

Методы адресации

Наиболее простым способом указания адреса является включение его в состав команды:

Это так называемая ***абсолютная*** адресация.

Для использования данного способа разрядность команды должна быть больше, чем разрядность адреса, поэтому во многих архитектурах для указания полного адреса приходится использовать дополнительные машинные слова.

КОП | АДРЕС

КОП

АДР1

АДР2

Методы адресации

Необходимость различных методов адресации

организация обращения к ячейкам памяти, адреса которых вычисляются при выполнении программы

реализация массивов, стеков, буферов и других структур данных

минимизация числа разрядов, необходимых для указания адреса

обеспечение позиционной независимости программы

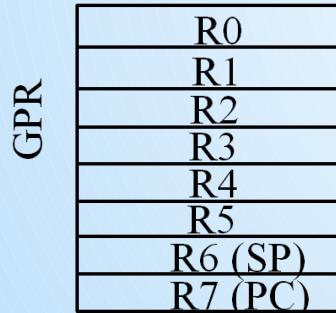
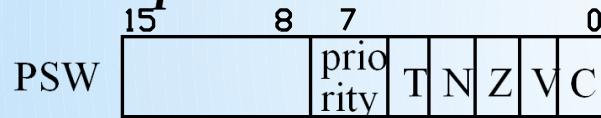
Возможность реализации различных методов адресации базируется на наличии регистров, для указания адреса которых необходимы всего 3-5 разрядов.

Архитектура процессоров DEC (*Digital Equipment Corp*)

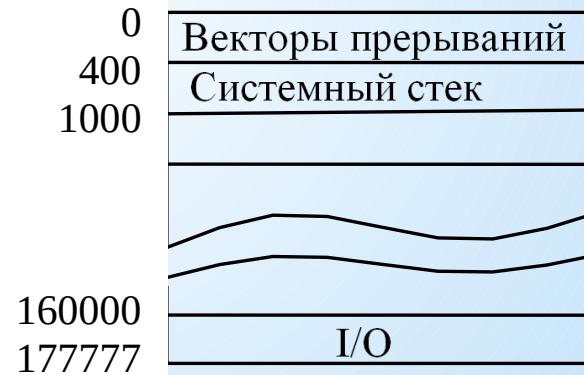
МиниЭВМ PDP-11 и микроЭВМ LSI-11 - примеры регулярной непротиворечивой архитектуры фон-Неймановского типа с CISC системой команд

- 16-разрядный процессор;
- единый набор команд для операций I/O
- форматы команд: безадресные, 1-, 2 – адресные;
- наличие регистров общего назначения

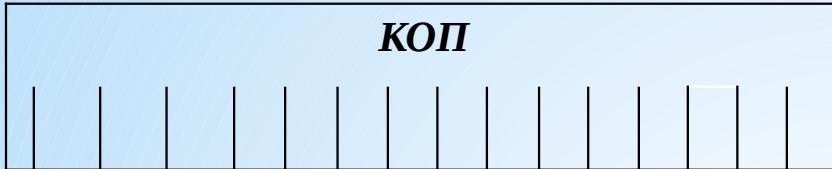
Регистровая модель:



Модель памяти



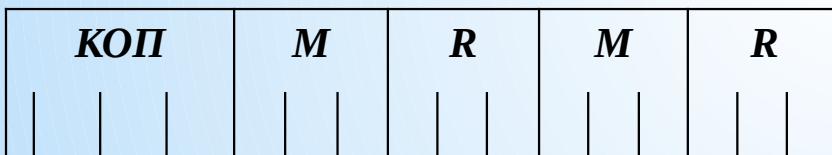
*Форматы команд:



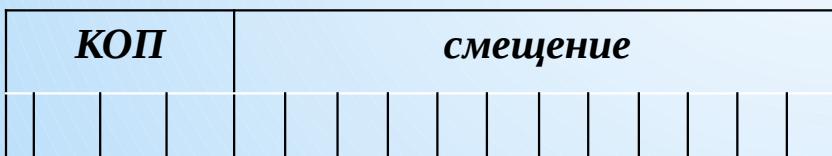
1 безадресная



2 одноадресная



3 двухадресная

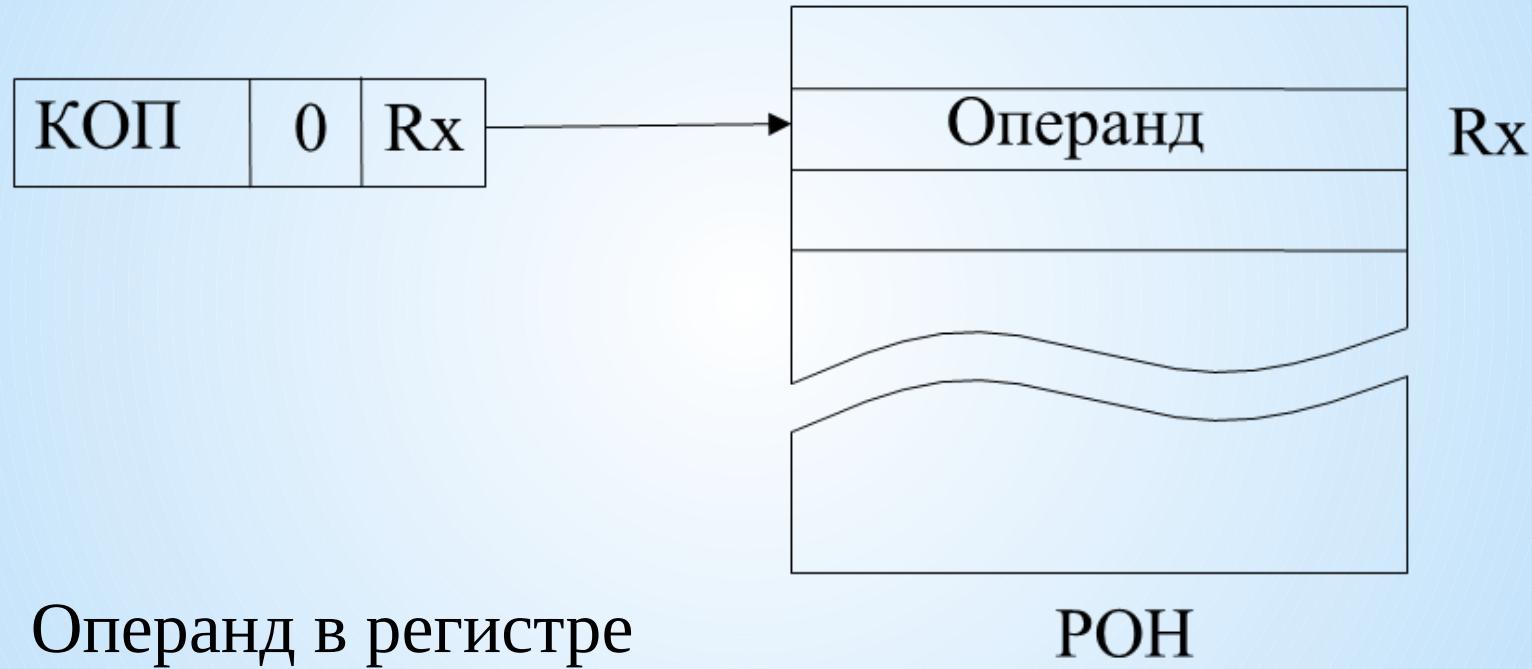


4 команда перехода

Классификация методов адресации

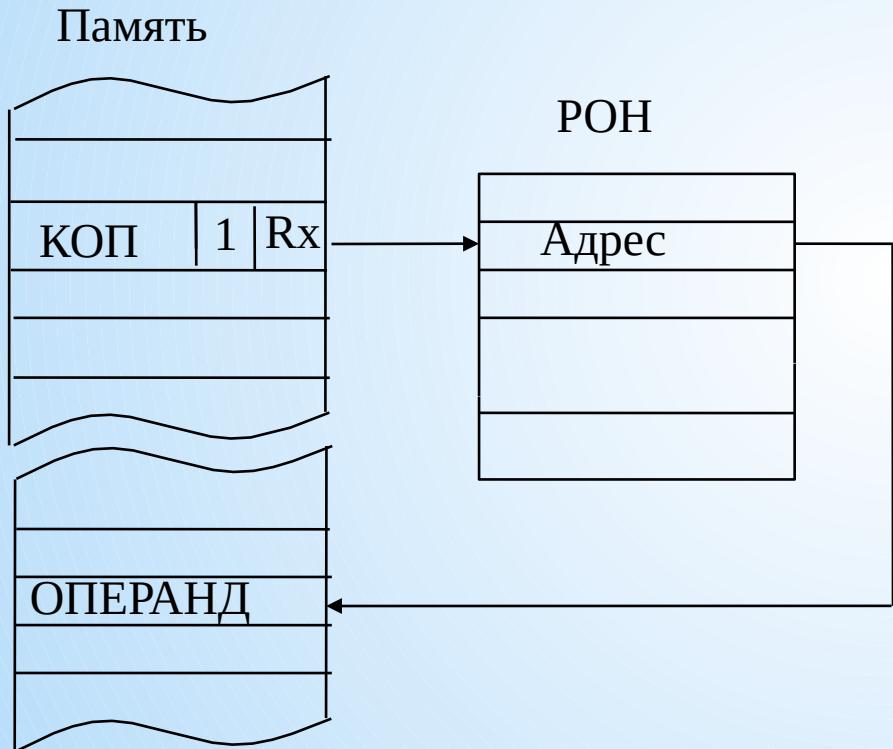
	Прямая	Косвенная		
Однокомпонентная	Регистровая $M=0 \ Rx$			Косвенно-регистровая $M=1 \ (Rx) @Rx$
	Автоинкрементная $M=2 \ (Rx) +$	Непосредственная $MR=27 \ #E$	Абсолютная $MR=37 \ @#E$	Косвенно-автоинкрементная $M=3 \ @(Rx) +$
	Автодекрементная $M=4 \ -(Rx)$			Косвенно-автодекрементная $M=5 \ @-(Rx)$
Многокомпонентная	Индексная $M=6 \ E(R)$	Относительная $MR=67 \ E$	Косвенно-относительная $MR=77 \ @E$	Косвенно-индексная $M=7 \ E(Rx)$
	С использованием счётчика команд			

*Прямая регистровая адресация ($M = 0$)



Косвенные методы адресации

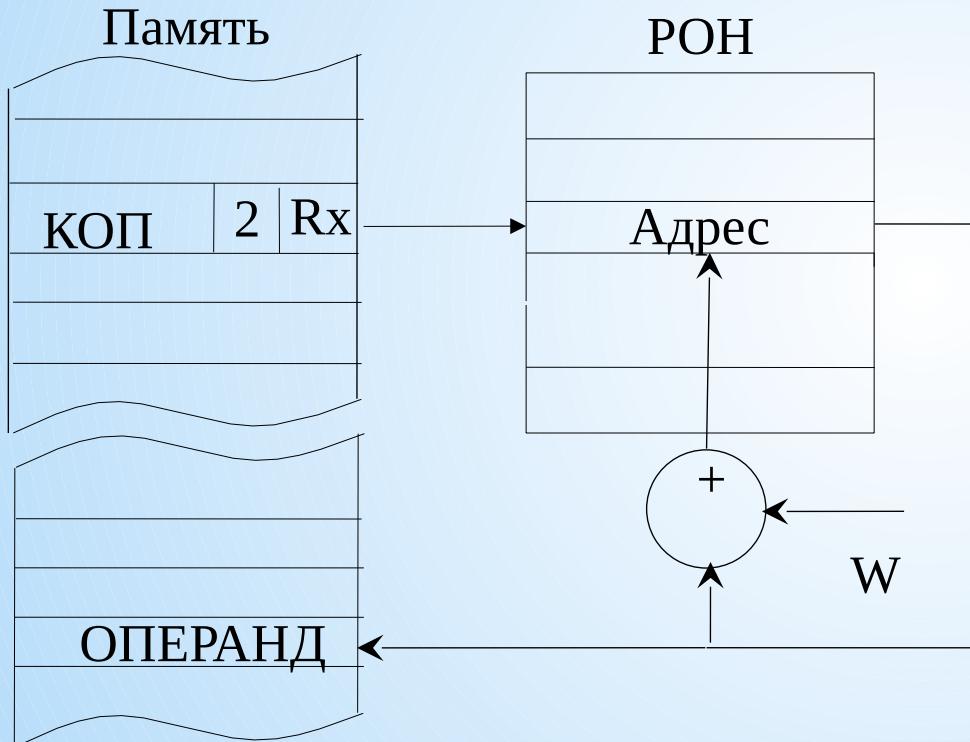
Косвенная ($M = 1$)



Адрес операнда, размещённого в памяти - в регистре, который указан в поле адреса команды

Косвенные методы адресации

Поставтоинкрементная ($M = 2$)

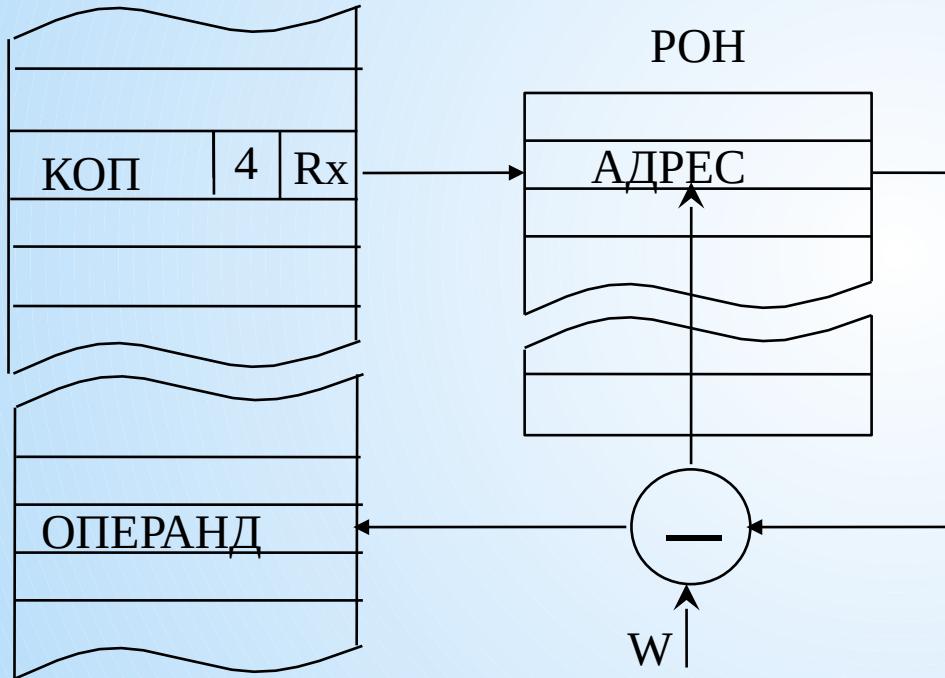


Адрес операнда в регистре, который указан в поле адреса команды

После использования содержимое регистра увеличивается на размер операнда W (в байтах или словах, в зависимости от способа адресации памяти – побайтный или пословный)

Косвенные методы адресации

Предавтодекрементная ($M = 4$)

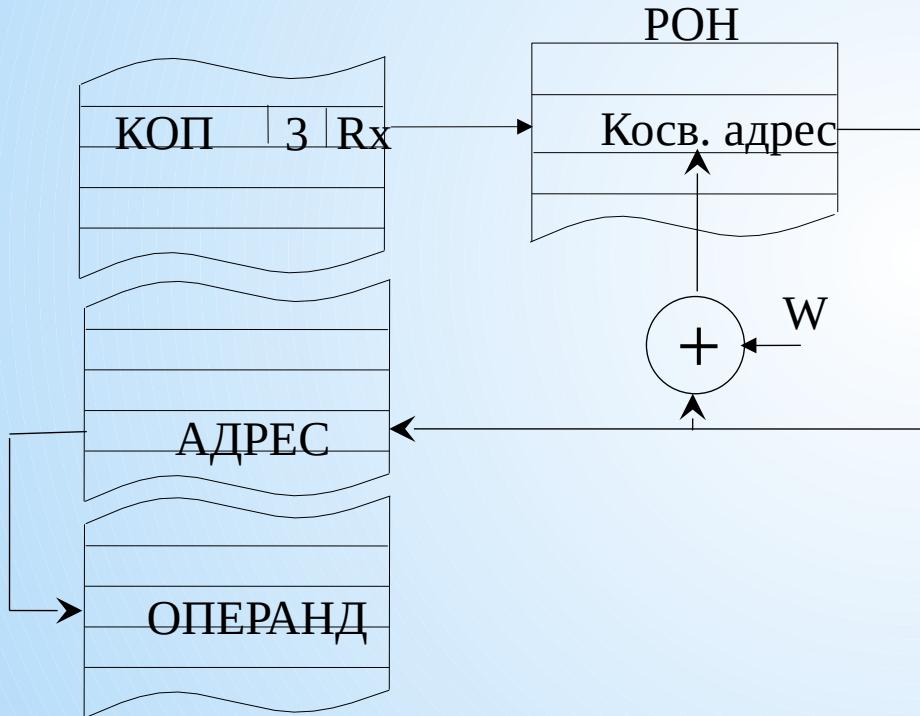


Адрес операнда в регистре, который указан в поле адреса команды.

