

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

И. П. Кобяк

**СПЕЦКОМПЬЮТЕР С НЕЙМАНОВСКОЙ
БАЗОВОЙ АРХИТЕКТУРОЙ**

*Рекомендовано УМО
по образованию в области информатики и радиоэлектроники
для специальности 1-40 02 01
«Вычислительные машины, системы и сети»
в качестве учебно-методического пособия*

Минск БГУИР 2013

УДК 004.2:004.3(076)
ББК 32.973.26-02я73
К55

Р е ц е н з е н т ы:

кафедра систем автоматического управления
Белорусского национального технического университета,
(протокол №4 от 21.11.2012);

главный научный сотрудник ОИПИ НАН Беларуси,
доктор технических наук А. А. Дудкин

Кобяк, И. П.

К55 Спецкомпьютер с неймановской базовой архитектурой : учеб.-метод. пособие / И. П. Кобяк. – Минск : БГУИР, 2013. – 105 с. : ил.
ISBN 978-985-488-986-3.

Пособие по дисциплине «Структурно-функциональная организация ЭВМ» предназначено для изучения принципов проектирования реальных компьютеров с архитектурой, близкой к неймановской. Учебный материал позволяет получить представление о подсистемах ОЗУ, обработки данных и блоке микропрограммного управления, а также о способах их объединения в систему с помощью схем внутреннего интерфейса.

УДК 004.2:004.3(076)
ББК 32.973.26-02я73

ISBN 978-985-488-986-3

© Кобяк И. П., 2013
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2013

Содержание

Введение	5
1. Проектирование архитектуры компьютера специального назначения	7
1.1. Базовые структуры спецкомпьютеров	7
1.2. Реализация вычислительного процесса в реальном масштабе времени	9
1.3. Классификация спецкомпьютеров по условиям эксплуатации. Требования к спецсистемам.....	10
1.4. Исследование арифметической функции, решаемой спецсистемой	12
1.5. Проектирование системы команд.....	18
1.6. Кодирование системы команд	23
1.7. Переход от системы команд к блок-схеме компьютера.....	25
1.8. Расчет компьютерной памяти и схемы управления адресами.....	28
2. Проектирование блока обработки данных (БОД) спецкомпьютера	36
2.1. Проектирование БОД с использованием МПС К1804ВС1	36
2.2. Проектирование БОД с использованием МПС К1804ВС2	40
2.3. Включение сформированных примитивов в блок-схему компьютера	43
2.4. Пример микропрограммирования БОД на МПС К1804ВС1.....	47
3. Проектирование блока микропрограммного управления (БМУ)	48
3.1. Проектирование блока микропрограммного управления на основе СУА М К1804ВУ1	48
3.2. Проектирование блока микропрограммного управления на основе СУАМ К1804ВУ2.....	53
3.3. Проектирование блока микропрограммного управления на основе УПМ К1804ВУ4.....	54
3.4. Микропрограммирование БОД на МПС К1804ВС1 и БМУ на УПМ К1804ВУ4	58

ВВЕДЕНИЕ

Проектирование компьютеров специального назначения является одной из важнейших задач, стоящих перед разработчиками цифровой аппаратуры. Основное назначение спецкомпьютеров – это обработка определенного класса алгоритмов в некоторой заданной проблемно-ориентированной области. В частности, большой класс вычислительных систем используется в блоках управления механическими подвижными платформами в целях коррекции их траектории движения. Эффективность решения нетривиальных задач при этом во многом зависит от качества бортовых компьютеров и их программного обеспечения, входящих в состав управляемых объектов.

Назначение бортовых компьютеров, как правило, состоит в следующем [1].

Во-первых, это сбор и обработка данных об окружающей среде.

Во-вторых, организация обмена информацией между иерархическими звеньями системы управления.

В-третьих, расчет координат при наведении объектов на цель и т.д.

В целом процесс проектирования спецкомпьютера определяется условиями применения подвижной платформы, комплексом алгоритмов, подлежащих реализации на борту носителя, точностью представления входной и выходной информации.

Как правило, функционирование каждого бортового спецкомпьютера платформы происходит не автономно, а в тесном взаимодействии с системами наземных стационарных или подвижных комплексов. Поэтому сложность разработки структуры и программного обеспечения обычно достаточно велика и требует существенных материальных затрат.

Стационарные комплексы управления в целом оказывают существенное влияние на архитектуру бортовых систем. Так, в состав средств наземного управления предполагается включение следующих подсистем:

- 1) центральный компьютер локальной сети;
- 2) средства контроля и диагностики подсистем;
- 3) подсистема навигации и радиопеленгации;
- 4) процессор(ы) для расчета координат целей;
- 5) подсистемы для реализации информационного и служебного программного обеспечения.

Кроме того, при проектировании спецкомпьютеров приходится дополнительно учитывать возможность субъективного вмешательства в процесс управления, а также необходимость оперативного принятия сложных решений в условиях, близких к экстремальным.

Структурный анализ бортового компьютера позволяет выделить в нем следующие технические компоненты.

1. Информационные средства или датчики первичной информации, предназначенные для сбора данных об окружающей среде, объекте управления и взаимном расположении подвижных и стационарных систем.

2. Линии и схемы передачи данных, осуществляющие связь рассредоточенных систем и исполнительных механизмов носителя с бортовым ком-

пьютером, а также средства для надежной передачи управляющей информации в соответствии с принятым протоколом обмена.

3. Вычислительные средства для обработки информации, принятия решений и формирования команд управления. Эти средства занимают центральное место в системе управления и определяют всю специфику работы проектируемого компьютера. В целом постоянное усложнение пользовательских задач и алгоритмов управления определяет и постоянство тенденции усложнения аппаратуры и программного обеспечения. В связи с этим на практике все чаще применяются многомашинные вычислительные комплексы, а также мультипроцессорные системы, способные за короткий промежуток времени решить практически любую задачу.

4. Исполнительные механизмы, предназначенные для отработки команд в соответствии с условиями применения того или иного подвижного объекта. К этим средствам относят: устройства, напрямую связанные с механической коррекцией положения системы в пространстве (микродвигатели, сельсины, бесконтактные переключатели и т.д.), системы индикации, отображения, жизнеобеспечения и другие электромеханические модули.

В настоящее время основной принцип организации бортовых вычислительных комплексов в управляющую систему базируется на иерархической подчиненности всех подсистем управляемой платформы одному из компьютеров верхнего уровня. При этом протокол взаимодействия устройств определяется условиями эксплуатации и назначением спецкомпьютера. Кроме того, каждая подсистема бортового комплекса должна функционировать автономно или включаться в состав системы при необходимости получения требуемой конфигурации вычислителя.

В общем случае практика проектирования спецкомпьютеров показала, что создание сложных управляющих систем представляет собой трудно формализуемую задачу. Вследствие этого проектирование бортовых компьютеров обычно основывается на личном опыте инженерно-технического персонала, использовании экспертных систем и баз знаний, анализе и модернизации базовых компьютерных моделей.

В общем случае процесс проектирования спецсистем управления является сложной задачей, трудоемкость решения которой достаточно велика даже для коллектива квалифицированных инженеров. В связи с этим данное пособие ориентировано только на изучение принципов построения компьютеров с неймановской базовой архитектурой, реализуемой на основе четырех основных подсистем: блока обработки данных, оперативной памяти, блока микропрограммного управления, подсистемы ввода-вывода. Иными словами, данное пособие является компьютерной «таблицей умножения» для любого современного специалиста.

Процесс проектирования спецкомпьютера требует базовых знаний по дисциплинам: «Арифметические и логические основы ВТ», «Схемотехника», «Структурная и функциональная организация ЭВМ».

1. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЬЮТЕРА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

1.1. Базовые структуры спецкомпьютеров

Учитывая специфику эксплуатации бортовых машин, различают два основных вида структурных схем вычислительных устройств:

- 1) спецкомпьютер в контуре управления автономным объектом;
- 2) спецкомпьютер в составе управляемого комплекса оперативно-технических средств.

Включение управляющего компьютера в систему управления, например, летательного аппарата, осуществляется как по первой, так и по второй схемам.

В первом случае структура комплекса будет иметь вид, показанный на рис. 1.1.

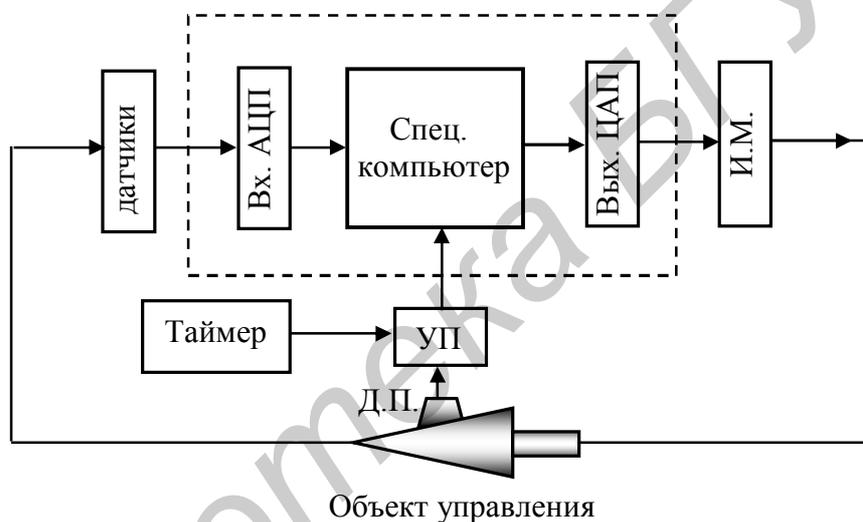


Рис. 1.1

Характерными особенностями включения спецсистем в контур управления подвижной платформой являются:

- 1) сопряжение цифровой и аналоговой аппаратуры;
- 2) многократное повторение алгоритмов приема, обработки и выдачи сигналов управления в систему;
- 3) функционирование вычислительной системы в реальном масштабе времени;
- 4) наличие временных ограничений на выполнение алгоритмов обработки информации;
- 5) повышенные требования к надежности аппаратуры и программного обеспечения спецкомпьютера.

Рассмотрим каждую из особенностей организации бортовой спецсистемы более подробно.

Первая характерная особенность состоит в том, что измеряемые датчиком физические величины по своей природе являются непрерывными.

Вследствие этого ввод информации в вычислительную систему требует преобразования вида «аналог–код», т.е. квантования во времени и по уровню. Кроме того, большинство исполнительных устройств работает под управлением аналоговых напряжений. Поэтому дискретные результаты расчетов должны быть преобразованы в аналоговые величины, иными словами, требуется обратное преобразование вида «код–аналог». Следует учитывать также, что различный физический характер преобразуемой АЦП и ЦАП информации приводит к значительному усложнению системы ввода–вывода. Создание же преобразователей, обладающих высокоточными или прецизионными характеристиками, представляет собой известную научную и практическую проблему. Таким образом, процесс ввода и преобразования сенсорной и другой информации в целом есть трудоемкая процедура. Поэтому задача проектирования данного блока спецкомпьютера тесно связана с научными исследованиями и инженерными разработками в соответствующей области знаний.

Вторая особенность работы компьютера в контуре управления заключается в том, что алгоритмы решаемых задач, описанные на машинном языке, хранятся в ПЗУ и не изменяются в процессе всего периода эксплуатации. Любой алгоритм выполняется многократно, а при каждом повторении меняются лишь исходные данные, вводимые извне. Частота повторения программ при этом зависит от скорости решения задач в процессоре, а в ряде случаев – от скорости приема данных, ввод–вывод которых сопровождается инерционностью преобразования.

Третья особенность заключается в реализации реального масштаба времени, а также в необходимости оперативного управления подсистемами комплекса при сохранении точности решения задач.

Четвертая особенность заключается в предоставлении процессору кванта времени требуемой длины, т.е. промежутка времени, необходимого для выполнения части реального алгоритма. Ограничение на длительность кванта налагаются лишь из соображений точности решения задач и соображений оперативного управления объектом или платформой.

Пятая особенность проектирования спецсистемы подчеркивает важность такой характеристики, как надежность всего комплекса. При этом под надежностью понимается не только техническая долговечность аппаратуры, но и надежность программного обеспечения. Данная характеристика определяется выбором методов решения задач, алгоритмами представления исходных данных (СОК и т.д.), степенью наработки методов контроля за правильностью отработки программ и команд в процессоре.

Второй тип структуры специализированного вычислительного устройства учитывает особенности взаимодействия человека и компьютера в составе подвижной платформы (пилотирование самолетов, управление плавсредствами, управление атомной техникой и т.д.). Данная структура может быть реализована по схеме, представленной на рис. 1.2. В отличие от предыдущей структуры схема бортового компьютера содержит средства связи для управ-

ления удаленными объектами, средства приема и передачи информационных сообщений, пульт связи человека и компьютера.

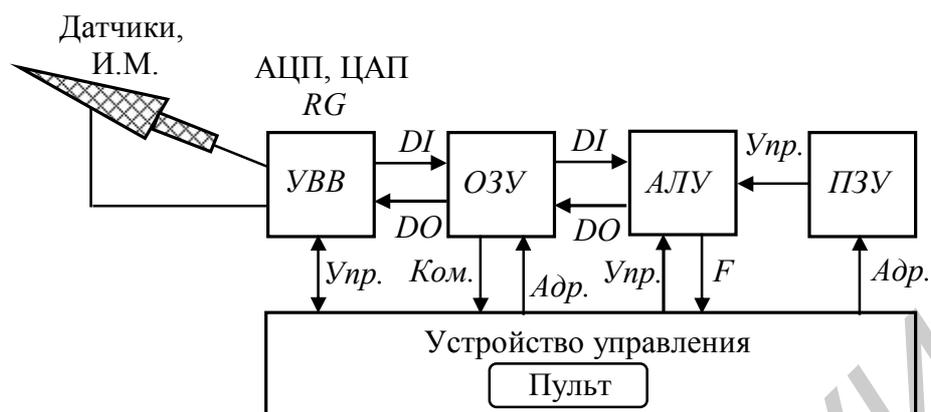


Рис. 1.2

Для ввода–вывода данных в приведенной структуре используются параллельные и последовательные порты, а также двоичные каналы прямого доступа в память. Кроме того, при проектировании схем ввода–вывода системы, приведенной на рис. 1.2, необходимо учитывать следующие две характеристики обрабатываемой информации: 1) рабочие диапазоны частот преобразования и 2) точность измерения входных величин. Анализ указанных параметров позволяет спроектировать наиболее рациональные устройства и обеспечить требуемую точность при управлении подвижной платформой. Реализация остальных блоков не отличается от предыдущей структуры и использует известные принципы проектирования систем высокой производительности.

Заметим, что в реальных управляющих системах используются многоядерные процессоры и многомашинные комплексы, способные обрабатывать информацию, поступающую в систему от множества первичных или вторичных источников информации.

1.2. Реализация вычислительного процесса в реальном масштабе времени

Под вычислительным процессом реального времени понимают алгоритмы обработки информации, при которых расчеты выполняются со скоростью, достаточной для воздействия результатов обработки данных на протекание процессов в некотором реальном объекте. Примером такой системы является спецкомпьютер, ориентированный на управление летательным аппаратом, и решение задач оптимизации параметров полета при переменных значениях параметров окружающей среды. Основным требованием, предъявляемым к системам реального времени (СРВ), является требование максимальной частоты ввода информации при условии сохранения качества процесса управления.

Упрощенный алгоритм функционирования спецкомпьютера в реальном масштабе времени состоит в выполнении следующих действий.

Движение объекта управления характеризуется группой параметров или вектором $\{x\}$. Эти параметры воздействуют на входные устройства датчиков информации и далее передаются на входы АЦП. Для управления объектом в пространстве или на подвижной наземной платформе наиболее часто используется синхронный принцип связи компьютера с управляемой платформой. При этом процесс управления разбивается на интервалы времени равной продолжительности с помощью синхросигналов, поступающих от таймера. Каждый квант времени отождествляется с одним циклом вычислений, который начинается по приходу синхроимпульса на устройство прерывания (УП). В начале очередного цикла выполняется опрос систем кодирования и ввода данных о текущем состоянии объекта в пространстве. Обычно считается, что измерительная информация не изменяется в пределах одного цикла. Таким образом, процесс ввода данных в компьютер всегда сопровождается квантованием аналоговых величин по уровню. Точность преобразования «аналог–код» определяется разрядностью входной шины.

Однако в ряде систем реального времени существует необходимость в более тесной связи управляемого объекта и спецкомпьютера. Для этого (дополнительно) используется асинхронный принцип связи управляющей и управляемой систем. Сигналы прерывания в данном случае формируются не синхронно от тактового генератора компьютера, а с помощью специальных датчиков, представляющих собой некоторые контактные или бесконтактные устройства. Каждый сигнал датчика для бортового компьютера является требованием по прекращению вычислений и переходу к выполнению подпрограммы, соответствующей текущему прерыванию. При этом сигналы управления воздействуют непосредственно на исполнительные механизмы, т.е. эти сигналы напрямую управляют перемещением, например, летательного аппарата. Соответственно такой режим работы системы получил в технике название «прямое цифровое управление».

В некоторых случаях движущаяся платформа может содержать несколько сложных агрегатов. При этом вся система управления должна строиться с использованием алгоритмов и систем автоматического регулирования (САР). Автоматические регуляторы следят за параметрами (например, за влажностью воздуха, количеством углекислоты, точностью наведения на цель и т.д.) и поддерживают их в заданных пределах. Центральный компьютер контролирует процессы в САР, координирует взаимодействие автономных систем, выдает в САР оптимальные компараторные настройки.

1.3. Классификация спецкомпьютеров по условиям эксплуатации. Требования к спецсистемам

В зависимости от условий эксплуатации компьютеры спецназначения разделяют на две большие группы:

- 1) стационарные системы, нетранспортируемые в рабочем состоянии;
- 2) транспортируемые компьютеры, т.е. устанавливаемые на подвижных механических платформах.

Компьютеры первой группы могут эксплуатироваться как в жилых отапливаемых помещениях, так и в составе подвижных платформ, работающих на открытом воздухе при различных климатических условиях. Поскольку стационарные системы доставляются к месту эксплуатации транспортными средствами, к ним предъявляются определенные требования по вибростойкости и уровню температур. Для компьютеров данной группы диапазон рабочих температур составляет от -50°C до $+70^{\circ}\text{C}$, уровень влажности 90–98 %, а уровень вибраций до 120 Гц при ускорении 6 g.

Транспортируемые спецсистемы устанавливаются на автомобилях, железнодорожном, морском и гусеничном транспорте, а также на различных летательных аппаратах. Специфика эксплуатации в данном случае определяется условиями передвижения платформы. Предельные значения и диапазоны рабочих температур, влажности и вибраций для рассматриваемых компьютеров должны значительно превосходить аналогичные значения, нормируемые для стационарных систем. Это обусловлено тем, что частота вибраций двигателей малых и больших ракет, например, может достигать 2500 Гц при ускорении платформы до 20 g. Данный фактор отрицательно сказывается на работе всех подсистем и модулей управляющего компьютера.

Требования или характеристики, которым должны удовлетворять спецкомпьютеры, разделяются на следующие группы.

1. Тактико-технические характеристики, включающие в себя информацию о назначении компьютера и его технических параметрах – быстродействии, емкости памяти, разрядности слова и т.д. В некоторых случаях бывает важной также информация об архитектуре системы. Это обусловлено тем, что особенности реализации процессора могут указывать на возможности распараллеливания вычислений. Таким образом, можно косвенно судить о производительности компьютера на определенных классах решаемых задач.

2. Конструкторско-технологические характеристики – включают в себя ограничения по габаритам, массе, принципам конструктивного исполнения. Сюда включают также особенности защиты компьютера от механических и климатических факторов, принципы организации взаимодействия с ВУ.

3. Эксплуатационные требования – это требования к простоте обслуживания и к пульту управления компьютером.

4. Экономические требования – это требования к стоимости компьютера. Они включают в себя затраты на проектирование и изготовление системы.

5. Требования к надежности – определяют такие параметры, как вероятность безотказной работы, время наработки на отказ, среднее время восстановления работоспособности. Параметры п. 5 имеют следующие определения.

Вероятность безотказной работы – это вероятность того, что в заданном интервале времени при определенных режимах и условиях эксплуатации в системе не произойдет ни одного отказа.

Наработка на отказ – это средняя продолжительность работы компьютера между двумя отказами.

Среднее время восстановления работоспособности определяется средним временем обнаружения и восстановления рабочего режима.

Анализ требований к надежности позволяет сделать вывод об их определенной противоречивости. Например, введение аппаратурной избыточности для устранения ошибок при сбоях и отказах приводит к увеличению числа контактов и разъемов в конструкции, к повышению плотности проводников печатного монтажа, что, естественно, снижает общую надежность системы. Таким образом, при проектировании каждому требованию присваивается свой весовой коэффициент, величина которого определяется спецификой использования компьютера. Выбор оптимального сочетания этих коэффициентов является одной из основных задач, решаемых на всех этапах проектирования техники.

1.4. Исследование арифметической функции, решаемой спецсистемой

Сущность этапа проектирования архитектуры заключается в разработке:

- 1) алгоритмов решения задач;
- 2) разработке системы команд;
- 3) в выполнении перехода от алгоритма к структуре системы.

Исходными данными для проектирования аппаратуры и ПО являются алгоритмы решения задач [2, 3].

В целом проектирование алгоритмов начинается с уяснения цели их разработки (для чего?), формулирования основных функций (каким образом?) и определения исходных данных (над чем?) для получения требуемых решений поставленных задач.

Ограничениями на разработку алгоритмов являются [3, 4]:

- 1) уровень финансирования и сроки разработки спецсистемы;
- 2) степень наработок в области техники, реализующей требуемые преобразования;
- 3) уровень научных результатов в решении заданных теоретических задач;
- 4) количество и достоверность данных, поступающих от источника первичной информации;
- 5) время исполнения и точность реализации алгоритмов на борту носителя.

Кроме того, следует учитывать критерии эффективности решаемых задач, значения обязательных и желательных показателей качества, помехозащищенность решений.

При разработке алгоритмов используют следующий порядок проектирования: во-первых, выбираются аналитические соотношения, позволяющие организовать заданные вычисления; во-вторых, разрабатывается блок-схема решения задачи с учетом заданной точности представления результата; в-третьих, оптимизируются вычисления, представленные граф-схемой алгоритма.

Пусть, например, требуется разработать алгоритм и программу для определения значений функции $y(x)=e^x$, $0 \leq x \leq 12$. Для решения поставленной задачи на первом этапе построим график, соответствующий заданному соотношению (рис. 1.3) и определим максимальное и минимальное значения функции, а также диапазон изменения аргумента.

Теоретически диапазон изменения аргумента лежит в пределах $-\infty \leq x \leq +\infty$, при этом граничные значения функции будут равны $0 \leq y(x) \leq +\infty$.

Для расчетов конкретных значений $y(x)$ используем разложение e^x в ряд Тейлора вида

$$y(x) = e^x = 1 + \sum_{i=1}^{\infty} \frac{x^i}{i!}, \quad x \geq 0. \quad (1.1)$$

Анализ соотношения (1.1) показывает, что одному значению функции соответствует бесконечное число членов ряда. Выбор же числа членов для расчета $y(x)$ при заданном x определяется с учетом двух практических ограничений:

- 1) точности представления результатов решения задачи в компьютере;
- 2) допустимой длительности расчета управляющей информации.

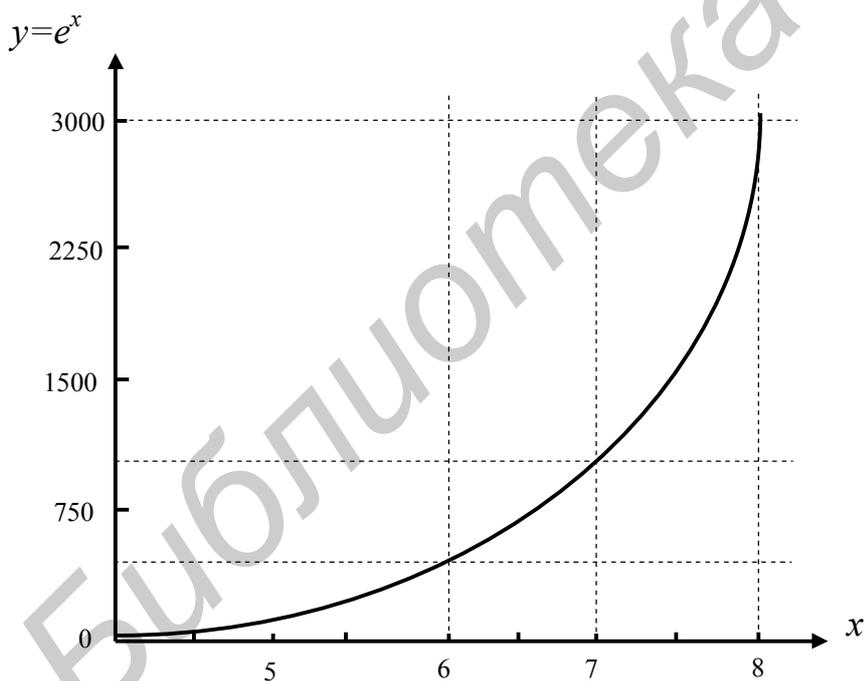


Таблица 1.1

x	$y=e^x$
1	2,718
2	7,389
3	20,09
4	54,6
5	148,4
6	403,4
7	1096,5
8	2981,4
9	8103,08
10	22026,47
11	59874,14
12	162754,79

Рис. 1.3

Для простоты будем считать, что общая ошибка вычислений в системе управления обуславливается двумя погрешностями: первая из них Δ_1 следует из ограничений на число членов в разложении в ряд Тейлора, вторая Δ_2 определяется ограничениями разрядной сетки.

Определим теперь разрядность сетки спецкомпьютера равной 36 разрядам и будем считать, что числа в памяти представлены в дополнительном коде с плавающей запятой. Тогда формат компьютерного слова будет иметь вид, показанный на рис. 1.4.



Рис. 1.4

При расчете разрядности порядка из общей длины 36-разрядного слова вычитается один разряд на знак мантииссы и один разряд на знак порядка. После этого рассчитывается длина поля порядка

$$n_p = \text{intlog}_2 M = 5.$$

В общем случае длина поля n_p может уточняться с учетом требования реализации функции сдвига для единичного операнда в пределах всей разрядности поля мантииссы.

Исследуем далее закон изменения общего члена ряда (1.1) при различных значениях аргумента x . Поставленной задаче будут соответствовать следующие табличные (табл. 1.2) и графические (рис. 1.5) величины.

Таблица 1.2

$x \setminus i$	1	2	3	4	5	6	7	8
1	1,0	0,5	0,17	0,042	0,008	0,0012	0,0002	0,000025
2	2,0	2,0	1,333	0,6666	0,2666	0,0888	0,0254	0,00635
3	3,0	4,5	4,5	3,375	2,025	1,0125	0,434	0,1627
4	4,0	8,0	10,666	10,6666	8,53333	5,68888	3,2513	1,6254
5	5,0	12,5	20,83333	26,041	26,041	21,701	15,501	9,688
6	6,0	18,0	36,0	54,0	64,8	64,8	55,54	41,657
7	7,0	24,50	57,166	100,04	140,058	163,401	163,401	142,97
8	8,0	32,0	85,333	170,666	273,066	364,08	416,102	416,102
9	9,0	40,5	121,5	273,375	492,075	738,11	949,0	1067,62
10	10,0	50,0	66,667	416,666	833,333	1388,88	1984,13	2480,16
11	11,0	60,5	221,833	610,042	1342,09	2460,5	3866,5	5316,44
12	12,0	72,0	288,0	864,0	2073,6	4147,2	7109,5	10664,0

$x \setminus i$	9	10	11	12	13	14	15	16
5	5,3823	2,6911	1,2234	0,50969	0,19603	0,0700	0,02333	0,00729
6	27,771	16,663	9,0888	4,5444	2,0974	0,89889	0,3595	0,1348
7	111,204	77,8426	49,536	28,896	15,5594	7,7797	3,63054	1,58883
8	369,868	295,894	215,196	143,464	88,285	50,448	26,906	...
9	1067,62	960,86	786,16	589,621	408,2	262,41	157,45	...
10	2755,73	2755,73	2505,21	2087,67	1605,9	1147,07	764,71	...
11	6497,87	7147,66	7147,66	6552,02	5544,01	4356,01	3194,41	...
12	14218,9	17062,8	8613,9	8613,9	7182,1	4727,5	1782,0	...

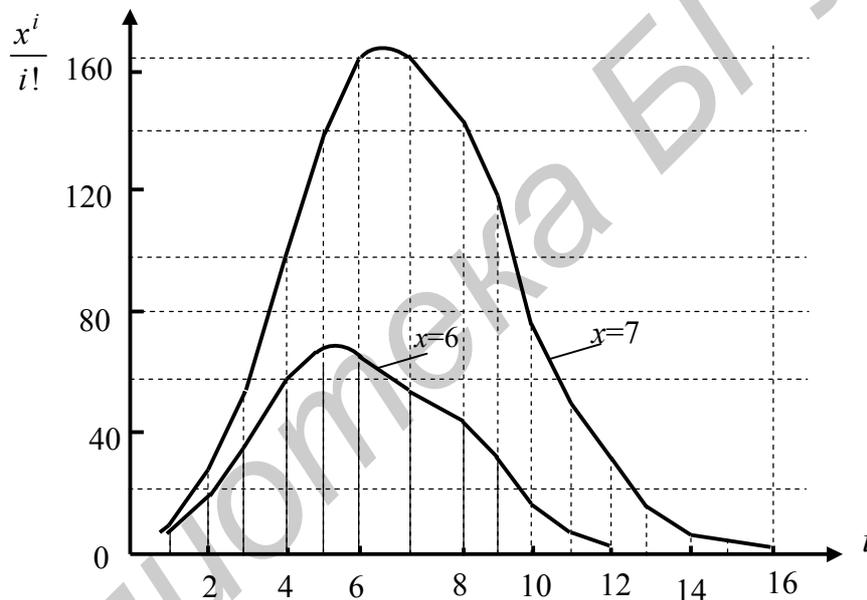


Рис. 1.5

Для расчета погрешности Δ_1 выполним суммирование членов ряда в табл. 1.2 в соответствии с ростом номеров i в табл. 1.3.

Таблица 1.3

$\begin{matrix} sum \\ x \end{matrix} \setminus$	1	2	3	4	5	6	7	8
1	2,0	2,5	2,6666	2,70833	2,71666	2,71805	2,71825	2,718278
2	3,0	5,0	6,333	6,9999	7,26666	7,35555	7,38095	7,38731
3	4,0	8,5	13,0	16,375	18,4	19,412	19,846	20,0085
4	5,0	13,0	23,666	34,3333	42,8666	48,5555	51,8068	53,4323

Продолжение табл. 1.3

$\frac{sum}{x}$	1	2	3	4	5	6	7	8
5	6,0	18,5	39,33333	65,3743	91,4153	113,116	128,617	138,305
6	7,0	25,0	61,0	115,0	179,8	244,6	300,14	341,797
7	8,0	32,50	89,666	189,706	329,764	493,165	656,566	799,536
8	9,0	41,0	126,333	296,999	570,065	934,145	1350,247	1766,349
9	10,0	50,5	172,0	445,375	937,45	1675,56	2624,56	3692,18
10	11,0	61,0	227,667	644,333	1477,666	2866,55	4850,68	7330,84
11	12,0	72,5	294,333	904,375	2246,465	4706,96	8573,47	13889,9
12	13,0	85,0	373,0	1237,0	3310,6	7457,8	14576,3	25231,3

Окончание табл. 1.3

$\frac{sum}{x}$	9	10	11	12	13	14	15	16
5	143,6873	146,378	147,6018	148,1149	148,3075	148,3775	148,3308	148,3381
6	369,568	386,231	395,32	399,864	401,9618	402,861	403,220	403,355
7	910,74	988,5826	1038,119	1067,015	1082,574	1090,35	1093,98	1095,57
8	2136,217	2432,111	2647,31	2790,77	2879,06	2929,50	2956,41	2969,863
9	4759,8	5720,66	6506,82	7096,44	7504,64	7766,64	7924,1	...
10	10086,6	12842,3	15347,51	17435,2	19041,08	20188,6	20952,9	...
11	20387,8	27535,43	34683,09	41235,11	46779,12	51135,1	54329,5	...
12	39450,2	56513,0	75126,9	93740,8	110922,9	125650,4	137432,4	...

Из формата компьютерного слова следует, что максимальное значение функции с учетом разрядности мантииссы $M = 29$ и порядка $P = 31$ определится формулой

$$2^{29} - 1 = 536\,870\,911, \quad (1.2)$$

$$y_{\max} = 0,536\,870\,911 \cdot 2^{31} = 1\,152\,921\,502,459\,363\,328.$$

Данному значению функции соответствует максимальное численное значение аргумента $x = 20,8$ (по условиям задачи оставляем $x_{\max} = 12$).

Минимальное значение будет определяться единичным битом в младшем разряде мантииссы с учетом порядка $P = -31$:

$$y_{\min} = -1 \cdot 2^{-29} \cdot 2^{-31} = -1 \cdot 2^{-60}. \quad (1.3)$$

Расчеты по формуле (1.1) показывают, что значению функции (1.2), например при $y(x) \geq 0$ и $x_{\max} = 21$, соответствует максимальное значение $i \approx 19$.

Следует, однако, заметить, что процесс расчета достаточно большого числа членов ряда задача длительная и не всегда обоснованная. В связи с

этим введем ограничения на расчетный параметр e^x , полагая, что допустимая погрешность вычислений равна $\Delta_1 = 0,5\%$. Тогда из формулы (1.1) для аргумента x можно записать соотношение

$$\Delta_1 = \frac{e^x - \left[1 + \sum_{i=1}^{i_{\max}} \frac{x^i}{i!} \right]}{e^x} = 0,005. \quad (1.4)$$

На основании ряда (1.1) при $x=1$ имеем

$$1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \dots \quad (1.5)$$

Тогда по формуле (1.4) найдем ряд погрешностей для значений i , равных:

$$\begin{aligned} i=1: & \quad 1+1=2, & \quad \Delta_1 = 0,26424, \\ i=2: & \quad 1+1+\frac{1}{2}=2,5, & \quad \Delta_1 = 0,0803, \\ i=3: & \quad 1+1+\frac{1}{2}+\frac{1}{6}=2,66(7), & \quad \Delta_1 = 0,01899, \\ i=4: & \quad 1+1+\frac{1}{2}+\frac{1}{6}+\frac{1}{24}=2,7083(3), & \quad \Delta_1 = 0,00366. \end{aligned} \quad (1.6)$$

Итак, из соотношений (1.6) следует, что заданный уровень погрешности вычислений при $x=1$ не превышает, если расчет функции осуществляется на основании 4 членов ряда.

