

# Структурная и функциональная организация ЭВМ (Computer Organization and Design)

БГУИР  
кафедра ЭВМ

Лекция 24-25  
«СВВ семейства x86»

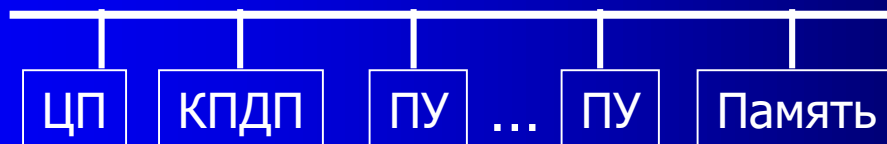
2019

## План лекции

1. Каналы ввода-вывода
2. Система ввода-вывода ЦП x86
3. Ввод/Вывод по прерываниям
4. Прямой доступ к памяти

# Прямой доступ к памяти

Механизм ПДП может быть реализован различными способами.



Вариант подключения КПДП в системе с общей системной шиной.

Минусы:

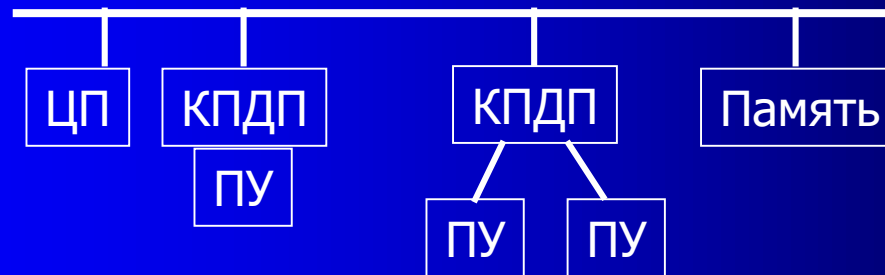
Эффективность – невысокая.

Каждая пересылка – два цикла шины.

Плюс: Простота, дешевизна.

# Прямой доступ к памяти

Механизм ПДП может быть реализован различными способами.



Варианты интеграции КПДП и ПУ.

# Прямой доступ к памяти

Механизм ПДП может быть реализован различными способами.



Система с шиной ввода/вывода.

Плюсы:

Лёгкость расширения.

Минимизация аппаратных затрат.

Обмен за 1 цикл системной шины.

# Каналы ввода-вывода

С развитием СВВ их функции усложняются – главная цель максимальное освобождение ЦП от управления процессами ввода/вывода.

## Дальнейшие шаги:

1. Расширение возможностей МВВ и предоставление ему прав процессора ВВ со спец. набором команд ввода/вывода. ЦП даёт указание о выполнении спец. программы ввода/вывода. Процессор ВВ извлекает из памяти и исполняет команды спец. программы самостоятельно и прерывает ЦП только после завершения всей программы ВВ.
1. К процессору ВВ добавляется лок. память. Управление множеством устройств ВВ. Минимизация привлечения ЦП к операциям ВВ.

# Каналы ввода-вывода

1. МВВ с расширенными возможностями называется *каналом ввода/вывода* (КВВ).
2. МВВ с расширенными возм. и лок. памятью называют *процессором ВВ*.

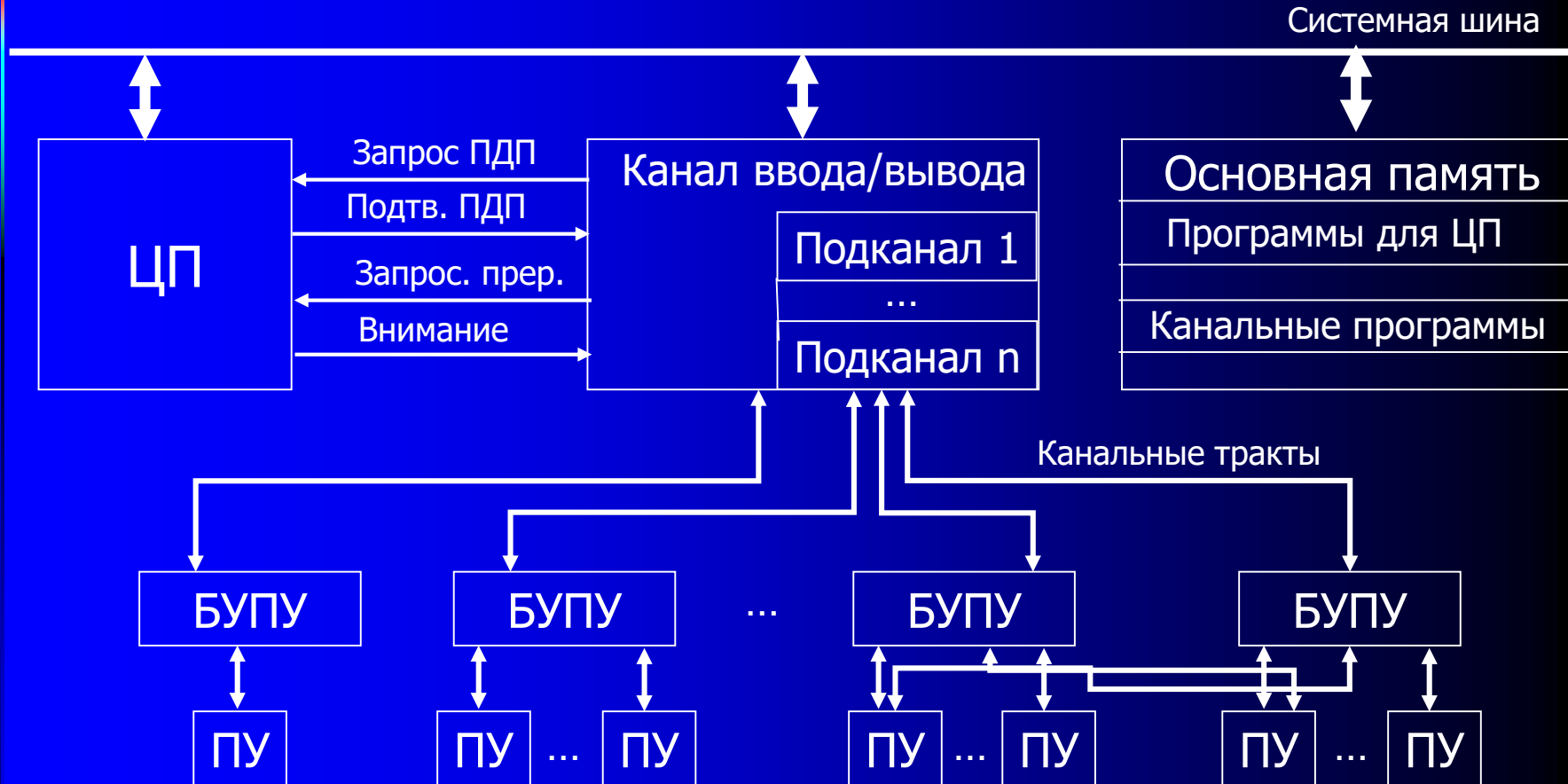
Различие достаточно условное – поэтому чаще используют термин – «канал».

СВВ с каналами ВВ была разработана и реализована в IBM 360 и совершенствована в сериях 370 и 390.

В подобной системе ЦП практически не участвует в непосредственном управлении ПУ - это задача спец. процессора ВВ. ЦП лишь запускает и останавливает операции КВВ и проверяет состояние канала и подключённых к каналу ПУ.

Плюсы: **минимизация команд ЦП для ввода/вывода.**

# Каналы ввода-вывода



ВМ с канальной системой ввода/вывода



# Каналы ввода-вывода

ПУ подключаются к КВВ через блоки управления периферийными устройствами. БУПУ принимает приказы по управлению ПУ (чтение, запись, перемотка и т.п.) и преобразует в сигналы свойственные конкретному ПУ.

Обычно один БУПУ может обслуживать несколько ПУ, но для быстродействующих ПУ – свой БУПУ.

Некоторые ПУ могут подключаться более чем к одному БУПУ – диверсификация подканалов.

Физически БУПУ может быть как самостоятельным устройством, так и частью ПУ или канала.

# Каналы ввода-вывода

В пределах канала каждое ПУ подключено к своему подканалу (с уникальными логическими номерам). Физически подканал – участок лок. памяти КВВ, где хранятся параметры операции ВВ.

Окончание операции либо ошибка – «Запрос на прерывание». ЦП может передавать КВВ спец. сигнал «Внимание».

КВВ реализует два режима работы – мультиплексный (разделение времени) и монопольный.

Передача м/у ВУ и КВВ байта или слова за сеанс – «байт мультиплексный», пакета данных – «блочно-мультиплексный».

# Система ввода-вывода ЦП x86

x86 – отдельная адресация памяти и портов(модулей) ввода-вывода.

