

ОСНОВЫ КОМПЬЮТЕРНОЙ ТЕХНИКИ

Дисциплина: «Архитектура аппаратных средств»

Преподаватель: Солодухин Андрей Геннадьевич

Оперативная память для MS-DOS

Оперативная память для MS-DOS

- Для операционных систем реального режима (MS-DOS) оперативная память может размещаться в стандартной области (640 Кбайт), дополнительной памяти и в некоторых областях UMA.
- *Стандартная память* является самой дефицитной в РС, на ее небольшой объем (типовое значение — 640 Кбайт) претендуют и BIOS, и ОС реального режима, а остатки отдаются прикладному ПО.

Оперативная память для MS-DOS

- Дополнительная память и UMA могут использоваться в качестве оперативной только со специальными программными интерфейсами, для которых были разработаны спецификации EMS и XMS.
- *Спецификация па отображаемую память* (Expanded Memory Specification, EMS) — это программная спецификация на использование дополнительной памяти DOS-программами реального режима.
- Спецификация *LIM EMS* — соглашение фирм Lotus, Intel, Microsoft на применение EMS.
- С помощью специальных аппаратных или программных средств любая область дополнительной памяти может быть отображена на небольшие страницы, расположенные в области UMA.

Оперативная память для MS-DOS

- Для поддержки EMS поначалу требовались специальные аппаратные средства. В компьютерах на процессорах 386 и выше появилась возможность программной эмуляции EMS, которую в MS-DOS 5+ выполняет драйвер EMM386.EXE.

Оперативная память для MS-DOS

- Система EMS в основном предназначена для хранения данных — для исполняемого в данный момент программного кода она неудобна, поскольку требует программного переключения страниц через каждые 16 Кбайт.
- Ее используют для создания виртуальных дисков, хранения очередей заданий для печати, а также для хранения данных и даже программного кода некоторых резидентных программ (в целях экономии стандартной памяти).

Оперативная память для MS-DOS

- *Спецификация на расширенную память* (extended Memory Specification, XMS) — это иная программная спецификация на использование дополнительной памяти DOS-программами, разработанная компаниями Lotus, Intel, Microsoft и AST для компьютеров на процессорах класса 286 и выше.
- Эта спецификация позволяет программе получить в распоряжение одну или несколько областей дополнительной памяти, а также задействовать область HMA.
- Распределением областей ведаёт диспетчер расширенной памяти — драйвер HIMEM.SYS.

Виртуальная память

Виртуальная память

- Виртуальная память (virtual memory) представляет собой программно-аппаратное средство расширения пространства памяти, предоставляемой программе в качестве оперативной.
- Виртуальное пространство памяти разбито на страницы фиксированного размера, а в физической оперативной памяти в каждый момент времени присутствует только часть из них.
- Остальные страницы хранятся на диске, откуда операционная система может «подкачать» их в физическую память, предварительно выгрузив на диск часть не используемых в данный момент модифицированных страниц.

Виртуальная память

- Обращение процессора к ячейке виртуальной памяти, присутствующей в физической памяти, происходит обычным способом.
- Если же затребованная область в данный момент не отображена в физической памяти, процессор выдает исключение (внутреннее прерывание), по которому операционная система программно организует замещение страниц, называемое свопингом (swapping).
- Виртуальную память поддерживают процессоры, работающие в защищенном режиме, начиная с 80286, но реально ее широко стали применять только в операционных системах и оболочках для 32-разрядных процессоров (80386+).
- Виртуальная память используется лишь при наличии дополнительной памяти.

Виртуальная память

- Суммарный объем виртуальной памяти, доступной всем приложениям, определяется объемом ОЗУ и файлов подкачки (их может быть и несколько).
- Объем файла подкачки может быть постоянным или же изменяться динамически по мере изменения потребностей системы.
- Для того чтобы приложениям хватало памяти, на диске, несущем динамический файл подкачки, должно быть достаточно свободного пространства (десятки и сотни мегабайт).
- В принципе, файл подкачки может располагаться и на сетевом диске, но при этом трафик сети оказывается напряженным.
- Конечно же, важен и объем установленной физической памяти - ее нехватка может быть принципиальным ограничением на запуск ряда приложений или установку операционных систем.

Пространство ввода-вывода

Пространство ввода-вывода

- Процессоры семейства x86 имеют отдельную адресацию памяти и портов ввода-вывода.
- Это разделение обеспечивается выделением специальных инструкций ввода-вывода, с помощью которых возможна передача данных между портами и регистрами процессора (или портами и памятью).
- Инструкции ввода-вывода порождают шинные циклы обмена, в которых вырабатываются сигналы чтения из порта и записи в порт.

Пространство ввода-вывода

- В инструкциях ввода-вывода используется 16-битная адресация, что обуславливает размер пространства ввода-вывода в 64 кбайт.
- Для дешифрации адресов портов в оригинальном ПК из 16 бит использовались только младшие.
- Старшие биты адреса хотя и поступают на шину, но устройствами игнорируются.

Пространство ввода-вывода

- Каждой шине назначается своя область адресов ввода.
- Дешифратор адресов, расположенный на системной плате, при чтении открывает соответствующие буферы данных.
- Так что реально считываться будут данные только с одной шины.
- При записи в порты данные (и сигнал записи) могут распространяться и по всем шинам компьютера (широковещательно).

