**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.

**1 РАЗРАБОТКА АРХИТЕКТУРЫ СПЕЦИАЛИЗИРОВАННОГО МИКРОКОМПЬЮТЕРА**

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

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:



Рисунок 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.

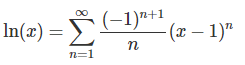
В рамках данного курсового проекта, при разработке спецкомпьютера предполагается использование вышеприведённой модели. В качестве базовых микросхем используются микросхемы серии К1804.

Проектируемый спецкомпьютер должен использовать разрядность данных равную 28 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: *ln* xi , x=*sum* ai,разработать программу контроля функционирования ОЗУ по алгоритму теста «Галоп».

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 [0;+∞). Значение функции изменяется в пределах (-∞;+∞). Для расчетов конкретных значений функции используем разложение *ln(x*) 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:

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

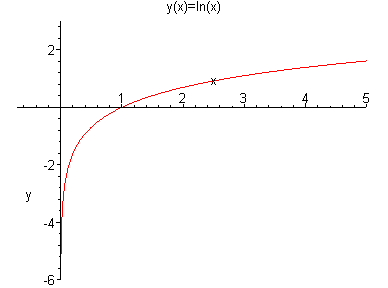


Рисунок 1.2 – График функции *y = ln(x)*

Для решения функции в целом предварительно необходимо выполнить функцию x=*sum* ai которая сводится к операции суммирования значения *(a1+a2+a3+ ... +ai+ ...)*. Таким образом, по логике наших рассуждений инициатором начала расчета должен быть появляющийся в ОЗУ массив данных *{ai}.*

Заданная разрядность нашего числа – 32 (см. рис. 1.3). Будем считать, что числа в памяти представлены в прямом коде с плавающей запятой.

Длина поля порядка , длина поля мантиссы – 27 разрядов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 31 |  | 6 |  | 4 | 0 |
| ± | Мантисса 25 разряда | | ± | Порядок 5 разрядов | |

Рисунок 1.3 – Формат компьютерного слова

Общая ошибка вычислений в компьютере обуславливается:

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

Исследуем закон изменения общего члена ряда при различных значениях аргумента *x* (*1.2 ≤ x ≤ 1.9*). Поставленной задаче будут соответствовать величины, представленные в табл. 1.1.

Таблица 1.1 – Значения общего члена ряда функции *ln(x)* при различных значениях аргумента x.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i  x  i | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |
| 1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
| 2 | -0,02 | -0,045 | -0,08 | -0,125 | -0,18 | -0,245 | -0,32 | -0,405 |
| 3 | 0,002667 | 0,009 | 0,021333 | 0,041667 | 0,072 | 0,114333 | 0,170667 | 0,243 |
| 4 | -0,0004 | -0,00203 | -0,0064 | -0,01563 | -0,0324 | -0,06003 | -0,1024 | -0,16403 |
| 5 | 6,4E-05 | 0,000486 | 0,002048 | 0,00625 | 0,015552 | 0,033614 | 0,065536 | 0,118098 |
| 6 | -1,1E-05 | -0,00012 | -0,00068 | -0,0026 | -0,00778 | -0,01961 | -0,04369 | -0,08857 |
| 7 | 1,83E-06 | 3,12E-05 | 0,000234 | 0,001116 | 0,003999 | 0,011765 | 0,029959 | 0,068328 |
| 8 | -3,2E-07 | -8,2E-06 | -8,2E-05 | -0,00049 | -0,0021 | -0,00721 | -0,02097 | -0,05381 |
| 9 | 5,69E-08 | 2,19E-06 | 2,91E-05 | 0,000217 | 0,00112 | 0,004484 | 0,014913 | 0,043047 |
| 10 | -1E-08 | -5,9E-07 | -1E-05 | -9,8E-05 | -0,0006 | -0,00282 | -0,01074 | -0,03487 |
| 11 | 1,86E-09 | 1,61E-07 | 3,81E-06 | 4,44E-05 | 0,00033 | 0,001798 | 0,007809 | 0,028528 |
| 12 | -3,4E-10 | -4,4E-08 | -1,4E-06 | -2E-05 | -0,00018 | -0,00115 | -0,00573 | -0,02354 |
| 13 | 6,3E-11 | 1,23E-08 | 5,16E-07 | 9,39E-06 | 0,0001 | 0,000745 | 0,004229 | 0,019553 |
| 14 | -1,2E-11 | -3,4E-09 | -1,9E-07 | -4,4E-06 | -5,6E-05 | -0,00048 | -0,00314 | -0,01634 |
| 15 | 2,18E-12 | 9,57E-10 | 7,16E-08 | 2,03E-06 | 3,13E-05 | 0,000317 | 0,002346 | 0,013726 |
| 16 | -4,1E-13 | -2,7E-10 | -2,7E-08 | -9,5E-07 | -1,8E-05 | -0,00021 | -0,00176 | -0,01158 |

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

График влияния *i*-го члена ряда на результат представлен на рис. 1.4.

Рисунок 1.4 – График влияния *i*-го члена ряда на результат

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

Таблица 1.2 – Суммирование членов ряда в соответствии с ростом номеров i

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x  sum | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |
| 1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
| 2 | 0,18 | 0,255 | 0,32 | 0,375 | 0,42 | 0,455 | 0,48 | 0,495 |
| 3 | 0,182667 | 0,264 | 0,341333 | 0,416667 | 0,492 | 0,569333 | 0,650667 | 0,738 |
| 4 | 0,182267 | 0,26197 | 0,334933 | 0,401037 | 0,4596 | 0,509303 | 0,548267 | 0,57397 |
| 5 | 0,182331 | 0,262456 | 0,336981 | 0,407287 | 0,475152 | 0,542917 | 0,613803 | 0,692068 |
| 6 | 0,18232 | 0,262336 | 0,336301 | 0,404687 | 0,467372 | 0,523307 | 0,570113 | 0,603498 |
| 7 | 0,182322 | 0,262367 | 0,336535 | 0,405803 | 0,471371 | 0,535072 | 0,600072 | 0,671826 |
| 8 | 0,182322 | 0,262359 | 0,336453 | 0,405313 | 0,469271 | 0,527862 | 0,579102 | 0,618016 |
| 9 | 0,182322 | 0,262361 | 0,336482 | 0,40553 | 0,470391 | 0,532346 | 0,594015 | 0,661063 |
| 10 | 0,182322 | 0,26236 | 0,336472 | 0,405432 | 0,469791 | 0,529526 | 0,583275 | 0,626193 |
| 11 | 0,182322 | 0,26236 | 0,336476 | 0,405476 | 0,470121 | 0,531324 | 0,591084 | 0,654721 |
| 12 | 0,182322 | 0,26236 | 0,336475 | 0,405456 | 0,469941 | 0,530174 | 0,585354 | 0,631181 |
| 13 | 0,182322 | 0,26236 | 0,336476 | 0,405465 | 0,470041 | 0,530919 | 0,589583 | 0,650734 |
| 14 | 0,182322 | 0,26236 | 0,336476 | 0,405461 | 0,469985 | 0,530439 | 0,586443 | 0,634394 |
| 15 | 0,182322 | 0,26236 | 0,336476 | 0,405463 | 0,470016 | 0,530756 | 0,588789 | 0,64812 |
| 16 | 0,182322 | 0,26236 | 0,336476 | 0,405462 | 0,469998 | 0,530546 | 0,587029 | 0,63654 |

Из формата компьютерного слова следует, что максимальное значение функции с учетом разрядности мантиссы *M=21* и порядка *P = 31* определится формулой:

221 – 1 = 134 217 727,

0, 134 217 727 ⋅ 231 = 288 230 374,004228 (1.2)

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

-1 ⋅ 2-21 ⋅ 2 -31 = -1 ⋅ 2-52  (1.3)

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

(1.4)

На основании ряда (1.1) при x = 1.2 имеем:

(1.5)

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

(1.6)

Итак, из соотношений (1.6) следует, что заданный уровень погрешности вычислений при *x=1.2* не превышается, если расчет функции осуществляется на основании 3-х членов ряда.

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

**

Таким образом, ввод данных должен сопровождаться обращением к служебной таблице вида (1.7) и определением числа вычислительных циклов, необходимых для расчета функции *ln(x)* для конкретного *x*. В простейшем случае число членов ряда выбирается равным 20.

Погрешность, вносимая ограниченностью разрядной сетки (с учетом бесконечной длины значения функции), будет определяться разностью между эталонным значением *ln(x)*, в нашем случае равным *ln(*20*)*, и значением, размещаемым в 21-разрядной сетке компьютера:

(1.8)

Из (1.8) 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.

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 алгоритма расчета функции *ln(x)* в соответствии с формулой (1.1) рис. 1.5 и исследуем методику решения поставленной задачи.

