

# **ЭЛЕКТРОПРИВОДЫ С СИСТЕМАМИ ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ**

**Ульяновск 2006**

**Федеральное агентство по образованию**  
Государственное образовательное учреждение высшего профессионального образования  
**Ульяновский государственный технический университет**

# **ЭЛЕКТРОПРИВОДЫ С СИСТЕМАМИ ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ**

**Учебное пособие**

**Составитель В. М. Иванов**

**Ульяновск 2006**

УДК 681.31 (076)

ББК 32.973.2.я73

Э46

Рецензент д-р техн. наук, профессор А. В. Кузнецов.

Ответственный за выпуск зав. кафедрой «Электропривод и АПУ» д-р техн. наук, профессор В. Н. Дмитриев.

Утверждено редакционно-издательским советом университета в качестве учебного пособия.

**Электроприводы с системами числового программного управления:**  
Э46 учебное пособие / сост. В. М. Иванов. – Ульяновск: УлГТУ, 2006. – 152 с.  
ISBN 5-89146-829-8

Рассматриваются принципы построения автоматизированных систем управления предприятиями и производственными процессами. Даются понятия об уровнях управления и о технических средствах их реализации. Проводится анализ устройств числового программного управления и тенденций их развития. Основное внимание уделено принципам построения цифровых электроприводов, включая следящие приводы, и алгоритмам их реализации. Подробно изложены особенности архитектуры микроконтроллеров AVR и их периферии.

Учебное пособие составлено в соответствии с учебной программой курса «Электроприводы с системами числового программного управления» и предшествующего курса «Микропроцессорное управление электроприводами» и может быть использовано студентами заочного и дневного факультетов в качестве дополнительного пособия по указанным курсам, в том числе при выполнении курсовых работ. Целью курсовой работы является получение навыков проектирования цифровых систем управления электроприводом применительно к системам числового программного управления.

Работа подготовлена на кафедре ЭП и АПУ.

**УДК 681.31(076)**  
**ББК 32.973.2.я73**

ISBN 5-89146-829-8

© Иванов В. М., составление, 2006  
© Оформление. УлГТУ, 2006

## ОГЛАВЛЕНИЕ

Предисловие.....	4
1. Тенденции развития автоматизированного производства.....	5
1.1. Уровни и состав систем управления ГПС .....	7
1.2. Комплексные средства автоматизации .....	11
2. Системы числового программного управления.....	14
2.1. Модернизация станков с ЧПУ .....	16
2.2. Концепции разработок и стратегия развития средств ЧПУ .....	19
2.3. Общая характеристика систем ЧПУ WINCNC .....	21
2.4. Аппаратные и технологические возможности УЧПУ SINUMERIC..	23
3. Принципы построения систем управления ЭП.....	29
3.1. Варианты сопряжения УЧПУ с ЭП .....	30
3.2. Примеры разработки следящего электропривода на основе МК.....	35
3.3. Принципы построения цифровых систем управления тиристорными преобразователями .....	37
3.3.1 Алгоритмы управления.....	41
4. Микроконтроллеры .....	47
4.1. Микроконтроллер ATmega128, ATmega128L .....	48
4.1.1. Краткое описание .....	49
4.1.2. Ядро центрального процессорного устройства AVR .....	54
4.1.3. Память .....	60
4.1.4. Аналогово-цифровой преобразователь .....	72
4.1.5. Порты ввода-вывода .....	84
4.1.6. Таймеры-счетчики 1 и 3.....	95
4.1.7. Системная синхронизация и тактовые источники.....	126
4.1.8. Прерывания .....	135
5. Общие указания по выполнению курсовой работы.....	145
5.1. Задание на курсовую работу .....	146
5.2. Требования к пояснительной записке .....	148
Библиографический список.....	149

## ПРЕДИСЛОВИЕ

Революционные изменения, произошедшие за последние десятилетия в области микроэлектроники, производстве средств вычислительной техники и информационных технологиях привели к смене концепций управления во всех сферах производства. Прогресс в области электронных и информационных технологий существенным образом изменяет как подходы к проектированию систем управления производственными механизмами и установками, так и их элементную базу. В многоуровневых системах управления электропривод (ЭП) как компонент относится к нижнему уровню управления. Следует отметить, что нижний (исполнительный) уровень составляет основу любого производства и именно он определяет конкурентоспособность и качество выпускаемой продукции. В системах числового программного управления (СЧПУ) механизмами к качественным показателям ЭП предъявляются особо жесткие требования. В цифровых системах управления ЭП с МК управлением достигаются показатели, недостижимые для средств аналогового управления.

Появление мощных, полностью управляемых полевых транзисторов MOSFET (Metal - Oxide - Semiconductor Field Effect Transistor) и биполярных транзисторов с изолированным затвором IGBT (Isulated Gate Bipolar Transistor) привело к развитию преобразовательной техники и расширению сферы применения синхронных и асинхронных электроприводов с преобразователями частоты. Другим фактором, обусловившим совершенствование регулируемого электропривода и расширение сфер его применения, было создание микропроцессоров (МП) и однокристальных микроконтроллеров (МК) достаточной вычислительной мощности.

Существующая широкая номенклатура МК предназначена для разнообразных сфер применения и удовлетворяет самые разнообразные требованиям к параметрам локальных цифровых регуляторов. Усилия разработчиков в области электропривода и электронных компонентов привели к созданию интеллектуальных драйверов, интегрированных модулей, включающих в своем составе силовой преобразователь, вычислительное ядро, преобразователи информации и датчики физических параметров. Необходимо помнить при этом, что новые решения в МП-технике, в том числе в архитектуре МК, исходят из обобщения аппаратных принципов систем управления. Именно совместные усилия специалистов различного профиля привели к созданию семейств МК, предназначенных для управления электродвигателями.

Область МП средств, предназначенная для управления двигателями, получила специальное название (Motor Control). Ведущие производители микропроцессорной техники для встроенных применений: Analog Devices, Atmel, Intel, International Rectifier, Motorola, Siemens, Texas Instruments.

## **1. ТЕНДЕНЦИИ РАЗВИТИЯ АВТОМАТИЗИРОВАННОГО ПРОИЗВОДСТВА**

В начале 70-х годов прошлого столетия в странах с передовой технологией были сформулированы две основные концепции развития производства, которые коренным образом повлияли на перестройку управления в производственных системах (ПС).

На основании этих концепций сформировались два глобальных направления, определивших развитие техники и технологии. Первое направление условно объединилось под названием CAD (Computer Aided Design – автоматизированное проектирование). В отечественной литературе аналогами этих терминов являются АСУ, САПР (автоматизированные системы управления, системы автоматического проектирования). Второе направление получило название CAM (Computer Aided Manufacturing – автоматизированное производство), а в отечественной литературе – АСУ ТП (автоматизированные системы управления технологическими процессами). В промышленно развитых странах наблюдалось развитие сразу двух направлений CAD и CAM, но доля капитальных вложений в одно или другое направление была разной. Так, в США основное внимание уделялось концепции CAD, и к концу 70-х годов США значительно опережали другие страны в этой области. Япония сосредоточила капитальные вложения в концепцию CAM и в результате значительно опередила США на международном рынке по целому ряду промышленных видов продукции. Достижения в направлениях CAD и CAM привели к необходимости и возможности создания объединенной концепции CAD/CAM (САПР/АСУ ТП). Этот информационный компонент получил обозначение MIS (management information system – информационная служба).

Системы CAD/CAM обладают следующими особенностями: строятся на базе аппаратных и программных средств ЭВМ для целей технического проектирования, графического представления информации, машинного анализа, управления производством. Благодаря модульному принципу построения системы имеют многотерминальный доступ со стороны пользователей; позволяют автоматически преобразовать информацию в команды управления средствами производства с устройствами числового программного управления.

Развитие и взаимное объединение CAD, CAM, MIS привело к созданию принципиально нового производства FMS (Flexible Manufacturing System – гибкая производственная система (ГПС)). С применением робототехники значительно повысилась гибкость управления ПС, и был сделан практический шаг в области разработки и создания «безлюдного» и «безбумажного» производства.

ГПС должна содержать автоматизированные системы технологической подготовки производства [1, 2]. Кроме того, ГПС не может нормально функционировать, если вовремя не будет планироваться ее работа, в том числе сменно-суточные задания, материально-техническое обеспечение, поставка

материалов, инструментов и др. Для быстрой плановой реализации этих работ необходима связь ГПС с автоматизированной системой управления производством (АСУП). Таким образом, ГПС в своем развитии вырастает до гибкого автоматизированного производства [3], в котором можно различать подсистемы: автоматизированную систему технологической подготовки производства (АСТПП), планирования, диспетчеризации, транспортно-накопительную, оборудования с ЧПУ, контроля качества, контроля исполнения, робототехническую и информационную службу доступа и управления. Причем информационная вычислительная подсистема объединяет все остальные. Каждая из этих подсистем представляет собой набор технических средств, программное обеспечение (ПО) и сопровождающую информационную документацию.

В общем случае в систему обеспечения функционирования ГПС входят:

АСУП – автоматизированная система управления предприятием;

АСНИ – автоматизированная система научных исследований;

АСТПП – автоматизированная система технологической подготовки производства;

САПР – система автоматизированного проектирования;

САК – система автоматизированного контроля;

АСУ ТП – автоматизированная система управления технологическим процессом.

На уровне управления технологическим процессом в свою очередь можно выделить ряд подсистем, осуществляющих распределение заданий и контроль выполнения, локальные системы управления, исполнительные механизмы и информационные датчики

На рис. 1.1 представлена базовая функционально-структурная схема производственной системы (ПС), которая включает в себя технологическую и складскую подсистемы. Организация единого технологического процесса достигается за счет транспортной и автоматизированной информационно-управляющей подсистем. Технологическая подсистема состоит из комплексов основного и вспомогательного оборудования, конкретный состав которого и количество определяются спецификой производства. В состав технологической подсистемы входит гибкие производственные модули ГПМ.

В рамках ГПС принципиально меняется роль и место транспортной и складской подсистем. Из вспомогательных систем традиционно производства в ГПС они переходят в состав основного производства и обеспечивают организацию технологического процесса в целом.

В качестве технологических модулей (ТМ) могут использоваться разнообразные промышленные установки (ПУ), которые представляют собой совокупность промышленного оборудования и системы управления.

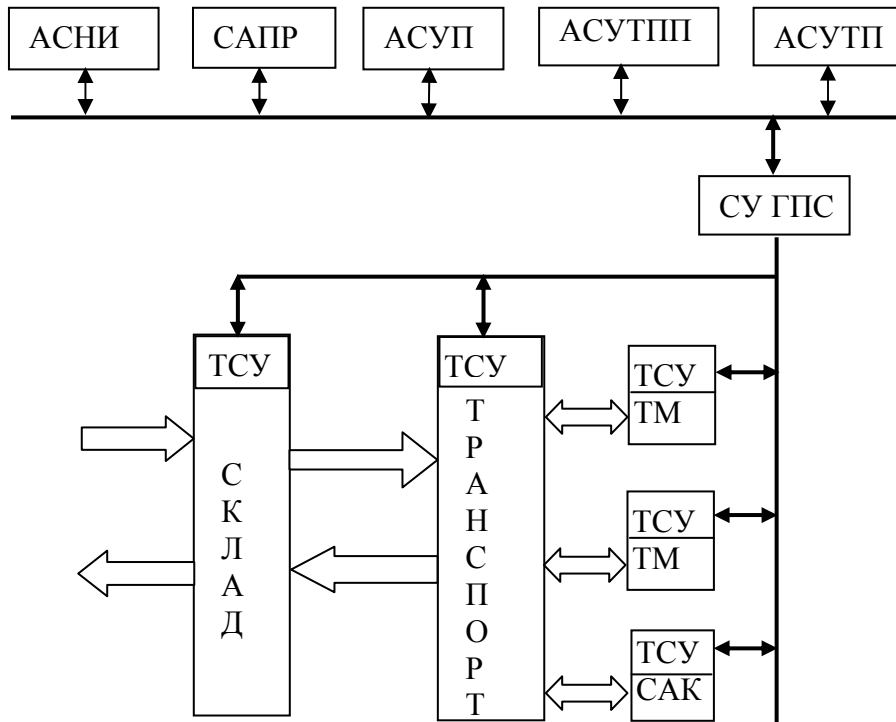


Рис. 1.1. Структурная схема гибкой производственной системы

Терминальные системы управления (ТСУ), реализованные на базе вычислительной техники, как правило, могут функционировать автономно или иметь связи с другими подсистемами. На нижнем уровне управления используется специализированные микропроцессорные средства и программное обеспечение, решающие задачи управления движением рабочих органов, программно-логического управления, ввода и вывода информации, централизованного контроля и регулирования, отображения информации и передачи данных. К ПУ можно отнести основные и вспомогательные подсистемы ПС: станки и станочные системы с ЧПУ, роботы и манипуляторы, автоматизированные транспортные и транспортно-складские системы, а также такие ПУ, как бумагоделательные машины, прокатные станы и другое оборудование.

## 1.1 УРОВНИ И СОСТАВ СИСТЕМ УПРАВЛЕНИЯ ГПС

Интегрированная система автоматизации предприятия может быть представлена в виде нескольких уровней управления.

Нижний, нулевой уровень системы включает набор датчиков и исполнительных устройств, встраиваемых в конструктивные узлы технологического оборудования и предназначенных для сбора первичной информации и реализации исполнительных воздействий. Этот уровень называется уровнем I/O (ввода-вывода).

Следующий, первый уровень служит для непосредственного автоматического управления технологическими процессами с помощью различных устройств сопряжения с объектом (УСО) и промышленных



контроллеров (ПК). Этот уровень получил наименование *control* (непосредственное управление).

Второй уровень, названный SCADA (Supervisory Control and Data Acquisition – сбор данных и диспетчерское управление), предназначен для отображения (или визуализации) данных в производственном процессе и оперативного комплексного управления различными агрегатами, в том числе и с участием диспетчерского персонала.

Третий уровень MES (Manufacturing Execution System) – средства управления производством – выполняет упорядоченную обработку информации о ходе изготовления продукции в различных цехах, обеспечивает управление качеством, а также является источником необходимой информации в реальном времени для верхнего уровня управления предприятием.

И, наконец, четвертый, верхний уровень управления определяется как MRP (Manufacturing Resource Planning) и ERP (Enterprise Resource Planning) – планирование ресурсов предприятия. В России системы этого уровня больше известны под именем АСУП (автоматизированные системы управления предприятием). Они предназначены для автоматизации планирования производства и финансовой деятельности, снабжения и продаж, анализа и прогнозирования и т. д. [4]. Наиболее известные системы этого уровня предлагаются компаниями SAP, Oracle, BAAN и др.

Эту модель комплексной автоматизации предприятия можно упрощать, объединяя любые два соседних уровня, но принципиально подход остается одинаковым.

В соответствии с современной идеологией основные задачи управления решаются на нижних уровнях системы, что позволяет повысить быстродействие системы и разгрузить вычислительную сеть от передачи излишней информации. На верхние уровни управления возлагаются только те задачи, для выполнения которых вычислительные средства нижних уровней не приспособлены, например, отображение текущего состояния автоматизируемого производства.

Система управления ГПС (СУ ГПС) предназначена для обеспечения совместной работы всех подсистем и компонентов ГПС. Данная система должна реализовывать следующие функции: хранение массивов сменно-суточных заданий (ССЗ), учет обеспечения заготовками, инструментом, технологической информацией на сутки; хранение и оперативное обновление данных о текущем состоянии автоматизированных складов; хранение управляющих программ (УП) для станков с ЧПУ (на сутки); вспомогательных технологических подсистем; ввод УП в УЧПУ станков и промышленных роботов (ПР), а также их корректировку или замену в связи с изменившимися условиями работы и ССЗ, обмен информацией с верхним уровнем.

При таком множестве задач СУ ГПС строятся по иерархическому принципу с декомпозицией общей задачи по уровням.

Современные методы проектирования АСУ ТП основаны на использовании систем SCADA (Supervisory Control And Data Acquisition),

которые представляют собой совокупность аппаратно-программных средств, обеспечивающих возможность мониторинга, анализа и управления параметрами технологического процесса человеком. Среда визуального проектирования и управления производственными процессами данных систем позволяют значительно сократить сроки проектирования и внедрения систем автоматизации производства.

Так, например, система Scada InTouch имеет открытый и расширяемый интерфейс, что позволяет обеспечить широкие возможности взаимодействия с множеством устройств промышленной автоматизации. Пользователь может создавать изображения в графическом редакторе Wonderware WindowMaker™ при помощи разнообразных инструментов: стандартных графических компонентов, растровых изображений (bitmap), элементов управления ActiveX, улучшенной графической библиотеки Symbol Factory, которая включает тысячи уже созданных изображений для промышленности и новой библиотеки графических символов SmartSymbols. Программное обеспечение предлагает широкий выбор инструментальных средств, серверов ввода-вывода, машину сценариев, а также возможности переключения из среды разработки в среду исполнения. Программное обеспечение InTouch позволяет создавать как отдельные автономные сетевые приложения, так и распределенные приложения с возможностью расширения до сотни узлов. Динамическая система разработки сетевых приложений (NAD) облегчает централизованное обслуживание эталона приложения Scada InTouch, используя один сетевой сервер. Каждый клиентский узел делает локальную копию основного приложения, что обеспечивает надежное резервирование. В случае, если сервер недоступен, клиентский узел продолжает функционировать, используя свои локальные копии приложения.

Общая функциональная схема современного автоматизированного производства представлена на рис. 1.2. Нижний уровень этой схемы составляют измерительные приборы и исполнительные механизмы [5]. Они могут быть аналоговыми или цифровыми (интеллектуальными). Для обмена информацией с приборами первого типа необходимо использовать АЦП и ЦАП (аналогово-цифровые и цифро-аналоговые преобразователи). С приборами второго типа можно обмениваться информацией непосредственно по сети передачи данных.

Следующий уровень схемы – контроллеры. Они выполняют функцию автоматического управления технологическим процессом. Управление исполнительными механизмами осуществляется по определенным алгоритмам путем обработки данных о состоянии технологических параметров, полученных посредством измерительных приборов.

Серверы технологических данных обеспечивают обмен информацией между технологическими устройствами и сетью персональных компьютеров. Они поддерживают протокол работы с технологическими устройствами и протокол работы с сетью персональных компьютеров.

Данные о текущих параметрах технологического процесса могут быть использованы для контроля состояния технологического процесса и управления

им с автоматизированных рабочих мест операторов; архивирования истории изменения технологических параметров; формирования суммарных отчётных форм с целью предоставления информации руководящему персоналу.



Рис. 1.2. Функциональная схема современного производства

В этой схеме SCADA система представлена серверами технологических данных и автоматизированными рабочими местами операторов.

Отметим функции SCADA систем:

сбор, первичная обработка и накопление информации о параметрах технологического процесса и состоянии оборудования от промышленных контроллеров и других цифровых устройств, непосредственно связанных с технологической аппаратурой;

отображение информации о текущих параметрах технологического процесса на экране ПЭВМ в виде графических мнемосхем;

отображение графиков текущих значений технологических параметров в реальном времени за заданный интервал;

обнаружение критических (аварийных) ситуаций;

обнаружение критических (аварийных) ситуаций;

вывод на экран ПЭВМ технологических и аварийных сообщений;

архивирование истории изменения параметров технологического процесса;

операторское управление технологическим процессом;  
предоставление данных о параметрах технологического процесса для их использования в системах управления предприятием.

## 1.2. КОМПЛЕКСНЫЕ СРЕДСТВА АВТОМАТИЗАЦИИ

Комплексные решения в производстве средств автоматизации предлагает ряд фирм [6, 7]. Автоматизированная система управления технологическими процессами АСУ ТП (Octagon Systems) показана на рис.1.3. В состав предлагаемого оборудования входят:

- PC-совместимые контроллеры Octagon под управлением SCADA-системы;
- измерители-регуляторы;
- высоконадежные сервера SCADA/HMI;
- автоматизированные рабочие места (АРМ) на базе ПК;
- встроенные панели управления с установленными графическими консолями.

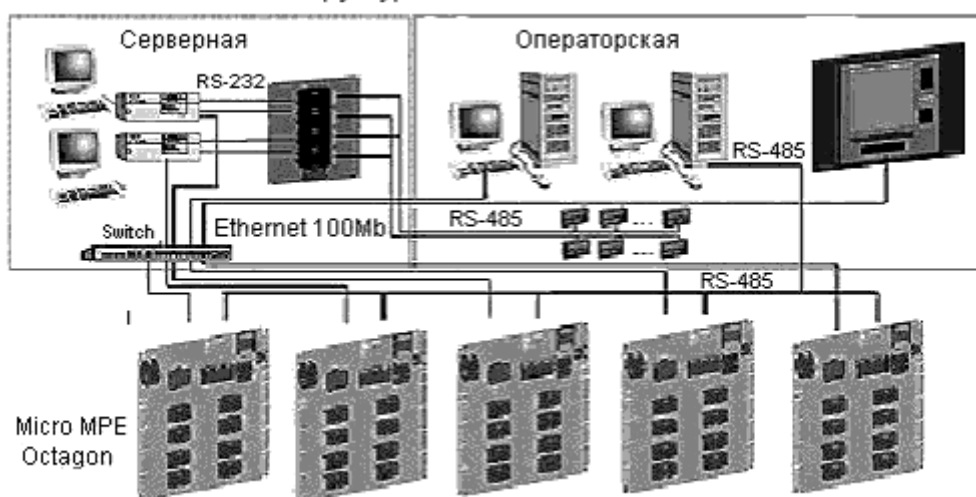


Рис. 1.3. Структурная схема АСУ ТП фирмы Octagon Systems

Человеко-машинный интерфейс (HMI) от компании Wonderware предназначен для управления технологическими процессами и диспетчерского контроля, поддерживает множество протоколов обмена с различными устройствами автоматизации посредством одного или нескольких драйверов ввода/вывода в реальном времени.

Вариант автоматизации технологических процессов предприятия фирмы Siemens показан на рис. 1.4.

Основное оборудование данного варианта составляют контроллеры семейства SIMATIC, которые широко используются в различных отраслях промышленности и имеют специализированный язык STEP, доступный для

широкого круга пользователей. Для построения интегрированных комплексов здесь предлагается использовать Ethernet и промышленную сеть Fieldbus, что позволяет обеспечить:

- высокую степень защиты при передаче информации;
- стандартную структуру сообщений;
- возможность подключения и отключения отдельных модулей/узлов при работе сети;
- наличие топологий с повышенной надежностью функционирования;
- наличие электрических, оптических и инфракрасных каналов связи;
- возможность использования сетевых компонентов других производителей;
- простую архитектуру сети.

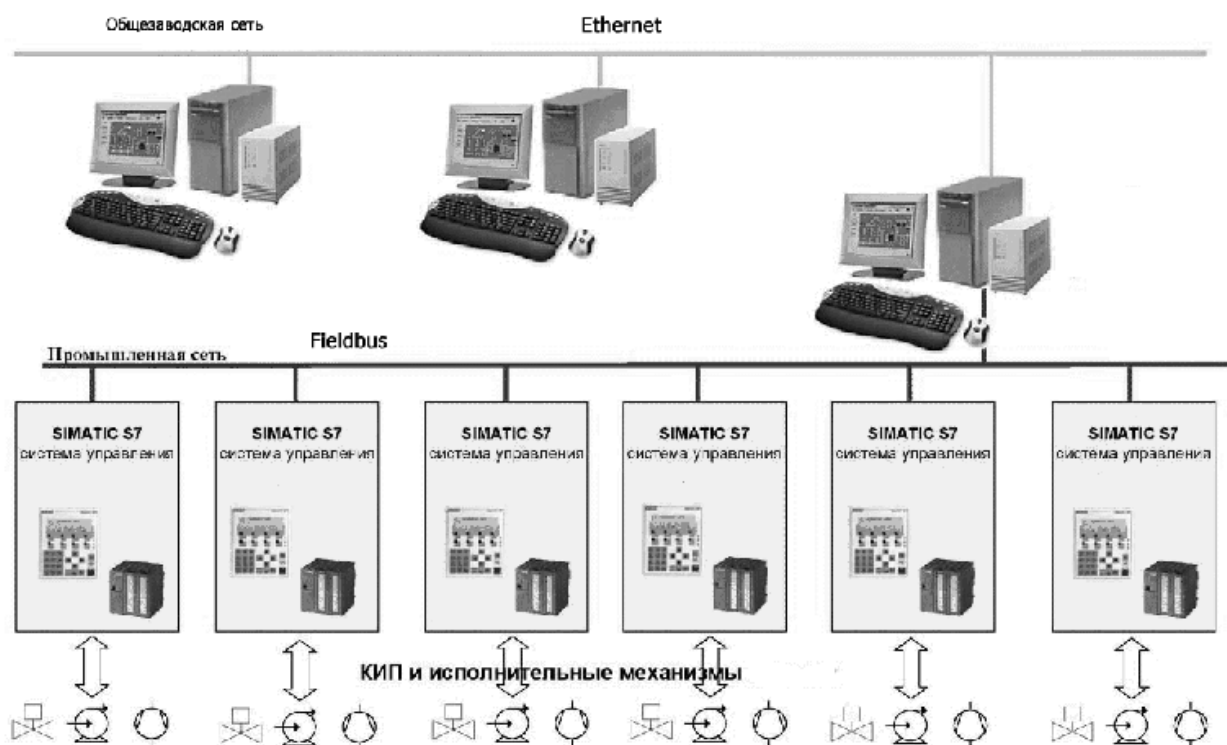


Рис. 1.4. Структурная схема автоматизации фирмы Siemens

Основные требования, предъявляемые к первому уровню (управления агрегатом) характеризуется следующими показателями:

- необходимостью обеспечения работы системы управления в режиме реального времени;
- предельной надежностью (на уровне надежности основного оборудования);
- возможностью встраивания в основное оборудование;
- функциональной полнотой модулей УСО;
- возможностью автономной работы при отказах комплексов управления верхних уровней;
- возможностью функционирования в цеховых условиях.

В промышленные контроллеры загружаются программы и данные из ЭВМ второго уровня, обеспечивающие координацию и управление агрегатом по критериям оптимальности управления технологическим процессом в целом. Для контроля состояния агрегата и технологического процесса на втором уровне управления выполняется вывод служебной, диагностической и оперативной информации.

Особенность обмена информацией между первым и вторым уровнями состоит в высокой степени регулярности. Необходимость обмена информацией в темпе реального процесса накладывает достаточно жесткие ограничения на этот режим. В данном случае применимы режимы обмена, соответствующие локальным промышленным сетям, которые в настоящее время выполняются в стандартах Bitbus, Profibus и т.п. Первый уровень управления реализуется, например, на промышленных контроллерах СМ 1820М ПК, Allen-Bradley, ЭМИКОН, МИК, СИКОН и др.

Второй уровень управления должен обеспечивать:

диспетчерское наблюдение за технологическим процессом по его графическому отображению на экране в реальном масштабе времени;

расчет и выбор законов управления, настроек и уставок, соответствующих заданным показателям качества управления и текущим (или прогнозным) параметрам объекта управления;

хранение и дистанционную загрузку управляющих программ в ПК;

оперативное сопровождение моделей объектов управления типа «агрегат», «технологический процесс», корректировку моделей по результатам обработки информации от первого уровня; синхронизацию и устойчивую работу систем типа «агрегат» для группового управления технологическим оборудованием;

ведение единой базы данных технологического процесса (реальное время);

конфигурацию комплекса для выбранного режима работы (в том числе переход на резервную схему);

связь с третьим уровнем.

Необходимо отметить, что в металлорежущих станках с программным управлением первые два уровня управления совмещены как по задачам, так по составу используемых средств вычислительной техники и, по сути, представляют АРМ оператора-станочника. Подходы к проектированию участков станочного оборудования также имеют свои особенности, связанные с возможностью автономной работы станка с программным управлением.

## 2. СИСТЕМЫ ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

В настоящее время станок с числовым программным управлением (ЧПУ) является основным производственным модулем современного производства. Станки с ЧПУ используются как для автоматизации мелкосерийного или штучного производства, так и для производства больших серий.

Процесс развития систем числового программного управления существенным образом связан с общими тенденциями разработки новых поколений микропроцессоров, каналов передачи информации и программного обеспечения. Вследствие процессов, происходящих в экономике России в последнее время, определяющим стало применение импортной микропроцессорной и компьютерной техники. Спад производства базовых отраслей отечественной промышленности привел к существенному отставанию машиностроения, в том числе, производства станков с ЧПУ. В действующем производстве до сих пор работают значительное количество станков со старыми моделями устройств числового программного управления (УЧПУ).

На рис. 4.1 показана типичная структурная схема устройства числового программного управления старого поколения типа 2С-42-65. Устройство является контурно-позиционным со свободным программированием алгоритмов. Количество управляемых координат – до 8. Одновременное управление при линейной интерполяции обеспечивается по 4-м координатам, а при круговой интерполяции – по 2-м координатам.

Одноплатная микроЭВМ МС 12.02 реализована на базе процессора 1801ВМ02. Обмен информацией между микро ЭВМ и внешними устройствами осуществляется по каналу ЭВМ типа «Общая шина». Для увеличения нагрузочной способности и емкости памяти используется расширитель канала (РК).

Конструктивно ЧПУ содержит 2 корзины. Одна из них предназначена для установки блоков общесистемного пользования, а вторая предназначена для установки специальных блоков для управления станком.

На станочной магистрали находятся блоки входных и блоки выходных сигналов, с помощью которых реализуется программная реализация задач логического управления. Это функции управления – электроавтоматики станка. Формирование аналоговых сигналов управления приводами подач и главного движения осуществляется через цифроаналоговые преобразователи (ЦАП) – группа «Привод». Для реализации обратных связей по положению используются преобразователи фаза-код (ПФК), составляющие группу «Датчики». Для решения задач адаптивного управления (например, систем стабилизации мощности резания) могут быть использованы аналого-цифровые преобразователи (АЦП) – группа «Адаптивное управление».

Пульт управления (ПУ) содержит набор алфавитно-цифровых клавиш, с помощью которых можно осуществлять ввод управляющей программы. Кроме того, имеются функциональные клавиши, с помощью которых задается режим работы УЧПУ и определяются специальные функции, соответствующие поиску, редактированию управляющих программ. Пульт коррекции (ПК) представ-

ляет собой набор декадных переключателей, с помощью которых можно осуществлять изменение значений скорости подачи и скорости вращения главного движения в процентном соотношении.

Для отображения текущего значения координат и технологических параметров используется алфавитно-цифровой дисплей – блок отображения символической информации (БОСИ). Для ввода и вывода управляющей программы могут быть использованы фотосчитывающее устройство (ФСУ) и ленточный перфоратор (ПЛ). В качестве носителя информации в этом случае используется перфолента. Другой вариант ввода-вывода информации основан на использовании канала последовательной связи (ИРПС – интерфейс радиальной последовательной связи). Для увеличения быстродействия используют аппаратный блок умножения (БУ) и блок преобразования кодов (БПК).

Базовое программное обеспечение УЧПУ записывается в постоянное запоминающее устройство (ПЗУ) и представляет собой набор подпрограмм, реализующих так называемые подготовительные G и вспомогательные функции M, а также сервисные функции по отработке управляющей программы. Управляющая программа представляет собой последовательность кадров, определяющих траекторию движения инструмента. В кадре с помощью G и M-функций определяется тип интерполяции (линейная, круговая), перемещения по координатам, скорости подач и частоты вращения привода главного движения, тип и коррекция на вылет режущего инструмента и другая информация, определяющая работу на участке траектории.

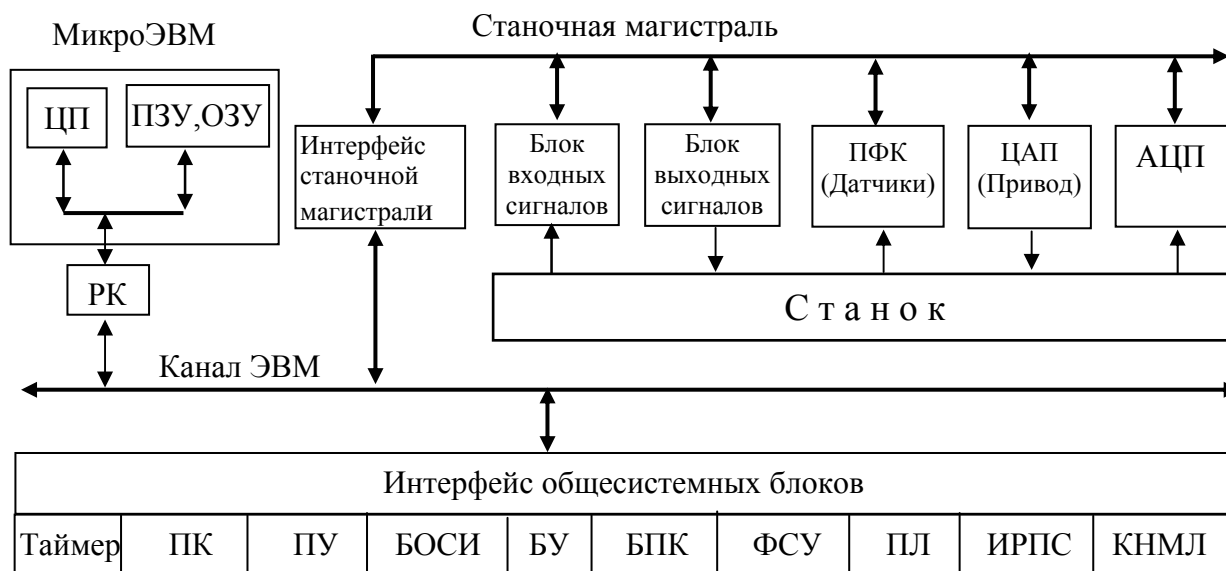


Рис. 2.1. Структурная схема УЧПУ 2С-42-65

Рассмотрим отработку управляющей программы с точки зрения функционирования и использования блоков УЧПУ.

Основное машинное время при отработке кадра затрачивается на расчет траектории движения инструмента. Движение по траектории в общем случае включает в себя участки разгона и торможения. Согласование движения по координатам и формирование задающих воздействий осуществляется программ-



ным интерполятором, который разворачивает требуемую траекторию во времени по прерываниям от таймера. Обработка этой траектории осуществляется следящими приводами подачи. Сигнал ошибки по положению формируется программным способом, а затем выдается через ЦАП в качестве сигнала управления скоростью электропривода. Привод подачи (главного движения) при этом представляет собой автономное устройство, которое должно быть замкнуто обратной связью по скорости. Работа интерполятора должна осуществляться в реальном масштабе времени. При использовании численных методов интегрирования шаг интегрирования определяется периодом прерывания от таймера. Для обеспечения частоты среза приводов порядка 50 Гц прерывания от таймера должны производиться на частоте не менее 100 Гц.

Во время обработки текущего кадра в фоновом режиме происходит подготовка информации для следующего кадра. Этот этап называется «Интерпретация кадра». Он включает в себя преобразование символьной информации в числовую. Числовая информация вводится в десятичной системе счисления. Вначале символьная информация преобразуется в двоично-десятичную систему, а затем с помощью БПК – в двоичную. Аналогичная задача преобразования информации возникает и в каналах обратной связи по положению. Контроль положения осуществляется в двоично-десятичном коде. Для согласования информации с преобразователя фаза-код преобразуется к машинному (двоичному) представлению.

При выводе информации возникает обратная задача – преобразование двоичной информации в двоично-десятичные числа, а затем в символьное представление.

Элементная база рассмотренного УЧПУ устарела, особенно это относится к устройствам ввода-вывода информации, носителям информации (перфоленте и магнитной ленте), используемым микросхемам памяти оперативных и постоянных запоминающих устройств

## 2.1. МОДЕРНИЗАЦИЯ СТАНКОВ С ЧПУ

Ведущие фирмы постоянно совершенствуют и расширяют возможность систем ЧПУ, систем подготовки данных и проектирования. Вместе с тем, учитывая значительный парк систем ЧПУ старого поколения, значительное внимание уделяется вопросам модернизации и создания систем передачи данных.

Модернизация отечественных станков с ЧПУ в настоящее время производится по двум основным направлениям: путем полной замены старых моделей устройств ЧПУ на новое поколение зарубежных фирм; модернизацией существующих устройств, за счет замены основных блоков, разработанных на новой элементной базе.

Один из вариантов модернизации систем ЧПУ 2С42-61, 2С42-65 и 2Р22, используемых для управления фрезерными, горизонтально-расточными, координатно-расточными и токарными станками, основан на замене микро-ЭВМ «Электроника-60» одноплатным микроконтроллером «Кварц-60.1» [8].

Микроконтроллер (рис. 2.2) заменяет в УЧПУ всю корзину ЭВМ и ряд суб-блоков:

- одноплатную микроЭВМ НМС 201.2;
- ЗУ с автономным питанием (1 плата SB-935 на 8 Кбайт);
- программное ПЗУ (3 платы SB-978 по 16Кбайт);
- страничное ОЗУ (1 плата SB-067 на 32 Кбайт);
- плату канала SB-473 из конструктива микроЭВМ (1 плата);
- плату расширителя канала SB-075 (1 плата);
- плату таймера SB-445 (1 плата);
- плату уставок SB-454 (1 плата);
- плату формирователя служебных сигналов SB-086 (1 плата).

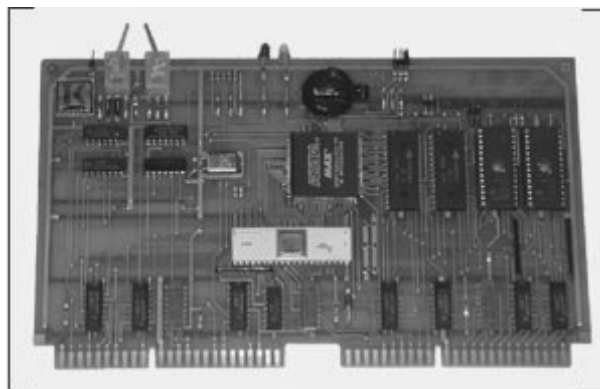


Рис. 2.2. Микроконтроллер «Кварц-60.1»

Микроконтроллер выполнен на базе микропроцессора 1801ВМ2 и программируемой логической интегральной схеме (ПЛИС) фирмы Altera в габарите одной платы стойки ЧПУ. Микроконтроллер имеет стандартные размеры и разъемы, соответствующие стойке ЧПУ. Информационная емкость ОЗУ увеличена до 64 Кбайт (2-е микросхемы типа 62256), емкость ППЗУ увеличена до 128 Кбайт (2-е микросхемы с ультрафиолетовым стиранием типа 27C512 или 27C256). Могут быть использованы и другие варианты организации страничной памяти ППЗУ.

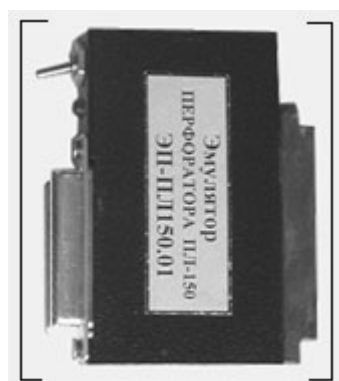
Для замены морально устаревших устройств ввода/вывода информации на бумажных и магнитных носителях используются модули электронной памяти – электронные картриджи. Подключение модуля памяти производится через штатный разъем блока, предназначенного для включения ФСУ (перфоратора) и не требует доработки УЧПУ. При этом устройство управления модулем памяти типа EEPROM имитирует работу заменяемого устройства.



Эмулятор  
ФСУ «Консул»  
для УЧПУ 2С42 (2Р22)



Модуль памяти  
MEM 32K



Эмулятор перфоратора "ПЛ150"  
для УЧПУ 2С42 (2Р22)

Рис. 2.3. Эмуляторы устройств ввода/вывода информации

Эмулятор ФСУ «КОНСУЛ 337.601» предназначен для ввода информации из модуля памяти MEM в УЧПУ через интерфейс фотосчитывающего устройства субблок SB466 УЧПУ 2С42 (2Р22). Эмулятор перфоратора «ПЛ150» предназначен для вывода информации из УЧПУ в модуль памяти через интерфейс перфоратора ПЛ150 – субблок SB-465 УЧПУ 2С42 (2Р22). Оба устройства конструктивно представляют собой блок размером 60×80×15 мм с двумя разъемами для подключения к плате SB466 (SB465) и к модулю памяти MEM.

В цехах основного производства использование физических носителей существенно снижает надежность и эффективность станков с ЧПУ. Для повышения мобильности, надежности и сокращения затрат при производстве изделий сложной номенклатуры используют DNC (Distributed Numerical Control) – распределительные системы числового программного управления станками. DNC является промежуточным этапом создания ГАП, с помощью данных систем осуществляется ведение архива управляющих программ (УП) на персональной ЭВМ и одноразовая их передача в УЧПУ по запросу оператора конкретного станка.

Персональная ЭВМ типа IBM PC/AT содержит мультиплексор связи, обеспечивающий подключение к ЭВМ группы станков через последовательный порт. Варианты мультиплексоров УМС-6, УМС-14. Мультиплексоры могут устанавливаться возле станков.

Используются следующие типы контроллеров для подключаемого станка:

ИМП-51 для станков с магнитной лентой, оснащенных пультами ПФСТЭ-12-500;

КМС-01 для станков с УЧПУ класса NC с вводом с перфоленты (Н33, Н22 и др.);

МК-51Р для станков с УЧПУ класса CNC с вводом с перфоленты (2С42, КМ43 и др.);

МК-51S для станков с УЧПУ класса CNC с вводом и выводом через RS-232 (Bosch, TNC, Fanuc);

МКНЦ-31 для станков с УЧПУ НЦ31 с вводом и выводом через порт КВП.

Другой вариант модернизации системы ЧПУ основан на замене микро-ЭВМ персональным компьютером. Согласование со станочной магистралью в этом случае осуществляется с помощью специализированной платы преобразования сигналов шины ISA в сигналы магистрали типа «Общая шина». В качестве комплектующих используются материнская плата с IBM-совместимым процессором, обязательно наличие свободного ISA слота, винчестер, гибкий дисковод, видео карта SVGA, компьютерный источник питания. В качестве видеотерминала используют обычный монитор или плоский и IBM-совместимую клавиатуру. В свободный ISA-слот материнской платы устанавливается плата интерфейса станочной магистрали и блока корректоров. Привязка к станку осуществляется с использованием программного обеспечения FMS3000-2С42 с помощью системы параметров и встроенного языка электроавтоматики. Рассмотренный вариант модернизации не требует перемонтажа

станка и замены датчиков обратной связи, сохранение интерфейса станочной магистрали и основных ее блоков дает известные преимущества в ремонте системы ЧПУ для обслуживающего персонала.

## **2.2. КОНЦЕПЦИИ РАЗРАБОК И СТРАТЕГИЯ РАЗВИТИЯ СРЕДСТВ ЧПУ**

Одну из наиболее продуманных стратегий разработки и выпуска средств автоматизации производства осуществляет фирма Сименс, решая задачи комплексного совершенствования своей продукции [9]. Одна из концепций этой стратегии – разработка систем ЧПУ – неразрывно связана с совершенствованием регулируемого электропривода, придания ему новых качеств за счет цифрового управления.

Учитывая разнообразного потребителя, спрос на самые простые, маленькие станки, которые выгодно использовать небольшим фирмам, предлагается семейство ЧПУ для простых станков (2 оси + шпиндель для токарных и 3 оси + шпиндель для фрезерных станков). В качестве приводов могут быть использованы как шаговые двигатели (ЧПУ SINUMERIK 802S), так и сервопривода с аналоговым интерфейсом (ЧПУ SINUMERIK 802C). Использование ЧПУ в данном случае с одной стороны позволяет поддерживать постоянный уровень качества и избавляет от проблемы поиска высококвалифицированных специалистов. Другая возможность при наличии высококвалифицированных специалистов связана с использованием специальных пультов управления, позволяющих осуществлять обработку в режиме ручного управления с помощью штурвалов. Наличие специального программного обеспечения при этом позволяет обеспечить одновременное формирование управляющей программы и использование ее в дальнейшей работе.

Для более сложных станков фирма Сименс предлагает СЧПУ SINUMERIK 810D с процессорами CCU1, CCU2 и SINUMERIK 840D с процессорами NCU571, NCU 572. Эти системы позволяют осуществлять управление от 6 до 12 осей и имеют в своем составе функции, позволяющие управлять не только токарной или фрезерной обработкой, но и такими технологиями, как шлифование, лазерная резка, вырубка и управление роботами. Эти системы работают совместно с цифровым приводом SIMODRIVE 611D, что позволяет достичь точности и динамики, превосходящих соответствующие показатели традиционного аналогового привода.

Отдельной линией проходит система ЧПУ SINUMERIK 840Di. Это так называемая система ЧПУ, базирующаяся на стандартном промышленном ПК. При этом интерфейс пользователя и ядро ЧПУ реализованы программно, а контроллер ПЛК и интерфейсы подключения периферии и приводов расположены на единой слот-плате. В качестве привода используется тот же универсальный привод SIMODRIVE 611U, что и для SINUMERIK 802D.

Тенденции широкого использования промышленных компьютеров в СЧПУ позволили в некотором смысле отечественным производителям преодо-

леть существенное отставание в производстве современных систем числового программного управления [10]. Одной из таких разработок является система числового программного управления серий FMS-3000 и FMS-3100.

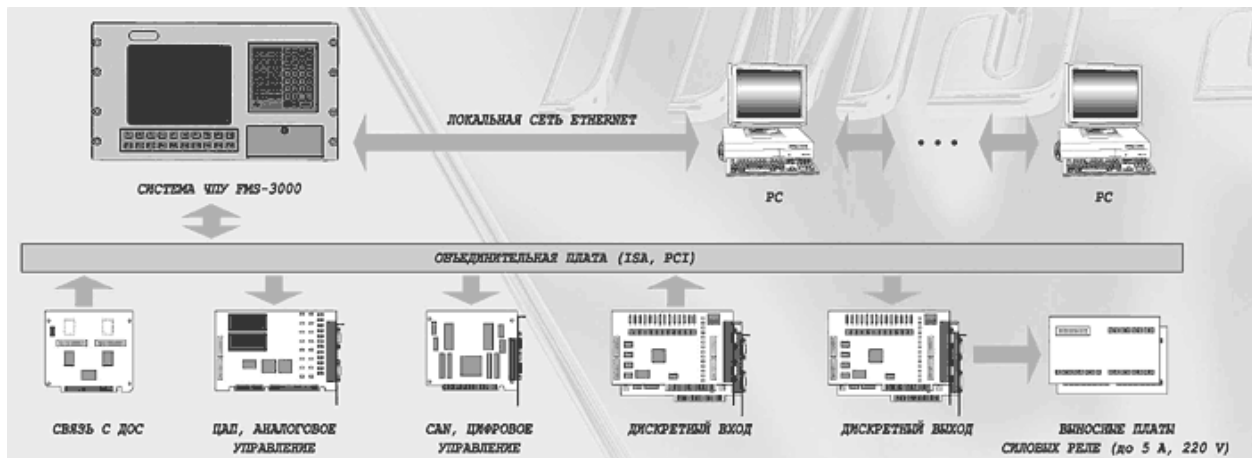


Рис. 2.5. Система числового программного управления на базе ПК

FMS-3000 и FMS-3100 предназначены для установки на различные типы технологического оборудования.

FMS-3000: фрезерные станки; расточные станки; копировальные станки; обрабатывающие центры; машины термической резки; другое оборудование с ЧПУ.

FMS-3100: намоточные станки; токарные станки; токарно-карусельные станки; другое оборудование с ЧПУ.

Адаптация к конкретному типу станка осуществляется с помощью программы электроавтоматики и разветвленной системы параметров **FMS-3000, FMS-3100**.

Основа FMS-3000 – персональный компьютер промышленного исполнения (ПК). Открытая архитектура ПК позволяют легко интегрировать в ее состав компьютерные компоненты ведущих производителей вычислительной техники. К числу стандартных компонентов промышленных компьютеров относятся: платы цифрового ввода/вывода, платы АЦП и ЦАП, платы программируемых счетчиков, платы релейных выходов и изолированных входов, плата интерфейса CAN, плата интерфейсов RS-232/ «токовая петля», панели для установки твердотельных реле и модулей ввода/вывода, модули преобразователя RS-232 в RS-422/485, модуль подключения RS-232 к оптоволоконному кабелю и т. д.

Доминирующие позиции открытых систем ЧПУ, построенных на базе персонального компьютера (PCNC – Personal Computer Numerical Control) – для условий промышленного производства заняли фирмы Advantech, Octagon Systems.

### 2.3. ОБЩАЯ ХАРАКТЕРИСТИКА СИСТЕМ ЧПУ WINPCNC

Система WinPCNC является одно-компьютерной системой ЧПУ, построенной на мощной платформе персонального компьютера с операционной системой Windows NT и расширением реального времени RTX 4.1 фирмы VentureCom. Она относится к классу PCNC (Personal Computer Numerical Control), т. е. к классу так называемых «персональных систем управления», который является сегодня наиболее перспективным классом систем ЧПУ нового поколения.

Система использует единственный процессор для обслуживания всех ее функций, включая функции электроавтоматики. Аппаратная часть представлена стандартной аппаратурой персонального компьютера и дополнительными интерфейсными модулями для связи со следящими приводами подачи и главного движения, платами электро-автоматики, панелью оператора. Общий вид системы ЧПУ WinPCNC показан на рис. 2.4.



Рис. 2.4.Общий вид ЧПУ WinPCNC

Система ЧПУ состоит из двух независимых блоков, системного (на рисунке снизу) и панели оператора.

Системный блок размещен в корпусе промышленного компьютера, который гарантирует защиту от производственных помех. Блок панели оператора должен быть встроен в свой корпус и конструктивно оформлен так, как это удобно потребителю. Системный блок показан в раскрытом виде на рис. 2.5 и представляет собой базовый набор плат на шине ISA или PCI. Существует также возможность установки дополнительных плат, например, для увеличения числа управляемых координат, увеличения общего числа параллельных входов-выходов электроавтоматики, для установки сетевых интерфейсных плат типа Fieldbus с целью управления приводами и электроавтоматикой. Дополни-

ные платы определяют опции системы ЧПУ, которые зависят от конкретного заказа.



Рис. 2.5. Системный блок ЧПУ

Основной особенностью системы WinPCNC на прикладном уровне является ее открытая архитектура, которая предоставляет производителям технологического оборудования и конечным пользователям широкие возможности по адаптации системы к собственным требованиям. Эти возможности поддержаны средствами конфигурации, а также дополнительными инструментальными системами. Так, интерфейс оператора открыт для включения любых приложений конечного пользователя, разработанных в среде Windows. При этом конечным пользователям предоставлен широкий открытый набор интерфейсных функций API (Application Programming Interface). Интерфейс оператора (его внешний вид, набор режимов, страниц и меню) может быть в кратчайший срок существенно перепроектирован с помощью специальной инструментальной системы State\_Machine\_Builder. Система ЧПУ может быть настроена на любую версию языка управляющих программ в коде ISO-7bit (DIN 66025). Настроенная версия поддерживается редактором-дебаггером управляющих программ и гибким интерпретатором управляющих программ (ISO-процессором). Интерпретатор может быть настроен на любое (разумное) число буферируемых кадров. Редактор-дебаггер располагает системой графического моделирования управляющей программы на входе системы и на выходе интерпретатора, т. е. с учетом эквидистант. Интерполятор системы имеет гибкую собственную архитектуру и фиксированный входной интерфейс IPD (Interpolator Data); он открыт для включения любых специальных алгоритмов интерполяции. В процессе управления может быть использована любая комбинация имеющихся алгоритмов. Гибкая система электроавтоматики построена на основе концепции SoftPLC, т.е. на основе программной реализации контроллера в составе прикладной компоненты системы ЧПУ. Взаимодействие всех модулей системы ЧПУ осуществляется через многофункциональную программную шину, которая служит глобальным сервером системы. Подобная организация допускает масштабирование системы.

