

## АВС. Лекция № 7. Система прерывания программ.

### Вопросы:

1. Прерывание;
2. Классификация прерываний;
3. Приоритеты;
4. Работа механизма прерывания;
5. Маскирование прерываний;
6. Особенности обработки аппаратных прерываний;
7. Функциональный аналог прерываний в составе Субъекта.

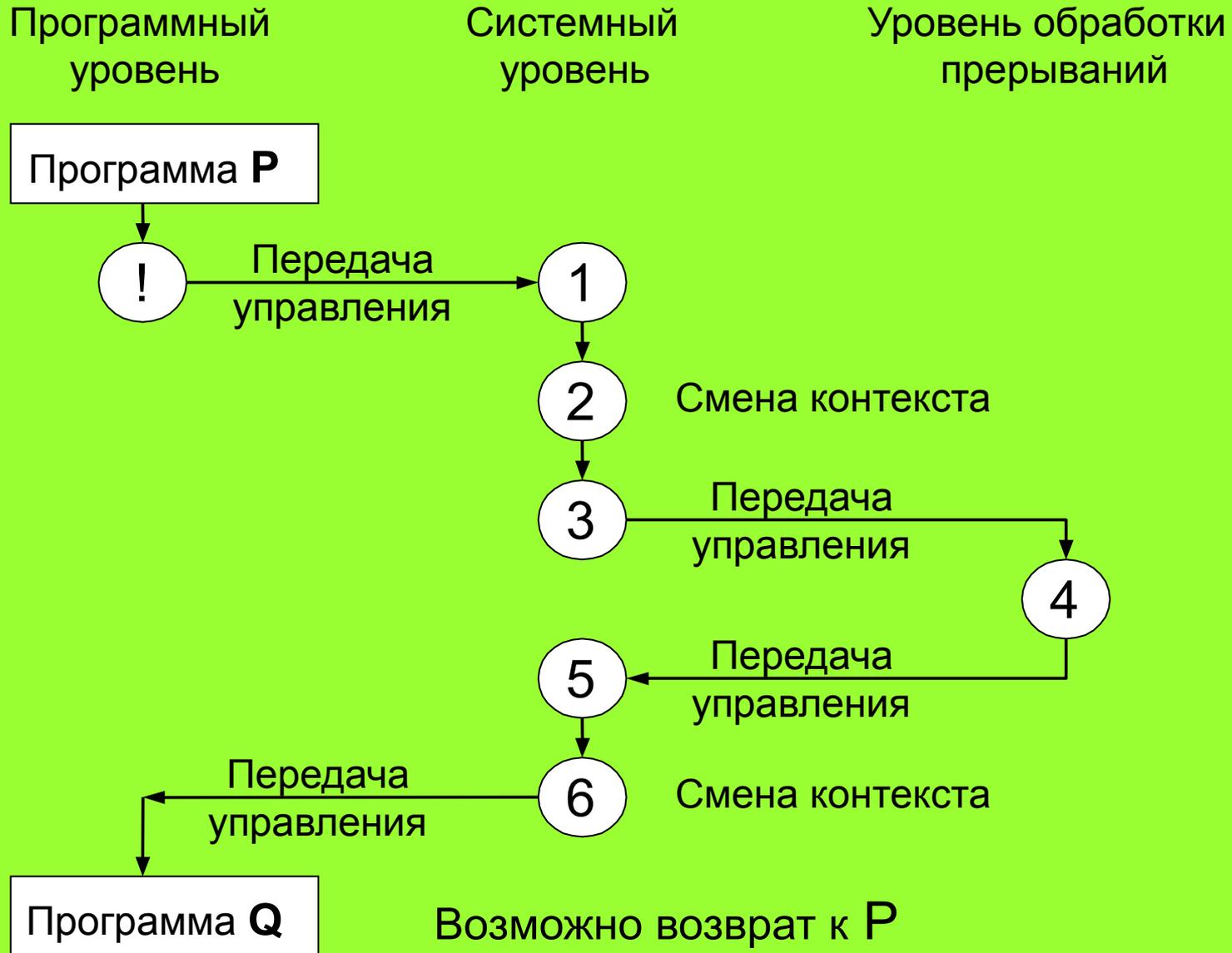
### • Главные функции механизма прерываний:

- распознавание и классификация прерываний
  - передача управления обработчику прерываний
  - корректное возвращение к прерванной программе.
- При этом переход от прерванной программы к обработчику прерываний и обратно должен выполняться как можно быстрее. Соответствующий промежуток времени **называется латентной задержкой** и является важной характеристикой вычислительной системы.

## Основные этапы реализации прерываний.

1. Передача управления осуществляется аппаратно.
2. Программой Q может быть и прерванная программа P.
3. Для корректной обработки прерывания очень важно запретить любые прерывания до окончания начатой обработки, поскольку иначе может возникнуть бесконечная последовательность вложенных прерываний. По окончании обработки прерывания необходимо восстановить прежний режим системы прерываний, т.е. разрешить их.

# Схема обработки прерываний:



## Обозначения:

- ! Прерывание (сигнал – установление факта прерывания)
- 1 Идентификация прерывания
- 2 Отключение всех других прерываний
- 3 Смена контекста\_1 (сохранение состояния прерванного процесса **P** из системных регистров, загрузка в системные регистры контекста соответствующего обработчика прерываний)
- 4 Обработка прерывания, включающая определение программы **Q**, которую следует запустить;
- 5 Смена контекста\_2 (загрузка в системные регистры контекста определенной на предыдущем шаге программы **Q**)
- 6 Установка прежнего режима системы прерываний.

Прерывания, возникающие при работе вычислительной системы, можно разделить на три класса:

- Внутренние аппаратные (синхронные);
- Внешние аппаратные (асинхронные);
- Программные.

