

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(Самарский университет)**

В.Г. ИОФФЕ

**СТРУКТУРНАЯ ОРГАНИЗАЦИЯ
ОДНОКРИСТАЛЬНЫХ
МИКРОКОНТРОЛЛЕРОВ**

**САМАРА
2017**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(Самарский университет)

В.Г. ИОФФЕ

СТРУКТУРНАЯ ОРГАНИЗАЦИЯ
ОДНОКРИСТАЛЬНЫХ
МИКРОКОНТРОЛЛЕРОВ

САМАРА

Издательство Самарского университета

2017

2

УДК 681.32
ББК 32.973

Рецензенты: д-р техн. наук Секисов Ю.Н.,
доцент, к.т.н. Полулех А.В.

Иоффе Владислав Германович

Структурная организация однокристалльных микроконтроллеров [Электронный ресурс]: учебное пособие / В. Г. Иоффе. - Самара.: Изд-во Самарского университета, 2017. 206 с, ил.. Электрон. текстовые и граф. дан. (1,98 Мбайт).- 1 эл. опт. диск (CD-ROM)

Электронное учебное пособие содержит описание обобщенной структуры однокристалльных микроконтроллеров и блоков, входящих в её состав. Рассмотрены основные принципы организации микроконтроллеров и резидентных периферийных устройств. Выполнен анализ тенденций развития микроконтроллеров.

Учебное пособие предназначено для бакалавров, обучающихся по направлению 09.03.01 - «Информатика и вычислительная техника».

Электронное учебное пособие разработано на кафедре информационных систем и технологий.

Стр.206, рис.68, табл.7, библи.124

УДК 681.32
ББК 32.973

© Самарский университет, 2017

Содержание

Введение	5
1. Основные характеристики ОМК	11
1.1. Операционные возможности	11
1.2. Быстродействие	18
1.3. Периферийные устройства	21
1.4. Инструментальные средства разработки	21
1.5. Резидентные средства программирования и отладки	25
1.6. Конструктивно-технологические характеристики	25
2. Обобщённая структура ОМК	31
2.1 Организация магистралей	33
2.2 Центральный процессор	37
2.3 Блок синхронизации и управления	40
2.4 Организация памяти	44
3. Периферийные блоки	48
3.1. Параллельные порты	54
3.2. Последовательные порты	58
3.2.1. Последовательный порт UART (USART)	66
3.2.2. Интерфейс SPI	68
3.2.3. Интерфейс I2C	71
3.3. Подсистема прерываний	77
3.4. Блок временных событий	81
3.5. Средства ввода/вывода аналоговой информации	98
3.5.1. Аналоговый компаратор	98
3.5.2. Особенности организации блока ЦАП ОМК	100
3.5.3. Аналого-цифровой преобразователь	102
3.5.4. Особенности организации блока АЦП ОМК	115
4. Особенности структурной организации ОМК	118
5. Способы реализации структур микроконтроллеров	127
6. Режимы работы ОМК	143
7. Структурная организация 32-разрядных микроконтроллеров	154
Контрольные вопросы	183
Список использованных источников	196

*«Знание некоторых принципов
легко возмещает незнание некоторых фактов»
Клод Адриан Гельвécий*

Введение

Современные системы автоматизации широко используют универсальные или специализированные *контроллеры* КН, которые могут работать как в автономном режиме, так и на разных уровнях иерархических автоматизированных систем.

К основным задачам, которые необходимо решать КН, относятся:

- сбор информации с датчиков различного назначения;
- обработка входной информации на основе определенных алгоритмов;
- формирование сигналов для управления объектом автоматизации;
- организация обмена информацией с автоматизированной системой;
- проверка работоспособности контроллера;
- ввод-вывод оперативной информации и её визуализация.

Структура, реализующая эти функции, представлена на рисунке 1.

Основные функции по обработке и управлению выполняет *процессорный блок ПБ*.

Постоянное ПЗУ (или репрограммируемое РПЗУ) содержит программы обработки, таблицы, константы, не изменяемые в процессе работы КН.

ОЗУ является местом хранения входных, выходных данных и промежуточных результатов.

Средства ввода-вывода дискретных сигналов СВВ ДС обеспечивают ввод информации, характеризующей состояние объекта ав-

томатизации, и вывод управляющих воздействий, формируемых в процессе обработки входных данных.

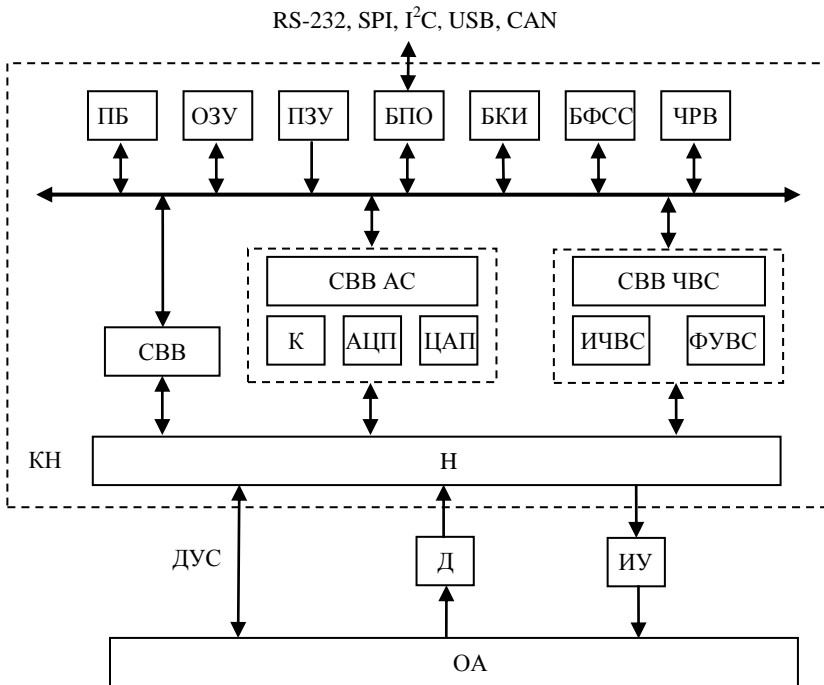


Рис. 1. Обобщенная структура контроллера

В состав средств ввода-вывода аналоговых сигналов СВВ АС обычно входят многоканальные *аналого-цифровые преобразователи* АЦП, выполняющие квантование и дискретизацию входных сигналов, *цифроаналоговые преобразователи* ЦАП, служащие для формирования аналоговых сигналов управления определенными типами исполнительных устройств и отображения информации, *компараторы* К, необходимые для аппаратного выполнения процедуры сравнения двух аналоговых величин.

Средства ввода-вывода частотно-временных сигналов СВВ ЧВС служат для измерения сигналов датчиков ИЧВС, информативными параметрами которых являются частота, временной интервал, момент появления различных событий, фазовый сдвиг, и *форми-*

вания управляющих воздействий различного вида с требуемыми временными характеристиками ФЧВС.

Оперативный ввод исходных данных, диагностической, управляющей информации и вывод информации о состоянии контроллера и объекта автоматизации выполняет блок клавиатуры и индикации БКИ.

Блок обмена информацией в последовательном коде БПО организует взаимодействие с интерфейсами автоматизированных систем. Обычно БПО поддерживает работу одновременно с несколькими наиболее распространенными интерфейсами RS-232, SPI, I2C, USB, CAN, Ethernet и так далее.

Блок формирования сигнала сброса БФСС является средством предотвращения «зависания» КН в случае возникновения определенных событий: снижения напряжения питания ниже допустимого предела, сбой в работе контроллера, превышение допустимого времени ожидания и так далее.

При работе в режиме реального времени в состав контроллера должны входить часы реального времени ЧРВ, имеющие автономное питание и генератор тактовых сигналов.

Нормализаторы Н выполняют приведение входных сигналов к виду и уровню, воспринимаемому средствами ввода данных, согласование выходных сигналов с исполнительными устройствами и, в случае необходимости, гальваническое разделение сигналов.

Нормализаторы осуществляют связь с объектом автоматизации ОА с помощью датчиков Д, исполнительных устройств ИУ, дискретных управляющих сигналов ДУС.

Широкое распространение контроллеры получили в связи с развитием электроники и, в частности, микропроцессорной техники.

В соответствии с законом Мура число транзисторов в производимых микропроцессорах удваивается каждые два года, каждые 18 месяцев удваивается их производительность, тактовая частота, вычислительная мощность, доступная за \$1, стоимость микросхемы падает вдвое каждые 18 мес.

Наиболее эффективные иллюстрации развития электроники по сравнению с другими энергоемкими отраслями промышленности, сформулированные лет 5-10 назад, приведены ниже.

Если бы автомобилестроение эволюционировало в соответствии с *законом Мура*, то Роллс-Ройс (самый дорогой автомобиль ручной сборки) стоил бы сегодня 3 доллара, 1 галона (3,785 л) бензина было бы достаточно, чтобы проехать полмиллиона миль, дюжина автомобилей разместились бы на булавочной головке и было бы дешевле его выбросить, чем заплатить за парковку.

Реализация этого же закона в авиаиндустрии привела бы к тому, что перелёт по маршруту Нью-Йорк-Париж, который в 1978 году стоил около 900 долларов и длился 7 часов, обошелся всего в цент и занял бы менее одной секунды!

Сейчас себестоимость производства транзисторов в микросхемах упала до такой степени, что цена изготовления одного транзистора и напечатанного типографского знака в газете (например, запятой) одинакова.

Хотя закон Мура был сформулирован на основании эмпирических зависимостей и строго не отражает реальные темпы развития электроники, но он характеризует ее вектор развития. [1,2] За последние годы практически перестали расти частоты микропроцессоров, а увеличение производительности обеспечивают многоядерные технологии. Это возможно потому, что плотность транзисторов продолжает расти, но темпы роста замедляются. Скорость удвоение числа транзисторов изменяется от 24 до 36 месяцев. Эксперты предполагают, что рост остановится при достижении плотности размещения 5 нм где-то между 2020 и 2030 годами. [3]. Фирма Intel обещала к 2015г выпустить чипы 10 нм, но в настоящее время наиболее полно представлены технологии 22 нм, хотя в ограниченном количестве выпускаются чипы с 14 нм (30 атомов кремния).

Элементной базой современных контроллеров являются *микропроцессоры* МП и *однокристалльные контроллеры* ОМК, основными достоинствами которых являются низкая стоимость в пересчете на выполняемую функцию, малые габариты и потребляемая мощность, универсальность и гибкость применения за счет программного управления, технологичность, повышенная надежность.

МП - программно управляемое устройство, обеспечивающее процесс обработки цифровой информации и управление, выполненное в виде одной или нескольких БИС.

В настоящее время широко используются только однокристалльные МП. МП содержит один или более блоков обработки цифровой информации, блок регистров, блоки управления, обеспечивающие информационные связи как внутри МП, так и с внешними устройствами. В терминах вычислительной техники МП - это операционное устройство, состоящее из операционных и управляющих автоматов.

Разработка КН на базе микропроцессоров требует применения большого числа микросхем различного назначения, что существенно увеличивает стоимость, габариты, энергопотребление, снижает надежность контроллера.

Например, для контроллера (рисунок 1) на базе микропроцессоров 8080/8086 требуется, как минимум, около 14 микросхем (без учета микросхем памяти и нормализаторов).

ОМК являются наиболее массовым представителем микропроцессорной техники. Первый патент на однокристалльную микро-ЭВМ (микроконтроллер) был выдан в 1971 году инженерам *М. Кочрену и Г. Буну*, сотрудникам Texas Instruments. Именно они предложили на одном кристалле разместить не только микропроцессор, но и память, устройства ввода-вывода. ОМК позволяют с минимальными затратами реализовать большую номенклатуру систем управления различными объектами и процессами. Они находят широкое применение в промышленной автоматике, контрольно-измерительной технике, автоматизированных системах различного назначения, аппаратуре связи, бытовой техники и многих других применений.

По оценкам компании Semico Research, в 2010 году каждый человек ежедневно должен был иметь дело с 350 микроконтроллерами, установленными в домашнем и офисном оборудовании, автомобильных системах, а также в устройствах личного пользования.

С появлением ОМК связывают начало эры компьютерной автоматизации в области управления.

Более рационально проектирование контроллера на базе ОМК. В принципе, КН (рисунок 1) можно выполнить с помощью одной микросхемы (без учета нормализаторов).

Основное внимание в пособии уделяется анализу структур и характеристик 8-разрядных ОМК. Однако принципы организации являются общими для ОМК любой разрядности.

В последние годы доля 8-разрядных ОМК в общем объеме продаж сокращается, а увеличивается доля 16- и 32-разрядных микроконтроллеров. Это связано со снижением стоимости ОМК. Причем аналитики предсказывают, что большими перспективами обладают 32-разрядные микроконтроллеры. Достаточно сказать, что стоимость 32-разрядного ОМК с ядром Cortex-M3 и богатой периферией на ноябрь 2011 года равна 0,99\$.

Однако развитие и применение 8-разрядных ОМК будет продолжаться из-за их малых размеров, низкой потребляемой мощности и стоимости. [4, 5]

1. Основные характеристики ОМК

Знание основных характеристик ОМК позволяет проводить их сравнительный анализ, оценивать возможность применения в конкретной системе и степень влияния на её параметры. Характеристики ОМК описываются, в основном, теми же параметрами, что и ЭВМ. В этом разделе приводятся обобщенные характеристики всех видов ОМК. Особенности отдельных типов ОМК будут отражены в соответствующих разделах.

1.1. Операционные возможности

Операционные возможности определяются множеством операций, реализуемых ОМК с целью обработки, хранения и управления вводом-выводом информации. Их можно оценить, анализируя функциональную и структурную организацию ОМК. [6]

Функциональную организацию определяют, в основном, форматы данных, команд, возможность реализации параллельных процессов, контроля и диагностики аппаратных и программных средств, сервисные функции.

Формат данных характеризуется длиной формата (разрядностью) и типом представления данных (числовые, логические, битовые, строки символов).

Числовые данные представлены, в основном, *форматами с фиксированной запятой*, поддерживающими целочисленную арифметику. Достоинством этого формата является относительно *высокое быстроедействие*, если диапазон изменения обрабатываемых данных известен и соответствует разрядности процессорного блока или незначительно её превышает.

Для обработки действительных чисел на ОМК следует использовать соответствующие алгоритмы обработки.

Данные могут быть *в виде кода* (беззнаковое представление) и *целого числа со знаком в дополнительном коде*.

Разрядность данных влияет на точность вычислений (величина машинной погрешности \pm единица младшего разряда) и диапазон представления чисел.

Наличие знака при фиксированной длине формата *уменьшает точность* обрабатываемых данных в два раза и требует формирова-

ния признака арифметического переполнения, который в ОМК может аппаратно не формироваться.

Кодовое представление данных обеспечивает большую точность из-за отсутствия знака.

Увеличение точности и быстродействия обеспечивается применением форматов удвоенной длины.

При изменении данных в широком диапазоне использование форматов с фиксированной запятой приводит к потере быстродействия за счет параллельно-последовательной обработки.

В этом случае более рационально использование форматов с плавающей запятой.

Для форматов с плавающей запятой точность зависит от разрядности мантиссы, а диапазон - разрядности характеристики. Характеристика - число без знака, которое равно базовому смещению \pm порядок, т.е. для отрицательного максимального порядка она равна нулю, а положительного - "все 1".

В целях унификации форматов данных с плавающей запятой разработан *стандарт IEEE-754*, поддержку которого следует контролировать при выборе ОМК.

Аппаратная реализация арифметических операций с плавающей запятой поддерживается только в ОМК повышенной производительности и цифровых сигнальных процессорах ЦСП (DSP).

В ОМК общего применения обработку чисел с плавающей запятой выполняют программно. Программы обработки доступны через Интернет и входят в состав соответствующих библиотек стандартных функций. Наиболее просто процедура обработки чисел с плавающей запятой реализуется при программировании на языках высокого уровня.

В ЦСП могут использоваться смешанные форматы, которые содержат фиксированную целую и действительную части [7,8].

Числовые данные могут использовать *двоичную* или *десятичную* систему счисления. В последнем случае данные представляются *двоично-десятичным кодом «8421»*.

Двоично-десятичная арифметика в ОМК поддерживается редко. Поэтому возможности по обработке двоично-десятичных данных в ОМК, как правило, ограничены. Аппаратная реализация подобных команд в 8-разрядных микроконтроллера практически отсутствует.

Наиболее часто используется двоичное сложение двоично-десятичных чисел с последующим применением команды коррекции. В некоторых ОМК команда коррекции после сложения отсутствует (например, AVR Atmel, HC08 Motorola), но формируется флаг переноса из 3 разряда в 4. Тогда после двоичного сложения следует использовать алгоритм коррекции на основании кода «с избытком 6» и последующем анализе значений переноса из старшего разряда (бит С) и из 3 разряда в 4 (биты АС или Н).

Двоично-десятичный код наиболее часто используется в промежуточных преобразованиях при вводе информации с клавиатуры и выводе на устройства индикации.

Логические данные представляются форматами фиксированной длины. Каждый бит обрабатывается независимо, а совокупность логических значений передается и обрабатывается как единое целое.

Наличие битовых данных и команд их обработки сокращает объем и время выполнения программ в задачах управления. В ОМК повышенной разрядности предусмотрены следующие форматы битовых переменных: бит, битовые поля, битовые строки.

Обработка строк символов в 8-разрядных ОМК не предусмотрена. Поэтому алфавитно-цифровая (текстовой) информация в коде ASCII обрабатывается программно.

Формат команды характеризуется разрядностью поля операций, адресностью, используемыми способами адресации, длиной формата. Формат команд во многом определяет операционные возможности ОМК и их структурную организацию.

В зависимости от типа ОМК число операций может изменяться в широком диапазоне: от нескольких десятков (*RISC* ОМК) до нескольких сотен в (*CISC* ОМК). Особенности RISC и CISC архитектур будут рассмотрены позднее. Однако в последних разработках ОМК система команд RISC и CISC отличаются незначительно.

Система команд любого ОМК является функционально полной, но в зависимости от числа команд и их сложности изменяются объем памяти, необходимый для хранения программ, время их выполнения и требуемые аппаратные затраты внутри кристалла ОМК.

В ОМК наблюдается тенденция формирования такого набора команд, который оптимизирует объектные коды программ за счет их

ориентации под конкретные компиляторы языков высокого уровня (Паскаль, СИ, Форт), команды нечеткой логики и так далее.

ОМК может содержать одно поле операций или несколько. В последнем случае возможно выполнение одновременно несколько операций. Подобная система команд характерна для некоторых 16-и 32-разрядных ОМК.

В общем случае выбор ОМК с требуемой системой команд следует производить, анализируя класс обрабатываемых алгоритмов и ряд технических требований: цена производительности, энергопотребление, габариты и т.д.

Адресность определяется количеством адресов, указанных в формате команде. В зависимости от адресности команды подразделяются на *безадресные, одно-, двух-, трехадресные*. С увеличением числа адресов в команде *увеличиваются затраты памяти* на хранение команд, но одновременно *уменьшается число обращений к памяти*, т.е. уменьшается время обработки и сокращается количество пересылок между регистрами и/или ячейками памяти. Однако одновременно с *адресностью необходимо контролировать разрядность шины*, по которой передаются команды. Например, эффективное быстрое действие двухадресных команд при длине в 2 байта, передаваемых через 8-разрядную шину, будет меньше, чем через 16-разрядную.

В общем случае, затраты памяти и быстрое действие существенно зависят от свойств алгоритмов.

Наиболее оптимальной является система команд, в которой используются команды с различной адресностью.

Для 8- и 16-разрядных ОМК характерно использование двухадресных, одноадресных и безадресных команд. Для одноадресных команд возможно использование дополнительного бита модификации, позволяющего записать результата в аккумулятор или по адресу, указанному в команде (например, ОМК фирмы Microchip).

Трехадресные команды используются, в основном, в 32-разрядных ОМК.

Эффективность команд во многом определяется используемыми *способами адресации*. Типовыми являются следующие способы: *прямая, прямая регистровая, непосредственная, косвенная, относительная, стековая, базовая, индексная, автоинкрементная, авто-*

декрементная, базовая индексная с масштабированием. [9]. Возможны различные модификации на основе этих способов.

Требуемая совокупность способов адресации определяется классом обрабатываемых алгоритмов. Поэтому необходимо знать не только правила вычисления исполнительного адреса, но и особенности их применения.

Прямая адресация используется для обращения к ячейкам памяти или регистрам, предназначенным для временного хранения информации или в качестве элементов с закрепленными функциями (параметры подпрограмм, место хранения индекса, базы и так далее). Предпочтение следует отдавать прямой регистровой адресации, так как в этом случае увеличивается быстродействие и сокращается длина команды.

Непосредственная адресация применяется при работе с константами, отражающими особенности реализуемого алгоритма (исходные значения параметров, фиксированное число повторений цикла и так далее). При использовании этого способ адресации необходимо контролировать разрядность памяти команд. Большее быстродействие будет в том случае, если команда считывается за одно обращение к памяти. В некоторых ОМК с 8-разрядной организации памяти (например, MCS-51/52) выполнение команды с непосредственной адресацией требует дополнительного цикла обращения к памяти.

Косвенная адресация предназначена для работы с основными структурами данных (таблицы, массивы, списки и так далее). В ОМК используется регистровая косвенная адресация. Чем больше число регистров косвенной адресации, тем большее быстродействие обеспечивается при работе с несколькими структурами данных. При этом необходимо контролировать разрядность регистров и их назначение (память команд, внутренняя или внешняя память данных). Недостатком канонической косвенной адресации является отсутствие аппаратной модификации содержимого регистра.

Сокращение объема программ и увеличение быстродействия обеспечивает *автоинкрементная и автодекрементная* адресации. Некоторые ОМК поддерживают масштабируемые способы адресации, при которых модификация регистра косвенной адресации производится в зависимости от разрядности адресуемых данных.

Частным случаем автоинкрементная и автодекрементная адресации является *стековая* адресация, наличие которой необходимо при работе с прерываниями, подпрограммами для сохранения адреса возврата и параметров этих процедур.

Перемещаемость программ и данных в памяти обеспечивают *относительные* способы адресации. Относительная адресация для программ реализуется на базе счетчика адреса команд, а данных - на основе индексных и/или базовых регистров, число которых расширяет функциональные возможности ОМК и сокращает объем программ за счет уменьшения команд пересылки. При этом необходимо контролировать величину смещения, обеспечивающую реализацию относительной адресации. Величина смещения может быть от десятков байт до десятков килобайт и выше.

При разработке программ всегда предпочтение следует отдавать относительной адресации, так как при любом начальном адресе остаются неизменными связи внутри программы.

Форматы команд современных ОМК могут содержать дополнительные поля, модифицирующие основную операцию (например, микроконтроллеры ARM)

Структурная организация определяется функциональной организацией ОМК и особенностями внутреннего и внешнего интерфейсов.

Структуру ОМК характеризуют:

- особенности организации процессорного блока;
- особенности организации и обращения к памяти данных и команд: принстонская (фон Неймана) или гарвардская;
- допустимый объем резидентной и внешней памяти
- система команд: сложный набор команд CISC, сокращенный RISC, смешанный MISC;
- встроенные средства контроля и программирования;
- возможность работы в режимах пониженного энергопотребления;
- состав и количество резидентных периферийных устройств, степень их интеллекта;
- особенности организации внутренних и внешних интерфейсов.

К особенностям организации процессорного блока относятся: разрядность арифметико-логического устройства, число операционных устройств (операционных ОА и управляющих автоматов УА), позволяющих одновременное выполнение нескольких процессов ввода-вывода и обработки данных, количество регистров и/или регистровых файлов и возможность их динамического распределения, используемые структурные методы повышения быстродействия: число ступеней конвейера, средства предсказания ветвлений, упорядочивание очереди команд, наличие кэш-память и так далее. [6]

При анализе внутренних интерфейсов следует контролировать разрядность шин адреса и данных, наличие дополнительных шин, обеспечивающих обмен данными между памятью, процессорным блоком и периферийными устройствами, и особенности их организации.

Внешний интерфейс ОМК характеризуют: количество линий ввода-вывода, способы организации внешних шин, режим конфигурации.

В зависимости от назначения ОМК количество внешних выводов изменяется от 8 до 300 и выше.

- При анализе внешних шин необходимо контролировать разрядность шины данных. (ШД). Она определяет потенциальные возможности по быстродействию;

- разрядность шины адреса (ША). Она определяет максимальное адресное пространство памяти, устройств ввода-вывода. Способ использования: ША позволяет судить о допустимых размерах адресуемой памяти и пространстве адресов УВВ. **Общее адресное пространство** предполагает его разделение между ресурсами памяти и устройств ввода-вывода, а в **раздельном** - адреса памяти и устройств ввода-вывода независимы;

- организацию шин: фон неймановская, гарвардская или смешанная (см.раздел 2.1)

- соотношение между разрядностью внутренней и внешней магистралей. В ряде ОМК разрядность внешней шины может быть меньше, чем внутренней;

- организацию и количество линий шины управления (ШУ). Это дает информацию о возможностях внешнего обмена ОМК (программный, по прерыванию, режим ПДП), количество запросов пре-

рываний и ПДП, линий командной информации (запись, чтение, сброс, управление масштабированием и т.д.);

- степень совмещения ША, ШД, ШУ. Максимальное быстродействие обеспечивают отдельные шины, а совмещенные - минимизируют число внешних линий и обладают повышенной помехозащищенностью, так как информация по ним передается в режиме разделения времени.

Для ОМК внешний интерфейс определяется количеством параллельных и последовательных портов ввода-вывода, наличием портов с выходом на стандартные международные интерфейсы, количеством каналов ввода-вывода аналоговой информации, частотно-временных сигналов, числом запросов внешних прерываний.

При организации взаимодействия с внешними устройствами более целесообразно использование последовательных интерфейсов.

В зависимости от особенностей конфигурации ОМК может работать в режимах:

- **микروпроцессора**, при котором управление внешней памятью программ и данных выполняется *аппаратно* и поддерживается соответствующими командами (например, MCS -51/52),

- **микроконтроллера**, не имеющего команд работы с внешней памятью (например, большинство ОМК AVR фирмы Atmel),

- **расширенного микроконтроллера**, в котором *аппаратно* может быть организована работа с внешней памятью данных (например, Atmega 8515 фирмы Atmel) или команд (Microchip).

Если ОМК работает в режиме микропроцессора или расширенного микроконтроллера, часть параллельных портов используется для организации внешней магистрали.

Однако следует понимать, что в режиме микроконтроллера возможно подключение внешней памяти данных, но для управления ею необходимо разработать *программу*.

1.2. Быстродействие

Быстродействие может определяться временем выполнения **регистровых** команд или числом команд в единицу времени. Для операций с фиксированной запятой в быстродействующих ОМК используется единица измерения **MIPS** (миллион операций в секунду).

Плавающая арифметика характеризуется единицей - **FLOPS** с различными десятичными приставками (**KFLOPS**, **MFLOPS**, **GFLOPS**). Эти параметры иногда называют также пиковым быстродействием. Для сравнения процессоров или микроконтроллеров с разной тактовой частотой используют характеристику **MIPS/MHz** или **FLOPS/MHz**

Косвенной характеристикой быстродействия является значение частоты внутреннего генератора, которое в зависимости от назначения ОМК может изменяться в широком диапазоне (от 0 до сотен МГц). Ряд ОМК допускает программное управление частотой внутреннего генератора.

При анализе быстродействия косвенным методом необходимо учитывать особенности архитектуры ОМК: RISC или CISC, гарвардская или неймановская и так далее (см. 2.1).

Для RISC ОМК и гарвардской архитектуры время выполнения большинства команд составляет один период генератора. Исключения составляют ряд команд передачи управления и длинные операции (умножение, деление).

Для CISC ОМК время выполнения определяется особенностями форматов команд и кратно периоду генератора.

При выполнении реальных прикладных программ быстродействие ОМК может быть существенно меньше, чем приведенная выше оценка. Это объясняется особенностями архитектуры ОМК, организацией памяти, внешнего интерфейса и так далее.

При работе в режимах микропроцессора или расширенного микроконтроллера время выполнения команд обращения к внешней памяти увеличивается по сравнению с внутренней.

Наиболее объективной характеристикой является среднее быстродействие, при котором

$$V_{\text{ср}} = 1 / (\sum P_i \cdot T_i)$$

где P_i - вероятность появления i -операции, T_i - время выполнения i -операции.

$V_{\text{ср}}$ определяют для некоторого набора операций (теста), характерного для той или иной области применения ОМК (так называемые бенч-марковские программы).

Оценка среднего быстродействия может производиться на основе универсальных или специализированных тестов.

К числу универсальных тестов можно отнести

Dhrystone MIPS - синтетический тест производительности, разработанный компанией ***Reinhold P. Weickeru*** для тестирования целочисленной производительности процессоров общего назначения ***DMIPS*** или ***DMIPS/MHz***.

CoreMark консорциума EEMBC, позволяющий эффективно оценить производительность центральных процессоров. В этом тесте производительность увязывается с исполнением простого кода, используя общие для практически всех приложений структуры данных (списки, строки и матрицы) и алгоритмы. В последнее время при оценке характеристик процессорных ядер ОМК используется, в основном, этот тест.

Для объективной оценки ОМК необходимо анализировать набор операций, входящих в данный тест, и процентное содержание отдельных операций. Большинство универсальных тестов дают интегральную характеристику ОМК.

Если известен класс алгоритмов, для обработки которых предназначен ОМК, то предпочтение следует отдавать специализированным тестам, адекватным области применения.

1.3. Периферийные устройства

Периферийные устройства определяют возможность адаптации ОМК с минимальными аппаратными средствами к классу решаемых задач.

При выборе ОМК необходимо анализировать количество, состав периферийных устройств, возможность их параллельной работы и технические характеристики. В состав ОМК входят порты параллельного и последовательного обменов, подсистемы прерываний и прямого доступа к памяти, таймеры, аналого-цифровые и цифроаналоговые преобразователи, встроенные средства контроля и программирования и так далее.

В отличие от систем на базе МП, в которых возможно изменение состава периферийных БИС, конфигурация ОМК, как правило, задана и определяется его проблемной ориентацией.

Исключение составляют программируемые системы на кристалле, в которых состав периферийных устройств можно изменять программно в зависимости от решаемых задач. Примером подобных ОМК являются СУ8Сххххх фирмы Cypress Semiconductor. [10, 11]

Чем больше число периферийных устройств, тем больше габариты кристалла, сложнее конструкция корпуса, выше энергопотребление, стоимость.

Поэтому при выборе ОМК следует оптимизировать количество периферийных устройств и выбирать конфигурацию в соответствие с решаемой задачей.

Известен ряд микроконтроллеров, в которых оптимизация количества периферийных устройств и габаритов корпуса достигается за счет применения программно управляемых коммутационных матриц, обеспечивающих присоединение различных резидентных устройств к определенным выводам ОМК в режиме с разделением времени. [12]

1.4. Инструментальные средства разработки

При выборе ОМК основное внимание следует уделять средствам, обеспечивающим разработку и отладку микропроцессорных систем. Особенно трудоемка разработка программного обеспечения, которая составляет 90-95% стоимости проекта. Поэтому грамотный

выбор инструментальных средств позволяет сократить время проектирования в несколько раз.

Средства разработки можно разделить на программные и программно-аппаратные. [13,14]

К числу программных средств относятся ассемблеры, макроассемблеры, компиляторы, в состав которых обязательно должен входить симулятор (эмулятор). Наиболее эффективно использование виртуальных средств моделирования ВСМ.

Симулятор - программное средство, способное имитировать работу ОМК и его памяти. Развитые симуляторы содержат в своём составе также модели внешних периферийных устройств, средства моделирования внешних сигналов и анализа результатов проектирования.

При выборе ВСМ необходимо учитывать:

- поддержку требуемого языка программирования;
- состав библиотек компонентов и возможность её расширения;
- возможность эмуляции требуемых внешних устройств;
- наличие средств для проверки качества проектирования (профайлеров);
- возможность контроля временных диаграмм;
- удобство интерфейса пользователя;
- возможность работы в многопроцессорном режиме (если это необходимо);
- наличие документации сопровождения, в которой описана методика работы с симулятором и приведены примеры программирования (желательно на русском языке);
- доступность (наличие бесплатных версий, стоимость).

Большинство доступных ВСМ (симуляторов) предназначены для работы с одним определенным семейством ОМК и поддерживают разработку **только программного обеспечения**: MCS-51/52 (Pinnacle, MCU 8051 IDE, Keil), Atmel (AVRStudio, CodeVisionAVR, VMLab), Motorola (HI-WAVE), MicroChip(MPLAB), MSP 430 (IAR Kickstart) и так далее. ВСМ IAR Embedded Workbench рассчитана на работу с несколькими семействами, но не объединена в единую систему. (IAR for AVR, IAR for ARM, IAR for 8051 и так далее). В по-

добных симуляторах виртуальные средства ввода-вывода отсутствуют или очень незначительны.

Исключением является VMLab for AVR, который удовлетворяет большинству требований, сформулированных выше. К его недостаткам следует отнести: ограниченное количество виртуальных моделей устройств ввода-вывода (в основном, цифровые), не отражающих характеристики реальных компонентов, среда проектирования не позволяет разрабатывать принципиальные электрические схемы.

При проектировании устройств *на базе ОМК* лучшими характеристиками обладает BCM Proteus Design Suite фирмы Labcenter Electronics. Это коммерческий пакет программ класса САПР, объединяющий в себе две основных программы: ISIS - средство разработки и отладки в режиме реального времени электронных схем и ARES - средство разработки печатных плат. Proteus позволяет работать с семействами микроконтроллеров MCS-51/52 (Atmel, Philips), AVR (Atmel), PIC 10,12,16,18,24 (MicroChip), HC11 (Motorola), MSP-430, TMS320 Picolo (TI), ARM7 (Philips), ARM Cortex -M3 (Luminary Micro, NXP Semiconductors), Basic Stamp (Parallax), электронным конструктором Arduino. В процессе развития Proteus число моделей ОМК постоянно увеличивается.

К основным достоинствам Proteus следует отнести обширную библиотеку ОМК и внешних компонентов, большой набор виртуальных устройств, необходимых для отладки МПУ, развитый интерфейс пользователя, многопроцессорный режим, возможность разработки принципиальной электрической схемы и печатной платы. [15]

Ряд фирм предлагают инструментальные средства с графической средой программирования, применение которых не требует знания ассемблера или языка высокого уровня. В этом случае написание программ заменяется составлением графической схемы, выполненной в соответствии с определенными требованиями, которая автоматически компилируется в коды выбранного ОМК (Algorithm Builder, IAR visualSTATE, FlowCode.).

В этом классе наиболее универсальной является система FlowCode. Это графическая универсальная среда *программирования и отладки* ОМК. Разработанная граф-схема после компиляции позволяет получать программы на языке СИ, ассемблере, создавать

HEX, COF-файлы, необходимые для прошивки ОМК или работе в среде Proteus, экспортировать написанный код для работы с выбранным ОМК (PIC, AVR, ARM). [16]

Наиболее эффективный способ разработки и отладки программ - использование языка высокого уровня, ориентированного на конкретные типы ОМК. Наибольшее распространение получили языки СИ, Pascal, Fort.

Недостаток программных средств - невозможность отладки программ в режиме реального времени с учетом физических особенностей конкретного ОМК и внешних устройств.

Эти недостатки устраняются при использовании программно-аппаратных средств.

Наиболее простым и дешевым средством этого класса являются *оценочные платы* (EVB) или *модули* (EVM). В их состав входят: ОМК, ОЗУ, репрограммируемые ПЗУ, средства контроля и управления, средства связи с ЭВМ, макетное поле или разъём для устройств пользователя. Программа, отлаженная на ЭВМ, по каналу связи загружается в оценочный модуль, а далее производится отладка на реальном оборудовании.

Оценочные платы могут использоваться в качестве контроллеров автоматизированных систем.

Лучшими характеристиками обладают внутрисхемные эмуляторы (ВСЭ). ВСЭ - программно-аппаратное средство, способное замещать собой эмулируемый ОМК и память в реальной схеме. Его применение делает процесс разработки и отладки микропроцессорных устройств легко контролируемым и управляемым. Стыковка ВСЭ с отлаживаемым устройством производится при помощи кабеля со специальными соединителями, которые замещают кристаллы памяти и ОМК. Как минимум, ВСЭ содержит средства разработки и отладки программ, загрузчик, эмулятор ПЗУ, средства контроля и диагностики состояния отлаживаемой системы. ВСЭ могут быть выполнены в конструктиве ПЭВМ или в виде выносного блока с последовательным или параллельным каналом связи. Однако это достаточно дорогое устройство - (1 - 10) тыс. долларов.

1.5. Резидентные средства программирования и отладки

К ним относятся программно-аппаратные средства, расположенные внутри кристалла, обеспечивающие контроль, отладку, программирование, управление режимами работы ОМК: сторожевой таймер, детектор падения частоты, фоновый отладчик, порт тестирования JTAG, средства программирования кристаллов с флэш-памятью, защиты информации от несанкционированного чтения ПЗУ, управления режимами энергопотребления и так далее. [17]

1.6. Конструктивно-технологические характеристики

К их числу относятся тип технологии, с использованием которой изготовлен ОМК, характеристики резидентной памяти, нагрузочная способность выходных буферов, номинальное значение напряжения питания, потребляемая мощность, условия эксплуатации, конструкция и габариты корпуса, стоимость.

ОМК могут быть выполнены на элементах *статической или динамической* логики.

Устройства *статической* логики работают на частотах от 0 до максимального значения. Это позволяет выбирать тактовую частоту ОМК в зависимости от требуемого быстродействия, что приводит к снижению энергопотребления. Элементной базой последовательностных схем таких устройств является канонический триггер, что при фиксированной плотности элементов приводит к снижению количества регистров, счетчиков, объема ОЗУ, размещенных внутри кристалла. Однако статическая логика более быстродействующая.

Элементной базой *динамической* логики является *КМОП-транзистор*, который запоминает значение логического уровня на своей входной емкости. При этом значительно увеличивается количество запоминающих элементов, размещенных внутри кристалла. Однако требуется процедура регенерации, так как напряжение на емкости в течение времени уменьшается, разряжаясь через входные цепи транзистора. Поэтому в ОМК с динамической логикой вводятся ограничения на минимальную тактовую частоту, а ОЗУ требуют процедуры регенерации. Контроль за минимальным значением осуществляет детектор падения частоты, который при снижении частоты ниже допустимой формирует соответствующий сигнал.

Современные ОМК выполняются на основе статической логики.

Основными характеристиками резидентной памяти являются её объем, разрядность шины данных, возможность увеличения объема за счет присоединения внешней памяти, быстродействие, технология изготовления.

В качестве *резидентной памяти программ* ОМК наиболее часто используются:

- масочные ПЗУ (ROM), содержимое которых записывается в момент изготовления микросхем и недоступно для модификации пользователем;
- однократно программируемые ПЗУ (PROM, OTP), информацию в которые может записывать пользователь, но только один раз;
- репрограммируемые ПЗУ.

В настоящее время наиболее часто используемым типом памяти для хранения программ является флэш (Flash) -память.

Память данных обычно состоит из ОЗУ и репрограммируемого ПЗУ, выполненного по технологии EEPROM. В ПЗУ данных находятся таблицы, константы, значения которых могут изменяться в процессе работы контроллера, но содержимое памяти должно сохраняться при отключении питания. Модификация содержимого этой памяти может выполняться в процессе работы по командам ОМК.

Однако при современных требованиях к электронным компонентам Flash, EEPROM обладают целым рядом недостатков: ограниченное число циклов перезаписи, относительно невысокая скорость чтения/записи и так далее.

На смену флэш и EEPROM приходят другие типы памяти: ферроэлектрическое (сегнетоэлектрическое) ОЗУ (FRAM),

магнитно-резистивное ОЗУ(MRAM) и в перспективе ЗУ на основе графенов, нанотрубок. [18, 19]

Разработан целый ряд ОМК с резидентной памятью на основе технологий FRAM, MRAM.

Новые возможности по революционному улучшению технических характеристик ОМК обеспечивает появление *новой элементной базы - мемристоров*.

Таблица 1. Сравнительные характеристики различных технологий памяти [19]

Свойства	Тип памяти					
	MRAM	DRA M	SRA M	FLAS H	EEPROM	FRA M
Высокая плотность	+	+	-	+	-	-
Энергонезависимость	+	-	-	+	+	+
Произвольный доступ	+	+	+	-	-	+
Неразрушающее чтение	+	-	+	+	+	-
Неограниченное число обращений	+	+	+	-	-	-
Быстрое считывание	+	+	+	+	+	+
Быстрая запись	+	+	+	-	-	+
Низкое потребление при записи	+	+	+	-	-	+
Цикл чтения	5-70 нс	~100 нс	5-100 нс	~100 нс	~100 нс	50-150 нс
Цикл записи	5-70 нс	~100 нс	5-100 нс	>1 мс	>1 мс	50-150 нс
Напряжение записи, В	<5	<5	<5	5	12	<5
Срок хранения данных (без питания), лет	бесконечно	0	0	>10	>5	>10
Стойкость (число циклов записи)	>10 ¹⁵	>10 ¹⁵	>10 ¹⁵	~10 ⁶	~10 ⁶	~10 ⁶
Радиационная стойкость	+	-	+	-	-	-/+*

Мемристор — пассивный элемент в микроэлектронике, способный изменять своё сопротивление в зависимости от протекавшего через него заряда (интеграла тока за время работы). Может быть описан как двухполюсник с нелинейной вольт-амперной характеристикой, обладающий гистерезисом.

Открытые ныне компоненты схем с памятью, или «мем-элементы» (мемристоры, мем-емкости и мем-индукторы), способны сами выполнять одновременно как обработку, так и хранение информации. В условиях такой платформы, объединяющей в своих элементах процессоры и все виды памяти, полный цикл работы с

начальными, промежуточными и финальными данными происходит быстро и параллельно в одном и том же месте.

Свойства мемристоров, позволяющие с помощью их одновременно обрабатывать и хранить данные, размещая их, при этом, в несколько слоев, создавая сложные трехмерные структуры, позволяют увеличить вычислительную мощность и объемы памяти компьютеров к значениям, совершенно невообразимым даже в наше время. При этом, это может произойти в ближайшие несколько лет. Согласно заявлению Стэна Вильямса, одного из ученых компании Hewlett-Packard, замена транзисторов мемристорами обещает улучшить электронику будущего в трех ключевых направлениях:

Использование мемристоров позволит одному и тому же устройству выступать одновременно в качестве процессора и памяти. Современные компьютеры, как известно, помимо процессоров, содержат и дополнительные устройства памяти, как оперативной, так и постоянной. Совместив все эти компоненты в одном, можно избежать затрат времени на обмен данными между устройствами, таким образом, система на мемристорах будет работать намного быстрее и потреблять, при этом, существенно меньшее количество энергии

Мемристоры могут иметь размеры, намного меньшие, чем размеры транзисторов. Как известно, законы квантовой механики ограничивают минимальные размеры транзисторов значениями, к которым вплотную подбираются современные технологические процессы. Использование мемристоров позволит соблюдать закон Гордона Мура еще достаточно длительное время, не прибегая к экзотическим и дорогостоящим уловкам, таким как графеновые транзисторы или квантовые вычисления.

В отличие от транзисторов, на основе которых создаются только линейные структуры чипов, из мемристоров можно реально формировать развитые трехмерные структуры, внутри которых число внутренних связей может вырасти по экспоненте, в зависимости от количества "вертикальных" слоев структуры. Вычислительный и логический потенциал таких трехмерных структур мемристоров настолько высок, что Леон Чуа, первооткрыватель мемристоров, полагает, что на основе этой технологии можно будет изготовить первый электронный мозг. [20,21]

При выборе типа корпуса ОМК необходимо учитывать особенности технологии изготовления печатных плат разрабатываемой микропроцессорной системы. Например, выводы DIP-корпуса имеют вертикальные выводы и расположены с двух сторон корпуса., в SOIC - горизонтальное (планарное). При большом количестве внешних выводов используются корпуса PLCC, QFP (TQFP, LQFP), у которых планарные выводы расположены по периметру корпуса, и так далее.

Различные модификации этих корпусов отличаются шагом расположения внешних выводов, их шириной, высотой корпуса, материалом корпуса.

Важной технологической особенностью является *способ формирования тактовой частоты*. Возможны варианты с внешними частотно-задающими элементами, в качестве которых могут быть *внешний генератор, RC-цепь, кварцевый резонатор, или внутренними*.

Большую стабильность частоты обеспечивают кварцевые резонаторы.

При пониженных требованиях к стабильности частоты более рационально применение ОМК с внутренними частотно - задающими элементами.

В зависимости от условий эксплуатации ОМК могут быть выполнены в коммерческом, промышленном или специальном исполнении. Отличия заключаются в материале корпуса, температурном диапазоне и так далее. Коммерческое исполнение предполагает работу в ограниченном температурном диапазоне (например, 0-40 или 0-70 С), промышленный - (-40-+100) С. Вообще, температурный диапазон может быть достаточно широк, например, (-55-+125) С.

Нагрузочная способность выходных буферов ОМК может изменяться в широком диапазоне - (1,5-25) мА. В ОМК с повышенной нагрузочной способностью возможно подключение внешней нагрузки (например, полупроводниковых светодиодов) без применения дополнительных формирователей. Однако при этом необходимо контролировать общую допустимую потребляемую мощность.

Номинальное напряжение питания у современных ОМК, как правило, $\pm 5\text{В}$, $3,3\text{В}$, хотя некоторые типы допускают работу в диапазоне $+(1,8-6)\text{В}$.

В зависимости от типа технологии и тактовой частоты ток потребления ОМК изменяется в диапазоне от единиц до сотен мА. Уменьшение энергопотребления обеспечивают различные “спящие” (Sleep) режимы работы ОМК, число которых может быть от трех и более.

Выбор требуемого режима определяется допустимыми энергопотреблением и временем выхода из режима энергосбережения в рабочий режим, числом работающих в этом режиме периферийных устройств. Наиболее широкое распространение получили режимы холостого хода (Idle), останова (Power Down) и их модификации (Power Save, Standby, Extended Standby и другие).

В режиме Idle прекращает работу центральный процессор ОМК, а все остальные периферийные устройства продолжают функционировать. Достоинством этого режима является быстрая реакция на события, приводящие к “пробуждению” ОМК.

В режиме Power Down функционирование всех систем ОМК прекращается. Единственными узлами, продолжающими работать в этом режиме, являются сторожевой таймер и подсистема обработки внешних прерываний. Выход из этого режима требует большего времени, чем в Idle.

Стратегия выбора ОМК состоит в том, чтобы на основании технических характеристик определить микросхему с минимальной стоимостью и удовлетворяющую всем требуемым спецификациям проектируемой системы. [22] Ведущие фирмы предоставляют программы параметрического поиска микроконтроллеров, позволяющие выбрать модель ОМК в зависимости от технических требований (Atmel, MicroChip, SiLabs и так далее).

При выборе ОМК полезным бывает свойство *«pin-to-pin совместности»*, что позволяет при одном размере корпуса заменять микроконтроллеры с разной функциональностью, не изменяя печатную плату.

2. Обобщённая структура ОМК

Обобщенная структура ОМК представлена на рисунке 2.1. Эта структура аналогична рисунку 1 и реализует те же функции.

