



*Томский межвузовский центр
дистанционного образования*

Д.А. Рождественский

МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА В СИСТЕМАХ УПРАВЛЕНИЯ

Учебное пособие

ТОМСК – 2003

Министерство образования Российской Федерации
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра компьютерные системы в управлении
и проектировании (КСУП)**

Д.А. Рождественский

МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА В СИСТЕМАХ УПРАВЛЕНИЯ

Учебное пособие

2003

Корректор: Красовкая Е.Н.

Рождественский Д.А.

Микропроцессорные устройства в системах управления: Учебное пособие. –
Томск: Томский межвузовский центр дистанционного образования, 2003. –
130 с.

© Рождественский Д.А., 2003

© Томский межвузовский центр
дистанционного образования, 2003

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ В МИКРОПРОЦЕССОРНУЮ ТЕХНИКУ	7
1.1. Назначение и области применения микропроцессорных устройств	7
1.2. Представление информации в микропроцессорных системах..	8
1.2.1. Использование аналоговых и дискретных сигналов	9
1.2.2. Последовательный и параллельный способ представления информации	10
2. ОБРАБОТКА ИНФОРМАЦИИ	11
2.1. Процессор	11
2.1.1. Архитектура процессора	11
2.1.2. Определение микропроцессора. Принцип действия и внутреннее устройство микропроцессоров, назначение их составных частей	12
2.1.3. АЛУ	13
2.2. Память	14
2.2.1. Основные характеристики полупроводниковой памяти	14
2.2.2. Оперативные запоминающие устройства	15
2.2.3. Типы микросхем оперативных запоминающих устройств (ОЗУ)	16
2.2.4. Типы микросхем постоянных запоминающих устройств (ПЗУ)	16
2.2.5. Буферная память	17
2.2.6. Стековая память	19
2.3. Периферийные устройства	19
2.3.1. Понятие интерфейса	19
2.3.2. Последовательный способ передачи информации	20
2.3.3. Параллельный способ передачи информации	21
2.3.4. Режимы обмена	21
2.3.5. Форматы передачи данных	22
3. МИКРОПРОЦЕССОРЫ И МИКРОКОНТРОЛЛЕРЫ	26
3.1. Краткая история микропроцессоров	26
3.1.1. Основные характеристики микропроцессоров	28
3.1.2. История архитектур	28
3.1.3. Основные черты RISC-концепции	30
3.1.4. Основные черты CISC-концепции	30
4. СТРУКТУРА И НАЗНАЧЕНИЕ ОСНОВНЫХ БЛОКОВ ТИПОВОГО МИКРОКОНТРОЛЛЕРА	32
4.1. Вычислительный блок	34
4.2. Память программ	36

4.3. Память данных	37
4.4. Тактовый генератор	37
4.5. Сторожевой таймер	38
4.6. Порты ввода/вывода	39
4.7. Контроллер прерываний	40
4.8. Таймеры/счетчики	41
4.9. Аналоговый компаратор	44
4.10. Аналого-цифровой преобразователь	44
4.11. Параллельный порт.....	45
4.12. Режимы пониженного энергопотребления.....	48
5. КЛАССИФИКАЦИЯ МИКРОКОНТРОЛЛЕРОВ.....	50
5.1. Четырехразрядные микроконтроллеры	50
5.2. Восемьразрядные микроконтроллеры.....	51
5.3. 16- и 32- разрядные микроконтроллеры.....	52
5.4. Дискретный процессор обработки сигналов (DSP).....	54
5.5. Программное обеспечение микропроцессоров.....	56
6. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ И ОТЛАДКИ ДЛЯ МИКРОКОНТРОЛЛЕРОВ	58
6.1. Внутрисхемные эмуляторы	59
6.1.1. Отладчик	60
6.1.2. Эмуляционная память.....	61
6.1.3. Трассировщик.....	61
6.1.4. Процессор точек останова.....	61
6.1.5. Профилировщик	62
6.1.6. Интегрированная среда разработки	62
6.2. Симуляторы.....	63
6.3. Отладочные мониторы	64
6.4. Платы развития	65
6.5. Эмуляторы ПЗУ	67
6.5.2. Набор ПО для МК Atmel AVR.....	67
7. ВЫБОР МИКРОКОНТРОЛЛЕРА	69
7.1. Процесс выбора.....	69
7.2. Критерии выбора	70
7.3. Системные требования.....	70
7.4. Основные особенности микроконтроллера.....	71
7.5. Возможности микроконтроллера	72
7.6. Набор команд микроконтроллера	72
7.7. Прерывания микроконтроллера	73
7.8. Характеристика поставщика.....	73
7.9. Характеристика производителя.....	74

7.10. Подведение итогов выбора микроконтроллера	74
8. СОВРЕМЕННОЕ СОСТОЯНИЕ РЫНКА	
МИКРОКОНТРОЛЛЕРОВ	76
8.1. Краткий обзор микроконтроллеров	76
8.2. ATMEL AVR	76
8.3. MICROCHIP семейство PIC	78
8.4. Scenix SX	80
8.5. Ангстрем	82
8.6. MOTOROLA. 68HC05, 68HC08, 68HC11	83
8.7. Z8 ZiLOG	87
8.8. SAMSUNG. SAM86, SAM88	89
9. ШКОЛА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	92
9.1. Достоинства цифровой обработки сигналов в измерительных приборах	93
9.2. Методы цифровой обработки сигналов в измерительных приборах	94
9.3. Особенности процессоров цифровой обработки сигналов	95
10. ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ. УРОВНИ	
ПРЕДСТАВЛЕНИЯ МИКРОПРОЦЕССОРНОЙ	
СИСТЕМЫ	96
10.1. Ошибки, неисправности, дефекты	97
10.2. Отладка	99
10.3. Обнаружение ошибки и диагностика неисправности	100
10.4. Свойство контролепригодности системы	101
10.5. Функции средств отладки	101
10.6. Этапы проектирования микропроцессорных систем	102
10.7. Источники ошибок	103
10.9. Аппаратная отладка	104
10.10. Отладка программ	106
10.11. Комплексная отладка микропроцессорных систем	107
ЛИТЕРАТУРА	109
Основная литература	109
Дополнительная литература	109
Интернет ресурсы:	109
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСУ "МИКРОПРОЦЕССОРНЫЕ	
УСТРОЙСТВА В СИСТЕМАХ УПРАВЛЕНИЯ"	109

1. ВВЕДЕНИЕ В МИКРОПРОЦЕССОРНУЮ ТЕХНИКУ

1.1. Назначение и области применения микропроцессорных устройств

Развитие микроэлектронной технологии ведет к повышению уровня автоматизации и снижению стоимости систем управления. И в большой мере это связано с развитием микропроцессорной техники.

Замечательным свойством микропроцессорных систем является их высокая гибкость, возможность быстрой перенастройки при необходимости даже значительных изменений алгоритмов управления. Как правило, перенастройка осуществляется программным путем без существенных производственных затрат. Более того, микропроцессоры позволяют легко реализовать принципы открытых систем, функциональные возможности которых могут наращиваться по мере необходимости или по мере появления новых технических средств. Тем самым обеспечивается соответствие технического уровня микропроцессорных систем управления самым современным требованиям в течение длительного времени. Универсальность микропроцессорных систем обеспечивает их доступность широкому кругу потребителей.

Характерные особенности микропроцессорных информационно-управляющих систем, предназначенных для автоматизации технологических процессов:

- наличие ограниченного набора четко сформулированных задач;
- требования оптимизации структуры системы для конкретного применения;
- работа в реальном масштабе времени, т.е. обеспечение минимального времени реакции на изменение внешних условий;
- наличие развитой системы внешних устройств, их большое разнообразие;
- существенное различие функциональных задач;
- высокие требования по надежности с учетом большой продолжительности непрерывной работы;
- сложные условия эксплуатации;
- обеспечение автоматического режима работы или режима с участием оператора как элемента системы.

1.2. Представление информации в микропроцессорных системах

Любой процесс функционирования технического объекта или системы связан с передачей, приемом и переработкой информации.

Информация - совокупность фактов, явлений, событий, представляющих интерес, подлежащих регистрации и обработке. В рассматриваемом понятии всегда существуют два партнера: источник и потребитель информации. Как первым, так и вторым могут быть объекты науки, техники, общества и природы, животные, люди. Во взаимодействии между ними и рождается информация. В зависимости от области знаний различают научную, патентную информацию, техническую, коммерческую и другие виды.

Информация, представленная в виде, удобном для обработки, называется данными. Определенная структура информационного объекта, подвергаемого обработке, именуется форматом.

Существует несколько форм представления информации. Символьная основана на использовании символов - букв, цифр, знаков, в том числе знаков пунктуации и других знаков. Текстовая также использует образующие тексты символы, но расположенные в определенном порядке. Самой емкой и сложной является графическая форма. К ней относятся различные виды изображений. Наименьшей единицей количества информации в двоичной системе счисления является бит.

Информация, вложенная и зафиксированная в некоторой материальной форме, называется **сообщением**. Сообщения делятся на:

- непрерывные (аналоговые);
- дискретные (цифровые).

Непрерывные сообщения представляются некоторой физической величиной, например, электрическим током, напряжением, изменение которого во времени отображают протеканием рассматриваемого процесса, например, изменение температуры в технологическом процессе печати.

Физическая величина, передающая непрерывные сообщения, может принимать любые значения и изменяться в произвольные моменты времени в зависимости от процесса, который она отображает.

Аналоговый сигнал - сигнал, величина которого непрерывно изменяется во времени. Аналоговый сигнал обеспечивает передачу данных путем непрерывного изменения во времени амплитуды, частоты либо фазы. В соответствии с этим, он имеет бесконечное число значений. К аналоговым относятся и шумоподобные сигналы.

Для **дискретного** сообщения характерно существование набора элементов, из которых в некоторые моменты времени формируются различные последовательности. Важным является не физическая природа элементов, а то, что набор элементов конечен. В соответствии с этим, любое дискретное сообщение конечной длины, передает конечное число значений некоторой величины.

Дискретный сигнал - сигнал, имеющий конечное, обычно небольшое, число значений.

Практически всегда дискретный сигнал имеет два либо три значения. Нередко его называют также цифровым сигналом. В системах используются двоичные сигналы, имеющие значения (+), (-). Процесс изменения напряжения от низкого уровня (-) к высокому (+), называется фронтом сигнала (положительным перепадом, положительным фронтом), а обратный процесс - спадом (отрицательным перепадом, отрицательным фронтом). Если существенно их взаимное расположение, то фронт может быть передним и задним.

Вместе с этим, при передаче данных применяются и троичные сигналы со значениями: (+), (0), (-). Здесь «единица» представляется отсутствием потенциала в канале, тогда как «ноль» характеризуется положительным либо отрицательным импульсом. При этом полярность импульсов, представляющих «нули», должна чередоваться, т.е. за положительным (+) импульсом должен следовать отрицательный (-) и наоборот. В форме троичного сигнала осуществляется не только кодирование передаваемых данных, но и обеспечивается синхронизация работы канала, проверка его целостности.

Дискретные сигналы, по сравнению с аналоговыми сигналами, имеют ряд важных преимуществ. К ним, в первую очередь, относятся помехоустойчивость, легкость восстановления формы, простота аппаратуры передачи.

1.2.1. Использование аналоговых и дискретных сигналов

Аналоговые сигналы естественным образом передают речь, музыку и изображения. Для их использования в системах и сетях возможно с любой необходимой степенью точности заменять непрерывные сообщения на цифровые путем квантования непрерывного сообщения по уровню и времени. Таким образом, любое сообщение может быть представлено в цифровой форме.

Аналого-цифровое преобразование заключается в формировании последовательностей n -разрядных двоичных слов, представляющих с заданной точностью аналоговые сигналы. Для выполнения этого преоб-

разования вначале осуществляется квантование аналогового сигнала. В результате преобразования получается дискретный сигнал, используемый в системах и сетях. Наименьшее изменение аналогового сигнала, которое регистрируется устройством, осуществляющим преобразование, называется разрешением.

1.2.2. Последовательный и параллельный способ представления информации

Цифровая информация может быть представлена последовательным и параллельными кодами.

При **последовательном** коде каждый временной такт предназначен для отображения одного разряда кода слова.

Такт (clock tick) - промежуток времени, между последовательными сигналами синхронизации.

Величина такта выбирается такой, чтобы во время его прохождения в рассматриваемом объекте заканчивались все переходные процессы, вызванные изменением входных сигналов.

При **параллельном** коде все разряды кода слова представляются в одном временном такте, фиксируются отдельными элементами и проходят через отдельные линии, каждая из которых служит для представления и передачи только одного разряда.

При параллельной передаче информации код слова развертывается в пространстве, в отличие от последовательной, в которой развертывается во времени.

2. ОБРАБОТКА ИНФОРМАЦИИ

Аппаратные средства любой алгоритмически универсальной ЭВМ можно условно разделить на три основные части:

1. Процессор.
2. Память.
3. Периферийные устройства.

2.1. Процессор

Процессор служит для обработки информации - ее перемещения, осуществления арифметических и логических операций по командам, которые он считывает из памяти. Последовательность команд называется программой

Процессор – совокупность устройств управления арифметико-логического устройства и связанных с ними регистров и триггеров. Он организует процесс обработки информации путем выборки и последовательного выполнения команд программы, находящейся в памяти ЭВМ.

2.1.1. Архитектура процессора

По работе с командами различают два типа архитектур процессоров:

1. Фон-неймановская архитектура.

Джон фон-Нейман, математик венгерского происхождения, предложил простейшую архитектуру, а именно – с объединенной памятью программного продукта и данных. С тех пор это простое решение стало стандартом. Машина фон-Неймана была создана в Принстонском институте новейших исследований в 1951 году.

Машины фон-Неймана хранят программу и данные в одной и той же области памяти. В машинах этого типа команды содержат указание, что выполнить, и адрес данных, подлежащих обработке.

2. Гарвардская архитектура.

Единственное отличие гарвардской архитектуры состоит в том, что память программ и память данных разделены, и они используют физически разделенные линии передачи. Это позволяет подобной машине пересылать команды и данные одновременно. Следовательно, такая конструкция может значительно увеличить производительность.

Гарвардская архитектура была разработана Говардом Айхеном (Howard Aiken) в конце 1930-х годов в Гарвардском университете. Первая машина Harvard Mark1 работала уже в 1944 году. За ней в 1946 году последовал «электронный числовой интегратор и калькулятор» (Electronic Numerical Integrator and Calculator - ENIAC), разработанный Пенсильванским университетом.

Персональные компьютеры общего назначения используют процессоры, построенные согласно архитектуре фон-Неймана. Другие семейства компьютеров также используют архитектуру фон-Неймана. Гарвардская архитектура является наиболее подходящей для специализированных микропроцессоров, предназначенных для решения прикладных задач в реальном времени.

2.1.2. Определение микропроцессора. Принцип действия и внутреннее устройство микропроцессоров, назначение их составных частей

Микропроцессор (МП) - это программно-управляемое электронное цифровое устройство, предназначенное для обработки цифровой информации и управления процессом этой обработки, выполненное на одной или нескольких интегральных схемах с высокой степенью интеграции электронных элементов.

Микропроцессор служит для обработки информации - осуществления арифметических и логических операций. Его работа состоит в считывания из памяти кодов команд, их расшифровки и выполнения.

Микропроцессор включает в себя:

- арифметическо-логическое устройство (АЛУ), которое служит для осуществления собственно арифметических и логических операций;
- регистры общего назначения (РОН), которые используются для хранения информации - сверхоперативного запоминающего устройства;
- 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 дешифрования кодов команд.

2.1.3. АЛУ

Арифметико-логическое Устройство (АЛУ) - часть процессора, выполняющая арифметические и логические операции над данными. АЛУ реализует важную часть процесса обработки данных. Она заключается в выполнении набора простых 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.). АЛУ состоит из регистров, сумматора с соответствующими логическими схемами и элемента управления выполняемым процессом. Устройство работает в соответствии с сообщаемыми ему именами (кодами) операций, которые при пересылке данных нужно выполнить над переменными, помещаемыми в регистры. 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.

К числу основных операций, выполняемых в АЛУ, относятся:

- передача операнда на регистр (прием и выдача);
- сдвиг операнда на заданное число разрядов (арифметический, логический, циклический);
- прибавление к слову или вычитание от слова единицы;
- сравнение операндов;
- поразрядные логические операции;
- суммирование двух операндов;
- преобразование кодов операндов, включая инверсию, дополнение, дешифрование и др.;
- операции над числами с плавающей запятой.

2.2. Память

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;

- **ПЗУ** - постоянное запоминающее устройство, предназначенное для хранения программ, поэтому часто эту память называют кодовой или памятью программ.

2.2.1. Основные характеристики полупроводниковой памяти

Полупроводниковая память имеет большое число характеристик и параметров, которые необходимо учитывать при проектировании систем:

1. Емкость памяти определяется числом бит хранимой информации. Емкость кристалла обычно выражается также в битах. Важной характеристикой кристалла является информационная организация кристалла памяти $M \times N$, где M - число слов, N - разрядность слова. Например, кристалл емкостью 16 Кбит может иметь различную организацию: 16 Кх1, 4 Кх2 Кх8. При одинаковом времени обращения память с большей шириной выборки обладает большей информационной емкостью.

2. Временные характеристики памяти.

Время доступа - временной интервал, определяемый от момента, когда центральный процессор выставил на шину адреса адрес требуемой ячейки памяти и послал по шине управления приказ на чтение или запись данных, до момента осуществления связи адресуемой ячейки с шиной данных.

Время восстановления - это время, необходимое для приведения памяти в исходное состояние после того, как ЦП снял с ША адрес, с ШУ сигнал «чтение» или «запись» и с ШД данные.

3. Удельная стоимость запоминающего устройства определяется отношением его стоимости к информационной емкости, т.е. определяется стоимостью бита хранимой информации.

4. Потребляемая энергия (или рассеиваемая мощность) приводится для двух режимов работы кристалла: режима пассивного хранения ин-

формации и активного режима, когда операции записи и считывания выполняются с номинальным быстродействием. Кристаллы динамической МОП-памяти в резервном режиме потребляют примерно в десять раз меньше энергии, чем в активном режиме. Наибольшее потребление энергии, не зависящее от режима работы, характерно для кристаллов биполярной памяти.

5. Плотность упаковки определяется площадью запоминающего элемента и зависит от числа транзисторов в схеме элемента и используемой технологии. Наибольшая плотность упаковки достигнута в кристаллах динамической МОП-памяти.

6. Допустимая температура окружающей среды обычно указывается отдельно для активной работы, для пассивного хранения информации и для нерабочего состояния с отключенным питанием. Указывается тип корпуса, если он стандартный, или чертеж корпуса с указанием всех размеров, маркировкой и нумерацией контактов, если корпус новый. Приводятся также условия эксплуатации: рабочее положение, механические воздействия, допустимая влажность и другие

2.2.2. Оперативные запоминающие устройства

Основным запоминающим устройством памяти, предназначенным для хранения всей информации непосредственно участвующей в вычислительном процессе, является оперативное запоминающее устройство (ОЗУ).

В современных микропроцессорах память ОЗУ представляет собой сложную многоуровневую систему. В этой системе обычно выделяют уровни сверхоперативной памяти (СОЗУ), ОЗУ, буферной памяти (БЗУ) и внешней памяти (ВЗУ).

Каждый последующий уровень отличается от предыдущего важнейшими техническими характеристиками памяти:

- емкостью;
- быстродействием.

Емкостью называется максимальное количество информации, которая может быть записана в память.

Быстродействие характеризуется длительностью операций чтения и записи – двух основных операций, выполняемых памятью ЭВМ.

Для указанных уровней памяти емкость растет в направлении от СОЗУ к ВЗУ, а быстродействие в противоположном направлении.

2.2.3. Типы микросхем оперативных запоминающих устройств (ОЗУ)

В микросхемы оперативных запоминающих устройств (ОЗУ) информация может быть записана неограниченное число раз, однако она исчезает с отключением питания.

Существует два типа микросхем ОЗУ:

- статические ОЗУ, в которых основой запоминающей ячейки служит триггер;
- динамические ОЗУ, в них основой запоминающих ячеек является конденсатор; в качестве конденсатора используется затвор полевого транзистора.

Ячейка динамического ОЗУ проще, поэтому ОЗУ этого типа дешевле и имеют большую емкость при том же количестве компонентов, однако они требуют периодической подзарядки всех запоминающих конденсаторов. Этот процесс называется регенерацией.

Типичное значение периода регенерации - миллисекунды; регенерация осуществляется при каждой операции чтения или записи. Также в динамических ОЗУ используется мультиплексированная адресная шина - адрес передается за два цикла, сначала одна половина разрядов (строки), потом другая (столбцы), для регенерации достаточно перебрать все номера строк.

Основными направлениями совершенствования ОЗУ является разработка:

- квазистатических ОЗУ - динамических «внутри», но со встроенной автономной схемой регенерации;
- энергонезависимых ОЗУ, хотя бы и в течение ограниченного периода времени. Одним из путей решения этой проблемы является использование микромощных статических ОЗУ со встроенным источником электропитания.

2.2.4. Типы микросхем постоянных запоминающих устройств (ПЗУ)

Микросхемы постоянных запоминающих устройств (ПЗУ) способны сохранять информацию при отключенном электропитании, но могут быть запрограммированы только один или очень ограниченное число раз.

Существуют следующие основные типы ПЗУ:

- масочные ПЗУ - они программируются в процессе их изготовления путем нанесения маски из замкнутых (высокий уровень) и разомкнутых перемычек (низкий уровень), этот тип ПЗУ наиболее дешев, но при изготовлении крупной партией;

- ПЗУ с плавкими перемычками или электрически программируемые (ЭПЗУ) - эти микросхемы программируются потребителем путем пропускания импульсов тока до разрушения перемычек, соответствующих битам, которые должны стать нулевыми;

- перепрограммируемые ПЗУ с электрической записью информации и стиранием ультрафиолетовым излучением (УФППЗУ) - основная ячейка памяти микросхемы данного типа – МОП-транзистор с полностью изолированным «плавающим» затвором, при программировании окисел пробивается и на затворе накапливается заряд, который сохраняется там пока микросхема не будет подвергнута УФ-облучению, под его действием окисел становится проводящим; сопротивление канала транзистора зависит от заряда на затворе и будет определять бит, записанный в ячейку;

- электрически стираемые ПЗУ устроены аналогично УФППЗУ, но стирание происходит, как и запись, при подаче импульсов напряжения; это самый дорогой, но и самый удобный тип ПЗУ.

В настоящее время все более популярной становится новая технология реализации ПЗУ - FLASH-память. Ее главное достоинство в том, что она построена по принципу электрической перепрограммируемости, т. е. допускает многократное стирание и запись информации с помощью программаторов. Минимальное гарантированное число циклов записи/стирания обычно превышает несколько тысяч. Это существенно увеличивает жизненный цикл и повышает гибкость микропроцессорных систем, так как позволяет вносить изменения в программу микропроцессора, как на этапе разработки системы, так и в процессе его работы в реальном устройстве.

2.2.5. Буферная память

В вычислительных системах используются подсистемы с различным быстродействием и, в частности, с различной скоростью передачи данных (рис. 2.1). Обычно обмен данными между такими подсистемами реализуется с использованием прерываний или канала прямого доступа к памяти. В первую очередь подсистема 1 формирует запрос на обслуживание по мере готовности данных к обмену. Однако обслуживание прерываний связано с непроизводительными потерями времени и при

пакетном обмене производительность подсистемы 2 заметно уменьшается. При обмене данными с использованием канала прямого доступа к памяти подсистема 1 передает данные в память подсистемы 2. Данный способ обмена достаточно эффективен с точки зрения быстродействия, но для его реализации необходим довольно сложный контроллер прямого доступа к памяти.



Рисунок 2.1. Применение буферной памяти

Наиболее эффективно обмен данными между подсистемами с различным быстродействием реализуется при наличии между ними специальной буферной памяти. Данные от подсистемы 1 временно запоминаются в буферной памяти до готовности подсистемы 2 принять их. Емкость буферной памяти должна быть достаточной для хранения тех блоков данных, которые подсистема 1 формирует между считываниями их подсистемой 2. Отличительной особенностью буферной памяти является запись данных с быстродействием и под управлением подсистемы 1, а считывание - с быстродействием и под управлением подсистемы 2 («эластичная память»). В общем случае память должна выполнять операции записи и считывания совершенно независимо и даже одновременно, что устраняет необходимость синхронизации подсистем. Буферная память должна сохранять порядок поступления данных от подсистемы 1, т.е. работать по принципу «первое записанное слово считывается первым» (First Input First Output - FIFO). Таким образом, под буферной памятью типа FIFO понимается устройство памяти, которое автоматически следит за порядком поступления данных и выдает их в том же порядке, допуская выполнение независимых и одновременных операций записи и считывания.

Типовой пример применения буферной памяти в микропроцессорах – последовательный приемо-передатчик.

2.2.6. Стековая память

Стековой называют память, доступ к которой организован по принципу: «последним записан - первым считан» (Last Input First Output - LIFO). Использование принципа доступа к памяти на основе механизма LIFO началось с больших ЭВМ. Применение стековой памяти оказалось очень эффективным при построении компилирующих и интерпретирующих программ, при вычислении арифметических выражений с использованием польской инверсной записи. В микропроцессорах она стала широко использоваться в связи с удобствами реализации процедур вызова подпрограмм и при обработке прерываний.

Аппаратный стек представляет собой совокупность регистров, связи между которыми организованы таким образом, что при записи и считывании данных содержимое стека автоматически сдвигается. Основное достоинство аппаратного стека - высокое быстродействие, а недостаток - ограниченная емкость.

Наиболее распространенным в настоящее время и, возможно, лучшим вариантом организации стека в ЭВМ является использование области памяти. Для адресации стека используется указатель стека, который предварительно загружается в регистр и определяет адрес последней занятой ячейки. В некоторых МП содержимое основных регистров запоминается в стеке автоматически при прерывании программ.

2.3. 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).

Взаимодействие микропроцессора с внешними устройствами происходит с помощью интерфейса.

2.3.1. Понятие интерфейса

Объединение модулей микропроцессорного устройства в единую систему производится посредством единой системы сопряжения, называемой интерфейсом - (от английского interface - сопрягать, согласовывать).

Интерфейс - определенная стандартами граница между взаимодействующими объектами.