Технические характеристики системы WinPCNC:

Тип компьютера и операционная система: промышленное исполнение; процессорный модуль Pentium-III, 750 МГц, ОЗУ – 512 Мбайт, HDD – 20 Гбайт, FDD, CDROM. Экран монитора – TFT. Операционная система Windows NT+RTX.

Режимы работы: автоматический, ручного управления; редактирования, отладки и моделирования управляющих программ; системной настройки.

Управляющая программа и стандартные циклы – в версии заказчика.

Число управляемых и интерполируемых координат – до 8-ми. Интерполируемая скорость – до 10 м/мин. Цена дискретности для тонкой интерполяции – 1 мкм.

Виды интерполяции: линейная, круговая, сплайновая. Сплайновую интерполяцию можно программировать непосредственно в кадре управляющей программы. Автоматические разгоны и торможения.

Управление приводами: +/- 10 В. Тип позиционного датчика обратной связи – импульсный. Возможно ручное управление с помощью маховичка ручного перемещения.

Электроавтоматика – типа SoftPLC. Связь с приводами электроавтоматики – через параллельный порт или по шине CANBus. Число входов/выходов – по заказу.

## **2.4. АППАРАТНЫЕ И ТЕХНОЛОГИЧЕСКИЕ ВОЗМОЖНОСТИ УЧПУ SINUMERIK 810D, 840D**

УЧПУ SINUMERIK 810D, 840D представляют собой интегрированные мультипроцессорные системы, в которых нельзя выделить ЧПУ и приводы в самостоятельные устройства.

Процессорный модуль ЧПУ SINUMERIK имеют линейку процессорных модулей, которые называются либо CCU (Compact Control Unit) для 810D типа CCU1 и CCU2, либо NCU (Numerical Control Unit) для 840D с типами от 571.2 до 573.2. Основные аппаратные характеристики процессоров ЧПУ приведены в таблице 2.1.

Пульт оператора состоит из следующих частей:

ММС-процессора (Human Machine Communication). Он представляет собой персональный компьютер MMC100.2 (Intel 486, MS-DOS) или MMC103 (Intel Pentium, Windows 95);

дисплея, который может быть цветным или монохромным (10,4" TFT плоский экран).

Кроме того, пульт может быть оснащен полноразмерной клавиатурой, дисководом, адаптером подключения к сети Ethernet. Весь интерфейс полностью русифицирован.

На ММС-процессоры можно установить дополнительное программное обеспечение фирмы Siemens (например, ShopMill – для экспресс-подготовки управляющих программ на рабочем месте, SINTDI – для управления инструментом, WinBDE – для сбора и анализа информации о работе станка) или третьих фирм.



Таблица 2.1

**Основные аппаратные характеристики УЧПУ**

Функции	SINUMERIK FM-NC	SINUMERIK 810DE	SINUMERIK 810D	SINUMERIK 840DE			
	NCU 570	CCU1	CCU2	NCU 571.2	NCU 572.2	NCU 573.2 12 осей	NCU 573.2 31 ось
Количество управляемых осей/шпинделей в базовом исполнении	4/2	5/2	5/2	5	5	5	5
Максимально возможное количество осей + шпинделей	5	5	5	6	10	12	31
Количество независимо от-рабатываемых программ в базовом исполнении	1	1	2	1	1	1	1
Максимальное количество независимо отработываемых программ	1	1	2	2	2	2	10
Расширение групп режимов работ (до)	—	—	2	—	—	2	10
Память пользователя для программ и данных	128 KB	256 KB	1,5 MB	256 KB	256 KB	2,5 MB	2,5 MB
Расширение памяти пользо-вателя в ЧПУ — с дискрет-ностью 256 Кбайт до 1,5 Мбайт	—	о	—	о	о	—	—
Память пользователя для программ и данных на жест-ком диске MMC 103 (1Гбайт)	+	+	+	+	+	+	+

В системах ЧПУ SINUMERIK 810D, 840D используются цифровые приво-ды (D – digital), в которых сигнал от ЧПУ передается по специальной цифро-вой шине. В каждом модуле привода имеется процессор, который выполняет задачи по управлению приводом и разгружает центральный процессор ЧПУ для других целей.

**Основные характеристики цифровых приводов:**

минимальное время (период квантования), через которое производится контроль положения – 0,125 мс;

разрешающая способность – 4,2 млн импульсов на один оборот двигате-ля;

диапазон регулирования скорости примерно в 50 раз больше по сравнению с аналоговыми приводами;

высокие динамические и статические характеристики.

Цифровое управление приводами позволяет повысить производительность станка и улучшить качество детали. Кроме того, имеются дополнительные сервисные возможности:

настройка привода через параметры, вводимые через ММС-процессор;

автоматическая оптимизация приводов, позволяющая более точно и быстро адаптировать приводы к механике станка;

представление информации о состоянии привода (температура, нагрузка и т. д.).

**Технологические возможности.** Хотя за последние годы язык программирования для УЧПУ претерпел серьезные изменения, однако остается преемственность программного обеспечения в виде набора базовых функций. Большинство программ, написанных для старых моделей УЧПУ, работают и с новыми моделями при минимальных переделках. Имеющиеся сегодня возможности визуального программирования и целый ряд высокоуровневых функций обеспечивают такие возможности, о которых раньше технолог мог только мечтать.

Мнемоника команд стала более ясной и читабельной. Базовый набор технологических функций УЧПУ позволяет использовать его с широкой гаммой станков (токарные, фрезерные, шлифовальные и т. п. станки и обрабатывающие центры). Остановимся на некоторых наиболее интересных из них.

**Сплайны и полиномы.** Эти функции позволяют создавать плавные непрерывные кривые. В системе возможно использование трех видов сплайнов (А, В, С) и кривых, заданных с помощью полиномов третьего порядка. Главное назначение сплайнов – быть интерфейсом между CAD/CAM-системами и УЧПУ. Основная область применения – производство штампов и пресс-форм. Использование сплайнов в обработке позволяет сократить управляющую программу, улучшить динамику движения приводов, повысить качество обрабатываемых поверхностей, отказаться от ручной доводки пресс-форм.

**Изменение величины подачи по заданному закону в пределах одного кадра.** Данная функция позволяет адаптировать величину подачи к режимам резания (например, изменять подачу при сверлении отверстия). Закон изменения определяется тремя способами: FLIN – линейное изменение, FCUB – по плавной кривой, FPO – по закону, заданному с помощью полинома.

**Автоматическое предотвращение зарезов.** Возможность обработки «узких мест» (карманов, пазов и т. п.) зависит от диаметра инструмента. При использовании опции Look Ahead происходит опережающий просмотр программы (до 1000 кадров). Система позволяет автоматически отслеживать подобные ситуации и предотвращать зарезы.

**Преобразование рабочей системы координат в пространстве.** Базовый набор команд (концепция FRAME) позволяет производить с рабочей системой координат следующие преобразования: TRANS – линейный сдвиг; ROT – на-

клон в заданной плоскости или вращение всей плоскости обработки; MIRROR – зеркальное отображение заданных осей, SCALE – масштабирование вводимых размеров. Используя эти возможности, можно исключить пересчет координат и задавать размеры прямо с исходного чертежа.

**Ориентация инструмента относительно плоскости обработки.** При наличии на станке поворотной инструментальной головы при условии 2,5D-обработки значительно упрощается обработка наклонных поверхностей. После задания соответствующей команды ориентации инструмента к плоскости обработки все дальнейшие действия по программированию перемещений производятся так, словно продолжается работа в стандартной системе координат.

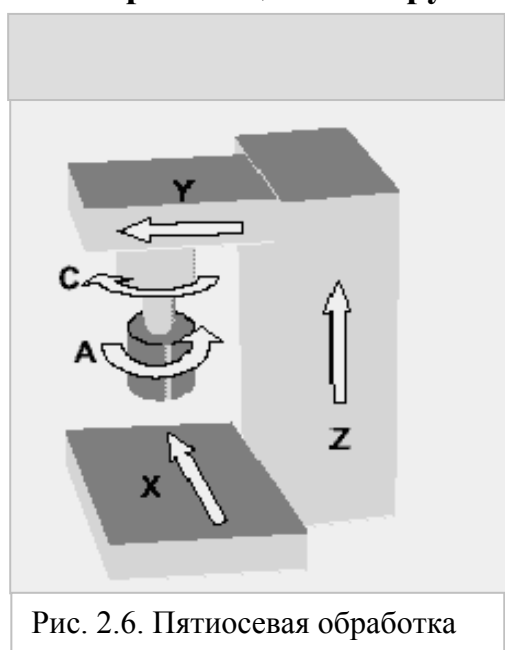


Рис. 2.6. Пятиосевая обработка

**Ориентация инструмента относительно криволинейных поверхностей.** Эта возможность позволяет отказаться от суррогатных решений при пятиосевой обработке криволинейных поверхностей в производстве штампов и пресс-форм (формирование траектории движения с помощью САМ-систем одновременным заданием в кадре пяти осей (рис. 2.6)). Необходимыми условиями использования данной возможности является наличие на станке двухосевой поворотной инструментальной головы и установка дополнительного пятиосевого пакета. Дополнительно к ориентации инструмента это обеспечит компенсацию длины и радиуса в объеме, ориентированный отвод в пространстве в случае поломки. Наиболее

перспективно использование этих возможностей в цепочке из CAD/CAM-систем.

**Специальные функции для токарно-фрезерных станков.** При выполнении фрезерных операций на телах вращения необходимо совмещать движения линейных и круговых осей. Для упрощения этих действий введены специальные функции TRANSMIT и TRACYL, которые позволяют свести работу программиста к работе на плоскости. Необходимое перемещение круговой оси система формирует автоматически.

**Задание запрещенных для перемещения зон на станке.** Отдельные зоны на станке можно объявить запретными. Конфигурация таких зон может быть произвольной. В этом случае система автоматически отслеживает возможность попадания в эти зоны и блокирует указанные ситуации (например, защита зоны патрона на токарном станке).

**Создание программных конструкций, аналогичных языкам высокого уровня.** Язык программирования для УЧПУ содержит много элементов из языков программирования высокого уровня (больше всего он напоминает Паскаль). Программист может создавать переменные различных типов, использо-

вать команды условных и безусловных переходов, арифметические и логические операции, циклы, выполняемые по условиям (WHILE – DWHILE; REPEAT – IL; FOR – DFOR). Система предоставляет доступ к внутренним переменным (например, к текущей памяти положения, к значениям остатков пути внутри кадра и т. .).

Имея эти возможности, можно создавать программы, построенные по принципу групповой технологии (для ввода новой детали необходимо просто заполнить таблицу внутри программы) и существенно облегчить работу оператора.

**Вызов внешних подпрограмм при возникновении аварийных ситуаций.** В случае возникновения нештатных ситуаций (например, при поломке инструмента) автоматически может быть вызвана специальная подпрограмма, внутри которой организуются все необходимые действия по устранению ситуации. Возврат на контур обработки производится одной командой, при этом можно вернуться в любую точку прерванного кадра.

**Выводы.** Развитие систем числового программного управления и наличие таких мощных графических пакетов как САПР-ЧПУ/2000 в настоящее время обеспечивает возможность вводить программу обработки детали в виде обычной технологической карты в среде визуального проектирования. Оператор в этом случае вводит по шагам контур детали, состоящий из таких простых графических элементов как прямая, дуга, конус и т. д. Все переходы между элементами просчитывает СЧПУ. Затем моделируются инструменты и траектории их перемещения, а также съем материала с детали. Каждый шаг поддерживается графически на экране СЧПУ, а в завершение оператор может получить трехмерное динамическое моделирование обработки детали. Такое графическое моделирование является настолько точным, что позволяет обойтись без проверки непосредственно на станке. Затем СЧПУ формирует весь технологический процесс, включая число проходов, выбор инструментов, скорость вращения шпинделя, скорость подачи и подбор компенсационных значений.

Если программирование производилось не на станке, а, например, с помощью САД-системы, оператор в большинстве случаев имеет возможность просмотреть программу на станке в графическом виде и внести необходимые коррективы.

Появление новых функций в ЧПУ позволяет сегодня существенно уменьшить время переналадки станка, которое в универсальном станке может занимать до 90% от общего времени работы. Другие функции позволяют избежать поломки дорогостоящего инструмента и повреждения дорогостоящей детали. Износ инструмента также можно снизить за счет программных функций ЧПУ.

Из вышеперечисленного видно, что в качестве приводов для станков наиболее целесообразно использовать цифровые приводы, т. е. интеллектуальные приводы со встроенным микропроцессором, которые принимают и обрабатывают сигнал с датчика измерения перемещения без участия ЧПУ. Управление от ЧПУ идет при этом по цифровой шине сетевого типа. Такое управление по-

вышает скорость реакции приводов и обеспечивает качественные показатели, недостижимые при классической схеме совмещения функций управления в однопроцессорных ЧПУ. Цифровой интерфейс позволяет также ускорить обмен данными между СЧПУ и приводами.

Все современные цифровые приводы имеют программные пакеты для отладки привода. Для настройки достаточно указать тип двигателя и преобразователь и система управления на базе МК автоматически настраивается на данные характеристики. Затем можно запустить автоматическую привязку двигателей к станку. Точную окончательную настройку можно произвести, используя программный осциллограф системы ЧПУ.

### 3. ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ УПРАВЛЕНИЯ ЭП

В системах ЧПУ задачи управления распределены между отдельными подсистемами, как на аппаратном, так и на программном уровне. Основной элементной базой при разработке современных систем управления являются: программируемые логические схемы (ПЛИС) и микроконтроллеры [12-14]. В многопроцессорных системах центральная микроЭВМ распределяет задачи управления между отдельными микроконтроллерами. При распределении задач микроконтроллер может решать все основные задачи управления ЭП или частичные функции, связанные с реализацией контура положения. В зависимости от архитектуры центральной ЭВМ обмен информацией с МК может быть осуществлен по магистрали ЭВМ типа Q-BUS, ISA, PCI или через интерфейс последовательной связи RS-485, MODBUS. В последнем случае решается задача территориально распределенных систем числового программного управления с локальными цифровыми электроприводами, что, например, для некоторых технологических установок имеет решающее значение.

Принцип отработки управляющей программы и организации следящего электропривода в УЧПУ показан на рис. 3.1.

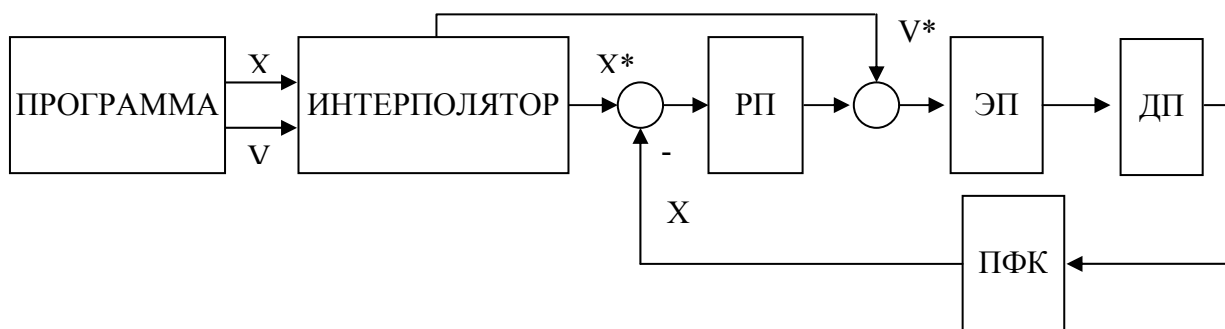


Рис.3.1. Функциональная схема следящего электропривода УЧПУ

Управляющая программа представляет собой последовательность кадров, определяющих траекторию движения исполнительных органов станка на участках обработки детали. Числовая информация, определяющая конечное значение координат  $X$  и скорость подачи  $V$  на участке траектории после подготовки и интерпретации поступает на интерполятор, где происходит развертка координат во времени и формирование текущих величин задания положения и скорости по каждой из координат. Перемещение механизма контролируется круговыми или линейными датчиками положения (ДП). Организация обратной связи и согласование с машинным представлением информации при работе ДП в фазовом режиме осуществляется с помощью преобразователя фаза-код (ПФК). Сигнал рассогласования по положению формируется регулятором положения (РП) и затем подается на отработку в электропривод. Для улучшения динамических свойств следящий электропривод строится по принципу комбинированной системы. Для этого с выхода интерполятора на вход регулятора скорости ЭП дополнительно подается сигнал задания скорости  $V^*$ , а сам ЭП должен быть охвачен обратной связью по скорости.

В приводах станков с ЧПУ в качестве датчиков положения используются: фотоэлектрические датчики импульсного и кодового типа, а также датчики положения типа СКВТ. Фотоэлектрические импульсные датчики (ИД) содержат два фотодиода, сдвинутые относительно друг друга на 90° по растровой решетке, и схему определения направления вращения. В зависимости от направления вращения импульсные последовательности формируются на одном из выходных каналов ИД. Использование импульсных датчиков предполагает наличие канала накопления информации (рис. 3.2).

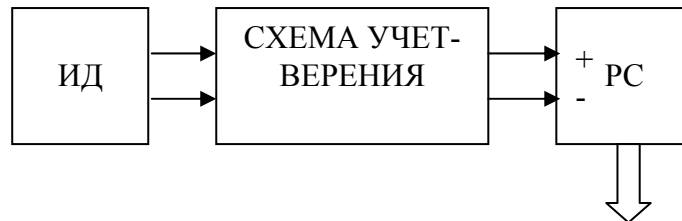


Рис. 3.2. Канал импульсно-фазового преобразователя

В канале накопления наиболее приемлемым вариантом является использование реверсивного счетчика (РС). Для увеличения разрешающей способности импульсных датчиков используется схема учетверения.

В большинстве случаев интерфейсные блоки программно-управляемых таймеров/счетчиков МК не имеет функций накопления реверсивных сигналов, тогда необходимо использовать два таймера/счетчика.

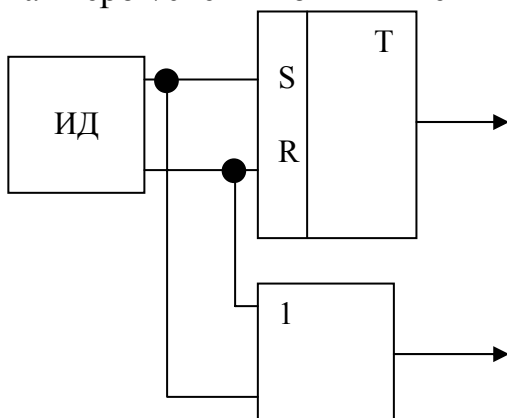


Рис. 3.3. Схема определения направления

В некоторых моделях МК используются таймерные сопроцессоры, которые имеют вход управления направлением накопления импульсных последовательностей.

Для определения направления движения в схему сопряжения с МК вводят триггер Т (см. рис. 3.3) и объединяют сигналы каналов ИД.

### 3.1. ВАРИАНТЫ СОПРЯЖЕНИЯ УЧПУ С ЭП

Варианты сопряжения ЭП с УЧПУ во многом определяются методом интерполяции и распределением функций между аппаратной и программной частью. Рассмотрим несколько вариантов сопряжения следящего ЭП с УЧПУ. На рис. 3.4 показана схема сопряжения при работе датчика положения (ДП) в фазовом индикаторном режиме [3, 15, 16].

С помощью генератора тактовых импульсов ГТИ, делителя частоты (ДЧ1) и формирователя опорных напряжений (ФОН) формируется ортогональная система синусоидальных напряжений для питания первичных обмоток ДП:

$$U_1 = U \sin \omega_o t; \quad U_2 = U \cos \omega_o t, \quad (3.1)$$

где  $U, \omega_o$  – амплитуда и круговая частота питающего напряжения.

Фаза  $\varphi$  выходного сигнала ДП в этом случае пропорциональна углу поворота вала двигателя:

$$U_3 = U \sin(\omega_o t + \varphi). \quad (3.2)$$

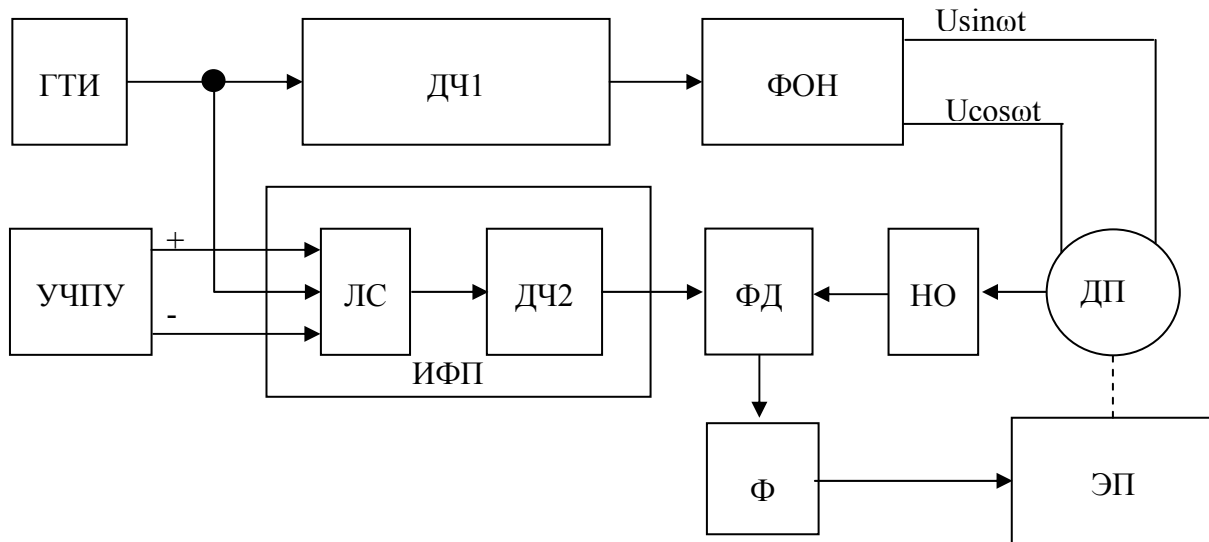


Рис. 3.4. Функциональная схема сопряжения ЭП при работе ДП в фазово-индикаторном режиме

Сигнал задания в рассматриваемом случае вырабатывается УЧПУ в виде единичных приращений, частота которых определяет скорость, а их количество – заданное положение фазы.

Накопление приращений осуществляется с помощью логической схемы ЛС и делителя частоты ДЧ2 импульсно-фазового преобразователя (ИФП). На вход ДЧ2 приходит результирующая частота  $f_{ГТИ} \pm \nabla f$  генератора и импульсов приращения. В результате этого выходной сигнал (последний разряд счетчика) ДЧ2 сдвигается по фазе относительно опорного сигнала ДЧ1. Сигнал рассогласования по фазе между задающим сигналом и обратной связью с выхода ДП формируется импульсно-фазовым дискриминатором (ФД), а затем через низкочастотный фильтр (Ф) подается на вход ЭП. Для согласования уровней сравниваемых сигналов используется нуль орган (НО). При наличии приращения на выходе «+» осуществляется формирование дополнительного импульса на вход делителя частоты ДЧ2, а по сигналу на выходе «-» – формируется запрет на прохождение импульса с ГТИ2 на вход ДЧ2. В последнем случае работа последнего разряда ДЧ2 по отношению к опорному сигналу запаздывает на число импульсов, поступивших с выхода «-» УЧПУ.

Без дополнительной синхронизации импульсных последовательностей подобное разделение выполнить затруднительно, поэтому целесообразно использовать распределитель импульсов (РИ) (рис. 3.5) и разделить импульсы



ГТИ на две последовательности, одну из которых использовать в качестве счетных импульсов, а другую для выделения и синхронизации импульсов приращений. На суммирующий вход реверсивного счетчика РС2 поступает суммарная частота развертки и приращений, а на вычитающий – разностная частота приращений.

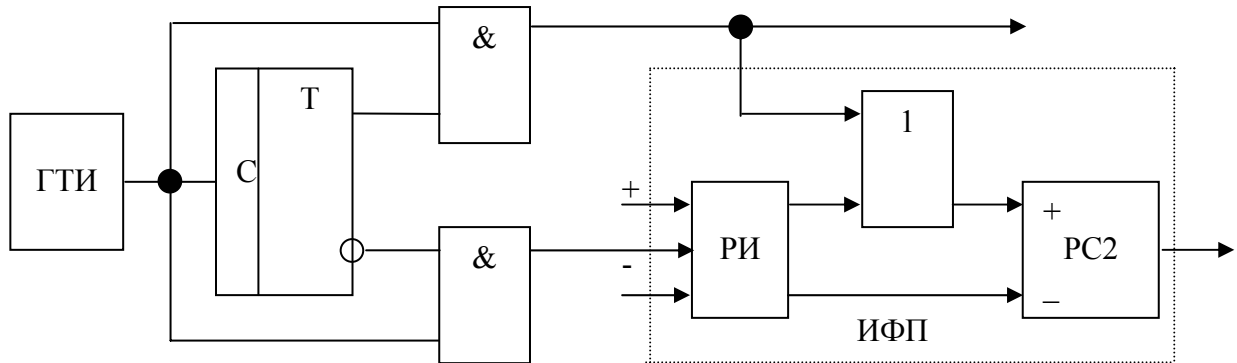


Рис. 3.5. Схема разделения импульсных последовательностей

Следующий вариант сопряжения УЧПУ (рис. 3.6) основан на реализации амплитудно-разностного режима работы ДП.

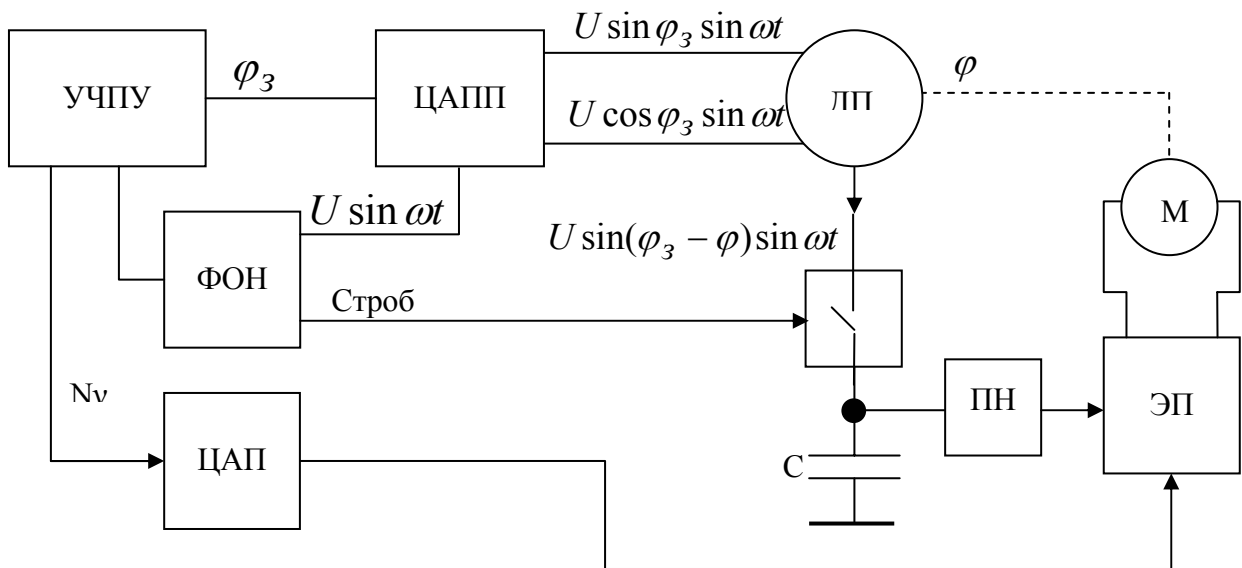


Рис. 3.6. Схема сопряжения следящего ЭП с УЧПУ при работе ДП в амплитудно-разностном режиме

С интерполятора УЧПУ выдается код  $\varphi_3$ , характеризующий задание на угол поворота. С помощью цифро-аналогового преобразователя перемещений ЦАПП и ФОН формируется система питающих напряжений:

$$U_1 = U \sin \varphi_3 \sin \omega_0 t, \quad U_2 = -U \cos \varphi_3 \sin \omega_0 t, \quad (3.3)$$

определяющая амплитудно-разностный режим работы ДП. Датчик в этом случае выполняет функции амплитудного дискриминатора:

$$U_3 = U \sin(\varphi_3 - \varphi) \sin \omega_0 t. \quad (3.4)$$

Для определения рассогласования в моменты времени, соответствующие максимуму амплитуды, формируется строб-импульс, по которому производится выборка выходного значения напряжения через ключ и запоминание его на конденсаторе С. Затем этот сигнал через повторитель напряжения (ПН) поступает на вход ЭП в качестве сигнала рассогласования между заданным и текущим положением. Система комбинированного типа – дополнительный сигнал управления поступает на ЦАП и определяет заданное значение частоты вращения  $U_v$  двигателя (М) подачи.

Другой вариант формирования ошибки основан на использовании фазочувствительного выпрямителя. Работа фазочувствительного выпрямителя должна быть привязана к опорному сигналу, который имеет прямоугольную форму.

Модуляция питающих напряжений (см. рис. 3.7) производится за счет двух постоянных запоминающих устройств (ПЗУ) и умножающих ЦАП [19]. ПЗУ1, 2 запрограммированы соответственно по закону  $\sin \varphi_3$  и  $\cos \varphi_3$ . С целью экономии памяти значения  $\sin \varphi_3$  и  $\cos \varphi_3$  записываются в пределах угла  $0..45^\circ$ .

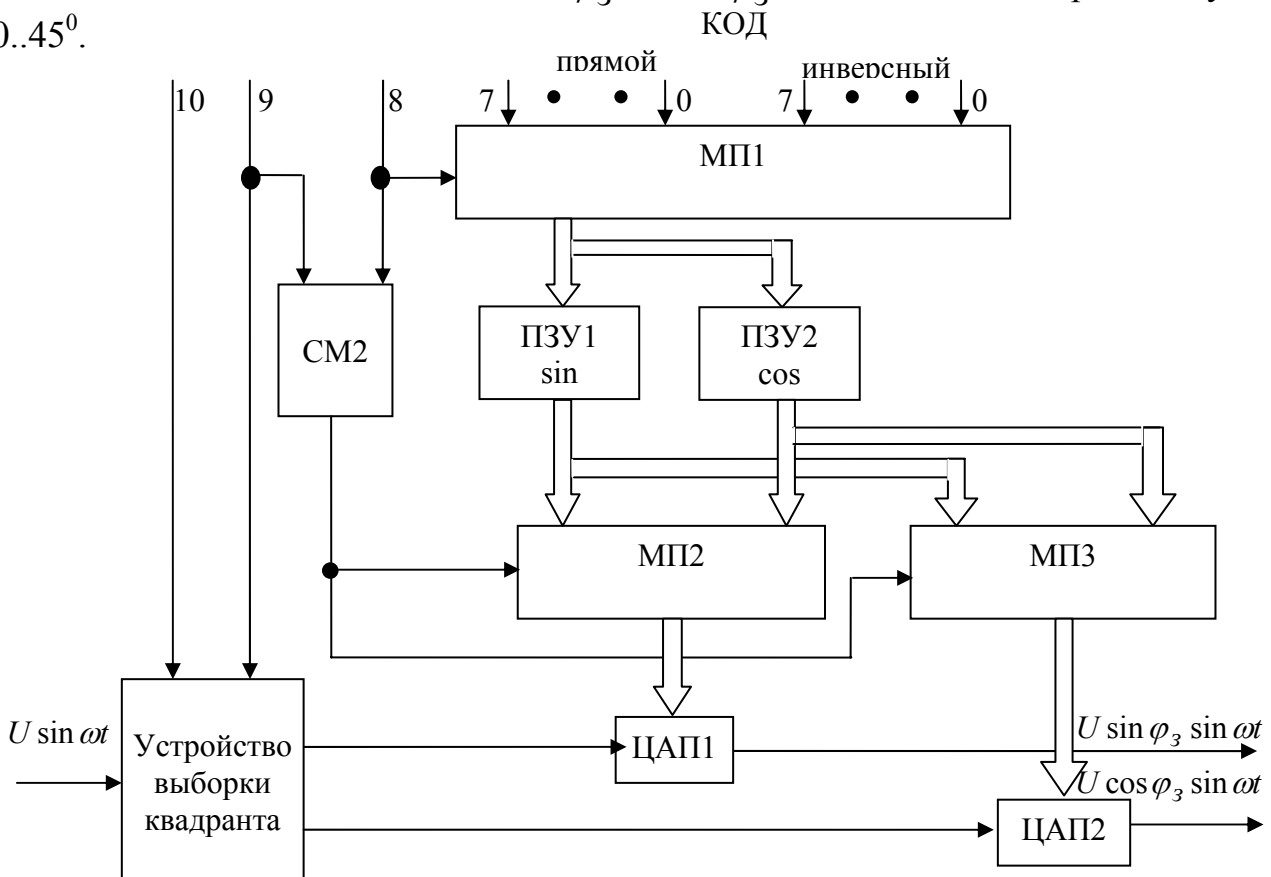


Рис. 3.7. Схема функционального преобразователя на основе умножающих ЦАП

При каждом переходе через  $45^\circ$  производится коммутация кодов с помощью мультиплексора МП1. Выходы ПЗУ подключаются к ЦАП1 и ЦАП2 через мультиплексоры МП2 и МП3, управляемые от сумматора по модулю 2 (СМ2),

на вход которого поданы разряды с весом  $45^0$  и  $90^0$ . Два старших разряда управляют выбором квадранта. При этом опорное напряжение подается на входы ЦАП в виде прямого или инверсного сигнала.

На рис. 3.8 показан вариант реализации следящего ЭП, где в качестве ПФК используется преобразователь считывания. С помощью ГТИ, Сч создается опорный интервал (см. рис. 3. 9), соответствующий периоду питания ДП.

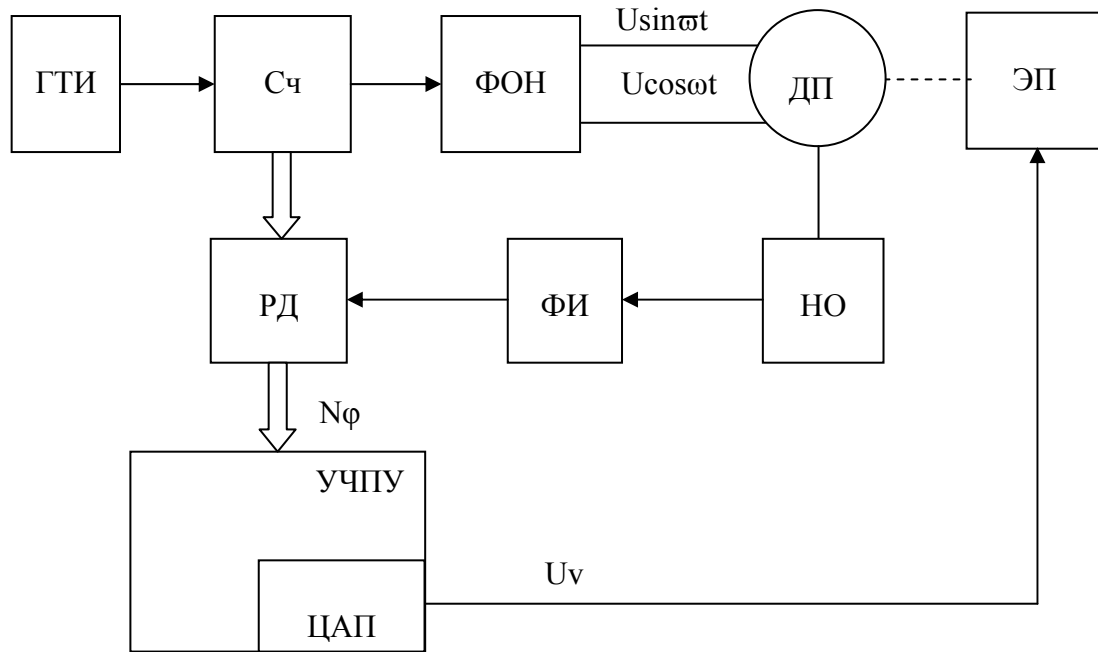


Рис. 3.8. Схема сопряжения при использовании преобразователя считывания

Последний разряд Сч поступает на ФОН, где производится выделение 1-ой гармоники и усиление по мощности выходных напряжений. Формирование системы питающих напряжений (3.1) можно осуществить путем сдвига опорного синусоидального сигнала, или за счет второго канала. В последнем случае необходимо сформировать прямоугольный сигнал, сдвинутый относительно последнего разряда счетчика на  $90^0$ .

В момент перехода напряжения с выхода ДП через нуль с помощью формирователя (ФИ) формируется импульс считывания, по которому осуществляется запись текущего кода счетчика Сч в регистр данных (РД). Для исключения неоднозначности считывания информации импульс считывания необходимо синхронизировать с работой счетчика. Код РД пропорционален углу поворота. Он подается на обработку в УЧПУ, где вычисляется рассогласование, а затем сигнал ошибки через цифровой регулятор положения выдается через ЦАП на вход ЭП.

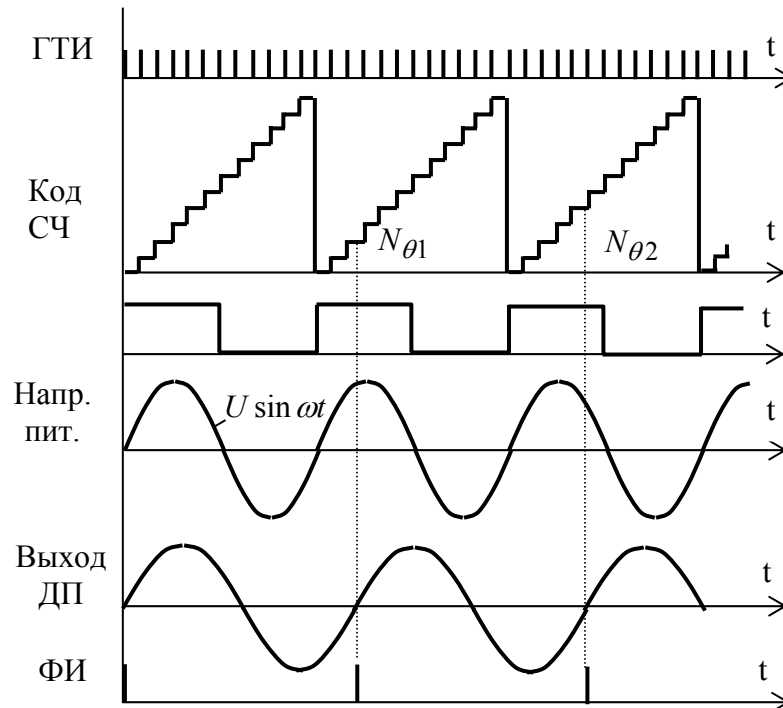


Рис. 3.9. Диаграммы импульсов и напряжений

Возможны и другие варианты рассмотренного преобразователя фаза-код, например, преобразователя с времяимпульсным формированием кода фазы [14].

### 3.2. ПРИМЕРЫ РАЗРАБОТКИ СЛЕДЯЩЕГО ЭП НА ОСНОВЕ МК

Рассмотренные варианты аппаратных решений схем сопряжения могут быть использованы как основа для реализации следящего электропривода на основе микроконтроллера. На рис. 3.10 показан пример разработки следящего ЭП с преобразователем считывания. В качестве микроконтроллера может быть использован МК типа AVR AT90S/L8535. МК включает в себя три таймера/счетчика, два из которых могут быть использованы для формирования ШИМ-сигналов, причем один из них Т1 имеет два сопряженных канала формирования ШИМ-сигнала.

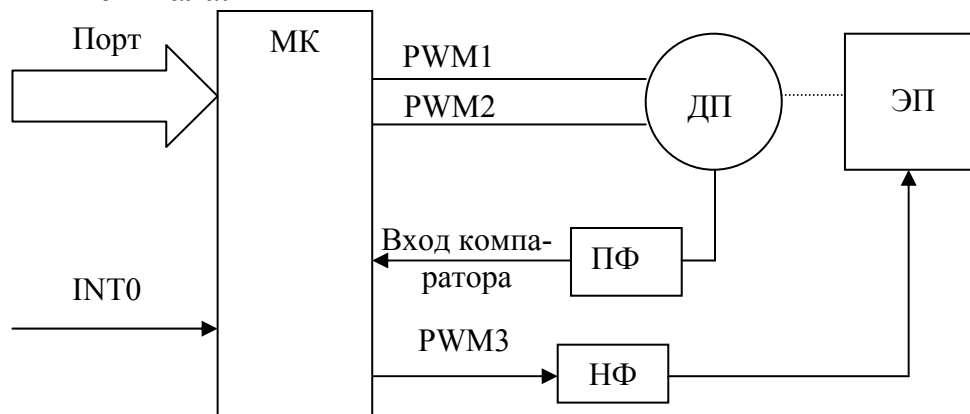


Рис. 3.10. Следящая система на базе МК с преобразователем считывания

Для формирования напряжений (3.1), модулированных по синусоидальному закону на выходах широтно-импульсных модуляторов PWM1, PWM2, используют таблицы, в которые записывают дискретные значения сигналов. Загрузку очередного элемента таблицы синусов в регистры ШИМ необходимо производить по таймеру T0, определяющему частоту развертки опорных сигналов.

Выходной каскад при этом может быть подобен схеме инвертора. В качестве силовых ключей могут быть использованы аналоговые коммутаторы, так как токи нагрузки достаточно низкие. Датчик является естественным фильтром высших гармоник. Однако для снижения влияния импульсных помех, его выход целесообразно подключить к входу аналогового компаратора МК через полосовой фильтр (ПС).

В момент перехода сигнала ДП через «0» компаратор МК будет вырабатывать сигнал прерывания, по которому производится считывание кода опорного счетчика T1 широтно-импульсного модулятора. Этот режим работы счетчика носит название режима «считывание на лету» и определяется путем записи соответствующей информации в регистр управления таймера/счетчика. Зафиксированное значение используется для формирования текущего положения фазы. При этом необходимо осуществить привязку к текущему интервалу развертки ШИМ-сигнала, т. к. фаза будет определяться номером интервала, умноженного на период развертки по несущей частоте, выраженной в количестве импульсов. Прием задания на перемещение от центральной микроЭВМ осуществляется через порт. Для сопряжения по шине можно использовать стандартный интерфейс, включающий в себя регистр данных и регистр состояния. Причем сигнал записи информации в регистре данных можно использовать в качестве сигнала внешнего прерывания INT0, по которому контроллер будет принимать данные. Сигнал ошибки по положению подается на вход ЭП в виде широтно-импульсного сигнала. Для выделения постоянной составляющей используется низкочастотный фильтр НФ. Формирование сигнала управления PWM3 электроприводом осуществляется таймером T2.

Рассмотренный вариант предъявляет довольно жесткие требования к быстродействию МК, так как выходная частота питающих напряжений составляет порядка 2,5 КГц. Однако избыточная информация (в качестве счетчика используются 8 или 10-разрядные счетчики, а соотношение между несущей и выходной составляет 12 и более) позволяет определять по приращению кода фазы значение скорости.

Семейство микроконтроллеров обычно имеет базовый вариант МК, например, AT90S8515, который используют в инструментальных средствах отладки [17,18]. Основной особенностью базового микроконтроллера является наличие внешней шины адресов/данных ШАД, позволяющей осуществлять сопряжение с внешней памятью и нестандартными периферийными устройствами.

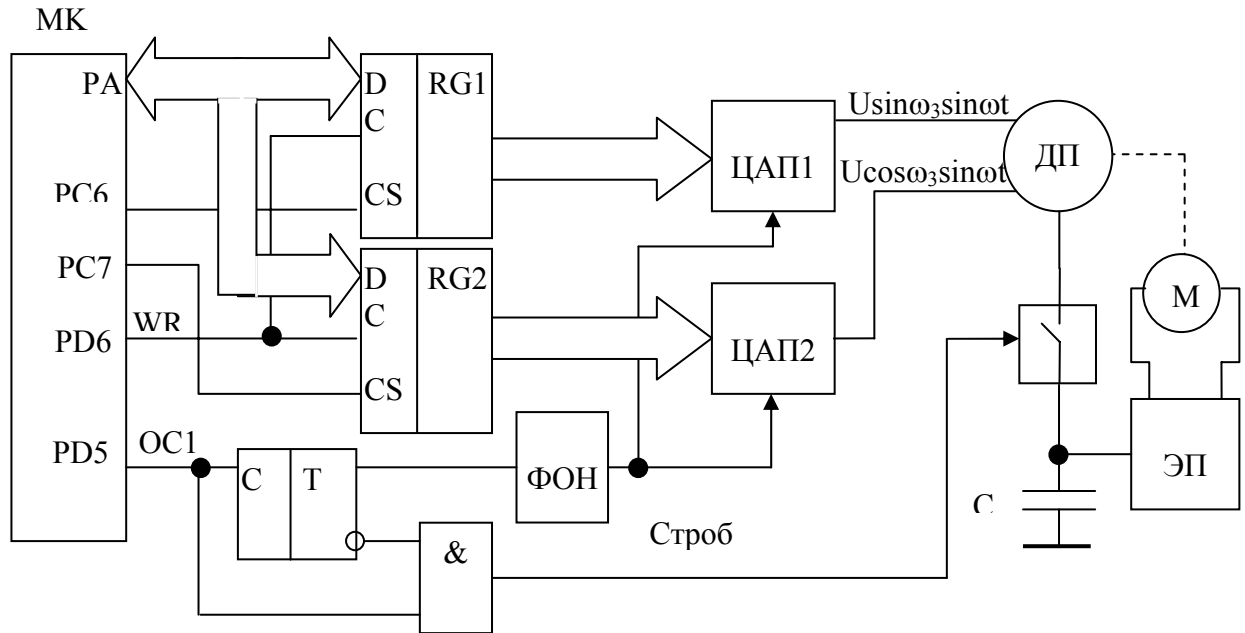


Рис. 3.11. Следящая система на базе МК при работе ДП в амплитудно-разностном режиме

Альтернативные выходы порта А (РА0..РА7) представляют собой мультиплексную ШАД (АДО – АD7), а порта С (РС0..РС7) – старшие разряды шины адреса (А8..А15). Вывод АLE используется как сигнал строба адреса. Сигналы записи/чтения (WR/ RD) данных формируются на выводах PD6, PD7.

Следящая система на базе МК при работе датчика положения в амплитудно-разностном режиме показана на рис. 3.11. Для формирования опорного напряжения используется сигнал с выхода таймера Т1, который определяет половину периода опорной частоты. Дополнительный триггер Т в данном случае расширяет разрядность счетчика таймера. При опрокидывании Т формируется сигнал «Строб», по которому происходит фиксация рассогласования по фазе через ключ на конденсаторе С, другой выход триггера поступает на ФОН, где происходит выделение 1-ой гармоники и формирование опорного напряжения. Для модуляции напряжений используют ЦАП1 и ЦАП2. Коды, пропорциональные  $\sin\varphi_3$  и  $\cos\varphi_3$  задающего угла поступают с регистров данных RG1 и RG2. Запись в регистры осуществляется по упрощенной схеме (без использования регистра адреса и дешифратора). В качестве селектирующих сигналов используются старшие разряды шины адреса порта РС.

### 3.3. ПРИНЦИПЫ ПОСТРОЕНИЯ ЦИФРОВЫХ СИСТЕМ УПРАВЛЕНИЯ ТИРИСТОРНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ

Использование простых МК для управления приводом по системе ТП-ДП, как правило, ограничены возможностями периферии, в том числе количеством таймеров, имеющихся у разработчиков. Для управления ТП в этом случае целесообразно использовать одноканальную СИФУ [11,16]. Угол отпирания тиристоров:

$$\alpha_3 = \frac{2\pi}{m}q + \alpha'_3, \quad (3.4)$$

где  $m$  – число фаз преобразователя,  $q = 0, 1, 2, \dots$  – номер интервала коммутации,  $\alpha'_3$  – угол управления относительно естественной точки коммутации.

Аппаратный вариант цифрового СИФУ, показанный на рис. 3.12 [20], содержит регистр данных, схему сравнения кодов ССК, счетчик Сч, распределитель импульсов РИ, триггер Т.

В регистр записывается угол управления. Устройство ССК сравнивает код задания и код счетчика и в момент их равенства формирует отпирающий импульс.

Распределитель импульсов РИ организован на базе кольцевого сдвигающего регистра, при поступлении каждого импульса происходит перераспределение отпирающих импульсов, поступающих с его выходов на тиристоры преобразователя (см. рис. 3.13). Каждый отпирающий импульс формирует сигнал прерывания, по которому МК из кода счетчика, равного в момент отпирания коду  $\alpha_3$ , вычитает код  $2\pi/m$  и вводит его в счетчик.

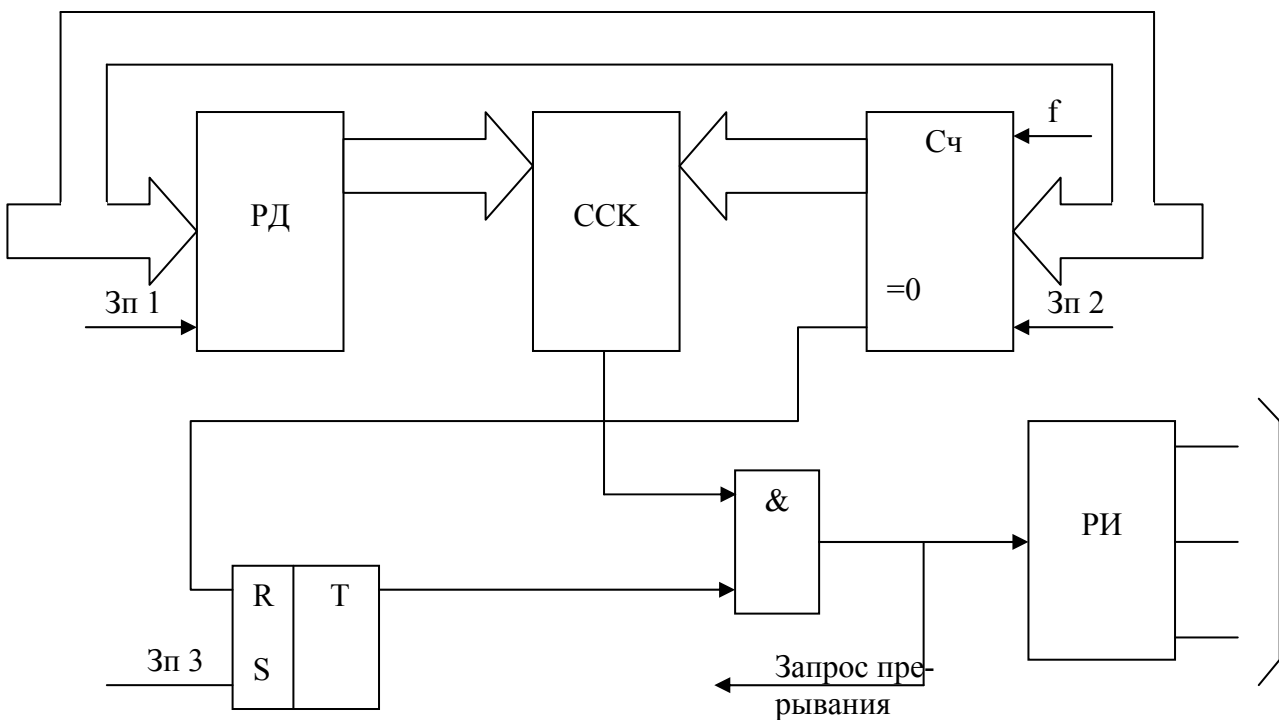


Рис. 3.12. Одноканальная цифровая СИФУ

Таким образом, следующий тиристор откроется через  $2\pi/m$  с тем же углом  $\alpha$ , если  $\alpha_3$  не изменится. При работе с  $\alpha < 2\pi/m$  разность  $\alpha_3 - 2\pi/m$  будет отрицательной (дополнительный код). Для исключения неправильной работы ССК МК формирует сигналом Зп 3 (запись 3), который взводит триггер Т, запрещая формирование отпирающих импульсов. Триггер сбрасывается, когда счетчик Сч переходит через нуль.

