

## 20. Система прерываний ПЭВМ.

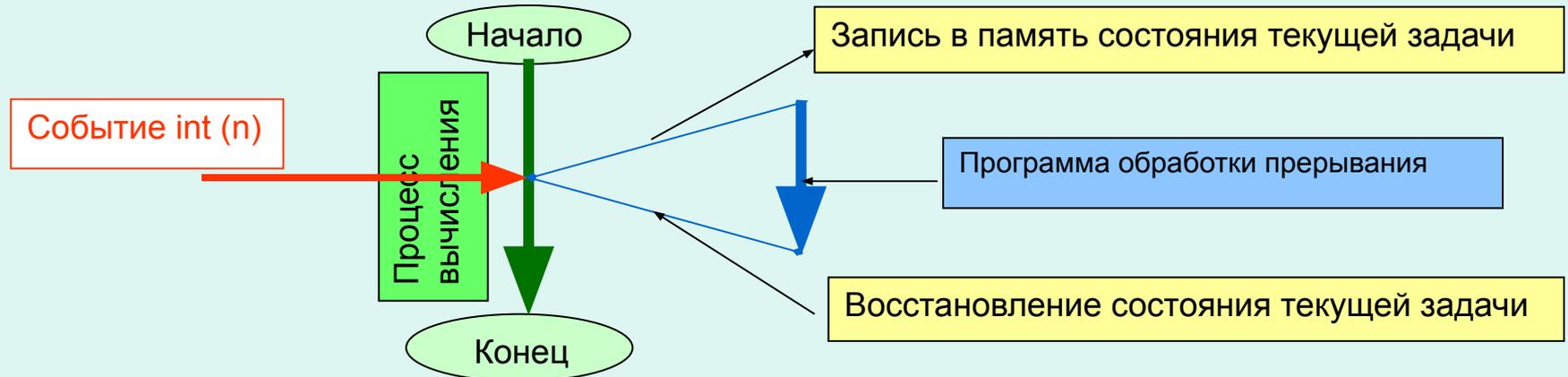
Основные цели, задачи и механизм прерываний исполнения программ.

**Цель прерывания программы:** остановка исполнения текущей программы в процессе её исполнения для выполнения другой программы, либо обработки какого либо события или ошибки исполнения программы.

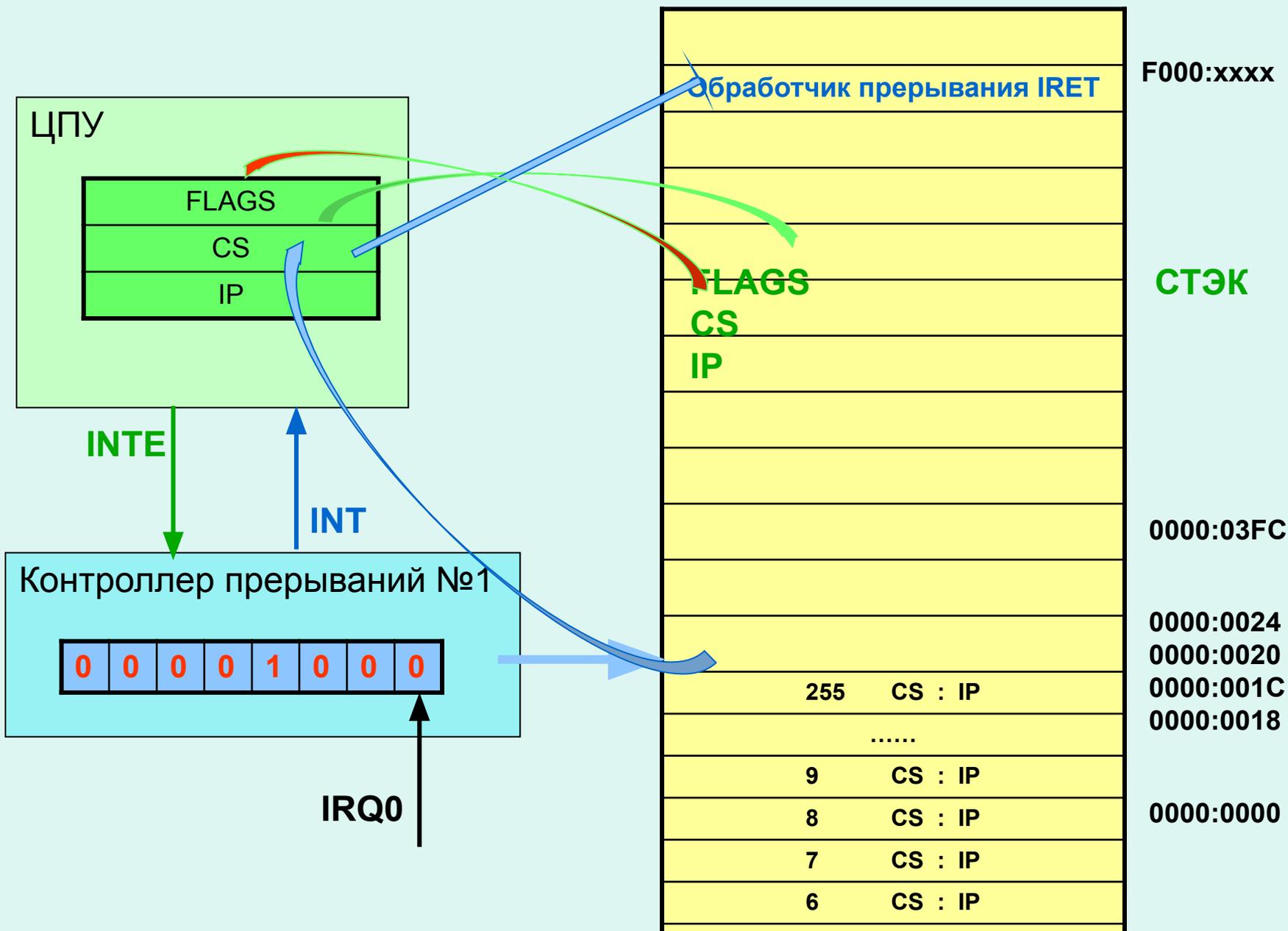
**Основные задачи обработки прерывания:**

1. Регистрация запросов на прерывание.
2. Определения готовности и приоритетов устройств, требующих обработки центральным процессором их требований.
3. Формирование адреса точки входа в программу - обработчик прерывания и инициализация процесса обработки прерывания с сохранением состояния текущей программы.
4. Восстановление исходного состояния текущей программы и продолжение вычислений после завершения обработки прерывания.

**Механизм прерываний исполнения программ.**



# Механизм обработки прерывания в реальном режиме.



## Типы прерываний в реальном режиме.

1. **Аппаратные (радиальные) прерывания** инициируются аппаратурой периферии, либо с системной платы или платы расширения. Они могут быть вызваны сигналом микросхемы таймера, сигналом от принтера, нажатием клавиши на клавиатуре и множеством других причин.

Аппаратные прерывания не связываются с работой программного обеспечения, но обязательно имеют соответствующие программы-обработчики, которые инициируются векторами прерываний, связанными с аппаратными прерываниями через контроллер прерываний.

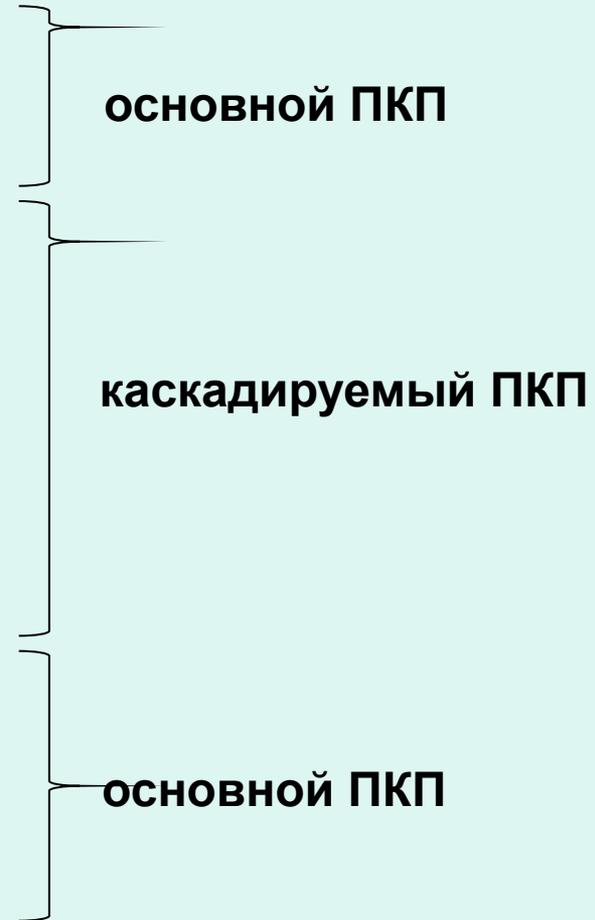
2. **Программные (векторные) прерывания** на самом деле ничего не прерывают. Это обычные процедуры или функции, которые вызываются прикладными программами для выполнения рутинной работы, такой, как прием нажатия клавиши на клавиатуре или вывод на экран. Эти программы находятся в составе операционной системы и механизм прерываний дает возможность обратиться к ним.