Основной недостаток алгоритма вычисления *ln(x)* заключается в сложности и длительности формирования результата с использованием команд:

Выполним модернизацию алгоритма, учитывая, что вычисления на каждом -м шаге выполняются по правилу

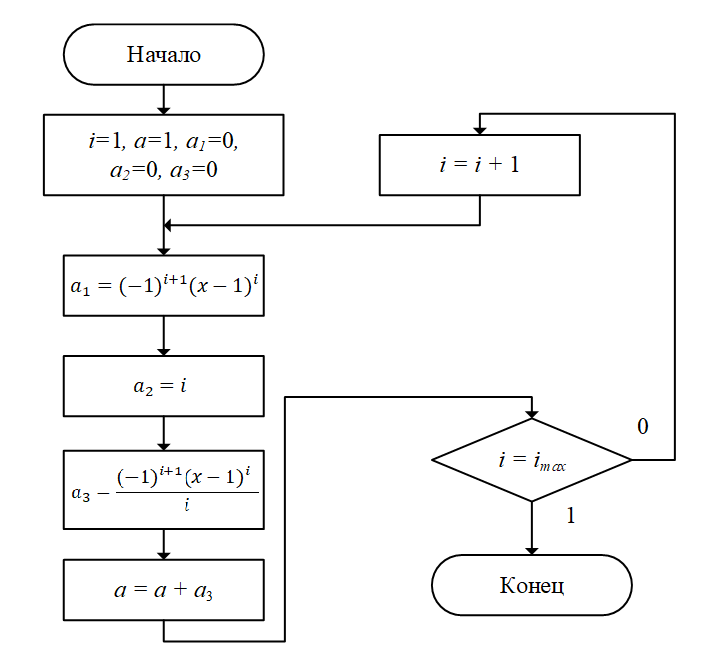
**

Рисунок 1.5 — Блок-схема алгоритма расчета функции ln(x)

Иными словами, в каждом *i*-ом цикле вычислений легко прослеживается зависимость:

(1.9)

С учетом равенства (1.9) модернизируем граф схему алгоритма (рис. 1.5) и поставим в соответствие всем вершинам ГСА команды компьютера (рис. 1.6).

i=1, a=1, a1=x-1,

a2=1, a3=0

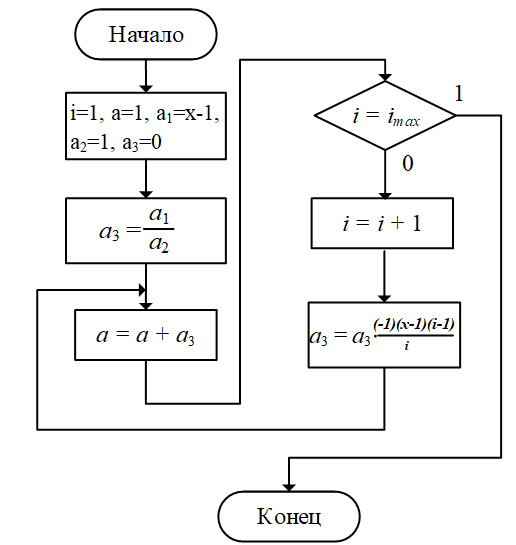


Рисунок 1.6 — Блок-схема модернизированного алгоритма расчета функции ln(x)

В полученной выше граф-схеме удалось избежать долговременных повторных вычислений в командах *Ifac(i) и Degr(x ,i)* и заменить их на множительно-делительные преобразования свойственные практически всем высокоуровневым языкам компьютера. Общая таблица команд для вычисления *ln(x)* при этом принимает следующий вид:

Таблица 1.3 — Таблица команд



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

Базовая система управляющих инструкций процессора без учёта команд служебного и специального ПО будет содержать 8 управляющих слов.

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

- прямая адресация;

- косвенная регистровая адресация;

- автоинкрементная адресация;

- косвенно-относительная адресация.

В рассматриваемых 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 приёмник пересылаемого операнда; *Rj* и *Rx* – 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*.

Рассчитаем разрядность поля кода операции базовой команды (КОП): intlog2(8 команд+2 ввод+1) = 4.

Рассчитаем разрядность поля модификатора команды, определяющий способ использования регистров общего назначения (РОН) при адресации ОЗУ (mod): log2(5 видов адресации) = 3.

При этом полная длина расширенного поля КОП будет иметь 7 разрядов.

При кодировании системы команд в рассматриваемом примере (см. рисунок 2.2) поле управления КОП принципиально может быть кодировано триадами в диапазоне 000-111.

Разрядность полей-указателей регистров выбирается по числу регистров процессорного РЗУ и в данном случае выбирается равной 4.

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 четвёртом формате не может превышать 32 – 19 = 13 бит.

Форматы команд, доработанных с учётом длины полей, могут быть представлены в следующем виде:

Таблица 1.6 – Форматы команд, доработанных с учётом длины полей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | КОП |  | *Ri* | ≤ 17 бит | | |  |
| *I1* | xxxx | 000 | xxxx | прямой адрес | | |  |
|  | КОП |  | *Ri* | *Rj* | Разрядность поля определяется ёмкостью памяти | | |
| *I2* | xxxx | 001 | xxxx | xxxx |  |  |  |
| *I3* | xxxx | 010 | xxxx | xxxx |  |  |  |
|  |  |  |  |  | *Rx* | ≤ 9 бит | |
| *I4* | xxxx | 011 | xxxx | xxxx | xxxx | смещение | |
| *I5* | xxxx | 100 | xxxx | xxxx | Разрядность поля определяется организацией памяти | | |

Команды пересылки приведены в таблице 1.7.

Таблица 1.7 – Форматы команды пересылки

|  |  |
| --- | --- |
| Способ адресации | Команда *Move A1,A2* |
| Прямая (*I1*) | 27 24 23 21 20 17 16   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | Ri | Прямой адрес | | память – регистр *Ri* | | | | |
| Косвенно-регистровая(*I2*) | 27 24 23 21 20 17 16 13 12   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | память (косвенно через *Rj*) – регистр *Ri* | | | | |
| Базово –индексная(*I4*) | 27 24 23 21 20 17 16 13 12 9 8   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | *Rx* | смещение | | память (баз.-инд. через *Ri*,*Rx*) – регистр *Ri* в командах *Move* не используется | | | | | | |
| Автоинкрементная(*I3*) | 27 24 23 21 20 17 16 9   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | память (автоинкр. через *Ri*) – регистр *Ri* в командах *Move* не используется | | | | |
| Регистрово-регистровая(*I5*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | регистр *Rj* – регистр *Ri* | | | | |

Команда деления имеет следующие форматы:

Таблица 1.8 – Форматы команды деления

|  |  |
| --- | --- |
| Способ адресации | Команда *Div A1, A2* |
| Прямая (*I6*) | 27 24 23 21 20 17 16   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | Прямой адрес | | Дм в регистре *Ri* – Дт в памяти | | | | |
| Косвенно-регистровая (*I7*) | 27 24 23 21 20 17 16 13 12   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | Дм в регистре *Ri* – Дт в памяти косвенно через *Rj* | | | | |
| Базово –индексная (*I9*) | 27 24 23 21 20 17 16 13 12 9 8   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | *Rx* | смещение | | Дт в регистре *Ri* – Дм в памяти – баз.–инд. через *Rj*, *Rx* | | | | | | |
| Автоинкрементная (*I8*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | Дт в регистре *Ri* – Дм в памяти автоинкр. через *Rj* | | | | |
| Регистрово-регистровая (*I10*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | KОП | *mod* | *Ri* | *Rj* | | Дм в регистре *Ri* – Дт в регистре *Rj* | | | | |

Для составления подмножеств команд суммирования *Sum*, вычитания *Sub* и умножения *Mult* может быть использованы аналогичная методология, что приводит к образованию управляющих слов с номерами: *I11* – *I25*.

Таблица 1.9 – Форматы команды суммирования

|  |  |
| --- | --- |
| Способ адресации | Команда *Sum A1,A2* |
| Прямая (*I11*) | 27 24 23 21 20 17 16   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | Прямой адрес | |
| Косвенно-регистровая (*I12*) | 27 24 23 21 20 17 16 13 12   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Базово –индексная (*I13*) | 27 24 23 21 20 17 16 13 12 9 8   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | *Rx* | смещение | |
| Автоинкрементная (*I14*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Регистрово-регистровая (*I15*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |

Таблица 1.10 – Форматы команды вычитания

|  |  |
| --- | --- |
| Способ адресации | Команда *Sub A1,A2* |
| Прямая (*I16*) | 27 24 23 21 20 17 16   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | Прямой адрес | |
| Косвенно-регистровая (*I17*) | 27 24 23 21 20 17 16 13 12   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Базово –индексная (*I18*) | 27 24 23 21 20 17 16 13 12 9 8   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | *Rx* | смещение | |
| Автоинкрементная (*I19*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Регистрово-регистровая (*I20*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |

Таблица 1.11 – Форматы команды умножения

|  |  |
| --- | --- |
| Способ адресации | Команда *Mult A1, A2* |
| Прямая (*I21*) | 27 24 23 21 20 17 16   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | Прямой адрес | |
| Косвенно-регистровая (*I22*) | 27 24 23 21 20 17 16 13 12   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Базово –индексная (*I23*) | 27 24 23 21 20 17 16 13 12 9 8   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | *Rx* | смещение | |
| Автоинкрементная (*I24*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |
| Регистрово-регистровая (*I25*) | 27 24 23 21 20 17 16 13   |  |  |  |  | | --- | --- | --- | --- | | КОП | *mod* | *Ri* | *Rj* | |

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

Таблица 1.12 – Форматы команды безусловного перехода

|  |  |
| --- | --- |
| Способ адресации | Команда *Jmp Metka* |
| Прямая (*I0*) | 27 24 23 21 20   |  |  |  | | --- | --- | --- | | КОП | *mod* | Прямой адрес | | «Метка» - прямой адрес | | | |

Таблица 1.13 – Форматы команды условного перехода

|  |  |
| --- | --- |
| Способ адресации | Команда *Jz Metka* |
| Прямая (*I26*) | 27 24 23 21 20 17   |  |  |  | | --- | --- | --- | | КОП | *mod* | Прямой адрес | | «Метка» - прямой адрес | | | |
| Косвенно-регистровая (*I27*) | 27 24 23 21 20 17   |  |  |  | | --- | --- | --- | | КОП | *mod* | *Ri* | | «Метка» памяти – косвенно через *Ri* | | | |
| Регистрово-регистровая (*I28*) | 27 24 23 21 20 17   |  |  |  | | --- | --- | --- | | КОП | *mod* | *Ri* | | «Метка» в регистре *Ri* | | | |

Команда *Inr* представляется одним одноадресным форматом с явным указанием регистра (см. таблицу 1.14).

Таблица 1.14 – Форматы команды икрементации

|  |  |
| --- | --- |
| Способ адресации | Команда *Inr A2* |
| Регистрово-регистровая (*I29*) | 27 24 23 21 20 17   |  |  |  | | --- | --- | --- | | КОП | *mod* | *Ri* | | «Метка» в регистре *Ri* | | | |

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 адресации операндов.

Порядок кодирования управляющей информации в командах компьютера *I0-I33*…приведена в таблице 1.15.

Таблица 1.15 – Порядок кодирования управляющей информации в командах компьютера *I0-I33*…

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | КОП | Код | Тип | mod | Примечание |
| 1 | 2 | 3 | 4 | 5 | 6 |
| *I0* | *Jmp* | 0000 | *RS* | 000 | Безусловный переход |
| *I1* | *Move* | 0001 | *RS* | 000 | Команды пересылки |
| *I2* | 0001 | *RS* | 001 |
| *I3* | - | *-* | - |
| *I4* | *Move* | - | *-* | - | Команды пересылки |
| *I5* | *Move*  *Div* | 0001 | *RR* | 100 | Команды пересылки  Команды деления |
| *I6* | 0010 | *RS* | 000 |
| *I7* | *Div*  *Sum* | 0010 | *RS* | 001 | Команды деления  Команды сложения |
| *I8* | 0010 | *RS* | 010 |
| *I9* | 0010 | *RX* | 011 |
| *I10* | 0010 | *RR* | 100 |
| *I11* | 0011 | *RS* | 000 |
| *I12* | *Sum*  *Sub* | 0011 | *RS* | 001 | Команды сложения  Команды вычитания |
| *I13* | 0011 | *RS* | 010 |
| *I14* | 0011 | *RX* | 011 |
| *I15* | 0011 | *RR* | 100 |
| *I16* | 0100 | *RS* | 000 |
| *I17* | *Sub*  *Mult* | 0100 | *RS* | 001 | Команды вычитания  Команды умножения |
| *I18* | 0100 | *RS* | 010 |
| *I19* | 0100 | *RX* | 011 |
| *I20* | 0100 | *RR* | 100 |
| *I21* | 0101 | *RS* | 000 |
| *I22* | *Mult*  *Jz* | 0101 | *RS* | 001 | Команды умножения  Команды условных переходов |
| *I23* | 0101 | *RS* | 010 |
| *I24* | 0101 | *RX* | 011 |
| *I25* | 0101 | *RR* | 100 |
| *I26* | 0110 | *RS* | 000 |
| *I27* | *Jz*  *Inr* | 0110 | *RS* | 001 | Команды условных переходов  Инкремент |
| *I28* | 0110 | *RS* | 010 |
| *I29* | 0111 | *RR* | 000 |
| *I30* | *In* | 1000 | *RS* | 000 | Команда ввода-вывода (операнд в аккумуляторе, адресация порта прямая или через РЗУ) |
| *I31* | *In*  Out | 1000 | *RS* | 001 | Команда ввода-вывода (операнд в аккумуляторе, адресация порта прямая или через РЗУ)  Спецкоманды |
| *I32* | 1001 | *RS* | 000 |
| *I33* | Out  … | 1001 | *RS* | 001 |
| *I34 –…* | 1010 | *…* | … |

**1.3 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. 1.16.

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.

Таблица 1.16 – Функции записи, хранения, чтения и регенерации информации

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *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 28-разрядной шиной команд\данных, определяется общей емкостью 28 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. 1.7. Обозначение банка, применяемое на структурных схемах, приведено на рис. 1.8.

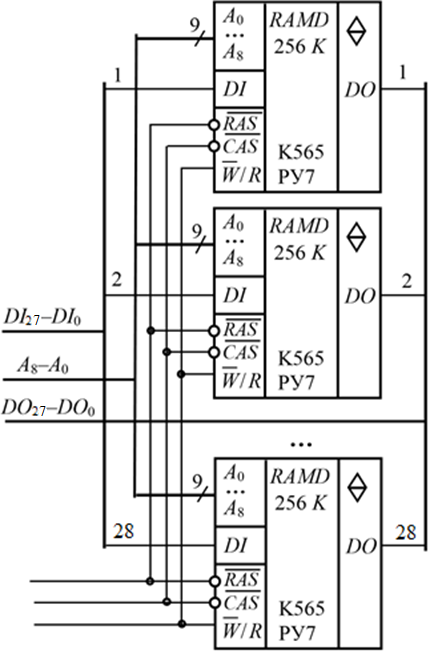


Рисунок 1.7 — Банк памяти



Рисунок 1.8 — Обозначение банка

Общее количество банков, составляющих подсистему ОЗУ (в соответствии с заданной емкостью памяти 1024 Кбайт), рассчитывается путем деления заданной емкости ЗУ на емкость 1 банка: 1024/1024 = 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. 1.17.

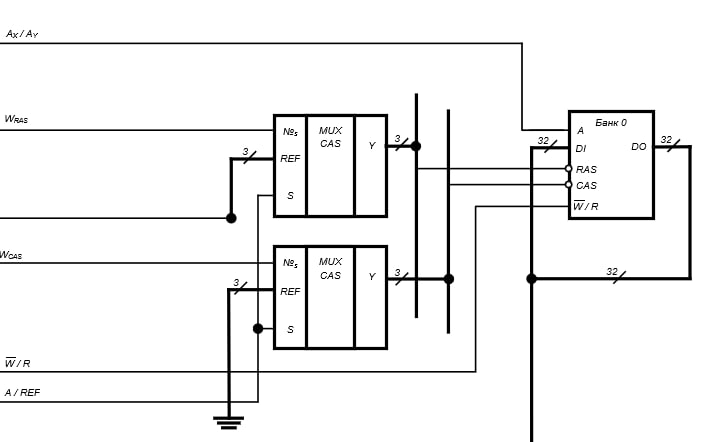


Рисунок 1.9 — Структурная схема блока ОЗУ

Управляющая информация из таблицы 1.17 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*.

*W \ R*

*А*

*DIO*

*W \ R*

*REF*

*Аx \ Ay*

*А \ REF*

***ОЗУ***

***ГТИ***

*RAS*

*W*

*СAS*

*W*

*А \ R*

*sА*

*RAS*

*W*

*СAS*

*W*

ШД

ША

32

18

+

2

ШУ

*m*

*D*

*O*

*А*

9

32

***П***

***ЗУ***

*RAS*

*W*

2

5

32

20

*С*

*S*

*Sw*

1

Рисунок 1.10 — Реализация подсистемы памяти

Таблица 1.17 – Управляющая информация

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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.

Формат команд *I1*, доработанный с учётом длины поля прямого адреса, представлен на рисунке 1.11.

|  |
| --- |
| 32 24 23 21 20 17 0 |

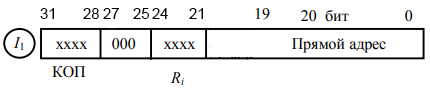


Рисунок 1.11 – Формат команды *I1*, доработанный с учетом длины поля прямого адреса

Будем считать, что для решения задач управления память компьютера разделена на 8 равных базовых областей, причём каждая область содержит по 512 страниц объёмом 256 машинных слов. Таким образом, разрядность поля «смещения» в команде формата *I4* (см. рисунок 1.12) должна быть выбрана равной 9 битам. Исполнительный адрес при этом определяется формулой:

|  |
| --- |
| 32 24 23 21 20 17 16 13 12 9 8 0 |



Рисунок 1.12 – Формат команды *I4*, доработанный с учётом для поля смещения

**1.4 Разработка обобщенной структуры микроЭВМ на основе системы команд**

Обобщённая структурная схема может быть представлена как совокупность функциональных блоков, соединенных между собой в соответствии с требованиями интерфейсов.