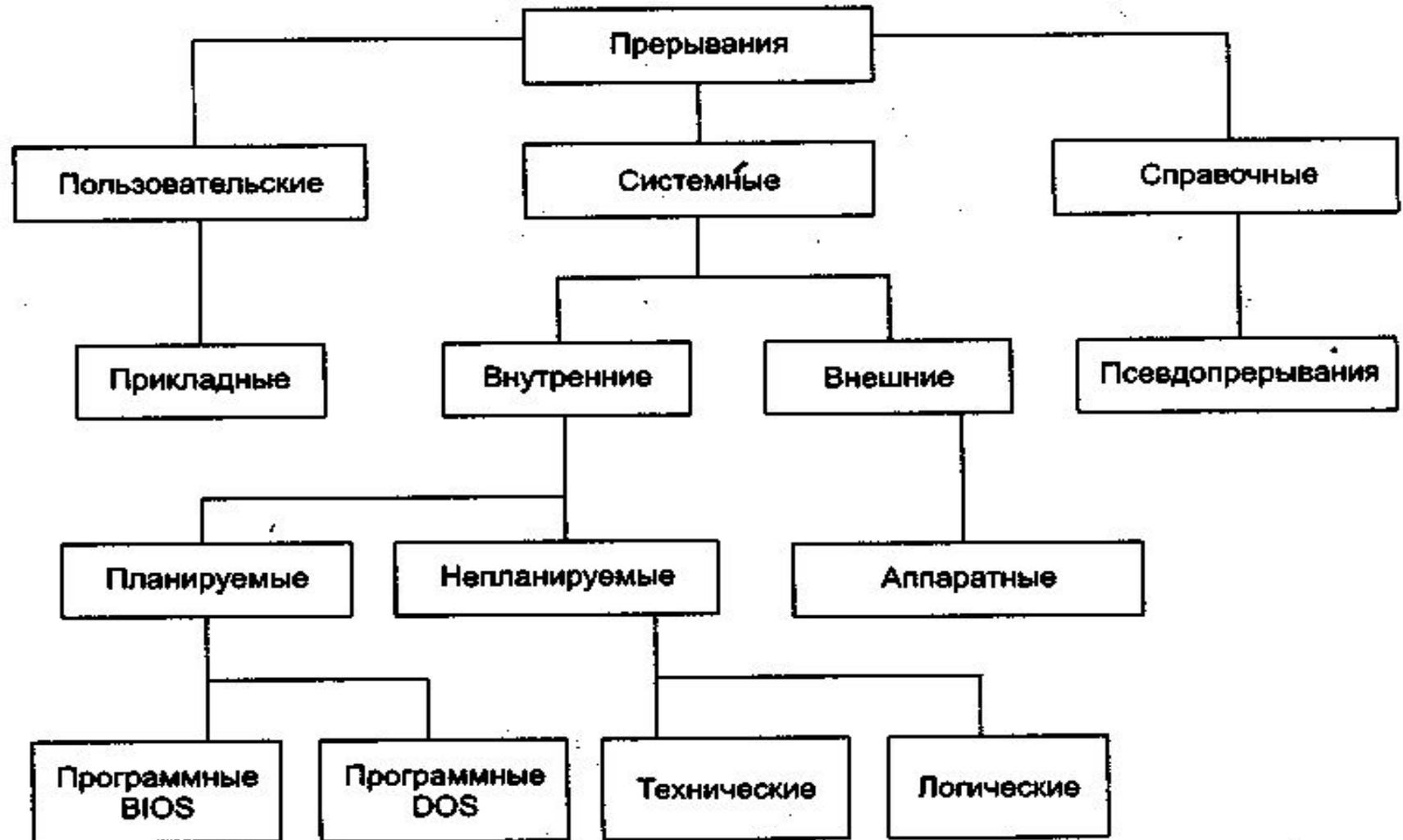
# Классификация прерываний

**Внешние аппаратные прерывания** асинхронные события, которые происходят вне процесса : 1. прерывания от таймера, 2. от внешних устройств, 3. прерывание по нарушению питания, 4. прерывание с пульта вычисления системы, 5. прерывание от другого процессора в вычислительной сети.

**Внутренние аппаратные прерывания** синхронные события, которые связаны с работой процессора и являются его синхронными операциями (сигнал прерывания формируется внутри процессора при выполнении программы, закрепленные за событиями номера прерываний защиты внутри процессора и не могут быть изменены): нарушение адресации; наличие в поле кода операции незадействованной двоичной операции; переполнение и исчезновение порядка; ошибки четности.

**Программные прерывания** (по команде внутри программы) также являются синхронными операциями. Любая программа может инициировать синхронное программное прерывание путем выполнения программы INT. MS-DOS использует для взаимодействия с программами прерывания с номерами 20H – 1FH, программы BIOS и прикладные программы используют другие номера. При этом распределение номеров условно и аппаратно не закреплено.

**Видов (номеров) прерывания может быть всего 256, и, соответственно, векторов прерывания в ОП насчитывается до 256. Более подробная классификация видов прерываний показана на рис. 17.2.**



**Рис. 17.2.** Классификация видов прерываний в МП

- *Прикладные прерывания* временно устанавливаются пользователем при многопрограммной работе МП для указания приоритета выполнения прикладных программ (при появлении необходимости выполнения более приоритетной программы текущая менее приоритетная программа прерывается).
- *Псевдопрерывания* служат для запоминания важных фиксированных адресов, которые могут быть использованы в программах, в частности, при условных и безусловных передачах управления (запоминание адресов передачи управления как векторов прерывания возможно благодаря аналогии выполнения прерывания и обращения к процедурам).

- *Аппаратные прерывания* инициируются при обращениях к МП со стороны внешних устройств (таймера, клавиатуры, дисководов, принтера и т. д.) с требованием уделить им внимание и выполнить совместно с ними те или иные процедуры. Прерывания от таймера, например, повторяются 18 раз в секунду, от клавиатуры — при каждом программно не запланированном нажатии некоторых клавиш и т. п. Аппаратные прерывания не координируются с работой программы и могут/быть весьма разнообразны. Для их систематизации и определения очередности выполнения при одновременном возникновении нескольких из них обычно используется контроллер прерываний.