Пространство ввода-вывода

- Для современных плат со встроенной периферией и несколькими шинами (ISA, PCI) распределением адресов управляет BIOS через регистры конфигурирования чипсета.
- А в иерархии шин PCI (и PCI-E) диапазоны адресов задаются программированием мостов.

Аппаратные прерывания

Аппаратные прерывания

- Аппаратные прерывания обеспечивают реакцию процессора на события, происходящие асинхронно по отношению к исполняемому программному коду.
- По возникновении такого события адаптер (контроллер) устройства формирует запрос прерывания, который поступает на вход контроллера прерываний.
- Контроллер прерываний формирует общий запрос прерывания для процессора, а когда процессор подтверждает получение этого запроса, контроллер сообщает процессору вектор прерывания, по которому выбирается программная процедура обработки прерываний.

Аппаратные прерывания

- Процедура должна выполнить действия по обслуживанию данного устройства, включая сброс его запроса, и отправить команду завершения в контроллер прерываний, что дает возможность отреагировать на следующие события.
- Логика работы системы прерываний и программная модель контроллера прерываний, совместимого с микросхемой i8259A, являются важной частью стандартизации архитектуры PC-совместимых компьютеров.

Немаскируемые прерывания

Немаскируемые прерывания

- *Немаскируемые прерывания* (Non-Maskable Interrupt, NMI) в РС используются для сигнализации о фатальных аппаратных ошибках.
- На немаскируемое прерывание процессор реагирует всегда (если завершено обслуживание предыдущего немаскируемого прерывания); этому прерыванию соответствует фиксированный вектор 2.

Немаскируемые прерывания

- Источниками *прерывания SMI* являются схемы чипсета, участвующие в управлении энергопотреблением, а также контроллер USB при эмуляции традиционных клавиатуры и мыши.
- Это прерывание имеет наивысший приоритет и обслуживается несколько иначе, чем «классические» прерывания.
- Здесь процессор не выполняет вызов процедуры, описанной в таблице прерываний, а переходит в режим SMM, что сопровождается установкой сигнала SMIACT#, по которому вместо обычной памяти процессору становится доступной память SMRAM.
- Выход из режима SMM происходит по выполнению инструкции RSM, завершающей процедуру обработки SMI.
- После обработки SMI возможен рестарт (повторное исполнение) инструкции останова (HALT) и инструкций ввода-вывода.

Немаскируемые прерывания

- Возможность рестарта инструкции ввода-вывода используют, например, когда прикладная программа (или системный драйвер) пытается обратиться операцией ввода-вывода к периферийному устройству, находящемуся в «спящем» режиме.
- Обработчик SMI «будит» устройство, после чего операция ввода-вывода рестартует и прикладное ПО (или драйвер) «не замечает», что устройство пребывало в спячке.
- Таким образом, управление потреблением может быть организовано на уровне BIOS способом, совершенно прозрачным для программного обеспечения (в том числе и ОС).
- Аналогично SMI позволяет незаметно выполнять манипуляции с контроллером USB по обращению программы к порту контроллера клавиатуры (портам 60h и 61h).

Маскируемые прерывания

Маскируемые прерывания

- Маскируемые прерывания используются для сигнализации о событиях в устройствах.
- Реакция процессора на маскируемые прерывания может быть задержана сбросом его внутреннего флага IF (инструкция CLI запрещает прерывания, STI - разрешает).
- По возникновении события, требующего реакции, адаптер (контроллер) устройства формирует *запрос прерывания*, который поступает на вход *контроллера прерываний*.
- Задача контроллера прерываний — довести до процессора запрос прерывания и сообщить вектор, по которому выбирается программная процедура обработки прерываний.
- В IBM PC-совместимых компьютерах применяется два основных типа контроллеров прерываний:

Маскируемые прерывания

- *Периферийный контроллер прерываний* (Peripheral Interrupt Controller, PIC) программно совместим с традиционным контроллером 8259А, использовавшимся еще в первых моделях IBM PC.
- Со времен IBM PC/AT применяется связка из пары каскадно соединенных контроллеров PIC, позволяющая обслуживать до 15 линий запросов прерываний.

Маскируемые прерывания

- *Усовершенствованный периферийный контроллер прерываний (Advanced Peripheral Interrupt Controller, APIC)* введен в компьютеры для поддержки мультипроцессорных систем на базе процессоров 4-5-го поколений (486 и Pentium) и используется поныне для более поздних моделей процессоров.
- Помимо поддержки мультипроцессорных конфигураций, современный контроллер APIC позволяет увеличивать число доступных линий прерываний и обрабатывать запросы прерываний от устройств PCI, посылаемые через механизм сообщений (MSI).

Маскируемые прерывания

- Компьютер, оснащённый контроллером APIC, обязательно имеет возможность функционировать и в режиме, совместимом со стандартной связкой пары PIC.
- Этот режим включается по аппаратному сбросу (и включению питания), что позволяет использовать старые ОС и приложения MS-DOS, «не знающие» APIC и мультипроцессирования.

Маскируемые прерывания

- На входы контроллеров прерываний поступают запросы от стандартных устройств (клавиатура, системный таймер, CMOS-таймер, сопроцессор), периферийных контроллеров системной платы и от карт расширения.
- Традиционно все линии запросов, не занятые перечисленными устройствами, присутствуют на всех слотах шины ISA/EISA.
- Эти линии обозначаются как IRQ_x и имеют общепринятое назначение.
- Часть этих линий отдается в распоряжение шины PCI.