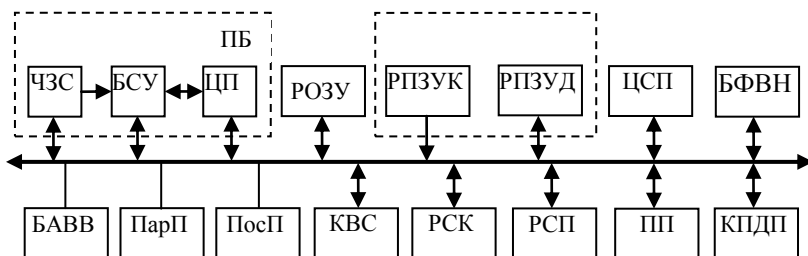


Рис. 2.1. Обобщенная структура ОМК

Процессорный блок выполнен на базе центрального процессора ЦП и блока синхронизации и управления БСУ, в функции которого входят формирование сигналов синхронизации для ЦП, внешней и внутренней магистрали ВМ, управление режимами работы ОМК, программированием, конфигурацией и так далее.

Частоту синхронизации определяет частотно-задающая схема ЧЗС, которая представляет собой генератор с внешним или внутренним частотно-задающим элементом, выполненном на кварцевом резонаторе или R, L, C контуре.

Объем резидентного ОЗУ РОЗУ в 8-разрядных ОМК изменяется в широких пределах: от 64 байт до десятков килобайт и более.

Резидентное ПЗУ РПЗУ может включать память команд и память данных. Имеются ОМК, в которых РПЗУ команд и данных отсутствует, но в большинстве случаев объем РПЗУ команд составляет от единиц Кбайт до десятков Мбайт, а РПЗУ данных - от десятков байт до нескольких килобайт.

Состав блока аналогового ввода-вывода БАВВ существенно зависит от назначения микроконтроллера. В ОМК общего применения в состав блока может входить восьмиканальный АЦП с разрядностью (8-10) бит и компаратор. В специализированных ОМК разрядность АЦП - (12 - 24) бита и БАВВ может содержать один или два ЦАП с разрядностью (10-14) бит.

В состав ОМК может входить от 1 до 8 8-разрядных параллельных портов *ПарП* с возможностью автономного управления каждым битом. Однако необходимо учитывать, что через линии параллельных портов обеспечивается функционирование АЦП, последовательных портов, счетчиков - таймеров, подсистемы прерываний и так далее.

В связи с широким применением последовательных интерфейсов количество и состав последовательных портов *ПосП* во многом определяет возможности ОМК по интеграции в автоматизированные системы. ОМК общего применения содержат от одного (как правило, с протоколом RS-232) до трех (RS-232, SPI, I2C) портов. В состав некоторых ОМК могут входить порты USB, CANbus и другие. Максимальными возможностями по работе с последовательными интерфейсами обладают коммуникационные контроллеры, количество портов которых может быть пять и более.

Контроллер временных событий *КВС* необходим для измерения и формирования частотно-временных сигналов. Он может содержать несколько счетчиков-таймеров с разрядностью (8-16) бит, управление которыми выполняется программно, или матрицу счетчиков-таймеров, в которых часть функций по измерению и формированию сигналов реализуется аппаратно.

В состав резидентных средств контроля *РСК* могут входить сторожевой таймер, выполняющий контроль за допустимым временем работы программы, монитор питания, который следит за значением напряжения питания контроллера, схемы сброса, тестирующие порты JTAG, Debug Wire, фоновый отладчик и так далее.

Резидентные средства программирования *РСП* выполняются на базе последовательных портов. Наиболее часто для этих целей используются SPI, JTAG.

Подсистема прерываний *ПП* является обязательным элементом любого ОМК. В ОМК используются, в основном, внутренние и внешние автовекторные прерывания, количество которых изменяется от единиц до десятков.

В качестве второго ядра в состав ОМК могут входить цифровые сигнальные процессоры *ЦСП*.

Контроллеры прямого доступа к памяти *КПДП* используются в ряде ОМК: Xmega фирмы Atmel, микроконтроллерах на основе ядра ARM, Cortex, специализированных ОМК, к числу которых относятся интегрированные системы сбора измерительной информации (например, в AduC 842 в режиме ПДП могут записываться результаты измерения АЦП), коммуникационные контроллеры.

Блок формирования внешнего интерфейса *БФВИ* необходим при работе ОМК в режиме микропроцессора. В состав БФВИ входят средства формирования шин данных, адреса и управления. Обычно они реализуются на основе определенных параллельных портов и представляют собой 8-разрядную шину, по которой в режиме с разделением времени передаются данные и младший байт адреса, 8-разрядную шину старшего байта адреса, шину управления, включающую сигналы записи\чтения внешней памяти, входы счетчиков, запросов внешних прерываний и так далее.

Технические характеристики ОМК и его функциональные возможности определяются особенностями внутренней и внешней магистрали, организацией ЦП и периферийных устройств.

2.1 Организация магистралей

Обмен информацией между блоками выполняется через магистраль, реализующую *фон неймановский* (принстонский) или *гарвардский* принципы организации.

Особенностью фон неймановского принципа (рисунок 2.2) является *единое адресное пространство для команд и данных и общая магистраль для их передачи*, что не позволяет одновременно извлекать команду и данные (узкое «горло фон Неймана»).

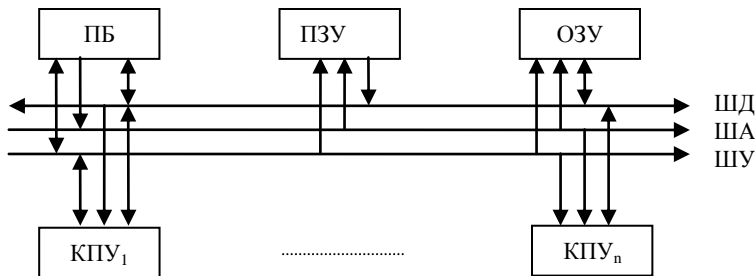


Рис. 2.2. Принцип фон Неймана.

ПБ - процессорный блок;
ПЗУ - постоянное запоминающее устройство;
ОЗУ - оперативное запоминающее устройство;
КПУ_i - контроллеры периферийных устройств.

При разработке этого принципа целью было экономия аппаратных средств. Поэтому принципиально подобная архитектура обладает меньшим быстродействием, чем гарвардская.

Однако для разработчика программного обеспечения общее адресное пространство позволяет относительно просто динамически перераспределять память между командами и данными, что существенно повышает гибкость вычислительной системы.

Стремления разделения ресурсов в рамках принципа фон Неймана при программировании привело к семантическому разделению общего поля памяти на сегменты программ, данных и стека.

Физическое и логическое разделение адресного пространства на память команд и память данных реализовано в магистрали на основе гарвардского принципа (рисунок 2.3).

Достоинства *гарвардского* принципа не ограничиваются только повышением быстродействия за счет одновременного обращения к памяти программ и данных.

Простая реализация двухуровневого конвейера.

Изменилась структура команд. Принцип фон Неймана не учитывает специфику обращения к памяти данных и команд, а использует ***общий*** формат команд. Однако в программах управления необходимый объем памяти данных МК, как правило, на порядок меньше требуемого объема памяти программ. Поэтому использование ***единого адресного пространства*** приводит к увеличению длины формата команд за счет увеличения числа разрядов для адресации операндов. Применение отдельной небольшой по объему памяти данных способствовало сокращению длины команд.

Различные форматы для команд и данных, используемые в гарвардском принципе, приводят к уменьшению длины команды, и, следовательно, к уменьшению требуемого объема программ.

Появилась возможность различной организации памяти команд и памяти данных. Например, в 8-разрядных ОМК за одно обращение к памяти команд выбирается информация длиной (12-16)

бит, а из памяти данных - 8 бит. Это приводит к дополнительному повышению быстродействия.

Поэтому большинство производителей современных ОМК используют Гарвардскую архитектуру.

К недостаткам гарвардского принципа следует отнести повышенные аппаратные затраты при реализации магистралей (большое число линий связи), невозможность динамического распределения ресурсов памяти команд и данных, фиксированный объём памяти команд. Поэтому она, главным образом, используется при организации внутренних магистралей ОМК.

Недостатки гарвардской архитектуры устраняются различными методами, которые объединяются общим названием - **модифицированная гарвардская архитектура**:

Количество **внешних линий** уменьшают за счет **общей шины** адреса и данных, но с использованием **раздельных команд** (и сигналов управления) для обращения к памяти команд и данных (8051).

Отсутствие обмена между магистральями команд и данных, которое делает невозможным динамически перераспределять ресурсы памяти и приводит к увеличению времени загрузки и неэффективным затратам памяти, устраняется использованием дополнительных аппаратных средств, обеспечивающих взаимодействие магистралей памяти и данных (УО рисунке 2.3).

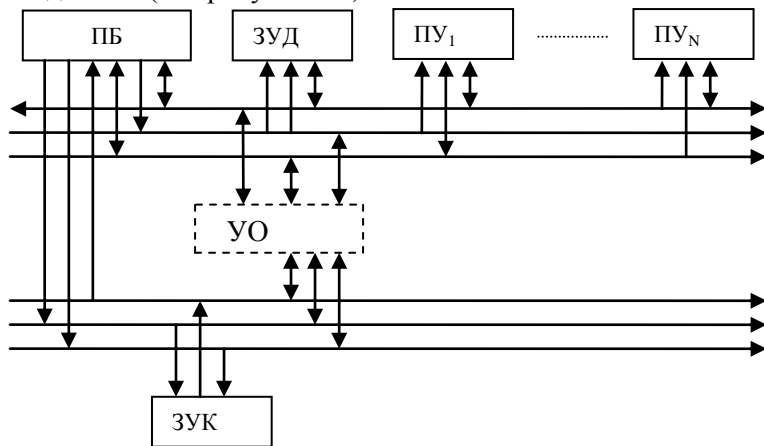


Рисунок 2.3 - Гарвардский принцип.

ЗУД - запоминающее устройство данных;

ЗУК - запоминающее устройство команд;

УО - устройство обмена магистралей данных и команд.

В микроконтроллерах семейства RX фирмы Renesas эту проблему решили объединением гарвардского и фон-неймановского принципов. В них используется 64-разрядная **шина для команд** и **общая магистраль памяти**, что позволило сохранить скорость гарвардской архитектуры и реализовать экономичное распределение ресурсов памяти. [23].

Современные CISC-микропроцессоры и микроконтроллеры используют кэш-память, построенную по гарвардскому принципу

Дальнейшим расширением возможностей гарвардского принципа является применение дополнительных магистралей. В ОМК PIC18xx2 фирмы MicroChip. Используется дополнительная магистраль стека. В сигнальных процессорах для реализации быстрого преобразования Фурье и цифровой фильтрации обычно применяют две шины для чтения данных, одну шину для записи данных и одну шину для чтения команд. Этот подход практикуют производители сигнальных процессоров — фирмы Analog Devices (семейства сигнальных процессоров BlackFin и Tiger Shark), Texas Instruments (семейства сигнальных процессоров C5000™ DSPs и C6000™ DSPs), Freescale (семейства сигнальных процессоров MSC8251 и DSP56K). [7]

При работе в **режиме микропроцессора** **внешняя магистраль** выполняется на основе общих шины данных, адреса, управления. В этом случае гарвардская организация обеспечивает в два раза больший объем адресуемой памяти за счет использования различных сигналов для обращения к памяти команд и данных.

Современные ОМК могут содержать дополнительные шины, работающие параллельно с базовой магистралью и обеспечивающие индивидуальные каналы связи между ОЗУ и периферийными блоками. Наиболее ярким примером является организация магистралей ОМК Xmega [24].

Фон-Неймановская (принстонская) архитектура используется, в основном, в персональных компьютерах и 32-разрядных микроконтроллерах, в которых требуется мобильность и универсальность программного обеспечения.

Гарвардская архитектура применяется в микроконтроллерах для реализации алгоритмов управления, где требуется обеспечить высокую надежность работы аппаратуры и высокую скорость выполнения программы.

Однако Гарвардская архитектура является недостаточно гибкой для реализации некоторых программных процедур. Поэтому выбор архитектуры ОМК следует выполнять в зависимости от класса решаемых задач.

2.2 Центральный процессор

Основным элементом, определяющим быстродействие и производительность ОМК, является центральный процессор ЦП. В качестве ЦП может использоваться хорошо зарекомендовавший себя микропроцессор (как в некоторых ОМК фирмы Motorola). Но наиболее часто ЦП выполняется на основе фирменной разработки.

ЦП относится к классу операционных устройств ОУ и состоит из операционного ОА и управляющего УА автоматов [25].

УА реализует выборку команд и последовательность их выполнения в зависимости от результатов операций.

ОА выполняет команды и формирует признаки результата операции. В состав ОА входят арифметико-логическое устройство, состоящее из определенного набора комбинационных схем, регистры общего назначения РОН, регистры специального назначения, функции которых закреплены для выполнения определенных действий: счетчик адреса команд, указатель стека, регистр маски, регистр управления последовательным портом и так далее.

Повышению быстродействия ОА способствует:

- повышенная разрядность комбинационных схем и регистров;
- распределенный характер АЛУ, при котором допускается одновременное выполнение нескольких микроопераций (число комбинационных схем соответствует числу одновременно выполняемых микроопераций, при этом должна обеспечиваться соответствующую связь между регистрами и комбинационными схемами);
- наличие аппаратных средств для реализации наиболее часто выполняемых функций: деления, умножения, умножения с накоплением, вычисление контрольной суммы CRC и так далее
- большой объем РОН;

- использование регистров с закрепленными функциями, при обращении к которым не требуется указывать их адрес: регистры косвенной и индексной адресации, счетчик команд, и так далее;
- многоаккумуляторная архитектура.

АЛУ с аккумуляторной и многоаккумуляторной структурами представлены на рисунке 2.4 соответственно, а и б. Недостатком аккумуляторного АЛУ является большое число пересылок и пониженное быстродействие, так как на его основе реализуются одноадресные команды, в которых один из операндов и результат операции записывается в аккумулятор АС.

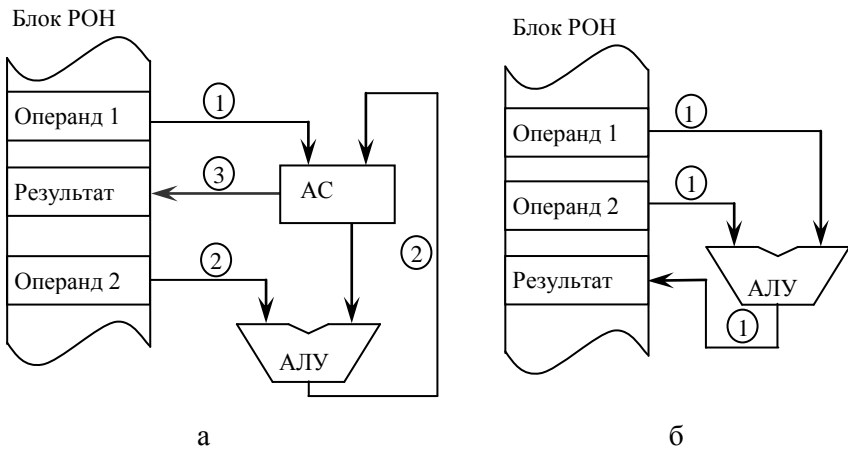


Рис. 2.4. Структура АЛУ: а - с аккумуляторной структурой; б - с многоаккумуляторной структурой

Этот недостаток несколько сглажен в PIC-контроллерах, в которых используется бит модификации, позволяющий записывать результат операции по адресу операнда, указанного в формате команды.

Большее быстродействие обеспечивает АЛУ (рисунок 2.4б), в котором каждый регистр может выполнять функции аккумулятора. На рисунке показано выполнение трехадресной команды. В 8-разрядных ОМК используются одноадресные и двухадресные команды. ОМК повышенной разрядности содержат трехадресные команды.

Реализация ЦП может быть выполнена на основе RISC (сокращенный набор команд), CISC (полный набор команд) или MISC (смешанный набор команд) -архитектур.

Особенностями CISC-архитектуры являются:

- большое количество команд, некоторые из которых могут быть идентичны операторам языков высокого уровня;
- система команд рассчитана на универсальное применение;
- большое количество способов адресации;
- переменная длина команды;
- команды выполняются за несколько тактов системного генератора;
- небольшое количество РОН.

К числу ОМК с CISC-архитектурой относятся MCS-51/52 и его клоны различных фирм, Motorola HC05, HC08, HC11 и другие.

Особенностями RISC- архитектуры являются:

- сокращенный набор команд, который может быть проблемно ориентирован;
- оптимизация системы команд на использование языков программирования высокого уровня;
- фиксированная длина команд, что существенно упрощает аппаратную часть управляющего и операционного автоматов и организацию конвейера;
- разделение команд обработки и обмена с памятью и периферийными устройствами;
- расширенный набор регистров, что позволяет большому объему данных храниться в РОН и упрощает работу компилятора по распределению регистров под переменные;
- логика выполнения команд направлена на аппаратную реализацию УА;
- большинство команд выполняется за один такт системного генератора;
- уменьшение количества и сложности команд позволило сократить аппаратные затраты внутри кристалла и использовать свободное место под размещение других элементов (регистры, многоступенчатый конвейер, кэш и т. д.).

К недостаткам следует отнести меньшее быстродействие и большие затраты памяти *при реализации определенных классов алгоритмов*.

К числу ОМК с RISC-архитектурой относятся PIC-контроллеры фирмы Microchip, AVR фирмы Atmel и другие.

На практике существенные различия между этими архитектурами наблюдались *на ранних стадиях развития ОМК*, например, Intel 8051 (MCS-51) обладал большими функциональными возможностями (255 команд), чем PIC (33 команды).

В настоящее время, если быстродействие и затраты памяти не являются определяющими, различия между MCS-51 и AVR, который является RISC контроллером, практически отсутствуют. Более того, AVR имеет команды обработки 16-разрядных данных, расширенное число команд умножения, больше способов адресации, чем в MCS-51.

В ОМК повышенной разрядности используется MISC-архитектура, в которой добавлены аппаратные средства для реализации ограниченного числа сложных команд (GoldFire, ARM и так далее). [26]

К этому же классу можно отнести и архитектуру Power, которая, обладая всеми признаками RISC-архитектуры, имеет набор команд, ориентированный на суперскалярную обработку (команды распределяются по трем независимым исполнительным устройствам: устройству переходов, устройству с фиксированной точкой и устройству с плавающей точкой.), команды типа процедур и средства для ускоренного выполнения условных переходов. [27]

2.3 Блок синхронизации и управления

Организацию взаимодействия во времени различных устройств ОМК обеспечивает блок синхронизации и управления БСУ и схемы синхронизации, расположенные в периферийных устройствах (последовательных и параллельных портах, таймерах АЦП и так далее).

Основными характеристиками БСУ являются: количество и типы генераторов, значение их выходной частоты, точность установки частоты в требуемых условиях эксплуатации, возможность переключения генераторов в процессе работы ОМК («на лету»), время выхода на номинальную частоту при включении и перезапуске, воз-

нии ОМК, а его значение записывается в калибровочный регистр. Точность установки частоты составляет 1-3%

НГВК - низкочастотный генератор с внешним «часовым» кварцевым резонатором частотой 32768 Гц. Используется для ЧРВ.

ВКВГ - внутренний калиброванный RC-генератор, настроенный на базовую частоту работы ОМК. Обычно этот генератор по умолчанию используется в качестве ИСТЧ. Точность установки частоты составляет 1-3 %

ГВСС - генератор внешнего сигнала синхронизации, который подается на линию подключения кварцевого резонатора XTAL1. При использовании внешнего источника синхронизации его параметры должны быть согласованы с параметрами ОМК (частота, скважность, минимальное значение частоты, уровни напряжения и так далее).

ВГВЧЭ - высокочастотный генератор с внешним частотно-задающим элементом, в качестве которого могут использоваться кварцевый или керамический резонаторы, или RC, RL-цепи. Выход генератора XTAL2 с соответствующим буферным усилителем используют как источник сигналов синхронизации для внешних устройств.

Значение частоты внутренних генераторов может быть установлено в фиксированных диапазонах при конфигурировании ОМК.

Для получения широкого спектра тактовых частот выходные сигналы генераторов могут быть преобразованы с помощью программно-управляемых делителей или умножителей.

Таблица 2. Сравнительные характеристики резонаторов [28]

Наименование	Цена	Размеры	Необх-ость настройки	Откл. част. от номинала	Темп. стаб-ость (на °C)
LC-контур	недорогая	большой	требуется регулировка	+/-2%	10↑-3 - 10↑-4
RC-контур	недорогая	малый	требуется регулировка	+/-2%	10↑-3 - 10↑-4
Кварцевый резонатор	дорогая	большой	не требуется регулировка	+/-0,001%	10↑-6
Керамический резонатор	недорогая	малый	не требуется регулировка	+/- (0,1 -0,5) %	10↑-5

Умножение частот выполняется блоком фазовой автоподстройки частоты *ФАПЧ*, на входе которого может находиться мультиплексор множителя частоты *МУЧ*.

Коэффициент умножения программируется в определенном диапазоне, например, от 1 до 31.

Делители частоты на рисунке не показаны. Они находятся в блоке управления тактовыми сигналами *БУТС* или в периферийных блоках и обеспечивают программно управляемое деление частоты в диапазоне от 1 до 2048 в зависимости от типа ОМК. БУТС формирует из *ИСТЧ*, поступающих на вход мультиплексора *МСС*, сигналы синхронизации для всех блоков ОМК. Настройка сигналов синхронизации выполняется автономно для каждого блока. Некоторые БУТС могут содержать множители частоты, позволяющие работу на частотах, превышающих тактовую частоту ЦП. Независимое тактирование отдельных блоков позволяет оптимизировать энергопотребление ОМК.

В состав БСУ могут входить блок внутренней калибровки *БК*, который сравнивает текущее значение внутреннего генератора с более точной частотой и осуществляет его автоматическую подстройку, и блок мониторинга *БМ*, контролирующий работу внешних источников сигналов. Если пропадает сигнал от внешнего генератора, то БМ аппаратно переключает блок тактирования на работу с внутренним генератором, формируя при этом немаскируемое прерывание.

Управление функциями БСУ выполняется с помощью определенного набора конфигурационных и калибровочных ячеек на стадии программирования ОМК, а в процессе работы - программно-доступными регистрами, число которых может быть от одного (8051) до 17 (XMEGA).

Приведенная структура соответствует более поздним моделям ОМК. Количество генераторов БСУ в различных ОМК может изменяться от одного до четырех и более.

При выборе конфигурации БСУ необходимо учитывать особенности задач, решаемых ОМК.

Меньшие аппаратные затраты обеспечивает использование резидентных генераторов с внутренними времязадающими элементами.

Однако они обладают относительно высокой погрешностью - не менее 1-3%. Подобные генераторы применяют в тех случаях, когда не требуется измерение или формирование частотно-временных сигналов с большой точностью.

Применение генератора с внешним кварцевым резонатором обеспечивает повышенную точность. Его использование целесообразно при измерении, формировании частотно-временных сигналов с большой точностью.

Керамические резонаторы используются, когда необходимы малые габариты и средняя точность. Кроме того, у них меньше время выхода на номинальную частоту при включении и перезапуске, чем у кварцевых резонаторов.

Возможность переключения источника тактового сигнала во время работы ОМК позволяет минимизировать энергопотребление.

Использование механизма калибровки увеличивает точность задания частоты при работе в реальных условиях, отличающихся от условий проведения заводской калибровки, так как частота внутреннего RC-генератора зависит от рабочего напряжения ОМК и окружающей температуры.

2.4 Организация памяти

При анализе характеристик памяти необходимо учитывать:

- расположение памяти (резидентная и/ или внешняя);
- разрядность памяти и ее объем;
- особенности физической реализации памяти (для ОЗУ статическая или динамическая память, для ПЗУ: масочная, однократно программируемая, перепрограммируемая (электрически перепрограммируемая, EEPROM, flash- память, FRAM);
- технические характеристики (последовательный или параллельный интерфейс, быстродействие, особенности временной диаграммы, энергопотребление).

Организация памяти определяется особенностями архитектуры и конфигурацией ОМК.

Структура памяти для архитектуры фон Неймана приведен на рисунке 2.6, где *PCH* - регистры специального назначения центрального процессора и периферийных устройств.

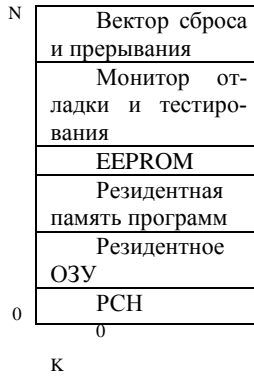


Рис. 2.6. Структура памяти ОМК с архитектурой фон Неймана

Разрядность шины данных памяти (0-K) определяется разрядностью аккумулятора, а её допустимый объем (шина адреса) - разрядностью счетчика адреса команд и регистров, участвующих в формировании адреса. Особенностью организации памяти является единое (общее) адресное пространство для PCN, памяти данных и команд и фиксированные адреса различных функциональных сегментов. Этот рисунок отображает структуру памяти ОМК семейства HC08 фирмы Motorola, работающего в режиме микроконтроллера.

Структура памяти ОМК с гарвардской архитектурой, работающей в режиме микропроцессора, приведена на рисунке 2.7, где РПП, ВПП - соответственно резидентная и внешняя память программ, РПД, ВПД - соответственно резидентная и внешняя память данных, PCN - регистры специального назначения (регистры ввода-вывода), EEPROM - электрически перепрограммируемая память данных.

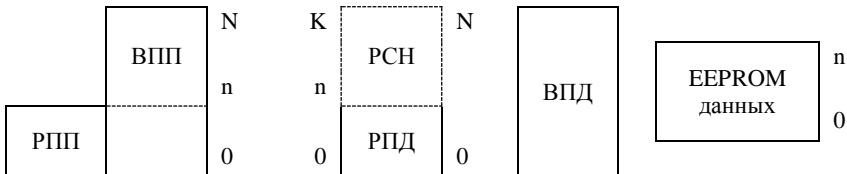


Рис. 2.7. Структура памяти ОМК с гарвардской архитектурой

Особенностью построения памяти в этом случае является возможность различной организации памяти данных и памяти программ.

Организация памяти программ определяется разрядностью счетчика адреса команд (шина адреса) и длиной наиболее часто используемых команд (шина данных). Шина данных может иметь разрядность 8, 12, 14, 16, 32 бита и более. Обычно РПП и ВПП образуют единое адресное пространство. В некоторых ОМК имеется возможность отключения РПП и работа только с ВПП, например, в MCS-51.

Организация памяти данных определяется *разрядностью АЛУ* (шина данных) и *архитектурными особенностями ОМК*.

В MCS-51 РПД и ВПД имеют раздельное адресное пространство. Объем РПД не превышает 256 байт, ВПД - 64 Кбайта, а РСН - 127 байт. В более поздних разработках этого класса ОМК появилась резидентная память данных, которая является частью адресного пространства ВПД.

В ОМК AVR Atmel РПД, РСН, ВПД находятся в общем адресном пространстве.

В состав большинства современных ОМК входит электрически перепрограммируемая память данных EEPROM, имеющая автономное адресное пространство. Эта память обычно используется для хранения информации, которая может изменяться в процессе работы ОМК, но должна сохраняться при выключении питания. Например, градуировочные характеристики датчиков, параметры технологического процесса и так далее.

С целью увеличения быстродействия и сокращения объема программ в РПД выделяются области памяти, обращение к которым выполняется с помощью определенных команд. К ним относятся битовая зона и зона регистров общего назначения РОН.

Зона РОН располагается в начальной области памяти с короткими адресами. Количество РОН невелико - (8-32). Команды, в которых используются РОН, обладают минимальной длиной и максимальным быстродействием. РОН могут быть частью резидентной памяти или выполнены в виде регистров, топологически размещенными в зоне ЦП. В последнем случае обеспечивается большее быстродействие.

В зависимости от решаемой задачи обращение к РОН может быть выполнено как к ячейкам памяти или регистрам резидентной памяти данных может, что снижает объем разрабатываемых программ и увеличивает быстроедействие.

Особенностью битовой зоны является возможность использования группы команд обработки битовых переменных.

Желание работать с короткими адресами памяти данных привело к появлению в ОМК оконной адресации, при которой память разделяется на зоны ограниченного объема (не более 255). Подобный прием искусственно расширяет область ячеек памяти, обращение к которым возможно с помощью короткой прямой адресацией, на все адресное пространство. Реализация оконной адресации похожа на страничную организацию памяти в ЭВМ.

Недостатком оконной адресации является в некоторых случаях необходимость межоконных переходов (такая же проблема существует в страничной адресации), поэтому модификацией этого способа является оконная адресация с регулируемым размером окна, которая используется, например, в MCS -196 [29]. В ОМК нашли применение горизонтальная и вертикальная оконные адресации.

Горизонтальное расположение окон используется в ПС-контроллерах (рисунок 2.8). РОН и РСН отображаются на ограниченное адресное пространство, что позволяет использовать 8-разрядные адресные регистры. Переключение окон (банков) выполняется с помощью специального регистра, а обращение к системным общедоступным регистрам ОДР возможно из любого окна.

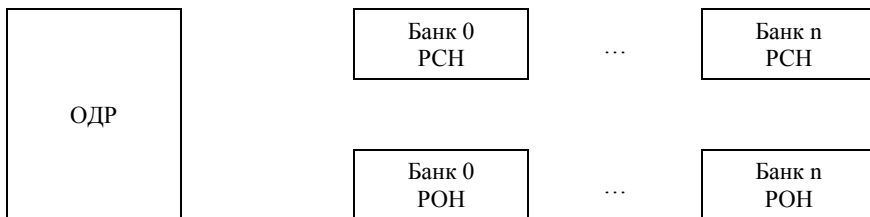


Рис. 2.8. Горизонтальное расположение окон

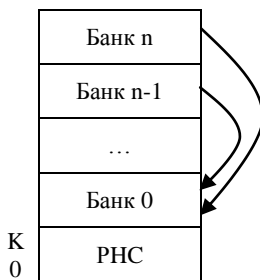


Рис. 2.9 Вертикальное расположение окон

Вертикальное расположение окон используют ОМК MCS-51. В 16-разрядных ОМК MCS-96, MCS-196 (рисунок 2.9) размер окна можно регулировать программно, а общее число окон зависит от объема памяти данных. В этом случае, выделяется окно в нижней части адресного пространства и на него отображается состояние верхних адресов данных.

Таким образом, механизм прямой короткой адресации расширяется на область памяти с 16-разрядными адресами [29].

3. Периферийные блоки

Прежде чем перейти к описанию периферийных блоков, следует вспомнить способы ввода-вывода информации, так как их грамотный выбор позволяет повысить быстродействие и производительность контроллеров на основе ОМК. Под *производительностью* будем понимать *число задач*, решаемых в единицу времени, а *быстродействием - время выполнения* операций.

Основными способами ввода-вывода являются:

- программный ввод-вывод под управлением процессора;
- ввод-вывод по прерыванию под управлением процессора;
- ввод-вывод в режиме прямого доступа к памяти ПДП.

Программный ввод-вывод используется в тех случаях, когда быстродействие периферийных устройств и процессорного блока отличаются незначительно или время, не является определяющим при решении задачи ввода-вывода. Достоинством этого способа яв-

ляются минимальные аппаратные затраты и, в некоторых случаях, повышенное быстродействие.

При выполнении программного ввода/вывода используются два способа обмена информацией: синхронный и асинхронный.

При **синхронном** обмене (рисунок 3.1) скорость ввода-вывода постоянна и определяется быстродействием ведущего и ведомого устройств. Реализация этого обмена может выполняться аппаратно (с формированием сигнала синхронизации СС) или программно - в соответствии с частотой счетчика-таймера или аналогичного устройства. Источником синхросигнала может быть ОМК или внешнее устройство.

К недостаткам синхронного обмена следует отнести пониженную достоверность передаваемой информации, так как отсутствуют сигналы обратной связи; и необходимость настройки частоты синхронизации при изменении длины линии связи или параметров внешних устройств

Синхронный обмен целесообразно использовать в том случае, если конфигурация системы в процессе работы не изменяется, быстродействие устройств, участвующих в обмене, существенно не отличается друг от друга или быстродействие не является определяющим параметром.

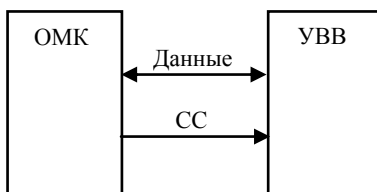


Рис. 3.1. Синхронный обмен

При **асинхронном** обмене (рисунок 3.2) используются сигналы квитирования ($ZГ$ - запрос готовности периферийного устройства $П_j$, $ПГ$ - подтверждение готовности $П_j$), формируемые ведущим и ведомыми устройствами. Ведущим может быть, как микроконтроллер, так и периферийное устройство.

На рисунке 3.2 ведущим является ОМК. В начале обмена ОМК формирует запрос $ZГ$, проверяющий готовность устройства ввода-вывода УВВ. Если устройство готово, оно передает МК сигнал под-

тверждения готовности ПГ. Если устройство имеет выход с тремя состояниями, то начало обмена задается сигналом разрешения РО. **Время ожидания сигнала ПГ необходимо контролировать (например, сторожевым таймером), чтобы предотвратить «зависание» контроллера.**

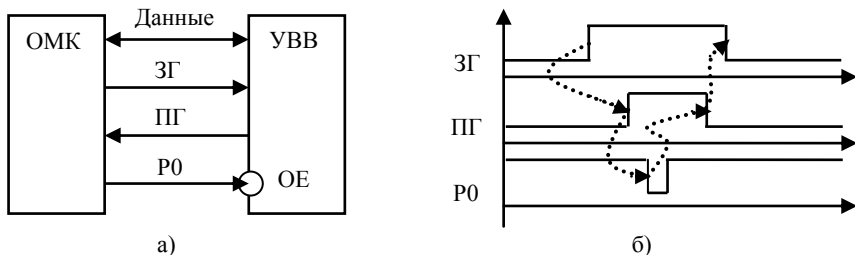


Рис. 3.2. Асинхронный обмен: а - обобщенная схема; б - временная диаграмма

Асинхронный обмен обладает повышенной достоверностью, обеспечивает автоматическую настройку на реальную скорость устройств при изменении их параметров или длины линии связи. Формирование и анализ сигналов квитирования требует дополнительного времени, что снижает скорость ввода-вывода

Возможны варианты асинхронного обмена с использованием только **флага готовности** периферийного устройства.

Формирование и контроль сигналов при программном обмене выполняет процессор, что приводит к его «простою» в циклах ожидания готовности устройств. Снижение скорости ввода-вывода особенно критично в мультипрограммном режиме и/или режиме реального времени.

Для повышения **производительности** ОМК используют **ввод-вывод по прерыванию**.

Прерывание - это реакция системы на программно непредсказуемые события, которые происходят асинхронно с работой процессора и не контролируются текущей программой.

Для реализации прерываний необходим контроллер прерывания и устройства, которые могут функционировать параллельно с процессором, не привлекая его ресурсы.

Время обработки прерываний: $T_{пр} = T_{рз+} + T_{с+} + T_{обр+} + T_{в}$, где $T_{рз}$ - время реакции на запрос (время между появлением сигнала прерывания на входе КП и входом в подпрограмму обработки прерываний); $T_{с}$ - время сохранения параметров прерванной программы; $T_{обр}$ - обработка прерываний; $T_{в}$ - время восстановления параметров прерванной программы и выхода из подпрограммы обработки командой возврата из прерывания.

Использование прерывания снижает быстрдействие операций ввода-вывода, но повышает производительность ОМК.

Уменьшение времени обработки прерывания обеспечивают:

применение «теневого» регистров, когда за каждой подпрограммой обработки прерываний закрепляется *свой* набор регистров. Это приводит к снижению $T_{с}$ и $T_{в}$;

использование аппаратного ядра, выполняющего определенные задачи без привлечения ресурсов процессора (например, блочный ввод-вывод по последовательному каналу) [29].

Быструю реакцию на запрос и увеличение скорости ввода-вывода обеспечивает **режим ПДП**. (рисунок 3.4).

Для его реализации необходимы контроллер ПДП и арбитр (на рисунке не показан). Контроллер ПДП формирует необходимые управляющие сигналы для обмена информацией между памятью и периферийными устройствами, а арбитр - разрешает конфликтные ситуации, возникающие при одновременном обращении к памяти данных процессора и различных устройств. [6]

На стадии инициализации процессор является ведущим устройством. Он настраивает контроллер ПДП на требуемый режим работы и передает основные параметры выбранного режима: вид запроса, одиночная или блочная передача, режим приоритета, начальный адрес ОЗУ и длину передаваемого блока и так далее.

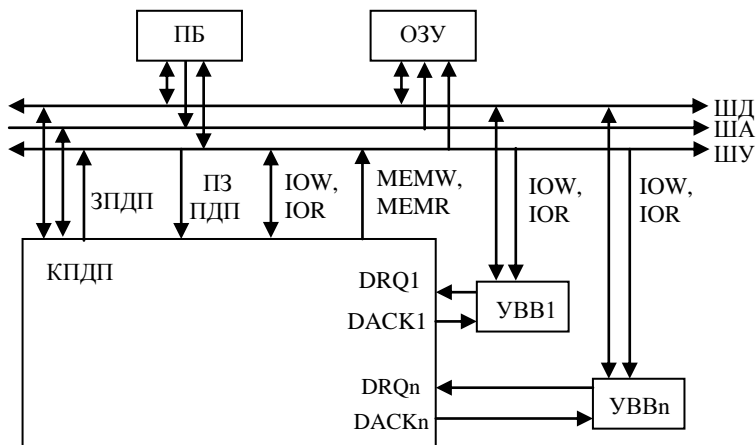


Рис. 3.4. Ввод-вывод в режиме прямого доступа к памяти

С приходом запросов DRQ_n на вход контроллера ПДП выбирается запрос с максимальным приоритетом и формируется сигнал запроса прямого доступа $ЗПДП$. После завершения текущей команды процессор переходит в режим ПДП (отключается от управления магистралью, переводя все выходы в высокоимпедансное состояние, формирует сигнал подтверждение запроса $ПЗПДП$ и передает управление контроллеру). Контроллер ПДП становится **ведущим устройством**, а процессор находится **в состоянии ожидания**. Содержимое внутренних регистров процессора не модифицируются, поэтому его не нужно запоминать в памяти, а затем восстанавливать, как при обработке прерываний

В процессе выполнения ввода-вывода контроллер передает в УВВ сигнал подтверждения ПДП $DACK_n$, переводя тем самым УВВ в **активное** состояние, $ША$ содержит начальный адрес передаваемого (или принимаемого) блока ОЗУ и формируются **одновременно** сигналы чтения памяти/записи в УВВ ($MEMR/IOW$) или ($MEMW/IOR$) при передаче в обратном направлении. Эта процедура продолжается до окончания передачи блока, после которой контроллер ПДП передаст управление процессору, возвращаясь в пассивное состояние.

Контроллер ПДП может использоваться не только для передачи данных между УВВ и памятью, но и из памяти в память, и из УВВ в УВВ.

Во время работы контроллера процессор не имеет доступа к внешним ресурсам. Поэтому, оценивая скорость ввода-вывода в режиме ПДП, следует учитывать быстродействия контроллера, ОЗУ, магистрали и периферийного устройства. Использование режима ПДП наиболее эффективно при блочном обмене информацией, а скорость одиночных передач может оказаться выше при программном вводе-выводе.

В микроконтроллерах может использоваться режим с «захватом цикла», при котором обращение контроллера к памяти выполняется в моменты времени, когда процессорный блок выполняет операции без использования магистралей памяти (взаимодействие с регистрами, кэш памятью). Однако для реализации такого режима требуются дополнительные аппаратные средства, а пересылка данных выполняется в *неопределенные моменты времени*.

Более эффективно использование режима ПДП при наличии нескольких независимых магистралей. Например, в ОМК XMEGA введены дополнительные шины для обмена между внешней и внутренней памятью данных, между периферийными блоками, EEPROM и так далее [20]. В этом случае увеличивается не только быстродействие, но и производительность.

3.1. Параллельные порты

Параллельные порты целесообразно использовать, если требуется согласование с внешними устройствами, передающими информацию в параллельном коде. Их применение обеспечивает более высокое быстродействие по сравнению с некоторыми последовательными портами: UART(USART), SPI, I2C. Наиболее часто параллельные порты используются для работы с внешней памятью.

Число внешних линий ввода-вывода ОМК, в основном, определяется количеством параллельных портов и их разрядностью. В состав ОМК может входить от одного до 8 восьмиразрядных портов.

Основными параметрами параллельного порта являются быстродействие (скорость ввода/вывода данных), схемотехнические осо-

бенности и нагрузочная способность, которые зависят от особенностей информационных и электрических характеристик.

К **информационным** характеристикам параллельного порта относятся:

- разрядность. Обычно 8-ми разрядные, но бывают и меньшей разрядности;

- степень совмещения функций. **Автономные порты** предназначены для выполнения только одной функции, например, цифровой ввод-вывод, **совмещенные** - могут программно перестраиваться на выполнение одной из двух или более функций. В ОМК используются, в основном, совмещенные порты;

- направление передачи данных. **Однонаправленный** порт не может изменять направление передачи и предназначен **только для ввода или вывода** информации, в **двунаправленном** - направление передачи изменяется **аппаратно** при выполнении команд пересылки данных (например, команда MOVX в ОМК MCS-51), направление передачи в **квазидвунаправленном** порте задается **программно** в соответствующем регистре управления;

- возможность побитового управления. Обычно порты находятся в битовом пространстве ОМК, что существенно упрощает процедуру анализа и формирования битовых переменных.

К **электрическим** характеристикам портов относятся нагрузочная способность по постоянному и переменному токам, особенности реализации входных и выходных линий порта, быстродействие. Эти характеристики определяются типом технологии, на базе которой выполнен ОМК, и структурой аппаратных средств.

Аппаратные средства должны обеспечить согласование ОМК с внешними устройствами при минимальном количестве дополнительных элементов.

Поэтому наиболее оптимальная структура должна содержать регистры для хранения входной и выходной информации, программно управляемые средства, позволяющие подключение внутренних резисторов, обеспечивающих согласование с необходимыми источниками сигналов, переход в высокоимпедансное (третье) состояние, изменение направления передачи.

Обобщенная структура параллельного порта представлена на рисунке 3.5, где R_2H - регистр направления передачи, $R_2D_{вых}$, $R_2D_{вх}$ -

соответственно регистр выходных и входных данных, РгК - регистр конфигурации, Бувх, Бувых - соответственно входной и выходной буферные усилители, ТШ - триггер Шмидта, R0, R1- «подтягивающие резисторы» к уровням нуля и лог.1.

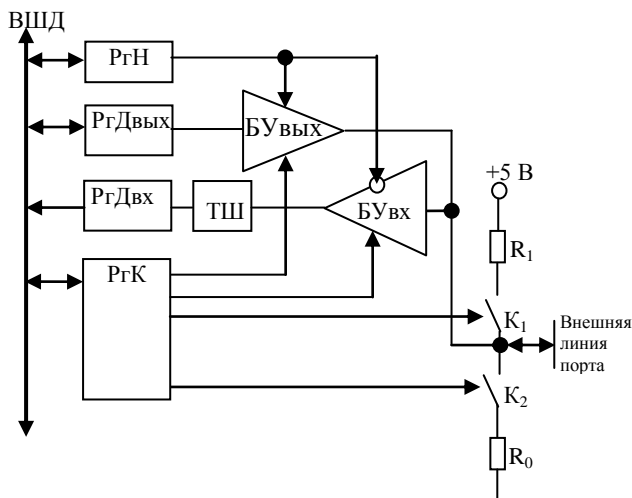


Рис. 3.5. Обобщенная структура параллельного порта

Входные цепи порта характеризуют входные сопротивление и емкость, возможность присоединения «подтягивающих» резисторов, наличие входных формирователей.

Входные формирователи выполняются на базе триггера Шмидта. Триггер Шмидта - пороговый элемент с петлей гистерезиса. Его особенностью является формирование импульсов с крутыми фронтами и срезом, что приводит к снижению влияния помех и повышению надежности работы. Фактически он является временным нормализатором входного сигнала - импульсы с различными фронтами и срезом на выходе триггера имеют унифицированный вид.

Выходные цепи могут иметь выход по току, по напряжению, иметь возможность присоединения «подтягивающего» резистора, переходить в третье состояние, особенно необходимое при работе с внешними магистралями.

Нагрузочная способность по постоянному току выходных буферных усилителей может быть от 1,5 мА до 24 мА, с повышенной нагрузкой свыше 24 мА. При высоких скоростях обмена необходимо дополнительно контролировать нагрузочную способность по переменному току, которая, как правило, задается в виде допустимой емкости $C_{\text{допуст}}=10\text{-}50\text{пФ}$.

При повышенных требованиях к нагрузке необходимо использовать внешние буферные усилители.

Регистр конфигурации РК управляет изменением схмотехники линий ввода-вывода порта. Наиболее часто это управление сводится к присоединению к линиям резисторов R0, R1. Резистор R1 необходим при согласованиях со схемами с «открытым» коллектором (или стоком), при подключении кнопок, тумблеров, контактов реле и так далее. R0 может служить нагрузочным резистором при согласовании с источниками напряжения.

В некоторых ОМК (например, в PSoC, xMEGA) возможно до 8 различных способов организации выходных линий. Каждый вывод может быть сконфигурирован как Push-Pull (двухтактный каскад с симметричной нагрузочной способностью), монтажное «И», монтажное «ИЛИ» с подтягивающими резисторами к шине питания или земле [11, 24].

Быстродействие - минимальное время, за которое может быть принята (в регистр) или передана (из регистра) информация. Оно определяется временем выполнения команд ввода-вывода. Лучшими характеристиками обладают порты, содержащие RгДвх., однако входной регистр присутствует у ограниченного числа ОМК.

В то же время возникает необходимость организации взаимодействия с быстродействующими процессами, для которых время выполнения команд ввода-вывода не является удовлетворительным. В этом случае необходимо использовать внешние регистры или триггеры, которые обеспечивают сохранение данных на время выполнения команды. При наличии внутреннего регистра- защелки необходимость в подобных регистрах отсутствует.

Программная модель параллельного порта содержит от одного (MCS-51) до четырех регистров (AVR Atmel).

Большинство параллельных портов восьмиразрядных ОМК не имеют в полном объеме приведенную структуру. Поэтому в ряде

случаев приходится использовать дополнительные программные и аппаратные средства, что усложняет разрабатываемое микропроцессорное устройство.

Ввод-вывод информации через параллельный порт может быть организован побитно и побайтно.

Битовые данные используются при программной реализации протоколов обмена по последовательным интерфейсам, формировании и анализе управляющих сигналов.

3.2. Последовательные порты

Последовательные порты используются для согласования ОМК с устройствами, поддерживающими последовательный обмен, передачи информации по внутренним или внешним интерфейсам контроллеров с целью сокращения аппаратных затрат: приемопередатчиков, устройств сопряжения с интерфейсами различного назначения, линий связи.

Необходимо учитывать, что в ОМК скорость ввода/вывода по последовательному каналу зависит от особенностей блока синхронизации и управления. В большинстве применений тактовый генератор ОМК *является общим источником синхронизации* параллельного и последовательного портов. Поэтому скорость ввода-вывода последовательных каналов ниже, чем параллельных.

При наличии независимой настройки тактовой частоты и использовании умножителей скорость последовательных каналов может быть выше, чем у параллельных.

Количество последовательных портов определяется назначением ОМК. В ОМК общего применения количество портов от 1 до 3. Наиболее часто «джентельменский набор» включает: UART (универсальный асинхронный приемопередатчик, работающий в большинстве ОМК и в синхронном режиме), I2C, SPI. Дополнительно могут использоваться порты, поддерживающие протоколы обмена USB и промышленного интерфейса CAN. В специализированных (коммуникационных) ОМК число последовательных портов может достигать 8-11.

