

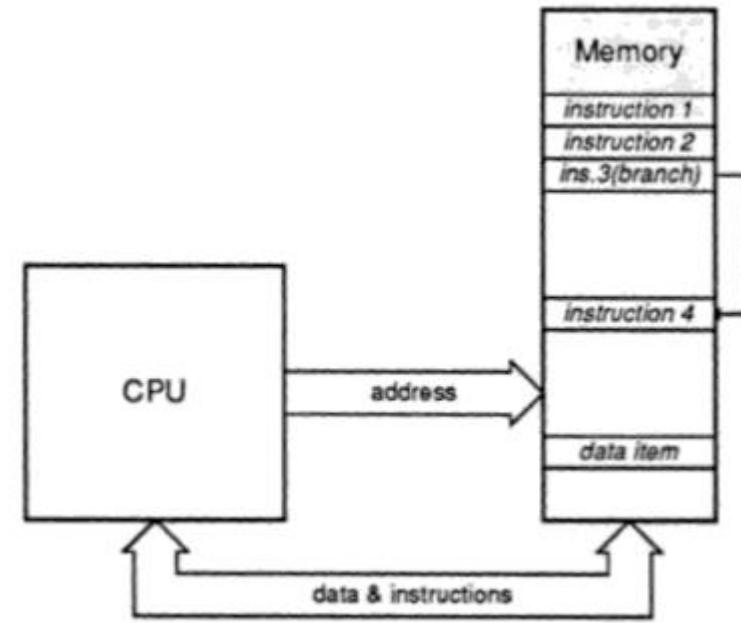
# ОРГАНИЗАЦИЯ АРХИТЕКТУРЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

# Классификация по способам расположения команд и данных

- В 1939 – 1946 по заказу Министерства обороны США двумя университетами (Гарвардским и Принстонским) были предложены две классические архитектуры компьютеров
  - Фон – Неймана
  - Гарвардская

# Архитектура Фон – Неймана

## Классические архитектуры



1. Принцип программного управления.

Компьютер работает под управлением программы

2. Общая память команд и данных

3. Принцип адресуемости памяти (процессору доступна любая ячейка памяти)

4. Вся информация кодируется в двоичной системе

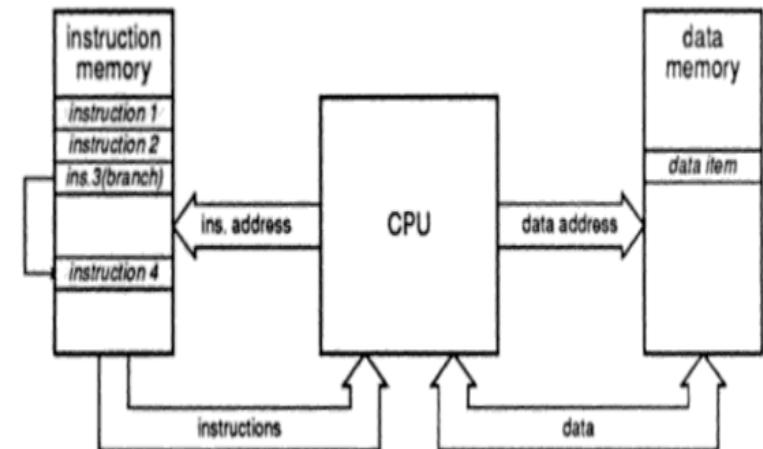
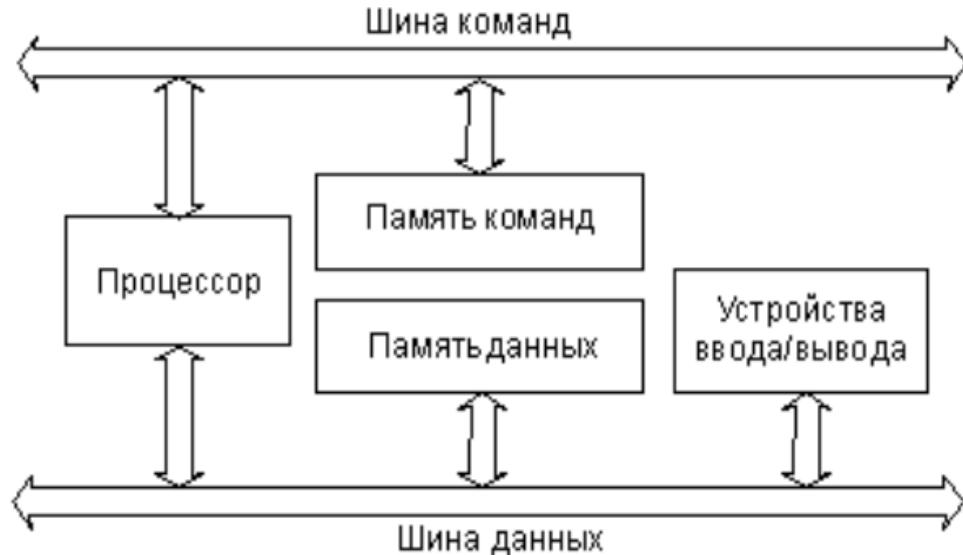
# Достоинства архитектуры Фон-Неймана.

- Более простая аппаратная реализация.
- Использование единой области памяти позволяет оперативно перераспределять ресурсы между областями программ и данных, что повышает гибкость программирования.
- Размещение стека в общей памяти облегчает доступ к его содержимому.
-

# Проблемы общей памяти

- Нельзя выбирать команды и данные одновременно
- Увеличение длины команд за счет увеличения адреса операнда.
- Увеличение времени поиска данных за счет длинного адреса

# Гарвардская архитектура

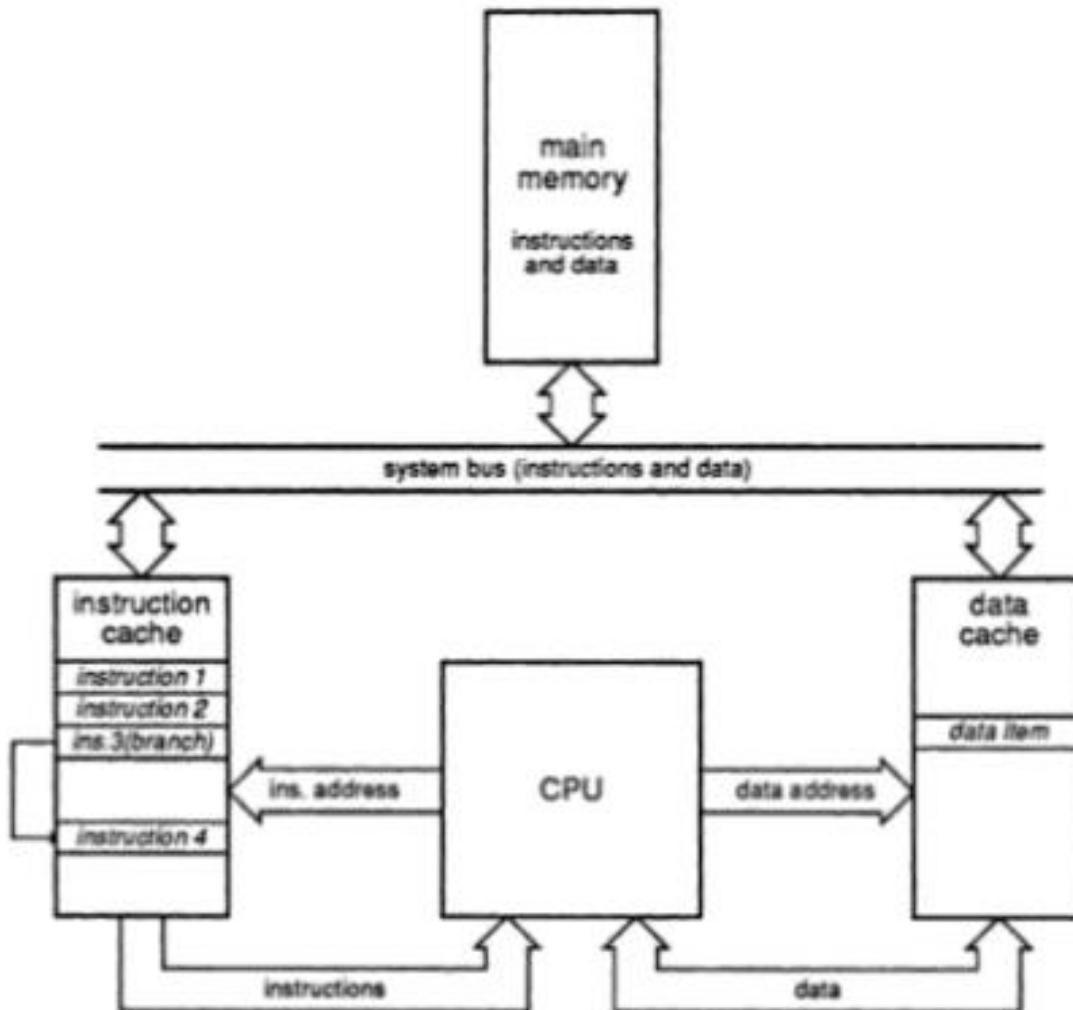


1. Коды команд и данные хранятся на разных устройствах памяти.
2. Канал чтения команд из памяти и канал чтения /записи данных физически разделены и могут работать параллельно, что позволяет проще организовать конвейер
3. Проще реализовать в пределах **одного** чипа.

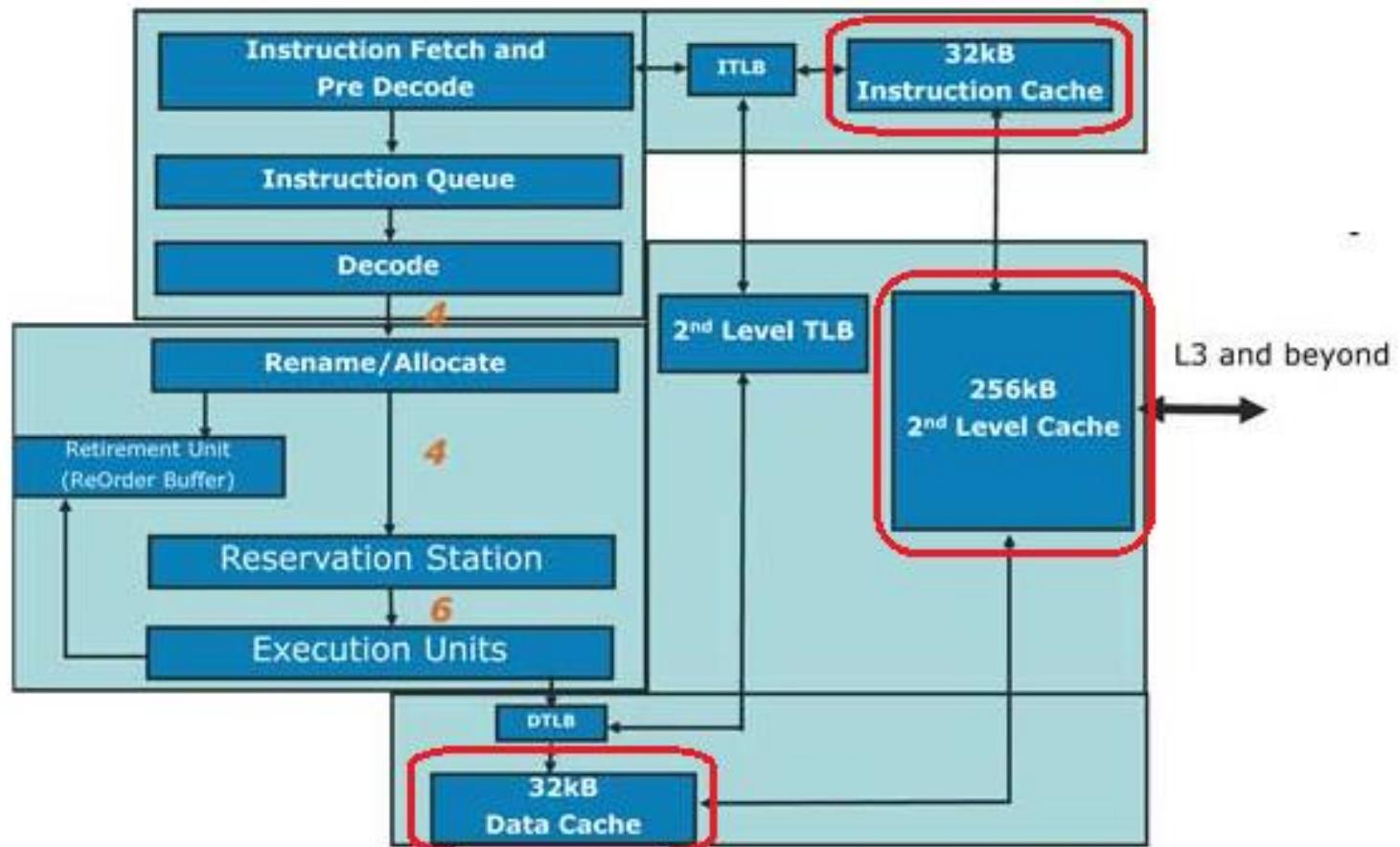
*Примеры :*

- в процессорах - два кэша: *данных и команд*;
- *системы на кристалле*.

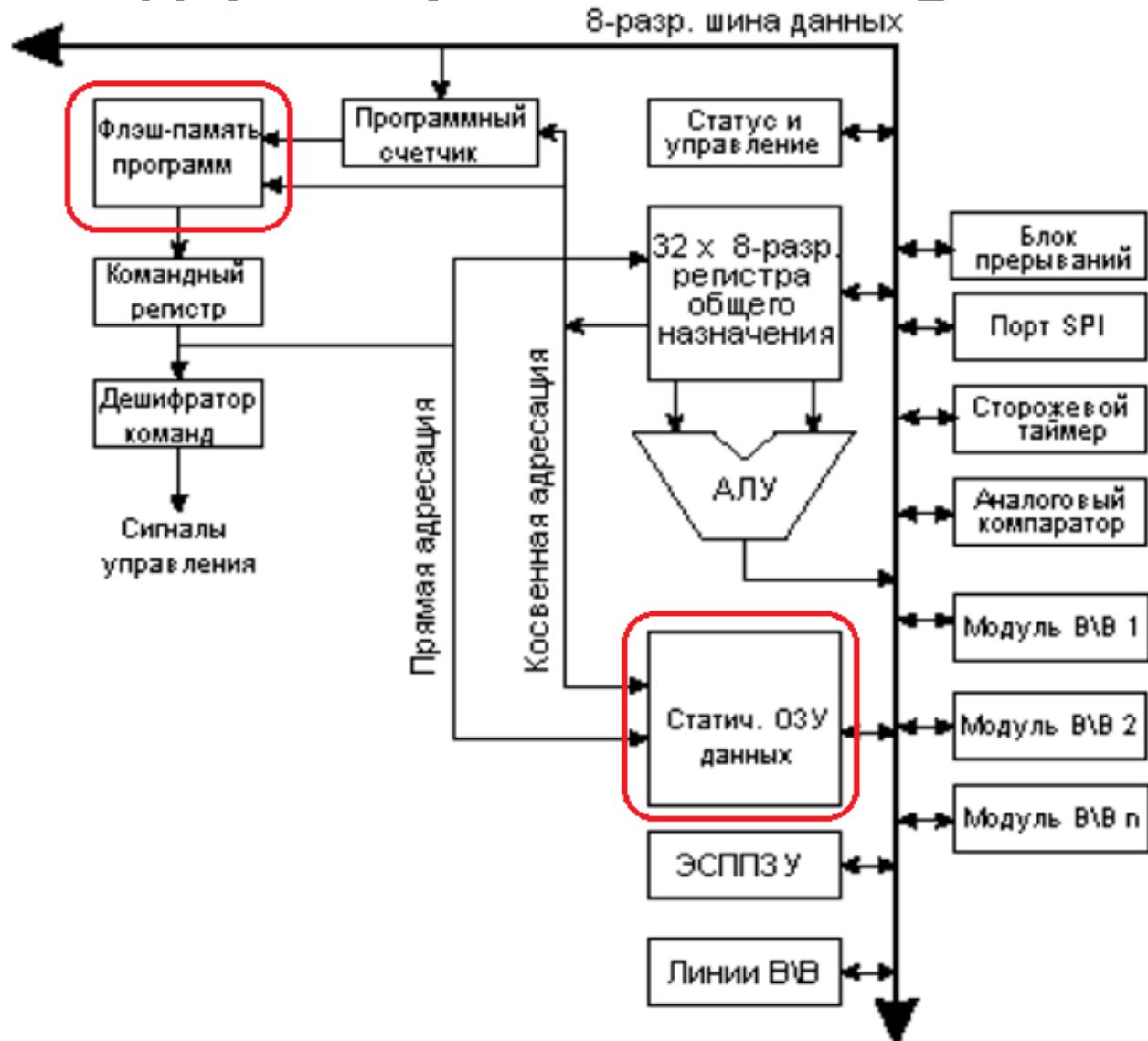
# Гарвардская модернизированная



# Гарвардская модернизированная Intel Nehalem



# Гарвардская модернизированная Atmega 8



Укрупненная функциональная схема микроконтроллера  
Atmega 8

# Архитектура компьютера ( по Фон-Нейману)



Архитектура компьютера показывает как взаимодействуют между собой отдельные функциональны устройства.

Системная шина связывает Процессор, Память, Контроллеры.

# Контроллеры

- Контроллеры можно разделить на :
- Контроллеры ввода вывода
  - Служат для связи процессора с определенным устройством ввода-вывода (периферийными устройствами) и управлени этим устройством.
- Системные контроллеры
  - Служат для управления компьютерной системой, располагаются на материнской плате.

