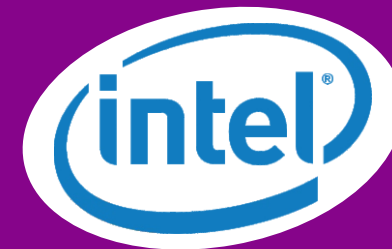




innovation**Studio**

Innovations & Startups Factory

Lomonosov Moscow State University, Faculty of Economics



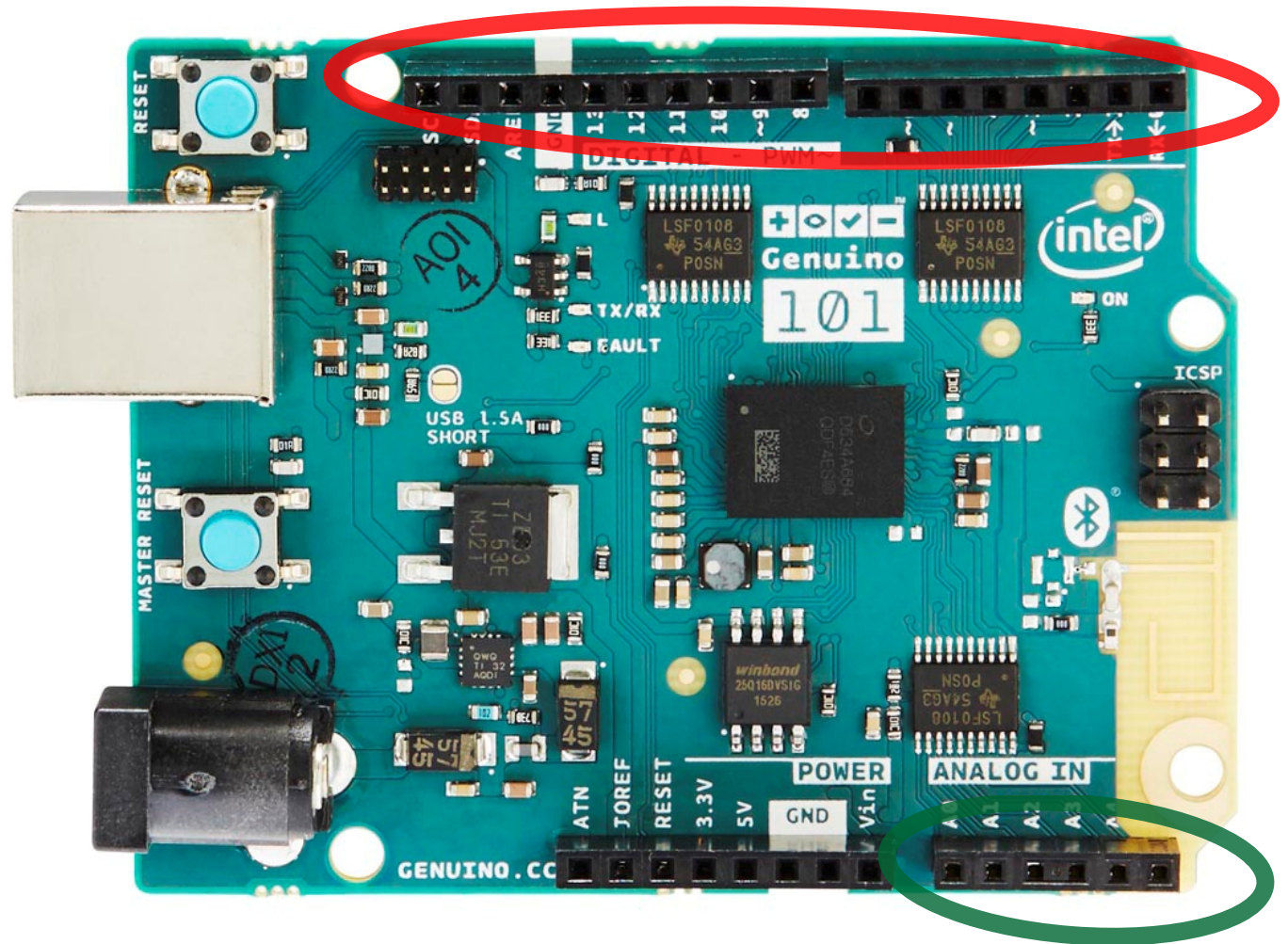
Работаем с портами и периферией

Виды портов

- Цифровые (вход, выход)
- Аналоговые (АЦП, ЦАП, ШИМ)