На основании табл. 1.3 находим погрешности для остальных значений аргумента:

$x=2, \quad i=6, \quad \Delta_1 = 0,004534,$	$x=6, \quad i=13, \quad \Delta_1 = 0,003636,$	
$x=3, \quad i=8, \quad \Delta_1 = 0,003835,$	$x=7, \quad i=15, \quad \Delta_1 = 0,002419,$	
$x=4, \quad i=10, \quad \Delta_1 = 0,00283,$	$x=8, \quad i=16, \quad \Delta_1 = 0,003722,$	
$x=5, \quad i=12, \quad \Delta_1 = 0,00201,$	\dots	
	$x=12, \quad i=22, \quad \Delta_1 = 0,0030646.$	(1.7)

Таким образом, ввод данных в спецкомпьютер должен сопровождаться обращением к служебной таблице вида (1.7) с определением числа вычислительных циклов, необходимых для расчета функции e^x для конкретного x . В простейшем случае число членов ряда выбирается равным 22.

Погрешность, вносимая ограниченностью разрядной сетки (с учетом бесконечной длины значения функции), будет определяться разностью между эталонным значением e^x , в нашем случае равным e^{12} , и значением, размещаемым в 29-разрядной сетке компьютера:

$$\Delta_2 = \frac{e^{12} - e_{M=29}^{12}}{e^{12}} \cdot 100\% =$$

$$= \frac{162754,79142... - 162754,791}{162754,79142...} \cdot 100\% = 0,0006147 \%$$
(1.8)

Из (1.8) следует, что $\Delta_{\Sigma} = \Delta_2 + \Delta_1 \approx \Delta_1$, т.е. Δ_2 может не учитываться как погрешность для данного класса функций.

В целом для решения каждой конкретной задачи может быть использовано несколько схем расчета. При этом для каждого алгоритма определяется точность представления результатов и максимальное время вычисления наиболее неблагоприятного параметра. Предпочтение конкретному способу решения задачи отдается в зависимости от поставленной цели управления.

1.5. Проектирование системы команд

В общем случае в спецкомпьютерах выделяют следующие группы алгоритмов, ориентированных на задачи управления подвижной платформой:

- 1) методы решения пользовательских задач;
- 2) методы решения служебных задач;
- 3) алгоритмы решения специальных задач.

Пользовательские задачи включают в свой состав:

- 1) задачи управления бортовыми системами;
- 2) задачи комплексной обработки входной информации;
- 3) задачи организации взаимодействия в локальной сети или между подсистемами управления агрегатами и механизмами подвижной платформы.

Служебное программное обеспечение, как правило, реализует алгоритмы решения более локальных задач. К ним относят:

- 1) алгоритмы диспетчеризации и прерываний вычислений;
- 2) алгоритмы защиты программ и исправления ошибок;
- 3) методы контроля работоспособности бортовых систем и другие утилиты.

К специальным задачам относят задачи имитации исходных данных для моделирования процессов передвижения используемой платформы, а также для решения задач тактического характера.

Анализ параметров алгоритмов, как правило, выполняется с использованием языковых и программных средств. С этой целью каждой вершине ГСА решаемой задачи ставится в соответствие команда машины. После этого выполняется расширение полученного набора команд с использованием заданных методов адресации и варьированием полей КОП. Полученная система дополняется командами контроля и диагностики, управления работой компьютера, командами ввода-вывода и другими управляющими словами, позволяющими получить требуемые режимы работы компьютера.

Рассмотрим показанную на рис. 1.6 блок-схему алгоритма расчета функции e^x в соответствии с формулой (1.1) и исследуем вычислительный аспект поставленной задачи компьютера.

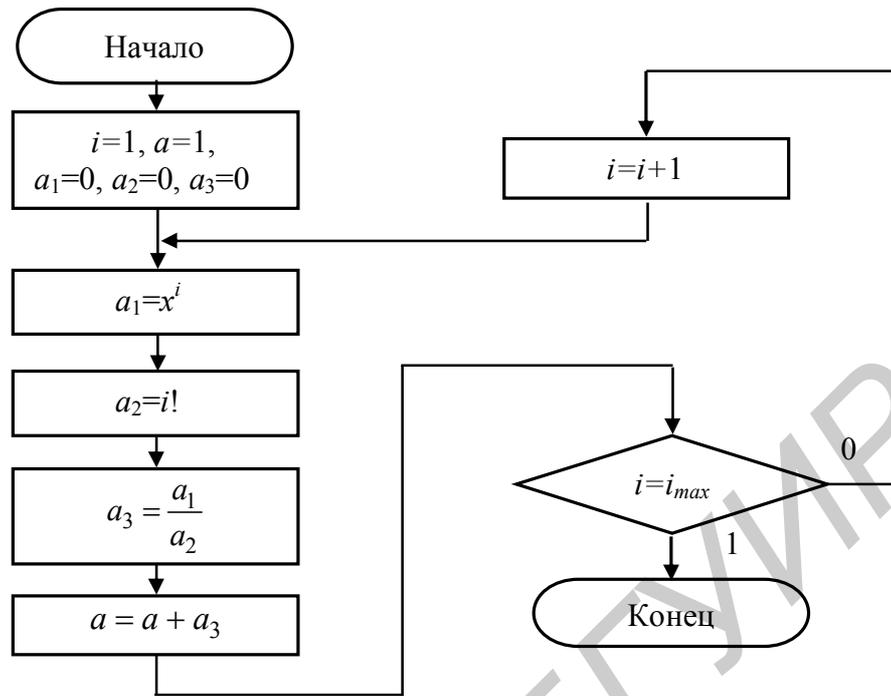


Рис. 1.6

Основной недостаток алгоритма вычисления e^x заключается в сложности и длительности формирования результата с использованием команд

$$\begin{aligned} \text{Ifac}(i) &= i!, \\ \text{Degr}(x, i) &= x^i. \end{aligned}$$

С целью сокращения повторных вычислений при расчете факториалов и степенных функций выполним модернизацию представленного алгоритма, учитывая, что расчеты на каждом i -м шаге могут быть преобразованы к виду:

	$a_1 = x^i$	$a_2 = i!$	$a_3 = \frac{x^i}{i!}$
$i = 1.$	x^1	$1!$	$\frac{x^1}{1!}$
$i = 2.$	$x^1 x = x^2$	$1! \cdot 2 = 2!$	$\frac{x^1}{1!} \cdot \frac{x}{2} = \frac{x^2}{2!}$
$i = 3.$	$x^2 x = x^3$	$2! \cdot 3 = 3!$	$\frac{x^2}{2!} \cdot \frac{x}{3} = \frac{x^3}{3!} \dots$

Иными словами, в каждом i -м цикле вычислений легко прослеживается зависимость

$$a_3(i) = a_3(i-1) \cdot \frac{x}{i}. \quad (1.9)$$

Теперь с учетом равенства (1.9) модернизируем граф-схему алгоритма (см. рис. 1.6) и поставим в соответствие всем вершинам ГСА команды компьютера.

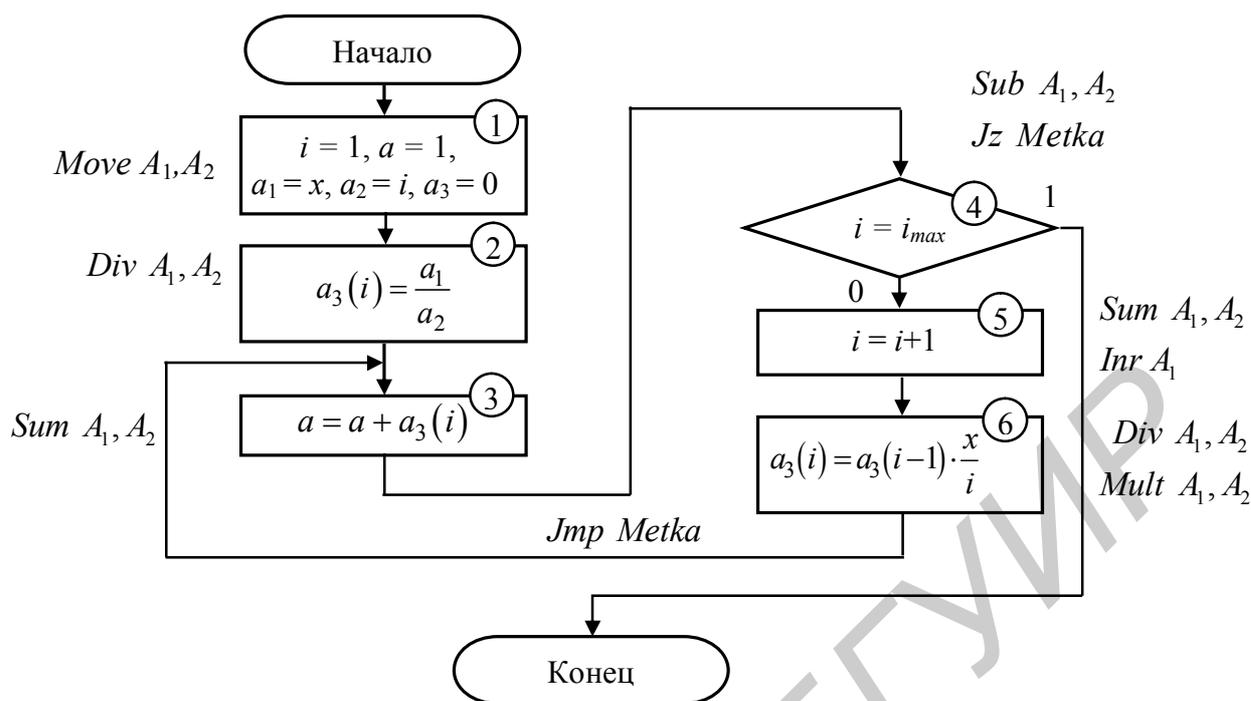


Рис. 1.7

В полученной выше граф-схеме удалось избежать долговременных повторных вычислений в командах $Ifac(i)$ и $Degr(x, i)$ и заменить их на множитительно-делительные преобразования, свойственные практически всем высокоуровневым языкам компьютера. Общая таблица команд для вычисления e^x представлена ниже:

Таблица 1.4

№	Команда	№	Команда
1	$Move A_1, A_2$	5	$Jz Metka$
2	$Div A_1, A_2$	6	$Mult A_1, A_2$
3	$Sum A_1, A_2$	7	$Inr A_2$
4	$Sub A_1, A_2$	8	$Jmp Metka$

Итак, базовая система управляющих инструкций процессора без учета команд служебного и специального ПО будет содержать 8 управляющих слов.

Выполним расширение системы команд спецкомпьютера, используя следующий (заданный) набор методов адресации операндов в памяти:

- 1) прямая адресация;
- 2) косвенная регистровая адресация;
- 3) автоинкрементная адресация;
- 4) базово-индексная адресация.

В данном случае расширенная система команд проектируемого компьютера может быть представлена следующими форматами.

Команды пересылки приведены на рис. 1.8.



Рис. 1.8

В приведенных форматах команд имеет место следующая интерпретация полей: КОП – поле кода операции базовой команды; *mod* – модификатор команды, определяющий способ использования регистров общего назначения при адресации ОЗУ; R_i – регистр источник и приемник пересылаемого операнда; R_j и R_x – регистры косвенной адресации. Команды I₁–I₃ представляют собой команды формата RS, команда I₄ – команду формата RX. Для полноты картины набор управляющих слов компьютера дополнен командой I₅ формата RR.

Команда деления имеет следующие форматы:



Рис. 1.9

Для составления подмножеств команд суммирования *Sum*, вычитания *Sub* и умножения *Mult* может быть использована аналогичная методология,

что приводит к образованию управляющих слов с номерами: $I_{11}-I_{25}$.

Команды перехода Jz и Jmp (I_0), применяемые при вычислении функции $y = e^x$, очевидно, являются одноадресными. В связи с этим нет необходимости включения в состав команд данной группы поля первого операнда R_i , а все множество управляющих слов в данной группе сократить до трех и представить тремя форматами (рис. 1.10).

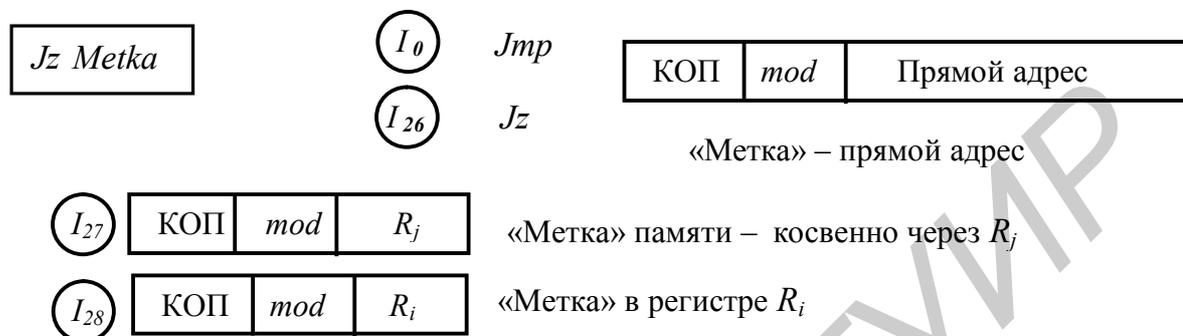


Рис. 1.10

Команда Inr представляется одним одноадресным форматом с явным указанием регистра (рис. 1.11).

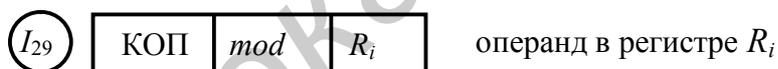


Рис. 1.11

Кроме использованных в граф-схеме алгоритма команд, компьютерные программы предполагают также реализацию процедур ввода-вывода информации в различных формах. При этом могут быть использованы команды с прямой или косвенной адресацией портов, а также с неявной или непосредственной адресацией операндов. В целом учитывая, что ввод-вывод достаточно сложная процедура и может быть организован как на программном, так и на микропрограммном уровне, данный вопрос подробно будет рассмотрен в соответствующем разделе ниже.

При создании системы команд в общем случае следует помнить, что:

1) в системе команд должны быть учтены все управляющие слова основной ГСА, определяющей главные вычислительные алгоритмы пользователя. В реальных спецсистемах эти команды в основном порождаются алгоритмами управления бортовой системой при расчете координат полета или наведении подвижной платформы на цель [1];

2) система команд должна быть дополнена командами ввода-вывода и командами управления режимами работы компьютера. Порождающей основой здесь являются алгоритмы комплексной обработки входной информации, задачи организации взаимодействия в локальной сети или между подсисте-

мами и механизмами платформы, задачи визуализации и автоподдержания параметров среды на борту носителя;

3) алгоритмы решения служебных задач могут потребовать применения специальных команд, не рассмотренных выше в пп. 1 и 2. При этом базой для создания управляющих слов являются: задачи диспетчеризации и прерывания вычислений, алгоритмы защиты программного обеспечения и исправления ошибок, методы тестирования и контроля работоспособности бортовых систем;

4) система команд может быть дополнена за счет решения специальных задач, к которым относят задачи моделирования процессов передвижения (полета) подвижной платформы, решение задач тактического характера [1].

1.6. Кодирование системы команд

При кодировании системы команд в рассматриваемом примере (рис. 1.7) поле управления КОП принципиально может быть кодировано триадами в диапазоне 000–111. Однако необходимость применения спецкоманд компьютера для реализации служебных и специальных функций приводит к необходимости расширения соответствующего поля с учетом общего числа управляющих слов.

Будем считать, что общее число команд проектируемого учебного устройства не превышает 15, в связи с чем разрядность кода операции может быть принята равной четырем битам.

Для каждой операции в данном компьютере определено пять разновидностей адресации. Следовательно, модификатор адресных регистров *mod* должен иметь разрядность, равную трем битам. При этом полная длина расширенного поля КОП будет иметь 7 разрядов.

Разрядность полей-указателей регистров выбирается по числу регистров процессорного РЗУ и в нашем случае выбирается равной 4. Данное значение соответствует архитектуре большинства известных процессоров.

Форматы команд, доработанных с учетом длины полей, могут быть представлены в следующем виде (рис. 1.12).



Рис. 1.12

Будем считать, что длина команды не может превышать разрядности машинного слова. Следовательно, разрядность прямого адреса в первом формате может составлять не более $36-11 = 25$ бит. Аналогично, разрядность смещения в четвертом формате не может превышать $36-19 = 17$ бит.

Порядок кодирования управляющей информации в командах компьютера I_0-I_{33} ... приведен в табл. 1.5.

Таблица 1.5

№ команды	КОП	Код	Тип	<i>mod</i>	Примечание
I_0	<i>Jmp</i>	0000	<i>RS</i>	000	Безусл. переход
$I_1 - I_5$	<i>Move</i>	0001	<i>RS</i>	000	Команды пересылки
		0001	<i>RS</i>	001	
		-	-	-	
		0001	<i>RR</i>	100	
$I_6 - I_{10}$	<i>Div</i>	0010	<i>RS</i>	000	Команды деления
		0010	<i>RS</i>	001	
		0010	<i>RS</i>	010	
		0010	<i>RX</i>	011	
		0010	<i>RR</i>	100	
$I_{11} - I_{15}$	<i>Sum</i>	0011	<i>RS</i>	000	Команды сложения
		0011	<i>RS</i>	001	
		0011	<i>RS</i>	010	
		0011	<i>RX</i>	011	
		0011	<i>RR</i>	100	
$I_{16} - I_{20}$	<i>Sub</i>	0100	<i>RS</i>	000	Команды вычитания
		0100	<i>RS</i>	001	
		0100	<i>RS</i>	010	
		0100	<i>RX</i>	011	
		0100	<i>RR</i>	100	
$I_{21} - I_{25}$	<i>mult</i>	0101	<i>RS</i>	000	Команды умножения
		0101	<i>RS</i>	001	
		0101	<i>RS</i>	010	
		0101	<i>RX</i>	011	
		0101	<i>RR</i>	100	
$I_{26} - I_{28}$	<i>Jz</i>	0110	<i>RS</i>	000	Команды условных переходов
		0110	<i>RS</i>	001	
		0110	<i>RS</i>	010	
I_{29}	<i>Inr</i>	0111	<i>RR</i>	000	Инкремент
$I_{30} - I_{31}$	<i>In</i>	1000	<i>RS</i>	000	Команда ввода-вывода (операнд в аккумуляторе, адресация порта прямая или через РЗУ)
		1000	<i>RS</i>	001	
$I_{32} - I_{33}$	<i>Out</i>	1001	<i>RS</i>	000	
		1001	<i>RS</i>	001	
$I_{34} - \dots$	\dots	1010	\dots	\dots	Спецкоманды

1.7. Переход от системы команд к блок-схеме компьютера

Рассмотрим неймановский компьютер с типичной архитектурой, т.е. со структурой и языком, обладающими основными чертами автоматического вычислительного устройства. По своей сути блок-схема проектируемого компьютера будет представлять собой четырехблочную систему с принстонской архитектурой, шинная организация которой определяется совмещенной шиной данных и команд, как показано на рис. 1.13.

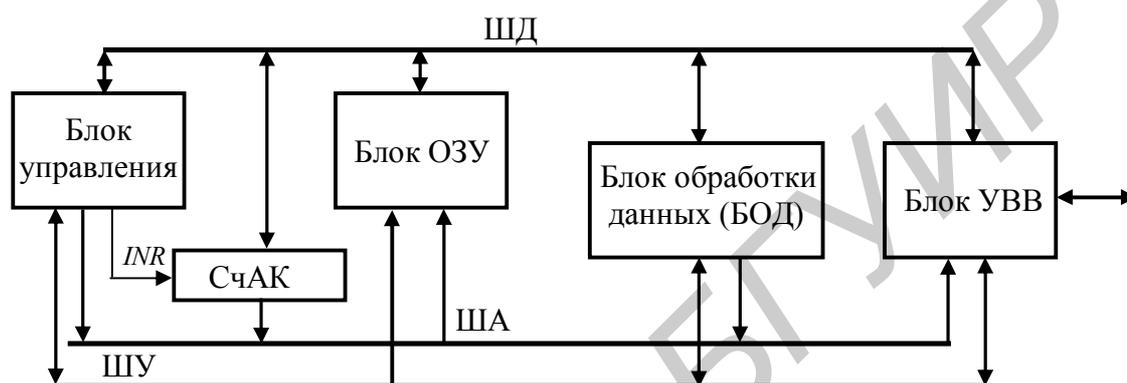


Рис. 1.13

Шинные связи между компонентами блок-схемы спецкомпьютера в основном определяются фазами выборки и исполнения команд.

Принципиально фаза выборки начинается с выдачи соответствующего адреса из СчАК на шину адреса системы. При этом текущее значение счетчика определяется естественным способом формирования адресов (используется линия *INR*) или алгоритмом отработки команд перехода. Последний, очевидно, и предполагает наличие двунаправленной связи между блоком управления, ШД и входом счетчика команд. На втором шаге контроллер ОЗУ, входящий в данной структуре в блок ОЗУ, принимает адресное слово, делит его на адреса строки и столбца и передает на адресный вход памяти. Блок управления включает память в режим чтения и принимает по ШД команду (или только ее первый байт) в свой внутренний регистр команд *RG K*. На этом фаза выборки завершается.

Фаза исполнения включает в себя четыре шага. На первом шаге осуществляется декодирование кода операции и распознавание общей длины управляющего слова. В результате выясняется последовательность действий, необходимая для выполнения преобразований и, если требуется, с использованием системной шины дочитывается оставшаяся часть команды.

На втором шаге совместные действия БУ (или БМУ – блока микропрограммного управления) и БОД позволяют сформировать адреса операндов и передать считанные данные во внутренние регистры БОД для обработки.

Третий шаг – это преобразование данных с учетом кода операции. При этом БМУ формирует сигналы управления для БОД и определяет характер

действий, необходимых для получения результата. Арифметикологическое устройство на основе сформированного результата формирует признаки (или флаги), которые передаются в устройство управления и далее используются для ветвления вычислительного процесса.

На четвертом шаге осуществляется запись результата в ОЗУ или другой приемник в соответствии с алгоритмом обработки текущей команды. В целом перечисленные действия определяют наличие двунаправленных линий связи между блоками и устройствами проектируемого компьютера. Последней микрооперацией фазы исполнения является микрооперация инкрементирования счетчика команд, после чего блок управления вновь реализует фазу выборки.

Для формирования более подробной блок-схемы компьютера выделим из блока управления регистр команды (RGK) в виде отдельной структурной компоненты и разместим в нем команду первого формата, например I_1 . При этом, устанавливая связи между RGK , блоками и модулями системы, добиваемся логического объединения воедино формата команды, аппаратных средств и принципа функционирования неймановской архитектуры (рис. 1.14). В приведенной структуре поля команды КОП и mod передаются на вход управляющего автомата и определяют механизм адресации операндов и алгоритм их преобразования. Блок управления при этом формирует сигналы микрокоманды I для БОД, УВВ и ОЗУ, а на свой вход принимает дополнительно признаки результата от процессора и сигналы квитирующих пар от УВВ. Адрес регистра РЗУ процессора определяет необходимость выделения в БОД входной адресной шины A , разрядность которой должна быть равна разрядности поля R_j . Иными словами, считаем, что процессор имеет 16 РОН, а поле адреса – длину 4 бита.

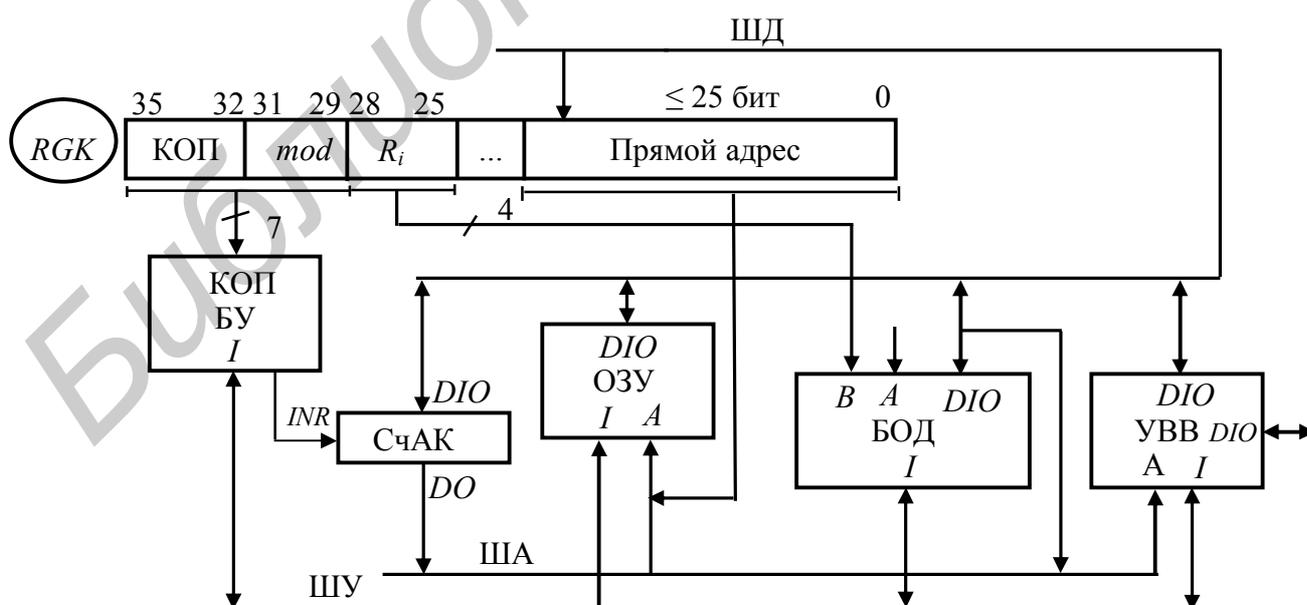


Рис. 1.14

В рамках совершенствования внутреннего интерфейса следует учитывать, что адресный вход памяти может принимать информацию от двух источников: от регистра команды RGK и из счетчика СчАК. Это предполагает включение в соответствующей точке интерфейса мультиплексора MUX с управлением от БМУ. Кроме того, работа ОЗУ, БОД и УВВ на общую шину данных также предполагает введение дополнительных линий для управления двунаправленными выводами DIO с целью арбитража ШД, т.е. общей шины компьютера.

Эмуляция в структуре спецкомпьютера следующих двух форматов, а именно команд $I_2 - I_3$ предполагает введение в сформированную блок-схему (см. рис. 1.14) дополнительных линий связи, как показано на рис. 1.15.

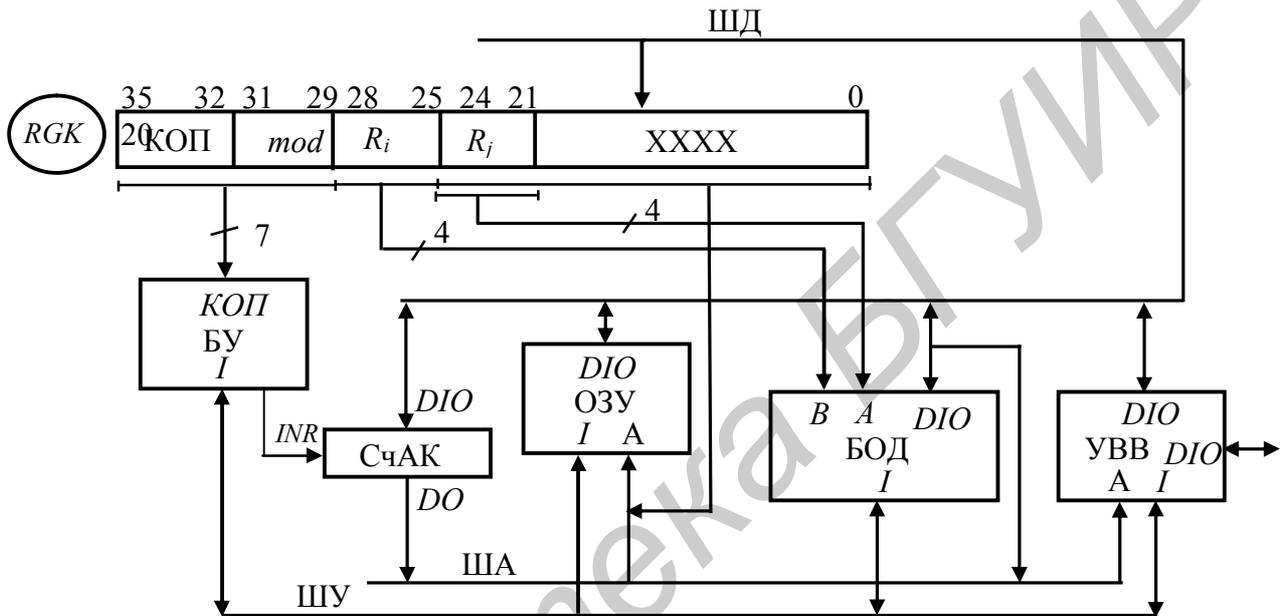


Рис. 1.15

Окончательный вариант укрупненной блок-схемы компьютера может быть получен путем эмуляции в полученной структуре команды I_4 формата RX (рис. 1.16).

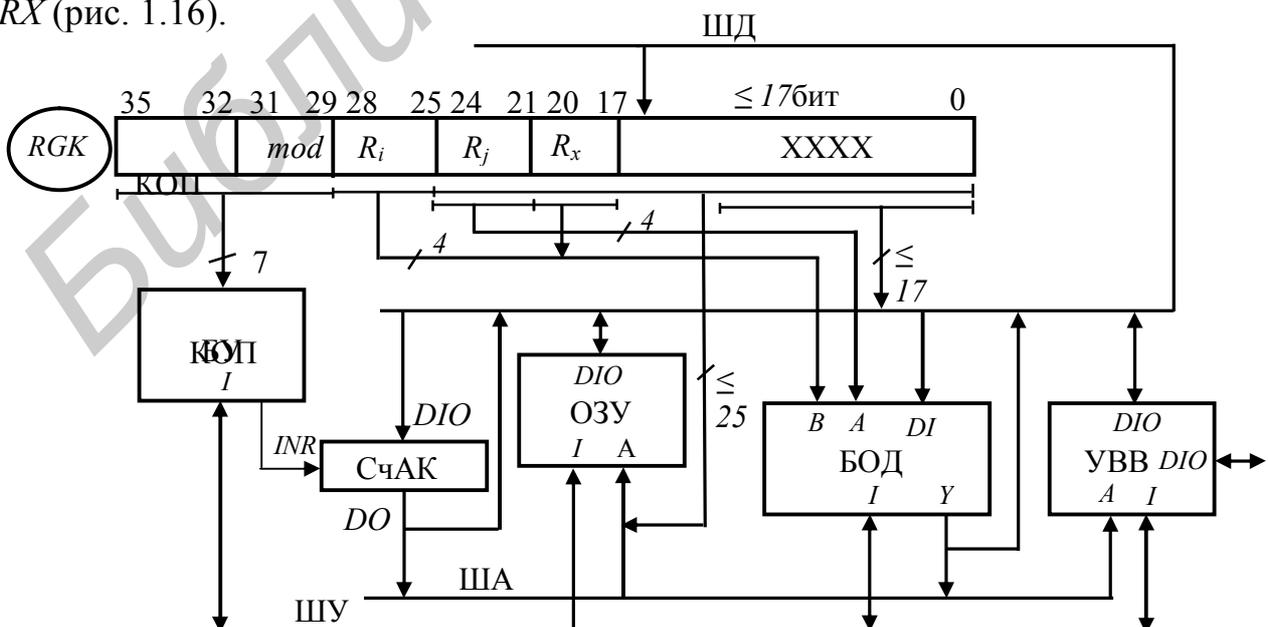


Рис. 1.16

На данном шаге эмуляции необходимо учитывать, что формирование адреса ОЗУ в четвертом формате осуществляется с использованием двух регистров процессора: R_j – регистра базы и R_x – регистра индекса. Данный факт предполагает одновременность передачи адресов регистров из $RG K$ на входы РЗУ A и B процессора.

Разряды смещения передаются по дополнительным линиям связи на ШД и далее через схемы коммутации компьютера на DI вход БОД. Таким образом, вычисление адреса операнда в ОЗУ может быть выполнено в течение только двух тактов работы системы. После данной процедуры на адресный вход B БОД может быть скоммутирован адрес второго операнда.

Очередной шаг совершенствования структуры, приведенной на рис. 1.16, может быть выполнен после расчета и синтеза функциональных блоков и соответствующих линий связи внутреннего интерфейса компьютера.

1.8. Расчет компьютерной памяти и схемы управления адресами

Будем считать, что по условиям задачи проектирования спецкомпьютер требует синтеза подсистемы оперативного ЗУ емкостью 4096 Кбайт. И пусть в качестве элементной базы задана интегральная схема ОЗУ K565PY7 с организацией запоминающего массива $256 \text{ Кбит} \times 1$. При этом (для простоты архитектурного проектирования) будем считать, что технология изготовления в дальнейшем используемых БИС серии K1804 и ИМС памяти идентичны и позволяют в условиях выбранной частоты тактирования получить стыкуемые подсистемы компьютера.

В соответствии со справочными данными выбранный модуль памяти содержит: одноразрядную входную DI и выходную DO шины, линии адреса, образующие 9-разрядную локальную ША, входы стробирования адресов строки \overline{RAS} и столбца \overline{CAS} , линию управления записью чтением $\overline{W/R}$. Модуль реализует функции записи, хранения, чтения и регенерации информации в соответствии с табл. 1.6.

Таблица 1.6

\overline{RAS}	\overline{CAS}	$\overline{W/R}$	A	DI	DO	Режим работы
1	1	X	X	X	z	Хранение
1	0	X	X	X	z	Хранение
10	0	X	X	X	z	Реген.(512)
0	0	0	A	0	z	Запись 0
0	0	0	A	1	z	Запись 1
0	0	1	A	X	D	Считывание

В структурную схему ИМС памяти входят выполненные на одном кристалле: матрица накопителя с $262 \ 144$ элементами памяти, расположенными на пересечениях 512 строк и столбцов, 512 усилителей считывания и регенерации, дешифраторы строк и столбцов, устройство управления и два регистра адреса.