Обобщенная структура последовательного порта представлена на рисунке 3.6.

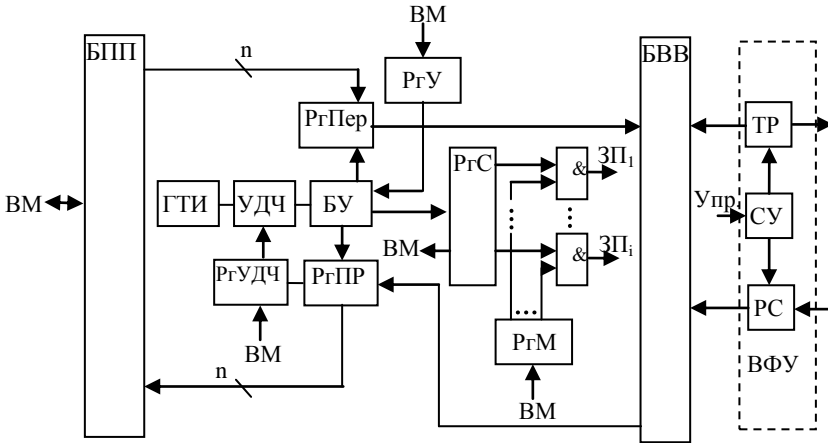


Рис. 3.6. Обобщенная структура последовательного порта

Работа последовательного канала начинается с процедуры инициализации, в процессе которой задают режим работы, скорость, формат обмена, способ ввода-вывода и разрешают прием и/или передачу информации.

Основная информация записывается в регистр управления $P2U$, а при задании скорости дополнительно используется управляемый делитель частоты $УДЧ$, на вход которого поступает частота от генератора тактовых импульсов $ГТИ$ ОМК.

Реализация $УДЧ$ может быть различной. В 8051 для этих целей используется счетчик-таймер 1, в AVR - автономный делитель частоты и так далее.

При передаче данные загружаются в регистр передатчика $P2Per$. Преобразование параллельного кода в последовательный начинается, если установлен бит разрешения. В процессе преобразования к данным добавляются дополнительные биты в соответствии с реализуемым протоколом передачи. Завершение передачи индицируется флагом готовности к следующему циклу, который устанавливается в регистре состояния. $P2S$.

При приеме данных необходимо предварительно установить бит разрешения, после чего происходит анализ входной информации на соответствие заданного формата, формирование поля данных, пре-

образование входных данных в параллельный код в $PzPr$ и передача его на внутреннюю магистраль ВМ ОМК.

Буфер приемопередатчика $BПП$ может содержать один общий регистр (8051) или два, один из которых служит буфером передатчик, а другой - приемника (AVR).

В регистре состояния PrC дополнительно устанавливаются флаги, характеризующие качество приема (бит паритета, правильность контрольной суммы, переполнение буферов, ошибка кадра и так далее).

В зависимости от способа ввода/вывода флаги, формируемые в регистре состояния, могут анализироваться программно или по прерыванию.

При использовании прерываний управление приоритетами между запросами выполняет регистр маски PzM и конъюнкторы, на выходе которых формируются запросы прерывания Zni .

Запись/чтение регистров порта выполняется через внутреннюю магистраль BM под управлением процессорного блока.

Координацию всех действий при работе порта выполняет блок управления $БУ$.

В некоторых 16-разрядных ОМК $БУ$ может содержать собственные средства обработки, позволяющие порту функционировать без привлечения ресурсов центрального процессора. [29]

Входные и выходные уровни сигналов последовательного порта BBB (буфер ввода-вывода) определяются напряжением питания ОМК. Поэтому при согласовании порта с определенными интерфейсами необходим соответствующий внешний формирователь уровня $ВФУ$, состоящий из трансивера TP , ресивера PC и схемы управления CV . Подобное согласование необходимо в интерфейсах RS-232, RS-485, CANbus, USB.

Приведенная структура (рисунок 3.6) носит гипотетический характер. В зависимости от типа последовательного порта состав регистров и место размещения управляющей информации может существенно отличаться.

Например, в 8051 программная модель порта UART состоит из двух регистров: регистра приемопередатчика $SBUF$, регистра управления и состояния $SCON$. Задание скорости обмена, управление

прерываниями осуществляется блоками, функционально не относящимися к последовательному порту.

В микроконтроллерах AVR семейства Mega программная модель порта включает 6 регистров: данных UDR, который состоит из регистров передатчика и приемника, управления UCSRC, UCSRB, состояния UCSRC, скорости обмена UBRRH:UBRRL.

В порте USB количество программно доступных регистров достигает 14.

Таким образом, программная модель изменяется в широких пределах, но везде в качестве базовых элементов присутствуют регистры управление, данных, состояния, скорости обмена, которые могут быть одно - двухбайтовые. Топологически эти регистры могут носить распределенный характер.

Основной характеристикой последовательного порта является **пропускная способность или скорость обмена**. Обычно в качестве этой величины указывается **пиковая скорость**, которая определяется структурой, электрическими характеристиками порта и линий связи. Скорость обмена указывается в бит/с, Кбит/с, Мбит/с.

На практике больший интерес представляет скорость передачи **данных**. Однако **поле данных** является только **частью кадра**, обобщенная структура которого представлена на рисунке 3.7. Кроме того, необходимо учитывать используемые способы повышения достоверности информации (битстаффинг, избыточное кодирование и так далее). [30,31]

старт	адрес	управление	данные	контроль	стоп
-------	-------	------------	--------	----------	------

Рис. 3.7. Обобщенная структура кадра

Наиболее простым является кадр RS-232 (10-11 бит). В других случаях длина кадра может быть от нескольких байт до нескольких килобайт. Поэтому реальная скорость передачи **данных** определяется отношением длины поля данных к длине кадра. Если длина кадра - 8 байт, пиковая скорость - 1 Мбит/с, длина поля данных - 2 байта, то скорость передачи **данных** будет 0,25 Мбит/с.

Для оценки потенциальных характеристик последовательного канала необходимо анализировать:

Направление обмена. Порты с симплексным обменом способны поддерживать только однонаправленные передачи, дуплексные - обеспечивают **одновременный** прием и передачу, полудуплексные - передают информацию в двух направлениях, но в **режиме с разделением времени**.

Режим работы. В **синхронном** режиме передача или прием каждого бита данных сопровождается формированием на дополнительной линии синхронизирующего импульса. В некоторых портах, работающих в синхронном режиме, линия синхронизации в явном виде может отсутствовать, но в структуре кадра передается поле (USB) или интервал (CANbus) синхронизации, за время передачи которого выполняется автоматическая подстройка частоты устройствами, участвующими в обмене. Подобный режим можно назвать **квазисинхронным**.

К синхронному режиму можно отнести обмен с использованием самосинхронизирующих кодов (RZ, Manchester II), но для его реализации требуются специальные формирователи.

Достоинством синхронного обмена при наличии линии синхронизации являются пониженные требования к стабильности генератора и высокое качество синхронизации, что особенно важно при передаче сообщений с длинными участками повторяющихся бит, и повышенная скорость.

Сигнал синхронизации *СС* формирует ведущее устройство (рисунк 3.8 а)

В **асинхронном** режиме линия синхронизации отсутствует, а согласование скоростей обеспечивается генераторами синхроимпульсов (*ГСИ*), находящимися в месте расположения приемника и передатчика (рисунк 3.8 б).

Частота *ГСИ* обычно более чем в 16 раз больше частоты передачи данных. Во время передачи бита выполняется 16-кратный опрос, а значение определяется по правилу «2 из 3» в середине 16-битовой серии.

Значения частот должны быть согласованы с точностью (0,5-1) % в зависимости от скорости обмена и быть стабильными за время передачи кадра.

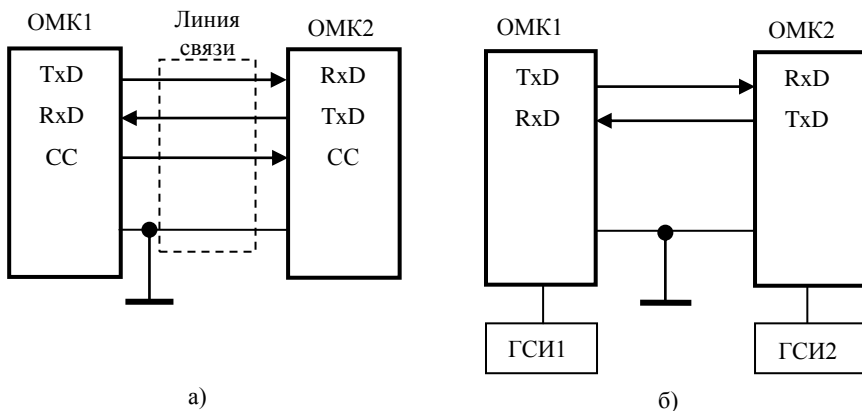


Рис. 3.8. Дуплексный обмен: а - синхронный; б - асинхронный

Формат кадра. Формат кадра характеризуется длиной, количеством и размером полей, возможностью программного изменения структуры кадра (рисунок 3.7).

Метод кодирования последовательной информации. В ОМК используется, в основном, код без возврата к нулю (*NRZ*). В портах USB применяется инверсный код без возврата к нулю (*NRZI*).

NRZ отображает биты информации последовательностью соответствующих уровней напряжения, не изменяющих значения за время передачи бита. Недостатком этого кода при отсутствии линии синхронизации является возможность потери синхронизации при передаче длинной последовательности бит, имеющих одно значение.

В *NRZI* логическая единица передается отсутствием изменения предыдущего значения бита, а логический ноль - инверсией этого значения. Таким образом, устраняется потеря синхронизации при передаче длинной последовательности нулей. При передаче длинной последовательности единиц, как и в коде *NRZ*, необходимо использовать методы, повышающие качество синхронизации.

Одним из распространенных методов является процедура **битстаффинга**. При передаче 5 повторяющихся бит в сообщение добавляется бит противоположного значения, а при приеме производится повторный контроль. Если приняты 6 бит одного значения,

то в этом случае фиксируется сигнал ошибки. В высокоскоростных интерфейсах может для этих целей использоваться кодирование «*8b/10b*», особенностью которого является представление байта в виде 10 бит. Информационная избыточность позволяет перекодировать значение байта, исключая длинные повторения одинаковых бит, а неиспользуемые комбинации служат для кодирования различной служебной информации. [30]

Методы повышения достоверности информации. Достоверность передаваемой информации повышается за счет использования информационной и временной избыточности. В сообщение добавляется некоторая дополнительная информация, обработка которой позволяет обнаружить и/или исправить ошибку.

Наиболее простым способом является использование **бита паритета**. В процессе передачи к сообщению добавляется бит, который дополняет количество единиц в кадре до четного или нечетного значения. В приемнике повторно определяется бит паритета и сравнивается с переданным значением. По результату сравнения формируется флаг, который обрабатывается центральным процессором. Подобный способ контроля не обнаруживает кратные ошибки. Поэтому он применяется в портах UART, использующих простые протоколы обмена.

Лучшие характеристики обеспечивает применение **контрольной суммы по модулю N** , где N представляет собой число кратное степени 2. Передаваемое сообщение суммируется побайтно без учета переполнения, а в приемнике происходит повторное формирование контрольной суммы и её проверка. Однако и в этом случае велика вероятность необнаруженных ошибок, так как суммирование выполняется **побайтно**.

Наиболее надежным способом является контрольная сумма, определяемая по методу **циклического избыточного кодирования CRC**, особенностью которого является участие **каждого бита** передаваемого сообщения в формировании контрольной суммы. В этом случае контролируемая информация преобразуется в битовый поток, который делится на характеристический полином. Остаток от деления и является контрольной суммой. При вычислениях используется арифметика «Сумма по модулю два (Исключающее ИЛИ)» Этот

способ является самым распространенным в современных последовательных интерфейсах, включая USB и CANbus [31].

В процессе обмена могут использоваться и другие способы повышения достоверности: контроль структуры кадра, переполнения буфера приема и передачи, бита подтверждения приема и передачи информации (I²C), пакета подтверждения приема и передачи информации (USB) и так далее [32,33].

Структурные особенности последовательного порта. В состав ОМК могут входить порты, выполняющие функции ведомого (Slave) и/или ведущего (Master) устройств.

В интерфейсах, реализующих режим “Master-Slave”, одно ведущее устройство обеспечивает обмен с несколькими ведомыми.

В интерфейсах с несколькими ведущими необходимо анализировать наличие средств, обеспечивающих арбитраж при попытке занятия магистрали одновременно несколькими устройствами.

Поэтому при выборе ОМК необходимо контролировать режим работы порта (master, slave), возможность его программного управления и наличие аппаратных средств, необходимых для реализации требуемого протокола обмена. Особенно это важно в портах **SPI, I2C, USB, CANbus**.

Согласование скорости обмена между внешними устройствами и внутренними магистралями ОМК, минимизирующее потери информации при вводе-выводе, осуществляется с помощью **буферной памяти, работающей по принципу FIFO**. Буферная память может быть установлена как на входе, так и на выходе порта.

При работе порта наиболее часто используют программный ввод-вывод и по прерыванию. В коммуникационных ОМК используются каналы прямого доступа к памяти ПДП.

Последовательные порты могут работать в автономном или совмещенном режимах.

Автономные порты функционирует одновременно, что обеспечивает большее суммарное быстродействие.

При работе в совмещенном режиме функции портов разделены во времени и настройка на требуемый протокол обмена реализуется программно. Наиболее часто используются совмещенные порты **USI**, предназначенные для работы с SPI и I2C.

Реализация обмена по интерфейсам RS-232 и SPI требует применения дополнительных сигналов (кроме данных и синхронизации). Например, управление модемом в RS-232, выборка ведомых устройств SPI. В ОМК эти сигналы могут формироваться аппаратно, что увеличивает скорость ввода-вывода. Если они отсутствуют, то следует использовать программное управление.

Желательно наличие автономного источника сигналов синхронизации последовательного порта, не связанного с тактовым генератором ОМК.

У некоторых ОМК в состав порта входят сопроцессоры ввода-вывода, выполняющие не только ввод-вывод информации, но и её обработку независимо от ЦП.

Электрические характеристики последовательного порта аналогичны параллельному, так как они используют одни и те же линии. Однако возможны нюансы при конфигурации порта.

3.2.1. Последовательный порт UART (USART)

Порт UART (УАПП - универсальный асинхронный приемопередатчик) является самым распространенным в ОМК последовательным каналом передачи данных. Он применяется практически во всех микроконтроллерах. Это связано с тем, что протокол UART используется на физическом уровне во многих последовательных интерфейсах: RS-232, RS-422, RS-485, Modbus, Vitbus, Profibus, IrDA и так далее. В некоторых ОМК (например, в MCS-51/52) UART может выполнять и синхронный обмен. В микроконтроллерах фирмы Atmel синхронно-асинхронный приёмопередатчик имеет аббревиатуру USART. Выход UART обозначают TxD (transmitted data), вход - RxD (received data).

Формат кадра асинхронного обмена включает: стартовый бит, поле данных, длина которого может быть от 5 до 9 бит, необязательный бит паритета (четность или нечетность), один или два стоповых бита (рисунок 3.9). Второй стоповый бит используется, если необходимо увеличить межкадровый интервал.



Рис. 3.9. Формат кадра асинхронного обмена

При отсутствии данных входные и выходные линии находятся в состоянии логической 1. Стартовый бит всегда логический 0, поэтому приёмник UART ждёт перепада из 1 в 0, а затем начинается прием кадра. Состав кадра и его длина задаются при инициализации порта. Стоповый бит (или биты) всегда равны логической 1. Передача выполняется младшими разрядами вперед.

Длительность всех битов одинакова и определяется скоростью передачи. Существует ряд стандартных скоростей передачи: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 б/с. Если *внутри одного устройства* связь можно осуществлять на произвольной скорости, то *для связи с внешними устройствами следует придерживаться стандартных величин*. Отсутствие рассинхронизации обеспечивается стабильностью тактовых генераторов приемника и передатчика, которая должна быть не хуже (0,5-1) %.

При обмене данными по каналам связи используются три метода передачи данных: симплексная, полудуплексная, дуплексная.

При выборе порта UART необходимо контролировать:

- Наличие буферных регистров приемника и передатчика;
- Возможность программирования структуры кадра и режимов работы;
- Особенности схемы формирования тактовых сигналов (автономная или использует счетчики-таймеры);
- Наличие аппаратных схем контроля ошибок (переполнение буферов, нарушения паритета, формата кадра);
- Количество векторов прерываний, формируемых в процессе работы;
- Возможность работы в синхронном режиме и его особенности (формат кадра, дуплексный или полудуплексный обмен).

К недостаткам порта можно отнести: неэффективная структура кадра (в 10-битовом кадре 2 или 3 служебных бита), что *снижает скорость передачи данных*, невысокая скорость в асинхронном режиме, использование бита паритета не позволяет обнаруживать четные ошибки.

3.2.2. Интерфейс SPI

Интерфейс SPI находит широкое применение как средство коммуникации между элементами цифровых систем.

Преимуществом SPI по сравнению с другими последовательными интерфейсами является легкость реализации. Устройства этого интерфейса представляют собой сдвиговые (универсальные) регистры со схемами управления, формат обмена - байт или слово, семантику которых определяет пользователь. [34]

Поэтому SPI поддерживает большое число производителей электронного оборудования. В этом стандарте выпускаются микросхемы памяти, мосты, ЦАП, АЦП и так далее. Интерфейсом SPI оснащены 85% однокристальных микроконтроллеров. При отсутствии SPI в ОМК его несложно реализовать программно.

Разработчик интерфейса SPI — фирма Motorola.

Интерфейс *синхронный, дуплексный, принцип обмена «ведущий-ведомый»*. Наиболее часто используется одно ведущее устройство, которое работает с несколькими ведомыми. Возможна работа в режиме мультимастер, но для его реализации требуются специальные схемные решения. Базовая топология - шина. Скорость обмена - несколько Мбит/с и более.

В состав интерфейса входят следующие линии: *СКК* - линия синхронизации, *MISO (SDI)* - вход ведущего устройства/выход ведомого, *MOSI (SDO)* - выход ведущего устройства/вход ведомого, *SSn* - выборка n-ого ведомого устройства.

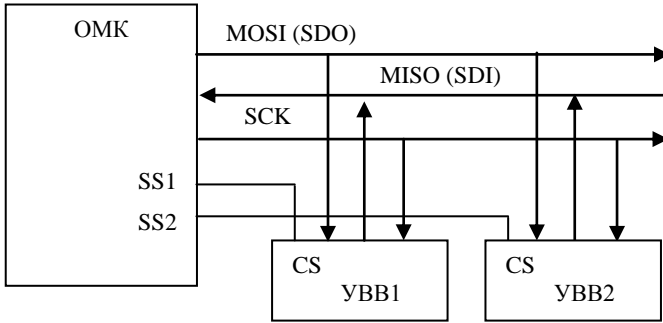


Рис. 3.10. Подключение устройств ввода/вывода к SPI

Схема соединения SPI представлена (рисунок 3.11), где в качестве ведущего устройства используется ОМК, а ведомых - устройства ввода-вывода УВВ.

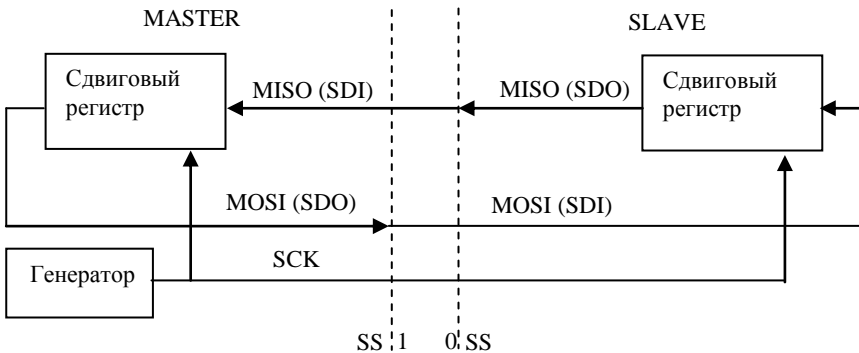


Рис. 3.11. Процедура обмена «master-slave» в SPI

Статус ведущего/ведомого устройств может быть задан программно или аппаратно. В последнем случае на входе SS ведущего устройства устанавливается, например, уровень 1, а ведомого - уровень 0.

В функции ведущего устройства входят: формирование сигналов синхронизации SCK, выборки SS, передача/прием информации. Принцип обмена информацией поясняет рисунок 3.10

Линии MISO, MOSI объединяют регистры ведущего и ведомого в кольцевой (циклический) сдвиговый регистр. После записи данных в

регистр ведущего устройства формируется сигнал SCK, под действием которого выполняется обмен информацией между регистрами. Время выполнения дуплексного обмена определяется разрядностью регистров сдвига (8 или 16 тактов SCK). По окончании обмена в ОМК устанавливается флаг готовности, на основании анализа которого выполняется чтение/запись новой информации.

В ряде случаев требуются передачи *только в одном* направлении. Например, при управлении ЦАП данные передаются от ведущего к ведомому, а при чтении внешней флеш-памяти - от ведомого к ведущему. В первом случае следует отключить линию MISO, во втором - производят запись любой информации в буфер передатчика ведущего устройства, инициируют обмен при отключенной линии MOSI, а после окончания обмена считывают значение принятого байта.

Для согласования с ведомыми устройствами требуются различные режимы синхронизации. Ведущие устройства могут генерировать синхроимпульсы различной полярности с активным фронтом или срезом. По заданному активному перепаду напряжения выполняется захват очередного бита, а по противоположному - сдвиг.

При выборе контроллера SPI в ОМК следует анализировать:

- автономный контроллер или совмещенный. В некоторых омик контроллер может выполнять функции SPI или I2C в режиме с разделением времени;
- возможность работы в режиме ведущего и ведомого;
- допустимую частоту SCK и программное управление частотой;
- направление обмена (младшими разрядами вперед или старшими);
- наличие сигналов SS и способов их формирования;
- режимы синхронизации;
- реализуемые способы ввода-вывода (программный, по прерыванию, режим ПДП)
- наличие буферов приемника и передатчика или буферов FIFO;
- возможность отключения контроллера с целью уменьшения энергопотребления;

- особенности настройки линий параллельного порта, используемых контроллером SPI, на ввод/вывод (аппаратно или программно).

К недостаткам SPI следует отнести относительно большое количество линий, которое зависит от числа ведомых устройств, отсутствие сигналов подтверждения приема/передачи информации, относительная сложность организации мультипроцессорного режима.

3.2.3. Интерфейс I2C

Последовательный интерфейс I2C, разработанный фирмой Philips, широко используется в микроконтроллерах для организации обмена в цифровом коде. Название шины I2C (или IIC) является сокращением её англоязычного наименования: «Inter-Integrated Circuit» - объединение интегральных схем. Область его применения аналогична SPI: АЦП, ЦАП, память, мосты, часы реального времени, бытовая электроника и так далее. Этот интерфейс может использоваться и с другими наименованиями: 2-Wire, TWI.

На основе I2C разработаны интерфейсы ACCESS.bus, который использовался для подключения к персональным компьютерам типовой периферии, и SMBUS - для управления питанием, конфигурирования памяти и других задач в системных платах современных ПЭВМ [35]

Интерфейс I2C синхронный, полудуплексный. Базовым источником информации при написании этого раздела является перевод фирменного стандарта, рисунки из которого используются в тексте [36]

Интерфейс содержит две сигнальные линии, одна из которых SDA предназначена для передачи данных, а вторая - сигнала синхронизации SCL.

Режимы работы: low-speed -100 КГц, fast-speed -400 КГц, HS-mode -4,3 МГц. Отличие I²C-абонентов, работающих в режиме fast-speed, состоит в наличии входных формирователей фронтов на основе триггеров Шмитта на линиях SDA и SCL. Обмен же информацией в режиме HS-mode требует внесения изменений, в том числе, в формат передачи данных. [35]

Далее будут рассмотрены особенности первых двух режимов, которые используются наиболее часто.

Приемопередатчики выполнены на основе схем с открытым стоком (или коллектором) и реализуют функцию «Монтажное И». Поэтому для функционирования интерфейса необходимы «подтягивающие» резисторы R_p , величина которых определяется длиной линии связи, скоростью передачи данных, значением напряжения питания и может изменяться в диапазоне (1-10) КОм.

Для обеспечения требуемой скорости и параметров импульсов вводятся ограничения на входные емкости устройств и общую суммарную паразитную емкость. Для частоты 100 КГц эти величины составляют соответственно (5-10) пФ и 400 пФ. Интерфейс используется, в основном, для внутримодульного (внутриплатного) обмена. При использовании интерфейса для связи с внешними устройствами дополнительно требуется линия «земли». Допустимая длина линий связи зависит от требуемой скорости обмена.

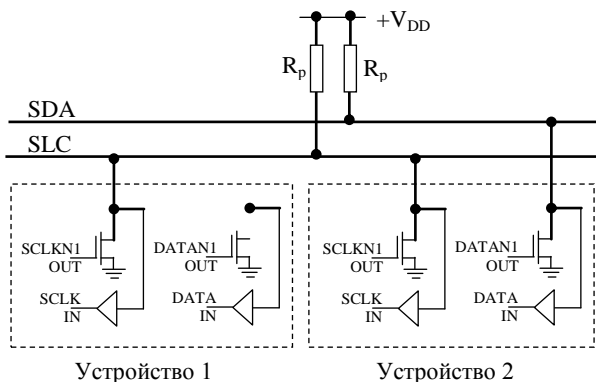


Рисунок 3.12. Схема соединения устройств I2C

Устройства I2C могут быть ведущим (Master) или ведомым (Slave). Ведущее устройство управляет формированием протокола обмена и генерирует тактовый сигнал SCL, ведомое- «прослушивает» интерфейс и, распознав свой адрес, выполняет команды ведущего. Наиболее часто используется режим с одним ведущим и несколькими ведомыми. Однако интерфейс поддерживает работу и с несколькими ведущими (режим multiMaster).

Формат кадра представлен на рисунке 3.13.

START	ADR	R/ W*	ACK*	D	ACK*		D	ACK*	STOP
-------	-----	-------	------	---	------	--	---	------	------

Рис. 3.13. Формат кадра I2C

Особенности формирования начала (START) и конца кадра (STOP) иллюстрирует рис. 3.14. В состоянии ожидания на линиях установлен высокий уровень. В начале кадра ведущее устройство формирует *срез* импульса на линии SDA при *единичном* значении SCL, а в конце - *фронт*.

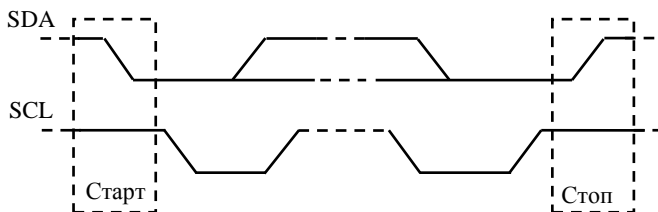


Рис. 3.14. Сигналы СТАРТ и СТОП

После старта ведущее устройство передает адрес ведомого ADR и бит направления обмена R/W*(чтение R -высокий уровень, W* - низкий). Адрес может быть 7-разрядный или 10-разрядный. Появление 10-разрядного адреса связано с большим числом выпускаемых устройств, поддерживающих интерфейс I2C. Если в первом случае ADD+R/W* занимает 1 байт, то во втором -два байта. Признаком 10-битовой адресации является код 11110, расположенный в старших разрядах старшего байта. *Адреса ведомых устройств, выпускаемых серийно, фиксируются при их проектировании и согласуются со специальной международной организацией.*

В некоторых случаях пользователю предоставляется возможность управлять значением двух-трех младших адресов.

Поле адреса может также использоваться как средство кодирования управляющей информации, как идентификатор устройств и так далее. Для этих целей резервируется часть адресного пространства. Например, 0000000 - общий вызов, 0000010 - зарезервирован для других форматов интерфейса, 11111xx - зарезервирован на будущее и так далее.

Передача/прием *каждого байта* завершается формированием и анализом бита подтверждения принятой информации АСК. Этот сигнал должны формировать и анализировать как ведущее, так и ведомое устройства.

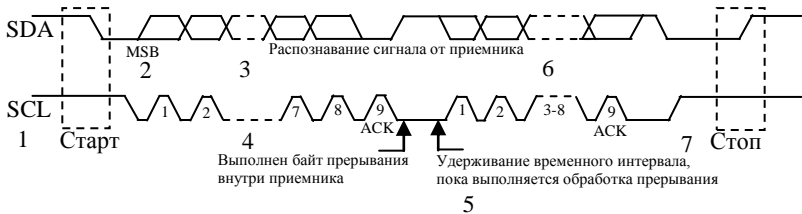


Рис. 3.15. Пересылка данных по шине I²C: 1 - сигнал СТАРТ; 2 - старший разряд байта; 3 - сигнал подтверждения от приёмника; 4 - прием байта завершен, прерывание внутри приемника; 5 - синхрония удерживается в низком состоянии, пока обслуживается прерывание; 6 - сигнал подтверждения от приемника; 7 - сигнал СТОП

Ведущее устройство начинает обмен, формируя данные и синхроимпульс. (рис. 3.15) Передача данных начинается по первому положительному импульсу на линии SCL, которым стробируется *старший бит* данных. **По фронту** SCL ведомое устройство фиксирует данные, а очередной бит данных должен быть сформирован **при низком уровне** SCL. Линия SDA ведомого устройства в момент приема информационных битов должна быть выставлена в единичное состояние. После формирования среза *восьмого* импульса на линии SCL ведомое устройство должно выставить на линию SDA нулевой уровень, подтверждая прием байта данных. По фронту девятого импульса ведущее устройство считывает состояние АСК, а по срезу - ведомое устройство выставляет высокий уровень.

Если ведомое устройство не успело выполнить требуемую функцию, то оно может задержать передачу АСК в линию данных на *определенное* время или удерживать нулевой уровень на линии синхронизации. Эти задержки определяются режимом работы интерфейса и могут составлять десятки-сотни мкс. Если АСК=1, то ведущий должен прервать передачу. При **приеме данных** АСК должно формировать ведущее устройство.

Длина поля данных стандартом не ограничивается, но следует учитывать, что длинное поле данных увеличивает время доступа к

интерфейсу, так как арбитраж между ведущими устройствами запрещен между START и STOP.

В процессе передачи кадра возможно многократное использование сигнала START без изменения STOP-условия (*повторный START*). Этот прием используется при изменении направления передачи к адресуемому устройству (например, «чтение- модификация- запись»), обращение к устройству с другим адресом в течение текущего кадра, для синхронизации нового пакета данных и так далее. Повторный START должен формироваться после сигнала ACK. При наличии нескольких ведущих устройств требуется реализация процедуры арбитража.

Процедура арбитража начинается, когда на линиях данных и синхронизации установлены высокие уровни (интерфейс свободен). Для арбитража после окончания кадра выделяется короткий межкадровый временной интервал, в течение которого ведущие устройства должны формировать стартовый сигнал.

Арбитраж происходит на шине SDA, в периоды, когда шина SCL находится в *высоком* состоянии. Все устройства одновременно побитно передают в линию данных свои адреса. *Большим приоритетом обладает низкий уровень сигнала, так как используется «монтажное И»*. Арбитраж может продолжаться на протяжении нескольких бит до тех пор, пока не определится устройство с максимальным приоритетом. В этой процедуре может быть задействовано не только поле адреса, но и данных.

В каждом такте синхронизации устройства передают значение очередного бита через выходной транзистор и его же читают. Если переданный бит и принятый совпадают, то арбитраж продолжается дальше. Если устройство выставило высокий уровень, а прочитало низкий, то оно переключается в режим ведомого и *только читает* информацию на линии данных.

Таблица 3. Процедура арбитража

Адрес устройства 1	Адрес устройства 1	Адрес устройства 1
1	1	1
0	0	0
0	0	1
1	1	ведомое
1	0	
ведомое	ведущее	

Арбитраж запрещен между START и STOP, повторным START и битом данных, битом данных и сигналом STOP, повторным START и сигналом STOP.

Преимуществами интерфейса I2C по сравнению с SPI являются: наличие мультипроцессорного режима, возможность контроля ошибок, аппаратная синхронизация устройств, меньшее количество линий, изменение числа устройств не требует дополнительных аппаратных затрат, стандартизованная структура кадра, используя только две линии и два резистора можно собрать систему требуемой конфигурации.

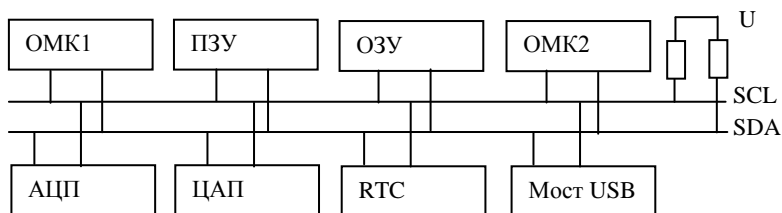


Рис.3.16. Структура микропроцессорной системы на основе I2C

К недостаткам можно отнести:

- при одной и той же частоте скорость передачи данных у SPI будет выше, так как кадр SPI не содержит поля адреса, девятого бита и реализуется дуплексный обмен;
- более сложная организация устройств и программного обеспечения.

При выборе микроконтроллера с интерфейсом I2C необходимо контролировать:

- автономный контроллер или совмещенный;
- возможность работы в режиме ведущего и ведомого;
- используемая адресация;
- объем аппаратных средств, реализующих протокол обмена;
- количество флагов и особенности информации, записанной в регистре состояния (коды, подтверждающие выполнение основных этапов обмена);
- максимальная скорость обмена;
- реализуемые способы ввода-вывода;
- особенности программной модели.

3.3. Подсистема прерываний

Обобщенная структура подсистемы прерываний *ПП* представлена на рисунке 3.17. Источником внутренних прерываний могут быть программа (исключение) или/и периферийные устройства *ОМК ВПр*. Внешние прерывания формирует оборудование, с которым взаимодействует *ОМК*.

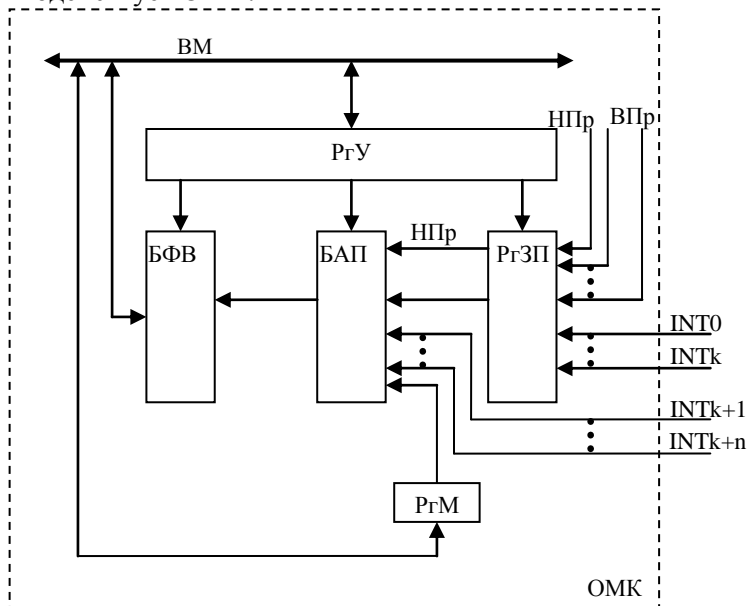


Рис. 3.17. Обобщенная структура блока обработки прерываний

Запросы $INT0 - INTk$ фиксируются в регистре запросов прерываний $P_2ЗП$. Часть запросов сбрасывается аппаратно в момент начала процедуры обработки прерываний, а некоторые - требуют программного сброса.

На входы блока анализа приоритетов $БАП$ поступает информация с выхода $P_гЗП$, внешние запросы $INTk+1 - INTk+n$, которые не фиксируются в $P_гЗП$, запросы немаскируемых прерываний НПР. Управление приоритетами выполняется с помощью специальных команд при инициализации ПП и регистра маски $P_2М$, который разрешает или запрещает прохождение определенных запросов. Некоторые $БАП$ содержат дополнительные средства управления уровнем приоритетов. Число уровней прерываний может быть от двух ($MCS-8051/52$) и более ($xMEGA$).

$БАП$ выбирает запрос с максимальным приоритетом и передает его в блок формирования вектора прерывания БФВ.

Вектор прерывания содержит информацию, необходимую для определения начального адреса подпрограммы обработки прерываний. В ОМК используются, в основном, **автовекторные** прерывания, при которых источнику или группе источников запросов соответствует **фиксированный** адрес начала подпрограммы обработки прерываний. Некоторые ОМК допускают управление адресами векторов.

Настройку подсистемы прерываний выполняется программно под управлением ЦП, который по внутренней магистрали ВМ пересылает информацию в регистры управления $P_2У$, маски $P_2М$ и другие.

В момент появления запроса, если прерывания разрешены, ЦП переходит к процедуре обработки: в стеке сохраняется информация об адресе следующей команды и текущее состояние прерванной программы (слово состояния процессора, регистры, используемые в подпрограмме обработки), в счетчик адреса команд записывается адрес подпрограммы обработки, выполняется обработка, после окончания которой восстанавливается информация о прерванной программе и осуществляется возврат из прерывания.

В большинстве ОМК в стеке **аппаратно** сохраняется и восстанавливается только адрес следующей команды, а работу с остальной

информацией необходимо выполнять *программно* в процедуре обработки.

Часть ОМК (например, HC05, HC08 фирмы Motorola) аппаратно в стеке сохраняют слово состояния процессора, аккумулятор, индексный регистр, счетчик адреса команд.

Основными характеристиками подсистемы прерываний являются:

- *количество входов внутренних и внешних запросов.* Число внешних запросов обычно не более двух. Если требуется большее количество, то следует использовать процедуру поллинга (рисунок 3.18).

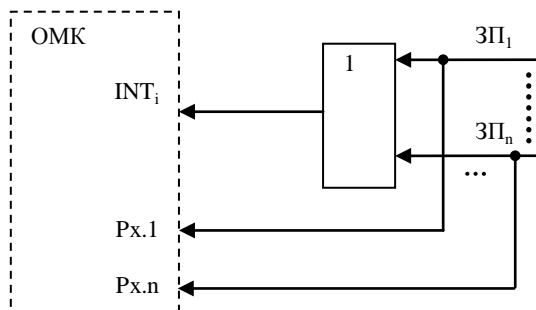


Рис. 3.18. Метод поллинга

Запросы $ZП_1-ZП_n$ объединяются по «ИЛИ» и подаются на вход внешнего запроса прерывания INT_i . Одновременно они поступают на свободные линии параллельного порта $P_x.i$. В подпрограмме обработки, соответствующего запроса INT_i , происходит анализ входных линий P_x и по его результатам выполняется требуемый фрагмент процедуры обработки прерываний. Приоритет опроса битов P_x устанавливается программно. При отсутствии свободных портов ОМК для хранения запросов может использоваться внешний регистр.

Применение поллинга увеличивает время обработки прерываний и требует применения дополнительных аппаратных средств.

Ряд семейств ОМК обладают расширенным количеством внешних прерываний: AVR MEGA, XMEGA, Motorola; [26,37]

- **соотношение между числом запросов и векторов прерываний.** Лучшими характеристиками обладают ОМК, в которых число запросов прерываний равно количеству векторов. В некоторых ОМК один вектор обслуживает несколько запросов. В подобном случае увеличивается время обработки прерываний, так как необходимо программно определять источник запроса методом поллинга;

- **способы управления приоритетом.** Для управления приоритетами запросов при одноуровневых прерываниях используется принцип маскирования, в многоуровневых - дополнительно регистры приоритетов;

- **параметры сигнала внешнего запроса.** Прерывание может вызвать фронт, срез или уровень сигнала запроса. Вид запроса выбирается при инициализации ПП в зависимости от условий работы и особенностей обработки прерываний ОМК.

Недостатком запросов по фронту или срезу является повышенное влияние импульсных помех, что может привести к ложному срабатыванию ПП.

Запросы по уровню обеспечивают большую надежность, но они могут не фиксироваться в РгЗП (AVR) или не сбрасываться аппаратно как в 8051.

Длительность сигнала определяется особенностями внутреннего цикла обработки запроса прерываний и индивидуальна для каждого ОМК;

- **особенности организации стека.** Стек является местом хранения параметров подпрограмм. В ОМК применяют аппаратный или программный стек.

Аппаратный стек имеет фиксированную глубину и не допускает программного обращения. При обращении к подпрограммам в него автоматически записывается обычно содержимое счетчика адреса команд. Поэтому сохранение контекста программы выполняется в свободных регистрах или ячейках памяти.

Если резидентная память достаточного объема, то **каждая подпрограмма** может использовать для своей работы **фиксированные ячейки памяти или регистры**. В этом случае сохранения контекста не требуется.

При программной организации стека местом его расположения могут быть внутренняя (резидентная) или внешняя память.

Параметрами стека являются: место расположения вершины стека (значение указателя стека), его глубина, которая определяется степенью вложенности подпрограмм и объемом записываемой в него информации, направление изменения указателя стека при записи информации (возрастающий или убывающий).

В структуру PIC18xxx, выполненного по гарвардскому принципу, введена дополнительная магистраль для работы со стеком [38].

При работе с подпрограммами необходимо контролировать объем информации, записываемый в стек аппаратно;

- **время реакции на запрос.** Это время от момента прихода запроса до входа в процедуру обработки прерываний. Время реакции на запрос определяется особенностями архитектуры ОМК и обычно задается количеством циклов ПБ.

Однако при оценке скорости ввода-вывода по прерыванию кроме времени реакции на запрос необходимо учитывать времена сохранения и восстановления контекста программы, возврата из прерывания, повторного вызова прерывания;

- **программная модель** подсистемы прерываний содержит регистры запросов прерываний, маски, управления видом запроса, приоритета. Особенностью программной модели подсистемы прерываний ОМК является распределенный характер перечисленных регистров. Например, в 8051 функции регистра запросов выполняют регистры TCON, SCON, регистра маски - IE, регистра приоритетов - IP, регистра управления - TCON.

В ОМК AVR количество регистров, обслуживающих подсистему прерываний, достигает 10. [24,37]

Для инициализации подсистемы прерываний необходимо: определить параметры стека, разработать подпрограмму обработки прерываний, при использовании внешних прерываний задать вид сигнала, установить биты общего разрешения прерываний и требуемых запросов.

3.4. Блок временных событий.

Функционирование любой автоматизированной системы невозможно без специальных средств, обеспечивающих анализ внутренних и внешних событий и формирование управляющих сигналов различного вида.

В ОМК эти функции реализует блок временных событий *БВС*, который должен решать следующие задачи:

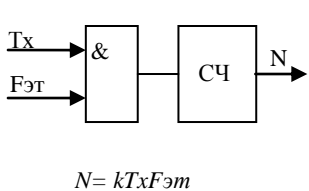
- Измерение временных интервалов, частоты, периода, скважности, фазового сдвига, числа импульсов и так далее;
- Фиксацию момента появления тех или иных событий (фронта, среза, заданного количества импульсов и так далее);
- Формирование интервалов времени для запуска внутренних и внешних событий;
- Формирование последовательности импульсов заданной частоты и скважности;
- Реализацию различных модуляторов: широтно-импульсных ШИМ, фазочастотных ФЧМ и так далее;
- Контроль времени работы программы;
- Выполнение функций часов реального времени.

Основными характеристиками БВС являются погрешность преобразования, разрешающая способность, быстродействие, количество каналов преобразования.

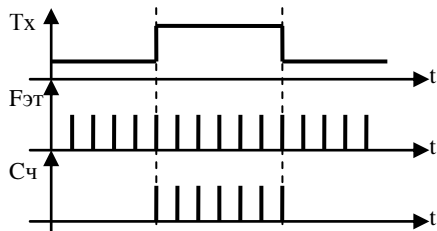
Для лучшего понимания работы БВС целесообразно напомнить принципы измерения и формирования частотно- временных сигналов.

Измерение - это процесс сравнения неизвестной величины с известной мерой (эталоном). Характеристиками процесса измерения являются диапазон изменения входной величины, основная и дополнительные погрешности (абсолютная, относительная, приведенная), разрешающая способность, время измерения.

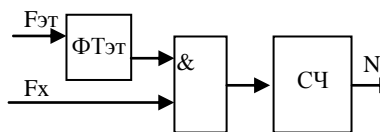
При измерении временных интервалов, частоты, периода, скважности, фазового сдвига в качестве источника эталонных сигналов обычно используют внутренний *тактовый кварцевый генератор*, на базе которого могут быть построены различные формирователи (рисунок 3.19)



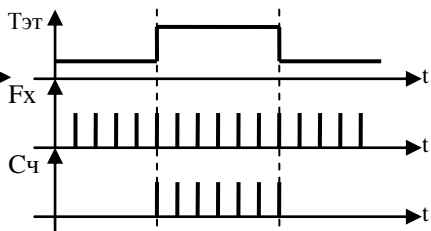
$$N = kT_x F_{эТ} m$$



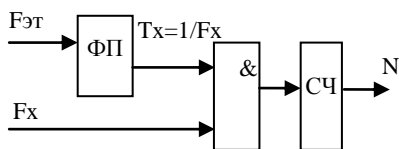
a)



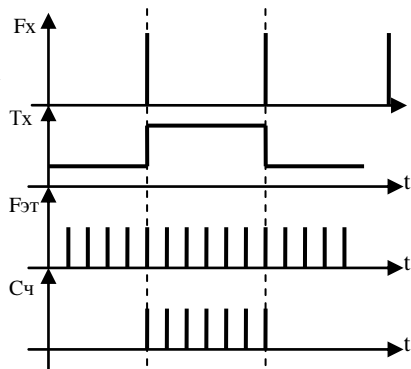
$$N = kT_{эТ} m \cdot F_x$$



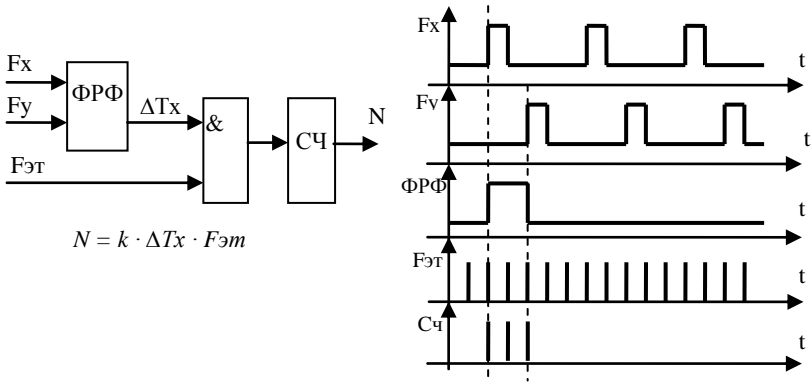
б)



$$N = kF_{эТ} m / F_x$$

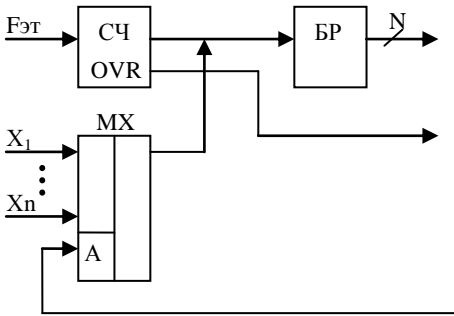


в)



$$N = k \cdot \Delta T_x \cdot F_{\text{эт}}$$

г)



$$T_{x_1} = \frac{k \cdot N_{x_1}}{f_{\text{эм}}}$$

$$T_{x_n} = \frac{k \cdot N_x}{f_{\text{эм}}}$$

д)

Рис. 3.19. Методы измерения временных событий: а - временной интервал на выходе формирователя разности фаз ФРФ преобразуется в код; б - метод среднего; в - измерении сигналов низкой частоты; г - измерение разности фаз двух импульсных последовательностей; д - определение времени появления событий.

