Московский Государственный Технический Университет имени Н.Э.Баумана

Конспект лекций по курсу

«Микроконтроллерные управляющие устройства»

Москва, 2009

Оглавление

[1. Микроконтроллеры. Назначение. Общие свойства архитектуры. Особенности микроконтроллеров различных фирм производителей. Управляющие восьмиразрядные микроконтроллеры семейства MCS-51 фирмы Intel и совместимые с ними. 2](#_Toc242211606)

[2. Обобщенная архитектура микроконтроллеров. Типовые периферийные устройства. 12](#_Toc242211607)

[3. Средства программирования и отладки микроконтроллеров. Инструментальные и программные средства. Внутрисхемный эмулятор, принцип работы. Классификация внутрисхемных эмуляторов. Дизассемблеры MCS-51. Макроассемблер. Компоновщик. Отладчик/симулятор. Подсистема точек останова. Процессор точек останова. Трассировщик. Профилировщик. Интегрированная среда разработки. 17](#_Toc242211608)

[4. Функциональные возможности внутрисхемных эмуляторов. Достоинства и недостатки внутрисхемных эмуляторов. Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051 PICE-51. Программные симуляторы. Платы развития. Отладочные мониторы. Эмуляторы ПЗУ. 23](#_Toc242211609)

[5. Структурная организация микроконтроллера i8051. Общие характеристики. Арифметико-логическое устройство. Назначение выводов. Организация портов ввода вывода. Альтернативные функции. 28](#_Toc242211610)

[6. Особые режимы работы (загрузка, стирание, холостой ход, пониженное энергопотребление). Режим ХХ. Режим ВНП. 28](#_Toc242211611)

[7. Организация памяти. Работа с внешней памятью. Память программ (ПЗУ). Память данных (ОЗУ). Регистры специальных функций. Регистр флагов (PSW). 28](#_Toc242211612)

[8. Таймеры / счетчики микроконтроллеров семейства 8051. Регистр режима работы таймера/счетчика TMOD. Регистр управления/статуса таймера TCON. Режимы работы таймеров-счетчиков. 28](#_Toc242211613)

[9. Режимы работы и регистры последовательного интерфейса. Регистр управления/статуса приемопередатчика SCON. Функциональное назначение бит регистра управления/статуса приемопередатчика SCON. Скорость приема/передачи информации через последовательный порт. Работа в сети. 28](#_Toc242211614)

[10. Организация системы прерываний. Регистры масок и приоритетов прерываний, флаги прерываний. Регистр масок прерывания (IE). Регистр приоритетов прерываний (IP). Выполнение подпрограммы прерывания. 28](#_Toc242211615)

[11. Система команд микроконтроллера i8051. Форматы представления данных и команд, группы команд, способы адресации операндов. Обозначения, используемые при описании команд. 28](#_Toc242211616)

[12. Классификация системы команд по функциональному признаку (пересылки данных, арифметических и логических операций, операций над битами, передачи управления), временные параметры исполнения команд. 28](#_Toc242211617)

[13. Организация взаимодействия микроконтроллера с объектом управления. Опрос двоичного датчика. Ожидание события. Устранение дребезга контактов. Подсчет числа импульсов. Опрос группы двоичных датчиков. 28](#_Toc242211618)

[14. Таймер счетчик Т/С2 микроконтроллера 8052. Регистр управление таймера/счетчика 2 T2COM. Режимы работы таймера/счетчика 2. Регистр режима таймера/счетчика 2 Т2МОD. Дополнительный регистр приоритетов прерываний IРН. 29](#_Toc242211619)

[15. Аналого-цифровой преобразователь микроконтроллеров семейства MCS-51. ADCON - Регистр управления преобразователем. ADDAT - регистр результатов преобразования. DAPR - регистр программирования опорных напряжений АЦП. Синхронизация АЦП и время преобразования. 29](#_Toc242211620)

[16. Расширения микропроцессоров семейства MCS-51/52. Дополнительные устройства и регистры. 29](#_Toc242211621)

# 1. Микроконтроллеры. Назначение. Общие свойства архитектуры. Особенности микроконтроллеров различных фирм производителей. Управляющие восьмиразрядные микроконтроллеры семейства MCS-51 фирмы Intel и совместимые с ними.

**Микропроцессор**

*Микропроцессор*(МП) — это программно-управляемое электронное цифровое устройство, предназначенное для обработки цифровой информации — ее перемещения, осуществления арифметических и логических операций по командам, которые он считывает из памяти.

Последовательность команд называется *программой*.

В 1970 году М.Э. Хофф из фирмы Intel сконструировал интегральную схему, аналогичную по своим функциям центральному процессору большой ЭВМ — первый микропроцессорIntel-4004, который уже в 1971 году был выпущен в продажу.

Микропроцессор включает в себя:

* *арифметическо-логическое устройство* (АЛУ), которое служит для осуществления собственно арифметических и логических операций: арифметической операцией называют процедуру обработки данных, аргументы и результат которой являются числами (сложение, вычитание, умножение, деление и т.д.). Логической операцией именуют процедуру, осуществляющую построение сложного высказывания (операции И, ИЛИ, НЕ и т.д.).
* *регистры общего назначения* (РОН), которые используются для хранения информации — сверхоперативного запоминающего устройства. Регистры предназначены для хранения операндов в процессе выполнения операций и функциональных схем, необходимых для выполнения преобразования операндов при передаче их с одного регистра на другой. Количество и назначение РОН в МП зависят от его архитектуры.
* *аккумулятор* — регистр, из которого берется одно из чисел, с которыми производятся арифметические или логические операции. В него помещается результат;
* *счетчик адреса команд*, в котором хранится адрес ячейки памяти, в которой записан код текущей команды;
* *регистр флагов или условий* — в него помещаются сведения об особенностях результата выполнения арифметических или логических операций, например, нулевой результат, переполнение (перенос), четность и пр.;
* *регистр адреса стека*, в котором записан адрес последний занятой под стек ячейки памяти;
* *блок управления шинами микропроцессорной системы*, схемы формирующей сигналы на внешних шинах микропроцессора и, тем самым, управляющей микропроцессорной системой;
* *блок дешифрирования кодов команд*.
* *Таймер-счетчик* — предназначен для подсчета внутренних событий, для получения программно-управляемых временных задержек и для выполнения времязадающих функций МП.
* *ОЗУ* — служит для приема, хранения и выдачи информации, используемой в процессе выполнения программы.
* *ПЗУ* — служит для выдачи констант, необходимых при обработке данных в АЛУ.
* *КЭШ-память* — хранит внутри МП копии тех команд операндов и данных, к которым производились последние обращения МП. Если МП необходимо считать данные, имеющиеся в КЭШ, то она их представляет, и нет необходимости обращаться к внешней памяти. В КЭШ помещаются результаты вычислений.
* *ША, ШД, ШУ* (адреса, данных, управления) — группы линий, по которым передается однотипная информация.
* *Шинный интерфейс* — выполняет функции согласования действий между внутренними устройствами МП и внешней системой, т.е. управляет потоками и форматами данных между МП и внешними устройствами.

В тех случаях, когда память и средства ввода/вывода размещаются на той же подложке интегральной схемы, что и микропроцессор, последний превращается в микроконтроллер. Более подробный анализ позволяет определить микроконтроллеры как устройства, имеющие память RAM

или ROM вместо кэш-памяти, присутствующей обычно в большинстве периферийных устройств. В противоположность микроконтроллерам, микропроцессоры имеют устройство управления памятью и большой объем кэш-памяти. Иногда разница определяется производительностью или разрядностью.

**Основные характеристики микропроцессора**

Микропроцессор характеризуется:

1. *тактовой частотой*, определяющей максимальное время выполнения переключения элементов;
2. *разрядностью*, т.е. максимальным числом одновременно обрабатываемых двоичных разрядов.
3. *архитектурой*

Разрядность МП обозначается m:n:k и включает:

* m — разрядность внутренних регистров, определяет принадлежность к тому или иному классу процессоров;
* n — разрядность шины данных, определяет скорость передачи информации;
* k — разрядность шины адреса, определяет размер адресного пространства. Например, МП i8088 характеризуется значениями m:n:k=16:8:20;

Архитектура МП дает представление о функциональном поведении логической структуры и ее организации (взаимодействие отдельных узлов и блоков МП при выполнении тех или иных вычислительных операций), определяет особенности построения программных средств, описывает внутреннюю организацию потоков данных и управляющей информации.

Понятие архитектуры микропроцессора включает в себя систему команд и способы адресации, возможность совмещения выполнения команд во времени, наличие дополнительных устройств в составе микропроцессора, принципы и режимы его работы.

В зависимости от набора и порядка выполнения команд процессоры исторически сформировались несколько классов.

**Архитектуры микропроцессора — RISC и CISC.**

**Основные черты CISC-концепции:**

Ранее других появились процессоры CISC. Термин **CISC** означает сложную систему команд и является аббревиатурой английского определения Complex Instruction Set Computer. Благодаря этому процессоры выполняют самые разнообразные задачи обработки данных.

При разработке набора команд CISC заботились об удобстве программиста/компилятора, а не об эффективности исполнения команд процессором. В систему команд вводили много сложных команд (производящих по несколько простых действий). Часто эти команды представляли собой программы, написанные на микрокоде и записанные в ПЗУ процессора. Команды CISC имеют разную длину и время выполнения. Зато машинный код CISC-процессоров — язык довольно высокого уровня. В наборе команд CISC часто присутствуют, например, команды организации циклов, команды вызова подпрограммы и возврата из подпрограммы, сложная адресация, позволяющая реализовать одной командой доступ к сложным структурам данных. Основной недостаток CISC — большая сложность реализации процессора при малой производительности.

Примеры CISC-процессоров — семейство Motorola 680x0 и процессоры фирмы Intel от 8086 до Pentium II. Наиболее известные микроконтроллеры с CISC-архитектурой фирм Zilog, Intel, Motololla, Siemens.

**Основные черты RISC-концепции:**

Со временем стало необходимо повысить скорость работы процессоров. Одним из путей к этому стал процессор RISC, который характеризуется сокращенным набором быстро выполняемых команд и происходит от английского Reduced Instruction Set Computer.

* одинаковая длина команд;
* одинаковый формат команд — код команды регистр-приемник два регистра-источника;
* операндами команд могут быть только регистры;
* команды выполняют только простые действия;
* большое количество регистров общего назначения (могут быть использованы любой командой);
* конвейер(ы);
* выполнение команды не дольше, чем за один такт;
* простая адресация.