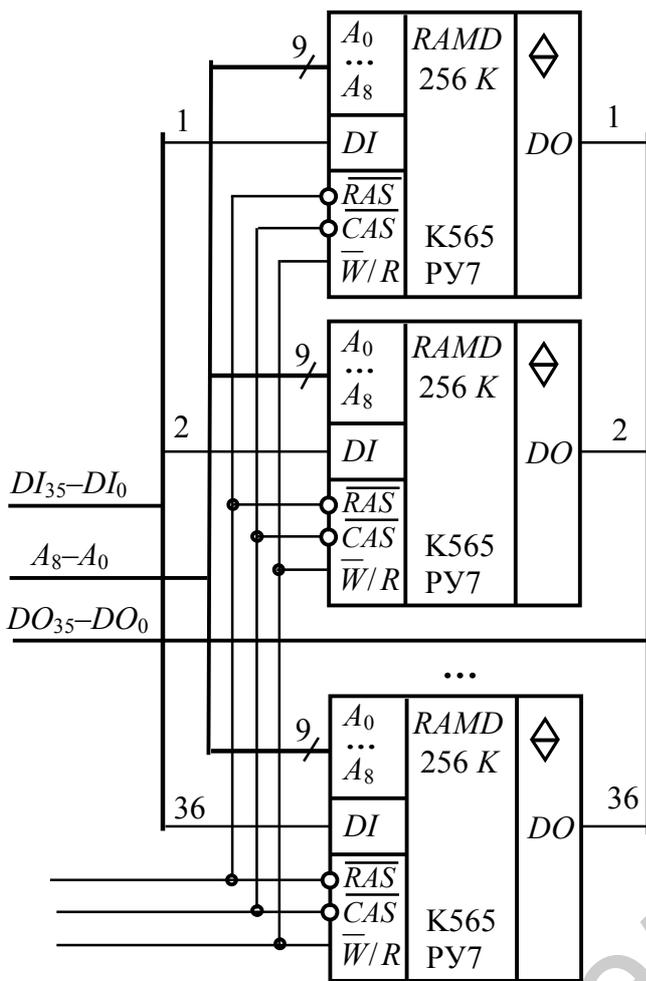


Рис. 1.17

Минимальная емкость памяти, используемая компьютером с 36-разрядной шиной команд/данных, определяется общей емкостью 36 ИМС заданного модуля ОЗУ. В нашем случае эта величина составляет ровно $256 \text{ Кбит} \times 36 = 9216 \text{ Кбит}$ или 1152 Кбайт. Соответствующая организация такой подсистемы получила название банка памяти и показана на рис. 1.17. Обозначение банка, применяемое на структурных схемах, приведено на рис. 1.18.

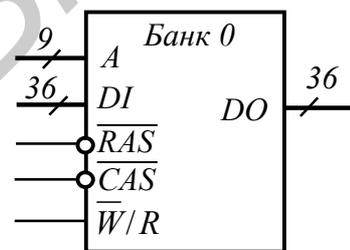


Рис. 1.18

Общее количество банков, составляющих подсистему ОЗУ (в соответствии с заданной емкостью памяти 4096 Кбайт), рассчитывается путем деления заданной емкости ЗУ на емкость 1 банка: $\frac{4096}{1152} = 3,55 \approx 4$ банка.

При формировании блока ОЗУ в проектируемой структуре необходимо использовать мультиплексор адреса, системная функция которого состоит в преобразовании 18 младших разрядов адресной шины в 9-разрядные адреса строки и столбца. Запись составляющих адреса во внутренние регистры ИМС осуществляется в соответствии с табл. 1.7.

Для стробирования соответствующего банка памяти по входу \overline{RAS} в схему ОЗУ включен мультиплексор $MUX RAS$. К первой группе входов указанного модуля подключены инверсные выходы дешифратора выборки банка $DC RAS$, а ко второй группе входов – выходная шина генератора импульсов, предназначенного для регенерации памяти.

Стробирование ЗУ по входу \overline{CAS} осуществляется с использованием $MUX CAS$. При этом первая группа входов принимает унитарный код с инверсных выходов дешифратора $DC CAS$, что позволяет выбрать требуемый

банк памяти, а вторая принимает нулевой уровень с шины «земля», необходимый для реализации функции регенерации. Для включения дешифраторов в активный режим используются входы w , позволяющие при нулевом уровне входного сигнала переводить выходы дешифратора в единичное состояние.

В целом для управления процессами обращения к памяти и регенерации предполагается использовать шину управления регистра микрокоманды.

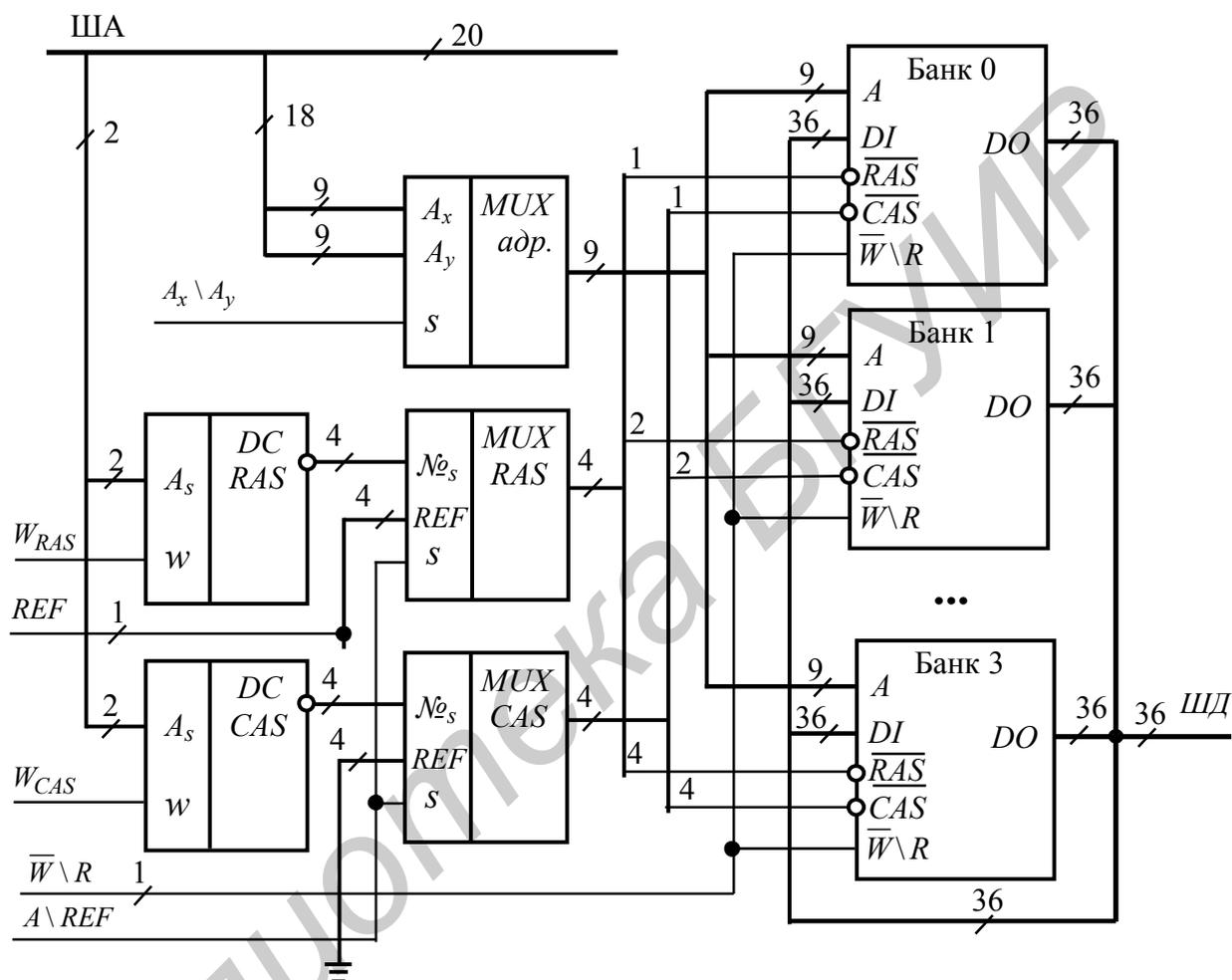


Рис. 1.19

Для хранения служебных программ, таких, например, как «Загрузчик», «Тест ОЗУ» и т.д., а также для хранения таблиц перекодирования данных или констант системного назначения в структуру ОЗУ включается блок ПЗУ (512 слов). Активация ПЗУ выполняется с помощью сигнала S_w ШУ, подаваемого на вход \overline{CS} при переводе ОЗУ в режим регенерации. Реализация подсистемы памяти (рис. 1.19) в виде модуля с соответствующими шинами управления, а также механизм включения ПЗУ в адресное пространство ОЗУ, показаны на рис. 1.20.

Управляющая информация из табл. 1.7 может быть использована для микропрограммирования задач, связанных с обработкой данных, хранимых в блоке ОЗУ. Указанные разряды включаются в состав регистра микрокоманды $RGMk$ в разряды 4...0. Сигнал управления ПЗУ S_w определяется как $RGMk$ [5].

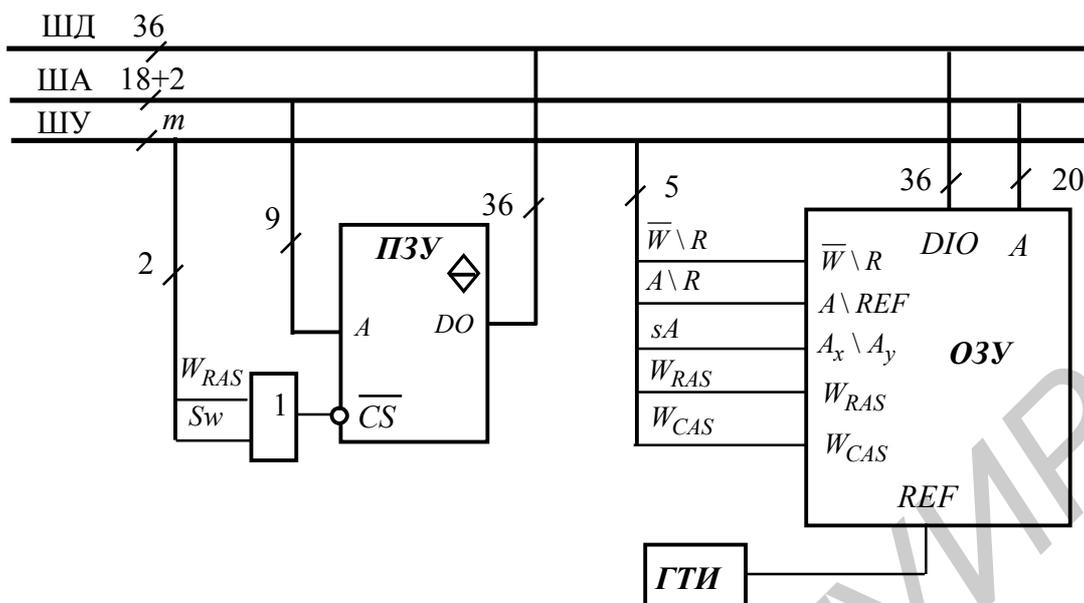


Рис. 1.20

Таблица 1.7

$\bar{W} \setminus R$	$A \setminus R$	sA	W_{RAS}	W_{CAS}	Комментарий
X	1	X	X	X	Регенерация
1	0	0	0	0	Переход к адресации
1	0	0	1	0	Запись адреса строки
1	0	1	1	1	Запись адреса столбца
0	0	1	1	1	Чтение ОЗУ

Полученная в данном параграфе структура компьютерной памяти позволяет уточнить форматы двух команд, ориентированных на обращение к ОЗУ.

Формат команд I_1 , доработанный с учетом длины поля прямого адреса, представлен на рис. 1.21.

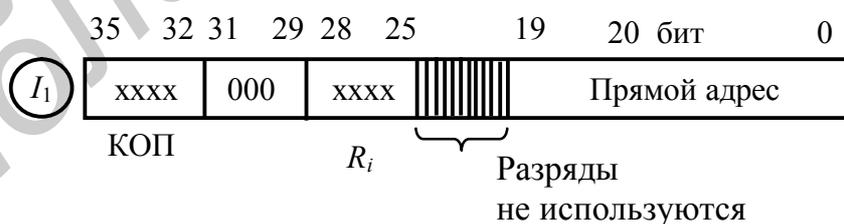


Рис. 1.21

Будем считать, что для решения задач управления память компьютера разделена на 8 равных базовых областей, причем каждая область содержит по 512 страниц объемом 256 машинных слов. Таким образом, разрядность поля «смещение» в команде формата I_4 (рис. 1.22) должна быть выбрана равной 9 битам. Исполнительный адрес при этом определяется формулой

$$A_{исп} = \langle R_j \rangle + \langle R_x \rangle + R G K [8, 0].$$

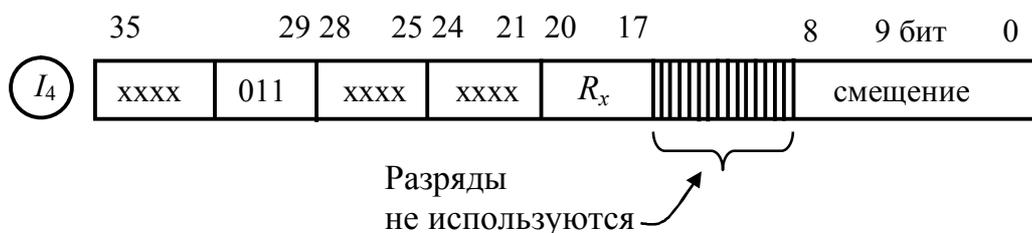


Рис. 1.22

Несмотря на то, что в командах приведенного формата не используется 8 разрядов слова, преобразование структуры ОЗУ к байтовой памяти выполнять нецелесообразно. Это обусловлено тем, что время выборки команд и данных существенно возрастает, если формат команды оказывается больше длины машинного слова. Данный факт существенно снижает производительность проектируемой системы, хотя пространство ОЗУ при этом используется более экономно.

Доработка блок-схемы компьютера, приведенной на рис. 1.16, с учетом подсистемы памяти (рис. 1.19 и 1.20) приведена на рис. 1.23.

Заметим, однако, что в случае использования процессоров с низким быстродействием относительно более быстродействующего ОЗУ, что свойственно учебным компьютерам, необходимо разрабатывать контроллер памяти, ускоряющий фазу выборки информации на системную шину. Принцип работы такого устройства состоит в следующем.

В состав регистра микрокоманды включается разряд *MEM*, который идентифицирует процесс обращения к памяти. Назначение данного сигнала состоит в запуске микроавтомата, показанного на рис. 1.24, формирующего временную диаграмму цикла обращения к ОЗУ. В качестве основы данного устройства выбран быстродействующий счетчик Грея *CTGr* с комбинационной схемой КС, на выходе которой генерируются требуемые сигналы управления. В момент запуска данного цифрового модуля сигнал микрокоманды $MEM = 1$ переключает специальный триггер схемы в состояние нуля. При этом считается, что исходное состояние триггера единичное, а блок памяти компьютера находится в режиме регенерации. Инверсный выход триггера разрешает прохождение импульсов счета от ГТИ на вход счетчика, при этом блок памяти (прямым плечом триггера) переводится в рабочий режим.

В процессе переключения разрядов счетчика происходит перебор его состояний с изменением одного бита. Это позволяет сформировать на выходе КС требуемую последовательность управляющих сигналов без сопутствующей генерации помех. Очевидно, что синтез КС осуществляется в соответствии с таблицей истинности, построенной на основе алгоритма адресации динамического ОЗУ. В частности, если счетчик находится в состоянии 000, то сигнал $A_x | A_y$ будет равен 0, что позволяет коммутировать адрес строки на вход памяти.

Остальные сигналы в данный момент времени неактивны (табл. 1.8).

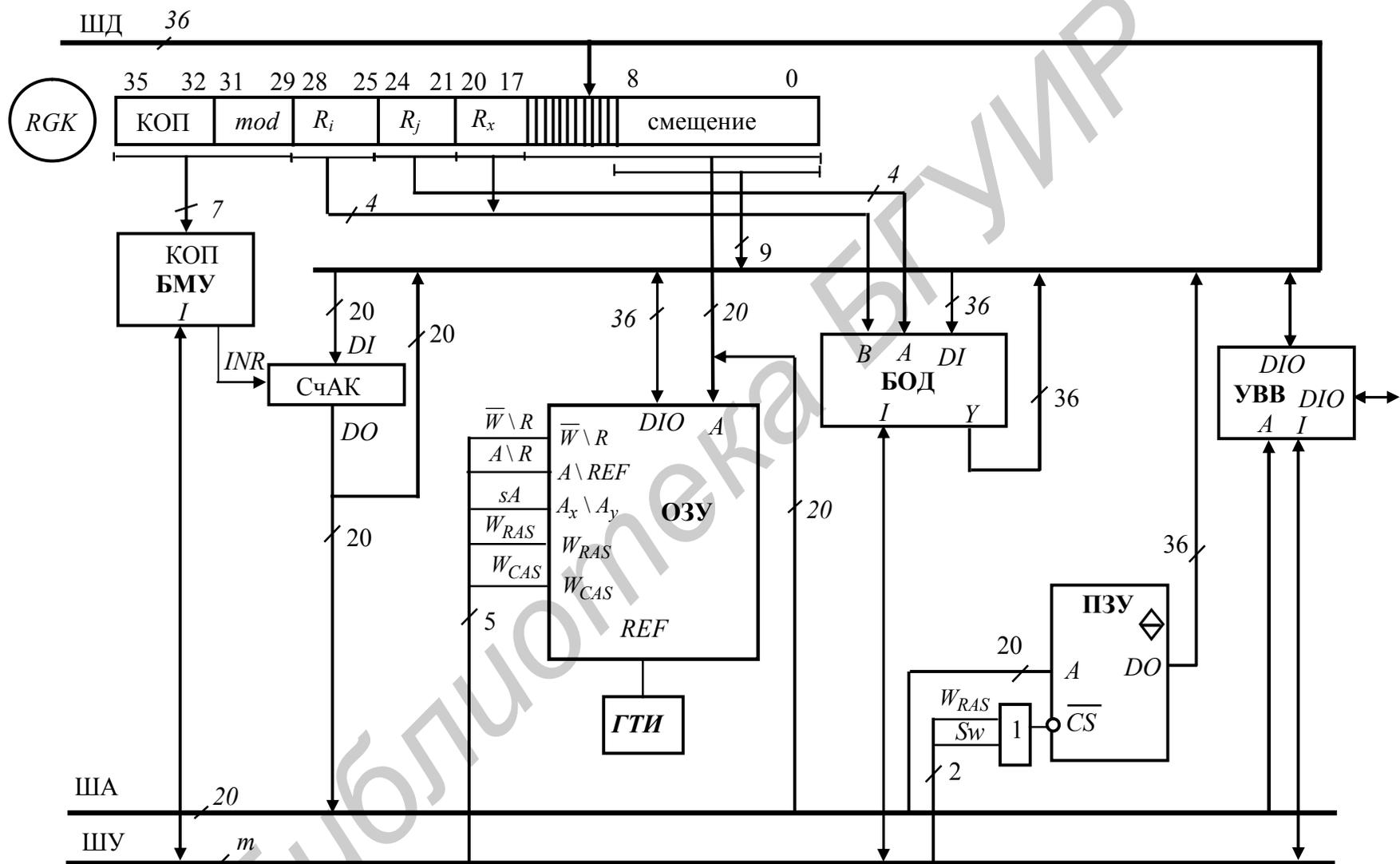


Рис. 1.23

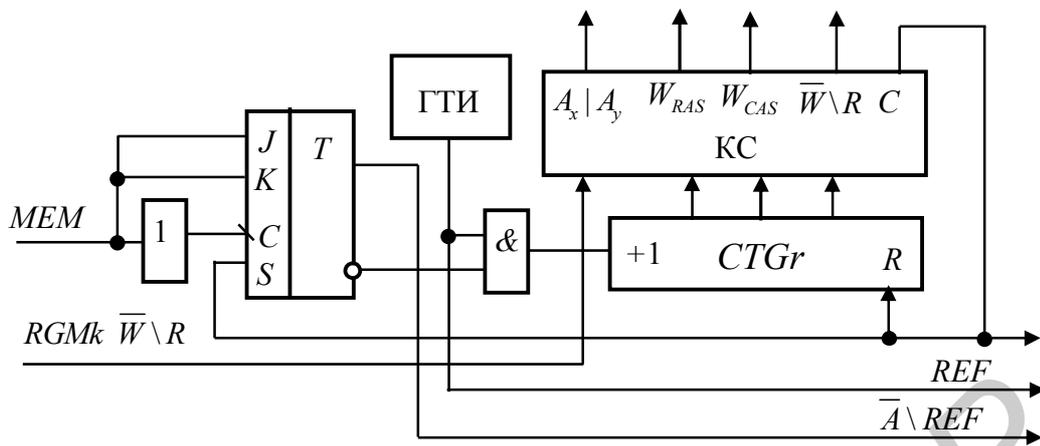


Рис. 1.24

Переключение счетчика в состояние 001 эквивалентно второму такту формирования временной диаграммы чтения/записи. При этом нулевое состояние сигнала $A_x|A_y$ сохраняется, т.е. адрес строки удерживается на адресных входах ОЗУ, а на выходе W_{RAS} формируется единичный уровень, разрешающий стробирование адреса по входу \overline{RAS} .

Таблица 1.8

CTGr 2 ² 2 ¹ 2 ⁰	RGMk $\overline{W}\backslash R$	$A_x A_y$	W_{RAS}	W_{CAS}	$\overline{W}\backslash R$	C
0 0 1	$\overline{W}\backslash R$	0	1	0	0	0
0 1 1		1	1	0	0	0
0 1 0		1	1	0	$\overline{W}\backslash R$	0
1 1 0		1	1	1	$\overline{W}\backslash R$	0
1 1 1		1	1	1	$\overline{W}\backslash R$	1

На третьем этапе $CTGr = 011$ сигнал $A_x|A_y$ переключается в единицу и на вход ОЗУ коммутируется адрес столбца. При этом сигнал $W_{RAS} = 1$, а W_{CAS} удерживается в «0». На четвертом такте временной диаграммы $CTGr = 010$, что позволяет передать сигнал регистра микрокоманды $\overline{W}\backslash R$ на соответствующий вход памяти.

На пятом этапе счетчик $CTGr$ устанавливается в состояние 110 и сигнал W_{CAS} также переключается в единицу, т.е. разрешает стробирование адреса столбца по входу \overline{CAS} . В результате, весь адрес памяти оказывается записан во внутренний регистр модуля ОЗУ. В схеме реализуется рабочий режим.

На этом процедура обращения к ОЗУ завершается, счетчик $CTGr$ переключается в состояние 111 и вновь сформированный сигнал $C = 1$ по цепи обратной связи переключает триггер управления в единицу. В схеме восста-

навливается режим регенерации. Дополнительной функцией сигнала $C = 1$ является функция записи выходного слова памяти в регистр выходных данных, если реализуется режим чтения.

В дальнейшем в ходе проектирования будем использовать подсистему ОЗУ, приведенную на рис. 1.23, так как данное включение блока в состав компьютера представляется классическим. Напомним, что в приведенной схеме технологии изготовления всех подсистем проектируемого устройства считаются адекватными друг другу с точки зрения быстродействия применяемых ИМС.

Последующее формирование структуры спецкомпьютера будем вести с учетом известного принципа проектирования «сверху вниз». При этом системе, аналогично вышеизложенной методологии, будем развертывать в структурную схему путем перехода от абстрактных представлений о вычислениях к реализации соответствующего аппаратного и микропрограммного обеспечения. Подобная методология проектирования получила название поэтапного усовершенствования системы.

С практической точки зрения структуру блоков компьютера во многом определяют требования, предъявляемые к решениям задач, требования к скорости получения данных решений, принципам верификации результатов. Однако применение конкретных комплектов БИС предопределило стандартность многих архитектурных реализаций подсистем, что привело к необходимости эмуляции заданных вычислений в микропрограммируемых средах. Соответственно и проектирование БОД, БМУ и УВВ далее будем выполнять с учетом стандартных схем, определяемых справочной литературой, а также путем формирования внутреннего интерфейса системы, позволяющего получить компьютер с заданной проблемной ориентацией.

В целом последующие этапы проектирования спецкомпьютера следует отнести к этапу логического синтеза структуры. Данный этап предполагает разработку схемы, наиболее полно удовлетворяющей основным характеристикам, сформулированным на предыдущем этапе, т.е. этапе алгоритмического синтеза. При этом производится выбор типа компьютера – параллельного или последовательного, выбор алгоритмов реализации основных макрокоманд системы, выбор элементов и устройств внутреннего интерфейса – счетчиков, дешифраторов, мультиплексоров. Выходной продукцией этапа логического синтеза является структурная схема компьютера, включающая в себя основные модули системы, представленные на уровне структур, а в случае необходимости – представление модулей и на функциональном уровне.

Конечным этапом синтеза любого цифрового устройства является этап разработки принципиальных схем. При этом осуществляется выбор разъемов типовых элементов замены, прорабатываются связи между разъемами и используемыми ИМС, организуется шинный интерфейс между модулями и схемами подсистемы. Условные графические обозначения всех составляющих принципиальных схем, так же как и всех других компонентов проекта, должны соответствовать стандартам ЕСКД.

2. ПРОЕКТИРОВАНИЕ БЛОКА ОБРАБОТКИ ДАННЫХ (БОД) СПЕЦКОМПЬЮТЕРА

2.1. Проектирование БОД с использованием МПС К1804ВС1

Проектирование БОД управляющего спецкомпьютера сводится в основном к решению трех задач:

- 1) организации ускоренного (параллельного) переноса в многоразрядных устройствах,
- 2) организации заданных видов сдвигов операндов одинарной и двойной длины,
- 3) хранения слова состояния процессора и признаков состояния системы, поступающих от внутренних схем контроля.

Для решения поставленных задач будем использовать интегральные схемы комплекта К1804: ВС1, ВС2, ВР1, ВР2 и ИР1.

В процессе проектирования спецкомпьютера следует помнить, что функциональное назначение БОД состоит в решении ряда системных задач. Во-первых, это обработка данных под управлением сигналов регистра микрокоманды, во-вторых, хранение счетчика команд (необязательно) и указателя стека (считается, что стек расположен в ОЗУ) и, в-третьих, обработка адресной информации. При этом третья задача предполагает включение в БОД регистра адреса для гальванической развязки системных шин адреса и данных. Как правило, для реализации данной функции используется регистр К1804ИР1.

С целью получения высокой скорости решения задач функция системного счетчика команд (см. рис. 1.13) может быть возложена на внешний (по отношению к БОД) модуль, однако если время решения задачи не критично к циклу управления, то СчАК может быть реализован в РЗУ процессора, что влечет за собой более высокую технологичность схемы компьютера.

Рассмотрим принцип создания вычислительного устройства на базе процессорной секции К1804ВС1 [5,6].

Модульность блоков проектируемого компьютера предполагает наличие стандартных подходов к созданию подсистем обработки данных и анализа результатов вычислений на основе формируемых признаков. В частности, для решения задачи организации ускоренного переноса справочная литература дает стандартное соединение модулей для 16-разрядной вычислительной подсистемы *Prim.1*, показанное на рис. 2.1. Сформированный блок является составляющей частью многоразрядного процессора с параллельным переносом. В общем случае схема требует доработки в части соединения с другими модулями СУСС К1804ВР2 при каскадном объединении блоков *Prim.1*. Как самостоятельный компонент компьютера данный модуль не может быть использован, так как не содержит выхода переноса в явной форме. Особенностью схемы *Prim.1* является параллельное функционирование четырех секций под управлением общей шины микрокоманд $I_8 - I_0$. При этом входные линии DI об-

разуют 16-разрядную шину данных устройства, а выходные линии Y – 16-разрядную выходную шину. Управление трехстабильными Y -выходами осуществляется с помощью общего для четырех секций входа \overline{OE} .

Второй особенностью блока *Prim.1* является наличие общего для младшей МПС БОД и схемы СУП входа переноса C_0 . Аппаратно данный вход соединяется с выходом C_0 СУСС K1804BP2. Таким образом, при микропрограммировании различных функций состояние указанной линии определяется путем задания управляющей информации по входам $I_{12} - I_0$ модуля СУСС.

Для организации ускоренного переноса выводы $\overline{P}, \overline{G}$ младшей (правой крайней) МПС соединяются со входами $\overline{P}_0, \overline{G}_0$ СУП BP1. Соответствующий выход C_x подключается ко входу C_0 первой средней МПС. Аналогично выводы $\overline{P}, \overline{G}$ первой средней МПС соединяются со входами $\overline{P}_1, \overline{G}_1$ ИМС BP1, а соответствующий выход C_y подключается ко входу C_0 второй средней МПС. Данный принцип сохраняется и при подключении второй средней МПС к блоку СУП, а также для соединения СУП со старшей МПС (левой крайней) рассматриваемого модуля. Старшая МПС подключается ко входам СУП $\overline{P}_3, \overline{G}_3$, при этом выходами всего 16-разрядного модуля являются выходы схемы K1804BP1 $\overline{P}, \overline{G}$.

Выходы z всех секций выполнены по схеме с открытым коллектором, в связи с чем в устройстве используется монтажное «И» для их логического объединения с подключением к источнику питания через резистор 470 Ом.

Выводы F_3, OVR, C_4 в промежуточных звеньях процессорного блока не используются, так как идентифицируемая данными сигналами ситуация в середине машинного слова возникнуть не может.

Для проектирования многоразрядного процессора представим 16-разрядный блок в виде модуля со входом переноса C_0 и выводами каскадирования $\overline{P}, \overline{G}$. Простой подсчет разрядов БОД показывает, что конечная реализация процессора будет состоять из двух модулей *Prim.1* и дополнительной МПС, входящей в состав устройства в качестве старшей подсхемы.

Общая схема уровня блок-схемы процессорной подсистемы с учетом основных линий связи с блоком СУСС показана на рис. 2.2.

С целью сокращения внешних связей БОД или внутреннего интерфейса компьютера будем считать, что СчАК располагается в одном из регистров (РЗУ) процессора и управляется микропрограммно из *RGMk*.

Выходом схемы СУСС является системная линия CT , определяющая условие перехода в микропрограмме или программе пользователя.

Двунаправленные выводы YN, YV, YC, YZ являются транзитными выходами признаков процессора или входами сигналов компьютера, формируемых внутренними схемами прерываний [7].

Для реализации команд формата *RR* на программном и микропрограммном уровне в проектируемую схему БОД включены мультиплексоры адресов *A* и *B* РЗУ (рис. 2.3). При этом мультиплексор канала *A* имеет три информационных входа, что необходимо для выполнения команд с базово-индексной адресацией. Управление коммутаторами адресов организуется с использованием сигналов s_1, s_0 , формируемых регистром микрокоманды (рис. 2.4). Табл. 2.1 определяет принцип мультиплексирования шин в соответствии с указанными сигналами управления.

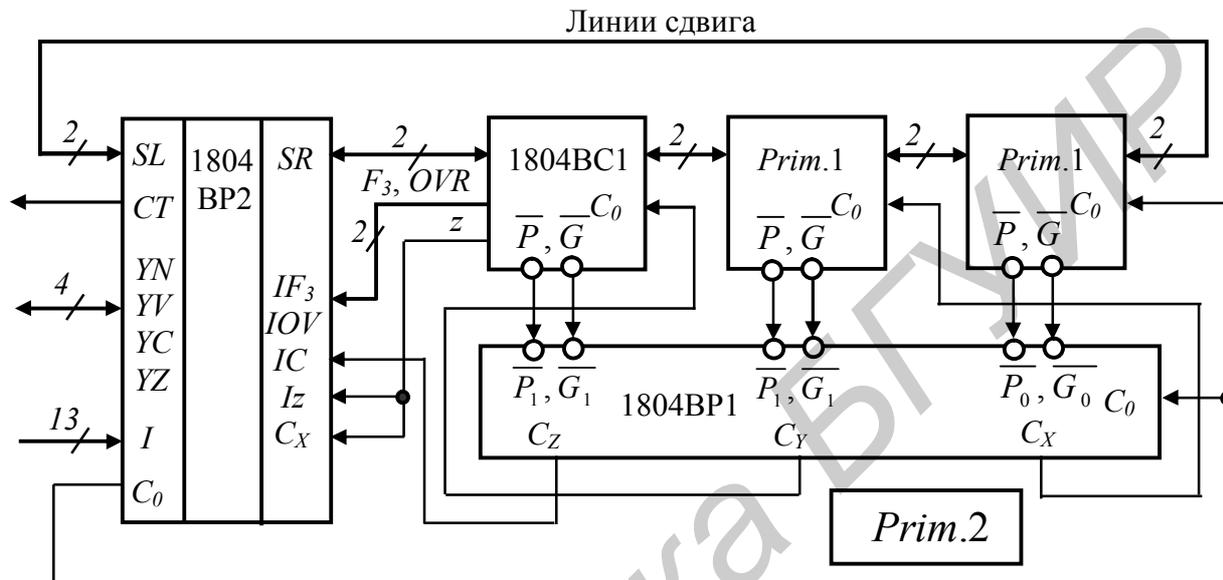


Рис. 2.2

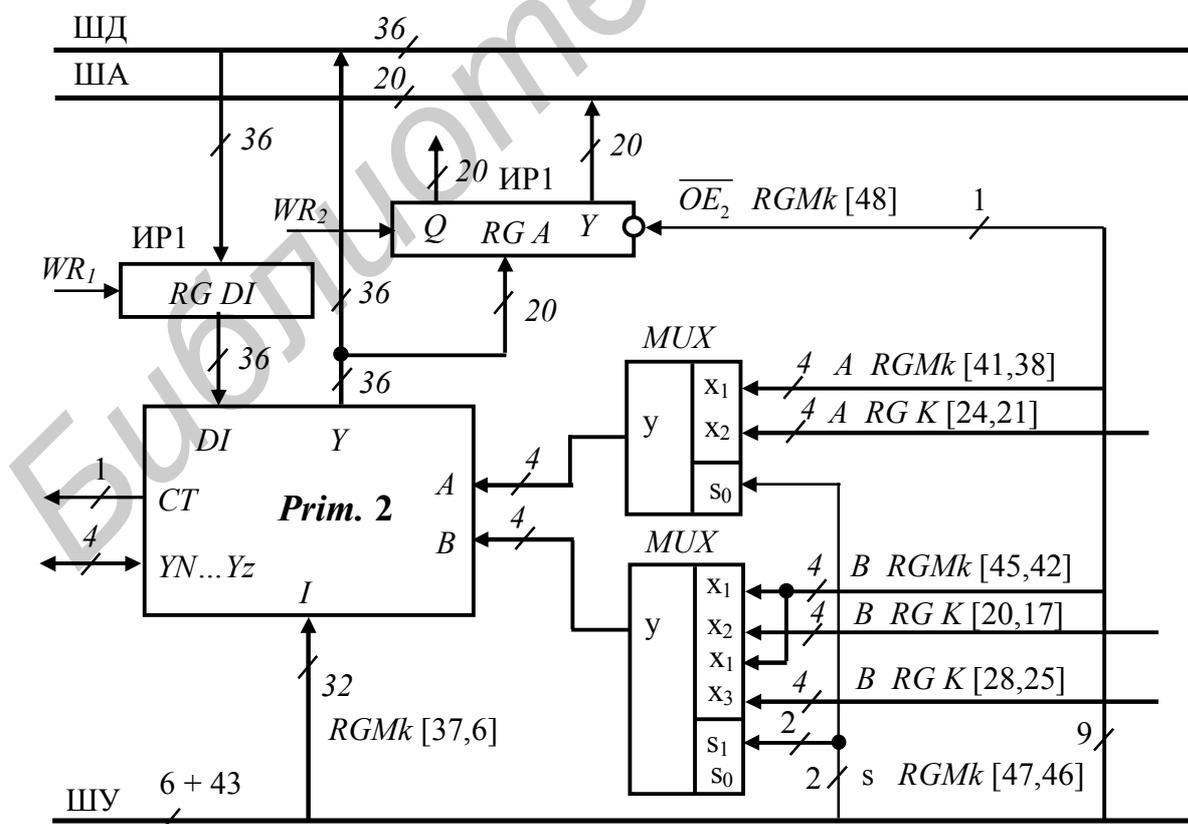


Рис. 2.3

Таблица 2.1

$s_1 s_0$	$y_{MUX A}$	$y_{MUX B}$
0 0	x_1	x_1
0 1	x_2	x_2
1 0	x_1	x_1
1 1	x_2	x_3

В состав управляющего (микрокомандного) слова БОД входят следующие поля (рис. 2.4):

- 1) разряд управления выдачей адреса из RGA на ША – $RGMk[48]$;
- 2) разряды задания адресов РЗУ в МПС К1804BC1 по каналам РЗУ A и B $RGMk [45,38]$;
- 3) сигнал разрешения выдачи данных из микропроцессорной секции на Y -шину \overline{OE}_1 $RGMk[37]$;
- 4) $I_8 - I_0$ – сигналы управления функцией МПС $RGMk[36,28]$;
- 5) $I_{12} - I_0$ – сигналы управления функцией СУСС $RGMk[27,15]$;
- 6) сигналы $RGMk[14,6]$ управления записью информации в регистры К1804BP2 и выдачи признаков на шину YN, YV, YC, YZ ; при этом если указанные двунаправленные выходы не используются, то разряд $RGMk[13] = 1$.