Задачей интерфейса является определение параметров, процедур и характеристик взаимодействия любых партнеров. Ими могут быть пользователи, сети, системы, уровни, функциональные блоки, устройства, программы и т.д.

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).

В зависимости от способа передачи данных различают два вида интерфейса: последовательный и параллельный.

2.3.2. Последовательный способ передачи информации

Последовательный интерфейс для передачи данных использует одну сигнальную линию, по которой информационные биты передаются друг за другом последовательно. Отсюда - название интерфейса и порта. Английские термины - *SerialInterface* и *Serial Port* (иногда их неправильно переводят как «серийные»). Последовательная передача позволяет сократить количество сигнальных линий и увеличить дальность связи. В ряде последовательных интерфейсов применяется гальваническая развязка внешних (обычно входных) сигналов от схемной земли устройства, что позволяет соединять устройства, находящиеся под разными потенциалами.

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 (СОМ-порты в IBM PC совместимых компьютерах).

2.3.3. Параллельный способ передачи информации

В *параллельном* интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно. В PC традиционно используется параллельный интерфейс *Centronics*, реализуемый LPT-портами. Параллельные интерфейсы используют логические уровни ТТЛ (транзисторно-транзисторной логики), что ограничивает длину кабеля из-за невысокой помехозащищенности ТТЛ-интерфейса. Гальваническая развязка отсутствует. Параллельный способ применяют в тех случаях, когда необходимо получить наивысшую пропускную способность канала передачи информации (например, все внутрисистемные интерфейсы микропроцессорных систем являются параллельными). Параллельные интерфейсы используют для подключения принтеров. Передача данных может быть как однонаправленной (*Centronics*), так и двунаправленной (*Bitronics*).

2.3.4. Режимы обмена

Для 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).

2.3.5. Форматы передачи данных

Рассмотрим некоторые общие вопросы, связанные с обменом данными между внешними устройствами и микропроцессором. Как уже говорилось, существуют два способа передачи слов информации по линиям данных: параллельный, когда одновременно пересылаются все биты слова, и последовательный, когда биты слова пересылаются поочередно, начиная, например, с его младшего разряда.

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.

При синхронной последовательной передаче, 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.

Синхронная последовательная передача начинается с пересылки в приемник одного или двух символов синхронизации (не путать с импульсами синхронизации). Получив такой символ (символы), приемник начинает прием данных и их преобразование в параллельный формат. Естественно, что при такой организации синхронной последовательной передачи она целесообразна лишь для пересылки массивов слов, а не отдельных символов. Это обстоятельство, а также необходимость использования для обмена сравнительно дорогих (четыре проводных или кабельных) линий связи помешало широкому распространению синхронной последовательности передачи данных.

Асинхронная последовательная передача данных означает, что у передатчика и приемника нет общего генератора синхроимпульсов и что

синхронизирующий сигнал не посылается вместе с данными. Как же в таком случае приемник будет узнавать о моментах начала и завершения передачи бит данных? Опишем простую процедуру, которую можно использовать, если передатчик и приемник асинхронной последовательной передачи данных согласованы по формату и скорости передачи.

Стандартный формат асинхронной последовательной передачи данных, используемый в микропроцессорах, 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. Когда передатчик бездействует (данные не посылаются на линию), на линии сохраняется уровень сигнала, соответствующий логической 1.



Рисунок 2.2. Формат асинхронной последовательной передачи данных

Передатчик может начать пересылку символа в любой момент времени посредством генерирования стартового бита, т.е. перевода линии в состояние логического 0 на время, точно равное времени передачи 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). При единичном бите контроля стоповый бит не изменяет состояния сигнала на линии. 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.

Для обеспечения оптимальной защищенности сигнала от искажения, шумов и разброса частоты синхроимпульсов приемник должен считывать принимаемый бит в середине его длительности. Рассмотрим работу приемника с того момента, когда он закончил прием символа данных и перешел в режим обнаружения стартового бита следующего слова.

Если линия перешла в состояние логического нуля и находится в этом состоянии в течение времени, не меньшего половины временного интервала передачи бита, то приемник переводится в режим считывания бит информации. В противном случае приемник остается в режиме обнаружения, так как вероятнее всего это был не стартовый бит, а шумовая помеха. В новом режиме приемник вырабатывает сигналы считывания через интервалы, равные времени передачи бита, т. е. выполняет считывание и сохранение принимаемых бит примерно на середине их передачи. Аналогичным образом будут считаны бит контроля четности и сигнал логической единицы (стоповый бит). Если оказалось, что на месте стопового бита обнаружен сигнал логического нуля, то произошла «Ошибка кадра» и символ принят неправильно. Иначе проверяется, четно ли общее число единиц в информационных битах и бите контроля, и если оно четно, производится запись принятого символа в буфер приемника.

Передний фронт стартового бита сигнализирует о начале поступления передаваемой информации, а момент его появления служит точкой отсчета времени для считывания бит данных. Стоповый бит предоставляет время для записи принятого символа в буфер приемника и обеспечивает возможность выявления ошибки кадра. Наиболее часто ошибки кадра появляются тогда, когда приемник ошибочно синхронизирован с битом 0, который в действительности не является стартовым битом. Если передатчик бездействует (посылает сигнал логической единицы) в течение одного кадра или более, то всегда можно восстановить правильную синхронизацию. Хуже обстоит дело при рассинхронизации генераторов передатчика и приемника, когда временной интервал между сиг-

налами считывания принимаемых битов будет меньше или больше времени передачи бита.

Например, если при считывании битов посылки, показанной на рис. 2.2,б, временной интервал между сигналами считывания станет на 6% меньше, чем время передачи бита, то восьмой и девятый сигналы считывания будут выработаны тогда, когда на линии находится бит контроля четности (рис. 2.3). Следовательно, не будет обнаружен стоповый бит и будет зафиксирована ошибка кадра, несмотря на правильность принятой информации. Однако при 18%-й рассинхронизации генераторов, когда вместо кода (01110001) приемник зафиксирует код (11100001), никаких ошибок не будет обнаружено - четность соблюдена и стоповый (девятый по порядку) бит равен 1 (см. рис. 2.3).

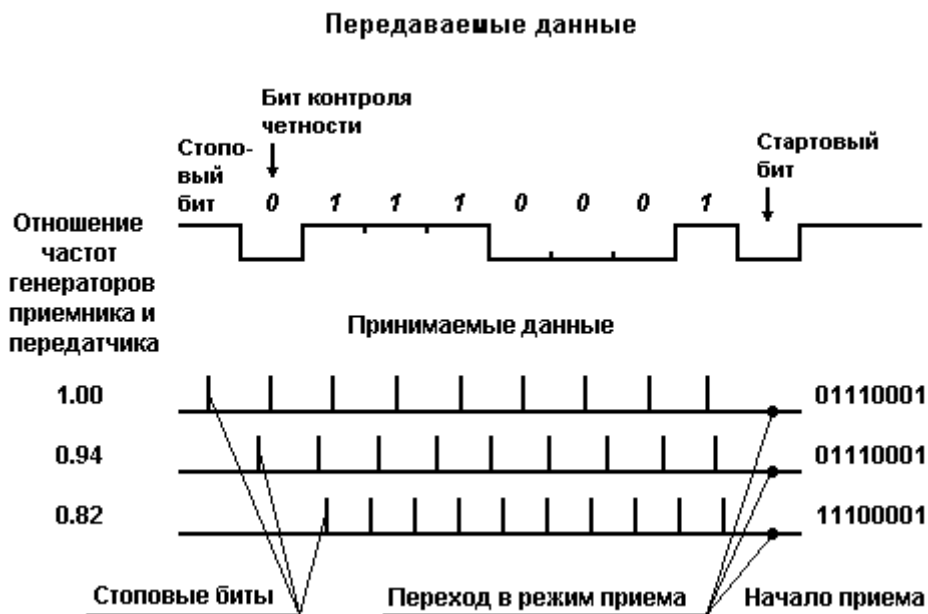


Рисунок 2.3. Ошибка из-за рассинхронизации генераторов передатчика и приемника

3. МИКРОПРОЦЕССОРЫ И МИКРОКОНТРОЛЛЕРЫ

Микропроцессор, как и любой другой процессор, является устройством, предназначенным для обработки данных или передачи данных. Он часто не имеет памяти, средств ввода/вывода данных. Эти задачи решаются внешними (по отношению к микропроцессору) интегральными схемами. Размеры слов, с которыми работают микропроцессоры все время растут. Все больше используются 32- и 64-разрядные микропроцессоры. Последние позволяют резко увеличивать адресуемую память и размер файлов, с которыми работают. Это, в свою очередь, увеличивает быстродействие микропроцессоров.

Создание микропроцессоров позволяет уменьшить стоимость и размеры технических средств обработки информации, увеличить их быстродействие, снизить энергопотребление.

3.1. Краткая история микропроцессоров

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

15 ноября 1971 г. можно считать началом новой эры в электронике. В этот день компания приступила к поставкам первого в мире микропроцессора Intel-4004.

Это был настоящий прорыв, ибо МП Intel-4004 размером менее 3 см был производительнее гигантской машины ENIAC. Правда работал он гораздо медленнее и мог обрабатывать одновременно только 4 бита информации (процессоры больших ЭВМ обрабатывали 16 или 32 бита одновременно), но и стоил первый МП в десятки тысяч раз дешевле. Кристалл представлял собой 4-разрядный процессор с классической архитектурой ЭВМ гарвардского типа и изготавливался по передовой р-канальной МОП-технологии с проектными нормами 10 мкм. Электрическая схема прибора насчитывала 2300 транзисторов. МП работал на тактовой частоте 750 кГц при длительности цикла команд 10,8 мкс. Чип i4004 имел адресный стек (счетчик команд и три регистра стека типа LIFO), блок РОНов (регистры сверхоперативной памяти или регистровый файл - РФ), 4-разрядное параллельное АЛУ, аккумулятор, регистр команд с дешифратором команд и схемой управления, а также схему связи с внешними устройствами. Все эти функциональные узлы объединялись между собой 4-разрядной ШД. Память команд достигала 4 Кбайт

(для сравнения: объем ЗУ мини-ЭВМ в начале 70-х годов редко превышал 16 Кбайт), а РФ ЦП насчитывал 16 4-разрядных регистров, которые можно было использовать и как 8 8-разрядных.

Такая организация РОНов сохранена и в последующих МП фирмы Intel. Три регистра стека обеспечивали три уровня вложения подпрограмм. МП i4004 монтировался в пластмассовый или металлокерамический корпус типа DIP (Dual In-line Package) всего с 16 выводами. В систему его команд входило всего 46 инструкций. Вместе с тем, кристалл располагал весьма ограниченными средствами ввода/вывода, а в системе команд отсутствовали операции логической обработки данных (И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ), в связи с чем их приходилось реализовывать с помощью специальных подпрограмм. Модуль i4004 не имел возможности останова (команды HALT) и обработки прерываний.

Цикл команды процессора состоял из 8 тактов задающего генератора. Была мультиплексированная шина адреса/шина данных, адрес 12-разрядный передавался по 4-разряда.

1 апреля 1972 г. фирма Intel начала поставки первого в отрасли 8-разрядного прибора i8008. Кристалл изготавливался по р-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 транзисторов. Процессор работал на частоте 500 кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

Сейчас микропроцессоры производятся большим числом фирм. На их основе создается подавляющее число компьютеров. Некоторые микропроцессоры, могут быть дополнены сопроцессорами, расширяющими возможности первых и набор выполняемых команд. Микропроцессор является основным компонентом микрокомпьютера.

В тех случаях, когда память и средства ввода/вывода размещаются на той же подложке интегральной схемы, что и микропроцессор, последний превращается в микрокомпьютер. Микроконтроллер является специализированным микрокомпьютером, применяемым для управления различными устройствами. Например, принтерами, терминалами, аппаратами передачи данных.

Мы стремимся делать различия между микроконтроллерами и микропроцессорами, хотя не существует точного определения различий между ними. На первый взгляд можно найти микроконтроллеры, связанные со сферой господства встроенных устройств, и микропроцессоры, активно применяемые в области настольных компьютеров. Более подробный анализ позволяет определить микроконтроллеры как устройства, имеющие память RAM или ROM вместо кэш-памяти, присутствующей обычно в большинстве периферийных устройств. В противопо-

ложность микроконтроллерам, микропроцессоры имеют устройство управления памятью и большой объем кэш-памяти. Иногда разница определяется производительностью или разрядностью. Например, 8- и 16-разрядные устройства обычно называют микроконтроллерами.

3.1.1. Основные характеристики микропроцессоров

Микропроцессор характеризуется:

1) тактовой частотой, определяющей максимальное время выполнения переключения элементов в ЭВМ;

2) разрядностью, т.е. максимальным числом одновременно обрабатываемых двоичных разрядов.

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 $m/n/k=16/8/20$;

3) архитектурой.

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.

3.1.2. История архитектур

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

Ранее других появились процессоры CISC. Они имеют классическую архитектуру, характеризуемую большим набором команд. Благодаря этому процессоры выполняют самые разнообразные задачи обработки данных.

Вместе с этим, стало необходимо повысить скорость работы процессоров. Одним из путей к этому стал процессор RISC, который характеризуется сокращенным набором быстро выполняемых команд. Так, в процессоре CISC для выполнения одной команды 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 так-

те. 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.

В настоящее время наиболее широко представлены RISC и CISC архитектуры.

Термин **CISC** означает сложную систему команд и является аббревиатурой английского определения Complex Instruction Set Computer. Аналогично, термин **RISC** означает сокращенную систему команд и происходит от английского Reduced Instruction Set Computer. Однако, несмотря на широкую распространенность этих понятий, необходимо признать, что сами названия не отражают главного различия между системами команд **CISC** и **RISC**. Основная идея **RISC**-архитектуры - это тщательный подбор таких комбинаций кодов операций, которые можно было бы выполнить за один такт тактового генератора. Основным выигрыш от такого подхода - резкое упрощение аппаратной реализации ЦП и возможность значительно повысить его производительность.

Первоначально реализовывать такой подход удавалось, лишь существенно сократив набор команд, отсюда и родилось название **RISC**. Очевидно, что в общем случае одной команде **CISC**-архитектуры должны соответствовать несколько команд **RISC**-архитектуры. 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**. Даже если для каждой инструкции потребуются выполнить три инструкции **RISC**-контроллера, то в итоге **RISC**-архитектура обеспечит четырехкратное увеличение производительности.

Попутно **RISC**-архитектура позволяет решить еще ряд задач. Ведь с упрощением ЦП уменьшается число транзисторов, необходимых для его реализации, следовательно, уменьшается площадь кристалла. А с этим связано снижение стоимости и потребляемой мощности.

Однако в настоящее время грань между этими двумя понятиями стремительно стирается. Например, МК семейства AVR фирмы Atmel имеют систему команд из 120 инструкций, что соответствует типу **CISC**.

Однако большинство из них выполняется за один такт, что является признаком **RISC**-архитектуры. Сегодня принято считать, что основным признаком **RISC**-архитектуры является выполнение команд за один такт тактового генератора. Число команд само по себе значения уже не имеет.

3.1.3. Основные черты RISC-концепции.

Основными чертами RISC-концепции являются:

- 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. В настоящее время именно RISC-процессоры наиболее распространены. Область их применения очень широка - от микроконтроллеров до суперкомпьютеров. RISC-процессоры лидируют по производительности среди процессоров общего назначения. 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.

3.1.4. Основные черты CISC-концепции

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 орга-

низации 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, Motorola, Siemens.

4. СТРУКТУРА И НАЗНАЧЕНИЕ ОСНОВНЫХ БЛОКОВ ТИПОВОГО МИКРОКОНТРОЛЛЕРА

Любой типовой микроконтроллер состоит из процессора гарвардской или фон-неймановской архитектуры, памяти программ, памяти данных, портов ввода/вывода, периферийных устройств и интерфейсных схем.

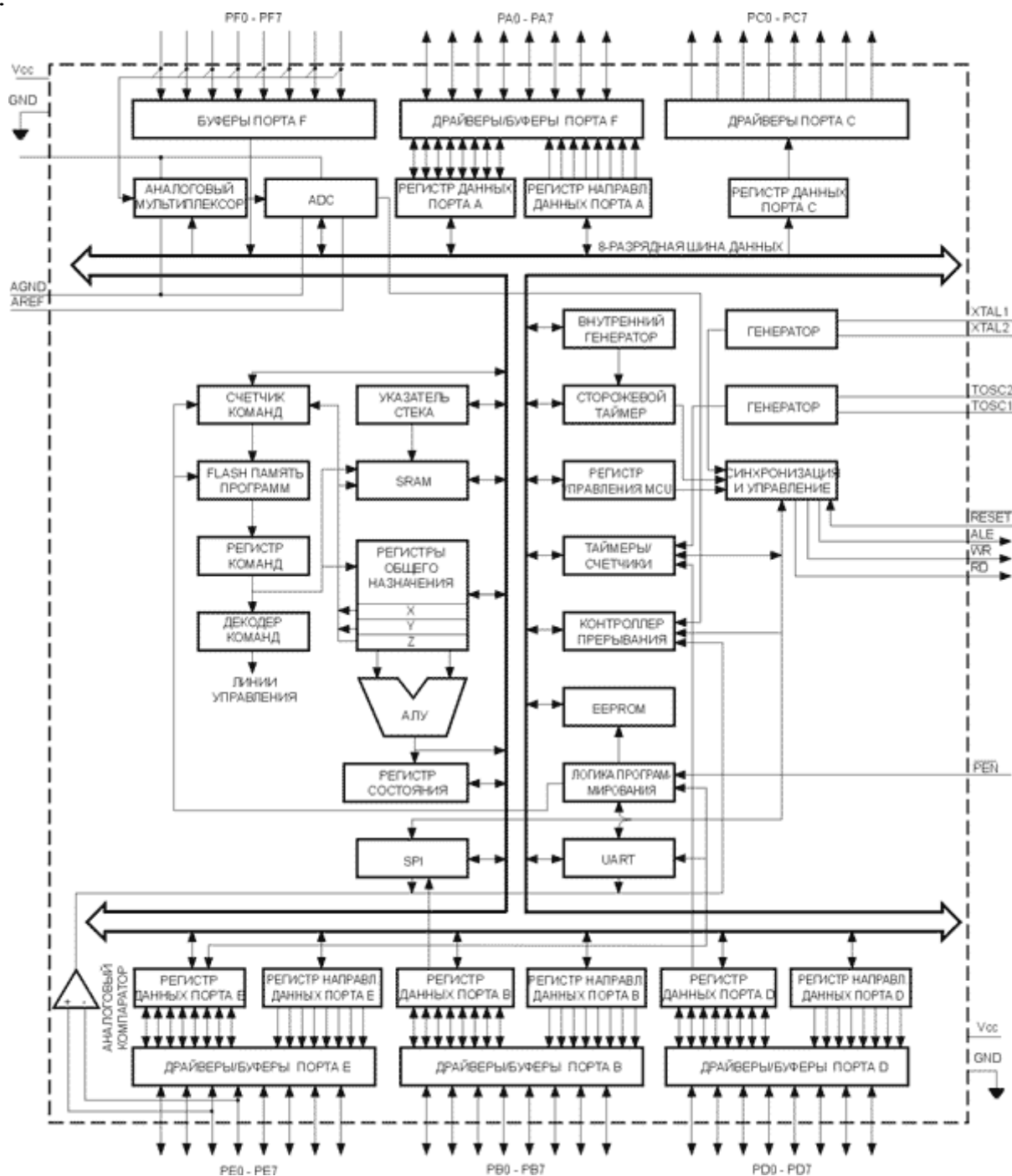


Рисунок 3.1 Структурная схема типового микроконтроллера

Обычно, когда говорят о каком-либо МК, то всегда упоминают **семейство**, к которому он принадлежит. К одному семейству относят изделия, имеющие одинаковое **ядро**, под которым понимают совокупность таких понятий, как система команд, циклограмма работы ЦП, организация памяти программ и памяти данных, система прерываний и базовый набор периферийных устройств. Отличия между различными представителями одного семейства заключаются, в основном, в составе периферийных устройств и объеме памяти программ или данных. Поскольку диапазон задач, решаемых МК, чрезвычайно широк, их производители стараются выпустить столько модификаций, чтобы удовлетворить самые разнообразные запросы потребителей. Во многих семействах число модификаций приближается к сотне или даже превышает это значение.

Наиболее важная особенность семейства - программная совместимость на уровне двоичного кода всех входящих него МК. Это позволяет разработчикам систем заменять одни МК семейства другими без потери наработок своего программного обеспечения. Естественно, чем большее число разновидностей входит в семейство, тем больше шансов выбрать оптимальный вариант, тем привлекательнее это семейство для разработчика. Вопрос правильного выбора семейства МК для новой разработки является стратегическим, так как проблема переноса программного обеспечения между изделиями разных семейств чрезвычайно сложна, и даже использование языков высокого уровня не всегда позволяет решить ее без больших потерь.

В данной главе рассматриваются характеристики и принципы работы основных блоков типового микроконтроллера.

В начале необходимо познакомиться с основными понятиями и сокращениями микропроцессорной техники:

- Flash ROM – тип энергонезависимой памяти программ МК;
- EEPROM - тип энергонезависимой памяти данных объем;
- RAM - объем статической памяти данных;
- External RAM - возможность подключения к микроконтроллеру дополнительной микросхемы внешней статической памяти данных (в килобайтах);
- ISP - возможность внутрисхемного программирования микроконтроллера, программирование осуществляется в системе (на целевой плате) при основном напряжении питания;
- SPM - функция самопрограммирования Flash ROM памяти микроконтроллера в системе без участия внешнего программатора;
- JTAG – стандартный интерфейс для программирования МК;
- I/O -линии ввода / вывода;

- USI - универсальный коммуникационный интерфейс;
- POR (Power-On Reset) – начальный сброс МК. Задача – удерживать сигнал сброса в активном состоянии пока не стабилизируется питание и кварцевый резонатор;
- BOD (BrownOut Detector) – отслеживает провалы питающего напряжения. В МК обычно с регулируемым порогом (2,7В, 4В);
- WDT (Watch Dog Timer) – сторожевой таймер для предотвращения «зависания» программы микроконтроллера;
- BDC - аппаратный программируемый блок защиты от сбоев при внезапном (в том числе и кратковременном) пропадании напряжения питания микроконтроллера;
- UART - асинхронный последовательный приемопередатчик;
- SPI - синхронный трехпроводной последовательный интерфейс;
- I2C - двухпроводной последовательный интерфейс;
- RTC - система реального времени;
- PWM - широтно - импульсный модулятор.

4.1. Вычислительный блок

Вычислительный блок является, пожалуй, самым отличительным блоком в МК. Он определяет концепцию построения и принцип работы с памятью. Рассмотрим как организован вычислительный блок у семейства AVR микроконтроллеров фирмы Atmel, являющейся несомненным лидером на рынке МК.

Гарвардская архитектура AVR реализует полное логическое и физическое разделение не только адресных пространств, но и информационных шин для обращения к памяти программ и к памяти данных, причем способы адресации и доступа к этим массивам памяти также различны. Подобное построение уже ближе к структуре цифровых сигнальных процессоров и обеспечивает существенное повышение производительности. Центральный процессор работает одновременно как с памятью программ, так и с памятью данных; разрядность шины памяти программ расширена до 16 бит.

Следующим шагом на пути увеличения быстродействия AVR является использование технологии конвейеризации, вследствие чего цикл «выборка - исполнение» команды заметно сокращен. Например, у микроконтроллеров семейства MCS51 короткая команда выполняется за 12 тактов генератора (1 машинный цикл), в течение которых процессор последовательно считывает код операции и исполняет ее. В PIC-контрол-

лерах фирмы Microchip, где уже реализован конвейер, короткая команда выполняется в течение 8 периодов тактовой частоты (2 машинных цикла). За это время последовательно дешифрируется и считывается код операции, выполняется команда, фиксируется результат и одновременно считывается код следующей операции (одноуровневый конвейер). Поэтому в общем потоке команд одна короткая команда реализуется за 4 периода тактовой частоты или за один машинный цикл. В микроконтроллерах AVR тоже используется одноуровневый конвейер при обращении к памяти программ и короткая команда в общем потоке выполняется, как и в PIC-контроллерах, за один машинный цикл. Главное же отличие состоит в том, что этот цикл у AVR составляет всего один период тактовой частоты.

Следующая отличительная черта архитектуры микроконтроллеров AVR - регистровый файл быстрого доступа. Каждый из 32-х регистров общего назначения длиной 1 байт непосредственно связан с арифметико-логическим устройством (ALU) процессора. Другими словами, в AVR существует 32 регистра-аккумулятора (сравните, например, с MCS51). Это обстоятельство позволяет в сочетании с конвейерной обработкой выполнять одну операцию в ALU за один машинный цикл. Так, два операнда извлекаются из регистрового файла, выполняется команда и результат записывается обратно в регистровый файл в течение только одного машинного цикла.

Шесть из 32-х регистров файла могут использоваться как три 16-разрядных указателя адреса при косвенной адресации данных. Один из этих указателей (Z Pointer) применяется также для доступа к данным, записанным в памяти программ микроконтроллера. Использование трех 16-битных указателей (X, Y и Z Pointers) существенно повышает скорость пересылки данных при работе прикладной программы.

Регистровый файл занимает младшие 32 байта в общем адресном пространстве SRAM AVR. Такое архитектурное решение позволяет получать доступ к быстрой «регистровой» оперативной памяти микроконтроллера двумя путями - непосредственной адресацией в коде команды к любой ячейке и другими способами адресации ячеек SRAM. В технической документации фирмы Atmel это полезное свойство носит название «быстрое контекстное переключение» и является еще одной отличительной особенностью архитектуры AVR, повышающей эффективность работы микроконтроллера и его производительность. Особенно заметно данное преимущество при реализации процедур целочисленной 16-битной арифметики, когда исключаются многократные пересылки меж-

ду различными ячейками памяти данных при обработке арифметических операндов в ALU.

Система команд AVR весьма развита и насчитывает до 133 различных инструкций. Почти все команды имеют фиксированную длину в одно слово (16 бит), что позволяет в большинстве случаев объединять в одной команде и код операции, и операнд(ы). Лишь немногие команды имеют размер в 2 слова (32 бит) и относятся к группе команд вызова процедуры CALL, длинных переходов в пределах всего адресного пространства JMP, возврата из подпрограмм RET и команд работы с памятью программ LPM. Различают пять групп команд AVR: условного ветвления, безусловного ветвления, арифметические и логические операции, команды пересылки данных, команды работы с битами. В последних версиях кристаллов AVR семейства «mega» реализована функция аппаратного умножения, что придает новым микроконтроллерам еще больше привлекательности с точки зрения разработчика.