К RISC процессорам причисляют MIPS, SPARC, PowerPC, DEC Alpha, HP PA-RISC, Intel 960, AMD 29000. RISC-концепция предоставляет компилятору большие возможности по оптимизации кода. Существуют стандарты на RISC-процессоры, например SPARC — Scalable Processor ARChitecture, MIPS, PowerPC часто их называют открытыми архитектурами. Наиболее известные микроконтроллеры с RISC-архитектурой это семейства AT90S, ATMega, ARM фирмы Atmel, микроконтроллеры фирм PIC, Scenix, Holtek.

**Сравнение архитектур**

Основная идея **RISC**-архитектуры — это тщательный подбор таких комбинаций кодов операций, которые можно было бы выполнить за один такт тактового генератора. Основной выигрыш от такого подхода — резкое упрощение аппаратной реализации ЦП и возможность значительно повысить его производительность.

Однако обычно выигрыш от повышения быстродействия в рамках **RISC**-архитектуры перекрывает потери от менее эффективной системы команд, что приводит к более высокой эффективности **RISC**-систем в целом по сравнению с **CISC**. Так, в процессоре CISC для выполнения одной ко-

манды необходимо, в большинстве случаев, 10 и более тактов. Что же касается процессоров RISC, то они близки к тому, чтобы выполнять по одной команде в каждом такте. Также с упрощением ЦП уменьшается число транзисторов, необходимых для его реализации, следовательно, уменьшается площадь кристалла. А с этим связано снижение стоимости и потребляемой мощности.

Следует также иметь в виду, что благодаря своей простоте процессоры RISC не патентуются. Это также способствует их быстрой разработке и широкому производству. Между тем, в сокращенный набор RISC вошли только наиболее часто используемые команды. Ряд редко встречающихся команд процессора CISC выполняется последовательностями команд процессора RISC. Позже появилась концепция процессоров MISC, использующая минимальный набор длинных команд. Вслед за ними возникли процессоры VLIW, работающие со сверхдлинными командами. Быстродействие процессоров определяется в миллионах операций в секунду MIPS.

**Память в микропроцессорных устройствах**

В микропроцессорных устройствах память служит для хранения исходных данных программ обработки информации промежуточных и окончательных результатов вычисления.

Выделяют два основных типа памяти:

* *ОЗУ* — оперативное запоминающее устройство, используемое для хранения данных, поэтому эту память называют еще памятью данных. Число циклов чтения и записи в ОЗУ не ограничено, но при отключении питающего напряжения вся информация теряется; В современных микропроцессорах память ОЗУ представляет собой многоуровневую систему, в которой выделяют уровни сверхоперативной памяти (СОЗУ), ОЗУ, буферной памяти (БЗУ) и внешней памяти (ВЗУ). Каждый последующий уровень отличается от предыдущего емкостью и быстродействием.
* *Емкостью* называется максимальное количество информации, которая может быть записана в память.
* *Быстродействие* характеризуется длительностью операций чтения и записи — двух основных операций, выполняемых памятью. Для указанных уровней памяти емкость растет в направлении от СОЗУ к ВЗУ, а быстродействие в противоположном направлении.
* *ПЗУ* — постоянное запоминающее устройство, предназначенное для хранения программ, поэтому часто эту память называют кодовой или памятью программ. Микросхемы ПЗУ способны сохранять информацию при отключенном электропитании, но могут быть запрограммированы только один или очень ограниченное число раз.

**ATMEL**

Микроконтроллеры изготавливаются по КМОП (CMOS) технологии и имеют полностью статическую структуру. Базовая структура микроконтроллеров совпадает с базовой структурой микроконтроллеров семейства MCS-51 и отечественных микроконтроллеров серий 1816/51 и 1830/51, однако микроконтроллеры многих типов содержат новые запоминающие и периферийные устройства, а некоторые устройства базовой структуры имеют иные характеристики. В число «старых» устройств входят:

* запоминающие: внутреннее постоянное запоминающее устройство (Internal ROM, IROM), предназначенное для хранения команд программы и констант, и внутреннее оперативное запоминающее устройство (Internal RАM, IRАM), предназначенное для хранения данных. IROM представляет собой репрограммируемое запоминающее устройство с электрическим стиранием записи, выполненное по Flash технологии. IROM выдерживает до 1000 циклов перепрограммирования. IRAM является статическим оперативным запоминающим устройством.
* периферийные устройства: восьмиразрядные параллельные порты ввода-вывода Р0, Р1, Р2, Р3, последовательный порт SP, таймеры-счетчики Т/С0, Т/С1, Т/С2 и контроллер прерываний.

Микроконтроллеры некоторых типов имеют «новое» запоминающее устройство - внутреннее репрограммируемое запоминающее устройство для хранения данных (Internal Data ROM, IDROM). Первоначальная запись данных в IDROM производится при программировании микроконтроллера. В процессе выполнения программы обращения к IDROM для чтения и записи выполняются с использованием команд с мнемокодами операции MOVX. После обращения для записи в IDROM выполняется цикл записи длительностью несколько мс, в течение которого новое обращение к IDROM невозможно. «Новыми» периферийными устройствами являются блок последовательного периферийного интерфейса (SPI), сторожевой таймер (WDT) и аналоговый компаратор (АС).

Блок SPI предназначен для последовательного ввода и вывода данных с использованием трех шин. При этом микроконтроллер может работать в качестве ведущего или ведомого устройства, а микроконтроллер типа АТ89S4D12 - только в качестве ведомого. Блок SPI может быть использован также для программирования микроконтроллера после установки его в аппаратуру. Сторожевой таймер WDT предназначен для перезапуска программы при появлении сбоев в ходе ее выполнения. Программа, работающая без сбоев, периодически сбрасывает сторожевой таймер, не допуская его переполнения.

Аналоговый компаратор сравнивает по величине напряжения сигналы, поступающие на входы Р1.0 и Р1.1. Результат сравнения подается на вход Р3.6, не имеющий внешнего вывода.

Микроконтроллеры семейства AT89 выпускаются для работы при разных значениях напряжения питания и тактовой частоты, определяемой частотой подключенного к микроконтроллеру кварцевого резонатора. Ток потребления зависит от величины напряжения питания и тактовой частоты. Кроме рабочего режима микроконтроллер может быть переведен в энергосберегающие режимы работы - режим холостого хода (Idle Mode) и режим пониженного энергопотребления (Power Down Mode).

В режиме холостого хода процессор остановлен, периферийные устройства продолжают работать, коды в IRAM сохраняются. Ток потребления уменьшается в 4-5 раз. Перевод в режим холостого хода выполняется по команде в программе, выход из режима - по сигналу сброса или при поступлении любого разрешенного запроса прерывания.

В режиме пониженного энергопотребления остановлен генератор тактового сигнала, коды в IRAM сохраняются. Ток потребления имеет величину 20-100 мкА. Перевод в режим пониженного энергопотребления выполняется по команде в программе, выход из режима - по сигналу сброса.

Микроконтроллеры выпускаются в корпусах разных типов с разным числом выводов, при этом число выводов, используемых для подключения микроконтроллера к схеме устройства, может отличаться от числа выводов корпуса.

Все микроконтроллеры семейства Atmel программируются и перепрограммируются пользователем. Кроме того, фирма Atmel выпускают аналоги микроконтроллеров типов АТ89С51 и АТ89С52, которые программируются по заказу покупателя на предприятии-изготовителе (микроконтроллеры типов AT80F51 и AT80F52 соответственно) или допускают лишь однократное программирование пользователем (микроконтроллеры типов AT87F51 и AT87F52 соответственно).

Новая линия микроконтроллеров Atmel к настоящему времени насчитывает более 20 различных типов, которые объединены в четыре группы:

* Mega AVR (префикс ATmegaXXX);
* Classic AVR (префикс AT90SXXX);
* Tiny AVR (префикс ATtinyXXX)
* AVR для Smart Cards (префикс AT90SCC)

Mega AVR имеют наибольшие объемы памяти, наибольшее количество выводов и наиболее полный набор периферийных узлов.

Группа Classic AVR содержит микроконтроллеры с различным сочетанием периферийных узлов, различными объемами встроенной памяти и различным количеством выводов. Таким образом, разработчик имеет возможность сделать оптимальный выбор и не переплачивать за неиспользуемые узлы.

Микроконтроллеры группы Tiny AVR появились последними. Это дешевые кристаллы, стоимостью порядка 1 доллара США, в маленьких 8-выводных корпусах, способные работать от источника пониженного напряжения и при этом обладающие такими функционально важными периферийными узлами, как, например, АЦП.

Можно назвать следующие отличительные особенности Mega AVR по сравнению с Classic AVR:

* Рекордные размеры встроенной памяти программ и данных;
* 6 цифровых программируемых портов ввода/вывода;
* Практически полный набор периферийных узлов Classic AVR (нет только схемы сброса при понижении питания). Так, например, микроконтроллеры Mega AVR имеют одновременно и 8 аналоговых входов, и возможность подключения внешней SRAM, тогда как кристаллы подсемейства Classic AVR имели ту или другую возможность выборочно, в зависимости от типа;
* Простейший 8-разрядный таймер у Mega AVR имеет более развитые режимы работы, чем у Classic AVR;
* Имеется возможность управления системной тактовой частотой при помощи программируемого предделителя, что позволяет по ходу выполнения программы снижать тактовую частоту и уменьшать тем самым энергопотребление кристалла;
* При помощи внешнего сигнала, подаваемого на вход PEN, можно разрешать/запрещать программирование кристалла по последовательному интерфейсу.
* Важной особенностью Tiny AVR является эффективное использование выводов кристалла. Так, за счет того, что Tiny AVR могут быть запрограммированы на работу без внешнего кварца и без внешнего вывода "Сброс", можно иметь 6 (16) цифровых программируемых входов/выходов при 8 (20)-выводном корпусе.
* Также отличительной особенностью Tiny AVR является возможность понижения напряжения питания у некоторых типов до 1,8 В.

