

Прямой доступ к памяти

Материал из Википедии — свободной энциклопедии

Прямой доступ к памяти (англ. Direct Memory Access, DMA) — режим обмена данными между устройствами или же между устройством и основной памятью (**RAM**), без участия **Центрального Процессора** (ЦП). В результате скорость передачи увеличивается, так как данные не пересылаются в ЦП и обратно.

Кроме того, данные пересылаются сразу для многих слов, расположенных по подряд идущим адресам, что позволяет использование т. н. «взрывного» (burst) режима работы шины — 1 цикл адреса и следующие за ним многочисленные циклы данных. Аналогичная оптимизация работы ЦП с памятью крайне затруднена.

В оригинальной архитектуре IBM PC был возможен лишь при наличии аппаратного DMA-контроллера (микросхема с номером 8237).

DMA-контроллер может получать доступ к системной шине независимо от центрального процессора. Контроллер содержит несколько регистров, доступных центральному процессору для чтения и записи. Регистры контроллера задают порт (который должен быть использован), направление переноса данных (чтение/запись), единицу переноса (побайтно/пословно), число байтов, которое следует перенести.

ЦП программирует контроллер DMA, устанавливая его регистры. Затем процессор даёт команду устройству (например, диску) прочитать данные во внутренний буфер. DMA-контроллер начинает работу, посылая устройству запрос чтения (при этом устройство даже не знает, пришёл ли запрос от процессора или от контроллера DMA). Адрес памяти уже находится на адресной шине, так что устройство знает, куда следует переслать следующее слово из своего внутреннего буфера. Когда запись закончена, устройство посылает сигнал подтверждения контроллеру DMA. Затем контроллер увеличивает используемый адрес памяти и уменьшает значение своего счётчика байтов. После чего запрос чтения повторяется, пока значение счётчика не станет равно нулю. По завершении цикла копирования устройство инициирует прерывание процессора, означающее завершение переноса данных. Контроллер может быть многоканальным, способным параллельно выполнять несколько операций.

Содержание

- [1 Захват шины \(bus mastering\)](#)
- [2 DMA и виртуальная память, IOMMU и AGP GART](#)
- [3 DMA и IDE/ATA, Ultra DMA](#)
- [4 Режимы Ultra DMA протокола ATA для ОС Linux](#)

Захват шины (bus mastering)

В шинах **MicroChannel**, **SBus**, разработанной под их большим влиянием **PCI** и её концептуальных производных **AGP** и **PCI-X** используется иная реализация DMA. Эти шины позволяют любому устройству заявить о возникновении потребности к захвату шины, каковая потребность удовлетворяется т. н. арбитром при первой возможности. Устройство, успешно осуществившее захват шины, самостоятельно выставляет на шину сигналы адреса и

управления, и исполняет в течение какого-то времени ту же ведущую роль на шине, что и ЦП. Доступ ЦП к шине при этом кратковременно блокируется.

В такой реализации DMA не существует DMA-контроллера, а также номера входа DMA-контроллера.

Некоторые старые устройства PCI, а именно реализации звуковых карт семейства Sound Blaster, использовали тот же DMA-контроллер 8237 из оригинальной архитектуры IBM PC. Такое использование является безусловно устаревшим для PCI, но поддерживалось с целью обеспечить полную совместимость по ПО и драйверу с версиями Sound Blaster для шины [ISA](#).

Данная поддержка называется Distributed DMA (D-DMA), и реализована аппаратным образом как в устройстве, так и в логике моста PCI-ISA, в которой на PCI-системах размещена и логика оригинального IBM PC DMA контроллера 8237. Реализация включает в себя 2 запроса — сначала от устройства мосту PCI-ISA, затем от моста основной памяти.

Кроме упомянутых реализаций Sound Blaster, практически никакие устройства PCI не используют понятие «номер входа DMA-контроллера», как и 8237 вообще.

DMA и виртуальная память, [IOMMU](#) и [AGP GART](#)

В ОС со [страничной виртуальной памятью](#), таких, как Windows и семейство UNIX, непрерывный регион виртуальных адресов может быть реализован разрывно расположенными физическими страницами.

Исполнение DMA по такому региону представляет собой довольно сложную задачу. Также сложной задачей является исполнение DMA по отгружаемой памяти.

Решение этой задачи требует выявления физических страниц, реализующих регион, и их блокировку от отгрузки обращением к подсистеме виртуальной памяти. Далее становится возможным нахождение физических адресов страниц региона, которые в общем случае не являются непрерывными и формируют так называемый scatter-gather list («список рассеяния/сборки») — SGL.

Задача исполнения DMA по такому списку может быть решена одним из следующих способов:

- выделение подряд идущей физической памяти в ядре ОС и промежуточное копирование всех данных туда/оттуда (т. н. «буфер отскока» — bounce buffer). Поддерживается всеми ведущими ОС, для активации поддержки в Windows требуется вызов IoGetDmaAdapter с DEVICE_DESCRIPTION::ScatterGather, установленном в FALSE.

Недостатки: трата времени процессора на копирование, потребление крайне ограниченного ресурса непрерывной физической памяти, занятие места в ограниченной части памяти, к которой есть доступ у DMA (первый гигабайт на x86).

