

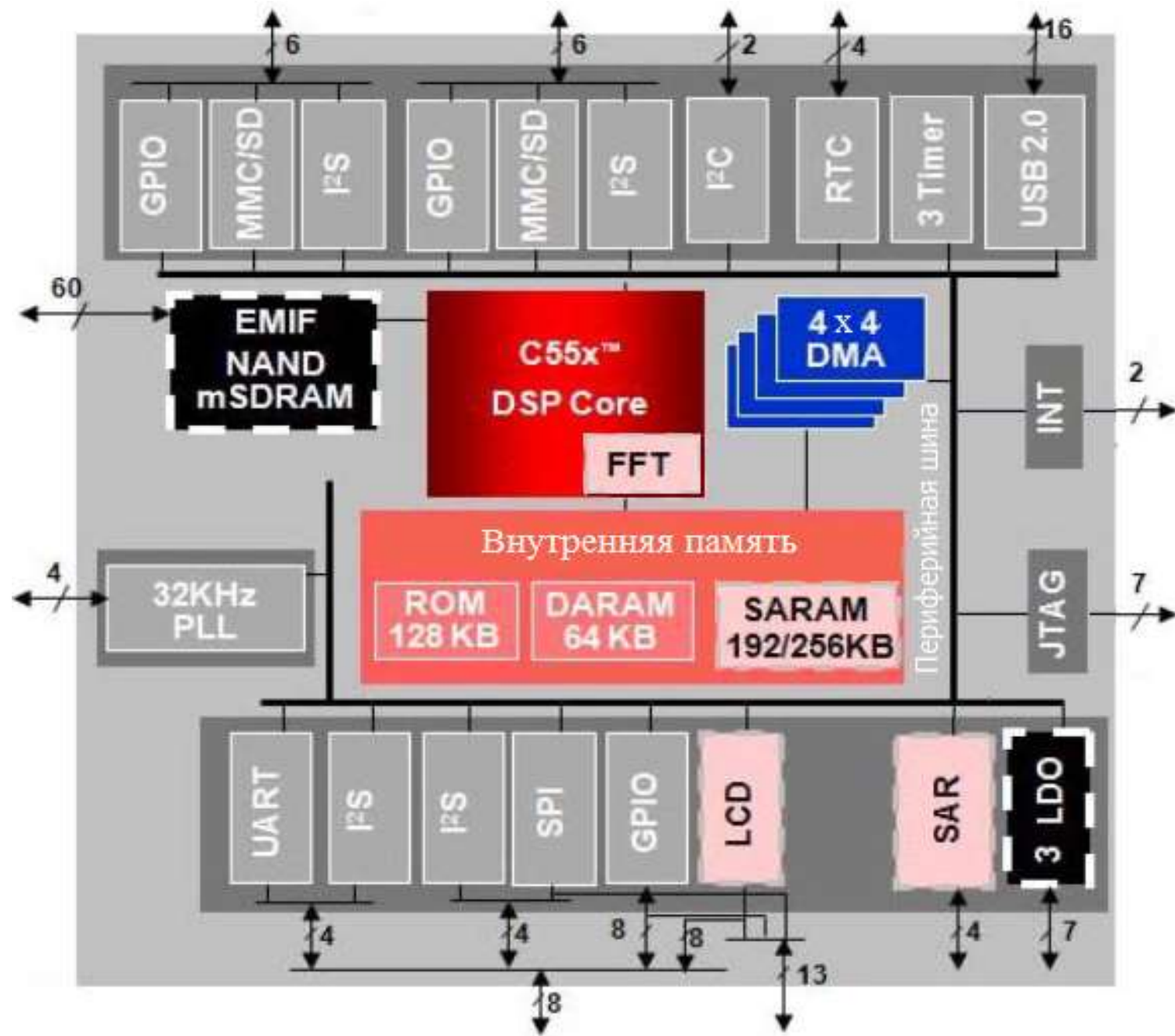


# Микропроцессорные устройства обработки сигналов

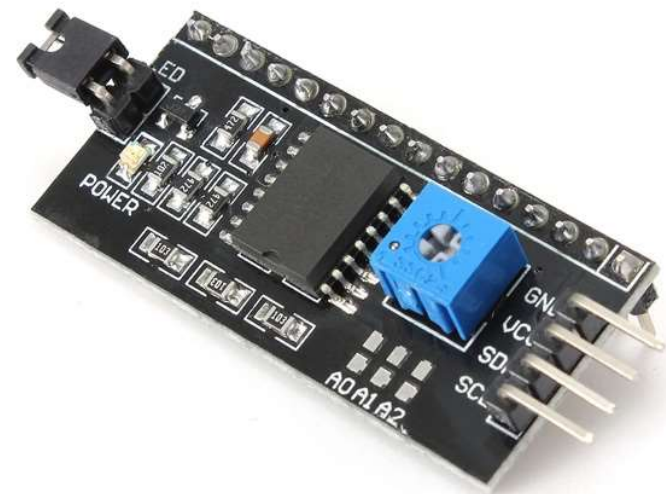
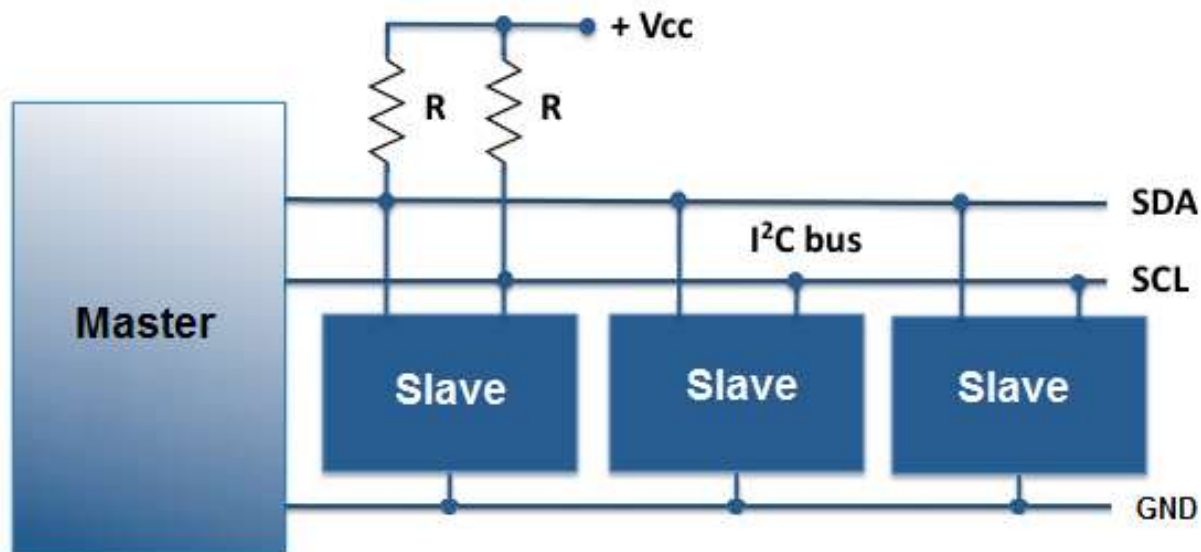
## Лекция L13 «Приборный интерфейс»