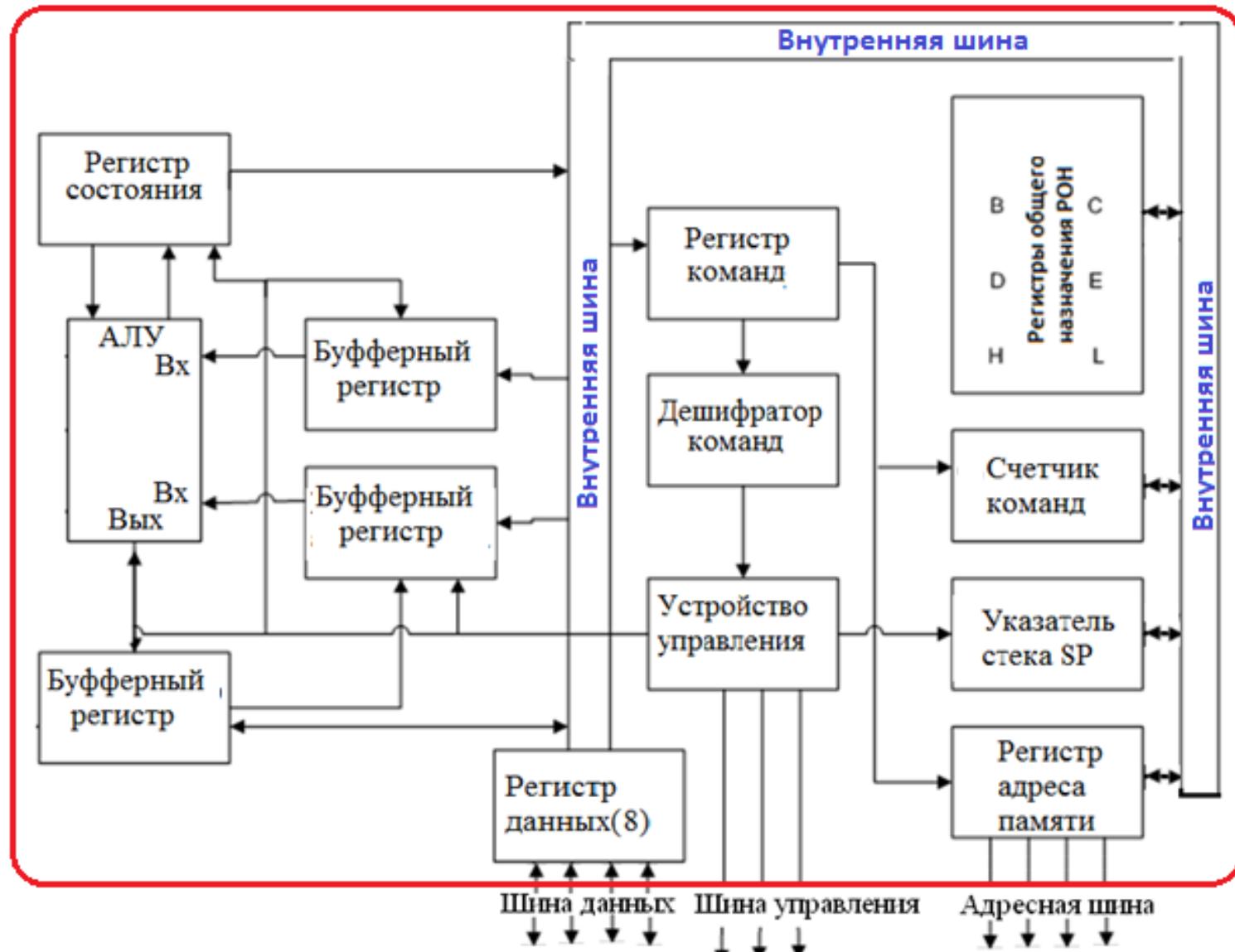
# Процессор

Архитектура процессора состоит из :

- **архитектуры набора команд** (Instruction Set Architecture, ISA);
  - как процессор виден программисту
    - количество и формат команд
    - количество доступных регистров
    - способы адресации
    - способ организации данных и.т.д.
- **микроархитектуры процессора**
  - его внутренняя аппаратная реализация
    - длина конвейера
    - предсказание переходов
    - декодер команд
    - количество одновременно выполняемых инструкций
    - доступ к памяти и.т.д.

# Микропроцессор - обобщенная микроархитектура

## Микропроцессор



# Работа процессора



- В счетчик команд заносится адрес ячейки памяти, содержащий  **первую команду** программы.
- Адрес поступает в регистр адреса памяти и храниться там на время обращения к памяти

# Работа процессора



- Адрес команды из регистра адреса поступает на шину адреса памяти.
- По сигналу из устройства управления процессор считывает команду, находящуюся по этому адресу на шину данных и записывает её в регистр данных процессора.

# Работа процессора



- Из регистра данных текущая команда поступает в регистр команды. Там команда дешифруется и анализируется устройством управления.
- В зависимости от кода операции Устройство управления вырабатывает управляющие сигналы для управления внутренними элементами процессора.

# Работа процессора



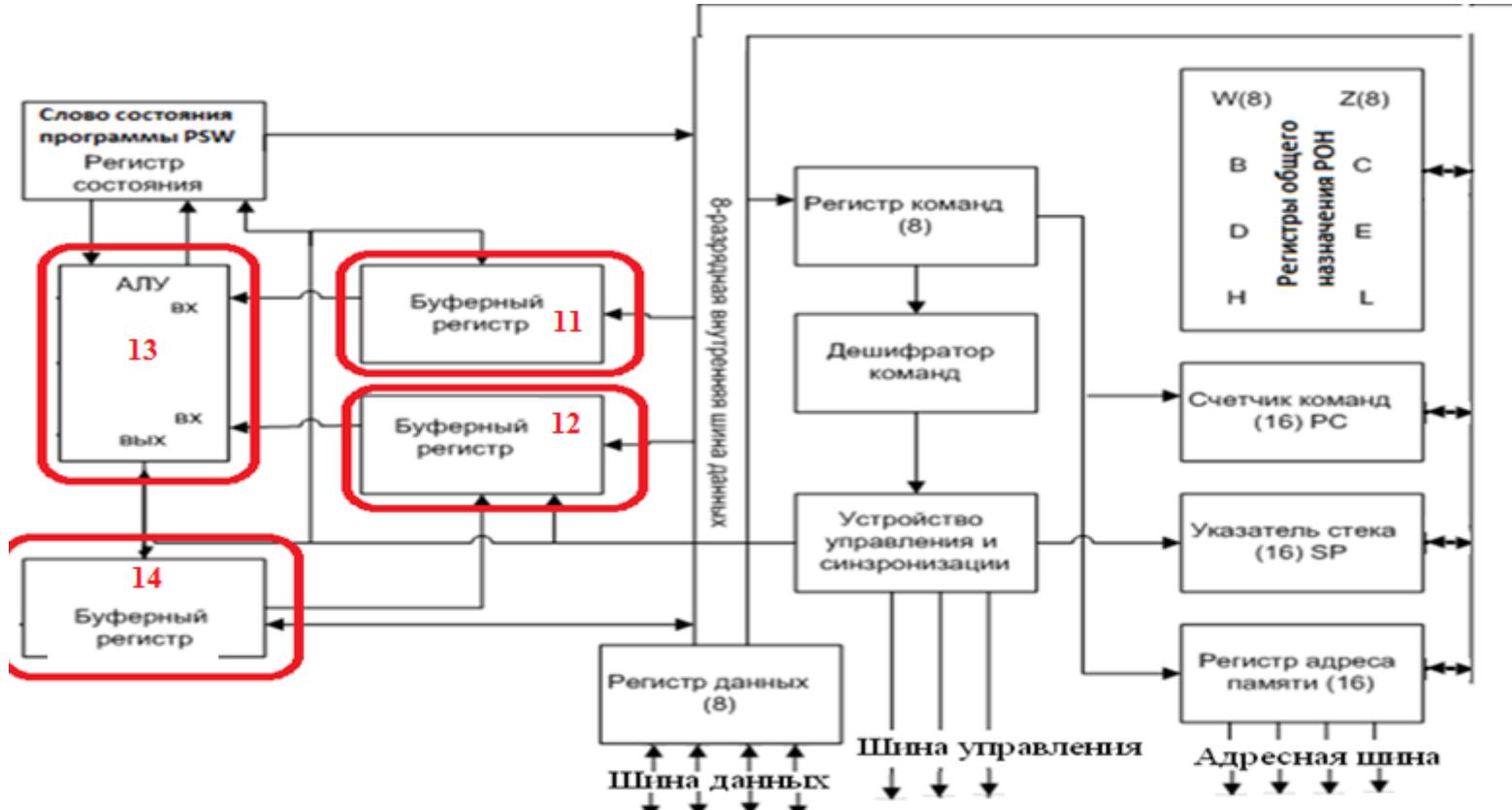
- Если машинная команда использует данные из памяти, УУ определяет их адреса и вырабатывает сигналы для чтения данных из памяти во внутренние регистры.

# Работа процессора



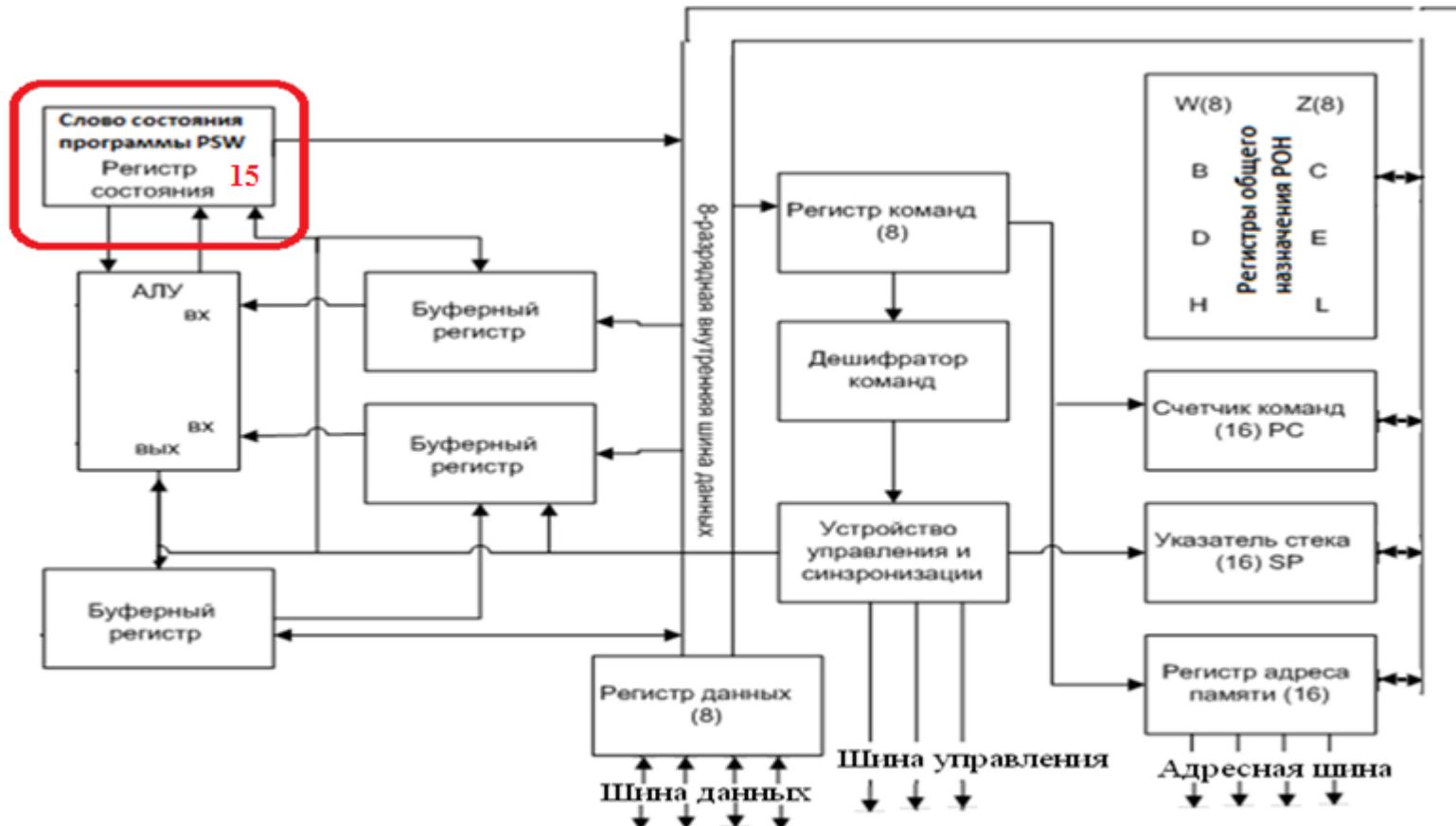
- Содержимое счетчика команд во время выполнения текущей команды увеличивается на длину команды и в нем уже содержится адрес следующей команды.

# Работа процессора



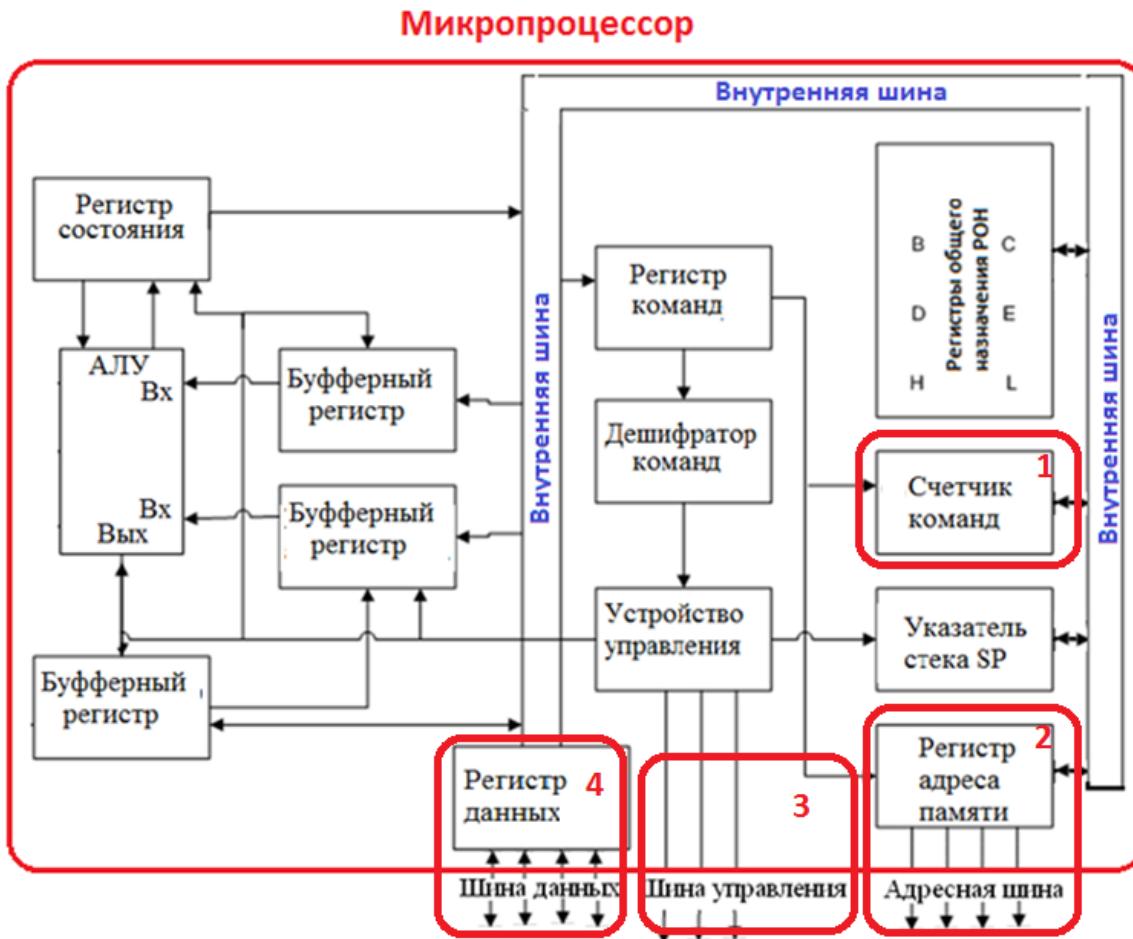
- АЛУ выполняет операцию над данными, находящимися в регистрах и сохраняет результат в регистре (аккумуляторе).

# Работа процессора



- В регистре слова состояния программы устанавливаются признаки выполнения команды

# Работа процессора



- Считывается следующая команда и цикл повторяется.

**ШИНА**

# Шина или аппаратный интерфейс

- Набор аппаратных средств и правила обмена (протоколов), обеспечивающих взаимосвязь устройств между собой
- Физически шина состоит из отдельных линий.
- Линия шины – это электрический проводник по которому распространяется электрический сигнал.
- Одна шина может состоять из нескольких подшин, каждая из которых соответствует своему функциональному назначению (шина адреса, шина данных, шина управления и.т.д.)

# Классификация интерфейсов

По способу передачи данных.

## Параллельные

- Одновременная передача всех разрядов (байта, слова и др.) по линиям интерфейса.
- Имеет столько линий сколько одновременно передается разрядов.

## Последовательные (пакетные)

- Последовательная передача бит данных в виде пакетов.
- Для передачи данных требуется минимум две физические линии.

## Последовательно - параллельные

- Параллельная передача пакетов по нескольким последовательным шинам.

# Классификация интерфейсов

## По направлению обмена:

- Симплексные – только передача или прием в одну сторону
  - Теле и радио вещание.
- Полудуплексные – передача или прием в обе стороны, но в разные моменты времени, по одной и той же линии.
  - Радиосвязь
- Дуплексные – передача и прием в обе стороны одновременно. Требуются отдельные линии для передачи и приема в каждую сторону.