В структуре проектируемого спецкомпьютера можно выделить следующие основные блоки:

- блок обработки данных (БОД);

- устройство управления (УУ);

- запоминающее устройство (ЗУ);

- устройства ввода-вывода (УВВ).

Обобщенная структура данного компьютера представлена на рисунке 1.13.

УУ является ядром проектируемого микрокомпьютера. Оно предназначено для формирования микрокоманд посылаемых в БОД, принятия соответствующего решения при анализе признаков поступающих от БОД (организация ветвления), прерывания выполнения текущей программы при возникновении прерывания от УВВ и выполнения микропрограммы обработки информации от УВВ, предназначено для управления ОЗУ и контроллером ПДП.



Рисунок 1.13 – Обобщённая структура специализированного микрокомпьютера

БОД предназначен для обработки данных, выдачи результата и признаков, сохранения данных в системе РОН. Непосредственная обработка данных выполняется в АЛУ, устройстве, входящем в состав процессора. Обработка в АЛУ осуществляется над всеми битами одновременно.

В состав ЦП также входит устройство управления выполнением программ.

Устройство управления необходимо для реализации следующих функций:

- дешифрация команды, выбираемой из памяти;

- выбор соответствующего цикла шагов;

- управление выполнением каждого шага;

- организация выполнения шагов в требуемой последовательности;

- принятие решений в соответствии с возникновением различных сигналов.

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 (счетчик адреса команды) 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. Арифметико-логическое 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 *I1*. При этом, устанавливая связи между *RG K*, блоками и модулями системы, добиваемся логического объединения воедино формата команды, аппаратных средств и принципа функционирования неймановской архитектуры (см. рисунок 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 *А*, разрядность которой должна быть равна разрядности поля *Rj*. Иными словами, считаем, что процессор имеет 16 РОН, а поле адреса – длину 4 бита.

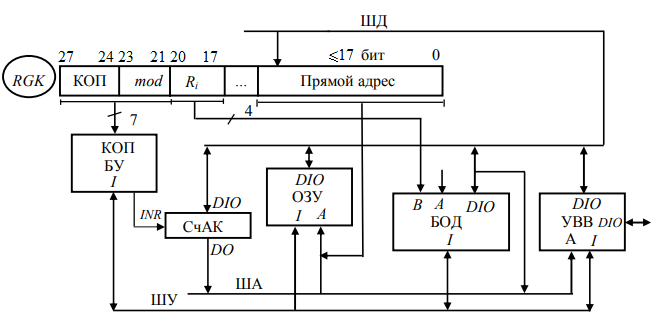


Рисунок 1.14 – Уточнённая структура компьютера (прямая и непосредственная адресация)

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 *I2* – *I3* предполагает введение в сформированную блок-схему (см. рисунок 1.14) дополнительных линий связи, как показано на рисунке 1.15.

Окончательный вариант укрупненной блок-схемы компьютера может быть получен путём эмуляции в полученной структуре команды *I4* формата *RX* (см. рисунок 1.16)

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. Данные 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 течении только двух тактов работы системы. После данной процедуры на адресный вход *B* БОД может быть скоммутирован адрес второго операнда.

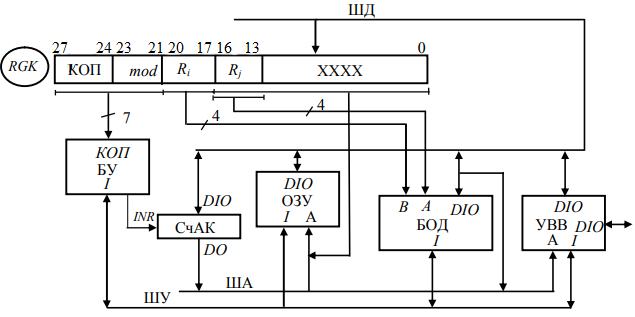


Рисунок 1.15 – Эмуляция форматов *I2* – *I3*

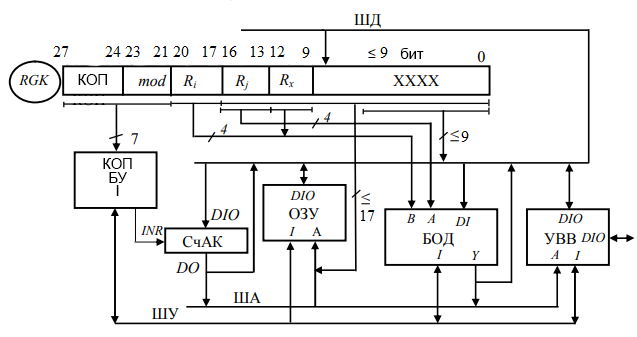


Рисунок 1.16 – Окончательный вариант укрупнённой блок-схемы

Доработка блок-схемы компьютера, приведённой на рисунке 1.16, с учётом подсистемы памяти (см. рисунки 1.9 и 1.10) приведена на рисунке 1.17.

Последующее формирование структуры спецкомпьютера будем вести с учётом известного принципа проектирования «сверху вниз». При этом систему аналогично вышеизложенной методологии, будем развёртывать в структурную схему путём 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.

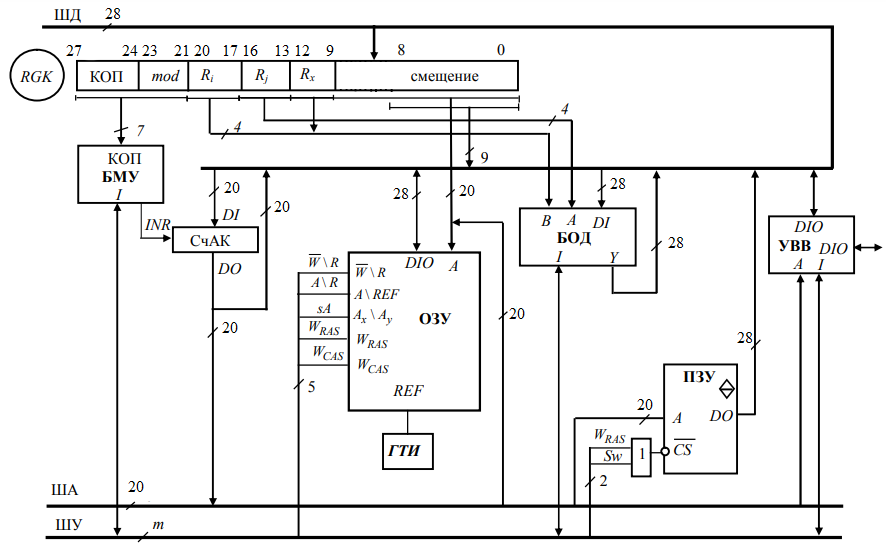


Рисунок 1.17 – Уточнение структуры компьютера с учётом подсистемы памяти

**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.

С целью получения высокой скорости решения задач функция системного счетчика команд (рисунок 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.

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

1) БВП – блок внутренней памяти;

2) АЛБ – арифметико-логический блок;

3) БР – блок регистра Q;

4) БУ – блок управления.

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 рисунке 2.2. 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.