<http://vykhovanets.ru/course67/>

# Состав микропроцессора



# Интерфейс I2C



**I<sup>2</sup>C** – Inter-Integrated Circuit (межмикросхемный интерфейс)

Master – ведущее устройство

Slave – ведомое устройство

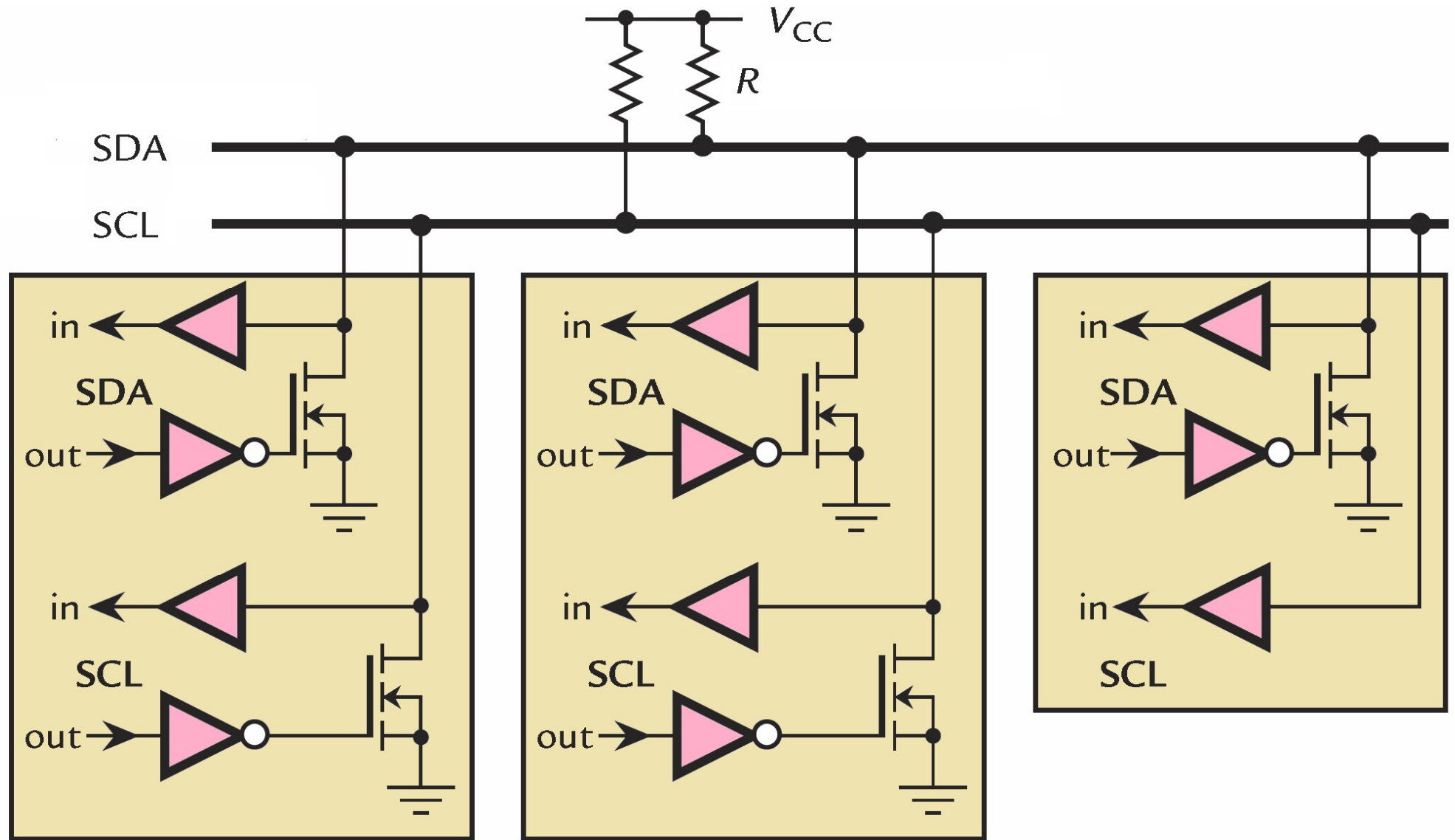
SDA – Serial Data (последовательные данные)

SCL – Serial Clock (последовательные тактовые сигналы)

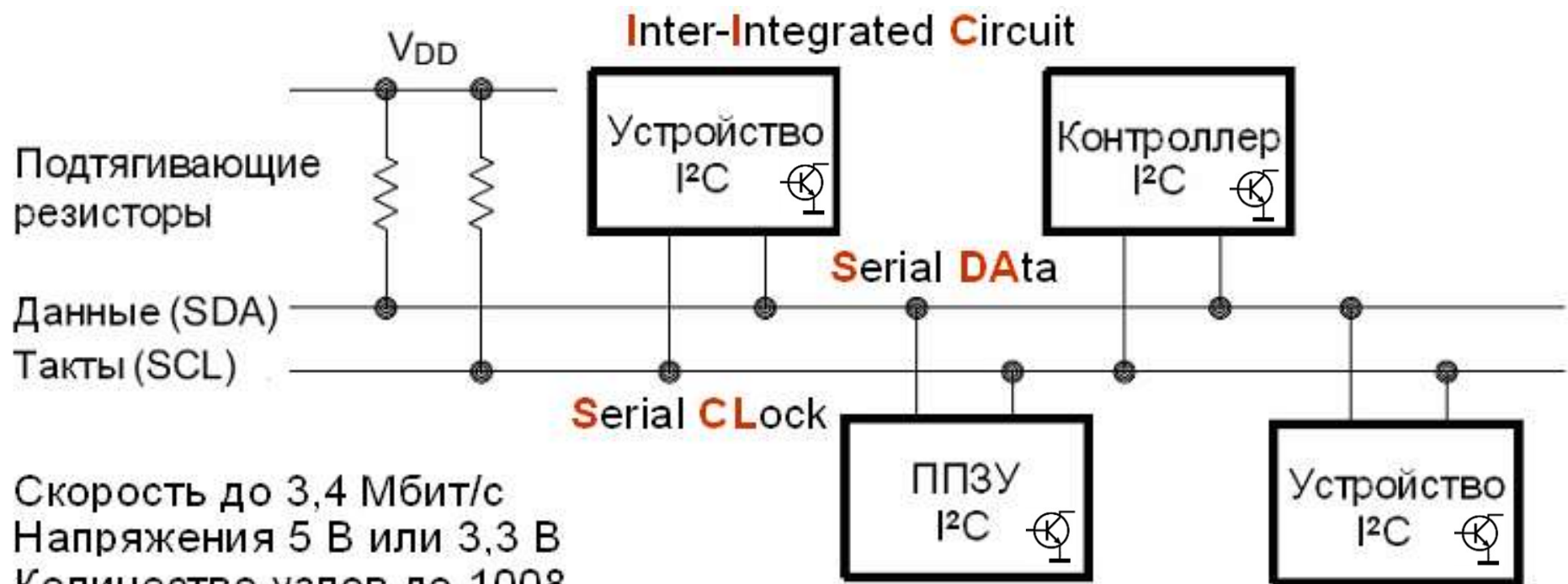
GND – Ground (общий проводник)

Vcc – Voltage Common Collector (напряжение питания)