Состояние остальных разрядов данной группы выходов определяется справочными таблицами.

БОД	К1804BC1 (БОД)						К1804BP2 (БОД)					
\overline{OE}_2 s_1 s_0	$B_3 - B_0$	$A_3 - A_0$	\overline{OE}_1	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE} \overline{OEY} \overline{OECT}	\overline{CEN} \overline{CEM}	\overline{EC} \overline{EV} \overline{EN} \overline{EZ}				
48 47 46	45 ... 42	41...38	37	36 ... 28	27 ... 15	14 13 12	11 10	9 ... 6				

Рис. 2.4

Реализация всех вычислений в БОД может осуществляться под управлением микропрограмм, представленных кодами полей микрокоманды, указанной на рис. 2.4.

2.2. Проектирование БОД с использованием МПС К1804BC2

Основные схемотехнические принципы создания вычислительных устройств на базе процессорной секции К1804BC2 состоят в следующем.

1. Каждая процессорная секция настраивается на определенное положение в системе: правая крайняя МПС определяется как младшая и настраивается путем подачи на вход \overline{LSS} потенциала низкого уровня GND . При этом линия $\overline{W}/\overline{MSS}$ становится выходом \overline{W} и используется для управления записью в память. Левая крайняя процессорная секция должна иметь входной

сигнал $\overline{LSS}=1$, что определяет линию $\overline{W}/\overline{MSS}$ как вход \overline{MSS} с подачей на него низкого уровня сигнала. Настройка МПС как средней (все модули между старшей и младшей секциями) осуществляется сигналами $\overline{LSS}=1$ и $\overline{MSS}=1$ (рис. 2.5). Учитывая, что выходные сигналы F_3 и OVR используются только в старшей МПС, в процессорах K1804BC2 выполнено совмещение выводов ускоренного переноса и указанных признаков в парах \overline{G}/F_3 и \overline{P}/OVR . Таким образом при настройке секций на заданное положение осуществляется и соответствующая коммутация выводов. Недостатком данной технологии является возможность формирования только последовательного переноса в старшей МПС.

2. В проектируемом устройстве из двух шин данных DA и DB будем использовать только одну DA -шину, задавая, в частности, уровень сигнала $\overline{OEB}=0$. Это определяет состояние шины DB как выхода, который далее не используется.

3. Проектируемый процессорный модуль *Prim.1* (см. рис. 2.5) не имеет старшей МПС, так как является некоторой составляющей частью всего БОД. Для каскадного соединения модулей *Prim.1* используются выводы СУП $\overline{P}, \overline{G}$, аналогично предыдущему проекту на базе БИС K1804BC1.

4. Для управления выходной шиной Y в МПС используется сигнал \overline{OEY} . В многоразрядных устройствах соответствующие сигналы всех модулей *Prim.1* соединяются электрически и подключаются к линии управления компьютера *RGMk* [37] (см. ниже рис. 2.7). Линии \overline{WE} всех модулей *Prim.1* также соединяются между собой, но управляются, как уже указывалось, выходом $\overline{W}/\overline{MSS}$ младшей секции.

5. Сигнал $\overline{IEN}=0$ используется для разрешения записи данных в регистр Q БОД, разблокировки выхода \overline{W} младшей МПС, а также для разрешения работы ряда дополнительных схем; если сигнал $\overline{IEN}=1$, то RGQ запирается и хранит информацию до установки $\overline{IEN}=0$, на выходе \overline{W} устанавливается единица и, кроме того, блокируется ряд дополнительных функций. В упрощенном варианте БОД рассмотренный сигнал подключается к шине GND .

6. Вход \overline{EA} МПС используется для управления внутренним мультиплексором данных – при $\overline{EA}=0$ на вход АЛУ коммутируется регистр RGA БВП, при $\overline{EA}=1$ источником операнда становится внешняя шина DA . Управление входом \overline{EA} осуществляется микропрограммно из *RGMk* [46] (рис. 2.7).

Реализация БОД из подсхем *Prim.1* показана на рис. 2.6. Устройство содержит основные элементы и линии связей модулей *Prim.1* со схемами СУП и СУСС при организации 36-разрядного вычислительного устройства *Prim.2*. Особенностью данного БОД является принцип включения старшей микропроцессорной секции, в которой распространение переноса осуществляется последовательно. Это приводит к использованию только правой половины схемы СУП во втором каскаде, остальная часть данного модуля не используется.

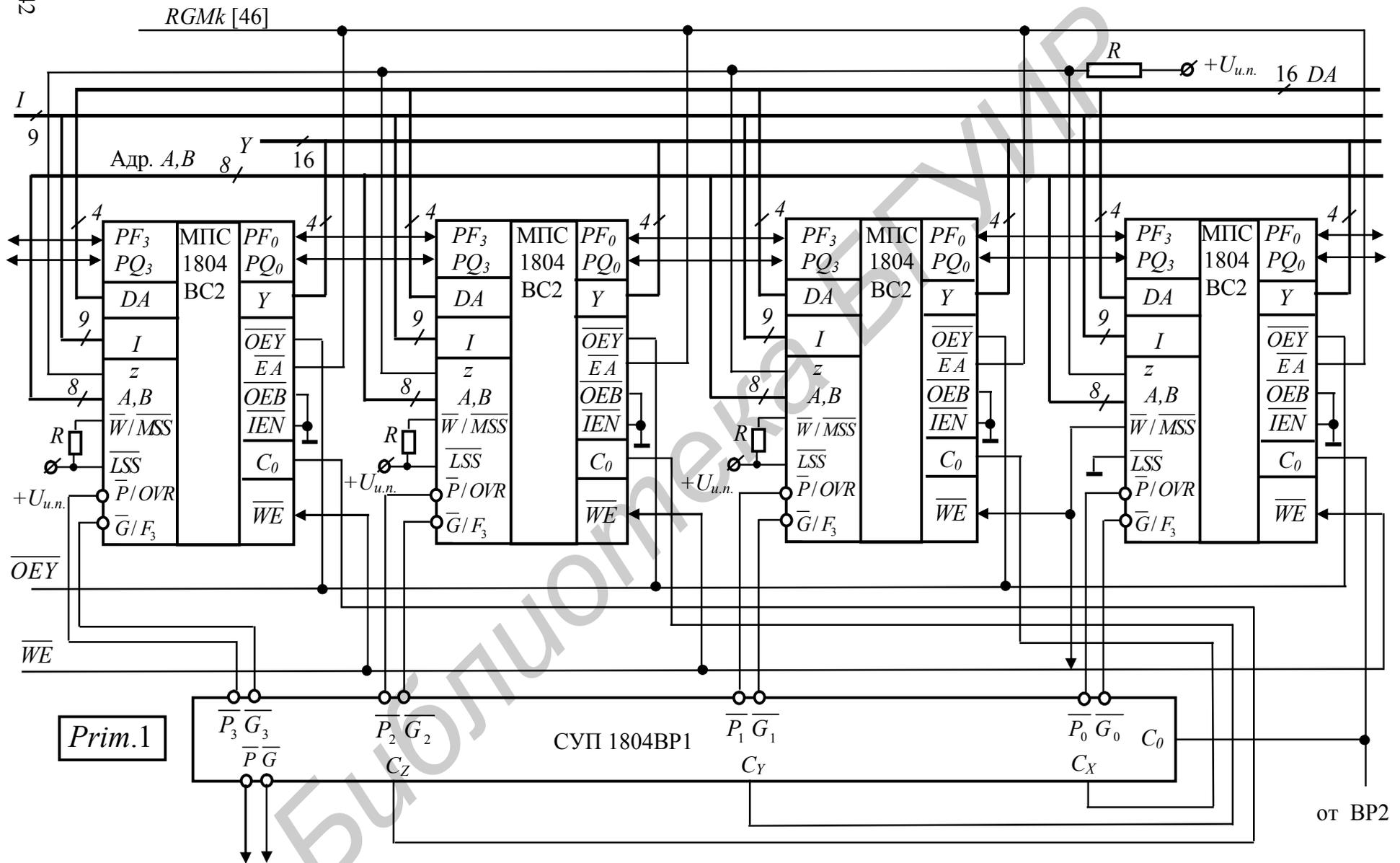


Рис. 2.5

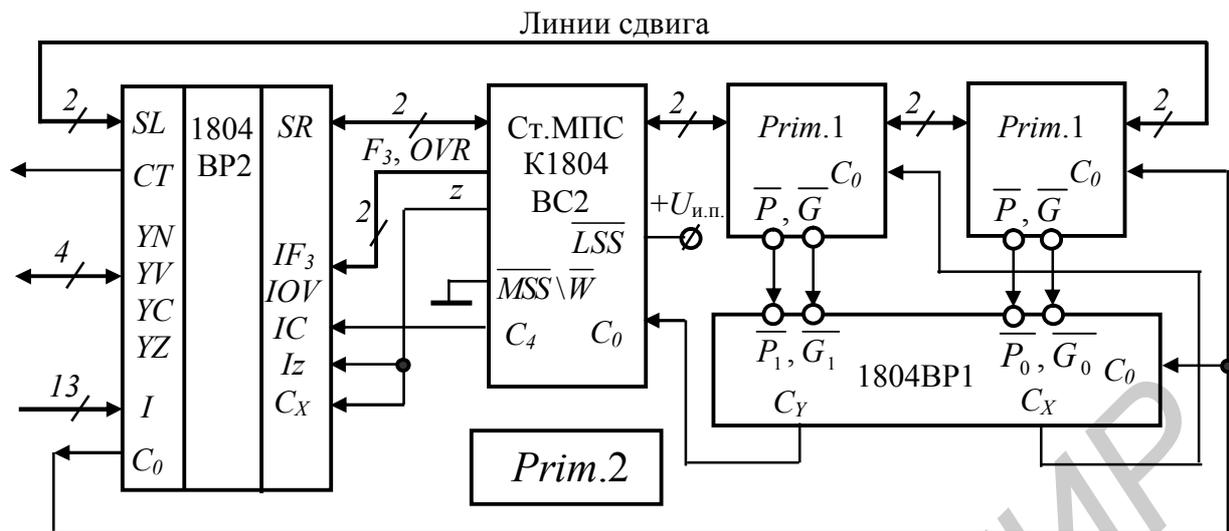


Рис. 2.6

Включение БОД в блок-схему компьютера на базе модуля *Prim.2* выполняется аналогично предыдущему случаю, показанному на рис. 2.3. Отличительной особенностью новой схемы является увеличенная на 1 разряд шина управления за счет включения в состав *RGMk* разряда управления входом \overline{EA} .

БОД			К1804BC2 (БОД)					К1804BP2 (БОД)									
\overline{OE}_2	s_1	s_0	\overline{EA}	$B_3 - B_0$	$A_3 - A_0$	\overline{OEY}	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE}	\overline{OEY}	\overline{OECT}	\overline{CEN}	\overline{CEM}	\overline{EC}	\overline{EV}	\overline{EN}	\overline{EZ}
49	48	47	46	45... 42	41...38	37	36...28	27...15	14	13	12	11	10	9	...	6	

Рис. 2.7

В целом управление БОД на основе микропроцессорной секции К1804BC2 может осуществляться с помощью микрокоманды, показанной на рис. 2.7. На данном этапе проектирования управляющее слово занимает 50 разрядов регистра, где выделены 4 поля сигналов:

- 1) сигналы управления ОЗУ *RGMk* [5,0];
- 2) сигналы управления СУСС К1804BP2 *RGMk* [27,6];
- 3) сигналы управления МПС К1804BC2 *RGMk* [46,28];
- 4) сигналы управления элементами коммутации *RGMk* [49,47].

По отношению к предыдущей реализации (см. рис. 2.4) сформированное управляющее слово содержит перенумерованные разряды \overline{OE}_2 и $s_1 s_0$, что необходимо учитывать при построении схемы, аналогичной рис. 2.3.

2.3. Включение сформированных примитивов в блок-схему компьютера

Для включения БОД на базе МПС К1804BC1 в состав компьютера преобразуем устройство, показанное на рис. 2.3, в модуль *Prim.3*, обобщая по

возможности внешние связи подсистемы к трехшинной организации блок-схемы компьютера. Реализация и включение *Prim.3* в схему проектируемого устройства показана на рис. 2.8.

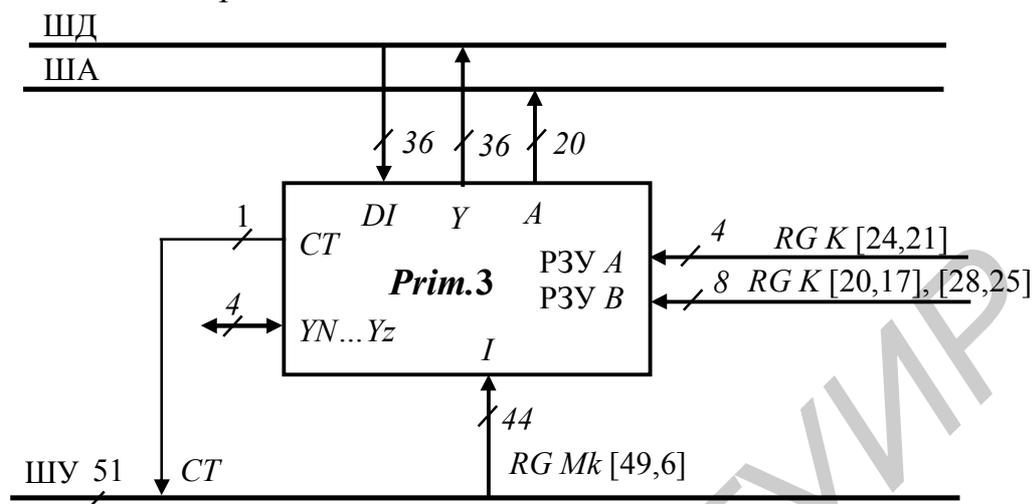


Рис. 2.8

По отношению к стандартной трехшинной системе полученный блок имеет дополнительные шины внутреннего интерфейса для реализации команд формата *RR, RX, RS – RGK [28,17]*. Данный факт, а также наличие объединенной шины команд и данных ядра компьютера позволяет классифицировать всю систему как принстонскую архитектуру. Таким образом, проектируемый компьютер будет всегда вносить ограничения на скорость обработки данных, что является достаточно сильным аргументом в пользу развития архитектур гарвардского типа.

Интеграция БОД в состав компьютера, показанного на рис. 1.23, требует объединения блоков ОЗУ и ПЗУ в единую подсистему. При этом вновь созданный модуль будет содержать 6-разрядную шину управления и шину входного 20-разрядного адреса. В новой базовой схеме (рис. 2.9) СчАК интегрирован в БОД и управляется микропрограммно из БМУ под действием сигналов *RG Mk*.

Для передачи информации на входы ПЗУ *A Prim.3* разряды регистра команды *RGK[20,17]* и *RGK[28,25]* объединены в общую 8-разрядную шину, а задача мультиплексирования возложена на внутренние схемы БОД. Что же касается коммутации адресов в точке «а» блок-схемы, то данную функцию необходимо реализовать путем введения мультиплексора адреса с управлением от *RG Mk* при доработке внутреннего интерфейса компьютера.

Линия *CT* представляет собой выход признака результата выполненной операции в БОД, по которому выполняется ветвление в программе или микропрограмме. Электрически данный вывод соединяется с блоком управления и является одной из линий его входной шины *I*.

В результате проектирования на данном этапе синтеза компьютера получаем разрядность шины управления, равную $m = 6 + 44 + 1 = 51$ разряду.

Однако здесь не учтены разряды управления записью в регистр команды *RGK* и управления точкой «а».

При синтезе БОД на МПС К1804BC2 следует также учитывать тот факт, что различная разрядность машинного слова (32 бита, например) может породить нетривиальное объединение секций при организации ускоренного переноса. Так, отсутствие возможности использовать выходы $\overline{P}, \overline{G}$ старшей МПС приводит к необходимости формирования модернизированного по отношению к *Prim.1* модуля *Prim.1.1*, показанного на рис. 2.10.

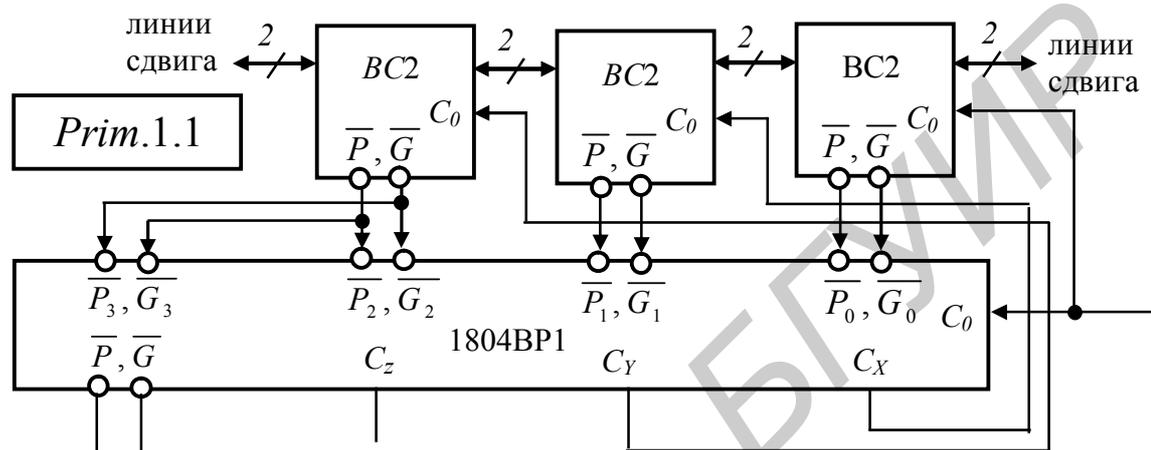


Рис. 2.10

Особенностью приведенного блока является наличие свободных выводов $\overline{P}_3, \overline{G}_3$ в интегральной схеме К1804BP1, которые подключаются с учетом двухкаскадного соединения СУП. В частности, если выходы $\overline{P}_3, \overline{G}_3$ подключены к шине «*GND*», то их потенциал будет определять константное нулевое состояние выходов $\overline{P}, \overline{G}$ СУП. Если указанные выходы не подключены, то состояние выводов $\overline{P}, \overline{G}$ используемой СУП будет соответствовать константным единицам. И в том, и другом случае перенос формируется неверно. Выходом из данной технологически сложившейся ситуации является соединение незадействованных входов $\overline{P}_i, \overline{G}_i$ параллельно с предыдущими входами $\overline{P}_{i-1}, \overline{G}_{i-1}$ схемы переноса. При этом состояние выводов левой крайней средней МПС транспортируется на выходы $\overline{P}, \overline{G}$ текущей СУП и достоверно учитывается в следующем каскаде. Соответственно устройство, показанное на рис. 2.6, вместо левого блока *Prim.1* будет содержать модуль *Prim.1.1*.

Отличительной особенностью модуля *Prim.3* (рис. 2.8), сформированного на основе процессорных секций К1804BC2, по отношению к реализации на МПС К1804BC1, является наличие расширенной на один разряд шины управления компьютером. Так, если входные линии ШУ в модуль *Prim.2* (рис. 2.3) образуют шину микрокоманд *RGMk* [37,6], то реализация устройства на процессоре BC2 требует дополнительного разряда *RGMk* [46]. Соответственно и в схемах, приведенных на рис. 2.8 и 2.9, шина управления *Prim.3* будет содер-

жать 44 бита, то есть $RGMk$ [49,6]. В остальном блок-схемы компьютеров на той и другой процессорных секциях оказываются идентичными.

2.4. Пример микропрограммирования БОД на МПС К1804ВС1

Ниже в табл. 2.2 приведен фрагмент микропрограммы формирования удвоенной суммы содержимого двух регистров МПС $RG_{12} = 2(RG_{12}+RG_{13})$ с записью результата в регистр источник и выдачей суммы $F = RG_{12}+RG_{13}$ на шину данных компьютера.

Таблица 2.2

БОД	К1804ВС1 (БОД)				К1804ВР2 (БОД)					
	$\overline{OE_2}$ s1 s0	$B_3 - B_0$	$A_3 - A_0$	$\overline{OE_1}$	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE} \overline{OEY} \overline{OECT}	\overline{CEN} \overline{CEM}	\overline{EC} \overline{EV} \overline{EN} \overline{EZ}	
48 47 46	45 ... 42	41...38	37	36 ... 28	27 ... 15	14 13 12	11 10	9 ... 6		
100	1100	X	1	337 ₈	X	111	11	1111		
$RG Mk A$ $RG Mk B$ $Y_{RGA} = R_{off}$	$B = 12$	$A = X$	$Y_{MPC} = R_{off}$	$R = D$ $S = 0$ $F = RvS$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет $RG M, RGN$	Запрет записи признаков в $RG M$		
Запись операнда с шины данных в регистр МПС №12										
100	1101	X	1	337 ₈	X	111	11	1111		
Запись операнда с шины данных в регистр МПС №13										
100	1100	1101	0	701 ₈	X	111	11	1111		
$RG Mk A$ $RG Mk B$ $Y_{RGA} = R_{off}$	$B = 12$	$A = 13$	$Y_{MPC} = F$	$R = A$ $S = B$ $F = R + S +$ $+ C_0$ $2F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет $RG M, RGN$	Запрет записи признаков в $RG M$		
Выдача на ШД результата $F = RG_{12}+RG_{13}$ с записью в РЗУ: $RG_{12} = 2(RG_{12}+RG_{13})$										

В приведенной таблице все три микрокоманды при отработке в БОД не используют СУСС К1804ВР2. В связи с этим при микропрограммировании информация в разрядах $RGMk$ [27,6] выбирается таким образом, чтобы вся ИМС находилась в неактивном состоянии. Функциональное назначение микроопераций, записанных в соответствующих полях микрокоманды, определяется комментариями в соответствующих столбцах.

Микропрограммирование полей, управляющих модулями МПС, выполняется в зависимости от выполняемых функций и основывается на справочных таблицах. При этом необходимо правильно выбрать источники операндов, задать функцию АЛУ и определить приемник результата.

3. ПРОЕКТИРОВАНИЕ БЛОКА МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ (БМУ)

3.1. Проектирование блока микропрограммного управления на основе СУАМ К1804ВУ1

Наличие стандартных модулей БИС СУАМ аналогично задаче проектирования БОД предполагает наличие стандартных схем и алгоритмов синтеза трех различных структур блоков микропрограммного управления. При этом, как правило, используются интегральные схемы К1804ВУ1, ВУ2, ВУ4, а также модуль постоянного запоминающего устройства К1804ВУ3 (или другие стандартные ПЗУ) и регистры К1804ИР1 [6, 8].

При проектировании БМУ на базе БИС СУАМ К1804ВУ1 или К1804ВУ2 следует помнить, что основная системная функция данного блока заключается в формировании адресов микрокоманд в микропрограммной памяти под воздействием внешних управляющих сигналов.

В формировании адресов на выходе СУАМ участвуют следующие входные сигналы ШУ: C_0 , $OR_0 - OR_3$, S_1, S_0 , PUP и \overline{FE} . Так при подаче «1» на вход C_0 происходит последовательное увеличение адресов на единицу, а при подаче «1» на i -й вход маски $OR_0 - OR_3$ адрес может быть изменен величину на 2^i . Сигналы на входах S_1, S_0 коммутируют на выход один из четырех источников адреса, а сигналы PUP и \overline{FE} определяют режим работы стека.

Для формирования адресов, разрядность которых превышает четыре бита, необходимо объединить несколько модулей СУАМ в единый блок. При этом стандартный блок из трех ИМС позволяет обращаться к памяти объемом в 4К слов. Соответственно при наращивании модулей объединяются линии шины управления $S_1, S_0, PUP, \overline{FE}, T, \overline{RE}, \overline{OE}$ всех схем, а также выполняется соединение выходов переноса C_4 предыдущей СУАМ со входами C_0 следующей. Сигнал переноса C_0 используется в арифметической функции инкрементирования, поэтому при выборе параметров тактовых импульсов следует учитывать время прохождения переноса через все модули (рис. 3.1).

Далее аналогично методике синтеза БОД в процессе проектирования будем использовать механизм объединения схемы, приведенной на рис. 3.1, в модуль *Prim.4* с набором шин, необходимым для синтеза устройства, показанного на рис. 3.2.

Для упрощения проектируемой системы входы БМУ $OR_0 - OR_3$ всех секций соединяются с шиной «земля», что позволяет исключить маскирование адресов, используемое в многопроцессорных системах. Вход \overline{OE} управления выходной шиной $Y_{11} - Y_0$ также подключается к нулю, что фиксирует адреса СУАМ на входах ПЗУ МПП. Сигнал \overline{zA} подключен к кнопке «Сброс» и используется для перевода компьютера в состояние первоначальной загрузки при включении.

Для управления режимом выполнения переходов используются выходы ПЗУ Q_5-Q_2 (рис. 3.2).

Декодирование кода операции текущей команды, расположенной в $RG K$, выполняется с использованием ПЗУ ПНА (преобразователя начального адреса). Системная функция данного блока заключается в аппаратной трансляции полей КОП и mod в начальный адрес микропрограммы, соответствующей выполняемой команде. Сформированный начальный адрес передается в модули К1804ВУ1 по шине данных $D_{11}-D_0$ на внутренний мультиплексор и далее на выходы $Y_{11}-Y_0$, соединенные со входами МПП.

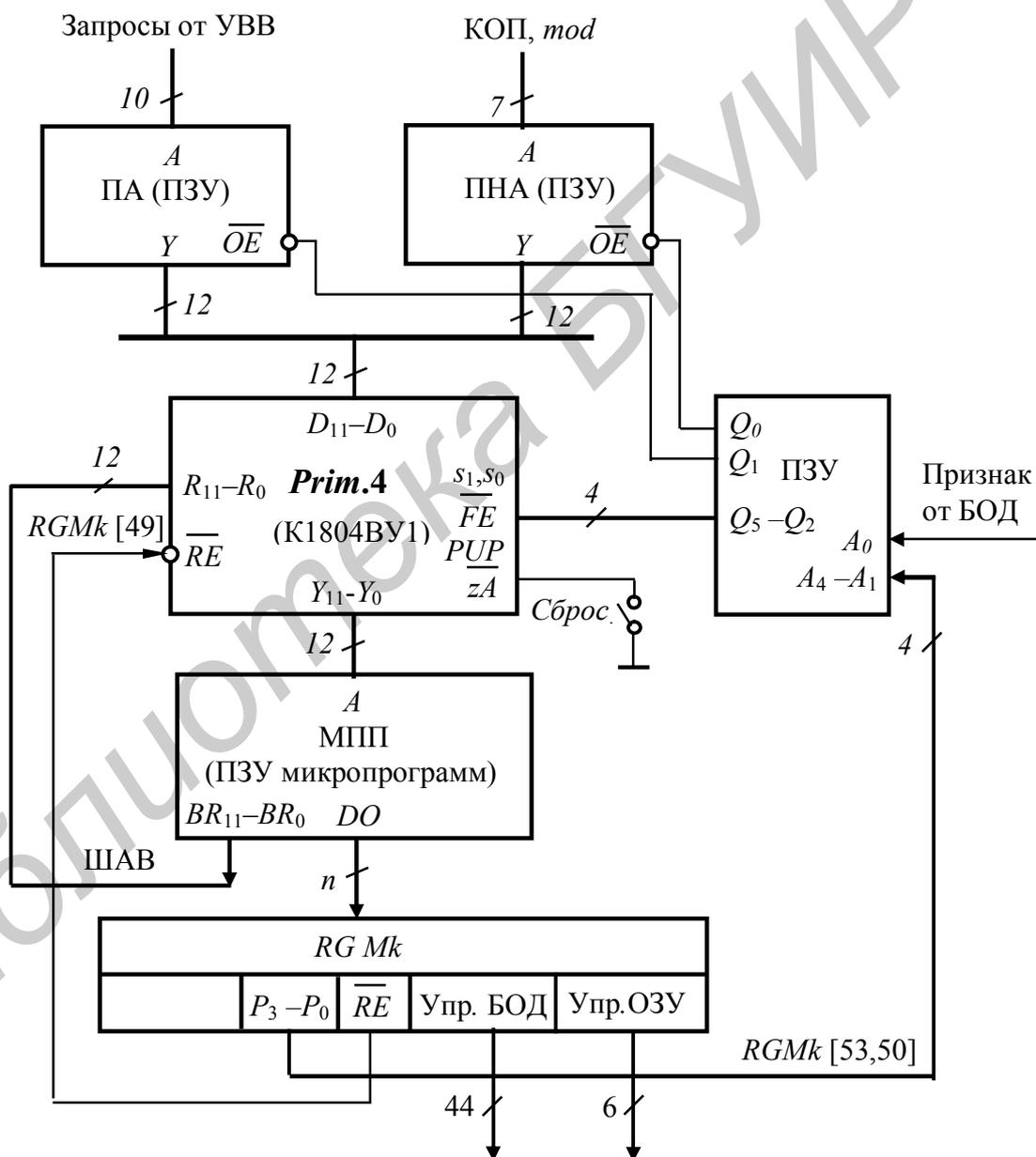


Рис. 3.2

Второй модуль ПЗУ в блоке управления ПА – это преобразователь адреса, который функционирует аналогично ПНА. Однако его входная шина используется для приема вектора запросов на прерывание, поступающих от

портов ввода-вывода К1804ИРЗ. При этом каждому конкретному вектору в соответствие ставится адрес микропрограммы обработки прерывания, имеющего наивысший приоритет среди запросов, составляющих данный вектор. Выполнение соответствующей микропрограммы, как правило, сопровождается переходом к подпрограмме, расположенной в ОЗУ, что позволяет обслуживать практически любое количество портов.

Арбитраж входной шины СУАМ $D_{11} - D_0$ между двумя источниками осуществляется путем перевода выходов одного из ПЗУ ПНА или ПА в состояние R_{off} . Данная функция возложена на выходы управляющего ПЗУ Q_0, Q_1 , которые всегда должны иметь противоположные значения.

Функция опроса системы прерываний чаще всего выполняется после выполнения очередной команды компьютера. При этом каждая микропрограмма, соответствующая команде высокого уровня, последней микрокомандой должна иметь микрокоманду перехода к опросу системы прерываний (ПА), как показано на рис. 3.3.



Рис. 3.3

Выполнение программы обработки прерываний всегда завершается командой передачи управления микропрограмме «Выборка команды».

Для выполнения условных и безусловных переходов на микропрограммном уровне структура микрокоманды содержит выходы явного указания адреса перехода $BR_{11} - BR_0$. Программируемый в данном поле адрес передается напрямую в схему СУАМ, минуя $RGMk$. Это необходимо для того, чтобы переход в заданную точку микропрограммы осуществлялся в текущий момент времени без задержки на такт. Выполненное соединение позволяет

использовать внутренний регистр *RG A* СУАМ в качестве фрагмента *RGMk*. Таким образом, в указанном режиме *RGMk* и *RGA* образуют единый регистр управления. Если бы управление выборкой микрокоманды выполнялось через *RGMk*, то адрес перехода в поле *BR* требовалось бы задавать на такт раньше, т.е. в предыдущей микрокоманде.

Кнопка «Сброс» внешним монтажом подключена ко входу \overline{zA} , обеспечивающему при подаче нулевого уровня переход к нулевой ячейке МПП. При этом в компьютере обрабатывается процедура тестирования внутренних схем и интерфейса, после чего выполняется переход к выполнению микропрограммы «Загрузчик». В общем случае программа тестирования может быть написана на языке высокого уровня и располагаться в системном ПЗУ. В данном случае микропрограмма в МПП, расположенная с нулевого адреса, должна содержать информацию, передающую управление в блок компьютерной памяти (ПЗУ). Если программный «Загрузчик» расположен также в системном ПЗУ, то соответствующая информация из МПП удаляется. В любом случае «Загрузчик» выполняет извлечение из ВЗУ программы «Первоначальная загрузка» и размещает ее в ОЗУ, после чего осуществляется загрузка в память служебных программ, драйверов, программ пользователя и установка блоков компьютера и интерфейса в требуемое состояние. Далее на экран монитора выводится командная строка, и пользователь имеет возможность запустить на выполнение свою программу. В автоматическом режиме работы управление может быть передано одной из программ операционной системы путем записи в СЧАК соответствующего адреса. В любом случае извлечение команды из памяти осуществляется под управлением микропрограммы «Выборка команды».

Интегрирование БМУ в единый модуль компьютера *Prim.5* приводит к интеграции в блок, показанный на рис. 3.4.

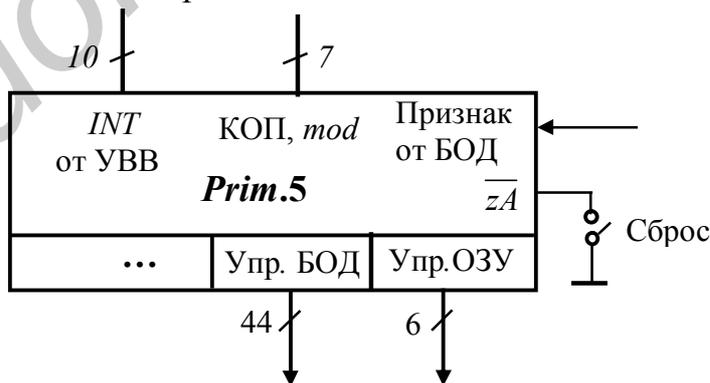


Рис. 3.4

При реализации сброса в автоматическом режиме следует помнить, что время действия сигнала \overline{zA} должно быть достаточным для протекания переходного процесса в комбинационной части СУАМ (элементы И и инкрементор СМК), а также достаточным для записи нуля в регистр счетчика микрокоманд. При этом чтение нуля на входы МПП реализуется по стандартному алгоритму.