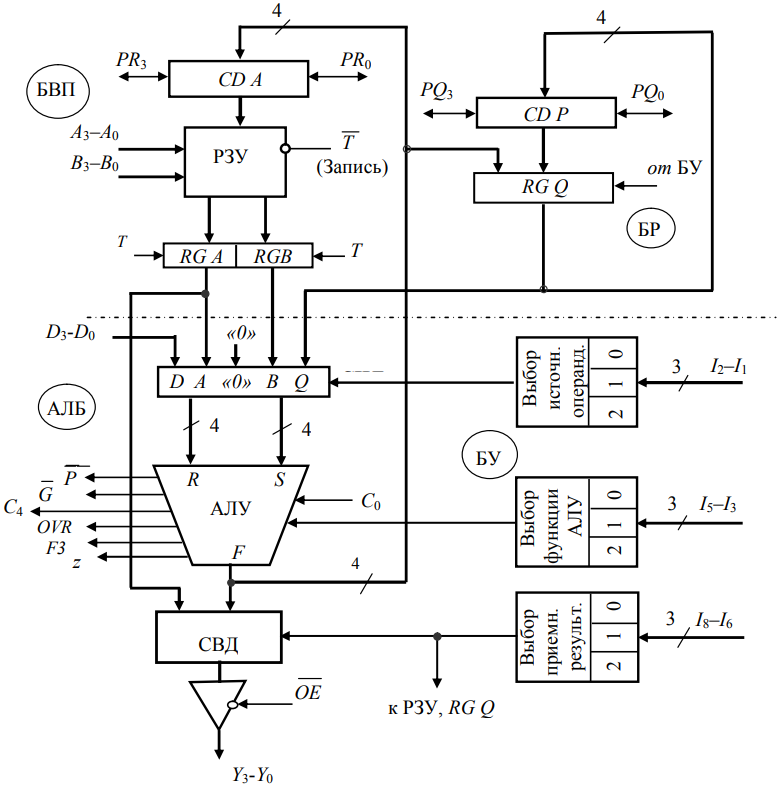


Рисунок 2.1 — Структурная схема К1804ВС1

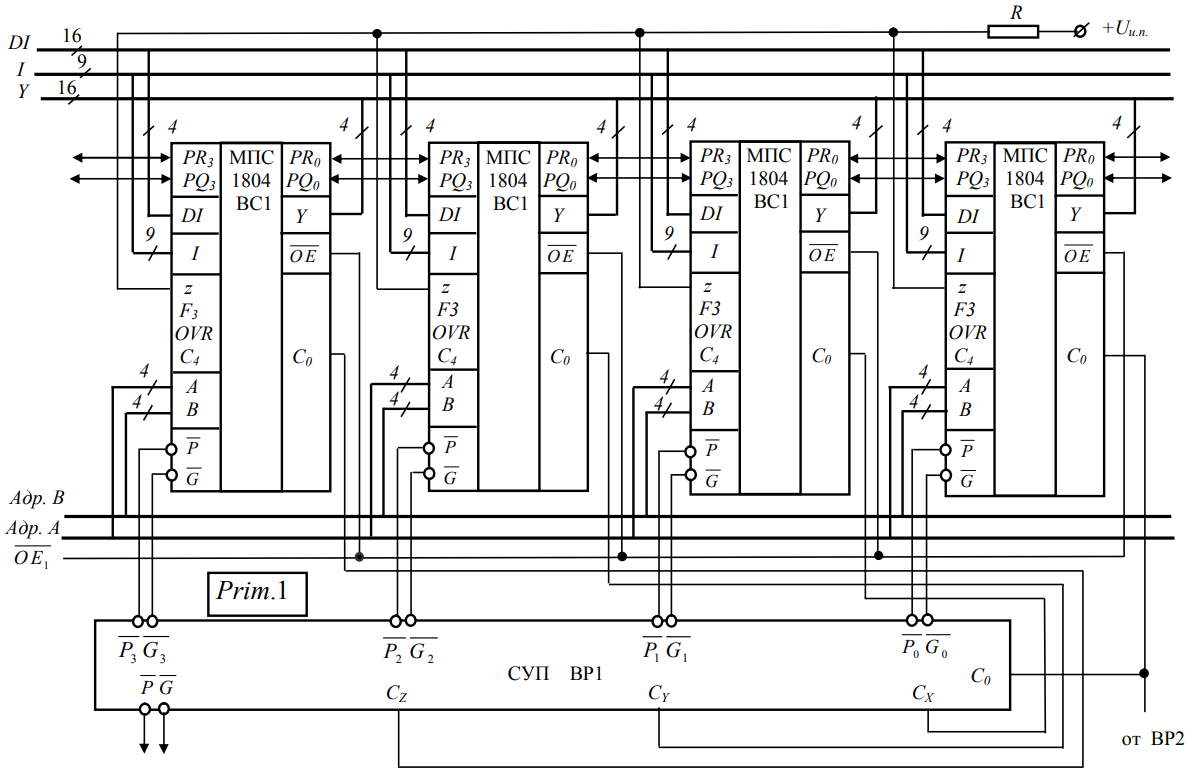


Рис. 2.2 – Схема *Prim*. 1

Особенностью схемы *Prim*. 1 является параллельное функционирование четырех секций под управлением общей шины микрокоманд *I*8 – *I*0. 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 .

Второй особенностью блока *Prim*. 1 является наличие общего для младшей МПС БОД и схемы СУП переноса *C*0. Аппаратно данный вход соединяется с выходом *C*0 СУСС К1804ВР2. 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 , младшей (правой крайней) МПС соединяются со входами , СУП ВР1. Соответствующий выход *Cx* подключается ко входу *C*0 первой средней МПС. Аналогично выводы , первой средней МПС соединяются ос входами , ИМС ВР1, а соответствующий выход *Cy* 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 , , 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 *F*3, *OVR*, *C*4 в промежуточных звеньях процессорного блока не используется, 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 , . 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 рисунке 2.3.

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*.

Выходы схемы СУСС является системная линия *CT*, определяющая условие перехода в микропрограмме или программе пользователя. Двунаправленные выводы *YN*, *YV*, *YC*, *YZ* являются транзисторными выходами признаков процессора или входами сигналов компьютера, формируемых внутренними схемами прерываний. Для реализации формата *RR* на программном и микропрограммном уровне в проектируемую схему БОД выключены мультиплексоры адресов *A* и *B* РЗУ (рисунок 2.4). При этом мультиплексор канала *A* имеет три информационных схода, что необходимо для выполнения команд с базово-индексной адресацией.

Исходя из задания к курсовому проекту, разрядность слова данных должна быть 32 бита и состоять из МПС 1804ВС1. Эта МПС предназначена для построения операционных блоков цифровых устройств с разрядностью кратной четырем. Следовательно, для получения 32-ух битной длины машинного слова необходимо использовать восемь микропроцессорных секций. Четыре МПС объединены в подсхему Prim. 1.

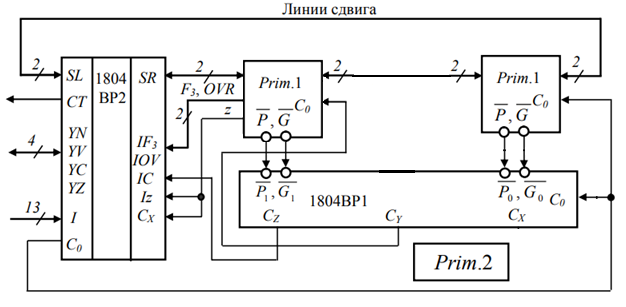


Рис. 2.3 – Схема блока обработки данных

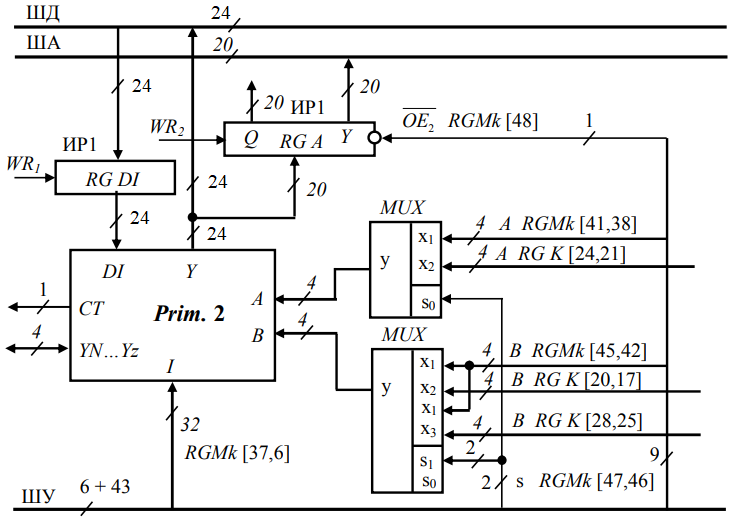


Рис. 2.4 – Включение блока обработки данных в схему компьютера

Для включения БОД на базе МПС К1804ВС1 в состав компьютера преобразуем устройство, показанное на рисунке 2.4, 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 рисунке 2.5.

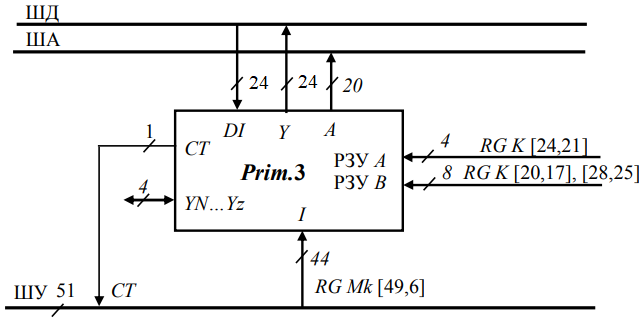


Рис. 2.5 – Включение блока обработки данных в схему компьютера

**2.2 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) К1804ВУ1

В формировании адресов на выходе СУАМ участвуют следующие входные сигналы ШУ: *C*0, *OR*0 – *OR*3, *S*1, *S*0, *PUP* и . Так при подаче «1» на вход *C*0 происходит последовательное увеличение адресов на единицу, а при подаче «1» на *i*-ый вход маски *OR*0 – *OR*3 адрес может быть изменен на величину 2*i*. Сигналы на входах *S*1, *S*2 коммутируют на вход один из четырех источников адреса, а сигналы *PUP* 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 их трех ИМС позволяет обращаться к памяти объемом в 4К слов. Соответственно при наращивании модулей объединяются линии шины управления *S*1, *S*0, *PUP*, , *T*, , всех схем, а также выполняется соединение выходов переноса *C*4 предыдущей СУАМ со входами *C*0 последующей. Сигнал переноса *С*0 используется в арифметической функции инкрементирования, поэтому при выборе параметров тактовых импульсов следует учитывать время прохождения переноса через все модули (рисунок 2.6).