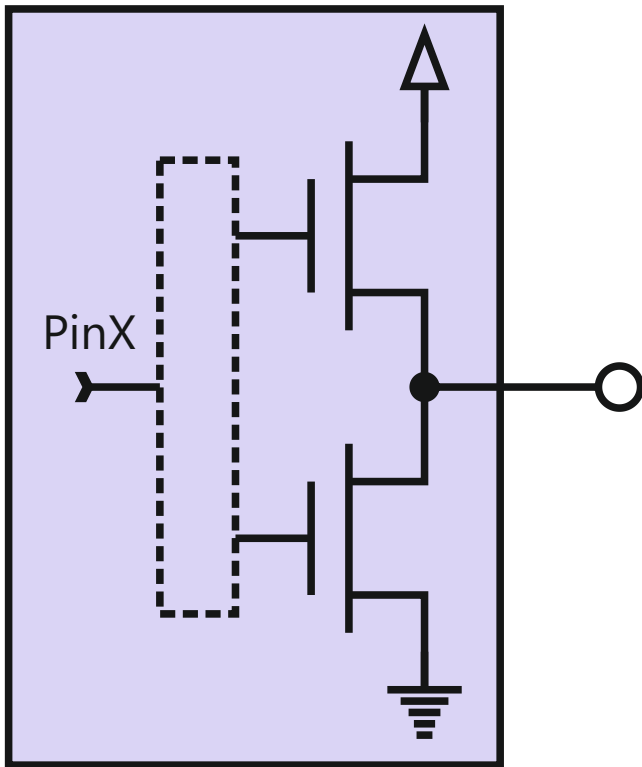
Цифровые I/O,
ШИМ

Входы АЦП

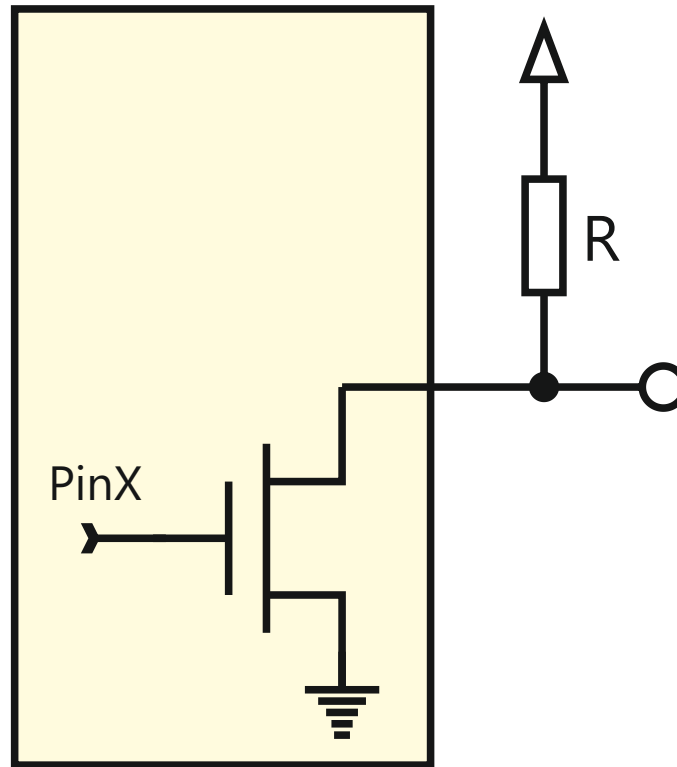


Виды цифровых выходов

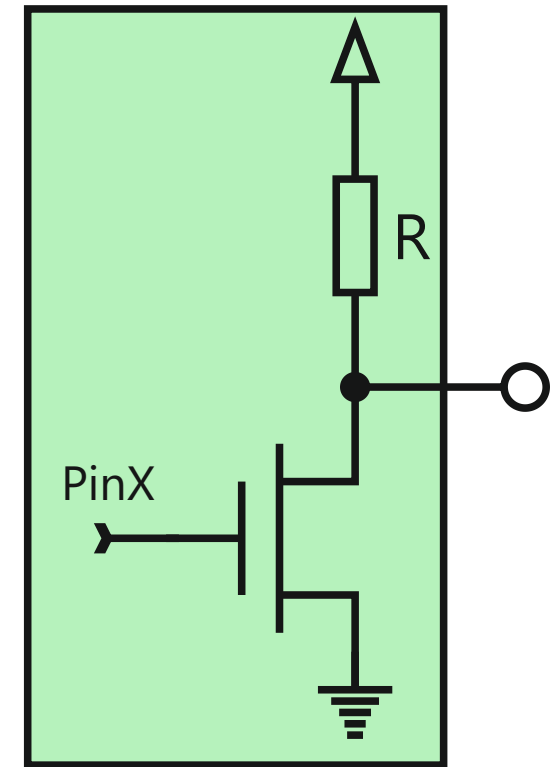
Push-pull



Open drain

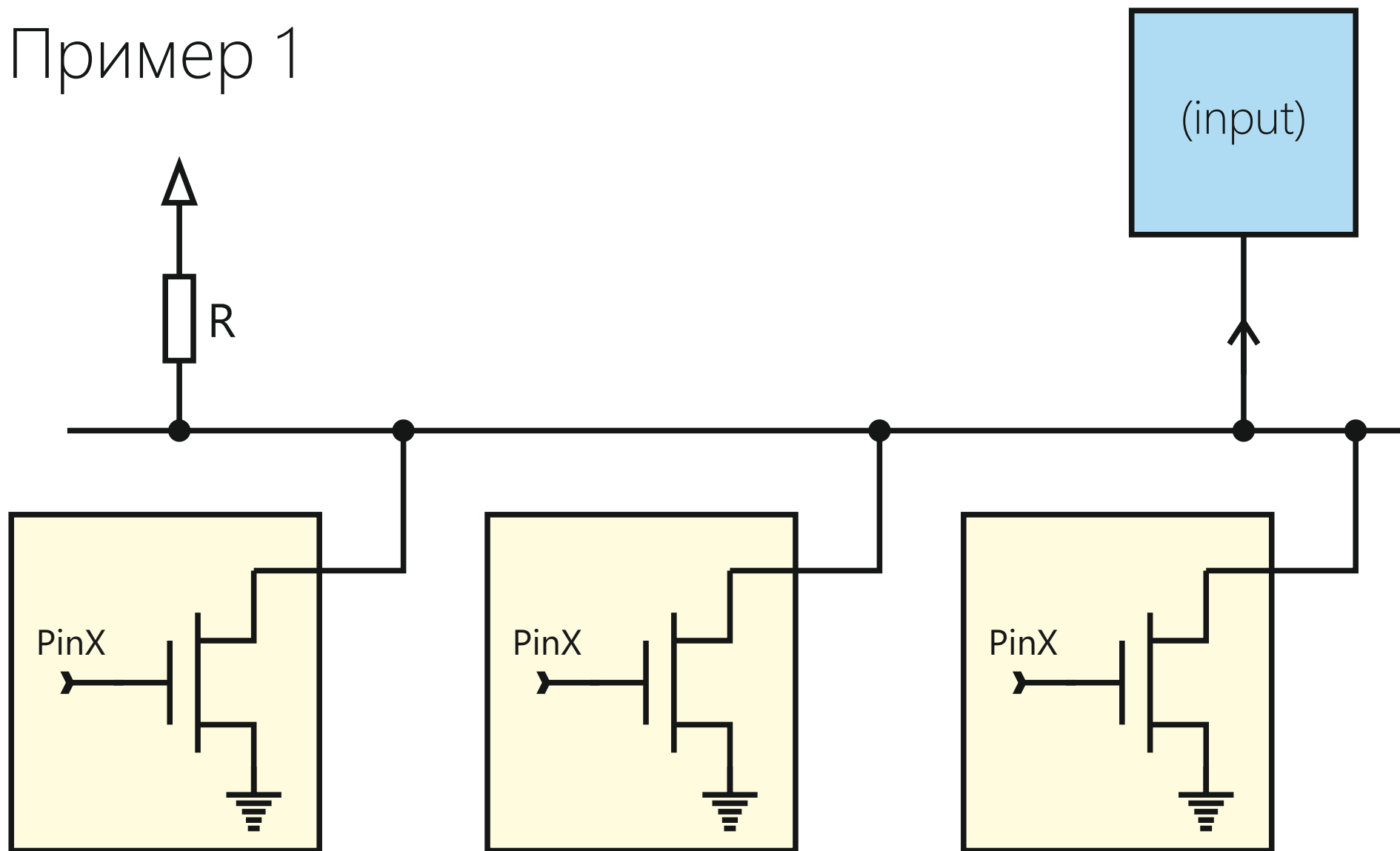


Pull-up



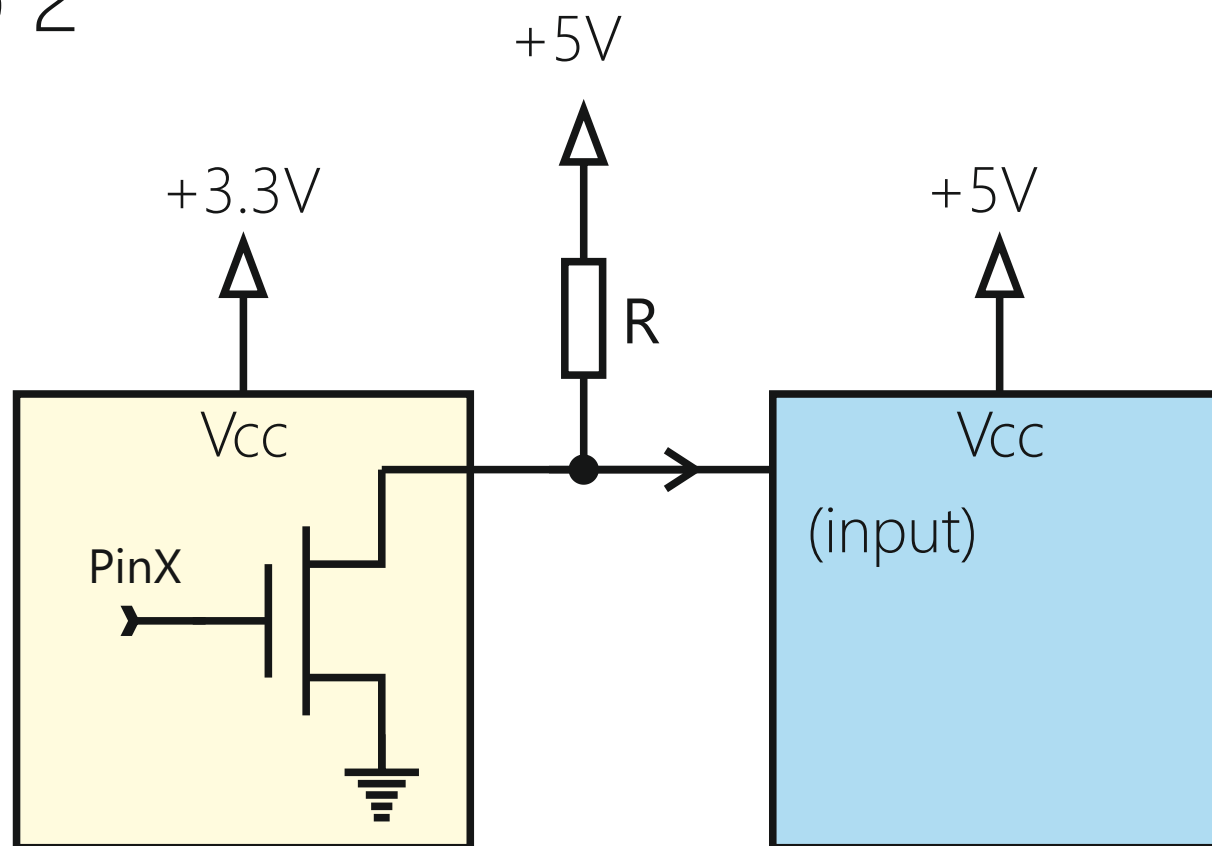
Зачем нужен open drain?

Пример 1



Зачем нужен open drain?