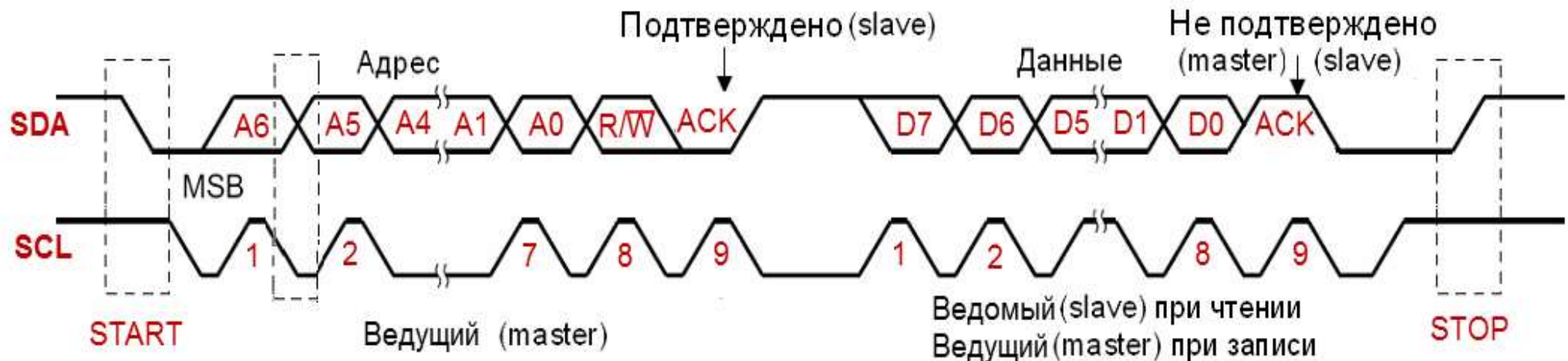
# Физический уровень



# Канальный уровень

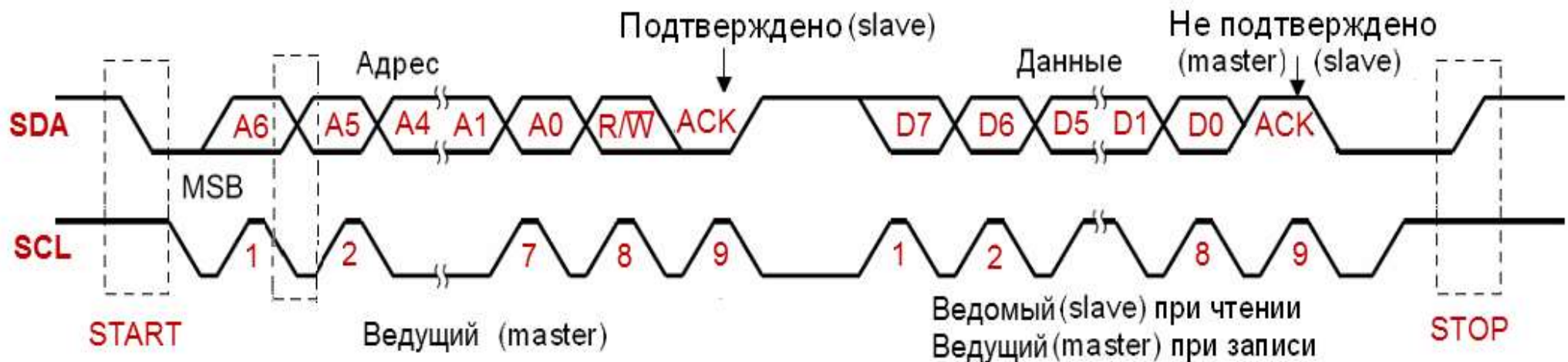
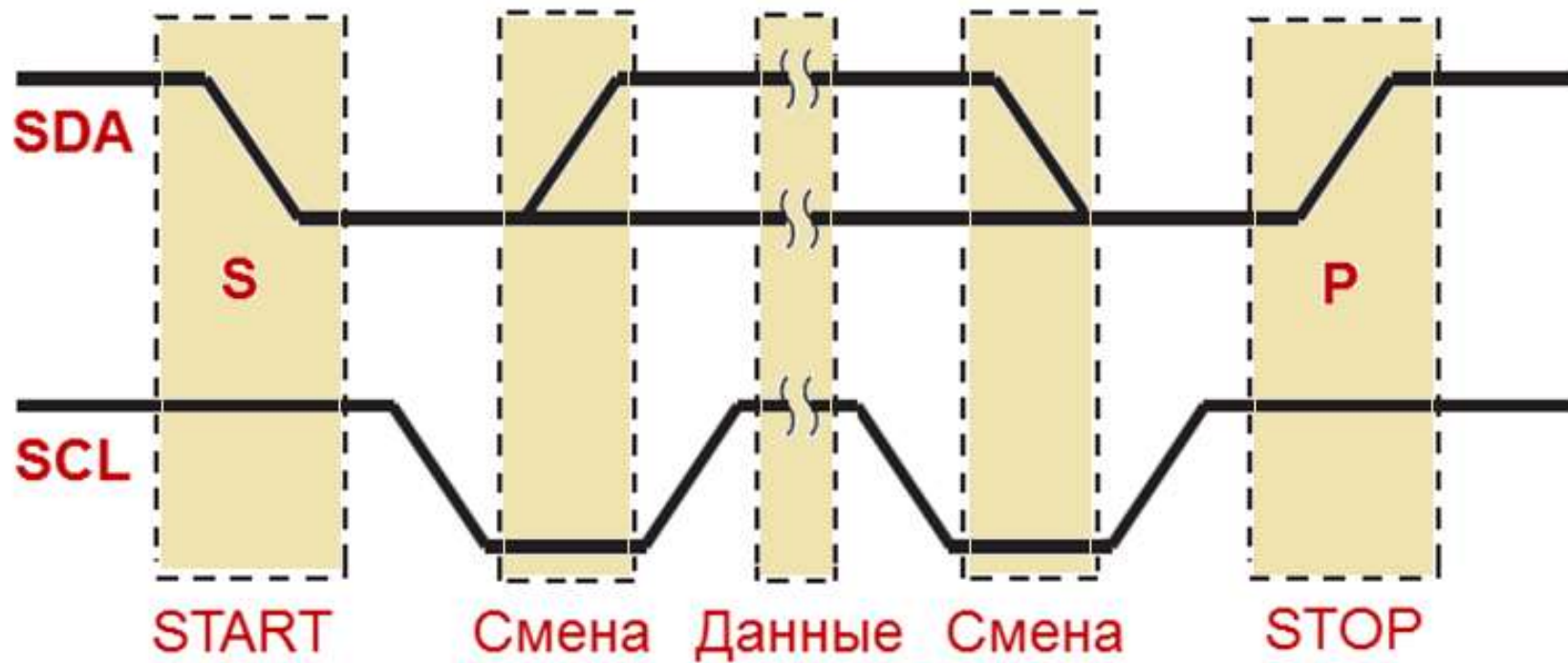


Скорость до 3,4 Мбит/с  
Напряжения 5 В или 3,3 В  
Количество узлов до 1008  
Максимальной емкость 400 пФ

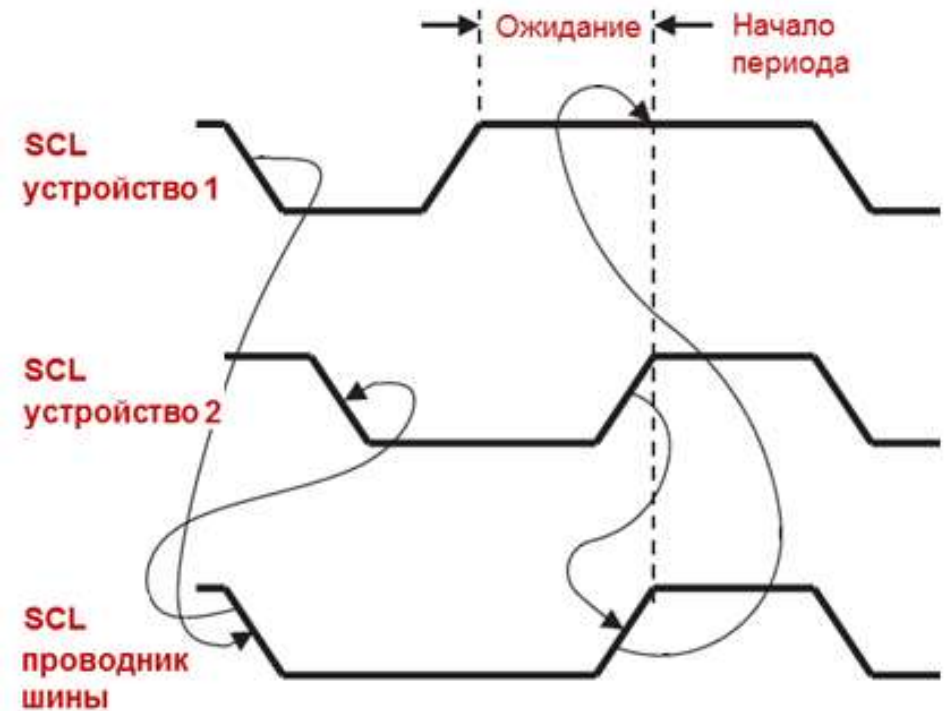
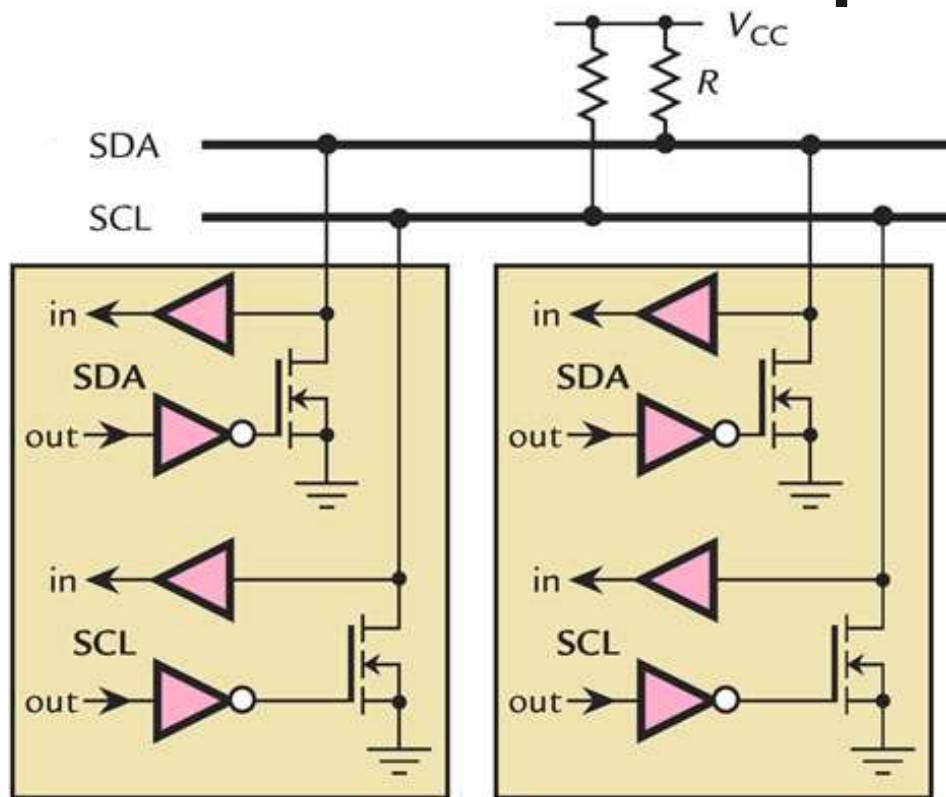