По разнообразию и количеству реализованных инструкций AVR больше похожи на CISC, чем на RISC процессоры. Например, у PIC-контроллеров система команд насчитывает до 75 различных инструкций, а у MCS51 она составляет 111. В целом, прогрессивная RISC архитектура AVR в сочетании с наличием регистрового файла и расширенной системы команд позволяет в короткие сроки создавать работоспособные программы с эффективным кодом как по компактности реализации, так и по скорости выполнения.

4.2. Память программ

В основном все современные микроконтроллеры имеют встроенную Flash-память программ, которая может быть загружена как с помощью обычного программатора, так и с помощью SPI-интерфейса, в том числе непосредственно на целевой плате. Число циклов перезаписи - не менее 1000. Некоторые кристаллы имеют возможность самопрограммирования. Это означает, что микроконтроллер способен самостоятельно, без какого-либо внешнего программатора, изменять содержимое ячеек памяти программ. То есть, существует возможность без перепрограммирования микроконтроллера изменять алгоритмы его функционирования и программы, заложенные в нем, и далее работать уже по измененному алгоритму или новой программе. Например, можно написать и сохранить несколько рабочих версий программы для конкретного приложения во внешней энергонезависимой памяти (DataFlash, EEPROM и т.п.), а

затем по мере необходимости или по реакции на какие-нибудь внешние или внутренние логические условия перегружать рабочие программы в тот же самый микроконтроллер, не извлекая его из печатной платы.

Для обеспечения этого механизма весь массив памяти программ делится на две неравные по объему области: блок загрузчика (программа, управляющая перезаписью Flash-памяти программ) и блок для размещения рабочего программного кода, причем свободная память в области загрузчика может быть использована в качестве дополнительного пространства для рабочего кода. Программа-загрузчик создается самим разработчиком и должна быть запрограммирована внешним программатором.

4.3. Память данных

Внутренняя оперативная память SRAM имеется у всех семейств микроконтроллеров. Ее размер варьируется от десятков байт до десятков килобайт. Для некоторых микроконтроллеров возможна организация подключения внешней памяти данных объемом до нескольких десятков килобайт. Данное подключение производится посредством параллельной шины микроконтроллера.

Практически все современные микроконтроллеры имеют также блок энергонезависимой электрически стираемой памяти данных EEPROM. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, таблиц перекодировок, калибровочных коэффициентов и т.п. EEPROM также может быть загружена извне как через SPI интерфейс, так и с помощью обычного программатора. Число циклов перезаписи - не менее 100 000. Два программируемых бита секретности позволяют защитить память программ и энергонезависимую память данных EEPROM от несанкционированного считывания.

4.4. Тактовый генератор

Тактовый генератор вырабатывает импульсы для синхронизации работы всех узлов устройства. Стандартный внутренний тактовый генератор микроконтроллера может запускаться от нескольких источников опорной частоты (внешний генератор, внешний кварцевый резонатор, внутренняя или внешняя RC-цепочка).

Поскольку микроконтроллеры полностью статические, минимальная допустимая частота ничем не ограничена (вплоть до пошагового режима). Максимальная рабочая частота определяется конкретным типом микроконтроллера. Верхние границы частотного диапазона, указанные в таблицах для выбранного микроконтроллера, гарантируют устойчивую работу микроконтроллеров при работе во всем температурном диапазоне. В некоторых микроконтроллерах имеются специальные аппаратные блоки для увеличения рабочей частоты. Например, микроконтроллер ATtiny15L семейства AVR фирмы Atmel содержит блок PLL для аппаратного умножения основной тактовой частоты в 16 раз. При номинальном значении последней 1,6 МГц получаемая вспомогательная периферийная частота равна 25,6 МГц. Эта частота может служить источником для одного из таймеров/счетчиков микроконтроллера, значительно повышая временное разрешение его работы.

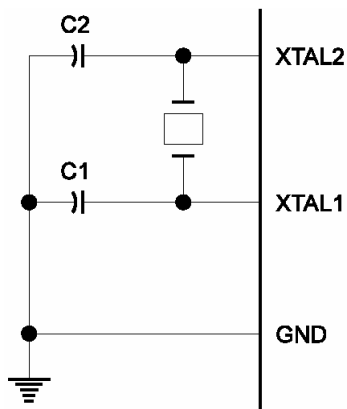


Рисунок 3.2 Пример подключения кварцевого резонатора к МК

4.5. Сторожевой таймер

Сторожевой (WATCHDOG) таймер предназначен для защиты микроконтроллера от сбоев в процессе работы, другими словами, это сторожевая собака, которая должна гавкнуть, если на нее долго не обращают внимание.

Он присутствует почти в каждом МК и предназначен для сброса МК в случае зависания его управляющей программы. Сторожевой таймер тактируется отдельным встроенным RC-генератором. Принцип его работы прост, после запуска текущее значение регистра сторожевого таймера инкрементируется с приходом каждого следующего тактового сигнала. И если данное значение не будет периодически сбрасываться в

программе, то при переполнении данного регистра произойдет рестарт МК.

WATCHDOG-таймер снабжен своим собственным предварительным делителем входной частоты с программируемым коэффициентом деления, что позволяет подстраивать временной интервал переполнения таймера и сброса микроконтроллера. WATCHDOG-таймер может быть отключен программным образом во время работы микроконтроллера как в активном режиме, так и в любом из режимов пониженного энергопотребления. В последнем случае это приводит к значительному снижению потребляемого тока.

WDR не рекомендуется использовать при отладке программ, так как ошибки в работе могут остаться незамеченными.

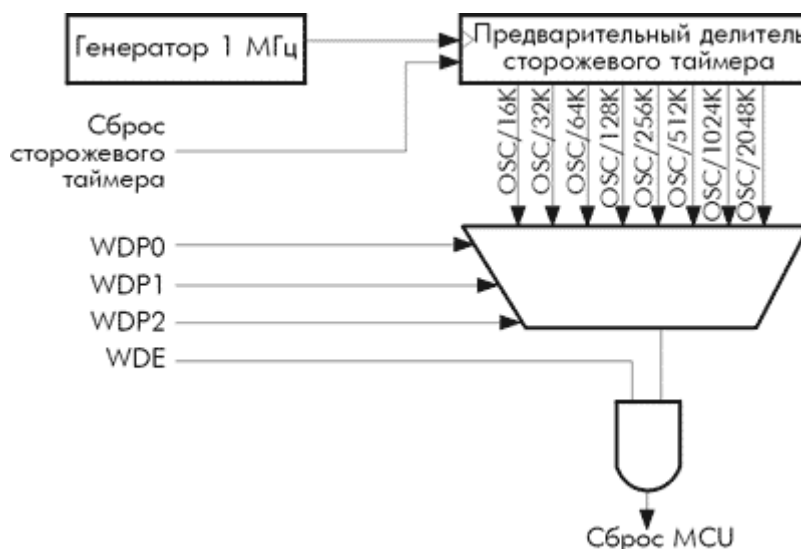


Рисунок 3.3 Блок схема сторожевого таймера

4.6. Порты ввода/вывода

Порты ввода/вывода - обязательная часть любого МК. Обычно их используют для связи с ближайшим окружением - датчиками и исполнительными механизмами. Число портов ввода/вывода напрямую зависит от числа выводов корпуса.

Обычно каждый разряд порта может быть запрограммирован на ввод или на вывод информации. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, напри-

мер, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы.

Из реализаций портов ввода/вывода, можно выделить интересную архитектурную особенность их построения у микроконтроллеров AVR. Она заключается в том, что для каждого физического вывода существует 3 бита контроля/управления:

- бит контроля направления передачи данных и привязки вывода к шине питания (VCC);
- бит привязки вывода к VCC и бит выходных данных;
- бит для отображения логического уровня сигнала на физическом выводе микросхемы.

Естественно возникает вопрос: а зачем необходимы именно 3 бита? Дело в том, что использование только двух бит контроля/управления порождает ряд проблем при операциях типа «чтение-модификация-запись». Например, если имеют место две последовательные операции «чтение-модификация-запись», то первый результат может быть потерян безвозвратно, если вывод порта работает на емкостную нагрузку и требуется некоторое время для стабилизации уровня сигнала на внешнем выводе микросхемы.

Также очень часто порты МК используются для выполнения альтернативных функций - в периферийных устройствах.

4.7. Контроллер прерываний

Контроллер прерываний - одна из важнейших частей МК. Особенность систем реального времени заключается в том, что для них чрезвычайно важным параметром является время реакции на внешние события. Система реального времени предполагает совершенно конкретную, рассчитываемую на этапе разработки скорость реакции системы управления на внешние события. Задержки сверх расчетных здесь просто недопустимы - они могут приводить к катастрофическим последствиям.

Проблемы быстрой реакции на события решаются организацией системы прерываний. Она подразумевает, что для каждого такого события разрабатывается отдельный «кусочек» кода, который формирует реакцию МК на него. Этот «кусочек» кода называют подпрограммой обработки запроса на прерывание (для краткости часто используют термин подпрограмма прерывания) и размещают в памяти программ по известному адресу. В момент возникновения заданного события сигнал об этом поступает на вход контроллера прерываний. Последний представляет со-

бой устройство, устанавливающее однозначное соответствие между входным сигналом о произошедшем событии и адресом программной памяти, по которому размещена точка входа в подпрограмму обработки запроса прерывания от данного события. Контроллер прерывает выполнение ЦП текущей программы и инициирует его переход на выполнение подпрограммы обработки прерывания. Время, прошедшее с момента возникновения события до начала выполнения первой инструкции подпрограммы прерывания, называют временем реакции МК на событие. После окончания обработки ЦП автоматически возвращается к выполнению прерванной программы. Каждый периферийный блок МК имеет хотя бы одно прерывание.

Другая функция контроллера прерываний - установка приоритетов событий. Понятие приоритет означает, что выполняемая подпрограмма прерывания может быть прервана другим событием только при условии, что оно имеет более высокий приоритет, чем текущее. В противном случае ЦП перейдет к обработке нового события после окончания обработки предыдущего.

4.8. Таймеры/счетчики

Все микроконтроллеры имеют в своем составе таймеры/счетчики количеством до десятка штук и с различной разрядностью (в основном 8 или 16 бит), которые могут работать и как таймеры от внутреннего источника опорной частоты, и как счетчики внешних событий с внешним тактированием.

Общие черты всех таймеров/счетчиков следующие:

- наличие программируемого предварительного делителя входной частоты с различными градациями деления;
- отличительной чертой является возможность работы таймеров/счетчиков на основной тактовой частоте микроконтроллера без предварительного ее понижения, что существенно повышает точность генерации временных интервалов системы;
- независимое функционирование от режима работы процессорного ядра микроконтроллера (т.е. они могут быть как считаны, так и загружены новым значением в любое время);
- возможность работы или от внутреннего источника опорной частоты, или в качестве счетчика событий. Верхний частотный порог определен в этом случае как половина основной тактовой частоты микрокон-

троллера. Выбор перепада внешнего источника (фронт или срез) программируется пользователем;

- наличие различных векторов прерываний для нескольких различных событий (переполнение, захват, сравнение).

Таймеры/счетчики можно использовать для:

- 1) точного формирования временных интервалов,
- 2) подсчета импульсов на выводах МК.

При работе таймера/счетчика в режиме подсчета импульсов от внешнего сигнала, внешний сигнал синхронизируется с тактовым генератором процессора. Для правильной обработки внешнего сигнала минимальное время между соседними импульсами должно превышать период тактовой частоты процессора. Подсчет импульсов ведется по одному из трех выбранных режимов:

- по нарастающему фронту;
- по спаду;
- по нулевому уровню.

Также в таймерах встречается аппаратно реализованный цифровой фильтр нижних частот.

3) формирования последовательности импульсов - ШИМ-режим (на основе ШИМ можно легко реализовать ЦАП).

4) реализации функций захвата/сравнения. Таймеры/счетчики способны вырабатывать запросы прерываний, переключая ЦП на их обслуживание по событиям и освобождая его от необходимости периодического опроса состояния таймеров. Типовые прерывания при переполнении таймера/счетчика, при достижении заданного уровня в функции захвата/сравнения.

5) реализации часов реального времени (RTC). Система RTC реализована во многих микроконтроллерах. Таймер/счетчик RTC имеет свой собственный предварительный делитель, который может быть программным способом подключен или к основному внутреннему источнику тактовой частоты микроконтроллера, или к дополнительному асинхронному источнику опорной частоты (кварцевый резонатор или внешний синхросигнал). Внутренний осциллятор, нагруженный на счетный вход таймера/счетчика RTC, оптимизирован для работы с внешним «часовым» кварцевым резонатором 32,768 кГц.

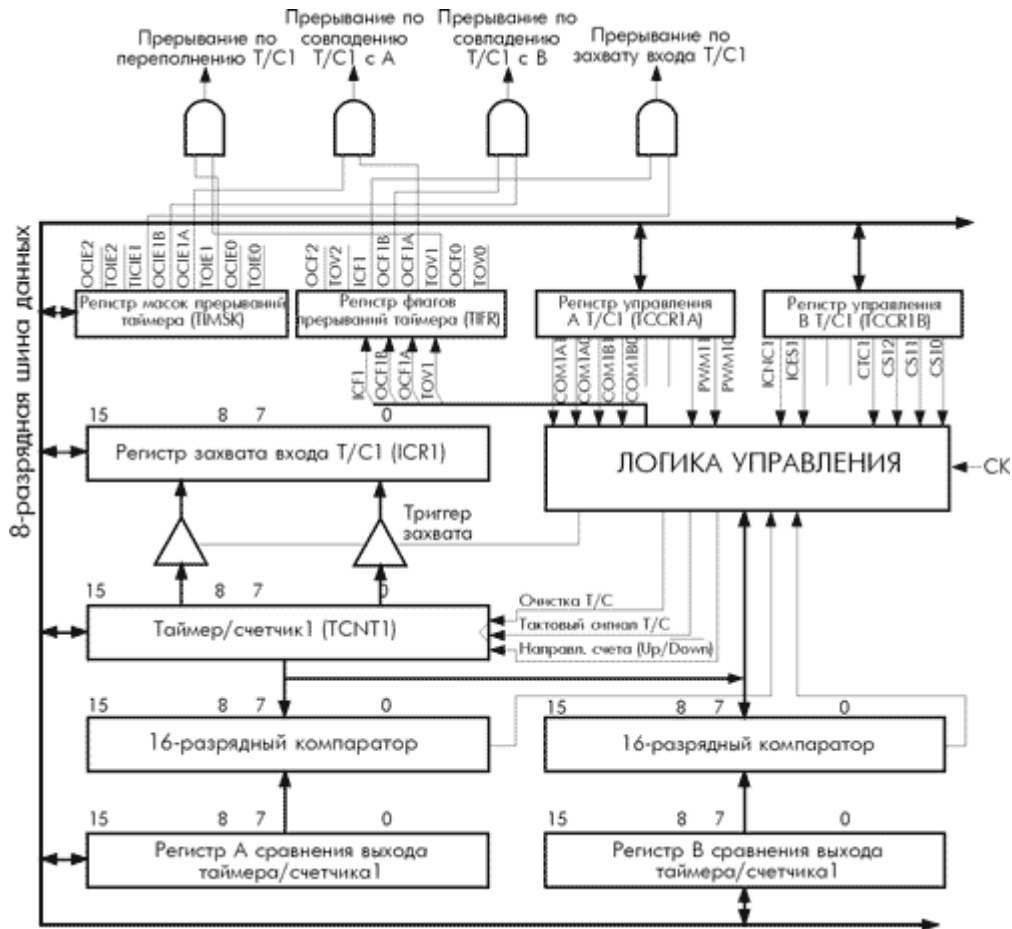
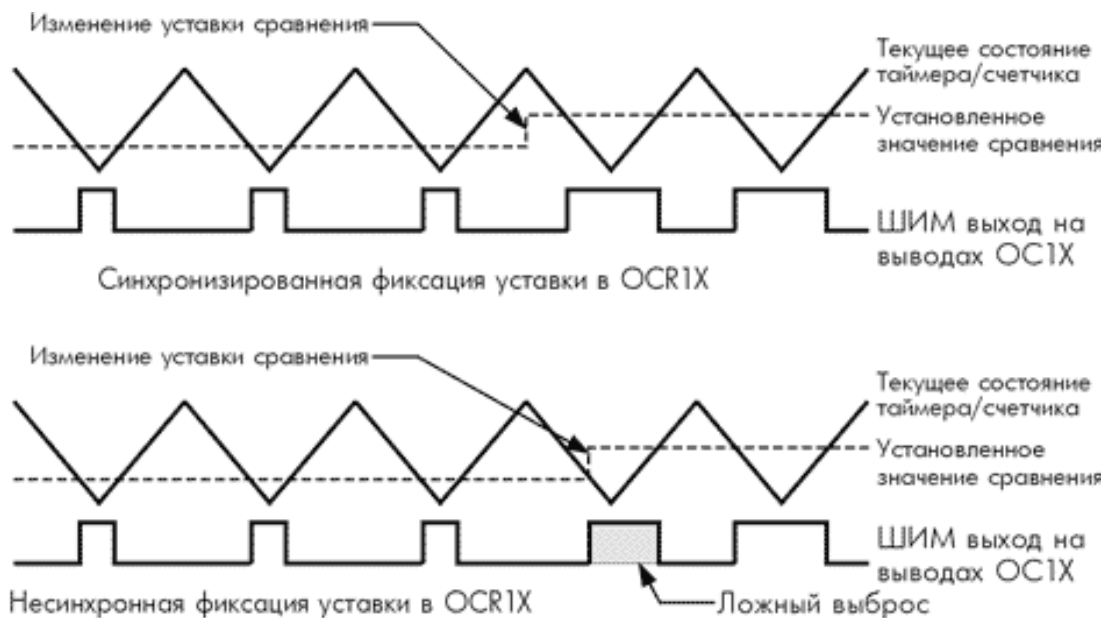


Рисунок 3.4 Блок-схема таймера-счетчика в МК семейства AVR



Примечание: X = A или B

Рисунок 3.5. Пример реализации ШИМ на основе таймера-счетчика в МК

4.9. Аналоговый компаратор

Аналоговый компаратор входит в состав большинства микроконтроллеров. Типовое напряжение смещения равно 10 мВ, время задержки распространения составляет 500 нс и зависит от напряжения питания микроконтроллера. Так, например, при напряжении питания 2,7 вольт оно равно 750 нс. Аналоговый компаратор имеет свой собственный вектор прерывания в общей системе прерываний микроконтроллера. При этом тип перепада, вызывающий запрос на прерывание при срабатывании компаратора, может быть запрограммирован пользователем как фронт, срез или переключение. Логический выход компаратора может быть программным образом подключен ко входу одного из таймеров/счетчиков. Это дает возможность измерять длительность аналоговых сигналов, а также максимально просто реализовывать АЦП двухтактного интегрирования.

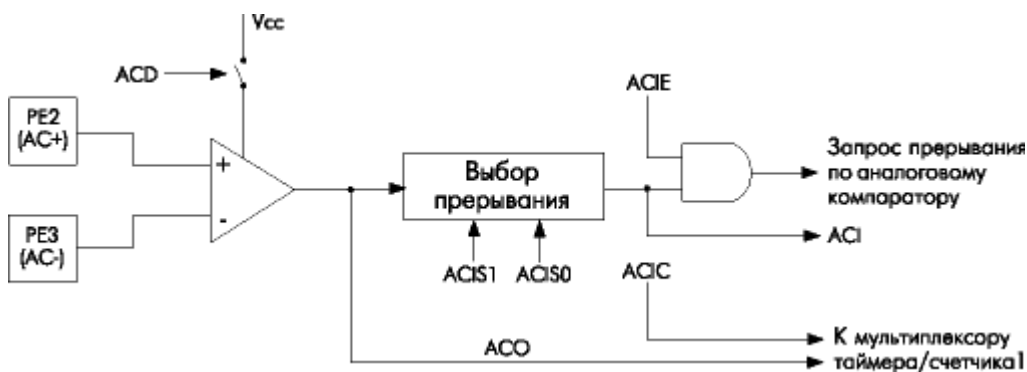


Рисунок 3.6 . Блок-схема аналогового компаратора в МК.

4.10. Аналого-цифровой преобразователь

Аналого-цифровой преобразователь (АЦП) в микроконтроллерах построен по классической схеме последовательных приближений с устройством выборки/хранения (УВХ). Каждый из аналоговых входов может быть соединен со входом УВХ через аналоговый мультиплексор. Устройство выборки/хранения имеет свой собственный усилитель, гарантирующий, что измеряемый аналоговый сигнал будет стабильным в течение всего времени преобразования. Разрядность большинства интегрированных АЦП составляет 10 бит при нормируемой погрешности ± 2 разряда.

АЦП может работать в двух режимах - однократное преобразование по любому выбранному каналу и последовательный циклический опрос всех каналов. Время преобразования выбирается программно с

- вначале процессор устанавливает на шине адреса адрес того периферийного устройства или ячейки памяти, с которой происходит информационный обмен;
- если на параллельной шине несколько устройств, то необходимо выбирать требуемое устройство ввода-вывода или микросхему памяти, формируя сигнал «ВЫБОР МИКРОСХЕМЫ» - «CHIP SELECT» (CS);
- на следующем такте процессор выставляет на шину данных информацию;
- далее формирует сигналы, соответствующие требуемому типу информационного обмена на шине управления (WR, RD, IOREQ, MREQ).

Таким образом, операции передачи адреса и данных оказываются разнесенными по времени, что позволяет в некоторых системах использовать для их передачи одну и ту же группу соединительных линий - мультиплексированная шина адреса-данных. Так, для реализации параллельного порта в МК используются два порта ввода/вывода: один - для выдачи старшего байта адреса, второй - для выдачи младшего байта адреса и байта данных. Младший байт адреса защелкивается во внешнем регистре сигналом с выхода микроконтроллера (ALE).

Последовательный порт - канал информационного обмена МК с внешним миром. Такие каналы связи занимают минимальное число выводов кристалла, обеспечивая связь на значительные расстояния с минимальными аппаратными затратами.

В большинстве МК реализованы два типа последовательных портов синхронные и асинхронные.

На сегодняшний день наиболее распространены два основных типа синхронных портов:

- I2C - двухпроводная двунаправленная шина;
- SPI - последовательный периферийный трехпроводный интерфейс.

Компания Philips разработала двунаправленную шину из двух проводов для эффективного контроля над интегральными схемами (ИС). Эта шина получила название Inter-IC или I2C. В настоящее время ИС компании Philips используются в более чем 150 видах CMOS и совместимых с шиной I2C двухполюсниках при производстве все трех вышеперечисленных видах продукции. Устройства совместимые с I2C имеют интерфейс прямо на микросхеме, который позволяет им соединяться при помощи шины I2C. Такой способ производства позволяет решить мно-

жество проблем, возникающих при проектировании интерфейсов между цифровыми схемами.

Основные характеристики шины I2C:

- Требуются только две линии: Serial Data Line (SDA) и Serial Clock Line (SCL).
- Каждое устройство, подключенное к шине, имеет свой уникальный адрес. На протяжении всего времени работы шины имеет место простая связь вида master/slave. Мастер может работать и как приемник, и как передатчик.
- Это действительно мульти-мастер шина, предусматривающая предотвращение ошибок и разрешение конфликтов, когда два или более мастера одновременно пытаются начать передачу данных.
- Двухнаправленная передача данных (8bit) производится на скоростях до 100 Кб/с в стандартном режиме или до 400 Кб/с в ускоренном режиме.
- Имеющийся на микросхеме фильтр, предотвращает скачки напряжения, предохраняя целостность данных
- Максимальное число ИС, одновременно подключенных к шине, ограничивается только максимальной емкостью шины в 400 пФ.

Основные характеристики SPI интерфейса:

- полнодуплексный трехпроводной синхронный обмен данными;
- два режима работы: ведущий (управляет процессом обмена данными - выдает импульсы синхронизации обмена) или ведомый (синхронизируется от ведущего);
- два режима передачи: старшим или младшим битом вперед;
- программируемые скорости обмена;
- прерывание по завершении передачи байта;
- буфер приемника – 1-байт;
- обнаружение ошибки искажения данных при передаче;
- линия выборки для реализации шинного обмена.

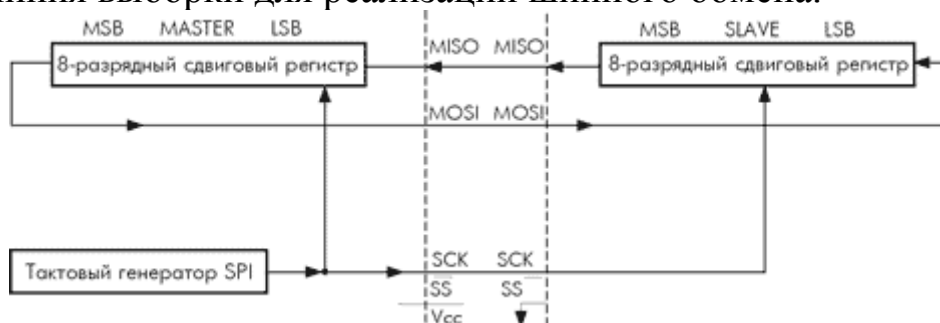


Рисунок 3.8. Реализация обмена между двумя устройствами (Master и Slave) по SPI интерфейсу.

UART - универсальный асинхронный последовательный приемопередатчик совместим с UART в персональном компьютере. На UART реализуются все популярные протоколы в мире встраиваемых систем: Bitbus (последовательная магистраль управления), CAN (межконтроллерный сетевой интерфейс) и многие другие.

Основные характеристики UART:

- скорость передачи до 155 200 бод;
- формат данных;
- обнаружение ошибок формирования кадра и переполнения;
- фильтрация шума;
- буфер приемника и передатчика;
- три отдельных прерывания: по завершении передачи, по завершении приема и по пустому буферу передатчика.