Абсолютная величина погрешности, если источники измеряемого и эталонного сигналов несинхронизированы, равна ± 1 единица младшего разряда. Тогда требуемая разрядность преобразователя N определяется из соотношения $N \geq \log_2 1/\gamma$, где γ - основная приведенная погрешность.

Если N определено, то значения $F_{\text{эт}}$ и $T_{\text{эт}}$ рассчитываются из представленных ниже соотношений при заданном верхнем диапазоне измеряемой величины.

Измерение длительности импульса иллюстрирует рисунок 3.19а, где $СЧ$ - счетчик импульсов, T_x - измеряемый временной интервал, $F_{эт}$ - эталонная частота, $N=K * F_{эт} * T_x$.

Например, для измерения временного интервала в диапазоне (0-1) мс с погрешностью 0,1% $N \geq 1024$, а $F_{эт}=1024/1\text{мкс}$.

Измерение частоты в зависимости от требуемого быстродействия может быть выполнено на основании схем рисунок 3.19 б, в.

Наиболее универсальным является метод «среднего» (рисунок 3.19 б). Для его реализации необходим формирователь эталонного временного интервала $\Phi T_{эт}$, значение которого вычисляется на основании изложенного выше алгоритма: $N=K * F_{эт} * T_x$.

Если максимальное значение измеряемой частоты $F_x=1$ МГц, допустимая погрешность $<0,1\%$, $N \geq 1024$, то $T_{эт} \geq 1$ мс. Недостатком этого метода при измерении низкочастотных сигналов является низкое быстродействие, так как при максимальной частоте 100Гц и том же значении погрешности $T_{эт} \geq 10,24$ с.

Поэтому при измерении сигналов низкой частоты, если время ограничено, используется схема, изображенная на рисунке 3.19 в (метод «мгновенного значения»). В этом случае измеряемая частота F_x поступает на вход формирователя периода ФП и далее измеряется как временной интервал. Время измерения равно длительности периода, а значение $F_{эт}$ определяется аналогично схеме рисунка 3.9, а: $N=K * F_{эт}/F_x$.

Недостатком этого метода является пониженная помехозащищенность, так как измеряется значение одного периода, и необходимость выполнения процедуры $1/N$ для получения значения частоты. Первый недостаток может быть устранен за счёт суммирования измерений нескольких периодов и последующего деления. Обычно количество периодов выбирают кратное степени 2, чтобы использовать команду сдвига вместо деления. Вычисление обратной функции для ОМК не представляет особых трудностей, хотя и требует дополнительного времени.

Схема измерения разности фаз двух импульсных последовательностей F_x и F_y представлена на рисунке 3.19 г. Временной интервал на выходе формирователя разности фаз ФРФ преобразуется в код аналогично схеме рисунок 3.19, а: $N=K * F_{эт} * T_x$

В качестве ФРФ может использоваться, например, RS-триггер.

Время появления событий можно определить с помощью схемы (рисунок 3.19 д). На вход счетчика СЧ поступает $F_{\text{эт}}$. В момент прихода фронта или среза импульса X_1 текущее значение счетчика «на лету» переписывается в буферный регистр *БР* и используется для дальнейшей обработки. При расчете параметров этой схемы необходимо контролировать возможность переполнения счетчика *OVR*. При большом числе событий используется мультиплексор *МХ*.

Обобщенная структура формирователя выходных сигналов представлена на рисунке 3.20.

Эталонные импульсы с выхода генератор *ГЭИ* поступают на вход счетчика *СЧ*. В зависимости от требуемой функции к выходу счетчика подключаются различные формирователи. Управляемые линии связи на рисунке обозначены крестом.

Для деления частоты требуется регистр *РзКД*, в который записывается коэффициент деления счетчика. В момент переполнения счетчика происходит повторная запись коэффициента деления.

Формирование импульса заданной длительности выполняется с помощью схемы сравнения *СС* и, например, *RS*-триггера. Значение длительности определяется частотой *ГЭИ* и значениям, записанными в регистр *РзДИ* и *СЧ*. В момент запуска формирователя триггер *Тз* устанавливается в ноль и запускается счетчик, предварительно установленный в исходное значение. В момент равенства содержимого *СЧ* и *РзДИ* на выходе *СС* формируется сигнал, изменяющий состояние *Тг* на противоположное. Получение сигнала требуемой полярности выполняет программно управляемый выходной буфер *ПУВБ*.

Для организации генератора программируемой частоты и скважности необходимы все ресурсы, представленные на рисунке 3.20. Скважность - это отношение длительности импульса к периоду импульсной последовательности. Частота выходного сигнала определяется значением, хранящемся в *РгКД*, а длительность - содержимым *РгДИ*. Временная диаграмма приведена на рисунке 3.20.

Широтно-импульсная модулятор *ШИМ* является устройством, в котором ширина импульса пропорциональна значению модулирующего сигнала. В качестве модулирующего сигнала могут быть, например, выходные коды *АЦП*, частотно-временного преобразователя, изменения которых пропорциональны измеряемому параметру.

Широкое распространение ШИМ получили в системах автоматического управления.

ШИМ можно рассматривать как частный случай генератора программируемой частоты и скважности, у которого частота постоянна, а длительность импульса зависит от величины преобразуемого сигнала. Код модулирующего сигнала должен быть записан в РгДИ.

Реализация ШИМ возможна на основе суммирующего (вычитающего) или реверсивного счетчиков.

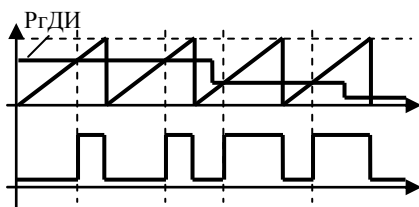
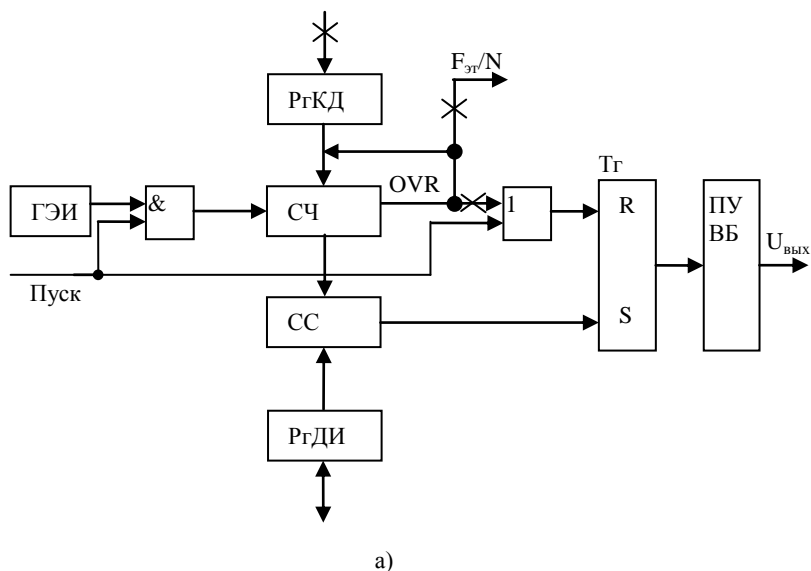


Рис. 3.20. Формирователь: а - обобщенная структура; б - временная диаграмма

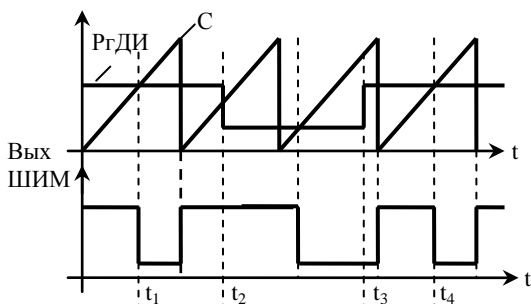


Рис. 3.21. Временная диаграмма широтно-импульсного модулятора

Для исключения пропуска в формировании импульса изменение состояния РГДИ следует проводить во время неинформативной части периода (Рисунок 3.21).

На основе ШИМ можно построить цифроаналоговый преобразователь ЦАП средней точности и относительно низкого быстродействия (рисунок 3.22).

На выходе ШИМ устанавливается ключевой элемент, выполненный на биполярном или полевом транзисторе T , который коммутирует источник эталонного напряжения (или тока) на фильтр низкой частоты ФНЧ. Фильтр формирует напряжение пропорциональное длительности импульса (рисунок 3.22 б). Согласование с приемником сигнала выполняет повторитель напряжения ПН, имеющий высокое входное и низкое выходное сопротивления. На рисунке в качестве примера приведен простейший однозвенный ФНЧ, состоящий из $R1$, C .

Погрешность ЦАП в данной схеме определяется метрологическими характеристиками источника эталонного сигнала, ключевого элемента, резисторов и конденсатора, а быстродействие - частотой ШИМ. Разрешающая способность зависит от разрядности счетчика, на котором выполнен ШИМ, и минимально возможного изменения скважности.

При пониженных требованиях к точности ЦАП ФНЧ можно подключать непосредственно к выходу ШИМ.

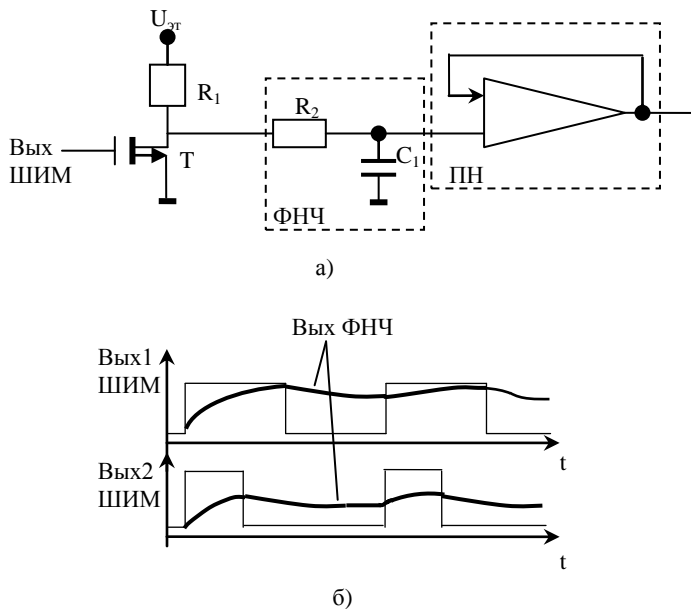


Рис. 3.22 ЦАП на базе ШИМ: а - обобщенная структура; б - временная диаграмма

Таким образом, для реализации БВС необходимы генератор эталонных сигналов, устройство, подсчитывающее количество импульсов от внешнего или внутреннего источника сигналов (счетчик-таймер), средства реализации сравнения эталонных и измеряемых сигналов, средства формирования выходных сигналов, коммутатор (мультиплексор) каналов.

Источником эталонных сигналов служит тактовый генератора ОМК, частота которого при повышенных требованиях к метрологии должна быть стабилизирована внешним кварцевым резонатором. Однако возможно применение и внешних генераторов.

Базовым элементом БВС является счетчик.

Основные характеристики счётчиков:

- разрядность;
- особенности формирования и доступа к сигналу переполнения. В большинстве ОМК флаг переполнения доступен только про-

граммно или по прерыванию. При вводе-выводе по прерыванию флаг переполнения обычно сбрасывается аппаратно. Ограниченное число ОМК выводят сигнал переполнения на внешние выходы, что обеспечивает возможность каскадного соединения счетчиков;

- быстродействие (максимальная частота);
- направление счёта (вычитающий, суммирующий, реверсивный);
- возможность выбора источника входного сигнала. В режиме счетчика на вход поступают импульсы от внешнего источника сигналов, таймера - от внутреннего генератора.
- режим работы счетчика (при синхронном режиме изменения состояния счётчика согласованы с частотой внутреннего генератора, в асинхронном - эта связь отсутствует);
- возможность программного управления входной частотой СТ, работающего в режиме *таймера*;
- способ запуска - программный или аппаратный (аппаратный запуск используется для синхронизации работы ОМК с внешними устройствами, в этом случае начало работы счетчика задается внешним сигналом);
- возможность программного управления режимами работы: счетчик или таймер, способ запуска, разрядность, аппаратная перезагрузка коэффициента деления СТ после переполнения и так далее;
- количество счётчиков.

В состав ОМК входят (8-16)-разрядные счетчики-таймеры, количество которых может быть от 1 до 4. Чем больше счетчиков-таймеров, тем больше параллельных процессов преобразования можно реализовать.

В зависимости от требуемых технических характеристик остальные элементы БВС могут быть реализованы программно, программно-аппаратно с использованием одного или нескольких счетчиков - таймеров ОМК или с помощью программируемого массива (матрицы счетчиков).

При программной реализации управление функциями БВС выполняет ЦП. Недостатком такого способа является низкое быстродействие и активное участие ЦП в процессе преобразования, что увеличивает время обработки информации.

К недостаткам подобных БВС относятся программное управление основными процедурами анализа и формирования временных событий, что приводит к увеличению погрешностей и потере быстродействия, и ограниченное число каналов.

В ОМК, ориентированных на задачи управления технологическим оборудованием в режиме реального времени, используется матрица счетчиков-таймеров МСТ. Упрощенная схема одного канала показана на рисунке 3.23.

В качестве базовых элементов используются счётчики-таймеры СТ1 и СТ2, которые могут соединяться каскадно, образуя СТ удвоенной разрядности. Источниками сигналов могут быть внутренний или внешний генератор, сигнал переполнения предыдущего СТ OVR. Коммутация сигналов выполняется программно- управляемыми мультиплексорами МХ1 и МХ2. В некоторых ОМК на входе СТ установлены программно управляемые делители частоты ПДЧ1, ПДЧ2. Управление режимами работы осуществляется с помощью регистров управления соответствующих счётчиков РГУСТ1 и РГУСТ2.

В *режиме захвата РЗ* входные воздействия поступают на вход детектора событий ДС, который в момент прихода запрограммированного события (фронт, срез, любое изменение сигнала) устанавливает флаги в регистре состояния РГС и фиксирует в буферном регистре БР текущее значение используемого СТ. Выбор используемого СТ выполняется с помощью мультиплексора каналов МХ, управляемого РГУК. Содержимое буферного регистра переписывается в регистр счетчика-таймера РГСТ. При этом устанавливается флаг в регистре состояния. Если код из РГСТ вовремя не считан, то следующее значение СТ будет записано в БР. При занятости БР возможна потеря кода. Эта ситуация фиксируется в регистре состояния как переполнение БР.

При обнаружении требуемого события схема управления СУ формирует внутренние сигналы для запуска АЦП, сброса и так далее.

В *режиме сравнения РС* (формирования внешних сигналов) информация с выхода СТ подается на один из входов схемы сравнения СС, а второй вход соединен с РГСТ. При равенстве кодов устанавливается флаг в РГС и с помощью схемы формирователя СФ формиру-

ется внешнее выходное событие: уровень лог.0, 1 или инверсия предыдущего состояния.

Ввод-вывод информации в МСТ может быть выполнен программно или по прерыванию.

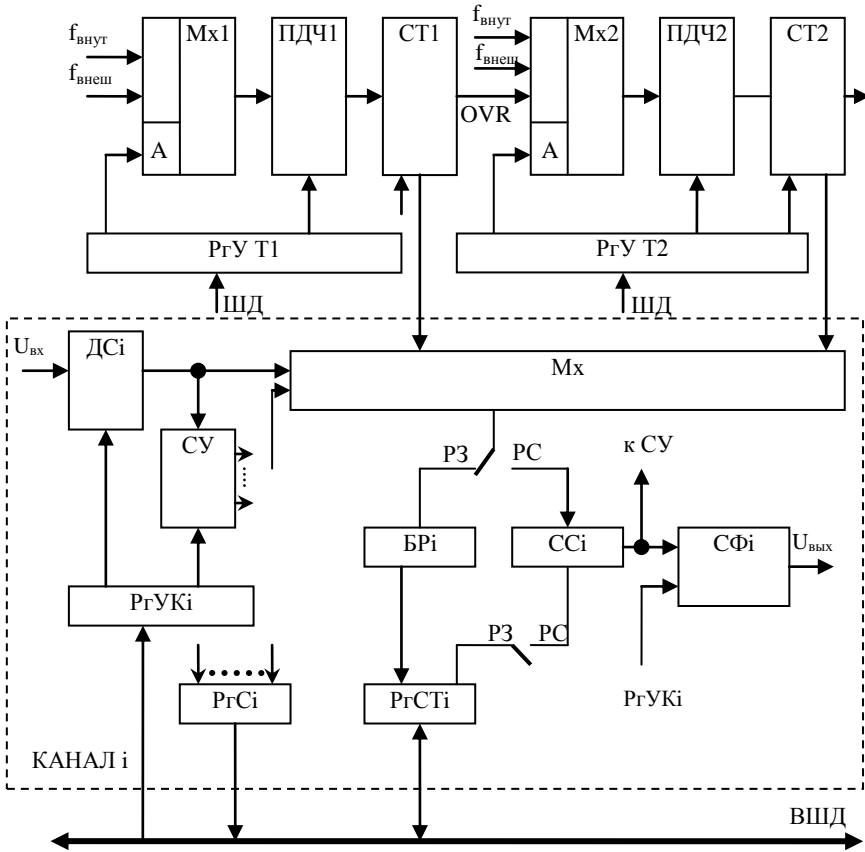


Рис. 3.23. - Обобщенная структура матрицы счетчиков-таймеров

Настройка на требуемый режим осуществляется с помощью регистра управления каналом PrУК.

В 8-разрядных ОМК используется обычно один 16-разрядный СТ, а количество каналов - от 2 до 5.

В матрице счетчиков - таймеров основная часть операций по анализу и формированию сигналов выполняется аппаратно. ЦП считывает информацию из PrCT, обрабатывает её в соответствии с требуемыми алгоритмами, реагирует на появление флагов в PrC.

Пример использования МСТ для измерения длительности импульса приведен на рисунке 3.24 а.

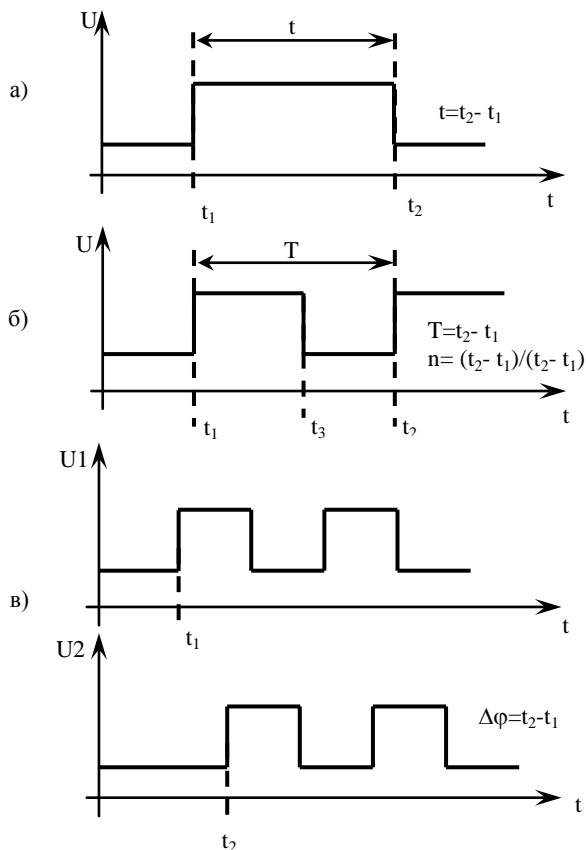


Рис. 3.24. Измерение частотно-временных сигналов МСТ

Детектор события настраивается на режим обнаружения фронта входного сигнала. При обнаружении данного события формируется, например, запрос прерывания, при обработке которого ЦП считыва-

ет время t_1 и перенастраивает ДС на восприятие среза. В момент второго прерывания считывается t_2 , вычисляется $t = t_2 - t_1$ и канал настраивается на восприятие следующего импульса.

Аналогично измеряется период и скважность (рисунок 3.14 б).

Особенностью измерения фазового сдвига высокочастотных колебаний является использование одновременной работы двух СТ (рисунок 3.15 в).

Для этого необходимо в РгСТ записать код, соответствующий периоду модулятора T , по достижении времени T установить на выходе «1», и перезагрузить РгСТ значением кода τ . По достижению времени τ изменить значение выходного сигнала на противоположное и загрузить в регистр код, соответствующий $(T - \tau)$. Далее этот цикл повторяется.

Из описания алгоритмов следует, что при проектировании устройств на основе МСТ необходимо учитывать временные задержки, возникающие при чтении, управлении и перезагрузке регистров.

Поэтому многие ОМК имеют аппаратные режимы формирования ШИМ. В этом случае необходимо только в регистре управления задать требуемый режим работы.

Формирование выходных сигналов рассмотрим на примере ШИМ (рисунок 3.25).

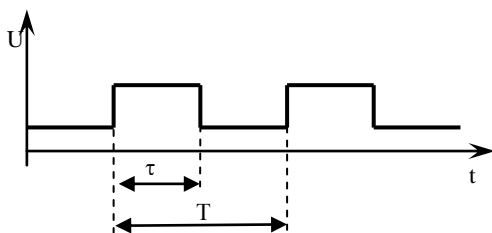


Рис. 3.25. Широтно- импульсная модуляция в МСТ

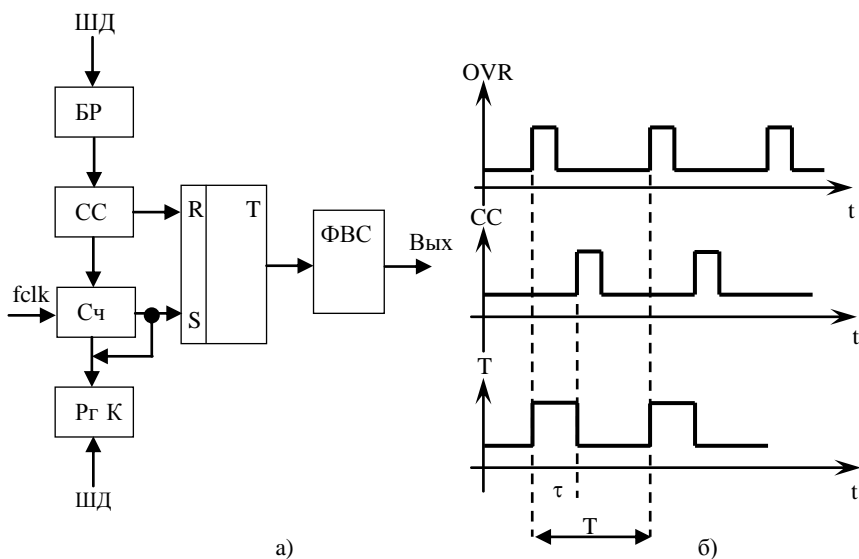


Рис. 3.26. Аппаратная реализация ШИМ: а - обобщенная структура; б - временная диаграмма

Пример аппаратной реализации ШИМ приведен на рисунке 3.26. На стадии инициализации в буферный регистр БР записывается код, определяющий длительность импульса, а в регистр РгК и счетчик Сч - код, соответствующий частоте ШИМ. Формирование ШИМ выполняет триггер Т. Сигнал переполнения Сч OVR устанавливает Т в единицу, а в момент равенства содержимого БР и СЧ выходной сигнал схемы сравнения СС сбрасывает триггер в ноль. При переполнении Сч содержимое РгК перезагружается и цикл повторяется.

В 16-разрядных ОМК фирмы Motorola используется сопроцессор временных событий СВС, который является интеллектуальной подсистемой, предназначенной для формирования и анализа высокочастотных сигналов в реальном масштабе времени без участия ЦП. В отличие от МСТ СВС имеет собственный специализированный микропроцессор СМП, который обрабатывает информацию в соответствии с назначением каналов [26]

Структура СВС приведена на рисунке 3.27.

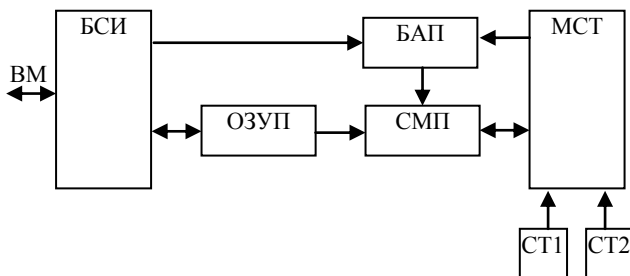


Рис. 3.27. Сопроцессор временных событий

Центральный процессор ОМК осуществляет взаимодействие с СВС путем записи-чтения информации в блок связи с интерфейсом БСИ, содержащий основные регистры управления и состояния. В ОЗУ параметров *ОЗУП* за каждым каналом закрепляется определенное количество ячеек памяти, в которых храниться информация, определяющая режим работы и текущее состояние канала. Обращение к ОЗУП возможно, как от ЦП, так и от соответствующего канала. *СМП* обеспечивает управление каналами в соответствии с их приоритетами, которые задаются на стадии инициализации в блок анализа приоритетов *БАП*, и программ, хранящихся в его ПЗУ.

ПЗУ обеспечивает выполнение определенного набора операций, который включает формирование сигналов с заданной задержкой и длительностью, генерацию импульсов заданной частоты и скважности, измерение различных частотно-временных параметров, управление шаговыми двигателями и так далее. Возможна разработка специализированных программ пользователя, которые могут храниться в ОЗУ микроконтроллера. Для этих целей разработаны специализированные инструментальные средства.

Структура матрицы счетчиков МСТ и взаимодействие с счетчиками таймерами *СТ1*, *СТ2* аналогичны рассмотренному ранее.

Сторожевой таймер СтТ (Watchdog) предназначен для контроля состояния процесса обработки и управления, выполняемого ОМК. Он предотвращает «зависание» ОМК при возникновении нештатных ситуаций.

На стадии инициализации СтТ задается максимально допустимое время работы программы или её фрагмента. Если установленный временной интервал закончился, генерируется системный сброс. При нормальном ходе процесса обработки СтТ должен программно сбрасываться через время меньшее установленного временного интервала.

Особенно важно применение сторожевого таймера в ОМК, работающих в условиях электромагнитных и электростатических помех.

Структура сторожевого таймера представлен на рисунке 3.28.

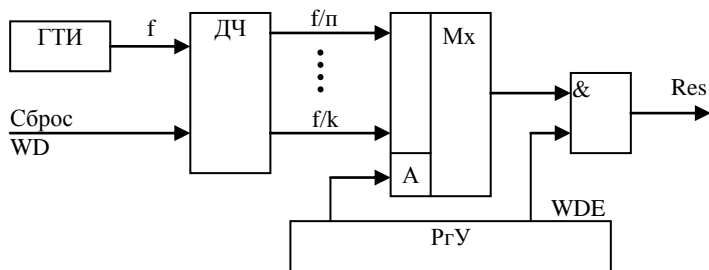


Рисунок. 3.28. Структура сторожевого таймера WD

Основным элементом сторожевого таймера является делитель частоты ДЧ, выполненный на основе счетчика. Выходы делителя частоты с помощью мультиплексора МХ коммутируются на вход схемы сброса, если работа СтТ разрешена сигналом WDE.

При включении контроллера счётчик сбрасывается. В процессе работы программы на его вход непрерывно поступают сигналы тактового генератора ГТИ. При прохождении наиболее критических точек программы производится принудительный сброс счетчика и его перезапуск. Если сигнал от контрольной точки вовремя не пришёл, счетчик переполняется. В момент переполнения формируется сигнал сброса ОМК, вызывающий повторный старт программы.

Допустимое время ожидания событий задается в регистре управления PrY, который разрешает работу СтТ и определяет временной интервал, коммутируя соответствующий вход МХ. Обычно коэффициенты деления частоты кратны степени 2.

Для реализации СтТ необходим счетчик, средства установки и контроля допустимого времени.

В зависимости от требуемых характеристик он может быть выполнен программно, с применением счетчиков- таймеров или аппаратно.

При оценке возможностей СтТ следует контролировать: максимально допустимое время переполнения таймера, количество диапазонов установки времени, особенности источника тактовых импульсов, возможность автономного питания, наличие индивидуального вектора прерывания.

Лучшими характеристиками обладают аппаратные сторожевые таймеры, имеющие автономные источник питания и генератор тактовых импульсов, собственный вектор прерывания. Подобные СтТ могут функционировать даже в «спящем режиме» при отключенном системном генераторе.

3.5. Средства ввода/вывода аналоговой информации

К средствам ввода-вывода аналоговой информации относятся компараторы, аналого-цифровые и цифроаналоговые преобразователи, датчики температур, источники опорного напряжения, источники питания датчиков и так далее.

3.5.1. Аналоговый компаратор

Аналоговый компаратор АК предназначен для сравнения двух аналоговых сигналов. Наиболее часто один из них является опорным (эталонным) сигналом. В момент сравнения на выходе АК формируется перепад напряжения. Поэтому его можно считать однобитовым аналого-цифровым преобразователем. По сравнению с аналого-цифровым преобразованием процедура сравнения в этом случае реализуется значительно быстрее.

АК может использоваться также в качестве формирователя импульсов с крутыми фронтами и срезом (например, преобразование синусоидального сигнала в прямоугольный).

Основными характеристиками АК являются:

- порог срабатывания - минимальная величина разности, при которой формируется выходной сигнал. Обычно единицы - десятки мВ;

- время отклика (переключения) - промежуток времени от начала сравнения до момента формирования выходного сигнала $U_{вых}$. Оно зависит от величины превышения $U_{вх}$ над опорным напряжением. В технических характеристиках это время указывается для фиксированной величины превышения и находится в диапазоне единиц нс - единиц мкс;

- диапазон входных напряжений. Определяется величиной $U_{пит}$;

- коэффициент подавления синфазного сигнала (1000 и более);
- величина смещения;
- входные параметры: ток, емкость, величина смещения;
- нагрузочная способность;
- потребляемая мощность.

При анализе характеристик АК в ОМК дополнительно необходимо контролировать:

- количество компараторов. При наличии двух компараторов представляется возможность аппаратного допускового контроля (порогового детектирования) двух величин $N_{min} \leq N_i \leq N_{max}$;

- количество источников сигналов, присоединенных к инвертирующему и неинвертирующему входам АК. В качестве внутренних входных сигналов могут использоваться выходы источника опорного напряжения, ЦАП, мультиплексора АЦП, источника питания или внутреннего делителя напряжения;

- особенности использования выходного сигнала компаратора. Он может служить для генерации прерываний при изменениях выходного сигнала из низкого состояния к высокому или из высокого состояния к низкому, запуска различных внутренних событий (например, захвата таймера, запуск АЦП). Выход компаратора может быть подключен к внешним выводам ОМК;

- возможность управления петлей гистерезиса. Программно можно выбрать работу компаратора с малым гистерезисом, с большим гистерезисом или без гистерезиса. Добавление гистерезиса позволяет избежать режима автоколебаний при незначительном отличии входных напряжений или наличии помех во входных сигналах;

- возможность управления энергопотреблением. В высокоскоростном режиме снижается время переключения, но увеличивается

энергопотребление. При выборе экономичного режима увеличивается время переключения, но снижается ток потребления.

Некоторые ОМК позволяют программно управлять временем переключения.

Обобщенная структура АК ОМК представлена на рисунке 3.29, где *АМ* - входной аналоговый мультиплексор, *ДМ* - выходной демultipлексор. Управление этими устройствами выполняется специальными регистрами.

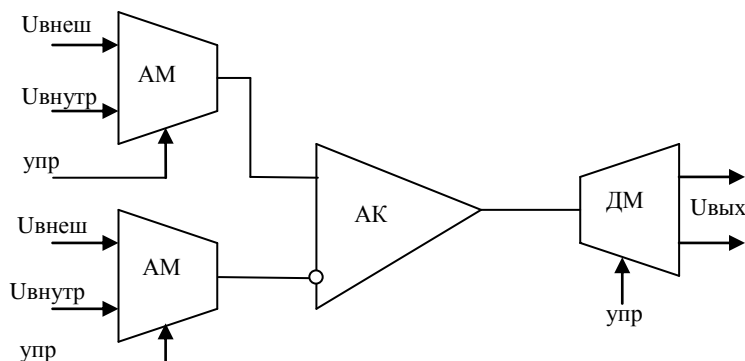


Рис. 3.29. Структура аналогового компаратора ОМК

Программная модель АК может содержать от одного до четырех регистров управления и состояния. [12,24, 26]

3.5.2. Особенности организации блока ЦАП ОМК

Цифро-аналоговый преобразователь ЦАП предназначен для преобразования цифрового кода в пропорциональный аналоговый сигнал. Область применения ЦАП довольно обширна: управление аналоговыми элементами автоматизированных систем, аудиовидеотехника, измерительные системы, цифровые генераторы сигналов различной формы, АЦП и так далее.

Преобразование заключается в суммировании токов или напряжений, пропорциональных весам двоичных разрядов, причем суммируются только токи тех разрядов, значения которых равны лог. 1. В двоичном коде вес от разряда к разряду изменяется вдвое

В зависимости от требуемой точности и быстродействия схемотехника ЦАП отличается большим разнообразием: по виду выход-

ного сигнала (ток, напряжение), принципом построения матрицы (суммирование токов или напряжения, резистивные матрицы, на переключаемых источниках тока конденсаторах, ШИМ и так далее) [40]. В ОМК наибольшее распространение получили ЦАП на основе резистивной матрицы «R-2R». Погрешности ЦАП определяется, в основном, метрологическими характеристиками источника опорного (эталонного) напряжения, матрицы и выходного усилителя. Основные характеристики ЦАП практически совпадают с характеристиками АЦП, изложенными в разделе 3.5.3 [40,41].

Обобщенная структура ЦАП приведена на рисунке 3.30.

Источники опорного напряжения *ИОН* подключаются к матрице ЦАП МЦ с помощью аналогового мультиплексора питания МП.

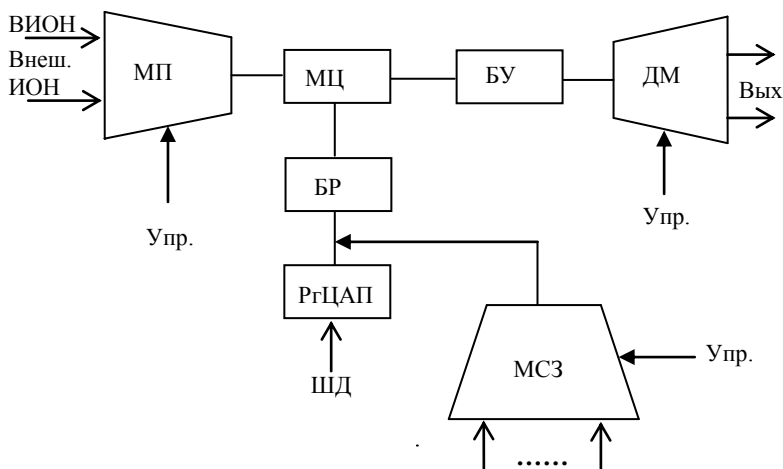


Рис. 3.30. Структура цифро-аналогового преобразователя ОМК

В качестве ИОН могут использоваться внутренний ИОН ВИОН или внешний. Текущее значение кода записывается в регистр *RгЦАП* с шины данных ШД и передается в буферный регистр *БР* по сигналам запуска, которые коммутируются с помощью мультиплексора *МСЗ*. Источниками запуска могут быть различные блоки ОМК или внешние события. Коммутация *МСЗ* выполняется специальными регистрами управления. Сигнал с выхода МЦ поступает на вход

буферного усилителя БУ, выходной сигнал которого с помощью аналогового демультиплексора ДМ может передаваться на внешние выводы или внутренние блоки ОМК.

В блоке ЦАП, кроме общих характеристик, необходимо контролировать:

- возможность коммутации ИОН и внешних источников в качестве напряжения питания ЦД. На основе ЦАП можно организовать аналоговый умножитель, одним из сомножителей которого является напряжение питания ЦД, а другим - код, пропорциональный значению другого сомножителя;
- программное управление значением выходного напряжения.
- программное управление разрядностью, быстродействием и временем нарастания выходного напряжения;
- тип выходного сигнала - ток или напряжение. Точковый выход при наличии двух ЦАП позволяет их объединять, увеличивая значение выходного тока в два раза;
- способ запуска. Источником сигнала запуска могут быть: программа, запрос прерывания или ПДП, внешний сигнал, переполнение счетчика-таймера и так далее;
- приемники информации выходного напряжения: внешний вывод ОМК, вход аналогового компаратора или АЦП;
- возможность калибровки напряжения смещения и коэффициента передачи;
- особенности работы в режиме пониженного энергопотребления.

Программная модель ЦАП состоит из регистров управления, состояния, данных, количество которых может изменяться от 3 до 8 [12, 24, 26].

3.5.3. Аналого-цифровой преобразователь

Большинство сигналов реальных объектов автоматизации относятся к классу аналоговых, которые представляют собой непрерывные функции времени (рисунок 3.31).

При вводе аналоговой информации входные сигналы преобразуются в дискретную форму с помощью аналого-цифровых преобразователей АЦП, обрабатываются ЦП и в некоторых случаях может

потребуется обратное преобразование, которое реализуется цифро-аналоговым преобразователем ЦАП.

Средства ввода-вывода аналоговой информации во многом определяют метрологические характеристики автоматизированной системы, в которой используется ОМК.

Процесс преобразования непрерывного сигнала $X(T)$ в дискретную форму $X(T_i)$ называется **дискретизацией или квантованием по времени**. Дискретизация представляет собой процесс получения мгновенных значений непрерывной функции в моменты времени T_i (рисунок 3.21).

Степень приближения мгновенного значения к действительному значению непрерывной функции в i -той точке ΔX определяется числом уровней квантования. Подобное преобразование называется **квантованием по уровню**.

Исходную функцию можно восстановить по её дискретным значениям с определенной погрешностью.

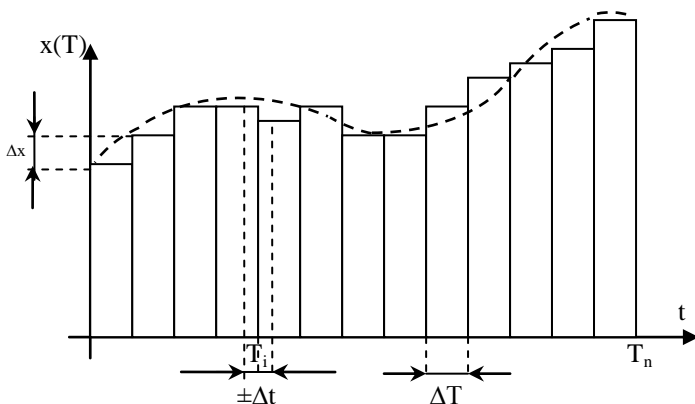


Рис. 3.31. Преобразование аналогового сигнала в дискретную форму

Чем больше число уровней квантования и меньше шаг дискретизации, тем выше точность воспроизведения исходной функции. Однако избыточные отсчеты требуют дополнительные программно-аппаратные ресурсы и приводят к увеличению времени обработки информации. Оптимальными будут такие шаг дискретизации и число уровней квантования, при которых входную функцию можно

восстановить с заданной погрешностью при минимальном числе выборок. Выбор частоты дискретизации изложен в соответствующей литературе и в данной работе не рассматривается.

В инженерной практике используется равномерная дискретизация, интервал которой определяется на основании теоремы Котельникова - Найквиста. Если исходная функция имеет ограниченный спектр частот, то она может быть восстановлена по значениям выборок с удвоенной максимальной частотой в спектре сигнала: $\Delta t \geq 1/(2f_B)$.

Реальные сигналы, как правило, имеют бесконечный спектр. Поэтому интервал дискретизации, определенный на основании теоремы Котельникова, рассматривают как приближенное значение для функций с неограниченным спектром. На практике интервал дискретизации уменьшается по сравнению с расчетной в (2-6) раз, а максимальную частоту определяют эмпирически (экспериментально).

Количество разрядов при квантовании по уровню связана с допустимой величиной приведенной относительной погрешности γ соотношением $n \geq \log_2 1/\gamma$.

В АЦП ОМК обычно **погрешность квантования** составляет величину \pm единица младшего разряда. При определенной технической реализации погрешность квантования АЦП может быть уменьшена до 0,5 единицы младшего разряда. Таким образом, при $\gamma \geq 0,1\%$, разрядность АЦП $n \geq 10$.

При преобразовании выходного кода АЦП в аналоговый сигнал необходимо учитывать погрешность **датирования** Δt , которая связана с вариацией момента измерения T_i . Нестабильность этого параметра приводит к погрешности при восстановлении сигнала из цифровой формы в аналоговую. [39,41]

К числу основных характеристик АЦП относятся [41]:

Разрядность измеряется в битах и определяет максимальное число кодовых комбинаций на выходе АЦП

Разрешающая способность - величина, обратная максимальному числу кодовых комбинаций на выходе АЦП. Разрешающая способность выражается в процентах, разрядах или децибелах и характеризует потенциальные возможности АЦП с точки зрения достижимой точности. Например, 12-разрядный АЦП имеет разрешаю-

щую способность $1/4096$, или $0,0245\%$ от полной шкалы, или $-72,2$ дБ.

Не следует считать, что разрядность и разрешающая способность определяют основные метрологические характеристики АЦП. Они характеризуют только *потенциальные возможности* АЦП и *погрешность квантования*, а значение *общей погрешности* определяется совокупностью параметров, которые приведены далее.

Статические параметры определяют отклонение характеристики преобразования АЦП $N=f(U_x)$ от идеальной прямой, где N - значение кода на выходе АЦП, а U_x - входное напряжение.

Погрешность полной шкалы - относительная разность между реальным и идеальным значениями выходного кода в максимальной точке характеристики преобразования при отсутствии смещения нуля. Эта погрешность является мультипликативной составляющей полной погрешности. Указывается в % от полной шкалы или в единицах младшего разряда ЕМР.

Погрешность смещения нуля - значение выходного кода при $U_x=0$. Является аддитивной составляющей полной погрешности. Указывается в % от полной шкалы или в единицах младшего разряда ЕМР.

Погрешности полной шкалы и смещения нуля АЦП могут быть уменьшены либо подстройкой аналоговой части схемы, либо коррекцией вычислительного алгоритма цифровой части устройства.

Интегральная нелинейность - максимальное отклонение реальной характеристики преобразования $N=f(U_x)$ от оптимальной. Оптимальная характеристика находится эмпирически так, чтобы минимизировать значение погрешности нелинейности. Наиболее часто для этих целей используют метод наименьших квадратов. Нелинейность обычно определяется в % от полной шкалы или ЕМР.

Дифференциальная нелинейность характеризует неравномерность значения кванта на выходе АЦП при одном и том же приращении входного сигнала. У идеальной характеристики преобразования разность напряжений между каждым изменением кода на выходе должна соответствовать ЕМР. Отклонение этой разности от ЕМР определяется как дифференциальная нелинейность. Дифференциальная нелинейность определяет два важных свойства АЦП: непропадание кодов и монотонность характеристики преобразования.

Непропадание кодов - свойство АЦП выдавать все возможные выходные коды при изменении входного напряжения от начальной до конечной точки диапазона преобразования.

Монотонность характеристики преобразования - это неизменность знака приращения выходного кода D при монотонном изменении входного преобразуемого сигнала. Монотонность не гарантирует малых значений дифференциальной нелинейности и непропадания кодов.

В идеале, повторяющиеся преобразования фиксированного постоянного входного сигнала должны давать один и тот же выходной код. Однако, вследствие неизбежного шума в схемах АЦП, существует некоторый диапазон выходных кодов для заданного входного напряжения. Если подать на вход АЦП постоянный сигнал и записать большое число преобразований, то в результате получится некоторое распределение кодов

Отношение сигнал/шум (SNR) - это среднеквадратическое значение входного сигнала к среднеквадратическому значению шума, который определяется, как сумма всех остальных спектральных компонентов, включая гармоники, но исключая постоянную составляющую. Для идеального АЦП $SNR = (6,02n + 1,76)$ дБ, где n - количество двоичных разрядов. Спектр шума квантования имеет равномерное распределение по частоте. Поэтому величина этого шума не может быть уменьшена путем увеличения времени преобразования и последующего усреднения. Шум квантования может быть снижен только за счет применения АЦП большей разрядности.

Диапазон входных сигналов указывает минимальное и максимальное значения входного сигнала. Минимальное значение входного сигнала определяется разрешающей способностью, а максимальное - величиной опорного напряжения.

Динамические параметры. Возникновение динамических погрешностей связано с дискретизацией сигналов, изменяющихся во времени. Можно выделить следующие параметры АЦП, определяющие его динамическую точность:

Максимальная частота преобразования - это наибольшая частота дискретизации входного сигнала, при которой **требуемый** параметр АЦП не выходит за заданные пределы. При этом предполагают, что входной сигнал не изменяется за время преобразования.

Выбранным параметром может быть, например, монотонность характеристики преобразования или погрешность линейности. Частота дискретизации задается в Гц, КГц, МГц или KSPS, MSPS (соответственно 1000 или 1000000 преобразований в секунду).

Время преобразования - интервал времени от скачка напряжения на входе, значение которого обычно равно максимальному, до установившегося значения кода, отличающегося от номинального на величину статической погрешности. Измеряется в секундах и её производных.

При выборе АЦП и оценке погрешностей измерения для конкретного применения необходимо учитывать не только его метрологические характеристики, но и целый ряд других параметров, связанных с особенностями источников входных сигналов, условий эксплуатации и так далее. К их числу относятся:

- **вид входного сигнала** (дифференциальный или потенциальный) и **полярность** (униполярный или биполярный). Преимуществом дифференциальных сигналов является большая помехозащищенность и возможность работы с изолированными от земли источниками. Для АЦП с дифференциальным входом необходимо учитывать **коэффициент подавления синфазного сигнала**, который определяет степень ослабления помехи, действующей на оба входа дифференциального усилителя. Коэффициент подавления синфазного сигнала указывается в децибелах дВ.

При измерении биполярных сигналов с помощью однополярного АЦП необходимо заботиться о смещении, что приводит к дополнительным погрешностям;

- **входные сопротивление и емкость**. Эти параметры обеспечивают согласование АЦП с источником сигналов и влияют на скорость преобразования;

- **характеристики источника опорного напряжения**. Они определяют такие параметры АЦП как диапазон преобразуемых сигналов, разрешающую способность, температурную и временную стабильность. ИОН может быть внутренним и/или внешним. Внутренний ИОН больше подвержен влиянию импульсных помех, возникающих в процессе работы АЦП и ОМК, температур, нестабильности напряжения питания. Лучшие метрологические характеристи-

ки можно обеспечить, используя внешний ИОН, но для этого требуется дополнительная аппаратура;

- **характеристики встроенного программно-управляемого усилителя ПУУ.** ПУУ предназначен для согласования амплитудного диапазона преобразуемого сигнала с диапазоном преобразования АЦП. Коэффициент преобразования этого блока может программироваться в диапазоне от 1 до 128 с шагом кратным степени 2. Некоторые ПУУ позволяют ослаблять сигнал. Например, в ОМК фирмы Silabs коэффициент преобразования изменяется в диапазоне 0,05-16, увеличивая диапазон преобразуемых напряжений до 60В; [11]

- **количество каналов преобразования.** Как правило, АЦП являются многоканальными: 2, 4, 8, 16 и так далее. Коммутация каналов выполняется с помощью аналогового мультиплексора, основными параметрами которого являются сопротивление ключей в открытом и закрытом состояниях, значение проходной емкости, которая влияет на скорость переключения каналов и паразитное прохождение сигнала из закрытого в открытый канал. Мультиплексоры могут коммутировать как потенциальные, так и дифференциальные сигналы. Для дифференциальных сигналов количество входов мультиплексора требуется в два раза больше;

- **способ передачи результатов преобразования:** параллельный и/или последовательный. При этом необходимо контролировать допустимые способы ввода/вывода и тип используемого интерфейса (SPI, I²C, USB, CANbus и так далее). Этот параметр необходимо анализировать при использовании внешних АЦП;

- **дополнительные погрешности,** связанные с отличием условий эксплуатации от условий метрологической аттестации. Необходимо учитывать зависимости приведенных выше характеристик от времени, температуры, напряжения питания, скорости изменения входного сигнала и так далее. Например, **температурная нестабильность** АЦП характеризуется **температурными коэффициентами погрешности** полной шкалы, смещения нуля, напряжения ИОН.

