|  |  |
| --- | --- |
| Курс: | Основы электроники |
| Модуль 9: | Цифровые интегральные схемы средней и большой степеней интеграции |

|  |  |  |
| --- | --- | --- |
| Авторы | Сапунов Арсений Русланович  Иванов Константин Александрович, к.т.н. | |
|  |  | |
| Рецензенты |  | |
|  |  | |
|  |  | |
| Длительность  (рекомендуемая) | | 2 часа |
|  | |  |
| Главная цель | | По окончании изучения темы обучаемый будет иметь общие сведения о цифровых ИМС средней и большой степеней интеграции, а также о некоторых популярных интерфейсах цифровой передачи данных, уметь применять и использовать их |
|  | |  |
| Промежуточные цели | | * Знать описание и определение счётчиков * Знать описание и определение сдвиговых регистров * Знать описание и определение мультиплексоров и демультиплексоров * Знать описание и определение ЦАП структуры R2R * Знать описание и определение АЦП на примере устройства последовательного приближения * Знать концепцию гибкой и жесткой логики * Знать структуру и базовые блоки микроконтроллера, его определение в рамках его представления как «серого ящика» * Знать и уметь использовать цифровые и аналоговые входы и выходы микроконтроллера * Знать и уметь применять распространённые цифровые интерфейсы, такие как I2C, SPI и UART |

1 Предмет лекции

**ИМС средней и большой степени интеграции** представляют собой ключевые элементы современных электронных устройств. Они позволяют создавать любые устройства, вплоть до высокопроизводительных устройств, такие как микропроцессоры и системы на кристалле (SoC), которые могут выполнять сложные вычисления и обрабатывать большие объемы данных. Такие системы имеют **замечательную производительность** и **малое потребление энергии**, что, в частности, делает их идеальными для мобильных и встраиваемых систем.

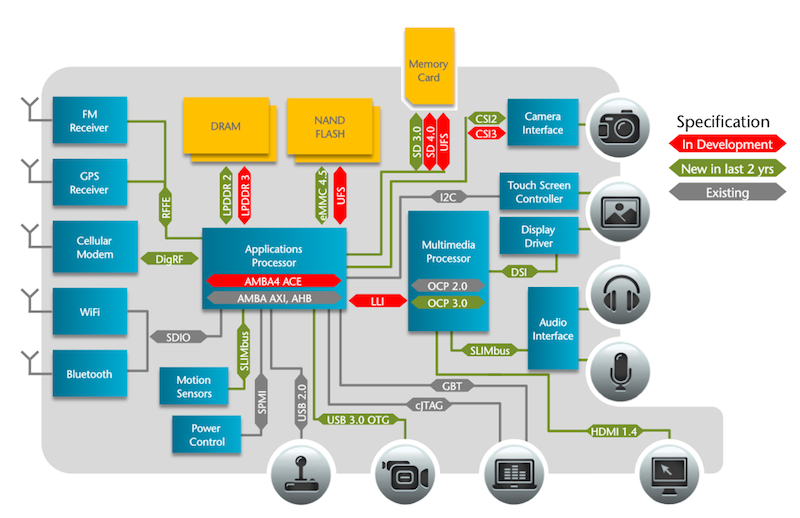


Рисунок 1 – Структура системы на кристалле (SoC)

С увеличением степени интеграции уменьшается размер устройства и повышается его функциональность. Однако, с этим также возникают проблемы, такие как **тепловыделение** и **сложности в проектировании**, что требует применения современных технологий и методов проектирования.



Рисунок 2 – SoC URX850 для линий связи

2 Счетчики

Сложные ИМС, такие как МК и SoC строятся на базе более простых функциональных блоков. Одним из них являются счетчики.

**Счетчики** — это цифровые устройства, которые используются для **подсчета импульсов** или событий. Они могут быть как синхронными, так и асинхронными. **Синхронные** счетчики обеспечивают одновременное изменение состояния всех разрядов, тогда как **асинхронные** изменяют состояние последующих разрядов последовательно. Счетчики могут быть как **увеличивающими**, так и **уменьшающими**, в зависимости от логики, используемой для управления их состоянием.

Внутренняя структура счетчика обычно включает в себя ряд **триггеров**, которые представляют собой базовые элементы хранения данных. Каждый триггер отвечает за один разряд счетчика. При поступлении импульсов на вход, триггеры переключаются, что приводит к изменению состояния счетчика. В зависимости от архитектуры, счетчики могут иметь дополнительные элементы управления, такие как **сброс** или **установка**.

Счетчики находят **широкое применение в цифровых системах**, включая таймеры, делители частоты и устройства для измерения времени.

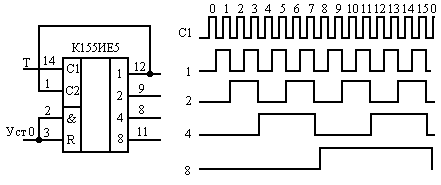


Рисунок 3 – Включение счетчика К155ИЕ5 и временная диаграмма

Один из вариантов УГО и счетчика на примере отечественной ИМС К155ИЕ5, а также временная диаграмма для его выводов показаны на рис. 3. На рис. 4 показана диаграмма для счетчика К176ИЕ2.

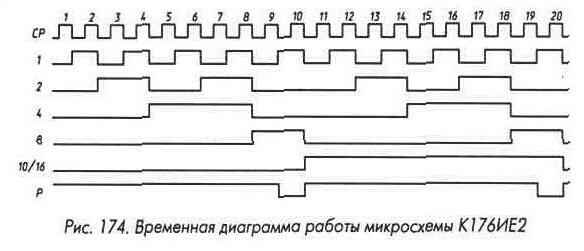


Рисунок 4 – Временная диаграмма ИМС К176ИЕ2

3 Сдвиговые регистры

**Сдвиговые регистры** — это цифровые устройства, которые используются для хранения и обработки двоичных данных. Они представляют собой последовательности триггеров, которые **могут сдвигать данные влево или вправо при каждом тактовом импульсе**. Сдвиговые регистры находят широкое применение в цифровых схемах, включая преобразование данных, где они могут преобразовывать параллельный код в последовательный и наоборот. Это особенно полезно в системах, где необходимо уменьшить количество используемых линий передачи данных, например, в интерфейсах связи. Вариант УГО для прибора представлен на рис. 5.