**INTEL**

Микроконтроллер семейства 8051 имеют следующие аппаратные особенности:

* внутреннее ОЗУ объемом 128 байт;
* четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;
* два 16-разрядных таймера-счетчика;
* встроенный тактовый генератор;
* адресация 64 КБайт памяти программ и 64 Кбайт памяти данных;
* две линии запросов на прерывание от внешних устройств;
* интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

Микроконтроллер 8751 снабжен УФ ПЗУ объемом 4 Кбайт. Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

Наряду с созданием сложных и высокоинтегрированных схем, Intel также совершенствует микросхемы, выпуск которых был освоен давно, например, 8-раз-рядные микроконтроллеры или однокристальные ЭВМ из семейства MCS-51 (отечественный аналог KM1816BE51). Эти микросхемы хорошо зарекомендовали себя в недорогих и сравнительно несложных устройствах. Основными направлениями модернизации данных микроконтроллеров являются:

* увеличение внутренней памяти программ до 32К, причем она может быть масочной, однократно программируемой или с ультрафиолетовым стиранием;
* снижение потребляемой мощности путем применения КМОП-технологии и специальных режимов пониженного энергопотребления;
* увеличение тактовой частоты до 20 Мгц;
* модификация режимов работы счетчиков-таймеров и последовательного порта;
* размещение на кристалле дополнительного оборудования.
* Эти микроконтроллеры выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:
* коммерческий (0°C — +70°C);
* расширенный (-40°C — +85°С):
* для военного использования (-55°C — +125°С).

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

**PHILIPS**

Микроконтроллеры изготавливаются по современному CMOS технологическому процессу. Все микроконтроллеры: оснащены памятью данных емкостью 256 байт, с возможностью внешнего расширения до 64 кбайт, объем встроенной памяти (RAM) составляет 8 кбайт, с возможностью внешнего расширения до 64 кбайт.

Микроконтроллеры располагают:

* пятью 8-разрядными портами I\O
* одним 8-разрядным входным портом, объединенным с аналоговыми входами.
* двумя 16-разрядными таймерами/счетчиками событий
* дополнительным 16-разрядным таймером/ счетчиком, связанным с четырьмя регистрами захвата и тремя регистрами сравенния
* 15 источниками прерываний с двумя уровнями приоритета
* 10-разрядным АЦП с восемью мультиплексирующими входами
* двумя PWM с 8-разрядным разрешением
* стандартным полнодуплексным UART , I|O порт последовательной I2C шины с байт-ориентированными функциями ведущего и ведомого
* встроенным сторожевым таймером
* встроенным генератором тактовых сигналов
* Напряжение питания: от 2,7 до 5,5 В
* Диапазон рабочих частот: от 3,5 до 30 МГц

Два программно устанавливаемых режима (IDLE и POWER-DOWN) способствуют энергосбережению. У модели 87С552 встроенный АЦП в режиме IDLE может быть оставлен в рабочем состоянии.

Микроконтроллеры могут работать как арифметические процессоры, поскольку обрабатывают и двоичную и BCD арифметику, обеспечивают манипулирование битами.

**ST MICROELECTRONICS**

Для всех микроконтроллеров ST, благодаря введению схем входной фильтрации и шумовой режекции, характерна высокая устойчивость к внешним электромагнитным помехам.

Статические микроконтроллеры ST6 предназначены для простых применений, связанных с контролем и обработкой аналоговых сигналов.

Процессорное ядро, построенное на гарвардской архитектуре, содержит:

* регистр- аккумулятор,
* два 8-разрядных регистра косвенной адресации
* два 8-разрядных регистра общего назначения
* аппаратный 6-уровневый стек

Объем памяти программ составляет от 1 до 8 кбайт, емкость ОЗУ – от 64 до 128 байт у микроконтроллера общего назначения и до 320 байт у микроконтроллера со встроенными контроллерами LCD-модулей. Некоторые модели имеют EEPROM данных объемом 64 и 128 байт.

Система команд содержит всего 40 инструкций, требующих для своего выполнения от 2 до 5 машинных циклов. Машинный цикл занимает 1,625 мкс при тактовой частоте 8МГц.

Большинство представителей семейства ST6 имеют встроенный АЦП с разрядностью 8 бит и числом каналов от 4 до 16 и 1 или 2 8-разрядных таймера с программируемым делителем. 8-разрядный таймер реализует функции выходного сравнения, входного захвата, перезапуск от внешнего прерывания, ШИМ.

На кристалле интегрированы схема сброса по включению и понижению напряжения питания, сторожевой таймер и внутренний тактовый генератор частотой 400 кГц, SPI и UART/

Напряжение питания от 3 до 6 В.

Микроконтроллеры семейства ST7. Процессорное ядро содержит:

* регистр-аккумулятор
* два индексных регистра X и Y
* регистр состояния процессора
* счетчик команд, адресующий 64 кбайт памяти
* 16-разрядный указатель стека

Система команд включает в себя 63 инструкции длиной от 1 до 4 байт. Время исполнения 1-байтной команды составляет 250 нс при внутренней тактовой частоте 8 МГц. В систему команд входят битовые операции и операции беззнакового умножения.

Семейство ST7 насчитывает более 50 представителей, ориентированных на выполнение различных задач:

* базовые микроконтроллеры общего назначения с 1 или 2 таймерными модулями, SPI И SCI
* микроконтроллеры, предназначены для обработки аналоговых сигналов, которые добавляют ко всему вышеперечисленному 8-разрядный АЦП с количеством каналов от 6 до 8, со временем преобразования 64 машинных цикла. Ряд моделей содержит встроенное EEPROM данных
* микроконтроллеры с последовательными интерфейсами SPI И SCI, IIC (скорость передачи данных до 400 кбит/с) и CAN (до 1 Мбит/с, USB со скоростью 1,5 Мбит/с)
* специализированные микроконтроллеры

Емкость однократно программируемого или FLASH ПЗУ составляет от 4 до 60 кбайт, емкость ОЗУ- от 256 байт до 2 кбайт. Версии с FLASH ПЗУ имеют возможность внутрисхемного программирования.

Таймерный модуль строится на основе 16-разрядного, так называемого «free-running» таймера (таймер без опции останова/запуска, включающийся после рестарта микроконтроллера, в дальнейшем его состояние можно только считать, а содержимое- только сбросить) с программируемым предделителем. Таймерный модуль может работать в одном из трех режимов: ШИМ, выходное сравнение (2 канала), входной захват (2 канала), формирование на внешнем выводе импульса заданной длины при наступлении внешнего события.

Семейство ST7 не требует применения внешних супервизорных схем, интегрируя на кристалле сторожевой таймер, схему сброса по включению и понижению питания, вспомогательный внутренний тактовый генератор.

Микроконтроллеры работают в диапазоне напряжений питания 3-5 В.

Большинство микроконтроллеров выпускается в пластиковых корпусах с количеством выводов 28/32/42. В зависимости от модели, часть или все линии параллельных портов ввода/вывода могут программироваться как источники внешних прерываний.

**FUJITSU**

Изготавливаемые по CMOC технологии с топологическими нормами 0,5 мкм недорогие микроконтроллеры идеально соответствуют применениям, для которых необходимы: высокая производительность, обеспечивающая обработку в реальном масштабе времени, малое потребление и невысокая стоимость, но для которых необходим богатый и мощный набор периферийных устройств.

Отличительные особенности:

* последние достижения КМОП-технологии
* масочное ПЗУ и флэш-память
* максимальная внутренняя тактовая частота от внешнего 4 МГц генератора до 24 МГц (42 нс)
* встроенный в кристалл контур ФАПЧ умножает частоту внешнего тактового генератора. Коэффициент умножения ФАПЧ может выбираться с помощью программного обеспечения
* ядро микроконтроллера с напряжением питания 3D обеспечивает низкий уровень электромагнитных помех и незначительное энергопотребление. Внутренний регулятор напряжения поддерживает работу при напряжении питания до 5В.
* Оптимизированный для приложений контроллера набор инструкций
* Данные битового и байтового типов, а также представленные в виде слова и двойного слова:
* 23 режима адресации
* разнообразие указателей
* 4-байтовая очередь команд
* знаковая команда умножения 16 бит х 16 бит
* EI2OS – независимая от ЦП функция автоматического преобразования с 16- канальным сервисом ввода/вывода
* 18-битовый таймер развертки
* сторожевой таймер
* быстрая обработка прерываний
* мощные функции прерывания – 8 программируемых уровней приоритета, свыше 30 возможных аппаратно-реализованных векторов и свыше 200 программно- реализованных векторов
* вычисления повышенной сложности, выполняемые на 32-разрядном аккумуляторе
* поддержка «заплаток» для программы – обход фрагментов кода в масочном ПЗУ через обновления во внешней памяти
* блок встраиваемой поддержки устранения ошибок обеспечивает дистанционное устранение ошибок
* режимы энергосбережения:
* 7 для часов с одним тактовым генератором или 10 для часов с двумя тактовыми генераторами
* режим пониженного энергопотребления, останов, прерывистый режим работы ЦП, аппаратное резервирование

**32-разрядные RISK микроконтроллеры CMOS серии FR**

Архитектура RISC представляет собой 32- разрядную архитектуру микропроцессорного ядра, которая призвана удовлетворить требования к высокой производительности в сочетании с низкой ценой, выдвигаемые существующими приложениями высокого класса для автомобильного, потребительского и телекоммуникационного сектора.

Будучи изначально разработанным в расчете на встраиваемые системы, ЦП обладает 16-разрядным форматом кодов команд, обеспечивающим максимум производительности при минимуме затрат, шириной внешней шины памяти и Кеша инструкций в ½ машинного слова, а также позволяет за каждый рабочий цикл шины захватывать две инструкции. Процессор обладает пятиступенчатым конвейером и блоком перемножения 32х32, а также он предусматривает новый блок циклического сдвига и модуль битового поиска, который позволяет отыскать первую единицу, нуль или различие в слове за один такт.

Несмотря на то, что семейство FR принадлежит к RISC- АРХИТЕКТУРЕ, ЦП содержит ряд расширенных инструкций для упрощения ассемблерного программирования, часто неизбежного во встраиваемых системах. Примером является сохранение половины регистрового банка в память.

Блоки, требующие быстрого отклика, соединены с ядром FR по 32-разрядной гарвардской шине. К числу таких блоков относятся: кэш инструкций, внутреннее ОЗУ, контроллер прямого доступа к памяти и блок побитового поиска. Ресурсы, к которым применяются обычные процедуры управления и опроса состояния, т.е. приемопередатчики UART? Таймеры и т.д., привязаны к 16-разрядной периферийной шине, называемой R-BUS. Шлюзом из R-BUS в ядро FR служит обычный модуль преобразования шины, сходный с тем, что применяется для интерфейса внешней шины.