# Состояния линии

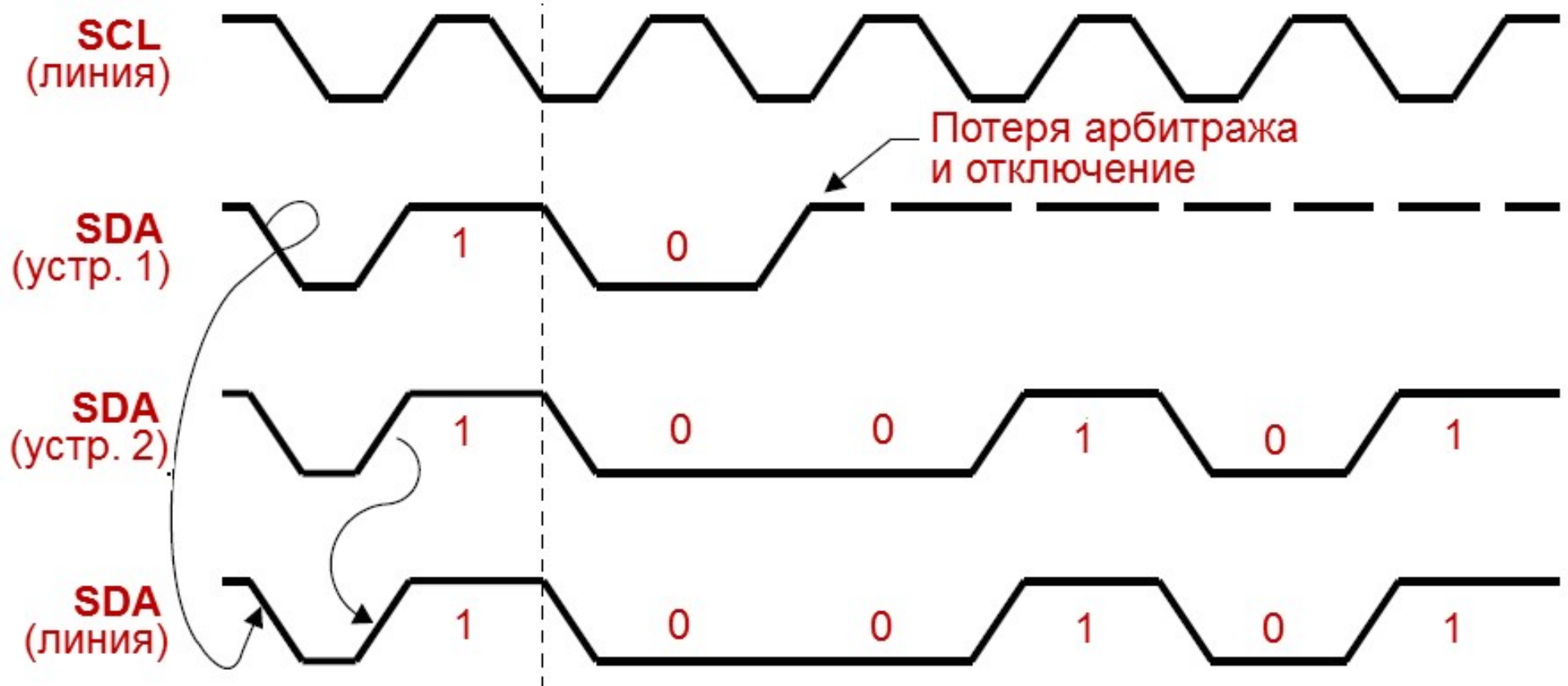


# Синхронизация



- Устройство, удерживающее низкий уровень тактового сигнала в линии, перекрывает такое удержание от всех других устройств.
- Если ведомое устройство удерживает низкий уровень тактового сигнала, остальные устройства переходят в состояние ожидания.
- Медленные ведомые устройства притормаживают быстрые ведущие устройства на время удержания тактового сигнала, достаточное для приема и сохранения данных (подготовки и передачи данных).

# Арбитраж

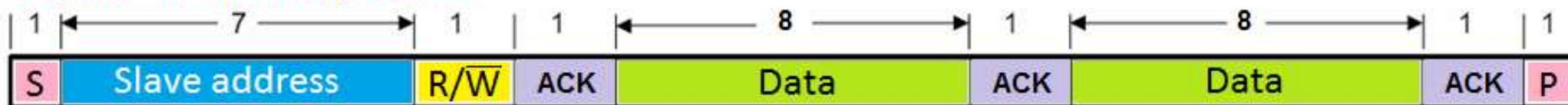


- Устройство, передающее данные с меньшим значением имеет больший приоритет на интерфейсе.
- Если несколько устройств передают похожие данные, то арбитраж наступает тогда, когда данные становятся различными.