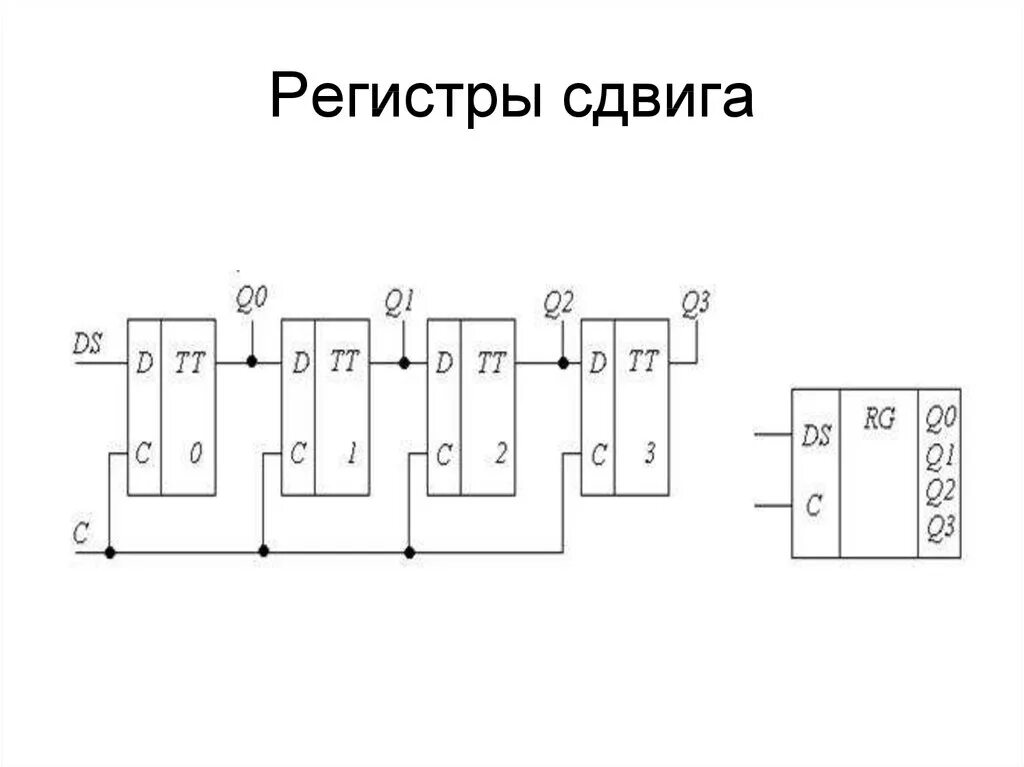


Рисунок 5 – Вариант УГО сдвигового регистра

Одной из ключевых функций сдвиговых регистров является возможность **размножения портов микроконтроллеров**. С помощью сдвиговых регистров можно управлять множеством выходов, используя всего несколько сигналов от микроконтроллера. Например, один сдвиговый регистр может управлять 8-16 выходами, что позволяет экономить ресурсы микроконтроллера. Это делает сдвиговые регистры крайне полезными в приложениях, где требуется управление множеством устройств или индикаторов.

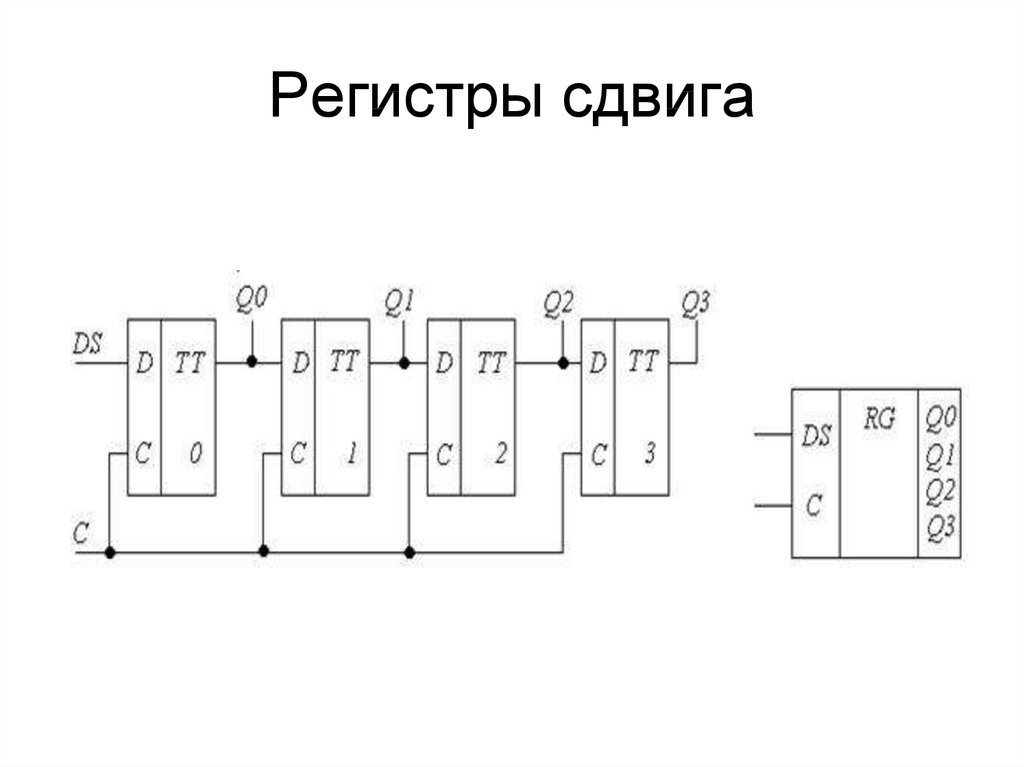


Рисунок 6 – Внутреннее устройство регистра

Кроме того, сдвиговые регистры также могут использоваться для **временного хранения информации**. Они позволяют хранить данные на определенное время, а затем передавать их в нужный момент. Это свойство делает их полезными в различных приложениях, таких как цифровые фильтры, преобразователи и системы передачи данных.

4 Мультиплексоры и демультиплексоры

**Мультиплексоры** (MUX) и **демультиплексоры** (DEMUX) являются важными компонентами цифровой электроники, используемыми для **управления потоками данных.** Мультиплексор позволяет выбрать один из нескольких входных сигналов и передать его на выход, используя управляющие сигналы. Это позволяет экономить ресурсы и упрощает проектирование схем, так как несколько входов могут быть объединены в один выход.

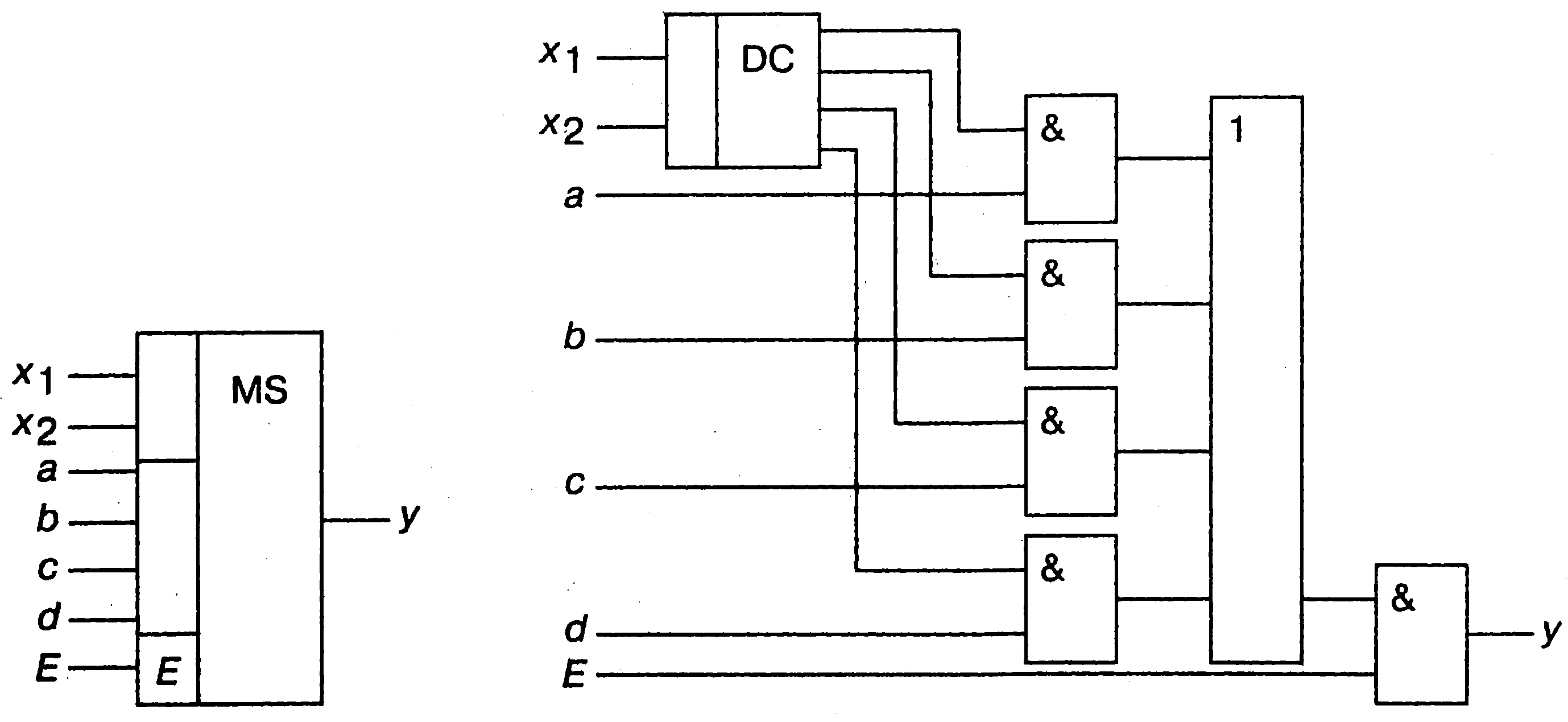


Рисунок 7 – Вариант УГО мультиплексора

Внутренняя структура мультиплексора состоит из логических элементов, таких как И- и ИЛИ-элементы, которые обрабатывают входные данные в зависимости от значений управляющих сигналов. Например, 2x1 мультиплексор имеет два входа и один выход, а его работа определяется одним управляющим сигналом, который указывает, какой из входов будет передан на выход. Варианта УГО мультиплексора на 4 канала и его внутренняя структура показаны на рис. 7 и 8 соответственно.