Сигналы чтения и записи из/в порт на шине ISA – **IORD#** и **IOWR#** (чтение/запись из/в память – MEMRD# MEMWR#) – порождают шинные циклы обмена чтения из порта и записи в порт.

На шине PCI – разделение иначе – тип операции кодируется 4 байтной командой, в зависимости от типа инструкции, выполняемой ЦП.

Пространство ВВ – 64kB, -> адресация по 16 линиям. В первых PC использовались лишь 10 бит адреса шины ISA (адреса 0-3FFh), в адаптерах для шин MCA и PCI – используются все 16 бит адреса.

# Карта портов ввода-вывода ЦП x86 (10-битное адресное пространство)

**Таблица**

Стандартная карта портов ввода-вывода

АТ и PS/2	PC/XT	Назначение
000–00F	000–00F	Контроллер DMA #1 8237
010–01F		PS/2 — расширение DMA #1
020–021	020–021	Контроллер прерываний #1 — 8259A
040–05F	040–043	Таймер (PC/XT: 8253, АТ: 8254)
060	060	Диагностический регистр POST (только запись)
	060–063	Системный интерфейс 8255
060, 064		Контроллер клавиатуры АТ 8042
061		Источники NMI и управление звуком
070–07F		Память CMOS и маска NMI
080		Диагностический регистр
080–08F	080–083	Регистры страниц DMA
090–097		PS/2-микроканал, арбитр
	0A0	Маска NMI
0A0–0BF		Контроллер прерываний #2 — 8259A

# Карта портов ввода-вывода ЦП x86

АТ и PS/2	PC/XT	Назначение
0C0-0DF		Контроллер DMA #2 8237A-5
0F0-0FF		Сопроцессор 80287
100-1EF		PS/2-управление микроканалом
170-177		Контроллер НЖМД #2 (IDE#2)
1F0-1F7		Контроллер НЖМД #1 (IDE#1)
200-207	200-20F	Игровой адаптер
	210-217	Блок расширений
238-23F		COM4
278-27F	278-27F	Параллельный порт LPT2 (LPT3 при наличии MDA)
	2A2-2A3	Часы MSM48321RS
2C0-2DF	2C0-2DF	EGA #2
2E0-2E7		COM4
<del>2E8-2EF</del>		<del>COM4</del>
2F8-2FF	2F8-2FF	COM2
300-31F		Плата прототипа
	320-32F	Жесткий диск XT
338-33F		COM3
370-377		Контроллер НГМД #2
376-377		Порты команд IDE#2
378-37F	378-37F	Параллельный порт LPT1 (LPT2 при наличии MDA)
380-38F	380-38F	Синхронный адаптер SDLC/BSC #2
3A0-3AF	3A0-3A9	Синхронный адаптер BSC #1
3B0-3BB	3B0-3BB	Монохромный адаптер (MDA)
3B4-3C9		PS/2-видеосистема
3BC-3BF	3BC-3BF	Параллельный порт LPT1 платы MDA
3C0-3CF	3C0-3CF	EGA #1
3C0-3DF	3C0-3DF	VGA
3D0-3DF	3D0-3DF	CGA/EGA
3E0-3E7		COM3
3E8-3EF		COM3
3F0-3F7	3F0-3F7	Контроллер НГМД #1
3F6-3F7		Порты команд IDE#1
3F8-3FF	3F8-3FF	COM1

# Карта портов ввода-вывода ЦП x86

АТ и PS/2	PC/XT	Назначение
<del>0C0-0DF</del>		<del>Контроллер DMA #2 8237A-5</del>
0F0-0FF		Сопроцессор 80287
100-1EF		PS/2-управление микроканалом
170-177		Контроллер НЖМД #2 (IDE#2)
1F0-1F7		Контроллер НЖМД #1 (IDE#1)
200-207	200-20F	Игровой адаптер
	210-217	Блок расширений
238-23F		COM4
278-27F	278-27F	Параллельный порт LPT2 (LPT3 при наличии MDA)
	2A2-2A3	Часы MSM48321RS
2C0-2DF	2C0-2DF	EGA #2
2E0-2E7		COM4
2E8-2EF		COM4
2F8-2FF	2F8-2FF	COM2
300-31F		Плата прототипа
	320-32F	Жесткий диск XT
338-33F		COM3
370-377		Контроллер НГМД #2
376-377		Порты команд IDE#2
378-37F	378-37F	Параллельный порт LPT1 (LPT2 при наличии MDA)
380-38F	380-38F	Синхронный адаптер SDLC/BSC #2
3A0-3AF	3A0-3A9	Синхронный адаптер BSC #1
3B0-3BB	3B0-3BB	Монохромный адаптер (MDA)
3B4-3C9		PS/2-видеосистема
3BC-3BF	3BC-3BF	Параллельный порт LPT1 платы MDA
3C0-3CF	3C0-3CF	EGA #1
3C0-3DF	3C0-3DF	VGA
<del>3D0-3DF</del>	<del>3D0-3DF</del>	<del>CGA/EGA</del>
3E0-3E7		COM3
3E8-3EF		COM3
3F0-3F7	3F0-3F7	Контроллер НГМД #1
3F6-3F7		Порты команд IDE#1
<del>3F8-3FF</del>	<del>3F8-3FF</del>	<del>COM1</del>

# Система ввода-вывода ЦП x86

Выбор базовых адресов стандартных устройств (3F8h, 3F0h, 378h и т.п.) – из экономии, с использованием максимального числа единичных битов и минимальное нулей – т.к. чаще применяли схемы «И-НЕ».

Каждой шине – своя область адресов ввода, при чтении дешифратор адресов системной платы открывает соответствующие буферы данных -> таким образом считывание только с нужной шины.

При записи – данные и сигнал записи может распространяться широковещательно – по всем шинам. В стандартном распределении адреса 0-0FFh закреплены за устройствами системной платы. При наличие ПУ, чтение по этим адресам не распространяется на шины расширения. В современных ВС с несколькими шинами, распределением адресов управляет BIOS через регистры конфигурирования чипсета, для шин PCI (PCI-E) – программированием мостов.

# Типы прерываний

Прерывания подразделяются на:

1. **Программные прерывания (Software Interrupts)** применяются для -
  - а) реализации системных вызовов (System Calls) необходимых для сигнализации ОС об инициации передачи данных.
  - б) уведомления дебаггера о наступлении определённых моментов – например отслеживания ошибок.
  - с) других целей...
2. **Периферийные или аппаратные прерывания – Маскируемые/Не маскируемые ( **Maskable / Non-Maskable** )**: - обеспечение реакции процессора на события, происходящие асинхронно к исполняемому программному коду **возобновления частично завершённой передачи/ для управления передачей данных от соответствующего устройства.**



# Типы прерываний

Немаскируемое NMI (Non-Maskable Interrupt): - сигнализация о фатальных ошибках. На NMI процессор реагирует всегда (при условии завершения обслуживания предыдущего NMI). В x86 – NMI – фиксированный вектор 2. На отдельный вход ЦП NMI поступает сигнал от схем контроля памяти от шины ISA (**IOCHK**) и шины PCI (**SERR#**).

До поступления сигнала на вход ЦП его можно блокировать установкой в «1» **бита 7 порта 070h**. Кроме того можно разрешать и определять источники NMI:

- ♦ **бит 2 порта 061h** (R/W) – разрешение контроля ОЗУ и сигнала SERR# шины PCI.
  - ♦ **бит 3 порта 061h** (R/W) – разрешение контроля шины ISA
  - ♦ **бит 6 порта 061h** (R) – ошибка контроля на шине ISA
  - ♦ **бит 7 порта 061h** (R) – ошибка чётности ОЗУ или SERR#
- шины PCI

# Типы прерываний

Немаскируемое SMI (System Management Interrupt): - перевод процессора в режим системного управления (System Management Mode) – функционирование ЦП в полной изоляции от прикладного и системного ПО (с запретом всех аппаратных прерываний и исключений). Чаще всего используется для управления энергопотреблением, но возможны и другие цели.

Сигнал SMI#- для ЦП немаскируемое прерывание с наивысшим приоритетом. При входе в SMM ЦП сохраняет своё состояние в SMRAM (**не в стеке!!!**) – доступ к этой памяти внешними устройствами только при нахождении ЦП в SMM. В ней же хранится обработчик прерывания SMI.

При выходе из SSM возможен рестарт – повторное выполнение инструкции, предшествующей появлению SMI#.

Режим SSM – полностью «прозрачен» для ПО и ОС -> можно управлять энергопотреблением на уровне BIOS...