# Классификация интерфейсов

По способу реализации:

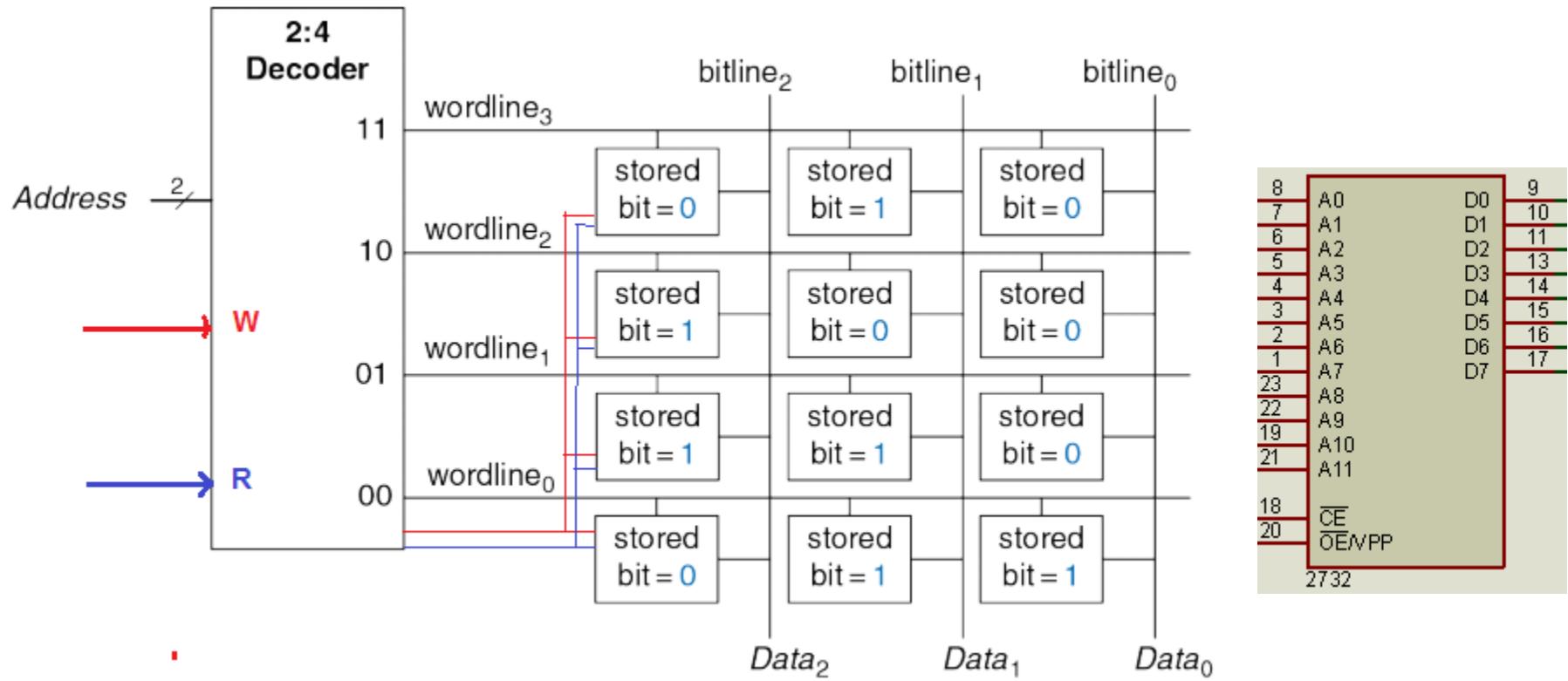
Внутренние - для связи электронных модулей внутри материнской платы:

Внешние (интерфесы ПУ) – для подключения внешних устройств ввода вывода и хранения

Память

# Байт ориентированная память

- Матрица памяти (прямоугольная)



- Ячейки выбираются построчно

# КОНТРОЛЛЕРЫ

# Контроллеры ввода вывода

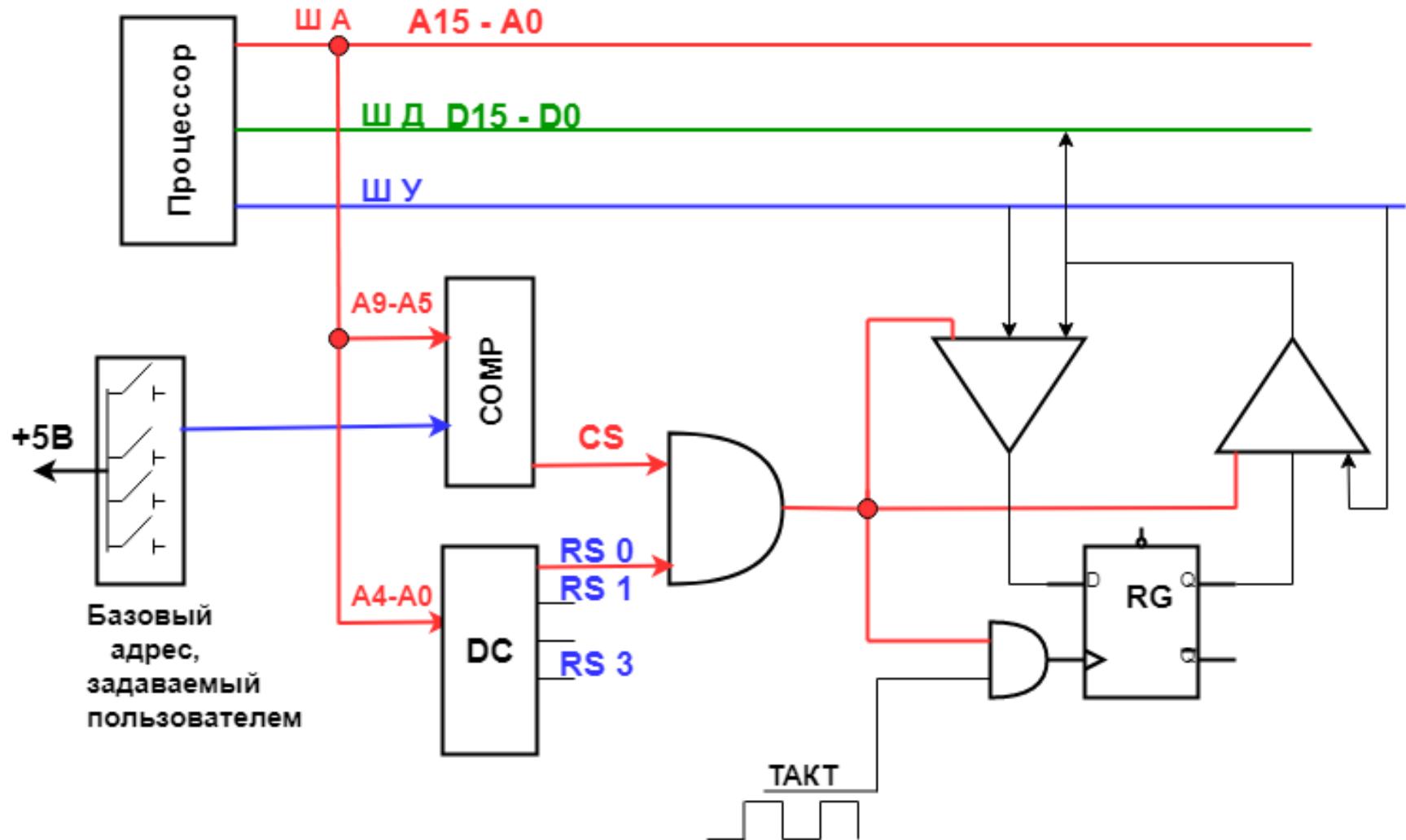
- КВВ - промежуточное устройство, которое преобразует сигналы шины ввода вывода процессора (системной шины) в сигналы интерфейса периферийного устройства и управляет работой ПУ.



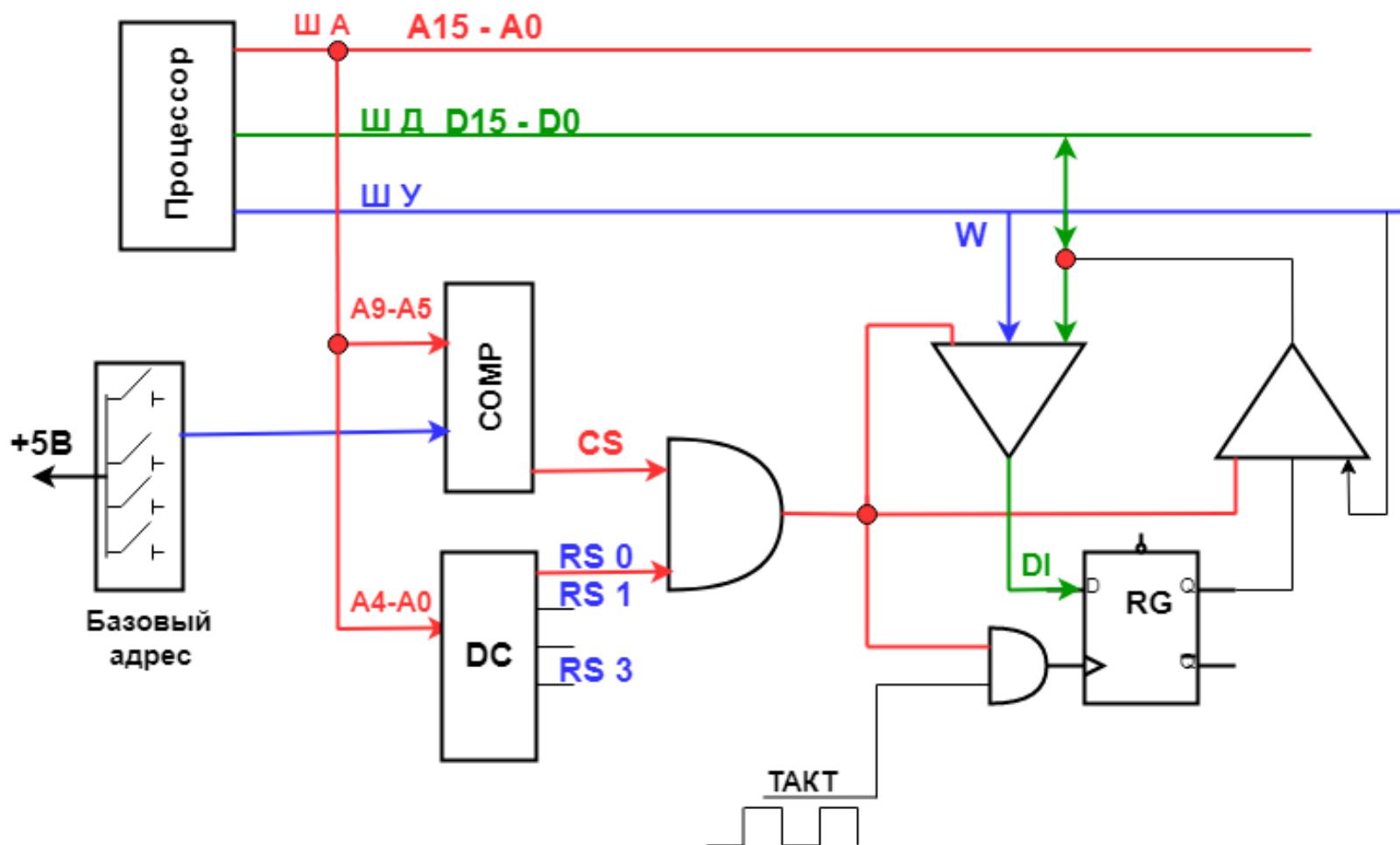
# Контроллер ввода-вывода

- Содержит, как правило, **группу регистров**, называемых **ПОРТАМИ** ввода-вывода.
- Типичный состав программно - доступных регистров/портов:
  - Регистр состояния – хранит текущее состояние контроллера В/В (готов/не готов);
  - Регистр команд – хранит команды управления контроллером В/В
  - Регистр данных – для передачи или приема данных;
  - Регистр адреса – хранит базовый адрес заданный контроллеру
    - Каждый регистр имеет уникальный адрес нашине ввода вывода процессора, по которому процессор может к нему обращаться.
    - Адрес регистра состоит из суммы базового адреса назначенного контроллеру ввода вывода + смещение, которое определяет номер регистра в контроллере (30h, 31h, 32h);
    - На первых моделях задавался перемычками на плате контроллера
- Буфер памяти
  - Временно хранит данные, принимаемые(передаваемые) периферийному устройству.
  - Служит для выравнивания скоростей работы процессора и ПУ

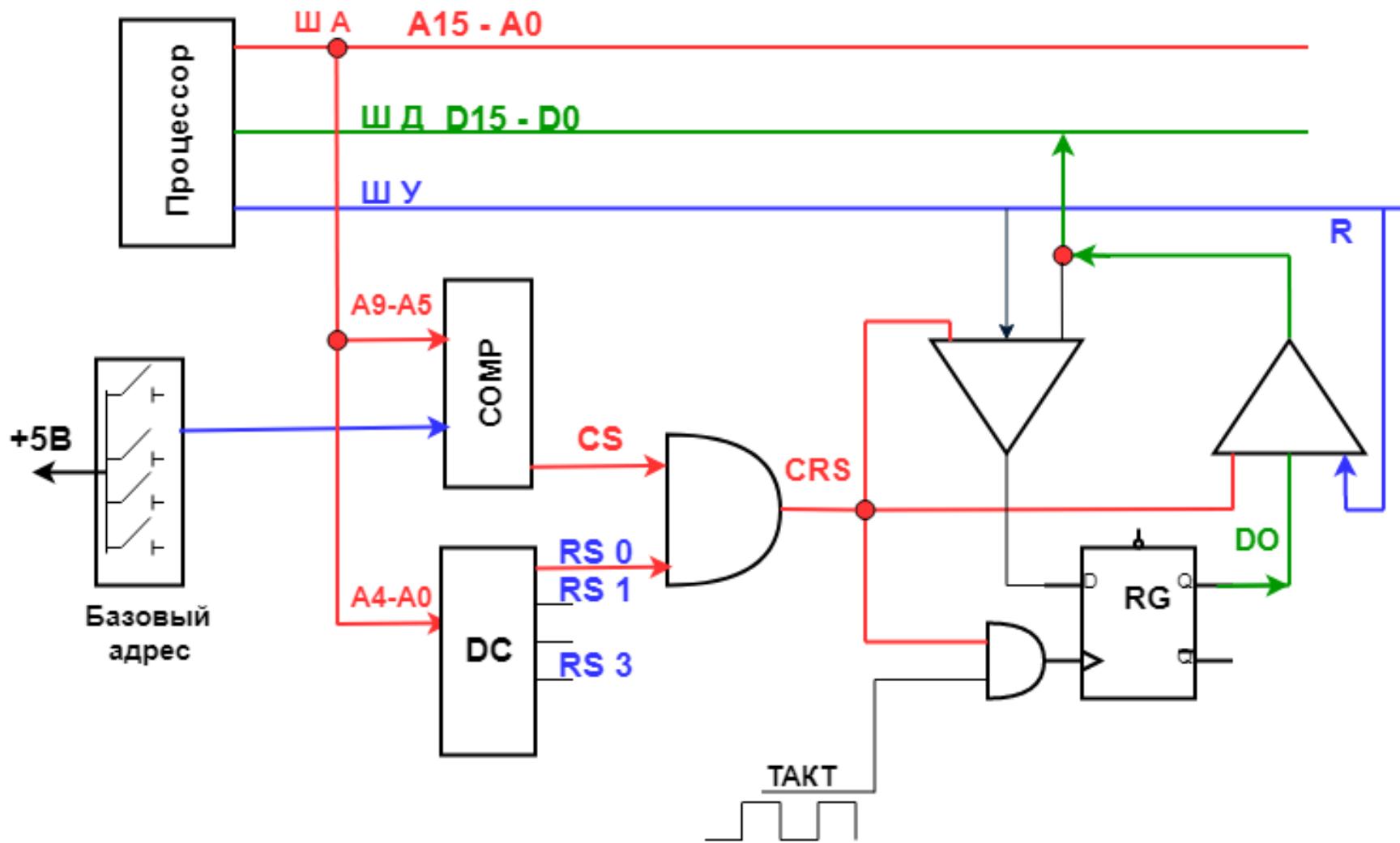
# Подключение контроллера



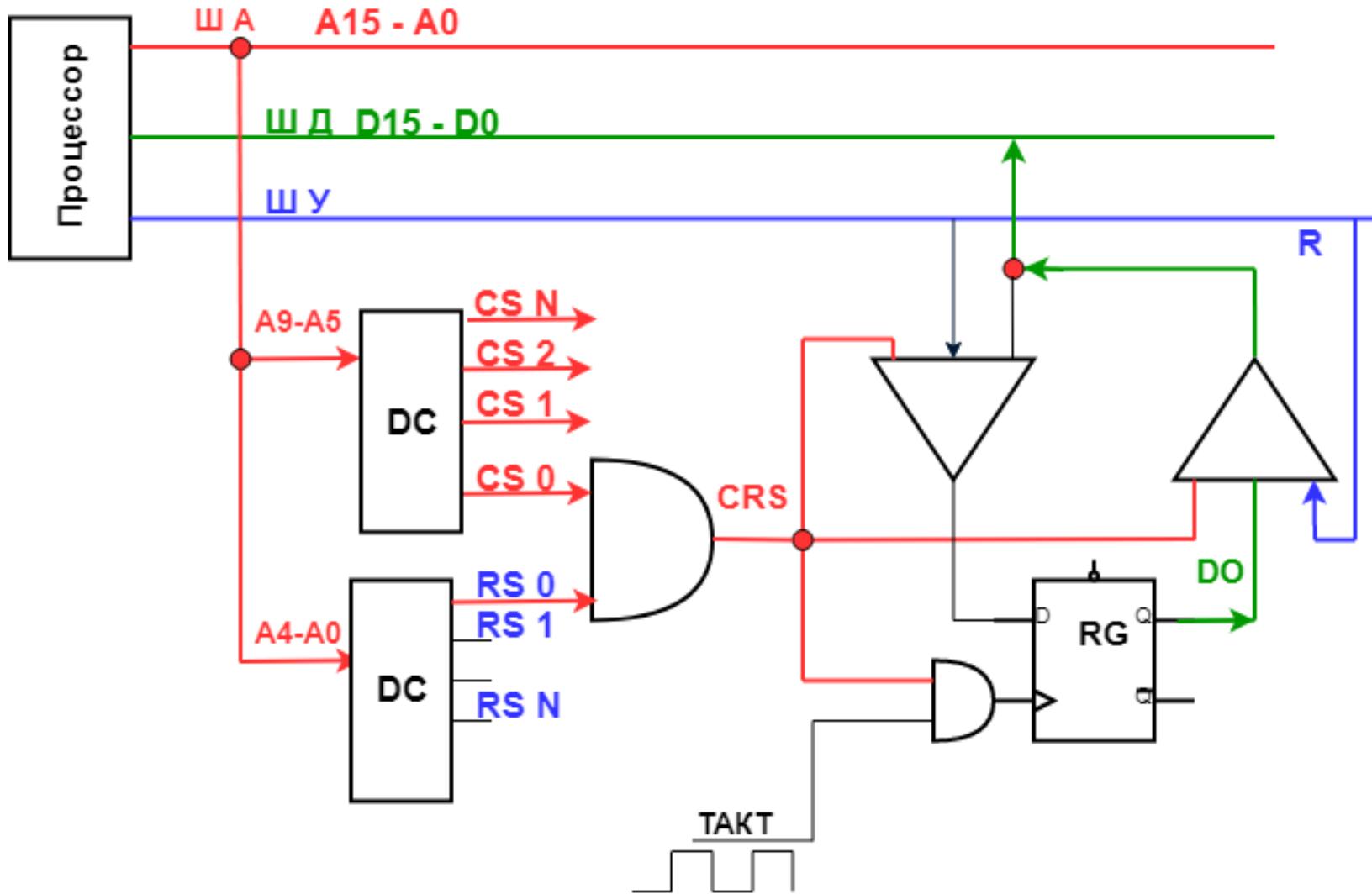
# Запись в регистр контроллера ввода-вывода