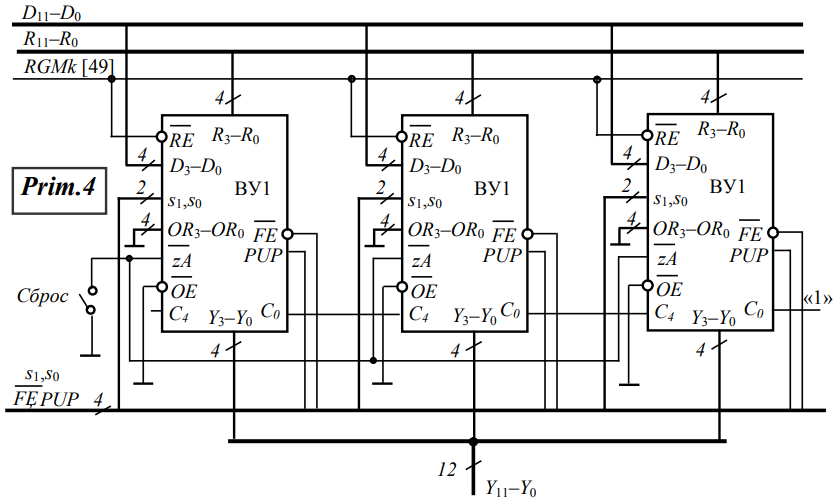


Рис. 2.6 – Схема *Prim*. 4

Для упрощения проектируемой системы входы БМУ *OR*0 – *OR*3 всех секций соединяются с шиной «земля», что позволяет исключить маскирование адресов, используемой в многопроцессорных системах. Вход управления выходной шиной *Y*11 – *Y*0 также подключается к нулю, что фиксирует адреса СУАМ на выходах ПЗУ МПП. Сигнал подключен к кнопке «Сброс» и используется для перевода компьютера в состояние первоначальной загрузки при включении.

Что касается процесса управления записью адресов с шины *R*11 – *R*0 во внутренний регистр адреса, то с этой целью вход К1804ВУ1 подключен к выходу регистру микрокоманды, например *RGMk*. Это позволяет сохранять адрес перехода в соответствующем регистре до требуемого момента времени.

Для организации условных и безусловных переходов на микропрограммном уровне в состав бока микропрограммного управления включается стандартное ПЗУ 556РТ14. Основное назначение данного модуля состоит в управлении источниками адреса микропрограммной памяти, расположенными в БИС СУАМ. В соответствии с поставленной целью в ПЗУ выбираются две соседние ячейки: четная и нечетная, адреса которых конкретизируются младшей адресной линией *A*0. Например, *XXXX*0 и *XXXX*1, где X – это произвольное значение бита. Как правило, в качестве адресной информации, подаваемой на вход *A*0, используется признак *CT*, поступающий от БОД. Если в процессе вычислений признак оказался равен «0», то из ПЗУ читается четное управляющее слово, а если признак *CT* = 1, то из ПЗУ читается нечетной 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.

Для управления режимом выполнения переходов используются выходы ПЗУ *Q*5 – *Q*2 (рисунок 2.7).

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.

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 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.

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 рисунке 2.8. Выполнение программы обработки прерываний всегда завершается командой передачи управления микропрограмме «Выборка команды».

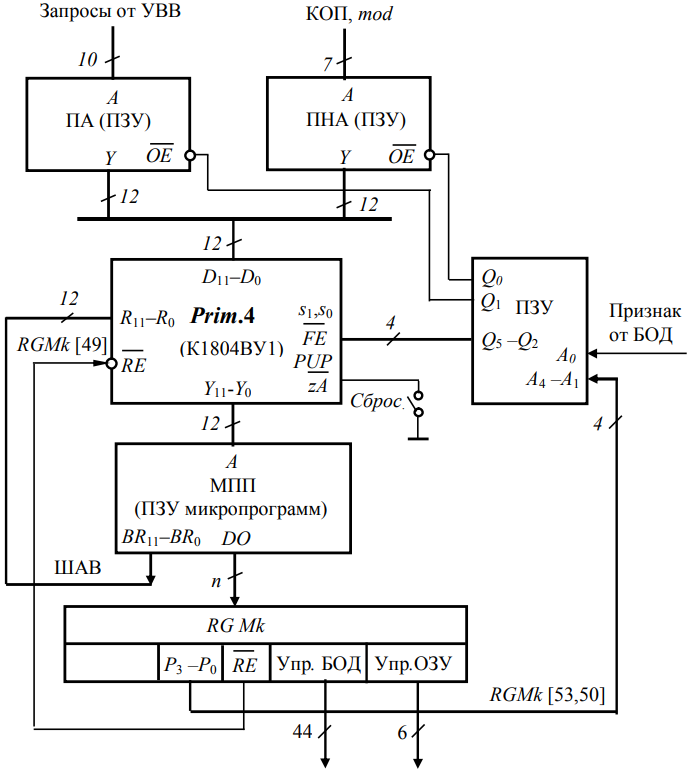


Рис. 2.7 – Структура БМУ на основе СУАМ К1804ВУ1

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 данный 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 *RG* *A* 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.

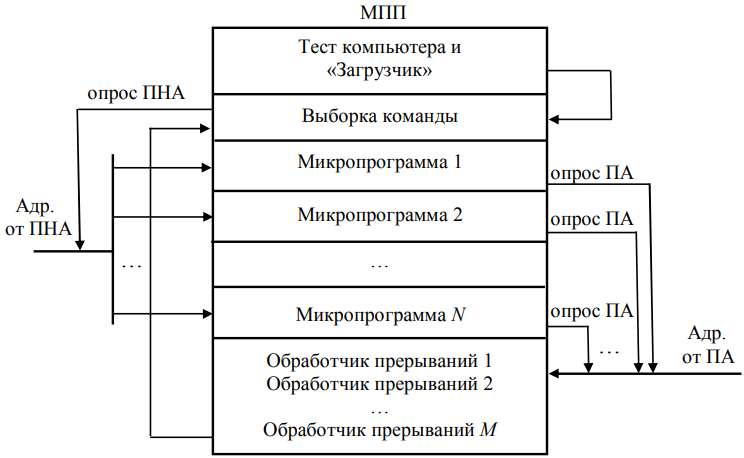


Рис 2.8 – ПЗУ микропрограмм

Кнопка «Сброс» внешним монтажом подключена ко входу , 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». 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 памяти служебных программ, драйверов, программ пользователя и установка блоков компьютера и интерфейса в требуемое состояние. Далее на экран монитора выводиться 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 рисунке 2.9. При реализации сброса в автоматическом режиме следует помнить, что время действия сигнала должно быть достаточным для проектирования 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.

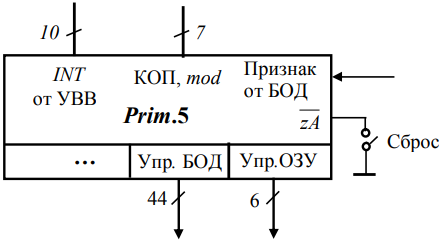


Рис 2.9 – Схема модуля устройства управления

**2.3 Разработка системы ввода-вывода данных**

В данном проекте используется подключение к микро-ЭВМ внешних носителей информации. Обмен информацией ведется через порты ввода-вывода.

В структуре специализированного микрокомпьютера выделено 12 портов ввода-вывода. 8 портов являются двунаправленными.

Система портов строится на основе регистров К1804ИР3. Данная микросхема представляет собой 8-разрядный параллельный двунаправленный регистр, предназначенный для организации порта ввода/вывода данных в микро-ЭВМ. Назначение выводов ИМС К1804ИР3 дано в таблице 2.3. Условно - графическое обозначение ИМС представлено на рисунке 2.9.

Таблица 2.1 – Назначение выводов К1804ИР3

|  |  |  |
| --- | --- | --- |
| Обозначение вывода | Назначение вывода | Вид выхода |
| 1 | 2 | 3 |
| DA0 – DA7 | Входы регистра R, выходы регистра S | Двунаправленный, три состояния |
| DB0 – DB7 | Входы регистра S, выходы регистра R | Двунаправленный, три состояния |
| FLS | Выход триггера запросов обмена FLS | Однонаправленный, два состояния |
| FLR | Выход триггера запросов обмена FLR | Однонаправленный, два состояния |
| RFLS | Вход обнуления триггеров запросов FLS | Однонаправленный, два состояния |
| RFLR | Вход обнуления триггеров запросов FLR | Однонаправленный, два состояния |
| EZA | Вход разрешения выхода регистра S | Однонаправленный, два состояния |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| EZB | Вход разрешения выхода регистра R | Однонаправленный, два состояния |
| CR | Вход синхронизации для триггера FLR и регистра R | Однонаправленный, два состояния |
| CS | Вход синхронизации для триггера FLS и регистра S | Однонаправленный, два состояния |
| ECR | Вход разрешения записи регистра R | Однонаправленный, два состояния |
| ECS | Вход разрешения записи регистра S | Однонаправленный, два состояния |

Интегральная микросхема предназначена для организации систем ввода-вывода компьютерных систем. В состав БИС входят: *RG R* и *RG S* – два 8-разрядных буферных регистра данных, *T* – триггер запросов обмена, *DF* – два детектора фронта, *DA* и *DB* – выходные каскады магистралей данных с третьим состоянием шины.