Маскируемые прерывания

- Каждому устройству, для поддержки работы которого требуются прерывания, должен быть назначен свой номер прерывания.
- Назначения номеров прерываний выполняются с двух сторон:
- во-первых, адаптер, нуждающийся в прерываниях, должен быть сконфигурирован на использование конкретной линии шины (джамперами или программно);
- во-вторых, программное обеспечение, поддерживающее данный адаптер, должно быть проинформировано о номере применяемого вектора.
- ОС имеют возможность изменить назначение запросов относительно распределения, сделанного через CMOS Setup.

Маскируемые прерывания

- В процессе назначения прерываний может участвовать система PnP для шин ISA и PCI; для распределения линий запросов между шинами служат специальные параметры CMOS Setup.
- Современные ОС имеют возможность изменить назначение запросов относительно распределения, сделанного через CMOS Setup.

Традиционный контроллер прерываний - PIC

Традиционный контроллер прерываний - PIC

- Контроллер прерываний (PIC) 8259A является периферийным устройством, которое связано с процессором через ту или иную шину расширения ввода-вывода.
- По этой шине процессор может обращаться к регистрам контроллера, программируя его режимы и управляя им, а также получать от контроллера 8-битный *вектор прерывания*, для чего в интерфейсе системной шины процессора и шины расширения имеется специальная команда подтверждения прерывания (*INTA*).
- Контроллер 8259A имеет 8 входов запросов от источников и один выход общего запроса.
- Каждому из входов соответствует свой вектор; программированием регистров контроллера задается номер вектора для входа 0, остальным входам соответствуют последующие номера векторов.

Традиционный контроллер прерываний - PIC

- Каждый вход может быть программно *замаскирован* — тогда он не вызывает сигнал общего запроса.
- Контроллер занимает два адреса в пространстве ввода-вывода, программное обращение позволяет управлять режимами работы контроллера, а также приоритетами и масками запросов.
- Кроме того, контроллер позволяет работать в *режиме опроса* (poll mode), или *полиинга* (polling), обеспечивая идентификацию источника прерывания (с учетом приоритетов) без выработки общего запроса.

Традиционный контроллер прерываний - PIC

- С каждым входом запроса в контроллере связано по одному биту в регистрах IRR, IMR и ISR; бит 0 каждого из этих регистров относится ко входу 0, бит 1 - ко входу 1, бит 7 - ко входу 7:
- IRR (Interrupt Request Register) — регистр запросов прерываний. В этом регистре бит устанавливается при обнаружении сигнала прерывания на соответствующем входе, независимо от маски.
- IMR (Interrupt Mask Register) — регистр масок прерываний. Единичное значение бита означает замаскированность данного входа — по запросу с замаскированного входа общий запрос прерывания не генерируется.
- ISR (Interrupt Service Register) — регистр обслуживаемого прерывания.

Традиционный контроллер прерываний - PIC

- Контроллер прерываний позволяет программировать свои входы на чувствительность к уровню или перепаду сигнала.
- *Чувствительность к уровню (level sensitive)* означает, что контроллер прерываний вырабатывает запрос прерывания процессора по факту обнаружения определенного уровня на входе IRQx.
- Если к моменту завершения обработки этого запроса (после записи команды *EOI* в регистр контроллера прерываний) контроллер снова обнаруживает активный уровень на том же входе DRQx, то он снова формирует запрос на прерывание процессора.
- *Чувствительность к перепаду (edge sensitive)* означает, что контроллер прерываний вырабатывает запрос прерывания процессора только по факту обнаружения перепада (на ISA — положительного) на входе IRQx. Повторно запрос по этому входу возможен только по следующему такому же перепаду, то есть сигнал предварительно должен вернуться в исходное состояние.

Традиционный контроллер прерываний - PIC

- В любом случае сигнал запроса аппаратного прерывания IRQ_x должен удерживаться генерирующей его схемой, по крайней мере, до цикла подтверждения прерывания процессором - именно в этот момент PIC определяет самый приоритетный незамаскированный запрос и по нему формирует вектор.

Традиционный контроллер прерываний - PIC

- Если к этому моменту запрос окажется снятым, источник прерывания корректно идентифицирован не будет, и контроллер сообщит *ложный вектор прерывания* (spurious interrupt), соответствующий его входу с максимальным номером (IRQ7 для первого контроллера и IRQ15 для второго).
- Обычно периферийные устройства строят так, что сигнал запроса сбрасывается при обращении программы обслуживания прерывания к соответствующим регистрам адаптера, так что ложных прерываний возникать не должно.

Традиционный контроллер прерываний - PIC

- Один контроллер PIC 8259A позволяет обслуживать 8 запросов прерываний; в PC/AT применяется *каскадное соединение двух контроллеров*, один из которых является ведущим, другой — ведомым.
- *Ведущий контроллер 8259A#1* обслуживает запросы 0, 1, 3-7; его выход подключается ко входу запроса прерываний процессора.
- К его входу 2 подключен *ведомый контроллер 8259A#2*, который обслуживает запросы 8-15.
- При этом поддерживается вложенность приоритетов — запросы 8-15 со своим рядом убывающих приоритетов вклиниваются между запросами 1 и 3 ведущего контроллера, приоритеты запросов которого также убывают с ростом номера.
- В XT каскадирование не применялось, и один контроллер 8259A обслуживал все 8 линий запросов.