- разбиение операции на подоперации по границам элементов SGL, с прерыванием в конце каждой операции. Использовалось в старых 8битных SCSI-контроллерах, поставляемых со сканерами типа HP ScanJet.

Недостатки: большое количество прерываний.

- поддержка SGL самим устройством, с требованием копирования SGL, преобразованного в формат, специфичный для устройства, в устройство через многочисленные обращения к регистрам устройства.

Недостатки: крайне высокая сложность устройства, невысокая производительность большого числа записей в регистры.

- поддержка SGL самим устройством, с требованием размещения SGL, преобразованного в формат, специфичный для устройства, в физически непрерывном регионе основной памяти. Устройство читает SGL тем же механизмом DMA с захватом шины, что и собственно данные, тем самым реализуя функциональность некоего процессора, читающего и исполняющего свою собственную «программу», реализованную как список дескрипторов SGL. Данная архитектура называется chain DMA («цепной DMA»), реализована в практически всем стандартном оборудовании современного компьютера — Intel IDE (в примитивном виде), UHCI и OHCI USB, OHCI 1394, а также в большинстве PCI-адаптеров Ethernet и SCSI (даже в устаревшем aic78xx). Как хороший пример реализации данной архитектуры в очень сложном и развитом виде см. спецификацию оборудования OHCI 1394. По некоторым сведениям, данная архитектура под названием «канальные программы» использовалась еще в IBM 360, известных в СССР как ЕС ЭВМ.

Недостатки: высокая сложность устройства, хотя и ниже в числе транзисторов, чем предыдущий вариант. Например, UHCI USB контроллер (см. спецификацию на сайте Intel) требует около 5000 транзисторов.

- поддержка SGL в межшинном оборудовании, при которой представление физически разрывного буфера для стороны устройства выглядит физически непрерывным. Такое оборудование называется IOMMU (IO memory management unit). Реализовывалось как на компьютерах фирмы Sun для шины SBus, так и на компьютерах DEC Alpha для шины PCI. До недавних времен практически никогда не реализовывалось в обычных x86/PCI системах, хотя в настоящее время есть тенденция к изменению этой ситуации, главным образом с целью повышения производительности гипервизоров виртуальных машин. Всегда реализовывалось для шины [AGP](#) под названием AGP [GART](#) с целью облегчить произвольный доступ трехмерного видеопроцессора к текстурам, находящимся в основной памяти. Со стороны устройства данное оборудование было стандартизировано спецификацией AGP, со стороны ПО стандартизации не существовало, и реализация зависела от производителя микросхемы северного моста между AGP и памятью (отсюда необходимость в «драйвере AGP», таком, как agr440.sys для микросхем Intel). Набор вызовов ядер развитых ОС, таких, как Windows, всегда содержал архитектурную абстракцию IOMMU (MapTransfer и GetScatterGatherList, эти же вызовы поддерживают и bounce buffer, понимаемый как разновидность IOMMU), позволяющую одному и тому же драйверу устройства поддерживать его при присоединении через различные IOMMU.

Недостатки: требование сложной логики уже не в устройстве, а в платформе.

DMA и IDE/ATA, Ultra DMA

Первоначальный контроллер жесткого диска IBM PC/AT не поддерживал DMA, и требовал передачи всех данных дискового ввода/вывода инструкциями REP INSW/REP OUTSW через порт 0x1f0.

В начале 90х годов диски MFM/RLL вымерли, сменившись дисками IDE, но регистровый интерфейс ПО к контроллеру не изменился.

Низкая производительность такого контроллера стала серьезной проблемой, особенно на системах PCI. Помимо требования нескольких циклов PCI на 2 байта переданных данных, это приводило к загрузке процессора дисковым вводом-выводом.

Для решения проблемы ряд компаний, в том числе Intel, разработали контроллеры IDE с поддержкой DMA. Контроллеры были и есть несовместимы по ПО между различными производителями, хотя совместимость всех Intel IDE/ATA/SATA снизу вверх более или менее поддерживается.

Также особенностью этой поддержки является использование новых команд протокола IDE/ATA, а значит, и требование поддержки DMA не только контроллером, но и самим жестким диском.

Около 2000 года поддержка DMA по шине IDE/ATA развилась в сторону увеличения тактовой частоты шины, что потребовало нового типа кабеля от контроллера к диску с удвоенным числом проводников меньшего размера. Эта технология называлась Ultra DMA (UDMA).

Многие операционные системы требовали действий администратора для использования IDE DMA. Так, например, стандартные ядра Linux до примерно 2004 года не имели такой поддержки, требовалось перестроение ядра с отредактированным файлом конфигурации.

В семействе Windows поддержка IDE DMA появилась сначала только для Intel в пакетах обновлений к Windows NT4, и требовала на большинстве систем ручного редактирования реестра для задействия.

В Windows 2000 это требование исчезло, но появилось требование обязательной вписки даже не-загрузочных дисков в BIOS и обязательного выставления режима DMA для них в настройках BIOS. Эти настройки BIOS становились видимы ядру ОС через технологию [ACPI](#), и ОС не позволяла включить DMA для диска, не вписанного в BIOS. Для сравнения: NT4 поддерживала и произвольный размер диска, и DMA без вписки диска в BIOS.

В системах Linux для включения IDE DMA также зачастую требуется команда `hdparm` (см. ниже).