Синхронизация с сетью осуществляется в момент естественного отпирания одной из фаз, например, фазы А. На диаграмме (рис. 3.14) показаны импульсы управления при изменении угла с  $\alpha_3=30^\circ$  на  $\alpha_3=150^\circ$ . Естественный порядок чередования управляющих импульсов изменяется в момент, соответствующий времени появления импульса синхронизации, по которому происходит загрузка счетчика новым углом отпирания.

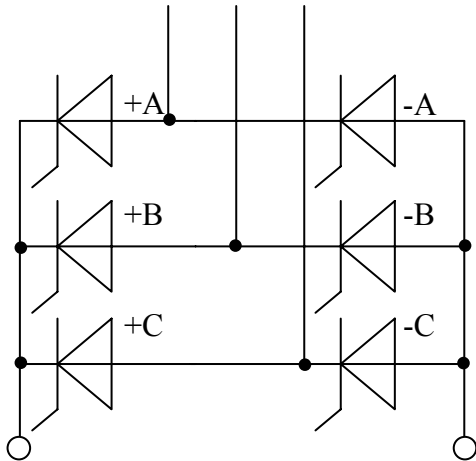


Рис. 3.13. Комплект тиристоров

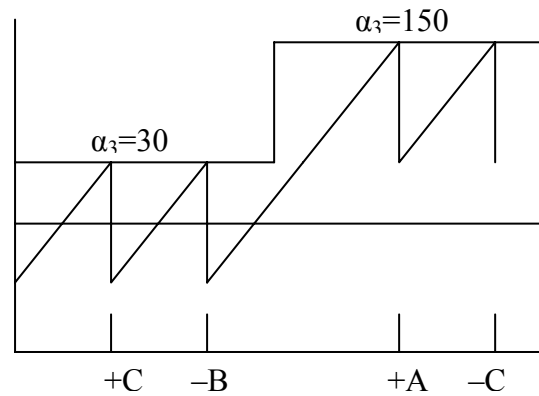


Рис. 3.14. Диаграммы импульсов управления при изменении угла  $\alpha_3=30^\circ$  на  $\alpha_3=150^\circ$

На рис. 3.15 приведены диаграммы управляющих импульсов при изменении угла задания с большего значения на меньшее. В момент, когда код счетчика равен углу  $120^\circ$ ,  $\alpha_3$  уменьшается до  $20^\circ$ . При этом сразу же формируется отпирательный импульс на тиристор +А, т. к.  $\alpha_3$  меньше кода счетчика.

Распределитель импульсов переключается в следующее положение. Из кода счетчика вычитается  $2\pi/m$ , и его код станет равным  $60^\circ$ . Так как  $\alpha_3$  по-прежнему меньше кода счетчика, то с приходом следующего тактового импульса формируется отпирательный импульс на тиристор – С. Распределитель РИ переключится в следующее положение. Код счетчика вновь уменьшится на  $2\pi/m$  и будет равен 0. Очередной отпирательный импульс будет выдан через  $20^\circ$  на тиристор +В.

Реализация ССК предполагает наличие вывода « $\geq$ » и схемы логики, формирующей дополнительный импульс при коде счетчика  $> \alpha_3$ .

Функциональная схема управления тиристорным преобразователем на основе МК показана на рис. 3.16. Синхронизация с сетью производится по

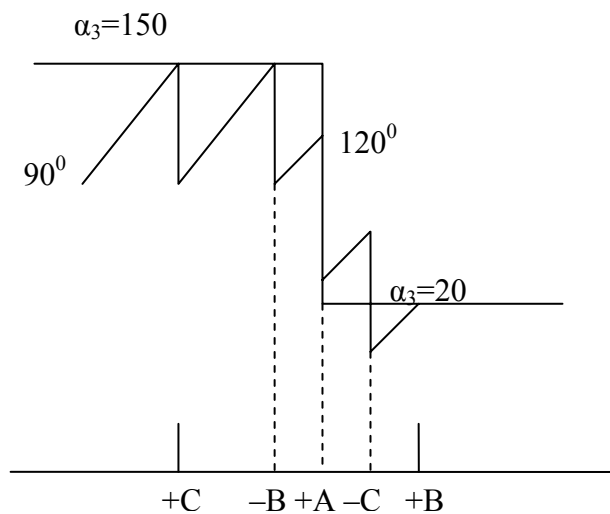


Рис.3.15. Диаграммы импульсов управления при изменении угла  $\alpha_3=120^\circ$  на  $\alpha_3=20^\circ$



внешнему выводу прерывания INT1, на вход которого поступает сигнал с нуля органа (НО). Подключение датчика тока ДТ к МК осуществляется через альтернативный вывод АЦП порта А. Для определения направления тока используется дополнительный дискретный вход МК.

Управление преобразователем осуществляется через порт В с помощью формирователей импульсов (ФИ). Состояние выходов порта определяет очередной импульс управления, поступающий на тиристоры моста.

Для реверсивного преобразователя рабочий комплект (выпрямителя или инвертора) определяется дополнительным сигналом «Комплект» с выхода МК. Управление двумя противофазными вентилями осуществляется со сдвигом на  $180^\circ$ . При использовании принципа многоканального управления это достигается двукратным запуском каждого канала СИФУ и привязкой его выходного импульса к отрицательной или положительной полуволне синхронизирующего напряжения. Для одноканального управления этот эффект достигается сдвигом последовательности импульсов, выдаваемой через порт управления вентилями. Выходной каскад формирователей импульсов, как правило, содержит импульсные трансформаторы. Для реализации ФИ могут быть использованы существующие решения. Однако для согласования ФИ с выходами МК наиболее целесообразно использовать драйверы.

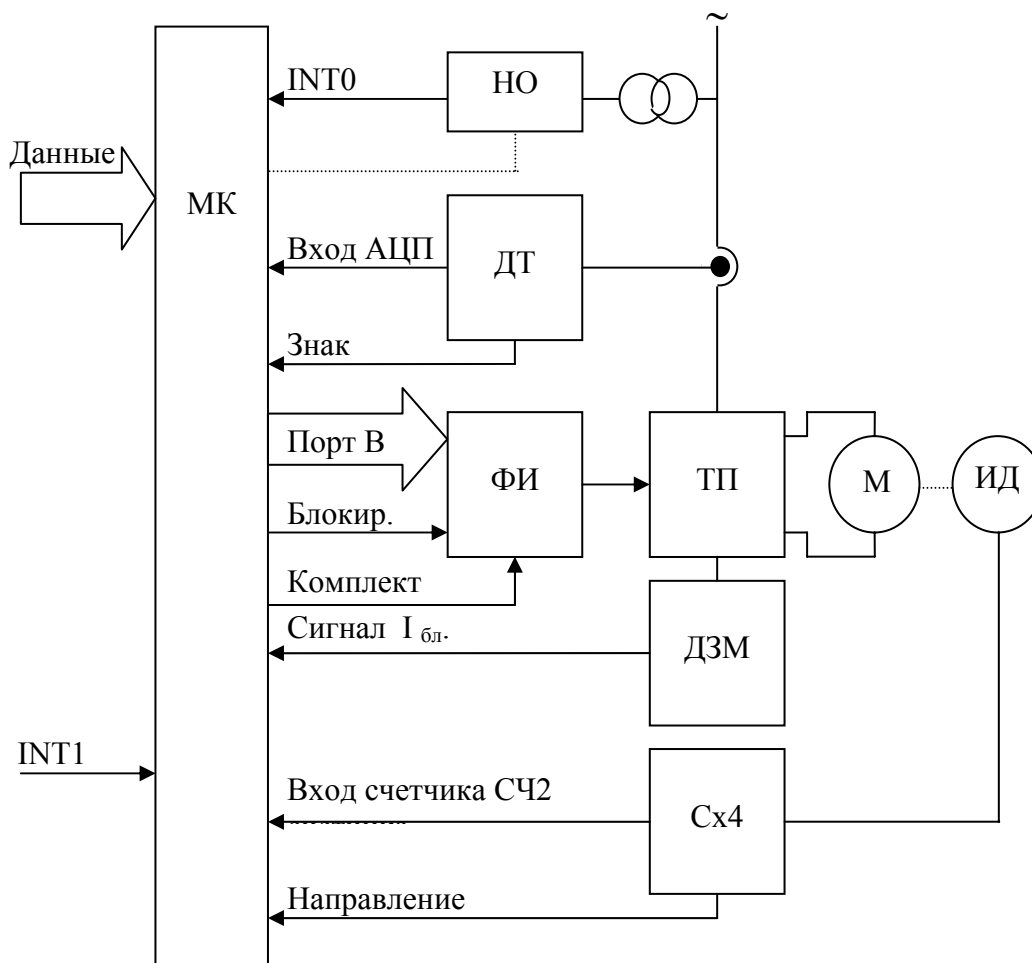


Рис. 3.16. Функциональная схема управления ТП на базе МК

Преобразователь с раздельным управлением переключает работу мостов в момент времени, соответствующий прохождению тока через 0 значение. При этом все тиристоры должны быть отключены. Для контроля состояния тиристоров используется специальный блок ДЗМ, который выполняет функции датчика контроля закрытого состояния тиристорного моста.

Для блокировки отпирающих импульсов может быть использован дополнительный выход МК или сброс выходных линий порта в нуль. Скорость контролируется с помощью импульсного датчика ИД. Для повышения разрешающей способности преобразователя частота-код, реализованного на базе счетчиков МК, может быть использована схема учетверения Сх4. Для определения направления вращения (см. ранее) используется дополнительный триггер.

Рассмотренный принцип построения ЭП практически использует все возможности МК типа AT90S8535.

Из имеющихся трех таймеров/счетчиков первый должен быть использован для задания периода квантования и обработки программ регулятора скорости и тока. Второй таймер используется для реализации системы импульсно-фазового управления, а третий совместно с первым для реализации преобразования частоты в код.

### 3.3.1. Алгоритмы управления

При включении питания или формировании сигнала СБРОС (RESET) происходит запуск работы МК и переход на 0 вектор, где должен быть предусмотрен переход на подпрограмму инициализации МК (рис. 3.17, а).

На этом этапе производится загрузка информации в регистры управления, определяющая режим работы периферийных устройств и конфигурацию МК.

После этого осуществляется переход на основную программу, где может быть предусмотрена обработка кнопок управления преобразователем, вывод информации на дисплей, переход в автономный режим работы для отладки и настройки параметров в режиме диалога с центральной микроЭВМ или с автономного пульта управления. Эта программа может работать в фоновом режиме, то есть во время, свободное от выполнения основных алгоритмов.

Этапу инициализации может предшествовать этап диагностики, на котором может быть осуществлена работоспособность системы в целом и МК, в частности, и выведена диагностическая информация. Все остальные алгоритмы (рис. 3.17, б, в, г) работают по прерыванию. Наивысшим приоритетом при этом обладает программа СИФУ. При частоте питающей сети 50 Гц 3-х фазный преобразователь должен работать с частотой прерывания 300 Гц.

Кроме подпрограмм регулятора скорости и тока в алгоритм управления должна быть включена подпрограмма блока логики, определяющего порядок работы тиристоров при изменении угла управления и работающий комплект тиристорного преобразователя.

Блок логики (рис. 3.18) содержит нуль-орган (операционный усилитель А); триггер заданного направления тока ТЗН; триггер истинного направления

тока ТИН; схему совпадения триггеров Э9..Э12, узел выдержки времени (R,C) и транзисторные ключи, к которым подключены первичные обработки импульсных трансформаторов тиристорных комплектов [21, 23].

В блок поступают два блокирующих сигнала. Первый сигнал  $U_{\text{И}}$  подается на зажим 3 с выхода элементов Э7 всех каналов СИФУ. При наличии управляющего импульса сигнал  $U_{\text{И}} = 0$ . Второй сигнал  $i_{\text{БЛ}}$  подается с выхода датчика проводимости вентилей. Оба этих сигнала блокируют переключения ТЗН и ТИН. Триггер ТИН, кроме того, блокируется схемой совпадения триггеров (выход Э12) через узел выдержки времени и разделительные диоды. ТЗН переключается сразу же после изменения полярности сигнала  $U_{\text{РТ}}$  с выхода регулятора тока, если нет управляющих импульсов и тока в тиристорах. Переключение второго триггера задерживается на время заряда конденсатора С до напряжения порога срабатывания. При использовании пороговой логики тика K511

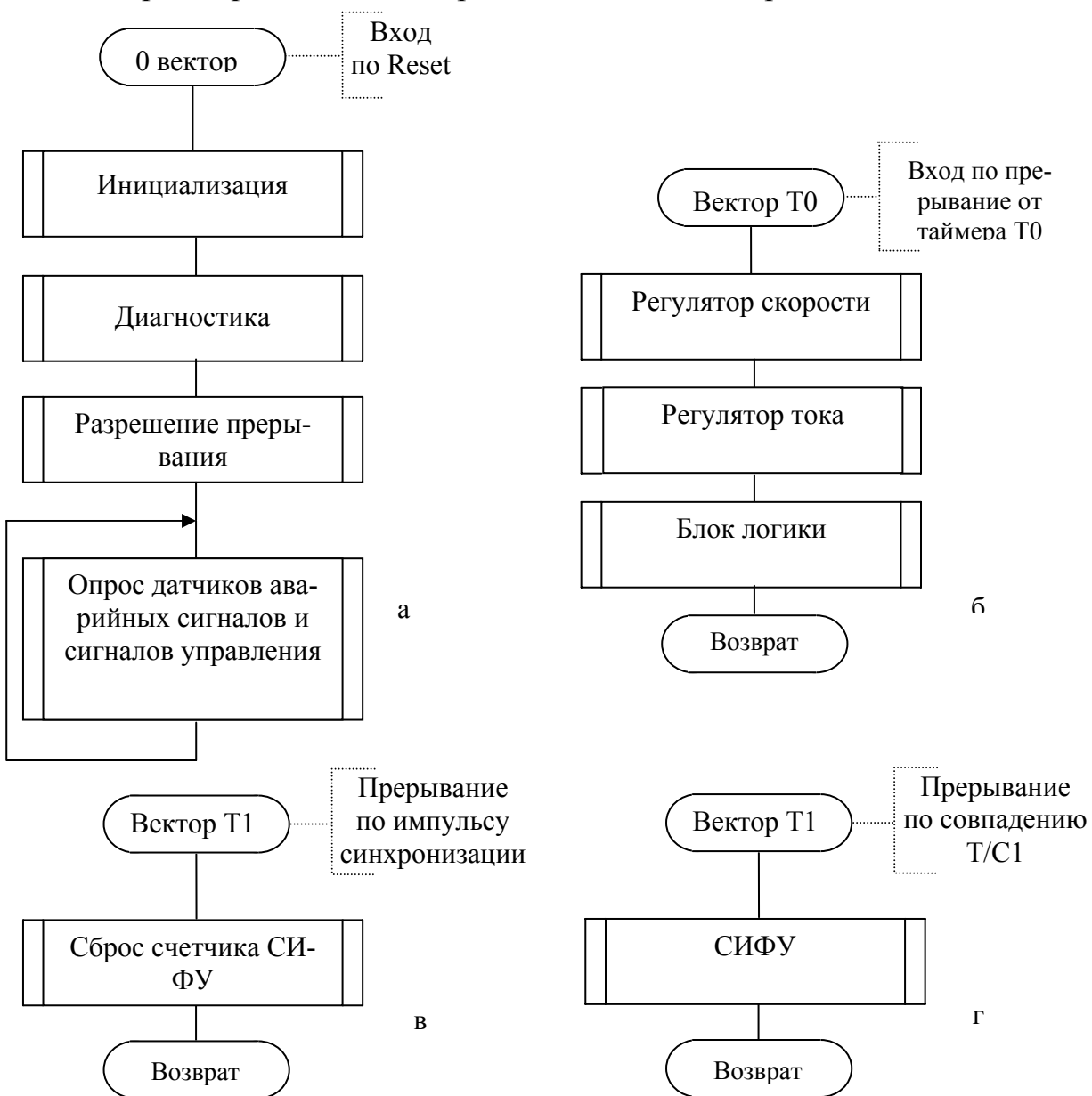


Рис.3.17. Алгоритмы обработки прерываний

это напряжение около 6В. На это время оба выходных ключа Т3 и Т5 закрыты. Если во время отсчета выдержки времени ТЗН вернется в прежнее положение, то сразу же отменяется отсчет выдержки времени и включается силовой ключ прежнего направления. Если во время отсчета выдержки появится ток в каком-либо из тиристоров (пробой, ложный управляющий импульс), то подача с импульсов запрещается. С выхода Э11 (зажим 9) в СИФУ поступает сигнал разрешения выдачи импульса.

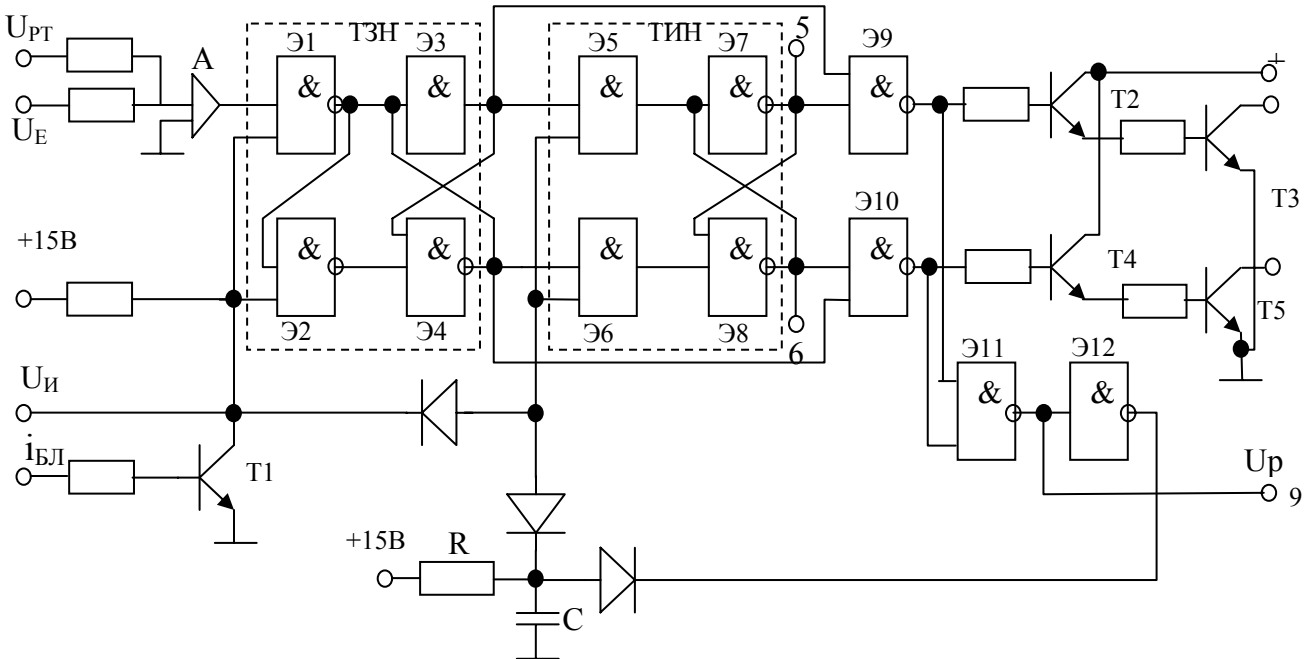


Рис. 3.18. Блок логики

Алгоритм блока логики (см. рис. 3.19) практически повторяет основные условия формирования импульсов управления комплектами реверсивного тиристорного преобразователя, рассмотренные выше. При наличии сигнала с датчика проводимости вентилей  $i_{БЛ}$  формируется обходная ветвь, соответствующая условию сохранения прежнего состояния (триггеры заданного ТЗН и истинного направления тока ТИН сохраняют свои прежние значения). После перехода тиристоров моста в непроводящее состояние  $i_{БЛ} = 0$  и создаются условия для переключения ТЗН. При этом ветвь решения (ТЗН = ТИН) по условию «Нет» формирует сигнал блокировки работы комплектов преобразователя и начинается отсчет паузы. При каждом вхождении в программу блокировки счетчик паузы СчП увеличивает свое значение на 1, и при достижении им заданного значения кода срабатывания N создаются условия для переключения триггера ТИН. Переключению на другой комплект преобразователя предшествует перевод распределителя импульсов РИ через три состояния, соответствующему сдвигу по фазе на  $180^0$ . Если на текущем цикле  $n$  триггер ТЗН вернется в состояние, определенное на предыдущем цикле  $n-1$ , то производится сброс счетчика СчП и выход из программы. На следующем цикле будет формироваться другая ветвь решения (ТЗН = ТИН) по условию «Да», если ТЗН сохранит свое значение и сигнал блокировки будет снят.

Возможны и другие варианты построения алгоритма, вытекающие из рассмотрения работы блока логики. Время паузы составляет порядка 10–20 мс и

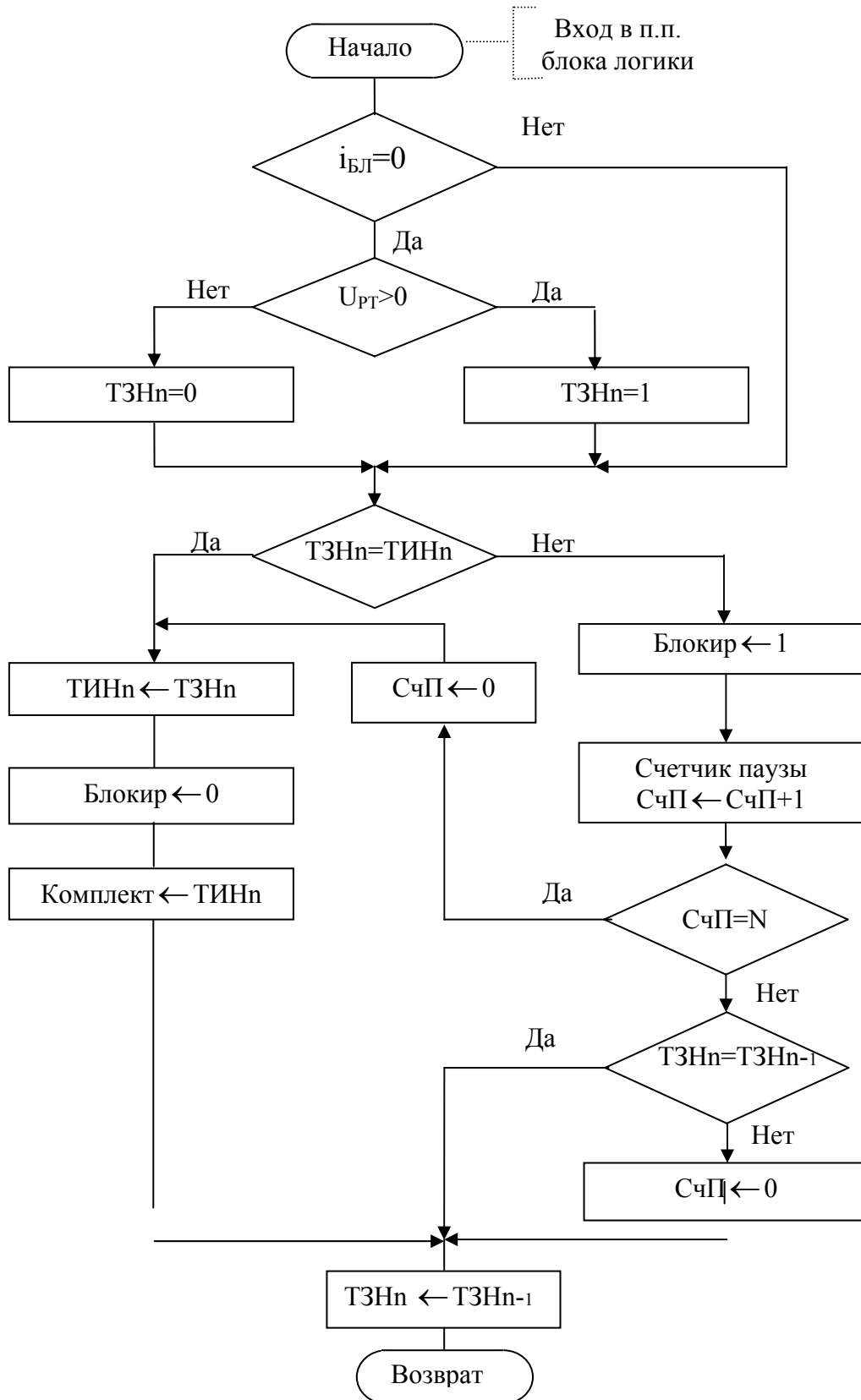


Рис. 3.19. Алгоритм блока логики

определяется фактически частотой прерывания таймера, по которому происходит обработка алгоритма. За это время при быстройдействии МК 1000000 операций в секунду выполняется порядка 10–20 тыс. команд, что намного превышает все программное обеспечение МК для реализации алгоритмов управления ЭП.

Вхождение в алгоритм работы СИФУ (рис. 3.20, а) осуществляется при совпадении содержимого счетного регистра таймера/счетчика (далее счетчика Сч) с кодом регистра сравнения (регистра данных угла задания  $\alpha_3$ ). При работе в стационарном режиме при каждом совпадении кода счетчика с кодом угла  $\alpha_3$  осуществляется сдвиг содержимого ячейки распределителя РИ и перезагрузка

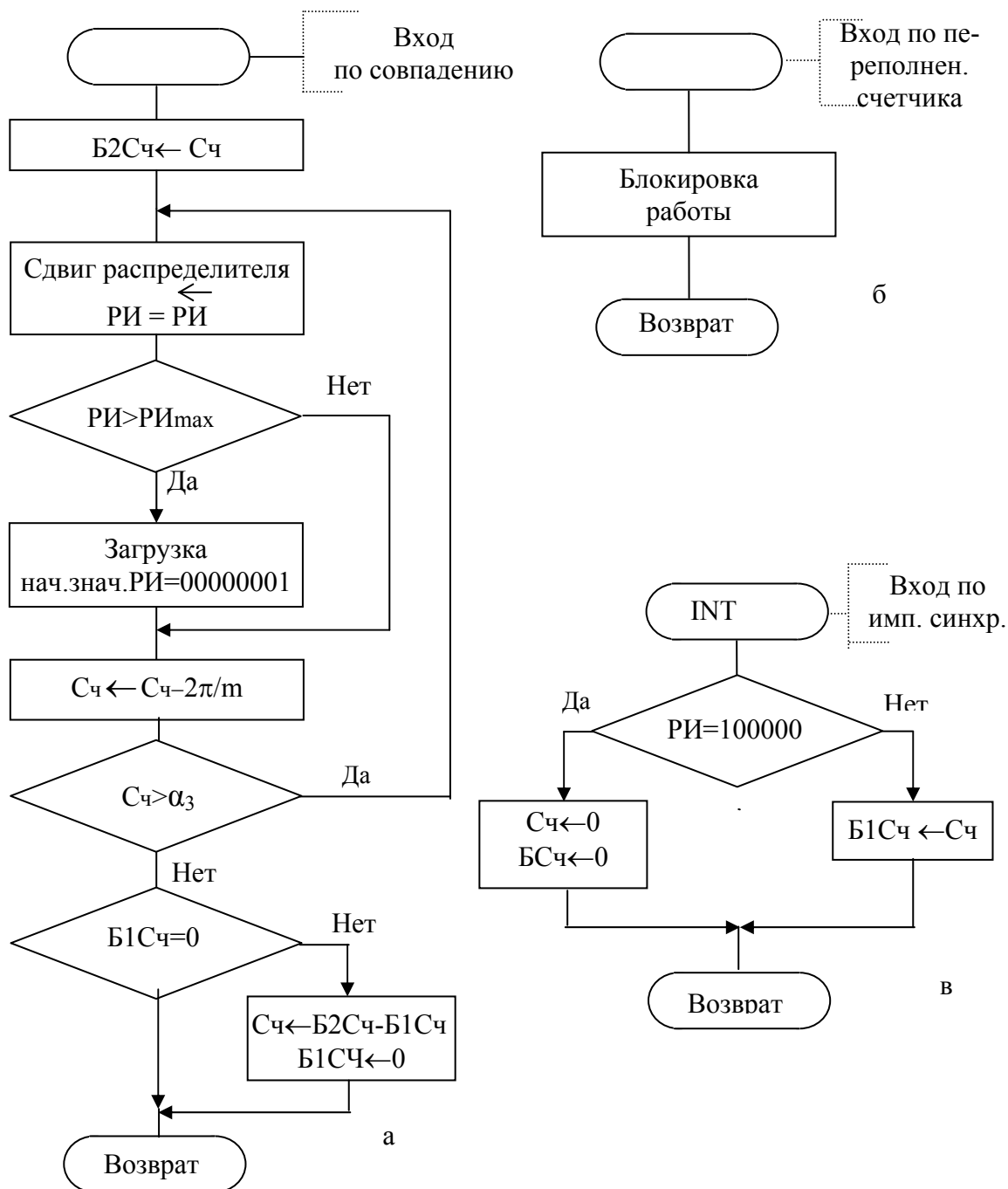


Рис. 3.20. Алгоритмы СИФУ

содержимого счетчика Сч. Значение угла  $\alpha_3$  определяется выходом регулятора тока и может изменяться в сторону увеличения или уменьшения.

В последнем случае перезагрузка Сч будет осуществляться до тех пор, пока его содержимое не станет меньше угла задания, РИ будет переходить в новое состояние и соответственно изменять состояние выходов порта. С учетом количества вентилей моста, равного 6, цикл работы распределителя неполный и после прохождения его через тах значение производится его установка в начальное состояние (рис. 3.20, в). Прерывание INT0 формируется один раз за период в точке естественной коммутации, например, тиристора фазы А. Если  $\alpha_3 < 2\pi/m$ , то последний (6-й) отпирающий импульс сформируется раньше, чем очередной импульс синхронизации. В этом случае счетчик блока таймера сбрасывается в нуль, чем достигается синхронизация. Если  $\alpha_3 > 2\pi/m$ , то импульс синхронизации сформируется раньше, чем сформируется последний отпирающий импульс, и сбросить счетчик нельзя. Поэтому в момент  $\alpha = 0$  производится запоминание содержимого Сч в буферной ячейке памяти Б1Сч, а после входа в алгоритм СИФУ производят вторичное запоминание значения Сч в буфере Б2Сч и по разнице буферных значений производят перезагрузку счетчика.

Операции чтения содержимого счетчика и его перезагрузки в ряде случаев целесообразно производить с остановкой таймера/счетчика. Для этого необходимо выбрать соответствующий режим работы через регистр управления блока таймера/счетчика. Операции перезагрузки в регистр данных порта или иные операции с данным регистром определяются при детализовке алгоритма или при написании программы. При разработке алгоритмов очень важно анализировать события, которые соответствуют внутренним и внешним прерываниям МК и его периферии. Каждое прерывание имеет свой приоритет, на время отработки прерывания другие прерывания можно замаскировать. Так, например, во время отработки алгоритма СИФУ целесообразно запретить прерывания от всех остальных устройств.

Рассмотренные примеры раскрывают некоторые возможности использования МК в цифровых системах управления. Для более полного ознакомления с принципами построения микропроцессорных систем управления, особенно приводами переменного тока, рекомендуется использовать сайты: [www.atmel.com](http://www.atmel.com), [www.atmel.ru](http://www.atmel.ru), [www.infineon.com](http://www.infineon.com), [www.analog.ru](http://www.analog.ru), [www.irf.com](http://www.irf.com), [www.motorola.com](http://www.motorola.com), [www.ti.com](http://www.ti.com).

#### 4. МИКРОКОНТРОЛЛЕРЫ

В настоящее время на рынке производителей микроконтроллеров лидирующее положение занимают ряд фирм: Atmel, AMD, Intel, Motorola, NEC, Philips Semiconductors, Texas Instruments, Infineon, Hitachi и др. Микроконтроллеры, как правило, имеют большое число портов ввода/вывода, встроенные многоканальные АЦП, программируемые таймеры/счетчики, каналы последовательной связи. Применительно к системам управления особое значение имеет развитая периферия МК, так как основные функции формирования управляющих сигналов и обработки сигналов обратных связей решаются на аппаратном уровне. Причем, с точки зрения потребителя, в ряде случаев различные семейства микроконтроллеров [18, 24, 25] имеют близкие характеристики, и основные отличия их заключаются в архитектуре вычислительного ядра и системе команд.

Область МП средств, предназначенная для управления двигателями, получила специальное название (Motor Control). С учетом тенденций и перспектив развития ЭП в последнее время особое внимание уделялось разработке микроконтроллеров, предназначенных для управления приводами переменного тока. К числу таких МК, специально разработанных для использования в системах управления частотно-регулируемого электропривода, относятся: MC68HC705M4, MC68HC708MP16 (Motorola), 8XC8196MC, 8XC8196MD, 8XC8196MH (Intel), C167CR (Infineon), TMS320F241 (Texas Instruments).

Среди микроконтроллеров общего назначения, приближающихся по свойствам своей периферии к группе Motor Control, можно отнести микроконтроллеры AVR фирмы Atmel [17, 18]. В рамках единой базовой архитектуры AVR-микроконтроллеры подразделяются на три семейства: «tiny», «classic», «mega». Микроконтроллеры этой серии приобрели большую популярность благодаря ценовой политике фирмы, доступности программно-аппаратных средств поддержки и свободно распространяемым программным обеспечением.

В связи с переходом в 2001–2002 г. фирмой «Atmel» на технологические нормы производства 0.35мкм изменена номенклатура выпускаемых микроконтроллеров семейства AVR®. В новое семейство вошли микроконтроллеры ATmega8515, ATmega8535 взамен микроконтроллеров AT90S8515, AT90S8535 семейства «classic» и другие представители семейств, нашедших сферы применения по своим потребительским свойствам.

Новые контроллеры имеют увеличенное в 2 раза быстродействие, до 10000 циклов перезаписи на страницу увеличен ресурс памяти команд (Flash) и добавлена возможность ее перепрограммирования встроенными средствами микроконтроллера. Также добавлены новые периферийные узлы, такие как:

- аппаратный умножитель 8\*8 (команда умножения выполняется за 2 такта);
- периферийный интерфейс TWI (I2C);
- интерфейс JTAG для внутрисхемной отладки и программирования;
- контроллер символьного ЖКИ (контроллер ATmega169L/V – без аналогов);
- система контроля напряжения питания.



Рассмотрим один из наиболее мощных представителей семейства «mega» микроконтроллер Atmega64/128, который может быть использован для реализации систем управления ЭП.

#### 4.1. МИКРОКОНТРОЛЛЕР ATMEGA128, ATMEGA128L

Отличительные особенности:

высокопроизводительный, 8-разрядный AVR-микроконтроллер с внутрисистемно программируемой флэш-памятью емкостью 128 кбайт;

RISC-архитектура, 133 инструкции, большинство из которых выполняются за один машинный цикл;

32 8-разр. регистров общего назначения + регистры управления встроенной периферией;

производительность до 16 млн. операций в секунду при тактовой частоте 16 МГц;

встроенное умножающее устройство выполняет умножение за 2 машинных цикла;

энергонезависимая память программ и данных;

износостойкость 128-ми кбайт внутрисистемно перепрограммируемой флэш-памяти – 1000 циклов запись/стирание;

опциональный загрузочный сектор с отдельной программируемой защитой;

внутрисистемное программирование встроенной загрузочной программой;

гарантированная двухоперационность – возможность чтения во время записи;

износостойкость 4 кбайт ЭСППЗУ – 100000 циклов запись/стирание;

встроенное статическое ОЗУ емкостью 4 кбайт;

опциональная возможность адресации внешней памяти размером до 64 кбайт;

программируемая защита кода программы;

интерфейс SPI для внутрисистемного программирования;

интерфейс JTAG (совместимость со стандартом IEEE 1149.1);

граничное сканирование в соответствии со стандартом JTAG;

обширная поддержка функций встроенной отладки;

программирование флэш-памяти, ЭСППЗУ, бит конфигурации и защиты через интерфейс JTAG.

Отличительные особенности периферийных устройств:

два 8-разр. таймера-счетчика с отдельными предделителями и режимами сравнения;

два расширенных 16-разр. таймера-счетчика с отдельными предделителями, режимами сравнения и режимами захвата;

счетчик реального времени с отдельным генератором;

два 8-разр. каналов ШИМ;

6 каналов ШИМ с программируемым разрешением от 2 до 16 разрядов;  
модулятор выходов сравнения;

8 мультиплексированных каналов 10-разрядного аналогово-цифрового преобразования: 8 несимметричных каналов, 7 дифференциальных каналов, 2 дифференциальных канала с выборочным усилением из 1х, 10х и 200х;

двухпроводной последовательный интерфейс, ориентированный на передачу данных в байтном формате;

два канала программируемых последовательных УСАПП;

последовательный интерфейс SPI с поддержкой режимов ведущий/подчиненный;

программируемый сторожевой таймер со встроенным генератором;  
встроенный аналоговый компаратор.

Специальные возможности микроконтроллера:

сброс при подаче питания и программируемая схема сброса при снижении напряжения питания;

встроенный калиброванный RC-генератор;

внешние и внутренние источники прерываний;

шесть режимов снижения энергопотребления: холостой ход (Idle), уменьшение шумов АЦП, экономичный (Power-save), выключение (Power-down), дежурный (Standby) и расширенный дежурный (Extended Standby);

программный выбор тактовой частоты;

конфигурационный бит для перевода в режим совместимости с ATmega103;

общее выключение подтягивающих резисторов на всех линиях портов ввода-вывода.

Ввод-вывод и корпуса: 53 программируемые линии ввода-вывода; 64 вывода, корпус TQFP.

Рабочие напряжения: 2.7 – 5.5 В для ATmega128L; 4.5 – 5.5В для ATmega128.

Градации по быстродействию: 0 – 8 МГц для ATmega128L; 0 – 16 МГц для ATmega128.

#### 4.1.1. Краткое описание

ATmega128 – 8-разр. КМОП микроконтроллер (рис. 4.1), основанный на расширенной AVR RISC-архитектуре. ATmega128 содержит 128 кбайт внутрисистемно программируемой флэш-памяти с поддержкой чтения во время записи, 4 кбайт ЭСППЗУ, 4 кбайт статического ОЗУ, что предоставляет широкие возможности разработчику при разработке программного обеспечения: размещения функций настройки, табличных данных, текстовых сообщений, отладчика и загрузчика. Расширенный интерфейс и увеличение числа выводов семейства «mega» существенно расширяет диапазон применения во встроенных системах управления и обмену информации с внешними устройствами. Так, например, можно организовать взаимодействие с центральной микроЭВМ через поч-

товые ящики или в простейшем случае через регистры данных и состояния. Дополнительные возможности организации многопроцессорных систем связаны с использованием последовательных портов (SPI), асинхронных приемопередатчиков (УСАПП) и двухпроводного интерфейса последовательной передачи.

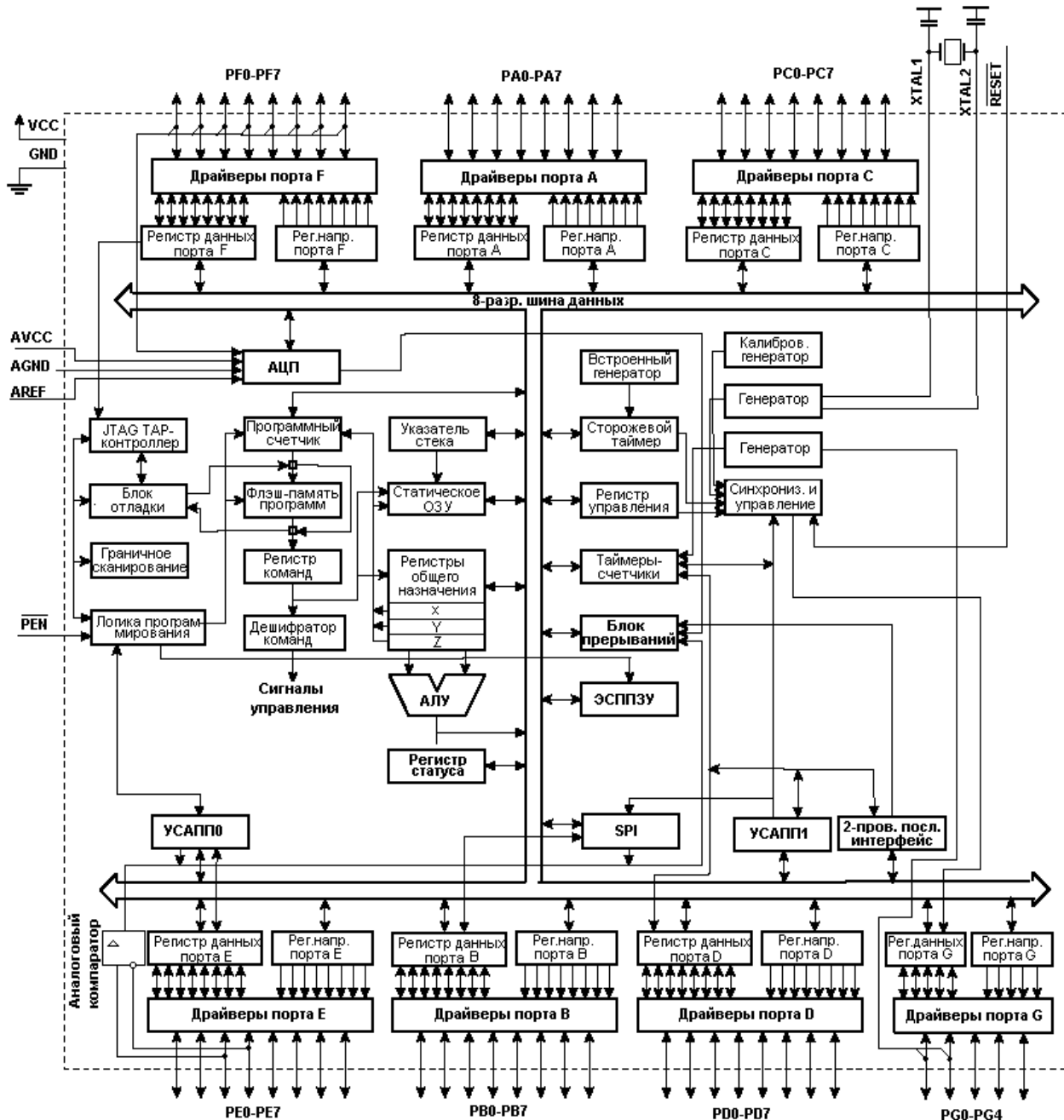


Рис. 4.1. Функциональная схема Atmega128

Периферия ATmega128 содержит 53 линии универсального ввода-вывода, многие из которых имеют альтернативные функции. Для реализации систем импульсно-фазового управления (СИФУ) и широтно-импульсных модуляторов могут быть использованы четыре гибких таймера-счетчика с режимами сравнения и ШИМ. Восьмиканальный 10-разрядный АЦП с опциональным дифференциальным входом и программируемым коэффициентом усиления позволяет

обрабатывать сигналы задатчиков и датчиков обратных связей, имеющих как однополярные, так и двухполярные уровни сигналов. Интерфейс JTAG, совместимый со стандартом IEEE 1149.1, предоставляет широкие возможности при отладке программы и программировании. Для исключения зависаний программы используется программируемый сторожевой таймер с внутренним генератором. Дополнительные возможности при сборе и обработке информации дает использование счетчика реального времени (RTC). Кристалл МК имеет шесть программно выбираемых режимов энергопотребления. Режим холостого хода (Idle) останавливает центральное процессорное устройство (ЦПУ), но при этом поддерживая работу статического ОЗУ, таймеров-счетчиков, SPI-порта и системы прерываний. Режим выключения (Powerdown) позволяет сохранить содержимое регистров, при остановленном генераторе и выключении встроенных функций до следующего прерывания или аппаратного сброса. В экономичном режиме (Power-save) асинхронный таймер продолжает работу, позволяя пользователю сохранить функцию счета времени в то время, когда остальная часть контроллера находится в состоянии сна. Режим снижения шумов АЦП (ADC Noise Reduction) останавливает ЦПУ и все модули ввода-вывода, кроме асинхронного таймера и АЦП для минимизации импульсных шумов в процессе преобразования АЦП. В дежурном режиме (Standby) кварцевый/резонаторный генератор продолжает работу, а остальная часть микроконтроллера находится в режиме сна. Данный режим характеризуется малой потребляемой мощностью, но при этом позволяет достичь самого быстрого возврата в рабочий режим. В расширенном дежурном режиме (Extended Standby) основной генератор и асинхронный таймер продолжают работать.

Встроенная программируемая флэш-память позволяет перепрограммировать память программ непосредственно внутри системы через последовательный интерфейс SPI с помощью простого программатора или с помощью автономной программы в загрузочном секторе. Загрузочная программа может использовать любой интерфейс для загрузки прикладной программы во флэш-память. Программа в загрузочном секторе продолжает работу в процессе обновления прикладной секции флэш-памяти, тем самым поддерживая операции чтения во время записи.

ATmega128 поддерживается полным набором программных и аппаратных средств для проектирования: Си-компиляторы, макроассемблеры, внутрисистемные эмуляторы, программные отладчики/симуляторы и оценочные наборы.

**Расположение выводов.** ATmega128 полностью совместим по расположению выводов (рис. 4.2, табл. 4.1.) с ATmega103 и может быть установлен на существующую плату для ATmega103. Если запрограммировать конфигурационный бит M103C, то ATmega128 будет функционировать в режиме совместимости с ATmega103, настраивая ОЗУ, линии ввода-вывода и векторы прерываний. Основные функции линий портов – дискретный ввод/вывод информации. Дополнительные (альтернативные) функции линий портов, предназначенные для подключения входов/выходов периферийных устройств МК, показаны на рис. 4.2 в скобках.

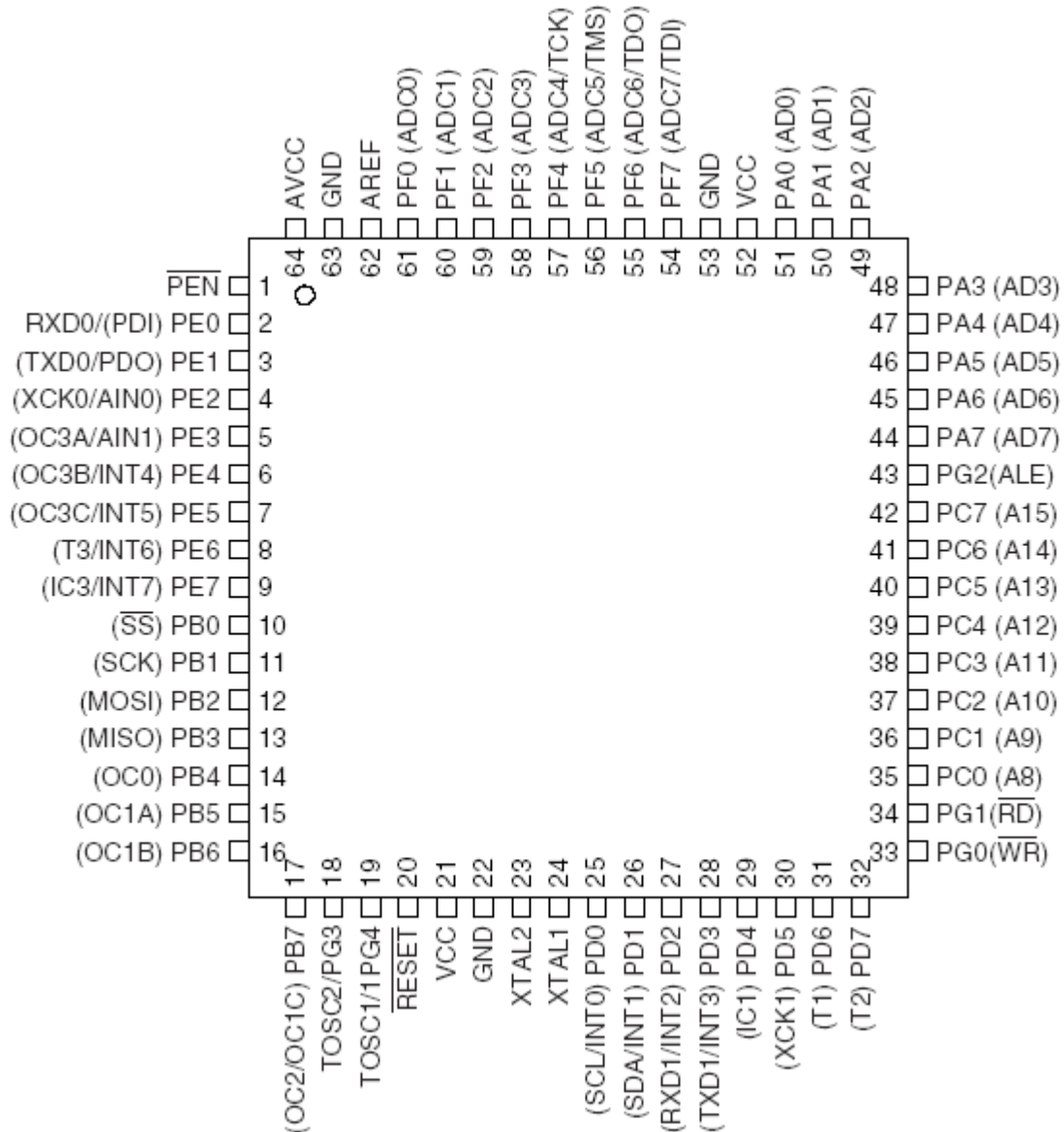


Рис. 4.2. Расположение выводов у ATmega128

Таблица 4.1

## Описание выводов

V <sub>CC</sub>	Напряжение питания цифровых элементов
GND	Общий
Порт А (PA7..PA0)	Порт двунаправленного ввода-вывода с внутренними подтягивающими к плюсу резисторами (выбираются отдельно для каждого разряда). Выходные буферы порта имеют симметричную выходную характеристику с одинаковыми втекающим и вытекающим токами. При вводе линии порта будут действовать как источник тока, если внешне действует низкий уровень и включены подтягивающие резисторы. Выводы порта находятся в третьем (высокоимпедансном) состоянии при выполнении условия сброса, даже если синхронизация не запущена. Специальные функции линий соответствующего порта описаны далее.
Порт В (PB7..PB0)	
Порт С (PC7..PC0)	
Порт D (PD7..PD0)	
Порт Е (PE7..PE0)	

Окончание табл. 4.1

Порт F (PF7..PF0)	Порт F действует как аналоговый ввод аналогово-цифрового преобразователя. Порт F также может использоваться как 8-разр. Порт двунаправленного ввода-вывода, если АЦП не используется. К каждой линии порта может быть подключен встроенный подтягивающий к плюсу резистор (выбирается отдельно для каждого бита). Выходные буферы порта F имеют симметричную выходную характеристику с одинаковыми втекающим и вытекающим токами. При вводе линии порта F будут действовать как источник тока, если внешне действует низкий уровень и включены подтягивающие резисторы. Выводы порта F находятся в третьем (высокоимпедансном) состоянии при выполнении условия сброса, даже если синхронизация не запущена. Если активизирован интерфейс JTAG, то подтягивающие резисторы на линиях PF7(TDI), PF5(TMS) и PF4(TCK) будут подключены, даже если выполняется Сброс. Вывод TDO находится в третьем состоянии, если не введено состояние TAP, при котором сдвигаются выводимые данные. Порт F также выполняет функции интерфейса JTAG. В режиме совместимости с Atmega103 порт F действует только на ввод.
Порт G (PG4..PG0)	Порт G – 5-разр. Порт двунаправленного ввода-вывода с внутренними подтягивающими к плюсу резисторами (выбираются отдельно для каждого разряда). Выходные буферы порта G имеют симметричную выходную характеристику с одинаковыми втекающим и вытекающим токами. При вводе линии порта G будут действовать как источник тока, если внешне действует низкий уровень и включены подтягивающие резисторы. Выводы порта G находятся в третьем (высокоимпедансном) состоянии при выполнении условия сброса, даже если синхронизация не запущена. Порт G также выполняет некоторые специальные функции Atmega128. В режиме совместимости с Atmega103 данные выводы используются как стробирующие сигналы интерфейса внешней памяти, а также как вход генератора 32 кГц, а при действии сброса они асинхронно принимают следующие состояния: PG0 = 1, PG1 = 1 и PG2 = 0, даже если синхронизация не запущена. PG3 и PG4 – выводы генератора.
RESET	Вход сброса. Если на этот вход приложить низкий уровень длительностью более минимально необходимой будет генерирован сброс независимо от работы синхронизации.
XTAL1	Вход инвертирующего усилителя генератора и вход внешней синхронизации.
XTAL2	Выход инвертирующего усилителя генератора.
AVCC	Вход питания порта F и аналогово-цифрового преобразователя. Он должен быть внешне связан с VCC, даже если АЦП не используется. При использовании АЦП этот вывод связан с VCC через фильтр низких частот.
AREF	Вход подключения источника опорного напряжения АЦП.
PEN	Вход разрешения программирования через интерфейс SPI. Микроконтроллер переходит в режим последовательного программирования через SPI, если во время действия сброса на этот вход подать низкий уровень. В рабочем режиме PEN не выполняет никаких функций.