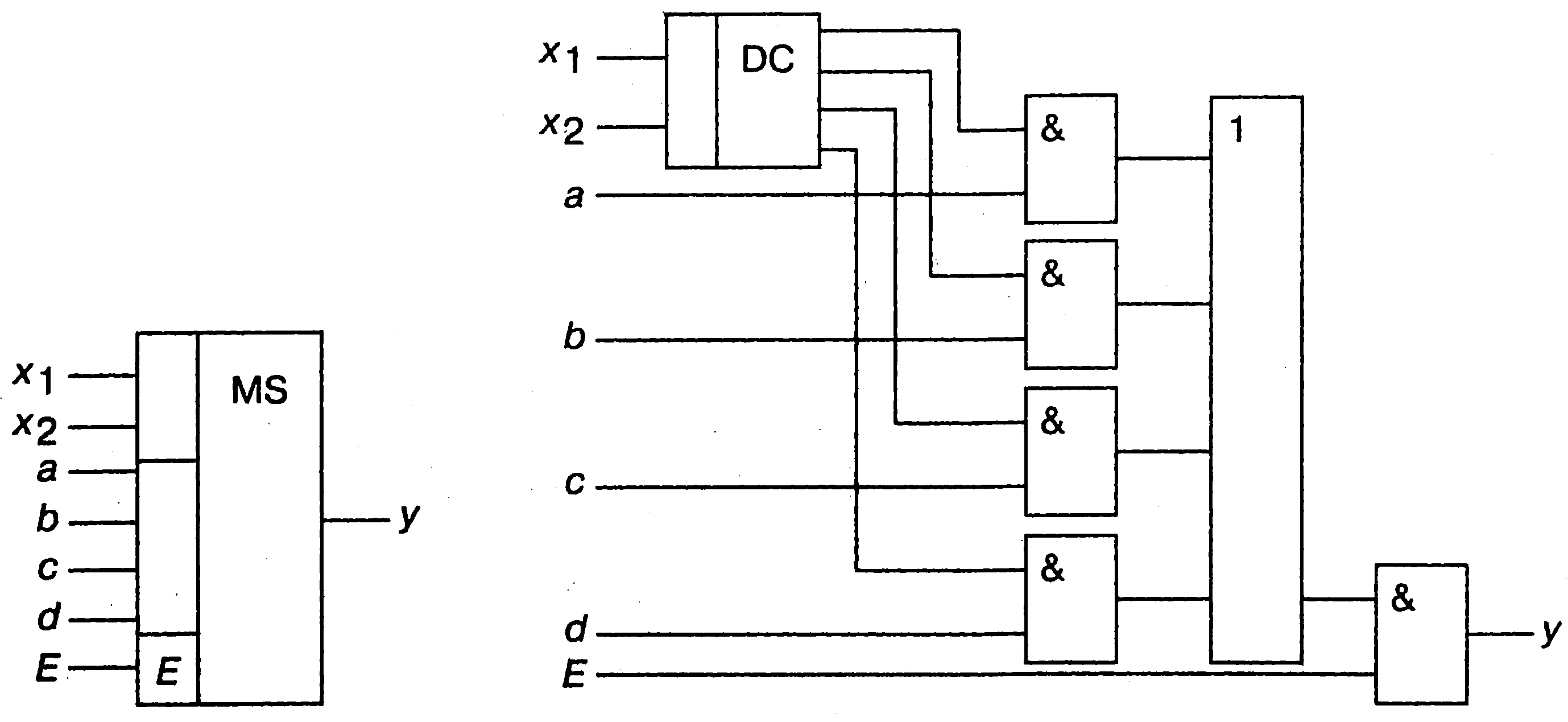


Рисунок 8 – Внутренняя структура мультиплексора структуры 4-1

**Демультиплексор**, наоборот, принимает один входной сигнал и распределяет его на несколько выходов. Это полезно в ситуациях, когда необходимо направить один сигнал на разные устройства или компоненты.

5 ЦАП

**Цифро-аналоговый преобразователь** (ЦАП, DAC) структуры **R2R** представляет собой устройство, которое **преобразует цифровые значения в аналоговые сигналы**. Эта структура основана на использовании резисторов с двумя значениями: R и 2R. Каждый бит цифрового входа управляет соответствующим резистором, создавая делитель напряжения, который формирует аналоговый выходной сигнал.

**Разрядность** ЦАП определяет количество бит, которые он может обрабатывать. Например, 4-битный ЦАП может принимать значения от 0 до 15, создавая 16 различных уровней выходного сигнала. Чем выше разрядность, тем более точным и плавным будет аналоговый сигнал, что имеет значение в приложениях, требующих высокой точности.

Концепция R2R ЦАПа интуитивно понятна и является базовой для рассмотрения принципа цифро-аналогового преобразования сигналов. Тем не менее, существует множество других видов ЦАП, например, на основе **ΔΣ** модуляции, **последовательного** **приближения**, на базе **ШИМ** и др. Они, как правило, подключаются к МК или SoC по цифровым интерфейсам.

Популярными ИМС ЦАП являются, например, **MCP4725** от Microchip (12 бит, I2C интерфейс), **AD5683R** от Analog Devices (16 бит, SPI интерфейс), **DAC8551** от Texas Instruments (16 бит, I2C интерфейс).

На рис. 9 показан вариант R2R ЦАП, а на рис. 10 – временная диаграмма для 3 бит. ЦАП.

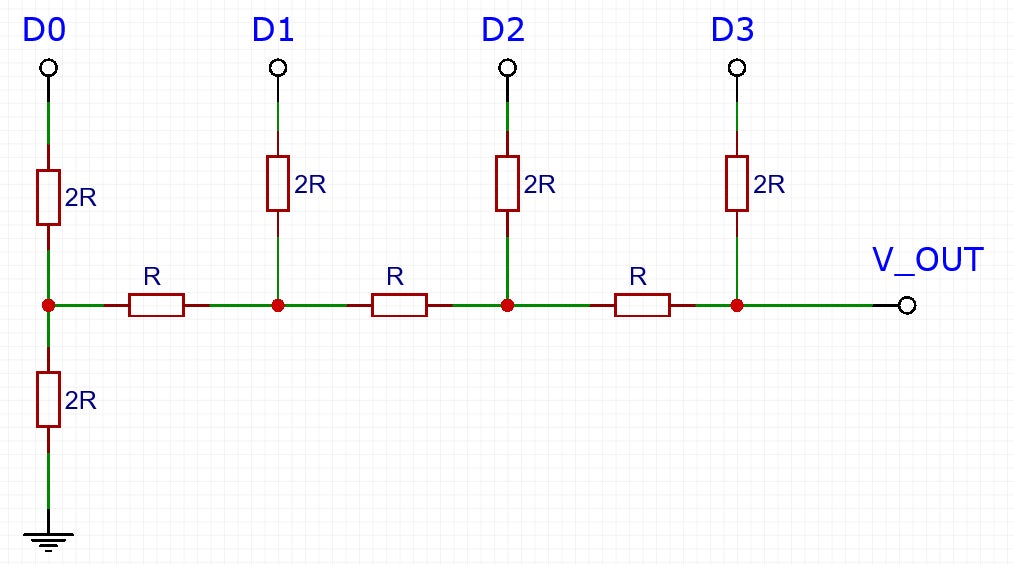


Рисунок 9 – R2R ЦАП разрядностью 4 бит

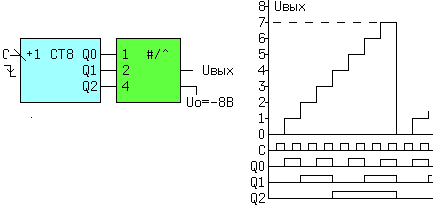


Рисунок 10 – Диаграмма состояний для 3 бит. ЦАП

6 АЦП

**Аналого-цифровой преобразователь** (АЦП, ADC) последовательного приближения — это устройство, которое **преобразует аналоговый сигнал в цифровой**, используя метод последовательного приближения. Этот тип АЦП обеспечивает высокую точность и скорость преобразования. Вариант УГО для прибора представлен на рис. 11.