# Протоколы I2C

## Формат 7-битной адресации



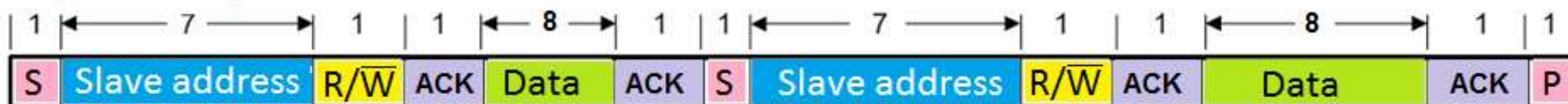
## Формат 10-битной адресации



## Свободный формат



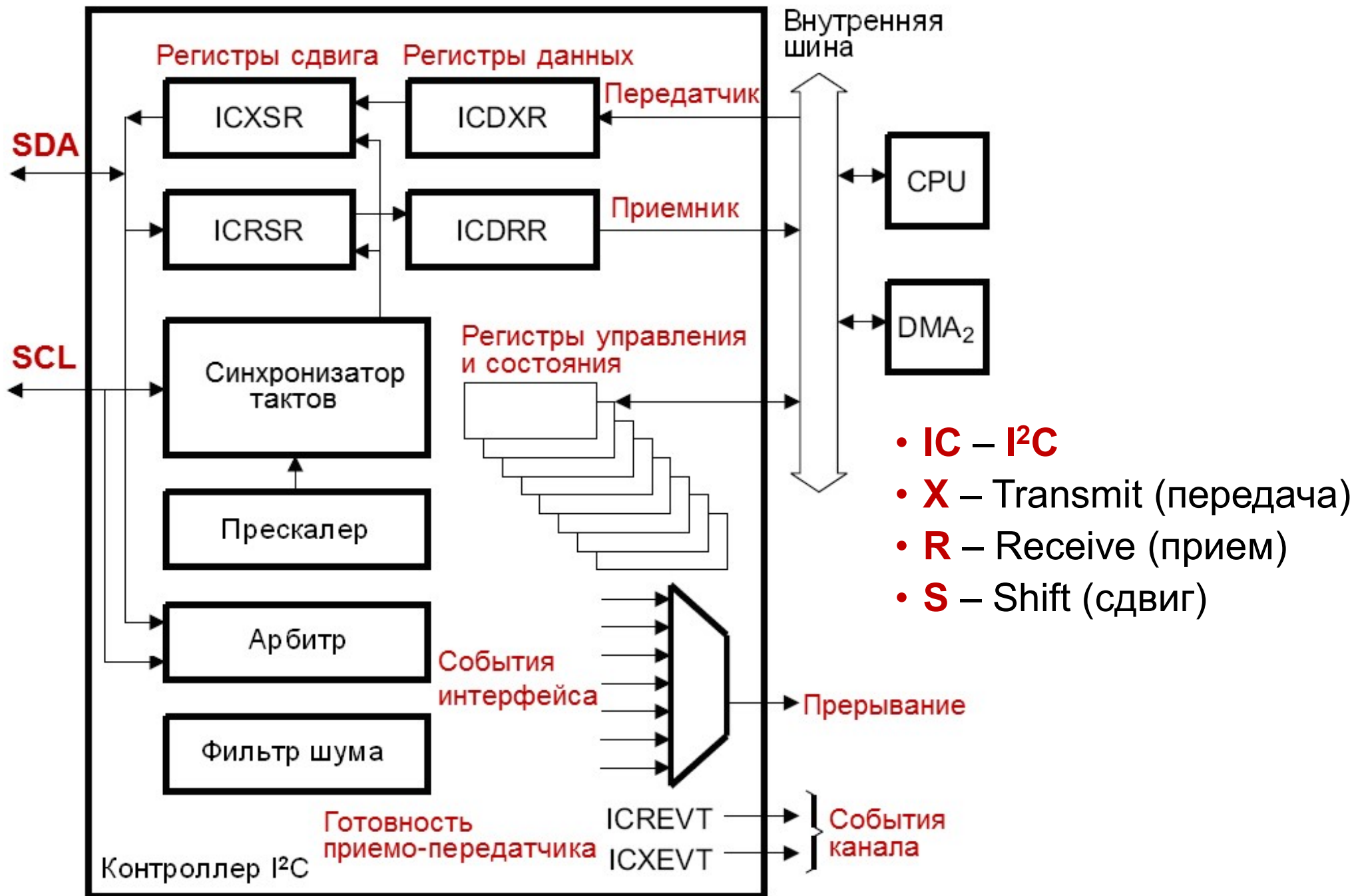
## Формат повторения



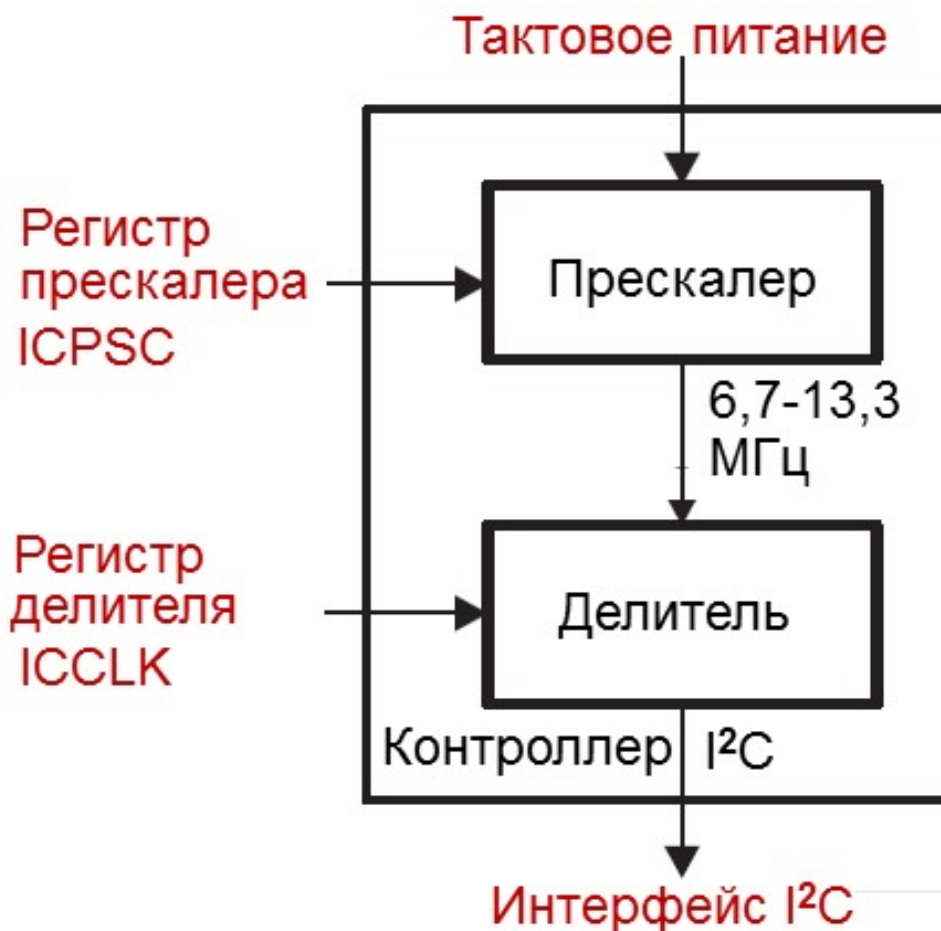
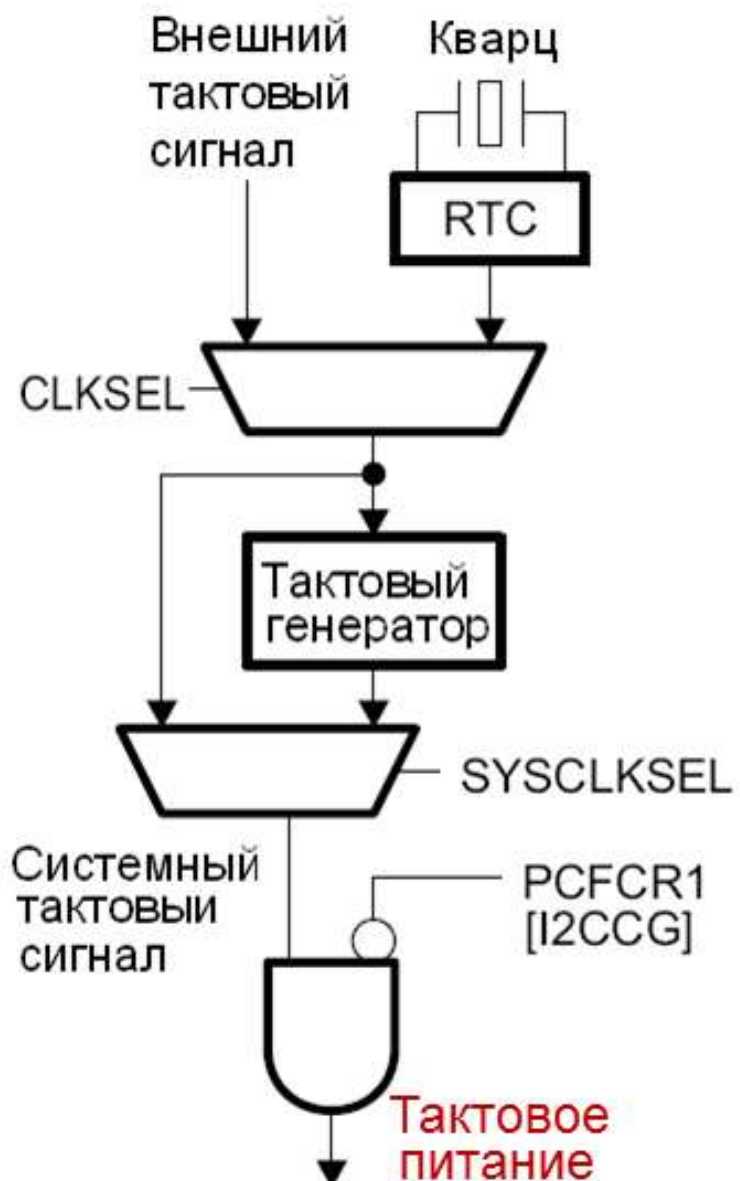
- **Режимы работы:**

- ведомый приемник (slave-receiver mode);
- ведомый передатчик (slave-transmitter mode);
- ведущий приемник (master-receiver mode);
- ведущий передатчик (master-transmitter mode).

# Организация контроллера



# Тактовое питание



$$f_{SCL} = \frac{f_{ICPSC}}{(N_{ICCLKH} + d)(N_{ICCLKL} + d)}, \quad d \in \{7, 6, 5\}$$

**PCGCR1 (1C02h)** – Peripheral Clock Gating Configuration Register 1

7	6	5	4	3	2	1	0
MMCS1CG	I2CCG	Reserved	MMCS0CG	DMA0CG	UARTCG	SPICG	I2S3CG

# Регистры I<sup>2</sup>C

Адрес	Обозначение	Описание
1A00h	<b>ICOAR</b>	<b>O</b> wn <b>A</b> ddress <b>R</b> egister (регистр собственных адресов)
1A04h	<b>ICIMR</b>	<b>I</b> nterrupt <b>M</b> ask <b>R</b> egister (регистр маски прерываний )
1A08h	<b>ICSTR</b>	<b>I</b> nterrupt <b>S</b> tatus <b>R</b> egister (регистр статуса прерываний )
1A0Ch	<b>ICCLKL</b>	<b>C</b> lock <b>L</b> ow-Time Divider Register (регистр делителя частоты )
1A10h	<b>ICCLKH</b>	<b>C</b> lock <b>H</b> igh-Time Divider Register (регистр делителя частоты )
1A14h	<b>ICCNT</b>	Data <b>C</b> ount (счетчик данных)
1A18h	<b>ICDRR</b>	<b>D</b> ata <b>R</b> ecieve <b>R</b> egister (регистр принятых данных )
1A1Ch	<b>ICSAR</b>	<b>S</b> lave <b>A</b> ddress <b>R</b> egister (регистр подчиненного адреса )
1A20h	<b>ICDXR</b>	<b>D</b> ata <b>T</b> ransmit <b>R</b> egister (регистр передаваемых данных )
1A24h	<b>ICMDR</b>	<b>M</b> ode <b>R</b> egister(регистр режима )
1A28h	<b>ICIVR</b>	<b>I</b> nterrupt <b>V</b> ector <b>R</b> egister (регистр вектора прерываний )
1A2Ch	<b>ICEMDR</b>	<b>E</b> xtended <b>M</b> ode <b>R</b> egister (регистр расширенного режима )
1A30h	<b>ICPSC</b>	<b>P</b> re <b>s</b> caler Register (регистр прескалера )
1A34h	<b>ICPID1</b>	<b>P</b> eripheral <b>I</b> dentification Register <b>1</b> (регистр идентификации 1)
1A38h	<b>ICPID2</b>	<b>P</b> eripheral <b>I</b> dentification Register <b>2</b> (регистр идентификации 2)

# Вектор прерываний

Вектор		Прерывание	Приоритет	Адрес
00	RESET	Сброса и инициализации	00	IVPD:00h
01	NMI	Внутреннее немаскируемое	01	IVPD:08h
02	INT0	Внешнее по входу INT0	03	IVPD:10h
03	INT1	Внешнее по входу INT1	05	IVPD:18h
04	TINT	Агрегированное таймера	06	IVPD:20h
...	...	...	...	...
08	DMA	Прямого доступа к памяти	11	IVPD:40h
...	...	...	...	...
13	SAR	Агрегированное АЦП	18	IVPD:68h
...	...	...	...	...
18	RTC	Часов реального времени	12	IVPH:90h
...	...	...	...	...
21	GPIO	Портов ввода-вывода	20	IVPH:A8h
...	...	...	...	...
23	I2C	Контроллера I2C	24	IVPH:B8h
...	...	...	...	...