К портам ввода-вывода, выполненных на микросхемах K1804ИР3, подключается внешние устройства (ВУ), через которые будет организовываться ввод и вывод данных.

Так как шина данных разрабатываемого микрокомпьютера 32-разрядная, а порты ввода/вывода 8-разрядные, для обмена информацией через порты ввода/вывода используются буферные регистры, служащие для преобразования выводимой информации из байта в слово.

Для организации требуемого количества портов ввода/вывода требуется 12 микросхем K1804ИР3. Сигнал *FLS* от каждого порта ввода/вывода является запросом на прерывание от данного порта. Выбор определённого порта осуществляется с помощью дешифратор отдельно от портов ввода и вывода, следовательно, необходимо использовать два дешифратора открываемых с помощью различных сигналов. Так как всего портов 8, то необходимо три бита для кодирования номера порта. Таким образом, для управления системой портов ввода-вывода необходимо пять бит от УУ.



Рисунок 2.9 – УГО К1804ИР3

В состав данной ИМС входят 2 8 разрядных регистра *R*, *S* для реализации двунаправленного режима работы, а также два триггера запросов обмена *FLS*, *FLR*. Синхронизация работы ИМС K1804ИР3 осуществляется по положительному фронту импульсах на входах *CS* и *CR*. Сигнал *FLS* от каждого порта ввода/вывода является запросом на прерывание от данного порта.

В режиме считывания из порта внешнее устройство формирует сигнал *WR* = 0, указывающий на запрос от ВУ. Переход *WR* из «1» в «0» формирует сигнал разрешения записи в *RGR* через *DA* (7…0). При этом устанавливается сигнал запроса прерываний *FS0* = «1». Процессор принимает сигнал запроса и формирует сигнал IOR на шине управления. Если *IOR* = «0» и адрес порта указан верно, то на вход *EZA* подается «1», что соответствует разрешению считывания данных из порта на ШД.

В режиме записи в порт процессор формирует на ШУ сигнал *IOW* = «0». На вход *ECR* передается «0», что соответствует перепаду из «1» в «0», т.к. сигнал *IOW* был установлен в «1». Подача «0» на вход *ECR* разрешает передачу данных с ШД на ВУ и в тоже время формирует сигнал *FR0* = «1» для передачи запроса к ВУ на вход *RDY*. При совпадении адреса порта, на вход *EDB* попадает «1», что разрешает выдачу данных с ШД на внешнее устройство.

Структурная схема портов ввода-вывода представлена на рисунке 2.10.



Рисунок 2.10 – Структурная схема портов ввода-вывода

**3 ПРОЕКТИРОВАНИЕ ВНУТРЕННЕГО ИНТЕРФЕСА**

**СПЕЦКОМПЬЮТЕРА**

**3.1 Разработка и описание внутреннего интерфейса компьютера для**

**реализации фаз выборки и исполнения команд**

Обобщённая структурная схема может быть представлена как совокупность функциональных блоков, соединенных между собой в соответствии с требованиями интерфейсов.

В структуре проектируемого спецкомпьютера можно выделить следующие основные блоки:

- блок обработки данных (БОД);

- устройство управления (УУ);

- запоминающее устройство (ЗУ);

- устройства ввода-вывода (УВВ).

Обобщенная структура данного компьютера представлена на рисунке 3.1.

УУ является ядром проектируемого микрокомпьютера. Оно предназначено для формирования микрокоманд посылаемых в БОД, принятия соответствующего решения при анализе признаков поступающих от БОД (организация ветвления), прерывания выполнения текущей программы при возникновении прерывания от УВВ и выполнения микропрограммы обработки информации от УВВ, предназначено для управления ОЗУ и контроллером ПДП.



Рисунок 3.1 – Обобщённая структура специализированного микрокомпьютера

БОД предназначен для обработки данных, выдачи результата и признаков, сохранения данных в системе РОН. Непосредственная обработка данных выполняется в АЛУ, устройстве, входящем в состав процессора. Обработка в АЛУ осуществляется над всеми битами одновременно.

В состав ЦП также входит устройство управления выполнением программ.

Устройство управления необходимо для реализации следующих функций:

- дешифрация команды, выбираемой из памяти;

- выбор соответствующего цикла шагов;

- управление выполнением каждого шага;

- организация выполнения шагов в требуемой последовательности;

- принятие решений в соответствии с возникновением различных сигналов.

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 (счетчик адреса команды) 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. Арифметико-логическое 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 *I1*. При этом, устанавливая связи между *RG K*, блоками и модулями системы, добиваемся логического объединения воедино формата команды, аппаратных средств и принципа функционирования неймановской архитектуры (см. рисунок 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 *А*, разрядность которой должна быть равна разрядности поля *Rj*. Иными словами, считаем, что процессор имеет 16 РОН, а поле адреса – длину 4 бита.

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 *I2* – *I3* предполагает введение в сформированную блок-схему (см. рисунок 1.14) дополнительных линий связи, как показано на рисунке 3.3.

Окончательный вариант укрупненной блок-схемы компьютера может быть получен путём эмуляции в полученной структуре команды *I4* формата *RX* (см. рисунок 3.4)

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. Данные 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 течении только двух тактов работы системы. После данной процедуры на адресный вход *B* БОД может быть скоммутирован адрес второго операнда.

Доработка блок-схемы компьютера, приведённой на рисунке 1.16, с учётом подсистемы памяти (см. рисунки 1.9 и 1.10) приведена на рисунке 1.17.

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.

**3.2 Разработка интерфейса спецкомпьютера для обработки прерываний   
 от схем ввода-вывода**

Система прерываний предназначена для прерывания программы пользователя, если возникло прерывание от внешних устройств, либо внутреннее прерывание. Прерывание программы пользователя осуществляется путём перехода компьютерной системы в режим анализа запросов и выполнения процедур распознавания портов. Причём этот переход инициализируется не командой в текущей программе, а внешним сигналом, который называют запросом прерывания.

Каждое прерывание имеет свой приоритет, что обуславливает выполнения того прерывания, которое имеет наивысший приоритет, когда прерываний возникает несколько. Система прерываний должна реализовывать данный аспект. Также система прерываний должна поддерживать выполнение пользовательской программы, если прерываний не поступает, т.е. можно сказать, что пользовательская программа имеет низший приоритет. Система прерываний также должна обеспечивать загрузку на выполнение микропрограммы, обслуживающей то прерывание, которое необходимо обслужить в данный момент.

При проектировании данного спецкомпьютера, необходимо реализовать систему прерываний вида 2/12-15, т.е. двухуровневую систему 15 прерываний, 12 из которых являются прерываниями от портов ввода/вывода и три внутренних прерываний. Можно выделить следующие уровни прерываний:

- прерывание от пользовательской программы (имеет низший приоритет);

- прерывания от портов ввода/вывода и внутренние прерывания (имеют высший приоритет).

В разрабатываемой системе количество прерываний 15, поэтому для организации выделения прерываний с высшим приоритетом требуется ПЗУ с разрядностью адреса – 11 и данных 12 (разрядность адреса микропрограммы). Так как имеется ПЗУ 556РТ14, разрядность адреса которой – одиннадцать, а данных – четыре, то потребуется три таких микросхемы.

Функция опроса системы прерываний чаще всего выполняется после выполнения очередной команды компьютера (см. рисунок 2.7). При этом каждая микропрограмма, соответствующая команде высокого уровня, последней микрокомандой должна иметь микрокоманду перехода к опросу системы прерываний (ПА).

Входная шина ПЗУ ПА (преобразователя адреса), которая представлена на рисунке 2.8, используется для приёма вектора запросов на прерывание, поступающих от портов ввода-вывода К1804ИР3 (см. рисунок 2.11). 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 программе, расположенной в ОЗУ, что позволяет обслуживать практически любое количество портов.

Если код поступивший на ПА равен нулю (прерываний нет), то ПА выдаст адрес микропрограммы, отвечающей за загрузку следующей макрокоманды из ОЗУ в регистр макрокоманд и начало выполнения микропрограммы соответствующей коду операции (КОП) находящегося в данный момент в регистре макрокоманд, т.е. при отсутствии прерываний процесс выполнения пользовательской программы продолжается.