#### 4.1.2. Ядро центрального процессорного устройства AVR

Основная функция ядра ЦПУ заключается в организации взаимодействия всех модулей в процессе выполнения программы и ее гарантированного выполнения. Для этого ЦПУ должен иметь возможность адресоваться к различным видам памяти, выполнять вычисления, управлять периферийными устройствами и обрабатывать прерывания.

В целях достижения максимальной производительности у AVR-микроконтроллеров используется Гарвардская архитектура (рис. 4.3) с отдельными памятью и шинами программ и данных. Команды выполняются с двухуровневой конвейеризацией. Данная концепция позволяет выполнять одну инструкцию за один машинный цикл.

АЛУ поддерживает арифметические и логические операции между регистрами, а также между константой и регистром. Кроме того, АЛУ поддерживает действия с одним регистром.

Регистровый файл с быстрым доступом содержит 32 восьмиразрядных рабочих регистра общего назначения, шесть из которых могут использоваться как три 16-разр. регистра косвенного адреса для эффективной адресации в пределах памяти данных. Данные 16-разр. регистра называются X-регистр, Y-регистр и Z-регистр.

Для ветвления программы поддерживаются инструкции условных и безусловных переходов и вызовов процедур, позволяющих непосредственно адресоваться в пределах адресного пространства.

Большинство инструкций представляют собой одно 16-разр. слово. Каждый адрес памяти программ содержит 16- или 32-разр. инструкцию. Флэш-память программ разделена на две секции: секция программы начальной загрузки и секция прикладной программы. Обе секции имеют отдельные биты защиты от записи и чтения/записи. Инструкция SPM (запись в секцию прикладной программы) должна использоваться только внутри секции программы начальной загрузки.

При генерации прерывания и вызове подпрограмм адрес возврата из программного счетчика записывается в стек. Размер стека ограничен общим размером статического ОЗУ и используемым его объемом. Указатель стека – SP – доступен на чтение и запись в пространстве ввода-вывода. Доступ к статическому ОЗУ данных может быть осуществлен через 5 различных режимов адресации архитектуры AVR.

Модуль прерываний содержит свои управляющие регистры в пространстве ввода-вывода. Бит общего разрешения работы системы прерываний находится в регистре статуса. Каждое из прерываний имеет свой вектор, адреса которых соответствуют таблице векторов прерываний. Прерывания имеют приоритет в соответствии с позицией их вектора. Прерывания с меньшим адресом имеют более высокий приоритет.

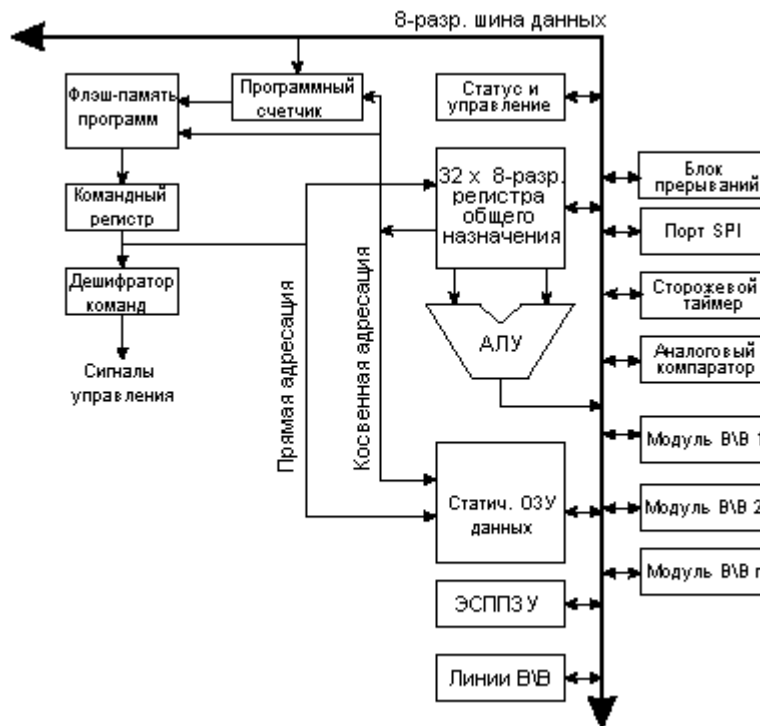


Рис. 4.3. Функциональная схема архитектуры AVR

Пространство памяти ввода-вывода содержит 64 адреса с непосредственной адресацией или может адресоваться как память данных, следующая за регистрами по адресам \$20-\$5F. Кроме того, ATmega128 имеет пространство расширенного ввода-вывода по адресам \$60-\$FF в статическом ОЗУ, для доступа к которому могут использоваться только процедуры ST/STS/STD и LD/LDS/LDD.

**АЛУ – арифметико-логическое устройство.** Высокопроизводительное АЛУ AVR-микроконтроллеров имеет непосредственный доступ ко всем 32 универсальным рабочим регистрам. АЛУ позволяет выполнить за один машинный цикл операцию между двумя регистрами или между регистром и константой. Операции АЛУ могут быть классифицированы на три группы: арифметические, логические и битовые. Кроме того, архитектурой ATmega128 поддерживаются операции умножения со знаком и без знака и дробным форматом.

**Регистр статуса.** Регистр статуса содержит информацию о результате только что выполненной арифметической инструкции в виде флагов. Данная информация может использоваться для ветвления программы по условию. Флаги обновляются, если инструкция предусматривает их формирование. Флаги этого регистра в большинстве случаев позволяют отказаться от использования инструкций сравнения, делая код программы более компактным и быстрым.

Регистр статуса SREG AVR-микроконтроллера имеет следующую структуру:

[illegible]



Разряд 7 – I: Общее разрешение прерываний.

Бит общего разрешения прерываний используется для активизации работы системы прерываний. Разрешение отдельных прерываний осуществляется в соответствующих управляющих регистрах. Если бит общего разрешения прерываний сбросить, то ни одно из прерываний не будет активным независимо от их индивидуальной конфигурации. Бит I сбрасывается в 0 аппаратно после генерации запроса на прерывание, а после выполнения инструкции возврата из прерывания RETI снова устанавливается к 1 для выполнения последующих прерываний. Бит I может также сбрасываться и устанавливаться с помощью инструкций CLI и SEI соответственно.

Разряд 6 – T: Хранение копируемого бита.

Специальные битовые операции BLD (копирование из T-бита) и BST (копирование в T-бит) используют в качестве источника и получателя данных бит T. Любой бит из регистрового файла может быть скопирован в бит T инструкцией BST, а также содержимое бита T может быть скопировано в любой бит регистрового файла с помощью инструкции BLD.

Разряд 5 – H: Флаг половинного переноса.

Данный флаг устанавливается при выполнении некоторых арифметических инструкций и индицирует о возникновении половинного переноса. Данный перенос широко используется в двоично-десятичной арифметике.

Разряд 4 – S. Бит знака, S = Искл. ИЛИ (N, V).

Бит S – результат выполнения логической операции исключающего ИЛИ между флагом отрицательного результата N и флагом переполнения двоичного дополнения V.

Разряд 3 – V. Флаг переполнения двоичного дополнения.

Флаг переполнения двоичного дополнения V поддерживает арифметику с двоичным дополнением.

Разряд 2 – N. Флаг отрицательного результата.

Флаг отрицательного результата N индицирует, что результатом выполнения арифметической или логической операции является отрицательное значение.

Разряд 1 – Z. Флаг нулевого результата.

Флаг нулевого результата Z индицирует, что результатом выполнения арифметической или логической операции является ноль.

Разряд 0 – C. Флаг переноса.

Флаг переноса C индицирует о возникновении переноса в результате выполнения арифметической или логической операции.

**Файл регистров общего назначения.** Файл регистров оптимизирован под расширенный набор инструкций AVR-микроконтроллеров. В целях достижения требуемой производительности и гибкости файлом регистров поддерживаются следующие схемы ввода-вывода:

один 8-разр. операнд и один 8-разр. результат;

два 8-разр. операнда и один 8-разр. результат;  
 два 8-разр. операнда и один 16-разр. результат;  
 один 16-разр. операнд и один 16-разр. результат.

На рисунке 4.4 показана структура 32 рабочих регистров общего назначения в ЦПУ.

	7	0	Адрес	
	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	Мл. байт X-регистра
	R27		\$1B	Ст. байт X-регистра
	R28		\$1C	Мл. байт Y-регистра
	R29		\$1D	Ст. байт Y-регистра
	R30		\$1E	Мл. байт Z-регистра
	R31		\$1F	Ст. байт Z-регистра

Рис. 4.4. Рабочие регистры общего назначения ЦПУ AVR

Большинство инструкций, работающих с файлом регистров, имеют непосредственный доступ ко всем регистрам, чем достигается выполнение их за один машинный цикл.

Как показано на рисунке 3.4, каждый регистр имеет свой адрес в области памяти данных, для чего отведено там первые 32 позиции. Регистры X, Y и Z могут быть использованы в качестве указателей ячеек памяти.

**X-регистр, Y-регистр и Z-регистр.** Регистры R26..R31 обладают некоторым дополнительными функциями для их общецелевого использования. Данные регистры являются 16-разр. указателями адреса для косвенной адресации в пределах памяти данных.

Три регистра косвенной адресации X, Y и Z представлены на рисунке 4.5.

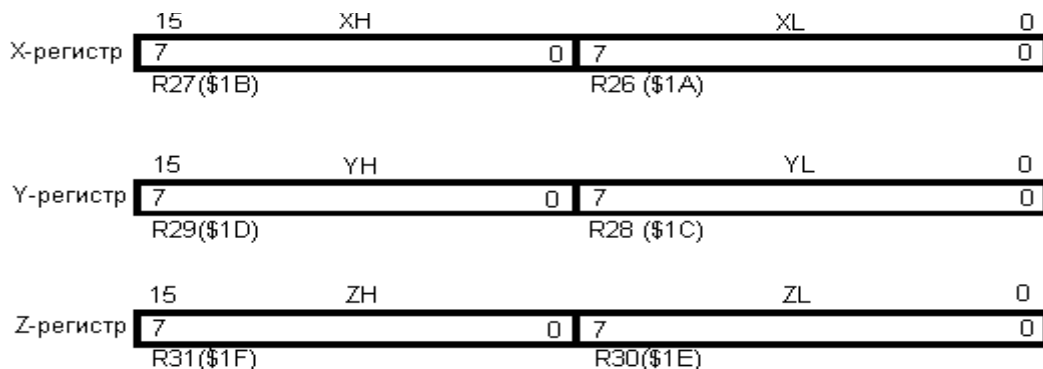


Рис. 4.5. X, Y и Z-регистры

В различных режимах адресации данные адресные регистры выполняют функции фиксированного смещения, автоматического инкрементирования и автоматического декрементирования.

**Стек** обычно используется для хранения временных данных, для хранения локальных переменных и для хранения адресов возврата при прерываниях и вызовах подпрограмм. Регистр указателя стека указывает на вершину стека. При отработке команды PUSH (сохранения байта в стеке) указатель стека уменьшает свое значение.

Указатель стека реализован как два 8-разр. регистра в области ввода-вывода. Число фактически используемых разрядов зависит от типа микроконтроллера. У некоторых AVR-микроконтроллеров область памяти данных настолько мала, что достаточно только регистра SPL. В этом случае регистр SPH отсутствует.

Разряды	15	14	13	12	11	10	9	8	
	<b>SP15</b>	<b>SP14</b>	<b>SP13</b>	<b>SP12</b>	<b>SP11</b>	<b>SP10</b>	<b>SP9</b>	<b>SP8</b>	<b>SPH</b>
	<b>SP7</b>	<b>SP6</b>	<b>SP5</b>	<b>SP4</b>	<b>SP3</b>	<b>SP2</b>	<b>SP1</b>	<b>SP0</b>	<b>SPL</b>
	7	6	5	4	3	2	1	0	
Чтение/Запись	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	
	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	
Начальное значение	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Разряды 7...2 – зарезервированные разряды. Данные зарезервированные разряды считываются как 0. При записи в данные разряды необходимо записывать нули для совместимости с последующими микроконтроллерами.

#### Регистр выбора Z-страницы ОЗУ – RAMPZ

Разряд	7	6	5	4	3	2	1	0	
	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>RAMPZ0</b>	<b>RAMPZ</b>
Чтение/Запись	Чт	Чт	Чт	Чт	Чт	Чт	Чт	Чт/Зп	
Начальное значение	0	0	0	0	0	0	0	0	

Разряд 1 – RAMPZ0: Расширенный указатель страницы ОЗУ

Регистр RAMPZ обычно используется для указания той страницы ОЗУ размером 64 кбайт, к которой выполняется доступ через Z-указатель. Т. к. ATmega128 не поддерживает память на статическом ОЗУ размером свыше 64 кбайт, то данный регистр используется только для выбора страницы памяти программ, доступ к которой осуществляется с помощью инструкций ELPM/SPM. Различные установки бита RAMPZ0 дают следующий результат:

RAMPZ0 = 0:	Инструкции ELPM/SPM осуществляют доступ к памяти программ в диапазоне адресов \$0000 - \$7FFF (младшие 64 кбайт)
RAMPZ0 = 1:	Инструкции ELPM/SPM выполняют доступ к памяти программ в диапазоне адресов \$8000 - \$FFFF (старшие 64 кбайт)

Обратите внимание, что действие инструкции LPM не зависит от установки RAMPZ.

**Сброс и обработка прерываний.** Все прерывания имеют свой индивидуальный вектор в памяти программ. Для каждого прерывания имеется собственный бит разрешения. Кроме того, имеется возможность общего разрешения работы прерываний с помощью установки соответствующего бита в статусном регистре. В зависимости от значения программного счетчика прерывания могут быть автоматически отключены, если запрограммировать биты защиты загрузочного сектора BLB02 или BLB12.

Сброс (RESET) имеет наивысший приоритет, за ним следует INT0 – запрос на внешнее прерывание по входу INT0. Векторы прерывания могут быть перемещены в начало загрузочного сектора флэш-памяти установкой бита IVSEL в регистре управления микроконтроллером (MCUCR). Вектор сброса может быть также перемещен в начало загрузочного сектора флэш-памяти путем программирования конфигурационного бита BOOTRST.

После возникновения прерывания бит I общего разрешения прерываний сбрасывается и все прерывания запрещаются. Пользователь может программно записать лог. 1 в бит I для разрешения вложенных прерываний. В этом случае все разрешенные прерывания могут прервать текущую процедуру обработки прерываний. Бит I автоматически устанавливается после выполнения инструкции выхода из прерывания RETI.

Имеется два основных типа прерываний. Первый тип прерываний активизируется событием, которое приводит к установке флага прерываний. Для данных прерываний программный счетчик загружается адресом соответствующего вектора прерывания для выполнения процедуры его обработки и затем аппаратно очищает флаг прерывания. Флаги прерывания также сбрасываются путем записи лог. 1 в соответствующий разряд. Если возникает условие прерывания, но данное прерывание запрещено, то флаг устанавливается и запоминается до разрешения этого прерывания или сбрасывается программно. Аналогично, если возникает одно и более условий прерываний при сброшенном флаге общего разрешения прерываний, то соответствующий флаг устанавливается и запоминается до возобновления работы прерываний, а затем прерывания будут выполнены в соответствии с приоритетом.

Второй тип прерываний активизируется сразу после выполнения условия прерывания. Данные прерывания не обязательно имеют флаги прерываний. Если условие прерывания исчезает до его разрешения, то данный запрос игнорируется.

После выхода из прерывания AVR-микроконтроллер возвращается к выполнению основной программы и выполняет еще одну инструкцию до обслуживания любого из отложенных прерываний.

При выполнении инструкции CLI все прерывания запрещаются. Запрос на прерывание не будет отработан после выполнения инструкции CLI, даже если оно возникает одновременно с выполнением команды CLI.

Для разрешения прерываний используется инструкция SEI, а следующая за SEI инструкция будет выполнена перед обработкой любого отложенного прерывания.

#### 4.1.3. Память

Память AVR-микроконтроллера разделена на две области: память данных и память программ. Кроме того, ATmega128 содержит память на ЭСППЗУ для энергонезависимого хранения данных. Все три области памяти являются линейными и равномерными.

Программируемая память программ ATmega128 содержит 128 кбайт внутренней, внутрисистемно перепрограммируемой флэш-памяти. Поскольку все AVR-инструкции являются 16 или 32-разр., то флэш-память организована как 64 кбайт\*16. Для программной защиты флэш-память программ разделена на два сектора: сектор программы начальной загрузки и сектор прикладной программы.

Программный счетчик PC у ATmega128 является 16-разр., что позволяет адресоваться к 64 кбайт памяти программ. Программирование памяти осуществляется через интерфейсы SPI, JTAG.

**Статическое ОЗУ памяти данных.** ATmega128 поддерживает две различные конфигурации статического ОЗУ памяти данных (см. табл. 4.2).

ATmega128 – сложный микроконтроллер с большим числом периферийных устройств, которые управляются через 64 ячейки памяти, зарезервированных в кодах операций инструкций IN и OUT. Для расширенной области ввода-вывода в статическом ОЗУ по адресам \$60-\$FF необходимо использовать только инструкции ST/STS/STD и LD/LDS/LDD. Область расширенного ввода-вывода не существует при переводе ATmega128 в режим совместимости с ATmega103.

Таблица 4.2

Конфигурации памяти

Конфигурация	Встроенное статическое ОЗУ памяти данных	Внешнее статическое ОЗУ памяти данных
Нормальный режим	4096	до 64 кбайт
Режим совместимости с ATmega103	4000	до 64 кбайт

В нормальном режиме первые 4352 ячейки памяти данных относятся к файлу регистров, памяти ввода-вывода, расширенной памяти ввода-вывода и встроенному статическому ОЗУ данных. В первых 32 ячейках расположен файл регистров, следующие 64 ячейки занимает стандартная память ввода-вывода, а

за ними следуют 160 ячеек расширенной памяти ввода-вывода. Замыкают внутреннюю память данных 4096 ячеек внутреннего статического ОЗУ данных.

Совместно с ATmega128 по выбору может использоваться статическое ОЗУ. Это статическое ОЗУ будет занимать оставшуюся часть от адресного пространства размером 64 кбайт. При использовании внешней памяти размером 64 кбайт (65536 байт) будет доступно 61184 байта в нормальном режиме и 61440 байта в режиме совместимости с ATmega103.

Доступ к внешнему статическому ОЗУ осуществляется автоматически с помощью тех же инструкций, что и для внутреннего ОЗУ, если указанное значение адреса находится за пределами внутренней памяти данных. При адресации внутренней памяти сигналы чтения и записи внешней памяти (выводы PG0 и PG1) неактивны в процессе всего цикла доступа. Работа внешнего статического ОЗУ разрешается путем установки бита SRE в регистре MCUCR.

Доступ к внешнему статическому ОЗУ требует еще одного машинного цикла на байт по сравнению с доступом к внутреннему статическому ОЗУ. Это означает, что на выполнение команд LD, ST, LDS, STS, LDD, STD, PUSH и POP потребуется один дополнительный цикл. Если стек будет размещен во внешнем статическом ОЗУ, то, соответственно, вызов и возврат из подпрограмм и процедур обработки прерываний будет длиться на три машинных цикла дольше за счет помещения в стек и извлечения из стека двухбайтного счетчика программы и неиспользования во время доступа к внешней памяти преимущества конвейерного доступа к внутренней памяти. Если интерфейс внешнего статического ОЗУ используется с состояниями ожидания (со сниженным быстродействием), то однобайтный внешний доступ потребует 2, 3 или 4 дополнительных машинных цикла для 1, 2 и 3 состояний ожиданий, соответственно. Таким образом, вызов и возврат из прерываний и подпрограмм потребует еще 5, 7 и 9 машинных циклов (в отличие от значений приведенных в описании набора инструкций) для 1, 2 и 3 состояний ожидания соответственно.

Реализовано пять различных способов адресации для охвата всей памяти: прямая, косвенная со смещением, косвенная, косвенная с предварительным декрементом и косвенная с последующим инкрементом. Регистры R26...R31 из файла регистров используются как регистры-указатели для косвенной адресации.

Прямая адресация позволяет адресоваться ко всей памяти данных.

Косвенная адресация со смещением позволяет адресовать 63 ячейки, начиная с адреса указанного в регистрах Y или Z.

При использовании инструкции косвенной адресации с предварительным декрементом и последующим инкрементом значения адресных регистров X, Y и Z, соответственно декрементируются до или инкрементируются после выполнения инструкции.

32 рабочих регистров общего назначения, 64 регистра ввода-вывода и 4096 байт внутреннего статического ОЗУ данных в ATmega128 доступны с помощью всех этих режимов адресации.

**Память данных на ЭСППЗУ.** ATmega128 содержит 4 кбайт памяти данных на ЭСППЗУ. Она организована как отдельная область памяти данных, в которой один байт может быть записан и считан. ЭСППЗУ характеризуется износостойкостью 100000 циклов чтения/записи.

Программирование ЭСППЗУ осуществляется через интерфейсы SPI, JTAG или параллельное программирование.

Доступ к ЭСППЗУ осуществляется через специальные регистры, расположенные в пространстве ввода-вывода.

В целях предотвращения неумышленной записи в ЭСППЗУ должна быть выполнена специфическая процедура записи.

Когда происходит считывание ЭСППЗУ ЦПУ задерживается на 4 машинных цикла до выполнения следующей инструкции. Во время записи в ЭСППЗУ ЦПУ задерживается на два машинных цикла до выполнения следующей инструкции.

**Адресные регистры ЭСППЗУ – EEARH и EEARL**

Разряд	15	14	13	12	11	10	9	8	
	–	–	–	–	EEAR11	EEAR10	EEAR9	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Чтение/запись	Чт.	Чт.	Чт.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Начальное значение	0	0	0	0	X	X	X	X	
	X	X	X	X	X	X	X	X	

Разряды 15..12 – Резерв

Данные зарезервированные разряды считываются как 0. При записи в данных разрядах необходимо указывать нули для совместимости с новыми версиями микроконтроллеров.

Разряды 11..0 – EEAR11..0: Адрес ячейки ЭСППЗУ

Регистры адреса ЭСППЗУ – EEARH и EEARL – определяют адрес ячейки ЭСППЗУ в 4 кбайтном пространстве. Байтные ячейки ЭСППЗУ адресуются линейно в диапазоне адресов 0...4096. Начальное значение EEAR неопределенное. Необходимое значение адреса должно быть записано до начала доступа к ЭСППЗУ.

**Регистр данных ЭСППЗУ – EEDR**

Разряд	7	6	5	4	3	2	1	0	
	Ст.							Мл.	EEDR <sup>1</sup>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	R/W <sup>1</sup>	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряды 7...0 – EEDR7..0: Данные ЭСППЗУ

Для выполнения записи в ЭСППЗУ в регистр EEDR необходимо указать записываемые данные, которые будут записаны по адресу, указанному в реги-

стре EEAR. После выполнения чтения из ЭСППЗУ в регистре EEDR содержатся считанные данные из ячейки по адресу указанному в EEAR.

Регистр управления ЭСППЗУ – EECR								EECR
Разряд	7	6	5	4	3	2	1	0
	–	–	–	–	EERIE	EEMWE	EEWE	EERE
Чтение/запись	Чт.	Чт.	Чт.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.
Исх.знач.	0	0	0	0	0	0	x	0

Разряды 7...4 – Резерв. Данные разряды у ATmega128 зарезервированы и считываются как 0.

Разряд 3 – EERIE: Разрешение прерывания по готовности ЭСППЗУ

Запись в EERIE 1 разрешает прерывание по готовности ЭСППЗУ, если кроме того установлен бит I в регистре SREG. Запись в EERIE нуля отключает это прерывание. Прерывание по готовности ЭСППЗУ генерируется, если бит EEWE сброшен.

Разряд 2 – EEMWE: Главное разрешение записи в ЭСППЗУ

Бит EEMWE разрешает установку бита EEWE, инициирующего запись в ЭСППЗУ. Данные будут записаны в ЭСППЗУ по указанному адресу, если в EEMWE записать 1, а затем в течение 4 машинных циклов записать 1 в EEWE. Если EEMWE = 0, то запись в EEWE лог. 1 не вызовет никаких действий. После программной установки бита EEMWE он автоматически сбрасывается аппаратно по истечении четырех машинных циклов.

Разряд 1 – EEWE: Разрешение записи в ЭСППЗУ

Сигнал разрешения записи EEWE является стробирующим сигналом записи для ЭСППЗУ. Для записи в ЭСППЗУ после корректной установки адреса и данных необходимо установить бит EEWE. Перед установкой бита EEWE должен быть установлен бит EEMWE, иначе запись в ЭСППЗУ не произойдет. При выполнении операции записи в ЭСППЗУ необходимо руководствоваться следующей последовательностью (порядок шагов 3 и 4 не важен):

1. Ожидание, пока EEWE не станет равным нулю
2. Ожидание равенства нулю бита SPMEN в регистре SPMCSR
3. Запись нового адреса ЭСППЗУ в EEAR (опционально)
4. Запись новых данных в регистр EEDR для записи в ЭСППЗУ (опционально)
5. Запись лог. 1 в EEMWE, когда в EEWE регистра EECR записан ноль
6. Запись лог. 1 в EEWE в течение четырех машинных циклов после установки EEMWE.

ЭСППЗУ нельзя программировать во время записи флэш-памяти из ЦПУ. С учетом этого, перед началом новой записи в ЭСППЗУ необходимо проверить завершение программирования флэш-памяти. Шаг 2 необходимо выполнять, если в приложении используется программирование из загрузочного сектора. Если программирование флэш-памяти под управлением ЦПУ не предусмотрено, то шаг 2 может быть исключен. Предостережения: Прерывание между шагами 5 и 6 может нарушить цикл записи из-за превышения установленного предела времени на выполнение этих шагов. Если процедура обработки прерыва-



ния, осуществляющая доступ к ЭСППЗУ, прерывается другим доступом к ЭСППЗУ, то EEAR или EEDR будут изменены, вызывая сбой прерванного цикла доступа. Во избежание этих проблем рекомендуется сбрасывать флаг общего разрешения прерываний при выполнении последних четырех шагов.

По окончании записи бит EEWЕ сбрасывается аппаратно. Данный бит может опрашиваться программно для определения возможности записи следующего байта (нулевое значение). После установки EEWЕ ЦПУ останавливается на два машинных цикла перед выполнением следующей инструкции.

**Разряд 0 – EERE:** Разрешение чтения из ЭСППЗУ

Сигнал разрешения чтения из ЭСППЗУ EERE является стробом чтения ЭСППЗУ. После записи корректного адреса в регистр адреса EEAR бит EERE должен быть установлен к лог.1 для запуска механизма чтения ЭСППЗУ. Чтение из ЭСППЗУ выполняется одновременно с выполнением инструкции, поэтому запрашиваемые данные готовы для считывания сразу по ее завершении. После чтения из ЭСППЗУ ЦПУ задерживается на четыре машинных цикла, а только затем выполняет следующую инструкцию.

Пользователь должен опросить флаг EEWЕ до начала операции чтения. Если осуществляется операция записи, то невозможно не только считать ЭСППЗУ, но и изменить регистр адреса EEAR.

Повреждение данных в ЭСППЗУ может быть легко предотвращено, если придерживаться следующих рекомендаций:

Микроконтроллер необходимо удерживать в состоянии сброса (низкий уровень на выводе RESET) при недостаточности уровня питания. Аналогично это можно выполнить, разрешив работу встроенного детектора питания (BOD). Если пороговый уровень встроенного детектора питания не соответствует необходимому порогу, то следует применить внешнюю схему сброса при снижении VCC (супервизор питания). Если сброс возникает во время действия операции записи, то запись будет завершена при условии достаточности уровня питания.

**Память ввода-вывода.** Все порты ввода-вывода и периферийные устройства в ATmega128 размещены в пространстве ввода-вывода. Доступ ко всем ячейкам ввода-вывода может быть осуществлен с помощью инструкций LD/LDS/LDD и ST/STS/STD путем передачи данных между одним из 32-х универсальным рабочим регистром и памятью ввода-вывода. Регистры ввода-вывода с адресами \$00-\$1F могут побитно адресоваться с помощью инструкций SBI и CBI. Состояние одного из разрядов в этих регистрах может тестироваться с помощью инструкций SBIS и SBIC. При использовании специфических команд ввода-вывода IN и OUT необходимо использовать адреса \$00-\$3F. Если адресоваться к регистрам ввода-вывода как к памяти данных с помощью инструкций LD и ST, то к указанным выше адресам необходимо прибавить \$20. ATmega128 является сложным микроконтроллером, для которого 64 адреса, зарезервированных в кодах операций IN и OUT, не достаточно для поддержки всех имеющихся периферийных устройств. Для расширенной области ввода-вывода, которая находится по адресам \$60-\$FF в статическом ОЗУ необходимо

использовать только инструкции ST/STS/STD и LD/LDS/LDD. Пространство расширенного ввода-вывода заменяется ячейками статического ОЗУ в режиме совместимости с ATmega103.

Не должна производиться запись в ячейки по зарезервированным адресам в пространстве ввода-вывода.

Некоторые флаги статуса сбрасываются путем записи в них лог. 1. Инструкции CBI и SBI работают только с регистрами по адресам \$00...\$1F.

Регистры управления вводом-выводом и периферийными устройствами описываются в следующих разделах.

**Интерфейс внешней памяти.** Характеристики интерфейса внешней памяти позволяет его использовать не только для подключения к внешнему статическому ОЗУ или флэш-памяти, но и для сопряжения с внешними периферийными устройствами, например, ЖК-дисплеи, АЦП и ЦАП. Его основными отличительными особенностями являются:

- возможность задания четырех различных по длительности состояний ожидания;

- возможность установки различных состояний ожидания для разных секторов внешней памяти (размер сектора конфигурируется);

- возможность выбора количества задействованных разрядов в старшем адресном байте;

- устройство запоминания состояния шины для минимизации потребления тока (опционально).

После разрешения внешней памяти (XMEM) становится доступным адресное пространство за пределами внутреннего статического ОЗУ через предопределенные для этой функции выводы.

**Использование интерфейса внешней памяти.** Элементы интерфейса:

AD7:0: Мультиплексированная младшая шина адреса/шина данных;

A15:8: Старшая шина адреса (с конфигурируемым числом разрядов);

ALE: Строб адреса внешней памяти;

RD: Строб чтения из внешней памяти;

WR: Строб записи во внешнюю память.

Биты управления интерфейсом внешней памяти расположены в трех регистрах: регистр управления микроконтроллером – MCUCR, регистр А управления внешней памятью – XMCRA и регистр В управления внешней памятью – XMCRB.

После разрешения работы интерфейс XMEM изменит настройки регистров направления данных портов, линии которых предопределены для выполнения функций интерфейса XMEM. Более подробная информация об изменении настроек порта приведена в разделе «Порты ввода-вывода» при рассмотрении альтернативных функций. Интерфейс XMEM автоматически определяет, к какой памяти – внешней или внутренней – осуществляется доступ. Во время доступа к внешней памяти интерфейс XMEM будет формировать сигналы шин адреса, данных и управления на линиях порта. При переходе ALE из 1 в 0 на линиях AD7:0 будут присутствовать действительные адресные сигналы. ALE на

ходится на низком уровне во время передачи данных. После разрешения работы интерфейса XMEM доступ к внутренней памяти будет вызывать изменения на шинах данных и адреса, а также строба ALE. Стробы RD и WR останутся неизменными. После запрета работы интерфейса внешней памяти используются обычные установки выводов и направления данных. Рисунок 4.6 иллюстрирует, как подключить внешнее статическое ОЗУ к AVR-микроконтроллеру с помощью 8-разр. регистра (например, «74×573» или аналогичный), который передает данные напрямую при высоком уровне на входе G.

**Требования по фиксации адреса.** Интерфейс внешней памяти разработан с учетом того, что после приложения низкого уровня на вход регистра G время удержания адреса  $t_H$  составит до 5 нс. Задержка установки адреса с входа D на выход Q ( $t_{PD}$ ) должна быть учтена при вычислении требования по времени доступа к внешней памяти. Необходимо также учитывать и влияние печатных проводников, которые представляют собой емкостную нагрузку и влияют на распространение сигналов. Время установки данных перед тем, как G станет равным 0 ( $t_{SU}$ ), не должно превышать разности времен начала формирования действительного адреса и низкого уровня ALE ( $t_{AVLLC}$ ).

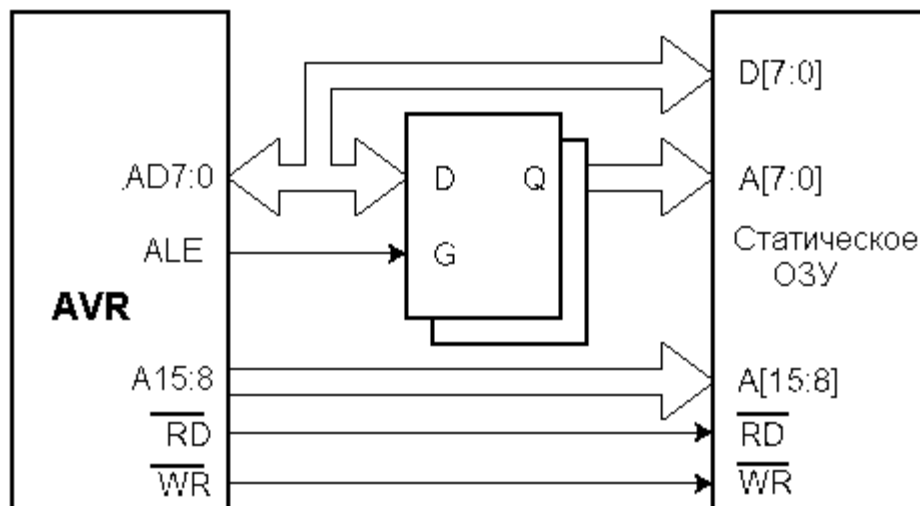


Рис. 4.6. Подключение внешнего статического ОЗУ к AVR-микроконтроллеру

**Подтягивающие резисторы и устройство запоминания состояния шины.** Подтягивающие к плюсу резисторы на линиях AD7:0 могут быть активизированы, если записать единицы в регистр соответствующего порта. Для снижения потребляемой мощности в режиме сна рекомендуется отключать подтягивающие резисторы путем записи нуля в регистр порта непосредственно перед переводом в режим сна.

Интерфейс XMEM также содержит устройство запоминания состояния шины на линиях AD7:0. Устройство запоминания состояния шины может быть программно подключено и отключено. После активизации устройство запоминания состояния шины будет сохранять предыдущее состояние шины AD7:0 при переводе этих линий интерфейсом XMEM в третье состояние.

**Временная диаграмма.** Микросхемы внешней памяти характеризуются различными параметрами временных диаграмм. Для удовлетворения этих тре-

бований интерфейс XMEM у ATmega128 обеспечивает различные состояния ожидания (см. табл. 4.4). Перед выбором состояний ожидания очень важно уточнить требования к временной диаграмме микросхемы внешней памяти. Время доступа к внешней памяти определяется как промежуток времени с момента выбора микросхемы памяти и установки адреса до появления действительных данных, соответствующих указанному адресу на шине. Время доступа не может превышать времени с момента установки импульса ALE к низкому уровню до стабильного установления данных во время чтения. Различные состояния ожидания устанавливаются программно. Реализована дополнительная функция, которая позволяет разделить внешнюю память на два сектора и для каждого из них индивидуально выполнить настройку состояний ожидания. Это делает возможным подключить две различных микросхемы памяти с различными требованиями к временной диаграмме доступа через один и тот же интерфейс XMEM.

Обратите внимание, что интерфейс XMEM – асинхронный и что форма сигналов (рис. 4.7) связана с внутренней синхронизацией. Расхождение между внутренней и внешней синхронизацией (XTAL1) не гарантируется (зависит от температуры и напряжения питания микросхем). Следовательно, интерфейс XMEM не подходит для синхронной работы.

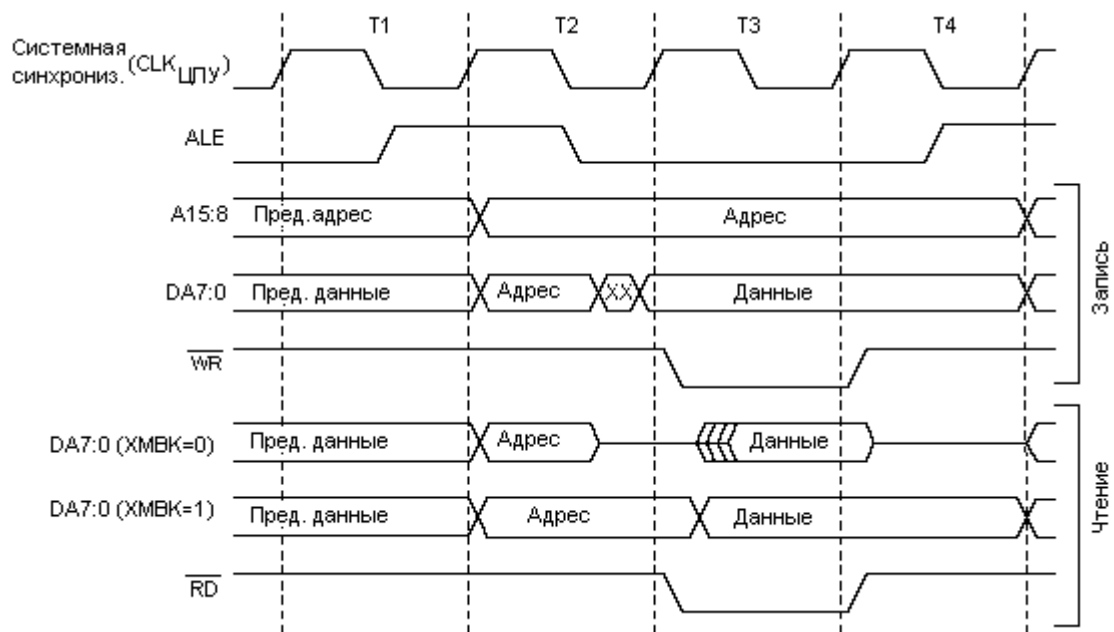


Рис. 4.7. Временная диаграмма доступа к внешней памяти без состояний ожидания (SRWn1=0 и SRWn0=0)

Примечание. SRWn1 = SRW11 (верхний сектор) или SRW01 (нижний сектор), SRWn0 = SRW10 (верхний сектор) или SRW00 (нижний сектор). Импульс ALE присутствует на такте T5, только если следующая инструкция осуществляет доступ к ОЗУ (внутреннему или внешнему).

## Описание интерфейса XMEM

### Регистр управления микроконтроллером – MCUCR

Разряд	7	6	5	4	3	2	1	0	
	<b>SRE</b>	<b>SRW10</b>	<b>SE</b>	<b>SM1</b>	<b>SM0</b>	<b>SM2</b>	<b>IVSEL</b>	<b>IVCE</b>	<b>MCUCR</b>
Чтение/Запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Начальное значение	0	0	0	0	0	0	0	0	

Разряд 7 – SRE: Разрешение внешнего статического ОЗУ/XMEM.

Запись в SRE лог. 1 разрешает работу интерфейса внешней памяти, после чего выходы AD7:0, A15:8, ALE, WR и RD выполняют свои альтернативные функции. После установки бита SRE игнорируются любые установки в соответствующих регистрах направления данных. Запись в SRE нуля отключает интерфейс внешней памяти, после чего вступают в силу обычные функции выводов и установки направления.

Разряд 6 – SRW10: Бит выбора состояния ожидания

При работе микроконтроллера не в режиме совместимости с ATmega103 описание действия данного бита подробно описывается ниже при рассмотрении бит SRWn регистра XMCRA. В режиме совместимости с ATmega103 запись в SRW10 лог. 1 разрешает состояние ожидания и один дополнительный период добавляется к стробу чтения/записи.

### Регистр A управления внешней памятью – XMCRA

Разряд	7	6	5	4	3	2	1	0	
	<b>–</b>	<b>SRL2</b>	<b>SRL1</b>	<b>SRL0</b>	<b>SRW01</b>	<b>SRW00</b>	<b>SRW11</b>	<b>–</b>	<b>XMCRA</b>
Чтение/Запись	Чт	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт/Зп	Чт	
Начальное значение	0	0	0	0	0	0	0	0	

Разряд 7 – Зарезервированный бит.

Данный разряд является зарезервированным и всегда читается как 0. Во время записи в данной позиции необходимо указывать 0 для совместимости с последующими микроконтроллерами.

Разряд 6..4 – SRL2, SRL1, SRL0: Задание границ секторов с состоянием ожидания.

Имеется возможность установить различные состояния ожидания для различных адресов внешней памяти. Адресное пространство внешней памяти может быть разделено на два сектора, каждый из которых имеет собственные биты выбора состояний ожидания. Биты SRL2, SRL1 и SRL0 определяют разбиение секторов (см. табл. 4.3). По умолчанию значение бит SRL2, SRL1 и SRL0 равно нулю и все адресное пространство внешней памяти обслуживается как один сектор. Если все адресное пространство статического ОЗУ конфигурируется как один сектор, то состояния ожидания определяются битами SRW11 и SRW10.

### Таблица 4.3

## Границы секторов памяти при различных настройках SRL2..0

SRL2	SRL1	SRL0	Границы сектора
0	0	0	Нижний сектор = Нет. Верхний сектор = 0x1100 - 0xFFFF
0	0	1	Нижний сектор = 0x1100 - 0x1FFF. Верхний сектор = 0x2000 - 0xFFFF
0	1	0	Нижний сектор = 0x1100 - 0x3FFF. Верхний сектор = 0x4000 - 0xFFFF
0	1	1	Нижний сектор = 0x1100 - 0x5FFF. Верхний сектор = 0x6000 - 0xFFFF
1	0	0	Нижний сектор = 0x1100 - 0x7FFF. Верхний сектор = 0x8000 - 0xFFFF
1	0	1	Нижний сектор = 0x1100 - 0x9FFF. Верхний сектор = 0xA000 - 0xFFFF
1	1	0	Нижний сектор = 0x1100 - 0xBFFF. Верхний сектор = 0xC000 - 0xFFFF
1	1	1	Нижний сектор = 0x1100 - 0xDFFF. Верхний сектор = 0xE000 - 0xFFFF

Разряд 1 и разряд 6 регистра MCUCR – SRW11, SRW10: Биты выбора состояний ожидания для верхнего сектора.

Биты SRW11 и SRW10 задают число состояний ожидания для верхнего сектора внешней памяти.

Разряды 4..2 – SRW01, SRW00: Биты выбора состояний ожидания для нижнего сектора.

Биты SRW01 и SRW00 задают число состояний ожидания для нижнего сектора внешней памяти (см. табл. 4.4).

### Таблица 4.4

## Состояния ожидания

SRWn1	SRWn0	Состояния ожидания
0	0	Нет состояний ожидания
0	1	Задержка на один машинный цикл во время stroba чтения/записи
1	0	Задержка на два машинных цикла во время stroba чтения/записи
1	1	Задержка на два машинных цикла во время stroba чтения/записи и задержка на один машинный цикл перед установкой нового адреса

Прим.:  $n = 0$  или  $1$  для нижнего или верхнего сектора, соответственно.

Разряд 0 – Зарезервированный бит

Данный бит является зарезервированным, поэтому всегда считывается как ноль. Если осуществляется запись в данную ячейку, то для совместимости с последующими микроконтроллерами рекомендуется в позиции данного бита указывать 0.

## Регистр В управления внешней памятью – XMCRB

[illegible]

Разряд 7– ХМВК: Разрешение работы устройства запоминания состояния шины внешней памяти.

Запись в ХМВК лог. 1 разрешает работу устройства запоминания состояния шины на линиях AD7:0. После его активизации AD7:0 будут запоминать последнее установленное состояние, даже если интерфейс ХМЕМ перевел линии в третье состояние. Запись в ХМВК лог. 0 означает запрет работы устройства запоминания состояния шины. ХМВК не подчинен SRE, так что даже если интерфейс ХМЕМ отключен, то устройство запоминания состояния шины будет активным до тех пор пока ХМВК = 1.

Разряды 6..4 – Зарезервированные разряды

Данные разряды являются зарезервированными для будущих микроконтроллеров. Для совместимости с ними рекомендуется записывать в данные позиции лог. 0..

Разряды 2..0 – ХММ2, ХММ1, ХММ0: Маска старших адресных разрядов внешней памяти

После разрешения внешней памяти все выводы порта С по умолчанию используются в качестве старшего адресного байта. Если нет необходимости адресоваться ко всему 60-кбайтному пространству внешней памяти, то свободные адресные линии возможно использовать в качестве универсального ввода-вывода (см. табл. 4.5). Использование бит ХММn позволяет адресоваться ко всем 64 кбайт ячейкам внешней памяти (см. «Использование всех 64 кбайт ячеек внешней памяти»).

Таблица 4.5

Использование старших адресных сигналов в качестве линий универсального ввода-вывода после разрешения внешней памяти

ХММ2	ХММ1	ХММ0	Число разрядов адреса внешней памяти	Освобождаемые адресные линии порта С
0	0	0	8 (Все пространство 60 кбайт)	Нет
0	0	1	7	PC7
0	1	0	6	PC7 - PC6
0	1	1	5	PC7 - PC5
1	0	0	4	PC7 - PC4
1	0	1	3	PC7 - PC3
1	1	0	2	PC7 - PC2
1	1	1	Старший байт адреса не используется	Полностью порт С

**Использование всех ячеек внешней памяти размером менее 64 кбайт.** Поскольку адресное пространство внешней памяти следует за адресным пространством внутренней, то к младшим 4352 ячейкам внешней памяти невозможно адресоваться (адреса 0x0000...0x10FF). Однако при подключении внешней памяти размером менее 64 кбайт, например, 32 кбайт к этим ячейкам можно легко адресоваться по адресам 0x8000...0x90FF. Поскольку адресный бит

внешней памяти A15 не подключен к внешней памяти, то адреса 0x8000...0x90FF будут выступать в качестве адресов 0x0000...0x10FF для внешней памяти. Адресация по адресам свыше 0x90FF не рекомендуется, т. к. может затронуть ячейку внешней памяти, доступ к которой уже осуществлялся по другому (меньшему) адресу. Для прикладной программы 32 кбайта внешней памяти будут представлять линейное адресное пространство с адресами 0x1100...0x90FF.

Когда микроконтроллер находится в режиме совместимости с ATmega103 внутренняя память занимает 4096 байт. Это означает, что 4096 байт внешней памяти могут быть доступны по адресам 0x8000...0x8FFF. Для прикладной программы внешняя память размером 32 кбайт в этом случае будет линейным адресным пространством в диапазоне адресов 0x1000...0x8FFF.

Использование всех 64 кбайт ячеек внешней памяти. Поскольку внешняя память располагается после внутренней памяти, то только 60 кбайт внешней памяти доступно по умолчанию (адресное пространство от 0x0000 до 0x10FF зарезервировано для внутренней памяти). Однако имеется возможность использовать весь объем внешней памяти путем маскирования старших адресных разрядов к нулю. Это может быть выполнено с помощью бит ХММn и программного управления старшими адресными разрядами. Интерфейс памяти будет иметь диапазон адресов 0x0000-0x1FFF, если установить на выходе порта С значение 0x00 и выбрать работу старших адресных разрядов как обычных линий ввода-вывода.

#### **4.1.4. Аналогово-цифровой преобразователь**

Отличительные особенности:

- 10-разрядное разрешение;
- интегральная нелинейность 0.5 мл. разр.;
- абсолютная погрешность  $\pm 2$  мл. разр.;
- время преобразования 65 - 260 мкс;
- частота преобразования до 15 тыс. преобр. в сек. при максимальном разрешении;
- 8 мультиплексированных однополярных входов;
- 7 дифференциальных входных каналов;
- 2 дифференциальных входных канала с опциональным усилением на 10 и 200;
- представление результата с левосторонним или правосторонним выравниванием в 16-разр. слове;
- диапазон входного напряжения АЦП 0...VCC;
- выборочный внутренний ИОН на 2.56 В;
- режимы одиночного преобразования и автоматического перезапуска;
- прерывание по завершении преобразования АЦП;
- механизм подавления шумов в режиме сна.



**Краткое описание.** ATmega128 содержит 10-разр. АЦП последовательного приближения. АЦП связан с 8-канальным аналоговым мультиплексором, 8 однополярных входов которого связаны с линиями порта F. Общий входных сигналов должен иметь потенциал 0В (т. е. связан с GND). АЦП также поддерживает ввод 16 дифференциальных напряжений. Два дифференциальных входа (ADC1, ADC0 и ADC3, ADC2) содержат каскад со ступенчатым программируемым усилением: 0 дБ (1х), 20 дБ (10х), или 46 дБ (200х). Семь дифференциальных аналоговых каналов используют общий инвертирующий вход (ADC1), а все остальные входы АЦП выполняют функцию неинвертирующих входов. Если выбрано усиление 1х или 10х, то можно ожидать 8-разр. разрешение, а если 200х, то 7-разрядное.