4.12. Режимы пониженного энергопотребления

Современные микроконтроллеры имеют несколько режимов пониженного энергопотребления, в которые могут быть переведены программным путем. Для разных семейств и разных микроконтроллеров в пределах каждого семейства изменяются количество и сочетание доступных режимов пониженного энергопотребления. Приведем шесть наиболее распространенных режимов пониженного энергопотребления.

1. Режим холостого хода (IDLE), в котором прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний и WATCHDOG-таймер продолжают функционировать.

2. Режим микропотребления (Power Down), в котором сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов. Выход из Power Down возможен либо по общему сбросу микроконтроллера, либо по сигналу (уровень) от внешнего источника прерывания. При включенном WATCHDOG-таймере ток потребления в этом режиме составляет около 60...80 мкА, а при выключенном - менее 1 мкА. Вышеприведенные значения справедливы для величины питающего напряжения 5 В.

3. Режим сохранения энергии (Power Save), который реализован только у тех микроконтроллеров, которые имеют в своем составе систему реального времени. В основном, режим Power Save идентичен Power Down, но здесь допускается независимая работа дополнительного таймера/счетчика RTC. Выход из режима Power Save возможен по прерыва-

нию, вызванному или переполнением таймера/счетчика RTC, или срабатыванием блока сравнения этого счетчика. Ток потребления в этом режиме составляет 6...10 мкА при напряжении питания 5 В на частоте 32,768 кГц.

4. Режим подавления шума при работе аналого-цифрового преобразователя (ADC Noise Reduction). Как уже отмечалось, в этом режиме останавливается процессорное ядро, но разрешена работа АЦП, двухпроводного интерфейса I2C и сторожевого таймера.

5. Основной режим ожидания (Standby). Идентичен режиму Power Down, но здесь работа тактового генератора не прекращается. Это гарантирует быстрый выход микроконтроллера из режима ожидания всего за 6 тактов генератора.

6. Дополнительный режим ожидания (Extended Standby). Идентичен режиму Power Save, но здесь работа тактового генератора тоже не прекращается. Это гарантирует быстрый выход микроконтроллера из режима ожидания всего за 6 тактов генератора.

Некоторые микроконтроллеры имеют еще одну примечательную архитектурную особенность, позволяющую значительно снизить энергопотребление всего кристалла в целом, когда в процессе работы возникают вынужденные паузы ожидания. В этом случае целесообразно уменьшить ток потребления центрального процессора и периферийных устройств как в активном режиме, так и в режиме холостого хода, понизив основную тактовую частоту микроконтроллера. Для этой цели на кристалле размещен специальный предварительный делитель, позволяющий делить основную тактовую частоту на целое число в диапазоне от 2 до 129. Включение/выключение данной функции осуществляется одной короткой командой в программе.

Большинство микроконтроллеров функционируют в широком диапазоне питающих напряжений от 1,8 до 6,0 вольт. Энергопотребление в активном режиме зависит от величины напряжения питания, от частоты, на которой работает микроконтроллер.

Температурные диапазоны работы микроконтроллеров:

- коммерческий ($0^{\circ}\text{C} \dots 70^{\circ}\text{C}$);
- промышленный ($-40^{\circ}\text{C} \dots +85^{\circ}\text{C}$);
- автомобильный ($-40^{\circ}\text{C} \dots +100^{\circ}\text{C}$);
- военный ($-55^{\circ}\text{C} \dots +125^{\circ}\text{C}$).

5. КЛАССИФИКАЦИЯ МИКРОКОНТРОЛЛЕРОВ

Сегодня в мире выпускаются тысячи типов МК.

Микроконтроллеры обычно классифицируют по разрядности данных, обрабатываемых арифметико-логическим устройством (АЛУ):

- четырехразрядные - самые простые и дешевые;
- восьмиразрядные - наиболее многочисленная группа (оптимальное сочетание цены и возможностей), к этой группе относятся микроконтроллеры серии MCS-51 (Intel) и совместимые с ними, PIC (MicroChip), HC68 (Motorola), Z8 (Zilog) и др.
- шестнадцатиразрядные - более высокопроизводительные, но более дорогостоящие;
- тридцатидвухразрядные - обычно являющиеся модификациями универсальных микропроцессоров.

Отдельно рассматриваются DSP-МК (DSP - Digital Signal Processor - цифровой сигнальный процессор), ориентированные на использование в системах обработки сигналов.

Сегодня наибольшая доля мирового рынка МК принадлежит восьмиразрядным устройствам (около 50 % в стоимостном выражении). За ними следуют 16-разрядные и DSP-МК (каждая из групп занимает примерно по 20 % рынка).

Внутри каждой группы МК делятся на CISC- и RISC-устройства. Наиболее многочисленной группой являются CISC-МК, но в последние годы среди новых МК преобладает RISC-архитектура.

5.1. Четырехразрядные микроконтроллеры

Четырехразрядные микроконтроллеры являются очень простыми и дешевыми устройствами, предназначенными для замены несложных схем на «жесткой» логике в системах с невысоким быстродействием. Типичные случаи применения - часы, калькуляторы, игрушки, простые устройства управления в промышленных устройствах и бытовой технике.

Типичные характеристики четырехразрядных микроконтроллеров:

- ОЗУ - объем - 16...64 четырехразрядные ячейки;
- ПЗУ - объем - 0.5 ...1 К восьмиразрядных ячеек, тип - масочное ПЗУ (наиболее дешевое);
- система команд - количество - 30...50;
- тактовая частота - 100 КГц...1МГц;

- периферийные устройства - 2...4 четырехразрядных параллельных порта, иногда контроллер жидкокристаллического индикатора;
- стоимость - порядка 0,1\$.

Примеры четырехразрядных микроконтроллеров - отечественные серии КР145ВМ1405, КР145ВМ1406, КР1834 и т.д.

Одним из крупнейших производителей четырехразрядных микроконтроллеров является фирма «Ангстрем», г. Зеленоград.

5.2. Восьмиразрядные микроконтроллеры

Современная тенденция в области разработки микропроцессорных управляющих систем состоит в наращивании мощности вычислителей, в переходе к 16- и 32-разрядным контроллерам и промышленным персональным компьютерам (ПК), в использовании операционных систем (ОС), построении многоуровневой иерархии интерфейсов. Подобный подход, при всех его положительных сторонах: модульности, снижении трудоемкости разработки, стандартизации протоколов и т. д., не всегда применим в российских условиях. Причина — значительная степень избыточности, приводящая к недопустимо высокой цене.

Между тем существует и продолжает активно развиваться класс восьмиразрядных микроконтроллеров и микросхем поддержки. Они могут использоваться не только для создания простейших устройств управления: регуляторов, контроллеров бытовой техники интеллектуальных периферийных модулей, но и для создания достаточно сложных локальных и распределенных комплексов промышленной автоматики, таких как системы учета энергоресурсов, контроля доступа, управления микроклиматом и др.

Совершенно закономерно, что и в России наиболее популярными стали именно восьмиразрядные МК.

Типичные характеристики восьмиразрядных микроконтроллеров:

- ОЗУ - объем – 256 байт - 4Кбайт;
- ПЗУ - объем – 1..64Кбайт, в новых МК тип - FLASH ПЗУ;
- энергонезависимая память данных тип EEPROM – 256 байт-2Кбайт;
- тактовая частота – 1..50МГц, основная доля 4..10МГц;
- широкий спектр периферийных устройств:
 - 2...4 восьмиразрядных порта,
 - таймеры, счетчики, ШИМ, часы реального времени, АЦП, ЦАП, аналоговый компаратор;

- последовательные порты – от 1 до 4;
- многоуровневая система энергосбережения;
- контроллеры ЖКИ, системы управления электродвигателями и т.д.;
- стоимость – единицы \$.

Наиболее яркими представителями восьмиразрядных CISC-МК являются изделия компаний Intel (семейства MCS-48, MCS-51), Motorola (68HC05, 68HC08, 68HC11) и Zilog (Z8).

Наиболее яркими представителями восьмиразрядных RISC-МК являются изделия компаний Atmel (семейства AT90S, ATmega, ATTiny), Microchip (PIC), Scenix (SX), АНГСТРЕМ (семейство ТЕСЕЙ - KP1878BE1). HOLTEK (HT48).

5.3. 16- и 32- разрядные микроконтроллеры

В последнее время наметилось существенное увеличение спроса на производственные тридцатидвухразрядные микроконтроллеры. В первую очередь это связано со значительным усложнением современной бытовой техники (производители которой потребляют до 70–80 % всех производимых микроконтроллеров), а также с резким снижением цен, что делает возможным создание более гибких и мощных систем, обладающих большим по сравнению с предыдущими устройствами набором возможностей.

Эти контроллеры ставятся в DVD-проигрыватели, CD-проигрыватели, автоответчики — словом туда, где надо работать с большими объемами памяти.

Типичные характеристики 16- и 32-разрядных микроконтроллеров:

- тактовая частота – 4...60 МГц, основная доля 10-20 МГц;
- ПЗУ – внутреннее и внешнее, адресное пространство до 64 Мбайт, обычно тип FLASH ПЗУ;
- мощная система команд и аппаратная реализация математического аппарата;
- из периферийных устройств:
 - линии ввода-вывода – 32-256;
 - интегрированный контроллер внешней памяти (SRAM или DRAM);
 - контроллеры DMA;
 - контроллер прямого доступа к памяти;
 - контроллер 16-разрядной шины ввода – вывода;
 - таймеры, счетчики, ШИМ, часы реального времени;

- АЦП, ЦАП;
- последовательные порты – от 2 до 6;
- развитые коммуникационные возможности, аппаратная поддержка IrDA, PCMCIA HDLC, USB, CAN, LCD;
- стоимость – 10-30 \$.

Наиболее яркими представителями шестнадцатиразрядных микроконтроллеров являются семейства компании Fujitsu (F2MC-16L/16LX/16F), Siemens (C16), Intel (MCS-96), Texas Instruments (MSP), Motorola (68HC), Mitsubishi (7700).

Особое внимание необходимо уделить фирме Advanced Micro Devices, которая является, единственным, кроме Intel, производителем процессоров семейства x86, предназначенных для применения во встраиваемых системах.

Идея применения процессоров семейства x86 во встраиваемых системах была заложена еще Intel, которая выпустила встраиваемые варианты своих кристаллов i186, i386, а позднее и i486. Применение подобных процессоров во встраиваемых системах удобно в первую очередь с точки зрения разработки, поскольку практически единственным средством, необходимым разработчику, является персональный компьютер. Кроме того, для семейства x86 накоплено огромное количество программного обеспечения, которое с минимальными переделками может быть адаптировано для работы во встраиваемых системах, а также отсутствует необходимость приобретения специальных средств симуляции и отладки, поскольку всю разработку ПО, включая отладку, можно производить с использованием РС в качестве эмулятора. Однако применение интеловских встраиваемых процессоров несколько сдерживали два фактора – их относительно большое энергопотребление и слабо развитая периферия, не отвечающая современным требованиям.

Для восполнения этого пробела параллельно развитию x86 фирма AMD разработала новое семейство – E86TM, состоящее из 16- и 32-разрядных встраиваемых.

Все семейство можно разделить на три группы:

- процессоры семейства x86 с пониженным энергопотреблением и расширенными средствами управления рабочей частотой (Am386, Am486, AMD-K6E, AMD-K6-2E);
- 16-разрядные микроконтроллеры Am186, базирующиеся на ядре процессора i186 и обладающие развитой встроенной периферией;
- 32-разрядные микроконтроллеры, базирующиеся на ядрах процессоров 386 (ElanSC300, ElanSC310), 486 (ElanSC400, ElanSC410) и

Am5x86 (ElanSC520) и содержащие все необходимое для построения РС-совместимой системы.

16- и 32-разрядные микроконтроллеры предназначены для использования при построении достаточно сложных систем, разработка которых практически невозможна без использования специальных программных и аппаратных средств. Поэтому на рынке представлен широкий спектр компиляторов и отладчиков для различных операционных систем применяемых в данных микроконтроллерах.

Разработка программного обеспечения (ПО) систем реального времени является сложным и многогранным процессом. Традиционно основные задачи по управлению ресурсами систем реального времени возлагаются на операционные системы реального времени (ОСРВ). ОСРВ координирует распределение ресурсов компьютерной системы (микропроцессор, память, порты ввода/вывода и т.п.) между конкурирующими по времени вычислительными процессами. И в этом случае применение ОСРВ оправдано и неоспоримо, ведь пользователь разрабатывает лишь прикладное программное обеспечение, а ОСРВ берет на себя основные заботы по распределению ресурсов, координации вычислительных процессов между собой, своевременной реакции на прерывания. И к настоящему времени существует сложившийся ряд операционных систем реального времени (OS-9, VxWorks, pSOS, RTLinux и т.п.), признанных лидеров в своей области.

5.4. Дискретный процессор обработки сигналов (DSP)

Digital signal processor - микропроцессор, предназначенный для обработки сигналов.

Архитектура дискретного процессора сигналов рассчитана на быстрое выполнение процессов, необходимых для обработки дискретных сигналов. И в первую очередь - умножения и накопления. Этот, обычно однокристальный, процессор, часто именуемый также цифровым процессором сигналов, характеризуется следующим:

- возможность программирования выполняемых процессов;
- наличие гибких высокоскоростных арифметических средств;
- возможность выполнения за один такт операции умножения, сдвига и других арифметических и логических операций;
- выполнение операций суммирования произведений с высокой скоростью;
- наличие буферов для циклического запоминания сигналов;

- отсутствие непроизводительных затрат на организацию программных циклов и ответвлений.

Использование дискретных процессоров обработки сигналов позволяет снизить стоимость, упростить структуру устройств. Например, в технике связи рассматриваемые процессоры используются для создания телефонных аппаратов, модемов, радиотелефонной связи, организации видеоконференций, мультиплексирования каналов.

Существуют различные типы процессоров сигналов. Одни из них предназначены для вычислений с фиксированной, другие - с плавающей запятой. Большое значение имеют векторные процессоры сигналов, которые используют методику векторных исчислений (производят операции над векторами). Применение векторных процессоров резко увеличивает пропускную способность, поскольку позволяет разработчику системы использовать лишь небольшое число команд вместо того, чтобы писать много строк программного кода. Стремление повысить быстродействие привело к созданию дискретных процессоров сигналов с параллельной обработкой.

Примеры DSP

Временем появления первого ЦСП принято считать 1982 год, когда компания Texas Instruments (TI) начала выпуск кристаллов TMS32010, имевших рекордную для того времени производительность в несколько MIPS (миллионов инструкций в секунду). Вскоре о производстве ЦСП заявили также фирмы Analog Devices, Motorola, AT&T (Lucent Technologies) и некоторые другие. Первые ЦСП создавались при использовании технологических норм около 1,2 мкм и в большинстве случаев предназначались для применения в военной индустрии. Но по мере развития технологии, вследствие совершенствования архитектуры ЦСП и сокращения их стоимости область применения сигнальных процессоров существенно расширилась и охватывает сегодня весь спектр приложений, где требуется высокопроизводительная обработка больших массивов данных.

Общепризнанным лидером по объему продаж является компания Texas Instruments, которая за последние три года увеличила свою долю с ~45 % до 48 %. Продукция компании хорошо известна в России и представлена как западными дистрибьюторами, работающими на российском рынке, так и отечественной компанией СКАНТИ-Рус (texas@inlinegroup.ru), являющейся прямым бизнес-партнером TI в России.

Производителем номер два на мировом рынке ЦСП является компания Lucent Technologies, бывшее подразделение компании AT&T. Имея около 28 % мирового объема продаж, компания мало известна в России, во многом благодаря тому, что выпускаемые ею ЦСП имеют достаточно узкую специализацию под конкретные телекоммуникационные приложения и не поставляются на массовый рынок. Авторам неизвестны дистрибьюторы, продающие ЦСП Lucent Technologies в России.

На третьем месте по объему продаж в 1999 году находилась компания Analog Devices (ADI) с 13 % рынка. Компания известна своей агрессивной политикой и добилась определенных успехов в конкуренции с TI на российском рынке, где активно работают два российских дистрибьютора — фирмы «Аргуссофт» и «Автекс». ЦСП производства ADI особенно эффективны при вычислениях с плавающей запятой и при построении многопроцессорных систем. Их архитектура будет рассмотрена более подробно.

Компания Motorola в 1999 году имела около 12 % рынка ЦСП, идя вплотную за ADI. В России компания представлена официальными западными дистрибьюторами и российской компанией «Гамма».

Пример использования DSP: MP3-плеер на базе ЦСП и флэш-памяти, позволяющий воспроизводить до 4 часов музыки, полученной через Интернет.

5.5. Программное обеспечение микропроцессоров

Процесс написания программ для микроконтроллеров состоит из нескольких этапов:

- 1) подготовка исходного текста программы на каком-либо из языков программирования;
- 2) компиляция программы;
- 3) отладка и тестирование программы;
- 4) окончательное программирование и подготовка к серийному производству.

На каждом из этапов необходимо применение специальных программных и аппаратных средств.

1. Подготовка исходного текста программы может производиться в любом текстовом редакторе, но в основном в среде используемого компилятора. Основные языки программирования микроконтроллеров приведены ниже:

- ассемблер – ранее единственный, а на сегодняшний день эффективен только для написания программных кодов критичных к времени исполнения или занимаемому размеру в памяти;
- СИ – стандарт для промышленных систем;
- BASIC, PASCAL, Fractal – используются в основном в учебных целях.

2. Компиляторы.

Компиляторы – это программы, которые преобразуют исходные тексты программ, написанные на языке программирования высокого уровня, в программу на машинном языке, «понятную» соответствующему микроконтроллеру. Полученный код, называемый исполняемой программой, устанавливается в микроконтроллер и запускается в нем без дополнительных преобразований.

3. Отладка и тестирование программы осуществляется аппаратно-программными средствами разработки и отладки программ для микроконтроллеров (рассмотрены ниже).

4. Окончательное программирование и подготовка к серийному производству.

Программаторы разделяются на два типа:

- пассивные – согласование сигналов между микроконтроллером и компьютером, который в данном случае выступает программатором;
- интеллектуальные – программатор выполняет все или часть операций программирования.

6. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ И ОТЛАДКИ ДЛЯ МИКРОКОНТРОЛЛЕРОВ

С развитием техники разработка любого микропроцессорного устройства становится все более затруднительной без использования средств разработки и отладки программного обеспечения микропроцессора.

Диагностика - процесс анализа состояния объекта. Объектом диагностики может быть сеть, система, компьютер, устройство, программа. В процессе диагностики изучаются характеристики, параметры и функции, выполняемые объектом. При этом осуществляется тестирование и анализ проведенного исследования. В этом процессе определяется характер, место, причина имеющихся неисправностей и ошибок, а также предпринимаются меры по их устранению. Благодаря диагностике становится возможным также прогнозирование поведения объекта в будущем.

При эксплуатации сети либо системы диагностика осуществляется всякий раз, как только обнаружится ошибка. Диагностическая программа исследует причину возникновения ошибки и предоставляет данные для последующего анализа. Ошибки же могут возникать при запоминании данных, их обработке и передаче. Для обнаружения ошибок в данные вводится определенная избыточность, позволяющая осуществлять необходимую диагностику. В особо важных случаях процесс обработки данных дублируется. При возникновении неисправности устройств осуществляется фиксация факта неисправности, определяется ее место и вид. Далее передаются сообщения о неисправности, устройство отключается и, если это возможно, заменяется резервным.

К числу основных инструментальных средств отладки микроконтроллеров 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.

6.1. Внутрисхемные эмуляторы

Внутрисхемный эмулятор - программно аппаратное средство, способное замещать собой эмулируемый процессор в реальной схеме. Внутрисхемный эмулятор - это наиболее мощное и универсальное отладочное средство.

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 выходы 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 эффективности программного кода);
- таймер реального времени;
- программно-аппаратные средства, обеспечивающие возможность чтения и модификации ресурсов эмулируемого процессора «на лету».

ту», 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.

6.1.1. Отладчик

Отладчик является своеобразным мостом между разработчиком и отладочным средством. Состав и объем информации, проходящей через средства ввода-вывода, доступность ее для восприятия, контроля и, при необходимости, для коррекции и модификации напрямую зависят от свойств и качества отладчика.

Хороший отладчик позволяет осуществлять:

- загрузку отлаживаемой программы в память системы;
- вывод на монитор состояния и содержимого всех регистров и памяти и, при необходимости, их модификацию;
- управление процессом эмуляции.

Более мощные отладчики, обычно их называют высокоуровневыми (High-Level Debuggers), помимо этого, позволяют:

- вести символьную отладку благодаря тому, что отладчик «знает» адреса всех символьных переменных, массивов и структур (за счет использования специальной информации, поставляемой компилятором). При этом пользователь может оперировать более приемлемыми для человека символьными именами, не утруждая себя запоминанием их адресов;

- контролировать и анализировать не только дисассемблированный текст, но и исходный текст программы, написанной на языке высокого уровня, и даже с собственными комментариями.

Такой отладчик позволяет пользователю одновременно контролировать ход выполнения программы и видеть соответствие между исходным текстом, образом программы в машинных кодах и состоянием всех ресурсов эмулируемого микроконтроллера.

Следует отметить, что высокоуровневый отладчик обеспечивает выполнение всех своих функций только в том случае, если используется кросс-компилятор, поставляющий полную и правильную отладочную информацию (не все компиляторы, особенно их пиратские версии, по-

ставляют такую информацию), и при этом формат ее представления должен быть «знаком» отладчику.

6.1.2. Эмуляционная память

Наличие эмуляционной памяти дает возможность использовать ее в процессе отладки вместо ПЗУ в отлаживаемой системе и, более того, отлаживать программу без использования реальной системы или ее макета. При необходимости внесения изменений в отлаживаемую программу достаточно загрузить новую или модифицированную программу в память эмулятора, вместо того чтобы заниматься перепрограммированием ПЗУ.

Существуют модели эмуляторов, которые позволяют пользователю «подставлять» вместо ПЗУ эмуляционную память не только целиком, но и поблочно (в некоторых моделях минимальный размер блока может достигать одного байта), в порядке, определенном пользователем. Для этого пользователю достаточно задать распределение памяти данных и памяти программ, в соответствии с которым процессор будет получать доступ и к содержимому ПЗУ в отлаживаемой системе, и к содержимому эмуляционной памяти внутрисхемного эмулятора.

6.1.3. Трассировщик

В сущности, трассировщик представляет собой логический анализатор, работающий синхронно с процессором и фиксирующий поток выполняемых инструкций и состояния выбранных внешних сигналов. Существуют модели внутрисхемных эмуляторов, которые позволяют трассировать не только внешние сигналы, но и состояния внутренних ресурсов микроконтроллера, например, регистров. Такие эмуляторы используют специальные версии микроконтроллеров (эмуляционные кристаллы).

6.1.4. Процессор точек останова

Процессор точек останова позволяет останавливать выполнение программы или выполнять иные действия, например запускать или останавливать трассировщик при выполнении заданных пользователем условий. В отличие от механизма обычных точек останова, процессор точек останова позволяет формировать и отслеживать условия практиче-

ски любой степени сложности, и при этом эмулируемый процесс не выводится из масштаба реального времени.

6.1.5. Профилировщик

Профилировщик (иначе анализатор эффективности программного кода) позволяет получить по результатам прогона отлаживаемой программы следующую информацию:

- количество обращений к различным участкам программы;
- время, затраченное на выполнение различных участков программы.

Анализ статистической информации, поставляемой профилировщиком, позволяет легко выявлять «мертвые» или перенапряженные участки программ и, в результате оптимизировать структуру отлаживаемой программы.

6.1.6. Интегрированная среда разработки

Совокупность программных средств, поддерживающая все этапы разработки программного обеспечения от написания исходного текста программы до ее компиляции и отладки, обеспечивающая простое и быстрое взаимодействие с другими инструментальными средствами (программным отладчиком-симулятором и программатором).

Наличие в программной оболочке эмулятора встроенного редактора, встроенного менеджера проектов и системы управления позволяет существенно облегчить работу разработчика, избавив его от множества рутинных действий. Для разработчика стирается грань между написанием программы, ее редактированием и отладкой. Переход от редактирования исходного текста к отладке и обратно происходит «прозрачно» и синхронно с активизацией соответствующих окон, менеджер проектов автоматически запускает компиляцию по мере необходимости и активизирует соответствующие окна программного интерфейса.

Столь же просто можно осуществить и переход к отладке проекта с помощью имеющегося отладчика-симулятора или приступить к «прошивке» ПЗУ отлаженной программой.

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 мультипроцессор-

ных 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.

6.2. Симуляторы

Симулятор - программное средство, способное имитировать работу микроконтроллера и его памяти. Как правило, симулятор содержит в своем составе:

- отладчик;
- модель ЦПУ и памяти.

Более продвинутые симуляторы содержат в своем составе модели встроенных периферийных устройств, таких как таймеры, порты, АЦП, системы прерываний.

Симулятор должен уметь загружать файлы программ во всех популярных форматах, максимально полно отображать информацию о состоянии ресурсов симулируемого микроконтроллера, а также предоставлять возможности по симуляции выполнения загруженной программы в различных режимах. В процессе отладки модель «выполняет» программу и на экране компьютера отображается текущее состояние модели.

Загрузив программу в симулятор, пользователь имеет возможность запускать ее в пошаговом или непрерывном режиме, задавать условные и безусловные точки останова, контролировать и свободно модифицировать содержимое ячеек памяти и регистров симулируемого микропроцессора. С помощью симулятора можно быстро проверить логику вы-

полнения программы, правильность выполнения арифметических операций.

В зависимости от класса используемого отладчика, различные симуляторы могут поддерживать высокоуровневую символьную отладку программ.

Некоторые модели симуляторов могут содержать ряд дополнительных программных средств, таких, например, как интерфейс внешней среды, встроенная интегрированная среда разработки.

В реальной системе микроконтроллер обычно занимается считыванием информации с подключенных внешних устройств (датчиков), обработкой этой информации и выдачей управляющих воздействий на исполнительные устройства. Чтобы в симуляторе, не обладающем интерфейсом внешней среды, смоделировать работу датчика, нужно вручную изменять текущее состояние модели периферийного устройства, к которому в реальной системе подключен датчик. Если, например, при приеме байта через последовательный порт взводится некоторый флажок, а сам байт попадает в определенный регистр, то оба эти действия нужно производить в таком симуляторе вручную. Наличие же интерфейса внешней среды позволяет пользователю создавать и гибко использовать модель внешней среды микроконтроллера, функционирующую и взаимодействующую с отлаживаемой программой по заданному алгоритму.