# Чтение регистра контроллера ввода-вывода



# Чтение регистра системного контроллера



Kommunikation



DE IN CHINA

R36

301



c5dx

HSP56

CMI8738/PCI-SX  
HRTF Audio Com  
M6124-037D  
0128UYT1EA

U1

C20 C24

C25 C

C12

C56

C26

R34

R35

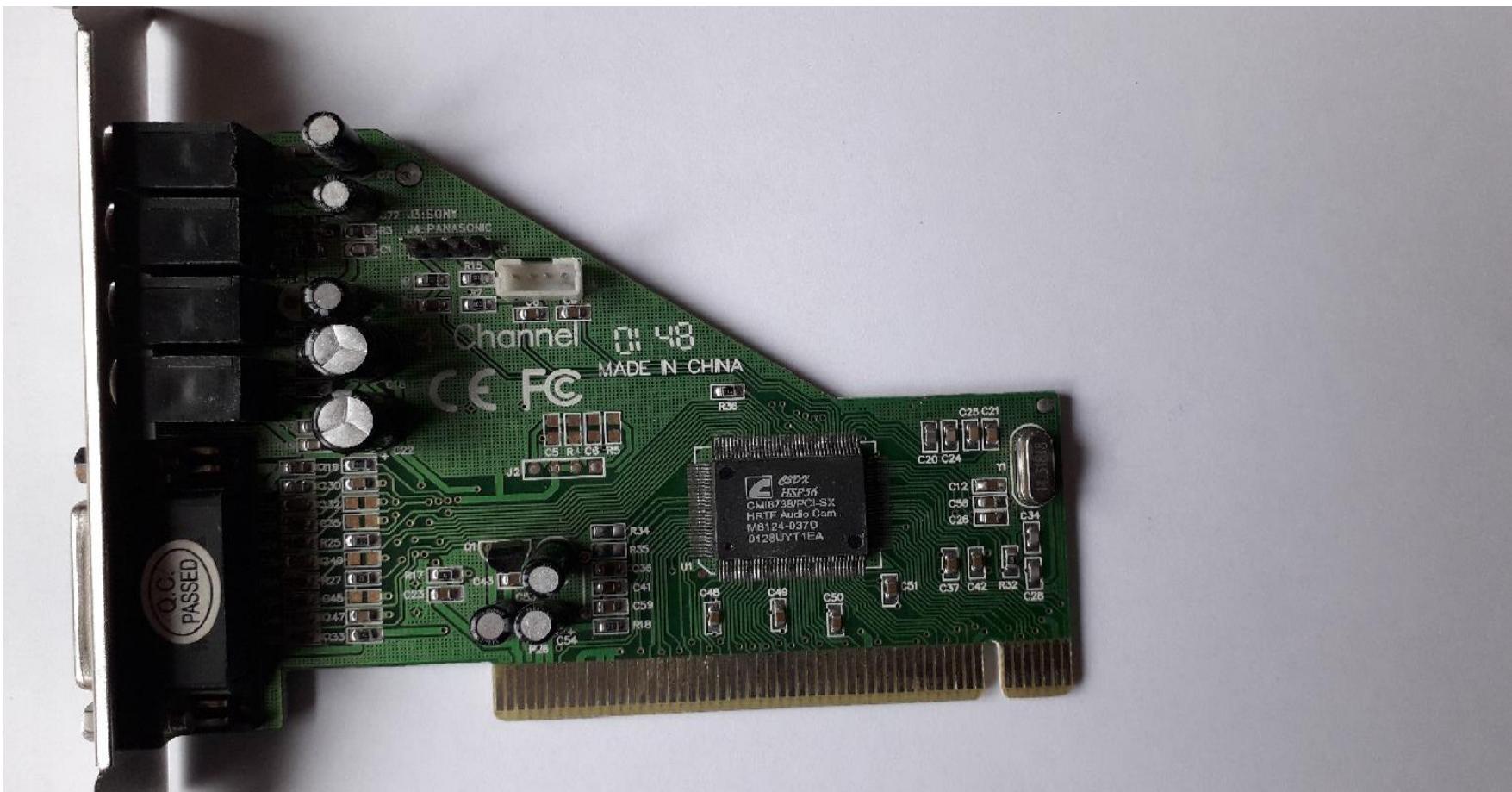
C36

C41

C51



# Контроллер PCI



# Контроллер в слоте

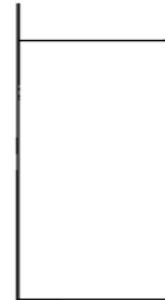


# Адресация портов контроллеров на шине ввода – вывода

## Одно адресное пространство

Адреса портов отображаются на адресное пространство памяти  
**MMIO – Memory Mapped IO**

Одно адресное  
пространство



- Каждому регистру контроллера, как ячейке памяти, выделен уникальный адрес в общем поле памяти,

Для обращения к портам используются те же команды, что и для обращения к памяти MOV...

# Карта памяти процессора ARM Cortex M3



# Адресация портов контроллеров на шине ввода - вывода

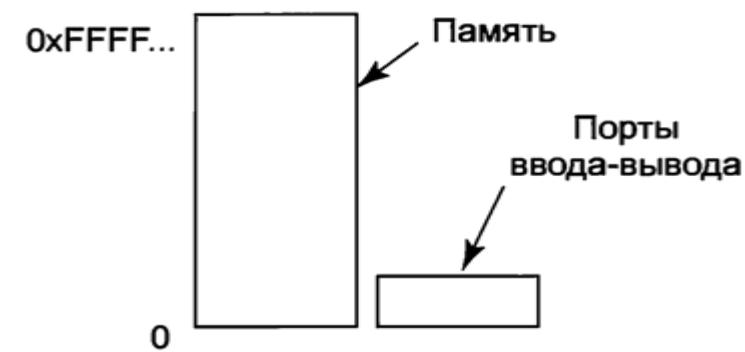
Два адресных пространства – пространство памяти и пространство портов.

Для адресации портов используется отдельное адресное пространство.

Для x86 оно составляет **64 Кбайта (16 бит)**

PMIO – Port Mapped IO

Два адресных  
пространства



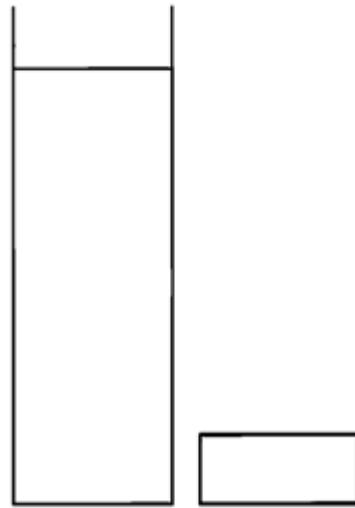
- Для обращения к портам используются специальные инструкции :

**in eax, 40h;  
out 40h, eax;**

# Адресация портов контроллеров на шине ввода - вывода

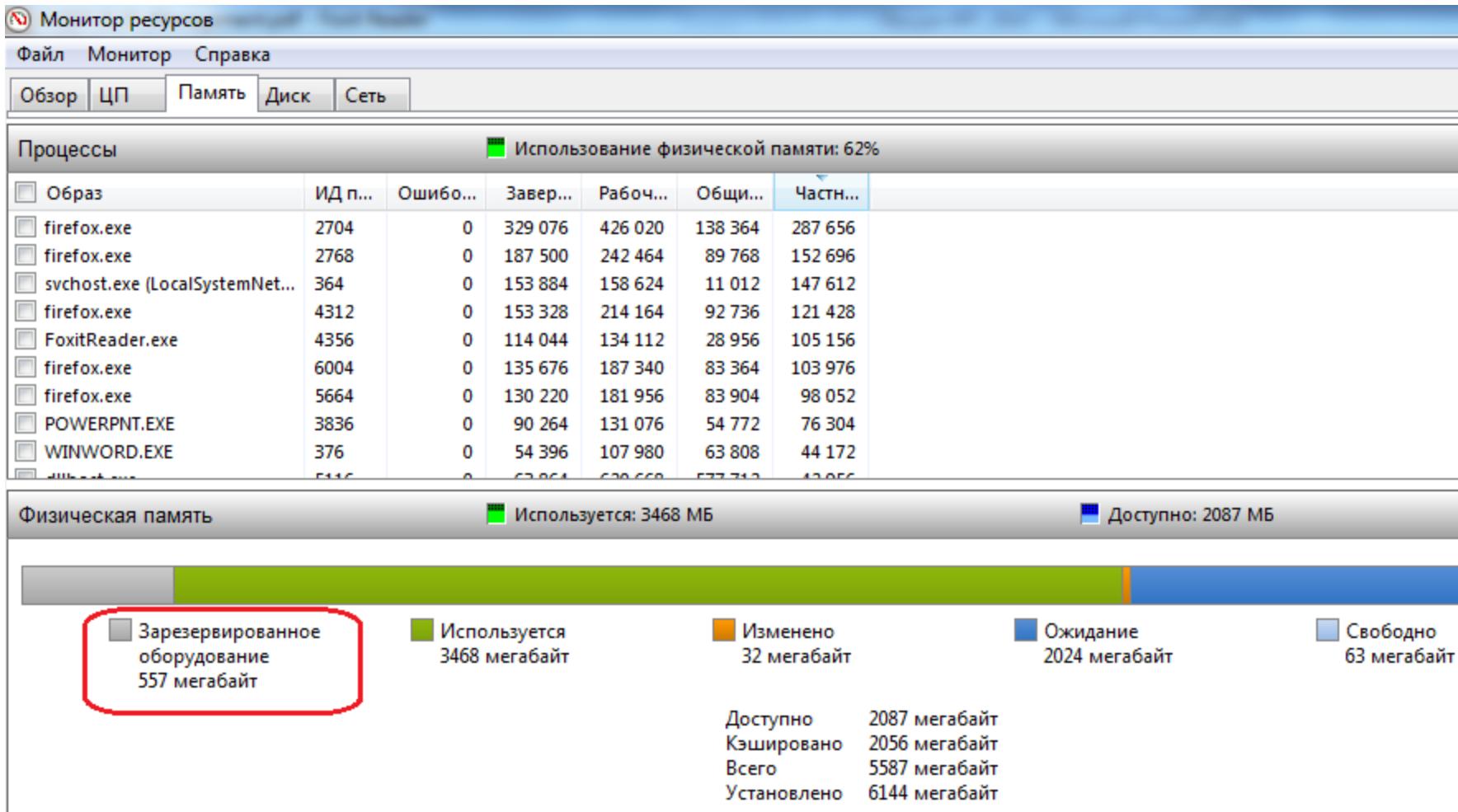
## Гибридный вариант

Два адресных пространства



- Используется в x86;
- Внутренняя память (буфер) контроллера (если имеется), находится в адресном пространстве памяти, а адресам портов ввода вывода отводится отдельное адресное пространство.

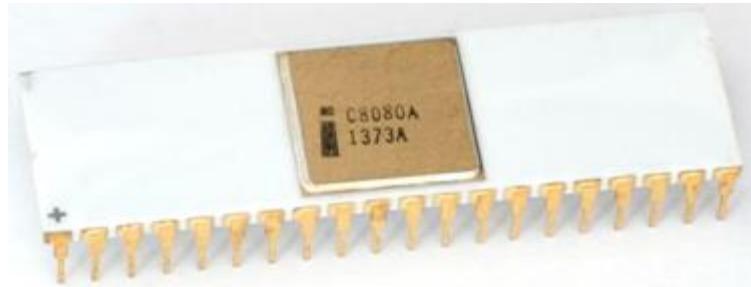
# Memory Mapped IO



# ОРГАНИЗАЦИЯ КОМПЬЮТЕРНОЙ АРХИТЕКТУРЫ НА БАЗЕ ПРОЦЕССОРА X8086

# Intel 8080

- Intel 8080 (*апрель 1974 года*) — 8-битный микропроцессор
  - Количество регистров: 7
  - Разрядность регистров: 8 бит
  - Разрядность шины данных: 8 бит
  - Разрядность шины адреса: 16 бит
  - Объём адресуемой памяти: 64 Кбайт
  - Количество инструкций: 80
  - Количество транзисторов 4758
  - Тактовая частота 2 – 4Мгц
- Отечественный аналог – микропроцессорный комплект серии **580ВМ80**



# Микропроцессорный комплект

- К процессору i8080 компанией Intel был выпущен микропроцессорный комплект поддержки (**Chipset**) в составе:
  - 8259 — контроллер прерываний;
  - 8257 — контроллер прямого доступа в память;
  - 8253 — 3-канальный таймер/счётчик;
  - 8251 — микросхема последовательного интерфейса;
  - 8255 — микросхема трёхканального параллельного интерфейса;
  - 8271 — контроллер накопителя на гибких магнитных дисках;
  - 8275 — контроллер монитора;
  - 8355 — микросхема интерфейса с периферией (с 16 Кбайт ПЗУ).
  - 8224 — тактовый генератор;
  - 8232 — процессор чисел с плавающей запятой (*позднее был заменен на сопроцессор Intel 8087*);

# Intel 8086

## ■ Intel 8086 (*июнь 1978*)

- Количество регистров: 14
- Разрядность регистров: 16 бит
- Разрядность шины данных: 16 бит
- Разрядность шины адреса: 20 бит
- Объём адресуемой памяти: 1 Мбайт
- Адресное пространство I/O: 1Кбайт
- Количество инструкций: 98
- Кэш-буфер команд: 6 байт
- Количество транзисторов: 29 000



# Архитектурные регистры

## Intel 8086

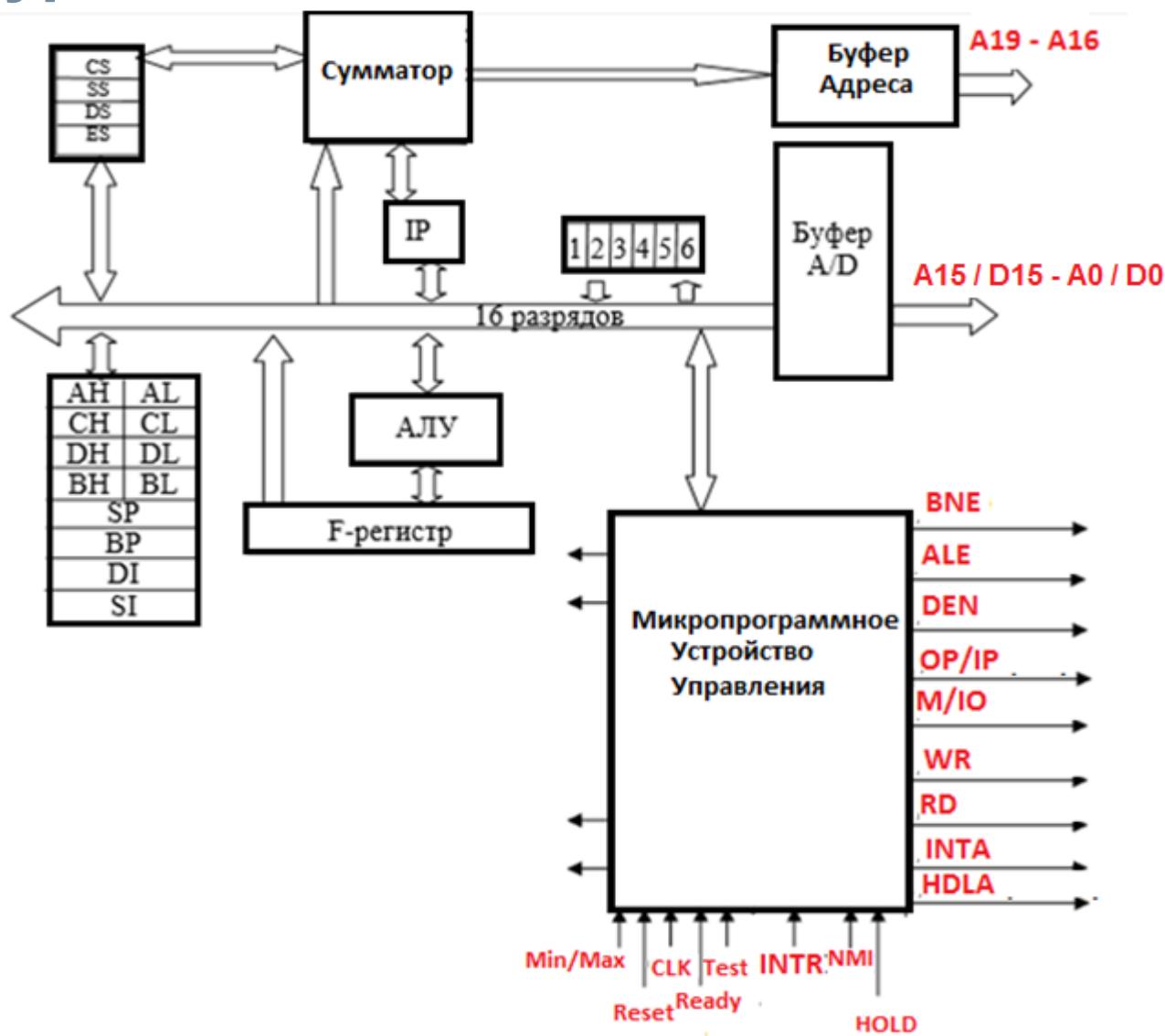
<u>Class</u>	<u>Register</u>	<u>Purpose</u>
Data:	AX,BX	“general” purpose
	CX	string and loop ops only
	DX	mult/div and I/O only
Address:	SP	stack pointer
	BP	base pointer (can also use BX)
	SI,DI	index registers
Segment:	CS	code segment
	SS	stack segment
	DS	data segment
	ES	extra segment
Control:	IP	instruction pointer (lower 16 bit of PC)
	FLAGS	C, Z, N, B, P, V and 3 control bits

# Intel 8088

- Intel 8088 (**июль 1979**)

- Количество регистров: 14
- Разрядность регистров: 16 бит
- Разрядность шины данных: **8** бит
- Разрядность шины адреса: 20 бит
- Объём адресуемой памяти: 1 Мбайт
- Адресное пространство I/O: 1 Кбайт
- Количество инструкций: 98
- Тактовая частота 5 -10 Мгц.