- *Программные прерывания* — это обычные процедуры, которые вызывает текущая программа для выполнения предусмотренных в ней стандартных подпрограмм, чаще всего подпрограмм — служебных функций работы с внешними устройствами, то есть, фактически, программные прерывания ничего не прерывают. Они делятся на две большие группы, вызывающие служебные функции:
  - базовой системы ввода-вывода — прерывания BIOS;
  - операционной системы — прерывания DOS.
- Программы обработки прерываний DOS, в отличие от программ обработки прерываний BIOS, не встроены в ПЗУ и для разных операционных систем могут быть разными. К программным прерываниям можно отнести также прерывания при пошаговом исполнении программы, при работе с контрольным остановом и т. д.

- *Технические прерывания* (или, иначе, прерывания от схем контроля) возникают при появлении отказов и сбоев в работе технических средств (аппаратуре) ПК. Большинство технических прерываний не маскируются, то есть они разрешаются всегда, а некоторые из них относятся к категории «аварийных» (например, отключение питания), и при их возникновении даже не запрашивается причина прерывания, а просто по возможности спасаются важные промежуточные результаты — записываются в безопасное место, в НЖМД, например.
- *Логические прерывания* возникают при появлении ошибок в выполняемых программах (деление на 0, потеря точности мантиссы, нарушение защиты памяти и т. п.). Многие из логических прерываний также относятся к категории немаскируемых

# Вектора прерывания

Между запросами (и соответствующими прерывающими программами) должны быть установлены *приоритетные соотношения*. Приоритетный выбор запроса для исполнения входит в процедуру перехода к прерывающей программе.

Назовем *вектором прерывания* вектор "начального состояния прерывающей программы. Вектор прерывания содержит всю необходимую информацию для перехода к прерывающей программе, в том числе ее начальный адрес. Каждому запросу (уровню) прерывания соответствует свой вектор прерывания. Векторы прерывания обычно находятся в специально выделенных фиксированных ячейках памяти.

Для того чтобы связать адрес обработчика прерывания с номером прерывания, используется *таблица векторов прерываний*, занимающая первый килобайт оперативной памяти - адреса от 0000:0000 до 0000:03FF. Таблица состоит из 256 элементов - FAR - адресов обработчиков прерываний. В первом слове элемента таблицы записано смещение, а во втором - адрес сегмента обработчика прерывания. Прерыванию с номером 0 соответствует адрес 0000:0000, прерыванию с номером 1 - 0000:0004 и т.д.

Инициализация таблицы происходит частично BIOS после тестирования аппаратуры и перед началом загрузки операционной системой, частично при загрузке DOS.

DOS может переключить на себя некоторые прерывания BIOS.

Таблицы векторов прерывания: Назначение некоторых наиболее важных векторов:

Номер	Описание
0	Ошибка деления. Вызывается автоматически после выполнения команд DIV или IDIV, если в результате деления происходит переполнение (например, при делении на 0). DOS обычно при обработке этого прерывания выводит сообщение об ошибке и останавливает выполнение программы.
1	Прерывание пошагового режима. Вырабатывается после выполнения каждой машиной команды, если в слове флагов установлен, бит пошаговой трассировки TF. Используется для отладки программ. Это прерывание не вырабатывается после выполнения команд MOV (Переслать) и POP (Извлечение слова из стека).
2	Аппаратное немаскируемое прерывание. Это прерывание может использоваться по-разному в разных машинах. Обычно вырабатывается при ошибке четности в оперативной памяти и при запросе прерывания от сопроцессора.
3	Прерывание для трассировки. Генерируется при выполнении однобайтовой машиной команды с кодом CCh и обычно используется отладчиками для установки точки прерывания

4	Переполнение. Генерируется машинной командой INTO при флаге OF. Если флаг не установлен, то команда INTO выполняется как NOP. Это прерывание используется для обработки ошибок при выполнении арифметических операций
5	Печать копии экрана. Генерируется при нажатии на клавиатуре клавиши PrtScr. Обычно используется для печати образа экрана. Для процессора 80286 генерируется при выполнении машинной команды BOUND, если проверяемое значение вышло за пределы заданного диапазона
6	Неопределенный код операции или длина команды больше 10 байт (для процессора 80286)
7	Особый случай отсутствия математического сопроцессора (процессор 80286)
8	IRQ0 - прерывание интервального таймера, возникает 18,2 раза в секунду
9	IRQ1 - прерывание от клавиатуры. Генерируется при нажатии и при отжати клавиши. Используется для чтения данных от клавиатуры

A	IRQ2 - используется для каскадирования аппаратных прерываний в машинах класса AT
B	IRQ3 - прерывание асинхронного порта COM2. Это и следующее прерывания особенно важны для нас, т.к. они используются при работе последовательными портами (модемами)
C	IRQ4 - прерывание асинхронного порта COM1
D	IRQ5 - прерывание от контроллера жесткого диска для XT
E	IRQ6 - прерывание генерируется контроллером флоппи-диска после завершения операции
F	IRQ7 - прерывание принтера. Генерируется принтером, когда он готов к выполнению очередной операции. Многие адаптеры принтера не используют это прерывание. В некоторых случаях можно использовать это прерывание для работы с последовательными портами
10	Обслуживание видеоадаптера

11	Определение конфигурации устройств в системе
12	Определение размера оперативной памяти в системе
13	Обслуживание дисковой системы
14	Последовательный ввод - вывод
15	Расширенный сервис для AT - компьютеров
16	Обслуживание клавиатуры
17	Обслуживание принтера
18	Запуск BASIC в ПЗУ, если он есть
19	Загрузка операционной системы
1A	Обслуживание часов
1B	Обработчик прерывания Ctrl - Break
1C	Прерывание возникает 18,2 раза в секунду, вызывается программно обработчиком прерывания таймера
1D	Адрес видео таблицы для контроллера видеоадаптера 6845
1E	Указатель на таблицу параметров дискеты
1F	Указатель на графическую таблицу символов с кодами ASCII 128-255
20-5F	Используется DOS или зарезервировано для DOS

60-67	Прерывания, зарезервированные для пользователя
68-6F	Не используются
70	IRQ8 - прерывание от часов реального времени
71	IRQ9 - прерывание от контроллера EGA
72	IRQ10 - зарезервировано
73	IRQ11 - зарезервировано
74	IRQ12 - зарезервировано
75	IRQ13 - прерывание от математического сопроцессора
76	IRQ14 - прерывание от контроллера жесткого диска
77	IRQ15 - зарезервировано
78-7F	Не используются
80-85	Зарезервированы для BASIC
86-F0	Используются интерпретатором BASIC
F1-FF	Не используются
IRQ0- IRQ15	Это аппаратные прерывания, о них будет рассказано позже.