Перед использованием содержимое регистра уменьшается на размер операнда W (в байтах или словах, в зависимости от способа адресации памяти – побайтный или пословный)

Косвенные методы адресации

Косвенно-автоинкрементная ($M = 3$)

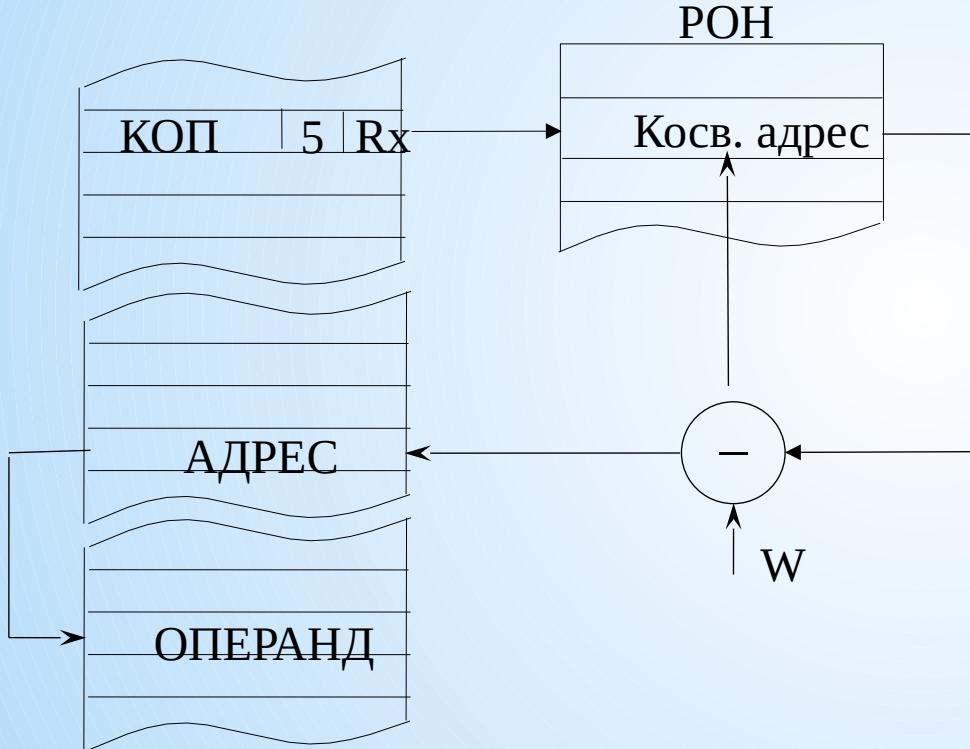


Адрес операнда в ячейке памяти, на которую ссылается регистр, указанный в поле адреса команды.

После использования содержимое регистра увеличивается на размер операнда W (в байтах или словах, в зависимости от способа адресации памяти – побайтный или пословный)

Косвенные методы адресации

Косвенно-автодекрементная ($M = 5$)



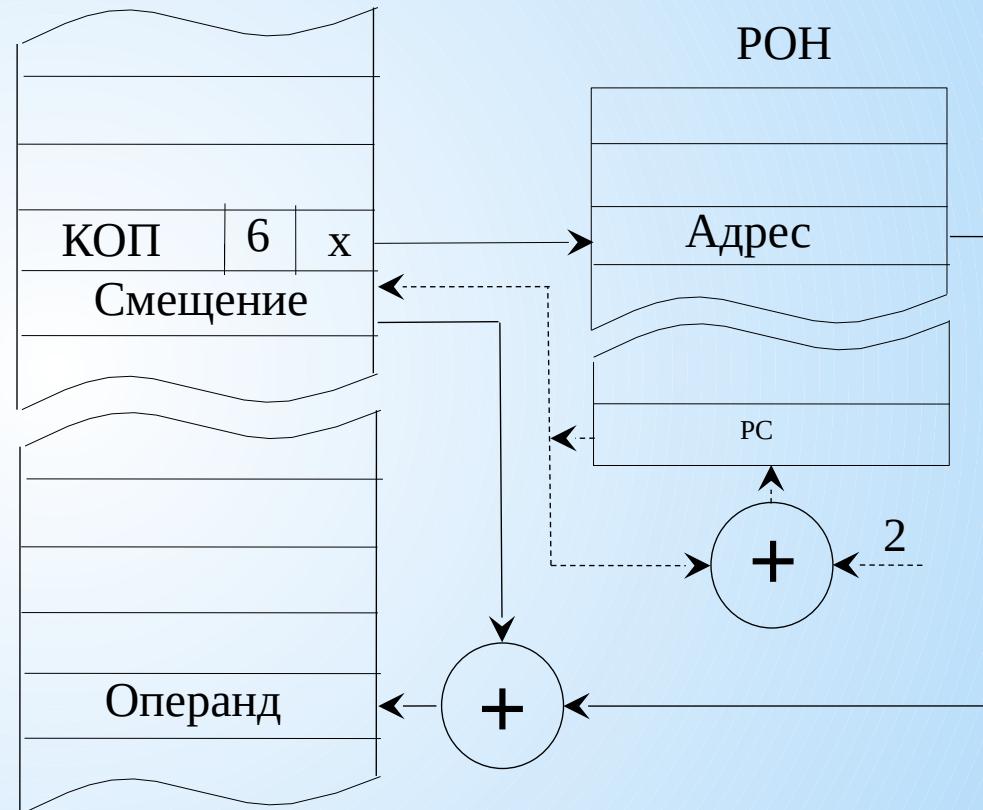
Адрес операнда в ячейке памяти, на которую ссылается регистр, указанный в поле адреса команды.

Перед использованием содержимое регистра уменьшается на размер операнда W (в байтах или словах, в зависимости от способа адресации памяти – побайтный или пословный)

*Многокомпонентные методы адресации

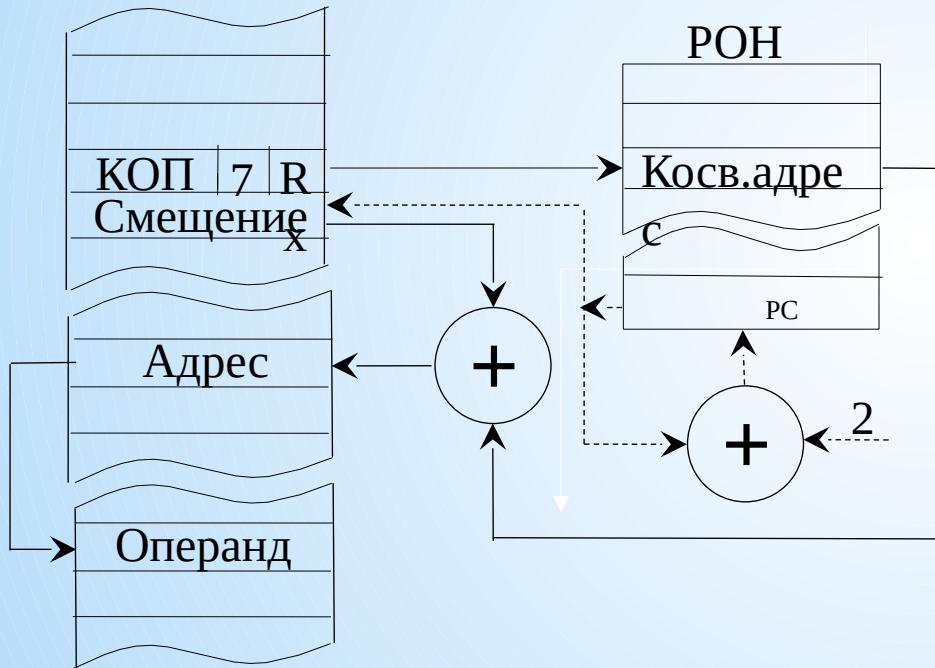
Индексный (M=6)

Исполнительный адрес вычисляется как сумма содержимого регистра, указанного в поле адреса, и смещения, находящегося во втором слове команды, с соответствующей коррекцией счётчика команд



*Многокомпонентные методы адресации

Косвенно-индексный (M=7)



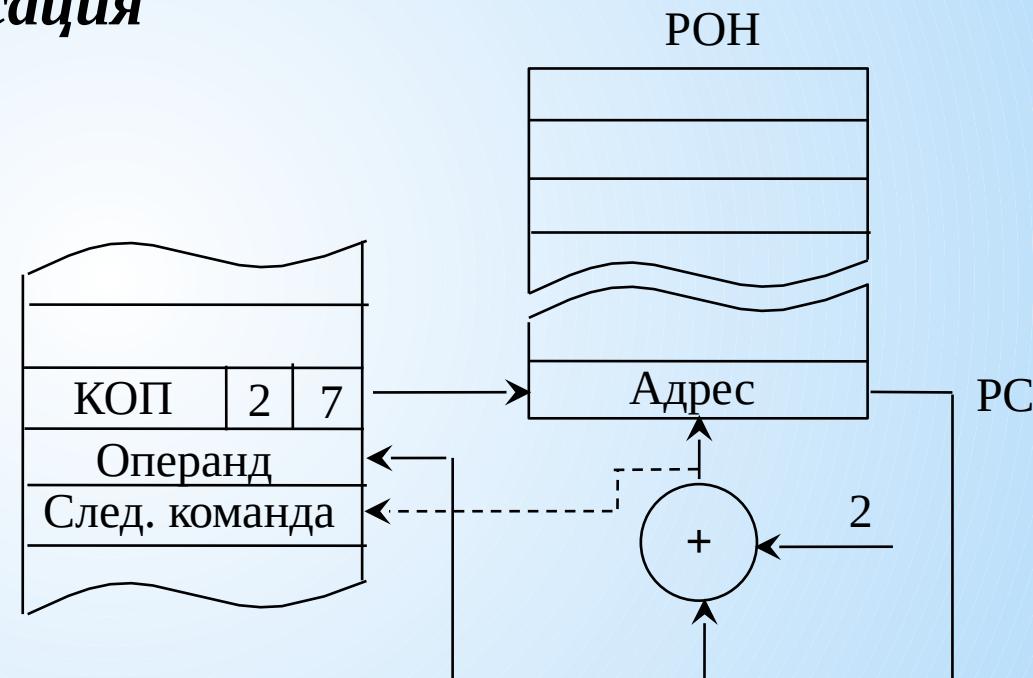
Исполнительный адрес извлекается из ячейки памяти, адрес которой равен сумме содержимого регистра, указанного в поле адреса, и смещения, находящегося во втором слове команды, с соответствующей коррекцией счётчика команд

Методы адресации с использованием счётчика команд

Непосредственная адресация (MR=27)

*Автоинкрементная адресация
по счётчику команд.*

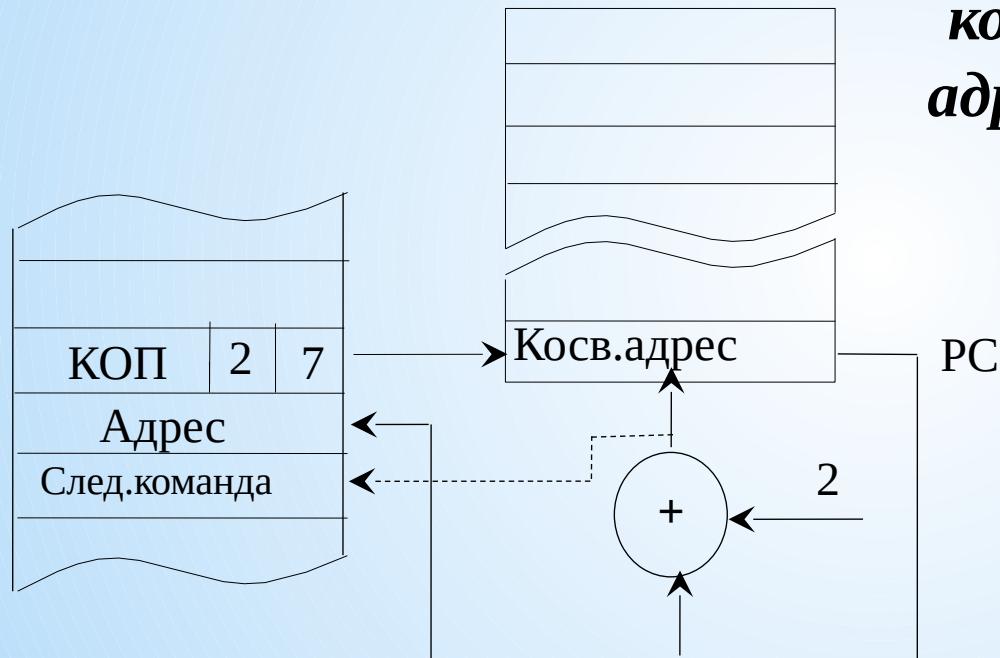
Размер команды составляет два машинных слова. После выполнения инкремента счётчик команд содержит корректный адрес следующей команды.