Внешняя шина дает пользователю возможность обращаться к внешней памяти и другим устройствам целевой системы, назначенным на диапазон адресов памяти- это достигается благодаря поддержке восьми полностью настраиваемых линий выбора кристалла с внешними выводами, параметры которых: диапазон памяти, ширина шины, параметры циклов ожидания и выравнивание адресов – могут настраиваться независимо.

**MOTOROLA**

**Семейство DragonBall**

Хотя процессор MC68VZ328 разрабатывался, в первую очередь, для персональных информационных устройств, он может быть использован в широком диапазоне применений, включая: навигационные системы, обучающие устройства, интеллектуальные телефоны и пр.

Основные характеристики процессора:

* Полностью статическое синтезируемое процессорное ядро АДЧ68000
* 32-разрядная/ 24-разрядная внутренняя/внешняя адресная шина, способная адресовать до 8Х 16 Мбайт блоков с сигналами CSA, CSB, СSС и СSDб 16-разрядная внутренняя шина данных
* Тактовая частота от 0 до 33 МГц
* Производительность 5,4 при 33 МГц
* Шина внешнего интерфейса M68000 с динамической настройкой на 8- и 16-разрядные порты данных
* Модуль системной интеграции, реализующий функции выполняемые, обычно, внешней логикой:
* конфигурирование системы, программируемое отображение адресов
* не требующий склеивающей логики интерфейс с SRAM, DRAM, SDRAM, EPROM и FLASH памятью.
* 8 программируемых сигналов выбора кристаллов периферии с логикой генерации сигнала WAIT состояния
* 4 программируемых I/O прерывания с возможностью клавиатурных прерываний
* 5 входов IRQ общего назначения, программируемых на срабатывание по фронту/ уровню/ полярности
* до 47 индивидуально программируемых параллельных портов I/O? Совмещаемых с другими функциями
* программируемый отклик на векторы прерывания встроенных модулей периферии
* управление электросбережением
* контроллер DRAM/SDRAM
* поддержка 8/16-разрядных DRAM/SDRAM с регенерацией CAS-перед-RAS и саморегенерацией
* Два универсальных приемопередатчика (UART)? Поддержка протокола IrDA 1.0
* Два раздельных порта последовательного интерфейса периферии (SPI)
* Два 16-разрядных универсальных счетчика/таймера (разрешение 30 нс при 33 МГц)
* Автоматическая генерация прерываний и возможность каскадирования до 32-разрядного таймера
* Два модуля PWM
* Режим загрузки (инициирование системы и запуска программ/данных в память системы посредством UART)
* Управление потреблением
* программируемый синтезатор частоты, работающий с кварцами 32, 768/38,4 кГц, с полным управлением частотой
* режим stop с малым потреблением
* индивидуальное выключение модулей
* Напряжение питания от 2,7 до 3,3 В
* Потребление в активном режиме – 40мА, в standby режиме- 0,06мА
* Диапазон рабочих температур 0-70 градусов Цельсия.

# 2. Обобщенная архитектура микроконтроллеров. Типовые периферийные устройства.

Периферийные устройства предназначены для преобразования формы представления информации в процессе передачи данных от микропроцессора к внешним устройствам. Типовые примеры — устройства преобразования сигналов (аналого-цифровые и цифро-аналоговые частотные преобразователи), устройства человеко-машинного интерфейса (клавиатура, дисплей), устройства связи с другими системами.

**АЦП и ЦАП**

Принцип работы **АЦП** состоит в измерении уровня входного сигнала и выдаче результата в цифровой форме. В результате работы АЦП непрерывный аналоговый сигнал превращается в импульсный, с одновременным измерением амплитуды каждого импульса. **ЦАП** получает на входе цифровое значение амплитуды и выдает на выходе импульсы напряжения или тока нужной величины, которые расположенный за ним интегратор (аналоговый фильтр) превращает в непрерывный аналоговый сигнал.

***Внимание****! Для правильной работы АЦП входной сигнал не должен изменяться в течение времени преобразования, для чего на его входе обычно помещается схема выборки-хранения, фиксирующая мгновенный уровень сигнала и сохраняющая его в течение всего времени преобразования.*

На выходе ЦАП также может устанавливаться подобная схема, подавляющая влияние переходных процессов внутри ЦАП на параметры выходного сигнала.

**Основные типы АЦП:**

* *параллельные* — входной сигнал одновременно сравнивается с эталонными уровнями набором схем сравнения (компараторов), которые формируют на выходе двоичное значение. В таком АЦП количество компараторов равно (2 в степени N) — 1, где N — разрядность цифрового кода (для восьмиразрядного — 255), что не позволяет наращивать разрядность свыше 10-12.
* *последовательного приближения* — преобразователь при помощи вспомогательного ЦАП генерирует эталонный сигнал, сравниваемый с входным. Эталонный сигнал последовательно изменяется по принципу половинного деления (дихотомии), который используется во многих методах сходящегося поиска прикладной математики. Это позволяет завершить преобразование за количество тактов, равное разрядности слова, независимо от величины входного сигнала.
* *с измерением временных интервалов* — широкая группа АЦП, использующая для измерения входного сигнала различные принципы преобразования уровней в пропорциональные временные интервалы, длительность которых измеряется при помощи тактового генератора высокой частоты.
* *последовательного счета*, или однократного интегрирования (single-slope) — в каждом такте преобразования запускается генератор линейно возрастающего напряжения, которое сравнивается со входным. Обычно такое напряжение получают на вспомогательном ЦАП, подобно АЦП последовательного приближения.
* *двойного интегрирования (dual-slope)* — в каждом такте преобразования входной сигнал заряжает конденсатор, который затем разряжается на источник опорного напряжения с измерением длительности разряда.
* *следящие* — вариант АЦП последовательного счета, при котором генератор эталонного напряжения не перезапускается в каждом такте, а изменяет его от предыдущего значения до текущего. Наиболее популярным вариантом следящего АЦП является Sigma-Delta, работающий на частоте, значительно(в 64 и более раз) превышающей частоту дискретизации выходного цифрового сигнала. Компаратор такого АЦП выдает значения пониженной разрядности (обычно однобитовые —0/1), сумма которых на интервале дискретизации пропорциональна величине отсчета. Последовательность малоразрядных значений подвергается цифровой фильтрации и понижению частоты следования (decimation), в результате чего получается серия отсчетов с заданной разрядностью и частотой дискретизации.

**Основные типы ЦАП:**

* *взвешивающие* — с суммированием взвешенных токов или напряжений, когда каждый разряд входного слова вносит соответствующий своему двоичному весу вклад в общую величину получаемого аналогового сигнала; такие ЦАП называют также параллельными или многоразрядными(multibit)
* *Sigma-Delta*, по принципу действия обратные АЦП этого же типа. Входной цифровой сигнал подвергается значительной (64x и более) передискретизации и подается на модулятор, формирующий малоразрядные(обычно однобитовые) значения. Полученные в результате малоразрядные отсчеты управляют схемой выдачи эталонных зарядов, которые со столь же высокой частотой добавляются к выходному сигналу. Типы ЦАП, выдающих истинно одноразрядный поток, называют bitstream (поток битов) или PDM (Pulse Density Modulation — модуляция плотностью импульсов). Несколько другой тип представляют ЦАП с широтно-импульсной модуляцией (ШИМ, Pulse Width Modulation, PWM), когда на схему выборки-хранения аналогового сигнала выдаются импульсы постоянной амплитуды и переменной длительности, управляя дозированием выдаваемого на выход заряда.

**Параметры АЦП**

При последовательном возрастании значений входного аналогового сигнала Uвх(t) от 0 до величины, соответствующей полной шкале АЦП Uпш выходной цифровой сигнал D(t) образует ступенчатую кусочно-постоянную линию. Такую зависимость по аналогии с ЦАП называют обычно характеристикой преобразования АЦП. В отсутствие аппаратных погрешностей средние точки ступенек расположены на *идеальной прямой* 1 (рис. 24), которой соответствует идеальная характеристика преобразования. Для количественной оценки отличий реальной характеристики от идеальной используются следующие параметры:

**Статические параметры**

* *Разрешающая способность* — величина, обратная максимальному числу кодовых комбинаций на выходе АЦП. Разрешающая способность выражается в процентах, разрядах или децибелах. Например, 12-разрядный АЦП имеет разрешающую способность 1/4096, или 0,0245% от полной шкалы, или -72,2 дБ. Разрешающей способности соответствует шаг квантования, номинальное значение шага квантования h=Uпш/(2N-1), где Uпш — номинальное максимальное входное напряжение АЦП(опорное напряжение), соответствующее максимальному значению выходного кода, N — разрядность АЦП.
* *Погрешность полной шкалы* — относительная разность между реальным и идеальным значениями предела шкалы преобразования при отсутствии смещения нуля.
* *Погрешность смещения нуля* — значение выходного кода, когда входной сигнал АЦП равен нулю. Является аддитивной составляющей полной погрешности.
* *Нелинейность* — максимальное отклонение реальной характеристики преобразования D(Uвх) от *оптимальной* (линия 2 на рис. 24). Оптимальная характеристика находится эмпирически так, чтобы минимизировать значение погрешности нелинейности.
* *Дифференциальной нелинейностью* АЦП в данной точке k характеристики преобразования называется разность между значением кванта преобразования hk и средним значением кванта преобразования h. В спецификациях на конкретные АЦП значения дифференциальной нелинейности выражаются в долях ЕМР или процентах от полной шкалы.
* *Погрешность дифференциальной линейности* определяет два важных свойства АЦП: непропадание кодов и монотонность характеристики преобразования. Непропадание кодов — свойство АЦП выдавать все возможные выходные коды при изменении входного напряжения от начальной до конечной точки диапазона преобразования. Пример пропадания кода i+1 приведен на рис. 25. При нормировании непропадания кодов указывается эквивалентная разрядность АЦП — максимальное количество разрядов АЦП, для которых не пропадают соответствующие им кодовые комбинации.
* *Монотонность характеристики преобразования* — это неизменность знака приращения выходного кода D при монотонном изменении входного преобразуемого сигнала.
* *Температурная нестабильность* АЦ-преобразователя характеризуется *температурными коэффициентами погрешности* полной шкалы и погрешности смещения нуля.

**Динамические параметры**

Возникновение динамических погрешностей связано с дискретизацией сигналов, изменяющихся во времени.