## Приоритеты

Различают *абсолютный* и *относительный приоритеты*. Запрос, имеющий абсолютный приоритет, прерывает выполняемую программу и инициирует выполнение соответствующей прерывающей, программы. Запрос с относительным приоритетом является первым кандидатом на обслуживание после завершения выполнения текущей программы.

Простейший способ установления приоритетных соотношений между запросами (уровнями) прерывания состоит в том, что приоритет определяется порядком присоединения линий сигналов запросов к входам системы прерывания. При появлении нескольких запросов прерывания первым воспринимается запрос, поступивший на вход с меньшим номером. В этом случае приоритет является жестко фиксированным. Изменить приоритетные соотношения можно лишь пересоединением линий сигналов запросов на входах системы прерывания.

# Стандартное распределение приоритетов внутри вычислительной системы

<p>Средства контроля процессора Системный таймер Внешние устройства Магнитные диски Сетевое оборудование Терминалы и мониторы Прочие периферийные устройства Программные прерывания</p>	<p><b>Высокий приоритет</b> <b>Низкий приоритет</b></p> 
---	---

Не всегда сигнал прерывания вызывает прерывание исполняющейся программы. Прерывание можно запрещать посредством отключения системы прерываний или операций маскирования отдельных прерываний.

## Маскирование прерываний

Часто при выполнении критических участков программы для того, чтобы гарантировать выполнение определённой последовательности команд целиком, приходится запрещать прерывания. Это можно сделать командой CLI. Её надо поместить в начало критической последовательности команд, а в конце расположить команду STI, разрешающую процессору воспринимать прерывания. Команда CLI запрещает только маскируемые прерывания, немаскируемые всегда обрабатываются процессором.

Если необходимо замаскировать не все прерывания, а только некоторые, например, от асинхронного адаптера, то для этого необходимо воспользоваться услугами контроллера прерываний. Подробно об этом ниже, сейчас только отметим, что выдачей в этот контроллер определённой управляющей информации можно замаскировать прерывания от отдельных устройств.

## Особенности обработки аппаратных прерываний

В отличие от программных прерываний, вызываемых запланировано самой прикладной программой, аппаратные прерывания всегда происходят **асинхронно по отношению к выполняемой программе**. Кроме того, может одновременно возникнуть несколько прерываний! Для того чтобы система знала, в какой последовательности необходимо обрабатывать запросы на прерывание, существует специальная **схема приоритетов**. Система приоритетов реализуется на микросхемах Intel 8259. Каждая микросхема обслуживает до восьми приоритетов. Для увеличения количества уровней приоритетов микросхемы объединяются (каскадируются). Приоритеты аппаратных прерываний, получаемых в результате **каскадирования двух микросхем** для машин АТ, представлены последовательностью номеров *векторов прерываний* (смотри таблицу), записанных в порядке возрастания приоритета обслуживания: 8; 9; А; 70...77; В: С: D: E: F. Из этой записи видно, что самый высокий приоритет у прерываний от интервального таймера, затем идёт прерывание от клавиатуры.