Традиционный контроллер прерываний - PIC

- В IBM PC/XT/AT используется специальный режим вложенных прерываний с фиксированным приоритетом и автоматическим неспецифическим завершением.
- После инициализации (процедурой POST и при загрузке ОС) все неиспользуемые входы контроллеров замаскированы (на запросы прерываний не реагируют), а их векторы прерываний указывают на «заглушку» — процедуру с единственной инструкцией IRET.

Традиционный контроллер прерываний - PIC

- Для подключения обработчика прерывания от устройства первым делом следует загрузить обработчик в память и установить указатель на него в таблице прерываний.
- Далее следует демаскировать соответствующий ему вход в контроллере прерываний.
- Если обработчик прерывания удаляется из памяти, предварительно должен быть замаскирован соответствующий ему вход контроллера.
- Все изменения в таблице прерываний должны выполняться при замаскированных прерываниях, чтобы избежать попытки использования вектора в процессе его модификации (это приведет к «вылету» программы — обращению по некорректному адресу).

Традиционный контроллер прерываний - PIC

- В современных системных платах функции контроллеров прерываний возлагаются на чипсет, который может иметь и более гибкие средства управления, чем пара контроллеров 8259А.
- Процедура инициализации контроллеров может и отличаться от традиционной, но ею занимается тест POST, который «знает» особенности системной платы.

Традиционный контроллер прерываний - PIC

- Однако в операционном режиме всегда сохраняется программная совместимость с 8259А.
- При работе с контроллером прерываний от программы требуется лишь управлять маской своего запроса (при инициализации программы нужно обнулить маску требуемого запроса) и корректно завершать обработку прерываний.
- Каждая процедура обработки аппаратного прерывания должна завершаться командой EOI (End Of Interruption), посылаемой контроллеру.
- Некорректно завершенная процедура не позволит повторно использовать данный или другие запросы прерываний.

Улучшенный контроллер прерываний -APIC

Улучшенный контроллер прерываний -APIC

- Контроллер APIC в первую очередь предназначен для симметричных мультипроцессорных систем (2, 4 процессора), в которых все процессоры разделяют общие устройства ввода-вывода и общие контроллеры прерываний.
- Однако APIC используется и в однопроцессорных системных платах.
- Система с APIC состоит из локальных контроллеров, установленных в процессорах, и контроллеров прерываний от ввода-вывода (одного или нескольких).

Улучшенный контроллер прерываний -APIC

- Все контроллеры APIC соединены между собой локальной шиной, по которой они обмениваются друг с другом сообщениями.
- Задача каждого *локального контроллера* (local APIC) - трансляция сообщений, принятых по локальной шине, в сигналы, вызывающие все аппаратные прерывания своего процессора, - маскируемые (INTR), немаскируемые (NMI) и системного управления (SMI).
- Кроме того, локальные контроллеры APIC позволяют каждому процессору генерировать прерывания для других процессоров.
- Локальный контроллер имеет внутренний интервальный таймер, позволяющий вырабатывать прерывания через программируемый интервал времени.

Улучшенный контроллер прерываний -APIC

- *Контроллер прерываний от ввода-вывода (I/O APIC)* преобразует запросы аппаратных прерываний от устройств в сообщения протокола локальной шины APIC.
- В мультипроцессорном режиме он отвечает за распределение прерываний по процессорам, для чего может потребоваться статическое или динамическое распределение.
- В случае статического распределения для каждого номера прерывания указывается номер процессора, который его обслуживает.
- В случае динамического распределения каждое прерывание направляется наименее приоритетному в данный момент процессору.



Проблема разделяемых прерываний

Проблема разделяемых прерываний

- Линии запросов прерываний в компьютере, насыщенном периферийными устройствами, являются самым дефицитным ресурсом, поэтому приходится использовать эти линии совместно, то есть применять *разделяемые прерывания (shared interrupts)* между несколькими устройствами.
- Обработчики прерываний (программы) от разных устройств, имеющих одну линию запроса (и следовательно, общий вектор прерывания), должны быть выстроены в цепочку.
- В процессе обработки прерывания очередной обработчик в цепочке чтением известного ему регистра своего устройства должен определить, не это ли устройство вызвало прерывание.

Проблема разделяемых прерываний

- Если это, то обработчик должен выполнить необходимые действия и сбросить сигнал запроса прерывания от своего устройства, после чего передать управление следующему обработчику в цепочке; в противном случае он просто передает управление следующему обработчику.
- Чтобы прерывания, одновременно возникающие от нескольких устройств, не терялись, контроллер прерываний должен быть *чувствительным к уровню*, а не к перепаду на входе запроса.

Проблема разделяемых прерываний

- В соответствии со схемотехникой логики ТТЛ и КМОП активным уровнем должен быть низкий; выходной формирователь сигнала запросов у адаптеров должен обладать открытым коллектором (ТТЛ) или открытым стоком (КМОП); вход запроса у контроллера должен быть «подтянут» к высокому уровню резистором.
- Тогда непосредственное соединение этих выходов со входом контроллера («монтажное И») даст требуемый результат в аппаратном плане, а в программном плане необходимо корректно выстроить обработчики в цепочку.