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.

6.3. Отладочные мониторы

Отладочный монитор - специальная программа, загружаемая в память отлаживаемой системы. Она вынуждает процессор пользователя производить, кроме прикладной задачи, еще и отладочные функции:

- загрузку прикладных кодов пользователя в свободную от монитора память;
- установку точек останова;
- запуск и останов загруженной программы в реальном времени;

- проход программы пользователя по шагам;
- просмотр, редактирование содержимого памяти и управляющих регистров.

Программа монитора обязательно должна работать в связке с внешним компьютером или пассивным терминалом, на которых и происходит визуализация и управление процессом отладки. Повторим, что отладочные мониторы используют тот процессор, который уже стоит на плате пользователя. Достоинством этого подхода являются очень малые затраты при сохранении возможности вести отладку в реальном времени. Главным недостатком является отвлечение ресурсов микроконтроллера на отладочные и связные процедуры, например: монитор занимает некоторый объем памяти, прерывания, последовательный канал. Объем отвлекаемых ресурсов зависит от искусства разработчика монитора. В последнее время появились изделия, которые практически не занимают аппаратных ресурсов процессора, о них будет рассказано ниже, в разделе «Эмуляторы ПЗУ».

6.4. Платы развития

Платы развития, или, как принято их называть в зарубежной литературе, оценочные платы (Evaluation Boards), являются своеобразными конструкторами для макетирования прикладных систем. В последнее время, при выпуске новой модели кристалла микроконтроллера, фирма-производитель обязательно выпускает и соответствующую плату развития. Обычно это печатная плата с установленным на ней микроконтроллером плюс вся необходимая ему стандартная обвязка. На этой плате также устанавливают схемы связи с внешним компьютером. Как правило, там же имеется свободное поле для монтажа прикладных схем пользователя. Иногда имеется уже готовая разводка для установки дополнительных устройств, рекомендуемых фирмой. Например, ПЗУ, ОЗУ, ЖКИ-дисплей, клавиатура, АЦП и др. Кроме учебных или макетных целей, такие доработанные пользователем платы стало выгодно (экономия времени) использовать в качестве одноплатных контроллеров, встраиваемых в малосерийную продукцию (5..20 шт.).

Для большего удобства платы развития комплектуются еще и простейшим средством отладки на базе монитора отладки. Однако здесь проявились два разных подхода: один используется для микроконтроллеров, имеющих внешнюю шину, а второй - для микроконтроллеров, не имеющих внешней шины.

В первом случае отладочный монитор поставляется фирмой в виде микросхемы ПЗУ, которая вставляется в специальную розетку на плате развития. Плата также имеет ОЗУ для программ пользователя и канал связи с внешним компьютером или терминалом. Примером здесь может служить плата развития фирмы Intel для микроконтроллера 8051.

Во втором случае плата развития имеет встроенные схемы программирования внутреннего ПЗУ микроконтроллера, которые управляются от внешнего компьютера. В этом случае программа монитора просто заносится в ПЗУ микроконтроллера совместно с прикладными кодами пользователя. Прикладная программа при этом специально должна быть подготовлена: в нужные ее места вставляют вызовы отладочных подпрограмм монитора. Затем осуществляется пробный прогон. Чтобы внести в программу исправления, пользователю надо стереть ПЗУ и произвести повторную запись. Готовую прикладную программу получают из отлаженной путем удаления всех вызовов мониторных функций и самого монитора отладки. Примерами могут служить платы развития фирмы Microchip для своих PIC контроллеров. Такой же принцип и у плат для отладки микроконтроллеров 80C750 Philips или 89C2051 Atmel.

Важно отметить, что иногда платы развития комплектуются еще и программами отладки, которые запускаются на внешнем компьютере в связке с монитором. Эти программы в последнее время заметно усложнились и зачастую имеют высокопрофессиональный набор отладочных функций, например, отладчик-симулятор или различные элементы, присущие в чистом виде интегрированным средам разработки. В состав поставляемых комплектов могут входить и программы прикладного характера, наиболее часто встречающиеся на практике.

Возможности по отладке, предоставляемые комплектом «плата развития плюс монитор», безусловно, не столь универсальны, как возможности внутрисхемного эмулятора, да и некоторая часть ресурсов микропроцессора в процессе отладки отбирается для работы монитора. Тем не менее наличие законченного набора готовых программно-аппаратных средств, позволяющих без потери времени приступить к монтажу и отладке прикладной системы, во многих случаях является решающим фактором. Особенно если учесть, что стоимость такого комплекта несколько меньше, чем стоимость более универсального эмулятора.

6.5. Эмуляторы ПЗУ

Эмулятор ПЗУ - программно-аппаратное средство, позволяющее замещать ПЗУ на отлаживаемой плате и подставляющее вместо него ОЗУ, в которое может быть загружена программа с компьютера через один из стандартных каналов связи. Это устройство позволяет пользователю избежать многократных циклов перепрограммирования ПЗУ. Эмулятор ПЗУ имеет смысл только для микроконтроллеров, которые в состоянии обращаться к внешней памяти программ. Это устройство сравнимо по сложности и по стоимости с платами развития. Оно имеет одно большое достоинство - универсальность. Эмулятор ПЗУ может работать с любыми типами микроконтроллеров.

Ранние эмуляторы ПЗУ позволяли только загружать программу, запускать ее и останавливать, используя общий сброс. Затем появились усложненные модели с аппаратной выработкой сигналов трассировки по достижении определенного адреса на осциллограф. Эмулируемая память в таких изделиях была доступна для просмотра и модификации, но очень важный контроль за внутренними управляющими регистрами микроконтроллера был до недавнего времени невозможен.

Однако появились модели интеллектуальных эмуляторов ПЗУ, которые позволяют «заглядывать» внутрь микроконтроллера на плате пользователя и вообще, по управлению отладкой, стали похожими на внутрисхемный эмулятор. Интеллектуальные эмуляторы ПЗУ представляют собой гибрид из обычного эмулятора ПЗУ, монитора отладки и схем быстрого переключения шины с одного на другой. Этим создается эффект, как если бы монитор отладки был установлен на плате пользователя, и при этом он не занимает у микроконтроллера никаких аппаратных ресурсов, кроме небольшой зоны программных шагов.

6.5.2. Набор ПО для МК Atmel AVR

В качестве примера рассмотрим набор программного обеспечения для микроконтроллеров Atmel.

1. AVR Studio 3.1

Достаточно удачным выбором при разработке программного обеспечения для микроконтроллеров семейства AVR может послужить интегрированная среда разработки AVR Studio 3.1 фирмы Atmel, которая включает в себя текстовый редактор с подсветкой синтаксиса, компилятор ассемблера, симулятор, отладчик и интерфейс с аппаратными эмуля-

торами. Данная система распространяется фирмой Atmel бесплатно и ее можно скачать с сайта фирмы Atmel (<http://www.atmel.com/>). Программа рассчитана на работу под управлением операционных систем Windows 9x, NT, 2000. К недостаткам AVRStudio можно отнести некоторую нестабильность работы отладчика, а также неполную симуляцию периферийных устройств (в частности, отсутствует симуляция АЦП). К плюсам же относится, в первую очередь, поддержка ее практически всех микроконтроллеров семейства AVR.

2. GNU/Linux AVR Assembler

Этот ассемблер, являясь полностью совместимым по входному коду с компилятором фирмы Atmel, имеет по сравнению с ним несколько преимуществ. Во-первых, он поставляется в исходных текстах, так что пользователь, в случае необходимости, может легко добавить новые возможности. Во-вторых, это один из немногих ассемблеров для микроконтроллеров фирмы AVR, работающих под управлением операционной системы Linux (кроме того, наличие исходных текстов позволяет легко перенести его на любую другую Unix-систему). И наконец, в GNU AVR Assembler значительно расширены, по сравнению с ассемблером фирмы Atmel, возможности по работе с макрокомандами (в частности, допускаются макросы внутри макросов). Программа распространяется по условиям GNU Public License (GPL). Домашняя страничка программы расположена по адресу <http://home.worldonline.dk/~tomo/>, а саму программу можно скачать по адресу <http://www.image.dk/~tomo/tavrasm.tar.gz> (Linux-версия), <http://www.image.dk/~tomo/tavrasm.zip> (DOS-версия) или <http://www.image.dk/~tomo/tavrasmw.zip> (консольная Windows-версия).

IAR C Compiler

Компилятор фирмы IAR общепризнанно является одним из лучших компиляторов C для микроконтроллеров семейства AVR. Связано это, в первую очередь, с наличием в нем богатых возможностей по оптимизации кода. Пожалуй, единственным существенным его недостатком является то, что в демонстрационной версии накладываются значительные ограничения на максимальный объем кода. Компилятор поставляется в составе интегрированной среды разработки IAR Embedded Workbench (EWB), включающей в себя также компилятор ассемблера, линкер, менеджер проектов и библиотек, а также отладчик. Скачать демонстрационную версию пакета можно с сайта фирмы IAR Systems, расположенного по адресу <http://www.iar.se/>.

7. ВЫБОР МИКРОКОНТРОЛЛЕРА

Цель данной главы – указать все возможные критерии выбора, которые должны быть приняты во внимание в процессе принятия решения по выбору МК.

Основная цель - выбрать наименее дорогой микроконтроллер (чтобы снизить общую стоимость системы), но в то же время удовлетворяющий спецификации системы, т.е. требованиям по производительности, надежности, условиям применения и т.д. Общая стоимость системы включает все: инженерные исследования и разработку, производство (комплектующие и труд), гарантийный ремонт, дальнейшее усовершенствование, обслуживание, совместимость, простоту в обращении и т.д.

7.1. Процесс выбора

Приступая к выбору, разработчик должен вначале задаться вопросом: «Что должен делать микроконтроллер в моей системе?» Ответ на этот простой вопрос определяет требуемые для разрабатываемой системы характеристики микроконтроллера и, таким образом, является определяющим фактором в процессе выбора.

Второй шаг – проведение поиска микроконтроллеров, которые удовлетворяют всем системным требованиям. Он обычно включает подбор литературы, технических описаний и технических журналов, а также консультации. Хорошо, если системным требованиям будет удовлетворять хорошо знакомый микроконтроллер, в противном случае должен быть проведен вторичный поиск, чтобы найти микроконтроллер, который наиболее полно удовлетворяет предъявляемые требования, имеет минимум внешних навесных компонентов и подходит по стоимости и габаритам.

Последняя стадия выбора состоит из нескольких этапов, цель которых – сузить список приемлемых микроконтроллеров до одного. Эти этапы включают в себя анализ цены, доступности, средств разработки, поддержки производителя, стабильности производства конкретных микроконтроллеров и наличия других производителей или поставщиков. Чтобы прийти к оптимальному решению, возможно, весь процесс придется повторить несколько раз.

7.2. Критерии выбора

Основные критерии выбора микроконтроллера представлены ниже в порядке значимости.

- Пригодность для прикладной системы. Может ли она быть сделана на однокристальном микроконтроллере или ее можно реализовать на основе какой либо специализированной микросхемы?
- Имеет ли микроконтроллер требуемое число контактов/портов ввода/ вывода, т.к. в случае их недостатка он не сможет выполнить работу, а в случае избытка цена будет слишком высокой.
- Имеет ли он все требуемые периферийные устройства, такие как последовательные порты ввода/вывода, RAM, ROM, A/D, D/A и т.д.?
- Имеет ли он другие периферийные устройства, которые не требуются в системе?
- Обеспечивает ли ядро процессора необходимую производительность, т.е. вычислительную мощность, позволяющую обрабатывать системные запросы в течение всей жизни системы на выбранном прикладном языке? Слишком много - расточительно, слишком мало - не будет работать.
- Выделено ли в бюджете проекта достаточно средств, чтобы позволить себе использовать данный микроконтроллер. Для ответа на этот вопрос, обычно требуются расценки поставщика. Если данный микроконтроллер не приемлем для проекта, все остальные вопросы становятся несущественными и вы должны начать поиски другого микроконтроллера.
- Доступность. Существует ли устройство в достаточных количествах? Производится ли оно сейчас? Что ожидается в будущем?
- Поддержка разработчика: ассемблеры, компиляторы, средства отладки (оценочный модуль, внутрисхемные эмуляторы, отладочные мониторы).
- Информационная поддержка: примеры применения, сообщения об ошибках, утилиты, в том числе «бесплатные» ассемблеры, примеры исходных текстов, поддержка применений у поставщика.

7.3. Системные требования

Проведение системного анализа вашего проекта позволит определить и требования к микроконтроллеру. Какие требуются периферийные устройства? Должна ли система управляться по прерываниям, по готов-

ности или по командам человека? Каким количеством устройств (битов ввода/вывода) необходимо управлять? Какие устройства из числа многих возможных типов должны контролироваться и управляться? Одно или несколько напряжений питания требуется для системы? Насколько отказоустойчив источник питания? Будет ли работать устройство при напряжении вашей сети питания? Должны ли напряжения удерживаться в узком фиксированном диапазоне изменений или же система может работать при большой нестабильности? Какой рабочий ток? Изделие должно работать от сети или от батарей? Если от батарей, должны ли использоваться перезаряжаемые батареи, и если это так, то каково время работы без перезарядки и какое для нее требуется время? Существуют ли ограничения по размеру, весу, эстетическим параметрам, таким как форма и/или цвет? Существуют ли какие-либо специфические требования к условиям окружающей среды, таким как военные условия, температура, влажность, атмосфера (взрывоопасная, коррозионная и т.д.), давление? Пользовательское программное обеспечение должно базироваться на дисках или ROM? Изделие работает в реальном времени, и если да, собираетесь ли вы создать или приобрести ядро программ реального времени или, возможно, будет достаточно обычной широко используемой версии? Достаточно ли персонала и времени для развития вашего собственного ядра программ? Как будут оплачиваться авторские права и программное обеспечение? Для решения задач реального времени требуется большая исследовательская работа, чтобы удовлетворить их особым требованиям.

7.4. Основные особенности микроконтроллера

Способен ли дешевый микроконтроллер удовлетворить требованиям системы или требуется дорогой 16- или 32-разрядный? Выбор прикладного языка (высокого уровня вместо ассемблера) может сильно повлиять на производительность системы, которая затем может диктовать выбор 8/16/32-разрядной архитектуры, но ограничение по цене может отвергнуть этот выбор.

Тактовая частота или, более точно, скорость шины определяет, сколько вычислений может быть выполнено за единицу времени. Некоторые микроконтроллеры, в основном ранних разработок имеют узкий диапазон допустимой тактовой частоты, в то время как другие могут работать вплоть до нулевой частоты. Иногда выбирается специфическая тактовая частота, чтобы сгенерировать другую тактовую частоту, требуемую в системе, например, для задания скоростей последовательной пере-

дачи. В основном вычислительная мощность, потребляемая мощность и стоимость системы увеличиваются с повышением тактовой частоты. Цена системы при повышении частоты увеличивается из-за стоимости не только микроконтроллера, но также и всех требующихся дополнительных микросхем, таких как RAM, ROM, PLD и контроллеры шины.

7.5. Возможности микроконтроллера

За счет достижения более высокого уровня интеграции и надежности при сохранении низкой цены, все микроконтроллеры оснащены встроенными дополнительными устройствами. Эти устройства под управлением микропроцессорного ядра микроконтроллера выполняют определенные функции. Встроенные устройства обладают повышенной надежностью, поскольку они не требуют никаких внешних электрических цепей. К наиболее известным встроенным устройствам относятся устройства памяти и порты ввода/вывода (I/O), таймеры, системные часы/генератор.

Устройства памяти включают оперативную память (RAM), постоянные запоминающие устройства (ROM), перепрограммируемую ROM (EPROM), электрически перепрограммируемую ROM (EEPROM). Таймеры включают и часы реального времени, и таймеры прерываний. Следует принимать во внимание диапазон и разрешение таймера. Средства I/O включают последовательные порты связи, параллельные порты (I/O линии), аналого-цифровые преобразователи (A/D), цифроаналоговые преобразователи (D/A), драйверы жидкокристаллического дисплея (LCD) или драйверы вакуумного флуоресцентного дисплея (VFD). Другими, реже используемыми, встроенными ресурсами являются внутренняя/внешняя шина, таймер слежения за нормальным функционированием системы сторожевая схема, система обнаружения отказов тактового генератора, возможность выбора конфигурации памяти и системный интеграционный модуль.

7.6. Набор команд микроконтроллера

Необходимо внимательно изучить набор команд и регистров каждого микроконтроллера, так как они играют важнейшую роль в определении возможностей системы в целом. Изучили ли ваши программисты индексные режимы адресации в связи с предполагаемыми нуждами вашей системы? Есть ли какие-либо специальные команды, которые будут использоваться в вашей системе, такие как умножение, деление и таб-

личное интерполирование? Есть ли какие-либо режимы энергосбережения для экономии батарейного питания, такие как стоповый, стоповый с низким потреблением мощности и/или с ожиданием? Есть ли какие-либо команды битовых манипуляций (установка бита, очистка бита, тест бита, изменение бита, команды перехода по установленному/очищенному биту), облегчающие применение микроконтроллера, или команды манипуляции с битовыми полями? Будьте осторожны с замечательными командами, которые совершают много действий в одной команде. Реальным критерием производительности является количество тактовых циклов, требуемое для выполнения задачи, а не количество исполненных команд. Для справедливого сравнения лучше закодировать одинаковую программу и сравнить полное число выполненных тактовых циклов и использованных байтов. Есть ли в карте операционных кодов нереализованные инструкции и что получится, если они случайно выполнятся? Обработает ли система подобную ситуацию корректно обработчиком «исключительных» событий или это приведет к выходу системы из строя?

7.7. Прерывания микроконтроллера

Проверка структуры прерываний необходима всегда, когда создается система реального времени. Сколько линий или уровней прерывания имеется и сколько их требуется для вашей системы? Имеется ли маска уровней прерывания? Когда уровень прерывания подтвержден, есть ли индивидуальные векторы для программы обработчика прерывания или должны опрашиваться все возможные источники прерывания, чтобы определить источник? В критических по скорости применениях, таких как управление принтером, критерием выбора подходящего микроконтроллера может быть время реакции на прерывание, т.е. время от начала прерывания до выполнения первой команды соответствующего обработчика прерывания.

7.8. Характеристика поставщика

Третий шаг в сокращении списка технически приемлемых микроконтроллеров - проверка производителей и поставщиков микроконтроллеров, т.е. компаний, с которыми вы планируете вступить в длительные отношения на взаимовыгодной основе. Поставщик может быть производителем микроконтроллеров, или он может быть дилером, который является полномочным представителем нескольких производителей. Наи-

лучшим образом удовлетворит ваши запросы поставщик с более широким ассортиментом продуктов и репутацией высокого качества, надежности, обслуживания и своевременной поставки при справедливой цене. Кроме того, чем больше продуктов вы покупаете у одного поставщика, тем большие преимущества вы получаете в отношении цены, услуг и поддержки. Всегда имейте в виду, что, хотя долларовый объем вашей покупки может казаться вам высоким, это всегда относительная величина к общему объему продаж поставщика. Поставщики, которые снабжают не только микроконтроллерами, но и памятью (RAM, ROM), дискретными устройствами (транзисторами, диодами и т.д.), стандартными цифровыми логическими устройствами (7400, 74HC00 и т.д.), специальными микросхемами, заказными приборами (CSIC), специализированными микросхемами (ASIC) и программируемыми логическими устройствами (PLD), смогут лучше удовлетворить ваши растущие запросы.

7.9. Характеристика производителя

Другими критериями в выборе производителя/поставщика микроконтроллера являются стабильность, его монопольное положение, сведения из литературы и поддержка. Стабильность может быть надежно проверена путем установления стажа работы производителя в этой области и его достижений. Отдел снабжения и кредитный отдел вашей компании могут помочь вам в этих вопросах. Монопольное положение поставщика, к сожалению, обычно норма, т.к. большинство производителей микроконтроллеров редко пересекаются в производстве с другими производителями. Если производитель имеет хорошие показатели в снабжении, доставке и цене, то его монопольное положение не должно являться препятствием.

7.10. Подведение итогов выбора микроконтроллера

Для окончательного шага в процессе выбора постройте таблицу, содержащую рассматриваемые микроконтроллеры в одной графе, а их важные характеристики - в другой. Затем приложите бланки технических описаний производителей, чтобы получить справедливое наглядное сравнение. Некоторые производители имеют предварительно сделанные сравнительные описания их микроконтроллеров, которые упростят вашу задачу, но проверьте по техническим описаниям, все ли новейшие продукты представлены.

Критерии оценки при выборе микропроцессора можно разбить на три группы:

- технические характеристики;
- эксплуатационные характеристики;
- потребительские свойства.

При этом критериями выбора считать потребительские свойства, т.е. соотношение показателей затраты/производительность/надежность, а технические и эксплуатационные характеристики – ограничениями для процедуры выбора.

Кроме того, необходимо разделить характеристики на прямые, для которых положительным результатом является их увеличение, и обратные для которых положительным результатом является ее уменьшение.

1. Технические характеристики:

- 1.1. Производительность (время выполнения одной команды);
- 1.2. Разрядность (АЛУ, внутренней шины данных и адреса);
- 1.3. Наличие и типы прерываний;
- 1.4. Размер и типы памяти программ и данных;
- 1.5. Количество и электрические характеристики линий ввода-вывода;
- 1.6. Наличие встроенной периферии: таймеры, счетчики событий, АЦП, ЦАП, ШИМ, RTC, параллельные и последовательные (синхронные, асинхронные) порты, компаратор, ЖКИ-драйверы;
- 1.7. Энергопотребление (режимы энергопотребления).

2. Эксплуатационные характеристики:

- 2.1. Диапазон рабочих температур;
- 2.2. Устойчивость к ЭМИ;
- 2.3. Размеры и тип корпуса.

3. Потребительские свойства:

- 3.1. Функциональность;
- 3.2. Поддержка технологических языков программирования;
- 3.3. Надежность – проходят ли данные МК выходной контроль на фирме производителя, отзывы о работе других разработчиков;

3.4. Затраты - Стоимость приобретения, доставки. Стоимость программно-аппаратного комплекса для разработки и отладки программ на данном МК.

8. СОВРЕМЕННОЕ СОСТОЯНИЕ РЫНКА МИКРОКОНТРОЛЛЕРОВ

Таблица 8.1. Популярные в России семейства современных
микроконтроллеров

8-разрядные	16-разрядные	32 –разрядные и более
Atmel AVR	80186	ARM processors
Intel 8051, MCS 151/251	Hitachi H8	80386, 80486
Microchip PICmicro family	Infineon 16x	Fujitsu SPARC-lite
Mitsubishi 740	Intel MCS-96/296	Hitachi SuperH Series
Motorola HC05, HCC06, HC11	Mitsubishi MELPS7700	Hyperstone E1-32X
Ангстрем Тесей	Mitsubishi M16C	IBM/Motorola PowerPC
National Semiconductor COP8	Motorola 68HC12	Infineon Tricore
NEC K Series	Motorola 68HC16	Intel i960
Oki nX 65K	Philips 80C51XA	MIPS R3000
Scenix SX	Toshiba TLCS-900	Mitsubishi M32Rx/D
STMicroelectronics ST6	Zilog Z80/80180/80380	Motorola M CORE, 683xx, 680x0
STMicroelectronics ST9		NEC V800 - Socket 7 processors
Texas Instruments TMS370		STMicroelectronics ST100
Toshiba TLCS-90		Sun microSPARC
Xemics CoolRISC		Tensilica Xtensa
Zilog Z8/Samsung SAM8X		MIPS R4xxx, R5xxx

8.1. Краткий обзор микроконтроллеров

В данном разделе приведена краткая информация о современных 8-разрядных микроконтроллерах широко распространенных в России.

RISC: Atmel, PIC, SX, Ангстрем.

CISC: Motorola, Zilog, Samsung.

8.2. ATMEL AVR

Компания ATMEL Corp. - один из мировых лидеров в производстве широкого спектра микросхем энергонезависимой памяти, FLASH-микроконтроллеров и микросхем программируемой логики, взяла старт

по разработке RISC-микроконтроллеров в середине 90-х годов, используя все свои технические решения, накопленные к этому времени.

Концепция новых скоростных микроконтроллеров была разработана группой разработчиков исследовательского центра ATMEL в Норвегии, инициалы которых затем сформировали марку AVR. Первые микроконтроллеры AVR AT90S1200 появились в середине 1997 г. и быстро снискали расположение потребителей.

AVR-архитектура, на основе которой построены микроконтроллеры семейства AT90S, объединяет мощный гарвардский RISC-процессор с отдельным доступом к памяти программ и данных, 32 регистра общего назначения, каждый из которых может работать как регистр-аккумулятор, и развитую систему команд фиксированной 16-бит длины. Большинство команд выполняются за один машинный такт с одновременным исполнением текущей и выборкой следующей команды, что обеспечивает производительность до 1 MIPS на каждый МГц тактовой частоты.

32 регистра общего назначения образуют регистровый файл быстрого доступа, где каждый регистр напрямую связан с АЛУ. За один такт из регистрового файла выбираются два операнда, выполняется операция, и результат возвращается в регистровый файл. АЛУ поддерживает арифметические и логические операции с регистрами, между регистром и константой или непосредственно с регистром.

Регистровый файл также доступен как часть памяти данных. 6 из 32-х регистров могут использоваться как три 16-разрядных регистра-указателя для косвенной адресации. Старшие микроконтроллеры семейства AVR имеют в составе АЛУ аппаратный умножитель.

Базовый набор команд AVR содержит 120 инструкций. Инструкции битовых операций включают инструкции установки, очистки и тестирования битов.

Все микроконтроллеры AVR имеют встроенную FLASH ROM с возможностью внутрисхемного программирования через последовательный 4-проводной интерфейс.

Периферия МК AVR включает: таймеры-счетчики, широтно-импульсные модуляторы, поддержку внешних прерываний, аналоговые компараторы, 10-разрядный 8-канальный АЦП, параллельные порты (от 3 до 48 линий ввода и вывода), интерфейсы UART и SPI, сторожевой таймер и устройство сброса по включению питания. Все эти качества превращают AVR-микроконтроллеры в мощный инструмент для построения современных, высокопроизводительных и экономичных контроллеров различного назначения.