Программные прерывания могут вызываться друг из друга. При этом не возникает конфликтов, так как каждая подпрограмма обработки прерывания сохраняет значения всех используемых ею регистров и затем восстанавливает их при выходе, тем самым не оставляя следов того, что она занимала процессор.

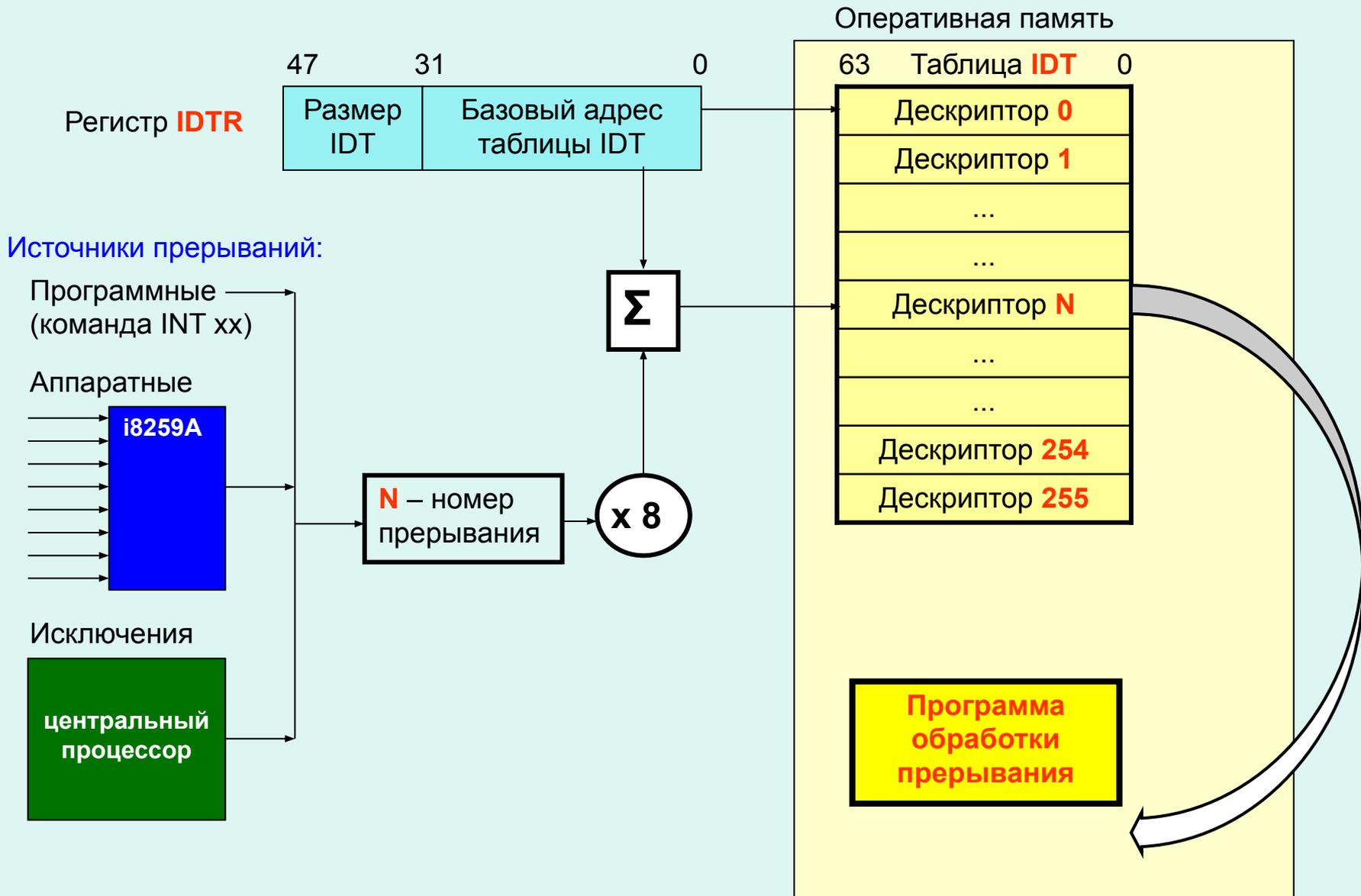
Адреса программ прерываний называют векторами. Каждый вектор имеет длину четыре байта. В первом слове хранится значение IP, а во втором - CS. Совокупность векторов составляет таблицу векторов.

**Система радиальных (аппаратных) прерываний.  
(Аппаратные прерывания в порядке их приоритета)**

- IRQ 0 таймер**
- IRQ 1 клавиатура**
- IRQ 2 канал ввода/вывода 2-го каскада**
- IRQ 8 часы реального времени**
- IRQ 9 свободный, PnP**
- IRQ 10 свободный, PnP**
- IRQ 11 свободный, PnP**
- IRQ 12 мышь PS/2**
- IRQ 13 мат. сопроцессор**
- IRQ 14 контроллер ATA**
- IRQ 15 свободный, PnP**
- IRQ 3 COM1 (COM2)**
- IRQ 4 COM2 (COM1)**
- IRQ 5 свободный, PnP**
- IRQ 6 контроллер FDD**
- IRQ 7 LPT1**



# Схема обработки прерываний в защищенном режиме.



# Исключения и прерывания.

**Исключения (особые ситуации) и прерывания** представляют собой принудительную передачу управления задаче или процедуре обработки особых случаев состояния процессора или аппаратной части ПЭВМ. Такая задача или процедура называется обработчиком.

**Прерывания** происходят в произвольные моменты времени выполнения программы в ответ на сигналы аппаратного обеспечения.

**Исключения** происходят вследствие выполнения команд, приводящих к этим исключениям. Обычно обслуживание прерываний и исключений выполняется способом, прозрачным для прикладных программ.

Прерывания используются для обработки событий, являющихся внешними по отношению к процессору, таких как запросы на обслуживание периферийных устройств.

Исключения обслуживают условия, обнаруживаемые процессором во время выполнения команд, например деление на 0.

Существует два источника прерываний и два источника исключений:

## 1. Прерывания

- Маскируемые прерывания, получаемые на входе INTR процессора x86.

Маскируемые прерывания не происходят до тех пор, пока не будет установлен флаг разрешения прерываний (IF).

- Немаскируемые прерывания, получаемые на вход NMI (Не-Маскируемое Прерывание) процессора. Процессор не обеспечивает механизма отключения немаскируемых прерываний.

## 2. Исключения

- Исключения, обнаруживаемые процессором.

Далее они классифицируются как сбой, ловушка или аварийное завершение.

- Программируемые исключения.

Команды INT 0, INT 3, INT n и BOUND могут служить программными переключателями исключений.

Эти команды часто называют "программными прерываниями", но они обрабатываются процессором как исключения.

## Типы исключений в защищенном режиме.

**Сбой (ошибка)** — прерывание или исключение, при возникновении которого в стек записываются значения регистров **cs: ip**, указывающие на команду, вызвавшую данное прерывание.

Это позволяет, получив доступ к сегменту кода, исправить ошибочную команду в обработчике прерывания и, вернув управление программе, фактически осуществить ее рестарт (в режиме прямой адресации при возникновении прерывания в стеке всегда запоминается адрес команды, следующей за той, которая вызвала это прерывание).