# Регистры адреса и масок

**ICOAR** (**1A00h**) – I2C Own Address Register (регистр собственного адреса)

15	10	9	0
Reserved			OADDR

**ICIMR** (**1A04h**) – I2C Interrupt Mask Register (регистр маски прерываний)

15								8							
Reserved															
7		6		5		4		3		2		1		0	
Reserved		AAS		SCD		ICXRDY		ICRRDY		ARDY		NACK		AL	

- **OADDR** – Own slave address (собственный адрес)
- **AAS** – Address-as-slave (обнаружение нулевого или своего адрес)
- **SCD** – Stop condition detected (обнаружение условия STOP)
- **ICXRDY** – Transmit-data-ready(готовность передатчика)
- **ICRRDY** – Receive-data-ready (готовность приемника)
- **ARDY** – Register-access-ready (готовность доступа к регистру)
- **NACK** – No-acknowledgment (нет подтверждения)
- **AL** – Arbitration-lost (потеря арбитража)

# Регистр статуса

**ICSTR (1A08h)** – I2C Interrupt Status Register (регистр состояния прерываний)

15	14	13	12	11	10	9	8
Reserved	SDIR	NACKSNT	BB	RSFULL	XSMT	AAS	AD0
7	6	5	4	3	2	1	0
Reserved		SCD	ICXRDY	ICRRDY	ARDY	NACK	AL

- **SDIR** – Slave direction (ведомое направление: 0 – ведомый приемник, 1 – ведомый передатчик)
- **NACKSNT** – No-acknowledgment sent (отправлено подтверждение)
- **BB** – Bus busy (занятость шины)
- **RSFULL** – Receive shift full (переполнение регистра сдвига приемника)
- **XSMT** – Transmit shift empty (пустота регистра сдвига передатчика)
- **AAS** – Addressed-as-slave (получен собственный адрес)
- **AD0** – Address 0 (получен нулевой адрес)
- **SCD** – Stop condition detected (обнаружено состояние STOP)
- **ICXRDY** – Transmit-data-ready (готовность передатчика)
- **ICRRDY** – Receive-data-ready (готовность приемника)
- **ARDY** – Register-access-ready (готовность доступа к регистру)
- **NACK** – No-acknowledgment (нет подтверждения)
- **AL** – Arbitration-lost (потеря арбитража)

# Регистры приема-передачи

**ICCLKL** (**1A0Ch**) – I2C Clock Low-Time Divider Register (регистр делителя 1)

15 0

ICCL

**ICCLKH** (**1A10h**) – I2C Clock High-Time Divider Register (регистр делителя 2)

15 0

ICCH

**ICCNT** (**1A14h**) – I2C Data Count Register (регистр счетчика данных)

15 0

ICDC

**ICDRR** (**1A18h**) – I2C Data Receive Register (регистр принятых данных)

15 8 7 0

Reserved

D

**ICSAR** (**1A1Ch**) – I2C Slave Address Register (регистр подчиненного адреса)

15 10 9 0

Reserved

SADDR

**ICDXR** (**1A20h**) – I2C Data Transmit Register (регистр передаваемых данных)

15 8 7 0

Reserved

D

- **ICCL** – Clock low-time divide-down (подчиненное направление)

- **ICCH** – Clock high-time divide-down (отправка подтверждения)

- **ICDC** – Data count (счетчик данных)

- **SADDR** – Slave address (подчиненный адрес)

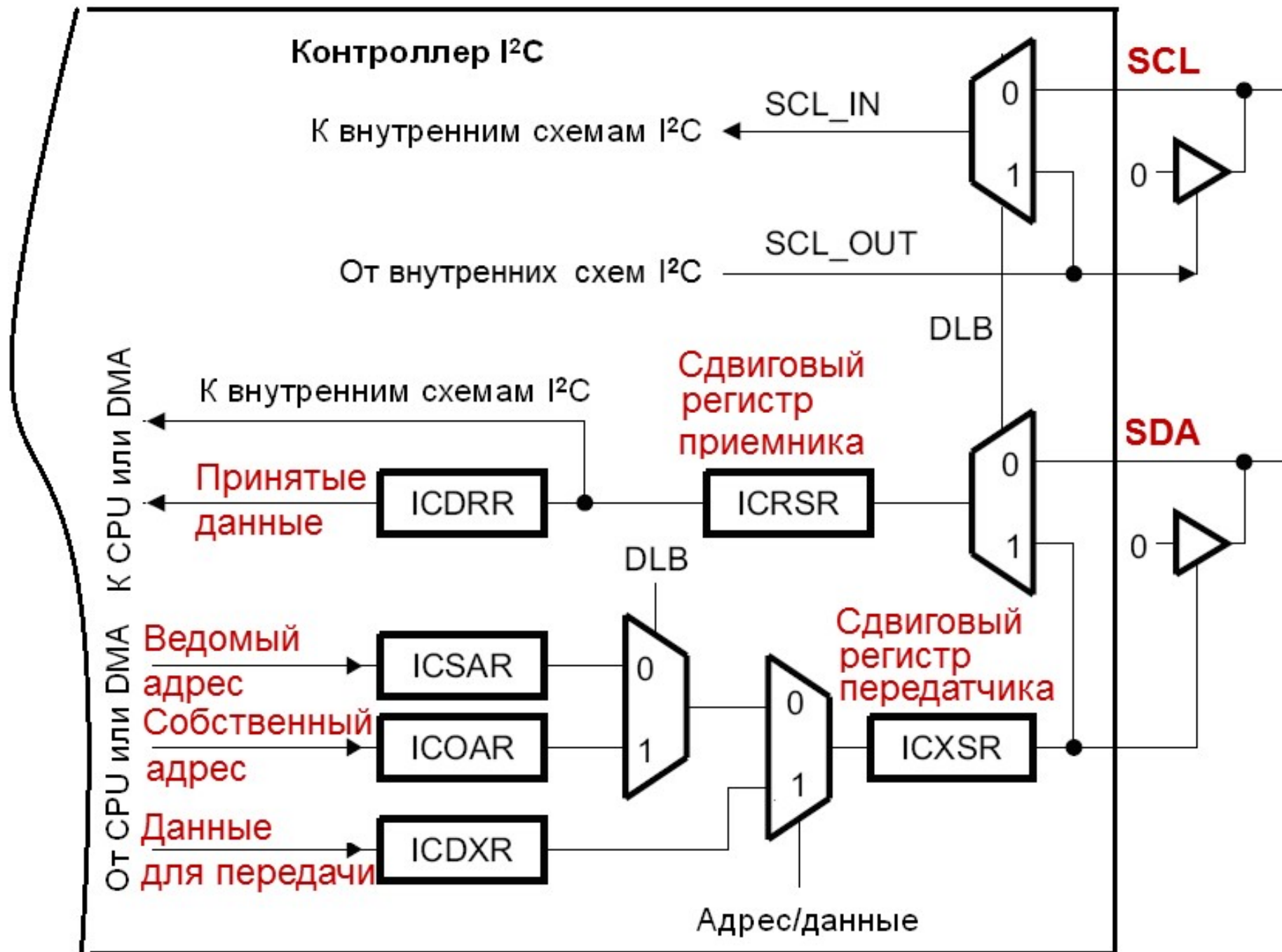
# Регистр режима

**ICMDR** (**1A24h**) – I2C Mode Register (регистр режима)

15	14	13	12	11	10	9	8
NACKMOD	FREE	STT	Reserved	STP	MST	TRX	XA
7	6	5	4	3	2	0	
RM	DLB	IRS	STB	FDF	BC		

- **NACKMOD** – No-acknowledge (NACK) mode (режим без подтверждения)
- **FREE** – Free emulation mode (работа в точке останова эмулятора)
- **STT** – START condition (генерация или обнаружение START)
- **STP** – STOP condition (генерация или обнаружение STOP)
- **MST** – Master mode (режим ведущего устройства)
- **TRX** – Transmitter mode (режим передачи)
- **XA** – Expanded address (режим расширенного адреса)
- **RM** – Repeat mode (режим повторения, число повторений в ICCNT)
- **DLB** – Digital loopback mode (режим цифровой петли)
- **IRS** – I2C ignore reset (разрешение работы контроллера I<sup>2</sup>C)
- **STB** – START byte mode (режим удлиненного стартового состояния)
- **FDF** – Free data format mode (режим свободных данных)
- **BC** – Bit count (число бит: 0 – 8 бит, 2 – 2 бита, ..., 7 – 7 бит)

# Режим цифровой петли





# Другие регистры

**ICIVR (1A28h)** – I2C Interrupt Vector Register (регистр вектора прерываний)

15		3	2	0
Reserved				INTCODE

**ICEMDR (1A2Ch)** – I2C Extended Mode Register (регистр расширенного режима)

15		2	1	0
Reserved			IGNACK	BCM

**ICPSC (1A30h)** – I2C Prescaler Register (регистр прескалера)

15		8	7	0
Reserved				IPSC

- **INTCODE** – Interrupt code (код прерывания:
  - 0 – нет прерывания;
  - 1 – потеря арбитража (AL);
  - 2 – нет подтверждения (NACK);
  - 3 – готовность доступа к регистру (ARDY);
  - 4 – готовность приемника (ICRRDY);
  - 5 – готовность передатчика (ICXRDY);
  - 6 – состояние STOP (SCD);
  - 7 – адрес как у ведомого (AAS))
- **IGNACK** – Ignore NACK (игнорировать отсутствие подтверждения)
- **BCM** – Backward compatibility (обратная совместимость – прерывание готовности передатчика генерируются при копировании ICDXR в ICXSR)
- **IPSC** – I2C prescaler divide-down (делитель прескалера)

# Регистры идентификации

**ICPID1 (1A34h)** – I2C Peripheral Identification Register 1

15	8	7	0
Class			Revision

**ICPID2 (1A38h)** – I2C Peripheral Identification Register 2

15	8	7	0
Reserved			TYPE

- **Class** – Класс устройства (01h)
- **Revision** – Версия (06h)
- **TYPE** – Тип (05h)

## Аппаратурный сброс

**PRCR (1C05h)** – Peripheral Reset Control Register (регистр управления сбросом)

7	6	5	4	3	2	1	0
PG4_RST	Reserved	PG3_RST	DMA_RST	USB_RST	SAR_RST	PG1_RST	I2C_RST

Запись нуля без эффекта, запись единицы запускает сброс, чтение единицы сигнализирует о состоянии сброса, чтение нуля сигнализирует об окончании сброса.