В приведенной схеме поле адреса перехода $BR_{11} - BR_0$ из разрядов ПЗУ МПП перенесено в разряды $RGMk$ [65,54], где реализовано управление соответствующим подрегистром по схеме с тремя состояниями. Подключение данного поля к шине адреса ветвления осуществляется выходом Q_6 управляющего ПЗУ. Очевидно, что сигналы, формируемые на выходах Q_0, Q_1, Q_6 , должны содержать унитарную управляющую информацию с целью арбитража шины ШАВ.

Интегрирование БМУ в единый модуль приводит к созданию блока, идентичного подсистеме, приведенной на рис. 3.4.

3.3. Проектирование блока микропрограммного управления на основе УПМ К1804ВУ4

Рассмотрим теперь структуру БМУ, организация которого базируется на применении БИС К1804ВУ4 (рис. 3.6).

Центральной частью блока, очевидно, является схема УПМ, позволяющая обрабатывать ряд сложных алгоритмов при управлении адресами. При этом вход схемы \overline{OE} в данном проекте технологически устанавливается в «0» с тем, чтобы выходы Y всегда передавались бы на шину адреса МПП. Считанная из МПП микрокоманда располагается в разрядах $RGMk$ и используется для формирования ШУ спецкомпьютера.

В качестве носителей входных сигналов в БМУ, проектируемом на основе БИС К1804ВУ4, используются семь линий схемы: вход \overline{CC} , принимающий условие CT от БОД, вход CCE , выполняющий функцию разрешения анализа условия при наличии единицы на входе (в нашем случае $CCE=0$), вход \overline{RLD} – разрешение записи в RGA Сч и четырехразрядная шина инструкции $I_3 - I_0$. Таким образом, в УПМ выполняется 16 видов переходов в соответствии с информацией, содержащейся в 32 ячейках встроенного ПЗУ управления.

Рассмотрим подробно каждую из 16 инструкций, выполняемых проектируемым БМУ.

$I_3 - I_0 = 0000$. JZ – переход к нулевому адресу. При выполнении данной микрооперации реализуется переход к микрокоманде МПП с нулевым адресом. В процессе исполнения инструкции выполняется также очистка стека. Аналогичная функция осуществляется и при нажатии кнопки «Сброс».

$I_3 - I_0 = 0001$. CJS – условный переход к подпрограмме. При выполнении условия (т.е. $CCE=0$ при $\overline{CC}=0$) осуществляется переход к микропрограмме по адресу из $RGMk$, принимаемому по шине D УПМ. Точка возврата $N+1$ запоминается в стеке. Если условие не выполняется ($\overline{CC}=1$ при $CCE=0$), то осуществляется переход к следующей микрокоманде по адресу из СМК.

В данной микрокоманде осуществляется передача адреса из $RGMk$ на шину D К1804ВУ4 без запоминания его в регистрах УПМ.

$I_3 - I_0 = 0010$. *JMAP* – переход по адресу из ПНА. Выполняется безусловный переход по адресу из ПНА, передаваемому через шину *D* УПМ. Инструкция реализуется путем формирования отпирающего сигнала $\overline{ME} = 0$ на выходе УПМ, соединенном со входом управления \overline{OE} ПНА. Микрокоманда, как правило, выполняется после завершения фазы выборки команды из ОЗУ и размещения ее в регистре команд. Иными словами, микрооперация *JMAP* транслирует код операции текущей команды из *RG Mk* в начальный адрес соответствующей микропрограммы в МПП.

$I_3 - I_0 = 0011$. *CJP* – условный переход по адресу из *RG Mk*. При выполнении условия $\overline{CC} = 0$ при $CCE = 0$ осуществляется переход по адресу, принимаемому из *RG Mk* на шину *D*. В отличие от операции $I_3 - I_0 = 0001$ операция со стеком не производится. Если условие не выполняется ($\overline{CC} = 1$ при $CCE = 0$), то осуществляется переход к следующей микрокоманде по адресу из СМК.

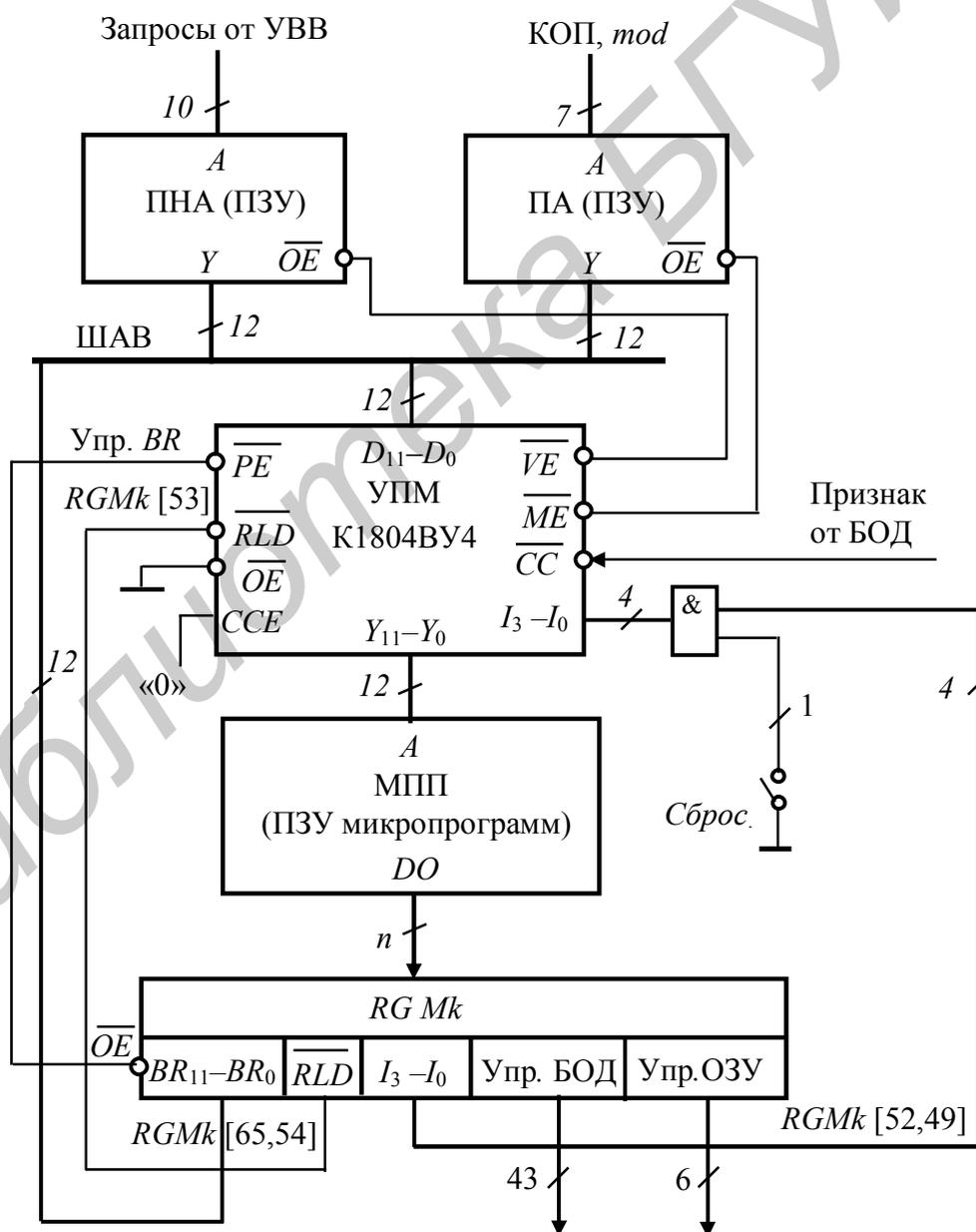


Рис. 3.6

$I_3 - I_0 = 0100$. *PUSH* – засылка в стек и условная загрузка счетчика. При выполнении текущей микрокоманды адрес следующей «заталкивается» в стек. Одновременно с этим, если условие выполняется ($\overline{CC} = 0$ при $CCE = 0$), осуществляется микрооперация загрузки счетчика УПМ из *RGMk* (сигнал $\overline{PE} = 0$). Если условие не выполняется, т.е. $\overline{CC} = 1$ при $CCE = 0$, то загрузка не производится. Для адресации следующей микрокоманды в любом случае выбирается СМК.

$I_3 - I_0 = 0101$. *JSRP* – условный переход к одной из двух подпрограмм. При выполнении данной инструкции адрес возврата из СМК засылается в стек, одновременно осуществляется переход к одной из двух подпрограмм: если условие выполняется ($\overline{CC} = 0$ при $CCE = 0$), то переход выполняется на адрес из *RGMk*, если условие не выполняется ($CCE = 0$ и $\overline{CC} = 1$), то адрес следующей микрокоманды извлекается из счетчика *RGA\Сч*. Адрес перехода 1 предварительно загружается в *RGA\Сч*, а адрес перехода 2 (из *RGMk*) устанавливается на входе *D* модуля УПМ. Содержимое счетчика не изменяется.

$I_3 - I_0 = 0110$. *CJV* – условный переход на адрес вектора. При выполнении условия $\overline{CC} = 0$ при $CCE = 0$ осуществляется переход к микропрограмме по адресу, формируемому преобразователем ПА. Переход реализуется путем генерирования на выходе УПМ сигнала $\overline{VE} = 0$, соединенного со входом \overline{OE} ПА. Если условие не выполняется ($CCE = 0$ и $\overline{CC} = 1$), то реализуется переход на адрес из СМК. Данная инструкция может использоваться при вызове подпрограмм обслуживания прерываний.

$I_3 - I_0 = 0111$. *JSP* – переход на адрес, условно выбираемый из *RGA\Сч* или из *RGMk*. При выполнении условия ($\overline{CC} = 0$ при $CCE = 0$) осуществляется переход на адрес, принимаемый из *RGMk* через шину *D*. Если условие не выполняется, т.е. $CCE = 0$ и $\overline{CC} = 1$, осуществляется переход на адрес из *RGA\Сч* УПМ. В отличие от микроинструкции 5-я операция со стеком не выполняется.

Изначально для выполнения микрооперации 7 адрес перехода предварительно засылается в *RGA\Сч*. Сигнал $\overline{PE} = 0$, что позволяет принимать адрес из *RGMk* через шину *D* на линии внутреннего интерфейса K1804BY4.

$I_3 - I_0 = 1000$. *RFCT* – повторить цикл, если содержимое счетчика не равно нулю. Данная инструкция используется для выполнения цикла из одной или нескольких микрокоманд заданное число раз. Для реализации микроинструкции в стек записывается адрес возврата или начала цикла (используется микроинструкция 4 *PUSH*), а в *RGA\Сч* – требуемое число циклов.

При выполнении микроинструкции 8 проверяется содержимое счетчика *RGA\Сч* и, если оно не равно нулю, то выполняется возврат на адрес из стека. Выполнение цикла повторяется. Одновременно осуществляется декрементирование счетчика *RGA\Сч* и, если содержимое *RGA\Сч* оказалось равным нулю, то вместо начала цикла выполняется переход на адрес из СМК. Адрес возврата при этом выталкивается из стека.

В общем случае адрес, предварительно записанный в стек, может быть использован и как адрес произвольной ветви микропрограммы, расположенной в МПП. Тогда рассматриваемая микроинструкция может интерпретироваться как условный переход на адрес из стека при неравенстве нулю $RGA\backslash Сч$. $I_3 - I_0 = 1001$. *RPCT* – повторять цикл, начиная с адреса из *RGMk*, пока содержимое счетчика не станет равным нулю. Данная инструкция аналогична инструкции 8, за исключением того, что адрес ветвления извлекается из *RGMk* (через шину *D*), а не из стека. Содержимое стека не изменяется.

Если содержимое $RGA\backslash Сч$ не равно нулю, то осуществляется переход в начало цикла по адресу из *RGMk*, принимаемому по шине *D*. Одновременно выполняется уменьшение счетчика на 1. Если содержимое счетчика равно нулю, то выполняется переход по *СМК*.

Микроинструкция 9 может быть использована при необходимости увеличения глубины стека на 1.

$I_3 - I_0 = 1010$. *CRTN* – условный возврат из подпрограммы. Если условие выполняется ($\overline{CC} = 0$ при $CCE = 0$), то осуществляется переход по адресу, принимаемому из вершины стека; содержимое вершины выталкивается. Если условие не выполняется ($\overline{CC} = 1$ при $CCE = 0$), то осуществляется переход по *СМК*.

$I_3 - I_0 = 1011$. *CJPP* – безусловный переход на адрес из *СМК* и выталкивание из стека.

$I_3 - I_0 = 1100$. *LDCT* – загрузить счетчик и продолжить. Данная инструкция относится к классу безусловных. Производится загрузка адреса в $RGA\backslash Сч$ из *RGMk* через шину *D* так как $\overline{PE} = 0$. Одновременно выполняется переход к микрокоманде, адрес которой извлекается из *СМК*.

$I_3 - I_0 = 1101$. *LOOP* – контроль конца цикла. Данная инструкция обеспечивает возможность выхода из цикла по условию. При выполнении условия ($\overline{CC} = 0$ при $CCE = 0$) переход выполняется по *СМК*. Одновременно адрес возврата в цикле выталкивается из стека. Если условие не выполняется ($\overline{CC} = 1$ при $CCE = 0$), то выполняется переход по стеку в начало цикла.

В общем случае данная инструкция может использоваться для условного перехода по стеку на любую ветвь микропрограммы в МПП.

$I_3 - I_0 = 1110$. *CONT* – продолжить. Выполняется безусловный переход на адрес из *СМК*. Никаких других микроопераций не производится.

$I_3 - I_0 = 1111$. *TWB* – ветвление на три направления. Данная инструкция предполагает выбор одного из трех источников адреса: стека, *СМК* или *RGMk* в зависимости от внешнего условия и от содержимого $RGA\backslash Сч$. Для этого необходимо предварительно записать 1-й адрес перехода (возврата) в стек, а число на единицу меньшее, чем количество исполняемых циклов в $RGA\backslash Сч$.

Если содержимое $RGA\backslash Сч$ не равно нулю, а программируемое условие выполняется ($\overline{CC} = 0$ при $CCE = 0$), то осуществляется переход по *СМК*. Одновременно адрес, хранимый в вершине стека, выталкивается, а содержимое $RGA\backslash Сч$ декрементируется. Если содержимое $RGA\backslash Сч$ не равно нулю, а программируемое условие не выполняется ($\overline{CC} = 1$ при $CCE = 0$), то осуществляет-

ся переход по стеку, а содержимое $RGA\setminus Cч$ также декрементируется. При этом адрес, хранимый в вершине стека, не изменяется.

Если содержимое $RGA\setminus Cч$ равно нулю, а программируемое условие выполняется ($\overline{CC} = 0$ при $CCE = 0$), то осуществляется переход по СМК. В противном случае, если $\overline{CC} = 1$ при $CCE = 0$, к микрокоманде, адрес которой извлекается из $RGMk$ на шину D . При этом независимо от выполнения условия адрес возврата выталкивается из стека.

Использование данной инструкции полезно при выполнении многих компьютерных программ, например, при поиске в памяти (операнда), завершающемся нахождением заданного значения, или достижением границы массива, при выполнении операций над числами переменной длины и т.д.

В целом БМУ данного вида может быть интерпретирован рассмотренным выше модулем *Prim.5* (рис. 3.4) и включаться в структуру компьютера на базе МПС К1804ВС1 в соответствии со схемой, приведенной на рис. 3.8.

3.4. Микропрограммирование БОД на МПС К1804ВС1 и БМУ на УПМ К1804ВУ4

Ниже в табл. 3.1 приведена микропрограмма формирования частного от деления двух целых чисел больших нуля ($Дм < Дт$) по методу с восстановлением остатка в соответствии с алгоритмом, приведенным на (рис. 3.7).

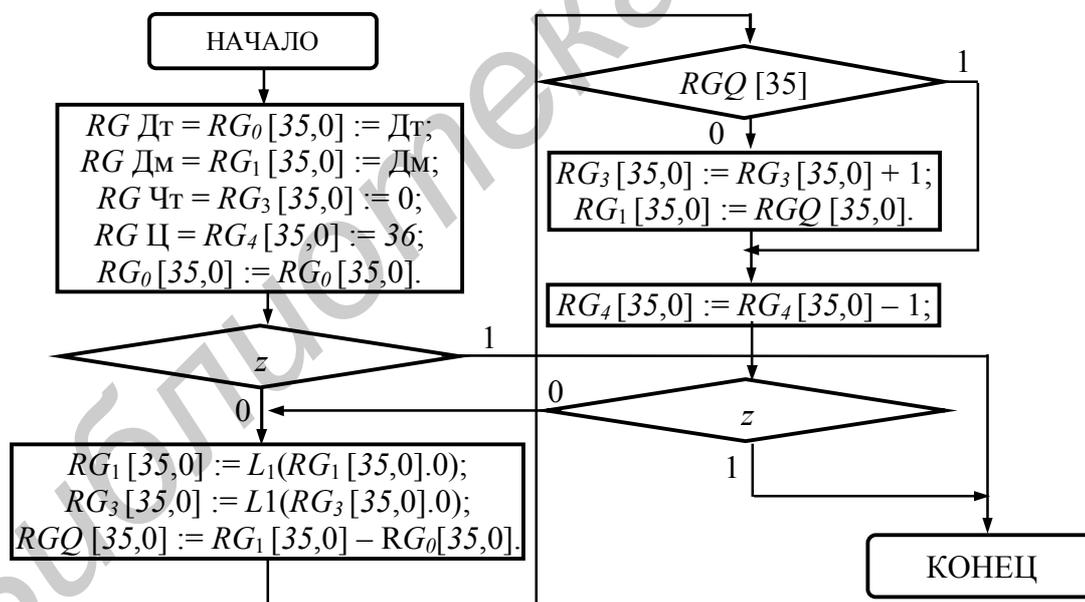


Рис. 3.7

Первой микрокомандой приведенной ГСА деления является микрокоманда № N загрузки делителя в регистр RG_0 . Данная функция не использует БИС К1804ВР2, в связи с чем поля управления регистра микрокоманды $RGMk$ [27,6] могут содержать практически любую информацию. Для конкретности выбираем код $RGMk$ [9,6] = 1111, который запрещает запись признаков в регистр RGM . Эту же функцию интегрально выполняет бит

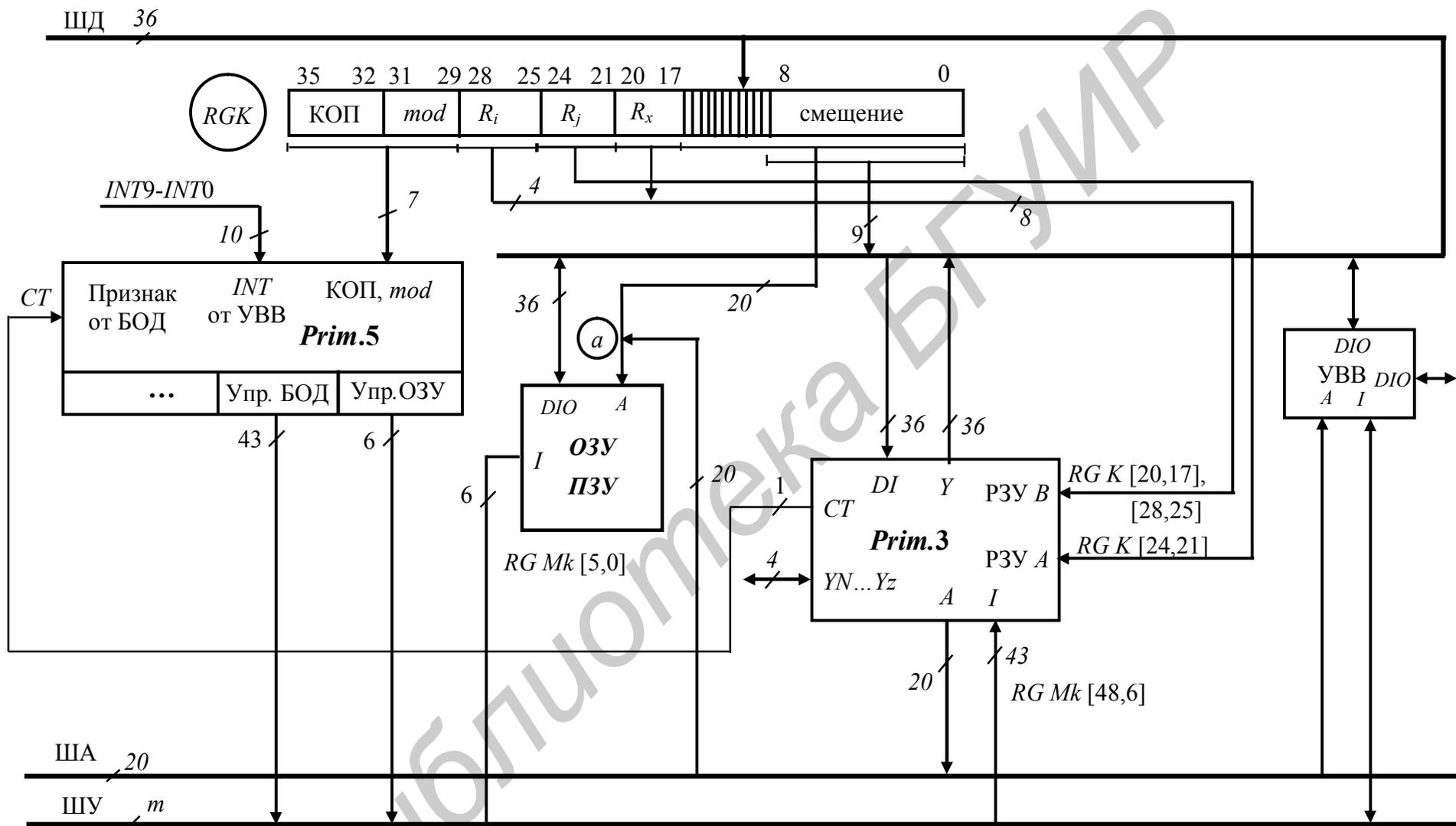


Рис. 3.8

$RGMk [10] = 1$. Запрет RGN осуществляет бит $RGMk [11] = 1$. Разряды $RGMk [14,12] = 111$ переводят выходы БИС К1804BP2 в третье состояние, что позволяет произвольно определить разряды поля управления СУСС $RGMk [27, 15] = X$.

В операционной части первой микрокоманды необходимо установить режим передачи операнда с шины данных компьютера (без изменения, через АЛУ МПС) в регистр RG_0 . С этой целью (для коммутации данных на вход АЛУ) в разряды $RGMk [30,28]$ записывается код 111 (при этом выбираются операнды $R = 0$ и $S = D_{35} - D_0$). Функция АЛУ определяется кодом $RGMk [33,31] = 011$ (операция ИЛИ над нулем и шиной $D_{35} - D_0$), а приемник результата программируется разрядами $RGMk [36,34] = 011$, что предполагает запись результата в выбранный в поле $B_3 - B_0$ нулевой регистр $RGMk [45,42]=0000$. Итого получаем набор сигналов вида

$$RGMk [36,28] = 011011111 = 337_8.$$

Содержимое поля адреса $A_3 - A_0$ может иметь произвольное значение, т.к. в данном преобразовании не используется.

Поле $RGMk [48,46]$ программируется кодом 100, что позволяет управлять адресами РЗУ МПС из соответствующих разрядов $RGMk$.

Адресная часть микрокоманды в данном управляющем слове должна обеспечить передачу управления в следующую ячейку МПП. В связи с этим в поле управления БИС К1804BY4 программируется функция перехода по СМК, т.е. содержимое $RGMk [52,49]$ определяется кодом 100. Остальные разряды операционной части при инкрементировании адреса не используются и программируются произвольно.

Примечание. Все коды полей выбираются из справочных таблиц, приводимых в описании соответствующих БИС в гл.4, имеющей статус технического приложения [6,7].

Вторая микрокоманда с № $N+1$ данной микропрограммы загружает делимое в регистр RG_1 , третья микрокоманда с № $N+2$ – число циклов деления в RG_4 , а четвертая с № $N+3$ обнуляет регистр частного RG_3 . Все поля указанных микрокомандных слов программируются аналогично первому.

Микрокоманда с номером $N+4$ выполняет проверку делителя на равенство нулю. Для реализации данной функции в процессоре осуществляется пересылка вида $RG_0 \rightarrow RG_0$. При этом если признак z окажется равным единице, то управление передается в микрокоманду END с номером $N+11$.

Заполнение полей микрокоманды $N+4$ выполняется следующим образом. Так, программирование схемы СУСС BP2 осуществляется с учетом ориентации ее на передачу признака нуля на вход БМУ. При этом поле $RGMk [9,6]$ не изменяется. Разряды $RGMk [14,12]$ устанавливаются в состояние 110, что запрещает сдвиги информации, но переводит вывод признака CT в рабочий режим. В поле $RGMk [20,15]$ устанавливается код 110101, который коммутирует на выход CT инверсное значение признака нуля. Данная микрооперация диктуется архитектурой УПМ К1804BY4, где нулем определяется факт выполнения условия.

Таблица 3.1

№	K1804BY4 (БМУ)			БОД		K1804BC1 (БОД)				K1804BP2 (БОД)				
	$BR_{11} - BR_0$	\overline{RLD}	$I_3 - I_0$	$\overline{OE_2}$ s ₁ s ₀	$B_3 - B_0$	$A_3 - A_0$	$\overline{OE_1}$	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE} \overline{OEY} \overline{OECT}	\overline{CEN} \overline{CEM}	\overline{EC} \overline{EV} \overline{EN} \overline{EZ}		
	65 ... 54	53	52...49	48 47 46	45 ... 42	41...38	37	36 ... 28	27 ... 15	14 13 12	11 10	9 ... 6		
N.	X	X	1110	100	0000	X	1	337 ₈	X	111	11	1111		
			CMK	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 0$ (Дт)	$A = X$	$Y_{MPC} = R_{off}$	$R = D$ $S = 0$ $F = R \vee S$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет RGM, RGN	Запрет записи признаков в RGM		
	Запись делителя с шины данных в регистр RG0													
N+1.	X	X	1110	100	0001	X	1	337 ₈	X	111	11	1111		
	Запись делимого с шины данных в регистр RG1													
N+2.	X	X	1110	100	0100	X	1	337 ₈	X	111	11	1111		
	Запись числа циклов деления с шины данных в регистр RG4													
N+3.	X	X	1110	100	0011	1101	1	344 ₈	X	111	11	1111		
			CMK	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 3$	$A = 0$	$Y_{MPC} = R_{off}$	$R = 0$ $S = A$ $F = R \& S$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет RGM, RGN	Запрет записи признаков в RGM		
	Запись нуля в регистр частного RG3													
N+4.	N+11 (END)	X	0011	100	0000	0000	1	333 ₈	00065 ₈	110	01	1111		
			RG Mk $z = 0$, CMK $z = 1$	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 0$	$A = 0$	$Y_{MPC} = R_{off}$	$R = 0$ $S = B$ $F = R \vee S$ $F \rightarrow B$	$CT = \bar{z}$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = \text{раб.реж.}$	Запрет RGM, разрешен. RGN	Запрет записи признаков в RGM		
	Проверка делителя на равенство нулю: $RG0 \rightarrow RG0$ и переход по z													

	K1804BY4 (БМУ)			БОД	K1804BC1 (БОД)				K1804BP2 (БОД)				
	$BR_{11} - BR_0$	\overline{RLD}	$I_3 - I_0$	\overline{OE}_2 s ₁ s ₀	$B_3 - B_0$	$A_3 - A_0$	\overline{OE}_1	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE} \overline{OEY} \overline{OECT}	\overline{CEN} \overline{CEM}	\overline{EC} \overline{EV} \overline{EN} \overline{EZ}	
	65 ... 54	53	52...49	48 47 46	45 ... 42	41...38	37	36 ... 28	27 ... 15	14 13 12	11 10	9 ... 6	
N+5.	X	X	1110	100	0001	X	1	633 ₈	X	110	01	1111	
			CMK	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 1$ (Дм)	$A = X$	$Y_{MPC} = R_{off}$	$R = 0$ $S = B$ $F = R \vee S$ $2F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = \text{раб.реж.}$	Запрет RGM, разрешен RGN	Запрет записи признаков в RGM	
				Сдвиг регистра делимого RG1 влево на 1 разряд									
N+6.	X	X	1110	100	0011	X	1	633 ₈	X	110	01	1111	
				Сдвиг регистра частного RG3 влево на 1 разряд									
N+7.	N+10	X	0001	100	0001	0000	1	011 ₈	04037 ₈	110	01	1111	
	адрес перехода		$F3 = 1$ CMK. $F3 = 0$ RGMk	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 1$	$A = 0$	$Y_{MPC} = R_{off}$	$R = A$ $S = B$ $F = S - R - 1 + C0.$ $F \rightarrow Q.$	$CT = \overline{F3},$ $C0 = 1$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = \text{раб.реж.}$	Запрет RGM, разреш. RGN	Запрет записи признаков в RGM	
				$RGQ = RG \text{ Дм} - RG \text{ Дт}$ и переход по знаку результата									
N+8	X	X	1110	100	0011	X	0	303 ₈	X	111	11	1111	
			CMK	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 3$	$A = X$	$Y_{MPC} = F$	$R = 0$ $S = B$ $F = R + S + C0.$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет RGM, RGN	Запрет записи признаков в RGM	
				Инкремент регистра частного RG3									

	K1804BY4 (БМУ)			БОД	K1804BC1 (БОД)				K1804BP2 (БОД)				
	$BR_{11} - BR_0$	\overline{RLD}	$I_3 - I_0$	\overline{OE}_2 s1 s0	$B_3 - B_0$	$A_3 - A_0$	\overline{OE}_1	$I_8 - I_0$	$I_{12} - I_0$	\overline{SE} \overline{OEY} \overline{OECT}	\overline{CEN} \overline{CEM}	\overline{EC} \overline{EV} \overline{EN} \overline{EZ}	
	65 ... 54	53	52...49	48 47 46	45 ... 42	41...38	37	36 ... 28	27 ... 15	14 13 12	11 10	9 ... 6	
N+9.	X	X	1110	100	0001	X	1	632 ₈	X	110	01	1111	
			CMK	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 1$ (ДМ)	$A = X$	$Y_{MPC} = R_{off}$	$R = 0$ $S = Q$ $F = R \vee S$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = \text{раб.реж.}$	Запрет RGM, разрешен. RGN	Запрет записи признаков в RGM	
Пересылка содержимого регистра RGQ в RG1													
N+10	N+5	X	0001	100	0100	X	1	313 ₈	00065 ₈	110	01	1111	
	адрес перехода		$z = 0,$ CMK. $z = 1,$ RGMk	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 4$	$A = 0$	$Y_{MPC} = R_{off}$	$R = 0$ $S = B$ $F = S - R - 1 + C0.$ $F \rightarrow B.$	$CT = \bar{z},$ $C0 = 0.$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = \text{раб.реж.}$	Запрет RGM, разреш. RGN	Запрет записи признаков в RGM	
Декремент RG4 и переход по признаку z													
N+11	N+11	X	0010	100	0011	X	0	333 ₈	X	111	11	1111	
			RGMk	RG Mk A RG Mk B $Y_{RGA} = R_{off}$	$B = 3$	$A = X$	$Y_{MPC} = F$	$R = 0$ $S = B$ $F = R \vee S$ $F \rightarrow B$	$I = X$	$SL, SR = R_{off}$ $Y_{BP2} = R_{off}$ $CT = R_{off}$	Запрет RGM, RGN	Запрет записи признаков в RGM	
Останов													

В разрядах $RGMk$ [27,26] программируются нули, определяющие выход $C_0 = 0$. Остальные разряды данного поля указывают на тип сдвига и могут быть выбраны произвольно, например, $RGMk$ [25,21] = 00000. Интегрируя все указания в код микрокоманды, получаем:

$$RGMk[27,15] = 0000000110101 = 00065_8.$$

Последние два бита поля управления СУСС $RGMk$ [11,10] устанавливаются в состояние 01, что позволяет отпереть регистр признаков RGN для записи флагов со входов IC, IV, IN, IZ .

В операционной части микрокоманды $N+4$ программируется микрооперация перезаписи операнда-делителя вида $RG_0 \rightarrow RG_0$. Поэтому в поле адреса канала B РЗУ устанавливается нулевой код $RGMk$ [45,42] = 0000. Поле адреса канала A может иметь произвольное значение, однако (избыточно) может быть также установлено в нуль $RGMk$ [41,38] = 0000.

Для управления функцией МПС следует выбрать микрооперации:

- 1) $RGMk[30,28] = 011$, $R = 0, S = B$, выбор в качестве операндов RG_0 и нуля;
- 2) $RGMk$ [33,31] = 011, $F = R \vee S$, операция ИЛИ над RG_0 и нулем;
- 3) $RGMk$ [36,34] = 011, $F \rightarrow B$, перезапись не измененного RG_0 в RG_0 .

Интегрально данное поле микрокоманды записывается в виде

$$RGMk$$
 [36,28] = 011011011 = 333₈.

В адресной части микрокоманды $N+4$ необходимо запрограммировать ветвление по признаку нуля z . С этой целью в поле $I_3 - I_0$ необходимо указать код $RGMk$ [52,49] = 0011. При этом, если условие не выполнено (в данной реализации системы $z=1$), то компьютер переходит к выполнению команды деления; если условие выполнено ($z=0$), то осуществляется выход из микропрограммы на конечную вершину ГСА.

Особенностью ветвления в созданной архитектуре является возможность выполнения перехода в текущей микрокоманде управления. Однако недостатком такой технологии является снижение тактовой частоты процессора, что следует из длительности протекания переходного процесса при распространении сигнала признака от МПС через СУСС в схему ветвления УПМ. При этом потери производительности на простых микрооперациях существенны и очевидны.

Указанный недостаток можно преодолеть, если сформированный в МПС признак априорно ветвлению записывать в регистр RGN СУСС. В данной реализации микропрограммы требуемый переход реализуется за два такта, причем частота тактов может быть выбрана значительно большей по отношению к предыдущей реализации системы. В данном случае потери времени на переходах ГСА достаточно эффективно компенсируются на большинстве остальных микроопераций МПС.

