Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: СиФОВМ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

СПЕЦИАЛИЗИРОВАННЫЙ КОМПЬЮТЕР

БГУИР КП 1-40 02 01 406 ПЗ

Студент: группы 150504,   
Горбачевский К. В.

Руководитель:   
Кобяк И. П.

МИНСК 2024

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                            (подпись)

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**З А Д А Н И Е**

**по курсовому проектированию**

Студенту Горбачевскому Кириллу Витальевичу\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

1. Тема проекта Спецкомпьютер

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Срок сдачи студентом законченного проекта с 22.02.2024 по 09.08.2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к проекту:

1. Разрядность – 32.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Кодирование – обратный код.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Способы адрессации – прямая, непосредственная, автоинкремент, относительная.              \_\_\_\_\_\_

4.

5.

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.Разработка архитектуры специализированного компьютера.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Проектирование основных структурных компонентов схемы компьютера.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.Проектирование внутреннего интерфейса.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Разработка микропрограмного обеспечения.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заключение.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Литература.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перечень графического материала (с точным указанием обязательных чертежей)

1. Структурная схема спецкомпьютера (формат А2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Принципиальная схема спецкомпьютера (формат А2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультант по проекту (с назначением разделов проекта)\_И.П. кобяк     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Дата выдачи задания 22.08.2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Календарный график работы над проектом на весь период проектирования (с назначением сроков исполнения и трудоемкости отдельных этапов):

разделы 1,2 к 24.09  –  50 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел    3    к 15.10  –  25 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

раздел    4    к 05.11  –  25 %;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оформление пояснительной записки и графического материала к 01.03 – 15 %;\_\_\_\_\_\_

защита курсового проекта с 08.08 по 12.08.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РУКОВОДИТЕЛЬ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_доцент каф. ЭВМ  Кобяк И.П.

(подпись)

Задание принял к исполнению 22.02.2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т. В. Грек

(дата и подпись студента)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc162185373)

[ИСХОДНЫЕ ДАННЫЕ 7](#_Toc162185374)

[1 ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЬЮТЕРА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ 8](#_Toc162185375)

[1.1 Анализ известных реализаций спецкомпьютеров, формулирование требований к разрабатываемому компьютеру 8](#_Toc162185376)

[1.2 Проектирование алгоритмов, выбор состава макроопераций и программирование задач 9](#_Toc162185377)

[1.3 Проектирование системы команд 13](#_Toc162185378)

[1.4. Кодирование системы команд 17](#_Toc162185379)

[1.5. Переход от системы команд к блок-схеме компьютера 19](#_Toc162185380)

[1.4 Проектирование ЗУ микрокомпьютера 22](#_Toc162185381)

[2 ПРОЕКТИРОВАНЕ ОСНОВНЫХ СТРУКТУРНЫХ КОМПОНЕНТОВ СХЕМЫ МИКРОКОМПЬЮТЕРА 27](#_Toc162185382)

[2.1 Разработка схемы блока обработки данных 27](#_Toc162185383)

[3 ПРОЕКТИРОВАНИЕ БЛОКА МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ (БМУ) 33](#_Toc162185384)

**ВВЕДЕНИЕ**

Проектирование компьютеров специального назначения является одной из важнейших задач, стоящих перед разработчиками цифровой аппаратуры. Основное назначение спецкомпьютеров – это решение определенного класса задач 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 решений при этом во многом зависит от качества бортовых компьютеров (и программного обеспечения), входящих в состав управляемых объектов.

Назначение бортовых компьютеров как правило состоит в следующем.

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 компьютера платформы происходит не автономно, а в тесном взаимодействии с системами наземных стационарных или подвижных комплексов. Поэтому сложность разработки структуры и программного обеспечения спецсистем обычно достаточно велика и требует существенных материальных затрат.

Стационарные комплексы управления, в свою очередь, оказывают существенное влияние на архитектуру бортовых систем. Так в состав средств наземного управления могут входить следующие подсистемы:

1) центральный компьютер локальной сети;

2) средства контроля и диагностики подсистем;

3) подсистема навигации и радиопеленгации;

4) процессор для расчета координат целей;

5) подсистемы информационного и служебного программного обеспечения.

Таким образом, 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.

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

Во-вторых. Это линии и схемы передачи данных, осуществляющие связь

рассредоточенных систем и исполнительных механизмов носителя с бортовым компьютером, а также средства для надежной передачи управляющей информации в соответствии с принятым протоколом обмена.

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

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

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

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, на использовании экспертных систем и баз знаний, на анализе и модернизации базовых компьютерных моделей.

**ИСХОДНЫЕ ДАННЫЕ**

* Разрядность данных 32 бита.
* Представление данных в ЭВМ: обратный код.
* Способы адресации операндов: прямая, непосредственная, автоинкрементная, относительная.
* Емкость ОЗУ 800 кБайт.
* Тип интегральной схемы ОЗУ: 565РУ7.
* Микропроцессорный БИС блока обработки данных: 1804ВС1.
* БИС ПЗУ микрокоманд: 556РТ14.
* БИС блока микропрограммного управления: 1804ВУ2.
* Порты ввода-вывода: 1804ИР3 8/8.
* Система прерываний (уровень/источник): 2/8-11.
* Канал ПДП (длина блока передаваемых данных): 512 слов.
  + - * Программное обеспечение: арифметическая операция *ex*.

**1** **ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЬЮТЕРА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ**

**1.1 Анализ известных реализаций спецкомпьютеров, формулирование требований к разрабатываемому компьютеру**

В период развития и становления вычислительной техники появилось много моделей представления компьютера, но наиболее распространённой является трёхшинная модель. Её структуру можно представить рис. 1.1:



Рис. 1.1. Структура трёхшинной модели

Опишем назначение блоков:

***Устройство управления (УУ)*.** Предназначено для управления всеми блоками компьютера путем посылки сигналов предписывающие те или иные действия. *УУ* используется для автоматической работы компьютера и указывает на:

* функцию выполнения *АЛУ*
* источники информации для *АЛУ*
* приемники результатов полученных в ходе вычислений.

***Арифметико-логическое устройство (АЛУ)*.** Предназначено для выполнения арифметических и логических операций, представленных в формате с плавающей запятой или фиксированной запятой. Кроме данных *АЛУ* может обрабатывать адресную информацию (формирование исполнительного адреса), команды (преобразование форматов), признаки (выход переноса, признак нулевого результата, переполнение, знаковый разряд и т.д.).