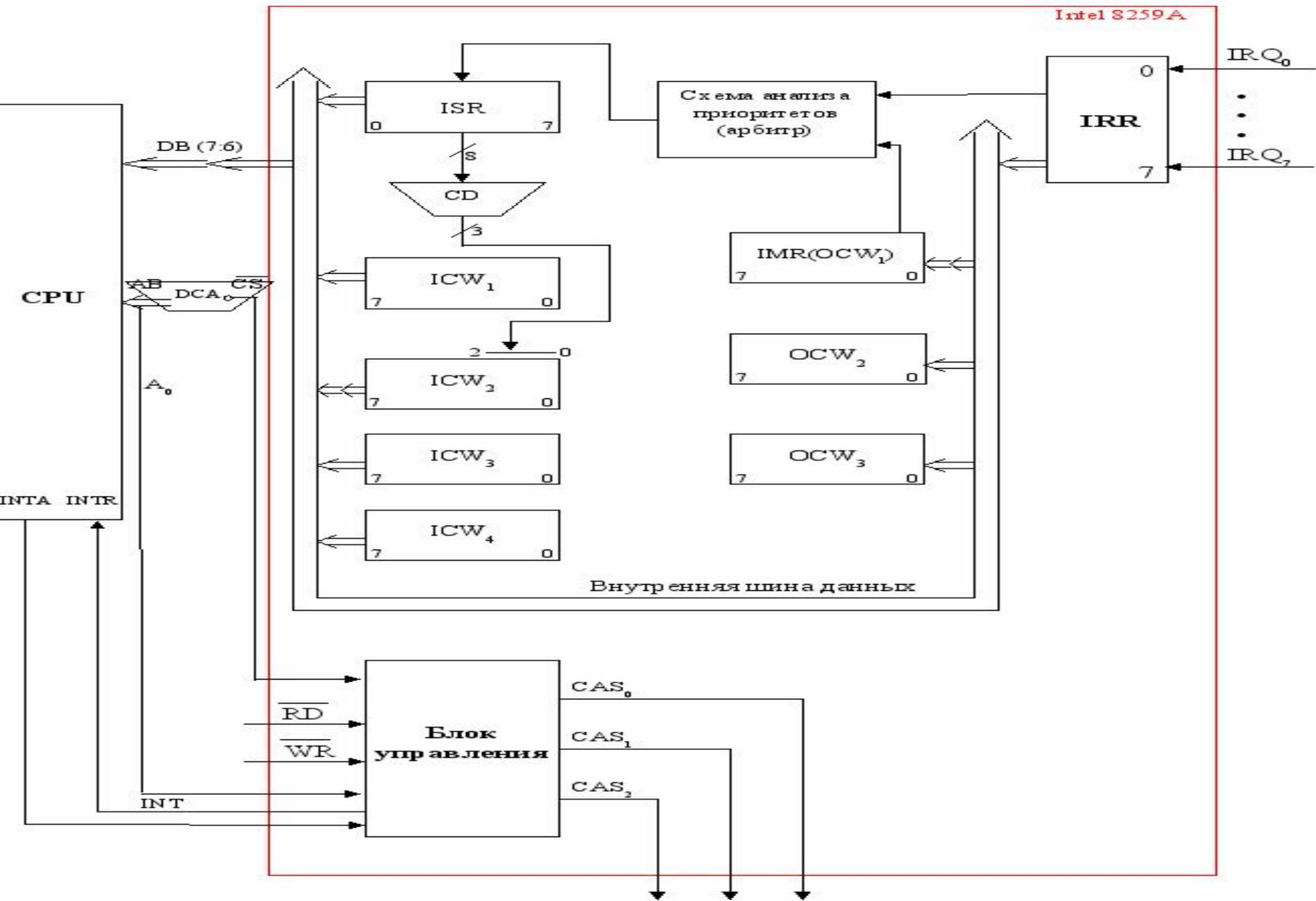
Для управления схемами приоритетов необходимо знать внутреннее устройство контроллера прерываний 8259. Поступающие прерывания запоминаются в **регистре запроса на прерывание IRR**. Каждый бит из восьми в этом регистре соответствует прерыванию. После проверки на обработку в настоящий момент другого прерывания запрашивается информация из **регистра обслуживания ISR**. Перед выдачей запроса на прерывание в процессор проверяется содержимое **восьмибитового регистра маски прерываний IMR**. Если прерывание данного уровня не замаскировано, то выдаётся запрос на прерывание.

Наиболее интересными, с точки зрения программирования контроллера прерывания, **являются регистры маски прерываний IMR и управляющий регистр прерываний**.

В машинах AT регистры маски первого и второго контроллеров имеют адреса 21h, A1h соответственно, а адреса управляющих регистров имеют значения: 20h; A0h.

Разряды регистра маски прерываний соответствуют номерам IRQ. Для того чтобы **замаскировать аппаратные прерывания какого-либо уровня, надо заслать в регистр маски байт, в котором бит, соответствующий этому уровню, установлен в 1**. Например, для маскирования прерываний от НГМД в порт 21h надо заслать двоичное число 01000000. При этом все остальные устройства будут продолжать нормально работать. В случае использования машинной команды CLI были бы отключены все аппаратные прерывания, включая клавиатуру.

# Упрощенная структурная схема ПИС



- IRR – регистр запросов прерываний - связан с внешними входами запросов ( $IRQ_0 - IRQ_7$ );
  - IMR – регистр маски запросов;
  - ISR – Interrupt Service Register – регистр обслуживаемых запросов;
  - $ICW_1 - ICW_3$  – Initialization Control Word – управляющее слово инициализации (приказы инициализации);
  - $OCW_1 - OCW_3$  - Operation Control Word – операционное управляющее слово (рабочие приказы);
- $OCW_1 = IMR$

Кроме регистров в состав PIC входят: блок управления и схема анализа приоритетов (арбитр).

Назначением блока управления является выработка внутренних и внешних сигналов управления, с помощью которых осуществляются те или иные элементарные действия (микрооперации) внутри микросхемы. Например, запись байта из внешней шины данных в один из регистров контроллера.

Сигналы  $CAS_0 - CAS_2$  используются для реализации каскадирования микросхемы.

Входной сигнал CS (Chip Select – выбор кристалла) генерируется в

**Контроллер прерываний 8259** допускает каскадирование, обеспечивающее **до 64 уровней прерываний**. Контроллер имеет несколько режимов работы, которые устанавливаются программным способом. **За первоначальную установку режимов контроллера отвечает BIOS**. Однако часто возникает необходимость изменения текущего режима работы (запрет или разрешение прерываний определённого, или всех уровней, обработка кода прерывания) или опроса состояния внутренних регистров контроллера.

Если контроллеры каскадированы, то ведомой микросхеме присваивается код (выдачей в микросхему соответствующего командного слова). Этот код равен номеру входа IRQ ведущей микросхемы, с которой связан выход запроса прерывания INT ведомой микросхемы. Внутри микросхемы приоритет зависит от номера IRQ . **Самым высоким приоритетом внутри группы, обслуживаемой каждым контроллером, обладает вход IRQ0**.

Однако, возможно программное **изменение приоритетов в рамках так называемого приоритетного кольца**. При этом дно кольца имеет самый низкий приоритет. Возможные варианты задания приоритетов приведены в **таблице**.

## Таблица кольцевых вариантов задания приоритетов

<i>Вход</i>	<i>Уровни приоритета</i>	<i>Вход</i>	<i>Уровни приоритета</i>
IRQ0	7 6 5 4 3 2 1 0	IRQ4	3 2 1 0 7 6 5 4
IRQ1	0 7 6 5 4 3 2 1	IRQ5	4 3 2 1 0 7 6 5
IRQ2	1 0 7 6 5 4 3 2	IRQ6	5 4 3 2 1 0 7 6
TRQ3	2 1 0 7 6 5 4 3	IRQ7	6 5 4 3 2 1 0 7

Наибольший приоритет у входа IRQ с обозначением 0 приоритетного кольца, наиболее низкий - с обозначением 7.