* *Максимальная частота дискретизации (преобразования)* — это наибольшая частота, с которой происходит образование выборочных значений сигнала, при которой выбранный параметр АЦП не выходит за заданные пределы. Измеряется числом выборок в секунду. Выбранным параметром может быть, например, монотонность характеристики преобразования или погрешность линейности.
* *Время преобразования (tпр)* — это время, отсчитываемое от начала импульса дискретизации или начала преобразования до появления на выходе устойчивого кода, соответствующего данной выборке. Для одних АЦП, например, последовательного счета или многотактного интегрирования, эта величина является переменной, зависящей от значения входного сигнала, для других, таких как параллельные или последовательно-параллельные АЦП, а также АЦП последовательного приближения, примерно постоянной. При работе АЦП без УВХ время преобразования является апертуным временем.
* *Время выборки (стробирования)* — время, в течение которого происходит образование одного выборочного значения. При работе без УВХ равно времени преобразования АЦП.

**Параметры ЦАП**

При последовательном возрастании значений входного цифрового сигнала D(t) от 0 до 2N-1 через единицу младшего разряда (ЕМР) выходной сигнал Uвых(t) образует ступенчатую кривую. Такую зависимость называют обычно характеристикой преобразования ЦАП. В отсутствие аппаратных погрешностей средние точки ступенек расположены на идеальной прямой 1 (рис. 22), которой соответствует идеальная характеристика преобразования. Реальная характеристика преобразования может существенно отличаться от идеальной размерами и формой ступенек, а также расположением на плоскости координат. Большинство параметров аналогично рассмотренным выше параметрам для АЦП, ниже рассмотрены специфические:

Динамические параметры ЦАП определяются по изменению выходного сигнала при скачкообразном изменении входного кода, обычно от величины "все нули" до "все единицы" (рис. 23).

***Время установления*** — интервал времени от момента изменения входного кода (на рис. 23 t=0) до момента, когда в последний раз выполняется равенство |*U*вых-*U*пш|=*d*/2

***Скорость нарастания*** — максимальная скорость изменения Uвых(t) во время переходного процесса. Определяется как отношение приращения Uвых ко времени, за которое произошло это приращение. Обычно указывается в технических характеристиках ЦАП с выходным сигналом в виде напряжения. У ЦАП с токовым выходом этот параметр в большой степени зависит от типа выходного ОУ. Для перемножающих ЦАП с выходом в виде напряжения часто указываются частота единичного усиления и мощностная полоса пропускания, которые в основном определяются свойствами выходного усилителя.

***Интерфейсы***

Объединение модулей микропроцессорного устройства в единую систему и взаимодействие микропроцессора с внешними устройствами происходит с помощью интерфейса (от англ. «interface» — сопрягать, согласовывать).

Интерфейс должен обеспечивать:

* простое и быстрое соединение данного устройства с любым другим, имеющим такой же интерфейс;
* совместную работу устройств без ухудшения их технических характеристик;
* высокую надежность.

Под стандартным интерфейсом понимается совокупность аппаратных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных компонентов в системах и направленных на обеспечение информационной, электрической и

конструктивной совместимости компонентов.

Основными элементами интерфейса являются:

* совокупность правил обмена информацией (временные диаграммы и диаграммы состояний сигналов интерфейса)
* аппаратная реализация (контроллеры)
* программное обеспечение интерфейса (драйверы)

Для любого интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена — дуплексный, полудуплексный и симплексный:

* ***Дуплексный*** режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если пропускная способность в направлениях «туда» и «обратно» имеет существенно различающиеся значения, или симметричным.
* ***Полудуплексный*** режим позволяет передавать информацию «туда» и «обратно» поочередно, при этом интерфейс имеет средства переключения направления канала.
* ***Симплексный*** (односторонний) режим предусматривает только одно направление передачи информации (во встречном направлении передаются только вспомогательные сигналы интерфейса). В зависимости от способа передачи данных различают два вида интерфейса: последовательный и параллельный.

***В параллельном интерфейсе*** все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно (за один квант времени), то есть информация разворачивается в пространстве. Параллельный способ применяют в тех случаях, когда необходимо получить наивысшую пропускную способность канала передачи информации*.* Так как между отдельными проводниками шины для параллельной передачи данных существует электрическая емкость, то при изменении сигнала, передаваемого по одному из проводников, возникает помеха (короткий выброс напряжения) на других проводниках. С увеличением длины шины (увеличением емкости проводников) помехи возрастают и могут восприниматься приемником как сигналы. Поэтому рабочее расстояние для шины параллельной передачи данных ограничивается длиной 1-2 м, и только за счет существенного удорожания шины или снижения скорости передачи длину шины можно увеличить до 10-20 м

***Последовательный интерфейс*** для передачи данных использует одну сигнальную линию, по которой информационные биты передаются друг за другом последовательно, на каждый из них отводится свой квант времени (битовый интервал). При последовательной передачи информации разворачивается во времени. Последовательная передача позволяет сократить количество сигнальных линий и увеличить дальность связи. В последовательном канале асинхронный режим работы соответствует передаче всего массива информации без специальных сигналов синхронизации и пауз между словами, синхронный — с синхронизацией после передачи каждого слова, при этом возможна пауза любой длительности между моментами передачи. Пример стандартного последовательного интерфейса — RS-232 (COM-порты в IBM PC совместимых компьютерах). Последовательный интерфейс подразделяют на синхронный и асинхронный.

В ***синхронном*** интерфейсе каждый передаваемый бит данных сопровождается импульсом синхронизации информирующим приемник о наличии на линии информационного бита. Следовательно, между передатчиком и приемником должны быть протянуты минимум три провода: два для передачи импульсов синхронизации и бит данных, а также общий заземленный проводник. Если же передатчик (например, микропроцессор) и приемник (например, персональный компьютер) разнесены на несколько метров, то каждый из сигналов (информационный и синхронизирующий) придется посылать либо по экранированному кабелю, либо с помощью витой пары проводов, один из которых заземлен или передает сигнал, инверсный основному.

В ***асинхронном*** интерфейсе у передатчикаи приемника нет общего генератора синхроимпульсов и синхронизирующий сигнал не посылается вместе с данными. А для синхронизации процесса передачи данных используются внутренние встроенные генераторы, настроенные на одну частоту, и некий оговоренный двумя взаимодействующими сторонами формат передачи данных. Данный формат разработан еще в 70-х годах прошлого столетия поддерживается практически всеми микропроцессорными устройствами.

Стандартный формат асинхронной последовательной передачи данных содержит n пересылаемых бит информации (при пересылке символов n равно 7 или 8 битам) и 3-4 дополнительных бита: стартовый бит, бит контроля четности (или нечетности) и 1 или 2 стоповых бита (рис. 2.2,а). Бит четности (или нечетности) может отсутствовать.

Рисунок 2.2. Формат асинхронной последовательной передачи данных

Передатчик может начать пересылку символа в любой момент времени посредством генерирования стартового бита. Затем происходит передача битов символа, начиная с младшего значащего бита, за которым следует дополнительный бит контроля по четности или нечетности. Далее с помощью стопового бита линия переводится в состояние логической 1 (рис.2.2,б). Состояние логической 1 должно поддерживаться в течение промежутка времени, равного 1 или 2 временам передачи бита.Промежуток времени от начала стартового бита до конца стопового бита (стоповых бит) называется кадром. Сразу после стоповых бит передатчик может посылать новый стартовый бит, если имеется другой символ для передачи; в противном случае уровень логической 1 может сохраняться на протяжении всего времени, пока бездействует передатчик. Новый стартовый бит может быть послан в любой момент времени после окончания стопового бита, например, через промежуток времени, равный 0,43 или 1,5 времени передачи бита. Передний фронт стартового бита сигнализирует о начале поступления передаваемой информации, а момент его появления служит точкой отсчета времени для считывания бит данных(запуск тактового генератора приемника). Стоповый бит предоставляет время для записи принятого символа в буфер приемника и обеспечивает возможность выявления ошибки кадра.

# 3. Средства программирования и отладки микроконтроллеров. Инструментальные и программные средства. Внутрисхемный эмулятор, принцип работы. Классификация внутрисхемных эмуляторов. Дизассемблеры MCS-51. Макроассемблер. Компоновщик. Отладчик/симулятор. Подсистема точек останова. Процессор точек останова. Трассировщик. Профилировщик. Интегрированная среда разработки.

О правильности функционирования микропроцессорной системы на уровне "черного ящика" с полностью неизвестной внутренней структурой можно говорить лишь тогда, когда произведены ее испытания, в ходе которых реализованы все возможные комбинации входных воздействий, и в каждом случае проверена корректность ответных реакций. Однако исчерпывающее тестирование имеет практический смысл лишь для простейших элементов систем. Следствием этого является тот факт, что ошибки проектирования встречаются при эксплуатации, и для достаточно сложных систем нельзя утверждать об их отсутствии на любой стадии жизни системы. В основе почти всех методов испытаний лежит та или иная гипотетическая модель неисправностей, первоисточником которой служат неисправности, встречающиеся в практике. В соответствии с моделью в рамках каждого метода предпринимаются попытки создания тестовых наборов, которые могли бы обеспечить удовлетворительное выявление моделируемых неисправностей. Любой метод тестирования хорош ровно настолько, насколько правильна лежащая в его основе модель неисправности.

Важным моментом является правильный выбор соотношения между степенью общности модели, стоимостью и степенью сложности формирования и прогона тестов, ориентированных на моделируемые неисправности. Чем конкретнее модель, тем легче создать для нее систему тестов, но тем выше вероятность того, что неисправность останется незамеченной. Если же модель неисправностей излишне общая, то из-за комбинаторного возрастания числа необходимых тестовых наборов и/или времени вычислений, требуемого для работы алгоритмов формирования тестов, она станет непрактичной и пригодной только для несложных систем.

**Обнаружение ошибки и диагностика неисправности**

Дефект не может быть обнаружен до тех пор, пока не будут созданы условия для возникновения из-за него неисправности, результат которой должен быть, в свою очередь, передан на выход испытуемого объекта, для того чтобы сделать неисправность наблюдаемой. Метод испытаний должен позволить генерировать тесты, ставящие испытуемый объект в условия, при которых моделируемые неисправности проявляли бы себя в виде обнаруживаемых ошибок. Если испытуемый объект предназначен для эксплуатации, то при обнаружении ошибки необходимо произвести локализацию неисправности с целью ее устранения путем ремонта или усовершенствования испытуемого объекта.

*Диагностика неисправности -* процесс определения причины появления ошибки по результатам тестирования.