**Ловушка** — прерывание или исключение, при возникновении которого в стек записываются значения регистров **cs:ip**, указывающие на команду, следующую за командой, вызвавшей данное прерывание.

Здесь так же, как и в случае ошибок возможен рестарт программы. Для этого необходимо лишь исправить в обработчике прерывания соответствующие код или данные, послужившие источником ошибки. После этого перед возвратом управления нужно скорректировать значение **ip** в стеке на длину команды, вызвавшей данное прерывание.

Механизм ловушек похож на механизм прерываний в реальном режиме, за исключением случая, когда прерывание типа ловушки возникло в команде передачи управления **jmp**, то содержимое пары **cs: ip** в стеке будет отражать результат этого перехода, то есть соответствовать команде назначения.

**Аварийное завершение** — прерывание, при котором информация о месте его возникновения недоступна или неполна и поэтому рестарт команды практически невозможен, если только данная ситуация не была запланирована заранее созданием точки останова программы в режиме отладки.

Микропроцессор должен однозначно определять, какие прерывания являются ошибками, ловушками и авариями, а также определять возможное наложение аппаратных прерываний на прерывания защищенного режима.

Для этой цели в пересекающихся прерываниях дополнительно генерируется и записывается в стек так называемый **код ошибки**.

# Таблица векторов прерываний в реальном и защищенном режимах

№ вектора	Реальный режим	Защищенный режим	Тип прер.	Код ошибки
<b>0</b>	Деление на ноль	Деление на ноль	Ошибка	Не формируется
<b>1</b>	Пошаговой работы	Пошаговой работы	Ошибка /Ловушка	Не формируется
<b>2</b>	Сигнал на входе NMI	Сигнал на входе NMI	Ловушка	Не формируется
<b>3</b>	Контрольная точка	Контрольная точка	Ловушка	Не формируется
<b>4</b>	Переполнение	Переполнение	Ловушка	Не формируется
<b>5</b>	BIOS: Печать экрана /нарушение границ массива	Нарушение границ массива	Ошибка	Не формируется
<b>6</b>	Недопустимая команда	Недопустимая команда	Ошибка	Не формируется
<b>7</b>	Обращение к отсутствующему сопроцессору	Обращение к отсутствующему сопроцессору	Ошибка	Не формируется
<b>8</b>	Микросхема таймера /двойная ошибка	Двойная ошибка	Авария	Формируется (всегда 0)
<b>9</b>	Клавиатура	Не используется	—	Формируется
<b>10 (0Ah)</b>	Сигнал на линии IRQ 2	Ошибочный TSS	Ошибка	Формируется
<b>11 (0Bh)</b>	Сигнал на линии IRQ 3	Отсутствие сегмента	Ошибка	Формируется
<b>12 (0Ch)</b>	Сигнал на линии IRQ 4 /выход за пределы стека	Выход за пределы стека или отсутствие стека	Ошибка	Формируется
<b>13 (0Dh)</b>	Сигнал на линии IRQ 5 /нарушение общей защиты	Нарушение общей защиты	Ошибка	Формируется
<b>14 (0Eh)</b>	Контроллер FDD - сигнал на линии IRQ 6	Отсутствие страницы памяти	Ошибка	Формируется
<b>15 (0Fh)</b>	Сигнал на линии IRO 7	Не используется	—	—
<b>16 (10h)</b>	BIOS: функции видеосистемы /ошибка сопроцессора	Ошибка сопроцессора	Ошибка	Не формируется
<b>17 (11h)</b>	Используется для прерываний BIOS и DOS	Ошибка выравнивания	Ошибка	Формируется (всегда 0)
<b>18-31 (12h-19h)</b>	Используется для прерываний BIOS и DOS	Зарезервировано	—	—
<b>32-255 (20h-FFh)</b>	Используется для прерываний BIOS и DOS	Аппаратные прерывания; прерывания, определяемые пользователем и OS	Ловушка	Не формируется

## Содержание информации, сохраняемой в стеке в защищенном режиме.



### Содержание битов кода ошибки:

- 0** =1 вызвана аппаратным прерыванием,  
=0 вызвана последней выполняющейся командой
- 1** =1 в поле **index** содержится индекс дескриптора в таблице IDT  
=0 в поле **index** содержится индекс дескриптора в таблице GDT или LDT
- 2** =1 значение в поле **index** соответствует номеру дескриптора в таблице LDT  
=0 значение в поле **index** соответствует номеру дескриптора в таблице GDT
- 3...15** – номер (**index**) дескриптора в одной из таблиц IDT, LDT или GDT в соответствии с битами 1 и 2 или 000 для прерываний 8 и 17.

Для корректного возврата из обработчика прерывания с кодом ошибки необходимо применять в нем команду **POP** перед командой **IRET** для очистки стека от кода ошибки.

Дескрипторы в таблице IDT строго упорядочены в соответствии с номерами прерываний.

Размерность таблицы IDT — не более 256 элементов размером по восемь байт, по числу возможных источников прерываний.

Допускается неполное заполнение таблицы IDT (в отдельных случаях описываются не все 256 дескрипторов этой таблицы), для неиспользуемых номеров прерываний формируются нулевые шлюзы-заглушки. Если этого не сделать, то при незапланированном прерывании или с номером, превышающим пределы IDT для данной задачи, будет возникать ситуация исключения - сбой общей защиты (с номером 13 (0Dh)).

Микропроцессор всегда определяет местоположение таблицы IDT по полю базового адреса в регистре **IDTR** независимо от режима (в реальном режиме поля базового адреса **IDTR** равны нулю).

## Формат дескрипторов в таблице IDT.

Дескрипторы в таблице прерываний IDT называются **шлюзами** (в отличие от дескрипторов в таблицах GDT и LDT).

Шлюз предназначен для указания точки входа в программу обработки прерывания.

В дескрипторной таблице прерываний IDT могут содержаться шлюзы трех типов: шлюз ловушки, шлюз прерывания и шлюз задачи. Что объясняется наличием различных источников прерываний и особенностями передачи управления в программы их обработки. Центральный процессор отличает шлюзы по значению в поле типа **type** (биты 40-43).

**Шлюз ловушки:**

63	47	type				39	36	32	31	15	0
Offset_2	p	dpl	0	1111	000	Не используется	indicator	Offset_1			

**Шлюз прерывания** (запрещает обработку аппаратных прерываний):

63	47	type				39	36	32	31	15	0
Offset_2	p	dpl	0	1110	000	Не используется	indicator	Offset_1			

**Шлюз задачи** (указывает через индикатор на сегмент задачи TSS – обработчика прерывания):

63	47	type				39	36	32	31	15	0
Не используется	p	dpl	0	0101	000	Не используется	indicator	Не используется			

Обозначения на схеме:

**p** – бит присутствия соответствующей программы обработки.

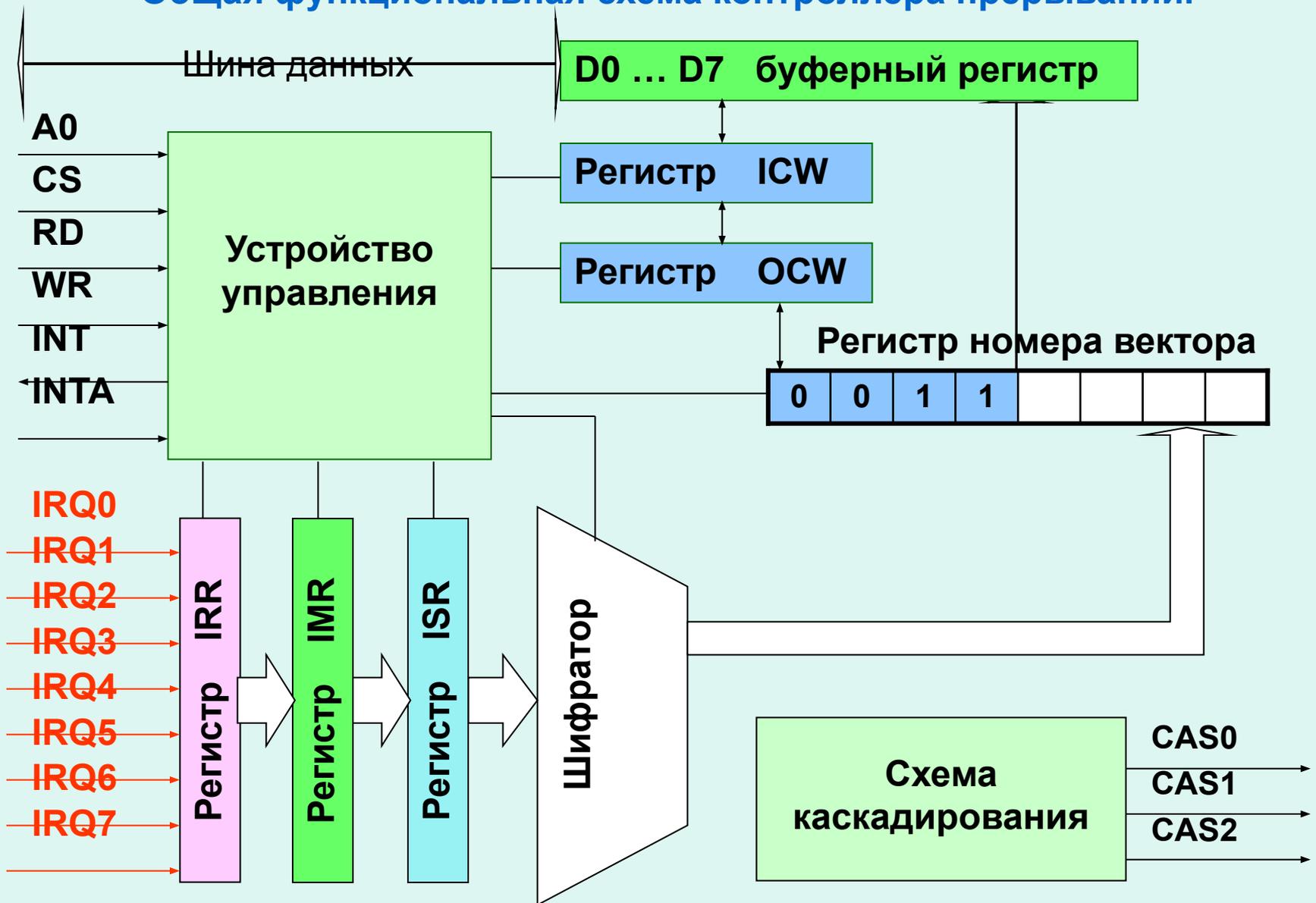
**dpl** – минимальный уровень привилегий задачи, которая может передать управление обработчику прерываний через данный шлюз (обычно 10 или 01).

**indicator** – селектор, указывающий на дескриптор в LDT или GDT.

**Offset\_1** и **Offset\_2** - смещение в сегменте, определяемом дескриптором, первое и второе слово соответственно.

# 21. Контроллер прерываний i8259A

Общая функциональная схема контроллера прерываний.



## **Основные функции контроллера:**

- 1. Фиксация запросов на прерывания от восьми внешних источников.**
- 2. Программное маскирование поступающих запросов.**
- 3. Присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы.**
- 4. Определение адреса и инициация вызова процедуры обработки поступившего аппаратного прерывания.**

## **Назначение основных устройств и регистров ПКП.**

- контроллер (устройство) управления;**
- схема каскадирования;**
- регистр фиксации поступления запросов на прерывания IRR;**
- регистр маскирования запросов на прерывания IMR;**
- регистр состояния исполнения запроса на прерывание ISR;**
- регистр управляющего слова ICW;**
- регистр операционного управления OCW.**

**ПКП может находиться в двух основных состояниях: настройки и обслуживания запросов на прерывания.**

**В состоянии настройки контроллер принимает управляющие слова инициализации (Initialization Command Words, ICW).**

**В состоянии обслуживания - операционные управляющие слова (Operation Control Words, OCW).**

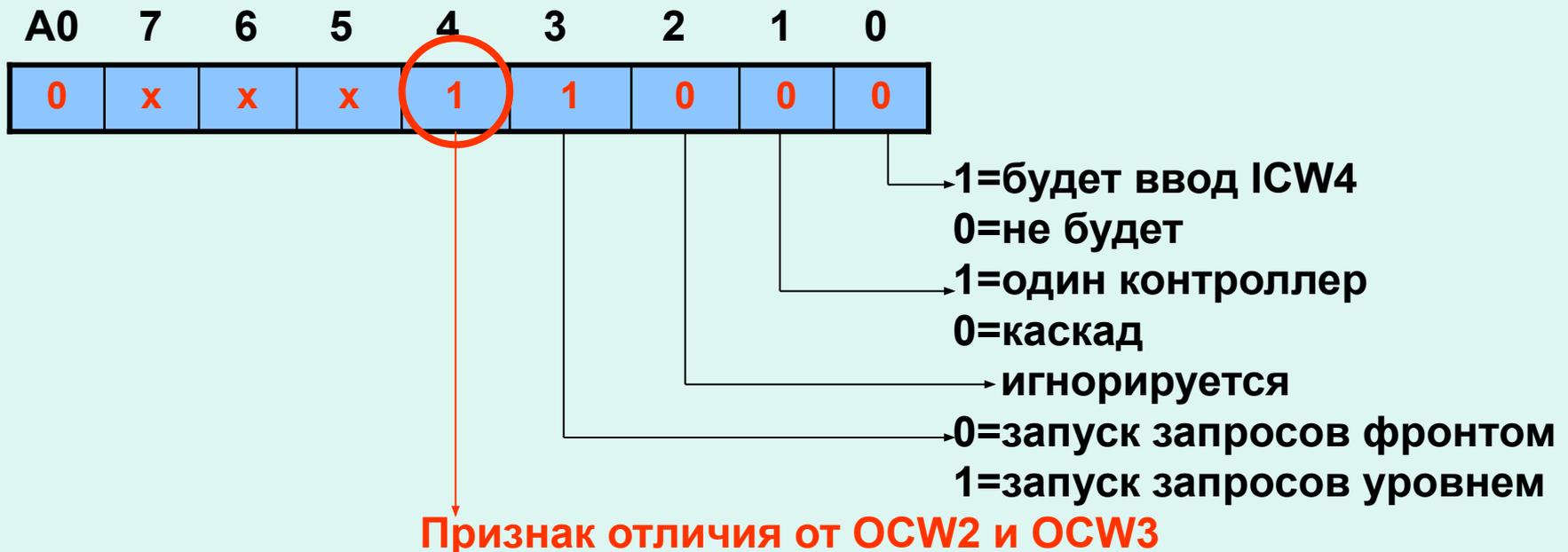
## Порядок программирования ПКП:

Для ввода информации в ПКП используются 2 порта ввода-вывода. Порт с четным адресом (20h/A0h) и порт с нечетным адресом (21h/A1h). Через эти порты могут быть переданы 4 слова инициализации, (ICW1- ICW4), задающие режим работы ПКП, и 3 операционных управляющих слова (слова рабочих приказов OCW1 - OCW3).

В порт с четным адресом выводятся ICW1, OCW2 и OCW3, а в порт с нечетным адресом – ICW2, ICW3, ICW4 и OCW1.

### Формат слов инициализации

#### Формат ICW1



## Формат ICW2

Величина смещения номера вектора от номера аппаратного прерывания

A0 7 6 5 4 3 2 1 0

1	0	0	0/1	0	1/0	0	0	0
---	---	---	-----	---	-----	---	---	---

для ведущего контроллера (8/32)

1	0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---

для ведомого контроллера (112)

## Формат ICW3

Для ведущего контроллера показывает к каким входам IRQn подключен ведомый (каскадируемый) контроллер -

A0 7 6 5 4 3 2 1 0

1	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---

для ведущего контроллера (2)

Для ведомого контроллера, работающего на уровне 1

1	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---

для ведомого контроллера (1)

## Формат ICW4

A0 7 6 5 4 3 2 1 0

1	0	0	0	x	x	x	x	x
---	---	---	---	---	---	---	---	---

завершение инициализации

тип центрального процессора

режим автоматического завершения

равен 1 для ведомого контроллера

режим буферизации

вложенный режим определения приоритетов ПКП

В процессе работы ПКП может быть перепрограммирован:

- допускается маскировать и размаскировать аппаратные прерывания;
- допускается изменять приоритеты уровней;
- можно давать команду завершения обработки аппаратного прерывания;
- допускается устанавливать/сбрасывать режим специальной маски;
- разрешен перевод контроллера в режим опроса и считывания

состояния регистров ISR и IRR;

Для этого используется одно из слов рабочих приказов OCW1 - OCW3.

