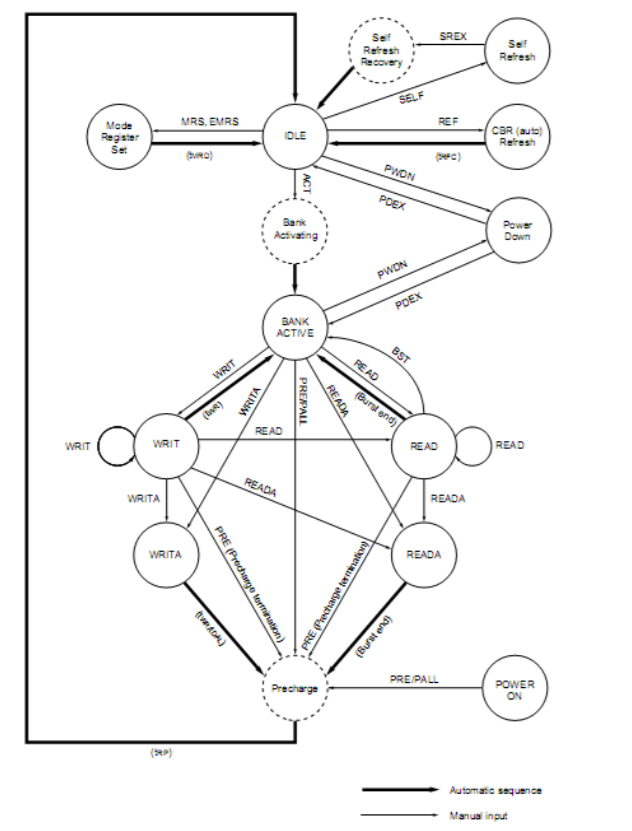


Рисунок 1. Диаграмма состояний УА DDR SDRAM

Данная диаграмма отражает переходы между состояниями автомата, который находится внутри памяти и управляет работой одного банка. Учитывая тот факт, что банков имеется большое количество и существует необходимость совмещения таких автоматов, часть состояний автомата относится сразу ко всем банкам памяти, а другая часть относится к каждому конкретному банку.

На данной диаграмме следующие состояния являются общими:

1. Mode Register Set
2. IDLE
3. Self Refresh Recovery
4. Self Refresh
5. CBR(auto) Refresh
6. Bank Activating
7. Power On
8. Power Down

В свою очередь оставшиеся относятся к каждому конкретному банку.

***Механизм работы****.*

При подаче питания попадаем в состояние **Power ON** и должны произвести некую цепочку инициализирующих команд (по стандарту **Precharge All**), т.е. перезарядить линии чтения/записи. После чего будет выполнен переход в состояние **IDLE**, когда все банки находятся в закрытом состоянии и линии чтения/записи перезаряжены. В этом состоянии можно сделать несколько инициализирующих действий: записать **Mode Register** (регистр управления), перейти в состояние **Power Down** и др. Для начала транзакции необходимо активировать банк (команда **Bank Activating**), на данное действие требуется достаточное количество времени, именно поэтому выделяют промежуточное состояние. К уже активированному банку можно обратиться с командами чтения/записи **Read/Writ**, **ReadA/WritA** (после которых автоматически следует внутренняя команда перезарядки линий). Следует отметить, что возможен прямой переход между состояниями **Writ>Read**, а в обратную сторону **Read>Writ** c задержкой (через состояние **Bank Active**).

Команда **Self Refresh** запускает внутренний контроллер регенерации, который будет перебирать адреса, пока не будет остановлен командой **SREX**. Это есть способ заполнения пауз в чтении/записи регенерацией. Команда **CBR(auto) Refresh** использует аналогичный счетчик однократно.