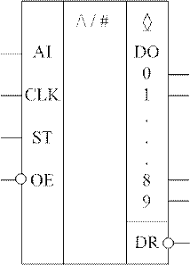


Рисунок 11 – Вариант УГО АЦП

Структурная схема АЦП последовательного приближения (рис. 12) включает в себя несколько ключевых компонентов: компаратор, регистр, **цифровой аналоговый преобразователь** и **логический контроллер**. Компаратор сравнивает входной аналоговый сигнал с выходом ЦАП, который генерирует опорное напряжение на основе текущего значения регистра. Логический контроллер управляет процессом, определяя, нужно ли увеличить или уменьшить значение, чтобы достичь совпадения.

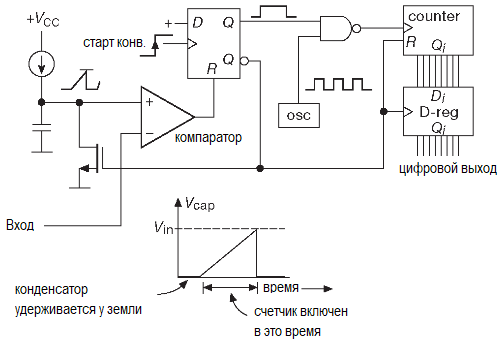


Рисунок 12 – Внутренняя структура АЦП последовательного приближения

**Преимущества** АЦП последовательного приближения включают в себя простоту конструкции и относительно низкую стоимость. Этот тип АЦП широко используется в различных приложениях, включая системы измерения, управления и обработки сигналов, где требуется высокая точность и скорость преобразования.

Существуют и другие типы АЦП, такие как **интегрирующие**, **ΔΣ**, **параллельные** и др. Они, также как и ЦАП, чаще всего к цифровым ИМС подключаются по интерфейсам.

Популярными ИМС АЦП являются, например, **MCP3008** от Microchip   
(10 бит, SPI интерфейс), **AD7606** от Analog Devices (16 бит, SPI интерфейс), **ADS1115** от Texas Instruments (16 бит, I2C интерфейс).

7 Гибкая и жесткая логика

**Гибкая и жесткая логика** — это две концепции, используемые в проектировании цифровых систем.

Жесткая логика подразумевает **использование фиксированных логических схем**, которые не могут быть изменены после проектирования. Это обеспечивает высокую надежность и предсказуемость работы системы, но ограничивает ее адаптивность к изменяющимся условиям или требованиям.

Гибкая логика, напротив, позволяет **изменять функциональность системы** путем программирования или перенастройки. Это может быть достигнуто с помощью программируемых логических устройств (PLD), таких как **FPGA** (Field Programmable Gate Array).

Кроме того, в контексте гибкой логики также следует упомянуть микроконтроллеры. Микроконтроллеры, как правило, основаны на **последовательной** обработке данных и **программируемы** для выполнения определенных задач. Они обеспечивают большую гибкость в разработке программного обеспечения, позволяя адаптировать функциональность системы к новым требованиям, но их архитектура обычно **не поддерживает параллельную обработку** на уровне, аналогичном FPGA.

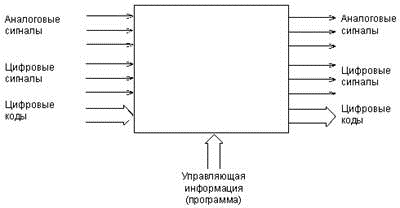


Рисунок 13 – концептуальный вид цифровой системы

Таким образом, гибкая логика, представленная ПЛИС, и микроконтроллеры предлагают разные подходы к проектированию цифровых систем: ПЛИС обеспечивают высокую производительность и параллелизм, в то время как микроконтроллеры предлагают простоту программирования и последовательную обработку данных.

8 Определение и концепция микроконтроллера

**Микроконтроллер** — это компактное интегральное устройство, которое включает в себя процессор, память и периферийные интерфейсы на одном чипе. Они предназначены для выполнения конкретных задач встраиваемых систем и широко используются в различных приложениях, от бытовой электроники до промышленных автоматизированных систем. Микроконтроллеры обеспечивают высокую степень интеграции, что позволяет уменьшить размеры и стоимость конечных устройств.

Основная концепция микроконтроллеров заключается в том, что они могут обрабатывать данные, взаимодействовать с внешними устройствами и выполнять программы, хранящиеся в их памяти. Это делает их идеальными для управления процессами и автоматизации. Микроконтроллеры могут быть настроены для выполнения различных функций, таких как управление двигателями, обработка сигналов и связь с другими устройствами.

Современные **микроконтроллеры часто имеют встроенные дополнения**, такие как аналогово-цифровые преобразователи (АЦП), таймеры и интерфейсы связи, что делает их универсальными инструментами для разработки. Они позволяют инженерам создавать сложные системы с минимальными затратами на аппаратное обеспечение и время разработки.