Очередная микрокоманда с номером $N+5$ предназначена для сдвига регистра делимого в RG_1 в цикле влево на один разряд. Данное преобразование

необходимо для последующего вычитания регистра делителя из регистра делимого с целью формирования цифры частного. Принципиально микропрограммирование указанной функции практически не отличается от алгоритма заполнения полей $RGMk$, использованного выше. Особенностью микрокоманды с номером $N+5$ является организация сдвига только в разрядах регистра МПС без использования СУСС К1804BP2. Это обусловлено тем, что биты старших разрядов делимого в процессе формирования частного могут быть потеряны, так как в последующих циклах не используются. Таким образом, в разрядах $RGMk$ в поле управления СУСС программируем произвольное слово управления, а именно:

$$RGMk [27,6] = X 110 01 1111.$$

Сущность выполнения сдвига в микропроцессорной секции К1804BC1 состоит в чтении операнда из РЗУ, выполнении неразрушающих делимое действий в АЛУ и записи его со сдвигом в регистр источник. Данную функцию реализует фрагмент микрокоманды вида

$$RGMk [36,28] = 110011011 = 633_8.$$

Заполнение остальных полей $N+5$ микрокоманды осуществляется аналогично предыдущим микроинструкциям.

Микрокоманда сдвига $N+6$ программируется практически так же, как и управляющее слов $N+5$.

Следующая микрокоманда расположена в МПП по адресу $N+7$ и предназначена для выполнения условного перехода по знаку результата вычитания регистра делителя из регистра делимого. Результат вычитания записывается в неявно адресуемый регистр Q . В отличие от микрокоманды $N+4$ в текущем управляющем слове программируется функция АЛУ вида

$$F = S - R - 1 + C_0.$$

Данное действие требует компенсации вычитаемой в АЛУ единицы с помощью сигнала входного переноса $C_0 = 1$. При этом на входы управления СУСС $I_{12} - I_0$ необходимо подать биты регистра микрокоманды

$$RGMk [27,15] = 0100000011111 = 04037_8.$$

Это обеспечивает установку на входе C_0 МПС логической единицы ($RGMk [27,26] = 01$), установку выхода CT СУСС в рабочий режим, а также коммутацию на выход CT инверсного значения знака результата:

$$RGMk [20,15] = 011111_8.$$

Функция сдвига в битах $RGMk [25,21]$ программируется произвольно, например кодом 00000.

Формирование оставшихся микрокоманд является, по сути дела, вариационным повтором предыдущих управляющих слов и не требует дополнительных разъяснений.

4. АППАРАТУРА И ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ БАЗОВЫХ БИС СПЕЦКОМПЬЮТЕРА

4.1. Микропроцессорная секция K1804BC1

Четырехразрядная микропроцессорная секция K1804BC1 предназначена для построения операционных блоков цифровых устройств с разрядностью, кратной 4.

На структурной схеме МПС условно выделяют четыре крупных блока (рис. 4.1):

- 1) БВП – блок внутренней памяти;
- 2) АЛБ – арифметико-логический блок;
- 3) БР – блок регистра Q ;
- 4) БУ – блок управления.

Блок внутренней памяти содержит в своем составе регистровое ЗУ с двумя независимыми каналами выбора информации – каналом адреса A и каналом адреса B . На входе РЗУ включен сдвигатель канала A ($CD A$), позволяющий записывать в ЗУ информацию как без сдвига, так и со сдвигом вправо или влево на один разряд. Запись в РЗУ возможна только по адресу, указанному на линиях канала B .

Регистры $RG A$ и $RG B$, установленные на выходах РЗУ, представляют собой 4-разрядные устройства с синхронной записью информации.

Выбор из БВП любого РОН в качестве источника информации осуществляется путем подачи на входы A и B адресной информации без программирования каких-либо других управляющих сигналов. Из РЗУ одновременно могут быть считаны два операнда. При этом, если на входах A и B установлены одинаковые адреса (выполняется обращение к одному и тому же РОН), то на обоих выходах РЗУ появляются идентичные данные. Запись данных в РЗУ выполняется только по каналу B , при этом адрес канала A игнорируется. Моментом начала записи в РЗУ является момент перехода тактового импульса из состояния «1» в состояние «0». При этом информация перед записью может быть сдвинута влево или вправо на один разряд. Эту операцию выполняет сдвигатель данных $CD A$, управляемый сигналами с дешифратора приемника результата. Схема $CD A$ имеет структуру, приведенную на рис. 4.2.

Арифметико-логический блок включает в свой состав двухвходовое АЛУ, выполняющее 8 арифметических и логических операций и формирующее 4 признака на основе выполненных действий:

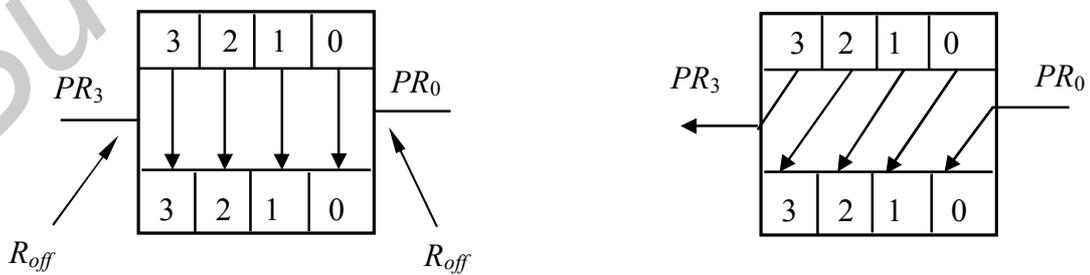
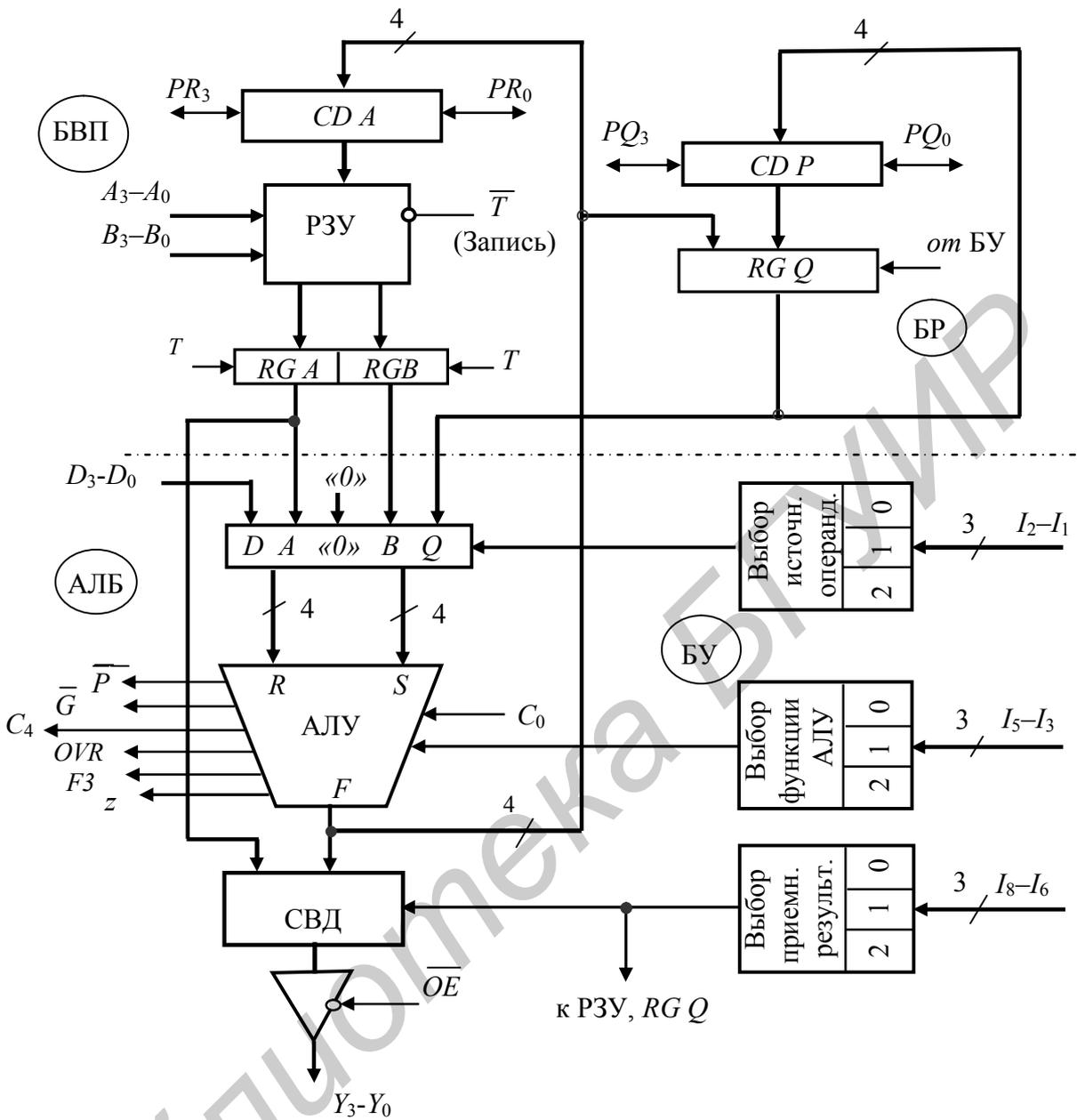
C_4 – перенос из старшего разряда результата;

OVR – переполнение; $OVR = 1$, если $C_4 \oplus C_3 = 1$;

F_3 – знак числа или значения старшего разряда на выходе АЛУ;

z – признак нулевого результата.

С выхода АЛУ информация подается на первый вход селектора выходных данных (СВД). На второй вход селектора данные передаются прямо с выхода $RG A$, минуя АЛУ. Это позволяет одновременно с выполнением действий



в арифметико-логическом блоке выводить данные (по каналу A) на вход СВД.

С выхода СВД информация через управляемые усилители передается на выходную шину МПС – трехстабильную шину Y .

Управление работой АЛБ осуществляется с помощью табл. 4.1, 4.2, 4.3.

Таблица 4.1

Микрокод				Источник операндов АЛУ	
I_2	I_1	I_0	δ -ричн.	R	S
0	0	0	0	A	Q
0	0	1	1	A	B
0	1	0	2	0	Q
0	1	1	3	0	B
1	0	0	4	0	A
1	0	1	5	D	A
1	1	0	6	D	Q
1	1	1	7	D	0

Таблица 4.2

Микрокод				Операция АЛУ
I_5	I_4	I_3	δ -ричн.	
0	0	0	0	$R+S+C_0$
0	0	1	1	$S-R-1+C_0$
0	1	0	2	$R-S-1+C_0$
0	1	1	3	$R \vee S$
1	0	0	4	$R \wedge S$
1	0	1	5	$\overline{R} \wedge S$
1	1	0	6	$\frac{R \oplus S}{R \oplus S}$
1	1	1	7	$\frac{R \oplus S}{R \oplus S}$

Селектор источников данных выбирает операнды для R и S входов АЛУ. Причем выбор источников осуществляется сигналами микрокоманды $I_2 - I_0$ (см. табл. 4.1), приемника результата – сигналами $I_8 - I_6$ (табл. 4.3), а функции АЛУ – сигналами $I_5 - I_3$ (табл. 4.2).

Арифметические операции в АЛУ выполняются с учетом значения сигнала входного переноса C_0 и по правилам дополнительного кода при представлении отрицательных чисел. Это, в частности, означает, что если оба операнда нулевые и выполняется вычитание при $C_0 = 0$, на выходе устанавливается значение 1111, обозначающее в дополнительном коде число -1 . При $C_0 = 1$ на выходе АЛУ формируется значение 0000.

Таблица 4.3

Микрокод				РЗУ		RGQ		Выход Y
I_8	I_7	I_6	δ -ричн.	Сдвиг	Загрузка	Сдвиг	Загрузка	
0	0	0	0	—	—	—	$F \rightarrow Q$	F
0	0	1	1	—	—	—	—	F
0	1	0	2	—	$F \rightarrow B$	—	—	A
0	1	1	3	—	$F \rightarrow B$	—	—	F
1	0	0	4	Вправо	$F/2 \rightarrow B$	Вправо	$Q/2 \rightarrow Q$	F
1	0	1	5	Вправо	$F/2 \rightarrow B$	—	—	F
1	1	0	6	Влево	$2F \rightarrow B$	Влево	$2Q \rightarrow Q$	F
1	1	1	7	Влево	$2F \rightarrow B$	—	—	F

При выполнении арифметических операций сигнал OVR , формируемый старшей микропроцессорной секцией, показывает, что результат преобразований в дополнительном коде занял знаковый разряд.

Выводы $\overline{P}, \overline{G}$ АЛУ позволяют с помощью внешних схем организовать между секциями ускоренный перенос при комплексировании их в блок с разрядностью, кратной четырем. При этом используется интегральная схема K1804BP1, применяемая в качестве сборки для четырех схем K1804BC1. Дальнейшее наращивание процессора по линиям переноса выполняется путем каскадного соединения сформированных 16-разрядных подсхем в единый блок.

Сигналы признаков, формируемые АЛУ, используются следующим образом.

Вывод F_3 – старший разряд АЛУ – может быть использован, например, для определения знака арифметической операции. При этом отсутствует необходимость отпираания трехстабильной выходной шины, что упрощает выполнение команд перехода в мультипроцессорных системах. При соединении нескольких МПС знаком является вывод F_3 старшей секции. Выводы F_3 остальных секций не используются.

Выход z выполнен по схеме с открытым коллектором. При объединении нескольких МПС все выводы z соединяются в общей точке, подключаемой через резистор к источнику питания. Потенциал этой точки имеет высокий уровень, если все выходы АЛУ одновременно нулевые.

Сигналы на выходах $\overline{P}, \overline{G}$, OVR и C_4 имеют смысл при выполнении в АЛУ арифметических операций. Однако информация на этих выводах появляется и при выполнении логических преобразований. Поэтому формирование сигналов состояния при выполнении различных операций осуществляется с помощью ряда булевых соотношений, приведенных в справочной литературе.

Блок регистра Q состоит из дополнительного регистра RGQ и сдвигателя CDP . Сдвигатель данных позволяет перезаписывать информацию в RGQ как без сдвига, так и со сдвигом влево или вправо на один разряд. Основное назначение вспомогательного регистра состоит в хранении результата удвоенной длины при выполнении операции умножения. Аналогичная ситуация возникает и при реализации деления, если существует необходимость в получении результата удвоенной точности. Запись данных в RGQ выполняется по положительному фронту сигнала синхроимпульса на соответствующем входе.

Блок управления формирует управляющие сигналы для остальных блоков МПС. Входами данной подсхемы является шина микрокоманды I_8-I_0 , которую условно можно разделить на три части. Блок управления соответственно также может быть представлен в виде совокупности трех частей, при этом первая вырабатывает сигналы управления для СИД, вторая используется для управления функцией АЛУ, а третья – для управления СВД, RGQ , CDP , CDA .

Инверсный сигнал \overline{OE} предназначен для управления Y -выходами МПС. При $\overline{OE} = 0$ разрешается вывод информации через Y -выходы на ШД; если $\overline{OE} = 1$, выходная шина отключается (переводится в состояние R_{off}).

4.2. Микропроцессорная секция K1804BC2

По отношению к предыдущей разработке процессорная секция K1804BC2 имеет ряд усовершенствований (рис. 4.3):

во-первых, АЛУ микропроцессора выполняет арифметические, логические и специальные функции;

во-вторых, сдвигатель данных АЛУ *CDA* выполняет как логические, так и арифметические сдвиги;

в-третьих, в микропроцессоре заложена возможность внешнего расширения РЗУ путем подсоединения любого числа дополнительных рабочих регистров, обращение к которым возможно в различных режимах адресации.

В целом микропроцессорная секция K1804BC2 состоит из 4 основных блоков: блока внутренней памяти (БВП), арифметико-логического блока (АЛБ), блока рабочего регистра (БР) и блока управления (БУ).

Блок внутренней памяти включает в свой состав шестнадцать 4-рядных РОН, объединенных в РЗУ, а также регистр *A* и регистр *B* (последний с трехстабильным выходом). Информация, размещаемая в регистрах, адресуется по каналам адреса $A_3 - A_0$ и $B_3 - B_0$ соответственно.

Управление выходом регистра *B* осуществляется сигналом \overline{OEB} , что позволяет передавать данные с выхода БВП на вход мультиплексора *MsS* АЛУ и на выходы $DB_3 - DB_0$ МПС. При этом $\overline{OEB} = 0$ и, кроме того, при $\overline{OEB} = 1$ имеется возможность вводить информацию со входов $DB_3 - DB_0$ через мультиплексор на вход АЛУ.

Каждый из РОНов может быть выбран в качестве источника или приемника полученного в АЛУ результата. При этом информация с выходов РЗУ записывается в *RGA* или *RGB* при наличии фронта логической единицы на входе *T*, если же $T = 1$, то регистры находятся в режиме хранения.

Запись информации в РЗУ может производиться только по адресу, указанному на линиях канала *B*. При этом необходимо, чтобы на входе \overline{WE} был установлен логический «0». Подача на входы \overline{WE} единичного потенциала превращает режим записи. Считывание информации из РЗУ производится одновременно по адресам *A* и *B*. Если на входах каналов *A* и *B* установлены одинаковые адреса, то на выходы РЗУ считывается одинаковая информация.

МПС может функционировать в режиме двухадресной $B = A + B$ и трехадресной $Q = A + B$ обработки данных. В двухадресном режиме на входы $A_3 - A_0$ подается адрес операнда *R*, а на входы $B_3 - B_0$ – адрес операнда *S* и результата. В трехадресном режиме приемником результата служит рабочий регистр *Q*, адресуемый в микрокоманде неявно.

Арифметико-логический блок состоит из двух мультиплексоров *MsS* и *MsR*, арифметико-логического устройства, сдвигателя *CDA* с трехстабильным выходом (рис. 4.4), а также формирователя признака нуля ФПН, который используется при выполнении арифметических, логических и специальных функций и при формировании сигналов состояния МПС.

Таблица 4.4

\overline{EA}	I_0	\overline{OEB}	R	S
0	0	0	A	B
0	0	1	A	DB
0	1	0	A	Q
0	1	1	A	Q
1	0	0	DA	B
1	0	1	DA	DB
1	1	0	DA	Q
1	1	1	DA	Q

Выбор операции АЛУ осуществляется под действием поля регистра микрокоманды $I_8 - I_0$ табл. 4.5, причем если разряды $I_4 - I_0$ имеют значение 00000, то АЛУ выполняет специальные функции. Выводы $\overline{P}, \overline{G}$ МПС используются для организации ускоренного переноса в многоразрядных процессорах. При этом выходы $\overline{P}, \overline{G}$ Ст.МПС не используются, а выходы F_3, OVR , наоборот, используются только у Ст.МПС. Этот факт позволяет объединить выходы \overline{G} и F_3 , а также \overline{P} и OVR

и соответственно уменьшить общее число контактов ИМС.

Информационные данные с выхода АЛУ могут быть переданы на входы сдвигателя CDP регистра Q и на входы сдвигателя CDA , соединенного с выходами F АЛУ. Таким образом, выходная информация может записываться в РЗУ и выводиться на шину $Y_3 - Y_0$ как без сдвига, так и со сдвигом влево или вправо на один разряд. Отличительной особенностью CDA АЛУ является процесс выполнения арифметических сдвигов. Схема сдвигателя для старшей МПС процессора при сдвиге влево или вправо на 1 разряд в этом случае имеет вид, показанный на рис. 4.4.

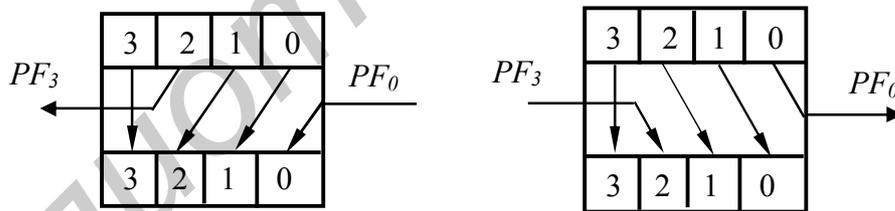


Рис. 4.4

Выполнение сдвиговых операций в процессоре K1804BC2 осуществляется под управлением сигналов микрокоманды $I_8 \div I_0$, при этом сигнал в блоке управления $\overline{IEN} = 0$, а сигналы микрокоманды $I_0 \vee I_1 \vee I_2 \vee I_3 \vee I_4 = 1$ (табл. 4.6).

Функционирование МПС при выполнении некоторых операций (например, арифметических) зависит от ее положения в системе, поэтому при соединении секций необходимо выполнить их настройку на заданное положение: старшая, средняя или младшая МПС (рис. 4.5). Такая настройка выполняется при помощи выводов \overline{LSS} и $\overline{MSS} / \overline{W}$. В младшей МПС на входе \overline{LSS} устанавливается «0», при этом линия $\overline{MSS} / \overline{W}$ становится выходом \overline{W} , причем необходимо, чтобы $\overline{W} = 0$ для каждого такта, в течение которого происходит запись в

РЗУ процессора, в противном случае $\overline{W} = 1$. Поэтому входы \overline{WE} всех МПС соединяют с выходом \overline{W} Мл.МПС, как показано на рис. 4.5.

В средней и старшей МПС на входе \overline{LSS} устанавливается логическая «1». При этом линия $\overline{MSS}/\overline{W}$ становится входом секции \overline{MSS} . В старшей МПС на этот вход подается уровень нуля, а в средних – уровень логической «1». Это позволяет вывести в средних и младшей МПС на выходы \overline{P}/OVR и \overline{G}/F_3 сигналы ускоренного переноса $\overline{P}, \overline{G}$, а в старшей МПС – сигналы OVR и F_3 .

Таблица 4.5

Код микрокоманды	Операция АЛУ	Выходные сигналы АЛУ					
		z	C_4	\overline{P}/OVR		\overline{G}/F_3	
$I_4I_3I_2I_1I_0$				Ст. МПС	Мл. и Ср. МПС	Ст. МПС	Мл. и Ср. МПС
00000	Специальные функции						
00001	$F_{ALU} := \langle 1111 \rangle$	0	0	0	0	F_3	\overline{G}
00011	$S-R-1+C_0$	z	C_4	OVR	\overline{P}		
00101	$R-S-1+C_0$						
00111	$R+S+C_0$						
01001	$\overline{S}+C_0$						
01011	$\overline{S}+C_0$						
01101	$\overline{R}+C_0$						
01111	$\overline{R}+C_0$						
10001	$F_{ALU} := \langle 0000 \rangle$	1	0	0	0		
10011	$\overline{R} \wedge S$	z	0	0	0		
10101	$R \oplus S$						
10111	$R \oplus S$						
11001	$\overline{R} \wedge S$						
11011	$\overline{R} \wedge S$						
11101	$R \vee S$						
11111	$R \vee S$						

Расширение знака при вычислениях может быть выполнено на несколько разрядов за 1 такт, для чего используется предпоследняя микрокоманда (код $I_8 - I_5 = E$, табл. 4.6). Микрокоманда E на входах «МПС-знака» передает знак с вывода PF_0 МПС на выводы $PF_3, Y_3 - Y_0$, что и приводит к реализации требуемой функции. Однако на входах $I_8 - I_5$ «МПС-данных» в это же время должен быть установлен код числа F , что требует определенной реорганизации шины управления, так как сигнал I_5 в данной микрокоманде должен быть равен нулю и единице одновременно. В целом принцип расширения знака используется не для модификации результата, а для формирования данных. Например, если 16-разрядный процессор должен интерпретировать 8-разрядные данные как двоичные числа со знаком, то знаковый разряд расширяется на один байт.

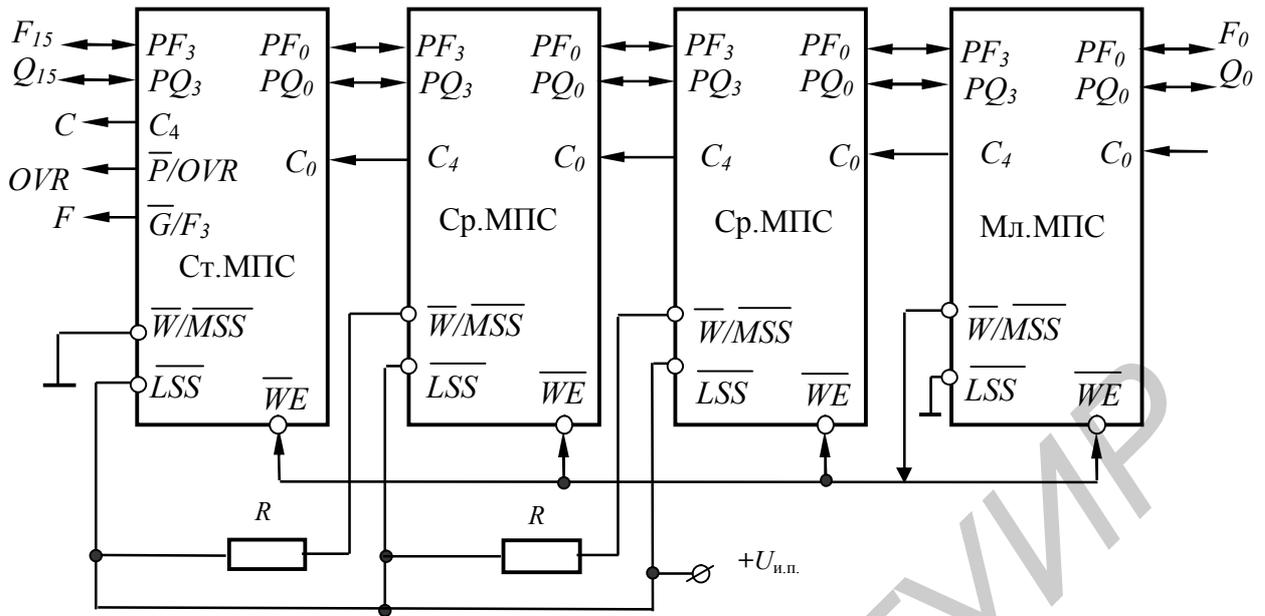


Рис. 4.5

Четыре микрокоманды (коды $I_8 - I_5 = 4, 5, 6, 7$) используются для формирования паритета на шине PF_0 . Паритет представляет собой результат операции $mod 2$ всех выходов АЛУ и сигнала, подаваемого на вход PF_3 . Паритетная логика обладает способностью наращивания путем соединения вывода PF_3 предыдущей МПС с выводом PF_0 последующей. При этом уравнение паритета имеет вид

$$PF_{0 \text{ Мл.МПС}} = F_{15} \oplus F_{14} \oplus \dots \oplus F_1 \oplus F_0 \oplus PF_{3 \text{ Ст.МПС}}$$

Специальные функции, выполняемые МПС, определяются сигналами $I_8 - I_5$ при наличии нулевой комбинации на входах $I_4 - I_0$. Схема может выполнять восемь специальных функций, которые перечислены в табл. 4.10. Восемь кодовых комбинаций не используются и являются запрещенными при функционировании МПС.

1. Умножение без знака $I_8 - I_5 = 0000$. Реализация данной функции предполагает выполнение базовых операций сдвига и сложения. Для умножения чисел с разрядностью, кратной четырем ($4n \times 4n$), требуется $4n$ тактов работы МПС. При выполнении умножения предполагается, что регистр R_0 РЗУ предварительно был очищен и далее будет использоваться для размещения старших бит частичных произведений результата. Множимое записывается в регистр R_1 , а множитель – в регистр R_2 . Регистры R_0 и R_1 могут быть расположены в РЗУ МПС и адресованы по входам A и B соответственно, кроме того, регистры R_0 и R_1 могут располагаться и во внешней памяти секции. В этом случае регистр R_1 используется как источник операнда R и соединяется с шиной $DA_0 - DA_3$, а регистр R_0 – как источник операнда S и как приемник результата и соединяется с шинами $DB_0 - DB_3$ и $Y_3 - Y_0$.

Таблица 4.6

I_8-I_5	Функц. СДА	Функц. СДР	Y_3		Y_2		Y_1	Y_0	PF_3		PF_0	\overline{W} $\neg LSS=0$	$RG Q$	
			Ст. МПС	Ср., Мл. МПС	Ст. МПС	Ср., Мл. МПС			Ст.МПС	Ср., Мл. МПС			PQ_3	PQ_0
0000	$F/2 \rightarrow Y$ Ар. С	$Q \rightarrow Q$	F_3	PF_3	PF_3	F_3	F_2	F_1	Вх.	Вх.	F_0	0	R_{off}	
0001	$F/2 \rightarrow Y$ Лог. С	$Q \rightarrow Q$	PF_3	PF_3	F_3	F_3	F_2	F_1	Вх.	Вх.	F_0	0	R_{off}	
0010	$F/2 \rightarrow Y$ Ар. С	$Q/2 \rightarrow Q$ Лог.С	F_3	PF_3	PF_3	F_3	F_2	F_1	Вх.	Вх.	F_0	0	Вх.	Q_0
0011	$F/2 \rightarrow Y$ Лог. С	$Q/2 \rightarrow Q$ Лог.С	PF_3	PF_3	F_3	F_3	F_2	F_1	Вх.	Вх.	F_0	0	Вх.	Q_0
0100	$F \rightarrow Y$	$Q \rightarrow Q$	F_3	F_3	F_2	F_2	F_1	F_0	Вх.	Вх.	P	0	R_{off}	
0101	$F \rightarrow Y$	$Q/2 \rightarrow Q$ Лог. С	F_3	F_3	F_2	F_2	F_1	F_0	Вх.	Вх.	P	1	Вх.	Q_0
0110	$F \rightarrow Y$	$F \rightarrow Q$	F_3	F_3	F_2	F_2	F_1	F_0	Вх.	Вх.	P	1	R_{off}	
0111	$F \rightarrow Y$	$F \rightarrow Q$	F_3	F_3	F_2	F_2	F_1	F_0	Вх.	Вх.	P	0	R_{off}	
1000	$2F \rightarrow Y$ Ар. С	$Q \rightarrow Q$	F_3	F_2	F_1	F_1	F_0	PF_0	F_2	F_3	Вх.	0	R_{off}	
1001	$2F \rightarrow Y$ Лог. С	$Q \rightarrow Q$	F_2	F_2	F_1	F_1	F_0	PF_0	F_3	F_3	Вх.	0	R_{off}	
1010	$2F \rightarrow Y$ Ар. С	$2Q \rightarrow Q$ Лог. С	F_3	F_2	F_1	F_1	F_0	PF_0	F_2	F_3	Вх.	0	Q_3	Вх.
1011	$2F \rightarrow Y$ Лог. С	$2Q \rightarrow Q$ Лог. С	F_2	F_2	F_1	F_1	F_0	PF_0	F_3	F_3	Вх.	0	Q_3	Вх.
1100	$F \rightarrow Y$	$Q \rightarrow Q$	F_3	F_3	F_2	F_2	F_1	F_0	F_3	R_{off}		1	R_{off}	
1101	$F \rightarrow Y$	$2Q \rightarrow Q$ Лог. С	F_3	F_3	F_2	F_2	F_1	F_0	R_{off}			1	Q_3	Вх.
1110	$PF_0 \rightarrow$ $Y_0 Y_1 Y_2 Y_3$	$Q \rightarrow Q$	PF_0	PF_0	PF_0	PF_0	PF_0	PF_0	PF_0	PF_0	Вх.	0	R_{off}	
1111	$F \rightarrow Y$	$Q \rightarrow Q$	F_3	F_3	F_2	F_2	F_1	F_0	R_{off}			0	R_{off}	

Примечания.

X – произвольное состояние, определяемое внешними сигналами;

P – признак равенства (паритет).

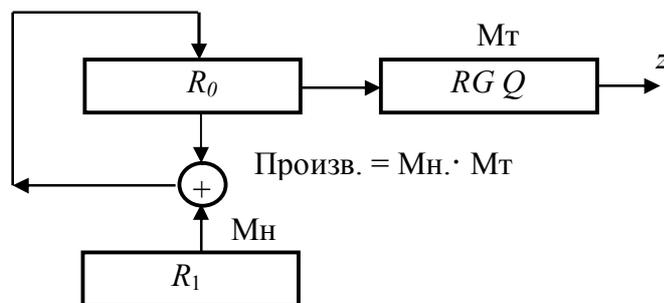


Рис. 4.8

При использовании типичного устройства управления для выполнения умножения без знака в МПС требуется хранить только две микрокоманды (табл. 4.7). Общее время умножения при этом составит $4n + 1$ тактов.

Таблица 4.7

Адрес	Микрокоманда								Комментарий
	I_0	$I_4I_3I_2I_1$	$I_8I_7I_6I_5$	\overline{OEB}	\overline{OEY}	$A_3 - A_0$	$B_3 - B_0$	C_0	
A_1	X	0110 $R + C_0$	0110 $F \rightarrow Y$ $F \rightarrow Q$	X	X	0010 R_2	XXXX	0	$RG_2 \rightarrow RGQ$ ($M_T \rightarrow RGQ$)
A_2	0 $S = B$	0000 $R + S + C_0$ $z = 1$ $S + C_0$	0000 $S = B$	0 $\frac{1}{2} F$ $\rightarrow Y$	0	0001 R_1	0000 RG_0	0	Умножение $R_0 := R_0 + R_1$

2. Умножение в дополнительном коде $I_8I_7I_6I_5 = 0010$. Данный алгоритм выполняется аналогично предыдущему, за исключением последнего такта, на котором производится коррекция результата. Начальная установка и первые $4n-1$ тактов умножения выполняются точно так же, как и при умножении чисел без знака. На $4n$ -м такте знаковый разряд множителя находится на шине z Мл.МПС. В это время необходимо подать микрокоманду последнего такта умножения в дополнительном коде или выполнить коррекцию результата. Объединение МПС при выполнении умножения в дополнительном коде представлено на рис. 4.9.