# Маскируемые прерывания

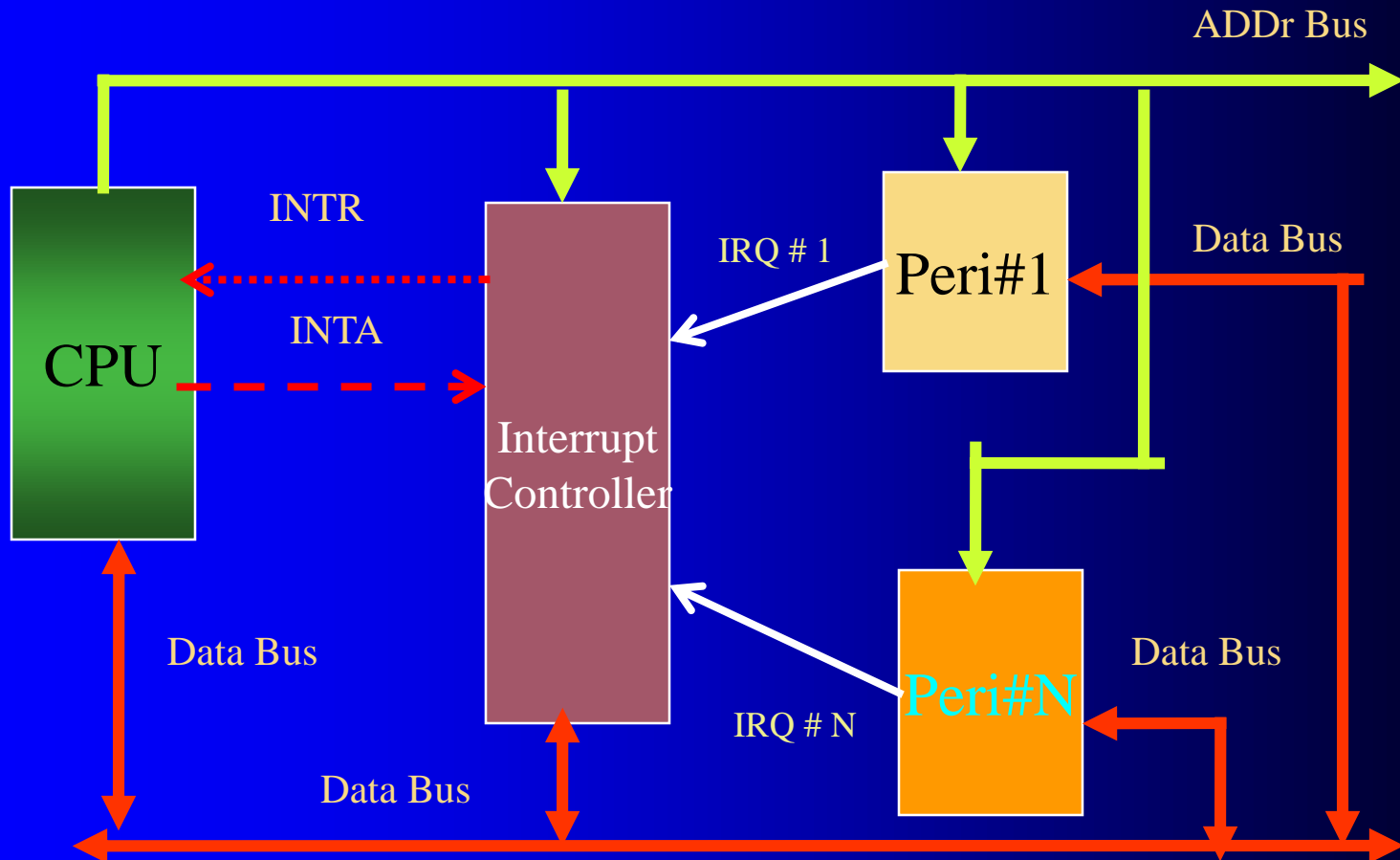
Маскируемые прерывания используются для сигнализации о событиях в устр-вах. Реакция ЦП может быть задержана сбросом его внутреннего флага IF (инструкции CLI/STI – запретить/разрешить).

При возникновении события, контроллер ВВ (ПУ) формирует запрос прерывания и подаёт его на вход контроллера прерываний (КП). Задача КП – довести до ЦП запрос прерывания и сообщить вектор, идентифицирующий процедуру обработки данного прерывания.

В IBM PC-совместимых ПК применяются два основных типа контроллера:

- Периферийный контроллер прерываний (Peripheral Interrupt Controller, PIC) – программно совместим с традиционным контроллером 8259А (с первых IBM PC/XT). С PC/AT -> связка из пары каскадов PIC -> до 15 линий запросов прерываний.

# Interrupting Peripherals ( *Device / Hardware Interrupts* )



Типовая схема взаимодействия ЦП – ПУ

# Контроллер прерываний 8259А

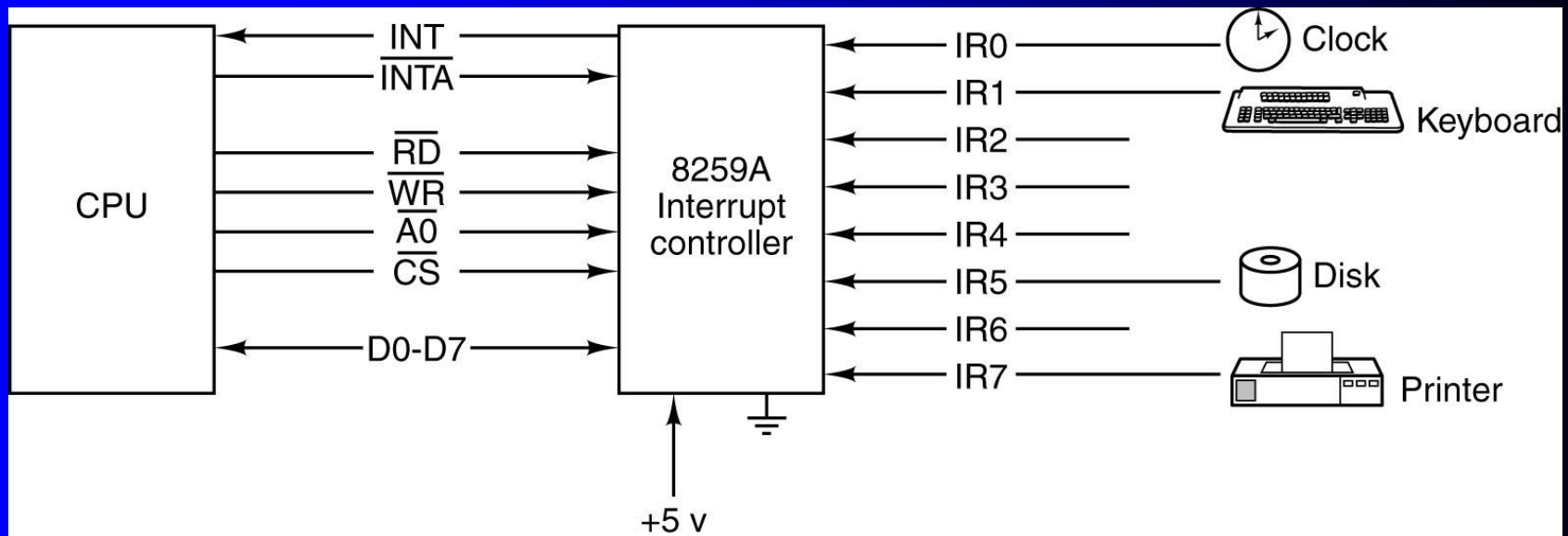
Контроллер прерываний (PIC) 8259А является периферийным устройством, связанным с ЦП через ту или иную шину расширения ВВ. По этой шине ЦП может обращаться к регистрам PIC, программируя их и управляя его работой, а так же получать от PIC 8-битный вектор прерывания (с пом. доп сигналов INT/INTA).

Контроллер имеет 8 входов запросов от источников прерываний и 1 выход общего запроса. Программированием контроллера задаётся адрес вектора прерывания для входа 0, остальные -> последующие номера...

Каждый вход может быть программно замаскирован.

За регистрами контроллера – два адреса в простр-ве ВВ: 020-021h. С каждым входом запроса в PIC связано по одному биту в регистрах IRR, IMR и ISR (0-й бит -> 0-му входу, 7-й -> 7-му).

# Контроллер прерываний 8259А



## Применение контроллера 8259А

# Управление контроллером 8259А

За регистрами контроллера – два адреса в пространстве ВВ: **020-021h**. С каждым входом запроса в PIC связано по одному биту в регистрах **IRR**, **IMR** и **ISR**:

- **IRR** (Interrupt Request Register) – регистр запросов прерываний, бит будет =1, если пришёл запрос на соотв. вход. От маски не зависит.
- **IMR** (Interrupt Mask Register) – регистр масок прер-ний. Единичное значение бита означает игнорирование запроса на прерывание, полученное с соотв. входа контроллера.
- **ISR** (Interrupt Service Register) – регистр обслуживаемого прерывания.

Вопрос – адреса 2, а регистров 3 – в чём фокус?

Контроллер позволяет программировать свои входы на чувствительность к уровню или перепаду сигнала.

В PC/AT – каскадное соединение двух контроллеров.