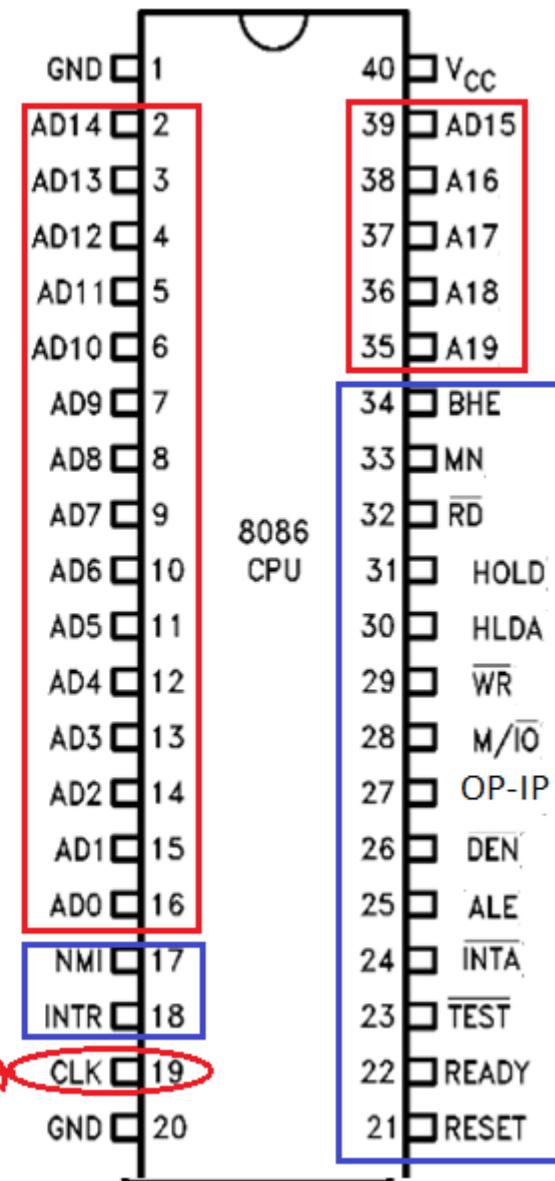
# Структурная схема i8086



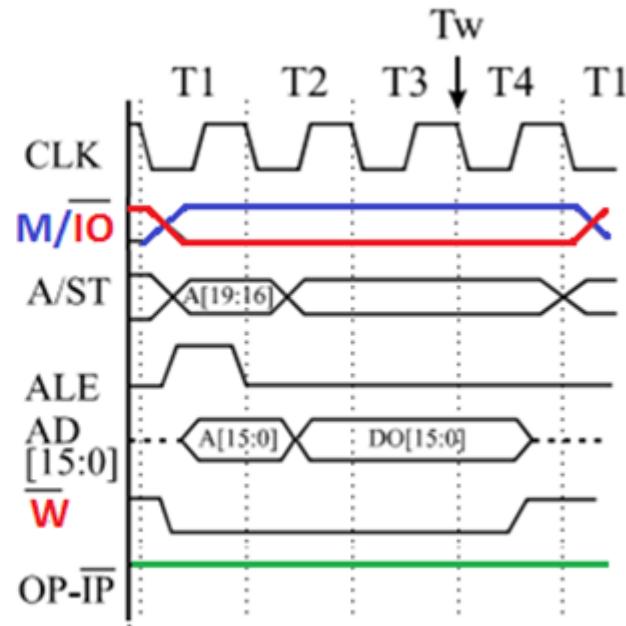
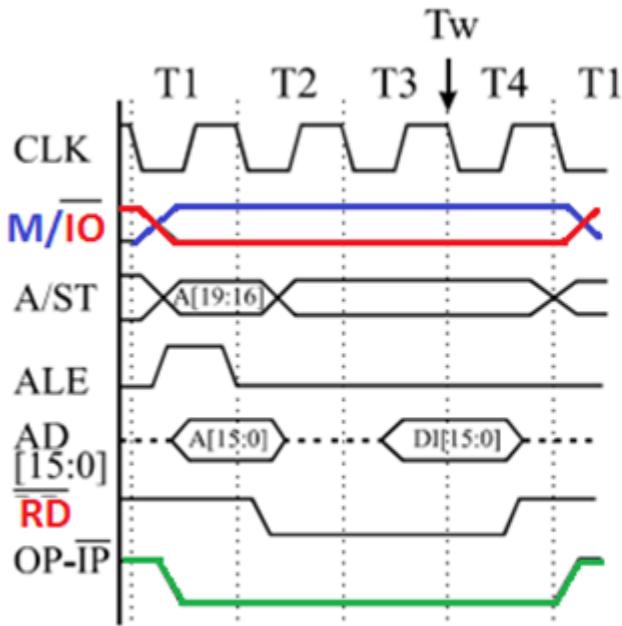
# Назначение выводов x8086

Назначение сигналов на выводах микропроцессора

Обозначение вывода	Вход/выход	Назначение вывода
AD15–AD0	Bх/вых	Мультиплексированная ША/Д
A16/S <sub>3</sub> –A19/S <sub>6</sub>	Вых	Линии адреса или состояния
BHE#/S <sub>7</sub>	Вых	Разрешение старшего байта шины
RD#	Вых	Управление чтением
WR#	Вых	Управление записью
M/IO#	Вых	Выбор памяти или ВУ
ALE	Вых	Разрешение фиксации адреса
<b>OP-IP#</b>	Вых	<b>Направление передачи данных</b>
DEN#	Вых	Разрешение пересылки данных
MN/MX#	Bх	Установка режима
TEST#	Bх	Сигнал окончания режима ожидания
HOLD	Bх	Запрос захвата шины
HLDA	Вых	Подтверждение захвата
INTR	Bх	Запрос прерывания
NMI	Bх	Запрос немаскируемого прерывания
INTA#	Вых	Подтверждение прерывания
READY	Bх	Готовность памяти или ВУ
RESET	Bх	Сброс (начальная установка)
CLK	Bх	Такты ГТИ
GND, +5B	Bх	Общий (земля), питание

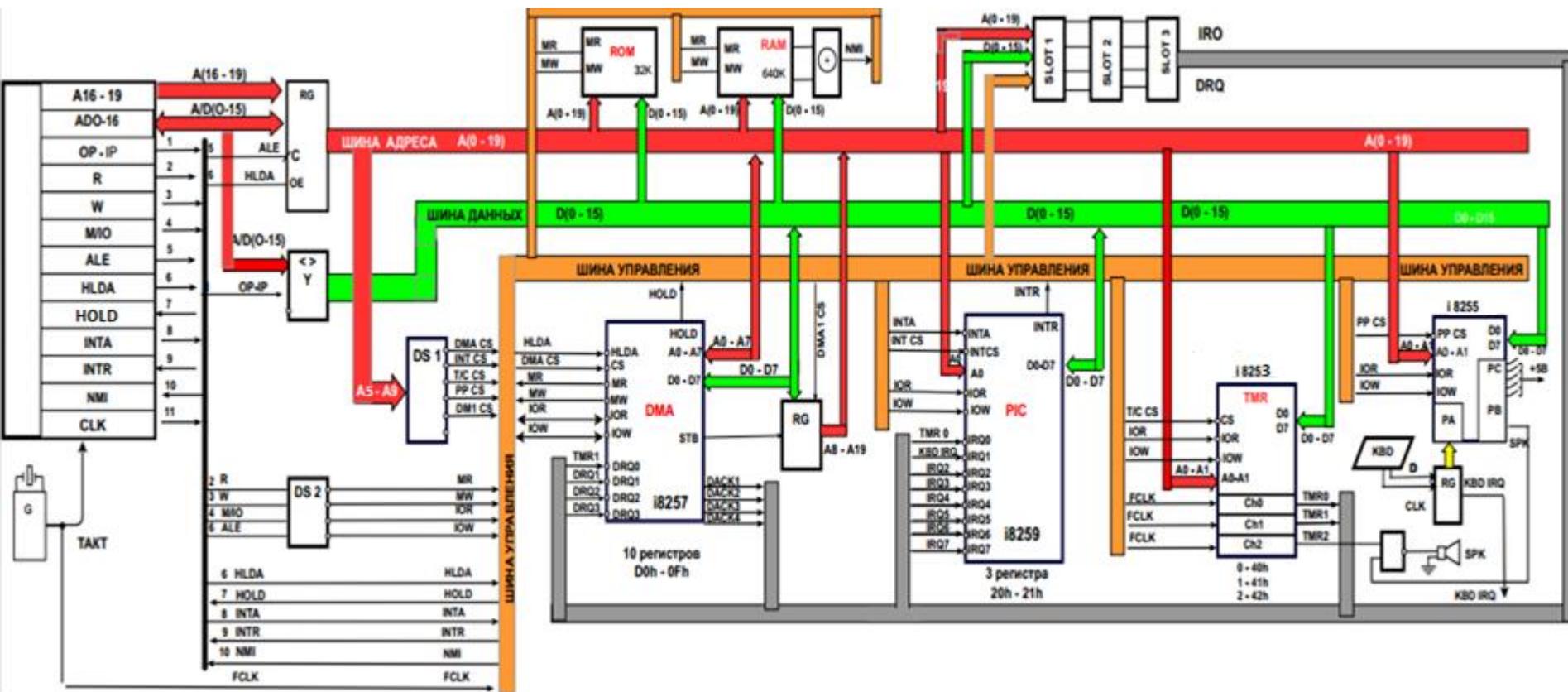


# Машинный цикл процессора (чтение/запись )

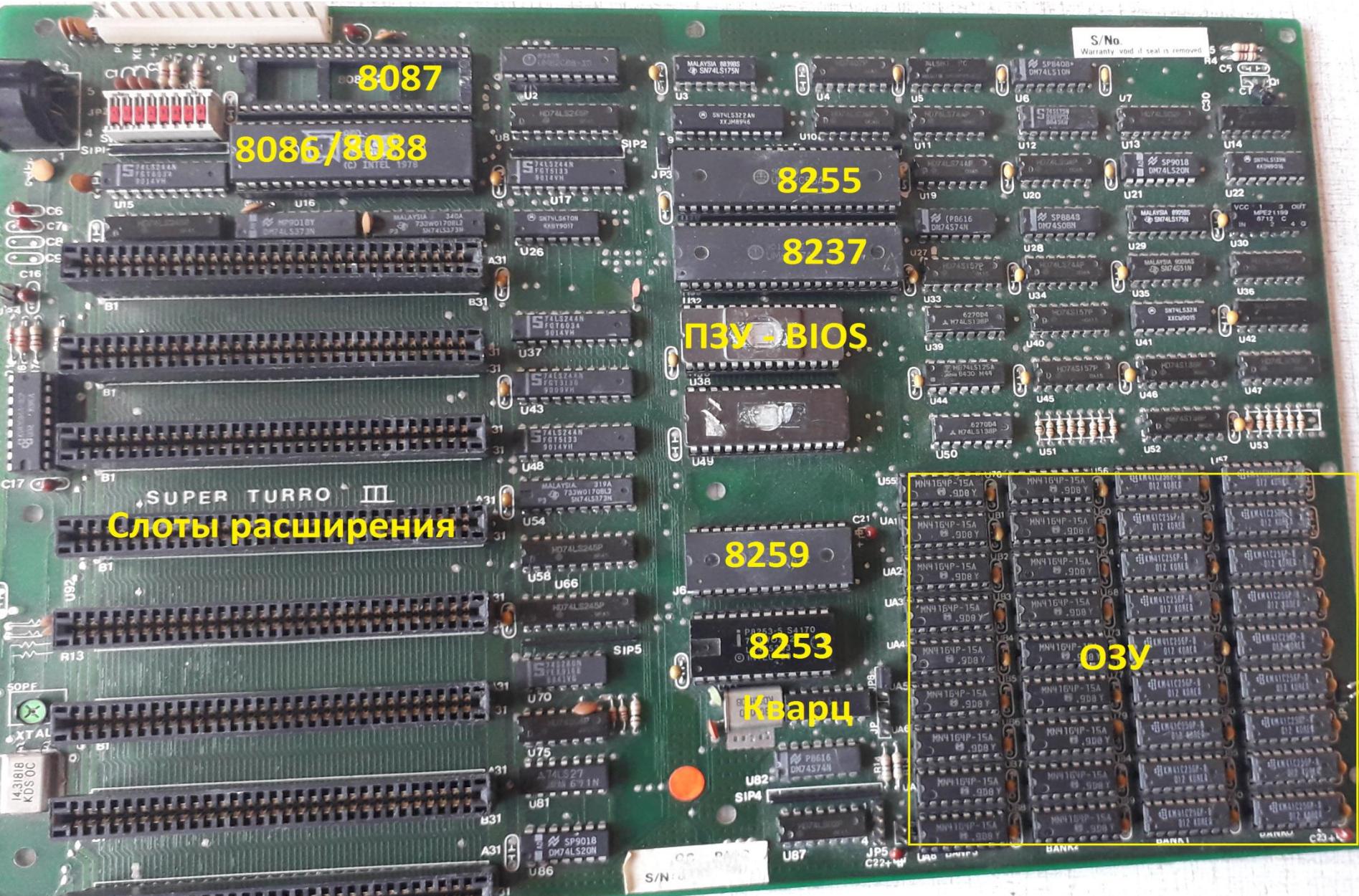


- Машинный цикл чтения / записи занимает четыре такта генератора

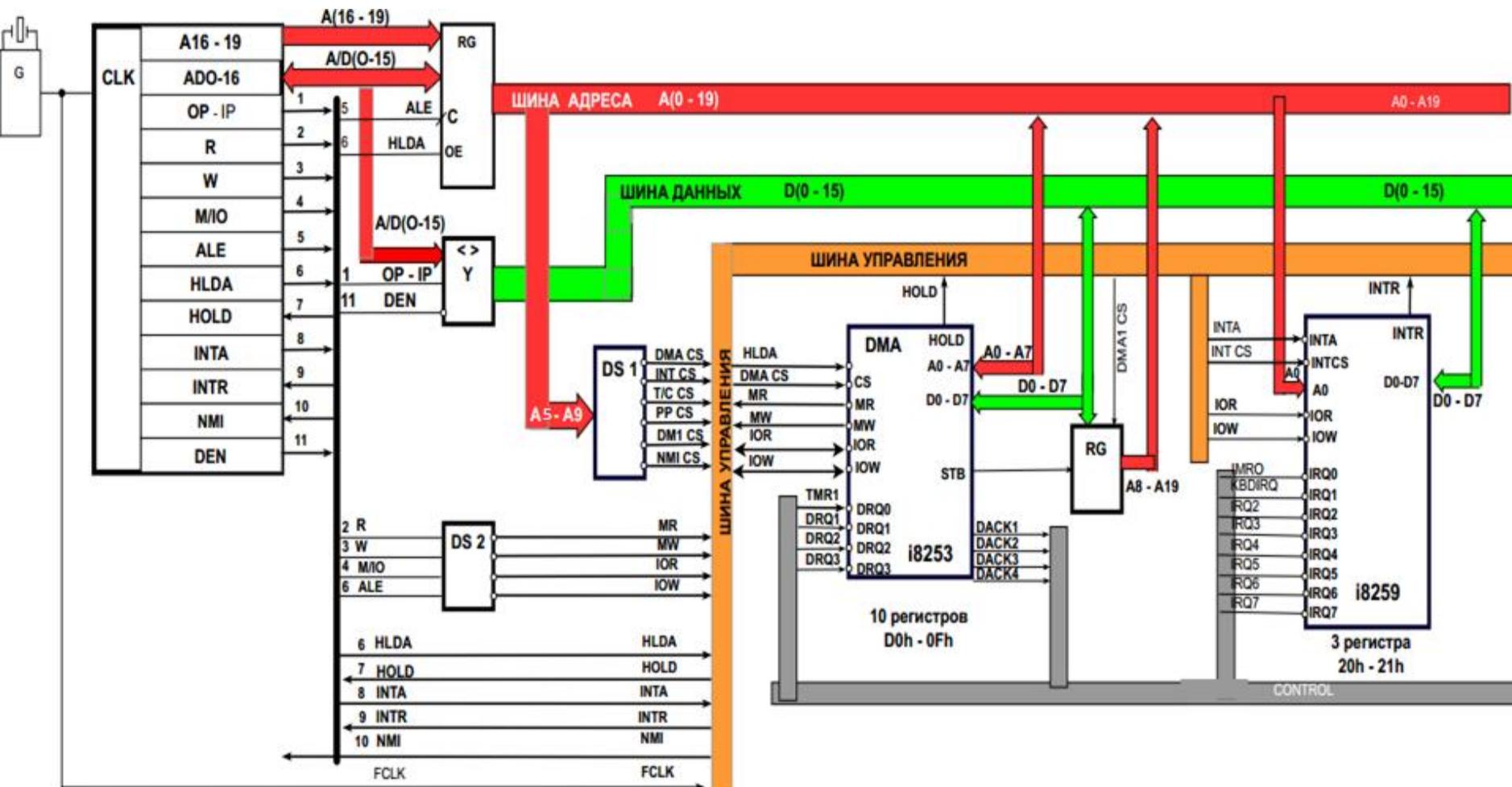
# IBM PC XT



# IBM PC XT



# Схема подключения процессора к системной шине



# Схема подключения процессора

## ■ RGA – регистр адреса

- Служит для временного хранения адреса памяти или порта В/В на ША в течение времени обращения процессора по данному адресу.