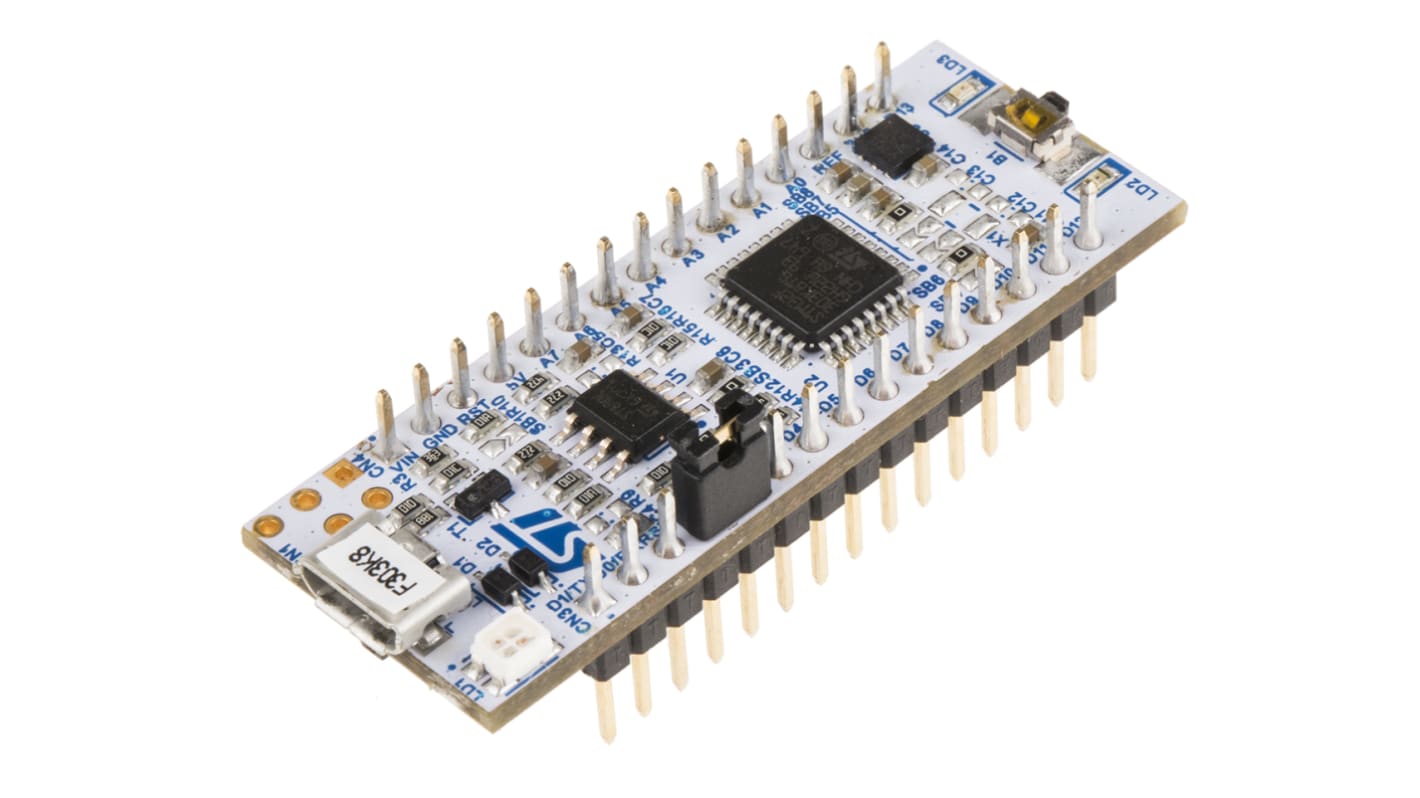


Рисунок 14 – Отладка STM32 Nucleo-32

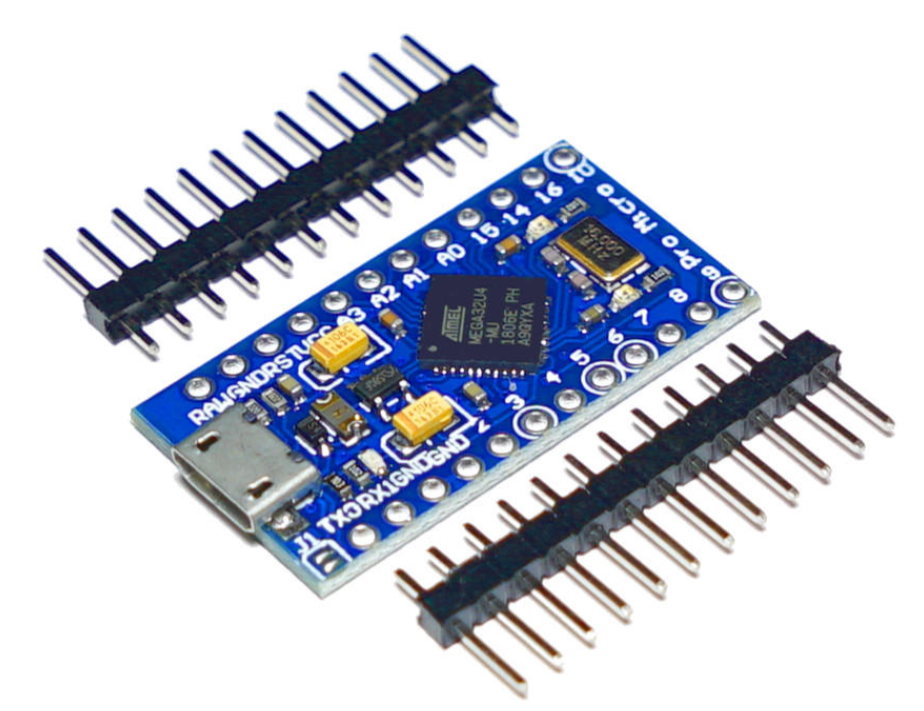


Рисунок 15 – Отладка Arduino Nano

В инженерной и образовательной среде достаточно популярны решения **отладочных плат** – плат, предлагающих МК с обвязкой, благодаря чему можно взаимодействовать непосредственно с микроконтроллером, не принимая во внимание схемотехнику. Особенно часто отладочные платы используются в прототипировании. Варианты таких плат показаны на рис. 14 и 15.

9 Микроконтроллер как «серый ящик»

Микроконтроллер можно рассматривать как «серый ящик», в котором скрыты сложные процессы и функции, доступные пользователю через интерфейсы. Основные блоки и модули, входящие в состав микроконтроллера, включают **центральный процессор (ЦП), память (как оперативную - ОЗУ, так и постоянную - ПЗУ)**, а также различные **периферийные устройства.**

Центральный процессор отвечает за выполнение инструкций программы, а память используется для хранения данных и кода. **Оперативная память** (RAM) временно хранит данные, используемые во время выполнения программы, в то время как **постоянная память** (ROM или Flash) хранит саму программу и настройки устройства.

Периферийные устройства могут включать в себя **таймеры**, **аналого-цифровые преобразователи** (АЦП), цифро-аналоговые преобразователи (ЦАП), **порты ввода-вывода**, а также **интерфейсы связи**, такие как UART, SPI и I2C. Эти модули позволяют микроконтроллеру взаимодействовать с внешними устройствами и выполнять широкий спектр задач.

Структура МК на примере линейки ESP32 от Espressif system как серого ящика показана на рис. 16.

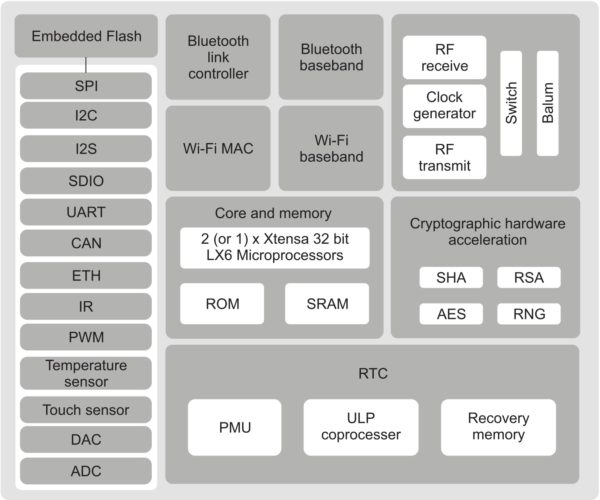


Рисунок 16 – Концептуальный вид структуры микроконтроллера ESP32

10 Типы входов и выходов микроконтроллера

Микроконтроллеры могут иметь различные **типы входов и выходов**, которые позволяют им взаимодействовать с окружающим миром. Цифровые входы и выходы (I/O) представляют собой базовые элементы, которые могут принимать или выдавать **только два состояния**: **высокий** (1) и **низкий** (0). Эти сигналы могут использоваться для управления светодиодами, реле и другими цифровыми устройствами.

Аналоговые входы и выходы, напротив, позволяют работать с **непрерывными сигналами**, что дает возможность микроконтроллерам обрабатывать данные с датчиков, например, микрофонов, термодатчиков и т.п.. Аналоговые входы обычно реализуются **через АЦП**, которые преобразуют аналоговые сигналы в цифровые, а аналоговые выходы могут быть реализованы **через ЦАП**.

Комбинирование цифровых и аналоговых входов и выходов **позволяет микроконтроллерам выполнять широкий спектр задач**, от простой обработки сигналов до сложного управления системами. Это делает их универсальными инструментами в разработке встраиваемых систем.

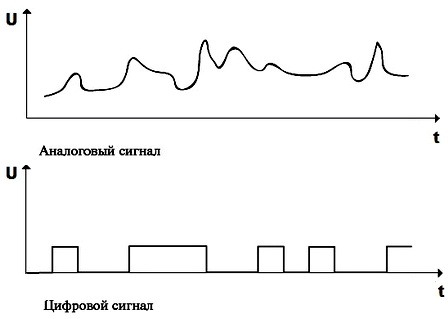


Рисунок 17 – сравнение цифрового и аналогового сигналов

Не все МК обладают интегрированными ЦАП. АЦП же встречается практически на всех МК, но также может отсутствовать. Это зависит от уровня выбранной модели и ее ценовой категории.