# Коммутация запросов прерываний ЦП x86

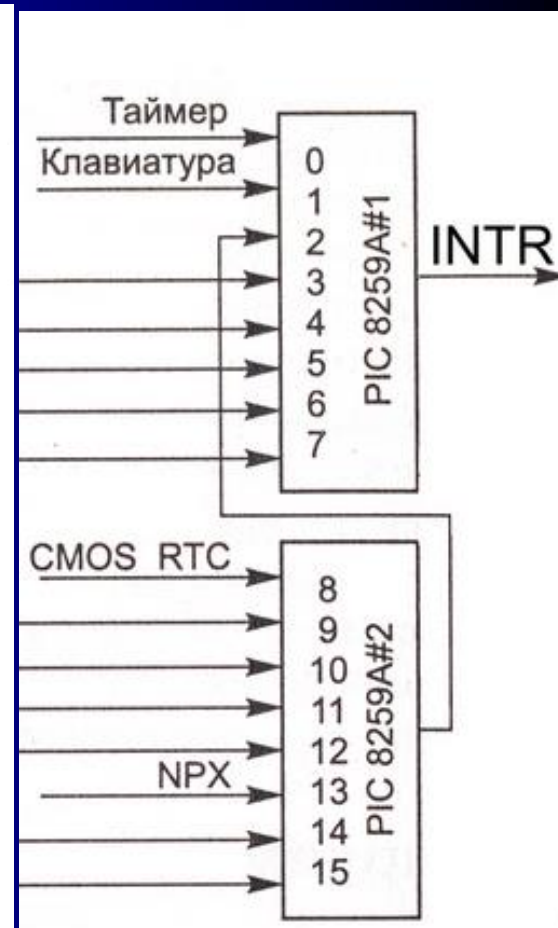
Таблица 4.2. Аппаратные прерывания (в порядке убывания приоритета)

Имя (номер <sup>1</sup> )	Вектор <sup>2</sup>	Вектор <sup>3</sup>	Контроллер/ маска	Описание
NMI	02h	02h	—	Контроль канала, четность памяти (в XT — сопроцессор)
IRQ0	08h	50h	#1/1h	Таймер (канал 0 8253/8254)
IRQ1	09h	51h	#1/2h	Клавиатура
IRQ2	0Ah	52h	#1/4h	XT — резерв, AT — недоступно (подключается каскад IRQ8–IRQ15)
IRQ8	70h	58h	#2/1h	CMOS RTC — часы реального времени
IRQ9	71h	59h	#2/2h	Резерв
IRQ10	72h	5Ah	#2/4h	Резерв
IRQ11	73h	5Bh	#2/8h	Резерв
IRQ12	74h	5Ch	#2/10h	PS/2-Mouse (резерв)
IRQ13	75h	5Dh	#2/20h	Математический сопроцессор
IRQ14	76h	5Eh	#2/40h	HDC — контроллер НЖМД
IRQ15	77h	5Fh	#2/80h	Резерв
IRQ3	0Bh	52h	#1/4h	COM2, COM4
IRQ4	0Ch	53h	#1/10h	COM1, COM3
IRQ5	0Dh	54h	#1/20h	XT — HDC, AT — LPT2, Sound (резерв)
IRQ6	0Eh	55h	#1/40h	FDC — контроллер НГМД
IRQ7	0Fh	56h	#1/80h	LPT1 — принтер

<sup>1</sup> Запросы прерываний 0, 1, 8 и 13 на шины расширения не выводятся.

<sup>2</sup> Указаны номера векторов при работе в реальном режиме процессора.

<sup>3</sup> Указаны номера векторов при работе в ОС Windows.





# Карта портов ввода-вывода ЦП x86 (10-битное адресное пространство)

**Таблица**      Стандартная карта портов ввода-вывода

<b>АТ и PS/2</b>	<b>PC/XT</b>	<b>Назначение</b>
000–00F	000–00F	Контроллер DMA #1 8237
010–01F		PS/2 — расширение DMA #1
020–021	020–021	Контроллер прерываний #1 — 8259A
040–05F	040–043	Таймер (PC/XT: 8253, АТ: 8254)
060	060	Диагностический регистр POST (только запись)
	060–063	Системный интерфейс 8255
060, 064		Контроллер клавиатуры АТ 8042
061		Источники NMI и управление звуком
070–07F		Память CMOS и маска NMI
080		Диагностический регистр
080–08F	080–083	Регистры страниц DMA
090–097		PS/2-микроканал, арбитр
	0A0	Маска NMI
0A0–0BF		Контроллер прерываний #2 — 8259A

# Нехватка линий запросов прерываний

В PC/XT – хватало одного PIC 8259A.

В PC/AT – каскадное соединение уже двух контроллеров.

Что делать дальше, если количество периферии растёт пропорционально производительности + надо поддерживать совместимость «сверху вниз»?

На каждом из слотов шин ISA/EISA присутствуют все незанятые входы контроллеров **IRQx**, часть отдаётся шине PCI. Но проблема остаётся – входов контроллеров всё равно не хватает. **Выход?**

Выход - «вешать» на один вход несколько источников прерываний. Но проблема – при запросе по положительному перепаду сигнала **IRQx** несколько устройств на одну линию «повесить» невозможно (**кстати, почему?**).

# Нехватка линий запросов прерываний

Доп. сложности – как плате вставленной в слот шины ISA/EISA/PCI определить какую линию **IRQx** использовать?

Ответ – назначения номеров прерывания выполняются с двух сторон: а) адаптер ПУ, нуждающегося в механизме прерываний, должен быть сконфигурирован (джамперами или программно –P&P) на использование конкретной линии **IRQx**, б) ПО, поддерживающее данный адаптер, должно знать номер применяемого вектора.

Изначально шины ISA не поддерживали P&P и использовали запрос по положительному перепаду сигнала **IRQx**.

Устройства ISA/EISA, поддерживающие P&P, позволя(-ли/-ют) настраиваться на прерывание по низкому уровню.

Устройства PCI работают только по низкому уровню.

# Нехватка линий запросов прерываний

Устройства PCI работают только по низкому уровню, по той причине, что линии IRQx, отведённые PCI недоступны ISA/EISA.

Разделение линий между шинами, а так же управление чувствительностью (к перепаду или уровню) отдельных линий обеспечивают параметры утилиты энергонезависимой области памяти - CMOS Setup: **Complementary metal–oxide–semiconductor**

- значения «ISA» или «Legacy» - статическое распределение
- значения «PCI/PnP» - динамическое распределение.

Чувствительность к уровню – КП, обнаружив низкий уровень на входе IRQx вырабатывает запрос на прерывание, и если к моменту завершения обработки ЦП этого прерывания КП обнаруживает низкий уровень на том же входе IRQx, то КП опять вырабатывает запрос на прерывание...

Чувствительность к перепаду – если КП, обнаружил фронт на входе IRQx, то до момента завершения обработки прерывания сигнал будет находиться в изменённом состоянии и фронт от другого ПУ (- на этом же IRQx) без возврата сигнала в исходное состояние зафиксирован не будет...