## ■ Усилитель

- служит для выбора направления передачи данных от процессора или к процессору и усиления сигналов данных на ШД

## ■ DS1

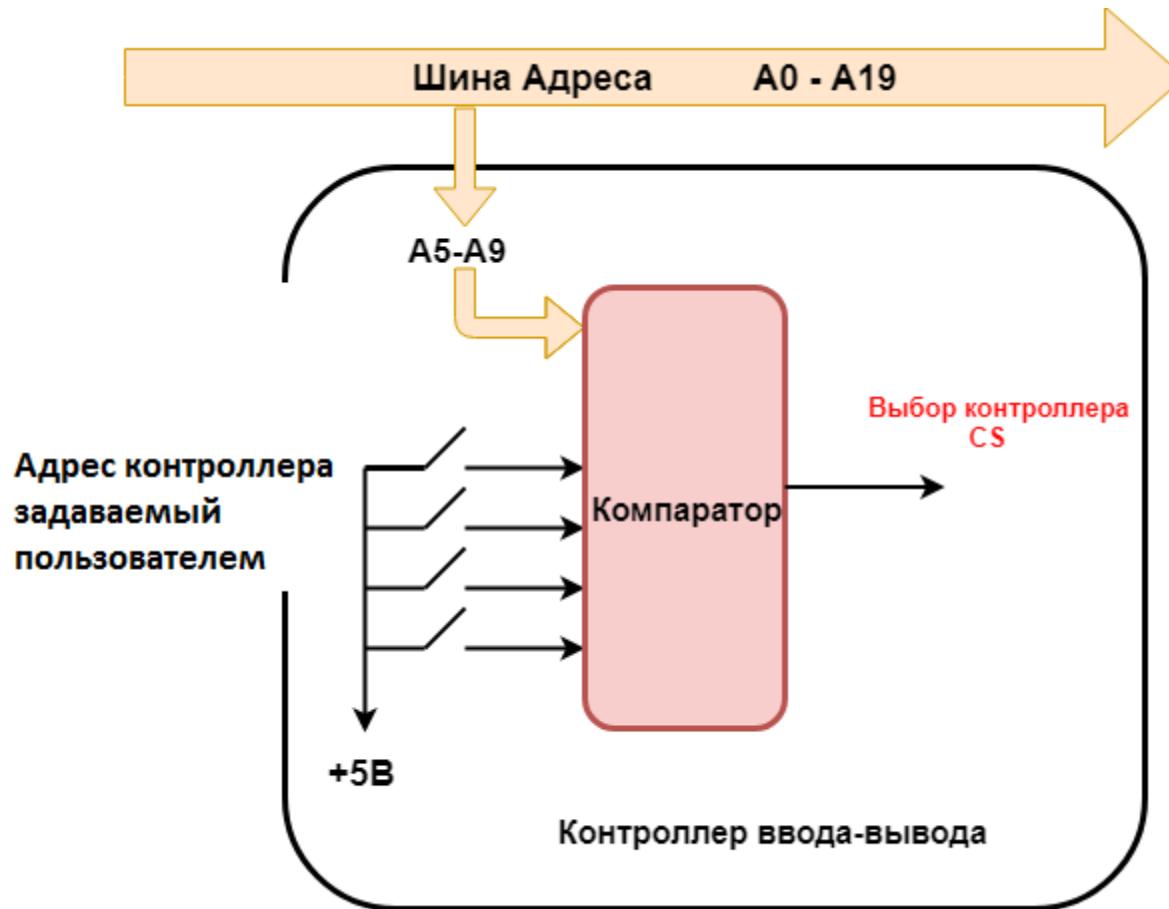
- дешифрирует разряды А5-А9 адреса для выбора системных контроллеров В/В (базовый адрес). Выходы дешифратора в виде сигналов CS (Chip Select ) подаются на аналогичные входы контроллеров ввода-вывода.
- Смещение относительно базового адреса задается разрядами А0-А4 адреса, поступающими на адресные входы выбранных контроллеров.

## ■ DS2

- дешифрирует сигналы R, W, M/IO процессора в сигналы на ШУ MR, MW, RIO, WIO для записи/чтения ячейки памяти или выбранного сигналом CS регистра контроллера ввода-вывода

## ■ КПДП – контроллер прямого доступа к памяти

# Адресация контроллеров в слотах расширения



Базовый адрес контроллера в слоте расширения задавался вручную перемычками

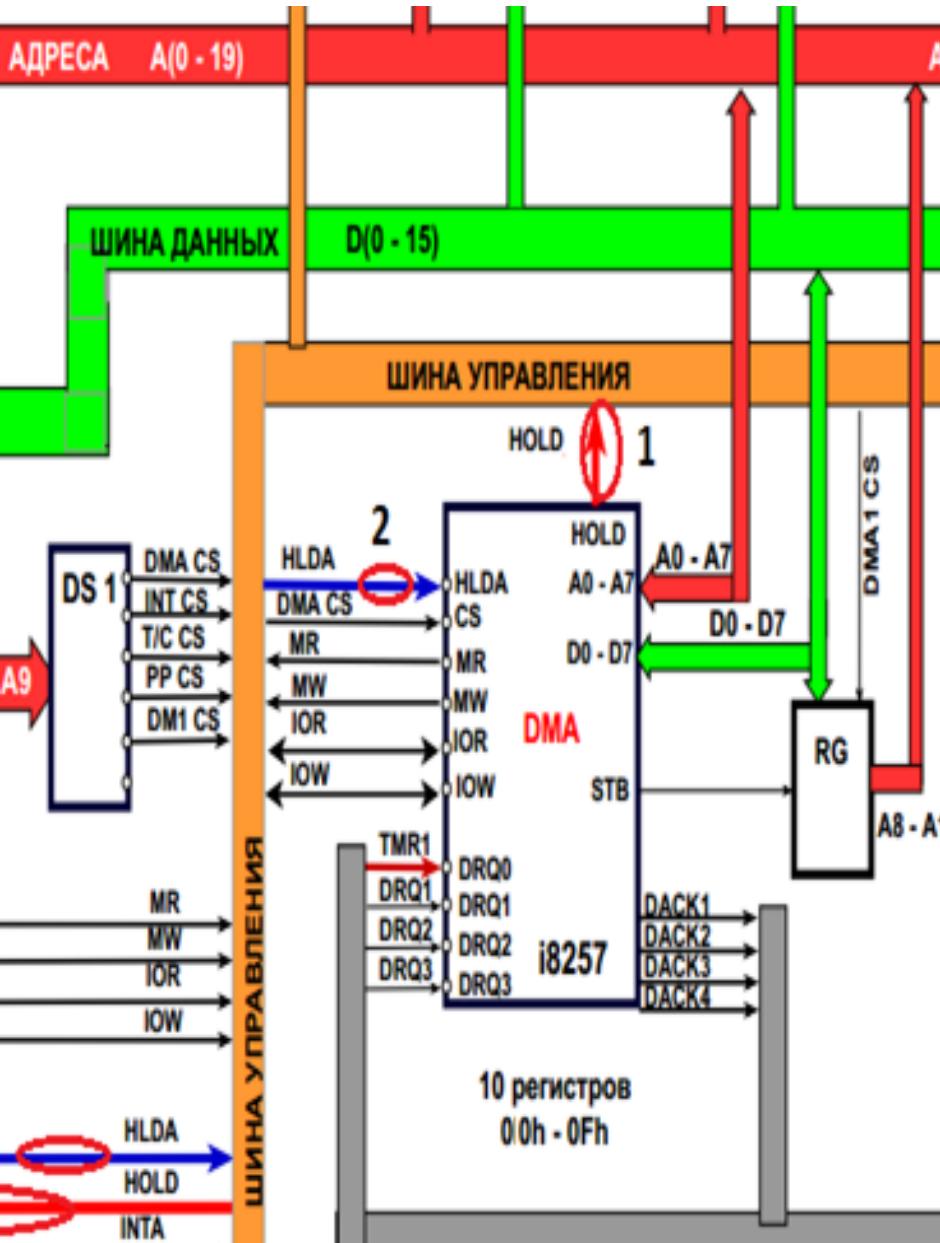
# Контроллер прямого доступа к памяти

*DMA (direct memory access)*

# Контроллер прямого доступа

- Контроллер прямого доступа к памяти (DMA - Direct Memory Access) обеспечивает высокоскоростной обмен данными между контроллером ввода-вывода и ОЗУ минуя центральный процессор.
- Берет на себя управление системной шиной и управляет передачей информации между контроллерами ввода-вывода и ОЗУ.
- Позволяет освободить процессор для выполнения вычислений параллельно с обменом.
- Функции системного контроллера ПДП выполняла микросхема i8237A

# Контроллер ПДП



**DMA CS** – сигнал выбора КПДП

**DRQ** – запрос от контроллеров В/В к КПДП

**HOLD** – запрос от КПДП к процессору на захват шины

**HLDA** – ответ процессора КПДП и отключение процессора от шины

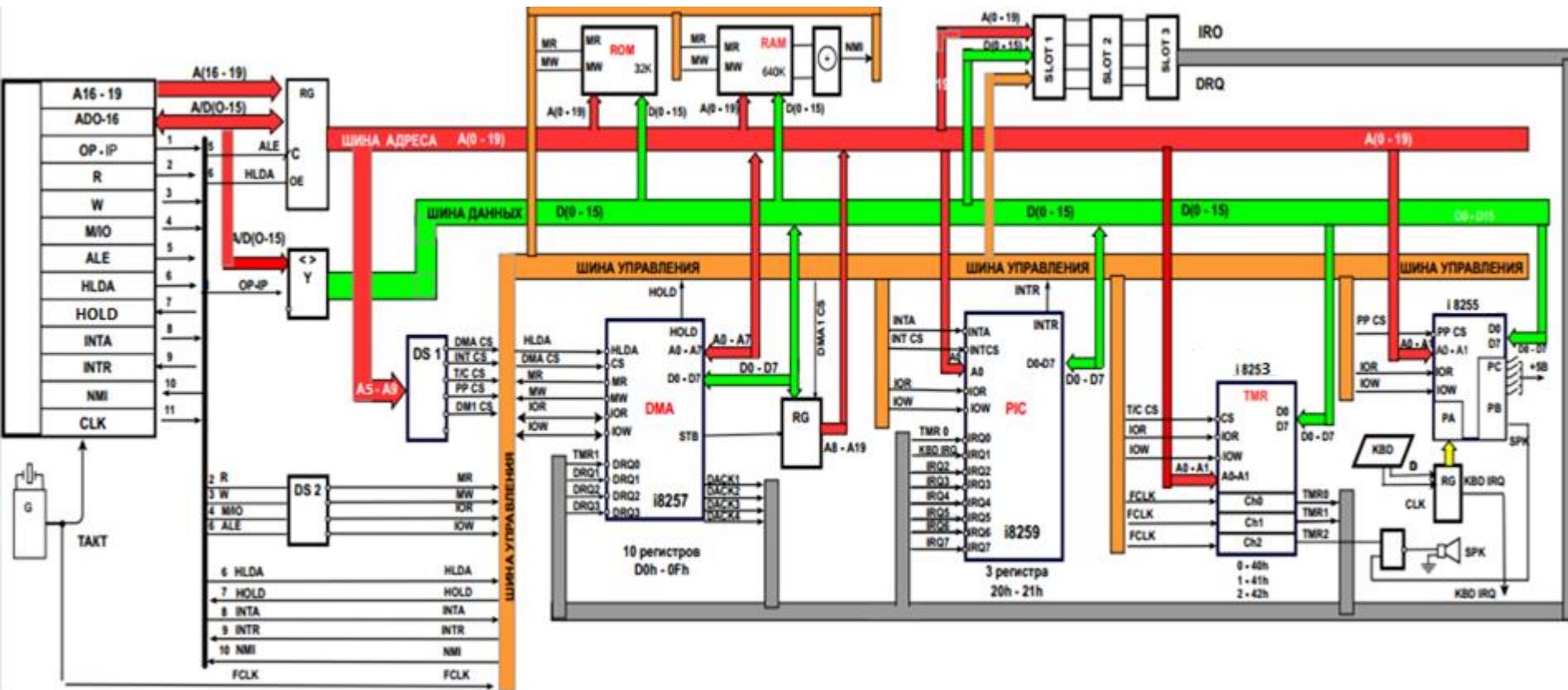
**DACK** – захват шины КПДП и разрешение устройству ПДП

**A0-A7, A9-A19** - адрес выставляемый КПДП

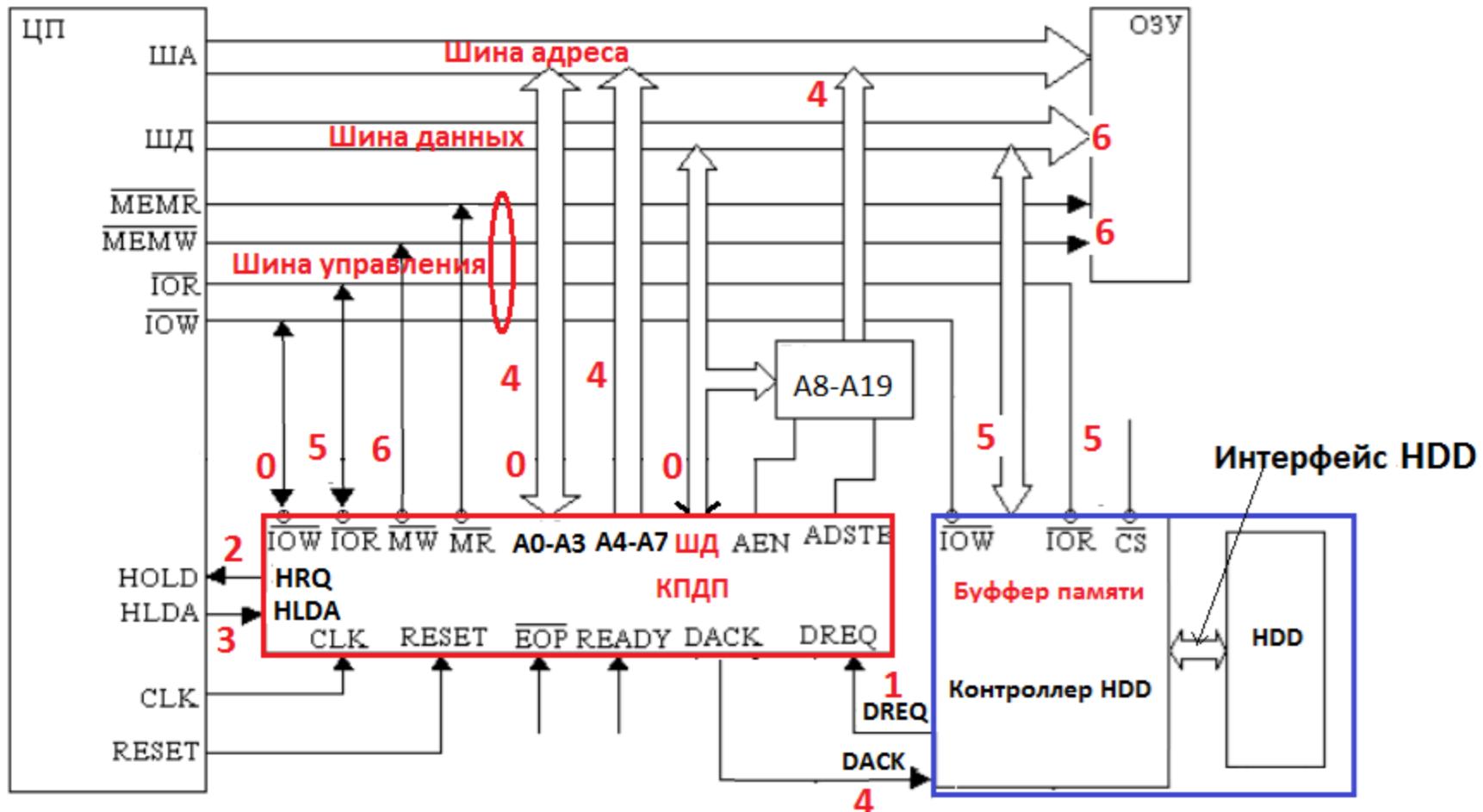
**IOR,IOW,MR,MW** – запись чтение устройства и памяти вырабатываляемые КПДП

**D0-D7** – данные

# КПДП в общей схеме



# Последовательность передачи при ПДП на примере контроллера HDD



# Инициализация контроллера DMA

Процессор «знает» с какой областью памяти работает внешнее устройство (начальный адрес области памяти и количество передаваемых байт)

При инициализации КПДП процессором задается :

- начальный адрес оперативной памяти для обмена;
- число передаваемых байт;
- направление обмена, а также устанавливаются требуемые режимы работы (*одиночная или блочная передача и т. д.*).

# Последовательность операций при ПДП

- 0. Инициализация контроллера
- 1. Принять Запрос (DREQ) на ПДП от контроллера HDD к контроллеру ПДП
- 2. Выставить запрос (сигнал HOLD) от контроллера ПДП к процессору на захват шины
- 3. Ответ процессора (сигнал HLDA) контроллеру ПДП (процессор переводит свои выводы в третье состояние и отключается от ША, ШД, ШУ)
- 4. Контроллер ПДП берет на себя управление шинами и выставляет адрес памяти на шину адреса и передает сигнал ответа DACK N в контроллер HDD
- 5. Контроллер ПДП вырабатывает сигналы IOR чтения данных с контроллера диска. Данные появляются нашине данных.
- 6 Контроллер ПДП вырабатывает сигнал MEMW записи в память . Байт данных переписывается напрямую из буфера КВВ HDD в ОП через шину данных..
- 6. Пункты 5-6 повторяются пока не будут переданы все данные, количество которых задано в КПДП.
- 7. Контроллер ПДП снимает сигнал HOLD и DACK, а процессор сигнал HLDA и возвращает себе управление системной шиной.

# Внутренние регистры

- Адреса програмно доступных регистров в диапазоне 00h – 0Fh

Режимы W-запись R-чтение	Число бит	Адреса DMA1 (h)	Назначение регистров (портов)
W	8	08	Регистр управления (команд)
R	8	08	Регистр состояния
W	8	0B	Регистр режима
W	4	0A	Регистр одиночны масок
W	4	0F	Регистр масок
W	4	09	Регистр запросов
W	*	0E	Сброс регистра маски
W	*	0C	Сброс триггера первый/последний (Тп/п)
*	*	*	
R	8	0D	Регистр данных
R/W	16	00	Регистр текущего адреса канала0/канала4

# Каналы ПДП для РС ХТ

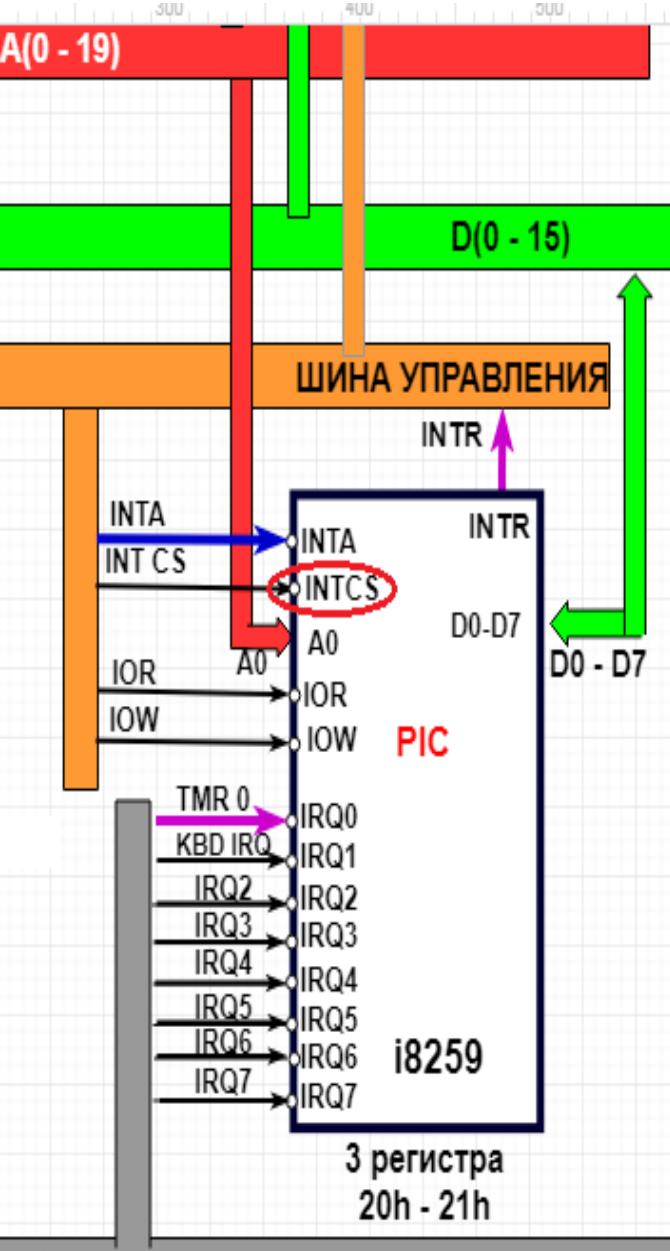
Каналы	Назначение
К0	для РС/ХТ регенерация памяти ,
К1	Резерв
К2	Работа с контроллером НГМД
К3	Резерв или для некоторых типов ПК работа с жестким диском

В РС ХТ на нулевой канал КПДП каждые 15 мкс подавался сигнал от первого канала таймера, по которому КПДП совершает «перебор» ячеек ОП для регенерации их содержимого.