Как уже рассматривалось выше, контроллер имеет регистры: IRR - запросов прерываний; ISR - обслуживания прерываний; IMR - маски прерываний. После выработки сигнала в процессор соответствующий разряд регистра ISR устанавливается в единичное состояние, что блокирует обслуживание всех запросов с равным и более низким приоритетом. Устранить эту блокировку либо сбросом соответствующего бита в ISR, либо командой специального маскирования.

# Типы команд, посылаемых программой в контроллер.

Имеются два **типа** команд, посылаемых программой в контроллер 8259, - **инициализации и операции**. Так как инициализация производится программой BIOS, рассмотрим более подробно команды операции. Существует **три типа команд операций**:

1. Маскирование запросов прерывания;
2. Команды обработки конца прерывания;
3. Опрос регистров и специальное маскирование.

Байты команды маскирования вводятся соответственно в порты 21h, A1h для первого и второго контроллера соответственно.

Команды операций второго и третьего типа используют порты с адресами 20h, A0h.

<i>Биты байта команды</i> D7 D6 D5 D4 D3 D2 D1 D0	<i>Описание</i>
0 0 1 0 0 0 0 0	Обычный конец операции
0 1 1 0 0 B2 B1 B0	Специальный конец операции, B0...B2 - двоично-десятичный код сбрасываемого разряда в регистре ISR.
1 0 1 0 0 x x x	Циклический сдвиг уровней приоритета с обычным концом прерывания, дно приоритетного кольца устанавливается по обслуженному запросу.
1 1 1 0 0 B2 B1 B0	Циклический сдвиг уровней приоритета со специальным концом прерывания. Дно приоритетного кольца устанавливается по коду B2 ... B0.
1 0 0 0 0 x x x	Разрешение вращения уровней приоритета
0 0 0 0 0 x x x	Сброс разрешения вращения уровней приоритета.
1 1 0 0 0 B2 B1 B0	Циклический сдвиг уровней приоритета без завершения прерывания, B2...B0 - двоично-десятичный код для приоритетного кольца.

<i>Биты байта команды</i> D7 D6 D5 D4 D3 D2 D1 D0	<i>Описание</i>
0 0 0 0 1 1 x x	Установка режима опроса
0 0 0 0 1 0 1 1	Разрешение чтения регистра обслуживания ISR.
0 0 0 0 1 0 1 0	Разрешение чтения регистра маски IRR.
0 1 1 0 1 0 0 0	Разрешение триггера специального маскирования.
0 1 0 0 1 0 0 0	Сброс триггера специального маскирования.

**По команде обычного конца прерывания устанавливается в нулевое состояние разряд ISR, соответствующий последнему обслуженному запросу.**

Команда **специального** конца прерывания устанавливает в нулевое состояние тот разряд ISR, **номер** которого указан **в разрядах** B0...B2 команды.

Команда циклического сдвига уровней приоритета с обычным концом прерывания устанавливает в ноль разряд ISR, соответствующий последнему обслуженному запросу и ему же присваивается низший уровень приоритета.

Аналогично работает команда циклического сдвига уровней приоритета со специальным концом прерывания, только **низший** уровень **приоритета** присваивается тому входу IRQ, номер которого **указан** в разрядах B0...B2 команды.

Команда циклического сдвига уровней приоритета устанавливает статус уровней приоритета без выполнения операции конца прерывания. Разряды B0...B2 указывают дно приоритетного кольца.

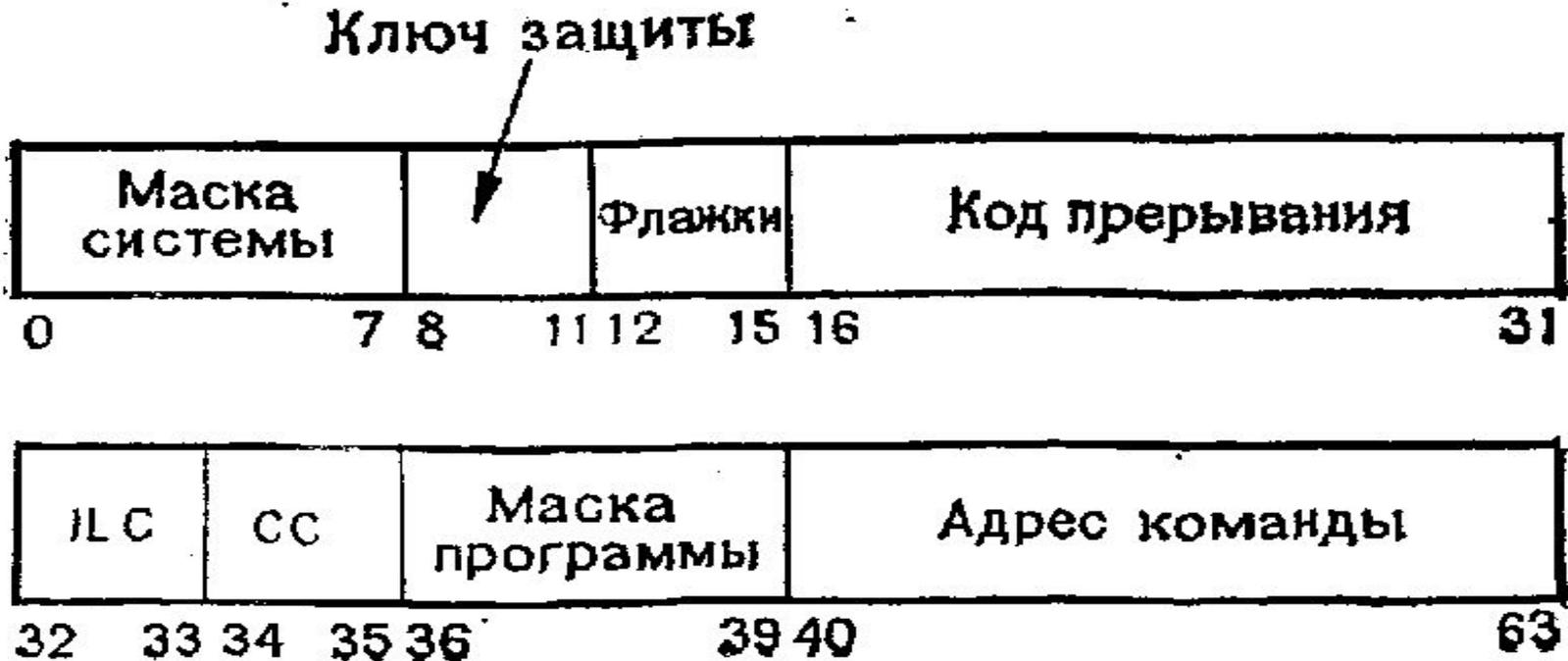
После выполнения команд разрешения чтения регистров ISR или IRR при выполнении команды ввода из порта 20h и A0h.