# Коммутация запросов прерываний ЦП x86

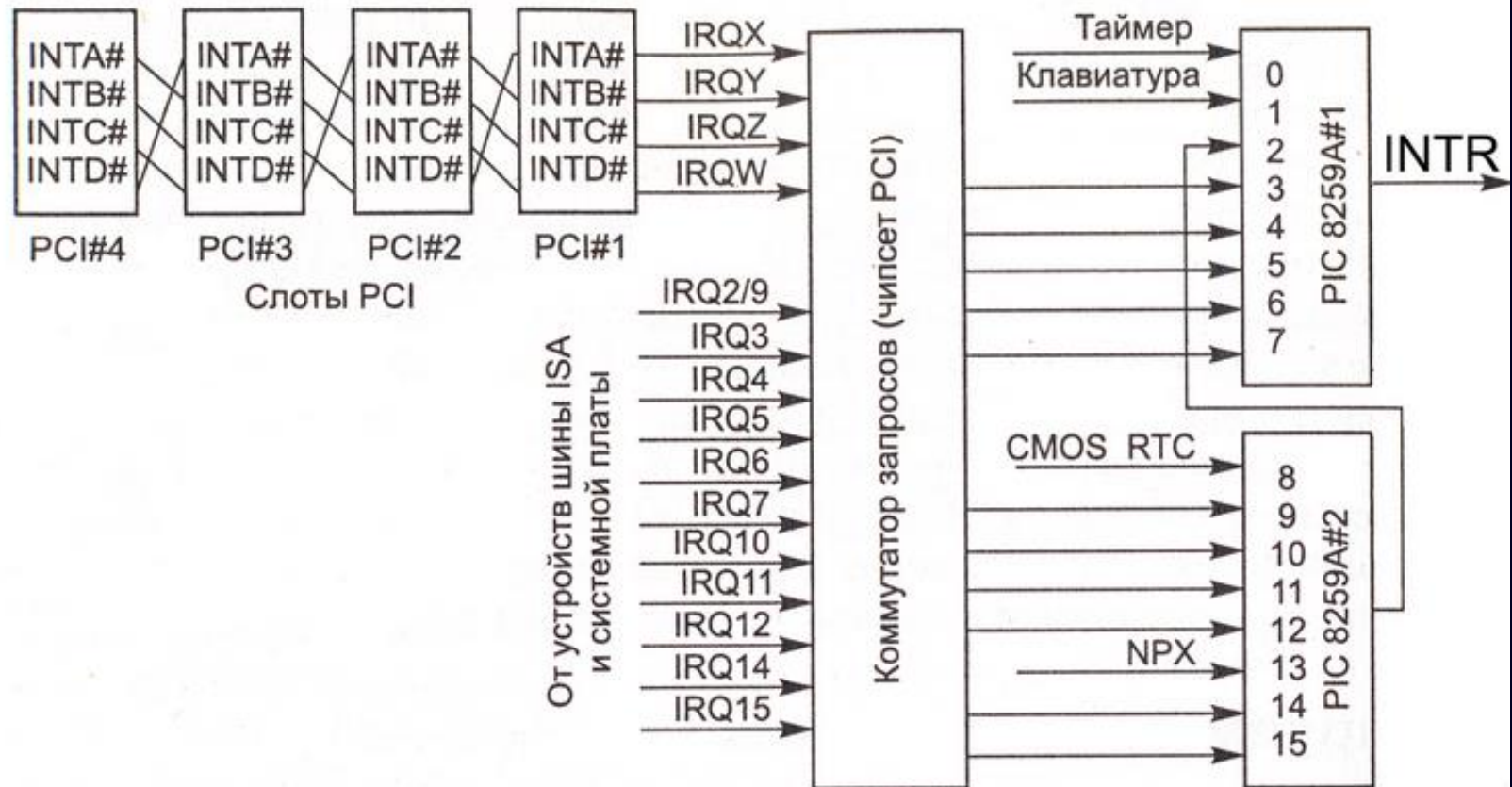


Рис Коммутация запросов прерываний

# Коммутация запросов прерываний ЦП x86

Для запросов прерываний от устройств с линии PCI задействуют 4 линии запросов прерывания – **INTR A,B,C,D**. Линии работают по низкому уровню, что даёт возможность их совместного использования.

Линии циклически сдвигаются в слотах и независимо коммутируются на доступные линии **IRQx** с помощью конфигурационных регистров чипсета.

Номера векторов, соответствующих линиям запросов контроллеров, система приоритетов и др. параметры задаются программно при инициализации контроллеров PIC или APIC.

Основные настройки остаются традиционными для совместимости со старым ПО, но различаются для ОС реального и защищённого режимов.



# Коммутация запросов прерываний ЦП x86

Таблица 4.2. Аппаратные прерывания (в порядке убывания приоритета)

Имя (номер <sup>1</sup> )	Вектор <sup>2</sup>	Вектор <sup>3</sup>	Контроллер/ маска	Описание
NMI	02h	02h	—	Контроль канала, четность памяти (в XT — сопроцессор)
IRQ0	08h	50h	#1/1h	Таймер (канал 0 8253/8254)
IRQ1	09h	51h	#1/2h	Клавиатура
IRQ2	0Ah	52h	#1/4h	XT — резерв, AT — недоступно (подключается каскад IRQ8–IRQ15)
IRQ8	70h	58h	#2/1h	CMOS RTC — часы реального времени
IRQ9	71h	59h	#2/2h	Резерв
IRQ10	72h	5Ah	#2/4h	Резерв
IRQ11	73h	5Bh	#2/8h	Резерв
IRQ12	74h	5Ch	#2/10h	PS/2-Mouse (резерв)
IRQ13	75h	5Dh	#2/20h	Математический сопроцессор
IRQ14	76h	5Eh	#2/40h	HDC — контроллер НЖМД
IRQ15	77h	5Fh	#2/80h	Резерв
IRQ3	0Bh	52h	#1/4h	COM2, COM4
IRQ4	0Ch	53h	#1/10h	COM1, COM3
IRQ5	0Dh	54h	#1/20h	XT — HDC, AT — LPT2, Sound (резерв)
IRQ6	0Eh	55h	#1/40h	FDC — контроллер НГМД
IRQ7	0Fh	56h	#1/80h	LPT1 — принтер

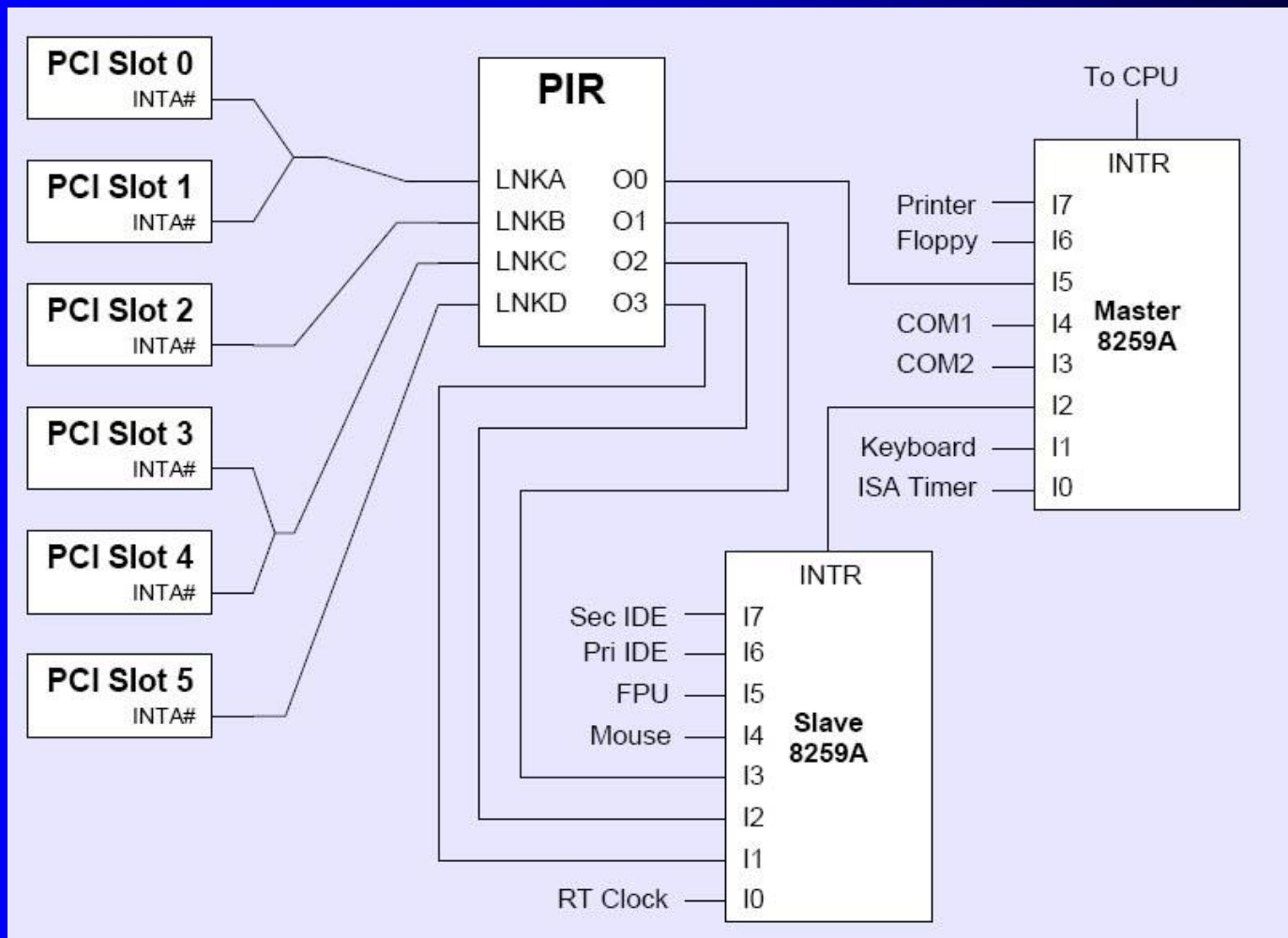
<sup>1</sup> Запросы прерываний 0, 1, 8 и 13 на шины расширения не выводятся.

<sup>2</sup> Указаны номера векторов при работе в реальном режиме процессора.

<sup>3</sup> Указаны номера векторов при работе в ОС Windows.

Различия в базовых адресах векторов в реальном и защищённом режимах.