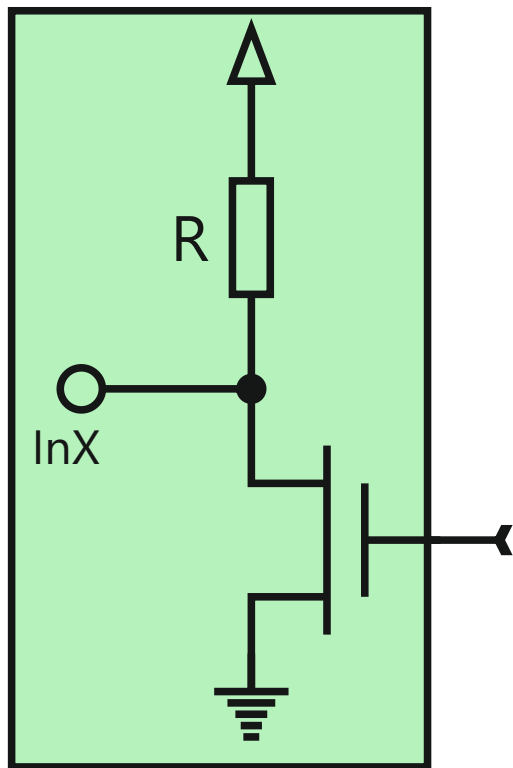
Пример 2*



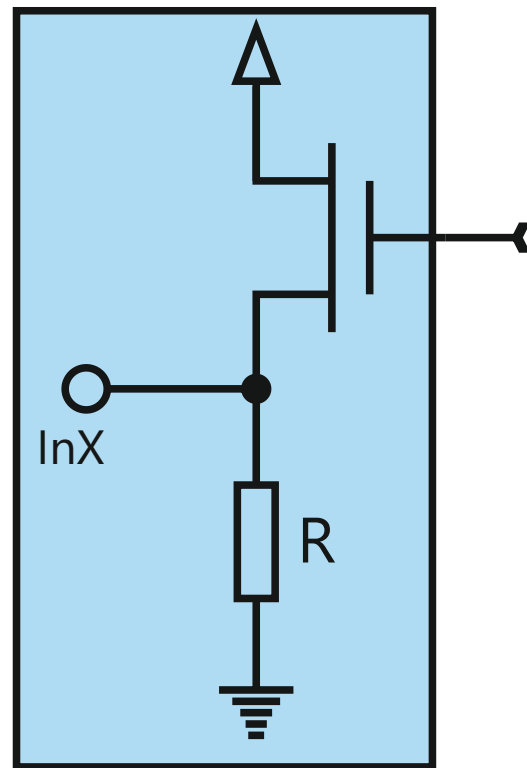
*При условии, что производитель разрешает так делать;
при обратном сценарии вход должен быть "5V tolerant"