При умножении на входе C_0 процессора должен присутствовать уровень нуля до последнего такта. На последнем такте на данном входе должен присутствовать сигнал, соответствующий логическому уровню линии z . В связи с этим на входе C_0 устанавливаются элемент 2И с входными сигналами z и разрешение. Сигнал «Разрешение» устанавливается в единичное состояние только на последнем такте умножения. В остальных тактах линия z используется так же, как и в предыдущей команде. На последнем такте умножения в дополнительном коде АЛУ реализует функцию $F = S + C_0$ при $z = 0$ и $F = S - R - 1 + C_0$ при $z = 1$. Таким образом, если множитель положительный,

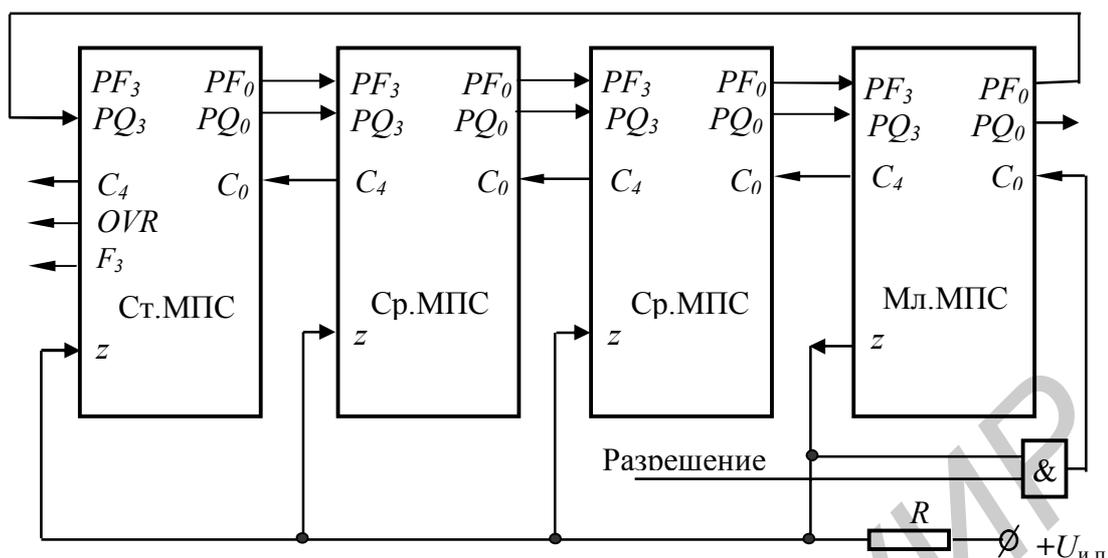


Рис. 4.9

то процесс умножения на этом такте заканчивается, а если множитель отрицательный, то множимое вычитается из полученного частичного произведения, после чего умножение также заканчивается. В конкретных реализациях процессоров сигнал «Разрешение» заменяется микропрограммным управлением от схемы СУСС K1804BP2.

Пример. Пусть $RG Mn = 7$, $RG Mt = -5$. Тогда в результате умножения получим произведение $[-RG Sm]_d = 1.1011101$. В соответствии с рис. 4.8 процесс формирования дополнительного кода результата будет иметь вид

$$\begin{array}{r}
 00000000 \quad RG Sm \\
 1 \text{ step: } \underline{01110000 \quad +Mn} \\
 00111000 \quad RG Sm + Mn, RG Sm \cdot 2; \\
 2 \text{ step: } \underline{01110000 \quad +Mn} \\
 01010100 \quad RG Sm + Mn, RG Sm \cdot 2; \\
 3 \text{ step: } \underline{00000000 \quad +0} \\
 00101010 \quad RG Sm + Mn, RG Sm \cdot 2; \\
 4 \text{ step: } \underline{01110000 \quad +Mn} \\
 01001101 \quad RG Sm + Mn, RG Sm \cdot 2; \\
 5 \text{ step: } \underline{10010000 \quad [-Mn]_d} \\
 11011101 \quad RG Sm - Mn = [-Sm]_d.
 \end{array}$$

3. Нормализация чисел обычной и двойной длины. Операция нормализации чисел в дополнительном коде выполняется путем сдвига числа в сторону старших разрядов до тех пор, пока два старших бита не будут иметь

различные значения. При этом знак нормализуемого операнда фиксируется в старшем разряде.

Нормализация выполняется для чисел как обычной, так и двойной длины. При нормализации чисел обычной длины используется только регистр Q , а при нормализации чисел двойной длины необходимо использовать еще АЛУ и блок внутренней памяти. В первом случае операнд помещается в регистр Q и по положительному фронту тактового сигнала выполняется его сдвиг в сторону старших разрядов. Режим сдвига реализуется до получения в двух старших битах регистра Q различных значений (0.1... или 1.0...), при этом через вывод PQ_0 МПС в младшие разряды регистра Q вдвигаются нулевые значения. Появление «1» на выходе C_4 Ст.МПС означает окончание операции нормализации; это обусловлено тем, что в Ст.МПС значение переноса формируется по закону $C_4 = Q_3 \oplus Q_2$. Кроме того, по признаку OVR можно определить предпоследний такт нормализации, так как $OVR = Q_2 \oplus Q_1$, т.е. опережает C_4 на один такт.

Пример.	0.000001011000111	–	0.101100011100000
	1.110101101111101	–	1.010110111110100
	0.000000000000000	–	не нормализуется
	1.111111111111111	–	1.000000000000000

Чтобы не выполнять нормализацию нулевого операнда, необходимо идентифицировать соответствующую ситуацию установкой какого-либо признака. С этой целью используется схема ФПН, на выходе z которого устанавливается «1», если все выходы регистра Q нулевые.

При появлении единицы на выходе C_4 операция нормализации прекращается, а на выходе F_3 Ст.МПС устанавливается знак числа, хранящегося в $RGQ (Q_3 \rightarrow F_3)$.

При выполнении операции нормализации чисел обычной длины можно подсчитать количество тактов, необходимое для выполнения данной операции. С этой целью на входе C_0 МПС устанавливается логическая единица, при этом регистр РЗУ, адресованный по каналу B , становится счетчиком тактов. В данном случае АЛУ выполняет функцию $F = S + C_0$.

Соединение выводов МПС при выполнении операции нормализации чисел обычной длины имеет вид, показанный на рис. 4.10.

При выполнении сдвигов счетчик тактов может быть организован во внешней памяти. В данном случае его содержимое передается на S вход АЛУ через внешнюю шину $DB_3 - DB_0$, а результат выводится на шину $Y_3 - Y_0$.

При выполнении нормализации чисел двойной длины старшие разряды числа находятся в ячейке РЗУ, адресуемой по B , а младшие – в RGQ . Вывод PQ_3 старшей МПС должен быть соединен с выводом PF_0 младшей МПС. На

5. Инкрементирование операнда на 1 или 2. Увеличение числа на единицу или двойку выполняется за один такт. Модифицируемый операнд подается на вход S АЛУ (например, из регистра RGB БВП, из RGQ или с внешней шины $DB_3 - DB_0$), после чего реализуется функция $F = S + C_0 + 1$, что при $C_0 = 1$ предполагает увеличение операнда на 2, а при $C_0 = 0$ – на единицу. Как правило, инкрементирование на единицу или двойку выполняется в процессе модификации счетчика команд при реализации адресации к полуслову или слову в ОЗУ. При этом счетчик команд может быть организован в РЗУ компьютера.

При выполнении данного преобразования используется код микрокоманды $I_8 - I_5 = 0100$ и стандартное включение МПС в структуру БОД.

6. Деление в дополнительном коде. Данная специальная функция выполняется в соответствии с микрокомандой, представленной в табл. 4.8. Однако в целом операция деления требует отработки не одной микрокоманды, а определенной микропрограммы, хранимой в МПП.

Таблица 4.8

Адрес	Микрокоманда								Комментарий
	I_0	$I_4I_3I_2I_1$	$I_8I_7I_6I_5$	\overline{EA}	\overline{OEY}	$A_3 - A_0$	$B_3 - B_0$	C_0	
A_1	0 $S = B$	0000	1100 $S + R + C_0$ $z = 0$ $S - R - 1 + C_0$ $z = 1$	0 $R = A$	X	0000 R_0 (Дт)	0001 R_1 (Дм. ст. часть)	z	Деление в доп. коде

Базовая схема соединения выводов МПС при выполнении операции деления в дополнительном коде показана на рис. 4.13.

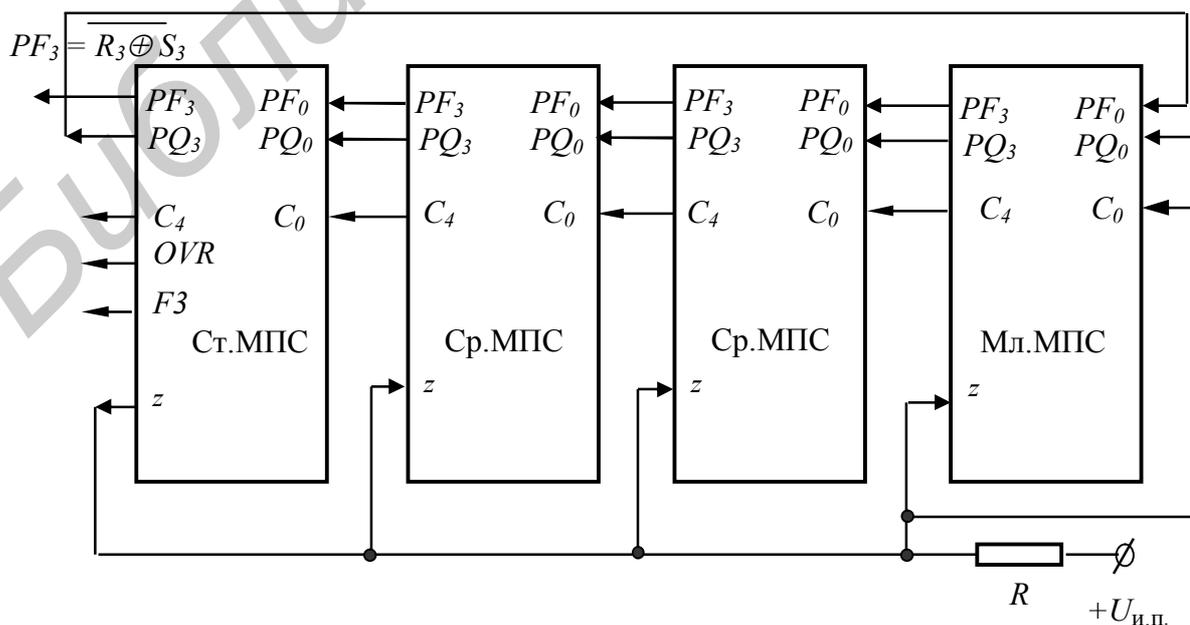


Рис. 4.13

В качестве примера, поясняющего данный алгоритм, рассмотрим принцип получения частного от деления двух целых отрицательных чисел:

$$[Dm]_d = 1.1011011_2 = -37_{10}, \quad [Dt]_d = 1.1010000_2 = -48_{10}.$$

Используемые числа не нормализованы.

На первом этапе при частном $Sht = 0$ и $z = 1$ выполняется функция $S - R - 1 + C_0$ при $C_0 = z = 1$, что предполагает формирование прямого кода делителя в виде

$$z = \overline{R_3 \oplus S_3}, \quad z = 1 \begin{cases} 1.101\ 1011\ [Dm]_d = -37 \\ 1.101\ 0000\ [Dt]_d = -48 \end{cases}$$

$$Sht = 0 \quad \begin{array}{r} 1.101\ 1011\ [Dm]_d = -37 \\ \underline{0.011\ 0000\ Dt = 48} \\ 0.000\ 1011\ Ost = 11 \end{array}$$

$Sign\ Ost \neq SignDm$

1) $Sht_1 = Sht + 1 = 1.$

При выполнении второго этапа деления необходимо выполнить масштабирование остатка путем сдвига на два разряда влево до совпадения единичных значений в старших разрядах прямых кодов делимого и делителя. После этого выполняется вычитание делителя из остатка в соответствии с вышеприведенным алгоритмом.

$$Sht_1 = 4Sht_1 = 4 \quad z = 0 \begin{cases} 0.010\ 1100\ Dm = 4 \cdot Ost = 44 \\ 1.101\ 0000\ [Dt]_d = -48 \end{cases}$$

$$Sign\ Ost \neq SignDm \quad \begin{array}{r} 1.111\ 1100\ [Ost]_d = -4 \\ \underline{} \\ 2) Sht_2 = 4 - 1 = 3 \\ Sht_2 = Sht_1 - 1 \cdot 2^{-2} = 0,11. \end{array}$$

В процессе преобразования частного следует помнить, что формируемое значение представляет собой набор двоичных чисел, расположенных после запятой. Данный факт предполагает интерпретацию результата деления как дроби, а не как целого числа, расположенного в регистре частного. Остальные этапы получения частного имеют следующий вид.

$$Sht_2 = 8Sht_2 = 24 \quad z = 0 \begin{cases} 1.110\ 0000\ [Dm]_d = 8 \cdot [Ost]_d = -32 \\ 0.011\ 0000\ Dt = 48 \end{cases}$$

$$Sign\ Ost \neq SignDm \quad \begin{array}{r} 0.001\ 0000\ Ost = 16 \\ 3) Sht_3 = 24 + 1 = 25 \\ Sht_3 = Sht_2 + 1 \cdot 2^{-3} = 0,11001. \\ Sht_3 = 2Sht_3 = 50 \quad z = 0 \begin{cases} 0.010\ 0000\ Dm = 2Ost = 32 \\ 1.101\ 0000\ [Dt]_d = -48 \end{cases} \\ Sign\ Ost \neq SignDm \quad \begin{array}{r} 1.111\ 0000\ [Ost]_d = -16 \\ 4) Sht_4 = Sht_3 - 1 = 49 \\ Sht_4 = Sht_3 - 1 \cdot 2^{-1} = 0,110001. \end{array} \end{array}$$

$$Sht_4 = 2Sht_4 = 98. \quad z = 1 \quad \begin{cases} 1.110\ 0000 & [Dm]_d = 2 \cdot [Ost]_d = -32 \\ 1.101\ 0000 & [Dt]_d = -48 \end{cases}$$

$$z = 0 \quad \begin{cases} 1.110\ 0000 & [Dm]_d = -32 \\ 0.011\ 0000 & Dt = 48 \end{cases}$$

$$Sign\ Ost \neq Sign\ Dm \quad 0.001\ 0000 \quad Ost = 16$$

$$5) \quad Sht_5 = Sht_4 + 1 \cdot 2^{-1} = 99$$

$$Sht_5 = Sht_4 + 1 \cdot 2^{-1} = 0,1100011.$$

$$Sht_5 = 2Sht_5 = 198. \quad z = 0 \quad \begin{cases} 0.010\ 0000 & Dm = 2 \cdot Ost = 32 \\ 1.101\ 0000 & [Dt]_d = -48 \end{cases}$$

$$Sign\ Ost \neq Sign\ Dm \quad 1.111\ 0000 \quad [Ost]_d = -16$$

$$6) \quad Sht_6 = Sht_5 - 1 \cdot 2^{-1} = 197$$

$$Sht_6 = Sht_5 - 1 \cdot 2^{-1} = 0,11000101.$$

Погрешность результата после шести этапов деления будет равна

$$\Delta = \frac{1}{Sht_\infty} (Sht_\infty - Sht_6) \cdot 100\ \% = 0,16891891\ \dots\ \%$$

Сводные сведения о специальных функциях представлены в табл. 4.9.

Таблица 4.9

Мк. код.	Функция АЛУ	PF ₃		PF ₀	PQ ₃	PQ ₀	\bar{W}	C ₄	\bar{P} / OVR		\bar{G} / F_3		z		
		См. МПС	Мл. Ср. МПС						См. МПС	Мл. Ср. МПС	См. МПС	Ср. МПС	Мл. МПС		
1. Умножение без знака															
0000	$F = S + C_0$ при $z = 0$, $F = S + R + C_0$ при $z = 1$	R _{off}	Bx	F ₀	Bx	Q ₀	0	C ₄	OVR	\bar{P}	F ₃	\bar{G}	Bx	Bx	Q ₀
2. Умножение в дополнительном коде															
0010	$F = S + C_0$ при $z = 0$, $F = S + R + C_0$ при $z = 1$	R _{off}	Bx	F ₀	Bx	Q ₀	0	C ₄	OVR	\bar{P}	F ₃	\bar{G}	Bx	Bx	Q ₀
3. Инкрементирование															
0100	$F = S + 1 + C_0$	Bx	Bx	P	R _{off}	R _{off}	0	C ₄	OVR	\bar{P}	F ₃	\bar{G}	z	z	z

Мк. код	Функция АЛУ	PF_3		PF_0	PQ_3	PQ_0	\bar{W}	C_4	\bar{P}/OVR		\bar{G}/F_3		z		
		Ст. МПС	Мл. Ср. МПС						Ст. МПС	Мл. Ср. МПС	Ст. МПС	Ср. МПС	Мл. МПС		
4. Преобразование «число со знаком – дополнительный код»															
0101	$F = S + C_0$ при $z = 0$, $F = \bar{S} + C_0$ при $z = 1$	Bx	Bx	P	R_{off}	R_{off}	0	C_4	OVR	\bar{P}	*	\bar{G}	S_3	Bx	Bx
5. Последний цикл умножения в дополнительном коде															
0110	$F = S + C_0$ при $z = 0$, $F = S - R -$ $-1 + C_0$ при $z = 1$	R_{off}	Bx	F_0	Bx	Q_0	0	C_4	OVR	\bar{P}	F_3	\bar{G}	Bx	Bx	Q_0
6. Нормализация обычной длины															
0111	$F = S + C_0$	F_3	F_3	R_{off}	Q_3	Bx	0	**	$Q_2 \oplus Q_1$	\bar{P}	Q_3	\bar{G}	***		
7. Нормализация двойной длины															
1010	$F = S + C_0$	$R_3 \oplus F_3$	F_3	Bx	Q_3	Bx	0	** **	$F_2 \oplus F_1$	\bar{P}	F_3	\bar{G}	*****		
8. Деление в дополнительном коде															
1100	$F = S + R + C_0$ при $z = 0$, $F = S - R -$ $-1 + C_0$ при $z = 1$	$R_3 \oplus S_3$	F_3	Bx	Q_3	Bx	0	C_4	OVR	\bar{P}	F_3	\bar{G}	*****	Bx	Bx
<p>Примечания.</p> <p>* F_3, если $z = 0$; $S_3 \oplus F_3$, если $z = 1$.</p> <p>** $Q_3 \oplus Q_2$ – для Ст. МПС, C_4 – для остальных МПС.</p> <p>*** z регистра Q.</p> <p>**** $F_3 \oplus F_2$ – для Ст. МПС, C_4 – для остальных МПС.</p> <p>***** z выходов АЛУ и регистра Q.</p> <p>***** $R_3 \oplus S_3$.</p>															

В схеме процессора К1804ВС2 реализован алгоритм деления в форме «стыковки» чисел частного в процессе вычитания делителя. Это приводит к совпадению многих бит реального и вычисленного результатов, однако, по сути, данная методология приводит к неточным вычислениям. Правильным следует считать алгоритм, реализующий знакопеременную коррекцию частного, формируемого в *RG Sht* [9].

4.3. БИС управления последовательностью микрокоманд УПМ К1804ВУ4

Двенадцатиразрядная схема управления последовательностью микрокоманд (УПМ) предназначена для построения блоков микропрограммного управления цифровых устройств. Основная функция схемы УПМ заключается в формировании последовательности адресов при адресации микрокоманд, хранящихся в микропрограммной памяти. В качестве внешних сигналов, воздействующих на БИС К1804ВУ4, предполагается использование микрокомандного управляющего слова, формирующего шину управления спецкомпьютера. Схема К1804ВУ4 построена на базе триггеров, срабатывающих по переднему фронту синхросигнала (рис. 4.14).

Архитектурно БИС включает себя СУАМ и управляющее ПЗУ, что позволило повысить уровень интеграции комплекта и сократить затраты на проектирование внутреннего интерфейса компьютера.

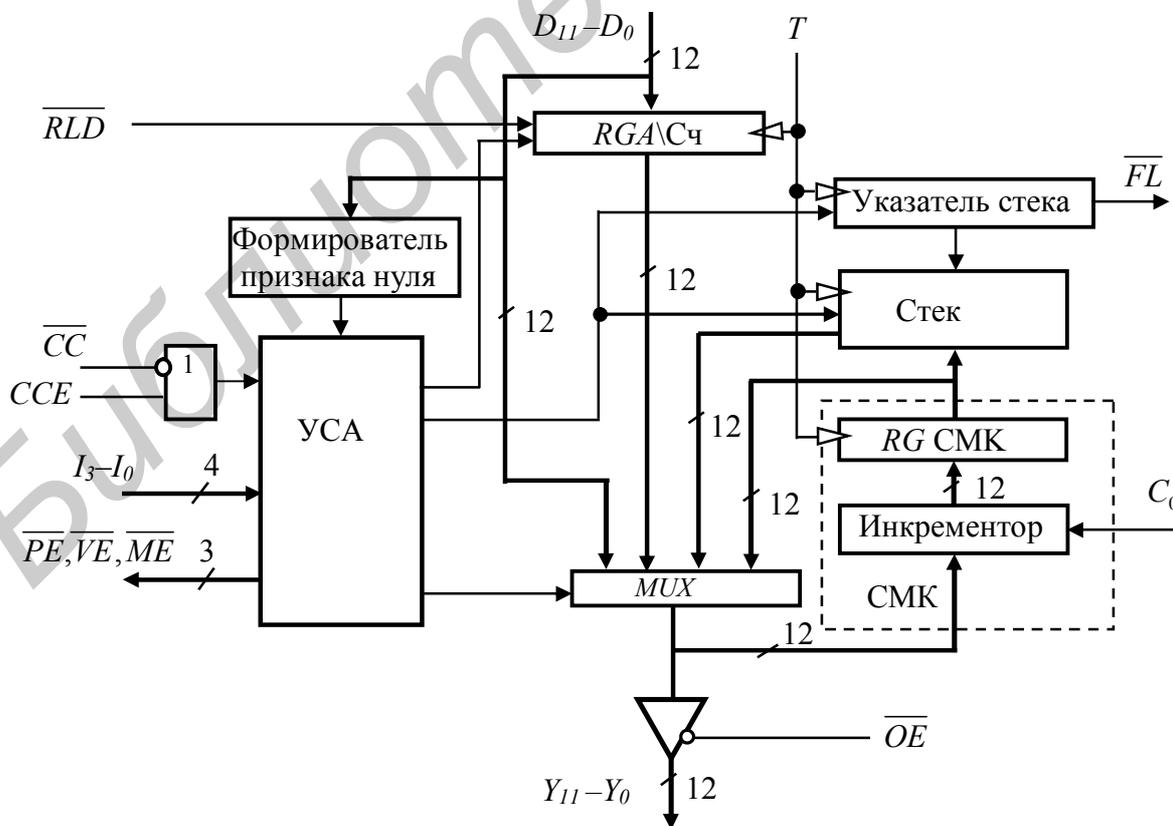


Рис. 4.14

БИС K1804ВУ4 имеет следующие архитектурные особенности:

- 1) двенадцатиразрядную внутреннюю и внешнюю шину данных, что позволяет адресовать 4096 слов в ПЗУ микрокоманд;
- 2) четыре встроенных источника адреса: внешнюю шину $D_{11}-D_0$, регистр адреса/счетчик, счетчик микрокоманд, стек глубиной пять;
- 3) возможность выполнения 16 инструкций управления, большинство из которых являются условными;
- 4) возможность арбитража внешней шины адреса ветвления между тремя источниками с помощью выходных сигналов схемы управления следующим адресом (УСА);
- 5) трехстабильные выходы.

Мультиплексор УПМ предназначен для выбора в качестве источника адреса следующей микрокоманды одной из четырех компонентов схемы. Источник выбирается в зависимости от состояния шины микрокоманды I_3-I_0 , а также управляющих сигналов \overline{CS} и CCE в соответствии с табл. 4.10. Выбранный с помощью мультиплексора адрес поступает на выходную трехстабильную шину $Y_{11}-Y_0$. При нуле на входе выдачи адреса (т.е. при $\overline{OE}=0$) разрешается вывод информации на шину Y , а при единичном значении сигнала выходная шина переводится в состояние R_{off} .

Регистр адреса/счетчик состоит из 12 триггеров D -типа, запись информации в которые производится по положительному фронту тактового сигнала при соответствующей инструкции или при $\overline{RLD}=0$ вне зависимости от инструкции. В зависимости от выполняемой инструкции $RG\ A\ Cч$ используется или в качестве буфера для записи и хранения адреса перехода либо числа циклов, принимаемых через шину D . При этом счетчик циклов может инкрементироваться или в каждом цикле выполнения микропрограммы или в каждом такте функционирования БИС. Если в $RG\ A\ Cч$ загружено число N , то при соответствующей инструкции цикл будет выполнен $N+1$ раз. Равенство нулю содержимого $RG\ A\ Cч$ является тестовым условием при выполнении некоторых условных инструкций схемы УПМ (см. табл. 4.10). Настройка $RG\ A\ Cч$ на тот или иной режим работы производится под действием внутренних управляющих сигналов, вырабатываемых с использованием микрокоманды I_3-I_0 .

Счетчик микрокоманд состоит из регистра счетчика микрокоманд ($RG\ CMK$) и схемы приращения – инкрементора. Любой текущий адрес с выхода мультиплексора передается через инкрементор в $RG\ CMK$ для запоминания.

Схема приращения имеет вход переноса C_0 . При нуле на входе C_0 адрес с выхода мультиплексора передается на вход $RG\ CMK$ без изменения. Это означает, что одна и та же микрокоманда может быть выполнена компьютером любое число раз. При единице на входе C_0 происходит увеличение адреса, передаваемого с выхода мультиплексора на вход $RG\ CMK$. Таким образом, имеется возможность выполнения последовательности микрокоманд в микропрограмме любой длины.

Таблица 4.10

I_3-I_0	Мнемоника	Содерж. инструкц.	Содерж. $RGA\setminus Cч$	$CCE = 0, \overline{CC} = 1$		$CCE = 0, \overline{CC} = 0$		$RGA\setminus Cч$	Активированный выход
				Y	Стек	Y	Стек		
0000	Jz	Переход к нулевому адресу	X	0	Сброс	0	Сброс	Хранен.	\overline{PE}
0001	CJS	Условный переход к подпрогр.	X	CMK	Хранен.	D	Загрузка	Хранен.	\overline{PE}
0010	$JMAP$	Переход по адресу из ПНА	X	D	Хранен.	D	Хранен.	Хранен.	\overline{ME}
0011	CJP	Условный переход по адресу из $RGMk$	X	CMK	Хранен.	D	Хранен.	Хранен.	\overline{PE}
0100	$PUSH$	Запись в стек и условная загрузка $RGA\setminus Cч$	X	CMK	Загрузка	CMK	Загрузка	Прим.1	\overline{PE}
0101	$JSRP$	Условный переход к одной из двух подпрограмм	X	$RGA\setminus Cч$	Загрузка	D	Загрузка	Хранен.	\overline{PE}
0110	CJV	Условный переход на адрес вектора прерыв.	X	CMK	Хранен.	D	Хранен.	Хранен.	\overline{VE}
0111	JRP	Условный переход на адрес из $RGA\setminus Cч$ или $RGMk$	X	$RGA\setminus Cч$	Хранен.	D	Хранен.	Хранен.	\overline{PE}
1000	$RFCT$	Повторить цикл, если $RGA\setminus Cч \neq 0$	$\neq 0$ $=0$	Стек CMK	Хранен. Выт.алк.	Стек CMK	Хранен. Выт.алк.	Декрем. Хранен.	\overline{PE} \overline{PE}

I_3-I_0	Мнемоника	Содерж. инструкц.	Содерж. $RGA \setminus Cч$	$CCE = 0, \overline{CC} = 1$		$CCE = 0, \overline{CC} = 0$		$RGA \setminus Cч$	Активированный выход
				Y	Стек	Y	Стек		
0111	<i>JRP</i>	Условный переход на адрес из $RGA \setminus Cч$ или $RGMk$	X	$RGA \setminus Cч$	Хранен.	D	Хранен.	Хранен.	\overline{PE}
1000	<i>RFCT</i>	Повторить цикл, если $RGA \setminus Cч \neq 0$	$\neq 0$ $= 0$	Стек СМК	Хранен. Выталк.	Стек СМК	Хранен. Выталк.	Декрем. Хранен.	\overline{PE} \overline{PE}
1001	<i>RPCT</i>	Повторить адрес из $RGMk$ при $RGA \setminus Cч \neq 0$	$\neq 0$ $= 0$	D СМК	Хранен. Хранен.	D СМК	Хранен. Хранен.	Декрем. Хранен.	\overline{PE} \overline{PE}
1010	<i>CRTN</i>	Условный возврат из подпрогр.	X	СМК	Хранен.	Стек	Выталк.	Хранен.	\overline{PE}
1011	<i>CJPP</i>	Условный переход на адрес из $RGMk$ и выталкив. из стека	X	СМК	Хранен.	D	Выталк.	Хранен.	\overline{PE}
1100	<i>LDCT</i>	Загрузить счетчик и продолжить	X	СМК	Хранен.	СМК	Хранен.	Загрузка	\overline{PE}
1101	<i>LOOP</i>	Контроль конца цикла	X	Стек	Хранен.	СМК	Выталк.	Хранен.	\overline{PE}
1110	<i>CONT</i>	Продолж.	X	СМК	Хранен.	СМК	Хранен.	Хранен.	\overline{PE}
1111	<i>TWB</i>	Ветвление на три направления	$\neq 0$ $= 0$	Стек D	Хранен. Выталк.	СМК СМК	Выталк. Выталк.	Декрем. Хранен.	\overline{PE} \overline{PE}
Примечание. Если $CCE = 0, \overline{CC} = 1$, то хранение, если $CCE = 0, \overline{CC} = 0$, то загрузка.									

Схема УПМ К1804ВУ4 не вырабатывает сигнала выходного переноса, что является препятствием для наращивания разрядности свыше 12 бит. Однако основная масса микропрограмм практически всегда содержит не более 4096 микрокоманд. В случае, если микропрограммное обеспечение превышает допустимый объем ПЗУ, следует организовать многостраничную подсистему с расширенной адресной шиной микропрограммной памяти.

При выполнении инструкции Jz (табл. 4.10) под действием внутренних управляющих сигналов происходит установка СМК в нуль.

Стек состоит из указателя стека, дешифратора, накопителя и схемы записи\считывания. Стек предназначен для запоминания адреса возврата при выполнении подпрограмм и циклов [10].

Указатель стека представляет собой реверсивный счетчик. Изменение информации в нем происходит по положительному фронту тактового сигнала T . Указатель стека всегда указывает на последнее записанное в накопитель слово.

Схема записи\считывания обеспечивает необходимую коммутацию для выполнения передачи информации из СМК в накопитель стека (при записи в стек) или из накопителя на вход мультиплексора (при считывании из стека).

Управление стеком осуществляется сигналами $I_3 - I_0$, \overline{CS} и CCE в соответствии с табл. 4.10.

Возможны следующие режимы работы стека:

- 1) очистка стека (путем установки в нуль УС);
- 2) хранение (содержимое УС не изменяется, при этом возможно считывание из регистра накопителя);
- 3) загрузка стека (при этом по положительному фронту тактового сигнала T происходит увеличение содержимого указателя стека по правилу $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, а по спаду синхроимпульса запись адреса в регистр накопителя). После заполнения стека на выходе \overline{FL} указателя стека появляется предупреждающий сигнал $\overline{FL} = 0$. Любая загрузка в заполненный стек стирает информацию в вершине стека, оставляя неизменным УС;

4) выталкивание из стека (при этом происходит считывание информации из вершины стека, после чего по спаду синхроимпульса производится уменьшение указателя по правилу $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$; если стек пуст, происходит считывание нулевого слова, а содержимое УС остается равным нулю.

Формирователь признака нуля вырабатывает внутренний управляющий сигнал при равенстве нулю содержимого RGA Сч.

Схема управления следующим адресом (УСА) вырабатывает три сигнала \overline{PE} , \overline{ME} , \overline{VE} , которые используются для отпириания одного из трех внешних источников, подключенных к шине D (регистра микрокоманд, преобразователя начального адреса и выходов регистра прерываний). Каждой микрокомандой вырабатывается один и только один сигнал разрешения для внешнего источника, чаще всего для регистра микрокоманд $\overline{PE} = 0$. При этом вы-

ходы остальных двух внешних источников, подключенных к шине D , должны находиться в третьем состоянии.

Схема УПМ К1804ВУ4 позволяет реализовать 16 инструкций управления последовательностью микрокоманд. Четыре инструкции являются безусловными, а остальные – условными. Десять инструкций являются функционально зависимыми от внешних условий, поступающих на вход \overline{CC} . Считается, что условие не выполнено, если $\overline{CC}=1, CCE=0$. В противном случае $\overline{CC}=0$ или $CCE=1$, условие считается выполненным. Кроме того, установив на входе $CCE=1$, можно условную инструкцию обратить в безусловную (т.е. независимо от состояния входа условия будет выполняться операция, соответствующая выполнению условия). Если условные инструкции в компьютере никогда не выполняются безусловно, то на вход CCE можно подать «0».

Действие трех инструкций $I_3 - I_0 = 8, 9, 15$ из шестнадцати зависит от содержимого $RGA \setminus Cч$, которое уменьшается до тех пор, пока не станет равным нулю. Эти микрооперации используются для организации заданного числа циклов. При этом действие инструкции 15 зависит и от внешнего кода условия и от содержимого $RGA \setminus Cч$ (рис. 4.15).

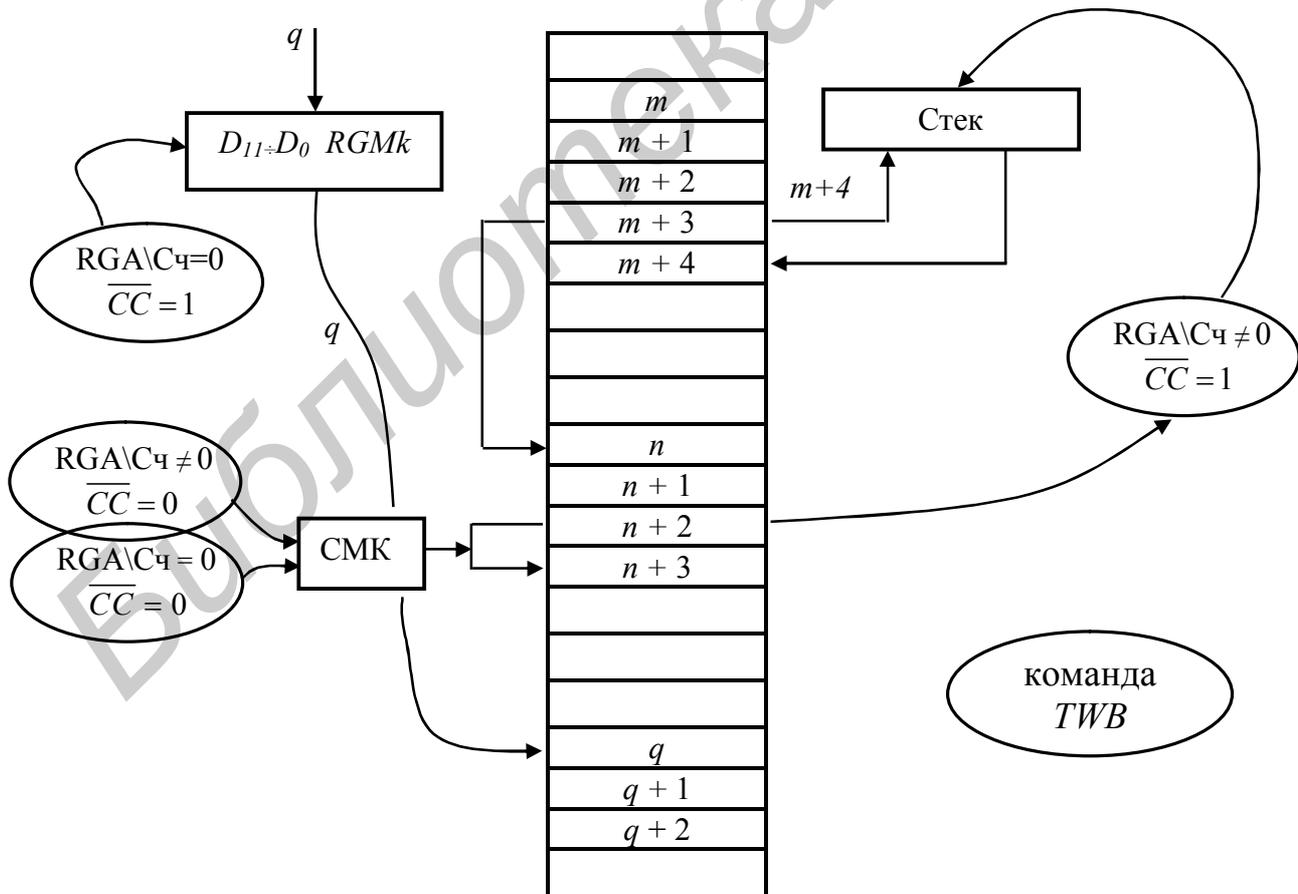


Рис. 4.15

Назначение выводов ИМС приводится в табл. 4.11.