АЦП содержит УВХ (устройство выборки-хранения), которое поддерживает на постоянном уровне напряжение на входе АЦП во время преобразования. Функциональная схема АЦП показана на рисунке 4.8.

АЦП имеет отдельный вывод питания AVCC (аналоговое питание). AVCC не должен отличаться более чем на  $\pm 0.3\text{В}$  от VCC.

В качестве внутреннего опорного напряжения может выступать напряжение от внутреннего ИОНа на 2.56В или напряжение AVCC. Если требуется использование внешнего ИОН, то он должен быть подключен к выводу AREF с подключением к этому выводу блокировочного конденсатора для улучшения шумовых характеристик.

**Принцип действия.** АЦП преобразовывает входное аналоговое напряжение в 10-разр. код методом последовательных приближений. Минимальное значение соответствует уровню GND, а максимальное уровню AREF минус 1 мл. разр. К выводу AREF опционально может быть подключено напряжение AVCC или внутренний ИОН на 1.22В путем записи соответствующих значений в биты REFSn в регистр ADMUX. Несмотря на то что ИОН на 2.56В находится внутри микроконтроллера, к его выходу может быть подключен блокировочный конденсатор для снижения чувствительности к шумам, т. к. он связан с выводом AREF.

Канал аналогового ввода и каскад дифференциального усиления выбираются путем записи бит MUX в регистр ADMUX. В качестве однополярного аналогового входа АЦП может быть выбран один из входов ADC0...ADC7, а также GND и выход фиксированного источника опорного напряжения 1,22 В.

В режиме дифференциального ввода предусмотрена возможность выбора инвертирующих и не инвертирующих входов дифференциального усилителя. Результат преобразования зависит от полярности входного сигнала. Для определения полярности достаточно опросить старший бит результата преобразования (ADC9 в ADCH). Если данный бит равен лог. 1, то результат отрицательный, если же лог. 0, то положительный. Результат преобразования отрицательных входных сигналов представляется в дополнительном коде.

Пример: Пусть ADMUX = 0xED (пара входов ADC3-ADC2,  $K_u = 1$ ,  $V_{\text{ион}} = 2.56\text{В}$ , результат с левосторонним выравниванием), напряжение на входе ADC3 = 300 мВ, а на входе ADC2 = 500 мВ, тогда:

$\text{КодАЦП} = 512 * 10 * (300 - 500) / 2560 = -400 = 0x270.$

С учетом выбранного формата размещения результата (левосторонний)  $\text{ADCL} = 0x00$ , а  $\text{ADCH} = 0x9C$ . Если же выбран правосторонний формат ( $\text{ADLAR} = 0$ ), то  $\text{ADCL} = 0x70$ ,  $\text{ADCH} = 0x02$ .

Если выбран дифференциальный режим аналогового ввода, то дифференциальный усилитель будет усиливать разность напряжений между выбранной парой входов на заданный коэффициент усиления. Усиленное таким образом значение поступает на аналоговый вход АЦП. Если выбирается однополярный режим аналогового ввода, то каскад усиления пропускается.

Работа АЦП разрешается путем установки бита  $\text{ADEN}$  в  $\text{ADCSRA}$ . Выбор опорного источника и канала преобразования невозможно выполнить до установки  $\text{ADEN}$ . Если  $\text{ADEN} = 0$ , то АЦП не потребляет ток, поэтому при переводе в экономичные режимы сна рекомендуется предварительно отключить АЦП.

АЦП генерирует 10-разрядный результат, который помещается в пару регистров данных АЦП  $\text{ADCH}$  и  $\text{ADCL}$ . По умолчанию результат преобразования размещается в младших 10-ти разрядах 16-разр. слова (выравнивание справа), но может быть опционально размещен в старших 10-ти разрядах (выравнивание слева) путем установки бита  $\text{ADLAR}$  в регистре  $\text{ADMUX}$ .

Практическая полезность представления результата с выравниванием слева существует, когда достаточно 8-разрядное разрешение, т. к. в этом случае необходимо считать только регистр  $\text{ADCH}$ . В другом же случае необходимо первым считать содержимое регистра  $\text{ADCL}$ , а затем  $\text{ADCH}$ .

После чтения  $\text{ADCL}$  блокируется доступ к регистрам данных со стороны АЦП. Это означает, что если считан  $\text{ADCL}$  и преобразование завершается перед чтением регистра  $\text{ADCH}$ , то ни один из регистров не может модифицироваться и результат преобразования теряется. После чтения  $\text{ADCH}$  доступ к регистрам  $\text{ADCH}$  и  $\text{ADCL}$  со стороны АЦП снова разрешается. АЦП генерирует собственный запрос на прерывание по завершении преобразования. Если между чтением регистров  $\text{ADCH}$  и  $\text{ADCL}$  запрещен доступ к данным для АЦП, то прерывание возникнет, даже если результат преобразования будет потерян.

**Запуск преобразования.** Одиночное преобразование запускается путем записи лог. 1 в бит запуска преобразования АЦП  $\text{ADSC}$ . Данный бит остается в высоком состоянии в процессе преобразования и сбрасывается по завершении преобразования. Если в процессе преобразования переключается канал аналогового ввода, то АЦП автоматически завершит текущее преобразование, прежде чем переключит канал.

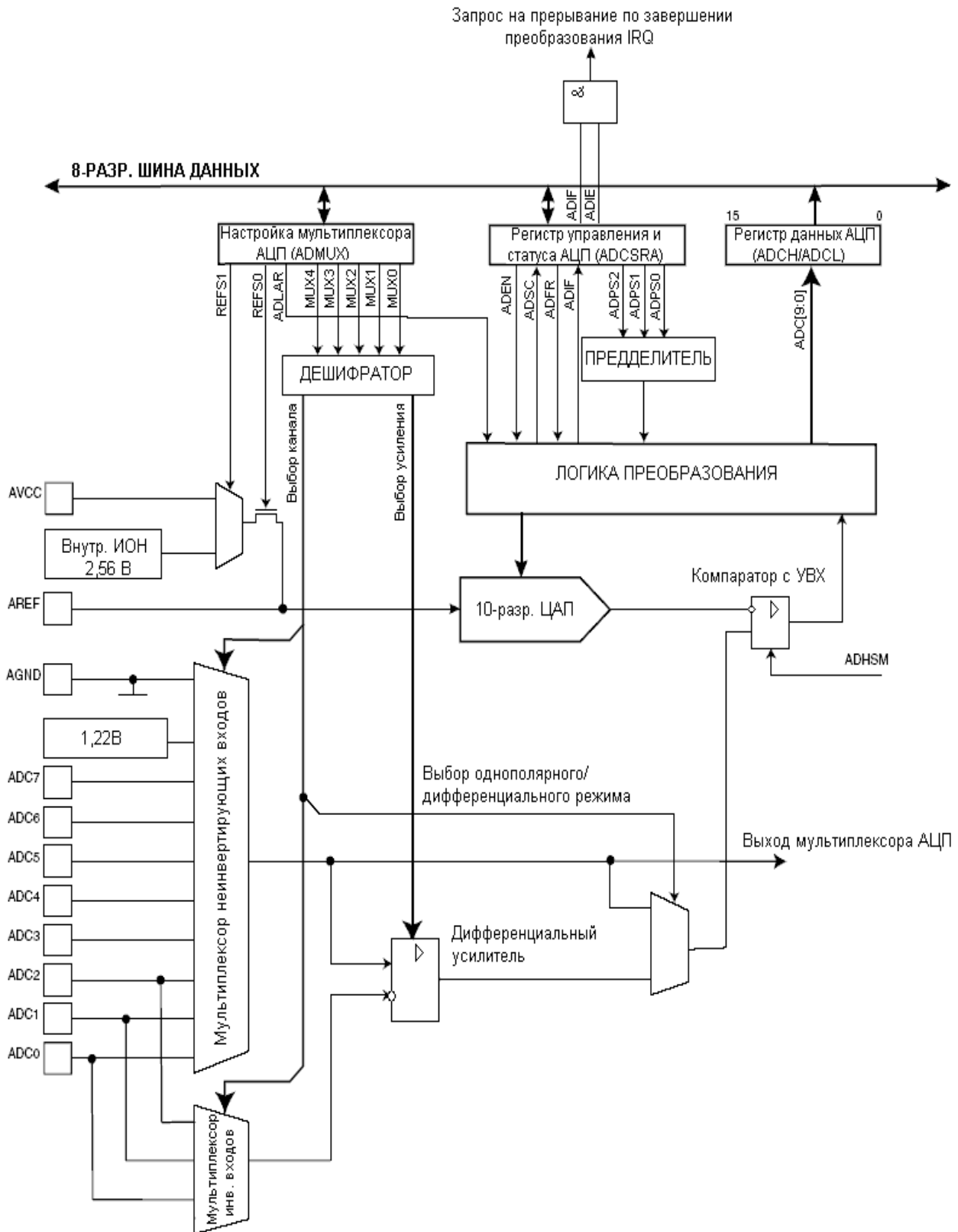


Рис. 4.8. Функциональная схема аналогово-цифрового преобразователя

В режиме автоматического перезапуска АЦП непрерывно оцифровывает аналоговый сигнал и обновляет регистр данных АЦП. Данный режим задается путем записи лог. 1 в бит ADFR регистра ADCSRA. Первое преобразование инициируется путем записи лог. 1 в бит ADSC регистра ADCSRA. В данном

режиме АЦП выполняет последовательные преобразования, независимо от того сбрасывается флаг прерывания АЦП ADIF или нет.

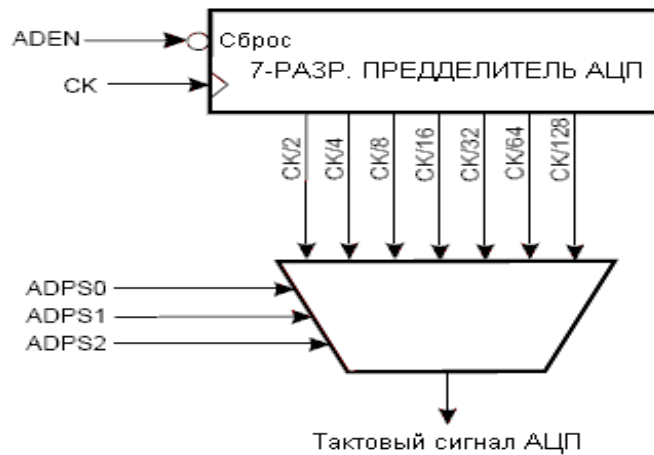


Рис. 4.9. Предделитель АЦП

Если требуется максимальная разрешающая способность (10 разрядов), то частота на входе схемы последовательного приближения должна быть в диапазоне 50...200 кГц. Если достаточно разрешение менее 10 разрядов, но требуется более высокая частота преобразования, то частота на входе АЦП может быть установлена свыше 200 кГц.

Модуль АЦП содержит предделитель (рис. 4.9), который формирует производные частоты свыше 100 кГц по отношению к частоте синхронизации ЦПУ. Коэффициент деления устанавливается с помощью бит ADPS в регистре ADCSRA. Предделитель начинает счет с момента включения АЦП установкой бита ADEN в регистре ADCSRA. Предделитель работает пока бит ADEN = 1 и сброшен, когда ADEN = 0.

Если инициируется однополярное преобразование установкой бита ADSC в регистре ADCSRA, то преобразование начинается со следующего нарастающего фронта тактового сигнала АЦП.

Нормальное преобразование требует 13 тактов синхронизации АЦП. Первое преобразование после включения АЦП (установка ADEN в ADCSRA) требует 25 тактов синхронизации АЦП за счет необходимости инициализации аналоговой схемы.

После начала нормального преобразования на выборку-хранение затрачивается 1,5 такта синхронизации АЦП, а после начала первого преобразования – 13,5 тактов. По завершении преобразования результат помещается в регистры данных АЦП и устанавливается флаг ADIF. В режиме одиночного преобразования одновременно сбрасывается бит ADSC. Программно бит ADSC может быть снова установлен и новое преобразование будет инициировано первым нарастающим фронтом тактового сигнала АЦП.

В режиме автоматического перезапуска новое преобразование начинается сразу по завершении предыдущего, при этом ADSC остается в высоком состоянии. Времена преобразования для различных режимов преобразования представлены в таблице 4.6.

Таблица 4.6

## Время преобразования АЦП

Тип преобразования	Длительность выборки-хранения (в тактах с момента начала преобразования)	Время преобразования (в тактах)
Первое преобразование	14.5	25
Нормальное однополярное преобразование	1.5	13
Нормальное дифференциальное преобразование	1.5/2.5	13/14

**Особенности каналов дифференциального усиления.** Дифференциальные преобразования синхронизированы по отношению к внутренней синхронизации СКАЦП2. Частота преобразования равна половине частоты синхронизации АЦП. Данная синхронизация выполняется автоматически интерфейсом АЦП таким образом, чтобы выборка-хранение инициировалась определенным фронтом СКАЦП2. Если преобразование (все одиночные преобразования и первое преобразование в режиме автоматического перезапуска) инициировалось пользователем, когда СКАЦП2 находился в низком лог. состоянии, то его длительность будет эквивалента однополярному преобразованию (13 тактов синхронизации АЦП). Если преобразование инициируется пользователем, когда СКАЦП2 равен лог. 1, оно будет длиться 14 тактов синхронизации АЦП вследствие работы механизма синхронизации. В режиме автоматического перезапуска новое преобразование инициируется сразу по завершении предыдущего, а т. к. в этот момент СКАЦП2 равен лог. 1, то все преобразования, которые были автоматически перезапущены (т. е. все, кроме первого), будут длиться 14 тактов синхронизации АЦП. Усилительный каскад оптимизирован под частотный диапазон до 4 кГц для любых коэффициентов усиления. Усиление сигналов более высоких частот будет нелинейным. Поэтому если входной сигнал содержит частотные составляющие выше частотного диапазона усилительного каскада, то необходимо установить внешний фильтр низких частот. Заметим, что частота синхронизации АЦП не связана с ограничением по частотному диапазону усилительного каскада. Например, период синхронизации АЦП может быть 6 мкс, при котором частота преобразования канала равна 12 тыс. преобр. в секунду, независимо от частотного диапазона этого канала.

**Изменение канала или выбор опорного источника.** Биты MUXn и REFS1:0 в регистре ADMUX поддерживают одноступенчатую буферизацию через временный регистр. До начала преобразования любые изменения канала и опорного источника вступают в силу сразу после их модификации. Как только начинается процесс преобразования, доступ к изменению канала и опорного источника блокируется, чем гарантируется достаточность времени на преобразование для АЦП. Модификации преобразования вступает в силу на последнем такте АЦП перед завершением преобразования (перед установкой флага ADIF в регистре ADCSRA). Запуск преобразования начинается следующим нарастающим фронтом тактового сигнала АЦП после записи ADSC. Таким образом,

пользователю не рекомендуется записывать новое значение канала или опорного источника в ADMUX до 1-го такта синхронизации АЦП после записи ADSC.

Особые меры необходимо предпринять при изменении дифференциального канала. Как только осуществлен выбор дифференциального канала усилительному каскаду требуется 125 мкс для стабилизации нового значения. Следовательно, в течение первых после переключения дифференциального канала 125 мкс не должно стартовать преобразование. Если же в этот период преобразования все-таки выполнялись, то их результат необходимо игнорировать.

Такую же задержку на установление необходимо ввести при первом дифференциальном преобразовании после изменения опорного источника АЦП (за счет изменения бит REFS1:0 в ADMUX).

Если разрешена работа интерфейса JTAG, то функции каналов АЦП на выводах порта F 7...4 отменяется.

**Входные каналы АЦП.** При переключении входного канала необходимо учесть некоторые рекомендации, которые исключают некорректность переключения.

В режиме одиночного преобразования переключение канала необходимо выполнять перед началом преобразования. Переключение канала может произойти только в течение одного такта синхронизации АЦП после записи лог. 1 в ADSC. Однако самым простым методом является ожидание завершения преобразования перед выбором нового канала.

В режиме автоматического перезапуска канал необходимо выбирать перед началом первого преобразования. Переключение канала происходит аналогично – в течение одного такта синхронизации АЦП после записи лог. 1 в ADSC. Но самым простым методом является ожидание завершения первого преобразования, а затем переключение канала. Поскольку следующее преобразование уже запущено автоматически, то следующий результат будет соответствовать предыдущему каналу. Последующие преобразования отражают результат для нового канала.

При переключении на дифференциальный канал первое преобразование будет характеризоваться плохой точностью из-за переходного процесса в схеме автоматической регулировки смещения. Следовательно, первый результат такого преобразования рекомендуется игнорировать.

**Источник опорного напряжения АЦП.** Источник опорного напряжения (ИОН) для АЦП (ВИОН) определяет диапазон преобразования АЦП. Если уровень однополярного сигнала выше ВИОН, то результатом преобразования будет 0x3FF. В качестве ВИОН могут выступать AVCC, внутренний ИОН 2,56В или внешний ИОН, подключенный к выв. AREF. AVCC подключается к АЦП через пассивный ключ. Внутреннее опорное напряжение 2,56В генерируется внутренним эталонным источником VBG, буферизованного внутренним усилителем. В любом случае внешний вывод AREF связан непосредственно с АЦП, и поэтому можно снизить влияние шумов на опорный источник за счет подключения конденсатора между выводом AREF и общим.

Если пользователь использует внешний опорный источник, подключенный к выв. AREF, то не допускается использование другой опции опорного источника, т. к. это приведет к шунтированию внешнего опорного напряжения. Если к выв. AREF не приложено напряжение, то пользователь может выбрать AVCC и 2.56 в качестве опорного источника. Результат первого преобразования после переключения опорного источника может характеризоваться плохой точностью и пользователю рекомендуется его игнорировать.

Если используются дифференциальные каналы, то выбранный опорный источник должен быть меньше уровня AVCC.

**Подавитель шумов АЦП.** АЦП характеризуется возможностью подавления шумов, которые вызваны работой ядра ЦПУ и периферийных устройств ввода-вывода. Подавитель шумов может быть использован в режиме снижения шумов АЦП и в режиме холостого хода. При использовании данной функции необходимо придерживаться следующей процедуры:

1. Убедитесь, что работа АЦП разрешена и он не выполняет преобразования. Выберите режим одиночного преобразования и разрешите прерывание по завершении преобразования.

2. Введите режим уменьшения шумов АЦП (или режим холостого хода). АЦП запустит преобразование как только остановится ЦПУ.

3. Если до завершения преобразования не возникает других прерываний, то АЦП вызовет прерывание ЦПУ и программа перейдет на вектор обработки прерывания по завершении преобразования АЦП. Если до завершения преобразования другое прерывание пробуждает микроконтроллер, то это прерывание обрабатывается, а по завершении преобразования генерируется соответствующий запрос на прерывание. АЦП остается в активном режиме пока не будет выполнена очередная команда sleep.

Следует обратить внимание, что АЦП не отключается автоматически при переводе во все режимы сна, кроме режима холостого хода и снижения шумов АЦП. Поэтому пользователь должен предусмотреть запись лог. 0 в бит ADEN перед переводом в такие режимы сна во избежание чрезмерного энергопотребления. Если работа АЦП была разрешена в таких режимах сна и пользователь желает выполнить дифференциальное преобразование, то после пробуждения необходимо включить, а затем выключить АЦП для инициации расширенного преобразования, чем будет гарантировано получение действительного результата.

**Схема аналогового входа.** Схема аналогового входа для однополярных каналов представлена на рисунке 4.10. Независимо от того, какой канал подключен к АЦП, аналоговый сигнал, подключенный к выв. ADCn, нагружается емкостью вывода и входным сопротивлением утечки. После подключения канала к АЦП аналоговый сигнал будет связан с конденсатором выборки-хранения через последовательный резистор, сопротивление которого эквивалентно всей входной цепи.

АЦП оптимизирован под аналоговые сигналы с выходным сопротивлением не более 10 кОм. Рекомендуется использовать источники только с малым

выходным сопротивлением и медленно изменяющимися сигналами, т. к. в этом случае будет достаточно быстрым заряд конденсатора выборки-хранения.

По отношению к каналам с дифференциальным усилением рекомендуется использовать сигналы с внутренним сопротивлением до нескольких сотен кОм. Следует предусмотреть, чтобы в предварительных каскадах формирования аналогового сигнала на вход АЦП не вносились частоты выше  $f_{\text{АЦП}}/2$ , в противном случае результат преобразования может быть некорректным. Если вероятность проникновения высоких частот существует, то рекомендуется перед АЦП установить фильтр низких частот.

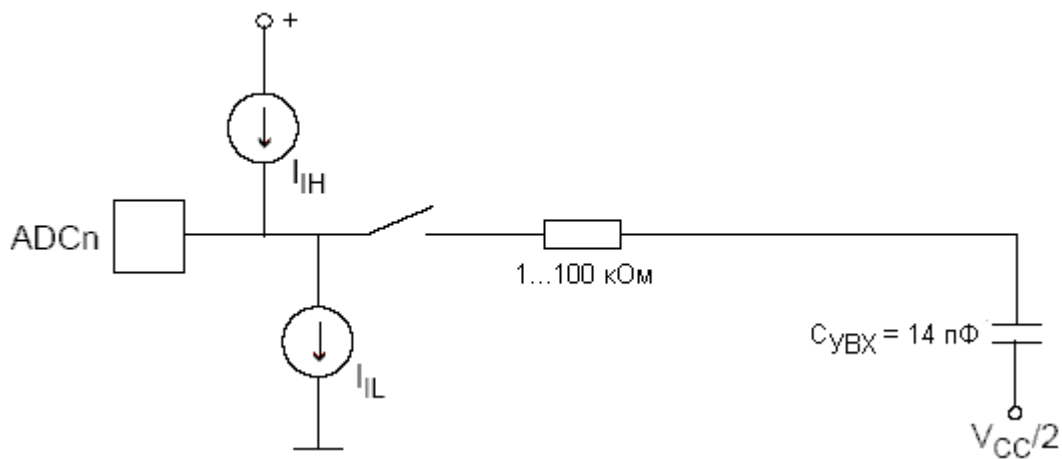


Рис. 4.10. Схема аналогового входа

#### Рекомендации по снижению влияния шумов на результат преобразования.

Если точность преобразования является критическим параметром, то уровень шумов можно снизить, придерживаясь следующих рекомендаций:

1. Выполняйте путь аналоговых сигналов как можно более коротким. Следите, чтобы аналоговые сигналы проходили над плоскостью (слоем) с аналоговой землей (экраном) и далеко от проводников, передающих высокочастотные цифровые сигналы.

2. Вывод AVCC необходимо связать с цифровым питанием VCC через LC-цепь в соответствии.

3. Используйте функцию подавления шумов АЦП, внесенных работой ядра ЦПУ.

4. Если какой-либо из выводов АЦП используется как цифровой выход, то чрезвычайно важно не допустить переключение состояния этого выхода в процессе преобразования.

**Методы компенсации смещения.** Усилительный каскад имеет встроенную схему компенсации смещения, которая стремится максимально приблизить к нулю смещение дифференциального измерения. Оставшееся смещение можно измерить, если в качестве дифференциальных входов АЦП выбрать один и тот же вывод микроконтроллера. Измеренное таким образом остаточное смещение можно программно вычесть из результата преобразования. Использование программного алгоритма коррекции смещения позволяет уменьшить ошибку преобразования до веса одного младшего разряда.



**Регистр управления мультиплексором АЦП– ADMUX**

Разряд	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряд 7:6 – REFS1:0: Биты выбора источника опорного напряжения

Данные биты определяют, какое напряжение будет использоваться в качестве опорного для АЦП (см. табл. 4.7). Если изменить значения данных бит в процессе преобразования, то новые установки вступят в силу только по завершении текущего преобразования (т. е. когда установится бит ADIF в регистре ADCSRA). Внутренний ИОН можно не использовать, если к выводу AREF подключен внешний опорный источник.

Разряд 5 – ADLAR: Бит управления представлением результата преобразования

Бит ADLAR влияет на представление результата преобразования в паре регистров результата преобразования АЦП. Если ADLAR = 1, то результат преобразования будет иметь левосторонний формат, в противном случае – правосторонний. Действие бита ADLAR вступает в силу сразу после изменения независимо от выполняющегося параллельно преобразования. Полное описание действия данного бита представлено в «Регистры данных АЦП – ADCL и ADCH».

Разряд 4:0 – MUX4:0: Биты выбора аналогового канала и коэффициента усиления

Данные биты определяют, какие из имеющихся аналоговых входов подключаются к АЦП. Кроме того, с их помощью можно выбрать коэффициент усиления для дифференциальных каналов (см. табл. 4.8). Если значения бит изменить в процессе преобразования, то механизм их действия вступит в силу только после завершения текущего преобразования (после установки бита ADIF в регистре ADCSRA).

Таблица 4.7

**Выбор опорного источника АЦП**

REFS1	REFS0	Опорный источник
0	0	AREF, внутренний ИОН отключен
0	1	AVCC с внешним конденсатором на выводе AREF
1	0	Зарезервировано
1	1	Внутренний источник опорного напряжения 2.56В с внешним конденсатором на выводе AREF

Таблица 4.8.

## Выбор входного канала и коэффициента усиления

MUX4..0	Однополярный вход	Неинвертирующий дифференциальный вход	Инвертирующий дифференциальный вход	Коэффициент усиления, Ку
00000	ADC0	Нет		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	Нет	ADC0	ADC0	10
01001		ADC1	ADC0	10
01010		ADC0	ADC0	200
01011		ADC1	ADC0	200
01100		ADC2	ADC2	10
01101		ADC3	ADC2	10
01110		ADC2	ADC2	200
01111		ADC3	ADC2	200
10000		ADC0	ADC1	1
10001		ADC1	ADC1	1
10010		ADC2	ADC1	1
10011		ADC3	ADC1	1
10100		ADC4	ADC1	1
10101		ADC5	ADC1	1
10110		ADC6	ADC1	1
10111		ADC7	ADC1	1
11000		ADC0	ADC2	1
11001		ADC1	ADC2	1
11010		ADC2	ADC2	1
11011		ADC3	ADC2	1
11100		ADC4	ADC2	1
11101		ADC5ADC21		
11110	1.23B (VBG)	Нет11110B(GND)		

## Регистр А управления и статуса АЦП – ADCSRA

Разряд	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряд 7 – ADEN: Разрешение работы АЦП

Запись в данный бит лог. 1 разрешает работу АЦП. Если в данный бит записать лог. 0, то АЦП отключается, даже если он находился в процессе преобразования.

#### Разряд 6 – ADSC: Запуск преобразования АЦП

В режиме одиночного преобразования установка данного бита инициирует старт каждого преобразования. В режиме автоматического перезапуска установкой этого бита инициируется только первое преобразование, а все остальные выполняются автоматически. Первое преобразование после разрешения работы АЦП, инициированное битом ADSC, выполняется по расширенному алгоритму и длится 25 тактов синхронизации АЦП вместо обычных 13 тактов. Это связано с необходимостью инициализации АЦП.

В процессе преобразования при опросе бита ADSC возвращается лог. 1, а по завершении преобразования – лог. 0. Запись лог. 0 в данный бит не предусмотрено и не оказывает никакого действия.

#### Разряд 5 – ADFR: Выбор режима автоматического перезапуска АЦП

Если в данный бит записать лог. 1, то АЦП перейдет в режим автоматического перезапуска. В этом режиме АЦП автоматически выполняет преобразования и модифицирует регистры результата преобразования через фиксированные промежутки времени. Запись лог. 0 в этот бит прекращает работу в данном режиме.

#### Разряд 4 – ADIF: Флаг прерывания АЦП

Данный флаг устанавливается после завершения преобразования АЦП и обновления регистров данных. Если установлены биты ADIE и I (регистр SREG), то происходит прерывание по завершении преобразования. Флаг ADIF сбрасывается аппаратно при переходе на соответствующий вектор прерывания. Альтернативно флаг ADIF сбрасывается путем записи лог. 1 в него. При выполнении команды «чтение-модификация-запись» с регистром ADCSRA ожидаемое прерывание может быть отключено. Это также распространяется на использование инструкций SBI и CBI.

#### Разряд 3 – ADIE: Разрешение прерывания АЦП

После записи лог. 1 в этот бит, при условии, что установлен бит I в регистре SREG, разрешается прерывание по завершении преобразования АЦП.

#### Разряды 2:0 – ADPS2:0: Биты управления предделителем АЦП.

Данные биты определяют соотношение между тактовой частотой ЦПУ и частотой входной синхронизации АЦП (см. табл. 4.10).

Таблица 4.10

Управление предделителем АЦП

ADPS2	ADPS1	ADPS0	Коэффициент деления
0	0	0	2
0	0	1	2
0	1	0	4

0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

По завершении преобразования результат помещается в этих двух регистрах. При использовании дифференциального режима преобразования результат представляется в коде двоичного дополнения.

#### Регистры данных АЦП – ADCL и ADCH

ADLAR = 0:

Разряд	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Чтение/запись	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	
	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	
Исх. значение	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1:

Разряд	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	-	-	-	-	-	-	ADCL
	7	6	5	4	3	2	1	0	
Чтение/запись	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	
	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	Чт.	
Исх. значение	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Если выполнено чтение ADCL, то доступ к этим регистрам для АЦП будет заблокирован (т. е. АЦП не сможет в дальнейшем модифицировать результат преобразования), пока не будет считан регистр ADCH.

Левосторонний формат представления результата удобно использовать, если достаточно 8 разрядов. В этом случае 8-разрядный результат хранится в регистре ADCH и, следовательно, чтение регистра ADCL можно не выполнять. При правостороннем формате необходимо сначала считать ADCL, а затем ADCH.

ADC9:0: Результат преобразования АЦП

#### 4.1.5. Порты ввода-вывода

Основная функциональное назначение портов ввода-вывода (ПВВ) – прием и выдача дискретных сигналов управления. Кроме того, через линии портов осуществляется подключение выводов периферийных устройств AVR-микроконтроллеров к внешним устройствам (альтернативные функции). Каж-

дая из линий порта может быть настроена на ввод или вывод независимо от других линий. Изменение направления ввода-вывода одной линии порта командами SBI и CBI не приводит к ложных изменений направления ввода-вывода других линий порта. Данное распространяется также и на изменение логического уровня (если линия порта настроена на вывод) или на включение/отключение подтягивающих резисторов (если линия настроена на ввод). Каждый выходной буфер имеет симметричную характеристику управления с высоким втекающим и вытекающим выходными токами. Выходной драйвер обладает нагрузочной способностью, которая позволяет непосредственно управлять светодиодными индикаторами. Ко всем линиям портов может быть подключен индивидуальный выборочный подтягивающий к плюсу питания резистор, сопротивление которого не зависит от напряжения питания. На всех линиях ПБВ установлены защитные диоды, которые подключены к VCC и Общему (GND), как показано на рисунке 4.11. Подробный перечень параметров ПБВ приведен в разделе «Электрические характеристики».

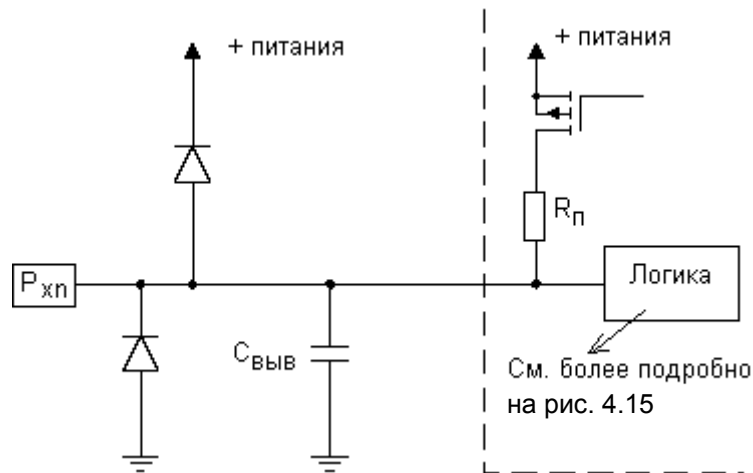


Рис. 4.11. Эквивалентная схема линии ПБВ

Ссылки на регистры и биты регистров в данном разделе даны в общей форме. Например, PORTB3, означающий разряд 3 порта В, в общем виде записывается как PORTxn, где символ «х» заменяет наименование ПБВ, а символ «n» заменяет номер разряда ПБВ. Однако при составлении программы необходимо использовать точную форму записи. Адреса физических регистров ввода-вывода и распределение их разрядов приведены в разделе «Описание регистров портов ввода-вывода».

Для каждого порта ввода-вывода в памяти ввода-вывода зарезервировано три ячейки: одна под регистр данных – PORTx, другая под регистр направления данных – DDRx и третья под состояние входов порта – PINx. Ячейка, хранящая состояние на входах портов, доступна только для чтения, а регистры данных и направления данных имеют двунаправленный доступ. Кроме того, установка бита выключения подтягивающих резисторов PUD регистра SFIOR отключает функцию подтягивания на всех выводах всех портов.

Большинство выводов портов поддерживают альтернативные функции встроенных периферийных устройств микроконтроллера. Описание альтерна-

тивных функций приведено далее в подразделе «Альтернативные функции порта» (см. также описание функций соответствующих периферийных модулей).

Заметим, что для некоторых портов разрешение альтернативных функций некоторых выводов делает невозможным использование других выводов для универсального цифрового ввода-вывода.

**Порты в качестве универсального цифрового ввода-вывода.** Все порты являются двунаправленными портами ввода-вывода с опциональными подтягивающими резисторами. Рисунок 4.15 иллюстрирует функциональную схему одной линии порта ввода-вывода, обозначенный как  $P_{xn}$ .

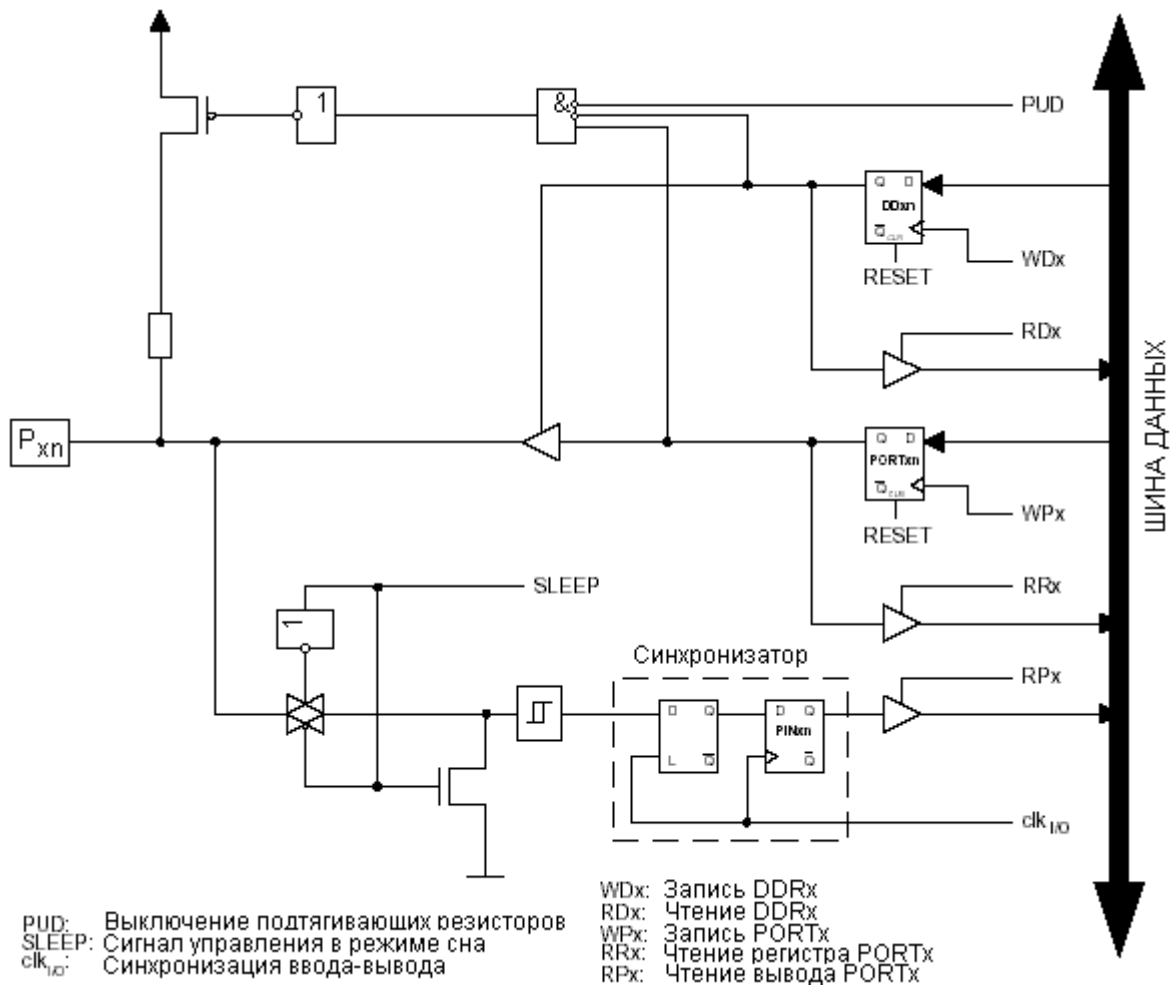


Рисунок 4.12. Организация универсального цифрового ввода-вывода (1)

Прим.: Сигналы  $WPx$ ,  $WDx$ ,  $RRx$ ,  $RPx$  и  $RDx$  являются общими в пределах одного порта.

Сигналы  $clk_{I/O}$ ,  $SLEEP$ , и  $PUD$  являются общими для всех портов

**Настройка выводов.** Режим и состояние для каждого вывода определяется значением соответствующих разрядов трех регистров:  $DDRn$ ,  $PORTn$  и  $PINn$ . Как показано в «Описании регистров портов ввода-вывода» доступ к битам  $DDRn$  возможен по адресу  $DDRn$  в пространстве ввода-вывода и, соответственно, к битам  $PORTn$  по адресу  $PORTn$ , а к битам  $PINn$  по адресу  $PINn$ .

Биты DD<sub>xn</sub> регистра DDR<sub>x</sub> определяют направленность линии ввода-вывода. Если DD<sub>xn</sub> = 1, то R<sub>xn</sub> конфигурируется на вывод. Если DD<sub>xn</sub> = 0, то R<sub>xn</sub> конфигурируется на ввод.

Если PORT<sub>xn</sub> = 1 при конфигурации линии порта на ввод, то разрешается подключение подтягивающего резистора. Для выключения данного резистора необходимо записать в PORT<sub>xn</sub> лог. 0 или настроить линию порта на вывод. Во время сброса все линии портов находятся в третьем (высокоимпедансном) состоянии, даже если не работает синхронизация.

Если PORT<sub>xn</sub> = 1 при конфигурации линии порта на вывод, то состояние выхода будет определяться значением PORT<sub>xn</sub>.

Поскольку одновременная запись в регистры DDR<sub>x</sub> и PORT<sub>x</sub> невозможна, то при переключении между третьим состоянием ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b00) и выводом лог. 1 ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b11) должно возникнуть промежуточное состояние или с подключенным подтягивающим резистором ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b01), или с выводом лог. 0 ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b10). Как правило, переход через состояние с подключением подтягивающего резистора эквивалентно состоянию вывода лог.1, если вывод микроконтроллера связан с высокоимпедансным входом. В противном случае, необходимо установить бит PUD регистра SFIOR для выключения всех подтягивающих резисторов на всех портах.

Переключение между вводом с подтягивающими резисторами и выводом низкого уровня связано с аналогичной проблемой. Поэтому пользователь вынужден использовать или третье состояние ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b00) или вывод лог. 1 ({DD<sub>xn</sub>, PORT<sub>xn</sub>} = 0b11) в качестве промежуточного шага.

В таблице 4.11 рассмотрено действие управляющих сигналов на состояние вывода.

Таблица 4.11

Настройка вывода порта

DD <sub>xn</sub>	PORT <sub>xn</sub>	PUD (в SFIOR)	Ввод-вывод	Подтягивающий резистор	Комментарий
0	0	X	Ввод	Нет	Третье состояние (Z-состояние)
0	1	0	Ввод	Да	R <sub>xn</sub> будет источником тока при подаче внешнего низкого уровня
0	1	1	Ввод	Нет	Третье состояние (Z-состояние)
1	0	X	Вывод	Нет	Вывод лог. 0 (втекающий ток)
1	1	X	Вывод	Нет	Вывод лог. 1 (вытекающий ток)

**Считывание состояние вывода.** Независимо от значения бита направления данных DD<sub>xn</sub> состояние вывода порта может быть опрошено через регистровый бит PIN<sub>xn</sub>. Как показано на рисунке 4.12, регистровый бит PIN<sub>xn</sub> и предшествующая ему триггерная защелка составляют синхронизатор. Данный подход позволяет избежать неоднозначности считывания информации, если изменение состояния на выводе произошло около фронта внутренней синхронизации. На рис. 4.13 представлена временная диаграмма синхронизации во время опроса, внешне приложенного к выводу уровня. Длительности мини-

мальной и максимальной задержек на распространение сигнала обозначены как  $t_{pd, \max}$  и  $t_{pd, \min}$  соответственно.

В следующих примерах показано как установить на линиях 0 и 1 порта В уровень лог. 1, а на линиях 2 и 3 – лог. 0, а также как настроить линии 4...7 на ввод с подключением подтягивающих резисторов на линиях 6 и 7.

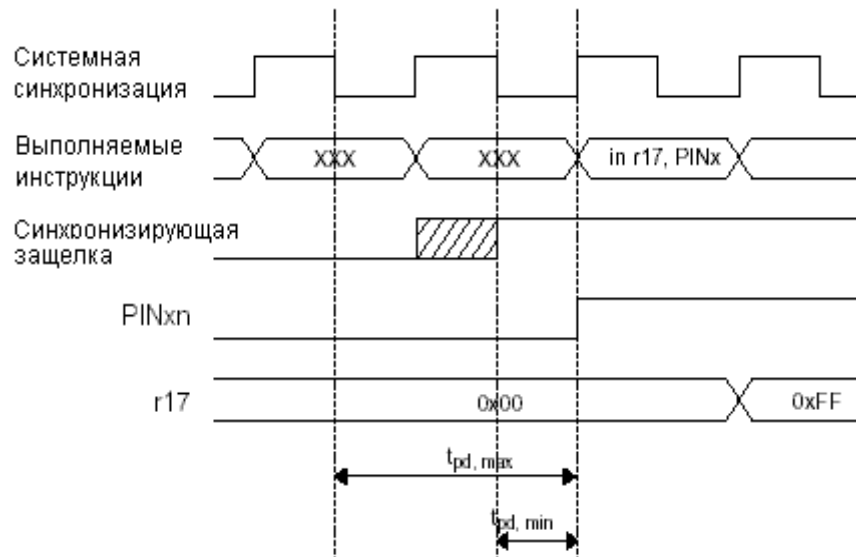


Рис. 4.13. Синхронизация во время опроса приложенного к выводу порта уровня

Результирующее состояние линий считываются обратно, но с учетом сказанного выше, включена инструкция **пор** для обеспечения возможности обратного считывания только что назначенного состояния некоторых выводов

### Пример кода на Ассемблере

```
...
; Разрешаем подтягивание и устанавливаем высокие выходные уровни
; Определяем направления данных линий портов
ldi r16,(1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
ldi r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out PORTB,r16
out DDRB,r17
; Вставляем инструкцию пор для синхронизации
пор
; Опрос состояния выводов порта
in r16,PINB
```

Прим.: В программе на Ассемблере используются два временных регистра для минимизации интервала времени от момента настройки подтягивающих резисторов на разрядах 0, 1, 6 и 7 до момента корректной установки бит направления. С момента установки разрешается вывод лог. 0 на линиях 2 и 3 и



замещается высокий уровень на разрядах 0 и 1, образованный за счет подключения подтягивающих резисторов, на высокий уровень сильноточного драйвера.

**Разрешение цифрового ввода и режимы сна.** Как показано на рисунке 4.12, входной цифровой сигнал может быть зашунтирован на входе триггера Шмита. Сигнал SLEEP устанавливается при переводе микроконтроллера в режим выключения (Power-down), экономичный режим, дежурный режим и расширенный дежурный режим. Это позволяет избежать повышения потребляемого тока в случае, если некоторые входные сигналы окажутся в плавающем состоянии или уровень входного аналогового сигнала будет близок к  $V_{CC}/2$ .

Сигнал SLEEP игнорируется по входам внешних прерываний. Если запросы на внешнее прерывание отключены, то SLEEP действует и на эти выводы. SLEEP также игнорируется на некоторых других входах при выполнении их альтернативных функций (см. «Альтернативные функции порта»).

Если на выводе внешнего асинхронного прерывания, настроенного на любое изменение, присутствует уровень лог. 1 и при этом внешнее прерывание не разрешено, то соответствующий флаг внешнего прерывания будет установлен при выходе из выше упомянутых режимов сна, т. е. функция шунтирования на входе в режимах сна приводит к возникновению логических изменений.

**Неподключенные выводы.** Если несколько выводов остаются неиспользованными, то рекомендуется гарантировать на них присутствие определенного логического уровня. Несмотря на то, что большинство цифровых входов отключены в режимах глубокого сна, как описано выше, необходимо избежать наличия плавающих входов во избежание повышенного потребления тока во всех других режимах работы микроконтроллера, где цифровой ввод разрешен (Сброс, Активный режим и режим холостого).

Самым простым методом гарантирования присутствия определенного уровня на неиспользуемом выводе является разрешение подключения внутреннего подтягивающего резистора. Однако в этом случае в режиме сброса подтягивающие резисторы будут отключены. Если требуется малое потребление и в режиме сброса, то необходимо устанавливать внешний подтягивающий резистор к плюсу или к минусу питания. Подключение выводов непосредственно к  $V_{CC}$  или  $GND$  не рекомендуется, т. к. может возникнуть опасный ток при случайной конфигурации такого вывода на вывод данных.

**Альтернативные функции порта.** Большинство выводов поддерживают альтернативные функции в дополнение к универсальному цифровому вводу-выводу. На рисунке 4.17 показано, как управляющие сигналы, представленные на упрощенном рисунке 4.15, могут быть отключены альтернативными функциями. Сигналы отключения могут присутствовать не на всех выводах, поэтому данный рисунок необходимо использовать как общее описание, применимое ко всем выводам портов семейства AVR-микроконтроллеров.

В таблице 4.15 сведены функции отключающих сигналов для активизации альтернативных функций. Указатели на выводы и порты не показаны в итоговых таблицах. Отключающие сигналы генерируются внутренне в модулях, поддерживающих альтернативные функции.

Таблица 4.15

Общее описание отключающих сигналов для активизации альтернативных функций

Наименование сигнала	Полное наименование	Описание
PUOE	Разрешение альтернативного управления подтягиванием	Если данный сигнал установлен, то подключение подтягивающего резистора определяется значением сигнала PUOV. Если данный сигнал сброшен, то подтягивающий резистор подключается, если {DDxn, PORTxn, PUD} = 0b010.
PUOV	Альтернативное управление подтягиванием	Если PUOE установлен, то подтягивающий резистор подключается/отключается, если PUOV установлен/сброшен независимо от состояния регистровых бит DDxn, PORTxn и PUD.
DDOE	Разрешение задания альтернативного направления	Если этот сигнал установлен, то разрешение работы выходного драйвера определяется значением сигнала DDOV. Если этот сигнал сброшен, то работа выходного драйвера разрешается регистровым битом DDxn.
DDOV	Значение альтернативного направления	Если DDOE установлен, то работа выходного драйвера разрешается/запрещается, когда DDOV устанавливается/сбрасывается независимо от состояния регистрового бита DDxn.
PVOE	Разрешение задания альтернативного выходного состояния порта	Если данный сигнал установлен и разрешена работа выходного драйвера, то состояние на выходе порта определяется сигналом PVOV. Если PVOE сброшен и разрешена работа выходного драйвера, то состояние на выходе порта определяется регистровым битом PORTxn.
PVOV	Альтернативное выходное состояние порта	Если PVOE установлен, то выход порта принимает состояние PVOV независимо от установки регистрового бита PORTxn.
DIEOE	Разрешение альтернативного цифрового ввода	Если данный бит установлен, то функция разрешения цифрового передается сигналу DIEOV. Если данный сигнал сброшен, то разрешение цифрового ввода определяется состоянием микроконтроллера (нормальный режим, режимы сна).
DIEOV	Альтернативное состояние цифрового ввода	Если DIEOE установлен, то цифровой ввод разрешен/запрещен, если DIEOV установлен/сброшен независимо от состояния микроконтроллера (нормальный режим, режимы сна).
DI	Цифровой ввод	Сигнал цифрового ввода для альтернативных функций. На рисунке сигнал подключен к выходу триггера Шмита перед синхронизатором. Если цифровой ввод используется как источник синхронизации, то модуль с альтернативной функцией будет использовать свой собственный синхронизатор.
AIO	Аналоговый ввод-вывод	Сигнал аналогового ввода/вывода к модулю/из модуля с альтернативной функцией. Сигнал подключается непосредственно к контактной площадке и может использоваться двунаправлено.

В следующих подразделах коротко описываются альтернативные функции для каждого порта и связь отключающих сигналов с альтернативными функциями выводов.

Регистр специальных функций ввода-вывода – SFIOR									
Разряд	7	6	5	4	3	2	1	0	
	<b>TSM</b>	–	–	–	<b>ACME</b>	<b>PUD</b>	<b>PSR0</b>	<b>PSR321</b>	<b>SFIOR</b>
Чтение/Запись	Чт./Зп.	Чт.	Чт.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Начальное знач.	0	0	0	0	0	0	0	0	

Разряд 2 – PUD: Отключение всех подтягивающих резисторов

Если в данный разряд записать лог. 1, то подтягивающие резисторы на всех портах будут отключены, даже если регистры DDxn и PORTxn настроены на их подключение ( $\{DDxn, PORTxn\} = 0b01$ ). См. «Настройка выводов» для детального изучения данной функции.

**Альтернативные функции порта A.** Альтернативной функцией порта A является мультиплексированная младшая шина адреса/шина данных внешнего интерфейса памяти.

Таблица 4.16.