Методы адресации с использованием счётчика команд

Абсолютная адресация ($MR=37$) –

РОН

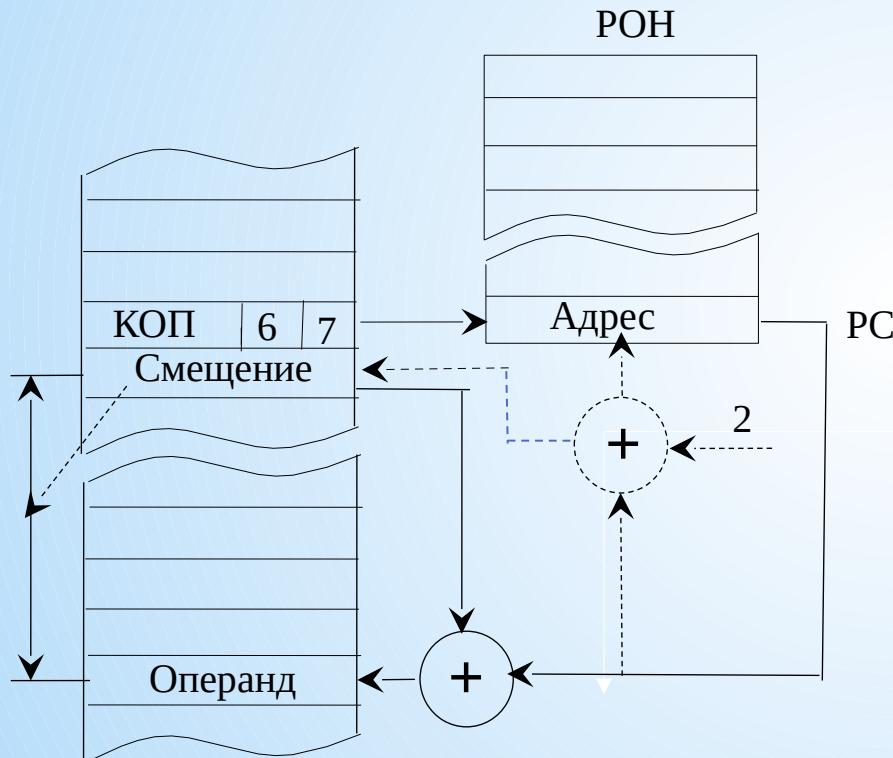


*косвенно-автоинкрементная
адресация по счётчику команд.*

Адрес операнда в
дополнительном
(в данном случае
втором) слове
команды

Методы адресации с использованием счётчика команд

Относительная (MR=67)

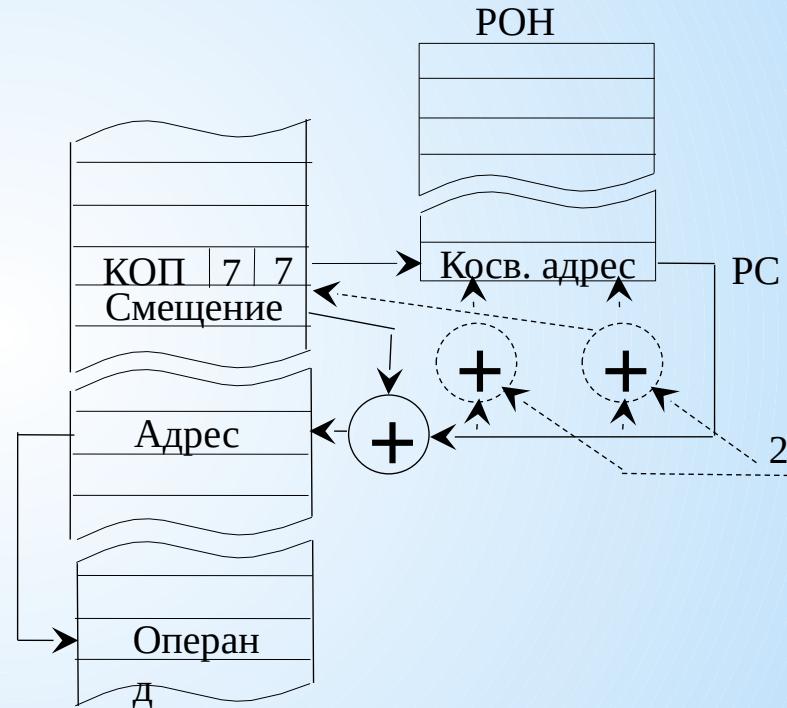


Адрес операнда вычисляется как сумма адреса текущей команды и смещения, помещаемого во втором слове команды, с двукратной коррекцией содержимого счётчика команд – первой для обращения к смещению, и второй – для выборки следующей команды

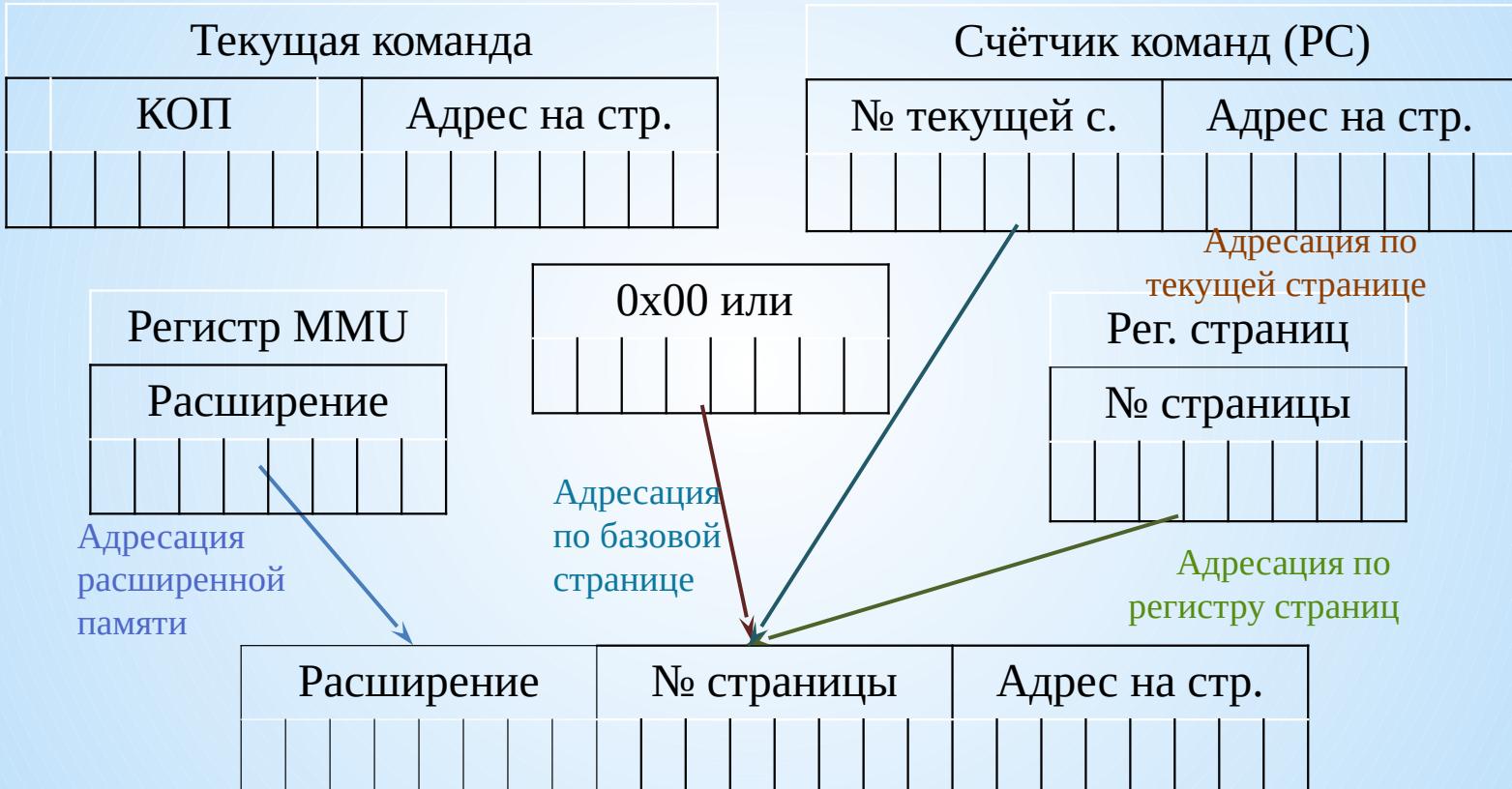
Методы адресации с использованием счётчика команд

Косвенно-относительная ($MR=77$)

Адрес операнда выбирается из ячейки, адрес которой вычисляется как сумма адреса текущей команды и смещения, помещаемого во втором слове команды, с двукратной коррекцией содержимого счётчика команд – первой для обращения к смещению, и второй – для выборки следующей команды



Страницчная адресация



Программное обеспечение (ПО) МП систем

ПО – совокупность программных средств, т.е. набор программ и инструкций по их применению, обеспечивающих возможность эксплуатации вычислительных средств, подготовки программ для них и функционирование вычислительной системы.



*Программное обеспечение разработки

Резидентное ПО – предназначено для разработки прикладного ПО той же вычислительной системы, на которой оно будет использоваться. Для этого такая ВС должна иметь средства взаимодействия с программистом, что не всегда возможно, особенно для встраиваемых вычислительных средств

Кросс ПО – ПО, предназначено для разработки программ для МПС, но реализованное на вычислительной машине с другой архитектурой (более удобной для разработки)

Функциональные модули ПО разработки

Кросс- и резидентное ПО выполняют аналогичные функции и содержат следующий набор функциональных модулей:

1. Трансляторы, компиляторы
2. Загрузчики
 - а) Редакторы связей
 - б) Объединяющие программы
 - в) Размещающие программы
3. Библиотекари
4. Отладчики
 - а) Кросовые – симуляторы
 - б) Резидентные – внутрисхемные эмуляторы

Интегрированная среда разработки

В современном ПО разработки весь комплекс перечисленных модулей объединяется в интегрированную среду разработки (ИСР, IDE – Integrated Development Environment), предоставляющую пользователю возможность выполнить весь цикл разработки и отладки ПО без обращения к сторонним программным средствам

Отладка МПС

Отладчики могут быть резидентными и кроссовыми. Процесс отладки на кроссовой системе осуществляют симуляторы, на резидентной – эмуляторы.

Резидентная отладка осуществляется с помощью внутрисхемного эмулятора: раньше это была аппаратная модель конкретного процессора с дополнительным отладочным интерфейсом, обеспечивающим доступ к внутренним ресурсам микроконтроллера, также состоянию его выводов, которая устанавливалась в целевую систему вместо реального микроконтроллера.

В современной МП технике отладочный интерфейс является составной частью каждого микроконтроллера, а внутрисхемный эмулятор в сущности является преобразователем интерфейсов: ПК (обычно это USB) – микроконтроллер (чаще всего JTAG – последовательный синхронный радиальный дуплексный интерфейс).

Отладка МПС

В процессе отладки с помощью симуляторов и эмуляторов используются следующие приёмы:

- ✓ задание точек останова программы и условия останова,
- ✓ задание пошагового и циклического выполнения программы,
- ✓ отображение состояния ресурсов вычислительной системы.

Кроме того, могут применяться специальные приборы:

- ✓ Логические анализаторы
- ✓ Генераторы слов
- ✓ Сигнатурные анализаторы и др.

* Номенклатура и семейства МК

Архитектура

Разрядность

Быстродействие

Энергопотребление

Периферийные
устройства

И др.

Функциональные возможности МК

В число периферийных устройств обычно входят

- параллельные порты ввода/вывода,
- каналы последовательного интерфейса,
- таймеры/счетчики,
- контроллер прерываний и т.п.

Кроме этого, многие микроконтроллеры содержат

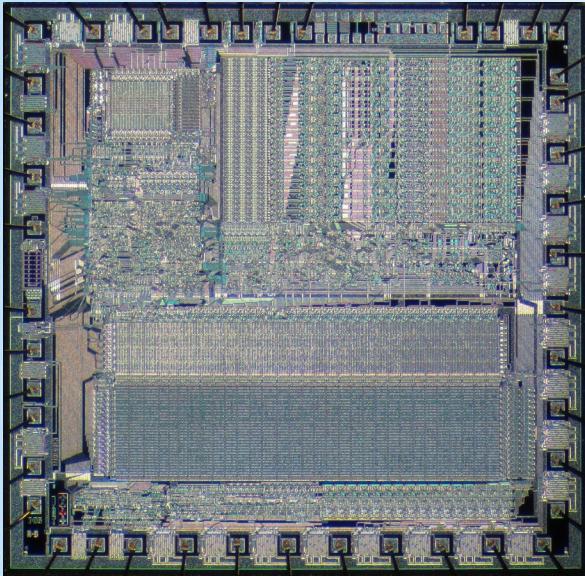
- дополнительные запоминающие устройства
- многофункциональные таймерные модули,
- многоканальные АЦП и ЦАП,
- широтно-импульсные модуляторы,
- цифровые компараторы,
- устройства для подключения жидкокристаллических и электролюминесцентных индикаторов и другие устройства

* Производители МК

Intel	Motorola
Texas Instruments	Renesas Electronics
STMicroelectronics	Samsung
AMD	Philips
NEC	Silicon Laboratories
Microchip + Atmel	Zilog
Dallas Semiconductor	Hitachi
Mitsubishi	Fujitsu
Toshiba	Holtek
Ubi.com (Scenix)	Infineon (Siemens)

и ряд других американских, европейских и южно-азиатских фирм.

* Отечественные разработки МК



* Классификация и производители МК

Наиболее распространенные МК

Из 8-разрядных МК на сегодняшний день наиболее популярными являются:

- семейства, совместимые с микропроцессорным ядром Intel MCS-51;
 - микроконтроллеры семейства AVR фирмы Atmel;
 - PIC-контроллеры фирмы Microchip
-
- Всё более заметную конкуренцию 8- и 16- разрядным МК составляют 32-разрядные семейства, наиболее известными из которых являются МК с архитектурой ARM, а также AVR-32 фирмы Atmel, микроконтроллеры производства японской фирмы Renesas Electronics и др.

* Встраиваемые вычислительные средства

* **Встраиваемая система (Embedded System)** — специализированная компьютерная вычислительная система, которая является модулем, блоком устройства (частью устройства) и встраивается непосредственно в него или входит в состав устройств, объединённых в единую сеть. Встраиваемые системы служат для управления устройствами, сбора данных, контроля и мониторинга.

* К встраиваемым системам предъявляется много требований: это способность работать в тяжёлых условиях, в условиях вибраций и ударов, при перепадах температур, в условиях повышенной нагрузки, в необслуживаемом режиме, поддерживать специализированные интерфейсы ввода-вывода, иметь долгий срок службы.

*COTS: Commercial Off-The-Shelf

COTS – Commercial Off-The-Shelf – готовые коммерческие продукты и технологии «с полки», которые можно купить на рынке.

* В двух словах суть идеологии COTS можно выразить следующим образом: **«покупать, а не разрабатывать самому»**. Применение готовых покупных продуктов и технологий позволяет сэкономить массу времени и сил, которые в противном случае пришлось бы потратить на «изобретение велосипеда», т.е. разработку базовых модулей, интерфейсов, плат и других низкоуровневых решений, которые давным-давно разработаны другими людьми, являющимися в своём деле профессионалами.

CALS (Computer-Aided Acquisition and Lifecycle Support)

Русскоязычным аналогом CALS является равнозначный термин ИПИ (Информационная Поддержка процессов жизненного цикла Изделий). Компьютерная поддержка этапов ЖЦИ получила свое оформление в методологии и стандартах ИПИ/CALS. Согласно концептуальным положениям ИПИ/CALS, реальные бизнес-процессы отображаются на виртуальную информационную среду, в которой определение продукта представлено в виде полного электронного описания изделия, а среды его создания и среды эксплуатации – в виде систем моделирования процессов и их реализации. Все три составляющие (определение продукта, среды его создания и среды эксплуатации) не только взаимосвязаны, но и непрерывно развиваются на всем протяжении жизненного цикла продукта.

Е.И. Яблочников, Ю.Н. Фомина, А.А. Саломатина. Компьютерные технологии в жизненном цикле изделия / Учебное пособие – СПб: СПбГУ ИТМО, 2010. – 188 с.

*Интерфейсы вычислительных систем

Под **стандартным интерфейсом** понимается совокупность унифицированных аппаратных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматических системах сбора и обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости указанных элементов.