## Форматы слов рабочих приказов

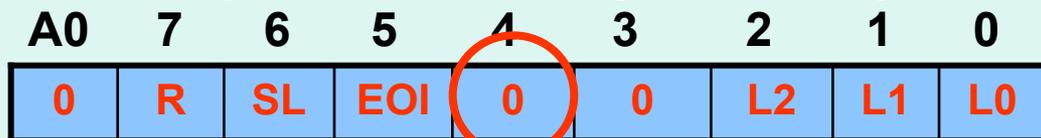
### Формат OCW1



Единичное значение одного из битов M0 - M7 означает, что прерывания соответствующего уровня (IR0 - IR7) маскируются и не будут обрабатываться контроллером.

Четвертый бит у ICW2, ICW3, ICW4 и OCW1 не является определяющим, т.к. все эти слова вводятся в порт с нечетным номером (A0=1) и используются все биты байта.

## Формат OCW2 - Порядок завершения аппаратного прерывания



3 бита, которые определяют номер уровня прерывания, если он требуется в команде

команда принудительного завершения обработки аппаратного прерывания

- 0 0 - принудительное завершение (используется совместно с EOI)
  - 0 1 - специфицированное завершение (используется совместно с EOI) в регистре ISR сбрасывается бит, определяемый полями L
  - 1 0 - циклический сдвиг приоритетов влево на одну позицию
  - 1 1 - назначение низшего приоритета уровню, определяемому полями L1 - L3.
- Признак отличия от ICW1**

Процедура обработки аппаратного прерывания должна перед своим завершением очистить свой бит в ISR выводом команды завершения обработки прерывания (End Of Interrupt, EOI).

Существует два варианта команды EOI: обычный и специфицированный. Обычный EOI очищает бит в ISR, соответствующий прерыванию с максимальным приоритетом.

Специфицированный EOI (R=0, SL=1, EOI =1, L0-L2 равно номеру уровня прерывания) очищает в ISR бит, соответствующий прерыванию с номером, указанным в L0 - L2 независимо от его приоритета.

## Формат OCW3 - Изменение режима и разрешение опроса IRR и ISR.

A0 7 6 5 4 3 2 1 0



Признак отличия от ICW1

Единичное значение бита P (бит опроса, Polling Bit) переводит контроллер в режим опроса. Если после этого считать данные из порта с четным адресом, получим байт следующего содержания:

7 6 5 4 3 2 1 0



Если I = 1, значит имеются запросы на прерывания и тогда L0 - L2 - это номер уровня с наивысшим приоритетом, по которому имеется запрос на прерывание.

Если P=0, можно считать информацию из ISR или IRR. Для этого необходимо дать команду чтения ISR или IRR и затем считать значение из порта с нечетным адресом.

## Система приказов BIOS для инициализации ПКП.

Код приказа	Вид приказа и в какой контроллер посылается	порт
00010001	ICW1 в контроллер 1 - будет указан тип архитектуры процессора	020h
00001000 *	ICW2 в контроллер 1 - величина смещения номера вектора на 8h	021h
00100000	<b>ICW2 в контроллер 1 - смещение вектора на 20h (защищ. режим)</b>	021h
00000100	ICW3 в контроллер 1 - указывает на ведущего в каскаде	021h
00000001	ICW4 в контроллер 1 - тип архитектуры процессора - x86	021h
00010001	ICW1 в контроллер 2 - будет указан тип архитектуры процессора	0A1h
01110000	ICW2 в контроллер 2 - величина смещения номера вектора на 70h	0A1h
00000010	ICW3 в контроллер 2 - указывает на ведомого в каскаде	0A1h
00000001	ICW4 в контроллер 2 - тип архитектуры процессора - x86	0A1h
10111000	В контроллер 1 - маска прерывания OCW1	021h
10111101	В контроллер 2 - маска прерывания OCW2	0A1h

Примечание: \* - при программировании смещения номера вектора относительно номера аппаратного прерывания необходимо изменять его величину на 20h для защищенного режима с целью исключения наложения адресов обработчиков внутренних прерываний и восстанавливать его значение 8h при возврате в режим прямой адресации. Перепрограммировать ведомый не требуется, т.к. он указывает на общепользовательские прерывания.

## Режимы формирования приоритетов:

1. **Режим фиксированных приоритетов.** Запросы прерываний имеют жесткие приоритеты от 0 до 7 (0 - высший) и обрабатываются в соответствии с приоритетами. При этом прерывание с меньшим приоритетом никогда не будет обработано, если в процессе обработки прерываний с более высокими приоритетами постоянно возникают запросы на эти прерывания.

2. **Автоматический сдвиг приоритетов.** В этом режиме дается возможность обработать прерывания всех уровней без их дискриминации. Например, после обработки прерывания уровня 4 ему автоматически присваивается низший приоритет, при этом приоритеты для всех остальных уровней циклически сдвигаются.

3. **Программно-управляемый сдвиг приоритетов.** Программист может сам передать команду циклического сдвига приоритетов ПКП, задав соответствующее управляющее слово. В команде задается номер уровня, которому требуется присвоить максимальный приоритет. После выполнения такой команды устройство работает так же, как и в режиме фиксированных приоритетов, с учетом их сдвига. Приоритеты сдвигаются циклически, таким образом если максимальный приоритет был назначен уровню 3, то уровень 2 получит минимальный и будет обрабатываться последним.

4. **Режим специальной маски.** Данный режим позволяет отменить приоритетное упорядочение обработки запросов и обрабатывать их по мере поступления. После отмены режима специальной маски предшествующий порядок приоритетов уровней сохраняется.