Некоторые виды погрешностей могут быть уменьшены за счет использования определенных структурных и алгоритмических методов:

- **снижение температурной погрешности** обеспечивает установка датчика температур **внутри** АЦП или ОМК, измерение с его помощью температуры T и последующая коррекция результатов измерения по известным характеристикам $N=f(T)$. Для ввода информации с датчика на АЦП используется дополнительный вход аналогового мультиплексора, который недоступен с внешних выводов. Если канал измерения температуры отсутствует, то можно использовать внешний датчик.

Погрешности смещения нуля и полной шкалы могут быть уменьшены за счет использования калибровки, по результатам которой корректируется измеренное значение.

Алгоритм и процедура калибровки может быть различной, но принцип её состоит в измерении напряжений U_0 и $U_{эт}$, соответствующих нижнему и верхнему диапазону измерения, и последующей коррекции результата измерения на основании полученных значений кода N_0 и $N_{эт}$.

$U'_x = (U_x - U_0)(N_x - N_0) / (N_x - N_0)$, где U'_x - скорректированное значение напряжения, N_x - текущее значение кода.

Этот режим предусмотрен во многих прецизионных АЦП и может быть реализован автоматически. При использовании внутренней (резидентной) калибровки используют аналоговую землю AGND и внутренний ИОН. Внешняя калибровка реализуется через определенные каналы аналогового мультиплексора.

Для уменьшения динамической погрешности, связанной с изменением входного сигнала за время преобразования, используется устройство выборки-хранения УВХ. Для АЦП необходимо, чтобы входной сигнал в процессе преобразования был постоянным, иначе возникает погрешность, так как за время измерения сигнал изменился на величину $(U_2 - U_1)$ (рисунок 3.32, а). При наличии УВХ эта погрешность существенно уменьшается, так как в момент запуска АЦП мгновенное значение входного сигнала фиксируется и сохраняется неизменным в течении цикла преобразования (рис/ 3.32 б).

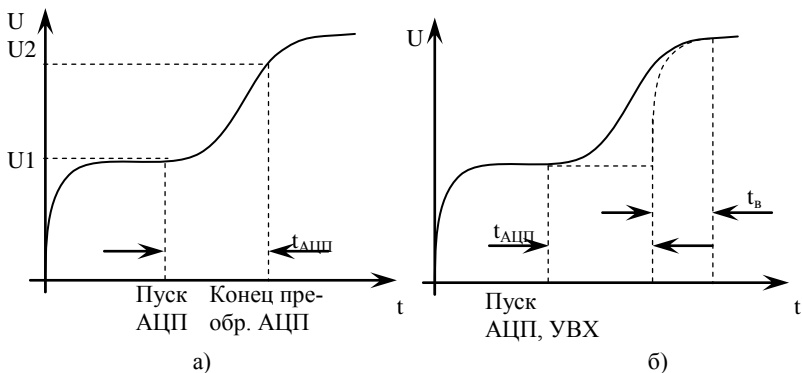


Рис. 3.22. Динамическая погрешность АЦП: а - без УВХ; б - с УВХ

Принцип работы УВХ иллюстрирует рисунок 3.33.

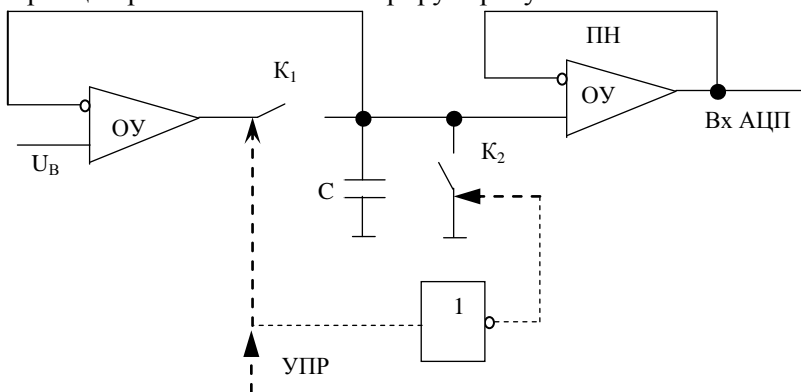


Рис. 3.23. Структура УВХ

Измеряемое напряжение $U_{вх}$ поступает на вход запоминающего устройства, выполненного на базе операционного усилителя ОУ и ключа K_1 . В исходном состоянии ключ K_1 замкнут и сигнал на выходе ОУ повторяет значение измеряемого сигнала. В момент запуска АЦП K_1 размыкается, на конденсаторе C запоминается мгновенное значение измеряемого напряжения и через повторитель напряжения ПН подается на вход АЦП. В момент окончания преобразования ключ K_1 замыкается и через некоторое время $t_{в}$ напряжение на C

вновь выходит на режим слежения входного сигнала. Ключ K_2 может использоваться для принудительного разряда конденсатора.

Использование УВХ устраняет неопределенность, связанную с привязкой результата преобразования к моменту запуска АЦП, что снижает погрешности восстановления аналогового сигнала. При этом общее время преобразования незначительно увеличивается за счет времени, необходимого для выхода на режим слежения (рисунок 3.32 б).

Увеличению быстродействия способствует наличие в схеме АЦП *буферов FIFO*, которые являются местом временного хранения информации, если интерфейс передачи данных занят.

Некоторые АЦП обеспечивают *автоматический выбор диапазона измерения* и ряд других сервисных функций.

Современные интегральные АЦП могут использовать все перечисленные методы уменьшения погрешностей. Однако необходимо контролировать учитываются ли в технических характеристиках перечисленные источники погрешностей.

Известно большое число методов аналого-цифрового преобразования, отличающихся погрешностью, скоростью преобразования и сложностью аппаратной реализации [39,41].

Наибольшее распространение в ОМК получили метод поразрядного уравнивания (последовательного приближения) и сигма-дельта АЦП.

Начало преобразования задает сигнал «Пуск», после прихода которого блок управления БУ устанавливает в регистре последовательного приближения РПП исходный код равный половине диапазона преобразования. Этот код передается в цифроаналоговый преобразователь ЦАП, на выходе которого формируется напряжение $U_{эт}$, определяемое значением опорного (эталонного) напряжения ИОН и переданным кодом. Сравнение измеряемой величины U_x и $U_{эт}$ выполняется аналоговым компаратором АК.

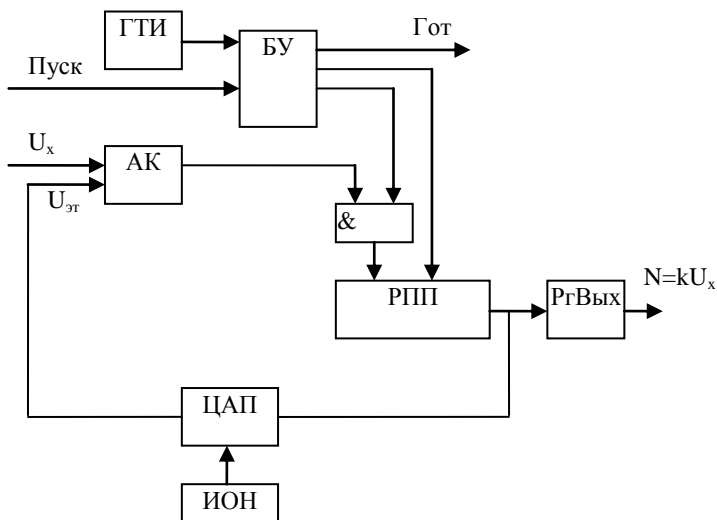


Рис. 3.24. Структура АЦП поразрядного уравнивания

В зависимости от результата сравнения (U_x больше или меньше $U_{эт}$) тактовый генератор ГТИ увеличивает или уменьшает значение кода, подаваемый с РПП на ЦАП. Алгоритмы уравнивания могут быть различны, но наиболее часто в ЦАП подается код в два раза отличающийся от предыдущего изменение кода. В момент равенства U_x и $U_{эт}$ значение РПП фиксируется в выходном регистре РгВых и формируется сигнал готовности «Гот», по которому РгВых считывается процессорным блоком.

Длительность цикла преобразования определяется разрядностью АЦП, частотой ГТИ и алгоритмом уравнивания. Возможно использование дополнительных тактов, связанных с особенностями конкретной реализации АЦП.

АЦП последовательного приближения обеспечивают быстродействие до $(0,2-1)$ мкс и имеют разрядность до 16 бит. Особенностью этих АЦП является независимость времени преобразования от величины входного сигнала и однозначная зависимость быстродействия и разрядности. При фиксированной частоте тактового генератора, время преобразования пропорционально количеству разрядов. По-

этому в некоторых АЦП предусмотрен режим пониженной разрядности, если возникла необходимость повысить быстродействие.

В тех случаях, когда не требуется высокое быстродействие, а необходима высокая разрешающая способность и помехозащищенность, применяют сигма-дельта АЦП, разрядность которых достигает 24 бит, но частота преобразования находится в диапазоне от 10 Гц до сотен КГц. Сигма-дельта АЦП и ЦАП с разрядностью (24-16) бит используются в ОМК, предназначенных для сбора и обработки измерительной информации: ADuC, MSC -1212 и так далее. [42,43]

В сигма-дельта АЦП аналоговый сигнал квантуется с очень низким разрешением (как правило, 1 бит) на частоте, во много раз превышающей максимальную частоту спектра сигнала.

Особенность сигма-дельта АЦП состоит в том, что спектр шума квантования у него распределен неравномерно. Он смещен в сторону высоких частот. Поэтому увеличивая время измерения и, соответственно, количество выборок измеряемого сигнала, накапливая и затем усредняя полученную выборку фильтром нижних частот, можно получить результат измерения с более высокой точностью. Для этих целей применяют метод передискретизации (повышение частоты дискретизации $F_{пд}$ в сотни и тысячи раз по сравнению с теоремой Котельникова) в сочетании с цифровой фильтрацией, что соответственно снижает погрешность квантования и повышает разрешающую способность.

Для снижения эффективной скорости поступления отсчетов на выходе АЦП применяется децимация. Однобитовые сигма-дельта АЦП и ЦАП обладают превосходной дифференциальной и интегральной линейностью благодаря линейности однобитного квантователя. Здесь не требуется высокоточная лазерная подгонка элементов ЦАП, как в других архитектурах АЦП. Одноразрядный ЦАП представляет собой разнополярные источники опорного напряжения, коммутируемые аналоговым ключом.

Блок-схема сигма-дельта АЦП первого порядка представлена на рисунке 3.35. Входная (аналоговая) часть такого класса приборов - сигма-дельта модулятор, преобразующий входной сигнал в последовательный непрерывный поток нулей и единиц, следующих с частотой $F_{пд} = K * F_s$, где K - коэффициент передискретизации, F_s - частота дискретизации, определенная на основании теоремы Котельникова.

Замкнутая цепь обратной связи состоит из вычитающего устройства Σ , интегратора I , компаратора K , однобитного ЦАП. Этот ЦАП принимает последовательный поток данных, а сигнал с его выхода вычитается из входного сигнала. Из теории обратной связи следует, что средняя величина напряжения на выходе ЦАП при достаточном петлевом усилении может достигать значения на входе модулятора.

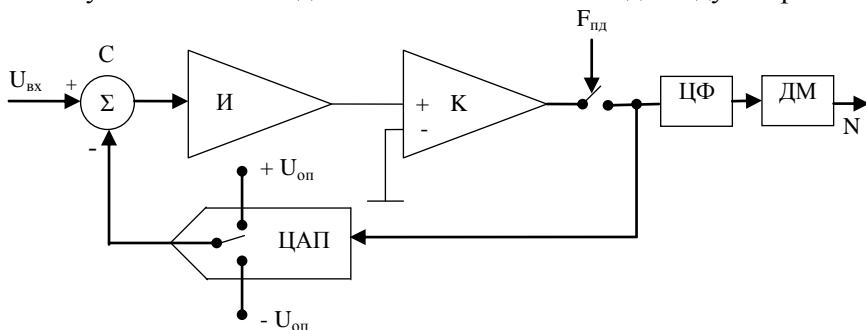


Рис. 3.35. Структурная схема сигма-дельта АЦП

Интегратор может быть представлен как фильтр, амплитуда отклика которого пропорциональна $1/f$, где f - частота входного воздействия.

Компаратор синхронизируется тактовыми импульсами, следующими с частотой $F_{пл}$, преобразуя медленный входной сигнал в сигнал переменного тока высокой частоты, меняющейся в зависимости от среднего значения напряжения на входе.

Таким образом эффективное значение шума квантования на низких частотах пренебрежимо мало, а интегратор выступает в роли фильтра высоких частот для шума квантования.

Если входной сигнал близок к верхнему диапазону измерений, то в битовом потоке на выходе больше единиц, чем нулей, и наоборот, если сигнал ближе к нижнему, то больше нулей. Для сигнала, близкого к середине шкалы, количество нулей и единиц примерно одинаково. Для декодирования выходного потока выходные отсчеты пропускаются через цифровой фильтр ЦФ низкой частоты.

Снижение частоты вывода данных выполняется с помощью дециматора ДМ. Децимация может также рассматриваться как метод устранения избыточной информации, связанной с передискретиза-

цией. В сигма-дельта АЦП широко используется совмещение функций цифрового фильтра и дециматора, что приводит к повышению быстродействия.

На рисунке 3.36 он показан для дискретного сигнала, где частота дискретизации входного сигнала $X(m)$ уменьшена в 4 раза. Сигнал $S(n)$ пересчитывается на более низкую частоту квантования (частоту децимации).

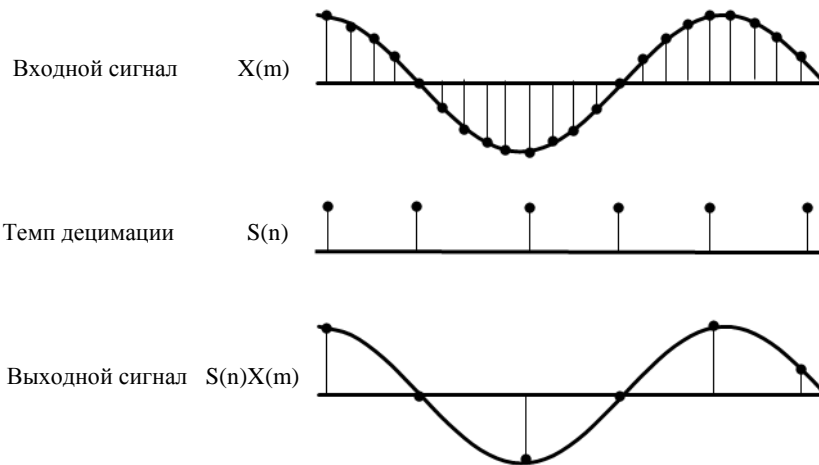


Рис. 3.36. Децимация дискретного во времени сигнала

Однобитные сигма-дельта АЦП используются для постоянных или медленно меняющихся входных сигналов из-за низкой скорости преобразования, так, как только за 2^N тактов цикла можно достичь N -бит эффективного разрешения.

Для повышения скорости преобразования и разрешающей способности сигма-дельта АЦП применяют специальные способы: модуляторы n -порядка, n -разрядные АЦП и ЦАП [44,45].

3.5.4. Особенности организации блока АЦП ОМК

Блок АЦП ОМК строится на основании принципов, описанных выше, и оценка пригодности его для решения задач автоматизации осуществляется аналогично интегральным АЦП. Однако имеется

ряд особенностей по сравнению с использованием внешних интегральных АЦП:

В большинстве ОМК имеется *регистра́вый файл* для хранения результатов многоканального измерения, объем которого следует контролировать.

Возможно использование *различных режимов ввода-вывода информации* (программного, по прерыванию, прямой доступ к памяти (ПДП)). В режиме ПДП управление обменом выполняется *аппаратно* без привлечения ресурсов процессора. Местом хранения кода АЦП может быть, как резидентная, так и внешняя память. Обычно ПДП используется в специализированных ОМК, предназначенных для сбора и обработки измерительной информации (AduC, MSP-430, xMEGA и так далее).

Наличие *регистра перегрузки*, каждый бит которого устанавливается в том случае, если значение входного сигнала соответствующего канала больше $U_{ион}$.

Использование *режима порогового детектирования*, при котором программно задаются минимальное и/или максимальное значения кода и при нарушении установленных границ поканально устанавливаются флаги, вызывающие процедуру прерываний. [12]. Этот механизм особенно эффективен в системах реального времени, работающими с прерываниями, так как позволяет уменьшить время реакции на превышение порога и объем программного обеспечения.

Возможность *программного управления коэффициентом усиления, частотой преобразования, разрядностью* (для АЦП поразрядного уравнивания), *коэффициентом передискретизации и децимации* (для сигма-дельта АЦП).

Наличие **режимов последовательного и выборочного опроса каналов**. При последовательном опросе каналы опрашиваются поочередно, поэтому время дискретизации для *каждого канала* $T = T_{ацп} * n$, где $T_{ацп}$ - быстродействие АЦП, а n - количество каналов. В тех случаях, когда необходимо увеличить частоту дискретизации для одного или нескольких каналов, используется выборочный опрос. Порядок коммутации каналов в этом случае может быть произвольным.

Различные *способы запуска АЦП*. Наиболее часто используется программный запуск по команде процессорного блока. Однако воз-

можно внешний запуск, запуск по переполнению счетчика-таймера или по формированию сигнала различных событий.

Сигнал готовности АЦП может в свою очередь **инициировать** работу периферийных устройств ОМК. Наиболее часто это режим захвата счетчика -таймера, но при наличии блока событий (xMEGA) возможны и другие варианты [12,24].

Контроль напряжений внутренних источников сигналов: датчика температур, ИОН, линейный выход ЦАП, питание ОМК и так далее.

Возможность работы **в режиме пониженного энергопотребления**.

К **недостаткам** использования **внутреннего АЦП** ОМК относятся

- сокращение числа линий, предназначенных для цифрового ввода-вывода, так как при наличии АЦП порт работает в совмещенном режиме;
- увеличенная погрешность и меньшее быстродействие по сравнению с автономными интегральными АЦП;
- большее влияние внутренних импульсных помех. Этот недостаток можно снизить, используя режимы энергосбережения, при которых отключаются наиболее существенные внутренние источники помех. Переход в рабочий режим осуществляется после завершения цикла преобразования по сигналу готовности АЦП. При этом необходимо учитывать, что использование такого метода приведет к снижению быстродействия. В некоторых ОМК имеются рекомендованная частота преобразования АЦП, на которых помехи минимальны. Однако это значение меньше максимально допустимой частоты преобразования.

Поэтому при повышенных требованиях к метрологии рекомендуется использовать внешние АЦП с последовательным каналом обмена (SPI, I2C).

Обобщенная структура блока АЦП приведена на рисунке 3.37, где $U1 - Un$ - внешние входные сигналы АЦП, $Uв$ - внутренние сигналы ОМК, $АМК$ - аналоговый мультиплексор каналов, ПУУ-программно-управляемый усилитель, $УВХ$ - устройство выборки хранения, $РзАК$ - регистр адреса каналов, $Рз КУ$ - регистр коэффициента усиления, $СВЗ$ - схема временной задержки, $АМ ИОН$ - анало-

говый мультиплексор источника опорного сигнала, F - выходной сигнал программно-управляемого генератора, ШУ- шина управления, ШД - шина данных.

Программная модель АЦП состоит из регистров

- управления, в которых задаются основные режимы работы;
- состояния, содержащих флаги готовности, переполнения, превышения допустимого значения;
- данных;
- регистров, задающих номер канала, коэффициент усиления, коэффициент деления частоты и так далее.

В различных ОМК число регистров может быть от 3 до 12 и более.

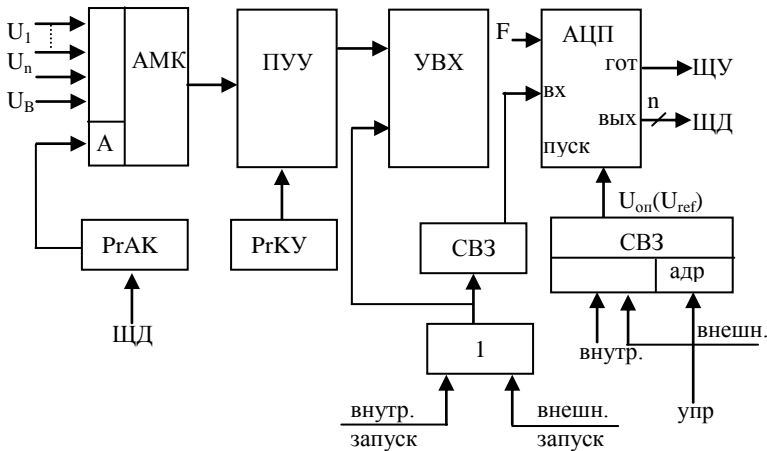


Рис. 3.37. Обобщенная структура блока АЦП ОМК

4. Особенности структурной организации ОМК

Структуры ОМК развиваются в соответствии с требованиями, предъявляемыми разработчиками современных автоматизированных систем.

Основными из этих требований являются: оптимальная система команд, позволяющая реализацию необходимых алгоритмов с минимальными затратами памяти, быстродействие (производительность), состав и количество периферийных устройств (функцио-

нальная насыщенность), их технические характеристики, количество линий ввода-вывода, условия эксплуатации, потребляемая мощность, конструктивные особенности корпуса, удельная стоимость требуемых функций.

Одновременно эти требования выполнены быть не могут, так как для реализации полного набора свойств требуются в некоторых случаях противоположные средства. Например, быстродействие и потребляемая мощность, функциональная насыщенность и количество линий ввода-вывода, количество линий ввода-вывода и габариты корпуса, шаг между контактами и так далее. Поэтому в зависимости от важности той или иной характеристики применяются различные структурные методы.

К общим **методам повышения быстродействия** относятся

- увеличение разрядности АЛУ, запоминающих устройств и внутренних магистралей. Разрядность - 8, 16, 32, 64.
- повышение частоты системного генератора и использование внутренних умножителей частоты;
- увеличение мощности (функциональности) команд и их аппаратная реализация: использование трехадресных команд, расширенное число способов адресации, возможность работы с двойной и четверной точностью, кодирование в команде одновременно нескольких операций;
- аппаратная реализация вычислительных операций и процедур: умножения, деления, умножения с накоплением ($A := (X * Y) + A$), сдвигов, вычисления контрольной суммы CRC, команд нечеткой логики и криптографических алгоритмов и так далее. Микроконтроллеры с командами нечеткой логики выпускают фирмы STMicroelectronics (STM 52 Duallogic, STM 62), Motorola (68HC12), Siemens (SAE81C99) и другие [46]. Криптографические микроконтроллеры содержат аппаратные акселераторы для генерации случайных чисел, шифрования и дешифрования, формирования и проверки цифровой подписи и тому подобное; [47, 48]
- децентрализация сбора и обработки информации за счет применения сопроцессоров, блоков управления памятью, периферийных блоков, содержащих локальные центры обработки информации: интеллектуальные порты ввода-вывода, сопроцессоры временных событий и так далее;

- применение многоядерных микроконтроллеров. Например, фирма XMOS выпускает микроконтроллеры универсального назначения семейства xCORE с количеством ядер до 16. Скорость для 8-ядерного ОМК составляет 1000 MIPS. [49] Многоядерные микроконтроллеры выпускают фирмы Texas Instruments, Parallax и другие; [50]

- использование проблемно-ориентированных ОМК для реализации заданного класса алгоритмов: цифровые сигнальные, коммутационные, мультимедийные микроконтроллеры;

- увеличение общего объема резидентной памяти команд и данных и совершенствование их организации, например, выборка за одно обращение нескольких команд;

- использование «теневых» банков регистров при обработке прерываний, которые позволяют не сохранять контекст программы, так как за определенным вектором прерываний закрепляется «свой» банк регистров (например, в микроконтроллерах MIPS). Этот способ можно использовать в ОМК, имеющих оконную организацию памяти. Например, в 8051 возможно закрепление банков РОН за определенными источниками прерываний;

- использование ОЗУ для хранения копии памяти программ. Например, FPSLIC;

- использование кэш-памяти, если не требуется жесткий режим реального времени;

- многоступенчатый конвейер с блоками предварительной выборки команд и предсказания ветвлений;

- увеличение количества внутренних магистралей, обеспечивающих одновременный обмен между несколькими устройствами. В качестве примера рассмотрим структуру микроконтроллера XMEGA фирмы Atmel. [24]

В ОМК используются индивидуальные шины для доступа к резидентной памяти, *EEPROM*, памяти ввода-вывода и внешней памяти, что обеспечивает одновременный доступ к различным запоминающим устройствам (рисунок 4.1).

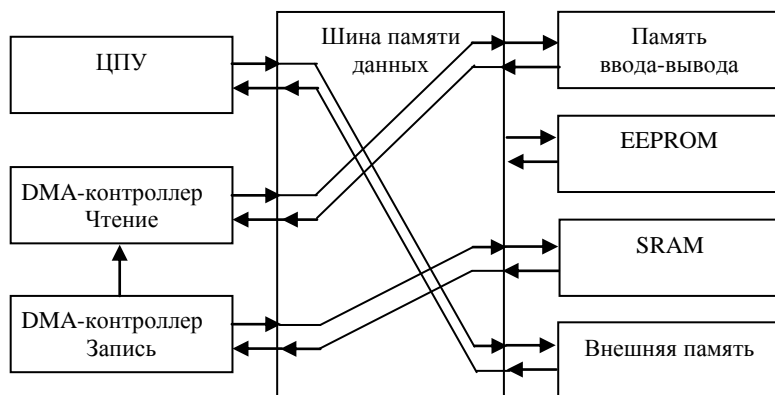


Рис. 4.1. Обмен данными в ОМК XMEGA

Как показано на рисунке 4.1, ЦПУ может выполнять доступ к внешней памяти, а DMA-контроллер одновременно с этим - передавать данные из внутреннего SRAM в память ввода-вывода. Если несколько ведущих устройств запрашивают доступ к одной шине, то доступ будет предоставляться с учетом определенных приоритетных правил.

Повышению быстродействия способствует применение системы событий, предназначенной для соединения между собой внутренних устройств ввода-вывода УВВ.

С ее помощью можно добиться автоматического выполнения определенных действий одним УВВ одновременно с произошедшими изменениями в другом УВВ. Взаимодействия УВВ задаются программно.

Данная система позволяет автономно управлять УВВ без использования прерываний, ЦПУ или ресурсов DMA. 8 каналов событий позволяют одновременно пропускать до 8 сигналов (рисунок 4.2).

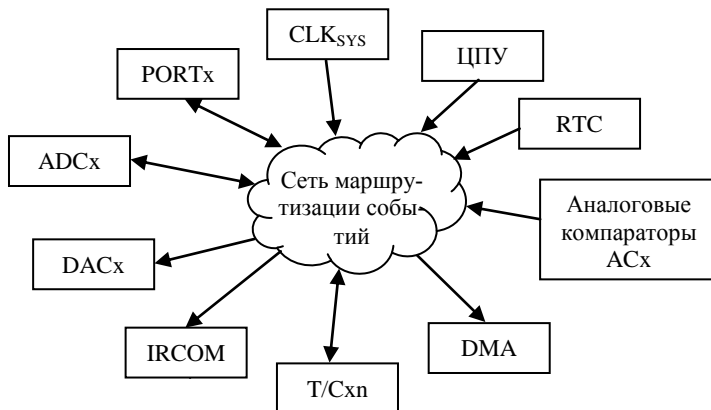


Рис. 4.2. Структура системы событий

На рисунке 4.2 *RTC* - таймер реального времени, *DMA* - контроллер прямого доступа к памяти, *T/C* - счетчики /таймеры, *IRCOM* - инфракрасный порт, *DAC* - цифроаналоговый преобразователь, *ADC* аналогоцифровой преобразователь, *CLK* - частота системного генератора.

Сеть маршрутизации событий состоит из восьми мультиплексов (CHnMUX), которые позволяют соединить любой из источников событий к любому из выходов мультиплексов (рисунок 4.3).

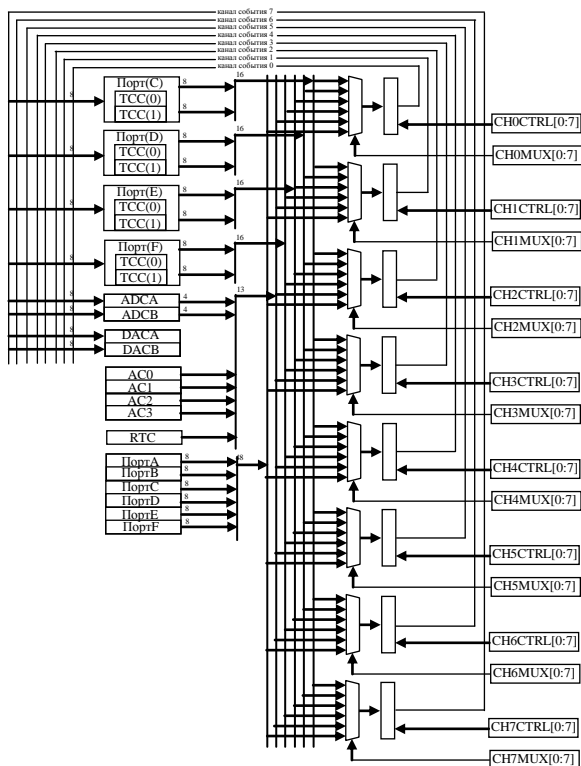


Рис. 4.3. Сеть маршрутизации событий XMEGA

Одновременно можно передавать до восьми событий. При необходимости одно и тоже событие может передаваться через несколько мультиплексов.

В процессе развития микроконтроллеров перечисленные выше методы находят своё применение во многих семействах ОМК, но наиболее полно они реализованы в ОМК с ядром ARM и Cortex. [51-56]

Большинство методов повышения быстродействия базируются на магистральной архитектуре внутреннего интерфейса (шины данных, адреса, управления). В многоядерных микроконтроллерах шинная архитектура становится причиной снижения быстродействия, так как *в каждый момент времени только одно из ядер может быть ведущим*, а остальные являются ведомыми. Большое быстродействие обеспечит переход от шинной архитектуры к сети с коммутацией пакетов Network on Chip (NoC). В архитектуре NoC каждое ядро или блок процессора соединён с маршрутизатором, через который происходит его общение с другими блоками. Сами маршрутизаторы объединены в сеть, по которой пакеты данных передаются от одного блока к другому, так же как пакеты в обычной компьютерной сети. Это значительно упрощает топологию микросхемы и снимает ограничения по масштабированию — в отличие от шины, множество блоков способно общаться одновременно, не мешая друг другу. [57,58]

Большое разнообразие методов, повышающих быстродействие, используется в системах на кристалле SoC, элементной базой которых являются программируемые логические интегральные схемы ПЛИС [59].

Снижение количества линий ввода-вывода ОМК без потери функциональных возможностей обеспечивает применение коммутационной матрицы. На рисунке 4.4 показана коммутационная матрица микроконтроллера C8051xxx фирмы SiLabs.

Программируемая коммутационная матрица ПКМ позволяет переклывать периферийные внутренние блоки (матрицу счетчиков/таймеров МСТ, последовательные интерфейсы, прерывания, аналоговые входы АЦП, выходы ЦАП и компараторов АК, регистры защелки портов и другие цифровые сигналы) на требуемые линии портов ввода-вывода. ПКМ представляет собой многоканальный аналоговый мультиплексор с числом входов, равным числу входов/выходов резидентных периферийных блоков, и числом выходов, равным числу линий (выводов) нескольких портов ввода/вывода (обычно 2-3 порта). Часть линий портов может быть закреплена за определенными периферийными блоками. Приоритетный дешифратор ПД задает определенные приоритеты между периферийными устройствами.

Имея в распоряжении все ресурсы микроконтроллера, пользователь может включить в работу только те блоки, которые ему необходимы для решения конкретной задачи. Причем конфигурация МК может быть изменена в процессе работы- «на лету». [12]

Аналогичный метод используется в микроконтроллерах PSoC, MicroChip. [11,38]

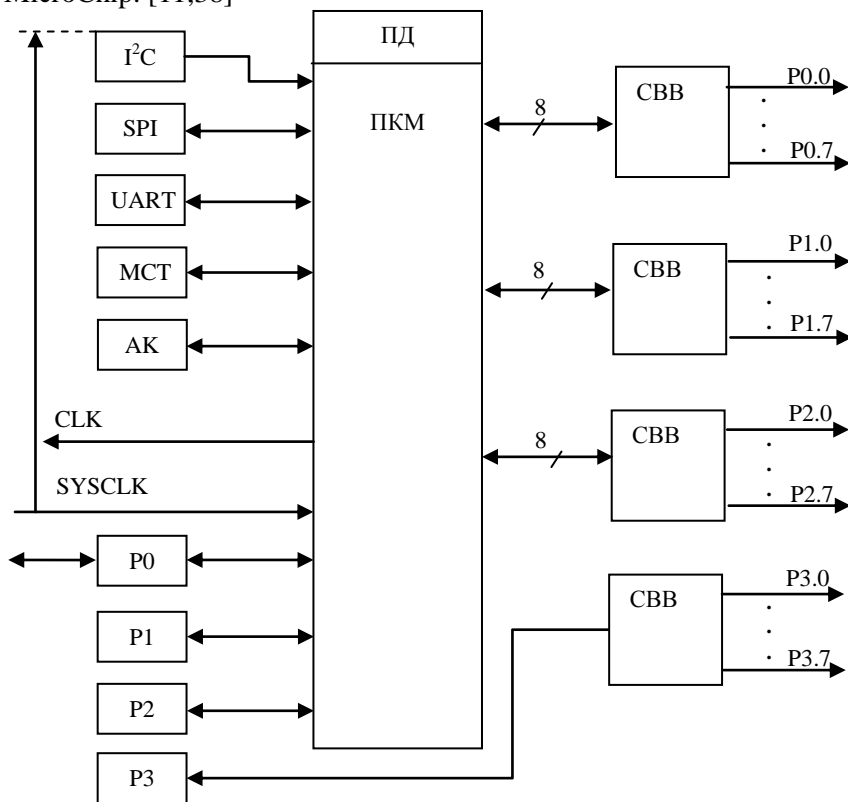


Рис. 4.4. Схема коммутационной матрицы

Снижения энергопотребление ОМК можно добиться несколькими способами:

- Уменьшение напряжения питания и тактовой частоты до уровня, достаточного для выполнения требуемого приложения или задачи. Этот уровень особенно важен для режима реального време-

ни. Напряжение питания может изменяться в диапазоне 1,6 В до 5В, а частота от 0 до максимума Гц. При этом следует контролировать, чтобы и аналоговые блоки работали в этих диапазонах;

- включение питания и подачу тактовой частоты *только на работающие* периферийные модули;

- Возможность независимого управления тактовой частотой для различных устройств ОМК, соответствующей их назначению (процессор, ОЗУ, флеш, периферия). Некоторые микроконтроллеры выполняют управление тактовой частотой автоматически при определенных программных настройках. Автоматическое управление тактовым сигналом разрешает поступление тактового сигнала только на те регистры, которые изменяют свое состояние. Если регистр не изменяет свое состояние, тактовый сигнал к нему не поступает;

- использование кварцевого генератора 32 кГц с ультранизким энергопотреблением;

- большим количеством «спящих» режимов, отличающихся быстродействием и коэффициентом снижения энергопотребления;

- оптимизацией технологии, схемотехники внутренних компонентов и их соединений. Для минимизации тока потребления необходимо уменьшать количества логических элементов и частоту их переключений, использовать *однофазную систему управления тактовыми сигналами*.

По мере уменьшения размеров элементов все большее значение приобретает энергопотребление соединительных проводников и задержки, вносимые ими. Наиболее остро эта проблема проявляется в шинных соединениях, поскольку они могут рассеивать до трети всей потребляемой мощности. Существует несколько подходов к оптимизации энергопотребления шины: уменьшение амплитуды сигнала, применение кодов, обеспечивающих минимальное количество переключений, использование бесконтактных технологий. [60]

Для реализации бесконтактных соединений между слоями микросхемы могут использоваться эффекты емкостной связи между миниатюрными контактными площадками или индуктивной - между спиральными катушками индуктивности, радиоинтерфейс на основе дипольных антенн, гибридные технологии из проводных и беспроводных каналов связи. [58] Например, при скорости передачи данных 23 Гбит/с традиционный проводной контакт рассеивает мощ-

ность около 200 мВт, а с индуктивной связью при технологической норме изготовления индукторов 90 нм и скорости 23 Гбит/с рассеиваемая мощность снижается до 2 мВт на контакт. [58]

Наиболее известные из малопотребляющих ОМК семейство MSP430F5xx [61,62], AVR-микроконтроллеры picoPower компании Atmel [63], EFM32 Zero Gecko фирмы SiLabs [64] и другие. [65]

5. Способы реализации структур микроконтроллеров

В зависимости от способа реализации микроконтроллеры можно классифицировать следующим образом:

ОМК со структурой фиксированной разработчиком и изготовителем (производителем). К этому виду можно отнести большинство микроконтроллеров общего применения и проблемно-ориентированных (цифровые сигнальные процессоры, коммуникационные контроллеры и так далее). Пользователь выбирает ОМК с конфигурацией, соответствующей решаемой задаче. Изменение технических требований может потребовать замены типа микроконтроллера.

ОМК со структурой, формируемой на стадии изготовления. В этом случае необходимые характеристики и конфигурация реализуются за счет использования унифицированных блоков и магистралей. Такой подход позволяет существенно сократить время проектирования микроконтроллера и обеспечивает масштабирование процессорного блока, памяти и периферийных устройств, то есть имеет возможность выбрать разрядность процессора, объем памяти, параметры портов, АЦП и так далее.

Пример ОМК фирмы Motorola с подобной структурой приведен на рисунке 5.1. Модули процессорного блока *ПБ*, *ПЗУ*, *ОЗУ*, сопроцессора *СП*, последовательного интерфейса *ПИ*, матрица счетчиков-таймеров *МСТ*, *АЦП*, системной интеграции *МСИ* объединены стандартизированной межмодульной магистралью *ММ*. *МСИ* содержит средства управления и синхронизации, координирует работу *ПБ* с внутренними модулями ОМК и внешними устройствами. [26]

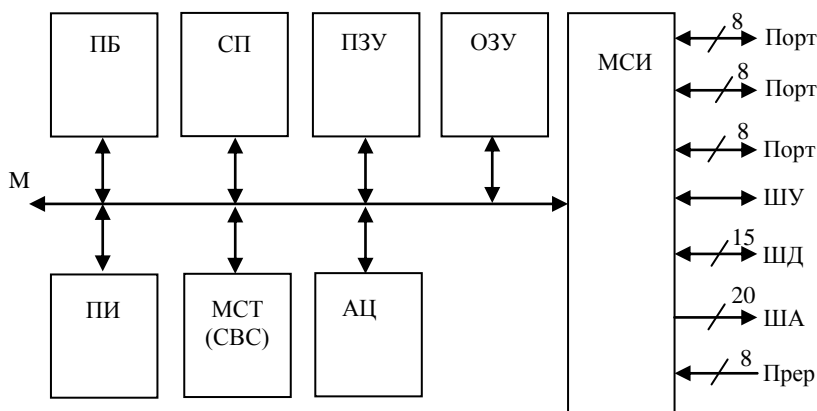


Рис. 5.1. ОМК с заказной структурой

Системы на кристалле. В настоящее время наиболее перспективной элементной базой для построения цифровой, аналого-цифровой и цифроаналоговой аппаратуры разнообразного назначения являются программируемые интегральные логические схемы **ПЛИС**.

К основным достоинствам ПЛИС по сравнению с микроконтроллерами следует отнести:

Возможность замены **программной интерпретации алгоритмов их аппаратной реализацией**. Для выполнения определенных операций применяются индивидуальные функциональные блоки, соединяемые в цепь определенной структуры. Функциональные характеристики блоков и характер соединений между ними соответствуют алгоритму решения задачи. Команды для выполнения отдельных операций могут отсутствовать. Это обеспечивает децентрализованную обработку информации.

Структура контроллера может быть разработана **под конкретную задачу**. Возможна модификация разрядности и функциональности процессорного блока, выбор необходимых периферийных блоков. Например, нужное количество счетчиков-таймеров, последовательных портов и так далее.

Репрограммируемые микросхемы ПЛИС позволяют разрабатывать системы, *структура которых может изменяться динамически в процессе функционирования*.

Наличие разнообразных программных средств автоматизированного проектирования с библиотеками типовых модулей, *малое время проектирования и отладки проектов*

Простота модификации проектов на любых стадиях их разработки. [59]

Уровень технологии производства интегральных схем приводит к постоянному росту количества элементов на кристалле, которое в сложных СБИС оценивается сотнями миллионов вентилей. Рабочие частоты таких СБИС составляют гигагерцы. Это позволяет на одной СБИС разместить законченную систему обработки данных с требуемыми техническими характеристиками. Подобные микросхемы называются *системами на кристалле* и обозначаются как SOC (System On Chip), или как SOPC (System On Programmable Chip), или как PSOC (Programmable System On Chip), или как CSOC (Configurable System On Chip) и т. д.

Системы на кристалле могут быть реализованы как проблемно-ориентированные *заказные* СБИС, структура которых не может быть изменена в процессе эксплуатации (*ASIC*), или программируемые пользователем (*FPGA/CPDL*).

CPLD (англ. complex programmable logic device — сложные программируемые логические устройства) содержат относительно крупные программируемые логические блоки — макроячейки (англ. macrocells), соединённые с внешними выводами и внутренними шинами. Функциональность CPLD кодируется в энергонезависимой памяти, поэтому нет необходимости их перепрограммировать при включении. Может применяться для расширения числа входов/выходов рядом с большими кристаллами, или для предобработки сигналов (например, контроллер COM-порта, USB, VGA).

FPGA (англ. field-programmable gate array) содержат блоки умножения-суммирования, которые широко применяются при обработке сигналов (DSP), а также логические элементы (как правило, на базе таблиц перекодировки — таблиц истинности) и их блоки коммутации. FPGA обычно используются для обработки сигналов, имеют больше логических элементов и более гибкую архитектуру,

чем CPLD. Программа для FPGA хранится в распределённой памяти, которая может быть выполнена как на основе энергозависимых ячеек статического ОЗУ (подобные микросхемы производят, например, фирмы Xilinx и Altera) — в этом случае программа не сохраняется при исчезновении электропитания микросхемы, так и на основе энергонезависимых ячеек Flash-памяти или перемычек antifuse (такие микросхемы производит фирма Actel и Lattice Semiconductor) — в этих случаях программа сохраняется при исчезновении электропитания. Если программа хранится в энергозависимой памяти, то при каждом включении питания микросхемы необходимо заново конфигурировать её при помощи начального загрузчика, который может быть встроен и в саму FPGA. Альтернативой ПЛИС FPGA являются более медленные цифровые процессоры обработки сигналов. FPGA применяются также, как ускорители универсальных процессоров в суперкомпьютерах (например, Cray — XD1, SGI — Проект RASC).

Структура микросхемы FPGA может модифицироваться в зависимости от решаемых задач в процессе эксплуатации и даже «на лету». Время регенерации структуры составляет 100 мкс и ниже.

Применение ASIC целесообразно при массовом производстве. Для большинства разработчиков по экономическим соображениям более предпочтительно проектирование на основе FPGA.

Программируемые системы на кристалле ПСК (SOPC) могут быть выполнены на однородной или блочной структурах.

ПСК с однородной структурой позволяют синтезировать функциональные блоки в *любой* области кристалла. При проектировании могут использоваться как собственные разработки, так и виртуальные компоненты, разработанные другими фирмами (soft-ядра, IP).

ПСК блочного типа состоят из **фиксированных** областей кристалла, в которых реализованы блоки **с неизменяемыми** в процессе работы функциями (hard-ядра), и областей, предназначенных для программирования пользователем (разработчиком). По сравнению с soft-ядрами hard-ядра обладают большим быстродействием и занимают на кристалле значительно меньшую область (в разы), так как они не содержат средств конфигурирования и оптимизированы для реализации **конкретной, фиксированной** функции. Обычно hard-ядрами из-за экономических соображений служат широко распространённые функциональные блоки: процессоры, микроконтролле-

ры, стандартные интерфейсы, контроллеры интерфейсов PCI, VME, USB, CAN, UART, I2C, SPI и другие. ПСК блочного типа следует применять в системах с высоким быстродействием. [59]

Обобщенная структура программируемой системы на кристалле представлена на рисунке 4.6., где *ПБ* - процессорный блок, *МВИ* - модуль внешнего интерфейса, *ВМ* - внешняя магистраль, *ПВВ* - программируемые порты ввода-вывода, *РП* - резидентная память, *ВКИ* - внутренний коммуникационный интерфейс, *ПУ* - периферийные устройства, *ВП* - внешняя память.

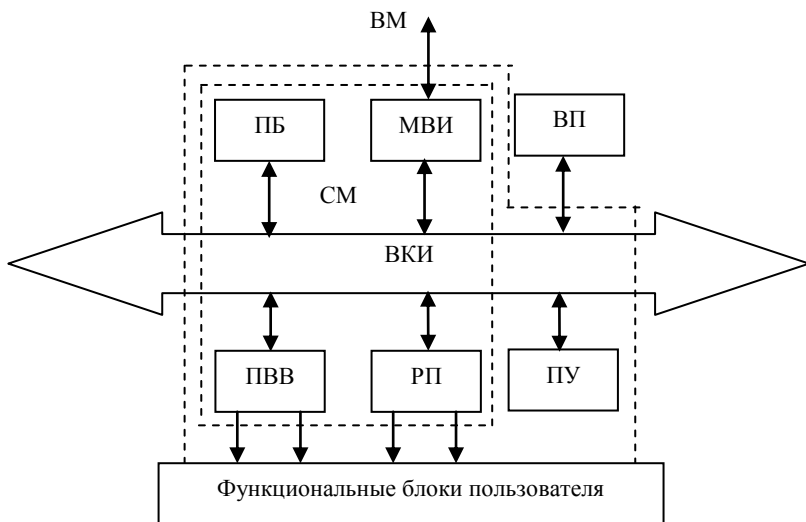


Рис. 5.2. Обобщенная структура программируемой системы на кристалле

Центральным блоком ПСК является системный модуль *СМ*, который может быть реализован на основе *soft-ядер*, *hard-ядер* или их комбинации. Остальную часть кристалла пользователь может использовать по своему усмотрению. В качестве процессорного блока могут использоваться как ядра широко известных *ОМК* (*MCS-51/52*, *ATMEG*, *Motorola*, *PIC*, *ARM*, *Cortex*, *MIPS*, *PowerPC* и так далее), так фирменные разработки.