Классификационные признаки интерфейсов

1. Назначение:

Магистральные
Внешние
Системные

2. Топология связей:

- } радиальный (точка-точка),
- } магистральный,
- } комбинированный

3. Способ передачи:

- } параллельный
- } последовательный
- } параллельно-последовательный

4. Способ синхронизации:

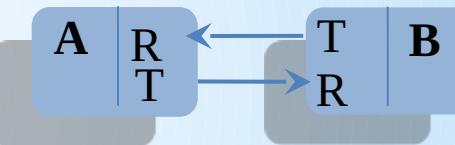
- } синхронный
- } асинхронный

5. Режим обмена:

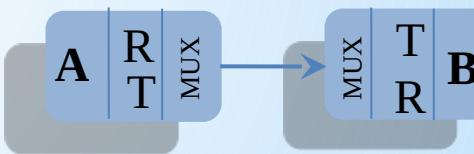
Симплексный



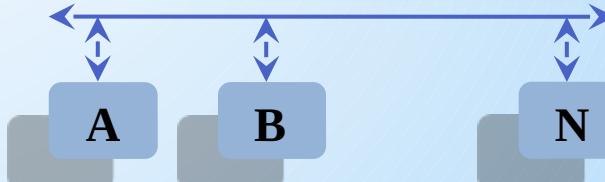
Дуплексный



Полудуплексный



Мультиплексный



Функциональное назначение интерфейсов

- * Аналоговый интерфейс\
- * Аудио интерфейс\
- * Видео интерфейс\
- * Интерфейсы БВС\
- * Интерфейсы микроконтроллеров\
- * Интерфейсы ММВС\
- * Интерфейсы ПК\
- * Интерфейсы пользователя\
- * Коммуникационные интерфейсы\
- * Мультимедийные интерфейсы\

Способы обмена информацией между устройствами вычислительной системы

1. Программно управляемые:

- Синхронный.
- Асинхронный с программной проверкой готовности.
- Асинхронный с аппаратной проверкой готовности.

*2. В режиме прямого доступа.

* Синхронный обмен данными

- * Синхронизация определяет согласование процессов взаимодействия между функциональными устройствами системы.
- * Синхронный обмен данными предполагает отсутствие ситуации неготовности обменивающихся сторон.
- * Основные достоинства:
 - Синхронный обмен – самый быстрый из всех
 - Синхронный обмен требует минимум аппаратного обеспечения.
- * Основной минус: синхронный обмен сложно (или вообще невозможно) организовать с асинхронными устройствами (т.е. с устройствами, имеющими разное время выполнения операций и/или множество производимых операций с сильно различающимися временами выполнения).
- * Два или более процесса являются синхронными, если смены состояний этих процессов взаимонезависимы и выполняются через одинаковые фиксированные интервалы времени.

* Асинхронный обмен данными

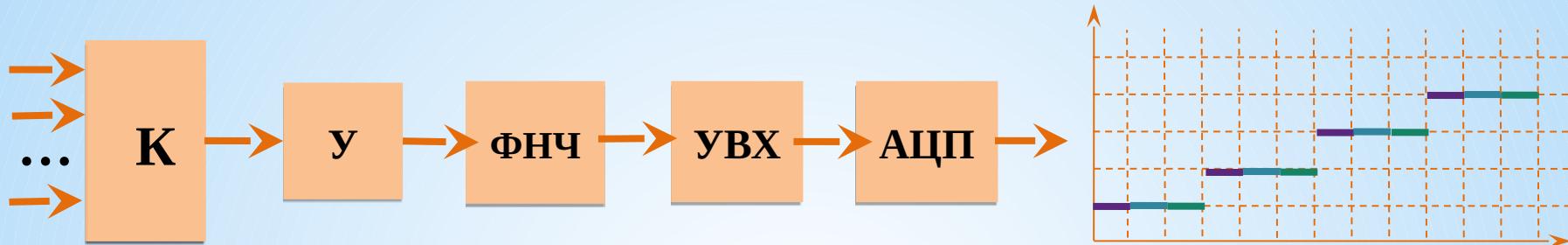
- * Два или более процесса являются асинхронными, если смены состояний этих процессов взаимозависимы, а время изменения состояния одного процесса определяется временем смены состояния другого процесса. При асинхронном взаимодействии между процессами передачи и приема существует отношение предшествования. Асинхронный принцип не означает, что синхронизация отсутствует, при асинхронном принципе период синхронизации является переменным, а при синхронном этот период постоянен и определяется частотой работы самого медленного устройства.
- * Реализация асинхронных процессов взаимодействия основывается на принципе обратной связи, которая может быть однопроводной и двухпроводной. Процессы взаимодействия интерфейсных блоков образуют иерархическую структуру. Уровни иерархии процессов взаимодействия соответствуют иерархии структурных элементов информации, передаваемой между устройствами системы. Процессам взаимодействия различных уровней иерархии соответствуют различные операции синхронизации.

* Аналоговый интерфейс

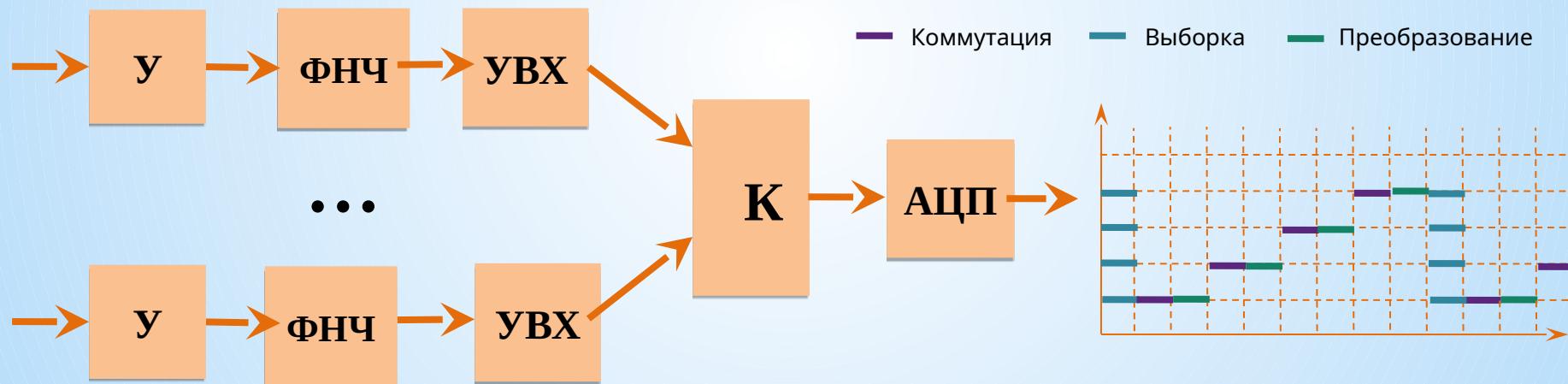


Структура тракта ввода-цифровой обработки-вывода аналоговых сигналов

Многоканальный аналоговый интерфейс



Многоканальная система сбора данных с аналоговым коммутатором на входе тракта

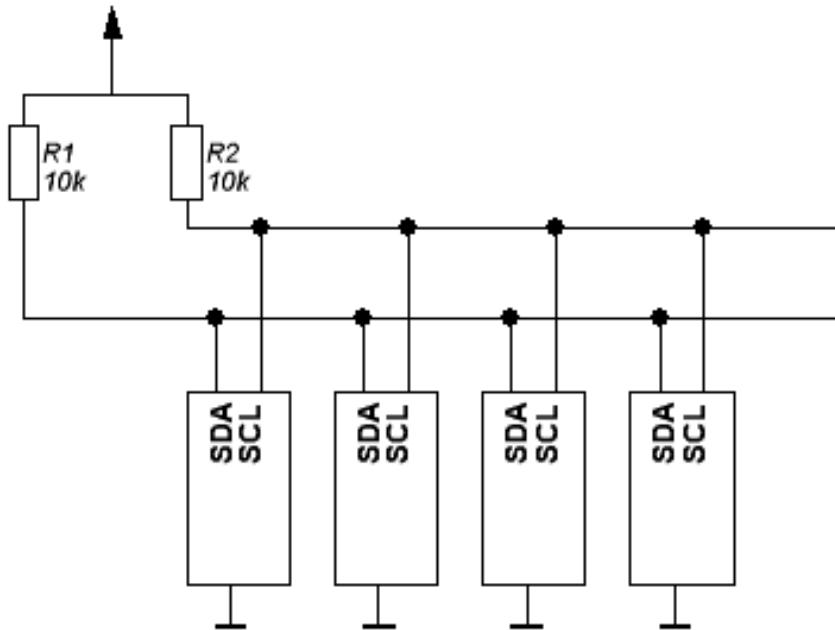


Многоканальная система сбора данных с аналоговым коммутатором на входе АЦП

*Стандартные интерфейсы

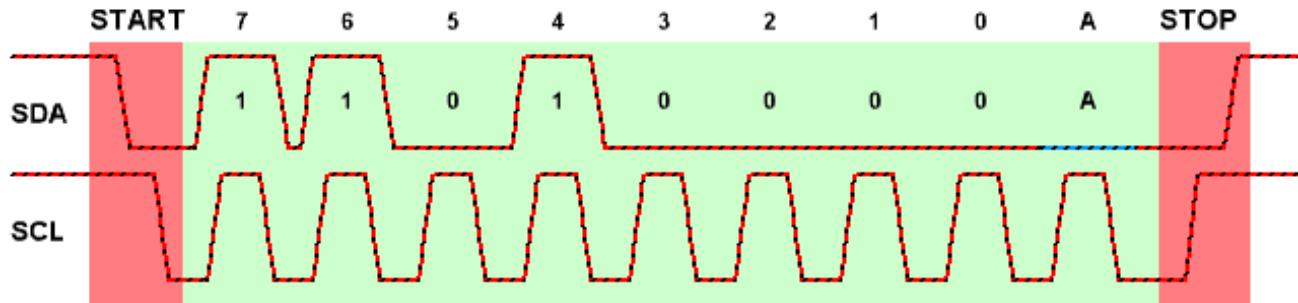
- I2C
- SPI
- UART/USART
- CAN
- SDIO
- USB
- I2S

* I2S физический уровень

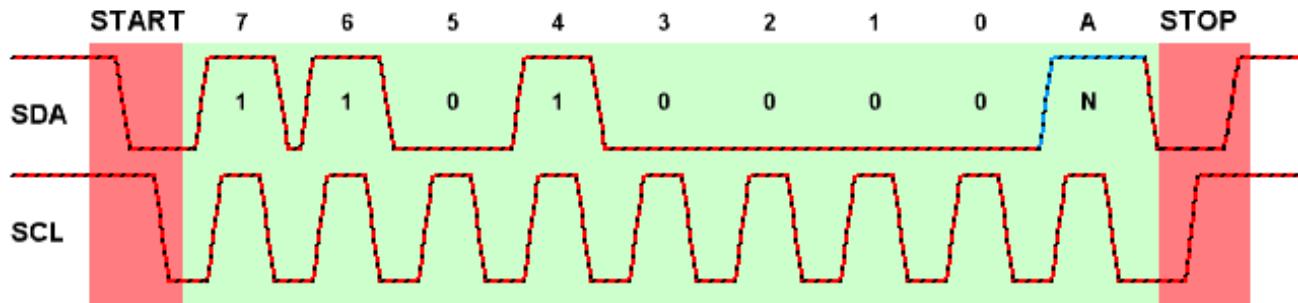


Данные передаются по двум линиям — линия данных и линия тактирования. Есть ведущий(master) и ведомый(slave), такты генерирует master. Всего на одной двупроводной шине может быть до 127 устройств. Схема подключения — **монтажное И**

Передается байт 11010000 в ответ получается бит подтверждения A



Передается байт 11010000 в ответ не получается бит подтверждения A

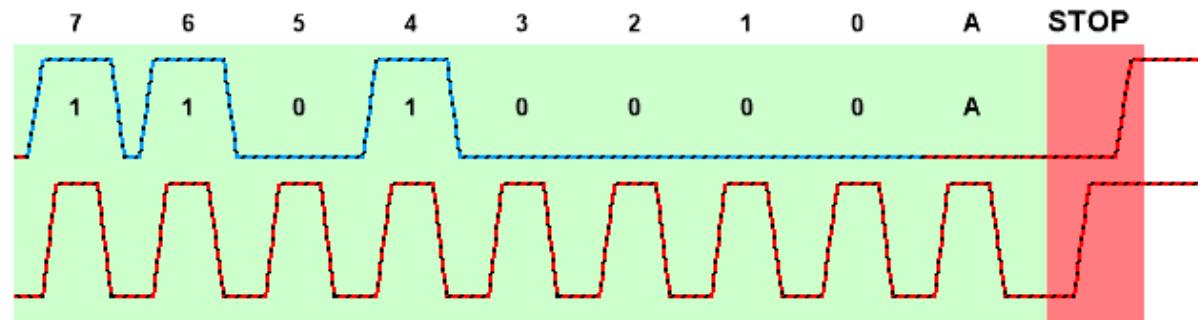


Уровень определяет ведомый

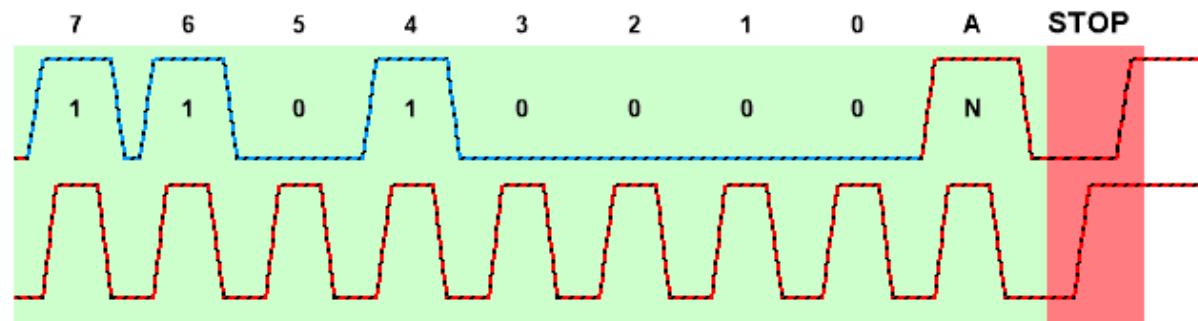
Уровень определяет ведущий

easyelectronics.ru

Принимается байт 11010000 в ответ отсылается бит подтверждения A



Принимается байт 11010000 в ответ не передается бит подтверждения A

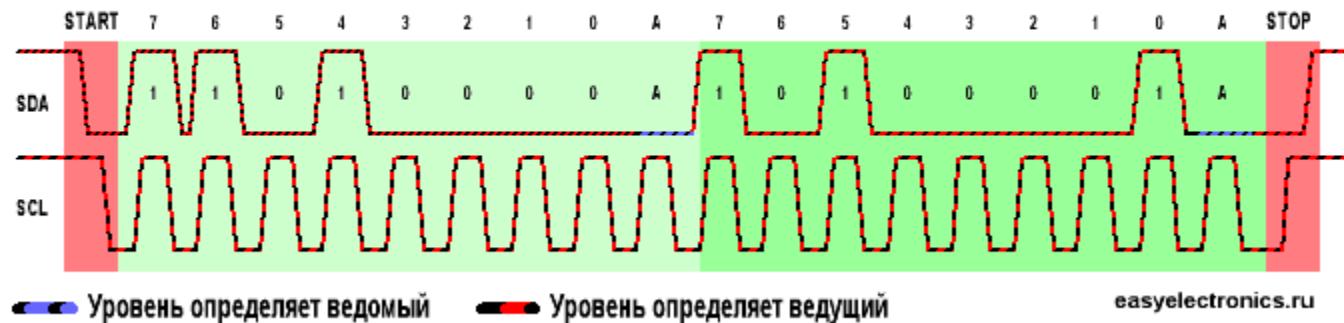


■ Уровень определяет ведомый

■ Уровень определяет ведущий

easyelectronics.ru

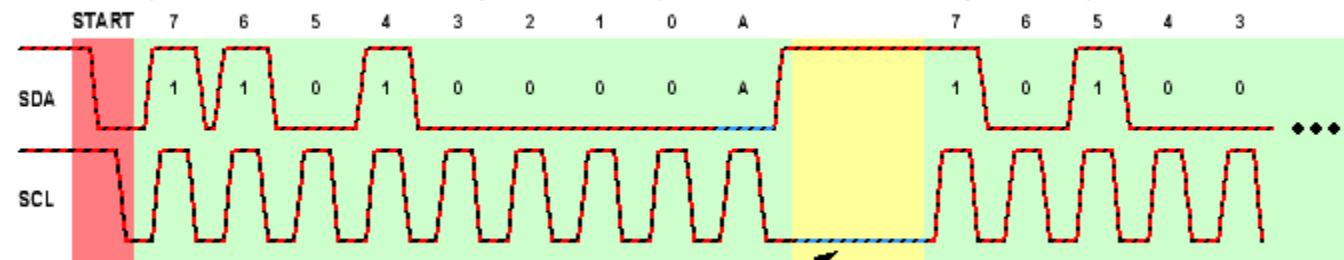
Передается два байта 11010000 и 10100001, в ответ на каждый получается бит подтверждения A



easyelectronics.ru

Передается байт 11010000 в ответ получается бит подтверждения A

Дальше передача второго байта как обычно

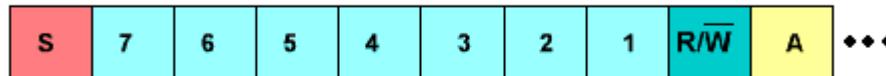


Но Slave тупит и просит подождать. Зажимая линию SCL на время занятости

easyelectronics.ru

*I2S логический уровень

Служебный пакет - запрос к Slave



Адрес Slave устройства

Что Slave должен сделать

Ответ ведомого (ACK). Или отсутствие ответа (NACK)

Последний байт! Потому NACK!!!