*Отладка -* процесс обнаружения ошибок и определение источников их появления по результатам тестирования при проектировании микропроцессорных систем. **Средствами отладки** являются приборы, комплексы и программы.

Точность, с которой тот или иной тест локализует неисправности, называется его разрешающей способностью. Требуемая разрешающая способность определяется конкретными целями испытаний. Например, при испытаниях аппаратуры в процессе эксплуатации для ее ремонта часто необходимо установить, в каком сменном блоке изделия имеется неисправность. В заводских условиях желательно осуществлять диагностику неисправности вплоть до уровня наименьшего заменяемого элемента, чтобы минимизировать стоимость ремонта. В лабораторных условиях в процессе отладки опытного образца необходимо определять природу неисправности (физического или нефизического происхождения). В случае возникновения и проявления дефекта требуется локализовать место неисправности с точностью до заменяемого элемента, а при проявлении субъективной неисправности - с точностью до уровня представления (программного, схемного, логического и т. д.), на котором была внесена неисправность, и места. Так как процесс проектирования микропроцессорной системы содержит неформализуемые этапы, то отладка системы предполагает участие человека.

**Свойство контролепригодности системы.**

Успех отладки зависит от того, как спроектирована система, предусмотрены ли свойства, делающие ее удобной для отладки, а также от средств, используемых при отладке. Для проведения отладки проектируемая микропроцессорная система должна обладать свойствами управляемости, наблюдаемости, предсказуемости.

*Управляемость -* свойство системы, при котором ее поведение поддается управлению, т. е. имеется возможность остановить функционирование системы в определенном состоянии, и затем снова ее запустить.

*Наблюдаемость -* свойство системы, позволяющее проследить за поведением системы, сменой ее внутренних состояний.

*Предсказуемость -* свойство системы, позволяющее установить систему в состояние, из которого все последующие состояния могут быть предсказаны.

**Функции средств отладки**

Сроки и качество отладки системы зависят от средств отладки. Чем совершеннее приборы, имеющиеся в распоряжении инженера-разработчика, тем скорее можно начать отладку аппаратуры и программ и тем быстрее обнаружить ошибки, локализовать источники, устранение которых обойдется дороже на более позднем этапе проектирования.

Средства отладки должны:

* управлять поведением системы или/и ее модели на различных уровнях абстрактного представления
* собирать информацию о поведении системы или/и ее модели, обрабатывать и представлять на различных уровнях абстракции
* преобразовывать системы, придавать им свойства контролепригодности
* моделировать поведение внешней среды проектируемой системы.

Под управлением поведением системы или ее модели понимаются определение и подача входных воздействий для запуска или останова системы или ее модели, для перевода в конкретное состояние последних. Чтобы определить место субъективной неисправности, которая может быть внесена на любой стадии проектирования, необходимо уметь собирать информацию о поведении системы и представлять ее в тех формах, которые приняты для данного проекта. Например, это могут быть временные диаграммы, принципиальные электрические схемы, язык регистровых передач, ассемблер и др.

В общем случае нельзя локализовать источник ошибки проектируемой системы, имея информацию о поведении системы только на ее внешних выводах, поэтому проектируемую систему преобразовывают. Например, прежде чем изготовлять однокристальную микроЭВМ с теми или иными "зашивками" ПЗУ, программы отлаживают на эмуляционном кристалле, у которого магистраль выведена на внешние контакты и вместо ПЗУ установлено ОЗУ.

**Автономная отладка**

Процесс отладки прототипа проектируемой системы должен начинаться с отладки аппаратуры и отладки программ.

Отладка аппаратуры предполагает тестирование отдельных устройств микропроцессорной системы - процессора, ОЗУ, контроллеров, блока питания, генератора тактовых импульсов путем подачи тестовых входных воздействий и приема ответных реакций. Тестовые входные воздействия и ответные реакции определяются, исходя из спецификаций на устройства, а также структурных схем устройств. При этом проверяются реальная аппаратура прототипа, спецификации, структурные схемы и отлаживаются тесты. После отладки отдельных устройств проверяется их взаимодействие. Процессор системы работает с шинами адресов, данных и управления. Анализируя их сигналы, можно проконтролировать выполнение программы в процессоре.

Поскольку ША и ШД синхронные, их работу лучше всего проверить с помощью методов логических состояний. Перед анализом последовательностей данных на этих шинах необходимо удостовериться в том, что сигналы, управляющие взаимодействием процессора с другими устройствами, выдаются в соответствующем порядке. Поскольку ШУ состоит из линий, работающих асинхронно, необходимо просматривать сигналы многих линий в течение одного и того же промежутка времени. Для анализа асинхронной работы линий управления необходимо также наблюдать за сигналами на них при возникновении определенного события, чтобы можно было четко разделить и идентифицировать различные состояния линий. Например, среди сигналов ШУ могут быть сигналы длительностью всего несколько наносекунд, но могут также возникать кратковременные ложные узкие импульсы, вызванные перекрестными помехами или шумами.

После того как доказана работоспособность ШУ, проводится дальнейшая проверка работы аппаратуры при различных режимах адресации процессора и кодах выбираемых данных. Для проверки выполнения процессором инструкций разрабатывается тестовая программа, которая помещается в ОЗУ или ППЗУ. При этом проверяется временная диаграмма сигналов и прохождения данных в системе (как осуществляется передача информации по отношению к строб-сигналам). Если тестовая программа - системный проверяющий тест, - пройдет успешно, можно утверждать, что автономно аппаратура отлажена.

При автономной отладке аппаратуры могут потребоваться приборы, умеющие:

1. выполнять функции аналогового прибора, т. е. измерять напряжение и ток; воспроизводить форму сигнала, подавать импульсы определенной формы и т. д.;
2. подавать последовательность сигналов одновременно на несколько входов в соответствии с заданной временной диаграммой или заданным алгоритмом функционирования аппаратуры, представленным в спецификации на языке высокого уровня, или другим способом;
3. собирать значения сигналов многих линий в течение одного и того же промежутка времени, который определяется задаваемыми, программируемыми событиями - комбинацией или последовательностью сигналов на линиях, например, ложным сигналом на линии;
4. обрабатывать и представлять собранную информацию либо в виде временной диаграммы, либо в виде диаграммы или таблицы логических состояний, либо на языке высокого уровня, например, языке регистровых передач.

Для автономной отладки аппаратуры широко используются осциллографы, вольтметры, амперметры, частотомеры, генераторы импульсов, позволяющие отлаживать аппаратуру на схемном уровне. Чтобы автономно отладить аппаратуру МПС на более высоком уровне, применяют логические анализаторы, генераторы слов, пульты, комплексы диагностирования.

**Отладка программ**

Отладка программ микропроцессорной системы проводится, как правило, на тех же ЭВМ, на которых велась разработка программ, и на том же языке программирования, на котором написаны отлаживаемые программы, и может быть начата на ЭВМ даже при отсутствии аппаратуры МПС. При этом в системном программном обеспечении ЭВМ должны находиться программы (интерпретаторы или эмуляторы), моделирующие функции отсутствующих аппаратных средств. Так, разработка и автономная отладка программных средств может вестись на больших ЭВМ, миниЭВМ, микроЭВМ, система команд которых не совпадает с системой команд используемого микропроцессора. Кроме того, при отладке программ может отсутствовать внешняя среда микропроцессорной системы, ее также необходимо моделировать.

Проверка корректности программ, т.е. проверка соответствия их внешним спецификациям, осуществляется тестированием. Программы проверяются на функционирование с различными исходными данными. Результаты функционирования программ сравниваются с эталонными значениями.

Отладка программ подразделяется на следующие этапы: планирование отладки; составление тестов и задания на отладку; исполнение программ; информирование о результатах исполнения программ по заданным исходным данным; анализ результатов, обнаружение ошибок и локализация неисправностей.

Существует два способа начального тестирования программ: пошаговый режим и трассировка программ.

В *пошаговом режиме* программа выполняется по одной команде за один раз, а пользователь анализирует содержимое памяти, регистров и т.д., чтобы проверить, соответствуют ли результаты ожидаемым. Пошаговый режим может быть трудоемким, если средства отладки будут требовать отдельных команд после каждого шага для того, чтобы показать необходимую информацию в понятном для пользователя виде. Имеются средства отладки, автоматически показывающие после каждого шага содержимое регистров процессора и ячеек памяти, используемых в последней команде, и несколько следующих команд. Пошаговый режим является весьма мощным средством предварительного тестирования, так как позволяет обнаруживать неисправности, прежде чем они существенно исказят программу и данные. Кроме того, неоднократно проходя отдельными шагами через один и тот же участок объектной программы, программист может легко изменять содержимое регистров и ячеек памяти, особенно если средства отладки имеют динамически обновляемый дисплей, и тем самым проверить работу программы в разных условиях. Этот интерактивный режим отладки программы позволяет разработчику постоянно упреждать, что будет делать его программа, и оперативно обнаруживать ошибку. Однако пошаговый режим с автоматическим показом результатов возможен только тогда, когда средства отладки содержат в своем составе дисплей с прямым доступом в память, так как после каждого шага на экране дисплея нужно показывать большой объем информации.

Исполнение программ осуществляется по шагам последовательно во времени и в соответствии с заданиями, содержащимися в операторах. При этом производится переработка значений переменных и определение оператора приемника. Если в ходе исполнения программы регистрируется последовательность операторов, реализуемых на каждом шаге процесса, то получается трасса или маршрут исполнения программы, который для конкретной программы зависит только от значений исходных данных.

Трассировка программ больше пригодна для отладочных средств, имеющих медленный, последовательный терминал. Программа-отладчик выполняет непрерывно команду за командой и выводит содержимое регистров процессора на терминал после каждого шага. Некоторые отладчики выводят также на терминал команды в дизассемблерной форме. Но при этом способе содержимое памяти не выводится на терминал и разработчик должен сам делать выводы об изменениях в ней. Отслеживание программы продолжается автоматически до тех пор, пока не будет остановлено извне. Результатом трассировки программы будут данные на экране дисплея или же в случае использования в качестве терминала печатающего устройства - длинная распечатка с ходом выполнения программы. Программист, анализируя эти данные, может обнаружить ошибки. Трассировка программ не дает, однако, возможности изменять содержимое памяти и регистров и может послужить причиной того, что программа разрушит себя или свои данные прежде, чем отслеживание будет остановлено.