В рамках единой базовой архитектуры AVR-микроконтроллеры подразделяются на три подсемейства:

1) Classic AVR - основная линия микроконтроллеров с производительностью отдельных модификаций до 16 MIPS, FLASH ROM-программ 2-8 Кбайт, EEPROM-данных 64-512 байт, SRAM 128-512 байт;

2) mega AVR с производительностью 4-6 MIPS для сложных приложений, требующих большого объема памяти, FLASH ROM-программ 64-128 Кбайт, EEPROM-данных 64-512 байт, SRAM 2-4 Кбайт, SRAM 4 Кбайт, встроенный 10-разрядный 8-канальный АЦП, аппаратный умножитель 8 \times 8;

3) tiny AVR - низкостоимостные микроконтроллеры в 8-выводном исполнении имеют встроенную схему контроля напряжения питания, что позволяет обойтись без внешних супервизорных микросхем.

AVR-микроконтроллеры поддерживают спящий режим и режим микропотребления. В спящем режиме останавливается центральное процессорное ядро, в то время как регистры, таймеры-счетчики, сторожевой таймер и система прерываний продолжают функционировать. В режиме микропотребления сохраняется содержимое всех регистров, останавливается тактовый генератор, запрещаются все функции микроконтроллера, пока не поступит сигнал внешнего прерывания или аппаратного сброса. В зависимости от модели AVR-микроконтроллеры работают в диапазоне напряжений 2,7-6 В либо 4-6 В (исключение составляет ATtiny12V с напряжением питания 1,2 В).

Средства отладки. ATMEL предлагает программную среду AVR-studio для отладки программ в режиме симуляции на программном отладчике, а также для работы непосредственно с внутрисхемным эмулятором. AVR-studio доступен с WEB-страницы ATMEL, содержит ассемблер и предназначен для работы с эмуляторами ICEPRO и MegaICE. Ряд компаний предлагают свои версии Си-компиляторов, ассемблеров, линковщиков и загрузчиков для работы с микроконтроллерами семейства AVR.

Микроконтроллеры ATMEL широко применяются в России и, как следствие, программируются многими отечественными программаторами. Ряд российских фирм предлагает также различные аппаратные средства отладки AVR-микроконтроллеров.

8.3. MICROCHIP семейство PIC

Первые микроконтроллеры компании MICROCHIP PIC16C5x появились в конце 80-х годов и благодаря своей высокой производительности

сти и низкой стоимости составили серьезную конкуренцию производимым в то время 8-разрядным МК с CISC-архитектурой.

Первое, что привлекает внимание в PIC-контроллерах - это простота и эффективность. В основу концепции PIC, единую для всех выпускаемых семейств, была положена RISC-архитектура с системой простых однословных команд, применение встроенной памяти программ и данных и малое энергопотребление.

Система команд базового семейства PIC165x содержит только 33 команды. Как ни странно, и это сыграло свою роль в популяризации PIC-контроллеров. Все команды (кроме команд перехода) выполняются за один машинный цикл (или четыре машинных такта) с перекрытием по времени выборок команд и их исполнения, что позволяет достичь производительности до 5 MIPS при тактовой частоте 20 МГц.

Микроконтроллеры PIC имеют симметричную систему команд, позволяющую выполнять операции с любым регистром, используя любой метод адресации. Правда, разработчики MICROCHIP так и не смогли отказаться от любимой всеми структуры с регистром-аккумулятором необходимым участником всех операций с двумя операндами. Зато теперь пользователь может сохранять результат операции на выбор, где пожелает, в самом регистре-аккумуляторе или во втором регистре, используемом для операции. В настоящее время MICROCHIP выпускает четыре основных семейства 8-разрядных RISC-микроконтроллеров, совместимых снизу вверх по программному коду:

- базовое семейство PIC15Cx с 12-разрядными командами, простые недорогие микроконтроллеры с минимальной периферией;
- PIC12Cxxx с 12-разрядными командами со встроенным тактовым генератором, выпускаемые в миниатюрном 8-выводном исполнении. Не так давно был анонсирован очередной такой «малыш» с внутренним 8-разрядным 4-канальным АЦП;
- Mid-range PIC16x/7x/8x/9x с 14-разрядными командами. Наиболее многочисленное семейство, объединяющее микроконтроллеры с разнообразными периферийными устройствами, в число которых входят аналоговые компараторы, аналогово-цифровые преобразователи, контроллеры последовательных интерфейсов SPI, USART и I2C, таймеры-счетчики, модули захвата/сравнения, широтно-импульсные модуляторы, сторожевые таймеры, супервизорные схемы и так далее;
- High-end PIC17C4x/5xx высокопроизводительные микроконтроллеры с расширенной системой команд 16-разрядного формата, работающие на частоте до 33 МГц, с объемом памяти программ до 16 Кслов. Кроме обширной периферии, почти все микроконтроллеры

этого семейства имеют встроенный аппаратный умножитель 8×8 , выполняющий операцию умножения за один машинный цикл.

Большинство PIC-контроллеров выпускаются с однократно программируемой памятью программ OTP с возможностью внутрисхемного программирования или масочным ROM. Для целей отладки предлагаются версии с ультрафиолетовым стиранием, надо признать, не очень дешевые. Полное количество выпускаемых модификаций PIC-контроллеров составляет порядка пятисот наименований. Как не без основания утверждает MICROCHIP, продукция компании покрывает весь диапазон применений 8-разрядных микроконтроллеров.

Особый акцент MICROCHIP делает на максимально возможное снижение энергопотребления для выпускаемых микроконтроллеров. При работе на частоте 4 МГц PIC-контроллеры, в зависимости от модели, имеют ток потребления меньше 1,5 мА, а при работе на частоте 32,768 КГц - ниже 15 мкА. Поддерживается «спящий» режим работы. Диапазон питающих напряжений PIC-контроллеров составляет 2,0...6,0 В.

В настоящее время готовится к запуску в производство новое пятое семейство PIC-контроллеров PIC18Cxxx. Новые микроконтроллеры будут иметь расширенное RISC-ядро, оптимизированное под использование нового Си-компилятора, адресное пространство программ до 2 Мбайт, до 4 Кбайт встроенной памяти данных и производительность 10 MIPS.

Из программных средств отладки наиболее известны и доступны различные версии ассемблеров, а также интегрированная программная среда MPLAB. Российские производители программаторов и аппаратных отладочных средств также уделяют внимание PIC-контроллерам. Выпускаются как специализированные программаторы, такие как PICPROG, программирующие почти весь спектр PIC-микроконтроллеров, так и универсальные: UNIPRO, STEPX, поддерживающие наиболее известные версии PIC.

8.4. Scenix SX

Следуя известному философскому принципу не умножать сущностей сверх необходимого, американская компания Scenix Semiconductors не стала изобретать принципиально новый микроконтроллер со своей системой команд, а решила кардинально переработать и улучшить уже упоминавшееся семейство PIC16Cx MICROCHIP. Scenix имеет патент на быструю флэш-технология. Это позволило создать микроконтроллеры с производительностью более 50 MIPS, анонсированные в декабре

1997 года. Еще через год появилась версия с удвоенной производительностью - 100 MIPS!

Новые микроконтроллеры SC18/20/28AC100 совместимы программно снизу вверх и аппаратно pin-to-pin с PIC16C5x и при этом предоставляют пользователю гораздо больше возможностей при ненамного большей стоимости.

RISC-ядро SX-процессора обеспечивает выполнение большинства 12-разрядных инструкций за один машинный цикл длительностью 10 нс на тактовой частоте 100 МГц. 4-уровневый конвейер команд позволяет одновременно выполнять выборку команды, декодирование, исполнение, запись результата одновременно для четырех последовательно расположенных команд. Новой особенностью процессора также является аппаратное сохранение содержимого служебных регистров и обратная загрузка для ускорения обработки прерываний без потери машинных циклов.

Кроме своей исключительной производительности, микроконтроллеры Scenix обладают следующими дополнительными возможностями по сравнению с PIC16C5x:

- расширенная система команд: 33 (как у PIC16C5x) плюс 10 дополнительных;
- у SX48BD/SX52BD 4096×12 FLASH-память программ с возможностью внутрисхемного программирования; внутрисхемная отладка на максимальной рабочей частоте, не требующая дорогих эмуляторов;
- 256 байт встроенного статического ОЗУ; два 16-разрядных универсальных таймера плюс один 8-разрядный; быстрый аналоговый компаратор; внутренний RC-генератор с предделителем; поддержка внешних прерываний со временем отклика 50 нс;
- сильноточные порты ввода/вывода (30 мА); расширенная схема сброса при пропадании питания; отсутствие «дрожания» фронта сигнала при обработке прерывания; возможность работы в режиме отладки программного кода с доступом к содержимому регистров и стека.

Наиболее же интересным моментом, пожалуй, является концепция Виртуальной Периферии - программной эмуляции периферийных устройств, активно разрабатываемой Scenix Sem. В ее основе заложены быстрые прерывания у SX-микроконтроллеров.

Многим разработчикам знакомы проблемы настойчивого поиска микроконтроллера, обладающего необходимым количеством таймеров, АЦП, ШИМ и контроллеров различных последовательных интерфейсов. И чтобы все это размещалось на одном кристалле.

Стандартный ответ компаний-производителей - размещение на кристалле как можно большего числа периферийных модулей для потенциального использования - приводит к необходимости решения многих технологических проблем (потребление, отвод тепла и так далее) и к значительному удорожанию кристалла.

В этом случае очень изящно выглядит решение Scenix Sem. - осуществлять программную эмуляцию периферийных компонент на базе очень быстрого микроконтроллера при посредстве стандартной библиотеки программных модулей. В настоящее время библиотека модулей насчитывает несколько десятков разновидностей, некоторые из них: АЦП, ШИМ, интерфейсы UART, I2C и т.д.

Доступен широкий спектр средств программирования и отладки для микроконтроллеров SX. Выпущено несколько коммерческих и бесплатных компиляторов Си, программных симуляторов, ассемблеров, программаторов. Постоянно растут списки примеров применения виртуальной периферии: Modem 1200 бод, IRDA-интерфейс, TCP/IP - делает доступным SX-микроконтроллер в сети Internet; а скоро - USB, CAN-протоколы.

В 2000 году компанией запланирован выпуск усовершенствованных микроконтроллеров SX48BD/SX52BD с расширенной памятью программ и данных.

8.5. Ангстрем

Разработчика отечественного производителя ОАО «АНГСТРЕМ» - микроконтроллерное RISC-ядро ТЕСЕЙ, на базе которого уже создано несколько микроконтроллеров, как универсальных, так и специализированного применения.

Характерной особенностью ядра ТЕСЕЙ являются:

- гарвардская RISC-архитектура, позволяющая выполнять любую из 52 команд 16-разрядного формата за два такта частоты процессора;
- единая система команд для всего семейства с возможностью адресации до двух операндов, находящихся в памяти;
- 4-ступенчатый конвейер выполнения команд;
- малое время отклика на прерывание и сохранение контекста;
- широкий диапазон конфигураций внутренних памяти команд, памяти данных и периферийных устройств.

Микроконтроллеры отличаются наличием энергонезависимой памяти данных, возможности многократного перепрограммирования памя-

ти программ, небольшим количеством внешних выводов и низким энергопотреблением.

Микроконтроллер КР1878ВЕ1 является представителем семейства ТЕСЕЙ и обладает следующими характеристиками:

- производительность до 4 MIPS на тактовой частоте 8 МГц;
- перепрограммируемая память программ 1К \times 16;
- память данных RAM 128 \times 8, EEPROM данных 64 \times 8;
- периферия: сторожевой таймер, 16-бит таймер-счетчик с предделителем, 12 линий ввода/вывода и поддержка прерываний;
- ток потребления меньше 2 мА при напряжении питания 5 В и тактовой частоте 5 МГц.

Микроконтроллер выпускается в 18-выводном исполнении.

8.6. MOTOROLA. 68HC05, 68HC08, 68HC11

История развития семейств MC68HC05/08/11 начинается с базового микропроцессора MC6800, выпущенного еще в 1974 году. Первые микроконтроллеры HC05 представляли собой типичные микропроцессоры 6800 с модифицированной архитектурой, пересаженные на «микроконтроллерную» почву.

Последующее продвижение MOTOROLA происходило в направлении совершенствования процессорного ядра «дедушки» (семейство HC11, основанное на микропроцессорной архитектуре II поколения MC6809, и семейство HC08 с усовершенствованным процессорным ядром MC6800) и ускорения работы самого процессора (семейство HC11 и, особенно, HC08).

MC68HC05

Основанный на фон-неймановской архитектуре, простой, по современным понятиям, процессор с классической аккумуляторной архитектурой и индексным регистром адресует до 64 Кбайт памяти и имеет укороченную систему команд микропроцессора 6800, состоящую в общей сложности из 65 инструкций. Определенным минусом, безусловно, является отсутствие регистров общего назначения для быстрых регистровых операций и соответственно работающих с ними команд.

Длительность исполнения большинства команд составляет от 2 до 6 машинных циклов или от 1 до 3 мкс при внутренней тактовой частоте 2 МГц. Исключением являются несколько команд, таких как, к примеру,

команда умножения 8х8, занимающая 11 машинных циклов, и команды, связанные с обработкой прерываний.

Из системы команд заслуживают упоминания команды битовых операций, установка и сброс бита во внутреннем ОЗУ и команды перехода по состоянию бита.

Номенклатура периферийных устройств зависит от выпускаемой серии, обозначаемой специальной литерой в названии микроконтроллера. В базовых моделях применяется простой 15-разрядный без опций запуска/останова таймер с блоком периодических прерываний, на основе которого также реализуется сторожевой таймер. Во многих моделях микроконтроллеров таймерный модуль строится на основе 16-разрядного free-running таймера и имеет также дополнительные каналы входного захвата и выходного сравнения. Важной особенностью является аппаратная поддержка внешних запросов прерываний. Даже самые простые и недорогие микроконтроллеры обеспечивают поддержку до пяти внешних прерываний, использующих, правда, один и тот же вектор прерывания. В рамках программы «No Excuses» поставляются микроконтроллеры серий В, С, J, К, L, Р.

Микроконтроллеры серии С общего назначения содержат таймер, 4 параллельных порта, SCI и SPI.

Дешевые МК серии J и К с минимальной периферией выпускаются в 16- и 20-выводном исполнении. Микроконтроллеры серии Р, производимые в 28-выводных корпусах, добавляют к этому последовательный порт, а также 8-разрядный АЦП.

Серия В обладает полным набором периферийных устройств – многоканальный АЦП, ШИМ, последовательный порт SCI.

Микроконтроллеры с литерой L предназначены для работы с ЖК-дисплеями.

Основной объем внутренней памяти занимает однократно программируемое ПЗУ емкостью от 1232 байт до 16 Кбайт с опцией защиты от чтения. Емкость внутреннего ОЗУ, в зависимости от модели, составляет от 64 до 512 байт. Микроконтроллеры серии В имеют в дополнение к этому EEPROM емкостью 256 байт. К сожалению, МК HC705, имея схему сброса по включению питания, не имеют таковой схемы по понижению напряжения питания, что требует применения внешних супервизорных схем во избежание повреждения данных EEPROM.

Стандартный диапазон напряжения питания микроконтроллеров HC05 составляет 3–5 В. Для обеспечения пониженного энергопотребления реализована возможность программного перехода в режим ожидания и останова.

Для поставляемых микроконтроллеров HC705 предлагаются специальные наборы, содержащие программатор и внутрисхемный симулятор по отпускной цене MOTOROLA от 99\$.

MC68HC08

Это семейство, первые представители которого появились в 1994 году, представляет собой эволюционное продолжение 68HC05 с полностью статической архитектурой, повышенной производительностью и расширенными функциональными возможностями.

Некоторые изменения коснулись процессорного модуля. Разрядность регистра стека и индексного регистра увеличена до 16 бит, в регистр состояния введен дополнительный разряд – флаг переполнения. Процессор работает на увеличенной до 8 МГц внутренней тактовой частоте – машинный цикл 125 нс! Для повышения производительности в процессорное ядро введен специальный регистр с романтическим названием «LookAhead», в который заносится операционный код следующей команды во время исполнения текущей (одноуровневый конвейер), что позволяет экономить лишний машинный цикл.

По сравнению с HC05, появились новые способы адресации, такие как адресация с постинкрементом и адресация память–память, значительно повысившие эффективность работы с массивами данных.

Также претерпела изменения система команд. Были введены специальные команды для работы со стеком (PUSH, POP), которых так не хватало в базовом HC05. Появились дополнительные арифметические команды: деление, десятичная коррекция, суммирование операндов с содержимым индексного регистра и стека, команды управления: ветвление по расширенному условию, а также новые команды пересылки. Появился ряд новых периферийных устройств:

- контроллер прямого доступа к памяти ПДП с тремя независимыми каналами, реализующий обмен блоками данных между памятью и внешними устройствами без участия процессора;
- 12-разрядный 6-канальный ШИМ-контроллер;
- модуль расширения адресации внешней памяти до 16 Мбайт.

Таймерный блок в основе своей аналогичен существующему в HC05, но обладает также рядом дополнительных функций, таких как реализация режима широтно-импульсной модуляции и возможность тактирования таймера внешним сигналом. Наконец-то появилась возможность программного включения/выключения таймера для подстройки временной базы.

Модуль системной интеграции обеспечивает расширенные возможности по контролю функционирования МК. Сброс микроконтроллера производится по включению питания, сигналу от сторожевого таймера, поступлении неправильного кода команды, формировании некорректного адреса и, что очень важно, при снижении напряжения питания ниже допустимого уровня.

Модули встроенной памяти микроконтроллеров семейства MC68HC08 содержат ПЗУ или ППЗУ емкостью от 16 до 56 Кбайт, ОЗУ емкостью от 128 до 1024 байт и EEPROM емкостью 512 байт.

В настоящее время поставляются микроконтроллеры серий MC68HC908JL3/JK3/JK1 со встроенным FLASH ПЗУ емкостью от 1536 байт до 4 Кбайт.

MC68C11

MC68C11 – полностью статические 8-разрядные микроконтроллеры, программно совместимые с 6800 и являющиеся связующим звеном (совместимость по исходному коду) между 8-бит МК MOTOROLA и их более мощными 16-разрядными родственниками MC68C16.

Основные отличия процессорного модуля HC11 от HC05 – присутствие двух 8-разрядных регистров-аккумуляторов (работающих отдельно или как совместная регистровая пара) и двух 16-разрядных индексных регистров. В зависимости от модели используемая внутренняя частота процессора составляет от 2 до 4 МГц. Система команд состоит из 108 инструкций длиной от 1 до 4 байт. Многие команды аналогичны существующим командам HC05, основные отличия связаны с появлением дополнительных рабочих регистров процессора. Наиболее интересны команды 16-бит арифметических операций: сложение, вычитание, целочисленное и дробное деление, инкремент, сдвиг 16-разрядных операндов, а также расширенные команды битовых операций с возможностью установки, сброса и тестирования сразу нескольких битов.

Микроконтроллеры HC11 позволяют адресовать как внутреннюю память (однокристалльный режим), так и внешнюю (расширенный режим). Некоторые версии позволяют адресовать до 1 Мбайт внешней памяти, используя страничную адресацию.

Семейство HC11 отличается большим разнообразием периферийных устройств. Таймерный блок, используемый в МК HC11, по своей структуре и функциям также аналогичен таймерному модулю HC05, но имеет дополнительные черты, такие как вспомогательный 8-бит таймер-счетчик с опцией программного включения/выключения, увеличенное

число каналов захвата/сравнения, возможность вывода заранее запрограммированного кода на линии параллельного порта ввода/вывода в определенные моменты времени.

Почти все микроконтроллеры имеют в своем составе встроенный 8-разрядный АЦП последовательных приближений. Время полного преобразования 8-разрядного АЦП занимает 32 машинных цикла на один канал.

Система последовательного ввода/вывода включает в себя асинхронный SCI и синхронный SPI.

Из прочих периферийных устройств, имеющих в составе ряда микроконтроллеров HC11, можно выделить 4-канальный ПДП, арифметический сопроцессор, обеспечивающий десятикратное ускорение операций деления и умножения, 6-канальный ШИМ.

Объем внутреннего ПЗУ (масочного или ОТР), в зависимости от версии, составляет от 4 до 48 Кбайт, ОЗУ – от 256 до 1278 байт. Некоторые версии содержат EEPROM емкостью до 2 Кбайт. Стандартное напряжение питания микроконтроллеров составляет 5 В, доступны также версии с пониженным напряжением питания. Недостатками микроконтроллеров HC11 можно считать большое энергопотребление (до 150–300 МВт) и относительно невысокую на сегодняшний день производительность для универсального МК.

Для семейств MC68HC05/08/11 предлагается большое количество ПО, включающее интегрированные отладочные среды, ассемблеры, Си-компиляторы, отладчики, программные симуляторы от множества третьих поставщиков (правда, в большинстве своем, ПО платное).

На сегодняшний день, в ряде крупных городов России сформировано более десяти учебно-методических и консультационных центров MOTOROLA, где можно получить исчерпывающую техническую информацию относительно продукции компании.

8.7. Z8 ZiLOG

История семейства универсальных микроконтроллеров Z8 начинается с легендарного микропроцессора Z80. Именно его процессорное ядро легло в основу новых (по тем временам) микроконтроллеров.

Процессор Z8 адресует три независимых области памяти: регистровый файл, память программ, память данных.

Регистровый файл может содержать до 256 8-разрядных регистров, из них до 239 универсальных рабочих регистров, каждый из которых может использоваться как аккумулятор, указатель адреса, индексный ре-

гистр, регистр данных или стека. Регистры специального назначения и периферийные регистры также доступны через регистровый файл. 16-разрядный регистр указателя стека может размещаться либо в регистровом файле, либо во внешнем ОЗУ (при его наличии). Адресное пространство рабочих регистров делится на банки по 16 регистров. Специальный регистровый указатель определяет текущую активную группу рабочих регистров. Это позволяет существенно сократить длину команд, использующих рабочие регистры.

50 базовых команд требуют для своего исполнения различное количество машинных тактов.

Внутренняя частота процессора Z8 может варьироваться от 4 до 8 МГц, в зависимости от модели. Соответственно для самых быстрых версий минимальная длительность выполнения команд, таких как регистровые операции, составляет 0,75 мкс, а максимальная (команды управления) – 2,5 мкс.

В дополнение к регистровому файлу, некоторые члены семейства имеют в своем составе расширенный регистровый файл для управления дополнительными периферийными устройствами и портами ввода/вывода.

Для повышения производительности процессор Z8 использует одноуровневую конвейеризацию команд. Правда, необходимо отметить, что из-за разной длины инструкций конвейер работает не самым оптимальным образом.

Z8 адресует до 64 Кбайт внешнего ОЗУ памяти данных и 64 Кбайт внешнего ПЗУ памяти программ. При отключении или отсутствии внутреннего ПЗУ, микроконтроллеры Z8 общаются с внешним миром через мультиплексированную шину адрес/данные.

В рамках одной архитектуры микроконтроллеры Z8 можно разделить на три группы, выпускаемые в виде версий с масочным или однократно программируемым ПЗУ или, что встречается редко, без ПЗУ вообще.

Диапазон напряжения питания, в зависимости от версии микроконтроллера (MASK ROM, OTP, ROMless, промышленный или коммерческий диапазон температур), составляет в среднем 3,0–5,0 В. Однократно программируемые микроконтроллеры начинают работать при более высоком значении напряжения питания. Выпускаются также версии с пониженным напряжением питания от 2 В. Такие МК имеют в названии литеру L. Микроконтроллеры Z8 имеют в своем составе схему сброса по включению питания, сторожевой таймер, два режима энергосбережения. Почти все микроконтроллеры семейства имеют схему сбро-

са по снижению напряжения питания и могут устанавливать защиту от чтения внутренней памяти.

Микроконтроллеры Z8Plus являются дальнейшим развитием Z8 для низкостоимостных применений и насчитывают сейчас 4 версии. Сохранив программную совместимость с первым семейством, МК Z8Plus обладают более высокой производительностью. За счет сокращения длительности машинного цикла и использования команд фиксированной длины новые микроконтроллеры работают в 1,5 раза быстрее, чем аналогичные версии Z8. Все команды процессора Z8Plus выполняются за 1 мкс при тактовой частоте 10 МГц.

Микроконтроллеры содержат однократно программируемое ПЗУ программ от 0,5 до 1 Кбайт, ОЗУ от 32 до 64 байт и выпускаются в 18- и 20-выводных корпусах.

Программно-аппаратные средства отладки. ZiLOG предлагает очень интересный и относительно недорогой (290\$) программатор-эмулятор Z8ССР00ZEM для микроконтроллеров Z8 группы ССР, позволяющий проводить полный процесс отладки, а затем программировать микроконтроллеры с однократным ПЗУ. Кроме этого, распространяются бесплатный ассемблер и программный симулятор. Существуют также версии Си-компиляторов.

8.8. SAMSUNG. SAM86, SAM88

8-разрядные микроконтроллеры SAM86 южнокорейской компании Samsung Electronics послужат нам переходным мостиком между известным Западом и таинственным Востоком. Микроконтроллеры этого семейства разработаны на основе ядра Z8 компании ZiLOG, но имеют несколько видоизмененную архитектуру и более простую систему команд.

Процессор SAM86, как и Z8, адресует три области памяти: регистровый файл, включающий от 112 до 208 регистров общего назначения, память программ до 8 Кбайт и память данных.

Как и инженеры ZiLOG, создавшие следующее поколение Z8Plus, разработчики Samsung также не остановились на достигнутом и разработали улучшенное процессорное ядро SAM88, адресуемое до 64 Кбайт памяти программ и данных.

В систему команд SAM88 вернулись команды работы с 16-разрядными операндами, команды загрузки с автоинкрементом, команды двоично-десятичной арифметики. Появились новые инструкции, отсутствовавшие в Z8, такие как команды умножения (24 системных цикла процессора), команды деления 16/8 (12 системных циклов), команды би-

товых операций, включающие в себя инструкции логических операций с битами. В результате количество команд увеличилось с 41 до 78 инструкций.

Минимальное время исполнения инструкций SAM86 и SAM88 составляет 750 нс при внешней тактовой частоте 8 МГц (для регистровых операций), что соответствует аналогичным характеристикам Z8. Некоторые версии микроконтроллеров SAM работают на тактовых частотах до 25 МГц.