Виды цифровых входов

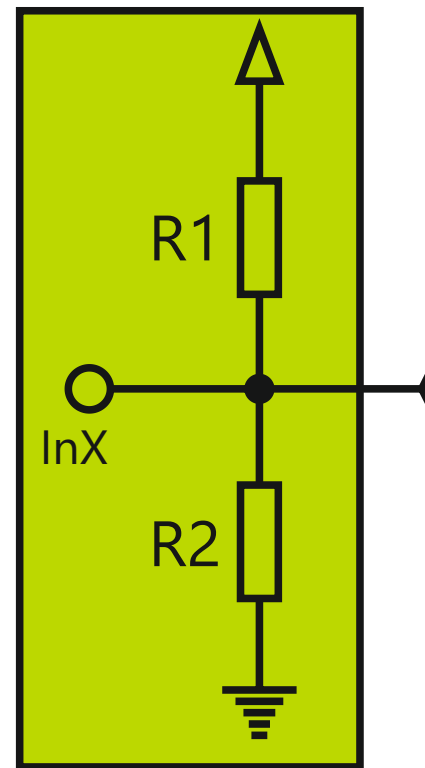
Pull-up



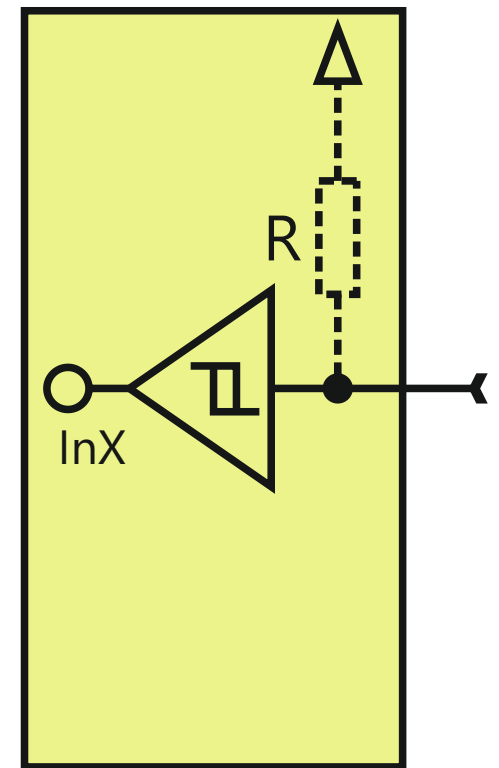
Pull-down



Pull-up
and down

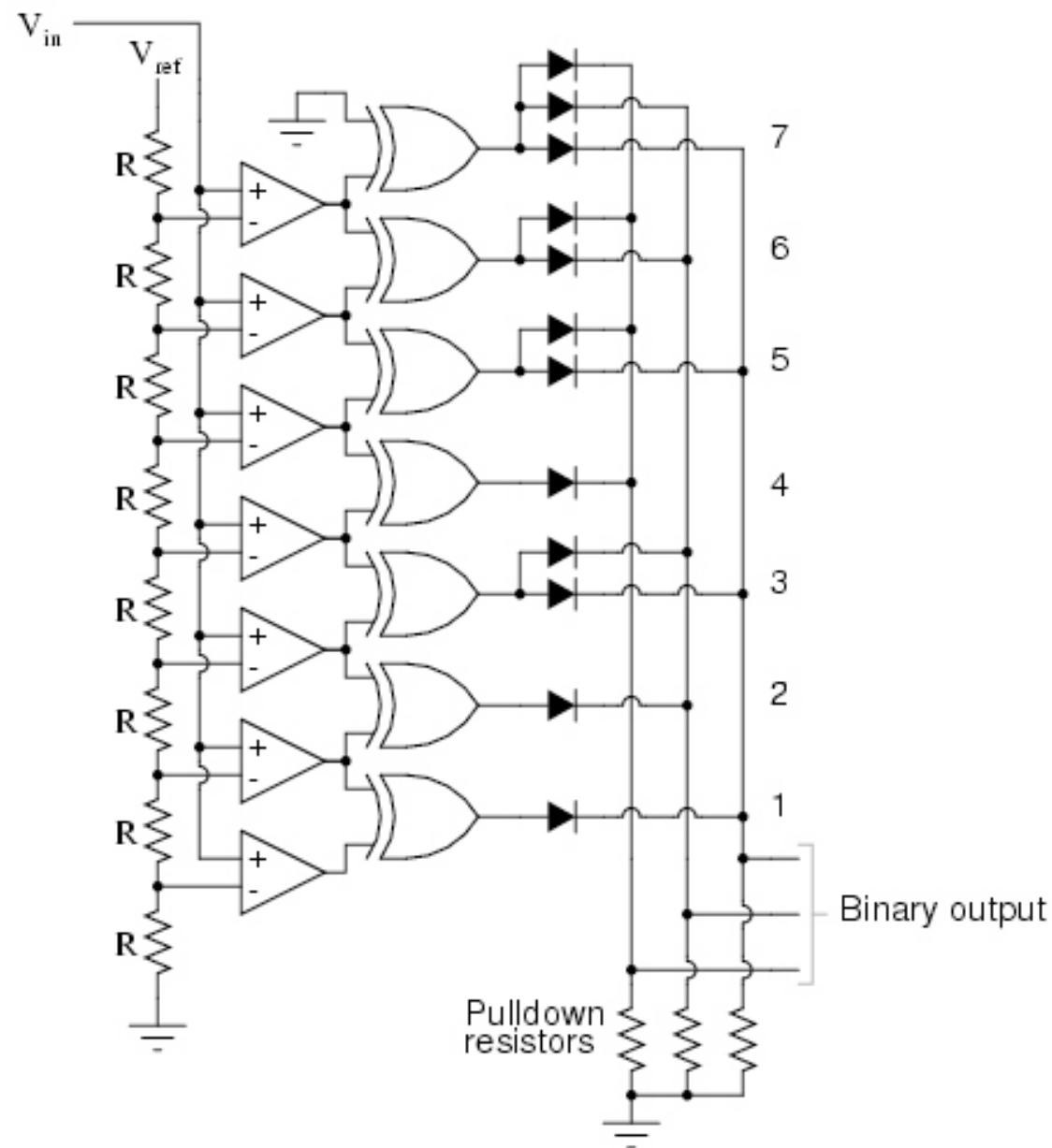


Schmitt



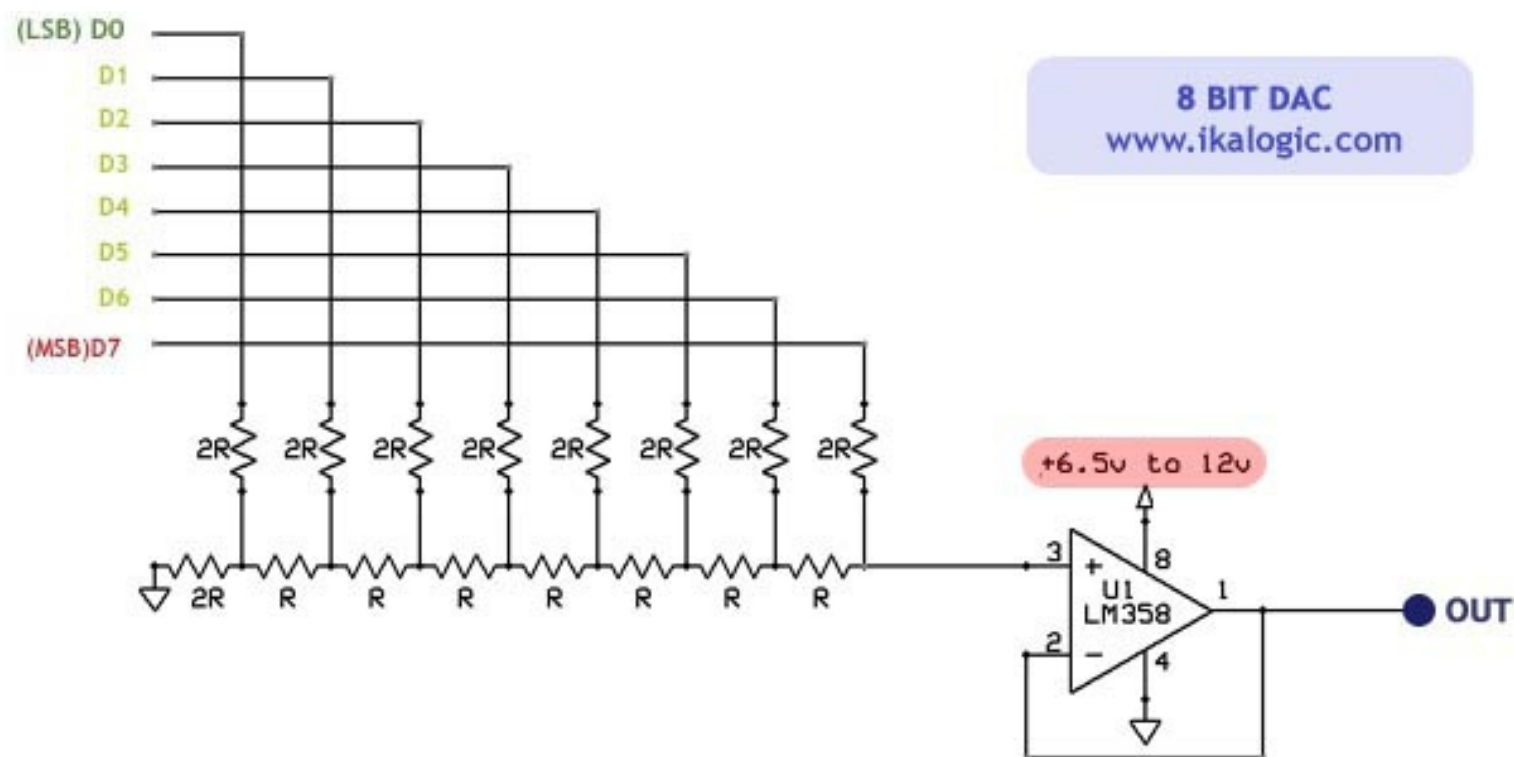
Коротко об АЦП

- Во встроенных АЦП, как правило, выход варьируется от нуля при $V_{in}=0$ до максимального двоичного числа разрядности АЦП при $V_{in}=V_{ref}$
- Не забываем про пин **AREF** ($V_{ref} \neq V_{cc}$)
- Если нужна большая разрядность, другая полярность, больше входов - ставим внешний АЦП
- Не стоит забывать про влияние входных каскадов на разрешение и качество работы АЦП



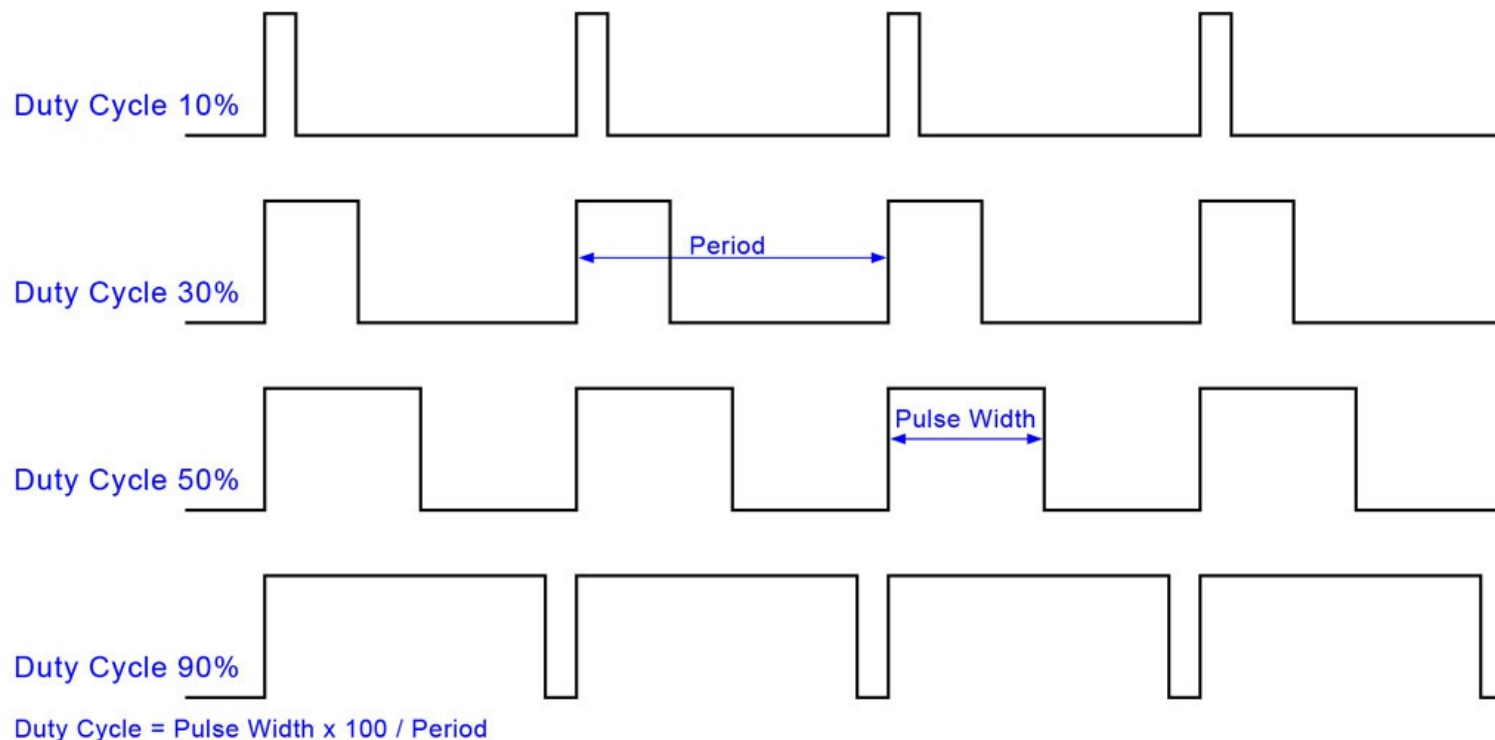
Коротко о ЦАП

- Выход варьируется от $V_{out}=0$ при $DAC_{in}=0$ до $V_{out}=V_{ref}$ при DAC_{in} , равном максимальному числу для его разрядности
- Встречается далеко не во всех микроконтроллерах, поэтому чаще всего используют внешние
- Не упускаем из внимания выходные каскады, импедансы и быстродействие