Отдельные участки программы после проверки, используя пошаговый режим или трассировку, можно объединить и проверить с помощью установки контрольных точек, вводимых в программу и прерывающих ее исполнение, для передачи управления программе-отладчику. По контрольным точкам можно по желанию выполнить избранные участки программы и проанализировать результаты. Контрольные точки устанавливаются обычно для конкретной команды, но в некоторых системах предусматриваются прерывания программы при чтении или записи данных в определенные ячейки памяти. Возможны и более сложные условия прерывания программы.

Расстановка контрольных точек предполагает, что программист связывает с ней точный адрес памяти. Для некоторых отладчиков программист задает абсолютный шестнадцатеричный адрес. Последние отладчики допускают символьные значения адресов, которые программист определяет в исходной программе; это позволяет значительно экономить время, распечатывая после каждого редактирования и транслирования программы новую копию листинга.

При тестировании можно планировать проверку всех возможных маршрутов исполнения программы для разных исходных переменных. Однако это реализуемо только для очень простых программ небольшого объема при малых диапазонах изменения исходных данных. Поэтому при планировании отладки программ применяют критерии полноты тестирования, которые, однако, не гарантируют полной проверки программ. Выбор критерия зависит от наличия ресурсов для тестирования и структурной сложности отлаживаемой программы. Критерии характеризуются глубиной контроля программ и объемом проверок.

В процессе отладки основная часть неисправностей в программах обнаруживается и затем устраняется. Однако всегда возможен пропуск нескольких неисправностей.

Средства отладки программ должны:

1. управлять исполнением программ (останавливать, изменять порядок, запускать и т. д.)
2. собирать информацию о ходе выполнения программы
3. обеспечивать обмен информацией (диалог) между программистом и ЭВМ на уровне языка программирования
4. моделировать работу отсутствующих аппаратных средств микропроцессорной системы.

**Комплексная отладка микропроцессорных систем**

Как правило, микропроцессорная система - это система реального времени, т. е. корректность ее функционирования зависит от времени выполнения отдельных программ и скорости работы аппаратуры. Поэтому система считается отлаженной после того, как рабочие программы правильно функционируют на действительной аппаратуре системы в реальных условиях. Дополнительным свойством, которым должны обладать средства комплексной отладки по сравнению со средствами автономной отладки, является возможность управления поведением МПС и сбора информации о ее поведении в реальном времени.

Тенденция развития средств отладки микропроцессорных систем состоит в объединении свойств нескольких приборов в одном комплексе, в создании универсальных средств, пригодных для автономной отладки аппаратуры, генерации и автономной отладки программ и комплексной отладки системы. Эти средства позволяют вести разработку и отладку, постепенно усложняя аппаратуру и программы. При этом разработка, изготовление и отладка планируются поэтапно с нарастанием сложности; новая, неотлаженная аппаратура и программа вводятся в создаваемую систему, присоединяются к проверенной ее части.

Если отладка программ ведется с использованием эмуляционного ОЗУ, а затем изготовляются микросхемы ПЗУ, то микропроцессорная система должна быть протестирована.

Средства отладки на последних этапах не должны влиять на правильность функционирования системы, вносить задержки, дополнительные нагрузки.

При комплексной отладке наряду с детерминированным используется статистическое тестирование, при котором МПС проверяется при изменении исходных переменных в соответствии со статистическими законами работы источников информации. Полнота контроля работоспособности проектируемой системы возрастает за счет расширения диапазона возможных сочетаний переменных и соответствующих им логических маршрутов обработки информации.

Существуют пять основных приемов комплексной отладки микропроцессорной системы:

1. останов функционирования системы при возникновении определенного события
2. чтение (изменение) содержимого памяти или регистров системы
3. пошаговое отслеживание поведения системы
4. отслеживание поведения системы в реальном времени
5. временное согласование программ.

Комплексная отладка завершается приемосдаточными испытаниями, показывающими соответствие спроектированной системы техническому заданию. Для проведения комплексной отладки МПС используют логические анализаторы и комплексы: оценочные, отладочные, развития микропроцессоров, диагностирования, средств отладки.

**Аппаратные средства разработки и отладки микроконтроллеров.**

К числу основных инструментальных средств отладки относятся:

* Внутрисхемные эмуляторы;
* Программные симуляторы;
* Платы развития (оценочные платы);
* Мониторы отладки;
* Эмуляторы ПЗУ.

Данный список не исчерпывает всех типов существующих инструментальных средств отладки. Кроме указанных, существуют и комбинированные устройства и наборы, которые позволяют компенсировать недостатки основных средств, взятых порознь.

# 4. Функциональные возможности внутрисхемных эмуляторов. Достоинства и недостатки внутрисхемных эмуляторов. Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051 PICE-51. Программные симуляторы. Платы развития. Отладочные мониторы. Эмуляторы ПЗУ.

**Внутрисхемные эмуляторы**

*Внутрисхемный эмулятор* - программно аппаратное средство, способное замещать собой эмулируемый процессор в реальной схеме. Внутрисхемный эмулятор - это наиболее мощное и универсальное отладочное средство. По сути дела, “хороший” внутрисхемный эмулятор делает процесс функционирования отлаживаемого контроллера прозрачным, т.е. легко контролируемым, произвольно управляемым и модифицируемым по воле разработчика.

Функционально внутрисхемные эмуляторы делятся на стыкуемые с внешней вычислительной машиной (обычно это бывает IBM PC), и функционирующие автономно.

Автономные внутрисхемные эмуляторы имеют индивидуальные вычислительные ресурсы, средства ввода-вывода, не требуют для своей нормальной работы стыковки с какими-либо внешними вычислительными средствами, но за это пользователю приходится расплачиваться либо существенно более высокой ценой, либо пониженными функциональными и сервисными возможностями по сравнению с аналогичными моделями, стыкуемыми с IBM PC.

Обычно, стыковка внутрисхемного эмулятора с отлаживаемой системой производится при помощи эмуляционного кабеля со специальной эмуляционной головкой. Эмуляционная головка вставляется вместо микроконтроллера в отлаживаемую систему. Если микроконтроллер невозможно удалить из отлаживаемой системы, то использование эмулятора возможно, только если этот микроконтроллер имеет отладочный режим, при котором все его выводы находятся в третьем состоянии. В этом случае для подключения эмулятора используют специальный адаптер-клипсу, который подключается непосредственно к выводам эмулируемого микроконтроллера.

Как минимум, эмулятор содержит следующие функциональные блоки:

* Отладчик;
* Узел эмуляции микроконтроллера;
* Эмуляционная память;
* Подсистема точек останова.
* Более продвинутые модели могут содержать дополнительно:
* Процессор точек останова;
* Трассировщик;
* Профилировщик (анализатор эффектив-ности программного кода);
* Таймер реального времени;
* Программно-аппаратные средства, обеспечивающие возможность чтения и модификации ресурсов эмулируемого процессора “на лету”, т.е. в процессе выполнения программы пользователя в реальном времени;
* Программно-аппаратные средства, обеспечивающие синхронное управление, необходимое для эмуляции в мультипроцессорных системах;
* Интегрированную среду разработки.

Некоторые модели внутрисхемных эмуляторов могут предоставлять пользователям и другие дополнительные возможности. Среди них отметим одну, хотя и достаточно специфическую, но в ряде случаев имеющую принципиальное значение: возможность построения многоэмуляторных комплексов, необходимых для отладки мультипроцессорных систем. Отличительной особенностью такого комплекса является возможность синхронного управления (с одного компьютера) несколькими эмуляторами.

В общем случае, различные модели внутрисхемных эмуляторов могут предоставлять пользователю возможности по контролю и управлению функционированием отлаживаемых устройств с разного рода ограничениями. Например, это может быть некорректное обрабатывание прерываний в пошаговом режиме, или запрет на использование последовательного порта и т.п.. Также необходимо помнить, что каждая реальная модель эмулятора имеет свой набор поддерживаемых компиляторов. Некоторые фирмы-производители эмуляторов сознательно идут на ограничение количества поддерживаемых компиляторов, в первую очередь это характерно для западных производителей. В этих случаях эмулятор умеет пользоваться только одним символьным форматом.

Конечно, столь широкий набор функциональных возможностей делает внутрисхемные эмуляторы наиболее мощным и универсальным средством отладки.

**Внутрисхемный эмулятор PICE-51**

PICE-51 — эмулятор нового поколения, созданный с применением передовых технологий разработки аппаратуры и программного обеспечения.

Применение программируемых матриц большой ёмкости позволило резко сократить размеры эмулятора без какого-либо ущерба его функциональным возможностям, минимизировать отклонения электрических и частотных характеристик эмулятора от характеристик эмулируемого процессора и, тем самым, добиться максимальной точности эмуляции на частотах до 30 МГц при напряжениях питания от 3,3 В до 5 В.

Перезагружаемая аппаратная структура эмулятора обеспечивает эмуляцию практически всех микроконтроллеров семейства 8051 как отечественного производства, так и фирм Intel, Philips, Siemens, Atmel, Dallas, Temic, OKI, AMD, MHS и других.

Мощный программный интерфейс в среде Windows, представляет собой интегрированную среду разработки, поддерживающую все этапы разработки программного обеспечения от написания исходного текста программы до её компиляции и отладки. Программа поддержки эмулятора ориентирована на отладку программ на языке высокого уровня по исходному тексту.

Эмулятор состоит из основной платы размером 80 х 76 мм, сменного адаптера под конкретный процессор и сменной эмуляционной головки под конкретный тип корпуса. На основной плате реализованы: трассировщик и процессор точек останова. Плата сменного адаптера содержит эмулирующий процессор под конкретный тип микроконтроллера. Эмуляционные головки обеспечивают установку эмулятора в колодки DIP и PLCC на плате пользователя. Питание эмулятора осуществляется от блока питания +5 В; 0,5 А или непосредственно от отлаживаемого устройства. Связь с компьютером обеспечивается по гальванически развязанному каналу RS-232C на скорости 115 Кбод.

Эмулируемые микроконтроллеры:

* *Intel* 8031, 80C31, 8032, 80С32, 8051, 80С51, 8052, 8хС52, 8хС54, 8хС58, 80C152JA/JB/JC/JD, 8xL/C51FA/FB/FC, 8xC51RA/RB/RC, 80C51GB;
* *Philips* 8031, 8051, 80С31, 80С32, 8хС51, 8хС52/54/58, 8хС51 FA/FB/FC; 8xC51RA+/RB+/RC+/RD+, 80CL410, 80С552, 80С562, 8ХС550, 8ХС652, 8ХС654, 8ХС528, 8ХС524, 8ХС575, 8ХС576, 80С851, 89С535, 89С536, 89С538;
* *Atmel* 89С51, 89С52, 89С55, 89С2051, 89С1051, 89S8252, 89S53 (последние два - только с внешней памятью программ);
* *Siemens* SAB501/502/504/505/511/513
* *Dallas Semiconductor* 80C310, 8ХС520, 8ХС530; MHS 80C31, 80C51, 80C32, 80C52, 80C154; OW80C31, 80C51, 80C154;
* *AMD* 80C31, 80C51, 80C52;

и другие микроконтроллеры семейства 8051.