Старт

Slave адрес

Чтение!
Ответ Slave

Считанный байт 1

Ответ Master

Считанный байт 2

Ответ Master
Стоп

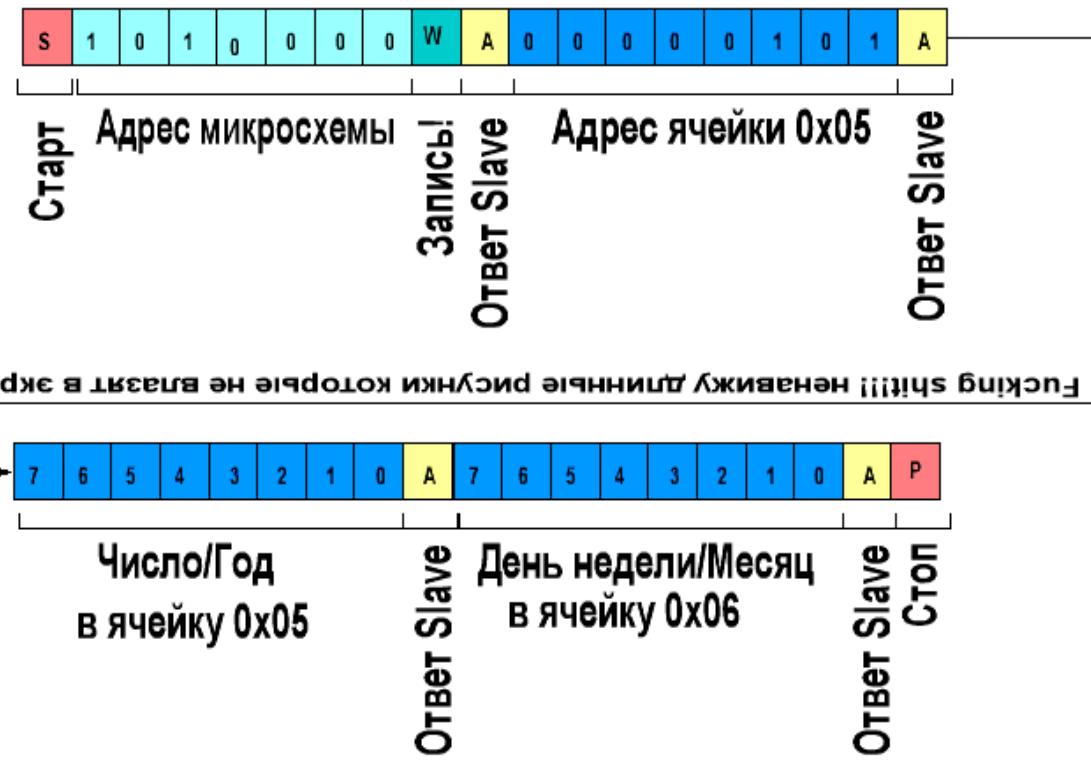
easyelectronics.ru

control/status	
hundredth of a second	
1/10 s	1/100 s
seconds	
10 s	1 s
minutes	
10 min	1 min
hours	
10 h	1 h
year/date	
10 day	1 day
weekday/month	
10 month	1 month
timer	
10 day	1 day
alarm control	
hundredth of a second	
1/10 s	1/100 s
alarm seconds	
alarm minutes	
alarm hours	
alarm date	
alarm month	
alarm timer	
free RAM	

CLOCK MODES

control/status	
D1	D0
D3	D2
D5	D4
free	
free	
free	
T1	T0
alarm control	
alarm D1	alarm D0
D3	D2
D5	D4
free	
free	
free	
alarm timer	
free RAM	

EVENT COUNTER
MRB015


easyelectronics.ru

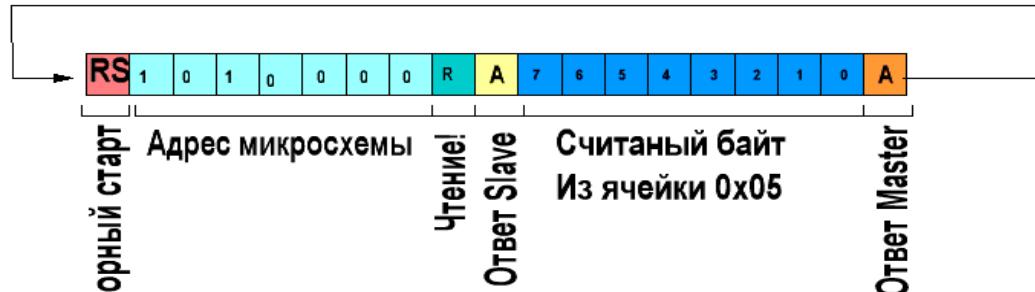
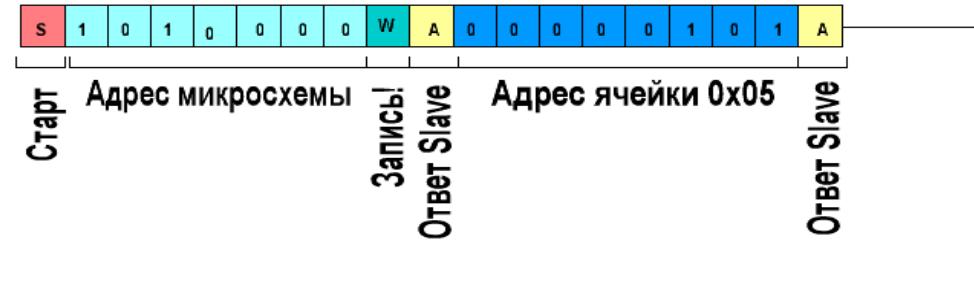
control/status
hundredth of a second 1/10 s 1/100 s
seconds
10 s 1 s
minutes
10 min 1 min
hours
10 h 1 h
year/date
10 day 1 day
weekday/month
10 month 1 month
timer
10 day 1 day
alarm control
hundredth of a second 1/10 s 1/100 s
alarm seconds
alarm minutes
alarm hours
alarm date
alarm month
alarm timer
free RAM

CLOCK MODES

control/status
D1 D0
D3 D2
D5 D4
free
free
free
T1 T0
alarm control
alarm D1 alarm D0
D3 D2
D5 D4
free
free
free
alarm timer
free RAM

EVENT COUNTER

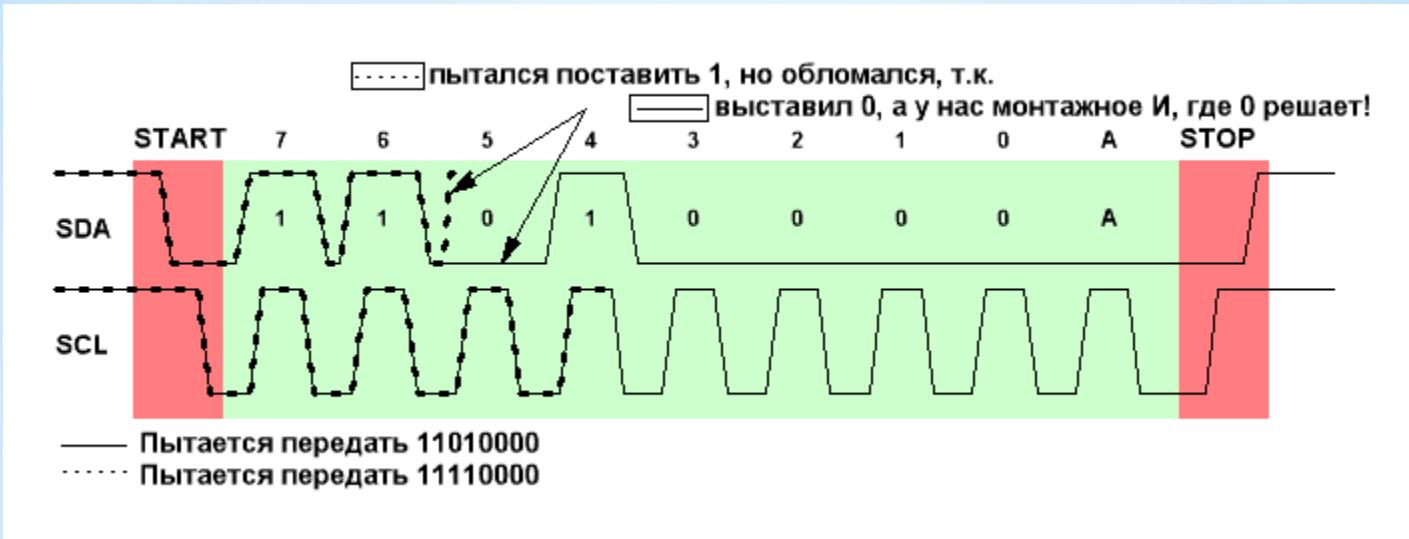
MRB015



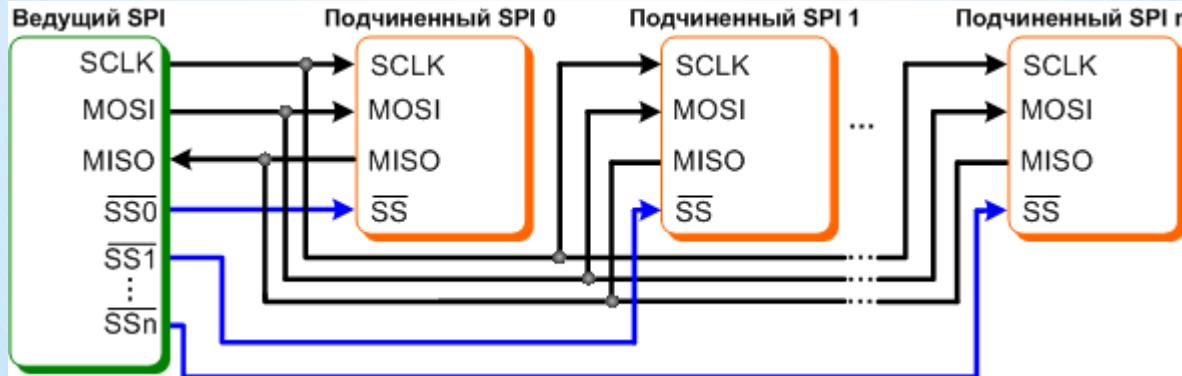
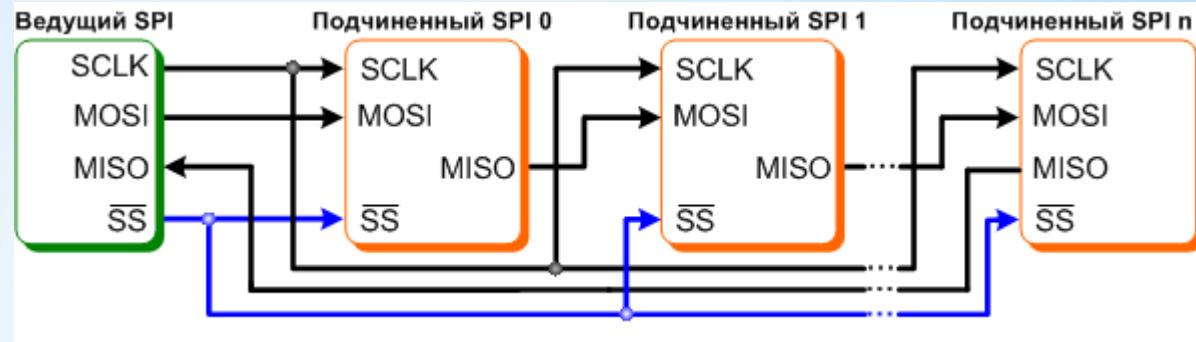
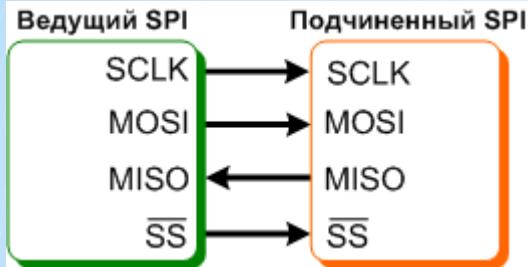
easyelectronics.ru

Считанный байт
Из ячейки 0x06
Ответ Master Стоп

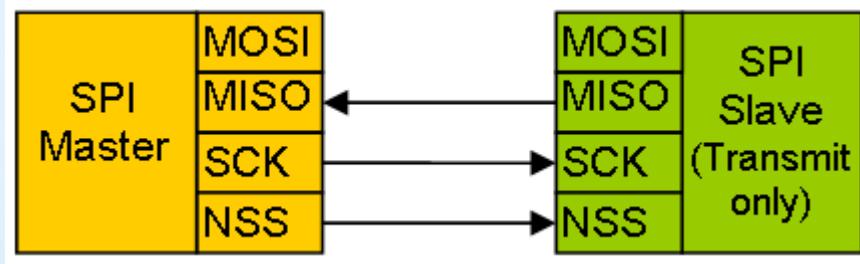
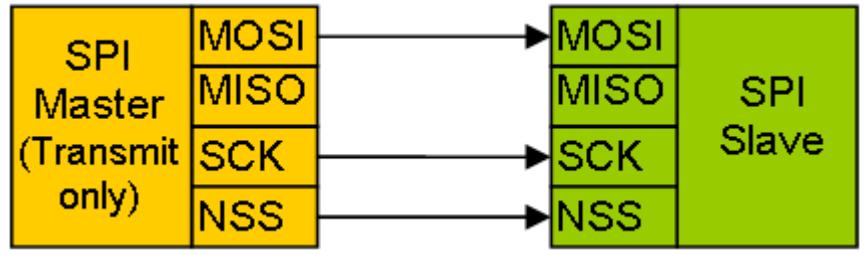
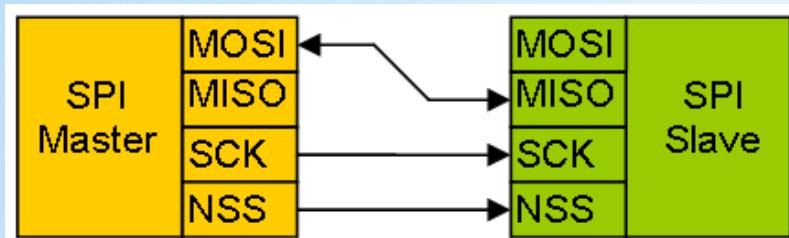
*I2S