## Режимы завершения прерываний ПКП:

### 1. Режим автоматического завершения обработки прерывания (AEOI).

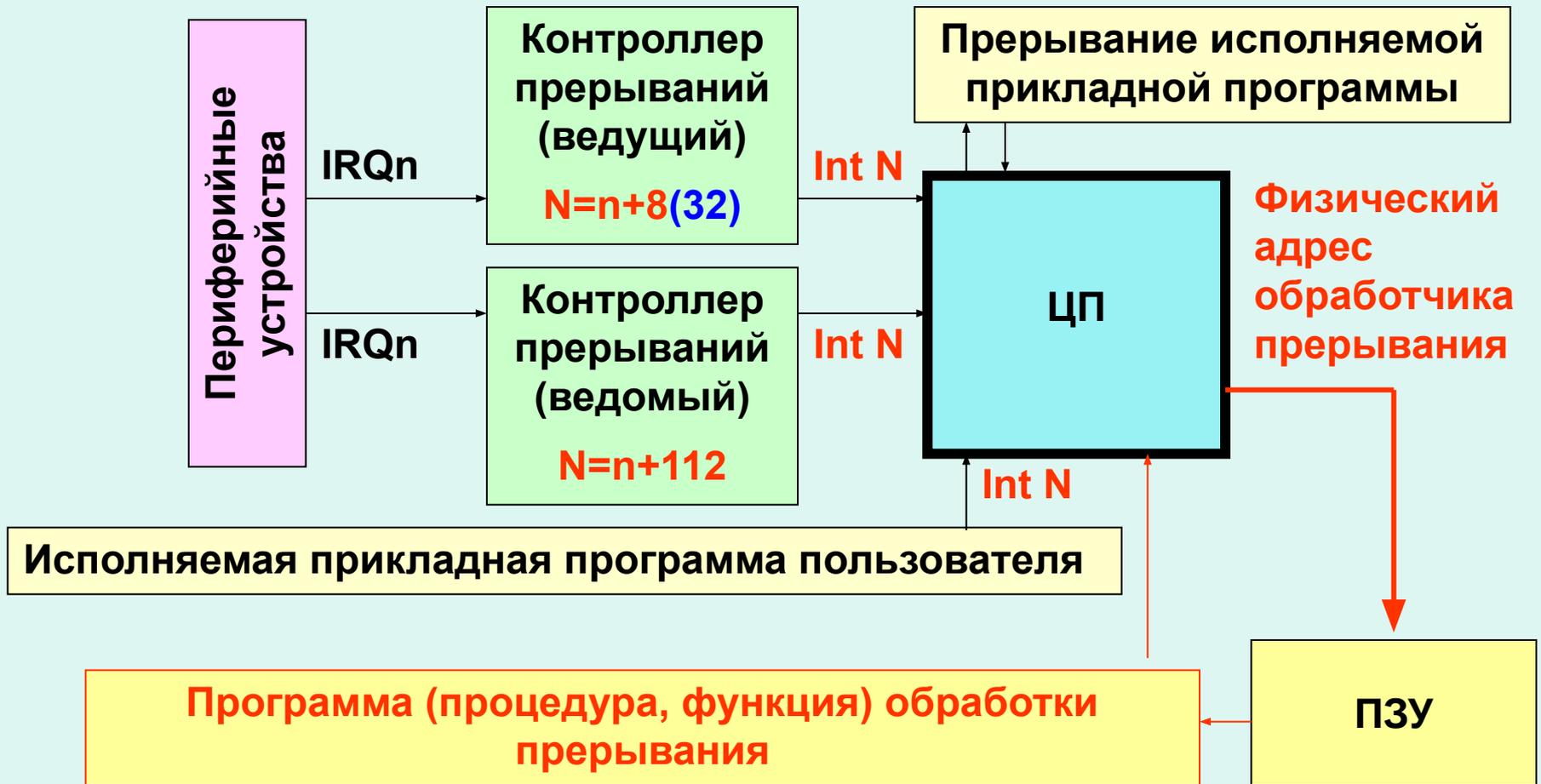
В обычном режиме работы процедура обработки аппаратного прерывания должна перед своим завершением очистить свой бит в регистре ISR специальной командой, иначе новые прерывания не будут обрабатываться ПКП. В режиме AEOI нужный бит в ISR автоматически сбрасывается в тот момент, когда начинается обработка прерывания нужной процедурой обработки и от нее не требуется издавать команду завершения обработки прерывания (EOI).

Сложность работы в данном режиме обуславливается тем, что все процедуры обработки аппаратных прерываний должны быть повторно входными, т. к. за время их работы могут повторно возникнуть прерывания того же уровня.

### 2. Режим опроса (Polling Mode).

В этом режиме аппаратные прерывания не происходят автоматически. Появление запросов на прерывание должно определяться считыванием IRR. Данный режим позволяет так же получить от ПКП информацию о наличии запросов на прерывания и, если запросы имеются, номер уровня с максимальным приоритетом, по которому есть запрос.

# Схема формирования адреса вектора по номеру аппаратного (радиального) прерывания для ведущего и ведомого контроллера прерываний



## 23. Подсистема прямого доступа к памяти.

### Назначение прямого доступа к памяти.

Прямой доступ к памяти (DMA - Direct Memory Access) обеспечивает высокоскоростной обмен данными между устройствами ввода-вывода и ОЗУ без использования центрального процессора, что позволяет освободить процессор для выполнения вычислений параллельно с обменом и независимо от него.

Наиболее часто возможности DMA используются при работе с дисковыми накопителями, однако реализовано использование DMA и адаптерами накопителей других типов, а также рядом других устройств.

Ощутимые преимущества дает использование DMA в процессе обмена данными с устройствами, принимающими или передающими данные достаточно большими порциями и с высокой скоростью.

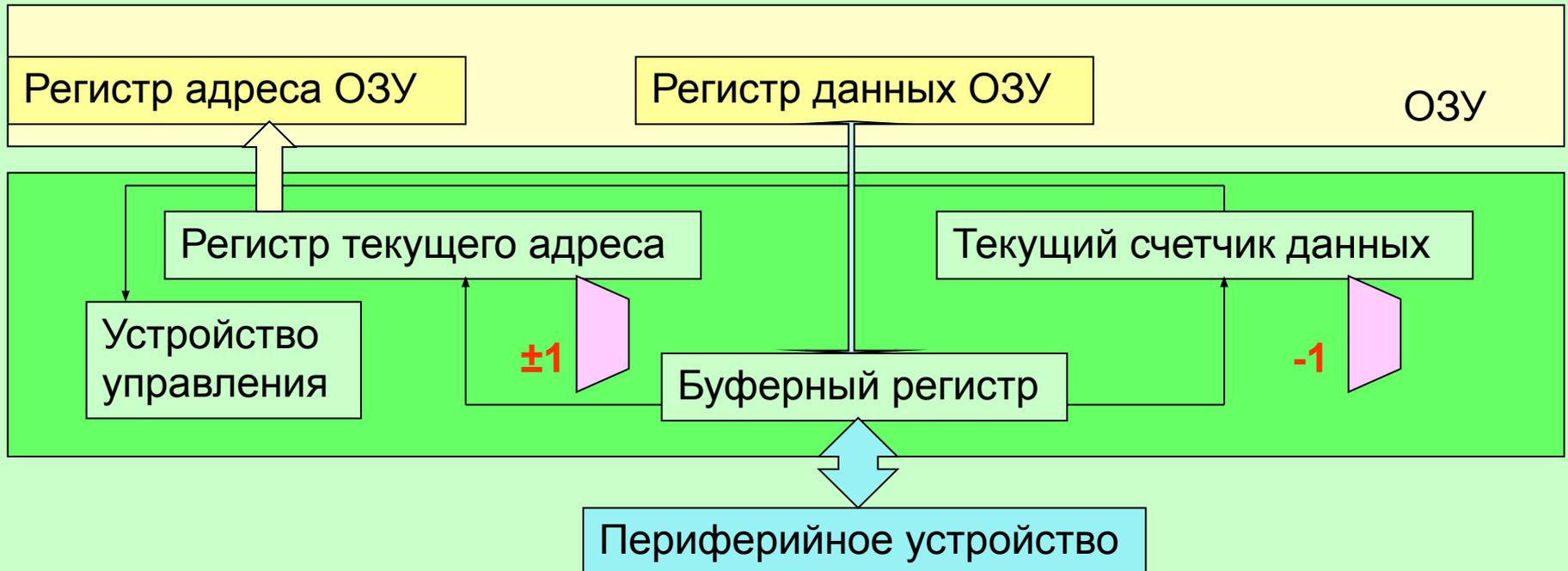
В ПЭВМ функции DMA выполняет контроллер прямого доступа к памяти (ПДП), построенный на базе микросхемы i8237A (INTEL) или ее аналоги i8237A-4 и i8237A-5, работающие с тактовой частотой 4 и 5 МГц соответственно. Контроллер имеет 4 независимых канала, каждый из которых может обслуживать одно периферийное устройство.

В современных схемах аппаратной части ПЭВМ частота контроллера ПДП соответствует частоте системной шины для обеспечения синхронной передачи данных в режиме память-память. Следует также учитывать, что многие современные устройства не поддерживают режим ПДП из-за сложности его инициализации и малой разрядности, предпочитая режим PIO.

## Основные задачи прямого доступа к памяти.

1. Управление инициализируемой периферийным устройством или центральным процессором передачей данных между ОЗУ и периферийным устройством.
2. Задание размера блока данных, который подлежит передаче, и область памяти, используемой при передаче.
3. Формирование адресов ячеек ОЗУ, участвующих в передаче.
4. Подсчет объема данных, передаваемых между периферийным устройством и ОЗУ, определение момента завершения операции ПДП.