***Характеристики аппаратуры***

* Точная эмуляция — отсутствие каких-либо ограничений на использование программой пользователя ресурсов микроконтроллера.
* До 256К эмулируемой памяти программ и данных. Поддержка банкированной модели памяти. Распределение памяти между эмулятором и устройством пользователя с точностью до 1-го байта.
* До 512К аппаратных точек останова по доступу к памяти программ и данных.
* Аппаратная поддержка для отладки программ на языках высокого уровня.
* Трассировка 8 произвольных внешних сигналов.
* 4 выхода синхронизации аппаратуры пользователя.
* Трассировщик реального времени с буфером объёмом от 16К до 64К фреймов по 64 бита с доступом «на лету». Трассировка адреса, данных, сигналов управления, таймера реального времени и 8-ми внешних сигналов пользователя.
* Программируемый фильтр трассировки.
* Аппаратный процессор точек останова с возможностью задания сложного условия останова эмуляции по комбинации сигналов адреса, данных, управления, 8-ми внешних сигналов, таймера реального времени, счётчиков событий и таймера задержки.
* Четыре комплексных точки останова, которые могут быть использованы независимо или в комбинациях по условиям AND/OR/IF-THEN.
* 48-разрядный таймер реального времени.
* Прозрачная эмуляция — доступ «на лету» к эмулируемой памяти, точкам останова, процессору точек останова, буферу трассировки, таймеру реального времени.
* Управляемый генератор тактовой частоты для эмулируемого процессора. Возможность плавного изменения тактовой частоты от 500 кГц до 40 МГц.
* Гальванически развязанный от компьютера канал связи RS-232C со скоростью обмена 115 Кбод.
* Встроенная система самодиагностики аппаратуры эмулятора.
* Тактовая частота базового варианта 28 МГц. С быстрой матрицей — 40 МГц.
* Поддержка низковольтовых микроконтроллеров с напряжениями питания от 3,3 В до 5 В.
* Двухпортовая память, обеспечивающая доступ «на лету» к памяти программ, внешней памяти данных, точкам останова, трассировщику, процессору точек останова.
* Установка карты памяти с точностью до 1 байта.
* Программируемый генератор тактовой частоты эмулируемого процессора.
* Более точная и надёжная эмуляция — отсутствие кабелей и шлейфов, снижающих надёжность работы и вносящих отклонения в электрические и временные характеристики процессора.
* Возможность поддержки микроконтроллеров, отличающихся от стандартного 51-го по циклу шины (Dallas 320/520)
* Более точная эмуляция процессоров 87C51FX/RX+ в режиме внутренней памяти программ при обращении к внешней памяти данных командами
* MOVX A,@Rx, MOVX @Rx,A
* Обновление версий эмулятора только заменой программного обеспечения — возможность обновления через web-сервер.
* Отсутствие конфигурационных перемычек — все установки аппаратных конфигураций осуществляются через программу поддержки

***Характеристики программного обеспечения***

* Программное обеспечение ориентировано на работу в среде Windows на IBM-совместимых компьютерах с процессорами типа 386/486/Pentium.
* Встроенный многооконный редактор предназначен для написания исходных текстов программ. Редактор поддерживает операции с блоками текста, поиск/замену, цветовое выделение синтаксических конструкций языка ассемблера и Си.
* Неотъемлемая, встроенная и действительно интегрированная среда разработки. Для разработчика стирается грань между редактором и отладчиком.
* Встроенный менеджер проектов обеспечивает автоматическую компиляцию программ. Все опции задаются в диалоговой форме. Переход от редактирования исходного текста к отладке и обратно происходит «прозрачно», т. е. менеджер проектов автоматически запускает компиляцию проекта при необходимости.
* PICE-51 обеспечивает символьную отладку и отладку по исходному тексту для программ, созданных с помощью следующих компиляторов:
* ассемблер ASM51 фирмы Intel;
  + компилятор PL/M фирмы Intel;
  + ассемблер и компилятор Си фирмы IAR Systems;
  + ассемблер и компилятор Си фирмы Avocet Systems Inc./HiTech; -
  + ассемблер и компилятор Си фирмы Keil Software Inc..
* Автоматическое сохранение и загрузка файлов конфигурации аппаратуры, интерфейса и опций отладки. Обеспечивается совместимость файлов конфигурации с симулятором PDS-51. Обеспечена переносимость проектов между эмулятором PICE-51 и симулятором PDS-51.
* Возможность настройки цветов, шрифтов и других параметров для всех окон одновременно и для каждого окна в отдельности.
* Эмулятор снабжён печатным руководством по эксплуатации и контекстным электронным руководством, в которых детально описаны его принципы работы, команды, меню, горячие клавиши.

**Программные отладчики-симуляторы**

Программный отладчик-симулятор - программное средство, способное имитировать работу микроконтроллера и его памяти. Как правило, хороший симулятор позволяет пользователю:

* Загружать файлы программ во всех популярных форматах;
* Контролировать и модифицировать состояние ресурсов симулируемого микроконтроллера. В процессе отладки модель «выполняет» программу, и на экране компьютера отображается текущее состояние модели;
* Запускать отлаживаемую программу в пошаговом или непрерывном режимах, вперёд и назад, задавать условные и безусловные точки останова, контролировать и свободно модифицировать содержимое ячеек памяти и регистров симулируемого микропроцессора. С помощью симулятора можно быстро проверить логику выполнения программы, правильность выполнения арифметических операций;
* Вести высокоуровневую символьную отладку программ. Такую возможность предоставляют наиболее «продвинутые» отладчики, обычно их называют высокоуровневыми (High-Level Debuggers), и естественно при условии использования компиляторов, поставляющих необходимую отладочную информацию;
* Контролировать и модифицировать параметры встроенных периферийных устройств, таких, как таймеры, порты, АЦП, системы прерываний;
* Получать, по результатам прогона отлаживаемой программы, статистическую информацию, необходимую для оптимизации её структуры

Некоторые модели симуляторов могут содержать ряд дополнительных программных средств, таких, например, как: интерфейс внешней среды, встроенную интегрированную среду разработки.

Наличие интерфейса внешней среды позволяет пользователю создавать и гибко использовать модель внешней среды микроконтроллера, функционирующую и взаимодействующую с отлаживаемой программой по заданному алгоритму.

Наличие в программной оболочке симулятора элементов интегрированной среды разработки позволяет управлять не только процессом симуляции, но и поддерживает в целом процесс отладки, начиная от написания исходного текста программы до её компиляции и отладки, обеспечивает простое и быстрое взаимодействие с другими инструментальными средствами (внутрисхемным эмулятором, программатором).

В общем случае, различные модели программных отладчиков-симуляторов могут предоставлять пользователю возможности по контролю и управлению функционированием отлаживаемых устройств с разного рода ограничениями. Основными характеристиками, определяющими качество и функциональность симулятора, принято считать:

* Перечень корректно поддерживаемых (симулируемых) микроконтроллеров;
* Перечень корректно поддерживаемых (симулируемых) встроенных в микроконтроллер узлов периферии;
* Скорость, детальность и точность симуляции;
* Перечень поддерживаемых компиляторов и форматов.

Очевидной особенностью программных симуляторов является то обстоятельство, что исполнение программ, загруженных в симулятор, происходит в масштабе времени, отличном от реального. Однако низкая цена, возможность ведения отладки даже в условиях отсутствия макета отлаживаемого устройства, делают программные симуляторы весьма эффективным средством отладки. Отдельно необходимо подчеркнуть, что существует целый класс ошибок, которые могут быть обнаружены только при помощи симулятора.

# 5. Структурная организация микроконтроллера i8051. Общие характеристики. Арифметико-логическое устройство. Назначение выводов. Организация портов ввода вывода. Альтернативные функции.

# 6. Особые режимы работы (загрузка, стирание, холостой ход, пониженное энергопотребление). Режим ХХ. Режим ВНП.

# 7. Организация памяти. Работа с внешней памятью. Память программ (ПЗУ). Память данных (ОЗУ). Регистры специальных функций. Регистр флагов (PSW).

# 8. Таймеры / счетчики микроконтроллеров семейства 8051. Регистр режима работы таймера/счетчика TMOD. Регистр управления/статуса таймера TCON. Режимы работы таймеров-счетчиков.

# 9. Режимы работы и регистры последовательного интерфейса. Регистр управления/статуса приемопередатчика SCON. Функциональное назначение бит регистра управления/статуса приемопередатчика SCON. Скорость приема/передачи информации через последовательный порт. Работа в сети.

# 10. Организация системы прерываний. Регистры масок и приоритетов прерываний, флаги прерываний. Регистр масок прерывания (IE). Регистр приоритетов прерываний (IP). Выполнение подпрограммы прерывания.

# 11. Система команд микроконтроллера i8051. Форматы представления данных и команд, группы команд, способы адресации операндов. Обозначения, используемые при описании команд.

# 12. Классификация системы команд по функциональному признаку (пересылки данных, арифметических и логических операций, операций над битами, передачи управления), временные параметры исполнения команд.

# 13. Организация взаимодействия микроконтроллера с объектом управления. Опрос двоичного датчика. Ожидание события. Устранение дребезга контактов. Подсчет числа импульсов. Опрос группы двоичных датчиков.

# 14. Таймер счетчик Т/С2 микроконтроллера 8052. Регистр управление таймера/счетчика 2 T2COM. Режимы работы таймера/счетчика 2. Регистр режима таймера/счетчика 2 Т2МОD. Дополнительный регистр приоритетов прерываний IРН.

# 15. Аналого-цифровой преобразователь микроконтроллеров семейства MCS-51. ADCON - Регистр управления преобразователем. ADDAT - регистр результатов преобразования. DAPR - регистр программирования опорных напряжений АЦП. Синхронизация АЦП и время преобразования.

# 16. Расширения микропроцессоров семейства MCS-51/52. Дополнительные устройства и регистры.