***Оперативная память (ОП)*.** Предназначена для хранения информации поступающей в компьютер из вне. Этот блок также предназначены для хранения программ, результатов промежуточных расчетов и другой машинной информации. *ОП* состоит из ячеек, в каждой из которых хранится машинное слово. Основными характеристиками *ОП* является емкость памяти и время обращения. Под временем обращения понимается длительность цикла записи или чтения операнда из любой ячейки ЗУ.

***Устройства ввода/вывода (УВВ)*.** Предназначены для связи компьютера с внешними периферийными устройствами. Устройство ввода обеспечивает считывание информации с внешних носителей и представление ее в форме электрических сигналов. Устройство вывода преобразует кодовую информацию, поступающую из памяти или других блоков машины, в форму, необходимую для обмена с внешней средой.

Кроме основных вышеперечисленных блоков в состав компьютера может входить ***система прямого доступа к памяти (ПДП)***и ***система прерываний*.**

Система *ПДП* позволяет осуществить непосредственный обмен данными между памятью и периферийными устройствами под управлением контроллера *ПДП* без участия *АЛУ*, что позволяет повышать скорость выполнения обмена.

***Система прерываний*** предназначена для прерывания программы пользователя, если возникло прерывание от внешних устройств, либо внутреннее прерывание. Например, при работе с внешними устройствами необходима работа с *УВВ*, обладающие небольшим быстродействием. Поэтому для синхронизации их с компьютером используется контроллер прерываний, выдающий соответствующие сигналы в *УУ*, когда *УВВ* готово к работе. При этом выполнение текущей операции спецкомпьютера приостанавливается и запускается программа обработки информации от *УВВ*. После завершения данной программы, выполнение прерванной операция возобновляется.

В рамках данного курсового проекта, при разработке спецкомпьютера предполагается использование вышеприведённой модели. В качестве базовых микросхем используются микросхемы серии К1804.

Проектируемый спецкомпьютер должен использовать разрядность данных равную 32 бит. Данные будут представляться в форме с плавающей запятой в обратном коде. Также спецкомпьютер будет содержать систему памяти для хранения информации, поддерживать ввод/вывод данных, использовать систему прерываний и систему прямого доступа к памяти.

**1.2 Проектирование алгоритмов, выбор состава макроопераций и программирование задач**

В соответствии с заданием необходимо разработать программное обеспечение для выполнения арифметических операций: *e‌x*.

Для решения поставленной задачи на первом этапе следует построить график заданной функции, определить ее максимальное и минимальное значения, а также диапазон изменения аргумента. График функции изображен на рис. 1.2. Диапазон изменения аргумента в пределах (-∞;+∞). Значение функции изменяется в пределах (0;2). Для расчетов конкретных значений функции используем разложение *e‌* в степенной ряд Тейлора:

(1.1)

Из данного равенства видно, что одному значению функции соответствует бесконечное число членов ряда. Выбор же числа членов для расчета функции при заданном *x* определяется:

1. точностью представления результатов в компьютере,
2. допустимой длительностью расчета управляющей информации.

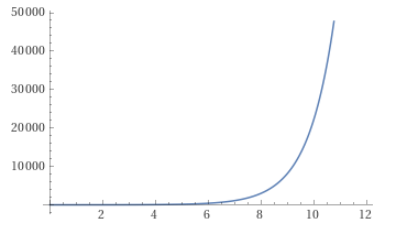


Рис. 1.2. График функции e‌x

Заданная разрядность нашего числа – 32 (см. рис. 1.3).

Длина поля порядка , длина поля мантиссы – 25 разрядов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 31 | *30* | 6 | 5 | 4 | 0 |
| ± | *М -* мантисса | | ± | *Р – порядок* | |
| Зн. *М* | 25 разрядов | | Зн. *P* | 5 разряда | |

Рис. 1.3

Общая ошибка вычислений в компьютере обуславливается:

1.  следует из ограничений на число членов в разложении в ряд Тейлора,
2.  определяется ограничениями разрядной сетки.

Исследуем закон изменения общего члена ряда при различных значениях аргумента *x* (*1 ≤ x ≤ 8*). Поставленной задаче будут соответствовать величины, представленные в табл. 1.1.

Таблица 1.1

x

i

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | 0,5 | 2 | 4,5 | 8 | 12,5 | 18 | 24,5 | 32 |
| 3 | 0,2 | 1,3 | 4,5 | 10,7 | 20,8 | 36 | 57,2 | 85,3 |
| 4 | 0,04 | 0,7 | 3,4 | 10,7 | 26,04 | 54 | 100,04 | 170,7 |
| 5 | 0,008 | 0,3 | 2,02 | 8,5 | 26,04 | 64,8 | 140,06 | 273,07 |

Продолжение таблицы 1.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 0,001 | 0,09 | 1,01 | 5,7 | 21,7 | 64,8 | 163,4 | 364,09 |
| 7 | 0,0002 | 0,03 | 0,4 | 3,3 | 15,5 | 55,5 | 163,4 | 416,1 |
| 8 | 0 | 0,006 | 0,2 | 1,6 | 9,7 | 41,7 | 143 | 416,1 |
| 9 | 0 | 0,001 | 0,05 | 0,7 | 5,4 | 27,8 | 111,2 | 369,9 |
| 10 | 0 | 0,0003 | 0,02 | 0,3 | 2,7 | 16,7 | 77,8 | 295,9 |
| 11 | 0 | 0,0001 | 0,004 | 0,1 | 1,2 | 9,09 | 49,5 | 215,2 |
| 12 | 0 | 0 | 0,001 | 0,04 | 0,5 | 4,5 | 28,9 | 143,5 |
| 13 | 0 | 0 | 0,0003 | 0,01 | 0,2 | 2,1 | 15,6 | 88,3 |
| 14 | 0 | 0 | 0,0001 | 0,003 | 0,07 | 0,9 | 7,8 | 50,4 |
| 15 | 0 | 0 | 0 | 0,0008 | 0,02 | 0,4 | 3,6 | 26,9 |
| 16 | 0 | 0 | 0 | 0,0002 | 0,007 | 0,1 | 1,6 | 13,5 |

Как видно из табл. 1.1, некоторые члены ряда с порядковыми номерами 8 – 16 близки к нулю и практически не влияют на результирующее значение функции. Причём, чем больше значение аргумента, тем меньшее число членов ряда влияют на результат.

Найдем значения целевой функции *e‌* в данных условиях (см. табл. 1.2).