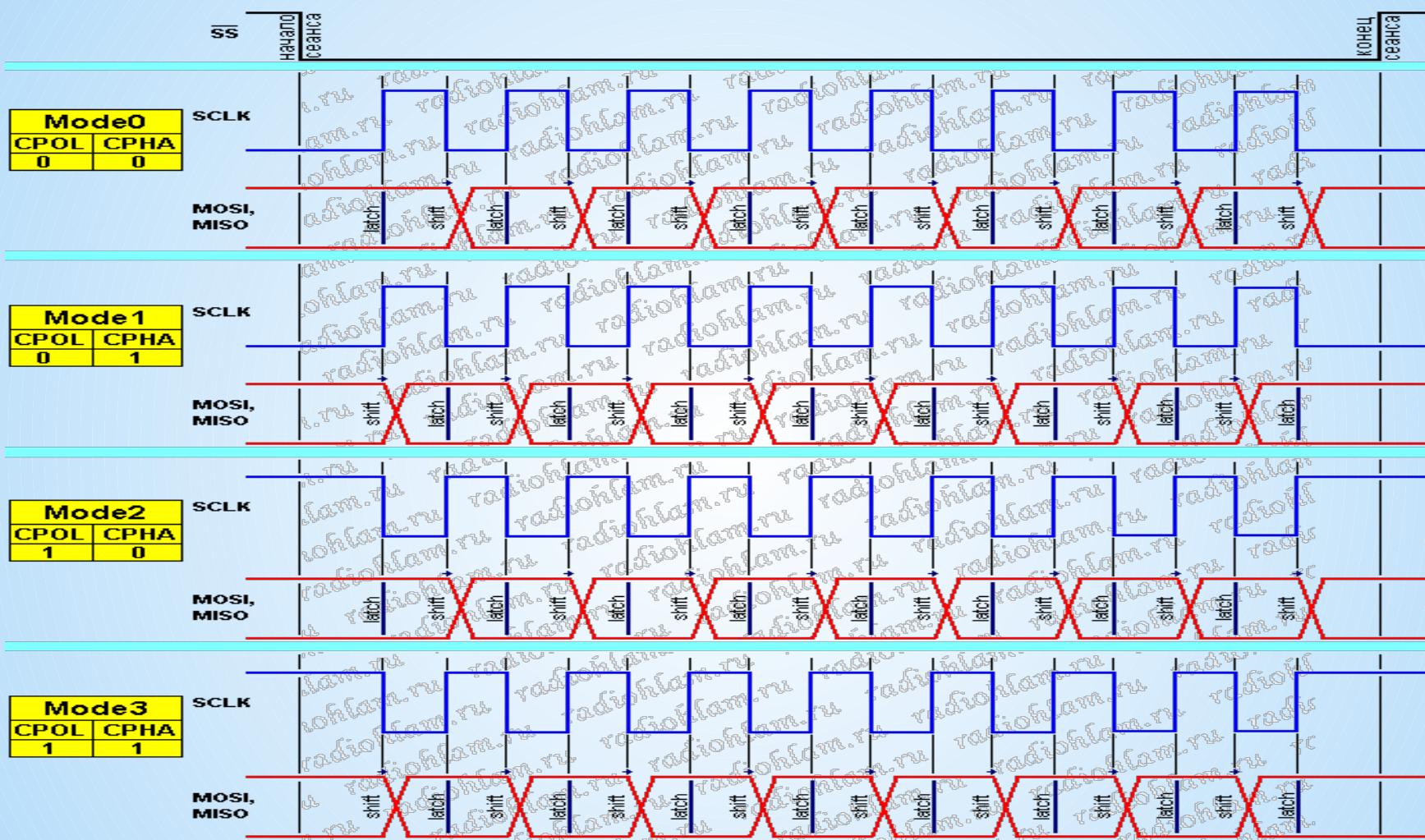


*SPI

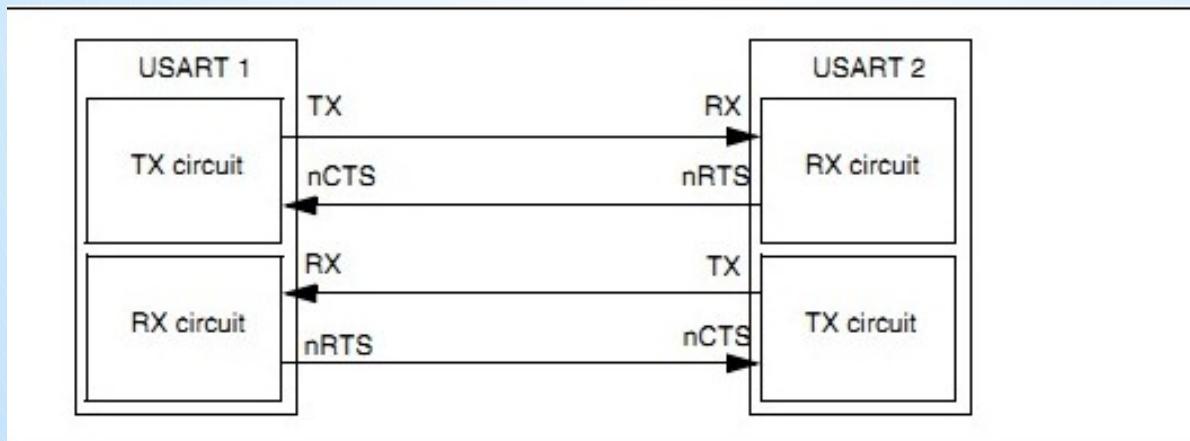
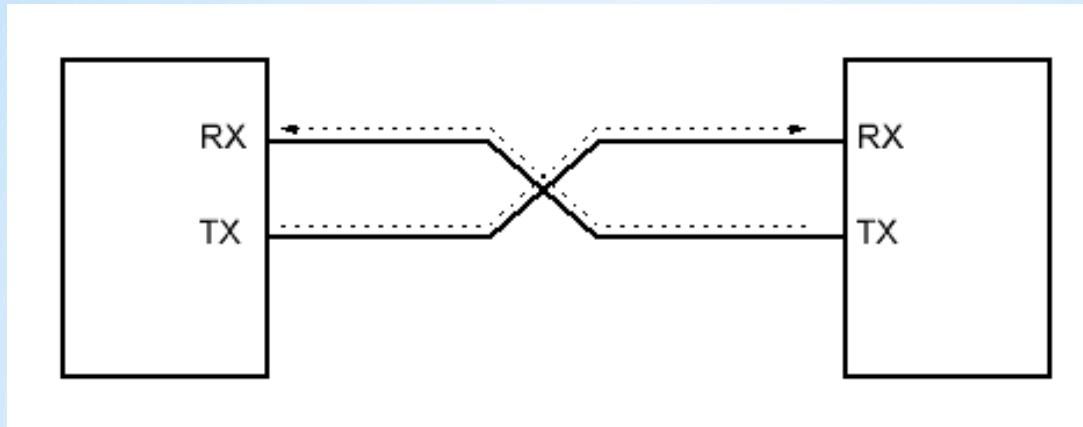


*SPI





*UART

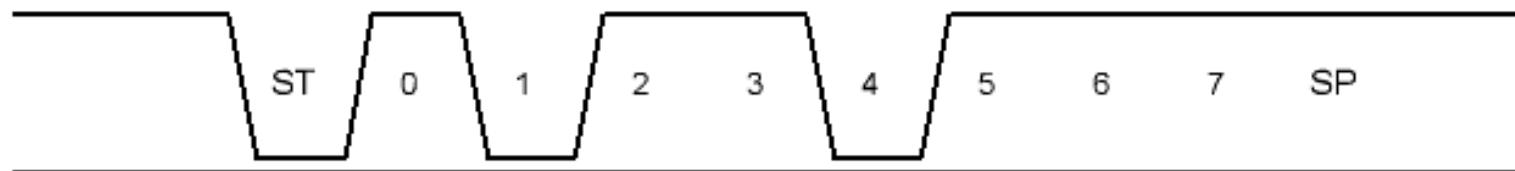
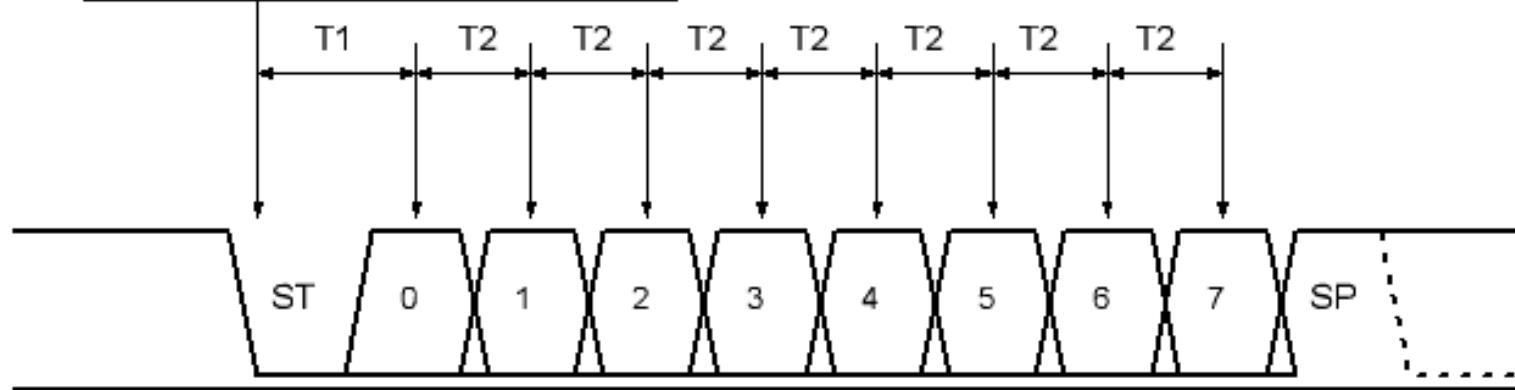


*UART

Скорость передачи (бод)	Время передачи одного бита (мкс)	Время передачи байта (мкс)
4800	208	2083
9600	104	1042
19200	52	521
38400	26	260
57600	17	174
115200	8,7	87

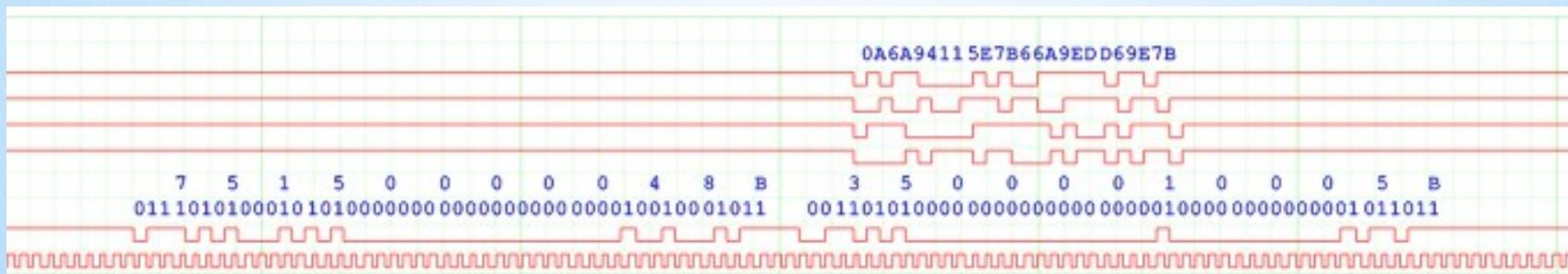
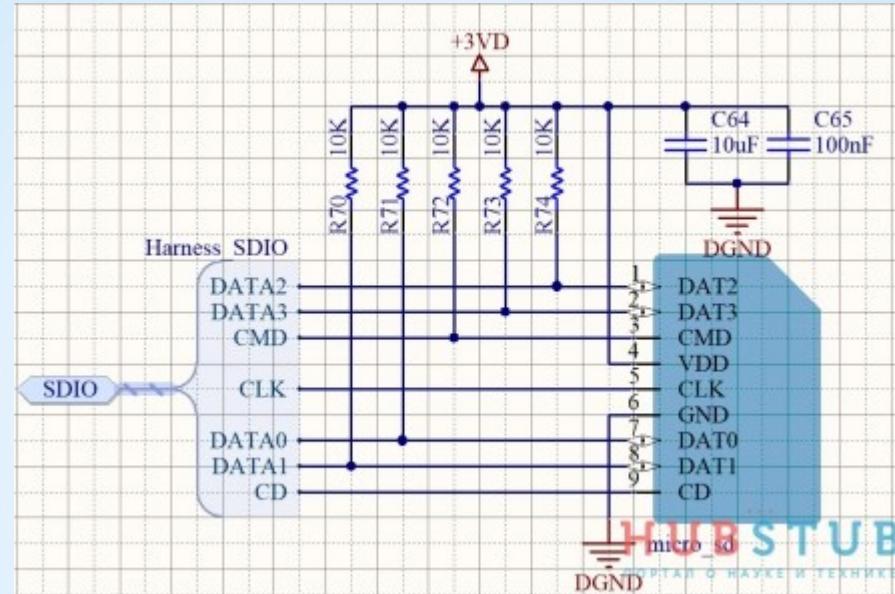
*UART

Тут мы узнали что надо принимать

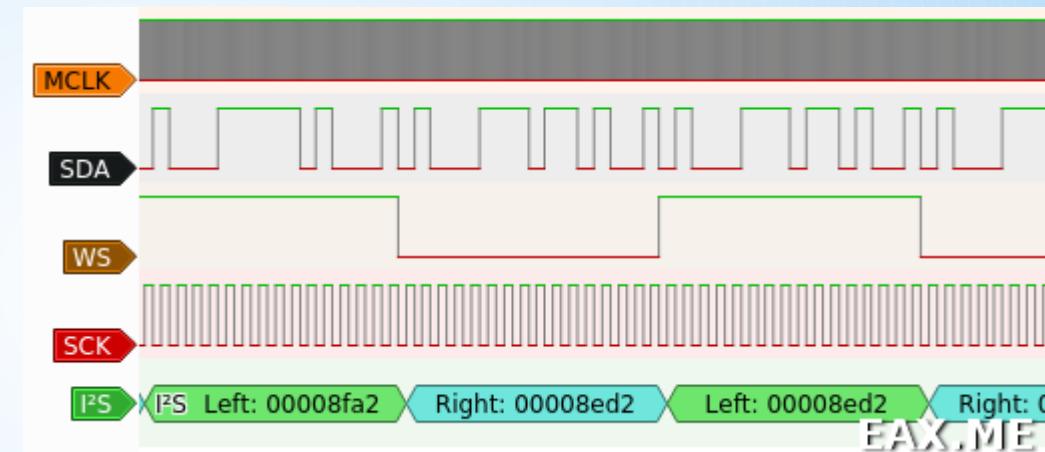
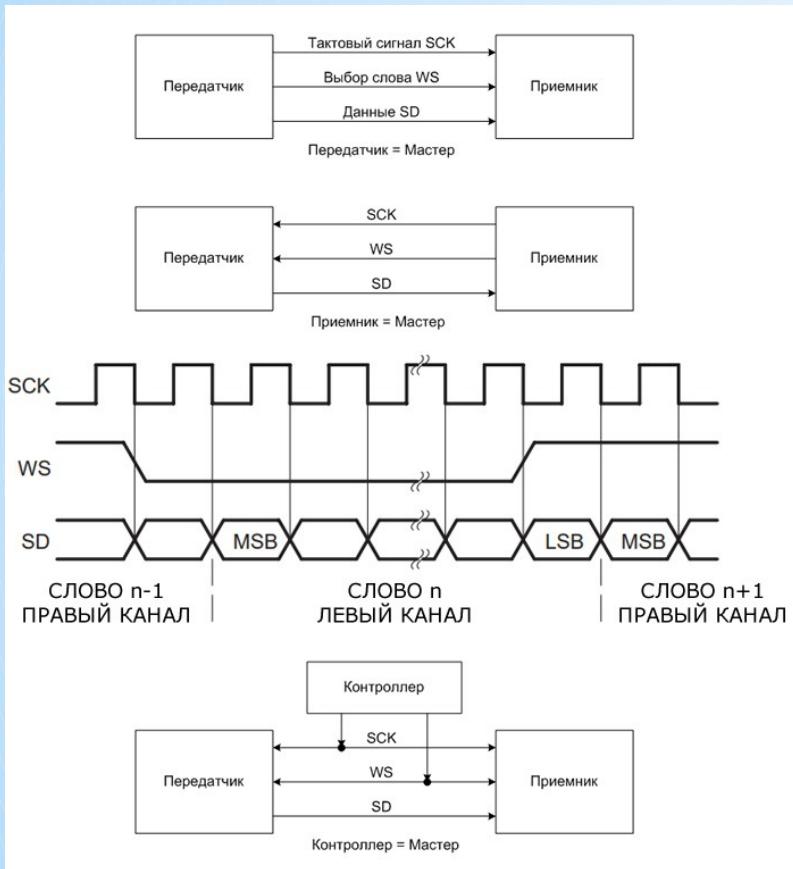


Передача байта 11101101

*SDIO



*I²S



Цифро-аналоговые преобразователи (ЦАП)

ИМС цифро-аналоговых преобразователей
классифицируются по следующим признакам:

- *По виду выходного сигнала:* с **токовым** выходом и выходом в виде **напряжения**.
- *По типу цифрового интерфейса:* **последовательный**
параллельный
- *По числу ЦАП на кристалле:* **одноканальные**
многоканальные.
- *По быстродействию:* **умеренного и высокого** быстродействия.