На рис. 18 приведены примеры работы цифрового входа МК **для опроса кнопки.** На рис. 19 показаны примеры работы цифрового выхода МК **для управления нагрузкой**. Обратите внимание, что порты МК не рассчитаны на ток больше **20-30 мА** (как правило, точное же значение всегда указывается в паспорте), а потому мощную нагрузку необходимо подключать через буферы или ключи на транзисторах.

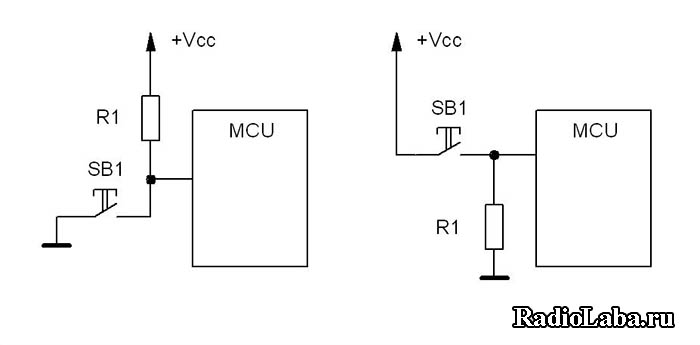


Рисунок 18 – Варианты подключения источника цифрового сигнала к МК

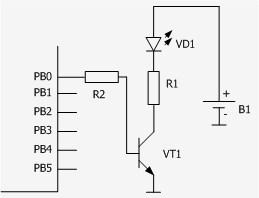
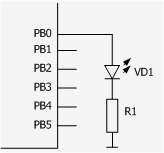


Рисунок 19 – Варианты подключения нагрузки к МК

На рис. 20 приведена демонстрация процесса обработки сигнала, поступающего с микрофона, проходящего аналого-цифровое преобразование, обработку на процессоре, а затем – обратное преобразование через ЦАП и выход на акустическую систему.

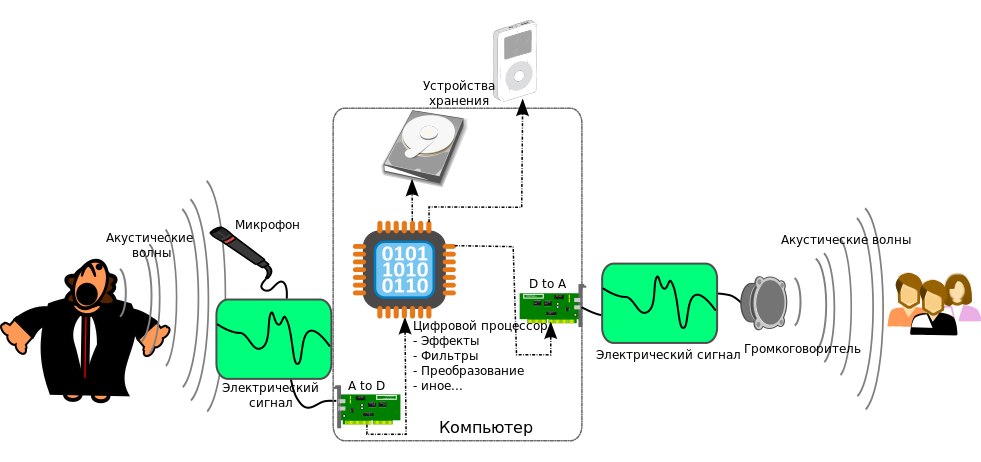


Рисунок 20 – Работа АЦ- и ЦА- преобразований

11 Интерфейсы передачи данных: I2C

**Интерфейс I2C** (Inter-Integrated Circuit) является популярным протоколом передачи данных, используемым для связи между микроконтроллерами и периферийными устройствами. Он **позволяет подключать несколько устройств к одной шине, используя всего два провода**: **SDA** (Serial Data Line) для передачи данных и **SCL** (Serial Clock Line) для синхронизации передачи. I2C поддерживает **многомастеровую** и **многоподчиненную** архитектуру, что позволяет одному или нескольким устройствам управлять шиной. Структура подключения по интерфейсу приведена на рис. 21.

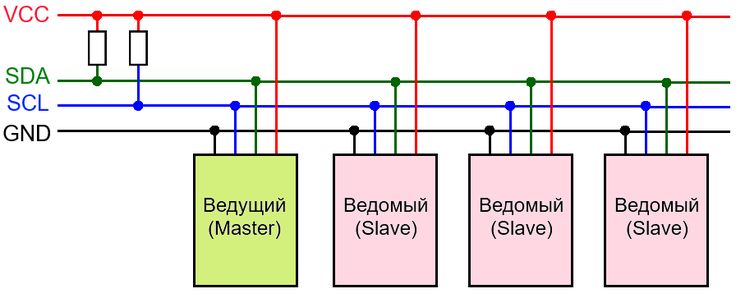


Рисунок 21 – Подключение устройств по I2С

Концепция I2C основана на **адресации** устройств, что позволяет каждому подключенному устройству иметь уникальный адрес. При передаче данных ведущий (мастер) отправляет адрес подчиненного устройства, с которым хочет установить связь, после чего начинается обмен данными. I2C поддерживает **различные скорости передачи**, что делает его универсальным для разных приложений.

Сопряжение устройств через I2C требует правильной конфигурации адресов и подключения линий SDA и SCL. Важно также учитывать **подтягивающие резисторы**, которые необходимы для обеспечения правильного уровня сигналов на шине. I2C широко используется в различных приложениях, включая датчики, EEPROM и дисплеи, благодаря своей простоте и многофункциональности.

На линии I2C могут быть **ведущие** и **ведомые** устройства. Максимальное количество устройств ограничивается длиной адреса и как правило равно 255. Устройства подключаются **параллельно**.



Рисунок 22 – временная диаграмма для интерфейса I2C

**Посылка от ведущего (master) к ведомому (slave) устройству** для I2C имеет вид как на рис. 22. Она включает несколько стадий:

1. **Начало передачи** (Start Condition).

2. **Адресация** ведомого устройства:

* После сигнала старта ведущее устройство отправляет 7- или 10-битный адрес ведомого устройства, к которому оно хочет обратиться.
* В конце адреса передается бит, указывающий, будет ли ведомое устройство принимать данные (бит 0) или отправлять данные (бит 1).

3. **Подтверждение** адреса (Acknowledge Bit):

* После отправки адреса ведущее устройство ожидает подтверждения (ACK) от ведомого устройства. Если ведомое устройство распознало свой адрес, оно отвечает, отправив бит подтверждения, изменяя уровень SDA на низкий. Если ведомое устройство не распознало адрес, оно не отправляет ACK, и линия SDA остается высокой.

4. **Передача** данных (если подтверждение было получено).

5. **Завершение** передачи (Stop Condition).

12 Интерфейсы передачи данных: SPI

**SPI** (Serial Peripheral Interface) — это еще один популярный интерфейс передачи данных, который используется для связи между микроконтроллерами и периферийными устройствами. В отличие от I2C, **SPI использует несколько линий для передачи данных, включая MOSI** (Master Out Slave In), **MISO** (Master In Slave Out), **SCK** (Serial Clock) и **CS** (Chip Select). Это позволяет обеспечить более **высокую скорость** передачи данных, что делает SPI идеальным для приложений, требующих быстрого обмена информацией.

Концепция SPI основана на мастер-подчиненной архитектуре, где один мастер может управлять несколькими подчиненными устройствами. При передаче данных мастер генерирует тактовый сигнал на линии SCK, а данные передаются по линиям MOSI и MISO. Каждое подчиненное устройство активируется с помощью сигнала CS, что позволяет избежать конфликтов на шине.

Пример реализации SPI приведен на рис. 23.