## Функциональная схема контроллера ПДП.



## **Порядок работы контроллера прямого доступа к памяти.**

- 1. При инициализации ПДП от центрального процессора в текущий счетчик данных заносится размер подлежащих передаче блока данных (число байт или слов).**
- 2. В регистр текущего адреса заносится начальный адрес области памяти ОЗУ, используемой при передаче.**
- 3. Центральный процессор освобождает шину данных и передает управление передачей данных контроллеру прямого доступа до завершения передачи (в соответствии с требованиями режима передачи данных).**
- 4. В процессе передачи каждого байта (или слова) содержимое регистра текущего адреса уменьшается или увеличивается на единицу и через масштабный коэффициент формируется адрес очередной ячейки памяти.**
- 5. Одновременно уменьшается на единицу содержимое текущего счетчика данных.**
- 6. Обнуление текущего счетчика данных указывает на завершение операции по прямой передаче данных между периферийным устройством и ОЗУ.**
- 7. Устройство управления формирует сигнал центральному процессору о завершении передачи данных и освобождении шины.**

**Контроллер ПДП обычно имеет наивысший приоритет в занятии шины и управление ОЗУ переходит к контроллеру ПДП сразу после его инициализации.**

## Режимы работы контроллера ПДП.

### 1. Режим одиночной передачи (Single Transfer Mode).

После каждого цикла передачи контроллер освобождает шину процессору, но сразу же начинает проверку сигналов запроса и, как только обнаруживает активный сигнал запроса, инициирует следующий цикл передачи.

### 2. Режим блочной передачи (Block Transfer Mode).

В этом режиме наличие сигнала запроса требуется только до момента выдачи контроллером сигнала "Подтверждение запроса на ПДП" (DACK), после чего шина не освобождается вплоть до завершения передачи блока.

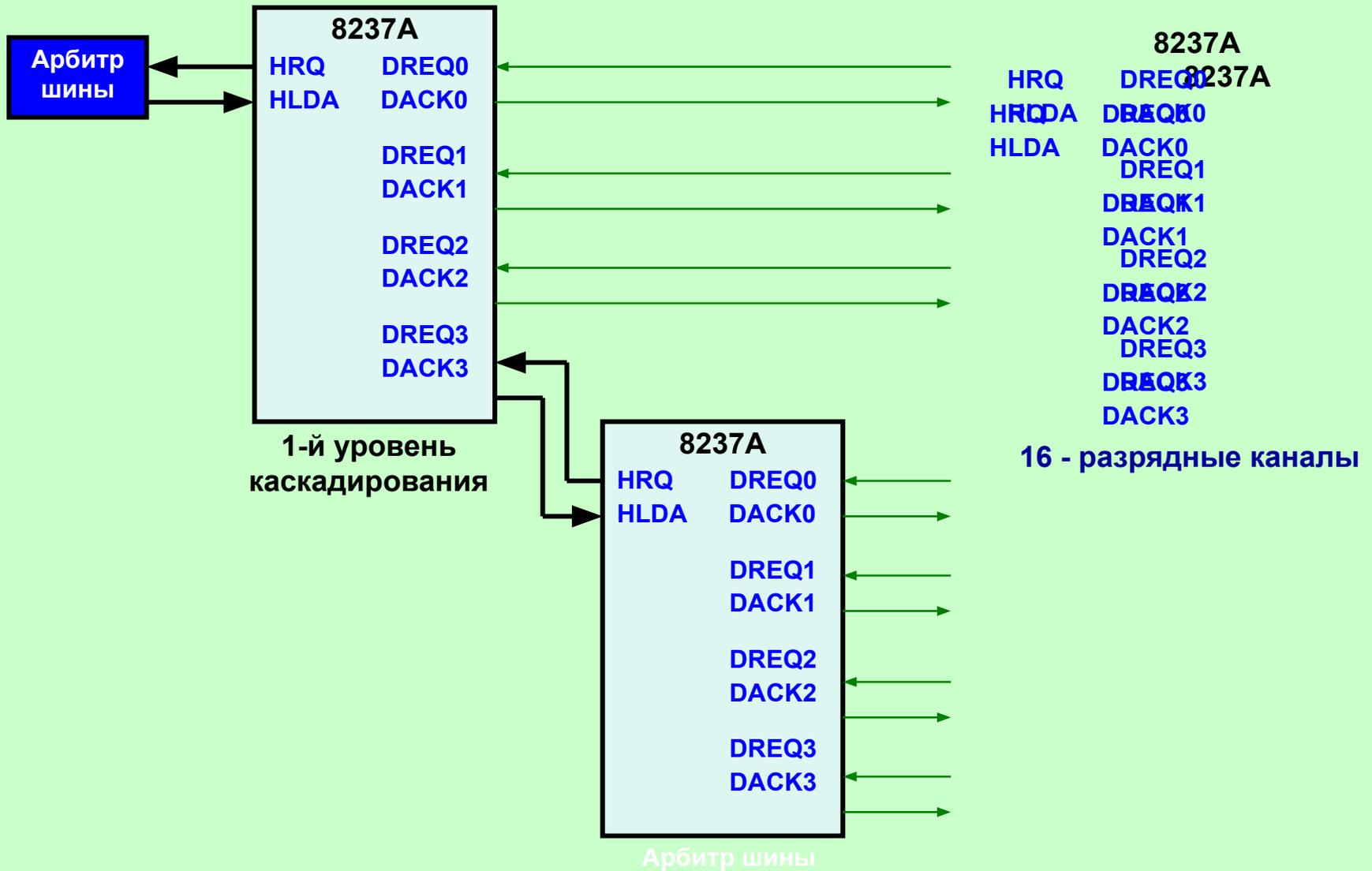
### 3. Режим передачи по требованию (Demand Transfer Mode).

Данный режим является промежуточным между двумя первыми: передача идет непрерывно до тех пор, пока активен сигнал запроса, состояние которого проверяется после каждого цикла передачи. Как только устройство не может продолжить передачу, сигнал запроса сбрасывается им и контроллер приостанавливает работу. Этот режим применяется для обмена с медленными устройствами, не позволяющими по своим временным характеристикам работать с ПДП в режиме блочной передачи.

### 4. Каскадный режим (Cascade Mode).

Режим позволяет включить в подсистему ПДП более одного контроллера в тех случаях, когда недостаточно четырех каналов ПДП. В этом режиме один из каналов ведущего контроллера используется для каскадирования с контроллером второго уровня. Для работы в каскаде сигнал HRQ ведомого контроллера подается на вход DREG ведущего, а сигнал DACK ведущего подается на вход HDLA ("Подтверждение захвата шины") ведомого.

# Блок-схема использования контроллеров ПДП в ПЭВМ



## Типы передач данных контроллером ПДП i8237A

### 1. Передача память-память (Memory-to-memory DMA).

Используется для передачи блока данных из одного места памяти в другое. Исходный адрес определяется в регистрах нулевого канала, выходной - в регистрах первого канала. Передача происходит с использованием рабочего регистра контроллера в качестве промежуточного звена для хранения информации.

### 2. Автоинициализация (автозагрузка, Autoinitialization).

После завершения обычной передачи использованный канал ПДП маскируется и должен быть перепрограммирован для дальнейшей работы с ним. При автоинициализации маскировка канала после окончания передачи не происходит, а регистры текущего адреса и счетчик циклов автоматически загружаются из соответствующих регистров с начальными значениями.

### 3. Режим фиксированных приоритетов.

В этом режиме канал 0 всегда имеет максимальный приоритет, а канал 3- минимальный и любая передача по каналу с более высоким приоритетом будет выполняться раньше, чем по каналу с более низким приоритетом.