В настоящее время выпускается около 40 моделей микроконтроллеров KS86 и KS88 построенных соответственно, на основе процессоров SAM86 и SAM88. Для большинства производимых МК доступны однократно программируемые версии (OTP).

Поскольку Samsung является довольно большой корпорацией, преуспевающей во многих областях производственной технической деятельности, то не стал удивительным тот факт, что многие модели выпускаемых МК приобрели полузаказной характер, соответствующий приложению, где они применяются. Так появились микроконтроллеры для аудио- и видеоприложений, для работы с жидкокристаллическими индикаторами, микроконтроллеры, встраиваемые в пластиковые карты, контроллеры шины USB и так далее.

Кроме специализированных МК, Samsung Electronics выпускает 19 моделей микроконтроллеров общего назначения, в число которых входят микросхемы обоих подсемейств KS86 и KS88. Кроме архитектурных отличий, микроконтроллеры KS88 имеют также расширенный регистровый файл (до 1040 байт, доступных в страничном режиме) и, как правило, больший объем внутренней памяти программ (от 16 до 32 Кбайт).

Производимые Samsung МК отличает большое разнообразие таймерных модулей на основе семи типов 8- и 16-разрядных таймеров-счетчиков, различающихся реализуемыми функциями.

В таймерный блок могут входить 8-разрядный базовый таймер с возможностью работы в качестве сторожевого таймера, 8- и 16-разрядные таймеры-счетчики, реализующие функции ШИМ, счета внешних событий, входного захвата, счета с автоперезагрузкой.

В периферию МК общего назначения также входят:

- два типа аналогово-цифровых преобразователей с разрядностью 8 и 10 бит и числом каналов от 4 до 12. Длительность цикла преобразования составляет для 8-разрядного АЦП 24 мкс, а для 10-разрядного АЦП – 25 мкс при внешней тактовой частоте 8 МГц. Кроме этого, аналоговый блок может включать схему детектора перехода состояния внешнего вывода через ноль (zero cross detection);

- ШИМ-контроллер. В зависимости от модели, он может быть одноканальным или двухканальным с разрешением от 8 до 14 бит;
- блок прерываний поддерживает до 11 внутренних и 12 внешних источников прерываний;
- система последовательного интерфейса представлена UART (до двух на кристалле), синхронным интерфейсом SCI, а также контроллером I2C.

Почти все микроконтроллеры общего назначения имеют специальный встроенный генератор сигналов звуковой частоты от 20 Гц до 20 КГц (buzzer output).

Наиболее дешевые версии МК KS86 и KS88 выпускаются в 16/18/20/32/40-выводных пластиковых корпусах.

Большинство микроконтроллеров начинают работать при напряжении питания 2,7 В, некоторые версии еще более «низковольтны» и устойчиво работают при напряжении 2 В.

Аппаратные отладочные средства для МК представлены внутри-схемным полнофункциональным эмулятором SMD32. Из программных средств доступны ассемблеры SASM88 и SAMA ASSEMBLER.

9. ШКОЛА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Типовые системы на базе цифрового процессора обработки сигналов (ЦПОС) состоят из кристалла ЦПОС, памяти, аналого-цифрового преобразователя (АЦП) и цифроаналогового преобразователя (ЦАП), а также каналов связи.

Кристалл ЦПОС может содержать большое разнообразие функциональных блоков, реализованных аппаратно. Некоторые наиболее важные из них, размещаемые внутри кристалла, перечислены ниже:

- Центральный арифметический блок - этот блок является той частью ЦПОС, которая выполняет наиболее важные арифметические функции, такие как умножение и сложение. Это именно та часть, которая делает ЦПОС столь быстрым по сравнению с традиционными процессорами;
- Вспомогательный арифметический блок - он предназначен для выполнения других математических вычислений и логических операций в то время, пока основной арифметический блок занят.

Порты

Порты связи необходимы любой системе ЦОС. На вход процессора поступает первичная информация, а обработанная информация выдается во внешний мир через эти порты.

Цифровые процессоры обработки сигналов преобладают в области кодирования сигналов, речи и изображения. Они весьма удобны для реализации фильтров, преобразований и многих других задач обработки сигналов

Кодирование аналоговых сигналов

Существуют три основных метода кодирования аналоговых сигналов:

- Амплитудно-Импульсная Модуляция (АИМ);
- Широтно-Импульсная Модуляция (ШИМ);
- Фазово-Импульсная Модуляция (ФИМ).

Каждый из перечисленных методов использует различные свойства импульса для описания изменений в аналоговом сигнале.

Амплитудно-импульсная модуляция

Амплитуда импульсов модулируется поступающим аналоговым сигналом. Амплитуда импульса пропорциональна амплитуде аналогового сигнала в момент отсчета. Частота следования импульсов должна, по меньшей мере, в два раза превосходить наивысшую частоту аналогового сигнала.

Широтно-импульсная модуляция

Ширина импульсов в последовательности модулируется аналоговым сигналом. На приеме аналоговый сигнал восстанавливается пропорционально ширине принятых импульсов.

Фазово-импульсная модуляция

Фазы (положения) импульсов пропорциональны амплитуде модулирующего сигнала. Импульсы сдвигаются относительно нормального положения в зависимости от амплитуды модулирующего сигнала. ФИМ широко используется в цифро-аналоговых преобразователях.

9.1. Достоинства цифровой обработки сигналов в измерительных приборах

Одним из основных направлений развития приборостроения является все более широкое использование цифровых методов представления, хранения, передачи и обработки информации. Данные методы позволяют не только повысить метрологические показатели по сравнению с аналоговыми приборами, но и максимально автоматизировать процесс измерения и обработки.

Чтобы выполнить цифровую обработку необходимо предварительно преобразовать аналоговый сигнал на входе в цифровой код с помощью аналого-цифрового преобразователя (АЦП). Затем происходит собственно цифровая обработка сигнала и, наконец, обратное преобразование на выходе цифрового сигнала в аналоговый вид выполняется с помощью цифроаналогового преобразователя (ЦАП).

Цифровая обработка сигналов (ЦОС), в том числе и измерительной информации, имеет ряд преимуществ по сравнению с другими способами:

- хорошая стабильность и воспроизводимость;
- отсутствие таких дестабилизирующих факторов, присущих аналоговым устройствам, как температурный и временной дрейф, разброс параметров, воздействие наводок и помех;

- гибкость - возможность программной реализации алгоритмов обработки измерительной информации и изменение параметров аппаратуры;
- возможность создания адаптивных систем с перестраиваемой структурой;
- простота настройки и повышение метрологических характеристик без использования трудоемких и прецизионных операций (подгонка образцовых резисторов).

Недостатки метода цифровой обработки сигнала связаны с некоторыми потерями информации при дискретизации сигнала и его квантовании, а также с ограничениями на сложность алгоритмов обработки, если необходимо выполнять обработку сигналов в реальном масштабе времени.

9.2. Методы цифровой обработки сигналов в измерительных приборах

Цифровая обработка сигналов позволяет выполнять, в принципе, любое формально описываемое преобразование сигнала по сколь угодно сложному алгоритму со сколь угодно большой степенью точности. Например, с помощью ЦОС в измерительных приборах можно решать следующие задачи:

- фильтрация сигнала;
- его масштабирование;
- сжатие (компрессия) измерительных сигналов;
- дискретное преобразование Фурье;
- преобразования спектра сигнала;
- определение сложных функциональных зависимостей, аппроксимация функцией данного вида;
- синтез нелинейных передаточных функций;
- статистический анализ полученных результатов.

Алгоритмы цифровой обработки сигналов могут реализовываться как в устройствах на жесткой логике, в том числе и на программируемых логических интегральных схемах (ПЛИС), так и в устройствах с программной логикой - универсальных либо специализированных микропроцессорах и микроконтроллерах, а также в универсальных компьютерах.

9.3. Особенности процессоров цифровой обработки сигналов

Особенности архитектуры цифровых процессоров обработки сигналов (ЦПОС) (а англоязычной литературе используется термин Digital Signal Processors и аббревиатура DSP) связаны с особенностью программ, по которым им приходится работать:

- программы выполняются, как правило, в реальном масштабе времени - по мере поступления входного сигнала, что придает критическую важность вопросам повышения быстродействия;
- программы содержат много логических и, особенно, арифметических операций и практически не содержат программ перехода;
- происходит постоянный и быстрый ввод-вывод данных, зачастую в аналоговой форме;
- программы относительно короткие и достаточно редко изменяются, зачастую остаются неизменными на протяжении всего срока эксплуатации процессора.

Из этого проистекают такие особенности архитектуры:

- в процессорах цифровой обработки сигналов очень часто используется так называемая Гарвардская архитектура - с отдельными блоками памяти для хранения программ и данных. Они могут иметь разную разрядность, к ним происходит обращение по разным командам;
- большая (иногда нестандартная) разрядность обрабатываемых данных - 16, 24, 32, 48, 64, 128, что позволяет увеличить диапазон обрабатываемых чисел без применения формата с плавающей запятой или обрабатывать по несколько чисел одновременно;
- блоки, предназначенные для ускорения выполнения команды умножения - сдвиговые регистры, матричные умножители;
- память команд и данных на самом кристалле процессора;
- возможность параллельного выполнения нескольких операций одновременно, например, ввода-вывода и арифметических команд;
- все команды имеют одинаковую длину и выполняются за одинаковое время, что позволяет использовать счетчик команд для отсчета временных интервалов.

10. ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ. УРОВНИ ПРЕДСТАВЛЕНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ

Микропроцессорная система может быть описана на различных уровнях абстрактного представления.

Существующую микропроцессорную систему можно описать на любом известном уровне представления, но в начальной стадии проектирования ее можно описать только на концептуальном уровне. В процессе разработки системы происходит переход от одного уровня ее представления к другому, более детальному. Каждая абстракция несет в себе только информацию, которая соответствует данному уровню и не содержит каких-либо сведений относительно более низких уровней. Микропроцессорная система может быть описана на одном из следующих уровней абстрактного представления:

- 1) «черный ящик»;
- 2) структурный;
- 3) программный;
- 4) логический;
- 5) схемный.

На уровне «черного ящика» микропроцессорная система описывается внешними спецификациями - перечисляются основные ее внешние характеристики.

Структурный уровень создается компонентами микропроцессорной системы: микропроцессорами, запоминающими устройствами, устройствами ввода/вывода, внешними запоминающими устройствами, каналами связи. Микропроцессорная система описывается функциями отдельных устройств и их взаимосвязью, информационными потоками.

Программный уровень разделяется на два подуровня: команд процессора и языковой. Микропроцессорная система интерпретируется как последовательность операторов или команд, вызывающих то или иное действие над некоторой структурой данных.

Логический уровень присущ исключительно дискретным системам. На этом уровне выделяются два подуровня: переключательных схем и регистровых пересылок. Подуровень переключательных схем образуется вентилями и построенными на их основе операторами обработки данных.

Переключательные схемы подразделяются на комбинационные и последовательностные; первые, в отличие от последних, не содержат запоминающих элементов. Поведение системы на этом уровне описывается

алгеброй логики, моделью конечного автомата, входными/выходными последовательностями 1 и 0. Комбинационные схемы представляются таблицей истинности, в которой каждому входному набору значений сигналов ставится в соответствие набор значений сигналов на выходах.

Последовательностные схемы могут описываться диаграммами или таблицами входов/выходов, в которых определены взаимно однозначные соответствия между входами схемы, внутренними состояниями (комбинациями значений элементов памяти) и выходами. Подуровень регистровых пересылок характеризуется более высокой степенью абстрагирования и представляет собой описание регистров и передачу данных между ними. Он включает в себя две части: информационную и управляющую. Информационная часть образуется регистрами, операторами и путями передачи данных. Управляющая часть определяет зависящие от времени сигналы, инициирующие пересылку данных между регистрами.

Схемный уровень образуется резисторами и конденсаторами. Показателями поведения системы на этом уровне служат напряжение и ток, представляемые в функции времени или частоты. Этот уровень описания дискретной системы широко используется в описаниях аналоговых систем и не является ни наинизшим из возможных, ни достаточным для полной характеристики системы.

10.1. Ошибки, неисправности, дефекты

В жизненном цикле микропроцессорной системы, как любой дискретной системы, выделяются три стадии: проектирование, изготовление и эксплуатация. Каждая из стадий подразделяется на несколько фаз, для которых существуют вероятности возникновения конструктивных или физических неисправностей, приводящих систему в неработоспособное состояние. Поэтому на каждой фазе необходимы процедуры тестового контроля, направленные на обнаружение и локализацию неисправностей.

Процедура тестового контроля может быть определена как проведение экспериментов с «черным ящиком». Дискретная система любой сложности или часть такой системы может рассматриваться как «черный ящик» с множеством входов и выходов. Правильность функционирования этого «черного ящика» должна устанавливаться путем подачи входных сигналов и наблюдения ответных выходных сигналов системы. В тех случаях, когда поведение «черного ящика» отличается от нормаль-

ного, характеризуемого его спецификацией или представлениями человека, говорят о наличии ошибки.

Ошибка вызывается некоторой неисправностью, представляющей собой некорректное состояние внутри «черного ящика».

Неисправности классифицируют в соответствии с их причинами: физическая, если причиной ее служат либо дефекты элементов, либо физическое воздействие окружающей среды; субъективная (внесенная, нефизическая), если ее причиной служат ошибки проектирования, неправильный монтаж элементов, грубые ошибки оператора.

Физические неисправности - непредусмотренные, нежелательные изменения значения одной или нескольких логических переменных в системе. Субъективные неисправности - конкретные проявления недостатков программного и аппаратного обеспечения и неправильных действий оператора, имеющих место при выполнении дискретной системой предписанных спецификацией действий.

Под субъективными неисправностями подразумеваются неисправности нефизические, вызванные недостатками различных схем, конструкций, программ, средств эксплуатации - компиляторов, ассемблеров, программ автоматизации проектирования, инструкции по эксплуатации, процедур и средств контроля и т.д. Субъективные неисправности делят на проектные и интерактивные.

Проектные неисправности вызваны недостатками, вносимыми в систему на различных стадиях реализации исходного задания при структурном проектировании, разработке алгоритмов, написании программ, трансляции в машинный код, детальном логическом и техническом проектировании, а также при последующих модификациях аппаратного и программного обеспечения.

Интерактивные неисправности возникают, когда в процессе работы, технического обслуживания или отработки системы оператор вводит в нее через интерфейс человек-машина ложную информацию, не соответствующую текущему состоянию системы. Как правило, это происходит в результате непонимания инструкции для оператора или вследствие неточностей ввода информации.

Ошибка - проявление неисправности (физической или субъективной). В зависимости от уровня иерархической структуры системы термин «ошибка» может иметь различный смысл. Так, для дискретного устройства он означает появление неверных двоичных сигналов («0» вместо «1» и «1» вместо «0»); для программы ошибка означает отклонение поведения программы от заданного, приводящее к выдаче неверных результатов.

Следует четко разграничивать понятия «ошибка» и «неисправность». Неисправность может приводить или не приводить к ошибке в зависимости от состояния системы. В то же время возникновение ошибки обязательно говорит о существовании какой-то неисправности. Одна и та же ошибка может быть вызвана множеством неисправностей, а одна неисправность может служить причиной целого ряда ошибок.

Дефекты - физические изменения параметров компонентов системы, выходящие за допустимые пределы. Их называют сбоями, если они носят временный характер, и отказами, если они постоянны.

Существует такая причинно-следственная связь:

1) дефект, представляющий собой изменение в значениях параметров компонентов, вызывает неисправность, т.е. отклонение от заданного значения (значений) логической переменной (переменных) в точке дефекта;

2) неисправность приводит к подаче неверных логических значений на вход (входы) остальной части системы и может вызывать ошибки, проявляющиеся при последующей работе других исправных логических схем;

3) ошибки приводят к появлению неправильных результатов или к отклонению от нормального хода исполнения программы.

10.2. Отладка

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 ров-

но 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 не сложных систем.

10.3. 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 испытываемого объекта, для того чтобы сделать неисправность наблюдаемой. Метод испытаний должен позволить генерировать тесты, ставящие испытываемый объект в условия, при которых моделируемые неисправности проявляли бы себя в виде обнаруживаемых ошибок. Если испытываемый объект предназначен для эксплуатации, то при обнаружении ошибки необходимо произвести локализацию неисправности с целью ее устранения путем ремонта или усовершенствования испытываемого 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 не формализуемые этапы, то отладка системы предполагает участие человека.

10.4. 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.

10.5. 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:

- 1) управлять поведением системы или/и ее модели на различных уровнях абстрактного представления;
- 2) собирать информацию о поведении системы или/и ее модели, обрабатывать и представлять на различных уровнях абстракции;
- 3) преобразовывать системы, придавать им свойства контролепригодности;

4) 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 изготавливать микропроцессорное устройство с теми или иными «зашивками» ПЗУ, программы отлаживают на эмуляционном кристалле, у которого магистраль выведена на внешние контакты и вместо ПЗУ установлено ОЗУ.

10.6. Этапы проектирования микропроцессорных систем

Микропроцессорные системы по своей сложности, требованиям и функциям могут значительно отличаться надежностными параметрами, объемом программных средств, быть однопроцессорными и многопроцессорными, построенными на одном типе микропроцессорного набора или нескольких, и т.д. В связи с этим процесс проектирования может видоизменяться в зависимости от требований, предъявляемых к системам. Например, процесс проектирования МПС, отличающихся одна от другой содержанием ПЗУ, будет состоять из разработки программ и изготовления ПЗУ.

При проектировании многопроцессорных микропроцессорных систем, содержащих несколько типов микропроцессорных наборов, необходимо решать вопросы организации памяти, взаимодействия с процессорами, организации обмена между устройствами системы и внешней средой, согласования функционирования устройств, имеющих различную скорость работы, и т.д. Ниже приведена примерная последовательность этапов, типичных для создания микропроцессорной системы:

1. Формализация требований к системе.
2. Разработка структуры и архитектуры системы.

3. Разработка и изготовление аппаратных средств и программного обеспечения системы.

4. Комплексная отладка и приемосдаточные испытания.

Этап 1. На этом этапе составляются внешние спецификации, перечисляются функции системы, формализуется техническое задание (ТЗ) на систему, формально излагаются замыслы разработчика в официальной документации.

Этап 2. На данном этапе определяются функции отдельных устройств и программных средств, выбираются микропроцессорные наборы, на базе которых будет реализована система, определяются взаимодействие между аппаратными и программными средствами, временные характеристики отдельных устройств и программ.

Этап 3. После определения функций, реализуемых аппаратурой, и функций, реализуемых программами, схемотехники и программисты одновременно приступают к разработке и изготовлению соответственно опытного образца и программных средств. Разработка и изготовление аппаратуры состоят из разработки структурных и принципиальных схем, изготовления прототипа, автономной отладки. Разработка программ состоит из разработки алгоритмов; написания текста исходных программ; трансляции исходных программ в объектные программы; автономной отладки.

Этап 4. Смотри пункт 10.11 Комплексная отладка.

На каждом этапе проектирования МПС людьми могут быть внесены неисправности и приняты неверные проектные решения. Кроме того, в аппаратуре могут возникнуть дефекты.

10.7. Источники ошибок

Рассмотрим источники ошибок на первых трех этапах проектирования.

Этап 1. На этом этапе источниками ошибок могут быть: логическая несогласованность требований, упущения, неточности алгоритма.

Этап 2. На данном этапе источниками ошибок могут быть: упущения функций, несогласованность протокола взаимодействия аппаратуры и программ, неверный выбор микропроцессорных наборов, неточности алгоритмов, неверная интерпретация технических требований, упущение некоторых информационных потоков.

Этап 3. На этом этапе источниками ошибок могут быть: при разработке аппаратуры - упущения некоторых функций, неверная интерпре-

тация технических требований, недоработка в схемах синхронизации, нарушение правил проектирования; при изготовлении прототипа - неисправности комплектующих изделий, неисправности монтажа и сборки; при разработке программных средств - упущения некоторых функций технического задания, неточности в алгоритмах, неточности кодирования.

Каждый из перечисленных источников ошибки может породить большое число субъективных или физических неисправностей, которые необходимо локализовать и устранить. Обнаружение ошибки и локализация неисправности являются сложной задачей по нескольким причинам: во-первых, из-за большого числа неисправностей; во-вторых, из-за того, что различные неисправности могут проявляться одинаковым образом. Так как отсутствуют модели субъективных неисправностей, указанная задача не формализована. Имеются определенные успехи в области создания методов и средств обнаружения ошибок и локализации физических неисправностей. Эти методы и средства широко используются для проверки работоспособного состояния и диагностики неисправностей дискретных систем при проектировании, производстве и эксплуатации последних.

Субъективные неисправности отличаются от физических тем, что после обнаружения, локализации и коррекции больше не возникают. Однако, как следует из перечня источников ошибок, субъективные неисправности могут быть внесены на этапе разработки спецификации системы, а это означает, что даже после самых тщательных испытаний системы на соответствие ее внешним спецификациям в системе могут находиться субъективные неисправности.

Процесс проектирования - итерационный процесс. Неисправности, обнаруженные на этапе приемосдаточных испытаний, могут привести к коррекции спецификаций, а следовательно, к началу проектирования всей системы. Обнаруживать неисправности необходимо как можно раньше, для этого надо контролировать корректность проекта на каждом этапе разработки.

10.9. Аппаратная 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 устройств, 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.

Затем проводится дальнейшая проверка работы аппаратуры при различных режимах работы 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:

а) выполнять функции аналогового прибора, т.е. измерять напряжение и ток; воспроизводить форму сигнала, подавать импульсы определенной формы и т.д.;

б) 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; 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; 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, пульта, комплексы диагностирования.

10.10. 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.

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.

Средства отладки программ должны:

- а) управлять исполнением программ (останавливать, изменять порядок, запускать и т.д.);
- б) собирать информацию о ходе выполнения программы;
- в) обеспечивать обмен информацией (диалог) между программистом и ЭВМ на уровне языка программирования;
- г) моделировать работу отсутствующих аппаратных средств микропроцессорной системы.

10.11. 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, не отлаженная аппаратура и программа вводятся в создаваемую систему, присоединяются к проверенной ее части.

Если отладка программ ведется с использованием эмуляционного ОЗУ, а затем изготавливаются 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.

ЛИТЕРАТУРА

Основная литература

1. Шарапов А.В.. Цифровые и микропроцессорные устройства /М-во общ.и проф.образования РФ.ТУСУР. - Томск : ТУСУР, 1998. - 161 с.
2. Шарапов А В Цифровые и микропроцессорные устройства /Мин-во общего и проф.образования РФ.ТУСУР. - Томск: ТМЦДО, 1999. - 161 с.: ил.
3. Уильямс,Г.Б. Отладка микропроцессорных систем /Перевод с рус. - М.: Энергоатомиздат, 1988. - 253 с.: ил.
4. Балашов Е. П., Пузанков Д. В Микропроцессоры и микропроцессорные системы /Под ред. В.Б.Смолова. - М.: Радио и связь, 1981. - 326 с.: ил.
5. Калабеков Б. А.Микропроцессоры и их применение в системах передачи и обработки сигналов. - М.: Радио и связь, 1988. - 368 с.

Дополнительная литература

Периодические издания. Журналы:

- ChipNews/Инженерная микроэлектроника (1998-2001)
- Современные технологии автоматизации (1998-2001)
- « Мир компьютерной автоматизации» (1998-2001)

Интернет ресурсы:

- www.automation.ru – путеводитель по промышленной автоматизации в России;
- Конференция "Микроконтроллеры и их применение" - <http://www.ts.aha.ru/>
- www.sterling.ru, www.dep.redline.ru – отечественные лидеры–фирмы производители контроллеров и средств автоматизации;
- www.motorola.com, www.atmel.com, www.sterling.com, www.zworld.com, www.advantech.com , www.wago.com – мировые лидеры– фирмы изготовители микроконтроллеров, микропроцессоров, ПЛК, промышленных компьютеров.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСУ «МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА В СИСТЕМАХ УПРАВЛЕНИЯ»

Цель изучения дисциплины

Данная дисциплина имеет следующие цели:

- формирование системного базового представления, первичных знаний, умений и навыков студентов по основам микропроцессорных систем, достаточных для дальнейшего продолжения образования и самообразования в области вычислительной техники и в смежных областях;
- изучение принципов построения, функциональных возможностей и архитектурных решений современных микропроцессорных систем (МПС), микроконтроллеров (МК) и промышленных ЭВМ;
- освоение методики проектирования микропроцессорных систем.

Задачи дисциплины

В результате изучения дисциплины у студентов должны быть сформированы представления о:

- принципах построения и функциональных возможностях микропроцессорных систем, микроконтроллеров и промышленных ЭВМ;
- состоянии развития современной элементной базы, ведущих мировых изготовителей и отечественных поставщиках электронных и микропроцессорных компонентов;
- методике проектирования микропроцессорных систем.

После изучения дисциплины студент должен приобрести систематизированные знания в следующих областях:

- принципы построения микропроцессорных систем и микроконтроллеров;
- основные микропроцессорные семейства отечественного и зарубежного производства;
- вопросы аппаратной и программной организации микропроцессорных систем;
- инструментальные средства отладки, диагностики и проектирования микропроцессорных систем и микроконтроллеров;
- применение микропроцессорных устройств и систем в автоматизированных системах управления технологическим процессом (АСУТП).

1. Содержание дисциплины

1.1. Основные понятия, обзорный анализ современных однокристальных микроконтроллеров, микропроцессоров и микропроцессорных комплектов

1.1.1. Назначение и области применения микропроцессорных устройств

- Предмет, объект, метод, цели и задачи дисциплины "Микропроцессорные системы".
- Первые определения и понятия. Контроллер, промышленный компьютер, микропроцессор, микроконтроллер, микропроцессорный комплект и т.д.
- Назначение и области применения микропроцессорных устройств: товары народного потребления, промышленность, АСУТП и т.д.
- Представление информации в микропроцессорных системах.
- Последовательный и параллельный способ представления информации.

1.1.2. Микропроцессор. Архитектура

- Основные части микропроцессорного устройства.
- Определение и назначение процессора.
- Обзор и характеристики архитектур микропроцессоров.
- Микропроцессор. Определение, типовой состав.
- Принцип действия и внутреннее устройство микропроцессоров.
- Назначение составных частей микропроцессора.
- АЛУ. Определение, функции, основные операции, выполняемые в АЛУ.