# Контроллер аппаратных прерываний

*PIC (Programmable Interrupt Controller)*

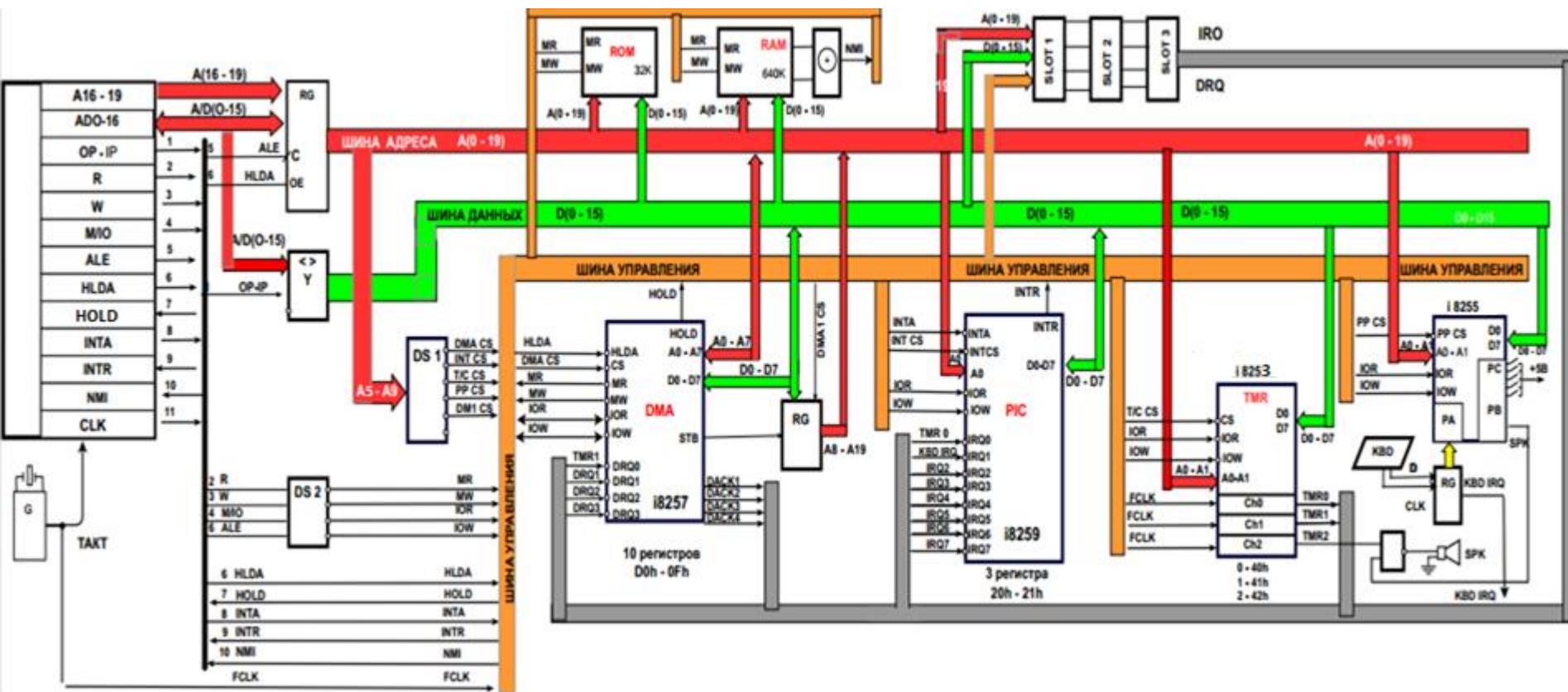
# Контроллер аппаратных прерываний



Сигналы:

**INT CS** – выбор чипа контроллера прерываний  
**IRQ0-IRQ7** входы запросов на прерывание  
**INTR** – запрос на прерывание к процессору  
**INTA** – ответ процессора  
**A0** – выбор внутренних регистров,  
**IOR/IOW** – запись/чтение в контроллер  
**D0-D7** шина данных,

# Контроллер аппаратных прерываний (в общей схеме)



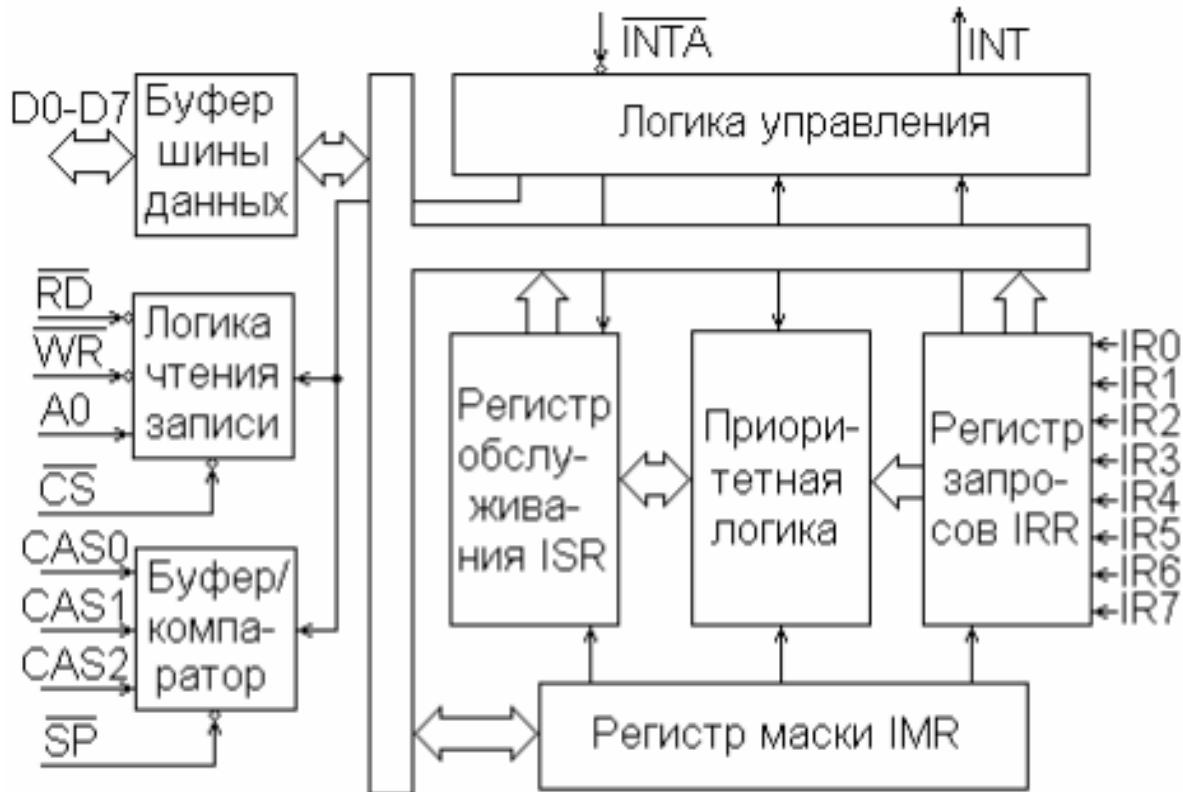
# Временная диаграмма



## Внутренние регистры

- **Регистр запросов IRR ( Interrupt request register )** – запоминает все запросы на входе IRQ.
- **Регистр состояния ISR ( Interrupt Status Register )** – задает приоритет обработки прерываний
- **Регистр маскирования IMR ( Interrupt Mask Register )** – запрещает отдельные прерывания
- Адреса регистров
- ISR: 20H  
IMR:21H
- Обращение к IRR с помощью специальных управляющих слов

# Регистры контроллера прерываний i8259

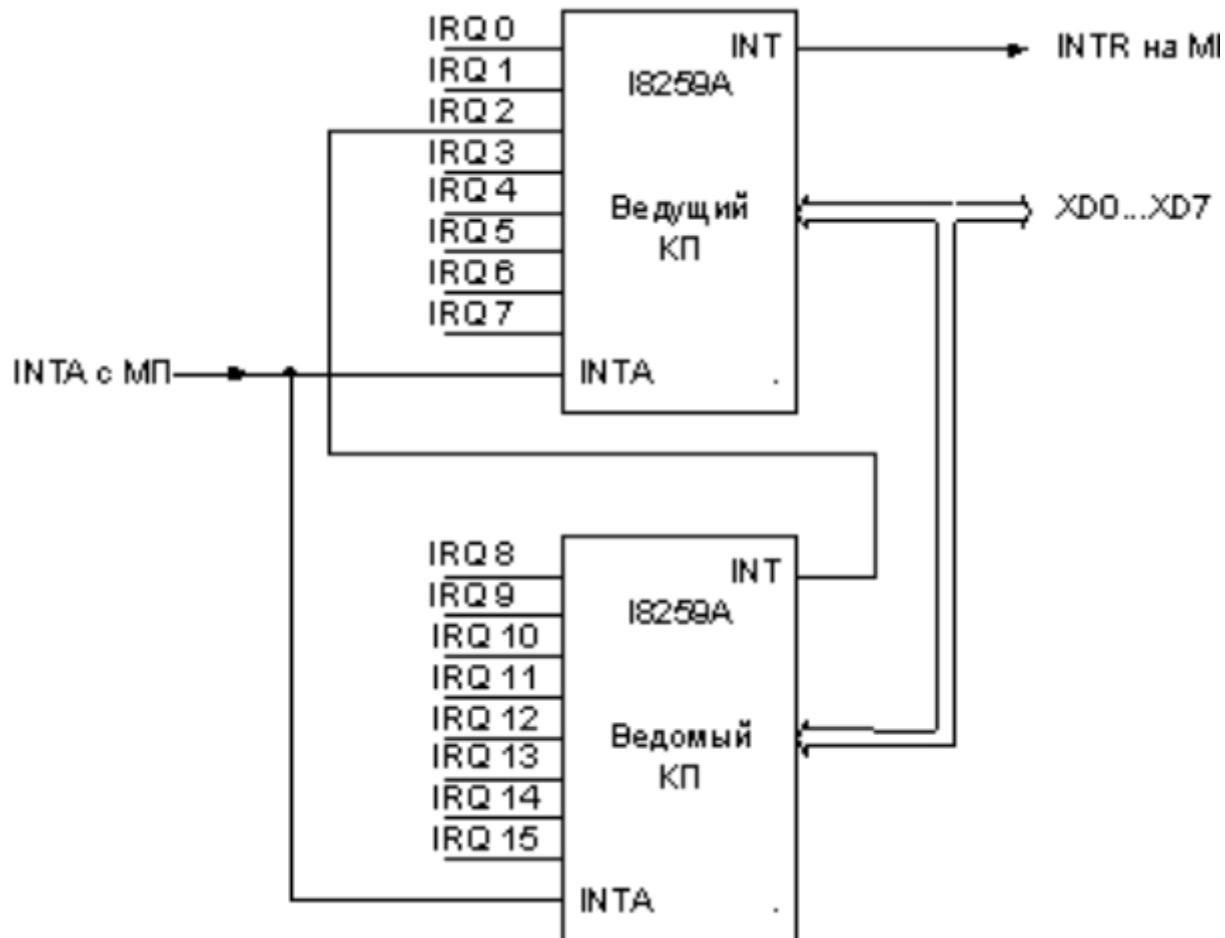


16	SP	PIC	$\leftrightarrow Z$	11
3	RD	D0	10	
2	WR	D1	9	
26	INTA	D2	8	
27	A0	D3	7	
1	CS	D4	6	
18	IR0	D5	5	
19	IR1	D6	4	
20	IR2	D7		17
21	IR3			
22	IR4			
23	IR5			
24	IR6			
25	IR7			
		INT		
			$\leftrightarrow$	
		CAS0	12	
		CAS1	13	
		CAS2	15	

# Порядок операций при прерывании

- 1. Инициализация контроллера через запись
- 2. Контроллер устройства В/В вырабатывает сигнал запроса на прерывание, который поступает на входы IRQ0-IRQ7 КПР
- 3. КПР вырабатывает сигнал запроса на прерывание INTR процессору, который поступает на соответствующий вход процессора
- 4. Процессор в ответ вырабатывает два раза сигнал подтверждения прерывания INT A контроллеру В/В.
  - Первый раз биты запроса на прерывание фиксируются в регистре запросов IRR.
  - Второй раз номер входа (вектора) прерывания IRQ по шине данных считывается в процессор, разрешая запись новых запросов прерываний IRQ.
- 5. В процессоре номер входа (вектора) умножается на 4 (сдвигается влево на 2 бита) и получается адрес в таблице векторов прерываний, по которому хранится адрес начала обработчика прерываний.

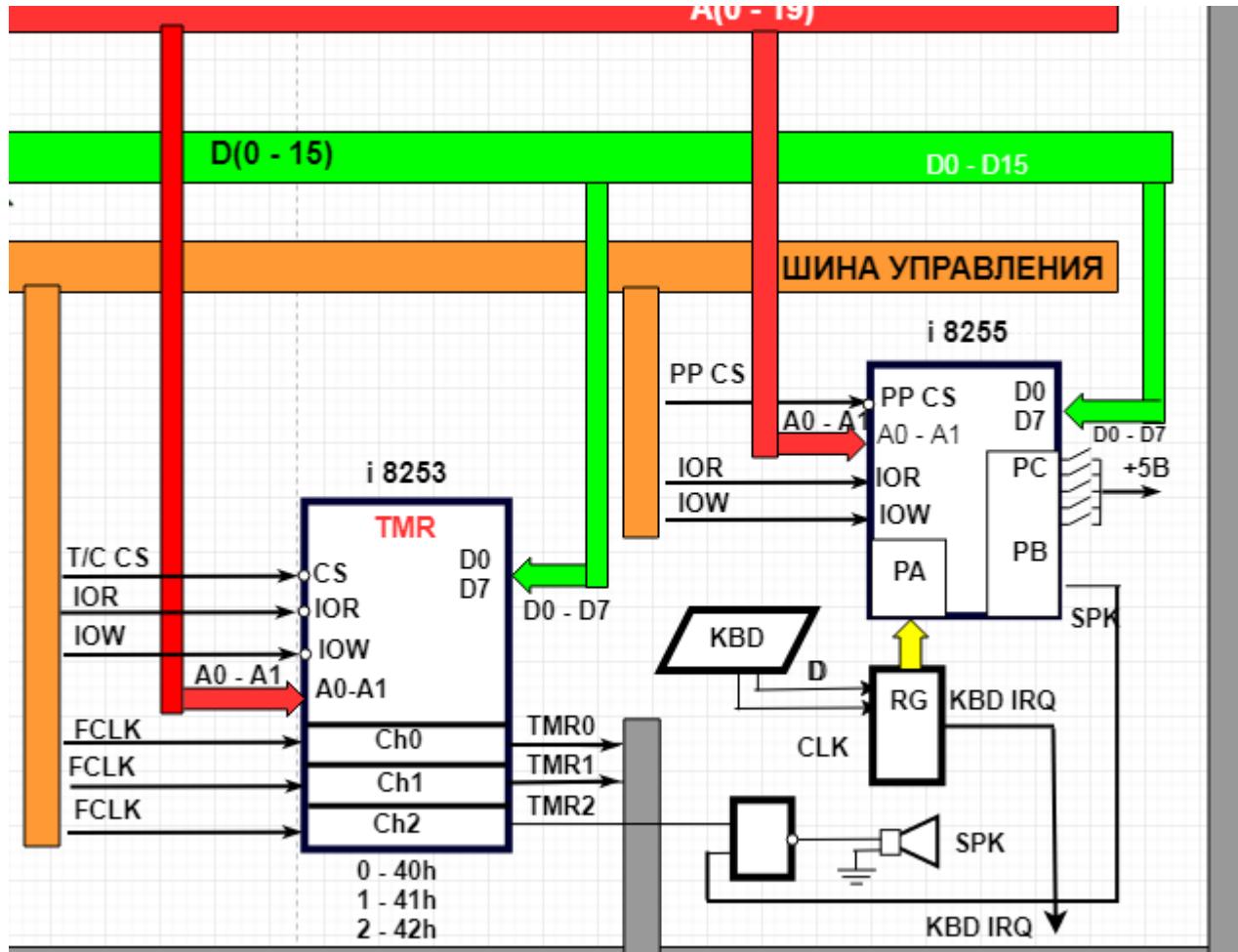
# Каскадирование контроллеров



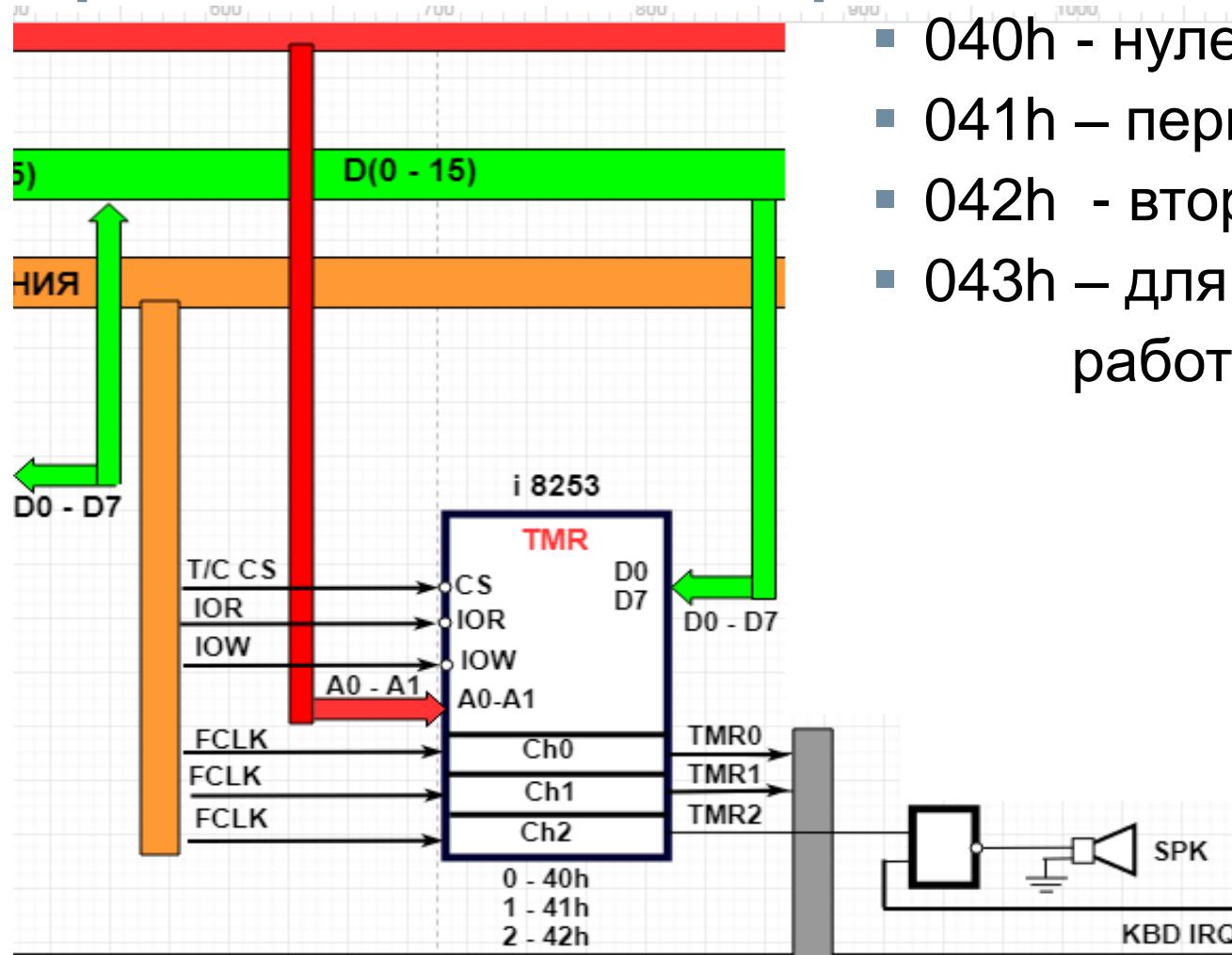
# Приоритеты и номера прерываний

Имя (номер <sup>1</sup> )	Вектор <sup>2</sup>		Контроллер/маска	Описание
	Реальный	Защищенный		
NMI	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