ШИМ

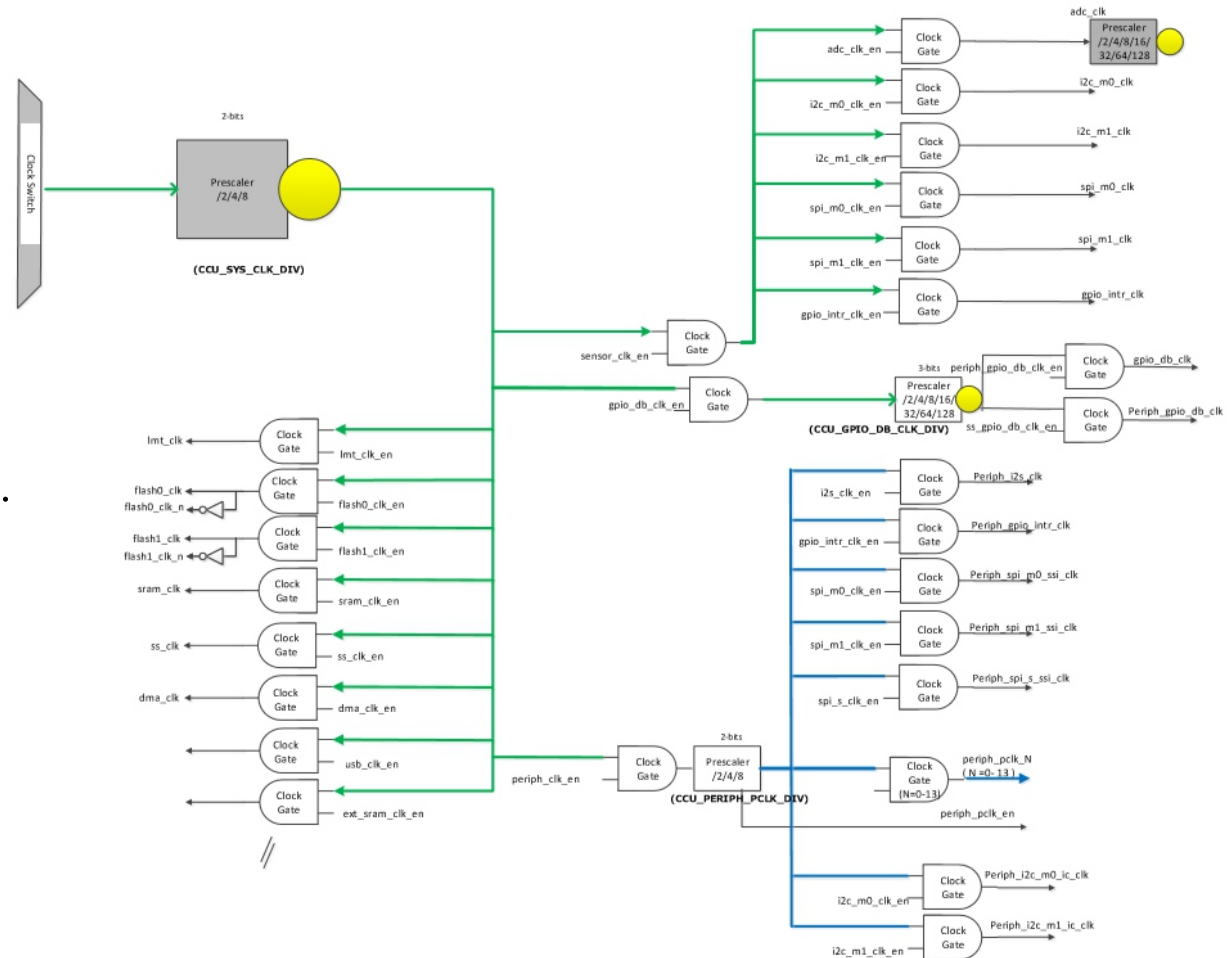
- Носитель информации - **ширина импульса** (в единицах времени или процентах от периода), или скважность, она же **duty cycle**.
- Хотя выход и относят к аналоговым, но он чаще всего мультиплексирован с цифровыми выходами на микроконтроллере
- Не забываем, что есть не только ширина импульса, но и **частота**



0 тактировании ШИМ

- В микроконтроллерах ШИМ, как правило, тактируется от внутреннего осциллятора для таймеров через делитель частоты. Частоту опорного генератора можно делить только на целое (чаще - кратное двум) число, поэтому выбор ограничен.

- У разных выходов ШИМ могут быть разные диапазоны рабочих частот. Например, в некоторых Arduino это так. Лучше это учесть до того, как начать что-то собирать.



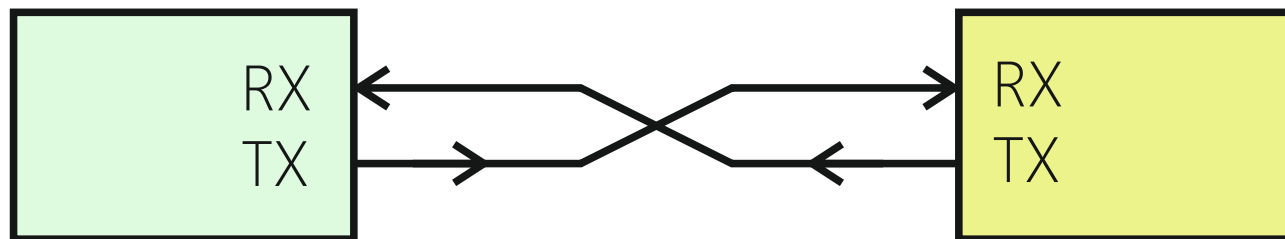
Низкоуровневые цифровые интерфейсы и протоколы

UART

- Чаще всего используется для передачи текстовых (ASCII) данных
- Основные параметры: **количество бит данных** (обычно 8 или 7), **скорость** (baud rate), **бит чётности** (parity - even/odd/no parity), **длительность стоп-бита**

Например, запись "115200, 8, N, 1" обозначает 115200 бод, данные по 8 бит, без бита чётности, длительность стопового бита равна 1 битовому интервалу.

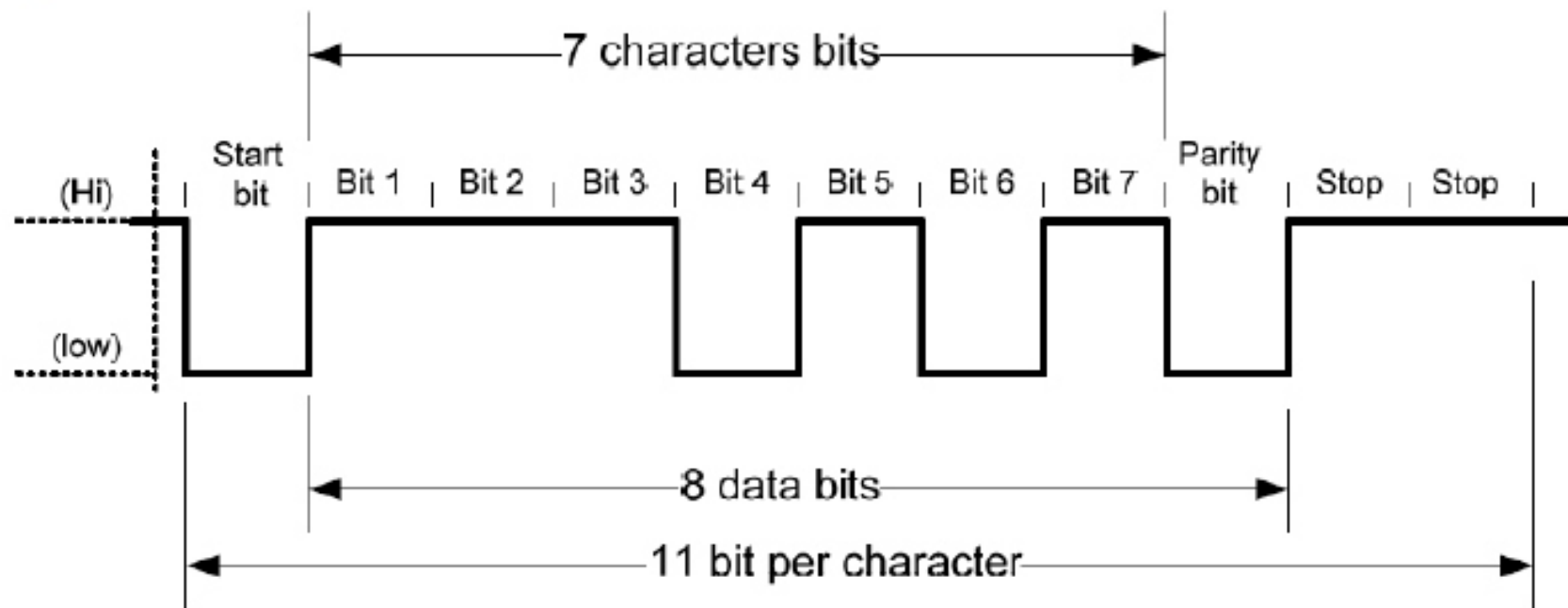
- Это асинхронный протокол. Получатель и отправитель заранее должны знать, с какой скоростью и какими параметрами будет идти передача.
- Конец строки может быть разный - **CRLF** (\r\n), **LF** (\n), **CR** (\r), так исторически сложилось в Windows, Linux и macOS соответственно.
- Разновидности - **USART**, **1-Wire**; высокоуровневые - **RS232**, **RS485**