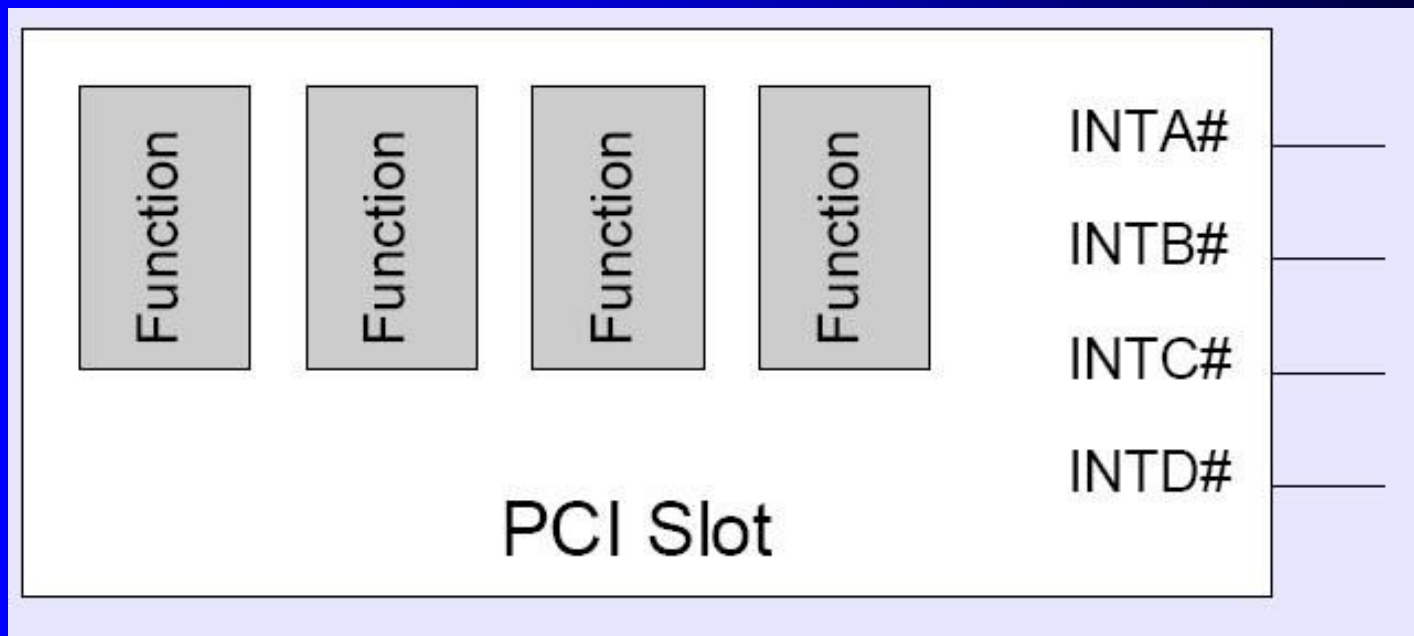
# Коммутация запросов прерываний ЦП x86



Альтернативная конфигурация коммутации PCI линий запросов прерывания.



# Коммутация запросов прерываний ЦП x86



Пример организации адаптера ПУ, подключаемого к PCI.

Т.к. PCI слот имеет 4 разделяемые линии запросов на прерывания, то ПУ **данного слота, может производить запросы на прерывания по 4 различным событиям/функциям.**

# Обработка маскируемых прерываний:

1. ПУ по событию возбуждает назначенную ему линию запроса прерывания – IRQx или INT A/B/C/D ->PIR-> IRQx.

1. Контроллер принимает сигналы запросов от источников прерываний и при наличии незамаскированного запроса подаёт сигнал INTR на вход ЦП x86.

1. ЦП, реагируя на запрос (если флаг IF не сброшен) сохраняет в стеке – регистр флагов (**почему только флаги?**) и адрес возврата, после чего формирует шинный цикл INTA (Interrupt Acknowledge – подтверждение прерывания) -> КП.

1. По получению INTA КП фиксирует состояние своих входов (оно могло измениться) и анализирует их в соотв. с уровнями приоритетов и посылает ЦП вектор прерывания.

# Обработка маскируемых прерываний:

5. Получив вектор ЦП по его номеру вызывает соотв. процедуру обработки прерывания (ISR – Interrupt Service Routine). ISR определяет аппаратное или программное прерывание (если программное допускается для данного вектора).

5. ISR идентифицирует источник прерывания – ПУ. При раздельном использовании IRQx – идентификации только путём последовательных обращений к регистрам всех ПУ и считывания их состояний. При этом следует учитывать возможность одновременных прерываний от нескольких ПУ или же появления ещё одного/двух... уже в процессе обработки.

# Обработка маскируемых прерываний:

7. ISR обслуживает ПУ-источник прерывания – ISR должна добиться от ПУ снятия сигнала прерывания. Если обслуживание долгое, то после выполнения критической секции, флагом IF ISR разрешает ЦП выполнять более приоритетные запросы -> возможны вложенные прерывания.

7. ISR в конце посылает команду EOI (End of Interrupt) – КП разрешает последующий приём с обслуженного входа и менее приоритетных IRQх. Команда посылается после снятия запросов обслуженными устройствами иначе КП после EOI пошлёт повторный запрос INTR. Обработчик прерывания ведомого контроллера должен послать EOI как ведомому, так и ведущему КП.

# Обработка маскируемых прерываний:

Фрагмент ISR от инструкции подачи команды EOІ до завершения (инструкции IRET) должен быть непрерываемым, т.е. являться критической секцией. Если обработчик разрешал вложенные прерывания, то он их должен запретить (инструкция CLI) перед подачей EOІ.

Вопрос – до самого выхода из ISR (т.е. включительно до IRET) флаг IF не может быть восстановлен – иначе есть риск, что данная ISR никогда не закончится. **Каким образом восстанавливается флаг IF?**

9. Обработка прерывания завершается инструкцией IRET, по которой ЦП возвращается к выполнению прерванного потока инструкций, предварительно восстановив регистр флагов (с **IF == 1 !!!**) и аппаратные прерывания опять разрешены. *Слайд 37*

## Зачем ещё может понадобиться маскирование прерываний?

При инициализации ВС (процедурой POST и при загрузке ОС) все неиспользуемые входы контроллеров замаскированы, а их векторы указывают на заглушку – ISR с единственной инструкцией IRET. Для подключения на такой вход ПУ, сперва следует загрузить его ISR в память и установить на него указатель в таблице прерываний, затем произвести демаскирование соотв. входа КПП. Если обработчик удаляется из памяти, то предварительно должен быть замаскирован соотв. вход IRQx.

Все изменения в таблице прерываний должны выполняться при сброшенном флаге IF=0(!) иначе возможно использование вектора в процессе его модификации и «вылет» по некорректному адресу.

# Прим.

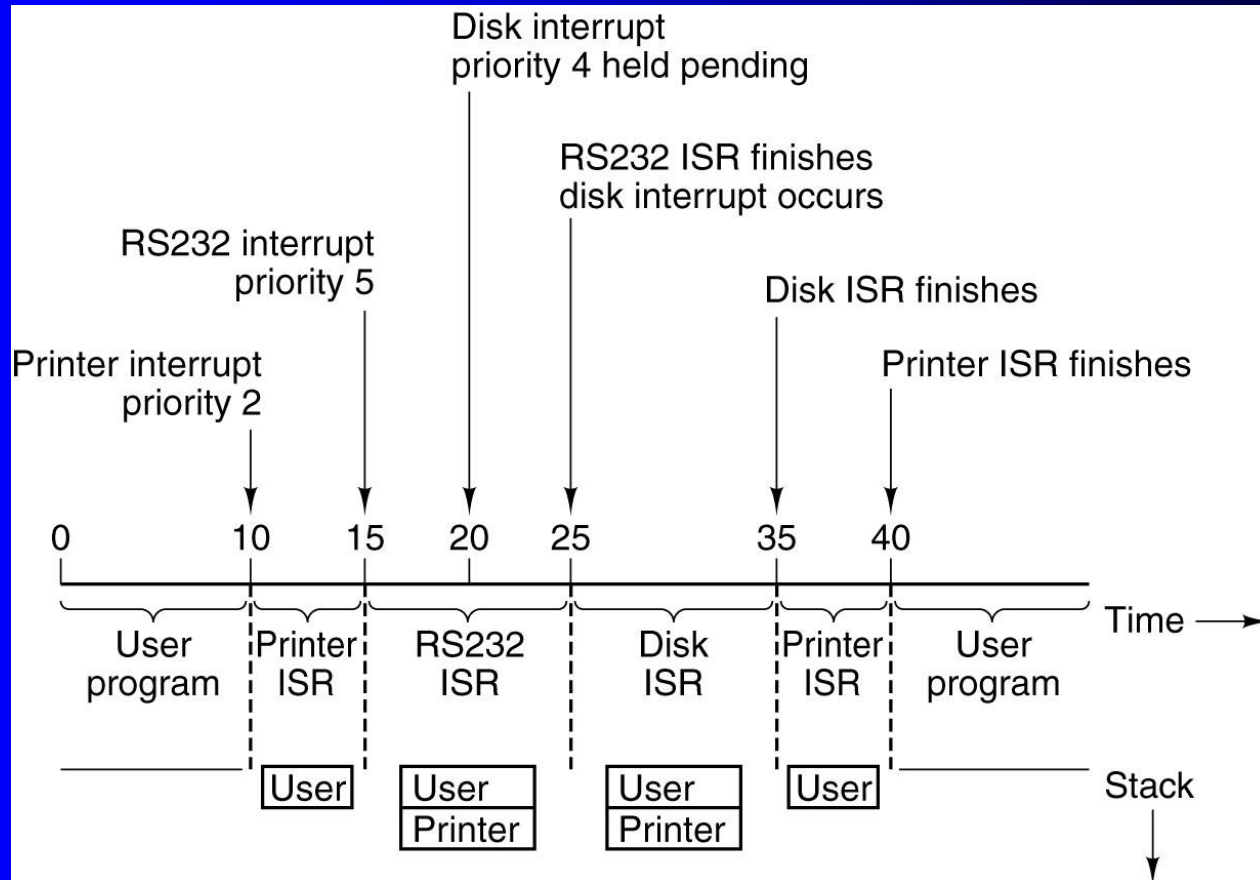
В IBM PC/XT/AT используется спец. режим вложенных прерываний с фиксированным приоритетом и автоматическим неспецифическим (т.е. некритичным завершением).