1.1.3. Память в микропроцессорных системах

- Память в микропроцессорных системах – определение, назначение, классификация.
- Основные характеристики полупроводниковой памяти.
- Типы микросхемы оперативных запоминающих устройств (ОЗУ).
- Типы микросхем постоянных запоминающих устройств (ПЗУ).
- Буферная и стековая память в микропроцессорных устройствах.

1.1.4. Интерфейсные устройства ввода/вывода информации в микропроцессорных системах

- Последовательный и параллельный способ передачи информации. Определение, характеристики, примеры.
- Структура и принцип работы параллельной шины.
- Режимы обмена между микропроцессорными устройствами: дуплексный, полудуплексный и симплексный.
- Реализация и применение синхронной и асинхронной последовательной передачи данных.
- Алгоритм работы асинхронной последовательной передачи данных.

1.1.5. Внутренняя структура современного микроконтроллера

- Краткая история микропроцессоров.
- Основные характеристики микропроцессоров.
- История архитектур. Основные черты RISC и CISC концепции построения микроконтроллера.
- Структура и назначение основных блоков современного микроконтроллера.
 - Вычислительный блок.
 - Память программ и данных.
 - Порты ввода/вывода.
 - Периферийные устройства: таймеры/счетчики, аналого-цифровой преобразователь, аналоговый компаратор, параллельный и последовательный порт.
- Режимы пониженного энергопотребления микроконтроллера.

1.1.6. Классификация микроконтроллеров

- Четырехразрядные микроконтроллеры.
- Восемьразрядные микроконтроллеры.
- Шестнадцати- и тридцати разрядные микроконтроллеры.
- Процессоры цифровой обработки сигналов.

1.1.7. Программное обеспечение микропроцессоров

- Общие принципы разработки программного обеспечения МПС.
- Компиляторы и программаторы.
- Инструментальные средства разработки и отладки программ для микроконтроллеров: внутрисхемные эмуляторы, программные симуля-

торы, платы развития (оценочные платы), мониторы отладки, эмуляторы ПЗУ.

1.1.8. Критерии выбора микропроцессора

- Основные системные и функциональные требования.
- Система и выполнение команд.
- Характеристика поставщика и производителя.
- Критерии оценки при выборе микропроцессора: технические характеристики, эксплуатационные характеристики, потребительские свойства.

1.1.9. Классификация и анализ современного состояния рынка микроконтроллеров на примере наиболее ярких представителей

- Восьмиразрядные RISC микроконтроллеры: Atmel, Microchip, Scenix, Ангстрем.
- Восьмиразрядные CISC микроконтроллеры: Motorola, Zilog, Samsung.
- Шестнадцатиразрядные микроконтроллеры фирм Hitachi и Advanced Micro Devices.

1.1.10. Школа цифровой обработки сигналов

- Типовой состав системы на базе цифрового процессора обработки сигналов (ЦПОС).
- Достоинства цифровой обработки сигналов в измерительных приборах.
- Особенности процессоров цифровой обработки сигналов.

1.1.11. Проектирование микропроцессорных систем

- Уровни представления микропроцессорной системы.
- Ошибки, неисправности, дефекты на всех стадиях жизненного цикла микропроцессорной системы.
- Этапы проектирования микропроцессорных систем. Функции и задачи, решаемые на каждом этапе. Источники ошибок при проектировании.
- Функции и средства отладки микропроцессорной системы
- Комплексная отладка микропроцессорных систем.

1.2. Лабораторные занятия

Лабораторные занятия проводятся в студенческой лаборатории «ЭлеСи – ТУСУР».

1. Изучение ОРС – сервера «Восток»:
 - 1.1. Конфигурация сервера;
 - 1.2. Создание сигналов;
 - 1.3. Работа с модулем логики.
2. Изучение ОРС – клиента «Восток»:
 - 2.1. Модуль оперативных алармов;
 - 2.2. Модуль трендов.
3. Изучение пакета программирования ПЛК OpenPCS:
 - 3.1. Выполнение задания по созданию рабочего проекта;
 - 3.2. Реализация алгоритма управления технологическим процессом.

2. Учебно-методические материалы по дисциплине

2.1 Основная литература

1. Шарапов А.В. Цифровые и микропроцессорные устройства: Учебное пособие. - Томск: ТУСУР, 1998. - 161 с.
2. Шарапов А.В. Цифровые и микропроцессорные устройства: Учебное пособие. - Томск: ТМЦ ДО, 1999. - 161 с.: ил.
3. Уильямс Г.Б. Отладка микропроцессорных систем /Перевод с рус. - М.: Энергоатомиздат, 1988. - 253 с.: ил.
4. Балашов Е. П., Пузанков Д. В Микропроцессоры и микропроцессорные системы / Под ред. В.Б. Смолова. - М.: Радио и связь, 1981. - 326 с.: ил.
5. Калабеков Б.А. Микропроцессоры и их применение в системах передачи и обработки сигналов. - М.: Радио и связь, 1988. - 368 с.

2.2 Дополнительная литература

- 4.2.1 Периодические издания. Журналы:
 - ChipNews/Инженерная микроэлектроника (1998-2001).
 - Современные технологии автоматизации (1998-2001).
 - « Мир компьютерной автоматизации» (1998-2001).

4.22. Интернет ресурсы:

- www.automatization.ru – путеводитель по промышленной автоматизации в России;
- Конференция "Микроконтроллеры и их применение" - <http://www.ts.aha.ru/wwwboards/mcontrol/7/wwwboard.html>
- www.sterling.ru, www.dep.redline.ru – отечественные лидеры–фирмы производители контроллеров и средств автоматизации;
- www.motorola.com, www.atmel.com, www.sterling.com, www.zworld.com, www.advantech.com, www.wago.com – мировые лидеры – фирмы изготовители микроконтроллеров, микропроцессоров, ПЛК, промышленных компьютеров.

3. Методические указания по выполнению контрольных работ

Всего в семестре две контрольные работы.

Контрольная работа состоит в проведении исследования по заданной теме и оформлению результатов в виде отчета. Список вариантов заданий приведен в приложениях №1 и №2 для первой и второй контрольной работы соответственно.

Структура отчета по контрольной работе №1 «Общие вопросы микропроцессорных устройств»

В каждом варианте после названия темы исследований, приведен ряд вопросов обязательных для рассмотрения.

Структура отчета:

1. Введение.

1.1. Анализ состояния данного вопроса: предпосылки возникновения, современное состояние и тенденции развития.

1.2. Место в системе и классификация.

1.3. Постановка задач исследования (не менее пяти). Необходимо представить краткое обоснование выбора каждой задачи исследования.

2. Основная часть.

2.1. Рассмотрение каждой выделенной задачи.

2.2. Рассмотрение вопросов указанных в задании.

3. Заключение - выводы по проведенному исследованию.

3.1. Приведение результатов по выделенным задачам исследования.

3.2. Приведение результатов по вопросам указанным в задании.

3.3. Общие выводы.

4. Список используемой литературы – приводится только те информационные ресурсы, на которые есть ссылки в тексте отчета.

Структура отчета по контрольной работе №2 « Исследование современного микропроцессора»

1. Введение.

1.1. Краткая информация о заданной фирме и семействе (не более 5% от всего отчета).

1.2. Место данного семейства в общей классификации микропроцессоров, занимаемое положение на рынке;

2. Основная часть.

2.1. Исследование данного семейства согласно критериям, приведенным в разделе 7. «ВЫБОР МИКРОКОНТРОЛЛЕРА» данного учебного пособия.

3. Заключение - выводы по проведенному исследованию.

4. Список используемой литературы – приводится только те информационные ресурсы, на которые есть ссылки в тексте отчета.

Приложение №1 Контрольная работа №1. Варианты заданий

Вариант 1. Аналоговые и дискретные сигналы в микропроцессорных устройствах.

- Характеристики каждого типа: описание, график, расчетные формулы.
- Сравнительный анализ по истории развития и области применения.

Вариант 2. АЦП в микропроцессорных устройствах.

- Типы, характеристики, виды аппаратной реализации (не менее пяти).
- Сравнительный анализ типов по: истории развития, области применения, точности, стоимости и т.д.

Вариант 3. ЦАП в микропроцессорных устройствах.

- Типы, характеристики, виды аппаратной реализации (не менее пяти).
- Сравнительный анализ типов по: истории развития, области применения, точности, стоимости и т.д.

Вариант 4. Фон-неймановская архитектура в микропроцессорных устройствах.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель архитектуры.

Вариант 5. Гарвардская архитектура в микропроцессорных устройствах.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель архитектуры.

Вариант 6. Периферийные устройства в ЭВМ – мышь и звуковая карта.

- Виды аппаратной реализации, сравнение интерфейсов обмена на физическом уровне.
- Программная реализация: алгоритм работы, драйвер связи, сравнение интерфейсов обмена на программном уровне.

Вариант 7. Периферийные устройства в ЭВМ – клавиатура и видео-карта.

- Виды аппаратной реализации, сравнение интерфейсов обмена на физическом уровне.
- Программная реализация: алгоритм работы, драйвер связи, сравнение интерфейсов обмена на программном уровне.

Вариант 8. Устройства хранения данных в ЭВМ – FDD и HDD.

- Виды аппаратной реализации, сравнение интерфейсов обмена на физическом уровне.
- Программная реализация: алгоритм работы, драйвер связи, сравнение интерфейсов обмена на программном уровне.

Вариант 9. Микропроцессор.

- История развития, области применения, основные типы.
- Характеристики, сравнительный анализ программных и аппаратных моделей.

Вариант 10. Арифметико-логическое Устройство.

- История развития, области применения, основные типы.
- Характеристики, программная и аппаратная модель, блок-схема.

Вариант 11. Статическая ОЗУ.

- Типы, характеристики, виды аппаратной реализации (не менее пяти).
- Сравнительный анализ типов по: истории развития, области применения, стоимости, быстродействия.

Вариант 12. Динамическая ОЗУ

- Типы, характеристики, виды аппаратной реализации (не менее пяти).
- Сравнительный анализ типов по: истории развития, области применения, стоимости, быстродействия.

Вариант 13. Энергонезависимая память программ в микропроцессорных устройствах.

- Виды реализации и типы микросхем, принцип действия, область применения.

Вариант 14. Энергонезависимая память данных в микропроцессорных устройствах.

- Виды реализации и типы микросхем, принцип действия, область применения.

Вариант 15. Буферная память в микропроцессорных устройствах.

- Виды реализации и типы микросхем, принцип действия, область применения.

Вариант 16. Стековая память в микропроцессорных устройствах.

- Виды реализации и типы микросхем, принцип действия, область применения.

Вариант 17. Последовательный интерфейс в микропроцессорных устройствах.

- Виды аппаратной реализации, сравнение на физическом уровне.
- Виды программной реализации: алгоритма работы, сравнение на программном уровне.
- Рассмотреть работу СОМ-порта в ПК (физическая реализация, схема подключения, работа с портом из программ пользователя).

Вариант 18. Параллельный интерфейс в микропроцессорных устройствах.

- Виды аппаратной реализации, сравнение на физическом уровне.
- Виды программной реализации: алгоритма работы, сравнение на программном уровне.
- Рассмотреть работу СОМ-порта в ПК (физическая реализация, схема подключения, работа с портом из программ пользователя).

Вариант 19. Режим работы интерфейса - дуплексный.

- Основные характеристики, примеры, области применения.
- Рассмотреть техническую реализацию в последовательном и параллельном канале передачи данных.

Вариант 20. Режим работы интерфейса - полудуплексный.

- Основные характеристики, примеры, области применения.
- Рассмотреть техническую реализацию в последовательном и параллельном канале передачи данных.

Вариант 21. Режим работы интерфейса - симплексный.

- Основные характеристики, примеры, области применения.
- Рассмотреть техническую реализацию в последовательном и параллельном канале передачи данных.

Вариант 22. CISC-концепция.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель.

Вариант 23. RISC-концепция.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель.

Вариант 24. MISC-концепция.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель.

Вариант 25. VLIW-концепция.

- История развития, области применения, сравнение с другими архитектурами.
- Характеристики, программная и аппаратная модель.

Приложение №2 Контрольная работа №2. Варианты заданий

Вариант	Фирма	Семейство
1	Atmel	AVR
2	Atmel	AT89S
3	Intel	8051
4	Microchip	PIC
5	Mitsubishi	8-битные
6	Motorola	HC05, HCC06, HC11
7	Ангстрем	Тесей
8	Scenix	SX
9	STMicroelectronics	ST6
10	Россия	1816BE51 /1816BE48
11	Texas Instruments	TMS370
12	Toshiba	TLCS-90
13	Samsung	SAM8X
14	Zilog	Z8
15	Hitachi	H8
16	Infineon	16x
17	Intel	MCS-96/296
18	Mitsubishi	M16C
19	Motorola	68HC12
20	Analog Devices	DSP
21	Philips	80C51XA
22	Zilog	Z80/80180/80380
23	IBM/Motorola	PowerPC
24	DALLAS SEMICONDUCTOR	8-битные
25	Texas Instruments	MSP430

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОМУ ПРОЕКТУ

1. Целью курсового проекта является закрепление теоретических знаний и формирование практических навыков при выполнении самостоятельной разработки микропроцессорного устройства, а также приобретение навыков работы с отечественным и зарубежным информационно-справочным материалом.

2. Типовое задание на курсовое проектирование предполагает выполнение этапов:

2.1. Анализ предложенного задания – изучение объекта и предмета исследований, обзор информационно-справочного материала по предложенной теме, поиск аналогов.

2.2. Разработка структурной и функциональной схемы разрабатываемого устройства, создание основного алгоритма работы устройства.

2.3. Разработка принципиальной схемы устройства и спецификации элементов.

2.4. Создание пакета конструкторских документов: печатная плата, чертеж конструкции, спецификация.

2.5. Оформление отчета по курсовому проекту.

3. Методические рекомендации по выполнению курсового проекта

Ниже приведены основные требования по выполнению перечисленных этапов курсового проектирования. Качество выполнения студентом данных требований является основным критерием при выставлении оценки за курсовой проект.

3.1. Анализ предложенного задания

На данном этапе студенту необходимо:

1) Изучить предложенное задание на курсовой проект:

- выделить объект и предмет исследований,
- произвести обзор информационно-справочного материала для детального ознакомления с объектом и предметом исследований.

2) Произвести поиск аналогов разрабатываемого устройства. Необходимо найти не менее пяти аналогов устройства, наиболее полно удовлетворяющих предложенному заданию. Для каждого аналога необходимо указать:

- назначение устройства, область применения,
- отличительные положительные черты (на ваш взгляд),

- отрицательные черты (на ваш взгляд).

После оценки всех пяти аналогов необходимо сформулировать актуальность Вашей разработки.

3) Оформить результаты исследований в виде технического задания на разработку.

Техническое задание должно содержать следующие пункты:

1. Введение

1.1. Наименование разрабатываемого устройства.

1.2. Назначение разрабатываемого устройства – область применения.

2. Обзорный анализ найденных аналогов, оформленный в виде таблицы, в которой необходимо еще раз подчеркнуть следующее:

2.1.1. Наименование и область применения.

2.1.2. Основные положительные черты, выгодно отличающие данное устройство от прочих аналогов.

2.1.3. Основные отрицательные черты, не позволяющие использовать данное устройство для ваших целей.

3. Обоснование актуальности разработки вашего устройства.

3.2. Разработка функциональной схемы разрабатываемого устройства и создание основного алгоритма работы устройства

На данном этапе необходимо:

1) Разработать структурную схему устройства.

В данном пункте разрабатываемое устройство рассматривается в качестве «черного ящика». На схеме обозначаются:

– входные и выходные каналы с указанием их основных характеристик: напряжение, ток, форма и тип сигнала;

– используемое для работы устройства внешнее оборудование – датчики и исполнительные механизмы, для каждого необходимо представить полное наименование, основные характеристики, а также характеристику канала связи его с разрабатываемым устройством.

В приложении к структурной схеме необходимо привести основной алгоритм функционирования устройства. В данном случае устройство рассматривается как цельный блок без детализации.

2) Разработать функциональную схему устройства:

В данной схеме необходимо рассмотреть внутреннюю структуру разрабатываемого устройства:

- представить внутреннюю структуру устройства с указанием основных его блоков и их взаимосвязи;
- для каждого блока необходимо привести краткий алгоритм его работы в устройстве, а для линий связи между блоками необходимо представить электрические и временные характеристики;
- привести основной алгоритм управляющего микропроцессора.

3.3. Разработка принципиальной схемы устройства и спецификации элементов

На данном этапе студенту необходимо:

- произвести выбор элементной базы с указанием требуемых расчетов и основных технических характеристик элементов;
- создать принципиальную схему устройства, желательно использовать пакеты PCAD или Orcad. На схеме для каждого элемента должны быть указаны:
 - 1) тип и порядковый номер;
 - 2) наименование (для микросхем) или значение (для пассивных элементов);
- создать спецификацию элементов принципиальной схемы.

Пример оформления спецификации элементов принципиальной схемы:

№	Поз.обозн.	Наименование	Кол.	Примечания
		Микросхемы		
1	DD1	AT90S8535 – 8AI	1	Корпус TQFP – 44pin
		Конденсаторы		
2	C1,C3..C5	0805 - 25В - 18 пФ 5	4	Чиповые

3.4. Создание пакета конструкторских документов

На данном этапе студенту необходимо:

- создать чертеж конструкции устройства в трех видах с указанием габаритных и установочных размеров, желательно использовать программу «Компас»;
- разработать печатную плату устройства, желательно использовать пакеты PCAD или Orcad.

3.5. Оформление отчета по курсовому проекту

В отчете должны быть представлены все вышеперечисленные пункты, и в конце необходимо привести вывод, в котором нужно отметить:

- полноту проведенного обзора аналогов устройства,
- степень соответствия разработанного устройства заданию,
- возможные пути совершенствования устройства,
- вероятность конкурентоспособности разработанного Вами устройства на рынке.

Варианты заданий на курсовое проектирование

Вариант 1

Регулятор освещенности в помещении с лампой накаливания постоянного тока.

Устройство должно измерять освещенность в помещении, сравнивать с заданным суточным графиком (ночь, утро, день, вечер) и плавно управлять выходным напряжением лампы накаливания постоянного тока напряжением 36В. Задание графика освещенности производится пользователем посредством четырех кнопок и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 2

Регулятор освещенности в помещении с лампой накаливания переменного тока.

Устройство должно измерять освещенность в помещении, сравнивать с заданным суточным графиком (ночь, утро, день, вечер) и плавно управлять выходным напряжением лампы накаливания переменного тока частотой 50Гц и напряжением 220В. Задание графика освещенности производится пользователем посредством четырех кнопок и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 3

Регулятор температуры в помещении.

Устройство должно измерять температуру в помещении, сравнивать с заданным суточным графиком температуры (ночь, утро, день, вечер) и управлять двумя нагревательными приборами. Задание графика температуры производится пользователем посредством четырех кнопок и жидкокристаллического индикатора. Необходимо обратить внимание

на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 4

Регулятор уровня воды в резервуаре.

Имеется резервуар, датчик давления и задвижка с электрическим приводом. Устройство должно измерять уровень воды в резервуаре, сравнивать с заданным суточным графиком воды (ночь, утро, день, вечер) и управлять электрическим приводом задвижки. Задание графика уровня воды производится пользователем посредством четырех кнопок и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 5

Регулятор температуры воды в аквариуме.

Устройство должно измерять температуру воды в аквариуме, сравнивать с заданным значением и управлять нагревательным прибором. Задание температуры воды производится пользователем посредством четырех кнопок, отображение текущей температуры воды на жидкокристаллическом индикаторе. Необходимо обратить внимание на меню пользователя, компактность и минимальную стоимость устройства.

Вариант 6

Регулятор влажности воздуха в помещении.

Устройство должно измерять влажность воздуха в помещении, сравнивать с заданным суточным графиком влажности (ночь, утро, день, вечер) и управлять нагревательным прибором и вентилятором. Задание графика влажности производится пользователем посредством четырех кнопок и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 7

Кодовый замок с возможностью перепрограммирования.

Прибор должен позволять с основной клавиатуры изменять код доступа. Должен быть продуман механизм защиты кода от взлома и переход на резервное питание прибора при отсутствии основного. Должна быть предусмотрена пожарная сигнализация, а также сигнализация при

попытке взлома самого замка или двери. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 8

Охранно-пожарная сигнализация дачного дома.

Прибор должен обеспечивать контроль трех контуров, переход на резервное питание при отсутствии основного. В случае взлома должна сработать звуковая и световая сигнализация, а в случае пожара, дополнительно, должен включиться поливочный насос. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 9

Измеритель перемещений с потенциометрическим датчиком пути.

Потенциометрический датчик пути преобразует перемещение механизма в напряжение постоянного тока. Устройство должно измерять это напряжение, производить первичную его обработку (фильтрация, нормирование) и отображать текущее положение механизма на жидкокристаллическом индикаторе. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 10

Измеритель скорости с импульсным датчиком.

Импульсный датчик скорости формирует двухфазную последовательность импульсов. Частота последовательности пропорциональна скорости вращения вала. Устройство должно измерять эту частоту, определять направление вращения и угловую скорость с заданной точностью и отображать их текущие значения на жидкокристаллическом индикаторе. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 11

Измеритель перемещения с кодовым датчиком.

Кодовый датчик измеряет перемещение механизма. Устройство должно определять положение механизма и сохранять в памяти данных информацию о перемещении в течение заданного интервала времени и

отображать текущее положение на двух семисегментных индикаторах. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 12

Система управления широтно-импульсным регулятором.

Прибор должен сформировать импульсные последовательности для управления ключами широтно-импульсного преобразователя при заданной частоте следования импульсов, которая вводится пользователем посредством четырех кнопок на передней панели и отображается на жидкокристаллическом индикаторе. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 13

Система управления автономным инвертором напряжения.

Прибор должен сформировать импульсные сигналы управления ключами автономного инвертора при заданной форме и частоте выходного напряжения, которые вводится пользователем посредством четырех кнопок на передней панели и отображаются на жидкокристаллическом индикаторе. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 14

Цифровой таймер для электробытовых машин и приборов.

Прибор должен коммутировать цепи питания электробытовых машин и приборов (~220В). Требования к прибору: бесконтактная коммутация, не менее трех управляющих выходов с программируемыми временными интервалами включения и выключения, контроль короткого замыкания в коммутируемых цепях. Задание всех временных уставок производится пользователем посредством четырех кнопок на передней панели и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 15

Цифровой таймер для автомобильных приборов.

Прибор должен коммутировать цепи питания автомобильных приборов (12В). Требования к прибору: бесконтактная коммутация, не менее трех управляющих выходов с программируемыми временными интервалами включения и выключения, контроль короткого замыкания в коммутируемых цепях. Задание всех временных уставок производится пользователем посредством четырех кнопок на передней панели и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 16

Звуковой таймер.

Устройство предназначено для записи коротких звуковых фрагментов и их воспроизведения в установленное пользователем время. Задание временных интервалов осуществляется посредством четырех кнопок и двух семисегментных индикаторов. Необходимо обратить внимание на меню пользователя, компактность и низкую себестоимость.

Вариант 17

Говорящие «часы-будильник-календарь».

Устройство предназначено для ведения времени, представление его на жидкокристаллическом индикаторе и звуковое оповещение в соответствии с заданными уставками. Пример режимов оповещения: начало часа, подъем, сон, дни рождения друзей, знакомых и т.д. Необходимо обратить внимание на меню пользователя, компактность и низкую себестоимость.

Вариант 18

Спорт-табло.

Устройство предназначено для управления спортивным табло, состоящим из 200 ламп (200Вт каждая). Взаимодействие с пользователем осуществляется через клавиатуру и жидкокристаллический индикатор. Необходимо обратить внимание на расположение ламп на табло, меню пользователя, защиту от неправильного подключения внешних цепей устройства, а также самотестирование прибора.

Вариант 19

Табло остановки городского транспорта.

Устройство предназначено для индикации на светодиодном табло графика движения городского транспорта через остановку. Необходимо обратить внимание на структурную организацию табло, связь с центральным диспетчерским пунктом, а также самотестирование прибора.

Вариант 20

Устройство оповещения в салоне городского транспортного средства.

Устройство предназначено для оповещения пассажиров салона городского транспортного средства информацией о маршруте транспортного средства, городских достопримечательностях и рекламе товаров. Необходимо обратить внимание на механизм записи новых голосовых сообщений в устройство (реклама и т.д.), механизм определения местоположения транспорта, а также самотестирование прибора.

Вариант 21

Устройство обработки внешнего аналогового сигнала.

Устройство предназначено для измерения, индикации и архивации внешнего аналогового сигнала. Диапазон напряжения от 0 до 10В, максимальная частота дискретизации 1кГц. Необходимо обратить внимание на механизм сжатия измерений, связь с персональным компьютером для копирования архивов, а также самотестирование прибора.

Вариант 22

Устройство обработки внешнего широтно-импульсного сигнала.

Устройство предназначено для измерения, индикации и архивации внешнего широтно-импульсного сигнала. Диапазон частоты от 0,1Гц до 10кГц, форма сигнала – прямоугольник, скважность от 0 до 100%, напряжение «логической единицы» 10В. Необходимо обратить внимание на механизм сжатия измерений, связь с персональным компьютером для копирования архивов, а также самотестирование прибора.

Вариант 23

Устройство обработки внешнего частотного сигнала.

Устройство предназначено для измерения, индикации и архивации внешнего частотного сигнала. Диапазон частоты от 0,1Гц до 10кГц, форма сигнала – прямоугольник, напряжение «логической единицы» 10В. Необходимо обратить внимание на механизм сжатия измерений,

связь с персональным компьютером для копирования архивов, а также самотестирование прибора.

Вариант 24

Адаптивный ПИД-регулятор.

Устройство предназначено для реализации ПИД-регулятора с возможностью изменения коэффициентов «на ходу». Задание всех коэффициентов и индикация текущего состояния ПИД-регулятора осуществляется посредством двенадцати кнопок (телефонная клавиатура) и жидкокристаллического индикатора. Необходимо обратить внимание на меню пользователя, защиту от неправильного подключения внешних цепей устройства пользователем, а также самотестирование устройства.

Вариант 25

Устройство идентификации объектов.

Устройство предназначено для идентификации проходящих через ворота объектов (скот, тележки и т.д.). На панели прибора отображается число объектов за сутки. Необходимо осуществлять бесконтактную идентификацию, а также самотестирование прибора.