UART

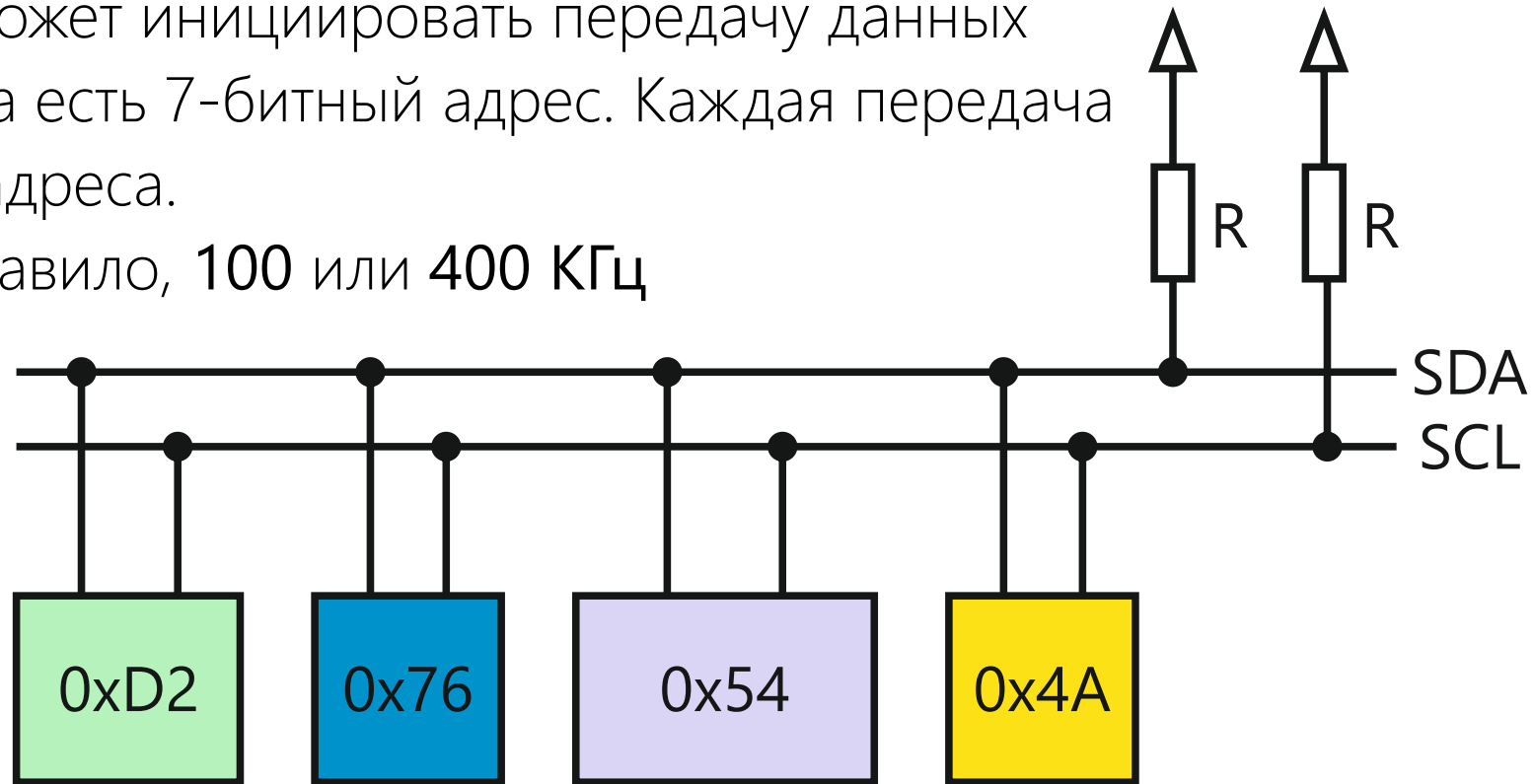
Пример передачи байта (буква "W")

Figure 14-3. Character ASCII "W" serial transmission.

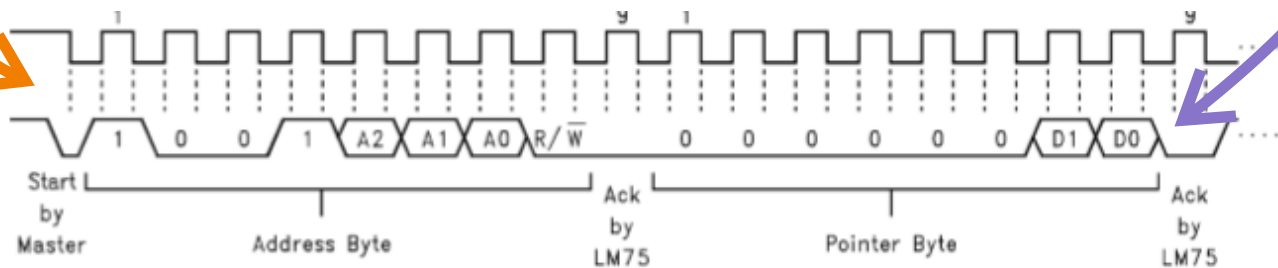


- Используется для синхронной передачи небольших (как правило) объёмов данных
- Присутствует практически во всех периферийных микросхемах и датчиках
- Две линии - данные (**SDA**) и тактовый сигнал (**SCL**)
- Предполагает наличие шины с множеством устройств
- Любое устройство может инициировать передачу данных
- У каждого устройства есть 7-битный адрес. Каждая передача данных начинается с адреса.
- Скорость SCL, как правило, **100** или **400** КГц

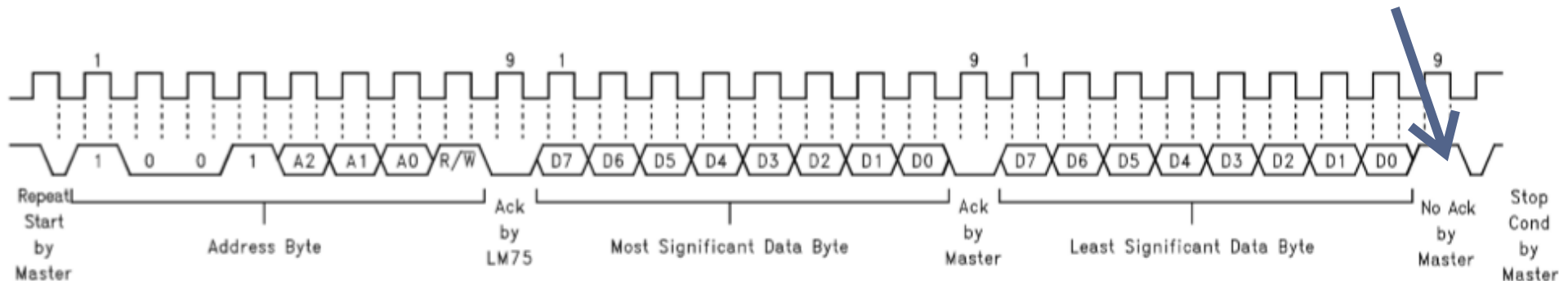
- Разновидности -
CAN, SMBUS,
PMBUS,
«пиратский» I²C
(TWI)



- Устройство, которому надо совершить передачу (Master), генерит старт передачи на шине, и генерирует тактовый сигнал на протяжении всего обмена. Пин SDA попеременно становится входом или выходом у Master и Slave.
- Если передача идёт успешно, то Slave генерит **ACK** (acknowledge) после каждого байта. Если что-то идёт не так, то на шине остаётся **NAK** (not acknowledge) и передача прерывается.



В конце передачи Master генерит NAK и Stop на шине



Разбор передачи данных по I²C

Master (write): Start. Устройство с таким-то адресом, сейчас кое-что запишу

Slave: ACK ("ok!")

Master: (пишет адрес регистра, который его интересует)

Slave: ACK

Master: Конец передачи. Stop.