*Цифро-аналоговые преобразователи (ЦАП)

ЦАП преобразует *цифровой* сигнал $D(d_{n-1}d_{n-2}\dots d_1d_0)$ в эквивалентное значение *аналогового* сигнала. Каждый разряд кода d_i имеет определенный вес. Число, задаваемое кодом D , можно представить в виде

$$D = d_{n-1} \cdot 2^{n-1} + d_{n-2} \cdot 2^{n-2} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0 = \sum_{i=0}^{n-1} d_i \cdot 2^i \leq 2^n - 1$$

При обработке сигналов чаще применяется дробный формат с точкой, фиксированной *перед старшим* разрядом:

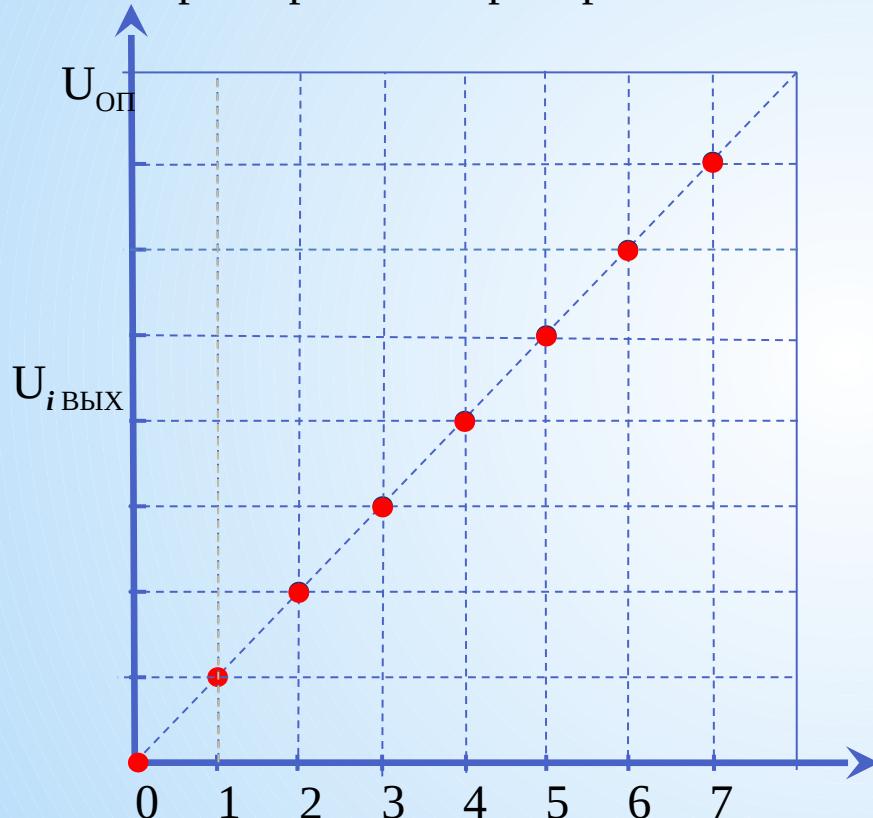
$$D = d_{n-1} \cdot 2^{-1} + d_{n-2} \cdot 2^{-2} + \dots + d_1 \cdot 2^{1-n} + d_0 \cdot 2^{-n} = \sum_{i=-1}^{-n} d_i \cdot 2^i \leq 1 - 2^{-n}$$

Дробный формат $m.l$

№	$m-1$	$m-2$	\dots	1	0		-1	-2	\dots	$1-l$	l
Вес	2^{n-1}	2^{n-2}	\dots	2^1	2^0		2^{-1}	2^{-2}	\dots	2^{1-l}	2^{-l}

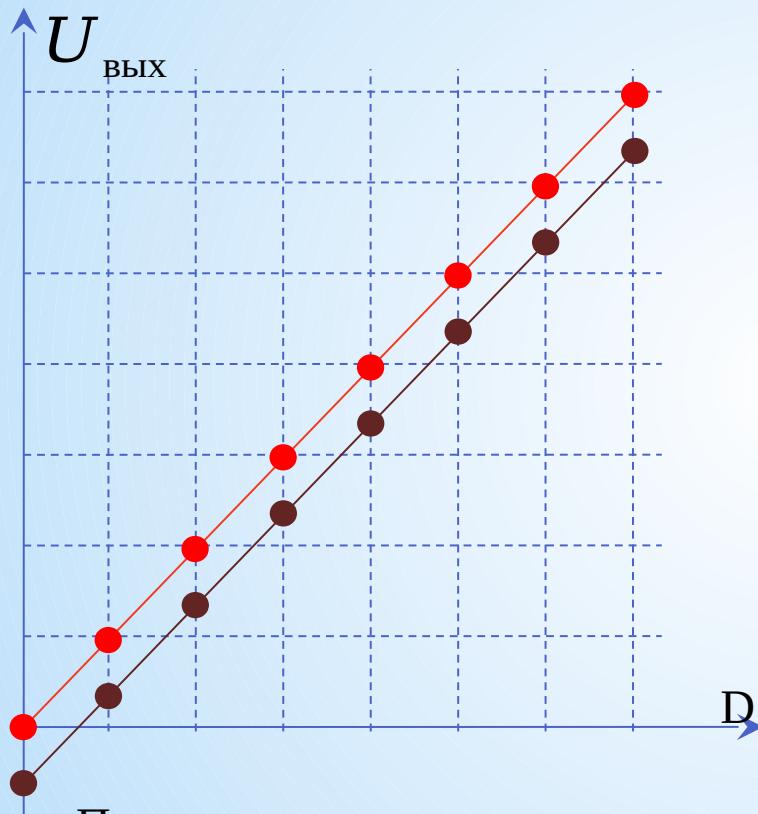
*Характеристика преобразования ЦАП

Характеристика преобразования – зависимость $U_{\text{вых}}$ от входного кода D

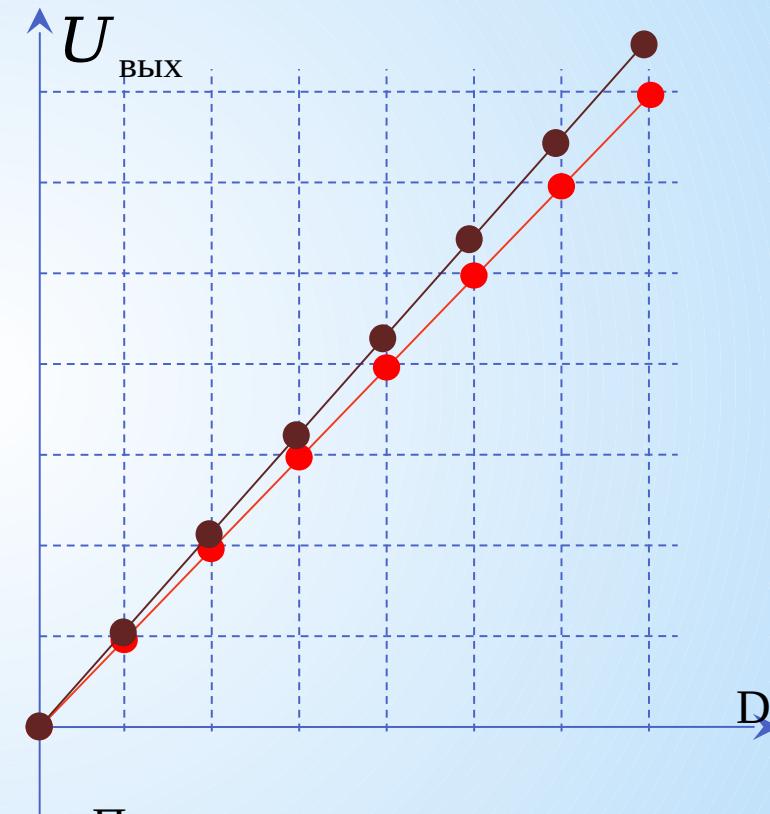


При последовательном возрастании значений входного цифрового сигнала D от 0 до 2^n-1 через единицу младшего разряда (МЗР) выходной сигнал $U_{\text{вых}}(i)$ образует 2^n точек. В отсутствие аппаратных погрешностей точки расположены на прямой, которая соответствует идеальной характеристике преобразования.

*Статические характеристики ЦАП

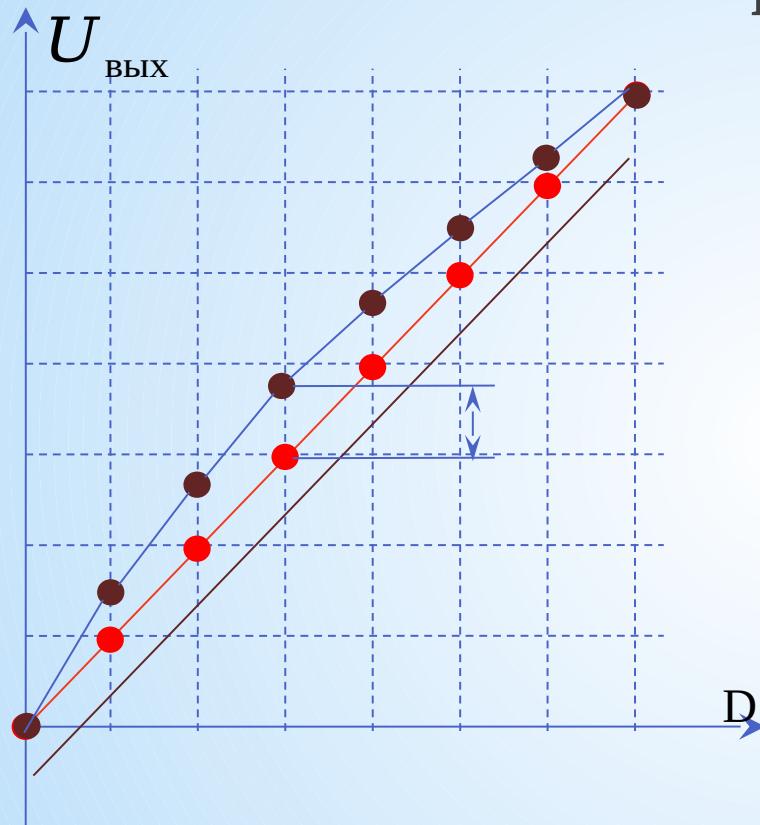


Погрешность смещения нуля

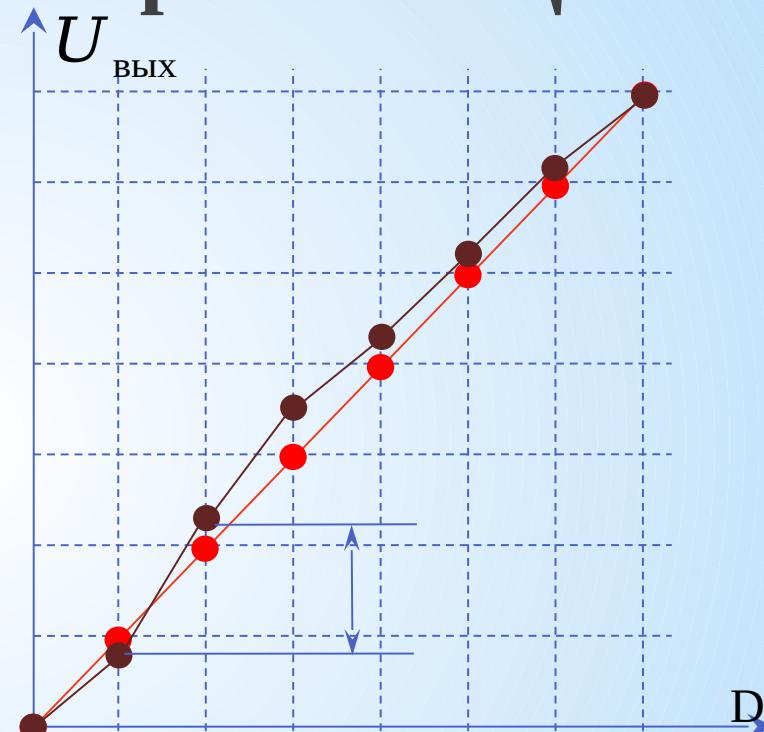


Погрешность усиления

*Статические характеристики ЦАП



Интегральная нелинейность – максимальное отклонение реальной характеристики от идеальной



Дифференциальная нелинейность – максимальная разность между соседними значениями выхода

*Характеристики ЦАП

*Статические характеристики

- Разрядность
- Разрешающая способность (относительная, абсолютная)
- Напряжение питания
- Уровни управляющего напряжения
- Величина опорного напряжения
- Максимальный выходной ток

*Погрешности преобразования:

- Погрешность полной шкалы (абсолютная погрешность преобразования);
- Погрешность смещения нуля;
- Погрешность линейности (нелинейность преобразования);
- Дифференциальная погрешность (дифференциальная нелинейность преобразования).

Характеристики ЦАП

Разрешающая способность - приращение $U_{\text{вых}}$ при преобразовании смежных значений N_j , т.е. отличающихся на ЕМЗР. Это приращение является шагом квантования.

Для двоичных кодов преобразования номинальное значение шага квантования

$$U_{\text{кв}} = U_{\text{пш}} / (2n-1),$$

где $U_{\text{пш}}$ - напряжение полной шкалы ЦАП (номинальное опорное напряжение),
n - количество разрядов двоичного числа, подаваемого на вход ЦАП

Чем больше разрядность преобразователя, тем выше его разрешающая способность.

Относительная разрешающая способность - это обратная величина от максимального числа уровней квантования ($2n-1$):

$$d_{\text{отн}} = 1 / (2n-1).$$

* Динамические характеристики ЦАП

Динамические характеристики. Идеальный ЦАП должен мгновенно выдавать аналоговый сигнал при подаче на вход цифрового кода. Реально же аналоговый сигнал на выходе появляется через какое-то время, которое включает внутренние задержки и время нарастания $U_{\text{вых}}$.

Время преобразования - интервал времени, в течение которого после подачи кода на выходе устанавливается сигнал с заданным отклонением от истинного (в пределах шага квантования).