Альтернативные функции выводов порта A

Вывод порта	Альтернативная функция
PA7	AD7 (Разряд 7 шины адреса и шины данных внешнего интерфейса памяти)
PA6	AD6 (Разряд 6 шины адреса и шины данных внешнего интерфейса памяти)
PA5	AD5 (Разряд 5 шины адреса и шины данных внешнего интерфейса памяти)
PA4	AD4 (Разряд 4 шины адреса и шины данных внешнего интерфейса памяти)
PA3	AD3 (Разряд 3 шины адреса и шины данных внешнего интерфейса памяти)
PA2	AD2 (Разряд 2 шины адреса и шины данных внешнего интерфейса памяти)
PA1	AD1 (Разряд 1 шины адреса и шины данных внешнего интерфейса памяти)
PA0	AD0 (Разряд 0 шины адреса и шины данных внешнего интерфейса памяти)

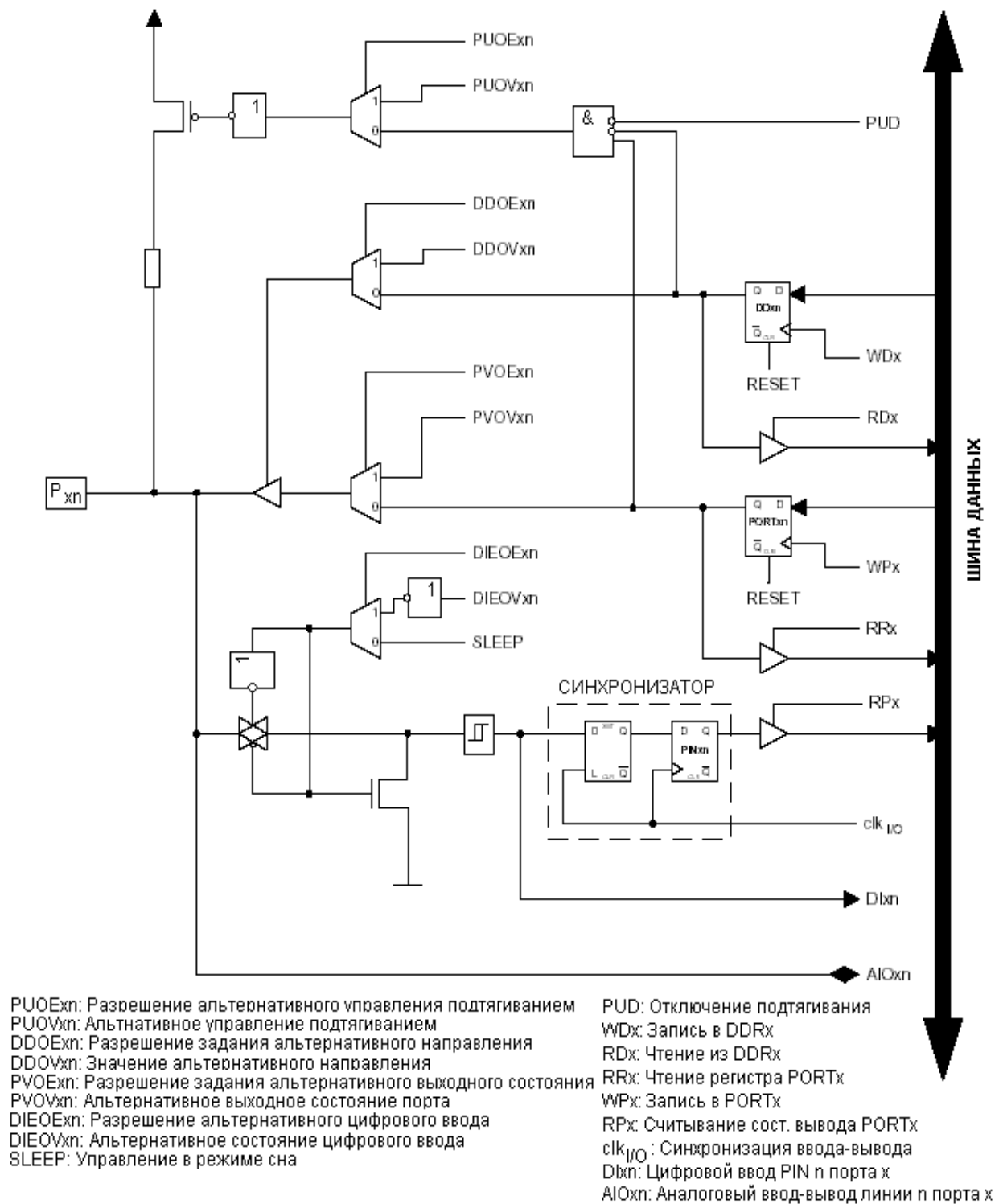


Рис. 4.14. Альтернативные функции порта

Прим.: Сигналы WPx, WDx, RLx, RPx и RDx являются общими в пределах одного порта. Сигналы clkI/O, SLEEP, и PUD являются общими для всех портов. Все остальные сигналы индивидуальны для каждого вывода.

В таблицах 4.14 и 4.15 приведена связь отключающих сигналов, представленных на рис. 4.14, и альтернативных функций выводов порта A.

Таблица 4.17

Отключающие сигналы для разрешения альтернативных функций на PA7..PA4

Наименование сигнала	PA7/AD7	PA6/AD6	PA5/AD5	PA4/AD4
PUOE	SRE	SRE	SRE	SRE
PUOV	$\sim(\text{WR} \mid \text{ADA} (1) \cdot \text{PORTA7} \cdot \text{PUD})$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA6} \cdot \text{PUD}$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA5} \cdot \text{PUD}$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA4} \cdot \text{PUD}$
DDOE	SRE	SRE	SRE	SRE
DDOV	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$
PVOE	SRE	SRE	SRE	SRE
PVOV	$\text{A7} \cdot \text{ADA} \mid \text{D7 OUTPUT} \cdot \text{WR}$	$\text{A6} \cdot \text{ADA} \mid \text{D6 OUTPUT} \cdot \text{WR}$	$\text{A5} \cdot \text{ADA} \mid \text{D5 OUTPUT} \cdot \text{WR}$	$\text{A4} \cdot \text{ADA} \mid \text{D4 OUTPUT} \cdot \text{WR}$
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	D7 INPUT	D6 INPUT	D5 INPUT	D4 INPUT
AIO	-	-	-	-

Прим.: ADA существует короткий интервал времени, когда выводятся адресные сигналы (см. также «Внешний интерфейс памяти»).

Таблица 4.18

Отключающие сигналы для разрешения альтернативных функций на PA3..PA0

Наименование сигнала	PA3/AD3	PA2/AD2	PA1/AD1	PA0/AD0
PUOE	SRE	SRE	SRE	SRE
PUOV	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA3} \cdot \text{PUD}$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA2} \cdot \text{PUD}$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA1} \cdot \text{PUD}$	$\sim(\text{WR} \mid \text{ADA}) \cdot \text{PORTA0} \cdot \text{PUD}$
DDOE	SRE	SRE	SRE	SRE
DDOV	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$	$\text{WR} \mid \text{ADA}$
PVOE	SRE	SRE	SRE	SRE
PVOV	$\text{A3} \cdot \text{ADA} \mid \text{D3 OUTPUT} \cdot \text{WR}$	$\text{A2} \cdot \text{ADA} \mid \text{D2 OUTPUT} \cdot \text{WR}$	$\text{A1} \cdot \text{ADA} \mid \text{D1 OUTPUT} \cdot \text{WR}$	$\text{A0} \cdot \text{ADA} \mid \text{D0 OUTPUT} \cdot \text{WR}$
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	D3 INPUT	D2 INPUT	D1 INPUT	D0 INPUT
AIO	-	-	-	-

**Альтернативные функции порта В.** Выводы порта В с альтернативными функциями показаны в таблице 4.19.

Таблица 4.19

## Альтернативные функции порта В

Вывод порта	Альтернативная функция
PB7	OC2/OC1C(1) (выход компаратора и выход ШИМ таймера-счетчика 2 или выход С компаратора и ШИМ таймера-счетчика 1)
PB6	OC1B (выход В компаратора и ШИМ таймера-счетчика 1)
PB5	OC1A (выход А компаратора и ШИМ таймера-счетчика 1)
PB4	OC0 (Выход компаратора и ШИМ таймера-счетчика 0)
PB3	MISO (Ввод для ведущей/вывод для подчиненной шины SPI)
PB2	MOSI (Вывод для ведущей/ввод для подчиненной шины SPI)
PB1	SCK (Синхронизация последовательной связи шины SPI)
PB0	SS (вход выбора подчиненного режима интерфейса SPI)

**Альтернативной функцией порта С** является старшая шина адреса внешнего интерфейса памяти.

Таблица 4.20

## Альтернативные функции выводов порта С

Вывод порта	Альтернативная функция
PC7	A15 (Разряд 15 шины адреса внешнего интерфейса памяти)
PC6	A14 (Разряд 14 шины адреса внешнего интерфейса памяти)
PC5	A13 (Разряд 13 шины адреса внешнего интерфейса памяти)
PC4	A12 (Разряд 12 шины адреса внешнего интерфейса памяти)
PC3	A11 (Разряд 11 шины адреса внешнего интерфейса памяти)
PC2	A10 (Разряд 10 шины адреса внешнего интерфейса памяти)
PC1	A9 (Разряд 9 шины адреса внешнего интерфейса памяти)
PC0	A8 (Разряд 8 шины адреса внешнего интерфейса памяти)

**Альтернативные функции порта D.** Выводы порта D с альтернативными функциями представлены в таблице 4.21.

Таблица 4.21

## Альтернативные функции выводов порта D

Вывод порта	Альтернативная функция
PD7	T2 (вход синхронизации таймера-счетчика 2)
PD6	T1 (вход синхронизации таймера-счетчика 1)
PD5	XCK1(1) (вход/выход внешней синхронизации УСАПП1)
PD4	IC1 (вход триггера захвата фронта таймера-счетчика 1)
PD3	INT3/TXD1(1) (вход внешнего прерывания 3 или выход передачи УАПП1)
PD2	INT2/RXD1(1) (вход внешнего прерывания 2 или вход приема УАПП1)
PD1	INT1/SDA(1) (вход внешнего прерывания 1 или ввод/вывод последовательных данных TWI)
PD0	INT0/SCL(1) (вход внешнего прерывания 0 или синхронизация последовательной связи TWI)

Прим.: XCK1, TXD1, RXD1, SDA и SCL отсутствуют в режиме совместимости с ATmega103.

**Альтернативные функции порта Е.** Альтернативные функции порта Е представлены в таблице 4.22.

Таблица 4.22

Альтернативные функции выводов порта Е

Вывод порта	Альтернативная функция
PE7	INT7/IC3(1) (вход внешнего прерывания 7 или вход триггера захвата фронта таймера-счетчика 3)
PE6	INT6/ T3(1) (вход внешнего прерывания 6 или вход синхронизации таймера-счетчика 3)
PE5	INT5/OC3C(1) (вход внешнего прерывания 5 или выход С компаратора и ШИМ таймера-счетчика 3)
PE4	INT4/OC3B(1) (вход внешнего прерывания 4 или выход В компаратора и ШИМ таймера-счетчика 3)
PE3	AIN1/OC3A (1) (инвертирующий вход аналогового компаратора или выход А компаратора и ШИМ таймера-счетчика 3)
PE2	AIN0/XCK0(1) (неинвертирующий вход аналогового компаратора или вход/выход внешней синхронизации УСАПП0)
PE1	PDO/TXD0 (вывод программируемых данных или вывод передачи УАПП0)
PE0	PDI/RXD0 (ввод программируемых данных или вывод приема УАПП0)

Прим.: IC3, T3, OC3C, OC3B, OC3V, OC3A и XCK0 отсутствуют в режиме совместимости с ATmega103.

**Альтернативные функции порта F.** Альтернативной функцией порта F является аналоговые входы АЦП (см. табл. 4.23).

Если некоторые выводы порта F используются как выходы, то необходимо следить, чтобы во время преобразования АЦП не происходило их переключение. Иначе, результат преобразования может быть некорректным. В режиме совместимости с ATmega103 порт F работает только на ввод. Если разрешена работа интерфейса JTAG, то подтягивающие резисторы на выводах PF7(TDI), PF5(TMS) и PF4(TCK) остаются подключенными, даже если микроконтроллер переведен в состояние сброса.

Таблица 4.23

Альтернативные функции выводов порта F

Вывод порта	Альтернативная функция
PF7	ADC7/TDI (Вход канала 7 АЦП или ввод данных при JTAG тестировании)
PF6	ADC6/TDO (Вход канала 6 АЦП или вывод данных при JTAG тестировании)
PF5	ADC5/TMS (Вход канала 5 АЦП или выбор режима JTAG тестирования)
PF4	ADC4/TCK (Вход канала 4 АЦП или синхронизация JTAG тестирования)
PF3	ADC3 (Вход канала 3 АЦП)
PF2	ADC2 (Вход канала 2 АЦП)
PF1	ADC1 (Вход канала 1 АЦП)
PF0	ADC0 (Вход канала 0 АЦП)

**Альтернативные функции порта G.** В режиме совместимости с ATmega103 могут использоваться только альтернативные функции порта G, а

функция универсального цифрового ввода-вывода не доступна. В таблице 4.24 приведены альтернативные функции порта G.

Таблица 4.24

Альтернативные функции выводов порта G

Вывод порта	Альтернативная функция
PG4	TOSC1 (Генератор часов реального времени таймера-счетчика 0)
PG3	TOSC2 (Генератор часов реального времени таймера-счетчика 0)
PG2	(Разрешение фиксации адреса внешней памяти)
PG1	RD (Строб чтения внешней памяти)
PG0	WR (Строб записи внешней памяти)

#### 4.1.6. Таймеры-счетчики 1 и 3.

Два 16-разрядных таймера-счетчика предназначены для точного задания временных интервалов, генерации прямоугольных импульсов и измерения временных характеристик импульсных сигналов.

Основные отличительные особенности:

16-разрядные счетчики (в т. ч. возможность организации 16-разр. ШИМ);

три отдельных блока сравнения;

двойная буферизация регистров порога сравнения (OCR);

один блок захвата;

подавитель шума на входе блока захвата;

режим сброса таймера при совпадении с порогом сравнения (автоматическая перезагрузка);

широотно-импульсная модуляция без генерации ложных импульсов при записи нового порога сравнения в OCR (двойная буферизация) и фазовая коррекция;

переменный период ШИМ;

частотный генератор;

счетчик внешних событий;

10 самостоятельных источников прерываний (TOV1, OCF1A, OCF1B, OCF1C, ICF1, TOV3, OCF3A, OCF3B, OCF3C и ICF3).

По сравнению с предыдущими версиями 16-разр. таймеров-счетчиков данные таймеры доработаны и улучшены. Совместимость этих таймеров соблюдается по следующим позициям:

адреса всех регистров, связанных с 16-разр. таймером-счетчиком, в т. ч.

регистры прерываний таймеров;

расположение бит внутри всех регистров 16-разр. таймеров, в т. ч. регистры прерываний таймеров;

векторы прерываний.

У следующих управляющих бит изменены наименования, но сохранено назначение и расположение в регистре:

PWMn0 заменен на WGMn0;



PWMn1 заменен на WGMn1;

CTCn заменен на WGMn2.

Ниже приведены регистры, которые были добавлены к 16-разр. таймеру-счетчику:

регистр управления таймером-счетчиком C (TCCRnC);

регистр C порога сравнения, OCRnCH и OCRnCL (или OCRnC).

Следующие биты добавлены в регистры управления 16-разр. таймером-счетчиком:

COM1C1, COM1C0 добавлены в TCCR1A;

FOCnA, FOCnB и FOCnC добавлены в новый регистр TCCRnC;

WGMn3 добавлен в TCCRnB.

Добавлены флаг прерываний и биты маски прерываний для канала сравнения C.

**Описание.** Укрупненная функциональная схема 16-разр. таймера-счетчика показана на рисунке 4.15. Описание регистров, расположение и назначение бит данных таймеров представлены в параграфе «Описание регистров 16-разр. таймеров-счетчиков».

Регистр таймера-счетчика (TCNTn, где n – номер таймера), регистры порогов сравнения (OCRnA/B/C, где A,B,C-канал), а также регистр захвата (ICRn) являются 16-разрядными регистрами. Доступ к этим регистрам осуществляется через временной регистр и должна быть соблюдена специальная процедура чтения и записи их содержимого. Регистры управления таймером (TCCRnA/B/C) являются 8-разр. регистрами, поэтому доступ к ним со стороны ЦПУ не связан с какими-либо ограничениями. Все сигналы запросов на прерывание представлены в регистре флагов прерываний таймеров (TIFR) и регистре флагов расширенных прерываний (ETIFR). Все прерывания индивидуально маскируются регистром макси прерываний таймеров (TIMSK) и регистром маски расширенных прерываний (ETIMSK). Регистры (E)TIFR и (E)TIMSK не представлены на функциональной схеме, т. к. они совместно используются другими таймерами микроконтроллера.

Таймер-счетчик может тактироваться внутренне через предделитель или внешне тактовым источником, подключенным к выводу Tn. Блок выбора тактового источника позволяет выбрать тактовый источник и фронт, по которому будет изменяться состояние таймера-счетчика. Если тактовый источник не задан, то таймер-счетчик находится в неактивном состоянии. Сигнал на выходе блока выбора тактового источника является тактовым сигналом таймера (clkTn).

Значение регистров порогов сравнения (OCRnA/B/C) непрерывно сравнивается со значением счетчика. Результат сравнения может использоваться для формирования сигналов с ШИМ или генерации прямоугольных импульсов с переменной частотой на выходах OCnA/B/C.

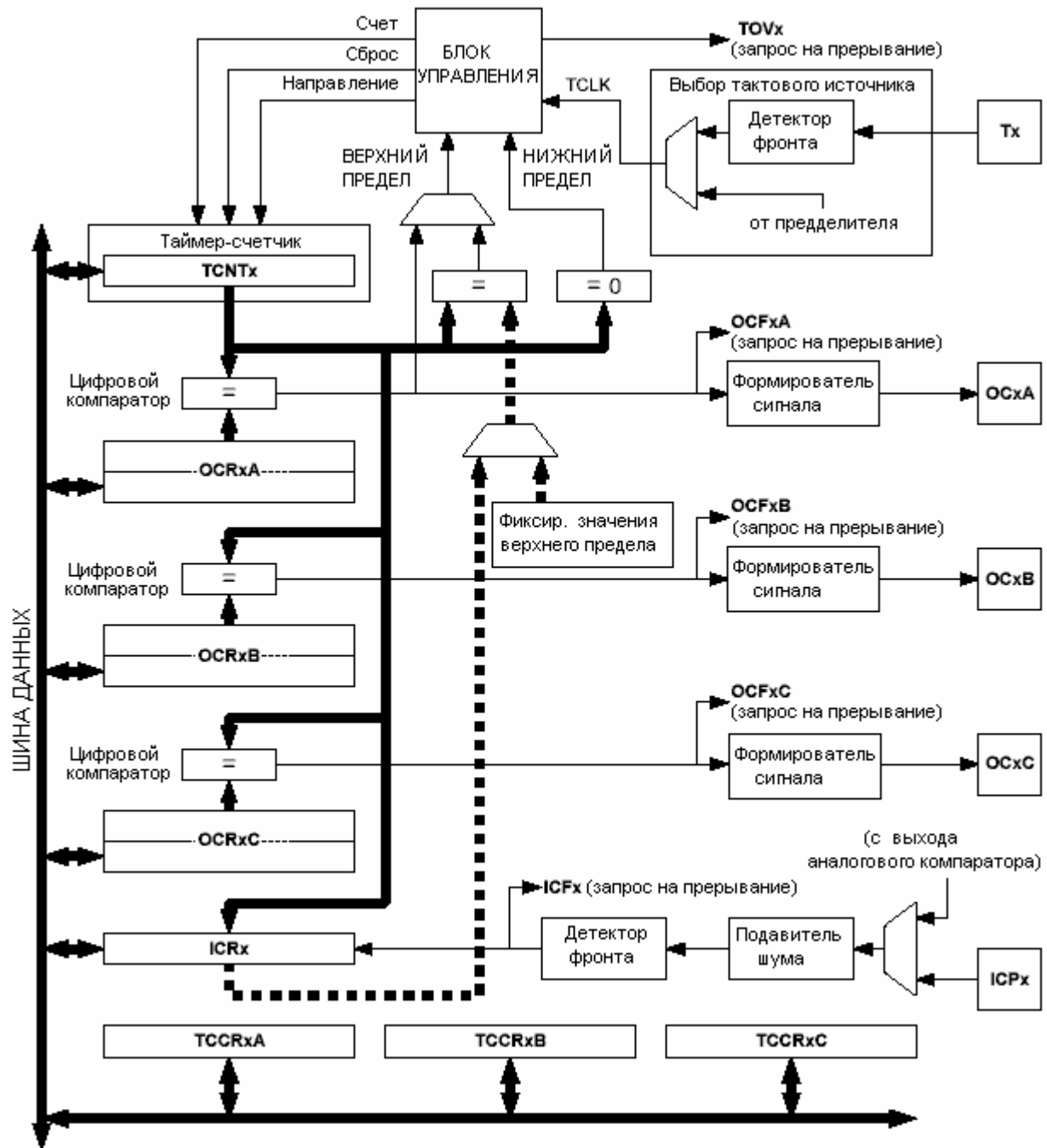


Рис. 4.15. Функциональная схема 16-разр. таймера-счетчика

В случае совпадения значений сравниваемых регистров устанавливается соответствующий флаг прерываний ( $OCFnA/B/C$ ), который в свою очередь может служить источником прерывания.

Регистр захвата позволяет запомнить состояние таймера-счетчика при возникновении заданного внешнего события (фронт внешнего сигнала) на входе захвата фронта  $ICPn$  или на выводах аналогового компаратора. На входе захвата фронта предусмотрена схема цифровой фильтрации (подавитель шума) для снижения риска срабатывания схемы захвата от помехи. Верхний предел или максимальное значение таймера-счетчика в зависимости от режима работы

таймера могут определяться значением в OCRnA, ICRn или иметь фиксированные значения. Если OCRnA задает верхний предел счета в режиме ШИМ, то он не может использоваться для генерации ШИМ-сигналов. Однако верхний предел в этом случае имеет двойную буферизацию, тем самым допуская изменение его значения в произвольный момент времени. Если верхний предел счета является постоянным значением, то альтернативно можно использовать регистр ICRn, освобождая регистр OCRnA для функции широтно-импульсной модуляции.

**Доступ к 16-разр. регистрам.** Регистры TCNTn, OCRnA/B/C и ICRn являются 16-разрядными, поэтому доступ к ним через 8-разр. шину данных AVR ЦПУ может быть осуществлен с помощью двух инструкций чтения или записи. У каждого 16-разр. таймера имеется свой 8-разр. регистр для временного хранения старшего байта данных. Во время доступа к 16-разр. регистрам одного таймера используется один и тот же временный регистр. Чтение/запись младшего байта инициирует 16-разр. операцию чтения/записи. Если выполняется запись младшего байта 16-разр. регистра, то за один такт ЦПУ одновременно записываются и младший байт, и старший байт из временного регистра. Если выполняется чтение младшего байта 16-разр. регистра, то за один такт ЦПУ параллельно с чтением младшего байта происходит копирование старшего байта 16-регистра во временный регистр.

Не все 16-разрядные регистры используют временный регистр для копирования старшего байта. Чтение 16-разр. регистров OCRnA/B/C не связано с использованием временного регистра.

Таким образом, чтобы записать данные в 16-разр. регистр, необходимо сначала записать старший байт, а затем младший. А при чтении 16-разр. регистра, наоборот, сначала считывается младший байт, а затем старший.

Ниже приведен пример на Ассемблере, показывающий, как осуществлять доступ к 16-разр. регистрам таймера. В примере предполагается, что во время обновления временного регистра не возникает прерываний. Аналогично может быть выполнен доступ к регистрам OCRnA/B/C и ICRn.

```
...
; Установка TCNTn = 0x01FF
ldi r17,0x01
ldi r16,0xFF
out TCNTnH,r17
out TCNTnL,r16
; Чтение TCNTn в r17:r16
in r16,TCNTnL
in r17,TCNTnH
...
```

В примере на Ассемблере значение TCNTn возвращается парой регистров r17:r16. При этом следует обратить внимание на проблему, которая связана с необходимостью выполнения двух инструкций для получения доступа к 16-разр. регистру. Если после выполнения первой инструкции доступа 16-разр. регистра происходит прерывание и в процедуре обработки прерывания также

происходит обновление этого же или другого регистра, но относящегося к тому же таймеру, то по завершении обработки прерывания изменится содержимое временного регистра и выполнение второй инструкции приведет к некорректному результату. Таким образом, когда и в основной программе и в прерываниях происходит обновление временного регистра, то в основной программе перед инициацией доступа к 16-разр. регистру необходимо запретить прерывания.

В следующем примере дан вариант чтения регистра TCNTn, без опасности изменения содержимого временного регистра, при возникновении прерывания. Доступ к регистрам OCRnA/B/C и ICRn можно осуществлять аналогично.

```
TIM16_ReadTCNTn:
; Запомнили состояние общего флага прерываний
in r18,SREG
; Запрет прерываний
cli
; Чтение TCNTn в r17:r16
in r16,TCNTnL
in r17,TCNTnH
; Восстановили состояние общего флага прерываний out SREG,r18
ret
```

В коде на Ассемблере значение регистра TCNTn перезаписывается в пару регистров r17, r16. В следующем примере показано, как избежать опасного влияния изменения содержимого временного регистра при возникновении прерывания во время записи в регистр TCNTn. На этом же принципе может быть выполнена запись в регистры OCRnA/B/C или ICRn.

```
TIM16_WriteTCNTn:
; Запомнили состояние общего флага прерываний
in r18,SREG
; Запрет прерываний
cli
; Копирование TCNTn в r17:r16
out TCNTnH,r17
out TCNTnL,r16
; Восстановили состояние общего флага прерываний
out SREG,r18
ret
```

**Тактовые источники таймера-счетчика.** Таймер-счетчик может использовать как внешний, так и внутренний тактовые сигналы. Источник тактового сигнала выбирается соответствующей схемой микроконтроллера под управлением бит выбора синхронизации (CSn2:0), которые находятся в регистре В управления таймером-счетчиком (TCCRnB). Более подробная информация по тактовым источникам и предделителю приведена в разделе «Предделители таймера-счетчика 3, таймера-счетчика 2 и таймера-счетчика 1».

**Блок счетчика.** Основным элементом 16-разр. таймера-счетчика является программируемый реверсивный 16-разрядный счетчик. На рисунке 4.16 представлена функциональная схема счетчика и сопряженных с ним элементов.

Описание внутренних сигналов:

Счет – Инкрементирует или декрементирует состояние TCNTn на 1.

Направление – Задаёт прямой счет (инкрементирование) или обратный счет (декрементирование).

Сброс – Сброс TCNTn (установка всех разрядов к лог. 0).

clkTn – Синхронизация таймера-счетчика.

Верхний предел – Сигнализирует о достижении TCNTn максимального значения.

Нижний предел – Сигнализирует о достижении TCNTn минимального значения (нуля).

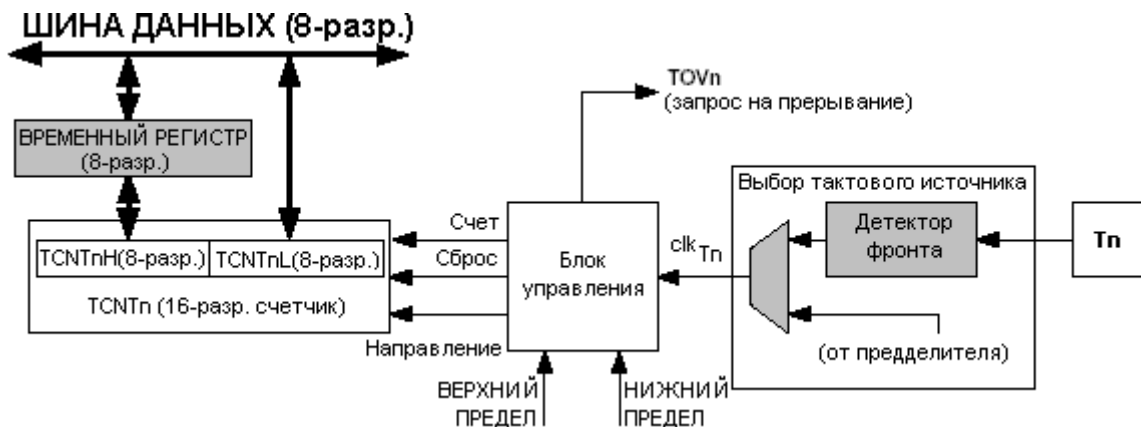


Рис. 4.16. Функциональная схема счетчика

Содержимое 16-разр. счетчика разбито на две 8-разр. ячейки, расположенных в памяти ввода-вывода: Старший байт счетчика (TCNTnH), в котором хранятся старшие 8-разрядов счетчика, и младший байт счетчика (TCNTnL), в котором хранятся младшие 8-разрядов. ЦПУ не имеет непосредственного доступа к регистру TCNTnH. Если ЦПУ выполняет доступ к TCNTnH, то фактически обращение происходит к временному регистру. Во временный регистр копируется значение TCNTnH, если выполняется чтение регистра TCNTnL и в TCNTnH копируется содержимое временного регистра, если выполняется запись в TCNTnL. Такой механизм реализован для считывания/записи 16-разр. значения счетчика за один такт ЦПУ в условиях 8-разр. шины данных.

В зависимости от используемого режима работы каждый такт синхронизации таймера clkTn счетчик будет сбрасываться, инкрементироваться или декрементироваться. Сигнал clkTn может быть внешним или внутренним, что задается битами выбора синхронизации (CSn2:0). Если тактовый источник не задан (CSn2:0 = 0), то таймер останавливается. Однако содержимое TCNTn остается доступным ЦПУ независимо от наличия синхронизации на clkTn. Если ЦПУ выполняет запись в TCNTn, то тем самым блокируется (запись имеет более высокий приоритет) любое действие счетчика: сброс или счет.

Алгоритм счета определяется значением бит режима работы таймера (WGMn3:0), расположенных в регистрах A и B управления таймером-счетчиком (TCCRnA и TCCRnB). Имеется четкая связь между алгоритмом счета счетчика и формой генерируемого на выходе OSpn сигнала. Более подробная

информация об этом приведена в «Режимы работы 16-разр. таймеров-счетчиков».

Установка флага переполнения таймера-счетчика (TOVn) происходит в зависимости выбранного с помощью бит WGMn3:0 режима работы. Флаг TOVn может использоваться для генерации прерывания ЦПУ.

**Блок захвата.** Таймер-счетчик содержит блок захвата, который запоминает состояние счетчика при возникновении внешнего события, тем самым определяя время его возникновения. В качестве события/событий выступает внешний сигнал, подключенный к выводу ICPn. Для таймера-счетчика 1 альтернативно может использоваться аналоговый компаратор в качестве источника внешнего события. Результат захвата состояния таймера может использоваться для вычисления частоты, скважности импульсов и других параметров импульсных сигналов. Альтернативно это значение может использоваться для создания журнала событий.

Функциональная схема блока захвата представлена на рисунке 4.17. Некоторые блоки на функциональной схеме, которые не относятся напрямую к блоку захвата, залиты серым цветом. Символ «n» в наименованиях бит и регистров заменяет номер таймера-счетчика.

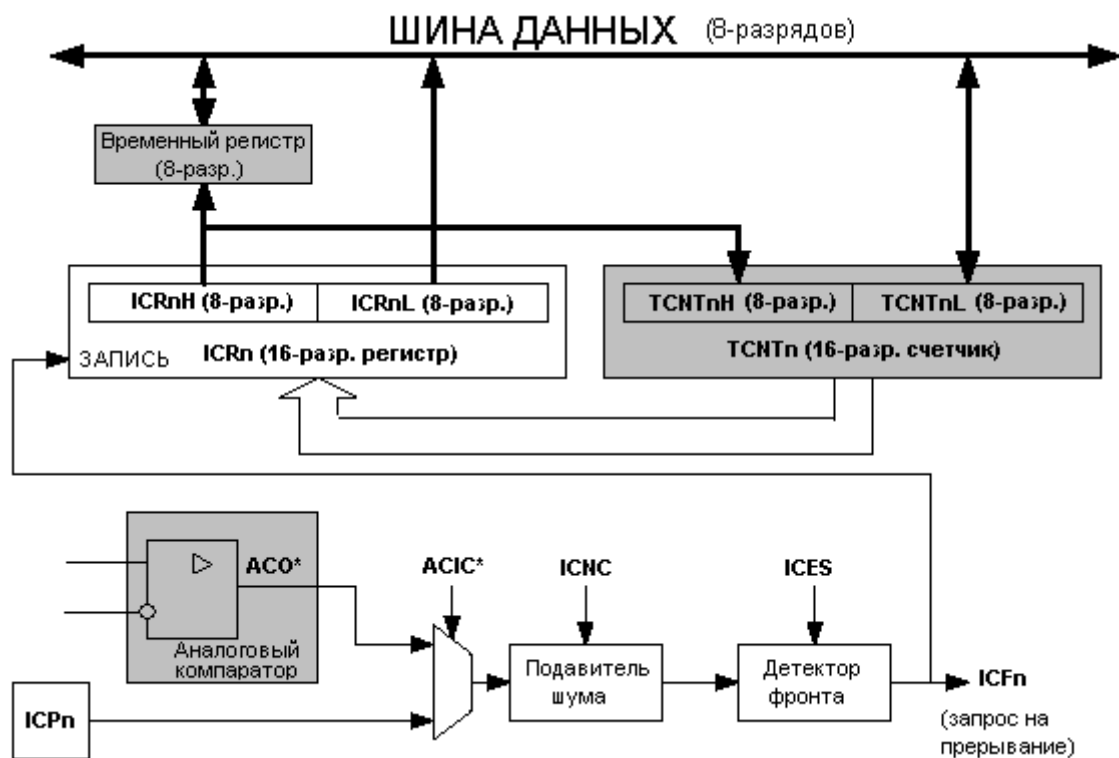


Рис. 4.17. Функциональная схема блока захвата

Прим.: Выход аналогового компаратора (ACO) только у таймера-счетчика 1 может выступать в качестве сигнала захвата. У таймера-счетчика 3 эта возможность отсутствует.

Если на входе захвата (ICPn) или альтернативно на выходе аналогового компаратора (ACO) возникает изменение логического уровня (событие), которое соответствует установкам детектора фронта, то выполняется захват состояния таймера. При этом 16-разр. значение содержимого таймера (TCNTn) помещается в регистр захвата (ICRn). Флаг захвата (ICFn) устанавливается на том же такте ЦПУ, на котором произошло копирование значения TCNTn в ICRn. Установка бита TICIEn = 1 разрешает прерывание по установке флага захвата. Флаг ICFn автоматически сбрасывается при переходе на вектор прерывания. Альтернативно флаг ICFn сбрасывается программно, если записать в него лог. 1. Считывание 16-разр. значения регистра захвата (ICRn) выполняется чтением сначала младшего байта (ICRnL), а затем старшего байта (ICRnH). При выполнении команды чтения младшего байта значение старшего байта автоматически копируется во временный регистр. Если ЦПУ выполняет команду чтения регистра ICRnH, то фактически считывается содержимое временного регистра. Запись в регистр ICRn возможна только в том случае, если битами задания режима работы таймера выбран режим, в котором значение регистра ICRn задает верхний предел счета. В этом случае необходимо выполнить соответствующую установку бит режима работы (WGMn3:0), а только затем выполнить запись значения верхнего предела в регистр ICRn. Запись 16-разр. значения в регистр ICRn выполняется путем записи сначала старшего байта в ICRnH, а только затем младшего байта в ICRnL (см. также «Доступ к 16-разр. Регистрам»).

**Источник срабатывания механизма захвата.** Основным источником, инициирующим захват состояния таймер-счетчика, является вывод захвата (ICPn). Таймер-счетчик 1 также альтернативно может использовать выход аналогового компаратора в качестве источника инициации захвата. Для этого необходимо установить бит разрешения захвата аналоговым компаратором (ACIC) в регистре состояния и управления аналогового компаратора (ACSR). Учтите, что изменение источника инициации захвата может привести к возникновению захвата. Поэтому после изменения источника должен быть сброшен флаг захвата.

Если разрешена работа подавителя шумов, то последовательно перед детектором фронта включается дополнительная логика, которая реагирует на изменение уровня, если он находился на постоянном уровне не менее 4 тактов ЦПУ. Обратите внимание, что вход подавителя шумов и детектора фронта всегда разрешен, если таймер-счетчик находится в режиме, где ICRn определяет вершину счета.

Захват можно инициировать программно путем управления настройками порта на выводе ICPn.

**Подавитель шума.** Подавитель шума позволяет повысить помехозащищенность за счет использования простой схемы цифровой фильтрации. Для изменения выходного состояния подавителя шума необходимо, чтобы на его входе совпало значение четырех выборок.

Работа подавителя шума разрешается путем установки бита разрешения подавления шумов на входе захвата (ICNCn) в регистре В управления тайме-

ром-счетчиком (TCCRnB). С разрешением работы подавителя шума вводится задержка на 4 такта между возникновением изменения входного уровня и собственно захватом состояния таймера. Подавитель шума использует системную синхронизацию и, следовательно, не зависит от настройки предделителя таймера.

**Использование блока захвата.** Основная проблема при использовании блока захвата состоит в конфликте отработки прерывания по захвату в случае обработки любого другого прерывания. Время между двумя событиями является критичным параметром. Если процессор не успевает считать содержимое ICRn до момента возникновения следующего события, то регистр ICRn обновит свое содержимое. В этом случае результат захвата будет некорректным.

Для измерения скважности (или коэффициента заполнения) импульсов необходимо после каждого захвата изменять фронт, по которому происходит захват. Данное изменение необходимо выполнять как можно раньше после считывания регистра ICRn. После изменения фронта необходимо сбросить флаг захвата ICFn записью в него лог. 1. Если требуется измерение только частоты, то сброс флага ICFn не требуется (если используется обработка по прерыванию).

**Блоки сравнения.** 16-разрядный цифровой компаратор непрерывно сравнивает значение TCNTn со значением регистра порога сравнения (OCRnx). Если значение TCNT равно OCRnx, то компаратор формирует сигнал совпадения (равенства). Следующий за совпадением такт ЦПУ устанавливает флаг сравнения (OCFnx). Если бит OCIEnx = 1, то установка флага сравнения приведет к генерации прерывания по результату сравнения. Флаг OCFnx автоматически сбрасывается после перехода на вектор обработки прерывания. Альтернативно флаг OCFnx сбрасывается программно, если записать в него лог. 1. Сигнал совпадения используется формирователем выходного сигнала, результирующая форма которого зависит от выбранного с помощью бит WGMn3:0 режима работы таймера и режима формирования импульсов (биты COMnx1:0). Сигналы ВЕРХНИЙ ПРЕДЕЛ и НИЖНИЙ ПРЕДЕЛ используются формирователем импульсов для отработки особых случаев задания экстремальных значений в некоторых режимах работы. У канала сравнения А имеется своя отличительная особенность, которая позволяет задать верхний предел счета (т. е. разрешающую способность счетчика). В дополнение к разрешающей способности верхний предел определяет период формируемых импульсов. На рисунке 4.18 показана функциональная схема блока сравнения. Символ «n» в наименованиях бит и регистров заменяет номер таймера (1 и 3), а «x» заменяет наименование канала сравнения (A/B/C).

В любом из 12 режимов широтно-импульсной модуляции ШИМ регистр OCRnx работает в качестве второго промежуточного буферного регистра. Если таймер работает в нормальном режиме или режиме сброса при совпадении (СТС), то двойная буферизация отключается. Двойная буферизация синхронизирует обновление регистра порога сравнения OCRnx по достижении верхнего или нижнего предела счета в зависимости от выбранного режима работы (алгоритма счета).



Если двойная буферизация разрешена, то ЦПУ фактически осуществляет доступ к буферному регистру OCRnx. Если же двойная буферизация отключена, то ЦПУ обращается к регистру OCRnx непосредственно. Содержимое регистра OCRnx считывается напрямую, а не через временный регистр старшего байта. Запись регистров OCRnx происходит через временный регистр, т. к. все 16 разр. участвуют в сравнении непрерывно. Первым необходимо записать старший байт OCRnxH. Если выполняется запись по адресу старшего байта, то содержимое временного регистра обновляется записываемым значением. Если выполняется запись младшего байта (OCRnxL), то параллельно копируется содержимое временного регистра в старшие 8-разряды буферного регистра OCRnx или регистра порога сравнения OCRnx, тем самым обновляя все 16 разрядов за один такт ЦПУ.

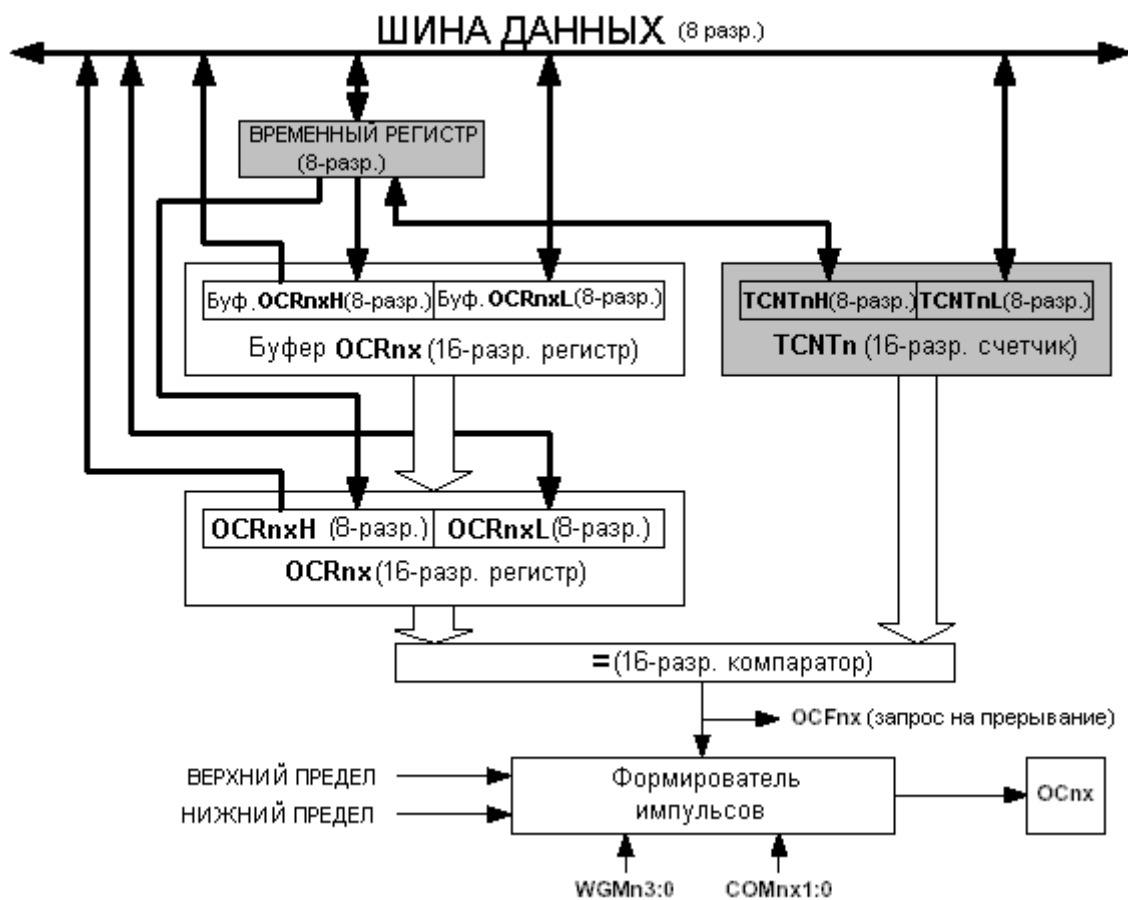


Рис. 4.18. Функциональная схема блока сравнения

**Принудительная установка результата сравнения.** В режимах генерации импульсов без ШИМ в формирователе импульсов результат сравнения может быть установлен непосредственно через бит принудительной установки результата сравнения FOCnx. Принудительная установка результата сравнения не приводит к установке флага OCFnx или сбросу/перезагрузке таймера, но влияет на состояние вывода ОСnx, который будет устанавливаться, сбрасываться или переключаться (инвертироваться) в зависимости от выбранной установки бит COMnx. Результат сравнения блокируется записью в TCNTn.

Если ЦПУ осуществляет запись в регистр TCNTn, то результат сравнения будет игнорироваться на следующем такте синхронизации таймера, даже если таймер остановлен. Данная функция позволяет установить в регистре OCRnx то же значение, что и в TCNTn без генерации запроса на прерывание, если разрешено тактирование таймера-счетчика.

**Использование блока сравнения.** Поскольку запись в TCNTn в любом из режимов работы блокирует отработку совпадения на один такт синхронизации таймера, то имеются некоторые опасные ситуации при изменении TCNTn во время использования любого из каналов сравнения независимо, работает таймер-счетчик или нет. Если в TCNTn записано значение, равное OCRnx, то возникающее совпадение игнорируется, тем самым вызывая некорректную генерацию импульсов. Следует избегать записи в TCNTn значения равного верхнему пределу в ШИМ-режимах с переменным значением верхнего предела. В этом случае совпадение по достижении верхнего предела игнорируется и счет продолжится до 0xFFFF. Аналогично, следует избегать записи в TCNTn значения равного нижнему пределу, если счетчик работает как вычитающий (обратный счет). Прежде чем настроить вывод OCnx на вывод в регистре направления данных необходимо выполнить инициализацию регистра OCnx. Самым простым способом решения этой задачи является использование бита принудительной установки результата сравнения (FOCnx) при работе таймера в нормальном режиме. Регистр OCnx сохраняет свое состояние даже при изменении режима работы таймера.

**Блок формирования выходного сигнала.** Биты задания режима формирования выходного сигнала (COMnx1:0) имеют двойное назначение. С одной стороны, биты COMnx1:0 используются формирователем сигнала и определяют какое логическое состояние должно быть на выходе OCnx при возникновении следующего совпадения. С другой стороны, биты COMnx1:0 используются для разрешения/запрета альтернативной функции вывода порта OCnx. На рисунке 4.19 представлена упрощенная логическая схема, где показаны только те регистры управления портом ввода-вывода (DDR и PORT), на которые оказывает действие биты COMnx1:0. Если происходит системный сброс, то выход регистра OCnx принимает нулевое состояние.

Функция линии универсального порта ввода-вывода заменяется на функцию выхода формирователя сигнала OCnx, если хотя бы один из бит COM01, COM00 установлен (логика ИЛИ). Однако управление направлением вывода OCnx (вход или выход) в этом случае остается за соответствующим битом регистра направления данных (DDR). Чтобы значение регистра OCnx присутствовало на выходе OCnx необходимо настроить данную линию на вывод (установить соотв. бит в DDRB). Управление вводом альтернативной функции не зависит от режима работы таймера за некоторыми исключениями (см. табл. 4.25 – 4.27).

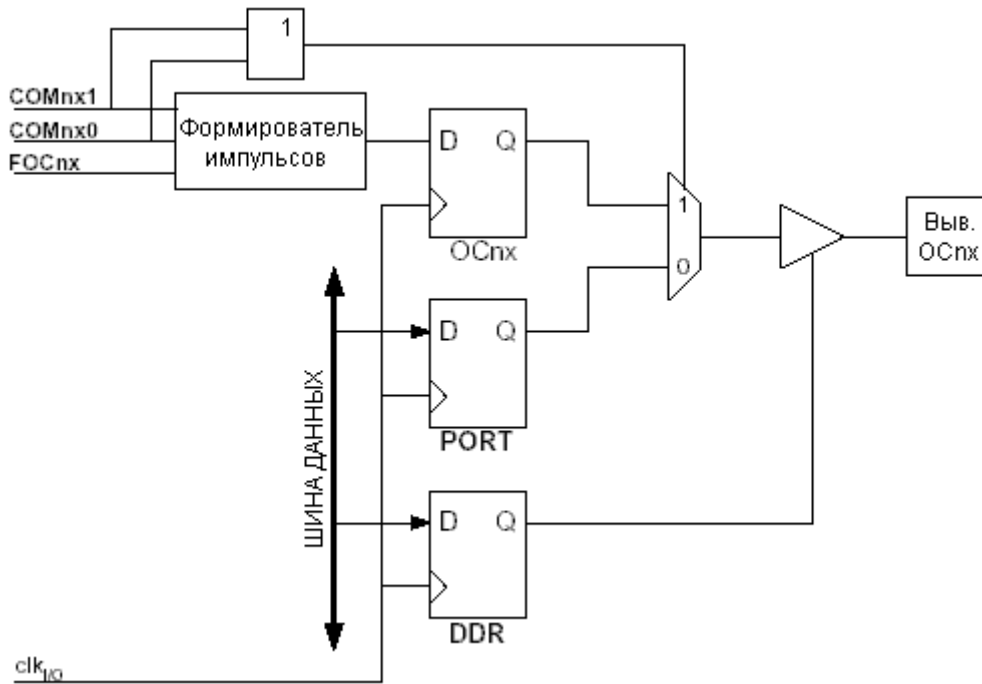


Рис. 4.19. Схема блока формирования выходного сигнала

Схемотехника выходной логики позволяет инициализировать состояние регистра OCnx перед разрешением настройки вывода OCnx в качестве выхода. Обратите внимание, что в некоторых режимах работы имеются зарезервированные состояния бит COMnx1:0. См. «Описание регистров 16-разр. таймеров-счетчиков». Установки бит COMnx1:0 не оказывают никакого влияния на работу блока захвата.

**Режимы генерации импульсов.** Установки бит COMnx1:0 оказывают различное влияние в зависимости от выбранного режима работы: нормального, сброса при совпадении и ШИМ. Общим для всех режимов работы является невыполнение каких-либо действий с регистром OCnx при возникновении совпадения, если COMnx1:0 = 0. В таблице 4.25 описано действие различных установок этих бит для режимов без ШИМ. Аналогичная информация для режима с быстрой ШИМ приведена в таблице 4.26, а для ШИМ с фазовой и частотной коррекцией в таблице 4.27.

Изменение состояния бит COMnx1:0 вступает в силу при следующем после их записи совпадении. В режимах без ШИМ воздействовать на генерацию импульсов можно с помощью стробирующего бита принудительной установки результата сравнения FOCnx.

**Режимы работы.** Под режимом работы 16-разр. таймера понимается его алгоритм счета и поведение связанного с ним выхода формирователя импульсов, что определяется комбинацией бит, задающих режим работы таймера (WGMn3-0) и режим формирования выходного сигнала (COMnx1:0). При этом биты задания режима формирования выходного сигнала не влияют на алгоритм счета, т. к. алгоритм счета зависит только от состояния бит задания режима работы таймера. В режимах с ШИМ биты COMnx1:0 позволяют включить/отключить инверсию на генерируемом ШИМ-выходе (т. е. выбрать ШИМ

с инверсией или ШИМ без инверсии). Для режимов без ШИМ биты COMnх1:0 определяют, какое действие необходимо выполнить при возникновении совпадения: сбросить, установить или инвертировать выход.

**Нормальный режим работы.** Самым простым режимом работы является нормальный режим ( $WGMn3-0 = 0b0000$ ). В данном режиме счетчик работает как суммирующий (инкрементирующий), при этом сброс счетчика не выполняется. Переполнение счетчика происходит при переходе через максимальное 16-разр. значение ( $0xFFFF$ ) к нижнему пределу счета ( $0x0000$ ). В нормальном режиме работы флаг переполнения таймера-счетчика TOVn будет установлен на том же такте синхронизации, когда TCNTn примет нулевое значение.

Фактически, флаг переполнения TOVn является 17-ым битом таймера-счетчика за тем исключением, что он только устанавливается и не сбрасывается. Однако программно это свойство может быть использовано для повышения разрешающей способности таймера, если использовать прерывание по переполнению таймера, при возникновении которого флаг TOVn сбрасывается автоматически. Для нормального режима работы не существует каких-либо особых ситуаций, поэтому запись нового состояния счетчика может быть выполнена в любой момент.

В нормальном режиме можно использовать блок захвата. Однако при этом следует учитывать влияние других внешних прерываний на корректность чтения результатов.

Блок сравнения может использоваться для генерации прерываний. Не рекомендуется использовать выход OCnх для генерации сигналов в нормальном режиме работы, т. к. в этом случае будет затрачена значительная часть процессорного времени.

**Режим сброса таймера при совпадении (СТС).** В режиме СТС ( $WGM01, WGM00 = 0b10$ ) регистр OCR0 используется для задания разрешающей способности счетчика. Если задан режим СТС и значение счетчика (TCNT0) совпадает со значением регистра OCR0, то счетчик обнуляется ( $TCNT0 = 0$ ). Таким образом, OCR0 задает вершину счета счетчика, а, следовательно, и его разрешающую способность. В данном режиме обеспечивается широкий диапазон регулировки частоты генерируемых прямоугольных импульсов. Он также упрощает работу счетчика внешних событий.