Проблема разделяемых прерываний

- Поясним, почему надежная разделяемость при чувствительности к перепаду на линии запроса невозможна.
- Если устройство 1 выработает сигнал запроса после того, как его выработает (но еще не снимет) устройство 2, то контроллер обработает только один запрос.
- Цепочка программных обработчиков окажется ненадежной: если обработчик устройства 1 в этой цепочке проверит свое устройство до возникновения прерывания, то прерывание будет потеряно.
- Поскольку прерывания по своей природе обычно асинхронны, работа этих устройств совместно с поддерживающими программами окажется загадочно нестабильной.

Проблема разделяемых прерываний

- Для шины PCI с аппаратной точки зрения проблема разделения прерываний решена — здесь активным уровнем запроса является низкий, и контроллер прерываний чувствителен к уровню, а не к перепаду.
- Для шины ISA с ее запросами прерываний по положительному перепаду разделяемость прерываний невозможна.
- Исключения составляют системные платы и устройства с поддержкой ISA PnP, которые можно заставить работать и по низкому уровню.

Проблема разделяемых прерываний

- После успешного решения аппаратной задачи обеспечения разделяемости линий запроса возникает задача *идентификации источника* каждого прерывания, что позволило бы запустить соответствующую процедуру обработки.
- Желательно, чтобы эта задача решалась средствами ОС и с минимальными потерями времени.
- В первых версиях (до PCI 2.2 включительно) не было общепринятого способа программной индикации и запрета прерываний.

Проблема разделяемых прерываний

- Обработчики прерываний устройств должны вести себя корректно, учитывая возможность попадания в цепочку обработчиков разделяемого прерывания.
- Встречается типичная ошибка обработчика прерываний: прочтя регистр состояния устройства и не обнаружив признака запроса, драйвер «на всякий случай» выполняет сброс всех источников запроса (а то и сброс всего устройства).
- Эту ошибку порождает незадачливый разработчик драйвера, не учитывающий возможности разделения прерываний и не доверяющий разработчикам аппаратных средств.

Прямой доступ к памяти - DMA

Прямой доступ к памяти - DMA

- *Прямой доступ к памяти* (Direct Memory Access, DMA) - это обмен между системной памятью (ОЗУ) и устройством, выполняемый без непосредственного участия процессора.
- Обмен осуществляет *контроллер прямого доступа*.
- Для устройств ISA в архитектуре PC/AT присутствует централизованный контроллер DMA, совместимый с «исторической» микросхемой i8237A.

Прямой доступ к памяти - DMA

- Для устройств шины PCI (и всех ее «родственников») контроллер является частью устройства — мастера шины (bus master).
- Штатного централизованного контроллера DMA, как это было в архитектуре ISA, для шины PCI нет.
- Многоканальный контроллер DMA, программируемый по командам от центрального процессора, присутствует на системной плате PC-совместимого компьютера.

Прямой доступ к памяти - DMA

- Первоначально он использовался устройствами шины ISA
- Теперь он используется интегрированной традиционной периферией: контроллером НГМД, портами LPT и COM и некоторыми встроенными аудиосредствами.
- Процессор при обмене по DMA занят только инициализацией контроллера, которая сводится к записи в его регистры нескольких байтов, задающих начальный адрес и размер пересылаемого блока памяти, направление и режим обмена.

Аксессуары системной платы IBM PC

Аксессуары системной платы IBM PC

- Системная плата первой модели PC содержала несколько функциональных узлов, которые благодаря открытому описанию приобрели надежный статус неприкосновенности.
- Гарантии неприкосновенности были получены благодаря несчетным количеством программ и программных продуктов, их использующих.

Аксессуары системной платы IBM PC

Эти узлы перечислены далее:

1. Схемы предоставления системных ресурсов — памяти, ввода-вывода, прерываний, прямого доступа к памяти.
2. Микросхемы ROM BIOS с программным кодом начального тестирования, запуска и функций ввода-вывода.
3. Системный таймер, реализованный на микросхеме 8253 и использовавшийся как генератор запросов регенерации памяти, интервальный таймер и тональный генератор для динамика. В AT те же функции выполняла аналогичная микросхема 8254.

Аксессуары системной платы IBM PC

4. Системный порт AT, предназначенный для управления немаскируемыми прерываниями и звуком.
5. Канал управления звуком (PC speaker) - логическая схема, использующая тональный сигнал таймера и программно-управляемые биты системного порта. На машинах AT такой «синтезатор» может исполнять даже записанную музыку и речь.
6. Последовательный интерфейс клавиатуры и мыши, реализуемый на AT с помощью микроконтроллера 8042 (в конструктиве VTX его уже нет).
7. Память конфигурации и часы-календарь - CMOS RTC - узел, появившийся с AT.

Системный таймер - 8253/8254

Системный таймер - 8253/8254

- Во всех моделях PC используется трехканальный счетчик-таймер, выполняющий:
 - генерацию прерываний от системных часов, вызывающих инкремент счетчика системного времени в ячейке 40:006E области переменных (data area) BIOS;
 - генерацию запросов на регенерацию памяти;
 - генерацию звуковых сигналов.