Таблица 4.11

Обозначение выводов	Наименование выводов	Назначение выводов
$D_{11}-D_0$	Прямой вход адреса	Используется как один из источников адреса следующей микрокоманды и как информационный вход при записи адреса в регистр адреса/счетчик. Определяет одну из 16 инструкций схемы УПМ.
I_3-I_0	Вход инструкции	
\overline{CC} CCE	Входы условия и разрешения условия	Оба входа используются для ввода признака выполненной в БОД операции при реализации условных инструкций. Если $\overline{CC} = 1$ и $CCE = 0$, считается, что условие не выполнено, т. е. блок УСА принимает нулевой бит.
C_0	Вход переноса	Используется как входной перенос в СМК; если $C_0 = 1$, то адрес поступающий в СМК увеличивается на 1, если $C_0 = 0$, то адрес не изменяется.
\overline{RLD}	Вход разрешения записи в $RGA\setminus Cч$	Используется для разрешения записи в регистр адреса/счетчик; если $\overline{RLD} = 0$, то разрешается запись информации со входов $D_{11}-D_0$ вне зависимости от инструкции.
\overline{OE}	Вход разрешения выдачи адреса	Используется для отпирания трехстабильной шины Y . При нуле на входе разрешения выдачи адреса \overline{OE} разрешается вывод адреса на шину Y . При единичном значении сигнала \overline{OE} выходная шина переводится в состояние R_{off} .
$Y_{11}-Y_0$	Выходы адреса	Выходы используются для подключения к адресным входам ПЗУ микрокоманд в блоке микропрограммного управления.
\overline{FL}	Выход признака заполнения стека	Используется для индикации состояния заполнения стека $\overline{FL} = 0$.
\overline{PE}	Выход разрешения $RGMk$	Используется для отпирания $RGMk$ при $\overline{PE} = 0$.
\overline{ME}	Выход разрешения ПНА	Используется для отпирания ПНА при $\overline{ME} = 0$.
\overline{VE}	Выход разрешения ПА	Используется для отпирания ПА при $\overline{VE} = 0$.

4.4. Схема управления состояниями и сдвигами СУСС К1804ВР2

Интегральная микросхема К1804ВР2 предназначена для организации сдвигов и управления признаками процессора с целью ветвления программ и микропрограмм. Структурная схема контроллера СУСС приведена на рис. 4.16.

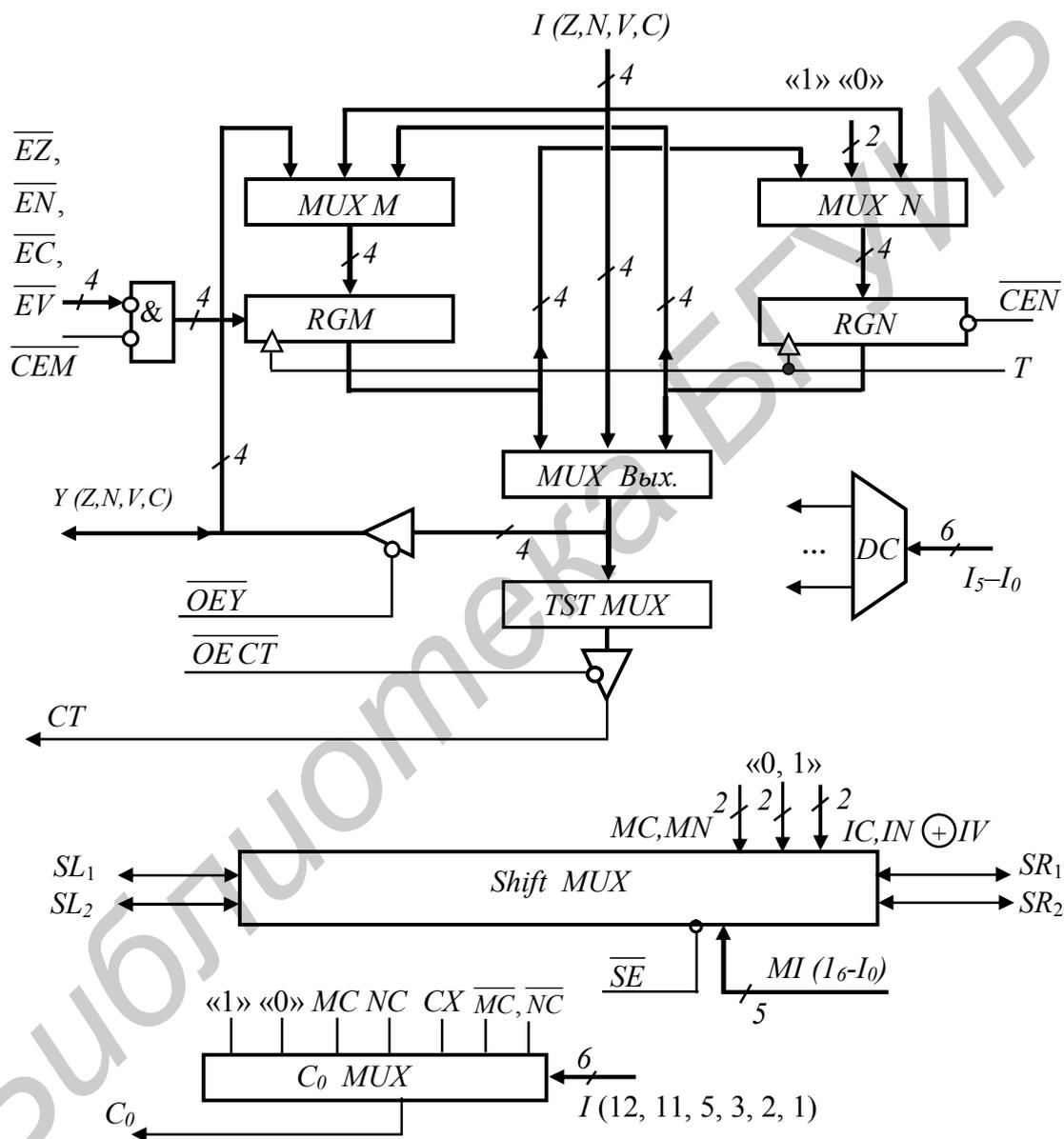


Рис. 4.16

СУСС предназначена для реализации следующих программируемых функций:

- 1) формирования сигнала входного переноса в МПС и схему ускоренного переноса К1804ВР1;

2) организации арифметических, циклических и логических сдвигов обычной и двойной длины (всего 32 варианта);

3) выполнения операций как с целым словом, так и с отдельными битами любого из двух регистров состояния;

4) проверки за один такт одного из 16 различных условий, которые поступают с выхода одного из двух регистров состояния или из МПС.

Назначение выводов ИМС и их нумерация приведены в табл. 4.12.

Таблица 4.12

Обозначение выводов	Номера контактов	Назначение вывода
$I(C, Z, N, V)$	12, 9, 14, 16	Входная 4-разрядная параллельная шина признаков условных переходов
$\overline{EC}, \overline{EZ}, \overline{EN}, \overline{EV}$	11, 8, 13, 15	Входы разрешения записи меток C, Z, N, V в регистр состояния RGM . При «1» на входах $\overline{EC}, \overline{EZ}, \overline{EN}, \overline{EV}$ запись в соответствующие разряды запрещена, а при нуле – разрешена, если одновременно $\overline{CEM} = 0$
\overline{CEM}	7	Вход разрешения записи в RGM ; при «0» на данном входе запись в регистр разрешена, а при «1» – запрещена
\overline{CEN}	2	Вход разрешения записи в RGN ; при «0» на данном входе запись в регистр разрешена, а при «1» – запрещена
$Y(C, Z, N, V)$	31, 32, 29, 28	Двунаправленная 4-разрядная параллельная шина признаков условных переходов. Допускает ввод данных в регистр RGM и вывод данных из RGM, RGN и внешней шины $I(C, Z, N, V)$
\overline{OEY}	20	Вход разрешения вывода информации по шине $Y(C, Z, N, V)$
\overline{OECT}	26	Вход разрешения вывода информации на шину CT
CT	27	Выход кода условия
PF_0, PF_3, PQ_0, PQ_3	35, 36, 33, 34	Двунаправленные выводы сдвига
\overline{SE}	37	Вход разрешения сдвига: если данный сигнал равен «0» – сдвиг разрешен, при $\overline{SE} = 1$ – выводы СУСС находятся в третьем состоянии
CX	24	Вход переноса – используется как один из источников при формировании выходного переноса C_0 из СУСС
C_0	25	Выход переноса для входа C_0 АЛУ
T	17	Вход синхронизации

В схеме СУСС выделяют пять основных блоков: блок обработки признаков, блок проверки условия, блок управления переносом, блок управления сдвигами, блок внутреннего управления.

Блок обработки признаков состоит из двух четырехразрядных регистров состояния RGM и RGN , двух входных $MUX M$ и $MUX N$ и выходного мультиплексора $MUX_{\text{вых}}$. Блок предназначен для хранения и модификации сигналов состояния МПС, таких как перенос C , знак числа N , переполнение OVR и признак нуля z .

Четырехразрядные регистры состояния построены на триггерах D -типа, запись информации в которые производится по положительному фронту сигнала T при наличии сигнала разрешения записи \overline{CEN} . Операции, выполняемые над битами регистра состояния RGN при $\overline{CEN} = 0$, приведены в табл. 4.13.

Таблица 4.13

I_5-I_0	Операция	Комментарий
001000	$0 \rightarrow NZ$	Уст. в «0» признака нуля z
001001	$1 \rightarrow NZ$	Уст. в «1» признака нуля z
001010	$0 \rightarrow NC$	Уст. в «0» признака C
001011	$1 \rightarrow NC$	Уст. в «1» признака C
001100	$0 \rightarrow NN$	Уст. в «0» признака N
001101	$1 \rightarrow NN$	Уст. в «1» признака N
001110	$0 \rightarrow NV$	Уст. в «0» признака OVR
001111	$1 \rightarrow NV$	Уст. в «1» признака OVR

При $\overline{CEN} = 1$ запись в регистр RGN запрещена.

БИС K1804BP2 допускает возможность выполнения четырех операций со словами регистра состояния RGN при $\overline{CEN} = 0$. Перечень соответствующих преобразований указан в табл. 4.14.

Таблица 4.14

I_5-I_0	Операция	Комментарий
000000	$M_i \rightarrow N_i$	Запись содержимого RGM в RGN
000001	$1 \rightarrow N_i$	Уст. в «1» всех разрядов RGN
000010	$M_i \leftrightarrow N_i$	Регистровый обмен
000011	$0 \rightarrow N_i$	Уст. в «0» всех разрядов RGN

Операции загрузки регистра состояния RGN ($\overline{CEN} = 0$) представляют собой запись информации в указанный регистр с выходов признаков состояний процессора, а соответственно и входов СУСС IC, IN, IV, Iz . Исчерпывающие сведения о данной операции приведены в табл. 4.15.

Таблица 4.15

I_5-I_0	Операция	Комментарий
000110 и 000111	$IZ \rightarrow NZ, IC \rightarrow NC$ $IN \rightarrow NN, IV \vee NV \rightarrow NV$	Запись сигналов с шины I в соответствующий разряд RGN
01100X, 10100X 11100X	$IZ \rightarrow NZ, \overline{IC} \rightarrow NC$ $IN \rightarrow NN, IV \rightarrow NV$	Загрузка с инверсией признака переноса
00010X, 010XXX 01101X, 0111XX 100XXX, 10101X 1011XX, 110XXX 11101X, 1111XX	$IZ \rightarrow NZ$ $IC \rightarrow NC$ $IN \rightarrow NN$ $IV \rightarrow NV$	Запись признаков с шины I

Основные функции регистра состояния RGM заключаются в следующем. Особенностью регистра является возможность управления записью в отдельные его разряды. Так при единице на входе \overline{CEM} запись запрещена во все разряды регистра. Если $\overline{CEM} = 0$, то запись в разряды RGM регламентируется сигналами $\overline{EZ}, \overline{EC}, \overline{EN}, \overline{EV}$. Основные сведения об операциях над словом в регистре RGM приведены в табл. 4.16.

Таблица 4.16

I_5-I_0	Операция	Комментарий
000000	$Y_i \rightarrow M_i$	Запись информации с шины Y в RGM
000001	$1 \rightarrow M_i$	Уст. в «1» всех разрядов RGM
000010	$M_i \leftrightarrow N_i$	Регистровый обмен
000011	$0 \rightarrow M_i$	Уст. в «0» всех разрядов RGM
000101	$\overline{M_i} \rightarrow M_i$	Инвертирование разрядов RGM

Как и регистр RGN , регистр RGM может выполнять операции над битами, а также операции загрузки регистра со входов IC, IN, IV, Iz . Операции над битами RGM выполняются с помощью сигналов разрешения признаков $\overline{EZ}, \overline{EC}, \overline{EN}, \overline{EV}$. Так при нуле на входе \overline{CEM} и нуле на одном из входов $\overline{EZ}, \overline{EC}, \overline{EN}$

или \overline{EV} производится запись признака в соответствующий разряд RGM . При единичном значении на выбранных входах $\overline{EZ}, \overline{EC}, \overline{EN}, \overline{EV}$ запись в соответствующий разряд запрещается. При единичном значении \overline{CEM} запись в разряды всего регистра запрещена.

Операции над битами регистра RGM ($\overline{CEM} = \overline{EZ} = \overline{EC} = \overline{EN} = \overline{EV} = 0$) перечислены в табл. 4.17.

Таблица 4.17

I_5-I_0	Операция	Комментарий
000100	$IZ \rightarrow MZ, MY \rightarrow MC$ $IN \rightarrow MN, MC \rightarrow MV$	Используется при организации сдвиговых операций
00100X, 01100X 10100X, 11100X	$IZ \rightarrow MZ, \overline{IC} \rightarrow MC$ $IN \rightarrow MN, IV \rightarrow MV$	Загрузка с инверсией признака переноса.
00011X, 00101X 0011XX, 010XXX 01101X, 0111XX 100XXX, 10101X 101XXX, 110XXX 11101X, 1111XX	$IZ \rightarrow MZ$ $IC \rightarrow MC$ $IN \rightarrow MN$ $IV \rightarrow MV$	Запись признаков с шины I

Управление выводом информации через трехстабильную шину Y осуществляется в соответствии с табл. 4.18.

Таблица 4.18

\overline{OEY}	I_5	I_4	Y	Комментарий
1	X	X	R_{off}	Состояние высокого сопротивления
0	0	X	$N_i \rightarrow Y_i$	При нуле на входах I_5-I_0 шина Y является входной независимо от значения сигнала \overline{OEY} .
0	1	0	$M_i \rightarrow Y_i$	$I_i = \{Z, C, N, V\}$
0	1	1	$I_i \rightarrow Y_i$	

Блок проверки условия состоит из схемы проверки условия и мультиплексора со схемой управления полярностью. Блок предназначен для формирования выходного сигнала кода условия под действием поля микрокоманды I_3-I_0 . Схема проверки условия выполняет 16 преобразований сигналов признаков, а результат выбранной операции передается на выход кода условия CT . Выбор операндов для выполнения операций в блоке осуществляется под действием сигналов микрокоманды I_5, I_4 .

Управление выходом CT при $\overline{OECT} = 0$ осуществляется в соответствии с табл. 4.19.

Блок выполняет ряд функций по обработке признаков, формируемых АЛУ K1804BC1, BC2 в результате выполнения операций вычитания регистров $RGA-RGB$ (табл. 4.20). При этом анализируются такие функции, как сравнение $RGA \geq RGB$, $RGA = RGB$, $RGA < RGB$ и др. Числа в RGA и RGB

Таблица 4.19

I_3-I_0	$I_5, I_4 = 00$	$I_5, I_4 = 01$	$I_5, I_4 = 10$	$I_5, I_4 = 11$
0000	$(NN \oplus NV) \vee NZ$	$(NN \oplus NV) \vee NZ$	$(MN \oplus MV) \vee MZ$	$(IN \oplus IV) \vee IZ$
0001	$\overline{(NN \oplus NV)} \wedge \overline{NZ}$	$\overline{(NN \oplus NV)} \wedge \overline{NZ}$	$\overline{(MN \oplus MV)} \wedge \overline{MZ}$	$\overline{(IN \oplus IV)} \wedge \overline{IZ}$
0010	$NN \oplus NV$	$NN \oplus NV$	$MN \oplus MV$	$IN \oplus IV$
0011	$\overline{NN \oplus NV}$	$\overline{NN \oplus NV}$	$\overline{MN \oplus MV}$	$\overline{IN \oplus IV}$
0100	NZ	NZ	MZ	IZ
0101	\overline{NZ}	\overline{NZ}	\overline{MZ}	\overline{IZ}
0110	NV	NV	MV	IV
0111	\overline{NV}	\overline{NV}	\overline{MV}	\overline{IV}
1000	$NC \vee NZ$	$NC \vee NZ$	$MC \vee MZ$	$IC \vee IZ$
1001	$\overline{NC} \wedge \overline{NZ}$	$\overline{NC} \wedge \overline{NZ}$	$\overline{MC} \wedge \overline{MZ}$	$\overline{IC} \wedge \overline{IZ}$
1010	NC	NC	MC	IC
1011	\overline{NC}	\overline{NC}	\overline{MC}	\overline{IC}
1100	$\overline{NC} \vee \overline{NZ}$	$\overline{NC} \vee \overline{NZ}$	$\overline{MC} \vee \overline{MZ}$	$\overline{IC} \vee \overline{IZ}$
1101	$NC \wedge \overline{NZ}$	$NC \wedge \overline{NZ}$	$MC \wedge \overline{MZ}$	$IC \wedge \overline{IZ}$
1110	$IN \oplus MN$	NN	MN	IN
1111	$\overline{IN \oplus MN}$	\overline{NN}	\overline{MN}	\overline{IN}

могут быть интерпретированы как целые без знака, так и как операнды в дополнительном коде. Результат одной из выбранных операций с помощью мультиплексора передается на схему управления полярностью, которая при необходимости инвертирует признак. Далее сформированное значение поступает на трехстабильную шину CT , управляемую сигналом разрешения кода условия \overline{OECT} . При $\overline{OECT} = 0$ разрешается вывод кода условия через шину CT . Если $\overline{OECT} = 1$, то вывод кода условия запрещен, а шина CT находится в состоянии высокого сопротивления. Проверка соотношения чисел A и B после выполнения операции $(A-B)$ осуществляется в соответствии с табл. 4.20.

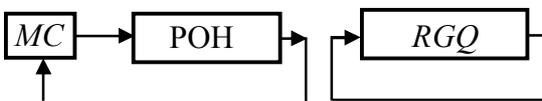
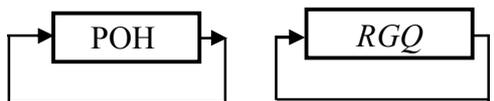
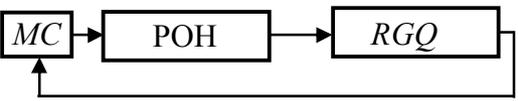
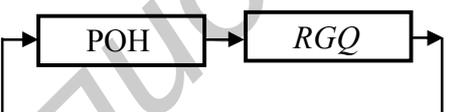
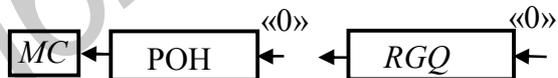
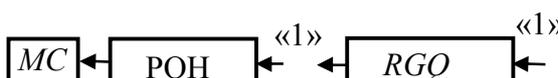
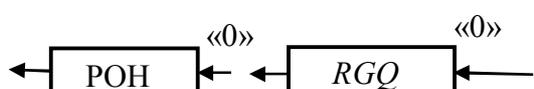
Таблица 4.20

Соотношение	Числа без знака			Числа в дополнительном коде		
	Состояние RGM, RGN , шина I	I_3-I_0	I_3-I_0	Состояние RGM, RGN , шина I	I_3-I_0	I_3-I_0
		$CT = 1$	$CT = 0$		$CT = 1$	$CT = 0$
$A = B$	$z = 1$	0100	0101	$z = 1$	0100	0101
$A \neq B$	$z = 0$	0101	0100	$z = 0$	0101	0100
$A \geq B$	$C = 1$	1010	1011	$\overline{N \oplus V} = 1$	0011	0010
$A < B$	$C = 0$	1011	1010	$N \oplus V = 1$	0010	0011
$A > B$	$C \cdot \overline{Z} = 1$	1101	1100	$\overline{(N \oplus V)} \cdot Z = 1$	0001	0000
$A \leq B$	$C \vee Z = 1$	1100	1101	$\overline{(N \oplus V)} \vee Z = 1$	0000	0001

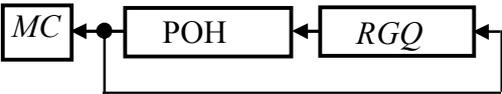
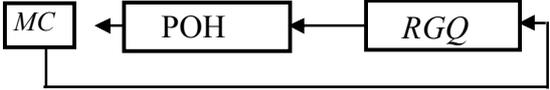
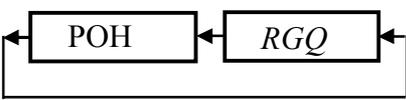
Блок управления сдвигами предназначен для организации различных вариантов арифметических, логических и циклических сдвигов (всего 32 варианта) в зависимости от значений сигналов микрокоманды $I_{10} - I_6$, как показано в табл. 4.21. Сигнал I_{10} определяет направление сдвига, поэтому он должен быть соединен со входом I_8 МПС 1804BC2 или со входом I_7 МПС 1804BC1. Выводы сдвига PF_0, PF_3, PQ_0, PQ_3 являются трехстабильными и управляются сигналом \overline{SE} . При $\overline{SE} = 0$ сдвиг разрешен. При $\overline{SE} = 1$ сдвиг запрещен, а все выходы сдвига находятся в состоянии высокого сопротивления.

Таблица 4.21

$I_{10} - I_6$	Тип сдвига	SL_1	SR_1	SL_2	SR_2	MC
00000	«0» → POH → «0» → RGQ →	R_{off}	0	R_{off}	0	—
00001	«1» → POH → «1» → RGQ →	R_{off}	1	R_{off}	1	—
00010	«0» → POH → RGQ → MC ← POH → MN	F_0	0	R_{off}	MN	F_0
00011	«1» → POH → RGQ →	F_0	1	R_{off}	SL_1	—
00100	MC → POH → RGQ →	F_0	MC	R_{off}	SL_1	—
00101	MN → POH → RGQ →	F_0	MN	R_{off}	SL_1	—
00110	«0» → POH → RGQ →	F_0	0	R_{off}	SL_1	—
00111	«0» → MC → POH → RGQ → RGQ → MC	F_0	0	Q_0	SL_1	Q_0
01000	MC → POH → RGQ → MC ← POH → MC	F_0	SL_1	Q_0	SL_2	F_0

$I_{10} - I_6$	Тип сдвига	SL_1	SR_1	SL_2	SR_2	MC
01001		F_0	MC	Q_0	SL_2	F_0
01010		F_0	SL_1	Q_0	SL_2	—
01011		F_0	IC	R_{off}	SL_1	—
01100		F_0	MC	Q_0	SL_1	Q_0
01101		F_0	SL_2	Q_0	SL_1	Q_0
01110		F_0	$IN \oplus IV$	R_{off}	SL_1	—
01111		F_0	SL_2	Q_0	SL_1	—
10000		0	R_{off}	0	R_{off}	F_3
10001		1	R_{off}	1	R_{off}	F_3
10010		0	R_{off}	0	R_{off}	—

$I_{10} - I_6$	Тип сдвига	SL_1	SR_1	SL_2	SR_2	MC
10011		1	R_{off}	1	R_{off}	—
10100		SR_2	R_{off}	0	Q_3	F_3
10101		SR_2	R_{off}	1	Q_3	F_3
10110		SR_2	R_{off}	0	Q_3	—
10111		SR_2	R_{off}	1	Q_3	—
11000		SR_1	F_3	SR_2	Q_3	F_3
11001		MC	F_3	SR_2	Q_3	F_3
11010		SR_1	F_3	SR_2	Q_3	—
11011		MC	R_{off}	0	R_{off}	—
11100		SR_2	F_3	MC	Q_3	F_3

$I_{10} - I_6$	Тип сдвига	SL_1	SR_1	SL_2	SR_2	MC
11101		SR_2	F_3	SR_1	Q_3	F_3
11110		SR_2	R_{off3}	MC	Q_3	—
11111		SR_2	F_3	SR_1	Q_3	—

Блок управления переносом формирует сигнал входного переноса C_0 для БОД и СУП под действием сигналов микрокоманды $I_{12}, I_{11}, I_5, I_3 - I_1$, как показано в табл. 4.22. При этом в качестве входного переноса выбирается один из семи источников ("0", "1", C_X , NC , MC , \overline{NC} , \overline{MC}), что позволяет легко реализовать операции сложения и вычитания чисел обычной и двойной длины. Вход C_X служит для организации выполнения в МПС1804BC2 некоторых специальных функций, при которых необходимо соединение входа C_X с выходом z МПС.

Таблица 4.22

I_{12}	I_{11}	I_5	I_3	I_2	I_1	C_0
0	0	X	X	X	X	0
0	1	X	X	X	X	1
1	0	X	X	X	X	C_X
1	1	0	0	X	X	NC
1	1	0	X	1	X	NC
1	1	0	X	X	1	NC
1	1	0	1	0	0	\overline{NC}
1	1	1	0	X	X	MC
1	1	1	X	1	X	MC
1	1	1	X	X	1	MC
1	1	1	1	0	0	\overline{MC}

Блок управления под действием сигналов микрокоманды $I_{12} - I_0$ формирует внутренние сигналы, управляющие подсхемами БИС. Для выполнения переходов в микропрограмме выходы признаков старшей МПС z, F_3, OVR, C_4 соединяются с соответствующими входами признаков состояния СУСС Iz, IN, IV, IC . Кроме того, при использовании ИМС К1804ВС2 вывод z МПС соединяется дополнительно со входом C_x К1804ВР2.

Выходы сдвига старшей МПС PF_3, PQ_3 К1804ВС2 или PR_3, PQ_3 К1804ВС1 соединяются с выводами сдвига SR_1, SR_2 СУСС соответственно. А выходы сдвига младшей МПС PF_0, PQ_0 К1804ВС2 или PR_0, PQ_0 К1804ВС1 соединяются с выводами сдвига SL_1, SL_2 .

Выход C_0 схемы К1804ВР2 является входом C_0 для МПС и схем ускоренного переноса при их каскадном соединении в многоразрядном процессоре.

Выход кода условия СУСС (CT) при использовании в блоке управления ПЗУ ветвления необходимо соединить с младшим разрядом адреса, старшие разряды которого формируются в регистре микрокоманды. При использовании в управляющем устройстве БИС К1804ВУ4 указанный разряд СУСС соединяется с выводом \overline{CS} . Однако при разработке и усложнении системы ввода-вывода спецкомпьютера схемотехника внутреннего интерфейса, в частности входа \overline{CS} , может быть усложняться.

Микросхема К1804ВР2 обеспечивает выполнение операции нормализации чисел обычной и двойной длины как в МПС К1804ВС1, так и в МПС ВС2. Признаком окончания нормализации служит появление сигнала на выходе C_4 старшей МПС, который формируется операцией «Исключающее ИЛИ» двух старших разрядов. Вторым признаком, информирующим об окончании нормализации (за один такт до ее окончания), служит сигнал на выходе OVR , представляющий собой функцию «Исключающее ИЛИ» двух средних разрядов старшей МПС. Эти сигналы (C_4 или OVR) через блок проверки условия СУСС подаются на вход кода условия \overline{CS} БИС К1804ВУ4. Так как признаки записываются в регистр состояния с опозданием на один такт, то при использовании сигнала на выходе C_4 МПС в качестве признака окончания нормализации необходимо в качестве последнего шага нормализации выполнить сдвиг операнда в сторону младших разрядов. Этот сдвиг выполняется СУСС специальной операцией сдвига при подаче на входы $I_{10} - I_6$ кода 00101 (нормализации чисел двойной длины) или 00010 (нормализация чисел обычной длины). В МПС К1804ВС1 нет индикатора окончания операции нормализации. Поэтому в СУСС предусмотрена операция $MN \oplus IN$ со знаковыми разрядами операнда на предыдущем и текущем тактах соответственно.

В процессе микропрограммирования выхода CT следует учитывать, что при использовании БИС К1804ВУ4 в блоке управления соответствующий вход признака является инверсным. Поэтому сформированный признак CT также должен формироваться с учетом операции инверсии.

ЗАКЛЮЧЕНИЕ

Рассмотренные вопросы проектирования спецкомпьютера с неймановской базовой архитектурой включают в себя: принципы синтеза ядра системы (аппаратуры процессор – память), организацию БМУ на основе микропрограммного управления, разработку сокращенного интерфейса для взаимодействия основных блоков компьютера. Использованный метод формирования вычислительного устройства на базе примитивов различного уровня позволил интегрировать подсистемы и подсистемы машины в компактные модули, взаимодействие которых было организовано на основе статического соединения внешних выводов. При этом ограниченная верификация проекта выполнялась с использованием микропрограммного обеспечения, позволяющего проследить направление информационных потоков в любой точке системы.

Основное внимание в проекте уделено аппаратным средствам и принципам компьютерного синтеза, которые с течением времени не так быстро девальвируются, несмотря на бурное развитие соответствующей отрасли. В частности использование систем ускоренного переноса, принципов обработки сложных ветвлений, а также многообразие алгоритмов сдвиговых операций всегда положительно воспринимается разработчиками аппаратуры и ПО. Применение же микропрограммного управления позволяет сформировать эффективную систему команд, исключая невостребованные пересылки и расчеты в процессоре специального назначения. Однако ограниченный объем данного пособия не позволил рассмотреть ряд функций спецкомпьютера, знание которых является обязательным для специалиста.

Положительной особенностью используемой в процессе проектирования элементной базы является аппаратная «прозрачность» микропроцессорного комплекта. Это позволяет проследить циркуляцию информации как в рамках системного интерфейса, так и в локальных интерфейсах БИС. Указанное обстоятельство позволяет изучить реальные схемотехнические вопросы синтеза устройств не только с точки зрения их функционального назначения, но и с позиции совершенствования имеющихся технических реализаций.

Во второй части работы планируется детальное изучение методов проектирования системы ввода-вывода, организации прямого доступа в память, использования БИС контроллера ОЗУ и рассмотрение системных вопросов взаимодействия подсистем компьютера в различных режимах работы. Кроме того, перспективным представляется процесс проектирования ядра системы в рамках использования БИС K1804BC2 с управлением от K1804BY4. Данный подход к синтезу спецсистемы позволит подробно изучить специальные функции, обрести навыки разработки сложных интерфейсов, а также микропрограммирования системы команд.

Интегральные знания о принципах работы неймановского компьютера на основе стандартных блоков: процессора, памяти, устройства управления и системы ввода-вывода, несомненно, являются «таблицей умножения» любого современного инженера-системотехника.

ЛИТЕРАТУРА

1. Матов, В. И. Бортовые цифровые вычислительные машины и системы / В. И. Матов, Ю. А. Белоусов, Е. П. Федосеев; под ред. В. И. Матова. – М.: Высш. шк., 1988. – 216 с.
2. Булдаков, А. В. Основы анализа и проектирования специализированных вычислительных средств / А. В. Булдаков. – Министерство Обороны СССР, 1970. – 314 с.
3. Специализированные ЦВМ / В. Б. Смолов [и др.]; под ред. В. Б. Смолова. – М.: Высш. шк., 1981. – 279 с.
4. Клингман Э. Проектирование специализированных микропроцессорных систем / Э. Клингман; пер. с англ. – М.: Мир, 1985. – 363 с.
5. Хвощ, С. Т. Микропроцессоры и микроЭВМ в системах автоматического управления : справочник / С. Т. Хвощ, Н. Н. Варлинский, Е. А. Попов; под общ. ред. С. Т. Хвоща. – Л. : Машиностроение, 1987. – 640 с.
6. Проектирование цифровых систем на комплектах микропрограммируемых БИС / С. С. Булгаков [и др.]; под ред. В. Г. Колесникова. – М. : Радио и связь, 1984. – 240 с.
7. Мик, Дж. Проектирование микропроцессорных устройств с разрядно-модульной организацией связей / Дж. Мик, Дж. Брик; пер. с англ. – М. : Мир, 1984. – 289 с.
8. Калабеков, Б. А. Микропроцессоры и их применение в системах передачи и обработки сигналов / Б. А. Калабеков. – М. : Радио и связь, 1988. – 368 с.
9. Кобяк, И. П. Элементы теории и техника множително-делительных операций в псевдотейронных системах / И. П. Кобяк // АВТ. – 2008. – №1. – С. 16–28.
10. Хамахер, К. Организация ЭВМ. 5-е изд. / К. Хамахер, З. Вранешич, С. Заки. – СПб.: Питер; Киев: Издательская группа ВНУ, 2003. – 848 с.

Учебное издание

Кобяк Игорь Петрович

***СПЕЦКОМПЬЮТЕР С НЕЙМАНОВСКОЙ
БАЗОВОЙ АРХИТЕКТУРОЙ***

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Г. С. Корбут*

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал макет *В. М. Задоя*

Подписано в печать 26.06.2013. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 6,39. Уч.-изд. л. 6,6. Тираж 100 экз. Заказ 64.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6