В режиме сброса таймера при совпадении ( $WGMn3-0 = 0b0100$  или  $0b1100$ ) разрешающая способность таймера задается регистрами OCRnA или ICRn. В режиме СТС происходит сброс счетчика (TCNTn), если его значение совпадает со значением регистра OCRnA ( $WGMn3-0 = 0b0100$ ) или с ICRn ( $WGMn3-0 = 0b1100$ ). Значение регистра OCRnA или ICRn определяет верхний предел счета, а, следовательно, и разрешающую способность таймера. В данном режиме обеспечивается широкий диапазон регулировки частоты генерируемых прямоугольных импульсов. Он также упрощает работу счетчика внешних событий. Временная диаграмма работы таймера в режиме СТС показана на рисунке 4.20. Счетчик (TCNTn) инкрементирует свое состояние до тех пор, по-

ка не возникнет совпадение со значением OCRnA или ICRn, а затем счетчик (TCNTn) сбрасывается.

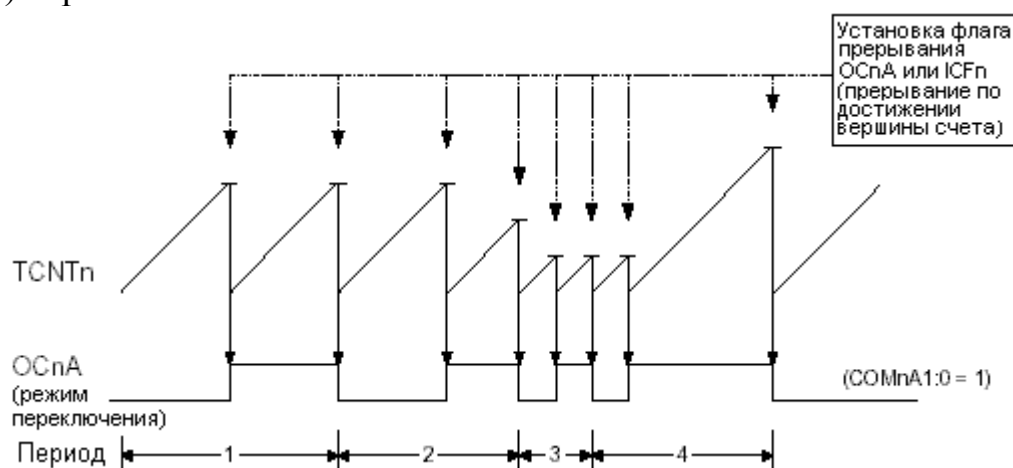


Рис. 4.20. Временная диаграмма для режима CTC

По достижении верхнего предела счета может генерироваться прерывание с помощью флагов OCFnA или ICFn, соответствующим используемым регистрам для задания верхнего предела счета. Если прерывание разрешено, то процедура обработки прерывания может использоваться для обновления верхнего предела счета. Однако задание значения вершины счета, близкого к значению нижнего предела счета, когда счетчик работает без предделения или с малым значением предделения, необходимо выполнять с особой осторожностью, т. к. в режиме CTC нет двойной буферизации. Если значение, записанное в OCRnA или ICRn, меньше текущего значения TCNTn, то сброс счетчика по условию совпадения наступит, когда он достигнет максимального значения (0xFFFF), затем перейдет в исходное состояние 0x0000 и достигнет нового значения OCRnA или ICRn. Во многих случаях возникновение такой ситуации не желательно. В качестве альтернативы может выступить режим быстрой ШИМ, где регистр OCRnA определяет верхний предел счета ( $WGMn3:0 = 0b1111$ ), т.к. в этом случае OCRnA имеет двойную буферизацию.

Для генерации сигнала в режиме CTC выход OCnA может использоваться для изменения логического уровня при каждом совпадении, для чего необходимо задать режим переключения ( $COMnA1, COMnA0 = 0b01$ ). Значение OCnA будет присутствовать на выводе порта, только если для данного вывода задано выходное направление. Максимальная частота генерируемого сигнала равна  $f_{OC0} = f_{clk\_I/O}/2$ , если  $OCRnA = 0x0000$ . Для других значений OCRn частоту генерируемого сигнала можно определить по формуле:

$$f_{OCnA} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnA)},$$

где переменная N задает коэффициент деления предделителя (1, 8, 32, 64, 128, 256 или 1024).

Так же, как и для нормального режима работы, флаг TOV0 устанавливается на том же такте таймера, когда его значение изменяется с 0xFFFF на 0x0000.

**Режим быстрой ШИМ.** Режим быстрой широтно-импульсной модуляции (ШИМ) ( $WGMn3:0 = 0b0101, 0b0110, 0b0111, 0b1110, 0b1111$ ) предназначен для генерации ШИМ-импульсов повышенной частоты. В отличие от других режимов работы в этом используется однонаправленная работа счетчика. Счет выполняется в направлении от нижнего к верхнему пределу счета.

Если задан неинвертирующий режим выхода, то при совпадении  $TCNTn$  и  $OCRnx$  сигнал  $OCnx$  устанавливается, а на верхнем пределе счета сбрасывается. Если задан инвертирующий режим, то выход  $OCnx$  сбрасывается при совпадении и устанавливается на верхнем пределе счета. За счет однонаправленности счета рабочая частота для данного режима в два раза выше по сравнению с режимом ШИМ с фазовой коррекцией, где используется двунаправленный счет. Возможность генерации высокочастотных ШИМ сигналов делает использование данного режима полезным в задачах стабилизации питания, выпрямления и цифро-аналогового преобразования.

Разрешающая способность ШИМ может быть фиксированной 8, 9 или 10 разрядов или задаваться регистром  $ICRn$  или  $OCRnA$ , но не менее 2 разрядов ( $ICRn$  или  $OCRnA = 0x0003$ ) и не более 16 разрядов ( $ICRn$  или  $OCRnA = 0xFFFF$ ). Разрешающая способность ШИМ при заданном значении верхнего предела (ВП) вычисляется следующим образом:

В режиме быстрой ШИМ счетчик инкрементируется до совпадения его значения с одним из фиксированных значений  $0x00FF$ ,  $0x01FF$  или  $0x03FF$  (если  $WGMn3:0 = 0b0101, 0b0110$  или  $0b0111$  соответственно), значением в  $ICRn$  (если  $WGMn3:0 = 0b1110$ ) или значением в  $OCRnA$  (если  $WGMn3:0 = 0b1111$ ), а затем сбрасывается следующим тактом синхронизации таймера. Временная диаграмма для режима быстрой ШИМ представлена на рисунке 4.21. На рисунке показан режим быстрой ШИМ, когда для задания верхнего предела используется регистр  $OCRnA$  или  $ICRn$ . Значение  $TCNTn$  на временной диаграмме показано в виде графика функции для иллюстрации однонаправленности счета. На диаграмме показаны как инвертированный, так и неинвертированный ШИМ-выходы. Короткой горизонтальной линией показаны точки на графике  $TCNTn$ , где совпадают значения  $OCRnx$  и  $TCNTnx$ . Флаг прерывания  $OCnx$  устанавливается при возникновении совпадений.

Флаг переполнения таймера-счетчика ( $TOVn$ ) устанавливается всякий раз, когда счетчик достигает верхнего предела. Дополнительно тем же тактовым импульсом вместе с флагом  $TOVn$  могут установиться флаги  $OCnA$  или  $ICFn$ , если для задания верхнего предела используется регистр  $OCRnA$  или  $ICRn$  соответственно. Если одно из этих прерываний разрешено, то в процедуре обработки прерывания может быть выполнено обновление верхнего предела счета и порогов сравнения.

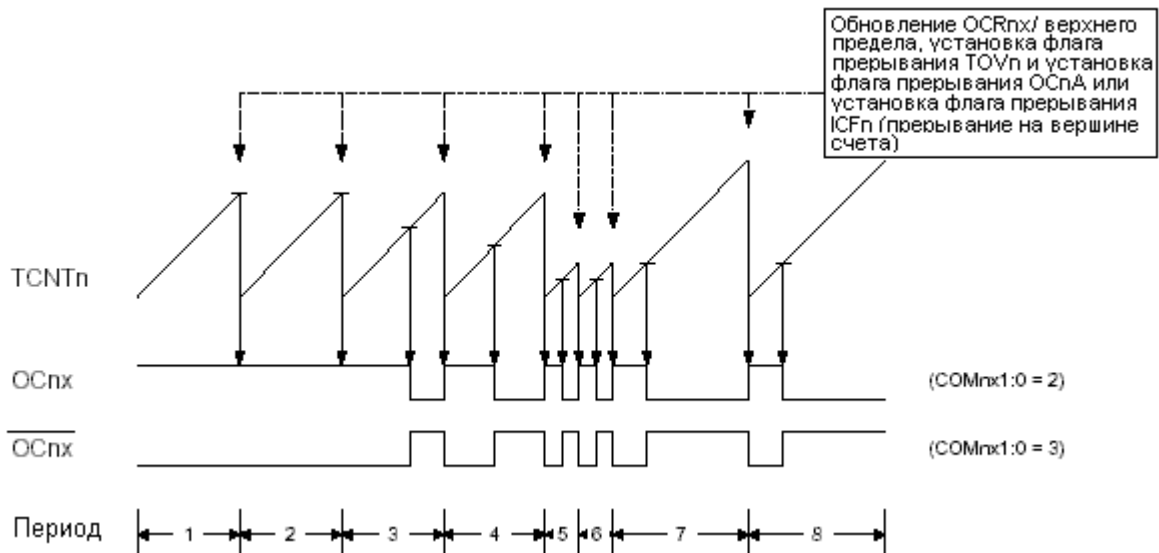


Рис. 4.21. Временная диаграмма для режима быстрой ШИМ

Механизм модификации регистра ICRn отличается от OCRnA в том случае, если он используется для задания верхнего предела. Регистр ICRn не имеет двойной буферизации. Это означает, что если в ICRn записывается малое значение во время работы счетчика с малым предделением или без него, то имеется опасность записи в регистр ICRn значения, которое окажется меньше текущего значения TCNTn. Как результат, в такой ситуации будет пропущено совпадение на вершине счета. В этом случае счетчик дойдет до максимального значения (0xFFFF), перезапустится со значения 0x0000, а только затем возникнет совпадение. Регистр OCRnA содержит схему двойной буферизации, поэтому, его можно модифицировать в любой момент времени.

Если выполняется запись по адресу OCRnA, то фактически значение помещается в буферный регистр OCRnA. Если же возникает совпадение между TCNTn и вершиной счета, то следующим тактом синхронизации таймера происходит копирование буферного регистра в регистр порога сравнения OCRnA. Обновление регистра выполняется тем же тактом, что и сброс TCNTn и установка флага TOVn.

Рекомендуется использовать регистр ICRn для задания верхнего предела, если верхний предел счета является константой. В этом случае также освобождается регистр OCRnA для генерации ШИМ-сигнала на выходе OCnA. Однако если частота ШИМ динамически изменяется (за счет изменения верхнего предела), то в этом случае выгоднее использовать регистр OCRnA для задания верхнего предела, т. к. он поддерживает двойную буферизацию.

В режиме быстрой ШИМ блоки сравнения позволяют генерировать ШИМ-сигналы на выводах OCnx. Если COMnx1:0 = 0b10, то задается ШИМ без инверсии выхода, а если COMnx1:0 = 0b11, то задается режим ШИМ с инверсией на выходе (см. таблицу 4.26). Фактическое значение OCnx можно наблюдать на выводе порта, если для него задано выходное направление (DDR\_OCnx). ШИМ-сигнал генерируется путем установки (сброса) регистра OCnx при возникновении совпадения между OCRnx и TCNTn, а также путем сброса (уста-

новки) регистра  $OCn_x$  вместе со сбросом счетчика (переход с верхнего предела на нижний предел).

Запись предельных значений в регистр  $OCRn_x$  связана с особыми случаями в генерации ШИМ-импульсов. Если  $OCRn_x$  установить равным нижнему пределу ( $0 \times 0000$ ), то на выходе будет возникать короткий импульс каждый  $(ВП+1)$ -ый такт синхронизации таймера. Запись в  $OCRn_x$  значения равного верхнему пределу приведет к установке постоянного уровня лог. 1 или 0 на выходе (зависит от выбранной с помощью бит  $COMn_x1$ : 0 полярности выходного сигнала).

Если требуется генерация меандра (прямоугольные импульсы со скважностью 2 или заполнением 50%) высокой частоты, то необходимо использовать режим быстрой ШИМ с установкой бит  $COMnA1:0 = 0b01$ , которая вызывает переключение (инвертирование) логического уровня на выходе  $OCnA$  при каждом совпадении. Данное применимо в том случае, если  $OCRnA$  используется для задания верхнего предела ( $WGMn3-0 = 0b1111$ ). Максимальная генерируемая частота меандра в этом случае  $fOCnA = fclk\_I/O/2$ , если  $OCRnA = 0 \times 0000$ . Данная особенность аналогична переключению  $OCnA$  в режиме СТС за исключением двойной буферизации, которая имеется в режиме быстрой ШИМ.

**Режим широтно-импульсной модуляции с фазовой коррекцией.** Режим широтно-импульсной модуляции с фазовой коррекцией (ШИМ ФК) ( $WGMn3-0 = 0b0001, 0b010, 0b0011, 0b1010$  или  $0b1011$ ) предназначен для генерации ШИМ сигнала с фазовой коррекцией и высокой разрешающей способностью. Режим ШИМ ФК основан на двунаправленной работе таймера-счетчика. Счетчик циклически выполняет счет в направлении от нижнего предела ( $0 \times 0000$ ) до верхнего предела, а затем обратно от верхнего предела к нижнему пределу. Если задан неинвертирующий режим выхода формирователя импульсов, то выход  $OCn_x$  сбрасывается/устанавливается при совпадении значений  $TCNTn$  и  $OCRn_x$  во время прямого/обратного счета. Если задан инвертирующий режим выхода, то, наоборот, во время прямого счета происходит установка, а во время обратного – сброс выхода  $OCn_x$ . При двунаправленной работе максимальная частота ШИМ-сигнала меньше, чем при однонаправленной работе, однако, за счет такой особенности, как симметричность в режимах ШИМ с двунаправленной работой, данные режимы предпочитают использовать при решении задач управления приводами.

Разрешающая способность ШИМ в данном режиме может быть либо фиксированной (8, 9 или 10 разрядов), либо задаваться с помощью регистра  $ICRn$  или  $OCRnA$ . Минимальная разрешающая способность равна 2-м разрядам ( $ICRn$  или  $OCRnA = 0 \times 0003$ ), а максимальная – 16-ти разрядам ( $ICRn$  или  $OCRnA = 0 \times FFFF$ ). Если задан верхний предел, то разрешающая способность ШИМ в данном режиме определяется следующим образом:

В режиме ШИМ ФК счетчик инкрементируется пока не достигнет одного из фиксированных значений  $0 \times 00FF$ ,  $0 \times 01FF$  или  $0 \times 03FF$  (соответственно для  $WGMn3-0 = 0b0001, 0b010$  или  $0b0011$ ), а также значения равного  $ICRn$  (если  $WGMn3-0 = 0b1010$ ) или  $OCRnA$  (если  $WGMn3:0 = 0b1011$ ). Далее, при дости-



жении верхнего предела, счетчик изменяет направление счета. Значение TCNTn остается равным верхнему пределу в течение одного такта синхронизации таймера. Временная диаграмма для режима ШИМ ФК представлена на рисунке 4.22. На рисунке показан режим ШИМ ФК с использованием регистра OCRnA или ICRn для задания верхнего предела. Состояние TCNTn представлено в виде графика функции для иллюстрации двунаправленности счета. На рисунке представлены, как неинвертированный, так и инвертированный ШИМ-выход. Короткие горизонтальные линии указывают точки на графике изменения TCNTn, где возникает совпадение со значением OCRnx. Флаг прерывания OCnx устанавливается при возникновении совпадения.

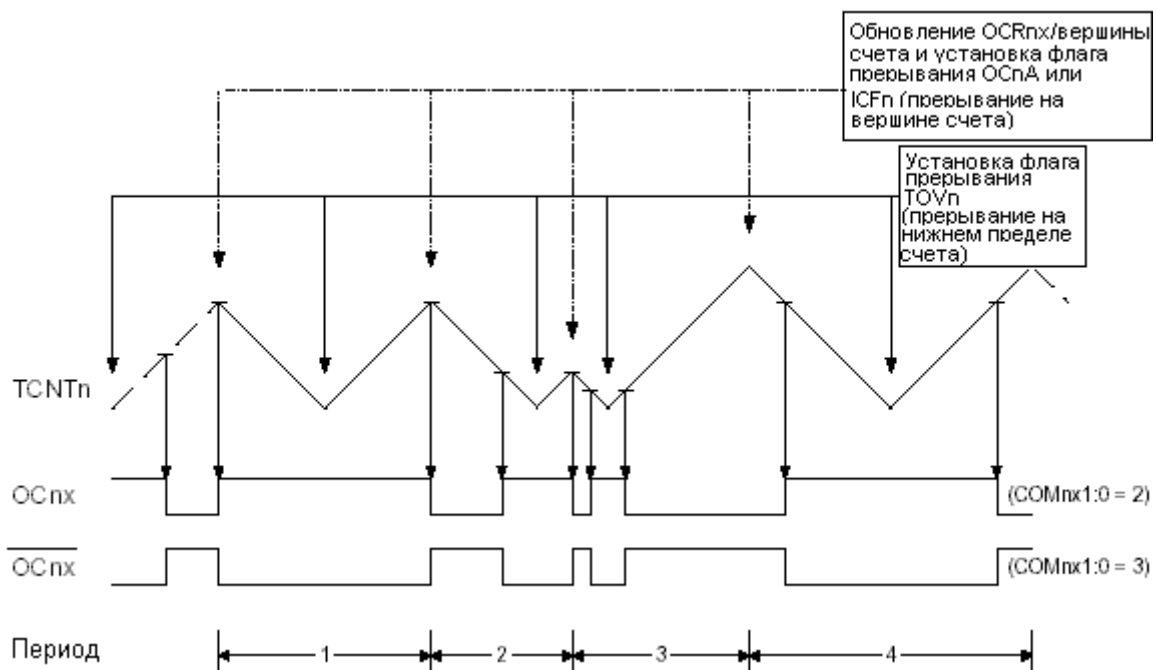


Рис. 4.22. Временная диаграмма для режима ШИМ ФК

Флаг переполнения таймера-счетчика (TOVn) устанавливается всякий раз, когда счетчик достигает нижнего предела. Если для задания верхнего предела используется регистр OCRnA или ICRn, то, соответственно, устанавливается флаг OCnA или ICFn тем же тактовым импульсом, на котором произошло обновление регистра OCRnx из буферного регистра (на вершине счета). Флаги прерывания могут использоваться для генерации прерывания по достижении счетчиком нижнего или верхнего предела.

При изменении значения верхнего предела счета необходимо следить, чтобы оно было больше или равно значениям во всех регистрах сравнения. В противном случае совпадение между TCNTn и OCRnx никогда не возникнет. Обратите внимание, что при использовании фиксированных значений верхнего предела счета во время записи в регистры OCRnx неиспользуемые разряды обнуляются. Третий период на рисунке 4.22 иллюстрирует случай, когда динамическое изменение верхнего предела счета приводит к генерации несимметричного импульса. Данная особенность основывается на времени обновления регистра OCRnx. Поскольку обновление OCRnx возникает на вершине счета, то и

период ШИМ начинается и заканчивается на вершине счета. Это подразумевает, что длительность обратного счета определяется предыдущим значением верхнего предела, а прямого – новым значением верхнего предела. Если два этих значения разные, то и длительность прямого и обратного счета будет также отличаться. Различие в длительности приводит к несимметричности выходных импульсов.

Если стоит задача изменения верхнего предела при работающем счетчике, то вместо этого режима рекомендуется использовать режим ШИМ ФЧК (фазовая и частотная коррекция). Если используется статическое значение верхнего предела, то между данными режимами практически нет отличий.

В режиме ШИМ ФК блоки сравнения позволяют генерировать ШИМ-сигналы на выводах ОС<sub>nх</sub>. Если установить COM<sub>nх</sub>1:0 = 0b10, то выход ШИМ будет без инверсии, а если COM<sub>nх</sub>1:0 = 0b11, то с инверсией (см. таблицу 4.27). Фактическое значение ОС<sub>nх</sub> можно наблюдать на выводе порта, если в регистре направления данных для данного вывода порта задано выходное направление (DDR\_OC<sub>nх</sub>). ШИМ-сигнал генерируется путем установки (сброса) регистра ОС<sub>nх</sub> при совпадении значений OCR<sub>nх</sub> и TCNT<sub>n</sub> во время прямого счета, а также путем сброса (установки) регистра ОС<sub>nх</sub> при совпадении между OCR<sub>nх</sub> и TCNT<sub>n</sub> во время обратного счета. Результирующая частота ШИМ-сигнала в режиме ШИМ ФК при заданном верхнем пределе (ВП) может быть вычислена по следующему выражению:

$$f_{OCnхPFCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP},$$

где N – коэффициент деления предделителя (1, 8, 32, 64, 128, 256 или 1024).

Запись предельных значений в регистр OCR<sub>nх</sub> связано с особыми случаями в генерации ШИМ-сигналов в режиме ШИМ ФК. Если задать режим ШИМ без инверсии и OCR<sub>nх</sub> установить равным нижнему пределу, то на выходе непрерывно будет установлен лог. 0, а если равным верхнему пределу, то на выходе постоянно присутствует лог. 1. Для ШИМ с инверсией указанные уровни необходимо заменить противоположными.

Если задать использование ОС<sub>nА</sub> в качестве верхнего предела (WGM<sub>n</sub>3:0 = 0b1011) и установить COM<sub>nА</sub>1:0 = 0b01, то на выходе ОС<sub>nА</sub> будет генерироваться меандр.

**Режим широтно-импульсной модуляции с фазовой и частотной коррекцией.** Режим широтно-импульсной модуляции с фазовой и частотной коррекцией (ШИМ ФЧК) (WGM<sub>n</sub>3:0 = 0b1000 или 0b1001) предназначен для генерации ШИМ-импульсов высокой разрешающей способности с фазовой и частотной коррекцией. Также как и режим ШИМ ФК режим ШИМ ФЧК основан на двунаправленной работе счетчика. Счетчик циклически считает от нижнего предела (0x0000) до верхнего предела, а затем обратно от верхнего предела к нижнему пределу. Если задан неинвертирующий режим ШИМ, то выход ОС<sub>nх</sub> сбрасывается, если возникает совпадение между TCNT<sub>n</sub> и OCR<sub>nх</sub> во время

прямого счета, и устанавливается, если возникает совпадение во время обратного счета. В инвертирующем режиме работа инверсная.

Основное отличие между режимами ШИМ ФК и ШИМ ФЧК состоит в моменте обновления регистра OCRnх из буферного регистра OCRnх (см. рис. 4.22 и рис. 4.23).

Разрешающая способность ШИМ в этом режиме может задаваться с помощью регистра ICRn или OCRnA. Минимальная разрешающая способность равна 2-ум разрядам (ICRn или OCRnA = 0x0003), а максимальная разрешающая способность – 16-ти разрядам (ICRn или OCRnA = 0xFFFF). Разрешающая способность ШИМ в разрядах может быть вычислена по следующему выражению:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}.$$

В режиме ШИМ ФЧК счетчик инкрементируется до совпадения со значением в ICRn (WGMn3:0 = 0b1000) или в OCRnA (WGMn3:0 = 0b1001). Это означает достижение вершины счета, после чего происходит изменение направления счета. Значение TCNTn остается равным вершине счета в течение одного такта синхронизации таймера. Временная диаграмма для режима ШИМ ФЧК показана на рисунке 4.28. На рисунке показан режим ШИМ ФЧК, когда вершину счета задает регистр OCRnA или ICRn. Значение TCNTn показано в виде графика функции для иллюстрации двунаправленности счета. На диаграмме показан как неинвертирующий, так и инвертирующий ШИМ выходы. Короткие горизонтальные линии указывают на точки график TCNTn, где возникает совпадение между OCRnх и TCNTn. Флаг прерывания OCnх устанавливается после возникновения совпадения.

Флаг переполнения таймера-счетчика (TOVn) устанавливается тем же тактом, когда произошло обновление регистров значением из буферного регистра (на нижнем пределе счета). Если для задания верхнего предела используется регистр OCRnA или ICRn, то по достижении счетчиком верхнего предела устанавливается флаг OCnA или ICFn соответственно. Флаги прерывания могут использоваться для генерации прерывания при достижении счетчиком верхнего или нижнего предела.

При изменении верхнего предела необходимо следить, чтобы новое значение было больше или равно значениям во всех регистрах порога сравнения. В противном случае, если задано значение верхнего предела меньше любого из значений регистров порога сравнения, совпадение между TCNTn и OCRnх никогда не наступит.

На рисунке 4.23 показано, что в отличие от режима ШИМ ФК, генерируемый выходной сигнал симметричен на всех периодах. Поскольку регистры OCRnх обновляются на нижнем пределе счета, то длительности прямого и обратного счетов всегда равны. В результате выходные импульсы имеют симметричную форму, а, следовательно, и откорректированную частоту.

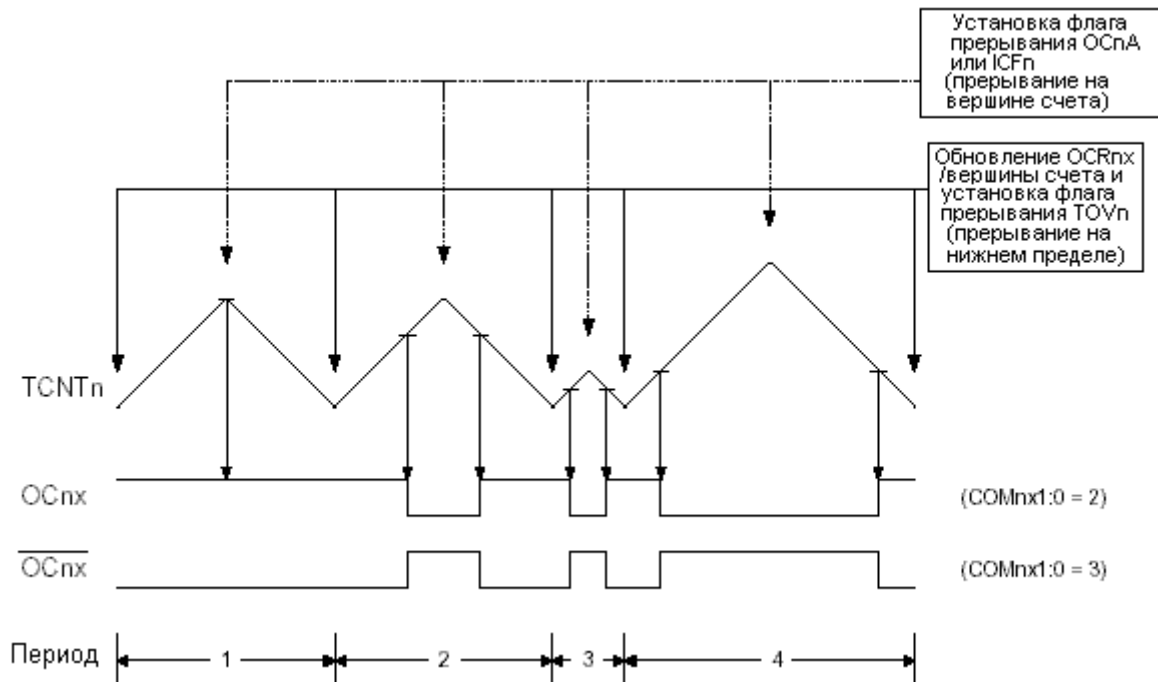


Рис. 4.23. Временная диаграмма режима ШИМ с фазовой и частотной коррекцией

Использование регистра ICRn для задания верхнего предела рекомендуется, если значение верхнего предела является константой. В этом случае также освобождается регистр OCRnA для широтно-импульсной модуляции импульсов на выводе OCnA. Однако если требуется динамическое изменение частоты ШИМ за счет изменения верхнего предела, то для задания верхнего предела рекомендуется использовать регистр OCRnA за счет наличия у него двойной буферизации.

В режиме ШИМ ФЧК блоки сравнения позволяют генерировать ШИМ-импульсы на выводе OCnx. Если COMnx1:0 = 0b10, то задается неинвертирующий ШИМ выход, а, если COMnx1:0 = 0b11, то инвертирующий (см. таблицу 4.27). Значение OCnx будет присутствовать на соответствующем выводе порта только в случае, если для него задано выходное направление. ШИМ сигнал генерируется путем установки (сброса) регистра OCnx при совпадении между OCRnx и TCNTn во время прямого счета и сброса (установки) регистра OCnx при совпадении между OCRnx и TCNTn во время обратного счета. Частота ШИМ в данном режиме при заданном верхнем пределе (TOP) счета определяется следующим образом:

$$f_{OCnxPFCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP},$$

где N – коэффициент деления предделителя (1, 8, 32, 64, 128, 256 или 1024).

Запись предельных значений в регистр OCRnx связана с особыми случаями в генерации ШИМ-сигналов в данном режиме. Если задать OCRnx равным нижнему пределу (0x0000), то в неинвертирующем режиме на выходе будет постоянно присутствовать низкий логический уровень, а при записи зна-

чения равного верхнему пределу на выходе будет длительно присутствовать высокий логический уровень. В инвертирующем режиме приведенные уровни будут противоположными.

Если OCRnA используется для задания верхнего предела ( $WGMn3:0 = 0b1001$ ) и  $COMnA1:0 = 0b01$ , то на выходе OCnA будет генерироваться меандр.

### Описание регистров 16-разр. таймеров-счетчиков

#### Регистр А управления таймером-счетчиком 1 – TCCR1A

Разряд	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	<b>TCCR1A</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

#### Регистр А управления таймером-счетчиком 3 – TCCR3A

Разряд	7	6	5	4	3	2	1	0	
	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	<b>TCCR3A</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряды 7:6 – COMnA1:0: Режим формирования выходного сигнала канала А

Разряды 5:4 – COMnB1:0: Режим формирования выходного сигнала канала В

Разряды 3:2 – COMnC1:0: Режим формирования выходного сигнала канала С

Биты COMnA1:0, COMnB1:0 и COMnC1:0 влияют на работу выводов OCnA, OCnB и OCnC, соответственно. Если один или оба бита COMnA1:0 равны 1, то вывод OCnA переходит к выполнению альтернативной функции, запрещающей его работу как обычного порта ввода-вывода. Аналогичные изменения происходят с выводами OCnB и OCnC во время записи лог. 1 в один из битов COMnB1:0 и COMnC1:0, соответственно. Однако необходимо учитывать, что остается влияние на работу данных выводов со стороны регистра направления данных (DDR) и в соответствующих разрядах этого регистра должно быть задано выходное направление для выводов OCnA, OCnB или OCnC.

Если выбрано подключение сигналов OCnA, OCnB или OCnC к выводам микроконтроллера, то назначение бит COMnx1:0 определяется выбранным с помощью бит WGMn3:0 режима работы таймера-счетчика. В таблице 4.25 показано назначение бит COMnx1:0, когда битами WGMn3:0 выбран режим сброса при совпадении (CTC) или нормальный режим, т.е. режимы без ШИМ.

Таблица 4.25

Режимы формирования выходного сигнала в режимах работы таймера без ШИМ

COMnA1/COMnB1/COMnC1	COMnA0/COMnB0/COMnC0	Описание
0	0	Нормальная работа порта, сигналы OCnA/OCnB/OCnC отключены.
0	1	Переключение (инвертирование) OCnA/OCnB/OCnC при совпадении.

Окончание табл. 4.25

1	0	Сброс ОСнА/ОСнВ/ОСнС при совпадении (установка лог. 0).
1	1	Установка ОСнА/ОСнВ/ОСнС при совпадении (установка лог. 1).

В таблице 4.26 представлено назначение бит COMnх1:0, когда с помощью бит WGMn3:0 выбран режим быстрой ШИМ.

Таблица 4.26

Режим формирования выходного сигнала в режиме работа таймера с быстрой ШИМ

COMnA1/COMnB1/COMnC1	COMnA0/COMnB0/COMnC0	Описание
0	0	Нормальная работа порта, сигналы ОСнА/ОСнВ/ОСнС отключены.
0	1	WGMn3:0 = 15: Переключение (инвертирование) ОСнА при совпадении, ОСнВ/ОСнС отключены (нормальная работа порта). Для всех других установок WGMn соответствует нормальная работа порта, когда ОСнА/ОСнВ/ОСнС отключены.
1	0	Сброс ОСнА/ОСнВ/ОСнС при совпадении, установка ОСнА/ОСнВ/ОСнС на вершине счета
1	1	Установка ОСнА/ОСнВ/ОСнС при совпадении, сброс ОСнА/ОСнВ/ОСнС на вершине

Прим.: Имеются особые случаи, когда OCRnА/OCRnВ/OCRnС равно верхнему пределу счета и установлен COMnA1/COMnB1/COMnC1. В этом случае возникшее совпадение игнорируется, но установка или сброс на вершине счета выполняется (см. “Режим быстрой ШИМ”).

В таблице 4.26 представлено назначение бит COMnх1:0 для режима ШИМ ФК и ШИМ ФЧК

Таблица 4.27

Режим формирования выходного сигнала в режимах работы таймера с ШИМ ФК и ШИМ ФЧК

COMnA1/COMnB1/COMnC1	COMnA0/COMnB0/COMnC0	Описание
0	0	Нормальная работа порта, сигналы ОСнА/ОСнВ/ОСнС отключены.
0	1	WGMn3:0 = 9 или 14: Переключение (инвертирование) ОСнА при совпадении, ОСнВ/ОСнС отключены (нормальная работа порта). Для всех других установок WGMn соответствует нормальная работа порта, когда ОСнА/ОСнВ/ОСнС отключены.
1	0	Сброс ОСнА/ОСнВ/ОСнС при совпадении во время прямого счета, установка ОСнА/ОСнВ/ОСнС при совпадении во время обратного счета
1	1	Установка ОСнА/ОСнВ/ОСнС при совпадении во время прямого счета, сброс ОСнА/ОСнВ/ОСнС при совпадении во время обратного счета

Прим.: Имеются особые случаи, когда OCRnА/OCRnВ/OCRnС равно верхнему пределу и установлен COMnA1/COMnB1/COMnC1 (см. “Режим ШИМ с фазовой коррекцией”).

Разряд 1:0 – WGMn1:0: Режим работы таймера-счетчика.

В сочетании с битами WGMn3:2 из регистра TCCRnB данные биты определяют алгоритм счета, источник для задания вершины счета (ВП) и тип генерируемой формы сигнала (см. табл. 4.28). Таймер-счетчик может работать в одном из следующих режимов: нормальный режим (счетчик), сброс таймера при совпадении (СТС) и три режима с широтно-импульсной модуляцией (ШИМ) (см. «Режимы работы 16-разр. таймеров-счетчиков»).

Таблица 4.28

Режимы работы таймера-счетчика

Режим	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Режим работа таймера-счетчика(1)	Верхний предел	Обновление OCRnх	Установка флага TOVn на:
0	0	0	0	0	Нормальный	0xFFFF	сразу после записи	МАКС
1	0	0	0	1	8-разр. ШИМ ФК	0x00FF	на вершине счета	нижнем пределе
2	0	0	1	0	9-разр. ШИМ ФК	0x01FF	на вершине счета	нижнем пределе
3	0	0	1	1	10-разр. ШИМ ФК	0x03FF	на вершине счета	нижнем пределе
4	0	1	0	0	СТС	OCRnA	сразу после записи	МАКС
5	0	1	0	1	8-разр. быстрая ШИМ	0x00FF	на вершине счета	на вершине счета
6	0	1	1	0	9-разр. быстрая ШИМ	0x01FF	на вершине счета	на вершине счета
7	0	1	1	1	10-разр. быстрая ШИМ	0x03FF	на вершине счета	на вершине счета
8	1	0	0	0	ШИМ ФЧК	ICRn	на нижнем пределе	нижнем пределе
9	1	0	0	1	ШИМ ФЧК	OCRnA	на нижнем пределе	нижнем пределе
10	1	0	1	0	ШИМ ФК	ICRn	на вершине счета	нижнем пределе
11	1	0	1	1	ШИМ ФК	OCRnA	на вершине счета	нижнем пределе
12	1	1	0	0	СТС	ICRn	сразу после записи	МАКС.
13	1	1	0	1	(резерв)	—	—	—
14	1	1	1	0	Быстрая ШИМ	ICRn	на вершине счета	на вершине счета
15	1	1	1	1	Быстрая ШИМ	OCRnA	на вершине счета	на вершине счета

Прим.: 1. Наименования бит CTCn и PWMn1:0 являются устаревшими, поэтому, необходимо использовать имена WGMn2:0. Однако назначение и расположение этих бит совместимо с предыдущими версиями таймеров.

**Регистр В управления таймером-счетчиком 1 – TCCR1B**

Разряд	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	<b>TCCR1B</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

**Регистр В управления таймером-счетчиком 3 – TCCR3B**

Разряд	7	6	5	4	3	2	1	0	
	ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30	<b>TCCR3B</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряд 7 – ICNCn: Подавитель шума на входе захвата. Установка данного бита (запись лог. 1) активизирует подавитель шума на входе захвата. После активизации подавителя шумов сигнал с вывода ICPn пропускается через фильтр. Логика работы фильтра состоит в определении четырех подряд равных по значению выборок и только в этом случае изменении своего выходного состояния. Следовательно, после разрешения подавления шумов сигнал с входа захвата будет задерживаться на 4 такта системной синхронизации.

Разряд 6 – ICESn: Выбор детектируемого фронта на входе захвата. Данный бит позволяет задать, какой фронт на входе захвата ICPn приведет к захвату состояния таймера. Если ICESn = 0, то падающий (отрицательный) фронт приводит к захвату состояния таймера, а если же ICESn = 1, то нарастающий (положительный) фронт приводит к возникновению захвата.

Если в соответствии с установкой ICESn возникает условие захвата, то содержимое счетчика копируется в регистр захвата ICRn. При этом также устанавливается флаг захвата ICFn, который может использоваться для генерации прерывания по захвату (если данное прерывание разрешено).

Если регистр ICRn используется для хранения значения верхнего предела счета (см. табл. 61), то вход ICPn отключается от соответствующего вывода микроконтроллера и функция захвата блокируется.

Разряд 5 – Зарезервированный бит. Данный бит зарезервирован для дальнейшего использования. В целях совместимости с будущими разработками рекомендуется во время записи в регистр TCCRnB в данном разряде указывать лог. 0.

Разряд 4:3 – WGMn3:2: Режим работы таймера-счетчика. См. описание регистр TCCRnA.

Разряд 2:0 – CSn2:0: Выбор тактового источника

Данный три бита позволяют выбрать тактовый источник для таймера-счетчика.



Таблица 4.30

## Описание бит выбора тактового источника

CSn2	CSn1	CSn0	Описание
0	0	0	Нет синхронизации. Таймер-счетчик остановлен.
0	0	1	clkI/O/1 (без предделения)
0	1	0	clkI/O /8 (с предделением)
0	1	1	clkI/O/64 (с предделением)
1	0	0	clkI/O/256 (с предделением)
1	0	1	clkI/O/1024 (с предделением)
1	1	0	Внешний тактовый источник с выв. Tn. Синхронизация по падающему фронту.
1	1	1	Внешний тактовый источник с выв. Tn. Синхронизация по нарастающему фронту.

Если для тактирования таймера выбран внешний вывод Tn, то данная функция за ним сохраняется, даже при его настройке на вывод. Данная функция позволяет программно управлять счетом.

## Регистр С управления таймером-счетчиком 1 – TCCR1C

Разряд	7	6	5	4	3	2	1	0	
	FOC1A	FOC1B	FOC1C	-	-	-	-	-	<b>TCCR1C</b>
Чтение/запись	Зп.	Зп.	Зп.	Чт.	Чт.	Чт.	Чт.	Чт.	
Исх. значение	0	0	0	0	0	0	0	0	

## Регистр С управления таймером-счетчиком 3 – TCCR3C

Разряд	7	6	5	4	3	2	1	0	
	FOC3A	FOC3B	FOC3C	-	-	-	-	-	<b>TCCR3C</b>
Чтение/запись	Зп.	Зп.	Зп.	Чт.	Чт.	Чт.	Чт.	Чт.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряд 7 – FOCnA: Бит принудительной установки результата сравнения для канала А.

Разряд 6 – FOCnB: Бит принудительной установки результата сравнения для канала В.

Разряд 5 – FOCnC: Бит принудительной установки результата сравнения для канала С.

Биты FOCnA/FOCnB/FOCnC становятся активными, когда с помощью бит WGMn3:0 выбран режим без ШИМ. В этом случае запись в FOCnA/FOCnB/FOCnC лог. 1 приводит к немедленной установке результата сравнения на входе блока формирователя сигнала. Выход OCnA/OCnB/OCnC изменяется в соответствии с установками бит COMnx1:0. Обратите внимание, что биты FOCnA/FOCnB/FOCnC реализованы как стробы. Стробы FOCnA/FOCnB/FOCnC не генерируют каких-либо прерываний и сбрасывают счетчик в режиме сброса таймера при совпадении (СТС), где OCRnA используется для задания вершины счета.





**Регистр захвата 3 – ICR3H и ICR3L**

Разряд	7	6	5	4	3	2	1	0	
	<b>ICR3C [15:8]</b>								<b>ICR3CH</b>
	<b>ICR3C [7:0]</b>								<b>ICR3CL</b>
Чтение/запись	Зп./Чт.	Зп./Чт.	Зп./Чт.	Зп./Чт.	Зп./Чт.	Зп./Чт.	Зп./Чт.	Зп./Чт.	
Исх. значение	0	0	0	0	0	0	0	0	

Регистры захвата обновляются содержимым соответствующего счетчика (TCNTn) при каждом определении условия захвата на входе ICPn (или альтернативно на выходе аналогового компаратора для таймера-счетчика 1).

Регистры захвата альтернативно могут использоваться для задания верхнего предела счета.

Регистры захвата также являются 16-разрядными, поэтому, одновременность записи младшего и старшего байтов достигнута за счет использования 8-разр. временного регистра старшего байта (TEMP). Временный регистр является общим для всех 16-разрядных регистров таймера (см. также «Доступ к 16-разр. Регистрам»).

**Регистр маски прерываний таймера-счетчика – TIMSK**

Разряд	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	<b>TIMSK</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Прим.: Данный регистр биты управления прерываниями для нескольких таймер-счетчиков, но в данном разделе детализированы только биты таймера 1. Описание остальных бит необходимо искать при описании соответствующих таймеров.

**Разряд 5 – TICIE1:** Разрешение прерывания по захвату состояния таймера-счетчика 1. Если в данный бит записана лог. 1, а также установлен флаг I в регистре статуса (активно общее разрешение прерываний), то разрешается прерывание по захвату состояния таймера-счетчика 1. Если устанавливается флаг в регистре TIFR, программа переходит на соответствующий вектор прерывания (см. раздел «Прерывания»).

**Разряд 4 – OCIE1A:** Разрешение прерывания по результату сравнения канала А таймера-счетчика 1. Если в данный бит записана лог. 1 и установлен флаг I в регистре статуса, то разрешается работа прерывания по результату сравнения канала А. Если устанавливается флаг OCF1A в регистре TIFR, то программа переходит на соответствующий вектор прерываний (см. раздел «Прерывания»).

**Разряд 3 – OCIE1B:** Разрешение прерывания по результату сравнения канала В таймера-счетчика 1. Действие аналогично предыдущему, но в отношении канала сравнения В.

Разряд 2 – TOIE1: Разрешение прерывания при переполнении таймера-счетчика 1. Если в данный бит записана лог. 1 и установлен флаг I в регистре статуса, то разрешается прерывание по переполнению таймера-счетчика 1. После этого, установка флага TOV1 в регистре TIFR приведет к переходу на соответствующий вектор прерывания.

**Расширенный регистр маски прерываний таймера-счетчика– ETIMSK**

Разряд	7	6	5	4	3	2	1	0	
	-	-	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	TOIE3C	ETIMSK
Чтение/запись	Чт.	Чт.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Прим.: Данный регистр не доступен в режиме совместимости с ATmega103.

Разряд 7:6 – Зарезервированные биты. Данные биты зарезервированы для дальнейшего использования. В целях совместимости с последующими версиями во время записи в регистр в данный бит необходимо записывать лог. 0.

Разряд 5 – TICIE3: Разрешение прерывания по захвату состояния таймера-счетчика 3. Если в данный бит записана лог. 1 и установлен флаг I в регистре статуса, то разрешается работа прерывания по захвату состояния таймера-счетчика 3. Если устанавливается флаг ICF3 в регистре ETIFR, то программа переходит на соответствующий вектор прерывания (см. раздел "Прерывания").

Разряд 4 – OCIE3A: Разрешение прерывания по результату сравнения канала A таймера-счетчика 3. Если данный бит равен 1 и установлен флаг I в регистре статуса, то разрешается работа прерывания по результату сравнения канала A таймера-счетчика 3. Если устанавливается флаг OCF3A в регистре ETIFR, то программа переходит на соответствующий вектор прерывания (см. раздел "Прерывания").

Разряд 3 – OCIE3B: Разрешение прерывания по результату сравнения канала B таймера-счетчика 3. Аналогично предыдущему, но по отношению к каналу B.

Разряд 2 – TOIE3: Разрешение прерывания по переполнению таймера-счетчика. Если в данный бит записана лог. 1 и установлен флаг I в регистре статуса, то разрешается прерывание по переполнению таймера-счетчика 3. После этого, установка флага TOV3 в регистре ETIFR приведет к переходу на соответствующий вектор прерывания (см. раздел "Прерывания").

Разряд 1 – OCIE3C: Разрешение прерывания по результату сравнения канала C таймера-счетчика 3. Если данный бит равен 1 и установлен флаг I в регистре статуса, то разрешается работа прерывания по результату сравнения канала C таймера-счетчика 3. Если устанавливается флаг OCF3C в регистре ETIFR, то программа переходит на соответствующий вектор прерывания (см. раздел "Прерывания").

Разряд 0 – OCIE3C: Разрешение прерывания по результату сравнения канала C таймера-счетчика 1. Если данный бит равен 1 и установлен флаг I в регистре статуса, то разрешается работа прерывания по результату сравнения канала C таймера-счетчика 1. Если устанавливается флаг OCF1C в регистре

ETIFR, то программа переходит на соответствующий вектор прерывания (см. раздел «Прерывания»).

**Регистр флагов прерываний таймеров-счетчиков – TIFR**

Разряд	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOF0	<b>TIFR</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Прим.: Биты данного регистра относятся к нескольким таймерам, но в данном параграфе рассматриваются биты только одного таймера. Описание остальных бит необходимо смотреть в соответствующих разделах.

Разряд 5 – ICF1: Флаг захвата состояния таймера-счетчика 1. Флаг устанавливается, если на входе ICP1 определяется условие захвата. Если регистр захвата ICR1 выбран с помощью бит WGMn3:0 в качестве источника верхнего предела счета, флаг ICF1 устанавливается по достижении верхнего предела счета.

ICF1 автоматически сбрасывается при переходе на вектор прерывания по захвату состояния таймера-счетчика. Альтернативно флаг ICF1 можно сбрасывать путем записи в него лог. 1.

Разряд 4 – OCF1A: Флаг результата сравнения канала А таймера-счетчика 1. Данный флаг устанавливается следующим тактом после совпадения значения TCNT1 с регистром А порога сравнения (OCR1A).

Обратите внимание, что строб принудительной установки результата сравнения (FOC1A) не устанавливает флаг OCF1A. Флаг OCF1A автоматически сбрасывается при переходе на соответствующий вектор прерывания. Альтернативно, флаг OCF1A сбрасывается путем записи в него лог. 1.

Разряд 3 – OCF1B: Флаг результата сравнения канала В таймера-счетчика 1. Данный флаг действует аналогично предыдущему, но в отношении канала сравнения В.

Разряд 2 – TOV1: Флаг переполнения таймера-счетчика 1. Установка данного флага зависит от значений бит WGMn3:0. В нормальном режиме и режиме СТС флаг TOV1 устанавливается при переполнении таймера-счетчика. См. табл. 61 для изучения поведения флага TOV1 при задании других значений WGMn3:0. Флаг TOV1 автоматически сбрасывается при переходе на вектор прерывания по переполнению таймера-счетчика 1. Альтернативно флаг TOV1 сбрасывается путем записи в него лог. 1.

**Расширенный регистр флагов прерываний таймеров-счетчиков – ETIFR**

Разряд	7	6	5	4	3	2	1	0	
	-	-	ICF3	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	<b>ETIFR</b>
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряды 7:6 – Зарезервированные биты. Данные биты зарезервированы для дальнейшего использования. В целях совместимости с последующими вер-

сиями необходимо в данные разряды записывать лог. 0 во время записи в регистр ETIFR.

Разряд 5 – ICF3: Флаг захвата состояния таймера-счетчика 3. Флаг устанавливается, если на входе ICP3 определяется условие захвата. Если регистр захвата ICR3 выбран с помощью бит WGM3:0 в качестве источника верхнего предела счета, то флаг ICF3 устанавливается по достижении верхнего предела счета.

ICF3 автоматически сбрасывается при переходе на вектор прерывания по захвату состояния таймера-счетчика. Альтернативно флаг ICF3 можно сбросить путем записи в него лог. 1.

Разряд 4 – OCF3A: Флаг результата сравнения канала А таймера-счетчика 3. Данный флаг устанавливается следующим тактом после совпадения значения TCNT3 с регистром А порога сравнения (OCR3A).

Обратите внимание, что строб принудительной установки результата сравнения (FOC3A) не устанавливает флаг OCF3A. Флаг OCF3A автоматически сбрасывается при переходе на соответствующий вектор прерывания. Альтернативно, флаг OCF3A сбрасывается путем записи в него лог. 1.

Разряд 3 – OCF3B: Флаг результата сравнения канала В таймера-счетчика 3. Действует аналогично предыдущему, но в отношении канала В таймера-счетчика 3.

Разряд 2 – TOV3: Флаг переполнения таймера-счетчика 3. Установка данного флага зависит от значений бит WGM3:0. В нормальном режиме и режиме СТС флаг TOV3 устанавливается при переполнении таймера-счетчика. См. табл. 61 для изучения поведения флага TOV3 при задании других значений WGM3:0. Флаг TOV3 автоматически сбрасывается при переходе на вектор прерывания по переполнению таймера-счетчика 3. Альтернативно флаг TOV3 сбрасывается путем записи в него лог. 1.

Разряд 1 – OCF3C: Флаг результата сравнения канала С таймера-счетчика 3. Данный флаг устанавливается следующим тактом после совпадения значения TCNT3 с регистром С порога сравнения (OCR3C).

Обратите внимание, что строб принудительной установки результата сравнения (FOC3C) не устанавливает флаг OCF3C. Флаг OCF3C автоматически сбрасывается при переходе на соответствующий вектор прерывания. Альтернативно, флаг OCF3C сбрасывается путем записи в него лог. 1.

Разряд 0 – OCF1C: Флаг результата сравнения канала С таймера-счетчика 1. Действует аналогично предыдущему, но в отношении TCNT1 и канала С таймера 1.

#### **4.1.7. Системная синхронизация и тактовые источники**

На рис. 4.24 представлены источники синхронизации и распределение синхроимпульсов к блокам ATmega128. Не обязательно вся синхронизация должна работать в одно и тоже время. В целях снижения энергопотребления тактирование неиспользуемых модулей может быть прекращено путем перевода в различные режимы сна командой sleep (см. «Управление энергопотреблением и режимы сна»).