Таблица 1.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x  sum | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1,000 | 2,000 | 3,000 | 4,000 | 5,000 | 6,000 | 7,000 | 8,000 |
| 2 | 1,500 | 4,000 | 7,500 | 12,000 | 17,500 | 24,000 | 31,500 | 40,000 |
| 3 | 1,667 | 5,333 | 12,000 | 22,667 | 38,333 | 60,000 | 88,667 | 125,333 |
| 4 | 1,708 | 6,000 | 15,375 | 33,333 | 64,375 | 114,000 | 188,708 | 296,000 |
| 5 | 1,717 | 6,267 | 17,400 | 41,867 | 90,417 | 178,800 | 328,767 | 569,067 |
| 6 | 1,718 | 6,356 | 18,412 | 47,556 | 112,118 | 243,600 | 492,168 | 933,156 |
| 7 | 1,718 | 6,381 | 18,846 | 50,806 | 127,619 | 299,143 | 655,569 | 1349,257 |
| 8 | 1,718 | 6,387 | 19,009 | 52,432 | 137,307 | 340,800 | 798,546 | 1765,359 |
| 9 | 1,718 | 6,389 | 19,063 | 53,154 | 142,689 | 368,571 | 909,749 | 2135,227 |
| 10 | 1,718 | 6,389 | 19,080 | 53,443 | 145,381 | 385,234 | 987,592 | 2431,121 |
| 11 | 1,718 | 6,389 | 19,084 | 53,548 | 146,604 | 394,323 | 1037,128 | 2646,317 |
| 12 | 1,718 | 6,389 | 19,085 | 53,583 | 147,114 | 398,868 | 1066,024 | 2789,781 |
| 13 | 1,718 | 6,389 | 19,085 | 53,594 | 147,310 | 400,965 | 1081,584 | 2878,067 |
| 14 | 1,718 | 6,389 | 19,086 | 53,597 | 147,380 | 401,864 | 1089,363 | 2928,516 |
| 15 | 1,718 | 6,389 | 19,086 | 53,598 | 147,403 | 402,223 | 1092,994 | 2955,422 |
| 16 | 1,718 | 6,389 | 19,086 | 53,598 | 147,410 | 402,358 | 1094,582 | 2968,875 |

Из формата компьютерного слова следует, что максимальное значение функции с учетом разрядности мантиссы *M=18* и порядка *P = 15* определится формулой:

225 – 1 = 33 554 431,

0, 33 554 431⋅ 231 = 720 575 918.904 442 88 (1.2)

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

-1 ⋅ 2-25 ⋅ 2 -31 = -1 ⋅ 2-56  (1.3)

Введем ограничения на расчетный параметр *e‌*, полагая, что допустимая погрешность вычислений равна . Тогда из формулы (1.1) для аргумента *{x}* можно записать соотношение:

|  |  |
| --- | --- |
|  | (1.4) |

Итак, из соотношений (1.6) следует, что заданный уровень погрешности вычислений при *x=1* не превышается, если расчет функции осуществляется на основании 2-х членов ряда.

На основании табл. 1.2 находим погрешности для остальных значений аргумента *x*:

|  |  |  |
| --- | --- | --- |
| x=1 i=4 ∆ = 0,00366  x=2 i=6 ∆ = 0,00453  x=3 i=8 ∆ = 0,00380  x=4 i=10 ∆ = 0,00284 | x=5 i=12 ∆ = 0,00202  x=6 i=13 ∆ = 0,00363  x=7 i=15 ∆ = 0,00241  x=8 i=16 ∆ = 0,00372 | (1.5) |

Таким образом, ввод данных должен сопровождаться обращением к служебной таблице вида (1.5) и определением числа вычислительных циклов, необходимых для расчета функции *e‌* для конкретного *x*. В простейшем случае число членов ряда выбирается равным 8.

Погрешность, вносимая ограниченностью разрядной сетки (с учетом бесконечной длины значения функции), будет определяться разностью между эталонным значением *e‌*, в нашем случае равным *ln(0.9)*, и значением, размещаемым в 18-разрядной сетке компьютера:

|  |  |
| --- | --- |
|  | (1.6) |

Из (1.6) следует, что , то есть  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.

**1.3 Проектирование системы команд**

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

Рассмотрим блок-схему (рис. 1.4) алгоритма расчета функции *e‌* в соответствии с формулой (1.1) и исследуем методику решения поставленной задачи.



Рис. 1.4. Блок-схема модернизированного алгоритма расчета функции e‌x

В полученной выше граф-схеме удалось избежать долговременных повторных вычислений факториала и степени и заменить их на множительно-делительные преобразования свойственные практически всем высокоуровневым языкам компьютера. Общая таблица команд для вычисления *e‌* при этом принимает следующий вид:

Таблица 1.4

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | № | Команда |
| 1 | *Move a1,a2* | 5 | *Div a1,a2* |
| 2 | *Sum a1,a2* | 6 | *Inr a1* |
| 3 | *Sub a1,a2* | 7 | *Jz metka* |
| 4 | *Mult a1,a2* | 8 | *Jmp metka* |

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

1. прямая адресация;
2. непосредственная адресация;
3. автоинкрементная адресация;
4. относительная адресация.

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

Команды пересылки приведены на рис. 1.6.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I1 | КОП | mod | Ri | Прямой адрес | | Память – регистр Ri | |
|  |  |  |  |  | |  | |
| I2 | КОП | mod | Ri | Константа | | | Константа – регистр Ri |
|  |  |  |  |  |  |  | |
| I3 | КОП | mod | Ri | Rj | Память (автоинкр. через Rj) – регистр Ri в *Move* не используется | | |
|  |  |  |  |  |  |  | |
| I4 | КОП | mod | Ri | Rj | Смещение | | Память (через Rj и смещение) – регистр Ri | |

Рис. 1.5

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; *Ri* – регистр источник и приемник пересылаемого операнда; *Rj*и *Rx* – регистры косвенной адресации. Команды *I*1–*I3* представляют собой команды формата *RS*, команда *I4* – команду формата *RX*.

Команда сложения имеет следующие форматы (рис. 1.6):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I5 | КОП | mod | Ri | Прямой адрес | | Память – регистр Ri | |
|  |  |  |  |  | |  | |
| I6 | КОП | mod | Ri | Константа | | | Сл 1 в Rj, Сл 2 задается явно |
|  |  |  |  |  |  |  | |
| I7 | КОП | mod | Ri | Rj | Память (автоинкр. через Rj) – регистр Ri в *Move* не используется | | |
|  |  |  |  |  |  |  | |
| I8 | КОП | mod | Ri | Rj | Смещение | | Сл 1 в Ri, Сл 2 в памяти через Rj и смещение | |

Рис. 1.6

Для составления подмножеств команд умножения *Mult*, вычитания *Sub* и деления *Div* может быть использована аналогичная методология что приводит к образованию управляющих слов с номерами: *I*9–*I20*.