Системный таймер - 8253/8254

- В качестве счетчиков-таймеров в ХТ применялась микросхема i8253, а в АТ — более быстродействующая i8254, которая с процессорами 80286 могла работать без тактов ожидания.
- На современных системных платах те же функции берет на себя чипсет, сохраняя программную совместимость с 8253/8254.
- Микросхемы 8253 и 8254 представляют собой трехканальные программируемые счетчики-таймеры, функционально почти совпадающие, но имеющие разное быстродействие (со стороны системной шины) и разное назначение выводов.

Системный таймер - 8253/8254

- Внутренние счетчики микросхемы имеют разрядность 16 бит, но общение с ними возможно только 8-битными операциями.
- При этом можно задавать значение только младшего байта счетчика (LSB), только старшего (MSB) или обоих (LSB/MSB), причем сначала передается младший, а потом старший байт.
- Программирование микросхемы осуществляется записью байтов в управляющий регистр по отдельности для каждого канала.

Канал управления звуком - PC
speaker

Канал управления звуком - PC speaker

- Стандартный канал управления звуком PC speaker рассчитан на подключение высокоомного малогабаритного динамика.
- Звук формируется из тонального сигнала от второго канала таймера
- Работой этого канала можно программно управлять.
- Частоту сигнала (тон) можно изменять, программируя коэффициент деления счетчика.

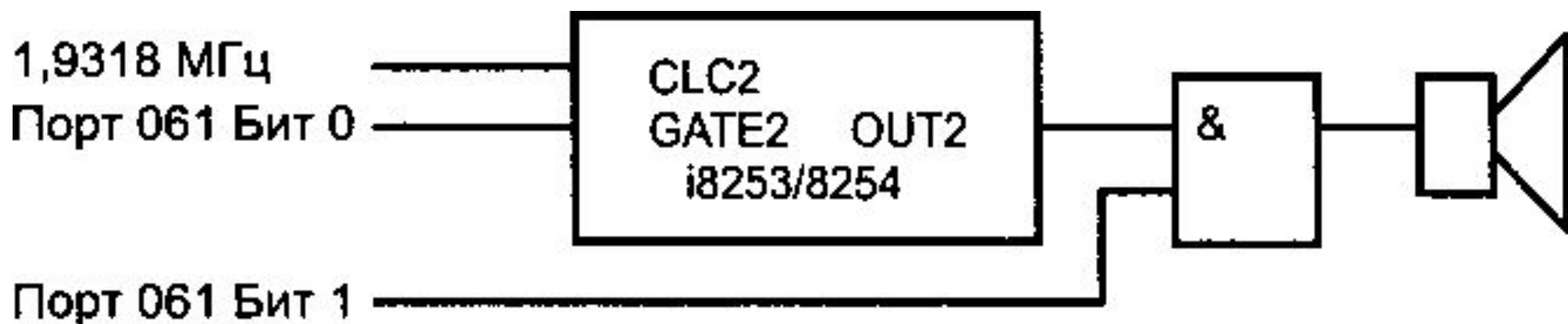
Канал управления звуком - PC speaker

- Разрешая/запрещая формирование сигнала программно-управляемым битом 1 системного порта 61h, можно подавать сигналы определенной длительности.
- Такой способ формирования звука мало загружает даже процессор 8086/88 и позволяет исполнять незамысловатые мелодии, причем в фоновом режиме, посылая команды из очереди по прерываниям от системного таймера.

Канал управления звуком - PC speaker

- С учетом физиологии слуха (инерционности восприятия) быстрым переключением частот можно достигать эффекта псевдомногоголосия.
- Таймер генерирует выходной сигнал при высоком уровне на входе GATE2 (при единичном значении бита 0 порта 61h).
- При низком уровне на входе GATE2 таймер формирует высокий уровень на выходе.

Канал управления звуком - PC speaker



Канал управления звуком

Канал управления звуком - PC speaker

- Более интересные звуки можно извлекать, используя принцип широтно-импульсной модуляции, программно осуществляемый через бит 1 порта 61h.
- При этом на входе GATE таймера должен быть низкий уровень (бит 0 порта 61h должен быть нулевым), чтобы на выходе OUT2 установился высокий уровень.
- В этом случае динамик играет роль фильтра нижних частот (инерционного звена) демодулятора.

Канал управления звуком - PC speaker

- Процессоры, начиная с 80286, способны формировать такой поток управляющих сигналов, который позволяет воспроизводить музыкальный или речевой сигнал с качеством карманного приемника.
- Это практически полностью загружает, например 286-й процессор.

Канал управления звуком - PC speaker

- Кроме того, качество воспроизведения в значительной степени зависит от частотных свойств динамика.
- Предпочтительнее более крупные динамики, у которых лучше воспроизведение нижних частот, - с ними можно добиться даже разборчивости речи.
- Возможно также комбинированное управление обоими битами порта 61h одновременно с программированием коэффициента деления счетчика-таймера.

Канал управления звуком - PC speaker

- Для ранних версий Windows существует даже драйвер, обеспечивающий извлечение звука через эти незамысловатые аппаратные средства.
- Однако в стандартную поставку ОС он не входит.
- В современной аппаратуре для звуковоспроизведения (и звукозаписи) используется аудиокодек.
- Роль стандартного звукового канала сводится к подачам гудков при загрузке, идентификации ошибок во время теста POST, когда сообщения на экран вывести еще нельзя, а также к сопровождению сообщений о системных ошибках.