В современных системных платах функции КП – за чипсетом, который может иметь более гибкие настройки, чем пара КП 8259А. Процедура их инициализации может отличаться от стандартной, но ею занимается POST от производителя платы → проблем не возникает. В операционном режиме чипсет обязан поддерживать программную совместимость с 8259А.

Каждая ISR обязана завершаться командой EOI – т.е. посылкой:

- 1-му контроллеру байта 20h на адрес 020h.
- 2-му – байта 20h на адрес 0A0h.

# Вложенные прерывания



Пример временной последовательности вложенных прерываний. ISR – (Interrupt Service Routines) – процедура обработки прерывания.



# Типы контроллеров прерываний

В IBM PC-совместимых ПК применяются два основных типа контроллера:

- Усовершенствованный периферийный контроллер прерываний (**Advanced PIC - APIC**) – введён для поддержки мультипроцессорных систем со времён x86 4-5 поколений (486-Pentium).

Помимо «мультипроцессорности», APIC позволяет увеличивать число доступных линий и обрабатывать запросы прерываний от устройств на шине PCI, посылаемые через механизм сообщений (**MSI**). Системы с APIC в обязательном порядке совместимы с PIC – включение по аппаратному сбросу (и вкл. питания) -> что позволяет запускать старые ОС и приложения MS DOS.

# Advanced Peripheral Interrupt Controller

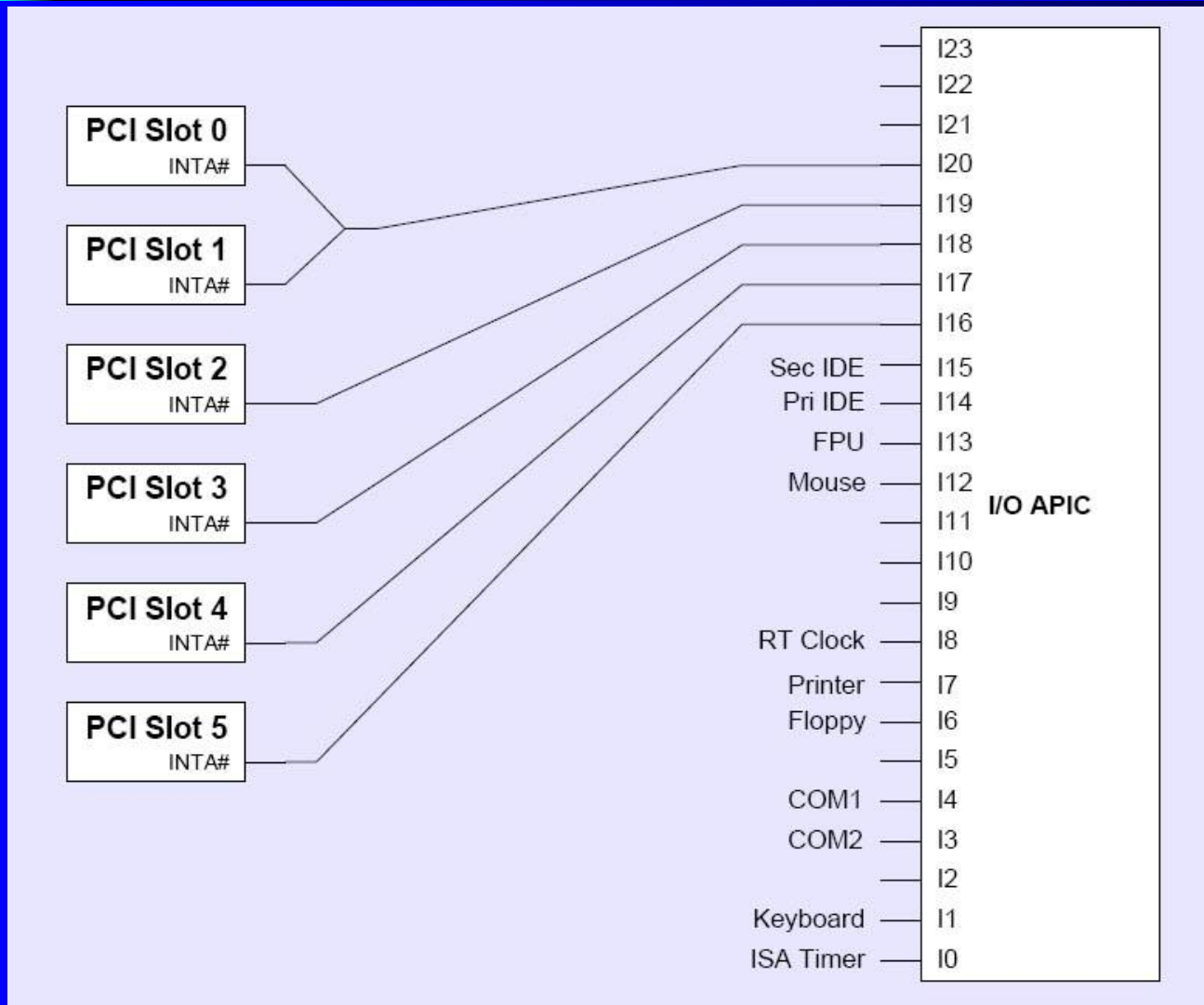
Система с **APIC** более сложная -> разделение функций КП на несколько частей: локальных контроллеров (интегрированы в процессоры) и КП от ввода-вывода (одного или нескольких) – **I/O APIC**.

Все контроллеры APIC соединены м/у собой локальной шиной, по которой они обмениваются друг с другом сообщениями.

Задача каждого **локального APIC** – трансляция сообщений, принятых по лок. шине в сигналы, вызывающие аппаратные прерывания своего процессора – маскируемые (INTR), немаскируемые (NMI) и системного управления (SMI) + обратная задача – генерация прерываний от своего ЦП к другим посредством сообщений по лок. шине.

Лок.APIC имеет внутренний таймер -> возможность генерации прерываний с заданным интервалом.

# I/O Advanced Peripheral Interrupt Controller



# Advanced Peripheral Interrupt Controller

**I/O APIC** преобразует запросы аппаратных прерываний от ПУ в сообщ. локальной шины APIC.

В мультипроц. режиме он отвечает за распределение апп. прер. по процессорам:

- статическое (опр. ЦП обслуживает только опр. запросы на прерывания).
- динамическое (каждое прерывание отправляется на обл. наименее занятому/приоритетному в тек. момент времени ЦП).

I/O APIC отвечает так же за распространение сигналов о системных событиях (**INTR, NMI, SMI**) и межпроцессорных прерываний -> является частью чипсета, напр. входит в хабы ICH2 и ICH3 от Intel.

# Advanced Peripheral Interrupt Controller

## Три режима работы APIC:

- режим PIC – эмуляция пары PIC 8259A с традиционной передачей сигналов прерывания одному процессору;
- режим виртуальных проводов (virtual wire mode) – аналогичен PIC, но с параллельной подачей сигналов по лок. шине APIC, при этом APIC может работать совместно с PIC 8259A, обеспечивая доп. возможности (напр. доп. входы запросов прерываний).
- симметричный режим BV (symmetric I/o mode) – сообщения прерываний от ПУ генерирует APIC, прерывания могут доставляться любому процессору, каждый вход запроса инд-но программируется с пом. таблицы перенаправления прерываний BV (I/O redirection table).

# Advanced Peripheral Interrupt Controller

I/O APIC – позволяет вырабатывать значительное число прерываний, каждое из которых описано I/O redirection table – способ реакции, вектор прерывания и процессор(-ы) назначения.