Команды перехода *Jz* и *Jmp* (*I0*), применяемые при вычислении функции *y* *e‌*, очевидно, являются одноадресными. В связи с этим нет необходимости включения в состав команд данной группы поля первого операнда *Ri*, а все множество управляющих слов в данной группе сократить до трех и представить тремя форматами (рис. 1.7).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I0 | КОП | mod | Прямой адрес | | | «Метка» – прямой адрес | | |
| I21 |
|  |  |  |  | | |  | | |
| I22 | КОП | mod | Ri | Константа | | | | Условный переход по адресу задается явно | |
|  |  |  |  |  |  | |  | | |
| I23 | КОП | mod | Ri | Rj | Смещение | | | Условный переход по адресу, хранящемуся в памяти (через регистр базы и смещение) | | |

Рис. 1.7

Команда *Inr*представляется одним одноадресным форматом с явным указанием регистра (рис. 1.8).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | I24 | КОП | mod | Ri | операнд в регистре *Ri* |

Рис. 1.8

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;
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 подсистемами 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.

## **1.4. Кодирование системы команд**

При кодировании системы команд в рассматриваемом примере (рис. 1.6) 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. 1.9).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | КОП |  | Ri | ≤ 21 бит | |  |
| I1 | xxxx | 000 | xxxx | Прямой адрес | | Память – регистр Ri |
|  |  |  |  | Разрядность поля опред. емкостью памяти | | |
| I2 | xxxx | 001 | xxxx | Константа | | Константа – регистр Ri |
|  |  |  |  | Rj |  |  |
| I3 | xxxx | 010 | xxxx | xxxx | Память (автоинкр. через Rj) – регистр Ri в *Move* не используется | |
|  |  |  |  |  | ≤ 17 бит |  |
| I4 | xxxx | 011 | xxxx | xxxx | Смещение | Память (через Rj и смещение) – регистр Ri | |

Рис. 1.9

Будем считать, что длина команды не может превышать разрядности машинного слова. Следовательно, разрядность прямого адреса в первом формате может составлять не более 32–11 = 21 бит. Аналогично, разрядность смещения в четвертом формате не может превышать 32–15= 17 бит.

Порядок кодирования управляющей информации в командах компьютера *I0*–*I33* … приведен в табл. 1.5.

Таблица 1.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № команды | КОП | Код | Тип | mod | Примечания |
| *I0* | *Jmp* | 0000 | RS | 000 | Безусловный переход |
| *I1-I4* | *Move* | 0001 | RS  RI  RS  RX | 000  001  010  011 | Команды пересылки |
| *I5-I8* | *Sum* | 0010 | RS  RI  RS  RX | 000  001  010  011 | Команды сложения |
| *I9-I12* | *Sub* | 0011 | RS  RI  RS  RX | 000  001  010  011 | Команды вычитания |
| *I13-I16* | *Mult* | 0100 | RS  RI  RS  RX | 000  001  010  011 | Команды умножения |

Продолжение таблицы 1.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *I17-I20* | *Div* | 0101 | RS  RI  RS  RX | 000  001  010  011 | Команды деления |
| *I21-I23* | *Jz metka* | 0110 | RS  RI  RS | 000  001  010 | Команды условных переходов |
| *I24* | *Inr* | 0111 | RR | 000 | Инкремент |
| *I25-I26*  *I27-I28* | *In*  *Out* | 1000  1001 | RS  RI  RS  RS | 000  001  000  001 | Команды ввода-вывода (операнд в аккумуляторе, адресация порта прямая или через РЗУ) |
| *I29-…* | … | 1010 | … | … | Спецкоманды |