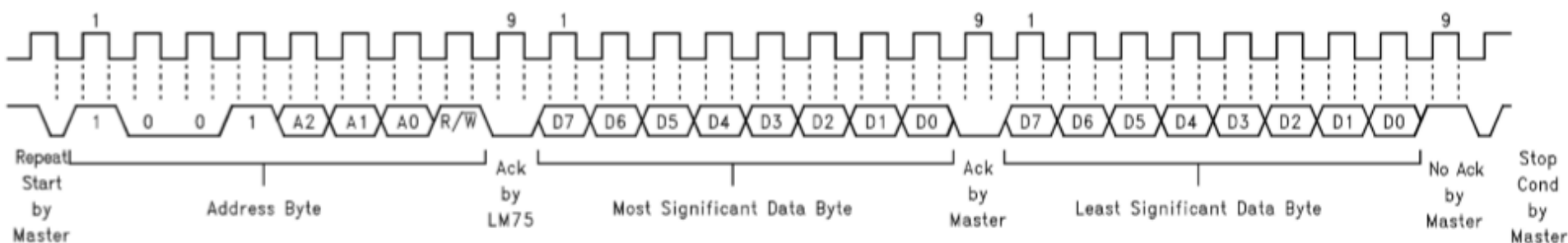
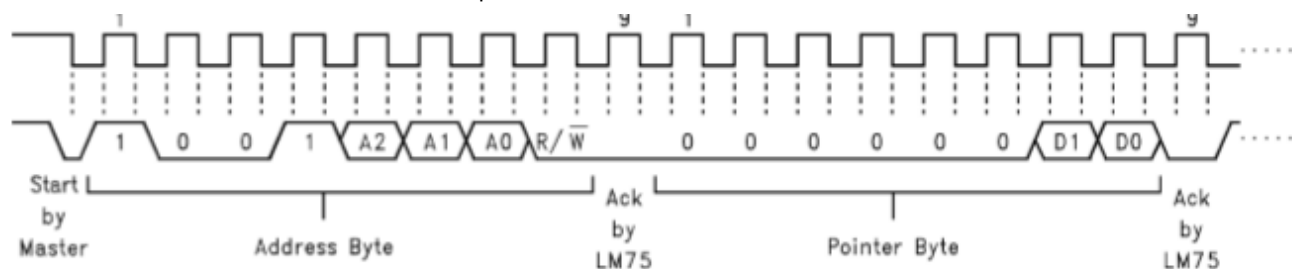
Master (read): Start. Устройство с таким-то адресом, дай мне байт из регистра, который я только что сказал

Slave: ok! (отправляет байт)

Master: ACK. Давай следующий байт

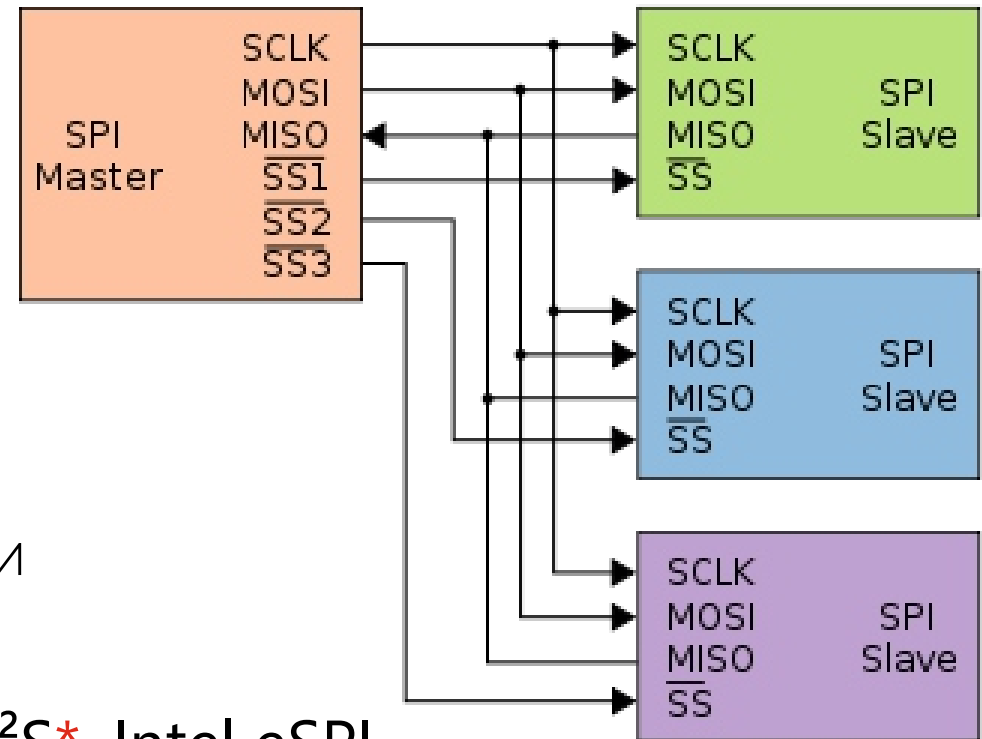
Slave: ok! (отправляет байт)

Master: Хватит. NAK. Stop



SPI

- Используется для синхронной передачи данных. Как правило, больших объёмов или на более высокой скорости, чем I²C.
- Чаще всего встречается в микросхемах памяти (в том числе и всех картах SD), а также в ПЛИС ввиду простоты реализации на стандартной логике
- Иногда присутствует в микросхемах вместе с I²C в качестве опции
- Тоже шина, как и I²C, но с дуплексной передачей и более примитивной адресацией.
- Параметры: скорость, полярность тактового сигнала (**CPOL**) и фронт тактового сигнала для приёма/передачи данных (**CPHA**)