Запросы – по инд. входам INTIN<sub>n</sub>, чувствительность по уровню или сигналу, вектор м.б. запрограммированы индивидуально для каждого входа.

**А хватит ли средств APIC для последующего роста BC?**

Более совершенные модели I/O APIC допускают вызов прерывания записью номера входа в регистр КП – используется при прерываниях MSI (Message Signaled Interrupts) на шине PCI – > экономия сигнальных входов->необязательно иметь инд. INTIN<sub>n</sub> -> число запросов ограничено размером таблицы I/O APIC.

# ПДП (DMA)

**Таблица** Стандартные каналы прямого доступа к памяти

Номер канала DMA#	0	1	2	3	4	5	6	7
Разрядность, байтов	1				2 с четного адреса			
Макс. размер блока	64 Кбайт				128 Кбайт, четный			
Граница блока	Кратна 1 0000h				Кратна 2 0000h			
Регистр страниц	8 бит A16–A23				7 бит A17–A23			
Адреса регистров:								
— страниц	087	083	081	082	08F	08B	089	087
— адреса	000	002	004	006	0C0	0C4	0C8	0CC
— счетчика	001	003	005	007	0C2	0C6	0CA	0CE

# ПДП (DMA)

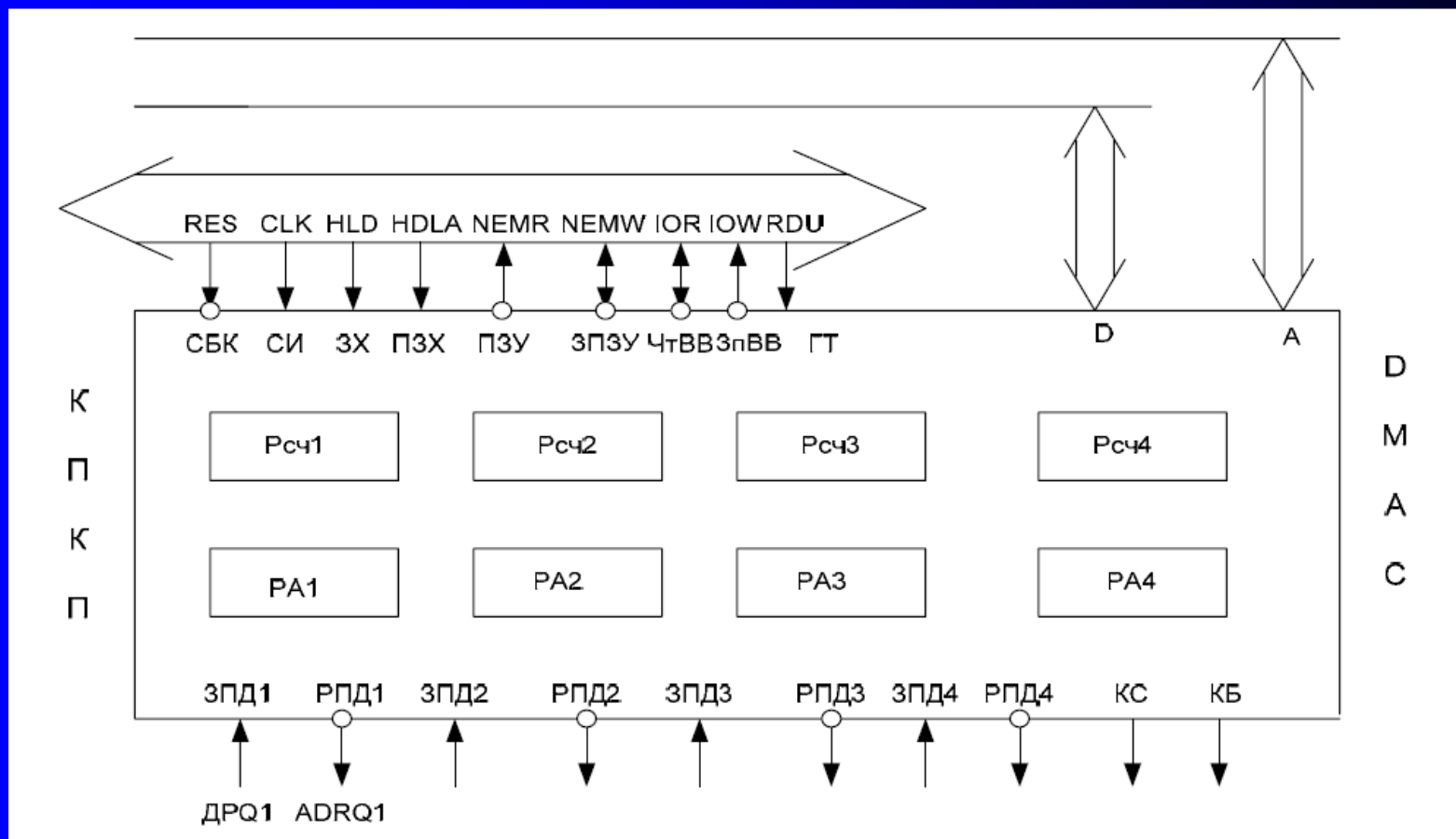


Схема контроллера ПДП



# ПДП (DMA)

Регистр адреса — куда писать. Регистр счётчика — количество слов. Сигналы сброса, синхронизации — очевидны.

ЗХ—захват(HLD-hold)- приходит ЗПД (DRQ1 DMA request)- запрос прямого доступа — КПДП передаёт на ЦП сигнал ЗХ, и тот отключается от СШ. И КПДП начинает работать после ПЗХ (подтверждение захвата—HLDA).

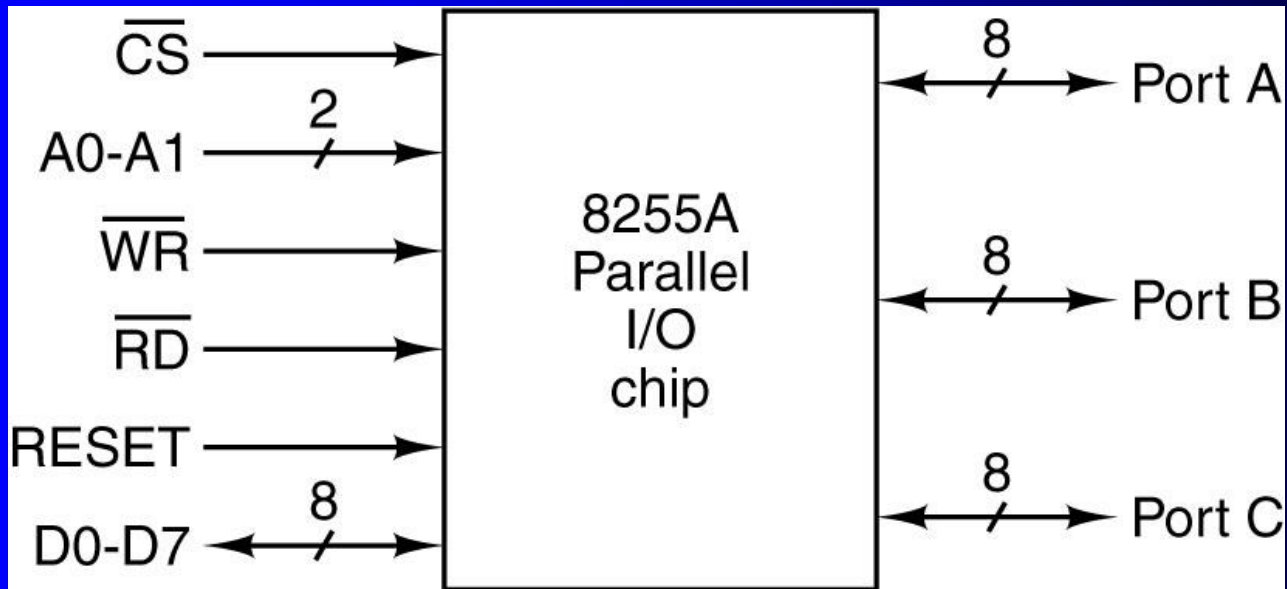
Устанавливается адрес и сигнал ЧтЗУ, КПДП передает сигнал ЗПВВ( запись данных во внешнее устройство), счетчик умножить на 1, адрес увеличить на 1. Для медленнодействующих устройств есть сигнал готовности(ГТ) . Наоборот — аналогично-сигналы ЧтВВ и ЗпЗУ. КПДП — является программируемым.

После передачи данных КПДП снимает сигнал ЗХ, восстанавливая работоспособность процессора.

В современных кэшируемых ЦП его работа в режиме ПДП может продолжаться, если ему не требуется СШ и все команды могут быть выполнены внутри кристалла.

# PIO Chips

An 8255A PIO chip.



# Example: The Pentium 4's Buses