Ведущими фирмами по производству и разработке ПСК являются *Xilinx* и *Altera*. [66-69]

Особенности ПСК рассмотрим на примере фирмы Altera, которая в качестве СМ использует ядро *Nios*, а ВКИ - интерфейс *Avalon*. [70]

Nios — это встраиваемый процессор общего назначения, с перестраиваемой конфигурацией, который легко вписывается в устройство Altera ® APEx™, оставляя много ресурсов для размещения периферийных устройств и функций пользователя. Встраиваемое ядро процессора *Nios 2* — конвейерный RISC-процессор, в котором команды выполняются за один цикл частоты синхронизации, частота генератора 250 МГц, разрядность данных 32 бита, адресное пространство 8 Гб., команды -трехадресные, длина формата -32 бита, 6-уровневый конвейер, отдельный кэш инструкций и кэш данных; аппаратное выполнение умножения и деления; возможно использование плавающей арифметики, большое количество векторов прерываний, число базовых регистров - 32, общее число регистров - до 512, что позволяет динамически распределять регистровый файл. Например, при обработке прерываний можно не сохранять в стеке используемые регистры, а иметь копии в регистровом файле. Программная модель процессора кроме стандартных инструкций поддерживает выполнение команд, формируемых пользователем. В процессе разработки имеется возможность варьировать функциональностью процессора, чтобы оптимизировать соотношение стоимость\производительность. [71, 72]

В состав soft-ядер входят так же средства отладки и программирования (порт JTAG), основные последовательные интерфейсы, широтно-импульсные модуляторы, контроллеры дисков и так далее.

Средством коммуникации между блоками ПСК является синхронная шина *Avalon*. В отличие от традиционных магистралей, в которых обмен информацией реализуется в режиме с разделением времени по принципу «ведущий-ведомый», шина использует **индивидуальные симплексные** линии данных, адреса и управления, мультиплексирование и **арбитраж на стороне ведомого устройства**, динамическое управление разрядностью шины, обеспечивает **работу нескольких ведущих устройств одновременно**. В режиме **мультимастера** данные могут передаваться **одновременно разным ведомым устройствам**. Арбитраж на уровне ведущих устройств отсутствует, а схема арбитража находится в ведомом устройстве и

включается в тот момент, когда происходит обращение к ведомому устройству со стороны нескольких ведущих. Это существенно увеличивает производительность.

Avalon это не набор пассивных линий, а модуль в состав которого входят активная логика, состоящая из индивидуальных шин, сигналов выборки, мультиплексоров, приоритетных шифраторов, коммутационной матрицы, устройств синхронизации.

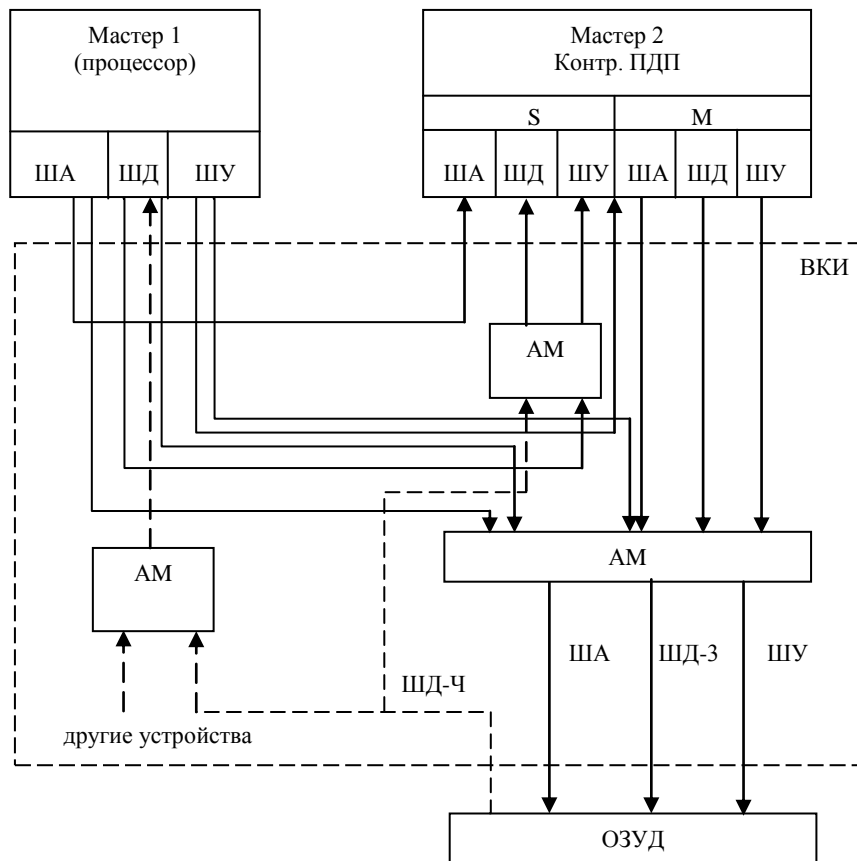


Рис. 5.3. Организация взаимодействия ведущих устройств с памятью данных на шине Avalon

На рис.5.3 показано взаимодействие процессора и контроллера ПДП с памятью данных ОЗУД, где сплошные линии - линии записи,

пунктирные - *линии чтения, ША, ШД, ШУ* - соответственно шины адреса, данных и управления, *АМ* - арбитраж-мультиплексор. Контроллер ПДП может работать в режиме ведомого (S), когда выполняется инициализация. или ведущего (M) при обмене информацией с памятью.

Продукция фирмы Altera динамично развивается, имеются разработки, в которых используются hard-ядра. Эти ядра базируются на RISC-архитектурах процессоров фирм ARM Limited, MIPS Technologies и IBM Microelectronics. Известны реализации Arria 10 SoC аппаратного двухъядерного процессорного блока ARM Cortex A9, работающего на частоте до 1,5 ГГц, Stratix® 10 SoC с 4-ядерным 64-битным процессором ARM Cortex-A53, RISC-ядром MP32 с архитектурой MIPS и так далее. [68]

Аналогична тенденция развития фирмы Xilinx. [69]

Несмотря на всю привлекательность ПСК на FPGA имеют ***некоторые недостатки: высокая стоимость для микросхем большого объема, необходимость приобретения лицензий на использование интеллектуальной собственности IP, при отсутствии исходного кода ядер усложняется процесс отладки, высокая стоимость отладочных средств.***

Более оптимальным в ряде приложений является использование в одном кристалле hard- и soft-ядер. Примером подобного подхода может служить семейство фирмы Atmel AT94K, называемое FPSLIC (Field Programmable System Level Integration Circuits - программируемые микросхемы системного уровня интеграции). [73-75]

FPSLIC содержит на кристалле программируемую логическую FPGA-матрицу объемом от 10 до 40 тысяч элементов, микроконтроллер, память и согласующую логику. Внешний вид кристалла FPSLIC показан на рис. 5.4. Особенность технического решения состоит в том, что микроконтроллер и память реализованы аппаратно. [75]

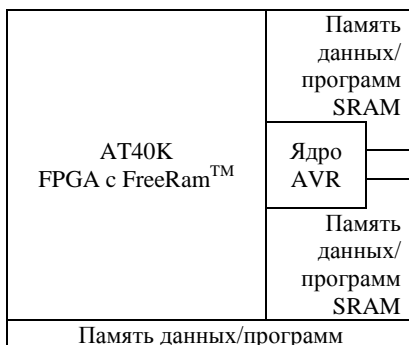


Рис. 5.4. Структура кристалла FPSLIC

Особенностями FPSLIC являются:

- использование механизма копирования памяти программ в статическое ОЗУ, что значительно повышает быстродействие;
- размещение на кристалле модуля FPGA, стандартного микроконтроллера семейства AVR и аппаратного блока, обеспечивающего их взаимодействие;
- архитектура статической памяти позволяет гибко управлять конфигурацией и объемом адресного пространства (рис 5.5). Фиксированный объем памяти программ 10Kx16, а данных -4Kx8;
- память организована как двухпортовое ОЗУ. FPGA и AVR могут осуществлять доступ к памяти данных независимо друг от друга. FPGA может записывать и считывать данные из SRAM в то же самое время, когда AVR записывает или считывает данные. Такая структура позволяет разработчику реализовать различные способы буферного обмена данными между FPGA и AVR: буферы типа LIFO, FIFO или другие. Предотвращение конфликтов при одновременном обращении обоих ядер к одному адресу ОЗУ должна обеспечивать схема арбитража. (рис 5.6)

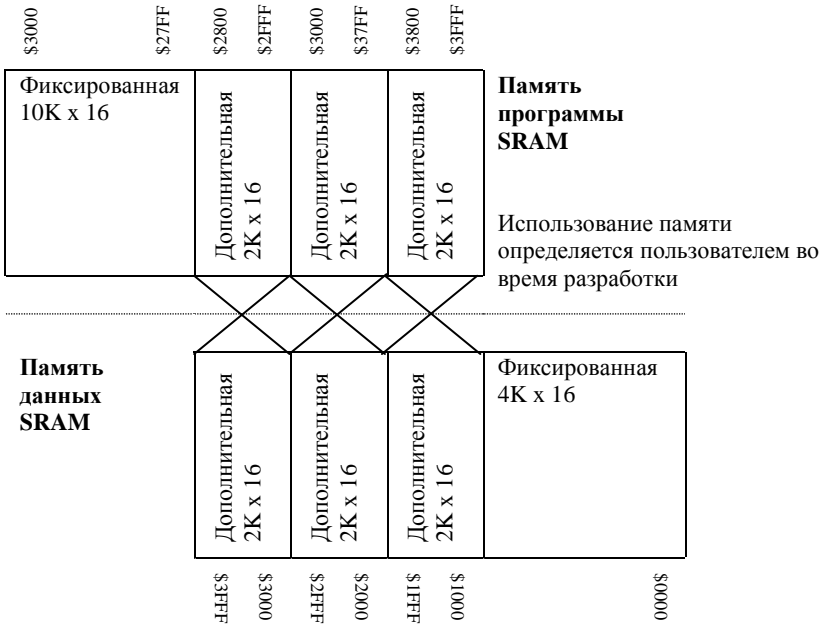


Рис. 5.5. Структура массивов памяти в FPSLIC

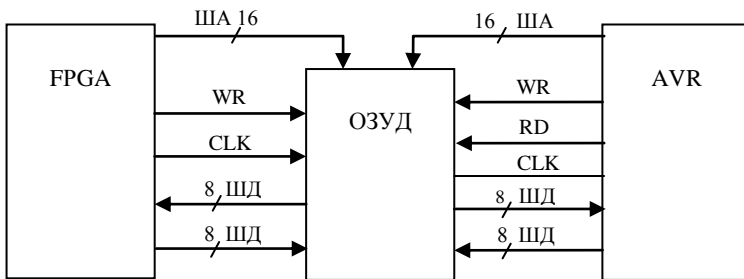


Рис.5.6. Организация двухпортовой памяти

В FPSLIC реализована возможность перепрограммирования FPGA под управлением AVR непосредственно в процессе работы. Структура механизма конфигурации FPGA под управлением AVR приведена на рис. 5.7.

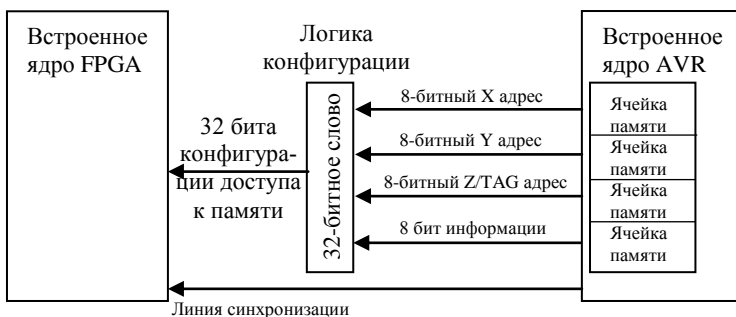


Рис. 5.7. Конфигурация массива FPGA в микросхеме FPSLIC под управлением AVR

Ядро AVR имеет доступ к конфигурационной памяти FPGA по записи. В адресном пространстве микроконтроллера имеются четыре ячейки памяти, которые используются для генерации 4-х байтной записи (24-разрядный адрес и собственно байт данных), необходимой для записи в FPGA. Фактически, в этом режиме FPGA рассматривается микроконтроллером как синхронное оперативное запоминающее устройство. Наличие такого режима позволяет изменять конфигурацию FPGA в системе алгоритмически, по ходу выполнения рабочей программы микроконтроллера. Можно, например, сохранить несколько вариантов конфигурации FPGA во внешней энергонезависимой памяти (например, DataFlash), а затем, по мере необходимости, переписывать их в AVR, переформатировать и загружать в FPGA.

Это позволяет пользователям найти оптимальный компромисс между логикой и памятью, экономить средства и получать дополнительные выгоды от гибкости, многофункциональности и энергопотребления конечных систем.

Упрощенная структура FPSLIC представлена на рис. 5.8.

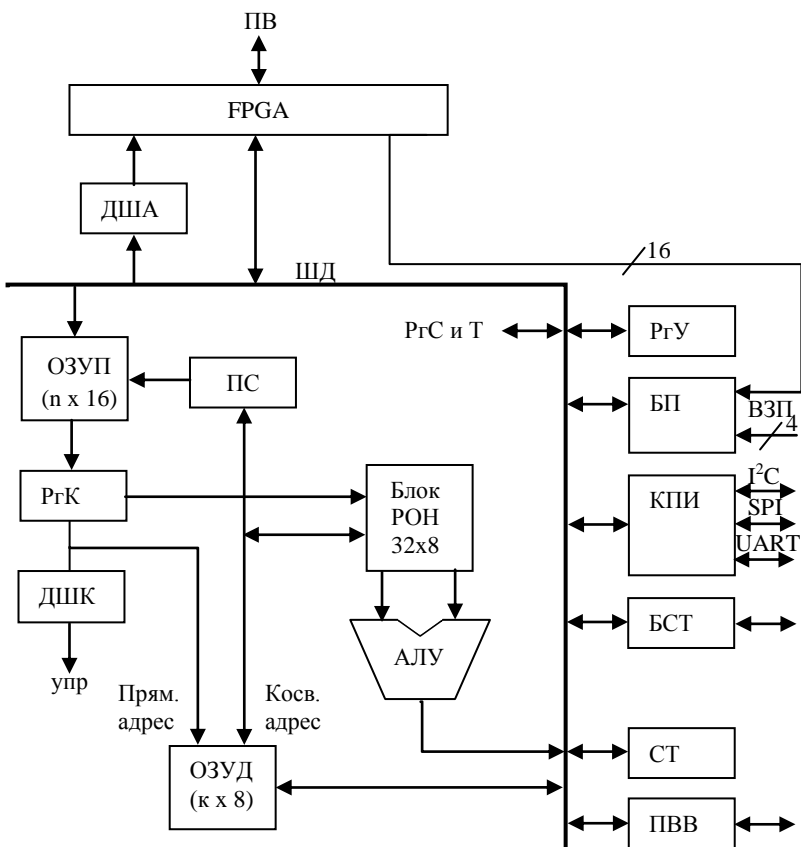


Рис. 5.8. Упрощенная структура FPSLIC

За основу ядра, интегрированного в FPSLIC, взят микроконтроллер Atmega161, имеющий каноническую для AVR гарвардскую архитектуру: блок РОН, 8-разрядное АЛУ, регистр команд PгК, программный счетчик ПС, регистры управления, блок прерываний БП, контроллеры последовательных интерфейсов КПИ, блок счетчиков-таймеров БСТ, сторожевой таймер СТ, порты ввода-вывода ПВВ, регистры состояния и тестирования PгСТ, ОЗУ данных ОЗУД.

Корпорация Atmel выпустила новую конфигурируемую система-на-кристалле CAP7L на базе ядра ARM7TDMI.

Конфигурируемая система-на-кристалле AT91CAP7L200A содержит процессор на базе ядра ARM7TDMI и вентиляющую матрицу объемом 200К логических вентиляей. Вентильная матрица используется для реализации IP-функций, дополнительных процессорных ядер или уникальной периферии, необходимой заказчику. Конфигурирование вентильной матрицы осуществляется методом metal programmable (MP) *в процессе производства микросхемы*. Корпорация Atmel имеет большую библиотеку IP-функций (лицензионных и бесплатных), которые были протестированы в блоках CAP7L. Микросхемы AT91CAP7L имеют низкое энергопотребление. Максимальное значение энергопотребления составляет 4мВт. [76]

Рассмотренные выше ПСК предназначены для ввода-вывода и обработки дискретной (цифровой) информации.

Однако при решении задач автоматизации приходится довольно часто обрабатывать аналоговые сигналы, поступающие с различных датчиков. Типовыми операциями в этом случае являются коммутация, нормализация, фильтрация, аналого-цифровое преобразование, цифровая обработка, вывод информации в аналоговом виде с помощью цифро-аналогового преобразователя. Поэтому в ряде случаев необходимо, чтобы ПСК содержали *программируемые* средства обработки аналоговых сигналов.

Компания Anadigm выпускает программируемые аналоговые интегральные схемы в виде автономных кристаллов (микросхем), но их использование в ПСК требует усложнения печатной платы. [77]

Наиболее привлекательна ПСК, разработанная фирмой Cypress, которая содержит на одном кристалле конфигурируемый массив элементов для обработки аналоговых и цифровых сигналов с интегрированным микроконтроллером. Фирменная аббревиатура этой разработки - PSoC. [78-80].

Упрощенная структура PSoC представлена на рис.5.9

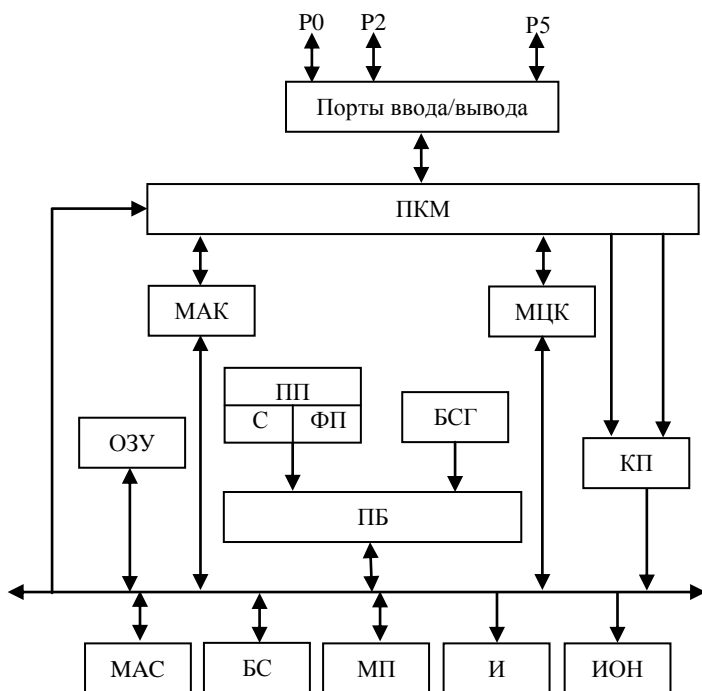


Рис. 5.9. Упрощенная структура PSoC

PSoC представляет собой матрицу цифровых и аналоговых элементов, объединяемых в систему **программируемыми связями**.

В зависимости от типа микросхемы процессорный блок ПБ может быть выполнен на базе фирменного 8-разрядного микропроцессора M8C (PSoC 1), микроконтроллера 8051(PSoC 3) или Cortex M3 (PSoC 5). Если быстродействие PSoC 1 составляет порядка 4 MIPS, то для PSoC 3 эта величина составляет 24 MIPS, а для PSoC 5 - 1,25 DMIPS/МГц при возможности работы до 80 МГц. [80,81]

Матрица цифровых компонентов МЦК содержит основные (basic) и коммуникационные (communication) блоки. Основные блоки включают: таймеры, счетчики, ШИМ, генераторы псевдослучайной последовательности, генератор CRC, генератор зоны не-

чувствительности (для ШИМ). В коммуникационные блоки входят: контроллеры ПДП, интерфейсы SPI (режимы Master/Slave), I²C, USB, CAN, инфракрасный порт, UART.

Матрица аналоговых компонентов МАК может включать: инструментальные усилители с программируемым коэффициентом усиления, компараторы, программируемые фильтры (низких частот, высоких частот, полосовые и т.д.), блок для работы с сенсорной клавиатурой, программируемый блок для реализации цифровых фильтров, аналоговые мультиплексоры, АЦП (накапливающие, сигма - дельта, последовательного приближения), ЦАП. Имеется возможность изменять разрядность АЦП от 8 до 20, ЦАП - от 6 до 9 разрядов.

Комплектация МЦК и МАЦ определяется типом ПМК. Количество цифровых блоков до 16, а аналоговых - до 12. Необходимо учитывать, что некоторые функциональные устройства требуют для своей реализации несколько блоков и определенные ресурсы флэш - памяти и ОЗУ. [79]

Наряду с использованием библиотечных функций пользователю предоставляется возможность создавать собственные функциональные устройства.

Ввод-вывод выполняется через порты P0-P5 с конфигурируемой схемотехникой (до 8 вариантов). Нагрузочная способность портов до 25мА для цифровых линий и 40 мА для аналоговых. Программируемая коммутационная матрица ПМК позволяет управлять функциональностью линий ввода -вывода (см. раздел 5)

Системный блок содержит оперативную память ОЗУ, объем которой в PSoC 1 до 2 КБ, PSoC 3 - до 8 КБ, PSoC 5 - до 64КБ, флэш-память программ ПП (PSoC до 32КБ, PSoC 3 до 64 КБ, PSoC 5 до 256 КБ), блок системных генераторов БСГ, содержащий несколько генераторов с внешними и внутренними частотозадающими элементами, блок синхронизации БС, схему умножения с накоплением МАС, монитор питания МП, интегратор (дециматор) И, источник опорного напряжения ИОН, контроллер прерываний КП, число входов которого зависят от типа микросхемы.

ПП хранит программы супервизора С и пользователя ФП. Дополнительно PSoC 3 и PSoC 5 имеют EEPROM объемом до 2 КБ.

PSoC обладает возможностью изменения конфигурации в процессе работы по внешнему или внутреннему событию. Для переконфигурирования PSoC достаточно изменить значения специальных динамических регистров, определяющих внутреннюю текущую архитектуру микросхемы.

Возможности PSoC можно расширить за счет применения расширителей ввода-вывода CY8C95xx, которые позволяют дополнительно увеличить число линий ввода-вывода до 36, содержат EEPROM до 27 КБ, широтно-импульсные модуляторы. Связь с ПСК осуществляется через интерфейс I2C.

На базе двух технологий PSoC и WirelessUSB, выпускаются микросхемы, объединяющие в одном корпусе микроконтроллер PSoC и приемопередатчик WirelessUSB. Эти микросхемы получили название PProC (Programmable Radio System-on-Chip).

Успешной разработке проектов способствует наличие интегрированных сред разработки PSoC Designer и PSoC Express, которые используются для PSoC 1.

Особенностью PSoC Express является создание законченного приложения на базе PSoC, которое включает в себя конфигурацию периферийных блоков, разводку внутренних коммутационных связей, обработку прерываний и программу процессорного ядра контроллеров PSoC. Разработка приложения на базе PSoC Express не требует от разработчика ни одной строчки кода! Процесс создания приложения с помощью PSoC Express состоит из следующих шагов: выбор входных (датчики, кнопки...) и выходных (двигатели, индикация...) функций, описание состояния выходов в соответствии с задачами, в зависимости от проекта PSoC Express предложит наиболее подходящий для задачи кристалл. После создания проекта PSoC Express позволяет провести симуляцию его работы.

Новая интегрированная среда разработки ***PSoC Designer версии 5.0*** объединила в себе два программных продукта: PSoC Designer 4.4 и PSoC Express 3.0.

Для разработки ПСК на основе PSoC 3, PSoC 5 используют ***PSoC Creator***. [80]

Применение PSoC обеспечивает высокую производительность, компактность, минимальное использование внешних компонентов, минимизирует площадь печатной платы, энергопотребление, время

проектирования, повышает надежность, снижает стоимость готовой продукции.

6. Режимы работы ОМК

Рабочий (активный) режим. Режим, при котором микроконтроллер выполняет программу при постоянно включенных всех функциональных блоках.

Режим пониженного энергопотребления. Энергопотребление микроконтроллера зависит от элементной базы, на которой он выполнен, статической и динамической составляющей тока нагрузки, напряжения питания.

Большинство микроконтроллеров выполняются на основе КМОП-технологии, статическое потребление которых, вызванное постоянными токами утечки, незначительно. Основной составляющей является динамическая нагрузка, определяемая переходными процессами в транзисторах при переключении логических элементов и зарядом-разрядом нагрузочных конденсаторов. Эти процессы происходят по сигналу тактового генератора: чем меньше тактовая частота, тем меньше энергопотребление.

Все методы снижения энергопотребления базируются на управлении этими составляющими: снижение напряжение питания, раздельное питание процессорного блока и периферии, уменьшение частоты тактового генератора, динамическое управление количеством одновременно работающих устройств и их частотой. Однако при выборе методов реализации режимов энергосбережения следует учитывать фактор реального времени, так как большинство из них приводит к снижению быстродействия.

Напряжение питания микроконтроллеров в зависимости от назначения изменяется в широких пределах от 1,8 В до 7В и выше.

В **рабочем режиме** используются: раздельное питание ядра и периферии, автоматическое отключение неработающих периферийных устройств, делители и умножители, позволяющие работу процессорного блока и функциональных блоков на требуемых частотах и так далее. (см. раздел 5)

Наиболее эффективно энергопотребление снижается при использовании специальных «спящих» режимов. При фиксированных значениях напряжения питания и тактовой частоты «спящие» режимы

можно разделить на два основных класса- режимы ожидания и останова (микрopotребления).

Режим ожидания (Idle mode) реализуется практически во все ОМК одинаково: прекращается тактирование процессорного блока и памяти программ, но система тактирования периферийных блоков работают в полном объеме. При этом энергопотребление снижается в 3-5 раз. **Достоинством является быстрая реакция на события, приводящие к переходу ОМК в рабочий режим (выходу ОМК из «спящего» режима)** (3-5 периодов тактовой частоты). Это может быть внутреннее или внешнее прерывания, сброс.

Режим останова (Stop, Sleep, Power Down) отличается в микроконтроллерах большим разнообразием. В этом режиме отключаются **почти все тактовые сигналы, поэтому не работают соответствующие синхронные функциональные блоки. Исключением могут быть сторожевой таймер, АЦП, I2C, USB, внешние прерывания, асинхронные счетчики.** В зависимости от конфигурации имеется несколько модификаций режима останова. Возврат в рабочий режим возможен по сбросу, внешним прерываниям, прерываниям от сторожевого таймера, при обнаружении заданного адреса в последовательных канал (**I2C, USB**) и так далее. [26,37]

В качестве примера рассмотрим режимы пониженного энергопотребления ОМК ATmega фирмы Atmel:

1. Idle (режим ожидания) - режим, в котором прекращается тактирование процессора и памяти программ. При этом вся остальная периферия микроконтроллера (АЦП, таймеры/счетчики, и пр.) продолжает функционировать. Благодаря этому при поступлении прерываний, как внешних, так и внутренних пробуждают микроконтроллер, и он продолжает функционировать в рабочем режиме;

2. ADC Noise Reduction (режим снижения шумов АЦП) - режим, в котором прекращается тактирование процессора, памяти программ, а также системы ввода/вывода. Это необходимо для того, чтобы снизить шумы и повысить точность преобразования АЦП. Если АЦП включен, то преобразование начинается сразу после перехода в данный режим. Выход из данного режима осуществляется по прерыванию от сторожевого таймера, при аппаратном сбросе, сбросе от системы BOD, по внешнему прерыванию, прерыванию от

АЦП, EEPROM и STM, а также по совпадению адреса по интерфейсу I2C (TWD);

3. Power-down (режим микропотребления) - в данном режиме прекращают работу все устройства микроконтроллера кроме тех, которые работают в асинхронном режиме. Выход из данного режима может осуществляться при аппаратном сбросе, сбросе от системы монитора питания, внешние прерывания, прерывания от сторожевого таймера, и прерывание по совпадению адреса в интерфейсе I2C (TWI);

4. Power-save (экономичный режим) - данный режим полностью повторяет режим Power-down, но в нем еще продолжает свою работу таймер/счетчик 2 в асинхронном режиме;

5. Standby (режим ожидания) - полностью повторяет режим Power-down, но микроконтроллер пробуждается всего за 6 машинных циклов. Данный режим работает только с внешним резонатором;

6. Extended Standby (расширенный режим ожидания) - полностью повторяет режим Power-save, но микроконтроллер пробуждается всего за 6 машинных циклов. Данный режим работает только с внешним резонатором. [82]

Как меняется энергопотребление в зависимости от выбранного режима работы наглядно продемонстрировано в таблице ниже (при 1 MHz, 3V, 25°C):

Таблица 4. Энергопотребление в зависимости от режима работы

Тип микроконтроллера	Активный режим	Режим ожидания	Режим останова
Attiny2313	2.5 mA	0.5 mA	0.5 μ A
Attiny2313a	190 μ A	24 μ A	0.1 μ A
Atmega48	250 μ A	15 μ A	0.1 μ A
Atmega8	3.6 mA	1.0 mA	0.5 μ A
Atmega8a	250 μ A	50 μ A	0.1 μ A
Atmega32a	0.6 mA	0.2 mA	<1 μ A

Вход в режим пониженного энергопотребления выполняется по специальным командам (для ATmega это Sleep), режим задается в регистре управления, а некоторые параметры в конфигурационных ячейках на стадии программирования. Выход - в зависимости от

особенностей выбранного режима, затем выполняется команда, следующая за Sleep.

Основными характеристиками режимов энергосбережения являются:

- степень снижения энергопотребления;
- требуемые сигналы для возврата в рабочий режим;
- время возврата в рабочий режим;
- количество активных устройств.

Режим сброса. Режим сброса предназначен для принудительной установки стартового адреса программы и основных системных регистров в фиксированное состояние при возникновении определенных условий. Переход в этот режим может быть вызван:

- включением напряжения питания;
- подачей внешнего сигнала определенного вида на вход RES (Reset);
- снижением напряжения питания ниже допустимого значения;
- переполнением сторожевого таймера;
- сбросом по интерфейсу отладки и программирования (например, JTAG).

При *появлении (наступлении)* одного из перечисленных событий управляющие регистры устанавливаются в исходные значения, а в счетчик адреса команд записывается адрес вектора сброса. Значение адреса может быть фиксированным (8051/52, Atmel - 0, Motorola - старшие адреса) или задаваться состоянием конфигурационных ячеек. По адресу вектора располагается информация о стартовом адресе программы. После адреса вектора обычно располагается зона векторов прерываний.

В большинстве ОМК установлен формирователь аппаратного сброса Power-On. Если напряжение питания стабильно во времени и подаётся резким скачком, то внешние элементы для сброса теоретически не нужны. Обнуление производится автоматически узлом Power-On по достижению определённого порога. Длительность внутреннего импульса сброса может определяться содержимым конфигурационной ячейки. При более медленной скорости нарастания напряжения питания рекомендуется установка внешних RC-цепочек. В контроллерах параллельно конденсатору присоединяют

кнопку, которая позволяет формировать внешний импульс сброса при «зависании» или перезапуске ОМК. Возможен сброс от внешнего источника. [83]

Блок монитора питания BOD (Brown-Out Detection) предназначен для контроля текущего значения напряжения питания. Монитор питания измеряет напряжение Vcc и вырабатывает сигнал сброса ОМК при снижении Vcc ниже допустимого значения. Микроконтроллер удерживается в состоянии сброса до тех пор, пока не восстановится номинальное напряжение. Монитор должен иметь зону нечувствительности (гистерезис), чтобы предотвратить формирование сброса при небольших по напряжению импульсных помехах. *Это повышает помехоустойчивость при питании от полупроводниковых батарей/аккумуляторов, а также при мощных электромагнитных наводках.* С помощью ячеек конфигурации BOD можно включить, выключить, задать напряжение срабатывания. [37]

Сброс по переполнению сторожевого таймера происходит, если процессорный блок не сформирует вовремя соответствующую команду. Длительность состояния сброса определяется особенностями ОМК.

Сброс через интерфейс отладки и программирования характерен для ОМК, в которых присутствует соответствующий интерфейс (JTAG, C2, debugWire и так далее). Их особенности будут изложены ниже.

Режим программирования. Программирование ОМК заключается в записи в резидентную (внутреннюю) память микроконтроллера необходимой информации. Резидентная память состоит из памяти данных, памяти программ и памяти, определяющей режимы работы микроконтроллера и его блоков. [26,37]

В процессе программирования могут быть выполнены следующие операции:

- чтение ячеек идентификатора;
- чтение калибровочных ячеек;
- чтение/запись конфигурационных ячеек;
- общее стирание памяти;
- чтение/запись памяти программ;
- чтение/запись памяти данных (EEPROM);
- чтение/запись ячеек защиты.

Ячейки **идентификаторов** содержат информацию о типе микроконтроллера, коде производителя, объеме и типе памяти программ и так далее. Они записываются в процессе изготовления ОМК и располагаются в автономном адресном пространстве. Содержимое идентификаторов можно **только прочитать в режиме программирования**.

В **калибровочные ячейки** при изготовлении ОМК записываются константы, необходимые для реализации функций подстройки (или автокоррекции) частоты внутренних генераторов и аналоговых блоков (АЦП, ЦАП, датчика температуры). Количество таких ячеек зависит от количества диапазонов частот генератора и организации аналоговых блоков. В зависимости от особенностей ОМК содержимое калибровочных ячеек **после чтения** может быть **переписано при программировании** во флэш-память или **аппаратно** - в специальные регистры в режиме сброса.

Конфигурационные ячейки (Fuse Bits) предназначены для настройки определенных системных функций ОМК. Они расположены в отдельном адресном пространстве, доступном только при программировании. Количество и содержание Fuse-бит зависит от конкретной модели микроконтроллера. С их помощью можно задать: режим работы тактового генератора, длительность состояния сброса, режим работы сторожевого таймера, включение/выключение монитора питания и значение его уровня срабатывания, разрешение/запрет программирования по SPI, JTAG, внутрисхемную отладку, определить размер секции загрузчика и положение вектора сброса (начальный адрес прикладной программы или загрузчика) и так далее. Для изменения содержимого конфигурационных ячеек используются специальные команды программирования.

Рабочая программа микроконтроллера не может проверить или изменить содержимое ячеек конфигурации. Поэтому к процессу конфигурирования следует относиться ответственно. **По умолчанию** Fuse-биты устанавливаются в определенное положение, которое индивидуально для каждого ОМК, и его следует контролировать.

Стирание кристалла должно выполняться перед каждым перепрограммированием микроконтроллера. Эта команда стирает содержимое флэш-памяти, EEPROM и ячейки защиты, записывая в них «1». На состояние конфигурационных ячеек команда не влияет.

В некоторых ОМК можно запретить стирание EEPROM соответствующим программированием конфигурационных ячеек. Часть ОМК не имеют отдельного режима «стирание». Для них стирание прежней информации в памяти происходит в «теневом» режиме, при каждом новом цикле программирования микроконтроллер.

Алгоритм *записи памяти программ и данных* определяется особенностями ОМК и метода программирования. Запись выполняется побайтно или постранично. Размер страницы может быть жестко задан или устанавливаться в конфигурационных ячейках. В исходном состоянии содержимое памяти равно единице, а при программировании записываются нули.

Содержимое Flash-памяти, её секторов и EEPROM может быть защищено от записи и/или чтения программированием *ячеек защиты (Lock bits)*. Lock-биты можно записать с помощью внешнего программатора или из прикладной программы, однако только в направлении *усиления* уровня защиты. Стереть lock-биты можно только командой «стирание кристалла» одновременно с flash-памятью. Поэтому *установка битов защиты выполняется в конце цикла программирования*, то есть после программирования памяти и конфигурационных ячеек.

Активизация режима программирования индивидуальна для микроконтроллеров и зависит от многих факторов: последовательное программирование при низком или высоком напряжении, параллельное программирование, программирование по интерфейсу JTAG и его модификациям, внешний программатор, внутрисхемное программирование (ISP mode). [84]

Режим внутрисхемной отладки и программирования обеспечивает отладку микропроцессорной системы с минимальными внешними аппаратными средствами.

В этом режиме отладка программы осуществляется в ОМК, установленном в плату разрабатываемой системы, и выполняется с учетом *реальных характеристик* периферийных устройств и внешних сигналов. Реализация режима требует:

- наличия *резидентных* аппаратно-программных средств, включающих контроллер отладки, набор регистров контрольных точек для доступа к прикладной программе или данным и управляющие подпрограммы (монитор), записанные в ПЗУ и недоступные

из программ пользователя. *Эти ресурсы формируются в ОМК на стадии изготовления;*

- модуля связи с ПК, осуществляющего взаимодействие последовательных портов ПК и ОМК;
- инструментальные средства моделирования, размещенные в ПК, с помощью которых выполняется разработка программного обеспечения, его отладка и программирование.

Ввод микроконтроллера в режим отладки может осуществляться программно или аппаратно (формированием на его выводах определенной комбинации сигналов).

В режиме отладки микроконтроллер работает под управлением команд, посылаемых ПК, через специальный интерфейс. Эти команды позволяют контролировать и модифицировать текущее состояние микроконтроллера, производить стирание и запись содержимого внутренней Flash-памяти и ЭПЗУ, запускать программы с точками останова и передавать в ПК состояние регистров, ячеек памяти. Данный режим обеспечивает выполнение основных операций, реализуемых в процессе отладки программ, без использования дорогостоящих схемных эмуляторов.

Внутрисхемный отладчик может работать в фоновом или отладочном режимах.

Фоновый режим используется для управления отладчиком и чтения содержимого резидентной памяти без остановки выполнения рабочей программы или с незначительной потерей быстродействия.

Режим отладки позволяет отладчику управлять работой процессорного блока, осуществлять чтения/запись внутренних регистров и памяти, контролировать выполнение прикладной программы в точках останова.

На ранних этапах развития микропроцессорной техники каждая фирма использовала собственные разработки внутрисхемных отладчиков. В настоящее время наиболее часто используется унифицированный внутрисхемный отладчик JTAG на основе стандарта IEEE 1149.1-2001 Test Access Port and Boundary-Scan Architecture (Порт тестового доступа и Архитектура Граничного сканирования). *Особенностью JTAG является возможность контроля и программирования не только микроконтроллера (или ПЛИС), но и подключенной к его выводам периферии.* [85,86]

В основу стандарта положена идея внедрение в компоненты цифрового устройства средств, обеспечивающих унифицированный подход к решению следующих задач:

- диагностика печатной платы с целью выявления технологических дефектов (отсутствие связей, короткое замыкание, тестирование межсоединений);
- конфигурация программируемых устройств (микроконтроллеры, ПЛИС);
- конфигурация и прошивка микросхем памяти (FLASH, EEPROM, FRAM);
- реализация режимов внутрисхемной отладки и мониторинга, OCD (On Chip Debug);
- решение задачи функциональной верификации (для ASIC или FPGA);
- импорт и анализ Boundary Scan Description Language (BSDL) файлов для микросхем различных производителей.

Тестирование по стандарту IEEE1149.1 позволяет выявить следующие дефекты:

- правильность монтажа резисторов Pull-up/Pull-down
- целостность монтажа выводов устройства, поддерживающего IEEE1149.1
- целостность цепей
- тестирование шин адреса, данных, управления у различных типов памяти SDRAM (SDR, DDR, DDR2, DDR3), микропроцессоров, микроконтроллеров, систем на кристалле, что актуально для запуска функциональных тестов
- тестирование ID, шин данных, адреса, управления у различных типов FLASH-памяти
- интерактивное тестирование с участием человека (светодиоды, кнопки и т.д.)
- тестирование цифровых интерфейсов PCI, SMI, I2C, TDM, SPI, UART (только для устройств с поддержкой эхоконтроля (loopback)), SDIO и т.д. [87]

Упрощенная логическая структура JTAG представлена на рис.6.1.

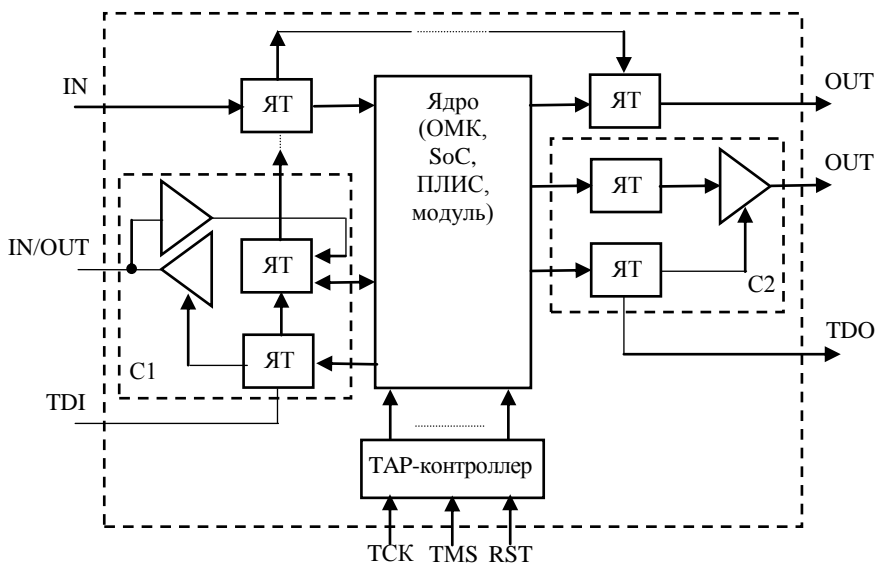


Рис.6.1. Логическая структура JTAG

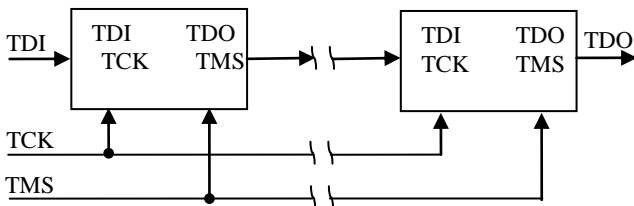


Рис.6.2. Соединение устройств JTAG

Ячейки тестирования ЯТ располагаются между ядром устройства и внешними выводами. ЯТ представляю собой структуру, состоящую из двух D-триггеров, на входе которой стоит мультиплексор, а на выходе - демultipлексор. Совокупность ТЯ образуют сдвиговый регистр сканирования границ (Boundary-Scan) РСГ, управляемый TAP- контроллером. Кроме этого регистра в состав TAP- контроллера входят регистр команд (инструкций), регистр данных, идентификационный регистр. На вход ЯТ поступают данные и команды TDI, управляющие сигналы, определяющие режим работы TMS, импульсы синхронизации TCK (последние два сигнала на

рис.6.1 не показаны), результаты тестирования получают с выхода TDO. Контроллер TAP представляет собой синхронный конечный автомат с 16 состояниями, изменяющий их по фронту TCK в соответствии с сигналом TMS. Эти состояния реализуют команды, определяемые стандартом JTAG, и команды, необходимые производителю устройства для выполнения определенных методик тестирования.

Количество ЯТ определяется особенностью ядра. На рис 6.1 показаны конфигурации однонаправленных линий ввода (IN), вывода (OUT), выводов с тремя состояниями (C2), двунаправленных выводов с тремя состояниями (C1). [59] Например, для микропроцессора MC68040 регистр тестовых данных содержит 184 разряда при 110 линиях ввода-вывода, а для MC68060 - 214 разрядов [26].

При наличии нескольких устройств, поддерживающих стандарт JTAG, они могут объединяться в цепочку (рис. 6.2). Стандарт не вводит никаких ограничений на количество устройств в цепочке. В процессе обмена вся цепочка представляет собой общий сдвиговой регистр, в котором каждому устройству будет соответствовать определенное количество разрядов. Ведущее устройство, разумеется, должно "знать" распределения разрядов между устройствами.

Порты JTAG входят в состав микроконтроллеров ARM7, ARM9, Cortex, MAXQ, ATmega, PIC, MCS -51/52 и так далее.

К недостаткам JTAG относятся низкая скорость, ограниченные возможности по функциональному тестированию, относительно большое количество линий, что ограничивает его применение в малогабаритных микроконтроллерах.

Эти недостатки устраняются в последующих совместимых версиях JTAG-стандартов. Например, стандарт 1149.7 (*compact JTAG*) уменьшает число обязательных контактов *до двух*, расширяет функциональность тестов, позволяет использовать звездообразную топологию, поддерживает работу с многоядерными микросхемами и так далее. [88]

Известны и другие модификации тестирующих портов однопроводный debugWire фирмы Atmel, двухпроводный C2 фирмы SiLabs и ряд других. [89,12]

Как правило, внутрисхемные отладчики могут работать в режиме программирования.

Многие микроконтроллеры обладают свойством *самопрограммирования*, то есть могут изменять содержимое флеш-памяти, не используя программатор. Это позволяет создавать системы, алгоритм работы которых будет изменяться в зависимости от различных внутренних или внешних событий. [90]

Для этого необходимо создать специальную программу загрузки *bootloader* и записать её при *обычном* программировании в зону загрузчика. Флеш-память разделяется на две зоны: секция прикладной программы и секция программы загрузчика. Программа, размещенная в зоне загрузчика, позволяет модифицировать прикладную программу с помощью команд записи в память (SPM для AVR, MOVX для SiLabs и так далее). Переход к программе загрузчика может быть по командам передачи управления или перемещением вектора сброса в начало секции загрузчика. Размер секций и положение сброса определяется конфигурационными ячейками.

Бутлоадер можно разработать самому или воспользоваться готовыми программами, свободно распространяемыми в Интернете. [91] Для передачи кода программы можно использовать интерфейсы, входящие в состав ОМК (SPI, I2C, RS-232, USB и так далее).

7. Структурная организация 32-разрядных микроконтроллеров

Последние годы развития микропроцессорной техники привели к широкому применению 32-разрядных микроконтроллеров. Это обусловлено снижением их стоимости и энергопотребления, высокой производительностью, большим набором периферийных модулей, развитыми и доступными средствами проектирования.

В 32-разрядных микроконтроллерах ОМК-32 используются практически все методы повышения быстродействия и снижения энергопотребления, изложенные в разделе 4.

ОМК-32 можно разделить на микроконтроллеры с фирменной архитектурой процессорного блока и архитектурой, использующие широко известные программные ядра (IP-ядра), поставляемые в электронном виде.

К первой группе можно отнести ряд микроконтроллеров фирмы Motorola, в которых используются микропроцессоры семейства

MC68xxx, фирмы AMD - с микропроцессорами x86, фирмы Atmel с ядром AVR32. [92,93]

Большее распространение получили микроконтроллеры второй группы с ядрами **ARM и MIPS**.

Особенности архитектуры ОМК-32 рассмотрим на примере микроконтроллеров с ядром ARM. [94,95,54]

Фирма ARM *не занимается производством* микросхем, а является *разработчиком моделей* процессорных ядер, сопроцессоров, периферийных устройств. Основной статьей дохода компании стали не сами процессоры, а лицензии на их архитектуру, а также разработка сопутствующего процессу создания микропроцессоров инструментария: компиляторов, систем отладки кода, тестовых стендов.

Фирма поставляет свои разработки в виде документации, в состав которой входят: описание алгоритма на языке VHDL или Verilog; электрические схемы; описания топологии и так далее. На основе этой документации *производители электронных компонентов* создают различные семейства микроконтроллеров, *дополняя их необходимым набором собственных периферийных устройств*.