Батарейная память и часы – CMOS Memory, RTC

Батарейная память и часы – CMOS Memory, RTC

- В АТ для хранения информации о конфигурации компьютера имеется специальная микросхема памяти КМОП небольшого объема, питание которой при выключенном компьютере осуществляется от батарейки.
- В той же микросхеме располагаются часы-календарь, тоже питающиеся от этой батарейки.
- Эти память и часы - *CMOS Memory and Real Time Clock* (CMOS RTC) - стали стандартным элементом архитектуры РС.

Батарейная память и часы – CMOS Memory, RTC

- Сначала содержимое этой памяти и дату модифицировали с помощью внешней загружаемой утилиты Setup, а позже эту утилиту встроили в BIOS.
- Микросхемы CMOS RTC имеют встроенную систему контроля непрерывности питания, отслеживающую также факт разрядки батареи ниже допустимого уровня.
- Достоверность информации конфигурирования проверяется контрольной суммой.

Батарейная память и часы – CMOS Memory, RTC

Таймер синхронизируется от собственного генератора с кварцевым резонатором, как правило, на 32,768 кГц.

Таймерная часть содержит:

- часы-календарь (год, месяц, число, час, минута, секунда);
- будильник, подающий сигнал в назначенные час, минуту и секунду;
- генератор меандра, позволяющий формировать запросы прерываний с заданной частотой (как правило, 1024 Гц).

Батарейная память и часы – CMOS Memory, RTC

- Микросхема CMOS RTC является источником аппаратного прерывания с номером 8.
- Прерывания могут возникать от будильника и генератора меандра, а также после смены времени в часах.
- Отдельные источники прерывания идентифицируются чтением ячейки 0Ch и разрешаются записью в ячейку 0Bh.
- Доступ к ячейкам CMOS RTC осуществляется через порты ввода-вывода 070h (индекс ячейки) и 071h (данные).

Батарейная память и часы – CMOS Memory, RTC

- Поскольку эта память имеет быстродействие порядка единиц микросекунд, необходима программная задержка между командами записи адреса и чтения-записи данных.
- Во время изменения состояния часов данные, считываемые из ячеек 0-9, могут оказаться некорректными.
- Признаком этой ситуации является единичное значение бита 7 ячейки 0Ah.

Батарейная память и часы – CMOS Memory, RTC

- Для определения момента окончания смены состояния часов можно пользоваться и разрешением соответствующего источника прерывания.
- Для работы с часами и будильником имеется сервис Int 1Ah BIOS (см. далее), который избавляет от необходимости программирования операций с учетом требуемых задержек, но может выдавать результаты и не с первого раза, если вызов попадет на момент изменения показаний часов.
- В этом случае вызов придется повторять.

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- *Сервисы Int 1Ah BIOS* позволяют считывать и модифицировать значения системного *таймера*, *даты* и *времени*, а также служат для установки *будильника* часов реального времени CMOS RTC.
- Перечисленные ниже номера функций указываются при вызове в регистре AH:

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AH = 0 — чтение системного таймера (двойного слова по адресу 40:006Eh в области переменных BIOS), инкрементируемого по прерываниям от канала 0 счетчика-таймера 8253/8254 примерно раз в 55 мс.
- Таймер обнуляется при выполнении теста POST после аппаратного сброса.
- Функция возвращает значение таймера, в CX — старшую часть, в DX — младшую. AL = 0, если за последние 24 часа не было переполнения таймера.
- В современных версиях регистр AL возвращает счетчик переполнений таймера, хранящийся в ячейке 40:0070h (в старых версиях содержимое AL было флагом, а не числом).

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AH = 1 — установка системного таймера (CX — старшая часть, DX — младшая) и сброс флага (счетчика) переполнения таймера в ячейке 40:0070h. В случае ошибки устанавливается флаг CF = 1.
- AH = 2 — чтение времени из RTC. Возвращает в упакованном BCD-формате час (в регистре CH), минуту (CL), секунду (DH) и признак коррекции летне-го/зимнего времени (DL = 1 — коррекция, DL = 0 — нет коррекции). Признаком успешной операции является флаг CF = 0. Во избежание ошибок некоторых версий BIOS при вызове флаг CF должен быть сброшен.

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AH = 3 — установка времени в RTC (назначение регистров и признак результата аналогичны функции 2).
- AH = 4 — чтение даты из RTC. Возвращает в упакованном BCD-формате век (в регистре CH), две старшие цифры года (CL), месяц (DH) и день (DL). Признаком успешной операции является флаг CF = 0. Во избежание ошибок некоторых версий BIOS при вызове флаг CF должен быть сброшен.

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AH = 5 — установка даты в RTC (назначение регистров и признак результата аналогичны функции 4).
- AH = 6 — установка времени срабатывания будильника RTC. Возвращает в упакованном BCD-формате час (в регистре CH), минуту (CL) и секунду (DH). Если будильник уже установлен, переустановка не производится и возвращается флаг CF = 1. При срабатывании будильник вызывает прерывание Int 4Ah.
- AH = 7 — отмена установки будильника.