# Подключение I<sup>2</sup>C

// Подключение I2C

uint16 I2C\_open( )

{

PCGCR1 &= ~PCGCR1\_I2CCG;

C5515\_wait( 100 );

ICMDR = ICMDR\_MST;

ICPSC = 20;

ICCLKL = 20;

ICCLKH = 20;

ICMDR = ICMDR\_MST

| ICMDR\_IRS;

return 0;

}

// Отключение I2C

uint16 I2C\_close( )

{

ICMDR = 0;

C5515\_wait( 100 );

PCGCR1 |= PCGCR1\_I2CCG;

return 0;

}

void C5515\_wait( Uint16 delay )

{

volatile Uint16 i;

for ( i = 0 ; i < delay ; i++ );

}

// Тактирование приборного интерфейса

// Ожидание завершения

// Режим ведущего и состояние сброса

// Делитель прескалера 12 МГц

// Делитель частоты 1 20kHz

// Делитель частоты 2 20kHz

// Ведущее устройство, 7-бит адреса

// Разрешение работы, 8 бит в посылке

// Сброс и очистка I2C

// Ожидание завершения

// Снять тактирование

# Передача данных I<sup>2</sup>C

```
uint16 I2C_write( uint16 i2c_addr, uint8* data, uint16 len )
{
    // Локальные данные
    int16 i = 0;
    // Инициализация передачи
    ICIER = 0;
    ICCNT = len;
    ICSAR = i2c_addr;
    ICMDR = ICMDR_STT
        | ICMDR_TRX
        | ICMDR_MST
        | ICMDR_IRS
        | ICMDR_FREE;
    c5515_wait(10);
    // Выдача данных
    for ( i = 0 ; i < len ; i++ )
    {
        ICDXR = data[i];
        do i = ICSTR & ICSTR_XRDY;
        while ( i == 0 );
    }
    ICMDR |= ICMDR_STP;
    c5515_wait(800);
    return 0;
}
```

// Запрет прерываний от I2C  
// Задание длины данных в посылках  
// Задать адрес ведомого  
// Генерация состояния START  
// Передатчик  
// Ведущее устройство  
// Разрешить работу  
// Работа совместно с эмулятором  
// Задержка перед передачей  
  
// Запись байта в регистр передатчика  
// Получить состояние передачи  
// Ожидать пока не передано  
  
// Генерация STOP  
// Перерыв в линии

# Примем данных I<sup>2</sup>C

```
uint16 I2C_read( uint16 i2c_addr, uint8* data, uint16 len )
{
    // Локальные данные
    int16 i = 0;
    // Инициализация передачи
    ICCNT = len;
    ICSAR = i2c_addr;
    ICMDR = ICMDR_STT
        | ICMDR_MST
        | ICMDR_IRS
        | ICMDR_RM;
    c5515_wait(10);
    // Прием данных
    for ( i = 0 ; i < len ; i++ )
    {
        do i = ICSTR & STR_RRDY;
        while ( i == 0 );
        data[i] = ICDDR;
    }
    ICMDR |= ICMDR_STP;
    c5515_wait(10);
    return 0;
}
```

// Задание длины данных в посылках  
// Задать адрес ведомого  
// Генерация START  
// Ведущее устройство  
// Разрешить работу  
// Работа в повторном режиме  
// Задержка перед приемом

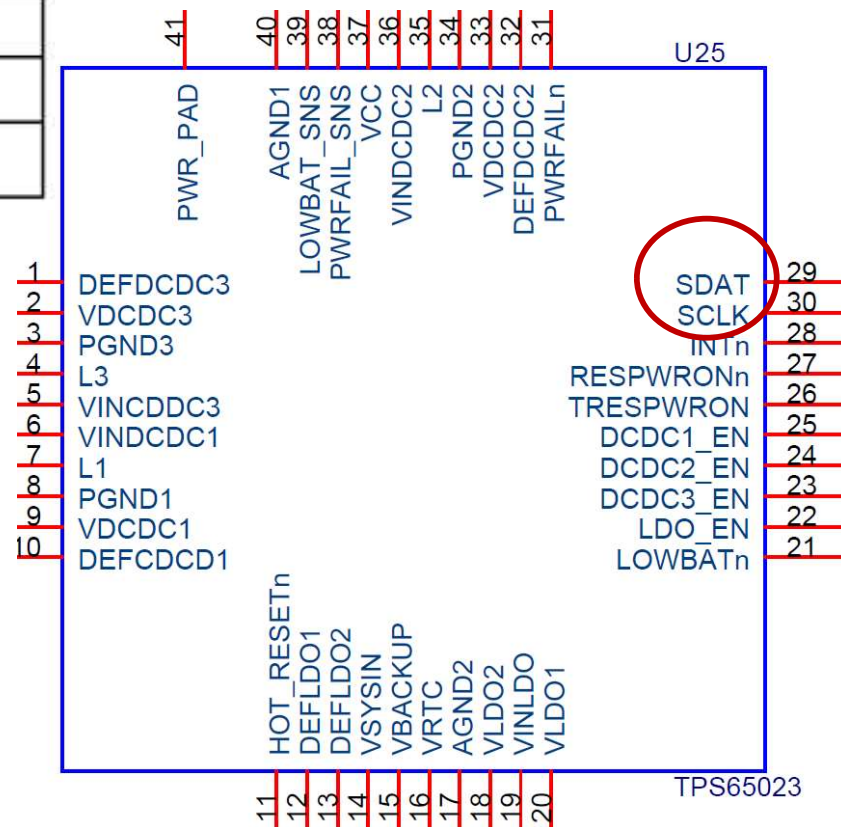
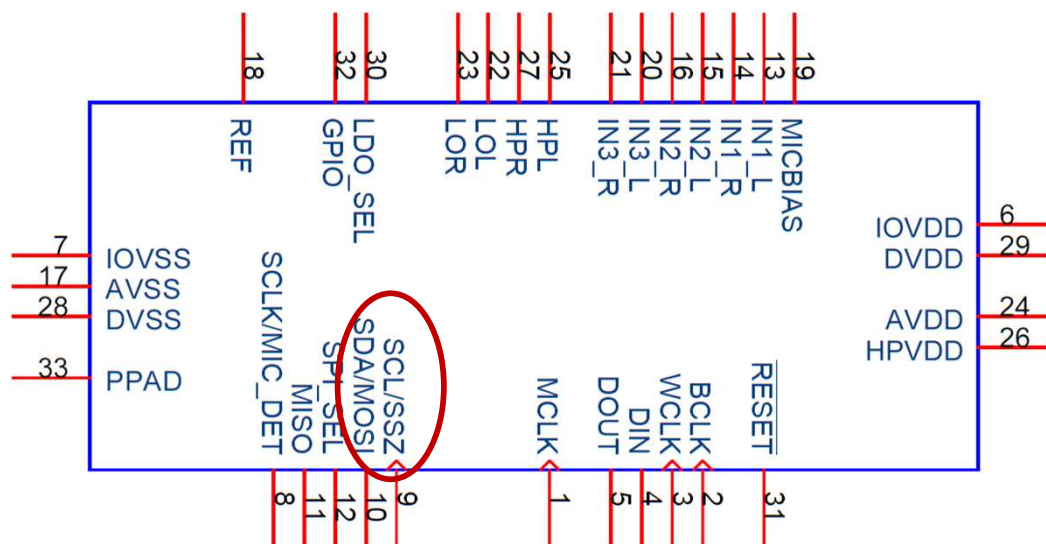
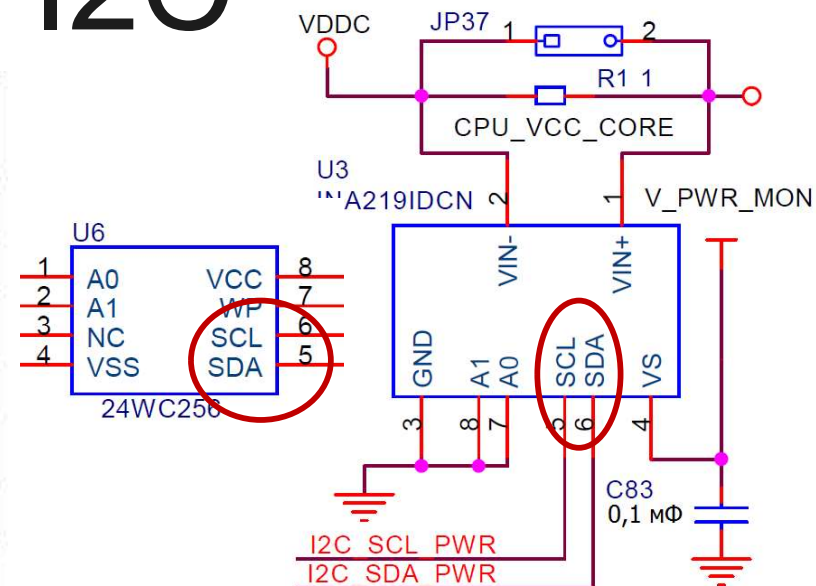
// Получить состояние приемника  
// Ожидание приема данных  
// Получить принятые данные