Клиентами ARM являются свыше 60 производителей микросхем, среди которых можно выделить такие известные компании, как Altera, Analog Devices, Atmel, Cirrus Logic, Fujitsu, MagnaChip (Hynix), Intel, Freescale (Motorola), National Semiconductor, NXP (Philips), OKI, ST Microelectronics и Texas Instruments. В настоящее время архитектура ARM занимает лидирующие позиции и охватывает 75% рынка 32-разрядных встраиваемых RISC микропроцессоров. [95]

Архитектуру ARM представляют два класса: «классические» процессоры, куда входят семейства ARM7, ARM9 и ARM11, и семейство **Cortex**.

ОМК с ядром ARM реализуют RISC-архитектуру с 32-разрядным процессорным блоком. Процессор ARM имеет несколько наборов (банков) 32-разрядных регистров, из которых в *конкретный момент времени* доступны программисту только 16. К ним относятся регистры общего назначения R12 - R0, указатель стека

R13, регистр возврата из подпрограммы R14, в котором сохраняется текущее содержимое PC для возвращения к основной программе. Регистр R15 является счетчиком адреса команды Регистр R15 (PC) может быть использован в обычных арифметических или логических операциях в качестве регистра приемника. Поэтому команда **add pc, pc, #8** является инструкцией для перехода на другой адрес.

Существует несколько режимов работы процессора, в зависимости от которого выбирается *соответствующий банк регистров*. Основные режимы работы: режим приложения (**USR, user mode**), режим супервизора или режим операционной системы (**SVC, supervisor mode**), режим обработки прерывания (**IRQ, interrupt mode**) и режим обработки «срочного прерывания» (**FIRQ, fast interrupt mode**). Процессоры ARM более старших версий кроме вышеперечисленных режимов работы имеют еще дополнительные режимы: **Abort** (используется для обработки исключений доступа к памяти), **Undefined** (используется для реализации сопроцессора программным способом) и режим привелигированных задач операционной системы **System**. Например, при возникновении прерывания процессор автоматически переходит к адресу программы обработчика прерываний и «переключает» банки регистров.

Основные форматы команд приведены на рис 7.1 Их особенностью является возможность выполнения трехадресных команд, повышенная функциональность, расширенное число способов адресации.

Type	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data Processing REGOP	Cond				0	0	1	Opcode				S	Rn			Rd			shifter_operand													
Multiply MULT	Cond				0	0	0	0	0	0	A	S	Rd			Rn			Rs			1	0	0	1	Rm						
Single Data Swap SWAP	Cond				0	0	0	1	0	B	0	0	Rn			Rd			0	0	0	0	1	0	0	1	Rm					
Single Data Transfer TRANS	Cond				0	1	1	P	U	B	W	L	Rn			Rd			Offset													
Block Data Transfer MTRANS	Cond				1	0	0	P	U	S	W	L	Rn			Register List																
Branch BRANCH	Cond				1	0	1	L	Offset																							
Coprocessor Data Transfer CODTRANS	Cond				1	1	0	P	U	N	W	L	Rn			CRd			CP#			Offset										
Coprocessor Data Operation COREGOP	Cond				1	1	1	0	CP Opcode				CRn			CRd			CP#			CP	0	CRm								
Coprocessor Register Transfer CORTTRANS	Cond				1	1	1	0	CP Opcode			L	CRn			Rd			CP#			CP	1	CRm								
Software Interrupt SWI	Cond				1	1	1	1	Ignored by processor																							

Рис 7.1. Форматы команд ARM

Кодирование команд довольно сложное. Подробное их описание приведено в соответствующей литературе. [94,97]. В обобщенном виде формат команды состоит из:

- **поля условий Cond** (31-28), в котором кодируются признаки результата выполнения операций N, Z, C, OV. На основе этих флагов можно сформировать различные комбинации условий (переходы по значению или их отрицанию, \geq , \leq и так далее). Это поле используется для организации условных переходов.

- Старшие четыре бита текущей команды сравниваются с признаками результата предыдущей. Если признаки совпали, то текущая команда выполняется, если нет, то выполняется команда NOP.

- **поле кода операций Opcode** (24 -21),

- **поле модификации кода** (27-25, 20) кодирует особенности выполнения команд: формировать признаки операции или нет, типы выполняемых операций, способы адресации, вид и особенности использования второго операнда (константа, адрес регистра и/или его смещение, использование операции сдвига,), длину слова и так далее.

- **поле адресов** регистров приемника Rd, первого операнда Rn, второго операнда Rm

- поле второго операнда **shifter_operand**. Любая операция выполняется над содержимым регистра Rn и shifter_operand. Результат операции помещается в регистр Rd. В зависимости от бита I_{25} **shifter_operand** трактуется как числовая константа, индекс второго регистра операнда, операция сдвига над значением второго операнда. Двенадцать бит **shifter_operand** разбиты на две части: четырехбитный показатель вращения **encode_imm** и собственно восьмибитное числовое значение **imm_8**.

Особенностями системы команд ARM по сравнению с ранее рассмотренными являются:

Наличие регистровой адресации с масштабированием. При масштабировании содержимое регистра Rm сдвигается на число разрядов указанное в команде (от 1 до 31). Реализуются логический сдвиг влево/вправо, арифметический сдвиг вправо; циклический сдвиг вправо.

Масштабирование применяется во многих операциях и, в частности, в регистровом, автоинкрементном, автодекрементном способах адресации.

Это позволяет за один такт выполнить операцию $Rd=Rn*Rm$ (*тип сдвига, n*), где * - код операции, а n- количество сдвигов. Например, $R0=R1 + (R2>>5)$ -сложить $R1$ и сдвинутое вправо на 5 разрядов (деление на 32) содержимое $R2$.

Сдвиг может осуществляться *не только на фиксированное число, записанное в формате команды, но и по содержимому третьего регистра*. Например, команда `add r0, r1, r1, lsr r3` выполнит сдвиг влево, количество разрядов которого указано в $r3$ $r0 = r1 + (r1>>r3)$.

Команда обратного вычитания позволяет изменить порядок записи операндов в команде вычитания. Классическая запись $Rd=Rn- Rm$, а обратная - $Rd=Rm-Rn$.

Команда умножения выполняется над содержимым R_s (регистр множителя) и R_m $Rd = Rm*R_s$, а регистр R_n игнорируется и должен быть равен нулю с целью совместимости с будущим расширением системы команд.

Команда умножения с накоплением $Rd = Rm*R_s + R_n$. Эта операция является базовой в цифровой обработке сигналов. При $R_m=1$ или $R_s=1$ команда по своим действиям эквивалентна выполнению команды сложения.

Наличие **команд блочной передачи** - загрузка содержимого нескольких последовательных слов из памяти в несколько регистров или наоборот (Register List - список регистров).

Специальная группа команд обеспечивает совместную *работу процессора с сопроцессорами*, размещенным на одном кристалле с процессором. К процессорному ядру ARM может быть подключено до 16 сопроцессоров, каждый из которых может иметь до 16 собственных регистров CR_n . Данные, передаваемые сопроцессору по внутренней шине, включают в себя номер сопроцессора, номера регистров сопроцессора, код операции и поле дополнительной информации (рис.7.1). От сопроцессора требуется возможность принимать команду, исполнять её и выдавать на шину результат. При этом один из регистров сопроцессора CR_n должен выполнять функцию регистра команд — воспринимать записанные в него данные как код

операции (команду сопроцессора). После ввода в этот регистр кода операции процессор должен послать команду CDP, которая будет выполняться сопроцессором.

Таким образом, при выполнении арифметических и логических команд можно *одновременно выполнить требуемую операцию, условный переход и масштабирование второго операнда, что соответствует процедуре if-the-else.*

Первоначальная система команд ARM содержала только 32-разрядные инструкции. Однако для систем, критичных к стоимости и объёму памяти программ, к которым относятся практически все встраиваемые системы реального времени, большая длина инструкций ограничивала распространение новой архитектуры. Поэтому система команд ARM дополнилась новым набором 16-разрядных инструкций, получившим название **Thumb**, что в вольном переводе означает «мальчик-с-пальчик».

Из набора команд ARM были отобраны наиболее употребляемые 36 команд. Каждая 32-разрядная инструкция в коде программы, выполняемой на процессоре ARM, предварительно сжимается компилятором и на вход процессора поступает набор 16-разрядных Thumb-инструкций. В процессе выполнения программы Thumb-инструкции декомпрессируются специальным модулем процессора, разворачиваясь в полноценные ARM-команды, которые и обрабатываются ядром. При этом программа может содержать как сжатые Thumb-инструкции, так и обычные команды, что позволяет разработчику программы принимать решение, какие из программных модулей сжимать, а какие оставить развернутыми, соблюдая тем самым компромисс между производительностью и плотностью упаковки кода. Таким образом, стандартное 32 разрядное ядро получило возможность работы с 16 битовыми командами.

По сравнению с базовой системой команд Thumb позволяет работать напрямую лишь с восьмью младшими РОН; а к старшим РОН (включая SP, LR, PC) применимы лишь несколько операций. Недоступны команды: умножение с накоплением/расширением, сопроцессорные операции, обмен SWAP, условное исполнение, сдвиг 2-го операнда в командах общего назначения, трехадресные операции (кроме ADD и SUB), автоинкрементный и автодекрементный режимы адресации и так далее. Команды Thumb существенно сокращают

необходимые объёмы памяти программ (порядка 30 %) и, кроме того, позволяет использовать более дешёвую 16-разрядную память.

Дальнейшим развитием этой идеологии стали команды **Thumb-2**. **Набор команд Thumb-2** является смешанным: в него входят как 16-, так и 32-разрядные команды. Такая комбинация является очень удачной с точки зрения нахождения оптимального соотношения плотности кода и производительности. Команды Thumb-2 дают улучшение плотности кода на 26% по сравнению с 32-битными инструкциями ARM и производительности на 25% по сравнению с 16-битными инструкциями Thumb. [97]

С целью расширения области применения ARM разработан целый ряд проблемно-ориентированных приложений.

Ускорение обработки мультимедийных приложений обеспечивают **технологии SIMD (Single instruction, multiple data — одиночный поток команд, множественный поток данных)** и сопроцессорное расширение **VFP (Vector Floating Point)** для данных с плавающей запятой.

Дальнейшее развитие SIMD и VFP получили в **технологии NEON**, обеспечивающей повышенную производительность при обработке видеоизображений, компьютерных играх, распознавание речи и так далее.

Для обработки сигналов в режиме реального времени разработан ряд ARM, в состав команд которых включены **инструкции, характерные для цифровых сигнальных процессоров**.

Для приложений с поддержкой языка программирования **Java** разработана **технология Jazelle**. Она предлагает уникальное сочетание высокой производительности, малой системной стоимости и невысоких запросов к энергопотреблению, что не может быть достигнуто одновременно, если использовать сопроцессор или специализированный Java процессор. Технология **Jazelle** позволяет ARM процессору выполнять **Java код на аппаратном уровне**. Аппаратно реализовано свыше 60% команд Java-байткода. Технология Jazelle Java обеспечивает аппаратное ускорение приложений, написанных в средах Java, .Net, MSIL, Python и Perl. Ускорение может быть **восьмикратным**, при этом технология Jazelle Java позволяет снижать энергопотребление при исполнении соответствующего кода на 80%. Разработчики могут реализовывать Java приложения, в том числе

операционные системы и прикладной код, на одном процессоре. [97, 98]

Технология безопасности TrustZone (ARM Security Extensions) используется в различных приложениях, включая управление цифровыми подписями и электронными платежами. TrustZone работает на уровне процессорного ядра. Она защищает данные, находящиеся в кэше и во внешней памяти, а также поддерживает работу с периферийными устройствами. В основе технологии разделение всех данных на общедоступные и закрытые. Закрытые данные будут помечаться специальным S-битом, и обрабатываться отдельно от обычных. Пользователь не сможет получить доступ к настройкам TrustZone. Например, производители мобильных устройств получают возможность внедрить в них цифровую систему защиты авторских прав (DRM), собирать информацию об установленных на устройстве программах и даже вести запись пользовательской активности. Она позволяет одному процессорному ядру исполнять код в двух контекстах, обычном и защищённом, и переключаться между ними, используя специальный режим процессора Monitor. [99]

Приведенные технологии расположены не в хронологической последовательности их появления, а как пример проблемно-ориентированных приложений, расширение которых происходит постоянно.

Особенности структурной организации ARM определяют три основных компонента:

- процессорное ядро,
- **набор периферийных компонентов PrimeCell**, созданный на основе перепрограммируемых вентилях. В составе PrimeCell реализованы контроллеры основных последовательных интерфейсов (UART, USB, CANbus, Ethernet, SPI, I2C), оперативной памяти и LCD-мониторов, интерфейсы для подключения смарт-карт и так далее;

- **фирменная магистраль AMBA (Advanced Microcontroller Bus Architecture)**, объединяющая периферию с ядром. AMBA делится на системную магистраль (**AHB**) и магистраль периферийных устройств (**APB**). Задачей **AHB** является обеспечение высокоскоростного обмена между ядром ARM и быстродействующими контроллерами, а **APB** предназначена для взаимодействия с более мед-

ленной периферией, что позволяет работать на меньшей частоте и приводит к снижению энергопотребления.

В качестве примера рассмотрим структурную организацию микроконтроллера LPC 2300/2400 с процессорным ядром ARM7TDMI фирмы NXP Semiconductors.

Особенностью ARM7TDMI является:

- 32-разрядное RISC-ядро с целочисленной арифметикой, трёхступенчатым конвейером и архитектурой фон Неймана;
- наборы команд ARM и Thumb;
- стандарт JTAG и модуль EmbeddedICE для внутрисхемной отладки программ;
- операции умножения с 64-разрядным результатом;
- частота - до 100 MHz;
- потребление - 0,25 МВт/МГц;
- технология 0,18 мкм;
- размер кристалла, необходимый для размещения ядра, - 0,53 мм².

Упрощенная структура микроконтроллера приведена на рис.7.2

В микроконтроллере используется иерархическая организация магистралей, каждая из которых предназначена для работы с определенным классом устройств. Наиболее быстродействующие устройства, имеющие контроллеры прямого доступа к памяти, подключены к магистрали АНВ. С целью уменьшения конфликтов АНВ выполнена разделенной на две магистрали: к одной подключены контроллер Ethernet и СОЗУ, а к второй - контроллер USB и **векторный контроллер прерываний VIC**.

Блок VIC обеспечивает приоритетную обработку 32 запросов прерываний, хранит и передает в ядро ARM7 вектор прерывания, позволяет программно изменять приоритет каждой линии и тип прерываний (быстрое, векторное, автовекторное), а также служит для организации программных прерываний. Его применение существенно сокращает время реакции на запрос.

Остальные периферийные модули подключены к APB, которая связана с АНВ мостом. В состав моста АНВ-APB входит **делитель частоты, позволяющий уменьшать частоту работы APB по сравнению с ядром ARM7 и АНВ, снижая общее энергопотребление.**

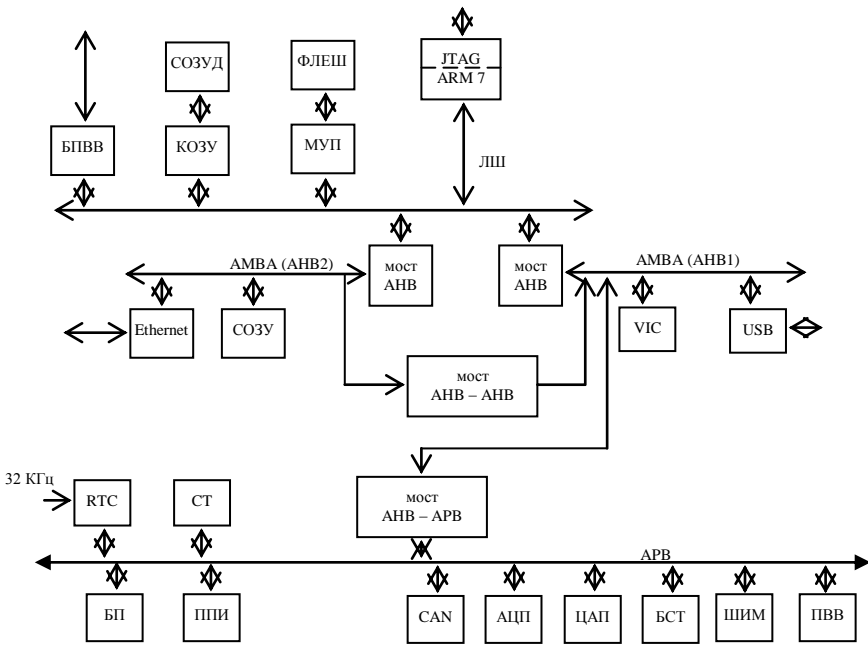


Рис.7.2 - Обобщенная структура ОМК с ядром ARM7

В состав модулей могут входить часы реального времени RTC, сторожевой таймер СТ, блок внешних прерываний БП, подсистема последовательных интерфейсов общего применения ППИ (UART, I2C, SPI, I2S), контроллер промышленной сети CAN, блок счетчиков-таймеров БСТ, широтно-импульсные модуляторы ШИМ, порты ввода-вывода ПВВ, ФЦП, ЦАП, контроллеры смарт-карт, ЖКИ и так далее. Часть модулей входит в набор периферийных компонентов PrimeCell.

Наиболее быстродействующие компоненты ОМК присоединены к дополнительной локальной шине. Её использование повышает производительность процессорного ядра, так как доступ к СОЗУ данных, флеш памяти и скоростным портам ввода-вывода БПВВ будет выполняться быстрее, чем через общую магистраль АНВ.

Несмотря на наличие нескольких внутренних магистралей и блоков памяти в ОМК используется общее линейное адресное пространство (принцип фон Неймана). Программа записывается в рези-

дентную флеш-память, для которой выделен определенный диапазон адресов. Одним из недостатков флеш является низкое быстродействие по сравнению с ОЗУ и процессорным ядром. Методы устранения этого недостатка были рассмотрены выше. В данном микроконтроллере для повышения быстродействия используется 128-битная организация памяти и модуль ускорения работы памяти МУП. МУП содержит быстродействующее локальное буферное ОЗУ, в которое записываются четыре 32-разрядные команды или восемь 16-разрядных команд Thumb, средства для сбора статистики и предсказания перехода. Команды помещаются в локальную память, а затем обрабатываются. МУП обладает недостатком аналогичным кэш-памяти - неопределенность времени обработки программы. Поэтому в системах реального времени имеется возможность его отключения.

Дальнейшее развитие семейства получило в ядрах ARM9, ARM11, выполненных соответственно по архитектуре *ARMv.5*, *ARMv.6*. В отличие от ARM7 *семейство ARM9* стало использовать гарвардскую архитектуру с разделёнными кэшами команд и данных; пятиступенчатый оптимизированный конвейер, усовершенствованную систему команд, включая инструкции эффективной реализации алгоритмов цифровой обработки сигналов; контроллер памяти с поддержкой виртуальной памяти (MMU) или функций защиты памяти (MPU). Это позволило увеличить вдвое быстродействие и снизить энергопотребление. Наличие MMU и MPU позволяло запускать на них программы, работающие под управлением практически полноценных операционных систем.

Семейство ARM11 базируется на новой архитектуре *ARMv6*, особенностью которой является восьмиуровневый конвейер с перепорядочиванием очереди команд и динамическим предсказанием переходов, параллельное выполнение операций загрузки/выгрузки и операций в АЛУ; возможность пересылки 64-битовых слов; поддержка адресации данных без выравнивания и со смешанным порядком байтов, использование технологий SIMD, TrustZone и сопроцессорного расширения VFP, системы команд Thumb-2, появление кэш-памяти первого и второго уровня. Кроме того, впервые в архитектуре ARMv6 была реализована многоядерная версия процессора

ARM11 (модель ARM11 MPCore), четыре ядра которой реализовывали симметричную многопроцессорную обработку.

В последние годы фирма ARM на основе архитектур *ARMv7* и *ARMv8* активно развивает *семейство Cortex*. *Cortex* переводится как «кора головного мозга» — структура, отвечающая за согласованную работу органов, мышление, высшую нервную деятельность.

В зависимости от области применения *Cortex* делится на три профиля:

- *Cortex-A (Application)* — высокопроизводительные процессоры для ресурсоёмких областей применения, таких как смартфоны, телевизоры, промышленные сетевые устройства и серверы;

- *Cortex-R (Real-time)* — высокопроизводительные процессоры, оптимизированные для работы в режиме реального времени, где основными требованиями являются надёжность, отказоустойчивость и время реакции. Например, контроллеры промышленного и медицинского оборудования, автомобильная электроника и коммуникационные системы, модули беспроводной связи, цифровые камеры и телевизоры, контроллеры жестких дисков и так далее.

- *Cortex-M (MCU)* — ядра для микроконтроллеров, пришедшие на смену 8- и 16-разрядным микроконтроллерам, для приложений, где требуются низкая стоимость и энергопотребление, малые габариты, а также обработка сигналов (например, измерительные устройства, интерфейсы, автомобильные и промышленные системы управления, медицинское оборудование).

В отличие от классических процессоров ARM, имеющих зачастую сложные аббревиатуры, разработчики и потребители электроники легко могут определить функции и потенциальные возможности того или иного процессора Cortex. Например, в названии ARM7TDMI (не самом длинном) закодировано: ARM 7 - семейство, T -поддержка режима Thumb, D -JTAG порт, M - быстрый умножитель, I - встроенный блок эмуляции.

Основные типы семейства ARM отражены на рис. [98]

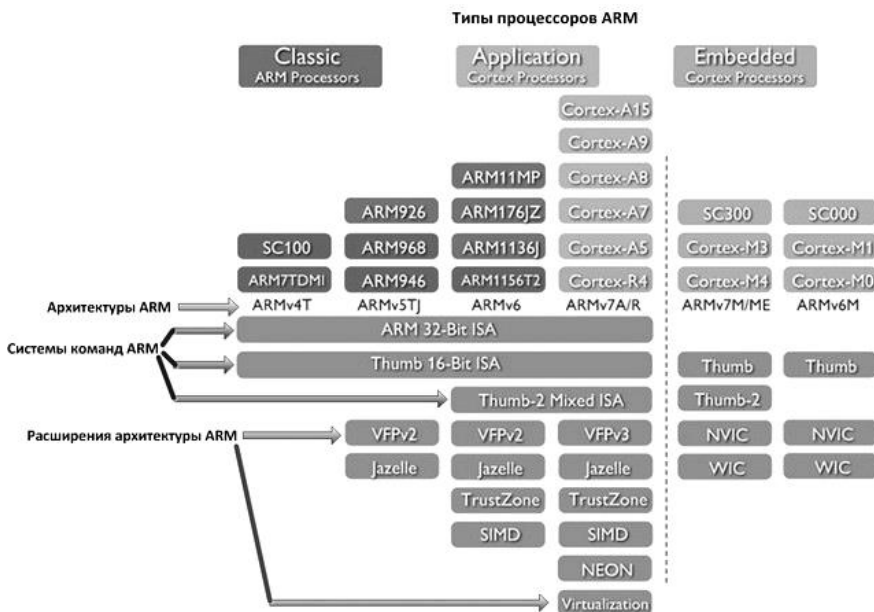


Рис. 7.3 Типы процессоров ARM

Архитектура ARMv7 отличается от ARMv6 большей производительностью и мощностью, наличием нескольких ядер, высокой тактовой частотой, поддержкой Flash и системы команд Thumb-2; использованием сопроцессора NEON, виртуальной памяти; технологии big.LITTLE, виртуализации.

Сопроцессор NEON реализует улучшенную технологию SIMD. Обладая независимым конвейерным модулем, собственными тридцатью двумя 64-разрядными и шестнадцатью 128-разрядными регистрами данных, NEON с легкостью работает с такими задачами, как кодирование и декодирование видео и звука и обработка, 2D- и 3D-графики. При этом модуль NEON тесно интегрируется с ARM-ядром, то есть в многоядерной архитектуре MPCore число модулей NEON совпадает с числом процессорных ядер. Уникальная архитектура NEON обеспечивает его производительность, как минимум в три раза превышающую производительность классических ARM-ядер на архитектуре ARMv5 и удвоенную производительность по

сравнению с модулями SIMD/VFP в архитектуре ARMv6. Энергопотребление в ряде приложений снижается за счет специальных режимов энергосбережения.

Технология big.LITTLE позволяет совместно использовать пару из высокопроизводительного и энергосберегающего процессоров для достижения одновременно высокой производительности и низкого потребления. Термин «big.LITTLE» можно перевести как «большой.маленький». Причем здесь имеется в виду не только более высокая и более низкая производительность, но и реальные отличия в физических размерах.

В зависимости от выполняемых задач архитектура позволяет отдельно задействовать высокопроизводительные и маломощные ядра в пределах одного устройства. Когда пользователь, например, запускает 3D игры и открывает браузер с множеством вкладок, активируются мощные ядра. В режиме ожидания, при совершении звонков, прослушивании музыки и прочих тривиальных задач мощные ядра временно отключаются и подключаются ядра, работающие на невысокой частоте. Таким образом, существенно снижается расход энергии (примерно вполтину) в обычном режиме использования смартфона, а быстродействие остается на высоком уровне.

Первая система big.LITTLE была построена на попеременной работе двух процессоров: большого Cortex-A15; и малого Cortex-A7. Процессоры имеют одну и ту же архитектуру, все инструкции выполняются одинаково. Отличаются они лишь их производительностью и микроархитектурой. За согласованную работу Cortex-A15 и Cortex-A7 отвечает система межсоединений (CCI-400). Еще один из самых важных элементов архитектуры является контроллер прерываний (GIC-400). A7 и A15 имеют полную программную совместимость, и подключены к единой подсистеме памяти. Оба кластера имеют собственный кэш. В итоге они могут практически мгновенно, незаметно для пользователя, переключаться между собой. Решения по распределению задач, в зависимости от запускаемых приложений, принимает планировщик задач. Планировщик определяет, какая мощность необходима для обеспечения оптимальной производительности при выполнении конкретной задачи. Затем подключаются соответствующие процессорные ядра. Изначально ядра A7 и A15 всегда работали только раздельно. Сейчас в случае необходимости

могут быть одновременно задействованы ядра big и LITTLE. Архитектура, которая позволяет активировать оба процессора для параллельной обработки данных получила название **big.LITTLE MP**. Количество ядер может быть асимметричное (например, 4+4+2). Также архитектура big.LITTLE нашла свое применение и в современных 64-битных процессорах.[100, 101]

Технология виртуализации позволяет запускать **на одной аппаратной платформе несколько операционных сред**, каждую на своей виртуальной машине. За счет этого достигается высокая безопасность и изолированность запущенных приложений. Благодаря виртуализации на одном мобильном телефоне можно будет устанавливать несколько операционных систем, одну из которых можно будет использовать, например, для работы, а другую в личных целях.[102,103]

Технология виртуализации **ARMv7** обеспечивает поддержку наиболее популярных гипервизоров виртуальных машин. **Гипервизор** осуществляет управление ресурсами и их разделение между различными операционными системами, выполняет изоляцию запущенных операционных систем друг от друга, а также может обеспечивать их взаимодействие (обмен файлами, сетевое взаимодействие и т.д.).

Эта технология должна способствовать **продвижению ARM на рынок серверных приложений**, где важной характеристикой является соотношение производительности и энергопотребления.

Архитектуру процессоров семейства **ARMv8 называют AArch64 или ARM64**. Она получила 64-битный набор инструкций и возможность работать с большим объемом оперативной памяти (4 Гбайт и больше). Естественно, предусмотрена совместимость с 32-битными приложениями (AArch32). Особенности ARMv8:

увеличено вдвое число регистров общего назначения (32 регистра с разрядностью 64 бита),

технология SIMD NEON поддерживает вычисления над **числами с плавающей запятой двойной точности и стандарта IEEE 754**, который является общепринятым форматом, используемый в программных реализациях арифметических действий (32-битная версия NEON поддерживала только одинарную точность и не в точности следовала стандарту). Увеличено с 16 до 32 количество 128-битных

регистров (совместимы с 64-битными регистрами), доступных сопроцессорам SIMD NEON и VFP. Набор инструкций SIMD NEON ускоряет работу приложений, отвечающих за обработку медиаданных и сигналов. В свою очередь VFP отвечает за вычисления над числами с плавающей запятой, обладая при этом малым энергопотреблением; новый набор инструкций с фиксированной длиной. Инструкции имеют размер 32 бита и многие совпадают с командами AArch32, хотя **условных инструкций стало меньше**, добавлены новые криптографические инструкции AES и SHA. [104]

Cortex-A наиболее активно развивающееся семейство. Одна из последних разработок - микросхема на базе архитектуры **Cortex-A73 и графика Mali-G71** обеспечат производительность уровня виртуальной реальности, смогут захватывать и воспроизводить видео 4К.

ARM Cortex-A73 будет выпускаться по 10-нм технологии и станет самым компактным и наиболее эффективным ядром на базе набора инструкций ARMv8-A. Он поддерживает TrustZone, NEON, виртуализацию и криптографию. Процессорные ядра получили 128-битный интерфейс AMBA 4 ACE, который позволит совмещать их в решениях big.LITTLE с Cortex-A53 и Cortex-A35. У Cortex-A73 имеет 64-килобайтный кэш инструкций, предвыборка на основе продвинутых алгоритмов. За счет этих и других оптимизаций Cortex-A73 окажется на 10% быстрее, чем Cortex-A72 на тех же частотах.

Графический процессор Mali-G71 на 50% производительнее, на 20% энергоэффективнее, чем предыдущее поколение. [105]

В отличие от профиля **A**, **Cortex-M** развивается не так активно. Широко известны ядра **M0, M3, M4 и M7**. [106 - 108]

Наиболее популярно **ядро Cortex M3**, выполненное по архитектуре **ARMv7-M** (профиль M). Все основные производители микроконтроллеров, которые присутствуют в России, используют это ядро: STMicroelectronics, Texas Instrument, NXP, ATMEL, Analog Devices, Renesas и т.д.

Его основой является 32-разрядное ядро **CM3Core**, состоящее из АЛУ с возможностями аппаратного деления и умножения данных, 13 регистров общего назначения, двух указателей стека, счетчика команд и ряда специальных регистров, в т.ч. регистр статуса программы. Кроме того, Cortex-M3 может оперировать с невыровненными данными, что отличает его от предшествующих архитектур

ARM, и поддерживает битовые операции. Ядро выполнено по Гарвардской архитектуре с трехступенчатым конвейером и предсказанием ветвлений.

Несмотря на использование Гарвардской архитектуры, все пространства памяти ЦПУ Cortex-M3 имеют общую линейную систему распределения адресов в пространстве размером 4 Гбайт.

В ядре Cortex-M3 реализован набор инструкций Thumb-2, который в некоторых случаях позволяет добиться увеличения удельной производительности на 70% по сравнению с ядром ARM7TDMI-S, исполняющим инструкции Thumb, и на 35% - по сравнению с тем же ядром, исполняющим инструкции ARM.

Процессор Cortex-M3 содержит систему прерываний NVIC, характеризующуюся задержкой вызова процедуры обработки прерывания всего лишь 12 машинных циклов (для сравнения, ARM7TDMI-S требует 24-42 цикла). Основу системы прерываний составляет контроллер векторизованных вложенных прерываний, который в стандартной реализации поддерживает одно немаскируемое прерывание и 32 прерывания общего назначения с 8 уровнями приоритетов (в общем случае число прерываний может достигать 240 при 256 уровнях приоритета). [109]

Блок защиты памяти является опциональным компонентом ядра Cortex-M3. Он позволяет повысить надежность встраиваемых систем за счет защиты критичной информации, используемой операционной системой, от действия пользовательских программ.

Доступ к встроенной отладочной системе реализован посредством порта, который с внешней средой связывается по 2-проводному последовательному отладочному порту SW или стандартному порту JTAG.

Для микроконтроллеров с масочным ПЗУ предусмотрен специальный блок Flash Patch, который во время отладки позволяет осуществлять выборку инструкций не из ПЗУ, а из статического ОЗУ, тем самым существенно упрощая процедуру отладки программного кода для таких микроконтроллеров.

Обмен информацией выполняется с помощью ***шинной матрицы и традиционных магистралей АНВ и АРВ.***

Матрица представляет собой совокупность высокоскоростных шин, управляемых контроллером, в состав которого входит арбитр. Это позволяет реализовывать множество параллельных каналов связи между шинами Cortex и периферийными устройствами, минуя процессорное ядро. *Подобный механизм описан в разделе 4.* Если же два устройства, например, процессорный блок и канал ПДП пытаются получить доступ к одному и тому же периферийному устройству, арбитр разрешит конфликт и предоставит доступ устройству с высшим приоритетом.

Cortex-M0 - это Cortex-M3 с усеченным набором команд. Микроконтроллеры, использующие ядро Cortex-M0, позиционируются производителями в качестве замены восьмиразрядных моделей. Их отличительной особенностью стала предельно низкая стоимость и малое энергопотребление, при сохранении многих возможностей архитектуры ARM.V его основе лежит архитектура ARMv6-M. *Cortex-M0 имеет одношинную фон-неймановскую архитектуру и использует только систему команд Thumb и некоторые команды Thumb2.* Это позволило упростить процессорное ядро и применять дешевую 16-ти разрядную память при некотором снижении производительности. Быстродействие ядра Cortex-M0 составляет 0.84 DMIPS / МГц. Это значит, что на максимальной частоте работы ядра в 50МГц, достигается производительность 45 DMIPS. Энергопотребление процессора M0, в зависимости от исполнения и решаемых задач, колеблется от 73 до 4мкВт/МГц. Модификация Cortex-M0+ по сравнению с Cortex-M0 дополнительно оснащен блоком защиты памяти MPU, буфером Micro Trace Buffer для отладки программ, а также имеет двухступенчатый конвейер вместо трехступенчатого и упрощенный доступ к периферийным блокам и линиям ввода/вывода. [110]

Ядро Cortex-M4, по сравнению с Cortex-M3, не характеризуется ростом общих показателей, построено по той же архитектуре, а отличается - введением DSP-инструкций. Модификация Cortex-M4F дополнительно содержит блок вычислений с плавающей запятой и поддерживает технологию SIMD. Это существенно ускоряет обработку потоковых данных и способствует его использованию в системах управления и обработки информации.

Основными особенностями *ядра Cortex-M7* являются:

- суперскалярная архитектура с динамическим предсказанием;
- шестиступенчатый конвейер с двойной выборкой;
- система команд ARMv7-M Thumb;
- кэш памяти с гарвардской структурой: 4 кбайта памяти команд I-cache, 4 кбайта памяти данных D-cache;
 - работа в режимах “Thumb” и “Debug”;
 - шинная матрица с расширенными интерфейсами памяти: 64-бит Instruction Tightly Coupled Memory (ITCM), 64-бит AXI master (AXIM), 2×32-бит Data Tightly Coupled Memory (DTCM), АНВ-Lite peripheral (АНВР);
 - интегрированный контроллер векторов прерываний (NVIC);
 - модуль защиты памяти (MPU);
 - блок вычислений с плавающей запятой FPU.

Наличие нескольких шин памяти делает возможным одновременное выполнение двух команд чтения или одной команды записи и одной команды чтения. [111]

Развитие профиля *Cortex-M* рассмотрим на примере микроконтроллеров семейства *STM32* фирмы *STMicroelectronics*. Их можно разделить на следующие группы:

семейства базового сегмента: STM32F0 (48 МГц, Cortex-M0), STM32F1 (72 МГц, Cortex-M3), STM32F3 (72 МГц, Cortex-M4);

- семейства с пониженным потреблением: STM32L0 (32 МГц, Cortex-M0+), STM32L1 (32 МГц, Cortex-M3), STM32L4 (84 МГц, Cortex-M4),

- семейства с повышенной производительностью: STM32F2 (120 МГц, Cortex-M3), STM32F4 (84...180 МГц, Cortex-M4), STM32F7 (216 МГц, Cortex-M7) [111]

Наиболее часто используется семейство STM32 с ядром Cortex-M3, в состав которого входят две группы микроконтроллеров:

Access (F101xx): частота тактирования 36 МГц, от 32 до 128 кбайт флэш-памяти от 6 до 16 кбайт SRAM, до 7 коммуникационных интерфейсов. Линейка Access разработана с тем, чтобы внедрить 32-разрядные микроконтроллеры в критичные к стоимости применения или в 16-разрядные проекты.

Performance (F103xx): частота тактирования 72 МГц, от 256 до 512 кбайт флэш-памяти, до 64 кбайт SRAM, контроллер статической памяти с поддержкой Compact Flash, SRAM, PSRAM, NOR и NAND

памяти, с поддержкой LCD параллельного интерфейса (F103Vx). Микроконтроллеры имеют до 13 коммуникационных интерфейсов, в том числе USB и CAN. Линейка микроконтроллеров Performance ориентирована на применения, которым необходимы одновременно повышенная производительность обработки и низкое энергопотребление. [112,113]

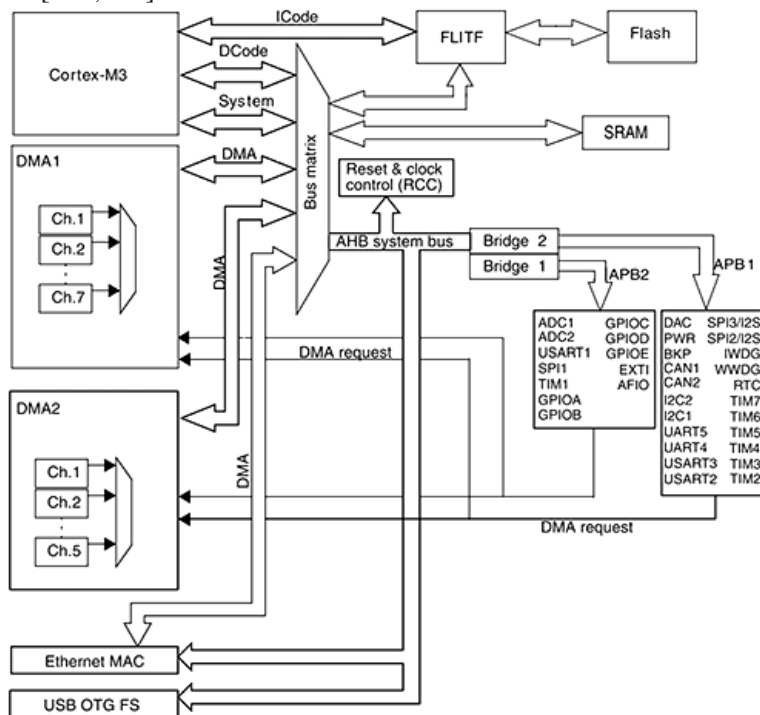


Рис. 7.4. Упрощенная структура микроконтроллера STM 32 с ядром Cortex-M3

Упрощенная структура микроконтроллера с ядром **Cortex-M3** представлена на рисунке 7.4, где

Bus matrix (матрица шин) - контроллер высокоскоростных шин, обеспечивающий независимую связь и арбитраж между системной шиной и шиной данных ядра, DMA, Ethernet (masters) и периферией - SRAM, FLASH, AHB (slaves).

Flash interface (**FLITF**) - интерфейс Flash-памяти, обеспечивающий чтение, запись, стирание, чтение с буфером предварительной выборки, защиту памяти (от записи или чтения).

ICode bus- шина команд (инструкций) и векторов прерываний.

DCode bus- шина данных — обеспечивает связь ядра с интерфейсом данных Flash для выборки/записи данных и отладочного доступа.

System bus- системная шина ядра, связывающая ядро и периферийные устройства

AHB system bus (Advanced High-performance Bus) — шина, которая связывает матрицу шин и периферийные шины **APB** (Advanced Peripheral Bus) с помощью мостов (**Bridge**).

Быстродействующие компоненты, Ethernet и USB OTG FS подключены к шине АНВ напрямую.

Периферийные шина APB1 и APB2 предназначены для управления устройствами с различным быстродействием.

APB2 работает на частоте ядра и обслуживает наиболее скоростные устройства: АЦП (до 1 миллиона выборок в секунду), ЦАП, некоторые таймеры, порты ввода/вывода GPIO, USART1 (до 4.5 МБит/с), SPI1 (до 18 МБит/с) и так далее.

Быстродействие APB1 ограничено частотой 36 МГц (половина частоты ядра) и к ней подключены более медленные устройства (контроллеры USART, CANbus, I2C, SPI/I2S, ШИМ и так далее).

DMA (Direct Memory Access) — каналы прямого доступ к памяти. Контроллеры **DMA** подключены к шинной матрице по собственной шине и имеют прямой доступ ко всей памяти, включая память, на которую отображены *управляющие регистры периферии*.

Reset & Clock Control (RCC) — обеспечивает тактирование ядра и периферии и сброс контроллера. **RCC** позволяет в процессе работы МК переключаться между источниками тактирования — внешним кварцем, керамическим резонатором и внутренней калиброванной RC-цепочкой, а также запрашивать у **NVIC** немаскируемое прерывание при сбое внешнего источника синхронизации и переключаться на внутренний, увеличивая надёжность системы.

Состав периферии и их характеристики могут отличаться в различных модификациях

Особенности архитектуры STM32F0,F3,F4 изложены в [109,110,112,113].

В качестве иллюстрации возможностей STM32F756 на рисунке 7.5 приведена его структурная схема. [111]

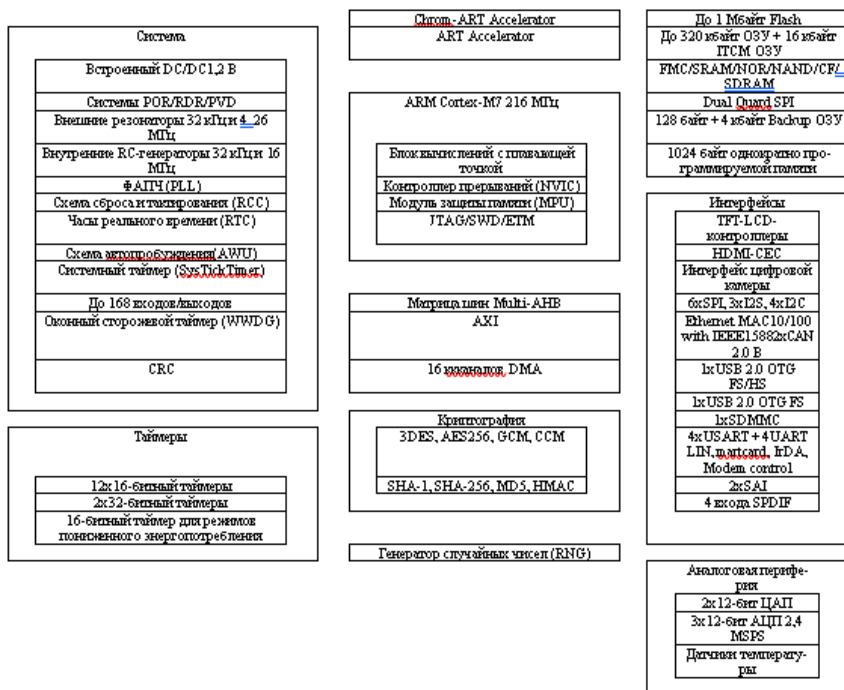


Рис. 7.5. Структурная схема STM32F756

К достоинствам семейства STM 32 можно отнести не только программную совместимость от младших моделей к старшим, но и *«pin-to-pin совместимость»*, то есть для одного размера корпуса все сигналы сохраняются на тех же линиях ввода/вывода для разных вариантов микроконтроллеров. Это позволяет в зависимости от решаемых задач выбирать требуемый объем флэш-памяти, ОЗУ, состав периферийных блоков, *не изменяя печатную плату*. Однако эту особенность необходимо контролировать.

Ядро Cortex-M7 используют в своих разработках фирмы Atmel (микроконтроллеры Atmel® SMART SAM570 с частотой 300 МГц), Freescale и другие. [114,115]

В последние годы на рынок активно выходят процессорные ядра (IP-модули) **MIPS (Microprocessor without Interlocked Pipeline Stages) - "микрпроцессор без блокировок в конвейере"**. Они широко используются и в отечественных разработках. [116-121]

Архитектура MIPS была одной из первых RISC-архитектур, получившей признание со стороны промышленности. Она была анонсирована в 1986 году. Затем MIPS Technology была первой компанией, выпустившей микропроцессоры с 64-битовой архитектурой в 1991 г.

Архитектуры, разработанные MIPS, занимают серьезную долю рынка сетевых и телекоммуникационных устройств: кабельные и DSL-модемы (95 и 52% рынка, соответственно), кабельные и спутниковые телевизионные приставки (76 и 30% рынка, соответственно), схемы для цифрового ТВ (59% рынка), DVD-проигрыватели (70%), сетевые концентраторы и коммутаторы (40%), принтеры (62%) и т.д. Компания MIPS имеет более 125 лицензиатов своих ядер, которые совместно производят более 500 миллионов процессоров каждый год. [121]

Однако по выпуску продукции они существенно уступают ARM.

Многие авторы подчеркивают уникальность базовой системы команд MIPS, которую разрабатывали университетские ученые.

Команды трехадресные, имеют более простые форматы, чем у ARM, выполняют, как правило, только одну операцию.

Форматы команд представлены в таблице 5.

Целочисленные арифметические и логические команды реализуют прямую регистровую и непосредственные способы адресации.

Для выполнения умножения и деления используются два специальных регистра HI, LO и **аппаратный блок деления и умножения (MDU) с автономным конвейером**, что позволяет выполнять умножение/деление параллельно с другими операциями

Логические команды помимо основных логических функций включают команды сдвигов на заданное число разрядов и сравнения с аппаратным формированием признаков.

Таблица 5. Форматы команд MIPS

Тип	-31-	Формат				-0-
R	opcode (6) КОП	Rs (5) Регистр операнда S	Rt (5) Регистр операнда T	Rd (5) Регистр результата	shamt (5) Управление сдвигом	funct (6) Функция расширения кода операции
I	opcode (6) КОП	Rs (5) Операнд S	Rt (5) Операнд T	immediate (16) Непосредственная константа		
J	opcode (6) КОП	address (26) Адрес				

Команды пересылки используют только косвенную адресацию со смещением (индексом) и обеспечивают работу с байтом, полусловом, словом.

Команды передачи управления используют, в основном, относительную адресацию. В отличие от аналогичных команд ARM признаки операций формируются аппаратно и в специальный регистр не записываются. В состав команд передачи управления входит ограниченное число условных переходов (меньше, чем в ARM).

Формат команд с плавающей запятой аналогичен целочисленному. Для работы с ним используется отдельный набор регистров. Состав команд минимизирован: 4 арифметических действия с одинарной и двойной точностью, сравнения, пересылки, условные переходы. Всего 14 команд. (в ARM - 44 команды, включая вычисления экспоненциальных и тригонометрических функций, логарифмов, квадратного корня).

Основными признаками MIPS- архитектуры являются развитая регистровая структура (32 регистра общего назначения, часть из которых может выполнять «закрепленные» функции),

банки «теневого» регистров (до 16), увеличивающие скорость обработки прерываний

большая часть команд содержит только одну операцию, а инструкции ARM выполняют несколько операций перед записью в регистр (смена операнда, проверка бита состояния и др.),

сокращенное количество способов адресации,

параллельная многопоточная обработка (*мультипредиинг*), основой которой являются простые и однотипные по структуре команды, иерархический многоступенчатый конвейер (до 15 ступеней), децентрализованная аппаратная обработка.

Развитие архитектуры MIPS идет аналогично ARM: 32 и 64-разрядные архитектуры (*MIPS32 и MIPS64*), 16-битные команды *MIPS16*, иерархическая организация интерфейсов, матрица шин, контроллеры прямого доступа к памяти, двухуровневый кэш, расширения для обработки цифровых сигналов, плавающей запятой, SIMD, суперскалярная архитектура, виртуализация, многопоточность, многоядерность и так далее.