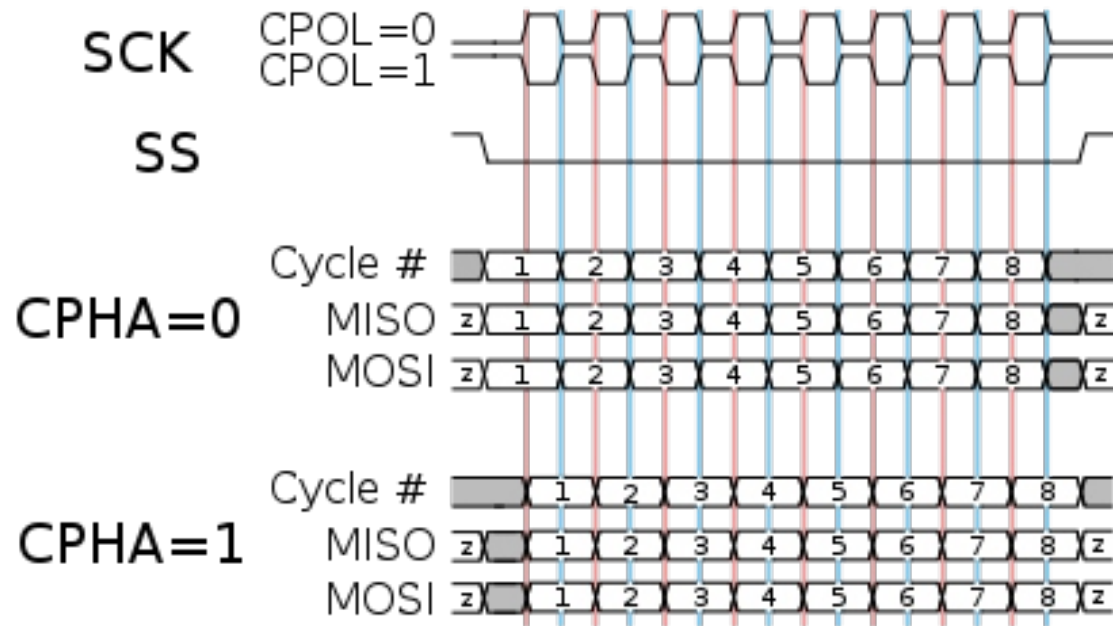


Разновидности: QuadSPI, Queued SPI, I²S*, Intel eSPI

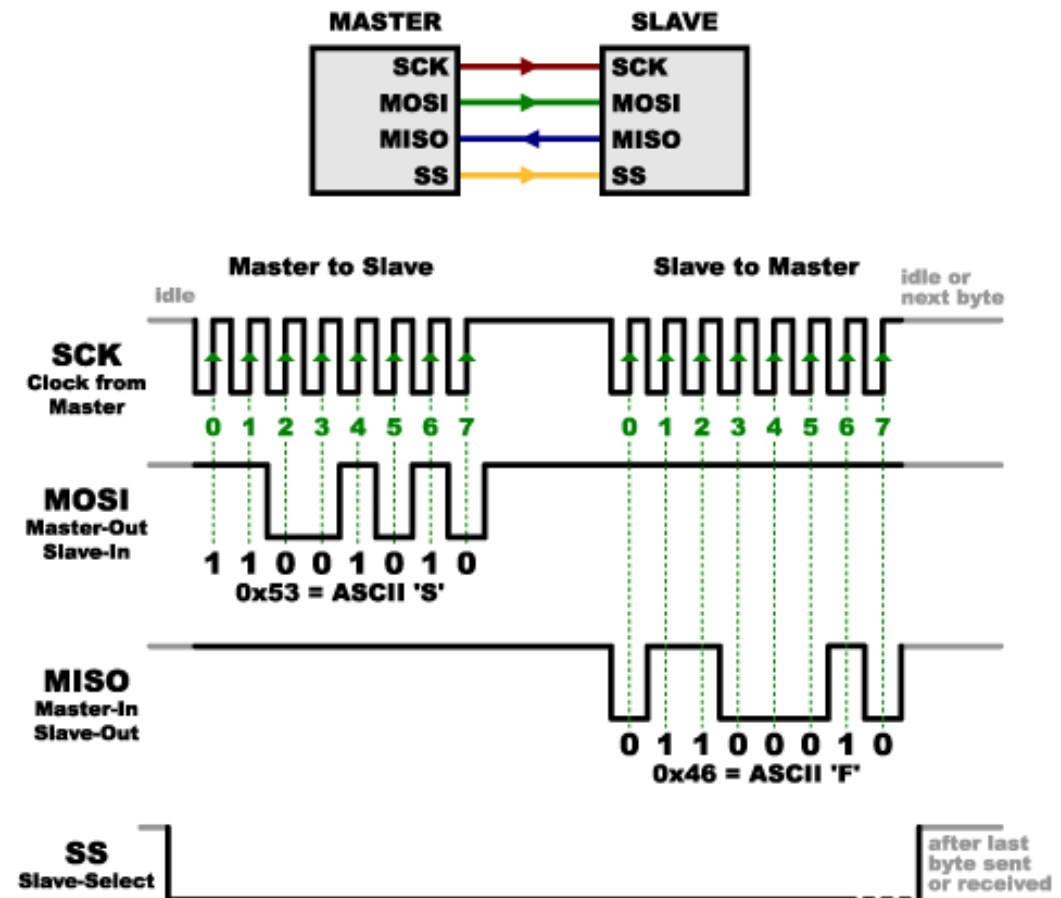
*похож, но формально является самостоятельным интерфейсом

SPI

Параметры CPOL и CPHA

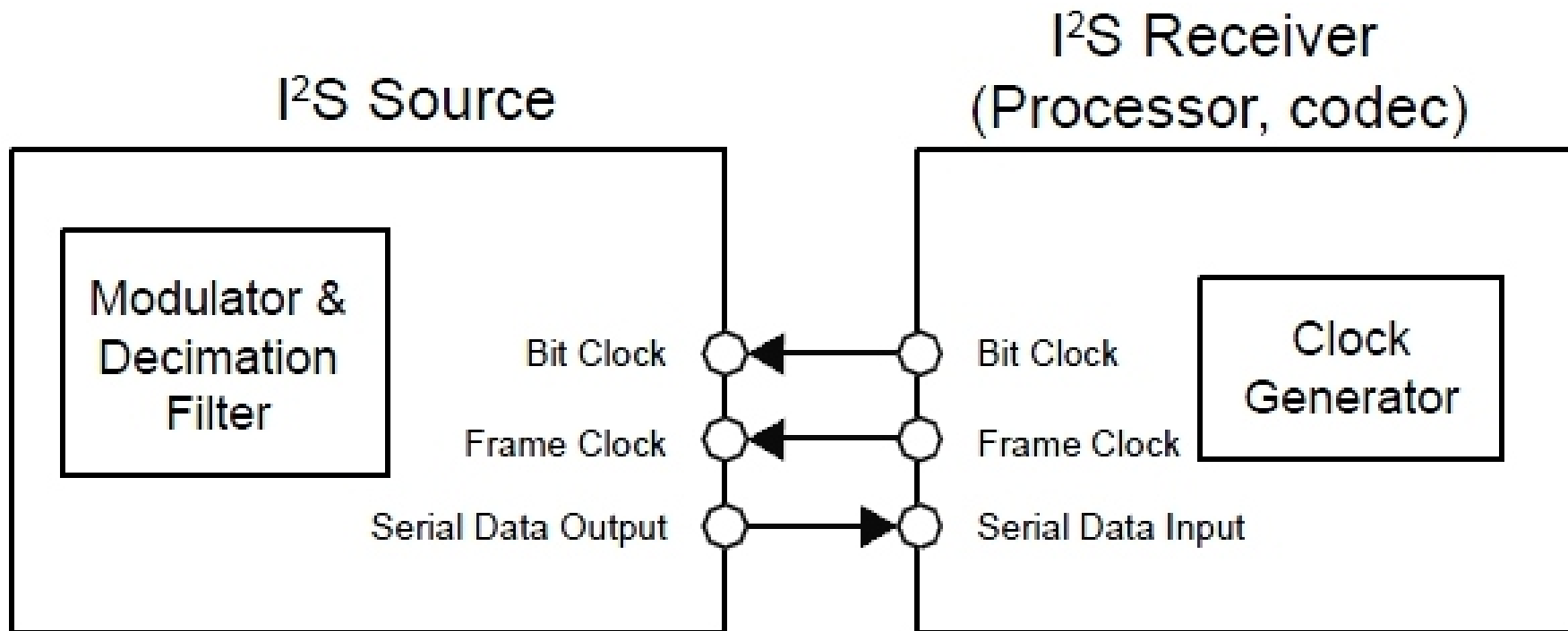


Пример записи и чтения байта



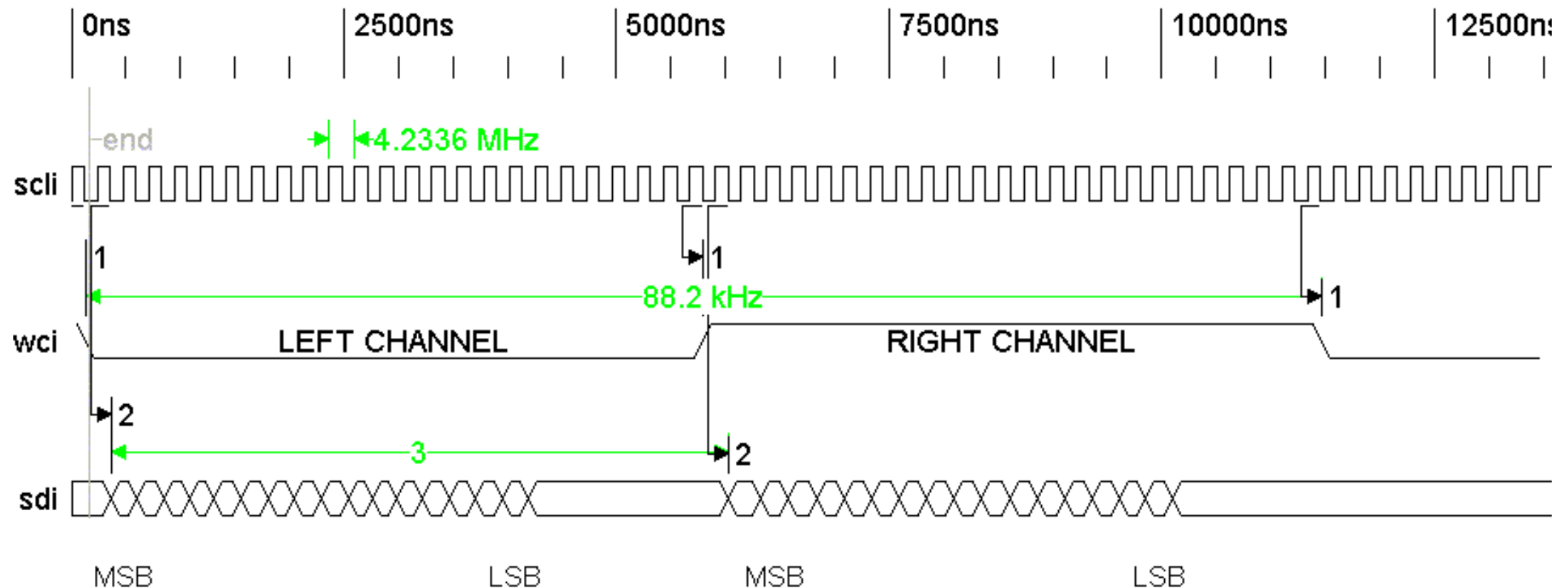
I²S

- Похож на SPI, используется для передачи цифрового аудио в реальном времени
- Чаще всего получатель сигнала (от микроконтроллера) - кодек, который уже преобразует этот сигнал в аналоговый. Источник сигнала - это, как правило, конвертер (АЦП).
- Этот интерфейс есть практически во всех микроконтроллерах



Частота SCL данных строго связана с частотой дискретизации звука и разрядностью звука и числом каналов (например, 2 для стерео)

$$f = \text{bitrate} * F_d * 2$$



Вопросы?