// Генерация STOP  
// Перерыв в линии

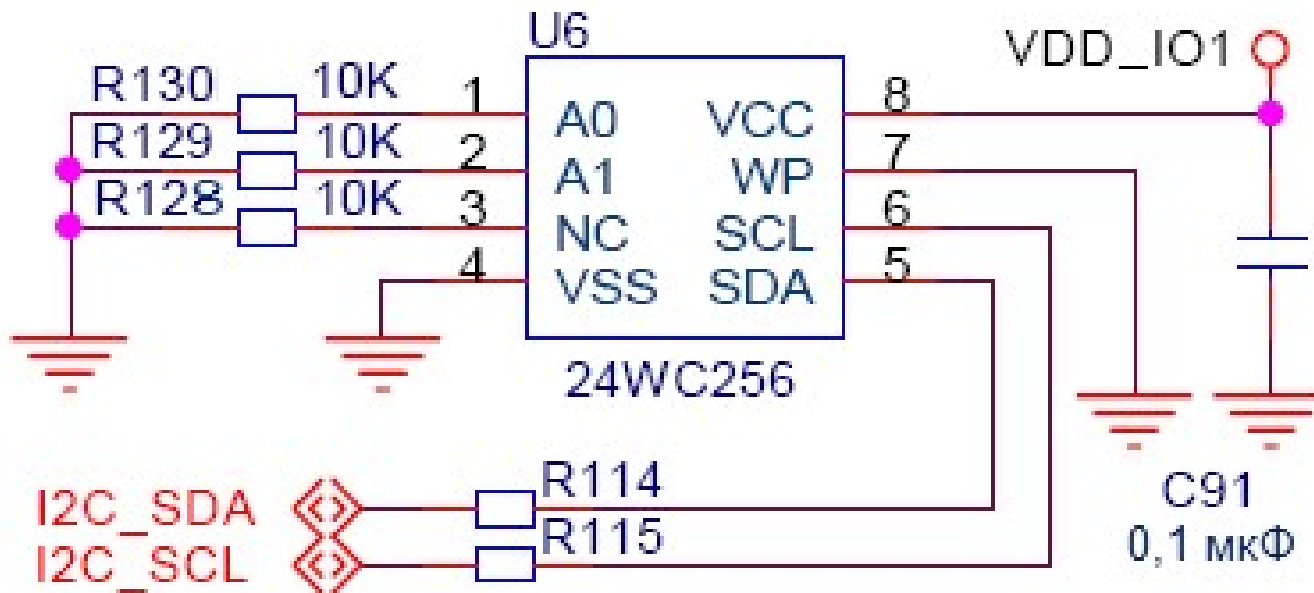


# Устройства I2C

Элемент	Адрес	Устройство
TLV320AIC3204	0x18	Аудио-кодек
INA219	0x40	Контроль питания ядра VDDC, U3
INA219	0x42	Контроль питания EMIF DC_VDD_IO2, U7
INA219	0x43	Контроль питания USB USB_VDD_IN, U21
INA219	0x44	Контроль питания платы DC_VDD_IO1, U20
INA219	0x45	Контроль питания 3,3 В, U27
INA219	0x46	Контроль питания 1,8 В, U23
INA219	0x47	Контроль питания 5 В, U24
TPS62023	0x48	PMIC - не используется
CAT24WC256X	0x50	I <sup>2</sup> C EEPROM

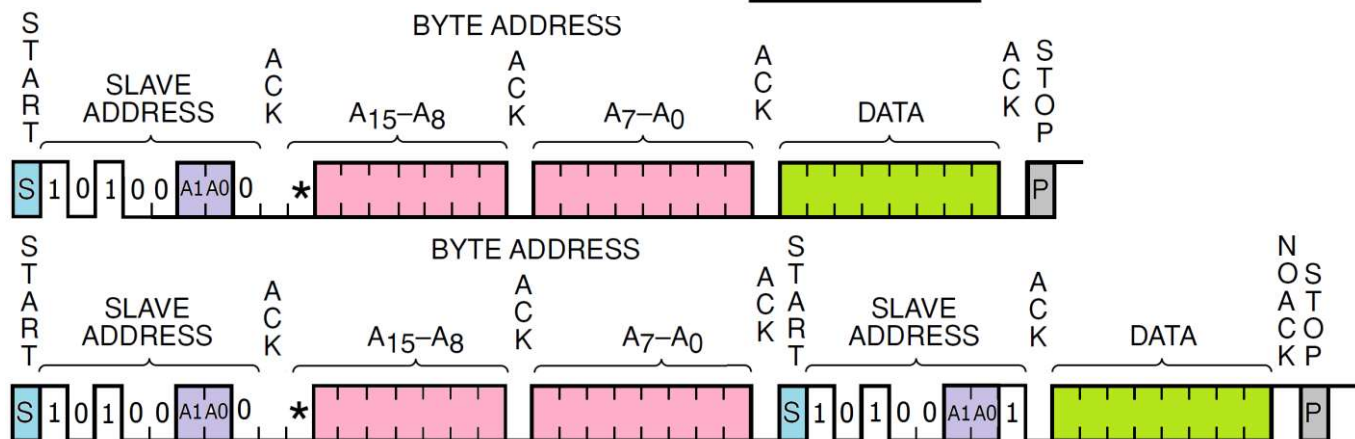
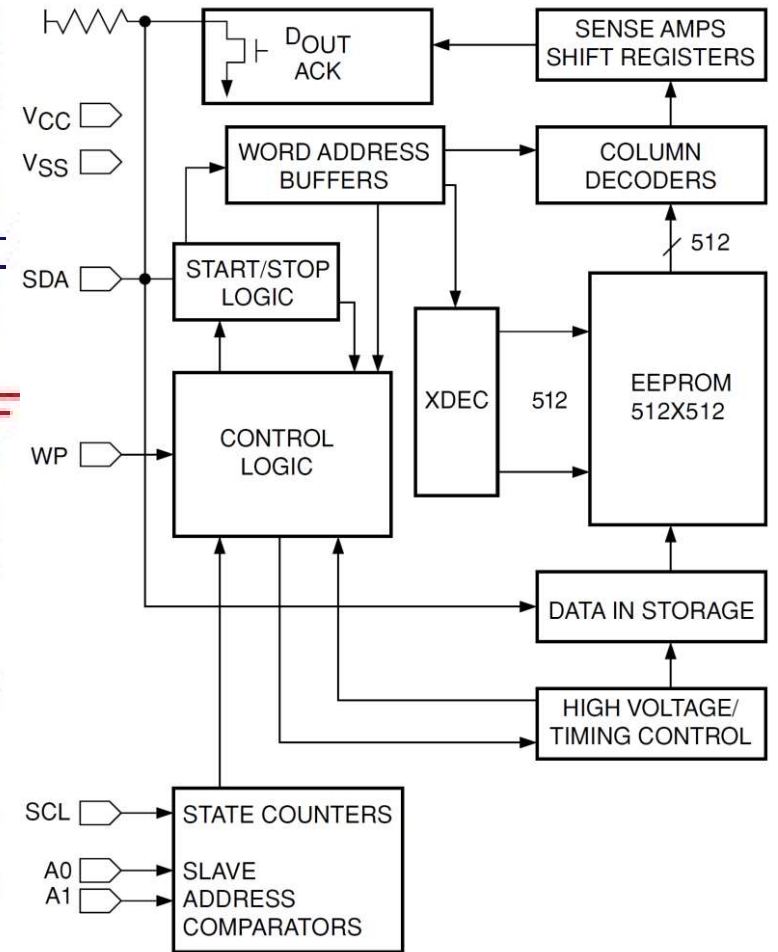
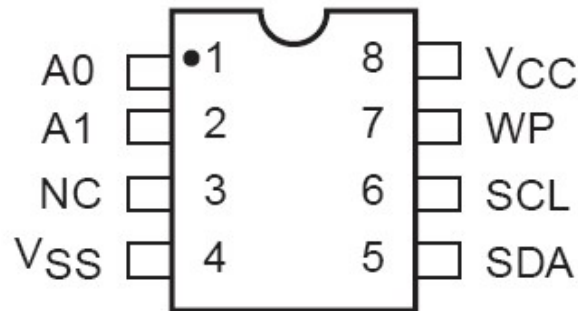


# ППЗУ CAT24WC256



Объем 256 кБит  
Частота до 1 МГц  
Адрес 10100xx

Вывод	Функция
A0, A1	Адресные входы
SDA	Последовательные данные
SCL	Тактовая частота
WP	Защита записи
VCC	+1,8 - +6,0 В
VSS	Земля
NC	Нет соединения



# Согласование уровней