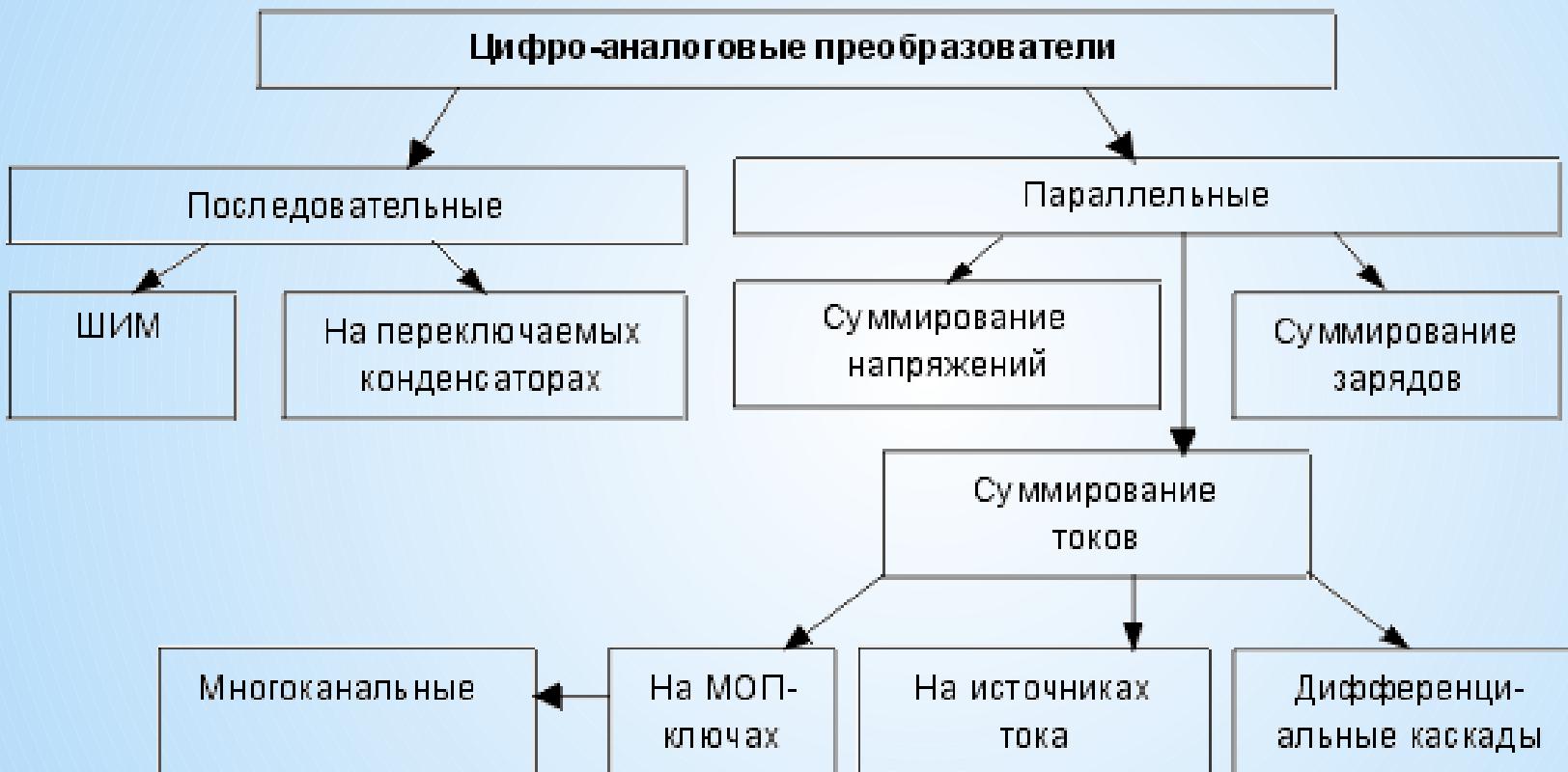
Максимальная частота дискретизации - максимальная частота, на которой ЦАП может работать, выдавая на выходе корректный результат.

Динамический диапазон (DD) - соотношение наибольшего и наименьшего сигналов (напряжения или тока), которые может воспроизвести ЦАП, выражается в дБ:

$$DD = 20 \cdot \lg \frac{U_{\max} - U_{\min}}{h}$$

где h – шаг квантования (связан с разрядностью и шумовым порогом).

Схемотехника цифро-аналоговых преобразователей



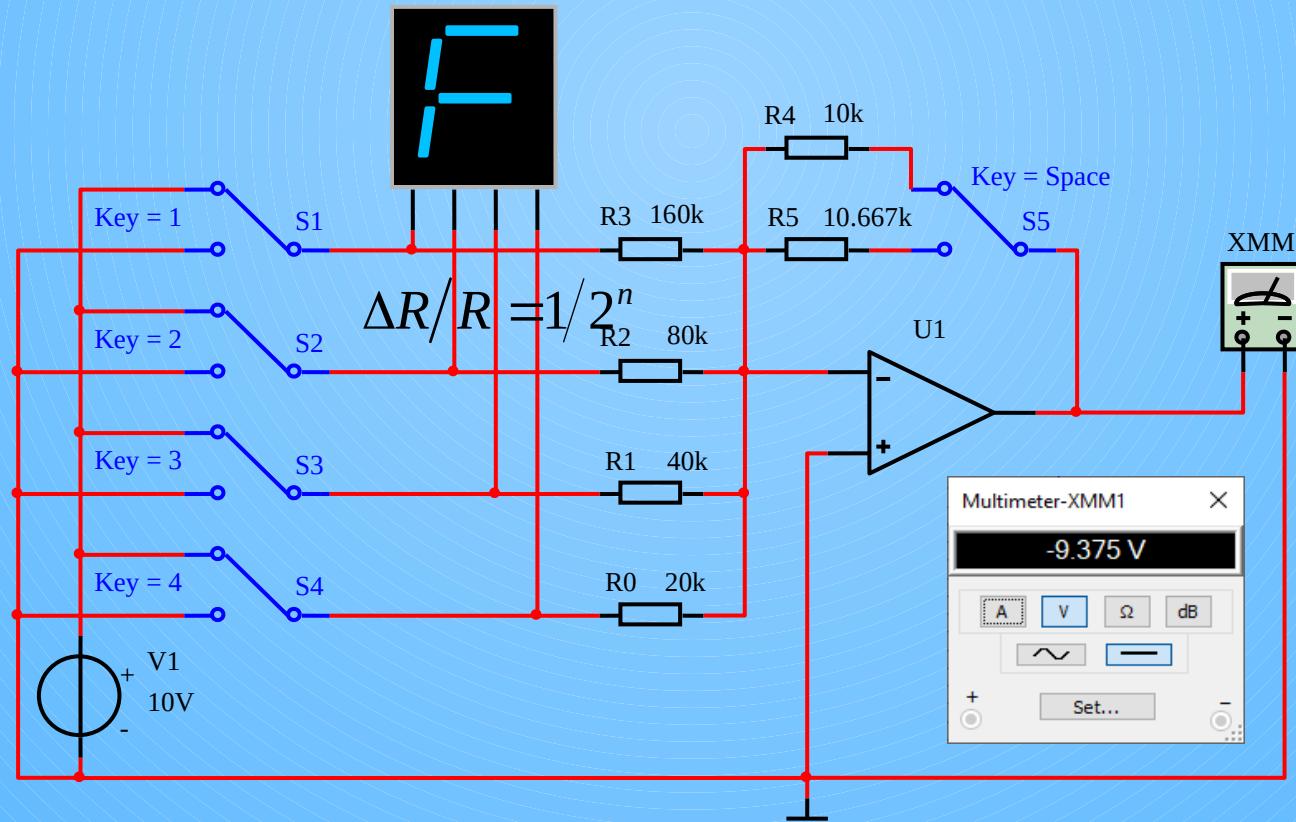
*Суммирующие ЦАП

Естественным способом преобразования двоичного кода в его аналоговый эквивалент представляется применение аналогового суммирования эквивалентов логических нуля и единицы с весами, соответствующими весам двоичных разрядов с некоторым масштабом, определяемым диапазоном аналоговых сигналов, принятым в конкретном устройстве.

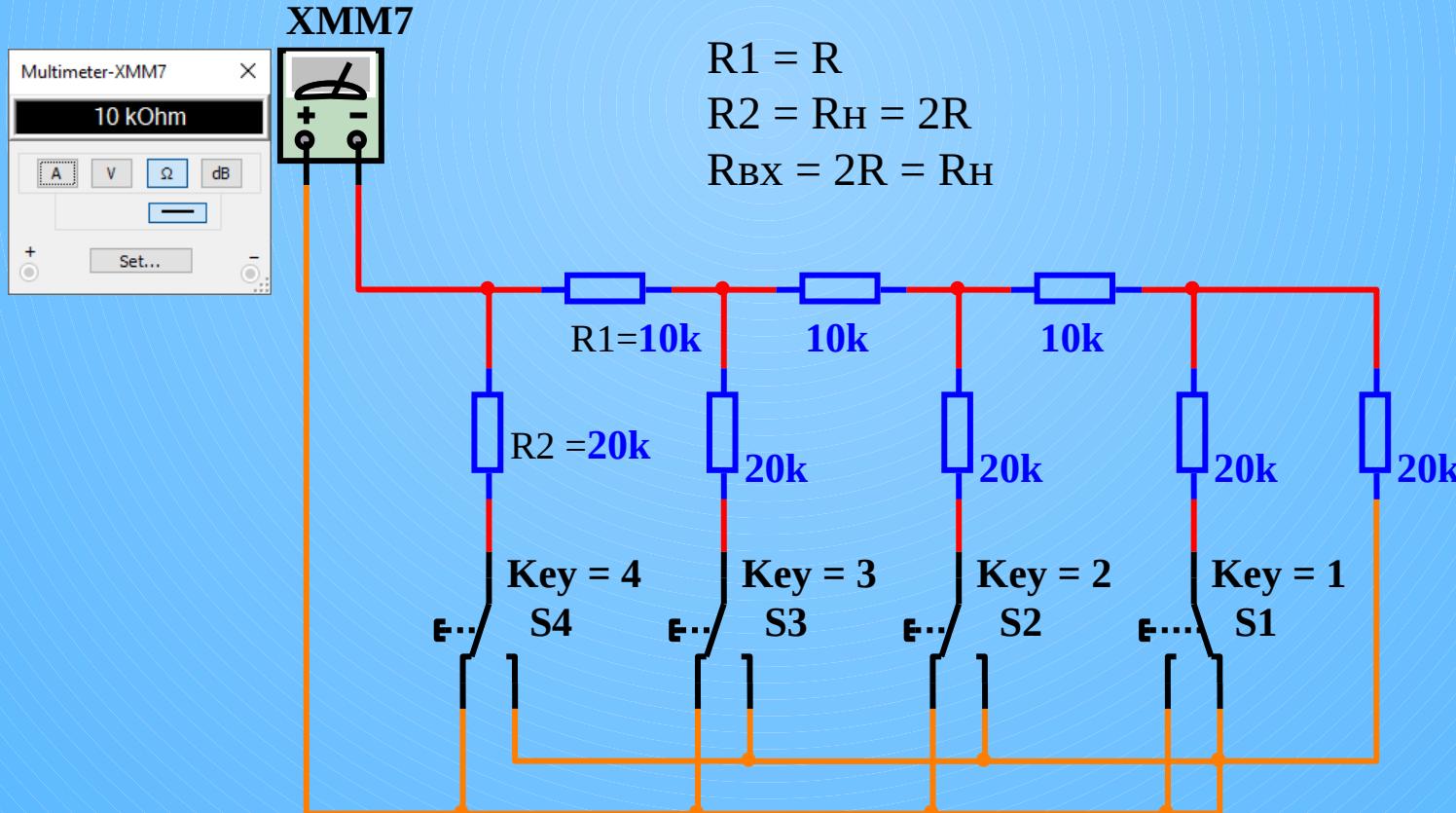
Очевидно, результат преобразования должен лежать в этом диапазоне, при том, что минимальному значению кода должен соответствовать минимальный уровень аналогового сигнала, а максимальному – соответственно, максимальный, точнее, меньше масштабной единицы на вес младшего значащего разряда (МЗР, LSB – Least Significant Bit), по аналогии с вышеприведёнными выражениями.

Величину аналоговой масштабной единицы в аналоговых устройствах задают в виде опорного напряжения $U_{\text{оп}}$ (U_{REF}), на несколько вольт меньшего, чем напряжение питания (чтобы обеспечить работу в линейном диапазоне аналоговых элементов, например, операционных усилителей).

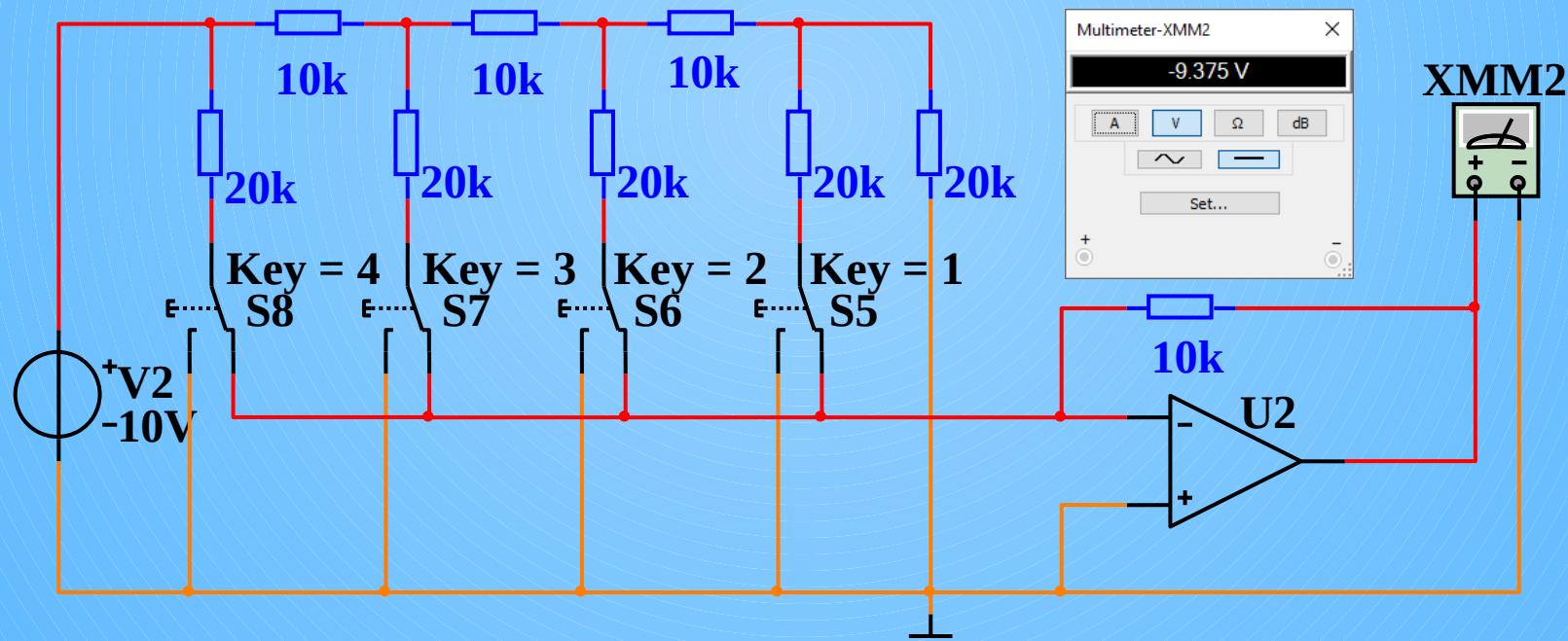
*Параллельная схема суммирования токов



*Матрица постоянного импеданса



Последовательная схема суммирования токов



$$U_{\text{выих}} = \frac{U_{\text{оп}} R_{\text{OC}}}{2R} \left(\frac{d^{n-1}}{2^1} + \dots + \frac{d^1}{2^{n-1}} + \frac{d^0}{2^n} \right) = \frac{U_{\text{оп}}}{2^n} \frac{R_{\text{OC}}}{2R} (d^{n-1} 2^{n-1} + \dots + d^1 2^1 + d^0 2^0)$$