## **1.5. 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. 1.10.

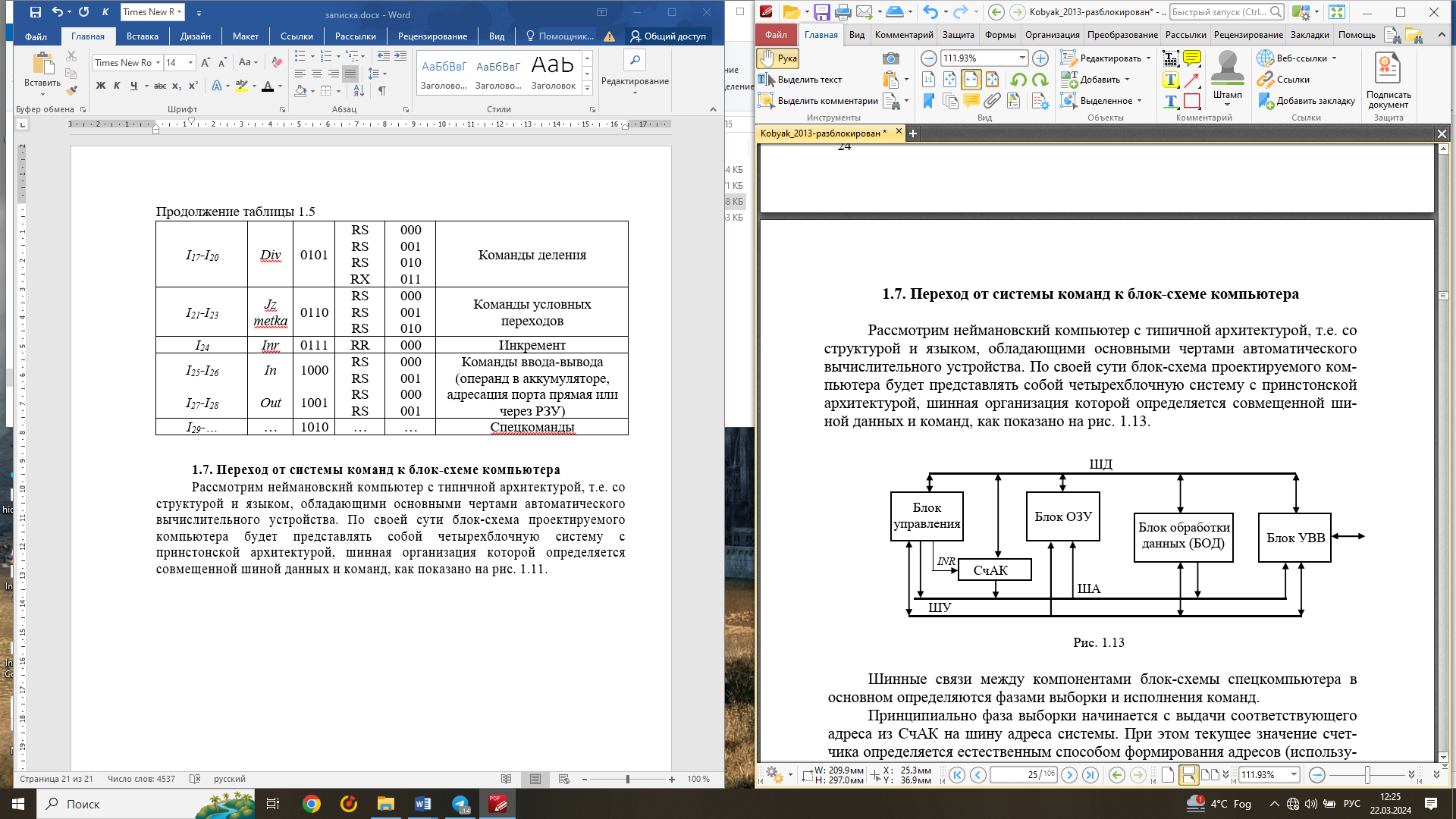


Рис. 1.10

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 отработки 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 (*RG K*) в виде отдельной структурной компоненты и разместим в нем команду первого формата, например, *I*1. При этом, устанавливая связи между *RG K*, блоками и модулями системы, добиваемся логического объединения воедино формата команды, аппаратных средств и принципа функционирования неймановской архитектуры (рис. 1.11). 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 *А*, разрядность которой должна быть равна разрядности поля *Rj*. Иными словами, считаем, что процессор имеет 16 РОН, а поле адреса – длину 4 бита.

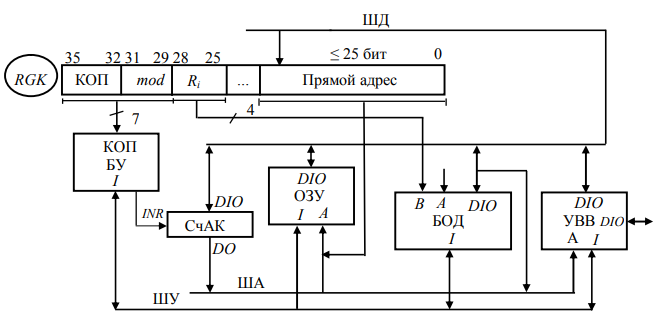


Рис. 1.11

В рамках совершенствования внутреннего интерфейса следует учитывать, что адресный вход памяти может принимать информацию от двух источников: от регистра команды *RG K* и из счетчика СчАК. Это предполагает включение в соответствующей точке интерфейса мультиплексора *MUX* c 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*3 – *I*4 предполагает введение в сформированную блок–схему (см. рис. 1.11) дополнительных линий связи, как показано на рис. 1.12.

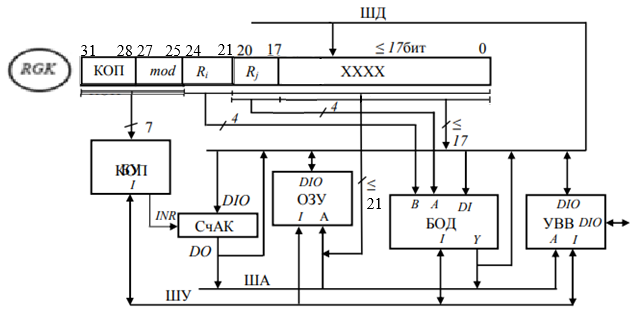


Рис. 1.12

На данном шаге эмуляции необходимо учитывать, что формирование адреса ОЗУ в четвертом формате осуществляется с использованием двух регистров процессора: *Rj –* регистра базы и *Rx –* регистра индекса. Данный факт предполагает одновременность передачи адресов регистров из *RGK* на входы РЗУ *А* и *В* 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. 1.13, может быть выполнен после расчета и синтеза функциональных блоков и соответствующих линий связи внутреннего интерфейса компьютера.

**1.4 Проектирование ЗУ микрокомпьютера**

Исходя из задания к курсовому проекту, локальная память разрабатываемой микроЭВМ должна быть, построена на основе БИС ЗУ К565РУ7.

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 и столбца , линию управления записью чтением . Модуль реализует функции записи, хранения, чтения и регенерации информации в соответствии с табл. 1.6.

В структурную схему ИМС памяти входят выполненные на одном кристалле: матрица накопителя с 262144 элементами памяти, расположенными на пересечениях 512 строк и столбцов, 512 усилителей считывания и регенерации, дешифраторы строк и столбцов, устройство управления и два регистра адреса.

Таблица 1.6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *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 | Регенерация |
| 0 | 0 | 0 | *A* | 0 | z | Запись 0 |
| 0 | 0 | 0 | *A* | 1 | z | Запись 1 |
| 0 | 0 | 1 | *A* | X | *D* | Считывание |

Минимальная емкость памяти, используемая компьютером с 32-разрядной шиной команд\данных, определяется общей емкостью 32 ИМС заданного модуля ОЗУ. В нашем случае эта величина составляет ровно 256 Кбит × 32 = 8192 Кбит или 1024 Кбайт. Соответствующая организация такой подсистемы получила название банка памяти и показана на рис. 1.13. Обозначение банка, применяемое на структурных схемах, приведено на рис. 1.14.

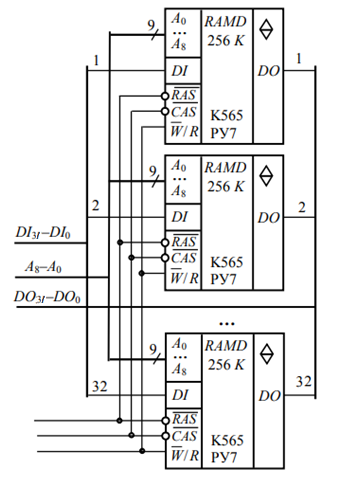


Рис. 1.13. Банк памяти

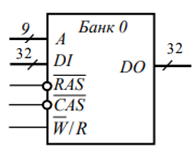


Рис. 1.14. Обозначение банка

Общее количество банков, составляющих подсистему ОЗУ (в соответствии с заданной емкостью памяти 800 Кбайт), рассчитывается путем деления заданной емкости ЗУ на емкость 1 банка: 800/1024 = 0,78 ≈ 1 банк.