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- *Функции Int 15h BIOS* позволяют программировать *таймер CMOS RTC* — вводить задержку или запускать таймер установки флага, указывая время в микросекундах (CX — старшее слово, DX — младшее).
- Нулевое значение интервала не вызывает никаких действий.
- Достижимое разрешение в зависимости от производительности ПК может достигать единиц миллисекунд, максимальная выдержка — около 70 часов.
- Перечисленные ниже номера функций указываются при вызове в регистре AH или AX:

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AH = 86h — задержка на заданное время. Управление возвращается вызвавшему процессу только через указанный интервал. По окончании задержки устанавливается бит 7 в ячейке BDA 0040:00A0. Таймер может оказаться занятым, тогда вызов сразу возвратит флаг CF = 1 (при успехе возвратится флаг CF = 0, а в AL окажется маска, записанная в 8259A#2).

Системная поддержка таймеров - Int 1Ah Int 15h BIOS

- AX = 8300h — запуск таймера, устанавливающего после указанной задержки бит 7 в ячейке, заданной регистрами ES:BX. При успешном запуске CF = 0; если таймер занят (он один) - CF = 1 и AL = 0. Управление возвращается процессу сразу, а флаг будет установлен через заданное время. Перед завершением программа, запускавшая таймер, должна его сбросить функцией 8301h (во-первых, чтобы освободить, во-вторых, чтобы снять «адскую машинку», которая неожиданно сама может изменить значение ячейки памяти, вполне возможно, уже задействованной другим, «ничего не подозревающим» процессом).
- AX = 8301h — сброс того же таймера.

Распределение системных ресурсов

Распределение системных ресурсов

- Для того чтобы программы могли взаимодействовать со *своими* устройствами, не мешая другим (и не получая от них помех), все системные ресурсы - адреса памяти и ввода-вывода, запросы прерываний и каналы DMA - должны быть бесконфликтно распределены между устройствами, подключенными к системной шине.

Распределение системных ресурсов

- *Для адресов памяти и портов ввода-вывода* бесконфликтность означает, что диапазоны соответствующих адресов всех имеющихся устройств не должны перекрываться.
- Это в первую очередь касается адресов, по которым выполняется чтение.
- Если устройства, конфликтующие по чтению, находятся на одной физической шине, то результат чтения из-за электрического конфликта оказывается неопределенным.

Распределение системных ресурсов

- Если конфликтующие устройства находятся на разных шинах, то будет прочитаны данные только одного устройства, но какого именно - зависит от настройки мостов, соединяющих шины.
- Конфликт по адресам для записи часто сознательно используют для одновременной передачи информации в несколько устройств.

Распределение системных ресурсов

- Информация при этом не искажается.
- Однако незапланированные конфликты по записи могут приводить к неожиданным побочным эффектам в работе устройств, не ожидающих данной записи.
- Для самого главного ведущего устройства — центрального процессора - безразлично, к какой из шин подключено устройство: он задает только адрес и тип операции.

Распределение системных ресурсов

- Операции записи могут выполняться широковещательно — распространяться по всем шинам.
- Операции чтения маршрутизируются - фактически, чтение по конкретному адресу памяти или порта обычно выполняется только с одной из шин.
- В иерархии шин PCI мосты выполняют маршрутизацию для всех транзакций.

Распределение системных ресурсов

- Для линий запросов прерываний бесконфликтность трактуется несколько сложнее.
- В классической *системе ISA* одну линию запроса может использовать только одно устройство, все остальные варианты — конфликтные.
- В *системах PnP ISA* (имеющих PnP BIOS, более гибко программируемый контроллер прерываний и устройства ISA PnP) при корректных настройках устройства PnP аппаратно могут использовать разделяемые (общие для нескольких устройств) линии прерываний.

Список литературы:

1. Аппаратные средства IBMPC. Гук М.Ю. Энциклопедия. 3-е изд. — СПб.: Питер, 2006.
2. Архитектура аппаратных средств. Конспект лекций. Барсукова Т. И.
3. Архитектура аппаратных средств. Конспект лекций. Забавина А. А.

СПИСОК ССЫЛОК:

<https://i2.wp.com/laptopmedia.com/wp-content/uploads/2017/06/900269711f3c.jpg>

<http://cart.softline.ru/pictures/products/16/35/05/99/af/f7/e1/63/ad/origin.jpeg>

[https://i.ebayimg.com/00/s/Njc1WDkwMA==/z/tkwAAOSweW5VAd64/\\$_57.JPG?set_id=880000500F](https://i.ebayimg.com/00/s/Njc1WDkwMA==/z/tkwAAOSweW5VAd64/$_57.JPG?set_id=880000500F)

<https://d.allegroimg.com/s1440/034db7/5bf73aa54f0ebb9f118bdae5d3ed>

<http://900igr.net/up/datas/55384/033.jpg>

<https://slide-share.ru/slide/4015074.jpeg>

<http://www.venuscomputers.pk/wp-content/uploads/2014/10/TG-3468.jpg>

<https://c-s.ru/uploads/29143/154716.jpg>

<https://go3.imgsmail.ru/imgpreview?key=65253deb8ce2d91f&mb=storage>

<https://i.ya-webdesign.com/images/pci-vector-slot.png>

[https://i.ebayimg.com/00/s/OTAwWDE2MDA=/z/ATkAAOSwAWlajflo/\\$_57.JPG?set_id=8800005007](https://i.ebayimg.com/00/s/OTAwWDE2MDA=/z/ATkAAOSwAWlajflo/$_57.JPG?set_id=8800005007)

Благодарю за внимание!

Преподаватель: Солодухин Андрей Геннадьевич

Электронная почта: asoloduhin@kait20.ru