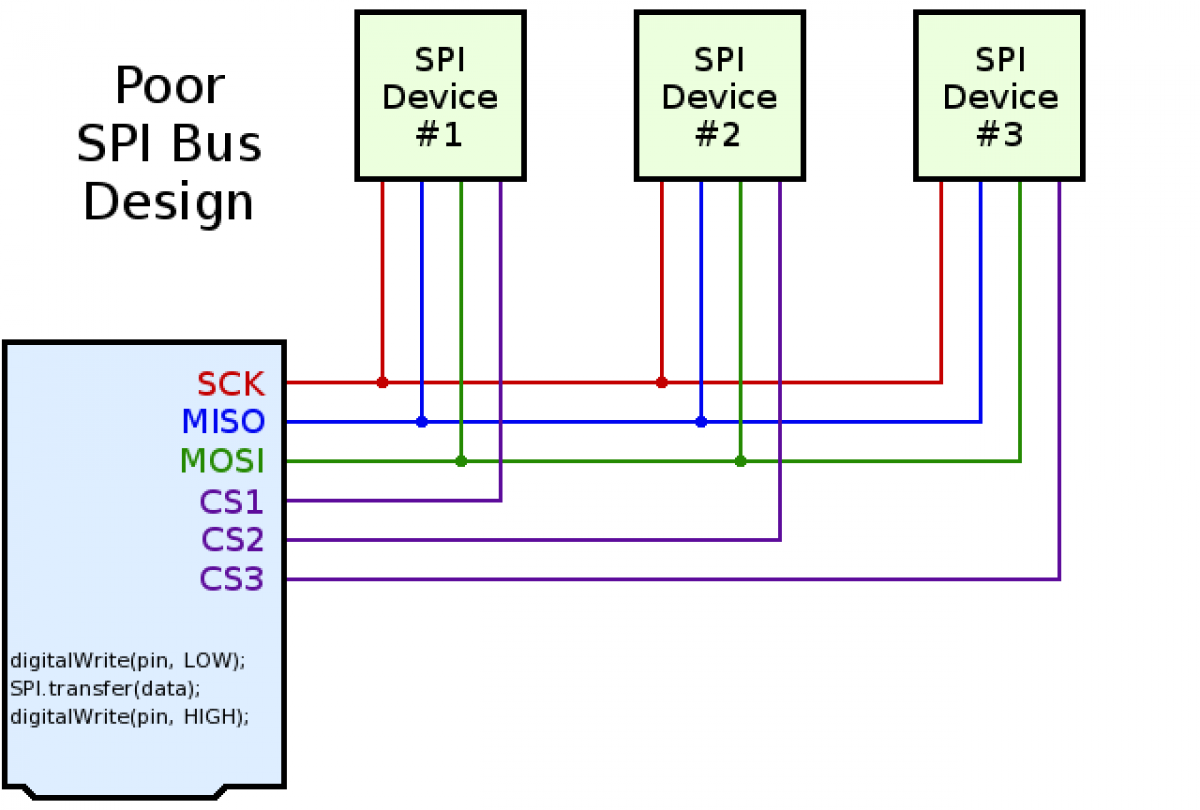


Рисунок 23 – Подключение устройств по SPI

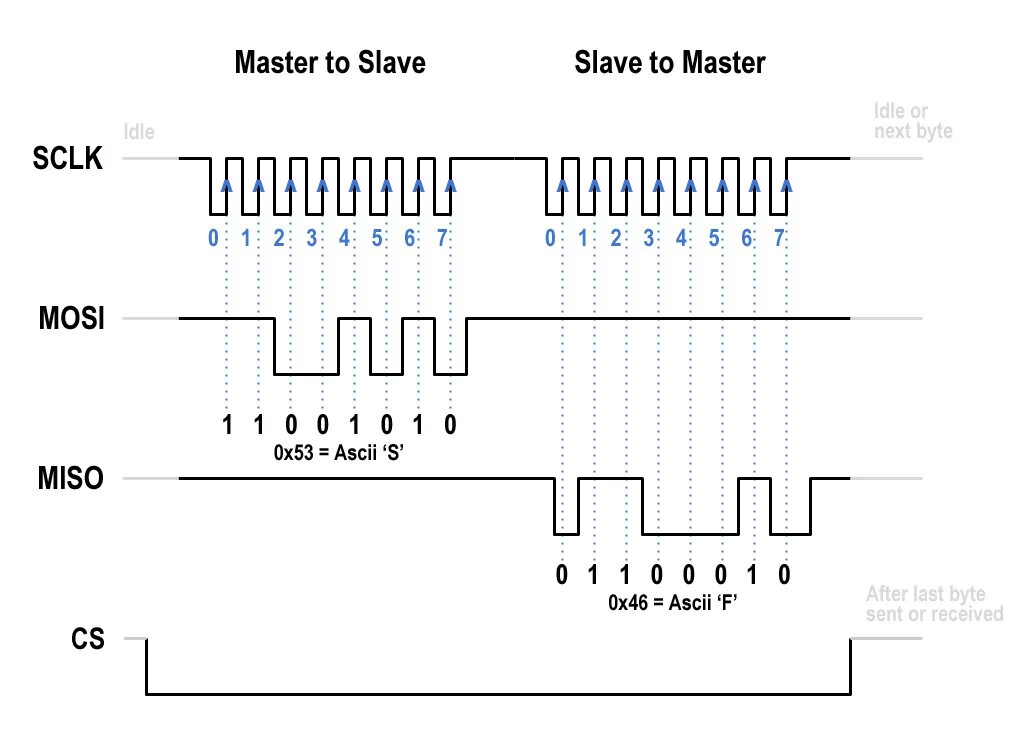


Рисунок 24 – Временная диаграмма для SPI

Временная диаграмма для SPI, приведенная на рисунке 24, описывается следующим образом:

1. **Выбор ведомого устройства** (линия СS):

* Ведущее устройство устанавливает сигнал СS в низкое состояние, чтобы выбрать ведомое устройство.

2. **Генерация** тактового сигнала (линия SCK):

* Ведущее устройство начинает генерировать тактовый сигнал SCK. В зависимости от конфигурации (полярности и фазы), данные могут быть захвачены на восходящем или нисходящем фронте SCK.

3. **Передача** данных (линии MOSI и MISO):

* Ведущее устройство отправляет данные по линии MOSI, а ведомое устройство может отправлять данные по линии MISO в ответ. Передача данных происходит синхронно с тактовым сигналом SCK.

4. **Завершение** передачи:

* После завершения передачи данных ведущий отключает сигнал СS (устанавливает его в высокое состояние), что завершает общение с ведомым устройством.

13 Интерфейсы передачи данных: UART

**UART** (Universal Asynchronous Receiver-Transmitter) — это асинхронный интерфейс передачи данных, который широко используется для последовательной связи между микроконтроллерами и другими устройствами. UART работает по принципу передачи данных в виде последовательности битов, где каждый байт данных включает стартовый бит, восемь бит данных, бит четности (опционально) и стоповый бит. Концепция UART заключается в том, что передатчик и приемник должны быть настроены на **одинаковую скорость передачи** (битрейт), количество бит данных и количество стоповых битов. Это позволяет обеспечить корректный обмен данными между устройствами. UART может работать в полудуплексном и полнодуплексном режимах, что означает, что данные могут передаваться в обоих направлениях, но не одновременно (**полудуплексный**) или одновременно (**полнодуплексный**).