считывается соответственно содержимое регистров ISR и IRR. Для получения содержимого регистра IMR необходимо выполнить чтение портов с адресами соответственно 21h и A1h.

**Команда разрешения триггера специального маскирования блокирует действие тех разрядов ISR, которые замаскированы командой типа 1** (маскирования индивидуальных приоритетных уровней запроса прерывания).

Чтение регистров ISR и IRR может использоваться **резидентными программами** при проверке возможности своей активизации - можно проверить, не выполняется ли в настоящий момент обработка какого-нибудь прерывания, которая может конфликтовать с действиями резидентной программы.

# Механизм прерываний IBM 370

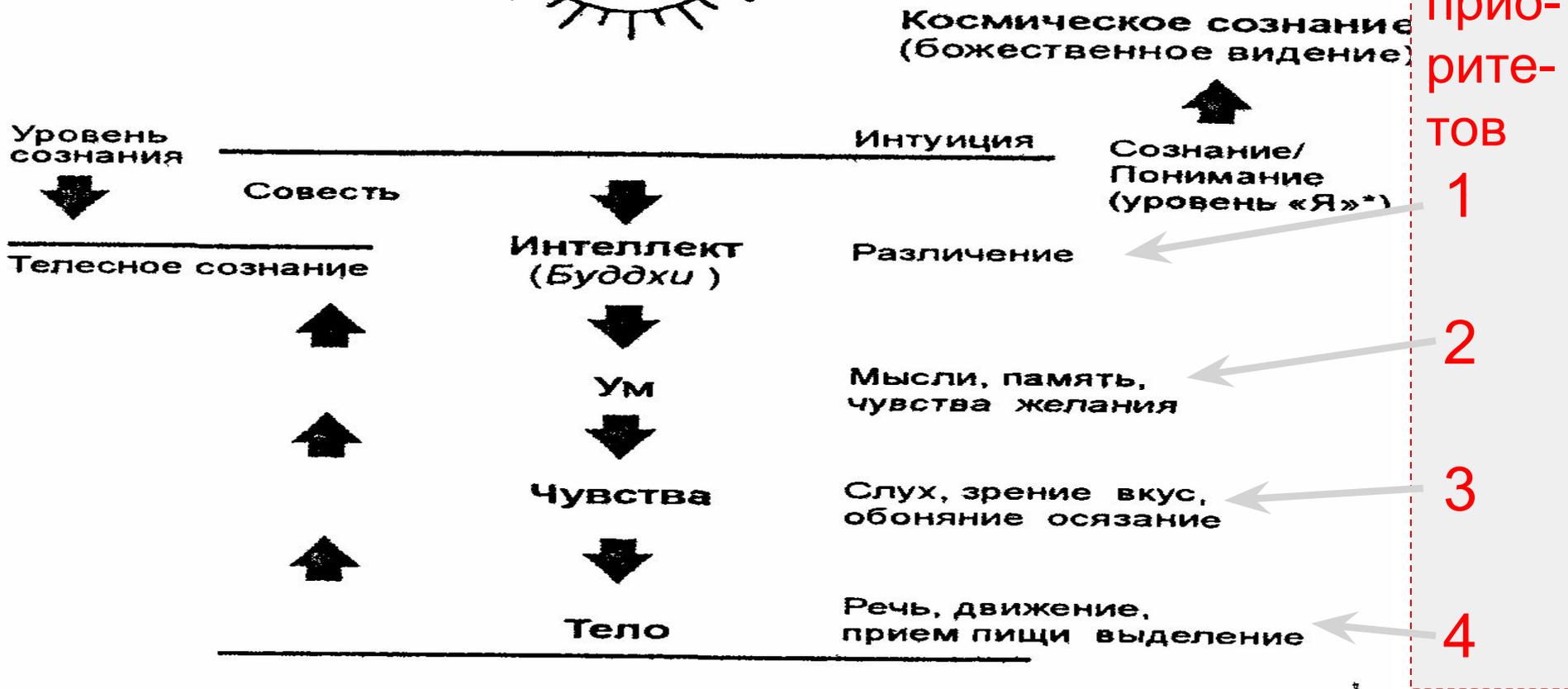


# Обработка прерываний



# Модель 1

## Пять уровней человеческого существа

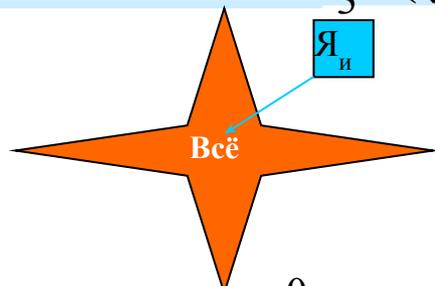


Интуиция происходит от высшей совести, находящейся за пределами личного «я» (*сатья*, или истина, проявляющаяся как интуиция). Склонности (тенденции) рождаются в низшей совести и находятся на уровне ума