Классификация современных микропроцессоров *MIPS*:

«Классические» ядра от MIPS Technologies

MIPS 4KE, M4K, M14K, M14Kc - малый размер, цена, энергопотребление

MIPS 24K, 34K, многоядерный 1004K - эффективность по производительности / милливатт

MIPS 74K, многоядерный 1074K - суперскаляр, высшая производительность

Новое (2012) поколение ядер от MIPS Technologies - Aptiv Generation

microAptiv - продолжает линейку M14K, добавляет DSP

interAptiv - продолжает 1004K, добавляет улучшенный менеджер когерентности

proAptiv - на 60-70% производительнее чем 1074K, дополнительные конвейеры ALU

Новейшее (2013г)- поколение ядер от Imagination Technologies - Warrior Generation

Анонсированный MIPS P5600 - основан на proAptiv, добавляет SIMD и виртуализацию. [120]

Для рынка встраиваемых систем MIPS предлагает несколько ядер, полностью совместимых снизу вверх. Для 32-битных микроконтроллеров, работающих в задачах реального времени, предназначены ядра M4K/M4KE, дальнейшее развитие этого семейства ядро M14K™ имеет *microMIPS™* архитектуру сжатия набора команд (аналогично Thumb 2), а ядро *microAptiv* дополнено средствами цифровой обработки сигналов (ЦОС) с поддержкой множественного потока данных SIMD и функциями защиты.

Результаты сравнения параметров ядер MIPS и Cortex приведены в таблицах 6, 7. Оценка производительности проводилась на основании тестов *CoreMark (CM)* и *Dhrystone MIPS (DMIPS)*.

Таблица 6. Сравнения ядер MIPS M4K и ARM Cortex-M3.

Ядро	M4K		Cortex-M3	
DMIPS/МГц	1.5		1.25	
CM/МГц	2.03		1.06-1.9	
Оптимизация ядра	По быстродействию	По площади	По быстродействию	По площади
По быстродействию	228	100	135	50
Площадь кристалла, мм ²	0.64	0.185	0.74	0.38
Типовая мощность, мВт/МГц	0.214	0.066	0.165	0.084

По тестам производительности ядро *microAptiv* показывает большую на 25% (DMIPS) и 40% (CoreMark) производительность относительно Cortex-M4. [122]

На основе ядра *microAptiv* компания Microchip выпускают 32-х разрядные контроллеры PIC32MZ с DSP расширением команд, контроллером кэш-памяти, контроллером управления памятью с буфером динамической трансляции адресов TLB MMU, микропроцессоры имеют до 2Мбайт Flash памяти программ, 512К ОЗУ, тактовую частоту до 200МГц. [123]

Упрощенная структура PIC32MZ приведена на рисунке 7.6.

Таб. 7. Сравнение архитектур microAptiv и Cortex-M4

Особенность	MIPS microAptiv	ARM Cortex-M4
Длина Конвейера	5	3
Версия с Кэшем/MMU	Да	Нет
Система команд	MIPS32 и microMIPS	Thumb2
Число инструкций	300	155
РОН (наборов)	32 (16)	16 (1)
Поддержка Closely coupled memory	Да	Нет
Реакция на прерывания	10 циклов	12 циклов
Трассировка команд	Да	Нет
Быстрый канал отладки	Да	Нет
DSP возможности		
Число DSP инструкций	159	80
SIMD команд	70	38
Команд Умножения/MAC	38	29
Выделенный DSP/MDU	Да	Нет
Аккумулятор	Да(64бит x 4)	Нет
16x8, два 8x8 Умножителя/MAC	Да	Нет
Команды сдвига	Да	Нет
Команды Сравнения/Pick	Да	Нет

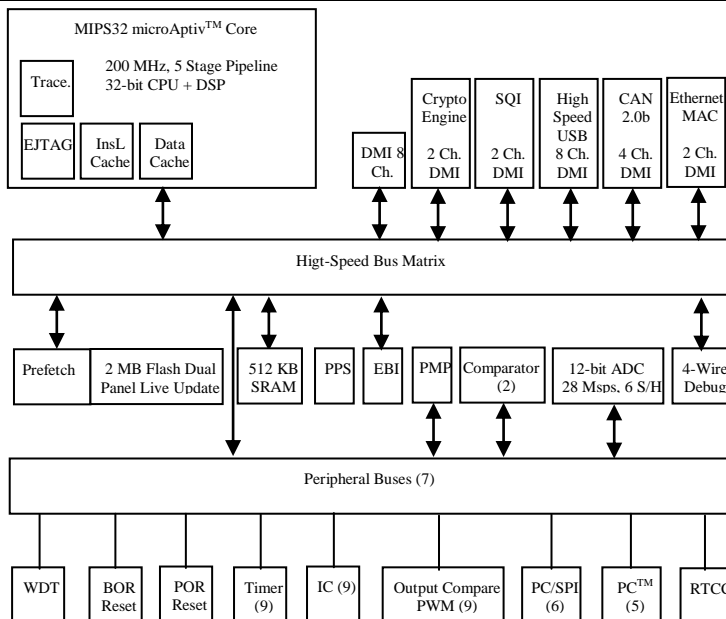


Рис. 7.6. Упрощенная структура PIC32MZ

Как показывает тестирование в открытой среде CoreMark, микроконтроллер PIC32 с ядром M4K обгоняет микроконтроллеры NXP и STMicroelectronics на базе ядра Cortex-M3 на 20—50%, а МК NXP на основе Cortex-M0 —на 63%, даже несмотря на то, что M4K выполняет два цикла ожидания при обращении к памяти, а Cortex-M — ни одного. Скорость работы M4K достигает 1,5 DMIPS/МГц, в то время как Cortex-M3 выполняет до 1,25 DMIPS/МГц, а Cortex-M0 — до 0,9 DMIPS/МГц.

Результаты сравнения, приведенные выше, относятся к 2013 году. ***Очевидно, что ядра ARM и MIPS достойные соперники и это соревнование будет продолжаться.***

Принимая во внимание характеристики 32-разрядных микроконтроллеров, возникает вопрос ***о целесообразности применения 8-разрядные ОМК.***

Основные различия между 8- и 32-разрядными микроконтроллерами - в стоимости, производительности процессорного ядра, функциональных возможностях, размерах корпуса и энергопотреблении в статическом режиме. Начиная новый проект, разработчики должны тщательно оценивать требуемые мощности процессорного ядра, типы и количество интерфейсов, энергопотребление, стоимость. Целесообразно выбирать устройство, наиболее подходящее для решения требуемой задачи, не обращая внимание на разрядность процессорного ядра.

Во многих приложениях не требуется высокая производительность процессорного блока, а необходим интенсивный обмен с различными периферийными устройствами.

Структуры современных 8-разрядных микроконтроллеров содержат контролеры основных последовательных интерфейсов, аппаратные средства для реализации шинных матриц, каналов ПДП, систему событий (например, XMEGA), что позволяет периферийным устройствам взаимодействовать без участия процессорного ядра.

При насыщенной функциональности число внешних контактов 8-разрядных микроконтроллеров может быть сокращено за счет использования коммутационных матриц.

Для выполнения одного и того же 8-разрядного приложения на 32-разрядном микроконтроллере нужно от полутора до трех раз

больше SRAM, чем на 8-разрядном, увеличиваются объемы флэш-памяти.

Преимущества 8-разрядных ОМК при пониженных требованиях к производительности можно было бы перечислять и далее.

Если сравнить стоимость 8- и 32-разрядных микроконтроллеров ведущих производителей с похожим составом периферии, объемом флэш-памяти, числом выводов и так далее, то окажется, что 8-разрядные ОМК примерно на 20% дешевле.

Потребность в 8-разрядных ОМК подтверждает и объем рынка микроконтроллеров: совокупный среднегодовой темп роста 6,4% против 6,9% у 32-разрядных.

Таким образом, не всегда мощные микроконтроллеры являются оптимальным выбором. Есть восьмиразрядные ОМК, которые удовлетворяют потребностям многих задач, в том числе *при реализации беспроводных решений*. [124]

Контрольные вопросы

Введение

1. Какие задачи решают контроллеры в автоматизированных системах?
2. В чем преимущества и недостатки децентрализованной обработки?
3. Какие функции реализуют нормализаторы?
4. Какие основные положения закона Мура?
5. В чем основные отличия микроконтроллера от микропроцессора? Функции и структура

Основные характеристики

6. Какие параметры характеризуют операционные возможности ОМК?
7. Какими параметрами характеризуются форматы данных?
8. В чем преимущества и недостатки формата с фиксированной запятой (точкой)?
9. Какие характеристики микроконтроллера определяет длина формата при представлении числовых данных?
10. В чем преимущества и недостатки формата с плавающей запятой (точкой)?
11. Какие характеристики микроконтроллера определяет длина поля характеристики? мантиссы?
12. Для каких целей используется двоично-десятичное представление чисел? Сравнить характеристик с двоичным представлением данных
13. В чем особенности выполнения арифметических операций в двоично-десятичном коде?
14. В чем особенность обработки логических данных? Для каких целей они используются?
15. С помощью каких средств обрабатывается алфавитно-цифровая информация?
16. Какими параметрами характеризуется формат команды?
17. Что определяет поле операций? Какая структура поля операции обеспечивает максимальную производительность (быстродействие)?

18. Как влияет адресность команды на характеристики микроконтроллера?

19. В каких случаях следует использовать прямую, прямую регистровую, непосредственную, косвенную, относительную, стековую, базовую, индексную, автоинкрементную, автодекрементную, базовую индексную с масштабированием адресации? Их сравнительные характеристики

20. Почему при разработке программ предпочтение следует отдавать относительной адресации?

21. Как влияет формат команды на характеристики микроконтроллера?

22. Какие характеристики процессорного блока необходимо контролировать?

23. Как влияет организация интерфейсов на характеристики ОМК?

24. На какие параметры ОМК влияют характеристики шин данных? адреса? управления?

25. В чем особенности режима микропроцессора? микроконтроллера? расширенного микроконтроллера?

26. Какие методы используются для оценки быстродействия ОМК?

27. Какие универсальные тесты используются для оценки быстродействия?

28. В чем преимущество специализированных тестов? Как их сформировать?

29. Как влияет на характеристики системы состав периферийных устройств ОМК?

30. В чем преимущества и недостатки виртуальных средств моделирования?

31. Какие параметры ВСМ необходимо контролировать?

32. В чем особенности применения ВСМ с графической средой программирования?

33. Какие функции реализуют оценочные модули? внутрисхемные эмуляторы?

34. Сравнительная характеристика статической и динамической логики.

35. В каких случаях следует использовать масочные и однократно программируемые ПЗУ?
36. В чем преимущества флэш-памяти? Какие у неё недостатки?
37. В чем преимущества памяти с электрическим стиранием информации? Для каких целей она используется?
38. Какие новые технологии должны заменить флэш-память?
39. В чем преимущества мемристоров по сравнению с транзисторами?
40. Какие способы формирования тактовой частоты используются в ОМК? Их сравнительные характеристики.
41. Для каких целей следует анализировать нагрузочную способность ОМК?
42. Как реализовать в ОМК режим энергосбережения? В каких случаях это необходимо?

Обобщённая структура ОМК

43. В чем достоинства и недостатки архитектуры фон Неймана?
44. Какие преимущества обеспечивает применение гарвардского принципа? В чем его недостатки?
45. Сравнительная характеристика гарвардской и неймановской организации.
46. В чем преимущества модифицированной гарвардской организации?
47. В каких направлениях происходит развитие гарвардской архитектуры?
48. Какие структурные особенности ЦП необходимо анализировать при выборе ОМК?
49. Какие недостатки характерны для ЦП, использующих регистровую архитектуру на основе аккумулятора?
50. В чем преимущества регистр-регистровой архитектуры?
51. В чем преимущества и недостатки CISC архитектур?
52. В чем преимущества и недостатки RISC архитектур?
53. В чем особенности MISC архитектур?
54. Какие основные характеристики блока управления и синхронизации БУС?

55. Какие разновидности генераторов входят в состав БУС? С чем связано их разнообразие?
56. В чем преимущества и недостатки внутренних (резидентных) синхрогенераторов?
57. В каких случаях следует использовать генераторы с внешним кварцевым резонатором?
58. Какие типы ПЗУ используются в качестве резидентной памяти программ? Их сравнительные характеристики.
59. Для каких целей используется ПЗУ данных в ОМК? Как оно реализовано?
60. Как организована память в ОМК с гарвардской архитектурой?
61. Как организована память в ОМК с неймановской архитектурой?
62. Какие принципы используются при организации резидентной памяти данных? Как разделяется адресное пространство?
63. Какие преимущества обеспечивает оконная адресация? В чем её недостатки?
64. Какие преимущества обеспечивает динамическая оконная адресация?
65. Какие способы ввода-вывода информации используются в ОМК?
66. В каких случаях следует использовать программный способ?
67. В каких случаях следует использовать синхронный обмен? Асинхронный? Сравнительная характеристика по скорости и аппаратным затратам.
68. Когда целесообразно использовать ввод-вывод по прерыванию?
69. Принцип обработки прерываний.
70. Функции контроллера прерываний.
71. Структура подпрограммы обработки прерываний.
72. В чем преимущества и недостатки режима прямого доступа к памяти ПДП?
73. Принцип обмена в режиме ПДП. Функции контроллера ПДП.
74. В каких случаях наиболее эффективен режим ПДП?

Параллельные порты

75. На основании каких характеристик следует оценивать потенциальные возможности параллельных портов ОМК?
76. Какие информационные характеристики следует анализировать?
77. Какие аппаратные средства должны входить в состав параллельного порта?
78. Какие преимущества обеспечивает использование триггера Шмитта во входных цепях портов?
79. Какова нагрузочная способность портов? Какие параметры нагрузки следует анализировать? Как обеспечить требуемую нагрузочную способность?
80. Как реализовать синхронный обмен через параллельный порт? асинхронный?
81. Какова программная модель параллельного порта?

Последовательные порты

82. В чем преимущества последовательного обмена по сравнению с параллельным? Недостатки?
83. Принцип работы последовательного порта и назначение элементов обобщенной структуры.
84. Какими параметрами последовательного порта определяются пиковая пропускная способность? Эффективная?
85. В каких случаях целесообразно использование синхронных форматов обмена в последовательных портах? квазисинхронных? асинхронных? Сравнительная характеристика этих способов
86. Какие методы кодирования используются в последовательных портах ОМК? В чем их недостатки?
87. Какие методы используются для повышения качества синхронизации?
88. Какие основные способы повышения достоверности информации используются в последовательных портах?
89. Какие структурные решения обеспечивают повышенную скорость обмена в последовательных интерфейсах?
90. Какие электрические характеристики последовательных портов необходимо анализировать?
91. Как оценить реальную скорость передачи данных в последовательных портах?

UART

92. В каких случаях следует использовать последовательный порт UART/USART?

93. Какова структура кадра и назначение отдельных полей?

94. Какие особенности порта необходимо контролировать при выборе ОМК? Почему?

95. В каких случаях следует использовать последовательный порт SPI? В чем его преимущества по сравнению с UART?

Интерфейс SPI

96. Основные характеристики SPI.

97. Принцип работы порта SPI и назначение его линий. Сколько линий требуется для организации обмена?

98. В чем особенности SPI при работе в режиме ведущего? Ведомого?

99. Как реализовать дуплексный режим? симплексный?

100. В чем особенности синхронизации SPI?

101. Какие особенности порта SPI необходимо контролировать при выборе ОМК? Почему?

Интерфейс I2C

102. В каких случаях целесообразно использовать I2C? Почему?

103. Основные технические характеристики I2C.

104. Сколько линий требуется для организации внутрисхемного обмена? Внешнего?

105. Как обеспечивается достоверность передаваемой информации в I2C?

106. Для каких целей используется поле адреса?

107. Краткая характеристика формата обмена.

108. Как обеспечивается синхронизация устройств, подключенных к интерфейсу?

109. Как обеспечивается согласование устройств с различным быстродействием?

110. Как обеспечивается электрическая совместимость модулей, подключенных к интерфейсу?

111. Функции ведущего и ведомого модулей.

112. Как реализуется процедура арбитража?

113. Какие аппаратные средства необходимы в ведущем устройстве? Ведомом?

114. В чем преимущества I2C при организации микропроцессорных систем?

115. Сравнительная характеристика SPI и I2C. Какой интерфейс обеспечивает большую скорость обмена данными?

Подсистема прерываний

116. В каких случаях целесообразно использование ввода-вывода по прерыванию?

117. Обобщенная структура контроллера прерываний. Назначение основных элементов.

118. На основании каких характеристик следует оценивать потенциальные возможности подсистемы прерываний ОМК?

119. Как влияет вид запроса прерываний на характеристики системы?

120. В чем отличие автовекторных прерываний от векторных? Какие прерывания наиболее часто используются в ОМК?

121. Как управлять приоритетом прерываний?

122. Какие способы организации стека используются в ОМК?

123. Как организовать программный стек и определить его глубину?

124. Как увеличить число внешних прерываний в ОМК? Какие технические средства для этого необходимы?

125. Какова структура подпрограммы обработки прерываний?

Блок временных событий

126. Какие основные функции реализует блок временных событий (БВС)? Возможна ли реализация этих функций на базе традиционных счетчиков-таймеров? Как это сделать?

127. Какой блок ОМК выполняет функции эталонного сигнала?

128. Как с помощью БВС измерить длительность импульса? Частоту? Скважность? Фазовый сдвиг? Период?

129. Как обеспечить требуемую погрешность измерения?

130. В каких случаях следует использовать метод мгновенного измерения частоты? метод среднего?

131. Как с помощью БВС реализовать генератор частоты? Широтно-импульсный модулятор (ШИМ)? Импульс заданной длительности?

132. В чем преимущество аппаратной реализации ШИМ? Как её выполнить?

133. Как на базе ШИМ организовать ЦАП?
134. Для решения каких задач используются счетчики-таймеры?
135. Какие основные характеристики счетчиков \ таймеров необходимо анализировать?
136. Какие преимущества обеспечивает синхронный режим работы счетчика? В чем его недостатки?
137. Принцип функционирования матрицы счетчиков-таймеров.
138. Режим захвата. Для каких целей используется? Как средства необходимы для его реализации?
139. Режим сравнения. Для каких целей используется? Как средства необходимы для его реализации?
140. Как в режиме захвата выполнить измерение приведенных выше параметров?
141. Как в режиме сравнения сформировать приведенные выше сигналы?
142. Какие преимущества обеспечивает применение сопроцессора временных событий? почему?
143. Для каких целей предназначен сторожевой таймер?
144. Принцип работы сторожевого таймера
145. Какие основные характеристики сторожевого таймера следует анализировать?

Средства ввода/вывода аналоговой информации

146. Для каких целей используется аналоговый компаратор? Его основные характеристики.
147. Какие дополнительные параметры аналогового компаратора необходимо анализировать в ОМК?
148. Как организовать допусковый контроль с помощью компараторов?
149. Как можно использовать выходной сигнал компаратора в ОМК? Приведите примеры.
150. Для каких целей используется гистерезис?
151. Для каких целей используется ЦАП? Его основные характеристики.
152. Какие параметры определяют погрешности ЦАП?
153. Какие функции реализуются в ОМК для управления ЦАП?

154. Как с помощью ЦАП реализовать требуемую аналоговую функцию? Какие погрешности ЦАП при этом необходимо учитывать?

155. Какие основные источники погрешностей характерны при преобразовании аналоговой величины в цифровую?

156. Как оценить погрешность квантования? Дискретизации? Датирования?

157. Какие характеристики АЦП необходимо контролировать при выборе ОМК?

158. Какие параметры АЦП влияют на погрешность квантования? дискретизации? восстановления аналогового сигнала?

159. В чем особенности измерения дифференциальных сигналов по сравнению с потенциальными?

160. Как влияет входное сопротивление и емкость на погрешность преобразования?

161. Какие дополнительные погрешности необходимо анализировать при оценке метрологических характеристик АЦП?

162. Какие методы используются для снижения температурной погрешности? погрешности смещения нуля и полной шкалы?

163. Какие функции выполняет устройство выборки- хранения (УВХ)? Для каких задач его наличие не требуется?

164. Принцип работы УВХ

165. Принцип работы АЦП на основе метода поразрядного уравнивания. В чем преимущества и недостатки этого метода?

166. Принцип работы сигма-дельта АЦП. Преимущества и недостатки этого метода.

167. Какие дополнительные характеристики необходимо контролировать в блоке АЦП ОМК?

168. В чем преимущества от использования режима ПДП? Всегда ли он будет эффективен?

169. Какие преимущества обеспечивает наличие режима порогового детектирования? Режим последовательного и выборочного опроса каналов?

170. Какое влияние оказывает ЦП ОМК на работу АЦП? Как устранить это влияние?

171. Принцип работы блока АЦП ОМК. Функции основных элементов.

172. Какие характеристики аналогового мультиплекса(коммутатора) необходимо анализировать?

Особенности структурной организации ОМК

173. Какие методы используются в ОМК для повышения скорости выборки команд? данных?

174. Какие методы нужно использовать для уменьшения времени выполнения команд?

175. Как ускорить процедуру обработки прерываний?

176. В чем преимущества и недостатки конвейера? С помощью каких приемов можно устранить недостатки?

177. В чем преимущества и недостатки кэш-памяти?

178. Какими преимуществами обладают проблемно-ориентированные микроконтроллеры? Привести примеры

179. В чем преимущества многоуровневых магистралей?

180. Какие преимущества обеспечивает применение матрицы шин?

181. Принцип работы системы событий. В чем отличие от магистрального способа обмена?

182. Почему многоядерные микроконтроллеры наиболее эффективны при использовании последовательных сетевых принципах обмена?

183. Для каких целей используется коммутационная приоритетная матрица?

184. Какие методы используются для снижения энергопотребления ОМК?

185. Какие технические решения используются для снижения энергопотребления магистралей?

Способы реализации структур микроконтроллеров

186. Сравнительные характеристики ОМК с фиксированной структурой и структурой, формируемой на стадии изготовления?

187. Какие основные преимущества систем на кристалле? В чем их недостатки?

188. Сравнительные характеристики технологий ASIC, FPGA, SPDL.

189. В чем особенности организации программируемых систем на кристалле ПСК с однородной структурой и блочного типа?

190. Обобщенная структура системы на кристалле. Назначение основных элементов.

191. Какие особенности процессорного ядра NIOS? Интерфейса AVALON?

192. Какие структурные методы используются в AVALON для повышения быстродействия?

193. Как организуется взаимодействие ведущих и ведомых устройств на примере рис5.7?

194. В чем особенности FPSLIC по сравнению с программируемыми системами на кристалле?

195. Какие основные особенности структурной организации FPSLIC? Какие преимущества обеспечивает двухпортовое ОЗУ? Как распределяются ресурсы памяти?

196. Какие методы используются для повышения быстродействия FPSLIC? Расширения функциональных возможностей?

197. В каких случаях следует использовать PSoC фирмы Cypress?

198. Какие основные отличия в структурной организации PSoC от рассмотренных ранее ПСК?

199. Какие возможности по реализации аналоговых компонентов? цифровых?

200. Каким образом реализуется компромисс между повышенной функциональностью и ограниченным числом линий ввода-вывода?

Режимы работы ОМК

201. Какие методы следует использовать в рабочем режиме для снижения энергопотребления?

202. Какие основные характеристики режимов пониженного энергопотребления?

203. В чем отличие в режимах ожидания и останова? С обеспечением каких характеристик связано многообразие режимов?

204. Сравнить режимы энергосбережения микроконтроллеров AVR.

205. Для каких целей используется режим сброса? Источники сигнала сброса?

206. Какие функции реализуются в режиме сброса?

207. Какие требования предъявляются к внешнему импульсу сброса? Как сформировать импульс положительной полярности? отрицательной?

208. Как осуществляется контроль напряжения питания?

209. Какие операции выполняются в режиме программирования?

210. Для каких целей используются идентификаторы? Калибровочные ячейки? конфигурационные ячейки? Ячейки защиты? Особенности программирования.

211. В чем особенности внутрисхемной отладки и программирования? Какие средства необходимы для реализации этого режима?

212. В чем особенности фонового режима внутрисхемного отладчика?

213. В чем основное преимущество порта JTAG по сравнению с другими резидентными средствами отладки? В чем его недостатки? Как они устраняются?

214. Какие функции можно реализовать, используя JTAG?

215. Какие дефекты можно выявить с помощью JTAG?

216. Какие программно-аппаратные средства необходимы для работы с JTAG?

217. Структура порта JTAG.

218. В чем особенности режима самопрограммирования? Для каких целей его используют? Как его реализовать?

Структурная организация 32-разрядных микроконтроллеров

219. Какие причины привели к существенному увеличению объема продаж 32-разрядных микроконтроллеров?

220. В чем основные отличия 8-разрядных микроконтроллеров от 32-разрядных?

221. В каких случаях целесообразно применение 8-разрядных микроконтроллеров?

222. Какие преимущества обеспечивает применение IP-ядер? В чем их недостатки?

223. Какие преимущества обеспечивают наличие нескольких режимов работы? увеличенное число регистров и банков регистров?

224. Основные характеристики формата команд ARM. Какие преимущества он обеспечивает? Сравнить с форматом AVR.

225. В чем особенности системы команд Thumb? Thumb-2?

226. Какие средства разработаны для мультимедийных приложений? Языка Java? Обеспечения безопасности? Снижения энергопотребления? Работы в режиме гипервизора? Их особенности.

227. На основе каких структурных компонентов разрабатывается ядро ARM? В чем их особенности?

228. Какие методы используются в структуре микроконтроллера LPC 2300/2400 для повышения быстродействия?

229. Почему в различных модификациях ARM используются гарвардская или неймановская архитектуры?

230. Какие профили Cortex использует ARM? В чем их особенности?

231. Тенденции развития семейства ARM. Какие последние достижения?

232. В чем особенности архитектуры ARMv7 и ARMv8?

233. Какие основные особенности архитектуры?

234. Какие преимущества обеспечивает в Cortex-M3 применение шинной матрицы?

235. В чем отличие Cortex-M3 от M0 и M4?

236. Какие основные особенности ядра Cortex-M7? Какие преимущества это обеспечивает?

237. Структура STM 32 с ядром Cortex-M3. Какие технические решения способствуют повышению быстродействия? Производительности?

238. В чем отличия в функциональной и структурной организации микроконтроллеров STM 32 с ядром Cortex-M3 и ядром Cortex-M7?

239. Какие преимущества обеспечивает «pin-to-pin совместимость»?

240. Какие основные признаки MIPS-архитектуры? Сравнить с ARM.

241. Сравнить структуру STM32F756 и PIC32MZ

Список использованных источников

1. Шашлов С. Закону Мура - 40 лет. URL: <http://www.ixbt.com/editorial/moorelaw40th.shtml>
2. Теги -Закон Мура. URL: <http://www.3DNews.ru>
3. Радиоежегодник. Радиолоцман №8,2015. URL: <http://www.rlocman.ru/book/book.html?di=160032>
4. Павлов П. Тенденция развития микропроцессоров и микроконтроллеров. Современная электроника №2, 2007, стр. 12-15
5. Майская В. Микроконтроллеры, микроконтроллеры, микроконтроллеры. URL: http://www.electronics.ru/files/article_pdf/0/article_602_213.pdf
6. Танненбаум Э. Архитектура компьютера. 5-е издание. -СПб.: Питер, 2010. -844 с.:ил.
7. Марков С. Цифровые сигнальные процессоры. Книга 1. М.: МИКРОАРТ, 1996. -144с.: ил.
8. Солонина А.И., Улахович Д.А., Яковлев Л.А. Цифровые процессоры обработки сигналов фирмы Motorola. СПб.: БХВ-Петербург, 2000. -512с.: ил.
9. Способы адресации. URL: <http://www.xServer.ru/computer/>
10. Кривченко И. Системы на кристалле: общее представление и тенденции развития. Компоненты и технологии №6,2001. URL: http://www.kit-e.ru/articles/plis/2001_06_48.php
11. Микроконтроллеры PSoC. [Электронный ресурс]. URL: <http://www.efo.ru/doc/Cypress/Cypress.pl?721>
12. Микроконтроллеры производства компании Silicon Laboratories. URL: <http://www.efo.ru/doc/Silabs/Silabs.pl?1165>
13. Инструментальные средства разработки и отладки для микроконтроллеров. URL: <http://www.phyton.ru/development-tools>
14. Средства программирования микроконтроллеров. URL: http://www.de.ifmo.ru/bk_netra/
15. Радиоежегодник. Proteus по-русски. Вып.24, 2013г. URL: <http://www.rlocman.ru/book/book.html?di=148418>
16. Радиоежегодник. FLOUCODE 6 - новые полеты. Вып.29, 2013г. RL: <http://www.rlocman.ru/book/book.html?di=150138>

17. Ключев А., Платунов А. Встроенные инструментальные средства современных микроконтроллеров. Электронные компоненты №6, 2002. URL: <http://www.elcomdesign.ru/magazine/archive/#>
18. Печерица Н. Что вслед за флеш?. Экспресс-Электроника, №6, 2005. URL: <http://citforum.ru/hardware/articles/>
19. Вихарев Л. Перспективные технологии производства памяти. Современное состояние. Компоненты и технологии №12, 2006. URL: http://www.kit-e.ru/articles/memory/2006_12_66.php
20. Точка зрения Hewlett-Packard: микрочипы на основе мемристоров- будущее компьютеров. URL: <http://www.dailytechinfo.org/electronics/>
21. Мемристоры: пора ли переписывать учебники? URL: <https://3dnews.ru/906763>
22. Выбор микроконтроллера. URL: <http://www.gaw.ru/html.cgi/txt/publ/micros/micros.htm>
23. Глубоков А. RX -новая вершина в эволюции микроконтроллеров. Компоненты и технологии №8, 2010. URL: http://www.kit-e.ru/articles/micro/2010_08_116.php
24. Руководство по 8-битным AVR-микроконтроллерам XMEGA URL: http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh_xmega_a/index.htm
25. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин. Л.: Машиностроение, Ленинградское отделение, 1979. 384 с.: ил.
26. Шагурин И.И. Микропроцессоры и микроконтроллеры фирмы Motorola: Справочное пособие. - М.: Радио и связь, 1998. - 560с.: ил.
27. Сравнение архитектуры POWER с другими RISC-архитектурами URL: <http://www.gaw.ru/html.cgi/adv/publ/micros/risk1.htm>
28. Керамические резонаторы Murata. URL: http://www.compitech.ru/html.cgi/arhiv/03_03/stat_6.htm
29. Козаченко В.Ф. Практическое руководство по применению 16-ти разрядных микроконтроллеров MSC 196/296 во встроенных системах управления. М.: ЭКОМ, 1997. 500с.: ил.

30. Ключев А.О., Ковязина Д.Р., Петров Е.В., Платунов А.Е. Интерфейсы периферийных устройств. -СПб., СПб ГУИТМО, 2010. - 290с.: ил.

31. Брейман А.Д. Сети ЭВМ и телекоммуникации. Учебное пособие. Часть 1. Общие принципы построения сетей. Локальные сети. - М.: МГАПИ, 2001. - 75с.: ил

32. Агуров П.В. Последовательные интерфейсы ПК. Практика программирования. -СПб.: БХВ-Петербург,2005. -496с.: ил.

33. Гук М.Ю. Шины PCI, USB, FireWire. Энциклопедия. -СПб.: Питер,2005. - 540с.: ил.

34. Новицкий Александр. Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация в ADuC70xx фирмы Analog Devices. Компоненты и технологии №3,5,9 2009. URL: <http://www.kit-e.ru/archive.php>

35. Востриков А.А., Балонин Н.А., Сергеев А.М. Внутриплатаые интерфейсы встраиваемых систем: Учеб. пособие/ СПбГУАП. СПб., 2012. URL: guap.ru/guap/kaf44/trud/vostrikov_balonin_sergeev_vpi.pdf

36. 2. Шина I²C и как её использовать (включая спецификацию). URL: <http://www.gaw.ru/html.cgi/txt/interface/iic/>

37. Евстифеев А.В. Микроконтроллеры AVR семейства Mega. Руководство пользователя. М.: Издательский дом «Додэка-XXI», 2007. -592 с.: ил.

38. Документация MicroChip на русском языке. URL: <http://www.microchip.ru/lit/?mid=1x0>

39. Никашин В.А. Аналого-цифровые и цифроаналоговые преобразователи. Справочник. СПб.: Корона принт, 2003. -224 с.:ил.

40. Цифро-анлоговые преобразователи. URL: <http://neo-chaos.narod.ru/useful/adc/dacs.pdf>

41. Интегральные микросхемы: Микросхемы для аналого-цифрового преобразования и средств мультимедиа. Выпуск1. М.: ДОДЕКА,1996. -384 с.: ил.

42. MSC1212. Прецизионный АЦП и ЦАП с микроконтроллером серии 8051 и флэш-памятью. URL: http://www.gaw.ru/html.cgi/txt/ic/Texas_Instruments/micros/mcs51/MSC1212.htm

43. ADuC 842. MicroConvertor с 16- разрядным АЦП и встроенным микроконтроллером 8052 с Flash памятью на 62 КБ. URL: http://www.autex.spb.su/download/microconverter/aduc842_pr_rus.pdf

44. Махлин А. Дельта-сигма модуляция: назад в будущее. Компоненты и технологии №11, 2010 URL: http://www.kit-e.ru/articles/circuit/2010_11_154.php

45. Сигма-дельта АЦП.. URL: <http://digteh.ru/digital/sigmaadc.php>

46. Шагурин И., Мокрецов М. Семейство 68HCS12 -новое поколение 16-разрядных микроконтроллеров компании Motorola, Компоненты и технологии, №2,2004. URL: http://www.kit-e.ru/articles/micro/2004_2_106.php

47. Защищенные криптографические микроконтроллеры компании Inside Secure. URL: http://insiderus.ru/article/mc_modules_is.html

48. Самоделов А. Криптография в отдельном блоке: криптографический сопроцессор семейства STM32F4xx. Новости электроники, №6, 2012. URL: <http://www.compel.ru/lib/ne/2012/6/4-kriptografiya-v-otdelnom-bloke-kriptograficheskiy-soprotsessor-semeystva-stm32f4xx>

49. X MOS выпускает новые многоядерные микроконтроллеры общего назначения. URL: <http://www.rlocman.ru/news/new.html?di=147064>

50. Texas Instruments представляет многоядерные системы-на кристалле с архитектурой KeyStone для облачных приложений. URL: <http://www.rlocman.ru/news/new.html?di=144168>

51. Редькин П.П. 32/16-битные микроконтроллеры ARM7 семейства AT91SAM7 фирмы Atmel. Руководство пользователя. М.: Издательский дом «Додэка-XXI», 2008. -704 с.: ил.

52. Тревор М. Микроконтроллеры ARM7 LPC 2300/2400. Вводный курс разработчика./ пер. с англ. Евстифеева А.В.- М.: «Додэка-XXI», 2010. -336 с.:ил.

53. Микроконтроллеры ARM (ARM7 и ARM9). Обзор. URL: <http://www.phyton.ru/development-tools/arm-cortex/arm7-arm9-microcontrollers>

54. Королев Б.Ю., Хартов В.Я. Сравнение архитектур микропроцессоров класса ARM. URL: <http://engsi.ru/doc/706095.html>

55. Егоров А. Микроконтроллеры NXP Semiconductors на базе архитектуры Cortex. Компоненты и технологии, № 5, 2010. URL: http://www.kit-e.ru/articles/micro/2010_05_90.php

56. Войтенко Ю. Обзор архитектуры ARM Cortex-A17. Маленький шаг в будущее. . URL: [http:// www. testlabs.kz/processors/1728-](http://www.testlabs.kz/processors/1728-)

[arm-cortex-a17-review.html](#)

57. Сеть на кристалле — мини-интернет внутри процессора.
URL: <https://geektimes.ru/post/143510/>

58. Д.Слюсар, В.Слюсар Беспроводные сети на кристалле. Перспективные идеи и методы реализации. ЭЛЕКТРОНИКА наука, технология, бизнес №6, 2011. URL: http://www.electronics.ru/files/article_pdf/2/article_2962_866.pdf

59. Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах программируемой логики. СПб.: БХВ -Петербург, 2002. — 608 с.: ил.

60. Архитектуры малопотребляющих процессоров и способы оптимизации энергопотребления. URL: <http://www.russianelectronics.ru/leader-r/review/2192/doc/52461/>

61. Квашин А. Новое семейство малопотребляющих микроконтроллеров MSP430F5xx. Компоненты и технологии №7, 2009. URL: http://www.kit-e.ru/articles/micro/2009_07_6.php

62. Самые маломощные в мире микроконтроллеры Texas Instruments теперь выпускает в миниатюрных корпусах. URL: <http://asonika-online.ru/news/21/>

63. AVR-микроконтроллеры picoPower компании Atmel с ультранизким потреблением. URL: http://www.gaw.ru/html.cgi/txt/publ/rts/atmel_pico_avr.htm

64. Silicon Labs выпускает самые малопотребляющие микроконтроллеры с ядром ARM Cortex-M0. URL: <http://randomstar.org/news/7083-silicon-labs-vypuskaet-samyemalopotrebyayuschie-mikrokontrollery-s-yadrom-arm-cortex-m0.html>

65. Микроконтроллеры с низким энергопотреблением для приложений Smart Power. Журнал РАДИОЛОЦМАН, декабрь 2012. URL: <http://www.rlocman.ru/review/article.html?di=144167>

66. Вычужанин В. Состояние рынка и расширение сферы применения ПЛИС. Компоненты и технологии №5, 2004. URL: http://www.kit-e.ru/articles/plis/2004_5_60.php

67. Архитектура ПЛИС (FPGA) URL: <https://marsohod.org/11-blog/265-fpga>

68. Программируемые логические интегральные схемы Altera. URL: <http://www.altera-plis.ru/Mikroshemi.html>

69. Тарасов И. Системы на кристалле на базе FPGA Xilinx со встроенными процессорами PowerPC. Компоненты и технологии №7, 2005 с.62-66. URL: http://www.kit-e.ru/assets/files/pdf/2005_07_62.pdf

70. Микросхемы семейства Excalibur. URL: <http://www.gaw.ru/html.cgi/txt/ic/Altera/excalibur.htm>

71. Nios II по-русски. URL: http://www.naliwator.narod.ru/nios_II.html

72. Микроконтроллер для встроенного применения - NIOS. Система команд и команды, определяемые пользователем. URL: http://www.compitech.ru/html.cgi/arhiv/02_08/stat_106.htm

73. Кривченко И. Системная интеграция в электронике -FPSLIC. URL: <http://www.efo.ru/doc/Atmel/Atmel.pl?673>

74. Королев Н. Atmel FPSLIC: элементная база 21 века. URL: <http://www.chipinfo.ru/literature/chipnews/200101/16.html>

75. Конфигурирование Xilinx FPGA с помощью микроконтроллера. URL: <http://microsin.net/programming/arm/configure-xilinx-fpga-with-mcu.html>

76. Новости электроники URL: <http://www.efo.ru/components/atmel/>

77. Щерба А. Программируемые аналоговые схемы Anadigm. Проекты, примеры применения. Компоненты и технологии №12,2012. URL: http://www.kit-e.ru/articles/plis/2012_12_6.php

78. Программируемые системы на кристалле (PSoC™) URL: <http://www.gaw.ru/html.cgi/txt/ic/Cypress/PSoC/start.htm>

79. Скуснов А. PSoC решения снижают стоимость миниатюрных систем «Электронные компоненты» №7-2004 URL: http://cec-mc.ru/techinfo/mikrokontrolleri_mikroprotessori/sistemi_na_kristalle_PSoC_PSD_PSD/PSoC/664/

80. Микроконтроллеры PSoC URL: <http://www.efo.ru/doc/Cypress/Cypress.pl?721>

81. Микроконтроллеры PSoC 3 на базе ядра 8051 и PSoC 5 на базе ARM Cortex-M3. URL: <http://www.kit-e.ru/news/PSoC3-PSoC5.php>

82. Режимы пониженного напряжения AVR. URL: <http://avrprog.blogspot.ru/2013/03/powermode.html>

83. Подсистема начального сброса в микроконтроллерах. URL:

<http://radiostorage.net/?area=news/1027>

84. Гаврилюк Д. Внутрисхемное программирование. Компоненты и технологии № 8, 2003. URL: http://www.kit-e.ru/articles/circuit/2003_8_114.php

85. JTAG– основы и немного теории. URL: <http://portal-ed.ru/index.php/testirovanie-i-kontrol/92-jtag-osnovy-i-nemnogo-teorii>

86. Интерфейс JTAG ?-Это очень просто. URL: <https://habrahabr.ru/post/190012/>

87. JTAG: тестирование и программирование микросхем. URL: <http://promwad.ru/tehnologii/jtag-testirovanie-programmirovanie-mikroshem>

88. Городецкий Ами. Новый JTAG- стандарт IEEE 1149.7. Компоненты и технологии №4,2010. URL: http://www.kit-e.ru/assets/files/pdf/2010_04_16.pdf

89. Внутрисхемный отладчик debugWire. URL: <http://fusecalc.mirmk.ru/help/help08.htm>

90. AVR 109. Самопрограммирование AVR. Перевод с англ. URL: <http://microsin.net/programming/avr/avr109-self-programming.html>

91. AVR. Учебный Курс. Использование Bootloader'a. URL: <http://easyelectronics.ru/avr-uchebnyj-kurs-ispolzovanie-bootloadera.html>

92. Сигаев А.Микроконтроллеры фирмы AMD. URL: <http://kasus.ru/articles/10/index.html>

93. Королев Н. Atmel:32-разрядные Flash-микроконтроллеры на ядре AVR32. Компоненты и технологии №11,2008. URL: http://www.kit-e.ru/articles/micro/2008_11_74.php

94. Емелин А., Шагурин И. RISC-микроконтроллеры с процессорным ядром ARM. Часть 1. URL: http://www.chipnews.ru/html.cgi/arhiv/02_06/6.htm

95. Микроконтроллеры ARM7 / ARM9 обзор URL: <http://lib.chipdip.ru/225/DOC000225753.pdf>

96. Изучение системы команд процессора ARM. URL: <https://marsohod.org/index.php/prodmarsohod2/amber-arm-soc/226-arm-instr>

97. Новодачный М. Микроконтроллеры на основе ядер ARM: широкое масштабирование аппаратных возможностей в рамках программной совместимости. Современная электроника № 9,2010. URL: <http://www.soel.ru/upload/clouds/1/iblock/a9d/a9d3b554ce2d9600a5cadd3045b35651/20100916.pdf>

98. Лебеденко Е Процессоры ARM: маленькие гиганты электронного мира. URL: http://www.mobimag.ru/Articles/5303/Processor_ARM_malenkie_giganty_yelektronnogo_mira.htm

99. Немного о ARM Security Extensions (aka ARM TrustZone) URL: <https://habrahabr.ru/post/309618/>

100. Секреты архитектуры big.LITTLE URL: <http://www.russianelectronics.ru/developer-r/review/2192/doc/58808/>

101. Изумрудный А Все что нужно знать об архитектуре big.LITTLE и как она работает в смартфонах. URL: <http://mobcompany.info/interesting/vse-chno-nuzhno-znat-ob-arxitekture-big-little-i-kak-ona-rabotaet-v-smartfonax.html>

102. Аппаратная виртуализация. Теория, реальность и поддержка в архитектурах процессоров. URL: <https://habrahabr.ru/company/intel/blog/196444/>

103. Микропроцессоры ARM Holdings угрожают Intel и AMD. URL: <http://slovar-ozhegova.ru/?p=850>

104. Пятковский Ю. Процессоры ARMv8: в чем преимущества 64-битной архитектуры? URL: <http://itc.ua/articles/protssoryi-armv8-v-chem-preimushhestva-64-bitnoy-arhitektury/>

105. Жуков А. Cortex-A73 - новое процессорное ядро для быстрых смартфонов. URL: <http://xdrv.ru/news/hardware/19722>

106. Микроконтроллеры Cortex-M0/M3/M4 URL: <http://mcucpu.ru/index.php/ucontrollers/mcu/113-mikrokontrollery-cortex-m0m3m4>

107. На что стоит променять Cortex-M3?

[URL: <https://habrahabr.ru/company/efo/blog/277491/>]

108. Попов Р., Квашин А. Web-семинар STM32Cube MX. URL: http://www.compel.ru/wordpress/wp-content/uploads/2015/11/Webinar_final.pdf

109. Староверов К. Микроконтроллеры на основе ядра ARM Cortex M3. Новости электроники №1,2008. URL:

<http://www.compel.ru/lib/ne/2008/1/4-mikrokontrolleryi-na-osnove-yadra-arm-cortex-m3>

110. Гавриков В. 32 бита за полдоллара. Новости электроники №8, 2013 [Электронный ресурс]. URL: <http://www.compel.ru/lib/ne/2013/8/3-32-bita-za-poldollara>.

111. Гавриков В. STM32F7: новый флагман- новые горизонты. URL: <http://www.compel.ru/lib/ne/2016/2/7-stm32f7-novyyiy-flagman-novyie-gorizontyi-3>

112. Крылов Е. STM32 - 32разрядные микроконтроллеры на основе ядра Cortex-M3, Компоненты и технологии №11, 2008. URL: http://www.kit-e.ru/articles/micro/2008_11_82.php

113. Ознакомительное руководство по ARM микроконтроллерам Cortex-M3. URL: <http://lishnih.net/arm/STM32book.pdf>

114. Микроконтроллеры Atmel | SMART на базе ARM Cortex-M7. URL: <http://www.atmel.com/ru/ru/technologies/cpucore/arm/cortex-m7.aspx>

115. Микроконтроллеры Freescale на новом ядре ARM — Cortex-M7. URL: <http://www.symmetron.ru/news/freescale-Cortex-M7.shtml>

116. Особенности архитектуры MIPS компании MIPS Technology. URL: http://citforum.ru/hardware/app_kis/glava_54.shtml

117. Изучаем MIPS-ассемблер. URL: <http://habrahabr.ru/post/147685/>

118. Афанасьев А. Микропроцессорные 32-разрядные ядра MIPS для высокопроизводительных встраиваемых систем. Компоненты и технологии №3,2014. URL: http://www.kit-e.ru/articles/micro/2014_3_91.php

119. Винин Г. MIPS или ARM?. URL: <http://www.russianelectronics.ru/leader-r/review/2192/doc/55583/>

120. Панчул Ю. Imagination Technologies. Презентация на семинаре Microchip Masters Russia. URL: http://gamma.spb.ru/media/pdf/masters2013/2013_P32mips_ru.pdf

121. Эволюция и современное состояние архитектуры MIPS. URL: <http://www.russianelectronics.ru/developer-r/review/2192/doc/40351/>

122. 32-бит ядра MIPS в высокопроизводительных микроконтроллерах Microchip. URL: <http://gamma.spb.ru/obuchenie/stati/microchip/254-32-bit-yadra-mips-v-vysokoproizvoditelnykh-mikrokontrollerakh-microchip>

123. Обзор нового поколения высокопроизводительных микроконтроллеров PIC32MZ.. URL: http://gamma.spb.ru/media/pdf/masters2013/2013_P32MZ_Pheripheral_ru.pdf

124. Эйленд А. Выбор микроконтроллера: 8- или 32-разрядный? URL: http://www.electronics.ru/files/article_pdf/4_article_4711_291.pdf

Учебное издание

Иоффе Владислав Германович

Структурная организация однокристалльных микроконтроллеров

Учебное пособие

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА»
(Самарский университет)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Изд-во Самарского университета.
443086, Самара, Московское шоссе, 34.