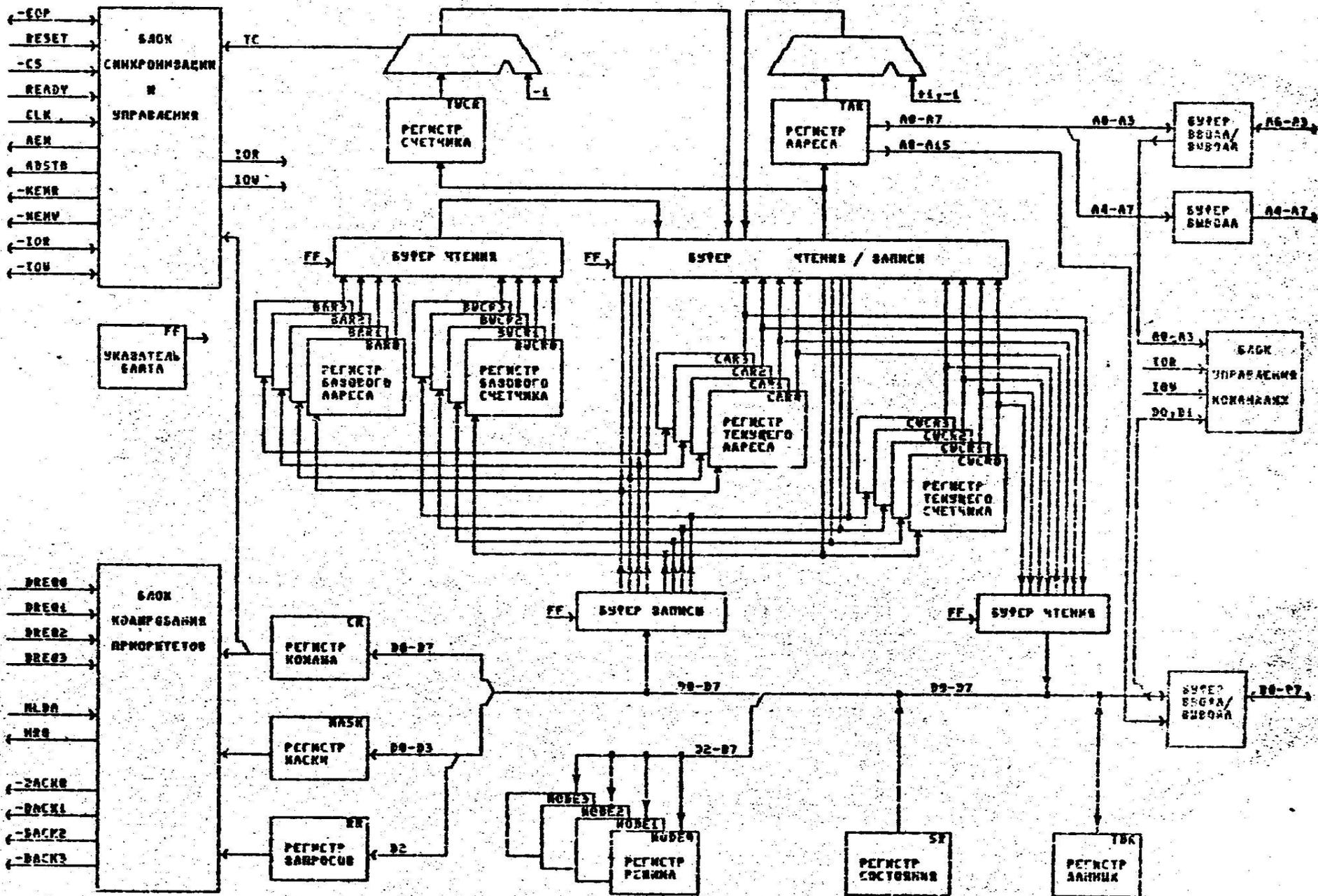
### 4. Циклический сдвиг приоритетов.

Каждому каналу, по которому прошла передача, автоматически присваивается низший приоритет, после чего право на передачу получает канал с наивысшим приоритетом.

### 5. Сжатие времени передачи (Compressed transfer timing).

При синхронной передаче ПДП может сократить время выполнения такта передачи на 2 цикла за счет тактов ожидания, входящих в каждый цикл.

# Блок-схема контроллера ПДП i8237A



## Сигналы контроллера ПДП i8237A

обозначение	тип	Функциональное назначение
RESET	Вход	Сброс состояния
READY	Вход	Готовность контроллера к ПДП
HLDA	Вход	Подтверждение захвата шины
DREQn	Входы	Запрос DMA периферийными устройствами n
D0 – D7	Вх/вых	Шина данных
-IOR	Вх/вых	Сопровождение чтения порта (активен низким уровнем)
-IOW	Вх/вых	Запись в порт (активен низким уровнем)
-EOP	Вх/вых	Завершение процесса ПДП
A0 – A3	Вх/вых	Младшая часть адреса ОЗУ
A4 – A7	Выход	Старшая часть адреса ОЗУ
HRQ	Выход	Запрос на захват шины данных контроллером ПДП
-DACKn	Выход	Подтверждение запроса DMA периферийному устройству
AEN	Выход	Разрешение адреса
ADSTB	Выход	Строб адреса
-MEMR	Выход	Сопровождение чтение памяти (активен низким уровнем)
-MEMW	Выход	Сопровождение записи в память (активен низким уровнем)
PIN5	Вход	Контроль питания
T/C	Выход	Бит завершения передачи
-CS	Вход	Выбор кристалла (каскада)
CLK	Вход	Сигнал синхронизации от генератора

## Регистры контроллера ПДП i8237A

Наименование регистра	разрядность	Кол-во
Регистр начального адреса	16	4
Регистр начального счетчика циклов	16	4
Регистр текущего адреса	16	4
Регистр текущего счетчика циклов	16	4
Рабочий регистр адреса	16	1
Рабочий регистр счетчика циклов	16	1
Регистр состояния	8	1
Регистр команд	8	1
Регистр режима	6	4
Рабочий регистр	8	1
Регистр масок	4	1
Регистр запросов	4	1

## Порты ввода-вывода основного контроллера ПДП i8237A

Порт	Режим	Назначение порта
0h	запись	Запись начального адреса в регистр начального адреса и регистр текущего адреса канала 0
2h	запись	то же для канала 1
4h	запись	то же для канала 2
6h	запись	то же для канала 3
0h	чтение	Чтение начального адреса из регистра начального адреса канала 0
2h	чтение	то же для канала 1
4h	чтение	то же для канала 2
6h	чтение	то же для канала 3
1h	запись	Запись в регистр начального счетчика циклов в регистр текущего счетчика циклов канала 0
3h	запись	то же для канала 1
5h	запись	то же для канала 2
7h	запись	то же для канала 3
1h	чтение	Чтение текущего значения из регистра текущего счетчика циклов канала 0
3h	чтение	то же для канала 1
5h	чтение	то же для канала 2
7h	чтение	то же для канала 3

## Порты ввода-вывода основного контроллера ПДП i8237A

Порт	Режим	Назначение порта
8h	запись	Запись регистра команд ПДП
8h	чтение	Чтение регистра состояния ПДП
9h	запись	Запись в регистр запросов ПДП
0Ah	запись	Запись бита маски для одного из каналов ПДП
0Bh	запись	Запись регистра режимов для одного из каналов ПДП
0Ch	запись	Очистка (сброс) триггера-защелки (триггера первого/последнего)
0Dh	запись	Программный сброс контроллера
0Eh	запись	Очистка битов масок всех 4-х каналов
0Fh	запись	Запись регистра масок для всех 4-х каналов
0Dh/0Fh	чтение	Чтение рабочего регистра ПДП
81h	запись	Задание номера страницы для канала 2
82h	запись	то же для канала 3
83h	запись	то же для канала 1
87h	запись	то же для канала 0

Аналогично распределены портовые адреса для каскадируемого контроллера ПДП:

0C0h - 0CCh, 89h – 8Fh, 0D0 – 0DAh

## **Программирование контроллера ПДП i8237A**

**В работе ПДП различаются 2 главных цикла: цикл ожидания (Idle cycle) и активный цикл (Active cycle). Каждый цикл подразделяется на ряд состояний, занимающих по времени один период часов (тактовый интервал).**

**Из цикла ожидания контроллер может быть переведен в состояние программирования (Program Condition) путем подачи на вход RESET сигнала высокого уровня, длительностью не менее 300 нс и следующей за ним подачи сигнала низкого уровня (уровня 0) на вывод CS (Chip Select).**

**В состоянии программирования контроллер будет находиться до тех пор, пока на выводе CS сохранится сигнал низкого уровня.**

**В процессе программирования контроллеру задаются:**

- начальный адрес памяти для обмена;**
- уменьшенное на единицу число передаваемых байтов;**
- направление обмена,**

**а также устанавливаются требуемые режимы работы (разрешить или запретить циклическое изменение приоритетов, автоинициализацию, задать направление изменения адреса при обмене и т. д.).**

**Загрузка 16-разрядных регистров контроллера осуществляется через 8-разрядные порты ввода-вывода. Перед загрузкой первого (младшего) байта должен быть сброшен (очищен) триггер-защелка (триггер первый/последний, First/Last flip-flop), который изменяет свое состояние после вывода в порт первого байта и таким образом дает возможность следующей командой вывода в тот же порт загрузить старший байт соответствующего регистра.**