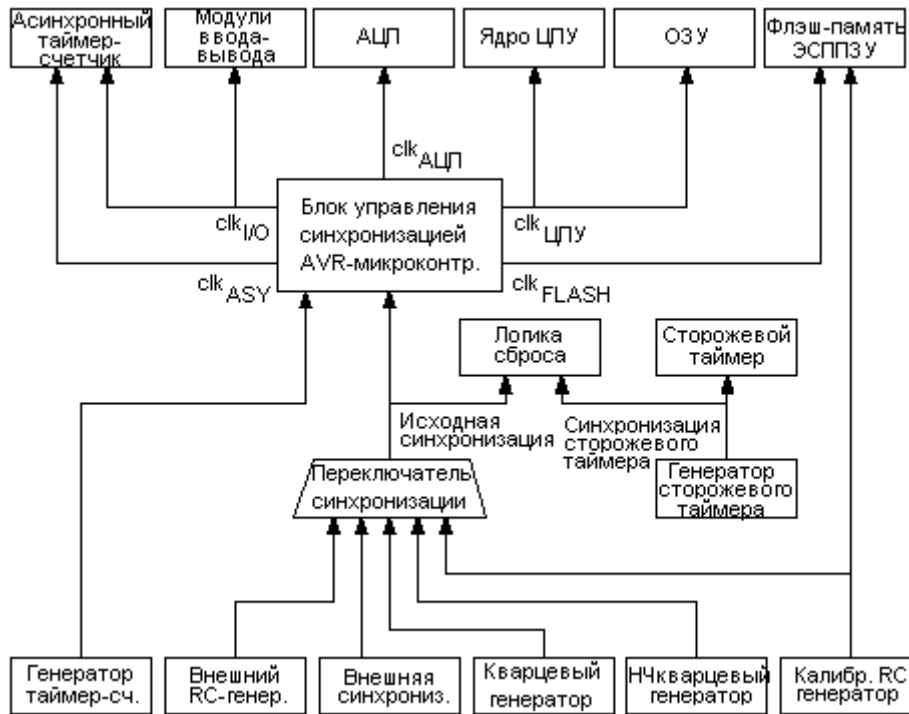


Рис. 4.24. Распределение синхронизирующих импульсов

**Синхронизация ЦПУ –  $\text{clkCPU}$ .** Синхронизация ЦПУ подключается к модулям микроконтроллера, которые связаны с работой ядра AVR. Примерами таких модулей являются файл регистров общего назначения, регистр статуса и память данных, выполняющая функцию стека. Остановка синхронизации ЦПУ приводит к прекращению выполнения ядром любых действий и вычислений.

**Синхронизация ввода-вывода –  $\text{clkI/O}$ .** Синхронизация ввода-вывода используется основными модулями ввода-вывода, в т. ч. таймеры-счетчики, SPI и УСАПП. Она также используется модулем внешних прерываний, но в некоторых случаях внешние прерывания детектируются в асинхронном режиме для поддержки работоспособности внешних прерываний даже при отключенной синхронизации. Также обратите внимание, что после отключения данной синхронизации (во всех режимах сна) двухпроводной интерфейс TWI продолжает наблюдать за передаваемым по шине адресом асинхронно.

**Синхронизация флэш-памяти –  $\text{clkFLASH}$ .** Синхронизация флэш-памяти тактирует работу интерфейса флэш-памяти. Обычно эта синхронизация работает одновременно с синхронизацией ЦПУ.

**Синхронизация асинхронного таймера –  $\text{clkASY}$ .** Синхронизация асинхронного таймера используется для тактирования асинхронного таймера-счетчика внешним кварцевым резонатором частотой 32 кГц. Данный тактовый генератор позволяет использовать таймер-счетчик как счетчик реального времени, даже при переводе микроконтроллера в режим сна.

**Синхронизация АЦП –  $\text{clkADC}$ .** АЦП тактируется обособленным блоком синхронизации. Это позволяет остановить работу синхронизации ЦПУ и ввода-вывода на время преобразования АЦП в целях снижения влияния цифро-



вого шума на результат преобразования. С помощью этого достигается более точный результат преобразования.

**Источники синхронизации.** С помощью конфигурационных бит имеется возможность выбора нескольких источников синхронизации. Сигнал синхронизации выбранного источника является входным для тактового генератора AVR и затем подключается к соответствующим модулям.

Таблица 4.35

Выбор опций синхронизации микроконтроллера

Источники синхронизации	CKSEL3..0(1)
Внешний кварцевый/керамический резонатор	1111 – 1010
Внешний низкочастотный кварцевый резонатор	1001
Внешний RC-генератор	1000 – 0101
Встроенный калиброванный RC-генератор	0100 – 0001
Внешняя синхронизация	0000

Прим.: Для всех конфигурационных бит «1» означает незапрограммированное состояние, а «0» – запрограммированное.

Подробное описание каждой из этих опций приведено в следующих разделах. При выходе ЦПУ из режима выключения (Power-down) или экономичного режима (Power-save) выбранный источник синхронизации используется по истечении времени на запуск, тем самым гарантируя стабильность работы генератора перед первым выполнением инструкции. Запуск микроконтроллера, инициированный сбросом (reset), сопровождается дополнительной задержкой для достижения питанием стабильного уровня перед переводом микроконтроллера в нормальный режим работы. Генератор сторожевого таймера используется для синхронизации данного модуля, который формирует задержку при запуске. Длительность генерируемой задержки определяется количеством импульсов генератора сторожевого таймера и для различных случаев приведена в таблице 4.36. Частота генератора сторожевого таймера зависит от напряжения питания, что показано в разделе «Типовые характеристики ATmega128: предварительные данные».

Таблица 4.36

Количество тактов сторожевого таймера

Типичное время переполнения (VCC = 5.0В)	Типичное время переполнения (VCC = 3.0В)	Количество тактов
4.1 мс	4.3 мс	4K (4096)
65 мс	69 мс	64K (65536)

**Первоначальный источник синхронизации.** Микроконтроллер поставляется с установками CKSEL = «0001» и SUT = «10». Эти значения соответствуют выбору в качестве источника синхронизации внутреннего RC-генератора с максимальным временем старта. Данная настройка гарантирует всем пользователям возможность установить требуемый источник синхронизации с помощью внутрисистемного или параллельного программатора.

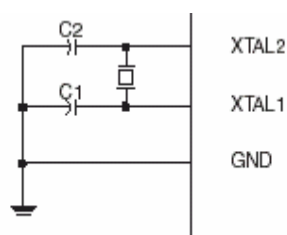


Рис. 4.25.

Кварцевый генератор

**Кварцевый генератор.** XTAL1 и XTAL2 – вход и выход инвертирующего усилителя, соответственно, который может быть использован в качестве встроенного генератора (см. рисунок 4.25). Для задания частоты может использоваться либо кварцевый, либо керамический резонатор. Конфигурационный бит СКОРТ выбирает один из двух режимов усилителя генератора. Если СКОРТ запрограммирован, то амплитуда колебаний выходного сигнала генератора будет ограничена уровнями питания.

Данный режим рекомендуется использовать при высоком уровне окружающих шумов или при использовании выхода XTAL2 в качестве источника синхронизации внешней схемы. Данный режим характеризуется широким частотным диапазоном. Если СКОРТ не запрограммирован, то амплитуда выходных колебаний генератора снижается. Использование данного режима позволяет существенно снизить потребляемую мощность, но при этом ограничен частотный диапазон и нельзя XTAL2 использовать для внешней синхронизации.

При использовании резонаторов максимальная частота равна 8 МГц, если СКОРТ не запрограммирован, и 16 МГц, если СКОРТ запрограммирован. C1 и C2 должны быть всегда равны независимо от использования кварцевого или керамического резонатора. Оптимальное значение емкостей конденсаторов зависит от используемого резонатора и от значения паразитной емкости. Рекомендации по выбору номиналов конденсаторов приведены в таблице 4.32. Для керамических резонаторов необходимо использовать конденсаторы с номиналом, рекомендуемым производителем.

Таблица 4.32

## Рабочие режимы кварцевого генератора

СКОРТ	CKSEL3..1	Частотный диапазон(1), МГц	Рекомендуемый диапазон номиналов C1 и C2 при использовании кварцевого резонатора
1	101	0.4-0.9	-
1	110	0.9-3.0	12пФ-22пФ
1	111	3.0-8.0	12пФ-22пФ
0	101, 110, 111	1.0-	12пФ-22пФ

Конфигурационные биты CKSEL0 совместно с битами SUT1..0 задают время старта в соответствии с таблицей 4.33.

Таблица 4.33

## Времена задержек при запуске для различных настроек кварцевого генератора

CKSEL0	SUT1..0	Длительность задержки при выходе из режима выключения и экономичного режима	Дополнительная задержка после сброса (VCC= 5.0В)	Рекомендуемая область применения
0	00	258 CK(1)	4.1 мс	Керамический резонатор, быстро нарастающее питание
0	01	258 CK(1)	65 мс	Керамический резонатор, медленно нарастающее питание

Окончание таблицы 4.33

0	10	1K СК(2)	—	Керамический резонатор, детектор питания (BOD) включен
0	11	1K СК(2)	4.1 мс	Керамический резонатор, быстро нарастающее питание
1	00	1K СК(2)	65 мс	Керамический резонатор, медленно нарастающее питание
1	01	16K СК	—	Кварцевый генератор, детектор питания (BOD) включен
1	10	16K СК	4.1 мс	Кварцевый резонатор, быстро нарастающее питание
1	11	16K СК	65 мс	Кварцевый резонатор, медленно нарастающее питание

Прим.:

1. Данные опции допускается использовать при работе микроконтроллера на частотах ниже максимальной рабочей. Данные опции не приемлемы при использовании кварцевых резонаторов.

2. Данные опции реализованы для использования керамических резонаторов и гарантируют стабильность частоты после запуска. При данных установках допускается использовать кварцевый резонатор, но при условии, что рабочая частота микроконтроллера меньше максимальной.

**Низкочастотный кварцевый генератор.** Для использования часового кварцевого резонатора 32.768 кГц в качестве источника синхронизации необходимо выбрать низкочастотный кварцевый генератор путем установки конфигурационных бит CKSEL равными «1001». Путем программирования конфигурационного бита пользователь может разрешить подключение встроенных конденсаторов к выводам XTAL1 и XTAL2, тем самым исключая необходимость применения внешних конденсаторов. Внутренние конденсаторы имеют номинал 36 пФ. После выбора данного генератора, длительности задержек при старте определяются конфигурационными битами SUT, как показано в таблице 4.34.

Таблица 4.34

Длительности задержек при старте для низкочастотного кварцевого резонатора

SUT1..0	Длительность задержки при выходе из режима выключения и экономичного режима	Дополнительная задержка после сброса (VCC=5.0В)	Рекомендуемая область применения
00	1K СК(1)	4.1 мс	Быстро нарастающее питание или включен детектор питания BOD
01	1K СК(1)	65 мс	Медленно нарастающее питание
10	32K СК	65 мс	Стабильная частота при старте
11	Зарезервировано		

Примечание: Данные опции необходимо использовать, если стабильность частоты при старте не важна для приложения.

**Внешний RC-генератор.** Для приложений некритичных к стабильности временных характеристик в качестве источника синхронизации может использоваться внешняя RC-цепь, подключение которой показано на рисунке 4.26.

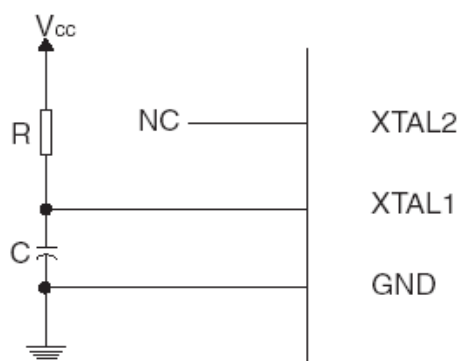


Рис. 4.26. RC-генератор

Тактовая частота грубо определяется выражением  $f = 1/(3RC)$ . Номинал конденсатора  $C$  должен быть не менее 22 пФ. Путем программирования конфигурационного бита CKOPT пользователь может разрешить подключение внутреннего конденсатора 36 пФ между XTAL1 и GND, тем самым исключая необходимость применения внешнего

конденсатора. Более подробная информация о работе генератора и о выборе номиналов  $R$  и  $C$  приведена в рекомендациях по применению внешнего RC-генератора. Генератор может работать в четырех различных режимах, каждый из которых ориентирован на специфический частотный диапазон. Рабочий режим выбирается конфигурационными битами CKSEL3..0 (см. табл. 4.35).

Таблица 4.35

Рабочие режимы внешнего RC-генератора

CKSEL3..0	Частотный диапазон, МГц
0101	- 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

После разрешения работы данного генератора длительность задержки при старте определяется установками конфигурационных бит (см. табл. 4.36).

Таблица 4.36

Длительность задержек при старте после выбора внешнего RC-генератора

SUT1..0	Длительность задержки при выходе из режима выключения и экономичного режима	Дополнительная задержка после сброса (VCC= 5.0V)	Рекомендуемая область применения
00	18 CK(1)	-	Включен детектор питания BOD
01	18 CK	4.1 мс	Быстро нарастающее питание
10	18 CK	65 мс	Медленно нарастающее питание
11	6 CK (1)	4.1 мс	Быстро нарастающее питание или включенный детектор питания BOD

Примечание: Данная опция не должна использоваться на тактовых частотах близких к максимальной.

**Встроенный калиброванный RC-генератор.** Встроенный калиброванный RC-генератор формирует фиксированные тактовые частоты 1.0, 2.0, 4.0 или 8.0 МГц. Выбор тактовой частоты осуществляется установкой конфигурационных бит CKSEL в соответствии с таблицей 4.37. После выбора микроконтроллер будет работать без внешних компонентов. Конфигурационный бит СКОРТ должен быть всегда незапрограммированным, если используется внутренний RC-генератор. В процессе сброса калибровочный байт аппаратно записывается регистр OSCCAL, тем самым автоматически выполняя калибровку RC-генератора. При питании 5В, температуре 25°C и выбранной частоте генератора 1.0 МГц данный метод калибровки обеспечивает погрешность генерации частоты не хуже  $\pm 3\%$  от номинального значения. При использовании данного генератора в качестве тактового генератора он также будет использоваться для тактирования сторожевого таймера и для задания длительности задержки при сбросе. Более подробная информация о предварительно запрограммированном калибровочном значении приведена в разделе «Калибровочный байт».

Таблица 4.37

Режимы встроенного калиброванного RC-генератора

CKSEL3..0	Номинальная частота, МГц
0001(1)	1.0
0010	2.0
0011	4.0
0100	8.0

Примечание: Микроконтроллер поставляется с данной установкой.

После выбора данного генератора длительность задержки при запуске микроконтроллера определяется установками конфигурационных бит SUT (см. табл. 4.38). Выводы XTAL1 и XTAL2 должны быть оставлены неподключенными (NC).

Таблица 4.38

Длительности задержек при запуске с различными настройками  
встроенного калиброванного RC-генератора

SUT1..0	Длительность задержки при выходе из режима выключения и экономичного режима	Дополнительная задержка после сброса (VCC= 5.0В)	Рекомендуемые условия для применения
00	6 СК	-	Включен детектор питания BOD
01	6 СК	4.1 мс	Быстро нарастающее питание
10(1)	6 СК	65 мс	Медленно нарастающее питание
11	Зарезервировано		

Примечание: Микроконтроллер поставляется с данной установкой.

Регистр калибровки генератора – OSCCAL								
Разряд	7	6	5	4	3	2	1	0
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.
Начальное значение	Калибровочное значение индивидуально для каждого микроконтроллера							

Разряды 7..0 – CAL7..0: Калибровочное значение для генератора

Запись значения калибровочного байта в данный регистр приведет к настройке генератора на номинальную частоту. В процессе сброса калибровочное значение для частоты 1МГц автоматически записывается в регистр OSCCAL. Если встроенный RC-генератор используется на других частотах, то калибровочный байт необходимо записывать программно. Для этого необходимо с помощью программатора считать значение калибровочного байта, затем сохранить его значение во флэш-память или ЭСППЗУ. После этого калибровочное значение может быть считано программно, а затем записано в регистр OSCCAL. Если в регистр OSCCAL записать ноль, то выбирается минимальная частота. Запись ненулевого значения приводит к повышению частоты генератора. Запись \$FF – к выбору максимальной частоты. Калиброванный генератор используется для синхронизации доступа к ЭСППЗУ и флэш-памяти. Во время выполнения записи в ЭСППЗУ или во флэш-память не следует выполнять калибровку на частоту выше на 10% от номинальной. В противном случае, запись в ЭСППЗУ или во флэш-память может быть некорректной. Результат подстройки при записи различных значений калибровочного байта приведен в таблице 4.40.

Таблица 4.40

Диапазон частот встроенного RC-генератора

Значение OSCCAL	Минимальная частота в процентах от номинальной, %	Максимальная частота в процентах от номинальной, %
\$00	50	100
\$7F	75	150
\$FF	100	200

**Внешняя синхронизация.** Если необходимо тактировать микроконтроллер от внешнего источника, то его необходимо подключить к выводу XTAL1 (см. рис. 4.27). В этом случае внешняя синхронизация должна быть разрешена записью в конфигурационные биты CKSEL значения «0000». Если запрограммировать конфигурационный бит CKOPT, то между XTAL1 и GND будет подключен внутренний конденсатор номиналом 36 пФ.

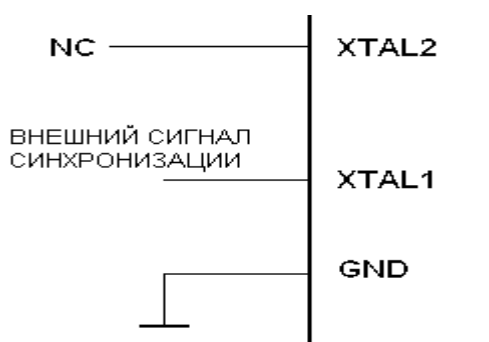


Рис. 4.27. Схема подключения внешнего источника синхронизации

После выбора данного источника синхронизации длительность задержки при запуске определяется битами SUT как показано в таблице 4.41.

Таблица 4.41

Длительность задержки при запуске при выборе внешней синхронизации

SUT1..0	Длительность задержки при выходе из режима выключения и экономичного режима	Дополнительная задержка после сброса (VCC= 5.0В)	Рекомендуемые условия для применения
00	6 СК	-	Включен детектор питания BOD
01	6 СК	4.1 мс	Быстро нарастающее питание
10(1)	6 СК	65 мс	Медленно нарастающее питание
11	Зарезервировано		

После подключения внешнего тактового источника необходимо избегать внезапных изменений его частоты для гарантирования стабильности работы микроконтроллера. Если на следующем такте частота изменится более чем на 2% по сравнению с предыдущим, то поведение микроконтроллера может стать непредсказуемым. Данный механизм реализован для гарантирования нахождения микроконтроллера в состоянии сброса в процессе таких изменений тактовой частоты.

**Генератор таймер-счетчика.** Выводы генератора таймера-счетчика TOSC1 и TOSC2 предназначены для непосредственного подключения кварцевого резонатора. В этом случае не требуются внешние конденсаторы. Генератор оптимизирован для совместной работы с часовым кварцевым резонатором 32.768 кГц. Подключение внешнего тактового источника к выводу TOSC1 не рекомендуется.

**Регистр управления делением XTAL – XDIV.** Регистр управления делением XTAL используется для деления частоты тактового источника на одно из значений в диапазоне 2-129. Данная функция может использоваться при необходимости оптимизации энергопотребления.

Разряд	7	6	5	4	3	2	1	0
	<b>XDIVEN</b>	<b>XDIV6</b>	<b>XDIV5</b>	<b>XDIV4</b>	<b>XDIV3</b>	<b>XDIV2</b>	<b>XDIV1</b>	<b>XDIV0</b>
Чтение/Запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.
Начальное значение	0	0	0	0	0	0	0	0

Разряд 7 – XDIVEN: Разрешение деления XTAL

Если в XDIVEN записать лог.1, то тактовые частоты ЦПУ и периферийных модулей (clkI/O, clkADC, clkCPU, clkFLASH) будут поделены на коэффициент, заданный значениями XDIV6 - XDIV0. Данный бит можно программировать в работающем приложении для варьирования тактовой частотой.

Разряды 6..0 – XDIV6..XDIV0: Разряды 6..0 коэффициента деления XTAL. Данные разряды определяют значение коэффициента деления, который вступает в силу после записи лог. 1 в XDIVEN. Если значение данных бит обо-

значит как  $d$ , то результирующая тактовая частота ( $f_{CLK}$ ) ЦПУ и периферийных модулей может быть найдена по выражению:

$$f_{clk} = \frac{\text{Частота тактового источника}}{129 - d}.$$

Изменять значение данных разрядов допускается, только если  $XDIVEN=0$ . Когда в  $XDIVEN$  записывается лог.1, то записываемые одновременно с ней значения  $XDIV6..XDIV0$  будут использоваться как коэффициент деления. Во время записи в  $XDIVEN$  лог. 0 одновременно записываемые значения в разряды  $XDIV6..XDIV0$  отклоняются. Поскольку, делитель выполняет снижение входной тактовой частоты микроконтроллера, то после разрешения деления также снижается быстродействие всех периферийных модулей.

Примечание: После разрешения деления тактовой частоты таймер-счетчик 0 может быть использован только в асинхронном режиме. Частота асинхронного источника должна быть не менее чем в 4 раза меньше результирующей (поделенной) частоты синхронизации. В противном случае не гарантируется определение запроса на прерывание и корректность доступа к регистрам таймера-счетчика 0.

#### 4.1.8. Прерывания

В данном разделе описывается специфика обработки прерываний, реализованная в ATmega128. Общее описание обработки прерываний приведено в разделе «Сброс и обработка прерываний».

Таблица 4.50

Векторы сброса и прерываний

№ вектора	Адрес памяти программ(4)	Источник	Условие возникновения прерывания
1	\$0000(1)	RESET	Внешний сброс, сброс при подаче питания, сброс при недопустимом снижении питания, сброс сторожевым таймером и сброс через JTAG-интерфейс
2	\$0002	INT0	Запрос на внешнее прерывание 0
3	\$0004	INT1	Запрос на внешнее прерывание 1
4	\$0006	INT2	Запрос на внешнее прерывание 2
5	\$0008	INT3	Запрос на внешнее прерывание 3
6	\$000A	INT4	Запрос на внешнее прерывание 4
7	\$000C	INT5	Запрос на внешнее прерывание 5
8	\$000E	INT6	Запрос на внешнее прерывание 6
9	\$0010	INT7	Запрос на внешнее прерывание 7
10	\$0012	TIMER2 COMP	Срабатывание компаратора таймера-счетчика 2
11	\$0014	TIMER2 OVF	Переполнение таймера-счетчика 2
12	\$0016	TIMER1 CAPT	Захват фронта таймером-счетчиком 1



Окончание табл. 4.50

13	\$0018	TIMER1 COMPA	Срабатывание компаратора А таймера-счетчика 1
14	\$001A	TIMER1 COMPB	Срабатывание компаратора В таймера-счетчика 1
15	\$001C	TIMER1 OVF	Переполнение таймера-счетчика 1
16	\$001E	TIMER0 COMP	Срабатывание компаратора таймера-счетчика 0
17	\$0020	TIMER0 OVF	Переполнение таймера-счетчика 0
18	\$0022	SPI, STC	Завершение последовательной передачи интерфейсом SPI
19	\$0024	USART0, RX	Завершение приема УСАПП 0
20	\$0026	USART0, UDRE	Регистр данных УСАПП0 свободен
21	\$0028	USART0, TX	Завершение передачи УСАПП 0
22	\$002A	ADC	Завершение преобразования АЦП
23	\$002C	EE READY	Готовность ЭСППЗУ
24	\$002E	ANALOG COMP	Аналоговый компаратор
25	\$0030(3)	TIMER1 COMPC	Срабатывание компаратора С таймера-счетчика 1
26	\$0032(3)	TIMER3 CAPT	Захват фронта таймером счетчиком 3
27	\$0034(3)	TIMER3 COMPA	Срабатывание компаратора А таймера-счетчика 3
28	\$0036(3)	TIMER3 COMPB	Срабатывание компаратора В таймера-счетчика 3
29	\$0038(3)	TIMER3 COMPC	Срабатывание компаратора С таймера-счетчика 3
30	\$003A(3)	TIMER3 OVF	Переполнение таймера счетчика 3
31	\$003C(3)	USART1, RX	Завершение приема УСАПП 1
32	\$003E(3)	USART1, UDRE	Регистр данных УСАПП1 свободен
33	\$0040(3)	USART1, TX	Завершение передачи УСАПП1
34	\$0042(3)	TWI	Двухпроводной последовательный интерфейс
35	\$0044(3)	SPM READY	Готовность записи в память программ

Прим.:

1. Если конфигурационный бит BOOTRST запрограммирован, то микроконтроллер выполняет переход на адрес сброса в загрузочном секторе, см. «Само-программирование из сектора начальной загрузки с поддержкой чтения во время записи».
2. Если установлен бит IVSEL в регистре MCUCR, то векторы прерываний перемещаются в начало загрузочного сектора флэш-памяти. В этом случае к адресу каждого вектора прерывания из таблицы прибавляется стартовый адрес загрузочного сектора флэш-памяти.
3. Прерывания по адресам \$0030 - \$0044 не существуют в режиме совместимости с ATmega103.

В таблице 4.51 показано расположение векторов сброса и прерываний в зависимости от различных установок BOOTRST и IVSEL. Если программа не использует прерывания, то она может быть размещена равномерно, используя ячейки с адресами векторов прерываний для хранения программного кода. Возможен также случай, когда вектор сброса располагается в секторе прикладной программы, а векторы прерываний – в загрузочном секторе или наоборот.

Таблица 4.51

Размещение векторов сброса и прерываний

BOOTRST	IVSEL	Адрес сброса	Начальный адрес векторов прерываний
1	0	\$0000	\$0002
1	1	\$0000	Адрес сброса в загрузочном секторе + \$0002
0	0	Адрес сброса в загрузочном секторе	\$0002
0	1	Адрес сброса в загрузочном секторе	Адрес сброса в загрузочном секторе + \$0002

Прим.: Для конфигурационного бита BOOTRST «1» означает незапрограммированное состояние, «0» - запрограммированное.

Ниже приведены типичные и общие программные установки адресов сброса и векторов прерываний у ATmega128:

Адрес	Инструкция	Комментарий
\$0000	jmp RESET	; Переход на обработку сброса
\$0002	jmp EXT_INT0	; Переход на обработку запроса IRQ0
\$0004	jmp EXT_INT1	; Переход на обработку запроса IRQ1
\$0006	jmp EXT_INT2	; Переход на обработку запроса IRQ2
\$0008	jmp EXT_INT3	; Переход на обработку запроса IRQ3
\$000A	jmp EXT_INT4	; Переход на обработку запроса IRQ4
\$000C	jmp EXT_INT5	; Переход на обработку запроса IRQ5
\$000E	jmp EXT_INT6	; Переход на обработку запроса IRQ6
\$0010	jmp EXT_INT7	; Переход на обработку запроса IRQ7
\$0012	jmp TIM2_COMP	; Переход на обработку при выполнении условия сравнения таймера 2
\$0014	jmp TIM2_OVF	; Переход на обработку при переполнении таймера 2
\$0016	jmp TIM1_CAPT	; Переход на обработку при захвате фронта таймером 1
\$0018	jmp TIM1_COMPA	; Переход на обработку при срабатывании компаратора А таймера 1
\$001A	jmp TIM1_COMPB	; Переход на обработку при срабатывании компаратора В таймера 1

```

$001C  jmp TIM1_OVF    ; Переход на обработку при переполнении таймера 1
$001E  jmp TIM0_COMP   ; Переход на обработку при выполнении условия сравнения тай-
мера 0
$0020  jmp TIM0_OVF    ; Переход на обработку при переполнении таймера 0
$0022  jmp SPI_STC     ; Переход на обработку при завершении передачи SPI
$0024  jmp USART0_RXC  ; Переход на обработку при завершении приема УСАПП0
$0026  jmp USART0_DRE  ; Переход на обработку при освобождении регистра данных UDR
УСАПП0
$0028  jmp USART0_TXC  ; Переход на обработку при завершении передачи УСАПП0
$002A  jmp ADC         ; Переход на обработку при завершении преобразования АЦП
$002C  jmp EE_RDY      ; Переход на обработку при готовности ЭСППЗУ
$002E  jmp ANA_COMP    ; Переход на обработку при срабатывании аналогового компара-
тора
$0030  jmp TIM1_COMPC  ; Переход на обработку при срабатывании компаратора С тайме-
ра 1
$0032  jmp TIM3_CAPT   ; Переход на обработку при захвате фронта таймером 3
$0034  jmp TIM3_COMPA  ; Переход на обработку при срабатывании компаратора А тайме-
ра 3
$0036  jmp TIM3_COMPB  ; Переход на обработку при срабатывании компаратора В тайме-
ра 3
$0038  jmp TIM3_COMPC  ; Переход на обработку при срабатывании компаратора С тайме-
ра 3
$003A  jmp TIM3_OVF    ; Переход на обработку при переполнении таймера 3
$003C  jmp USART1_RXC  ; Переход на обработку по завершении приема УСАПП1
$003E  jmp USART1_DRE  ; Переход на обработку при освобождении регистра данных UDR
УСАПП1
$0040  jmp USART1_TXC  ; Переход на обработку при завершении передачи УСАПП1
$0042  jmp TWI         ; Переход на обработку прерывания по двухпроводному последова-
тельному интерфейсу
$0044  jmp SPM_RDY     ; Переход на обработку прерывания при готовности выполнения
команды SPM
;
$0046  RESET:ldir16, high(RAMEND); Начало основной программы
$0047  out SPH,r16     ; Установка указателя стека в конце ОЗУ
$0048  ldi r16, low(RAMEND)
$0049  out SPL,r16
$004A  sei ; Разрешение прерываний
$004B  xxx
.....

```

Если конфигурационный бит BOOTRST незапрограммирован, размер загрузочного сектора установлен 8 кбайт и бит IVSEL установлен в регистре MCUCR перед разрешением любого прерывания, то можно использовать следующий пример распределения программы по адресам векторов сброса и прерываний.

Адрес	Инструкция	Комментарий
\$0000	RESET:ldi r16,high(RAMEND)	; Начало основной программы
\$0001	out SPH,r16	; Установка указателя стека в конце ОЗУ
\$0002	ldi r16,low(RAMEND)	
\$0003	out SPL,r16	
\$0004	sei	; Разрешение прерываний

```

$0005 xxx
;
.org $F002
$F002 jmp EXT_INT0 ; Переход на обработку прерывания IRQ0
$F004 jmp EXT_INT1 ; Переход на обработку прерывания IRQ1
... .. ;
$F044 jmp SPM_RDY ; Переход на обработку прерывания по готовности к записи в па-
мять программ

```

Если конфигурационный бит BOOTRST запрограммирован и установлен размер загрузочного сектора 8 кбайт, то можно использовать следующий шаблон программы:

Адрес	Инструкция	Комментарий
.org \$0002		
\$0002	jmp EXT_INT0	; Переход на обработку прерывания IRQ0
\$0004	jmp EXT_INT1	; Переход на обработку прерывания IRQ1
... .. ;		
\$0044	jmp SPM_RDY	; Переход на обработку прерывания по готовности к записи в память программ
;		
.org \$F000		
\$F000	RESET: ldi r16,high(RAMEND)	; Начало основной программы
\$F001	out SPH,r16	; Установка указателя стека в конец ОЗУ
\$F002	ldi r16,low(RAMEND)	
\$F003	out SPL,r16	
\$F004	sei	; Разрешение прерываний
\$F005	xxx	

Если конфигурационный бит BOOTRST запрограммирован, размер загрузочного сектора установлен 8 кбайт и бит IVSEL в регистре MCUCR установлен перед разрешением любого из прерываний, то распределение адресов в программе следующее:

Адрес	Инструкция	Комментарий
;		
.org \$F000		
\$F000	jmp RESET	; Переход на обработку сброса
\$F002	jmp EXT_INT0	; Переход на обработку прерывания IRQ0
\$F004	jmp EXT_INT1	; Переход на обработку прерывания IRQ1
... .. ;		
\$F044	jmp SPM_RDY	; Переход на обработку прерывания по готовности записи в память программ
\$F046	RESET: ldi r16,high(RAMEND)	; Начало основной программы
\$F047	out SPH,r16	; Установка указателя стека в конец ОЗУ
\$F048	ldi r16,low(RAMEND)	
\$F049	out SPL,r16	
\$F04A	sei	; Разрешение прерываний
\$F04B	xxx	

### Перемещение между секторами загрузочной и прикладной программы.

Общий регистр управления прерываниями задает размещение таблицы векторов прерываний.

**Регистр управления микроконтроллером – MCUCR**

Разряд	7	6	5	4	3	2	1	0	
	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	MCUCR
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

**Разряд 1 – IVSEL:** Выбор вектора прерывания

Если бит IVSEL сброшен (=0), то векторы прерываний размещаются в начале флэш-памяти. Если данный бит установлен (=1), то векторы прерываний перемещаются в начало загрузочного сектора флэш-памяти. Фактический адрес начала загрузочного сектора определяется значением конфигурационных бит BOOTSZ. Во избежание несанкционированных изменений таблицы векторов прерываний необходимо выполнить специальную последовательность записи при изменении бита IVSEL:

1. Записать лог. 1 в бит разрешения изменения вектора прерывания (IVCE).
2. В течение четырех машинных циклов записать желаемое значение в IVSEL, при этом записывая лог. 0 в IVCE.

Прерывания будут автоматически отключены при выполнении такой последовательности. Прерывания отключаются во время установки IVCE и останутся отключенными до перехода к инструкции следующей за инструкцией записи в IVSEL. Если IVSEL не записан, то прерывания будут находиться в отключенном состоянии 4 такта синхронизации. Состояние бита I в регистре статуса не затрагивается при автоматическом отключении прерываний.

Прим.: Если векторы прерываний помещаются в загрузочный сектор и бит защиты загрузочного сектора BLB02 запрограммирован, то прерывания будут отключены при выполнении программы с секторе прикладной программы. Если векторы прерываний размещены в прикладном секторе и бит защиты BLB12 запрограммирован, то прерывания становятся отключенными при выполнении программы в загрузочном секторе.

**Разряд 0 – IVCE:** Разрешение изменения вектора прерывания

В бит IVCE должна быть записана лог. 1, чтобы разрешить изменение бита IVSEL. IVCE сбрасывается аппаратно через четыре машинных цикла после записи лог. 1 в IVSEL. Установка бита IVCE приведет к отключению прерываний, что описано при рассмотрении бита IVSEL выше. Ниже приведен пример кода.

Пример кода на Ассемблере

```
Move_interrupts:
; Разрешение изменения векторов прерываний
ldi r16, (1<<IVCE)
```





Разряды 7..0 – ISC71, ISC70 - ISC41, ISC40: Бита выбора условия генерации внешнего прерывания 7-4

Внешние прерывания 7-4 активизируются через внешние выводы INT7:4, если установлены флаг I в регистре статуса SREG и соответствующая маска прерывания в регистре EIMSK. Условие, по которому генерируется прерывание, выбирается исходя из данных таблицы 4.54. Для определения фронтов на выводах INT7:4 осуществляется выборка их состояний. Если выбрано прерывание по фронту или изменению уровня, то прерывание будет сгенерировано, если на входе появляется импульс, длительность которого больше одного периода синхронизации. При действии на входе более коротких импульсов генерация прерывания не гарантируется. Если выбрано прерывание по низкому уровню, то прерывание генерируется, если до момента окончания выполнения текущей инструкции на входе по прежнему присутствует низкий уровень. Если разрешено прерывание по уровню, то оно будет генерироваться непрерывно до тех пор, пока на входе присутствует низкий уровень.

Таблица 4.54

Задание условия генерации запроса на прерывание (1)

ISCn1	ISCn0	Описание
0	0	Низкий уровень на INTn генерирует запрос на прерывание
0	1	Любое изменение логического состояния на INTn генерирует запрос на прерывание
1	0	Падающий фронт, выявленный по двум выборкам на INTn, генерирует запрос на прерывание.
1	1	Нарастающий фронт, выявленный по двум выборкам на INTn, генерирует запрос на прерывание.

Прим. 1: n = 7, 6, 5 или 4.

Перед изменением бит ISCn1/ISCn0 необходимо запретить работу прерывания путем очистки бита разрешения прерывания в регистре EIMSK. В противном случае прерывание может возникнуть после изменения данных бит.

Регистр маски внешнего прерывания – EIMSK

Разряд	7	6	5	4	3	2	1	0	
	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряды 7..0 – INT7 – INT0: Разрешение запроса на внешнее прерывание 7 – 0.

Если в бит INT7 – INT0 и в бит I регистра статуса SREG записать лог. 1, то разрешается работа внешнего прерывания по соответствующему выводу. Биты выбора условия генерации прерывания в регистрах управления внешними прерываниями EICRA и EICRB определяют по какому условию генерируется прерывание: по нарастающему фронту, по падающему фронту или по уровню. Любой из данных выводов сохраняет активность, даже если он настроен на вы-



вод. Данная особенность может использоваться для программной генерации прерывания.

Разряд	7	6	5	4	3	2	1	0	
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	EIFR
Чтение/запись	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	Чт./Зп.	
Исх. значение	0	0	0	0	0	0	0	0	

Разряды 7..0 – INTF7 - INTF0: Флаги внешних прерываний 7 - 0

Если фронт или изменение логического состояния на INT7:0 вызывает прерывание, то соответствующий флаг INTF7:0 принимает единичное состояние. Если установлены бит I регистра статуса SREG и соответствующий бит разрешения прерывания INT7:0 в регистре EIMSK, то микроконтроллер выполнит переход на вектор прерывания. Флаг сбрасывается аппаратно после выполнения процедуры обработки прерывания. Альтернативно флаг может быть сброшен программно путем записи лог. 1 в соответствующий бит. Если INT7:0 настроены на генерацию прерывания по уровню, то флаги постоянно находятся в сброшенном состоянии.

Примечание: Детальное описание работы интерфейса SPI, UART, TWI, аналогового компаратора приведено в [17, 18].

## 5. ОБЩИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

Сроки выполнения курсовой работы рассчитаны на один семестр, в течение которого студент должен, используя предшествующие знания в предметной области и специфические особенности курса «Электроприводы с системами числового управления», осуществить разработку цифрового электропривода (ЭП), как блока устройств/систем числового программного управления (УЧПУ/СЧПУ). Учитывая, многообразие задач, возникающих при разработке подобных систем, рекомендуется соблюдать следующую последовательность разработки:

- 1) освоить идеологию и архитектурные особенности числовых систем управления;
- 2) связать вопросы структурного построения системы регулирования с особенностями интерфейса УЧПУ и обработки информации в МП системах;
- 3) распределить решение задачи управления на аппаратную и программную части;
- 4) выбрать микроконтроллер (МК) и подробно ознакомиться с его интерфейсными устройствами;
- 5) сопоставить функциональные возможности внутреннего интерфейса МК с поставленными требованиями к системе управления ЭП и при необходимости дополнить его внешними интегральными схемами.

При разработке устройства сопряжения ЭП с центральным процессором (ЦП) СЧПУ в свою очередь целесообразно:

- 1) внимательно ознакомиться с заданием и при необходимости произвести доработку каналов связи центрального процессора (ЦП) и привода, уточнить функциональную схему устройств сопряжения МК с УЧПУ и с силовым преобразователем;
- 2) выделить функциональные элементы и ознакомиться с особенностями их схемотехники по справочной и учебной литературе;
- 3) скомпоновать схему в целом, обратив особое внимание на синхронизацию работы взаимосвязанных элементов и уровни согласования выходных и входных сигналов микросхем разных серий.

Разработка системы в целом требует детальной проработки всех вопросов: схемотехнических, алгоритмических, программного обеспечения. В курсовой работе подобная задача не ставится, однако, отдельные элементы программного обеспечения в части цифровых регуляторов должны быть проработаны полностью.

По согласованию с преподавателем возможны варианты разработки, предполагающие комплексную архитектурную проработку СЧПУ, включающую базовое программное обеспечение, протоколы обмена, среды визуальной подготовки управляющих программ, настройки и диагностики. В этом случае, допускается распределение задач между членами бригады и коллективная защита курсовой работы.

Сроки выполнения работы и ее успешная защита могут быть обеспечены лишь благодаря самостоятельной систематической работе студента. Контроль над выполнением работы возложен на преподавателя. Поэтому каждый студент не реже одного раза в неделю должен обращаться к преподавателю для консультации по конкретным вопросам и проверки законченных этапов работы. После защиты осуществляется дифференцированная оценка работы, которая кроме качества работы и знаний студента учитывает своевременность выполнения установленных сроков проектирования.

### 5.1. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Тема: Разработка цифрового ЭП на базе МК для устройств числового программного управления.

Исходные данные: определяются вариантом задания.

Вариант системы регулирования ЭП задается преподавателем в соответствии с уточняющими пунктами. При этом детализация разработки осуществляется с учетом выбранного микроконтроллера, каналов связи, шин сопряжения и интерфейсных БИС.

Содержание курсовой работы:

1) описание кинематики и рабочих органов станка, выбор двигателя по каталожным данным и определение его параметров;

2) разработка структурной схемы системы управления, математическое моделирование, проверка влияния дискретности регуляторов и их разрядности на качественные показатели системы управления;

3) разработка функциональной схемы блока управления ЭП как блока УЧПУ, описание принципа работы, в том числе интерфейса сопряжения и протоколов обмена информацией МК с центральной микроЭВМ;

4) разработка принципиальной схемы блока управления, выбор элементной базы, описание взаимодействия функциональных узлов схемы: МК, интерфейсных модулей МК и внешних интерфейсных схем, интеллектуальных драйверов управления силовыми преобразователями;

5) разработка алгоритмов работы и программного обеспечения.

Таблица 5.1

Вариант станка и привода

№	Станок	Привод	Источник информации	
0	Токарный	Поперечной подачи	Рис. 2.1	Конструкция и наладка станков с программным управлением и роботизированных комплексов: учеб. пособие / Л. Н. Грачев, В. Д. Косовский, А. Н. Ковшов и др. – М.: Высш.шк., 1989. – 271 с.
1		Продольной подачи		
2		Шпинделя		
3	Фрезерный	Привод перемещения фрезерной бабки	Рис. 3.29	Примечание: При выборе типа станка можно использовать любую доступную информацию.
4		Привод перемещения стола		
5		Привод главного движения фрезерной бабки		
6		Привод салазок		

Окончание табл. 5.1

7	Шлифовальный	Привод подачи стола	Рис. 6.4	
8		Привод механизма правки шлифовального круга		
9		Привод шлифовального круга		
10	Многоцелевой	Привод шпинделя	Рис. 5.19	
11		Привод перемещения шпиндельной бабки		

Таблица 5.2

Вариант следящего электропривода подачи

№ п. п.	Тип датчика положения	Режим работы	Контур регулирования	
			Скорости	Тока
0	СКВТ	Фазовращателя	Аналоговый	Аналоговый
1		Амплитудно-разностный		
2		Фазово-разностный		
3	Импульсный			
4	СКВТ	Фазовращателя	Цифровой	Аналоговый
5		Амплитудно-разностный		
6		Фазово-разностный		
7	Импульсный			
8	СКВТ	Фазовращателя	Цифровой	С цифровой моделью якорной цепи
9		Амплитудно-разностный		
10		Фазово-разностный		
11	Импульсный			

Таблица 5.3

Варианты сопряжения МК с ЦП

№	Магистраль сопряжения
0	Станочная магистраль типа общая шина Q-BUS
1	ISA
2	PCI
3	RS422/RS485

Таблица 5.4

## Вариант электропривода

№ п.п.	Система ЭП		Регулирование
	Преобразователь	Двигатель	
0	ТП-	ПТ	Однозонное
1	ТП		Двухзонное
2	ТП		Совместно с АКС
3	ШИП	ПТ	Однозонное
4			Однозонное с регулируемым динамическим торможением
5			Однозонное с регулируемым режимом противовключения
6			Двухзонное
7	АИН	БДПТ	С коммутатором обмоток и режимом ШИМ
8		ШД	
9		АД	Закон $u/f=\text{const}$
10			С компенсацией падения напряжения на статоре
11			С моделью АД
11			Частотно-токовое
12			Векторное управление

Прим.: Указанные пункты задания должны быть представлены в пояснительной записке в виде соответствующего описания

## 5.2. ТРЕБОВАНИЯ К ПОЯСНИТЕЛЬНОЙ ЗАПИСКЕ

Пояснительная записка должна включать следующие разделы:

- 1) задание на разработку;
- 2) общий вид, описание рабочих органов и кинематической схемы станка;
- 3) структурную схему системы управления, математическое описание электропривода, выбор регуляторов, результаты моделирования в среде MATLAB, проверку влияния дискретности регуляторов и их разрядности на качественные показатели системы управления;
- 4) функциональную схему микропроцессорной системы управления ЭП и ее описание, алгоритмы управления, программу.
- 5) лист принципиальной блока управления электропривода, краткое описание МК и принципиальной схемы, описание интерфейсных узлов МК, узлов сопряжения с интерфейсом УЧПУ и преобразователем.

Примечание. Обозначения на схеме должны соответствовать ЕСКД, ГОСТ 2.743 – 72 и др. В качестве элементной базы рекомендуется использовать серии зарубежных микросхем и отечественных аналогов.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Соломенцов Ю. М. Управление гибкими производственными системами / Ю. М. Соломенцов. – М. : Машиностроение, 1988. – 352 с.
2. Многоцелевые системы ЧПУ гибкой механообработкой / В. Н. Алексеев, В. Г. Воржев, Г. П. Гырдымов и др.; под общ. ред. В. Г. Колосова. – Л., 1984.
3. Системы программного управления промышленными установками и робототехническими комплексами : учебное пособие для вузов / Г. И. Прокофьев, Л. Н. Рассудов. – Л. : Энергоатомиздат, 1990. – 352 с.
4. Управляющие вычислительные комплексы : учебное пособие / Н. Л. Прохоров, Г. А. Егоров, В. Е. Красовский и др. – М. : Финансы и статистика, 2003. – 352 с.
5. Система «Контур». <http://www.scada.com>.
6. Автоматизированная система управления технологическими процессами АСУТП (Octagon Systems). <http://www.ya.ru>.
7. Автоматизированная система управления технологическими процессами предприятия (Siemens). <http://www.ya.ru>.
8. Модернизация станков с УЧПУ мод. 2С42, 2Р22, 2У22. <http://www.cncinfo.ru>.
9. Системы ЧПУ SINUMERIK 810D/840D: Аппаратные и технологические возможности. <http://ya.ru>
10. Системы числового программного управления серий FMS-3000 и FMS-3100. <http://ya.ru>.
11. Перельмутер В. М. Цифровые системы управления тиристорным электроприводом / В. М. Перельмутер, Л. К. Соловьев. – Киев : Техника, 1983. – 104 с.
12. Микропроцессорные средства производственных систем / В. Н. Алексеев, А. М. Коновалов, В. Г. Колосов и др.; под общ. ред. В. Г. Колосова. – Л., 1988.
13. Сташин В. В. Проектирование цифровых устройств на однокристальных микропроцессорах / В. В. Сташин, А. В. Урусов, О. Ф. Мологандцев. – М. : Энергоатомиздат, 1990. – 224 с.
14. Микропроцессорные системы автоматического управления / В. А. Бесекерский, Н. Б. Ефимов, С. И. Зиятдинов и др. – М., 1988.
15. Схемотехника цифровых преобразователей перемещения : справочное пособие / В. Г. Домчарев, В. Г. Матвеевский, Ю. В. Смирнов. – М., 1987.
16. Иванов В. М. Микропроцессорные системы управления электроприводом : методические указания и задания к курсовой работе / В. М. Иванов. – Ульяновск, 2001. – 36 с.
17. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel / В. В. Гребнев. – М. : ИП РадиоСофт, 2002. – 176 с.
18. Ефстифеев А. В. Микроконтроллеры AVR семейства Classic «ATMEL» / А. В. Ефстифеев. – М. : Издательский дом «Додэка-XXI», 2002. – 288 с.
19. Федорков Б. Г. Микросхемы ЦАП и АЦП: Функционирование, пара-

метры, применение / Б. Г. Федорков, В. А. Телец. – М. : Энергоатомиздат, 1990. – 320 с.

20. Файнштейн В. Г., Микропроцессорные системы управления тиристорными электроприводами / В. Г. Файнштейн, З. Г. Файнштейн. – М. : Энергоатомиздат, 1986. – 240 с.

21. Перельмутер В. М. Системы управления тиристорного электропривода постоянного тока / В. М. Перельмутер, В. А. Сидоренко. – М. : Энергоатомиздат, 1988. – 304 с.

22. Комплектные системы управления электроприводами тяжелых металлорежущих станков / Н. В. Донской, А. А. Кириллов, Я. М. Купчан и др. – М., 1980. – 288 с.

23. Михайлов О. П. Динамика электромеханического привода металлорежущих станков / О. П. Михайлов. – М. : Машиностроение, 1989. – 224 с.

24. Тавернье К. PIC-микроконтроллеры: Практика применения / К. Тавернье. – М.: ДМК Пресс, 2002. – 272 с.

25. Ремизевич Т. В. Микроконтроллеры для встраиваемых приложений: от общих подходов – к семействам HC05 и HC08 фирмы Motorola / Т. В. Ремизевич. – М. : ДОДЭКА, 2000. – 272 с.

26. Схемотехника электронных систем. Микропроцессоры и микроконтроллеры / В. И. Бойко, А. Н. Гурский, В. Я. Жуйков и др. – СПб., 2004. – 464 с.

27. Гук М. Аппаратные интерфейсы ПК. Энциклопедия / М. Гук. – СПб : Питер, 2002. – 528 с.

28. Алексеенко А. К. Применение прецизионных аналоговых микросхем / А. К. Алексеенко, Е. А. Колембет, Г. И. Стародуб. – М. : Радио и связь, 1985. – 256 с.

29. Гальперин М. В. Практическая схемотехника в промышленной автоматике / М. В. Гальперин. – М. : Энергоатомиздат, 1987. – 320 с.

30. Цифровые и аналоговые микросхемы: справочник / С. В. Якубовский, Л. И. Ниссельсон, В. И. Кулемова и др. – М. : Радио и связь, 1989. – 496 с.

31. Козаченко В. Ф. Микроконтроллеры: руководство по применению 16-разрядных микроконтроллеров Intell MCS-196/296 во встроенных системах управления / В. Ф. Козаченко. – М. : Изд-во ЭКОМ, 1997. – 688 с.

32. Куприянов М. С. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования / М. С. Куприянов, Б. Д. Матюшкин. – СПб. : Политехника, 1998. – 592 с.

Учебное издание

**ЭЛЕКТРОПРИВОДЫ  
С СИСТЕМАМИ ЧИСЛОВОГО ПРОГРАММНОГО  
УПРАВЛЕНИЯ**

Учебное пособие

Составитель ИВАНОВ Владимир Михайлович

Редактор *Г. А. Колденкова*

Подписано в печать 16.01.2006. Формат 60×84/16. Бумага офсетная.

Печать трафаретная. Усл. печ. л. 7,21.

Тираж 100 экз. Заказ

Ульяновский государственный технический университет  
432027, Ульяновск, Сев. Венец, 32.

Типография УлГТУ, 432027, Ульяновск, Сев. Венец, 32.