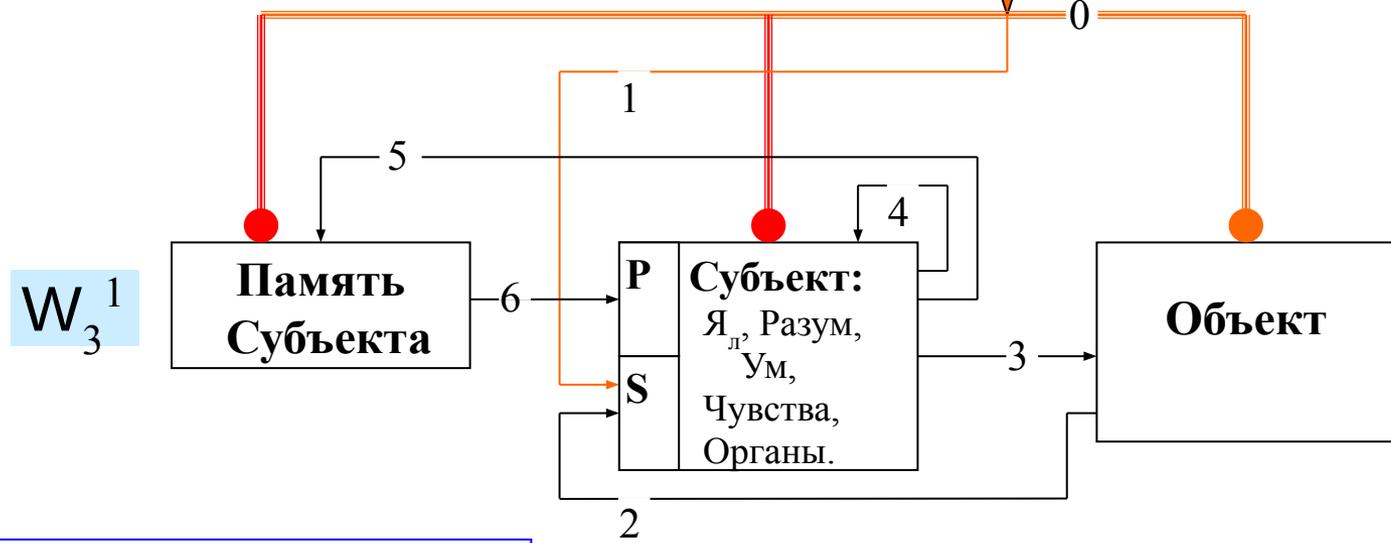
\*Чувство «Я» пронизывает всю совесть

$$D_3 = (((W_0 / W_1) / W_2^1) / W_3^1);$$

$((W_0 / W_1) / W_2^1)$   
(РЕАЛЬНОСТЬ)



**P** – Потребность  
**S** – Состояние  
**Я<sub>и</sub>** – истинное  
**Я<sub>л</sub>** – ложное (Эго)



**Отождествление себя**

000	Я – Ничто
001	Я – Объект
010	Я – Субъект
011	Я – Субъект & Объект
100	Я – Память
101	Я – Объект & Память
110	Я – Память & Субъект
111	Я – Всё

**Связи отождествления: Я – Всё (1,1)**

№	Содержание связей
0	Проявление / Растворение
1	Восприятие Субъектом Основы
2	Восприятие Субъектом Объекта / Воздействие Объекта на Субъект
3	Воздействие Субъекта на Объект / Восприятие Объектом Субъекта
4	Воздействие Субъекта на себя
5	Воздействие Субъекта на Память / Восприятие Памятью Субъекта
6	Воздействие Памяти на Субъекта / Обращение Субъекта к Памяти

# Таблица аналогий

	Система контроля чувств Человека		Система прерывания программ ЭВМ	
1	Интеллект	Различение	Диспетчер	Приоритеты обслуживания
2	Ум	Память, мысли, чувства, желания	Обработчик прерываний	Хранение запросов. Таблица векторов прерываний
3	Чувства	Слух, зрение, вкус, обоняние, осязание	Источники прерываний	Внешние и внутренние аппаратные прерывания. Программные прерывания
4	Физическое тело	Речь, движение, приём пищи, выделение	Физическая реализация	Контроллер прерываний. Программные ср.

# Приоритеты при целостном подходе к обучению.

Приоритет.	Описание приоритетов в обучении.
1	<b>Теоретическое и практическое познание своего Я как базовой сущности всего сущего.</b>
2	<b>Теоретическое и практическое познание организационной структуры ролевой игры как базовой сущности всего общества.</b>
3	<b>Теоретическое и практическое познание Информационных Технологий (АВС) как творения человека и общества, необходимого для познания единой основы всего сущего.</b>

# Свобода выбора

- Человек – сочетание двух видов особенностей. Один – животная природа. Другой – Божественная.
- Поддаваясь влиянию дурных качеств, таких как вожделение, гнев, жадность, обман, гордыня и эгоизм  
(шесть врагов человека – следствие отождествления себя с отдельным), человек принижает себя до уровня животного.
- С другой стороны, тот, кто правильно использует хорошие качества, данные Богом, такие как ум, мудрость и добродетель, следует по пути истины и праведности, посвящает себя святым делам, тот, в конце концов, достигает Божественного.
- *Человек может выбирать между путём, ведущим к животному состоянию, и путём, ведущим к Божественности, так как обе способности присутствуют в нём в равной мере.*
- Важен выбор, который он делает.

# Архитектура сознания

(Четыре типа устремлений в процессе обучения, или четыре варианта самосознания себя )

№	Тип архитектуры сознания	Содержание архитектуры сознания
1	Многообразиие в Многообразии (М в М)	Анархия – каждый сам по себе (Отсутствие науки)
2	Многообразиие в Единстве (М в Е)	Система – объединение элементов в систему (Специальные науки)
3	Единство в Многообразии (Е в М)	Принцип – всеобщие законы мира (Фундаментальные науки)
4	Единство в Единстве (Е в Е)	Абсолют – источник всего сущего (За пределами двойственности).