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. 1.11.

Для стробирования соответствующего банка памяти по входу *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 *CAS* осуществляется с использованием *MUX* *СAS*. 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.

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 *Sw* ШУ, подаваемого на вход при переводе ОЗУ в режим регенерации. Реализация подсистемы памяти (рис. 1.9) в виде модуля с соответствующими шинами управления, а также механизм включения ПЗУ в адресное пространство ОЗУ, показаны на рис. 1.10.

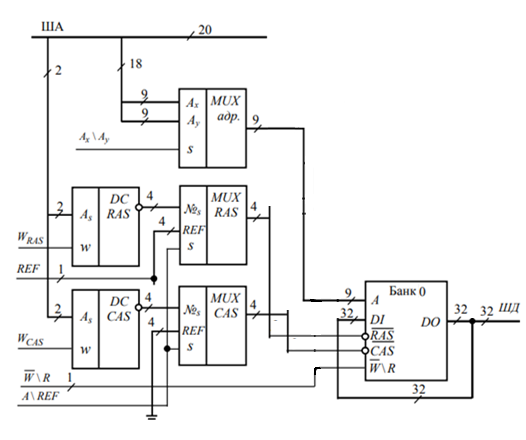


Рис. 1.9. Структурная схема блока ОЗУ

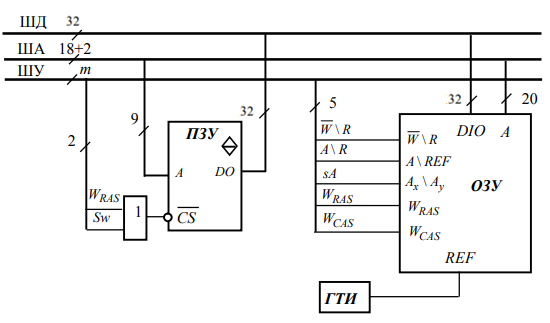


Рис. 1.10. Реализация подсистемы памяти

Управляющая информация из табл. 1.11 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 *Sw* определяется как *RGMk*.

Таблица 1.11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *A\R* | *sA* | *WRAS* | *WCAS* | 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 команд, ориентированных на обращение к ОЗУ (рис. 1.11).

31 28 27 25 24 21 19 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| КОП | *000* | *Ri* | **|||||||||||||||** | Прямой адрес |

31 28 27 25 24 21 20 17 8 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| КОП | *011* | *Ri* | *Rj* | **|||||||||||||||||||||||||||||** | Смещение |

Рис. 1.11. Уточнённый формат некоторых команд

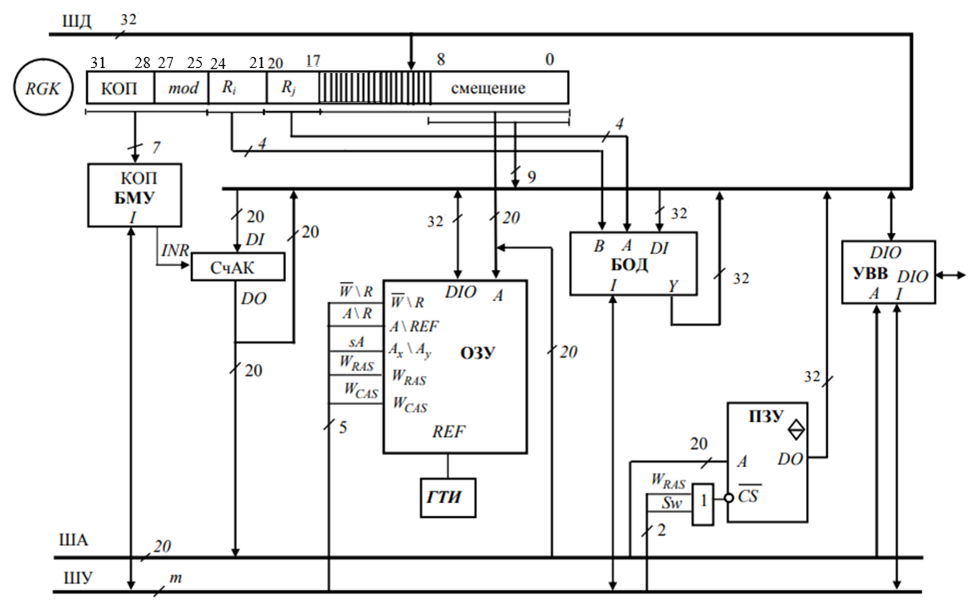
****

Рис. 1.12 – Блок схема компьютера с учетом ОЗУ

**2 ПРОЕКТИРОВАНИЕ ОСНОВНЫХ СТРУКТУРНЫХ КОМПОНЕНТОВ СХЕМЫ МИКРОКОМПЬЮТЕРА**

**2.1 Разработка схемы блока обработки данных**

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.

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.

Структурная схема БОД состоит из следующих основных блоков:

* процессорного блока К1804ВС1;
* схемы ускоренного переноса (СУП) К1804ВР1;
* схемы управления состоянием и сдвигами (СУСС) К1804ВР2.

Четырехразрядная микропроцессорная секция К1804ВС1 предназначена для построения операционных блоков цифровых устройств с разрядностью, кратной 4. На структурной схеме МПС условно выделяют четыре крупных блока (рис. 2.1):

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

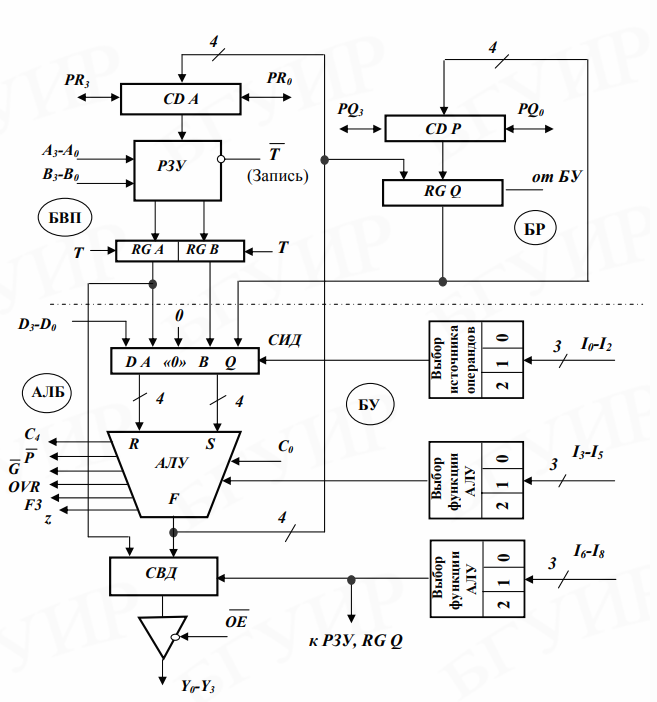


Рис. 2.1. Структурная схема К1804ВС1

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 I8 – I0. При этом входные линии DI образуют 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 OE.