# Вопрос 3. Способы отображения ОП в кэш: наборно-ассоциативный способ отображения

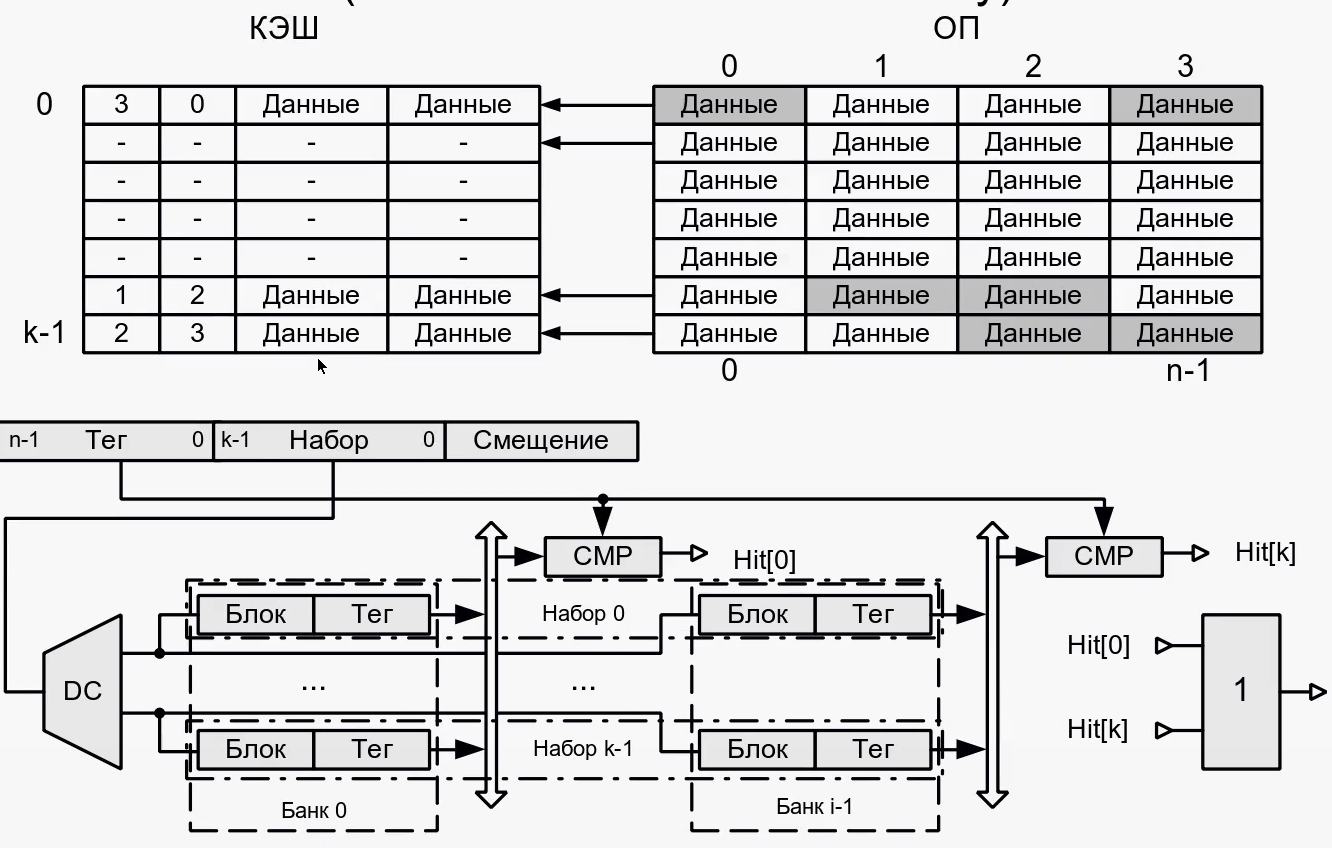


Рисунок 2. Схема наборно-ассоциативного отображения ОП в кэш

Является промежуточным способом отображения ОП в кэш между произвольной загрузкой и прямым размещением. Идея данного способа состоит в том, чтобы объединить несколько кэш с прямым размещением, дав вариативность, в какой из них производить запись значений. Как правило объединение происходит по 4 или по 8 банков. Данная величина называется степенью ассоциативности. Фактически появляется возможность размещения конфликтующих при прямом размещении данных в разные банки, что позволяет существенно сократить количество конфликтов. Строка в банках строго определяется адресом. Для любой ячейки памяти выделяется набор, в котором достаточно свободно выбрать одну из кэш-линеек. На сегодняшний день в современных процессорах используется объединение по 8 банков (для всех уровней кэша: **L1, L2, L3**).

Для реализации очистки ячеек используется в общем случае следующий алгоритм: к каждой линейке добавляется дополнительная информации о ее текущем состоянии и частоте использования и происходит очистка наименее используемых данных.

Наборно-ассоциативный способ отображения является наиболее распространённым благодаря своей вариативности. Один и тот же объем кэш памяти меняется как по горизонтали, так и по вертикали.