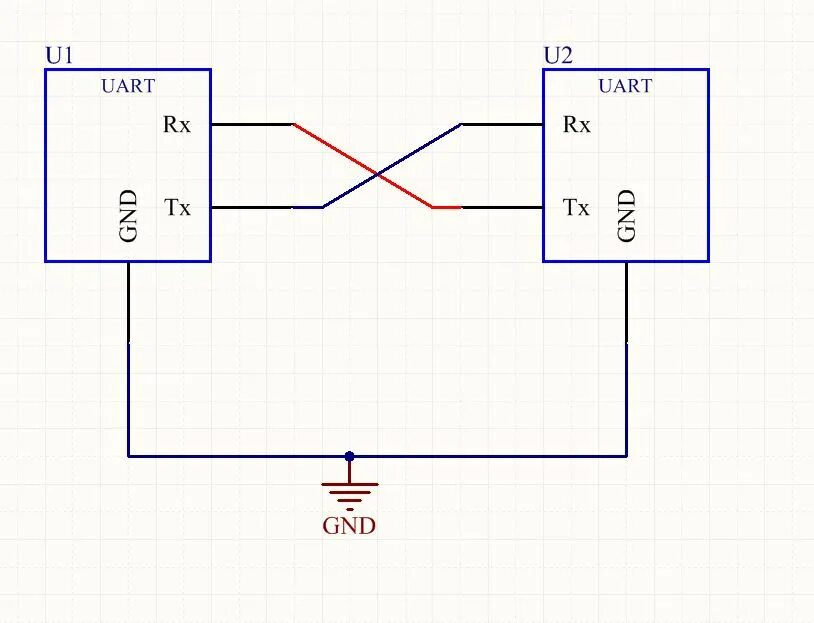


Рисунок 25 – Подключение устройств по UART

Сопряжение устройств через UART требует подключения линий **TX** (передача) и **RX** (прием) между устройствами согласно рисунку 25. Также важно учитывать уровни логических сигналов, чтобы избежать повреждения оборудования. UART широко используется в различных приложениях, включая модемы, GPS-устройства и системы управления, благодаря своей простоте и надежности.

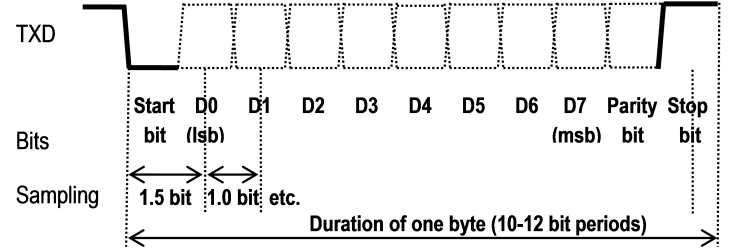


Рисунок 26 – Временная диаграмма для UART

Описание временной диаграммы для UART имеет вид, показанный на рисунке 26, и проходит следующим образом:

1. **Стартовый** бит:

* Передача начинается со стартового бита, который устанавливается в низкий уровень (0). Это сигнализирует о начале передачи.

1. **Данные**:

* После стартового бита передаются биты данных. Обычно это 8 бит, которые могут принимать значения от 0 до 255. Биты данных передаются в порядке от младшего к старшему (LSB к MSB).

1. **Контрольный** **бит** (если используется):

* Если используется контрольный бит (например, бит четности), он передается сразу после данных. Он может быть установлен для проверки четности переданных данных.

1. **Стоп**-**биты**:

* После передачи данных (и контрольного бита, если он есть) передаются один или два стоп-бита, которые устанавливаются в высокий уровень (1). Это сигнализирует о завершении передачи.

14 Заключение и перспективы

Цифровые интегральные схемы (ИМС) средней и большой интеграции стали **основой современного электронного мира**, обеспечивая значительное увеличение функциональности при уменьшении размеров устройств. Благодаря возможности размещения тысяч и миллионов транзисторов на одном кристалле, эти схемы значительно улучшают производительность и снижают энергопотребление. В последние годы наблюдается рост интереса к проектированию цифровых ИМС, что связано с развитием новых технологий, таких как искусственный интеллект, Интернет вещей (IoT) и 5G, которые требуют высокой степени интеграции для обеспечения компактности и эффективности.

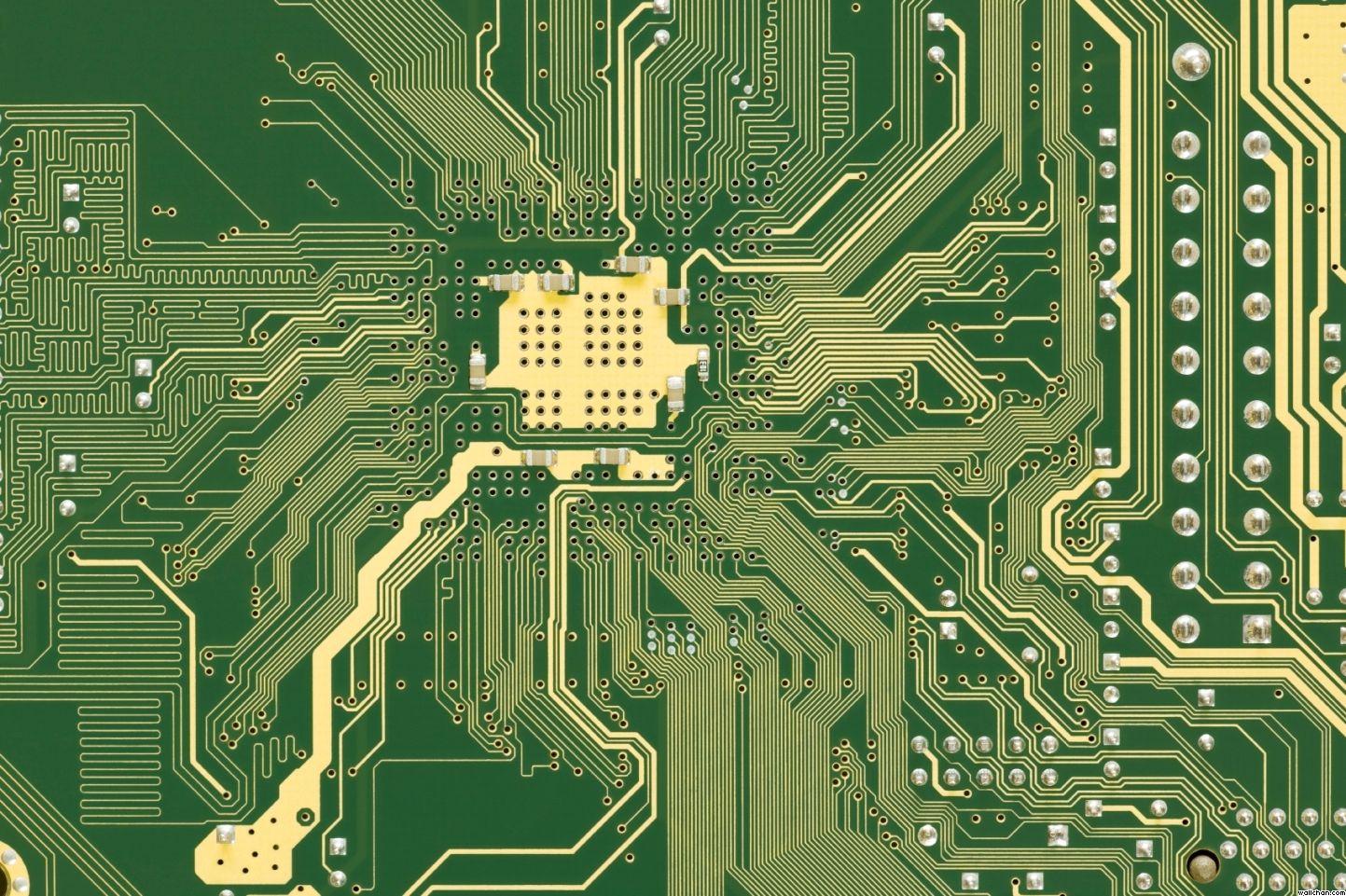


Рисунок 27 – Топология нижнего слоя материнской печатной платы

Тем не менее, индустрия цифровых ИМС сталкивается с рядом вызовов, включая усложнение проектирования и высокие затраты на разработку. Для решения этих проблем активно разрабатываются новые методы проектирования, такие как системный подход к проектированию и использование программируемых логических устройств. Эти инновации не только улучшают характеристики интегральных схем, но и открывают новые возможности для их применения в различных областях, от мобильных устройств до высокопроизводительных вычислительных систем, что делает будущее цифровых ИМС многообещающим.

Спасибо за внимание!