Второй особенностью блока Prim.1 является наличие общего для младшей МПС БОД и схемы СУП входа переноса C0. Аппаратно данный вход соединяется с выходом C0 СУСС К1804ВР2. Таким образом, при микропрограммировании различных функций состояние указанной линии определяется путем задания управляющей информации по входам I12 - I0 модуля СУСС.

Для организации ускоренного переноса выводы , младшей (правой крайней) МПС соединяются со входами , СУП ВР1. Соответствующий выход Cx­ подключается ко входу C0 первой средней МПС. Аналогично выводы , первой средней МПС соединяются со входами , ИМС ВР1, а соответствующий выход Cy подключается ко входу C0 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 , , при этом выходами всего 16-разрядного модуля являются выходы схемы К1804ВР1 , .

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 *F3, OVR, C4* 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-разряд-ный блок в виде модуля со входом переноса C0 и выводами каскадирования , . 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.

Для реализации команд формата RR на программном и микропрограммном уровне в проектируемую схему БОД включены мультиплексоры адресов А и В РЗУ (рис. 2.4). При этом мультиплексор канала A имеет три информационных входа, что необходимо для выполнения команд с базовоиндексной адресацией. Управление коммутаторами адресов организуется с использованием сигналов s1, s0, формируемых регистром микрокоманды (рис. 2.4). Табл. 2.1 определяет принцип мультиплексирования шин в соответствии с указанными сигналами управления.

Таблица 2.1

|  |  |  |
| --- | --- | --- |
| *s1s0* | *y MUX A* | *y MUX B* |
| *00* | *x1* | *x1* |
| *01* | *x2* | *x2* |
| *10* | *x1* | *x1* |
| *11* | *x2* | *x3* |

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 К1804ВС1 по каналам РЗУ А и В RGMk [45,38];

3) сигнал разрешения выдачи данных из микропроцессорной секции на Y -шину OE1 RGMk[37];

4) I8 - I0 – сигналы управления функцией МПС RGMk[36,28];

5) I12 - I0 – сигналы управления функцией СУСС RGMk[27,15];

6) сигналы RGMk[14,6] управления записью информации в регистры К1804ВР2 и выдачи признаков на шину *YN, YV, YC, YZ*; при этом если указанные двунаправленные выводы не используются, то разряд RGMk[13] = 1.

Состояние остальных разрядов данной группы выходов определяется справочными таблицами.

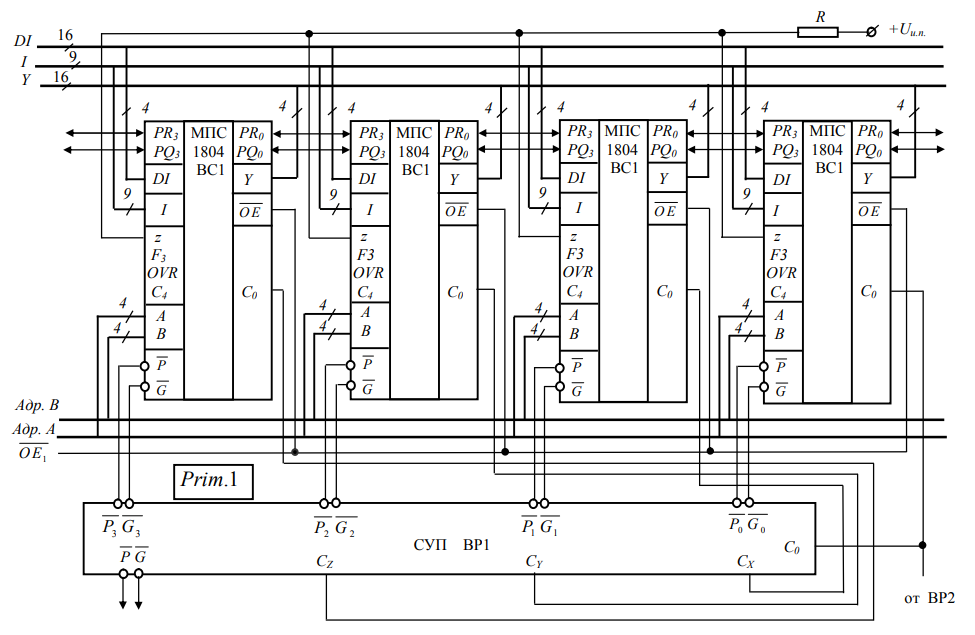


Рис. 2.2. Схема Prim 1.

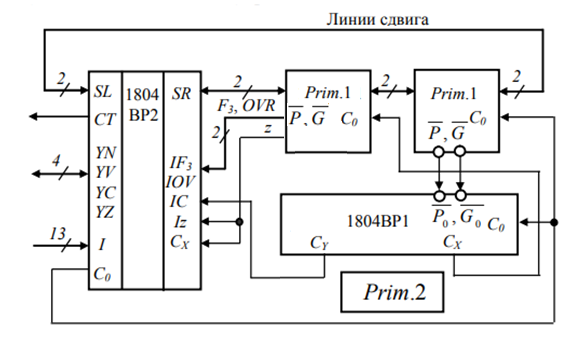


Рис. 2.3. Схема блока обработки данных (Prim 2).

С целью сокращения внешних связей БОД или внутреннего интерфейса компьютера будем считать, что СчАК располагается в одном из регистров (РЗУ) процессора (RG15) и управляется микропрограммно из RGMk.

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

Для реализации команд формата RR на программном и микропрограммном уровне в проектируемую схему БОД включены мультиплексоры адресов А и В РЗУ (рис. 2.4). Управление коммутаторами адресов организуется с использованием сигнала *s0*, формируемого регистром микрокоманды.