# Трехканальный таймер



# Трехканальный таймер i8253



- 040h - нулевой канал
- 041h – первый канал
- 042h - второй канал
- 043h – для задания режимов работы каналов таймера.

Каждый канал представляет собой независимый счетчик с программно-управляемым коэффициентом пересчета, который может работать в шести режимах

# Трехканальный таймер i8253 для РС ХТ

## ■ Нулевой канал

- Используется для ведения системного времени, вызывая каждую **1/18** секунды прерывание процессора и запуская программу, увеличивающую содержимое четырех байт памяти (начиная с адреса **046Ch**) на 1.

## ■ Первый канал

- Каждые 15мкс обращается к каналу контроллера ПДП и запускает процедуру **регенерации ОЗУ**.

## ■ Второй канал

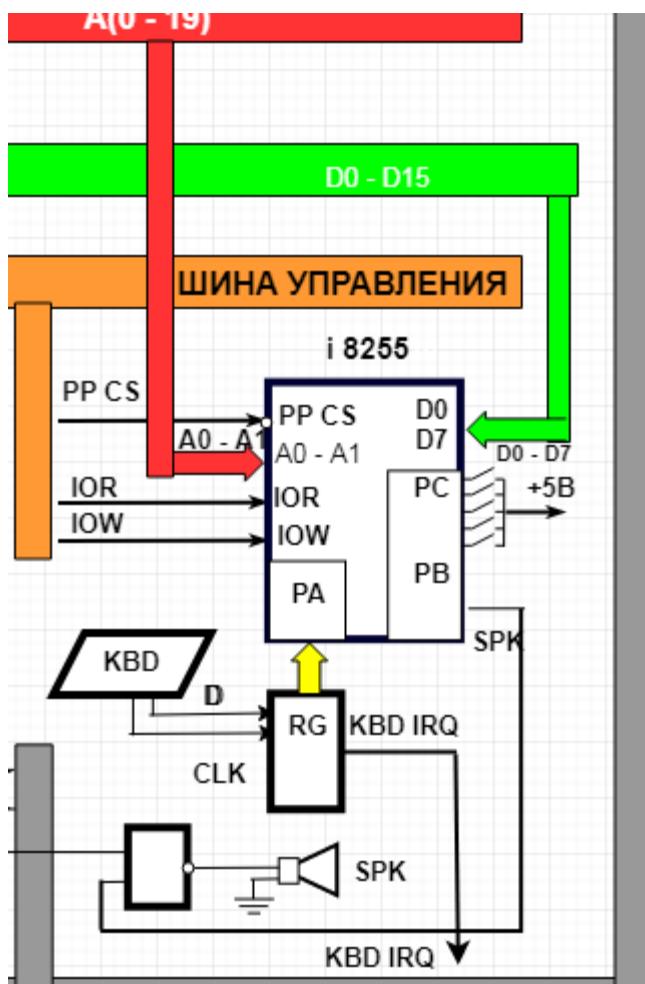
- Генерирует частоту, выдаваемую на звуковую головку для генерации звуковых сигналов. (например, при ошибках памяти )

# Программируемый параллельный интерфейс

- **PPI (Programmable Parallel Interface) i8255A.**
- Включает три двунаправленных байтовых порта (A, B и C).
- **Порт A** (адрес 60h)
  - Для чтения поступающего с клавиатуры сканкода нажатой клавиши (после нажатия клавиши, возникает прерывание **IRQ1** по которому обработчик считывает с порта А код нажатой клавиши)
- **Порт B** (адрес 61h)
  - Для вывода управляющих сигналов различного назначения (например для разрешения или запрещения звуковых сигналов).
- **Порт C** (адрес 62h)
  - используется BIOS и ОС для чтения свитчей конфигурации оборудования системы. Позже эта функция перенесена в CMOS – память BIOS

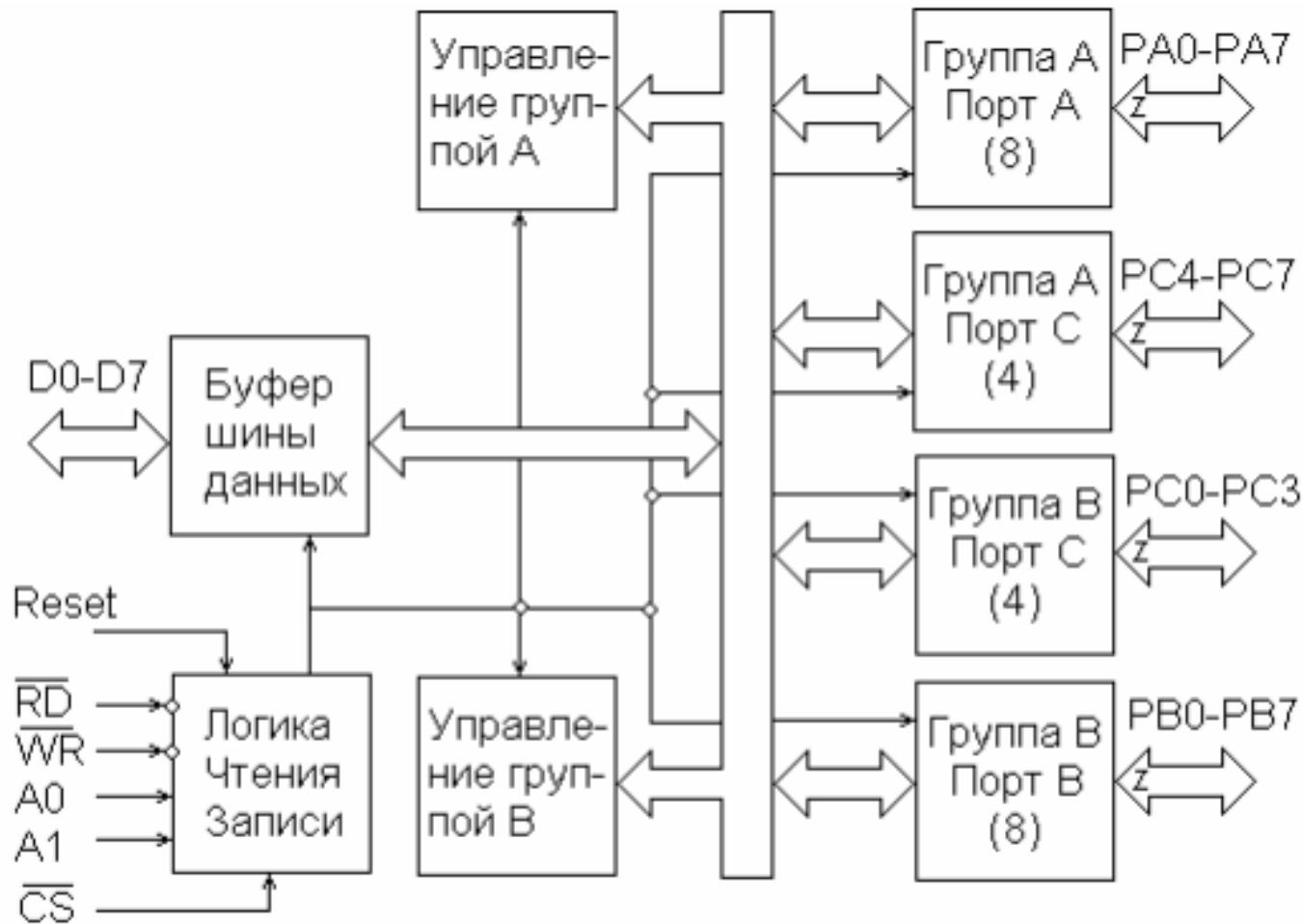
Позже микросхема i8255 использовалась в качестве параллельного порта (LPT интерфейса) для подключения различных устройств (например, принтера)

# PPI (Programmable Parallel Interface) i8255A.



Код нажатой клавиши запоминается в специальном регистре и по нему вырабатывается запрос на прерывание, который поступает на вход IRQ1, контроллера прерываний

# Структурная схема i8255

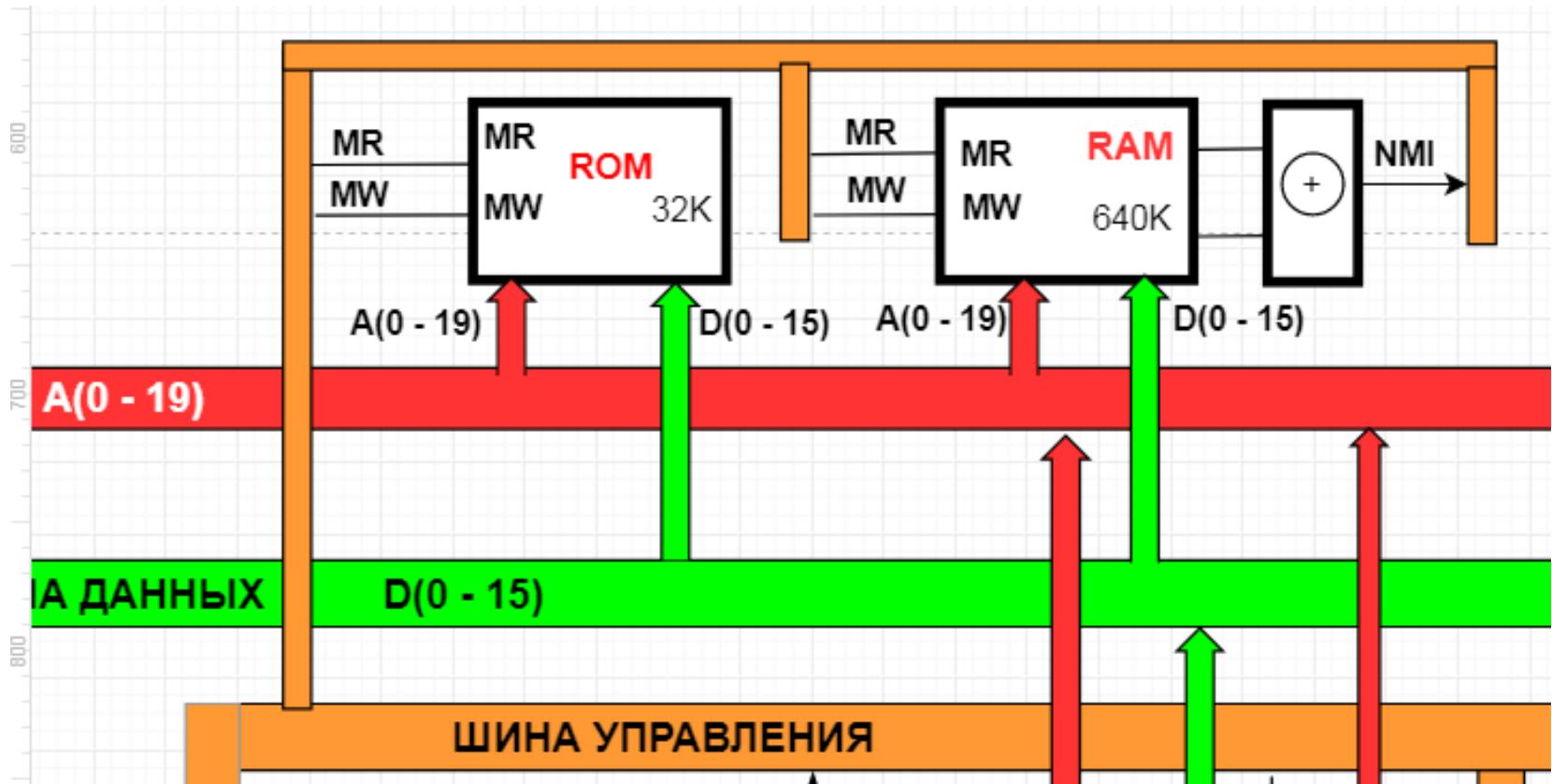


# Универсальный последовательный синхронно-асинхронный приемо-передатчик i8251 (**СОМ порт**)

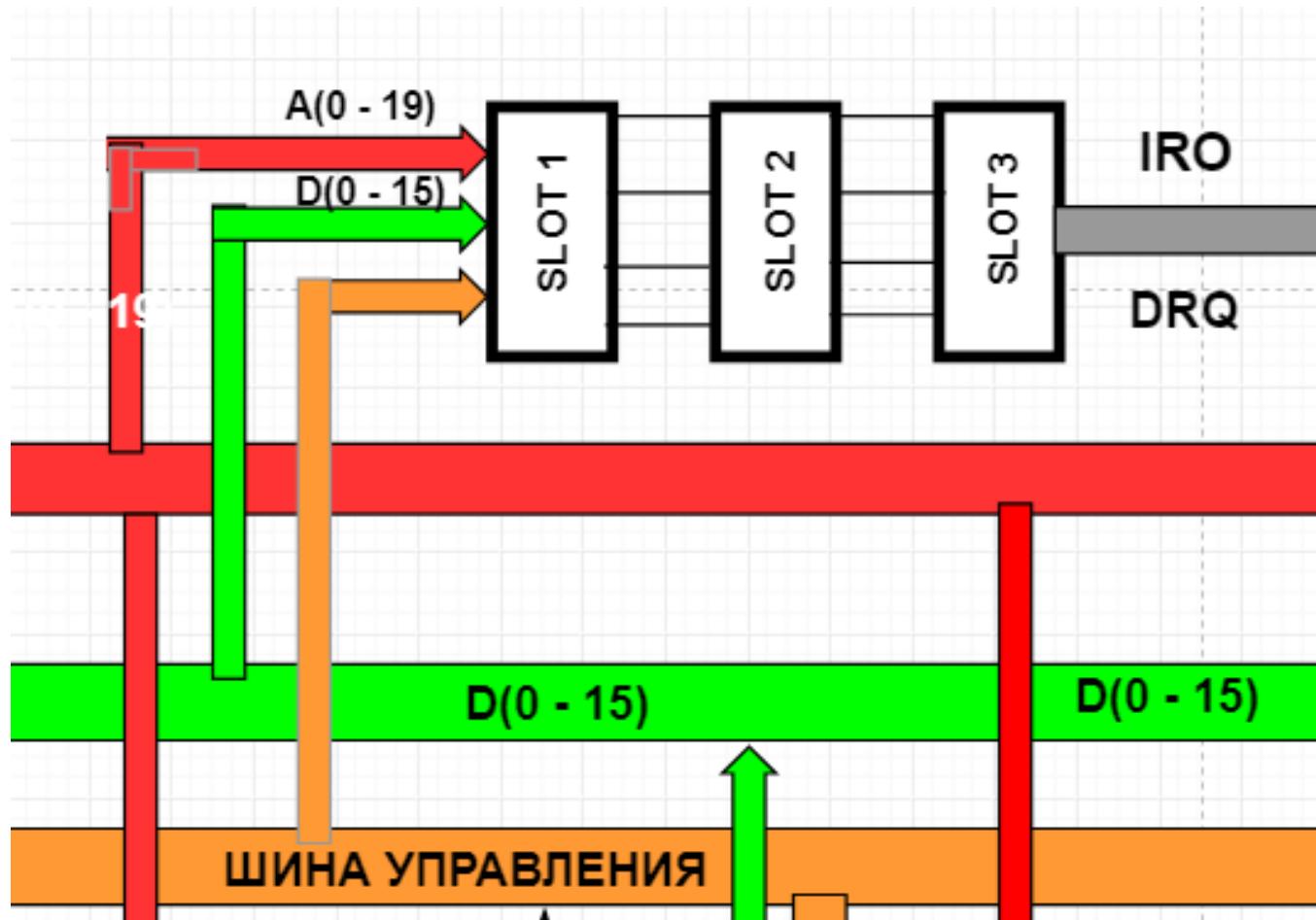
USART – Universal Synchronous/Asynchronous Receiver/Transmitter

- Реализует синхронно-асинхронными канал последовательной связи для подключения различных устройств (например, модемов).
- В некоторых моделях использовался для подключения мышки и других портов.
- В настоящее время в архитектуре ПК заменен USB.
- Широко используется во встроенных системах в силу своей простоты.

# ROM, RAM



# Слоты расширения



# Структура шина ISA

Industry Standard Architecture