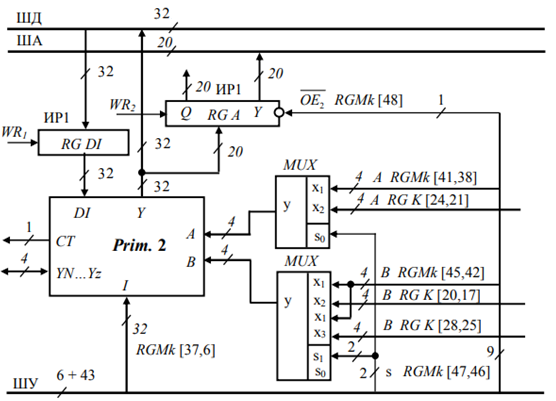


Рис. 2.4. Включение блока обработки данных в схему компьютера

Для включения БОД на базе МПС К1804ВС2 в состав компьютера преобразуем устройство, показанное на рис. 2.4, 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. 2.5.

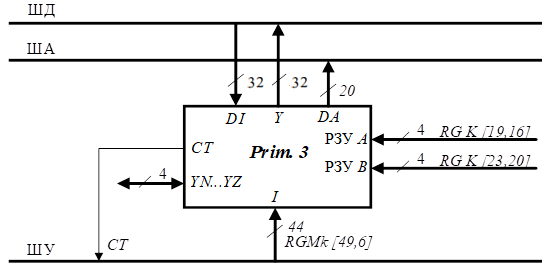


Рис. 2.5. Включение блока обработки данных в схему компьютера

## 

## **3 ПРОЕКТИРОВАНИЕ БЛОКА МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ (БМУ)**

#### 3.1 Разработка устройства управления

Устройство управления специализированного микрокомпьютера обеспечивает выполнение последовательности микроопераций в соответствии с кодом текущей команды и организует выборку команд программы в соответствии с выполняемой программой.

Устройство микропрограммного управления содержит: блок микропрограммной памяти, в котором хранятся микрокоманды; блок генерации адреса микрокоманды, формирующий адрес следующей микрокоманды, который в общем случае зависит от кода выполняемой микрооперации, кодов и признаков выполняемых в АЛУ операций, информации блоков синхронизации и прерывания процессора; блок синхронизации, предназначенный для приема управляющих сигналов и формирования последовательности синхросигналов для основных блоков специализированного микрокомпьютера для обеспечения определенной последовательности их работы; дешифратор микрокоманд, формирующий управляющие сигналы, поступающие в исполнительные блоки специализированного микрокомпьютера.

Согласно варианту задания данный блок должен быть реализован на основе схем управления адресом микрокоманды (СУАМ) К1804ВУ2.

В формировании адресов на выходе СУАМ участвуют следующие входные сигналы шины управления: , , , , и . Так, при подаче «1» на вход происходит последовательное увеличение адресов на единицу, а при подаче «1» на *i*-й вход маски адрес может быть изменен на величину . Сигналы на входах , коммутируют на выход один из четырех источников адреса, а сигналы и 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 таких схем позволяет обращается к памяти объемом в 4К слов. При наращивании модулей объединяются линии шины управления , , , , , , всех схем, а также выполняется соединение выходов переноса предыдущей СУАМ с входами последующей. Сигнал переноса используется в арифметической функции инкрементирования, поэтому при выборе параметров тактовых импульсов следует учитывать время прохождения переноса через все модули. Реализация каскада секций К1804ВУ2 представлена на рис. 2.4.

Для упрощения проектируемой системы входы всех секций соединяются с шиной «земля», что позволяет исключить маскирование адресов, используемое в многопроцессорных системах. Вход управления выходной шиной также подключается к нулю, что фиксирует адреса СУАМ на входах ПЗУ микропрограммной памяти (МПП). Сигнал подключен к кнопке «сброс» и используется для перевода компьютера в состояние первоначальной загрузки при включении.

Информация с входов по сигналу может быть записана во внутренний регистр и одновременно подана на вход мультиплексора адреса секций К1804ВУ2. Наличие сигнала расширяет формат микрокоманды еще на один разряд.



Рис. 2.4. Модуль Prim4

Для организации условных и безусловных переходов на микропрограммном уровне в состав блока микропрограммного управления включается ПЗУ 556РТ14 (согласно варианту задания). Основное назначение данного модуля состоит в управлении источниками адреса микропрограммной памяти в СУАМ. В соответствии с поставленной целью в ПЗУ выбираются две соседние ячейки: четная и нечетная, адреса которых конкретизируются младшей адресной линией . Например, и , где x – это произвольное значение бита. Как правило, в качестве адресной информации, подаваемой на вход , используется признак , поступающий из блока обработки данных. Если в процессе вычислений признак оказался равным «1», то из ПЗУ читается нечетное слово. При этом различное информационное наполнение указанных двух ячеек позволяет переводить модуль СУАМ из одного режима формирования адресов в другой, иными словами, в системе реализуется условный переход. Если же в четной и нечетной ячейках ПЗУ хранятся одинаковые слова, то в системе выполняется безусловный переход на адрес, соответствующий считанному из ПЗУ управляющему слову.

Таким образом, включив схемы ПЗУ с модулем Prim4, получим обобщенную схему устройства управления, представленную на рис. 2.5.

Для управления режимом переходов используются выходы ПЗУ . Выходы , , используются для арбитража шины адреса ветвления и должны содержать унитарную информацию. Поле адреса перехода находится в разрядах 65..54 регистра микрокоманды, где реализовано управление соответствующим подрегистром по схеме с тремя состояниями. Подключение данного поля к шине адреса ветвления осуществляется выходом .

Системная функция блока ПЗУ ПНА (преобразователь начального адреса) заключается в аппаратной трансляции полей КОП и Mod в начальный адрес микропрограммы, соответствующей выполняемой команде. Сформированный начальный адрес передается в модули К1804ВУ2 по шине данных на внутренний мультиплексор и далее на выходы , соединенные со входами микропрограммного ПЗУ.

Блок ПЗУ ПА – это преобразователь адреса, который функционирует аналогично ПНА. Однако его входная шина используется для приема вектора запросов на прерывание, поступающих от портов ввода-вывода К1804ИР3. 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.



Рис. 2.5. Схема устройства управления

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

Кнопка «Сброс» внешним монтажом подключена ко входу *zA*, обеспечивающему при подаче нулевого уровня переход к нулевой ячейке МПП. При этом в компьютере отрабатывается процедура тестирования внутренних схем и интерфейса, после чего выполняется переход к выполнению микропрограммы «Загрузчик».

Интегрирование блока микропрограммного управления в единый модуль компьютера приводит к интеграции в блок, представленный на рис. 2.6.



Рис. 2.6. Схема модуля устройства управления

При реализации сброса в автоматическом режиме следует помнить, что время действия сигнала *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.