В последней микрокоманде каждой макрокоманды должен содержаться соответствующий адрес для ПЗУ ветвлений в УУ, чтобы на его выходе бит, идущий на вход () ПА, был установлен для разрешения выдачи адреса микропрограммы обработки прерывания.

**3.3 Проектирование внутреннего интерфейса для выполнения**

**алгоритмов ветвления вычислительного процесса**

Для организации условных и безусловных переходов на микропрограммном уровне в состав блока микропрограммного управления включается стандартное ПЗУ К1804ВУ3 или нестандартное ПЗУ, программируемое разработчиком компьютера на стадии изготовления. Основное назначение данного модуля состоит в управлении источниками адреса микропрограммной памяти, расположенными в БИС СУАМ. В соответствии с поставленной целью в ПЗУ выбираются две соседние ячейки: четная и нечетная, адреса которых конкретизируются младшей адресной линией A0. Например, ХХХХ0 и ХХХХ1, где Х – это произвольное значение бита. Как правило, в качестве адресной информации, подаваемой на вход A0, используется признак CT, поступающий от БОД. Если в процессе вычислений признак оказался равен «0», то из ПЗУ читается четное управляющее слово, а если признак CT = 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.

**3.4 Проектирование системы ПДП и интерфейса связи канала и ОЗУ**

**компьютера**

Система прямого доступа к памяти позволяет осуществить непосредственный обмен данными между памятью и периферийными устройствами под управлением контроллера ПДП без участия БОД, что позволяет повышать скорость выполнения обмена.

Устройство, осуществляющее управлением передачей данных при ПДП, называется контроллером ПДП и выполняет следующие функции: управление шиной адреса, управление передачей данных, формирование адреса, подсчет числа слов, управление режимов передачи. Контроллер ПДП должен передавать соответствующий адрес на шину адреса памяти и вырабатывать сигналы управления передачей данных между памятью и устройством ввода/вывода. Контролер ПДП должен содержать указатель адреса, который формирует адрес ячейки памяти, из которой считываются или в которую записываются очередное слово передаваемых данных. Содержимое этого указателя должно увеличиваться или уменьшаться после передачи очередного слова данных. Перед началом передачи данных в контроллер ПДП поступают информация о числе передаваемых слов и начальный адрес. Во время передачи данных контроллер ПДП должен осуществлять контроль числа переданных слов и закончить передачу по достижении заданного числа слов. Таким образом, контроллер ПДП должен содержать схему управления режимом передачи, которая определяет направление потока данных, способ определения конца передачи и т. д.

Восьмиразрядный счетчик адресов ПДП КМ1804ВУ6 предназначен для использования в контроллерах ПДП. Основные функции адресного генератора ПДП заключается в формировании последовательности адресов ячеек памяти при передаче данных в память или из нее, в подсчет числа предаваемых слов и формировании соответствующего сигнала по завершении передачи данных.

Условно-графическое обозначение микросхемы КМ1804ВУ6 представлено на рисунке 3.2.

Назначение выводов микросхемы КМ1804ВУ6 показано в таблице 3.1.



Рисунок 3.2 – УГО микросхемы КМ1804ВУ6

Таблица 3.1 – Назначение выводов микросхемы КМ1804ВУ6

|  |  |
| --- | --- |
| Обозначение | Назначение |
| *D*0-*D*7 | Двунаправленная шина данных, используется для ввода/ вывода информации |
| *A0*-*A7* | Выход адреса, предназначен для вывода информации из счетчика адреса |
| *ОЕА* | Вход разрешения выдачи адреса. Используется для разрешения трёх стабильной шины *A*. При *OEA* = 0 разрешается вывод содержимого счётчика адреса на шину *А*, при *OEA* = 1 шина A переводится в высокоомное состояние. |
| *I0*-*I2* | Вход инструкции, определяет одну из восьми инструкций адресного генератора ПДП |
| *ACI* | Входной перенос счётчика адреса. Используется как вход переноса счётчика адреса (при *ACI* = 0 содержимое счётчика адреса увеличивается на 1, при *ACI* = 1 содержимое счётчика не меняется), а также для управления работой счётчика при его разрешении (при *ACI* = 0 разрешается режим счёта). |
| *АСО* | Выходной перенос счётчика адреса; *ACO* = 0, если *ACI* = 0 и во всех разрядах счётчика адреса единицы. |
| *WCI* | Входной перенос счётчика слов. Используется как вход переноса счётчика слов (при *WCI* = 0 содержимое счётчика слов увеличивается на, при *WCI* = 1 содержимое счётчика слов изменяется) и для управления работой счётчика слов при его разрешении (при *WCI* = 0 разрешается режим счёта). |
| *WCO* | Выходной перенос счётчика слов; *WCO* = 0, *WCI* = 0 и во всех разрядах счётчика слов единицы. |
| *DON* | Выход индикации конца передачи. Наличие единицы на этом выходе определяет окончание передачи данных в трех режимах управления. |

Адресный генератор ПДП КМ1804ВУ6 имеет следующую особенность: возможность наращивания для получения адреса любой разрядности, кратной восьми.

Функционирование системы ПДП осуществляется посредством шести разрядов от УУ. Сигнал DON идет на мультиплексор признаков в УУ, как признак окончания работы системы ПДП.

Так как разрядность адреса ОЗУ разрабатываемого спецкомпьютера составляет двадцать бит, то при создании системы ПДП необходимо использовать три микросхемы КМ1804ВУ6 (см. рисунок 3.3).



Рисунок 3.3 – Система ПДП

**4 РАЗРАБОТКА МИКРОПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**4.1 Формат микрокоманды. Микропрограммная интерпретация команд языка компьютера**

В разрабатываемом спецкомпьютере используются макрокоманды и микрокоманды. Макрокоманды вводятся пользователем и определяют программу. Каждой макрокоманде в данном спецкомпьютере ставится в соответствие микропрограмма, состоящая из микрокоманд. Все микропрограммы находятся в микропрограммной памяти (ПЗУ) и вводятся разработчиком. Микропрограммы могут быть изменены за счет возможности перепрошивки микропрограммной памяти. Это обуславливает гибкость разрабатываемого спецкомпьютера. Каждая макрокоманда состоит из кода операции (КОП) и полей операндов. КОП предназначен для кодирования макрокоманд. КОП поступает в ПНА в качестве адреса ПЗУ, в котором по данному адресу расположен адрес начала микропрограммы, соответствующей данной макрокоманде. Данная микропрограмма обуславливает выполнение требуемой макрокоманды.

Разработка формата микрокоманды представлена в таблицах 4.1 – 4.4.

Таблица 4.1 – Управление ОЗУ

|  |  |  |
| --- | --- | --- |
| Биты в RGMk | Название | Назначение |
| 0 | WCAS | Вход стробирования адреса столбца ОЗУ |
| 1 | WRAS | Вход стробирования адреса строки ОЗУ |
| 2 | sA | Выбор адреса строки/столбца в ОЗУ |
| 3 | A \ R | Выбор режимов регенерации или обращения к ОЗУ |
| 4 | \ R | Запись/чтение из ОЗУ |
| 5 | Sw | Активация ПЗУ |

Таблица 4.2 — Управление блоком обработки данных

|  |  |  |
| --- | --- | --- |
| Биты в RGMk | Название | Назначение |
| 1 | 2 | 3 |
| 6-9 |  | Входы разрешения записи меток *C, Z, N, V* в регистр состояния программ (RGM) К1804ВР2 |
| 10 |  | Вход разрешения записи в *RGM* К1804ВР2 |
| 11 |  | Вход разрешения записи в *RGN* К1804ВР2 |
| 12 |  | Вход разрешения вывода информации на шину *CT* К1804ВР2 |
| 13 |  | Вход разрешения вывода информации по шине  *Y (C, Z, N, V)* К1804ВР2 |
|  | | |
| Продолжение таблицы 4.2 | | |
| 14 |  | Вход разрешения сдвига в К1804ВР2 |
| 15-27 |  | Сигналы управления функцией СУСС К1804ВР2 |
| 28-36 |  | Сигналы управления функцией МПС К1804ВС2 |
| 37 |  | Управление выходной шиной Y в МПС К1804ВС2 |
| 38-41 |  | Адрес регистра в МПС К1804ВС2 по каналу А |
| 42-45 |  | Адрес регистра в МПС К1804ВС2 по каналу B |
| 46 |  | Выбор источника операндов для входа R АЛУ |
| 47 |  | Вход разрешения записи в РЗУ |
| 48 |  | Управление коммутаторами адресов каналов А и В |
| 49 |  | Управление выдачей адреса из *RGA* на ША |
|  |  |  |

Таблица 4.3 – Управление СУАМ

|  |  |  |
| --- | --- | --- |
| Биты в RGMk | Название | Назначение |
| 50 |  | Разрешение записи с шины D во внутренний регистр адреса |
| 51-54 | P0 – P3 | Часть адреса управляющего слова в ПЗУ |
| 55-56 | BR0 – BR11 | Выводы явного указания адреса перехода |

Таблица 4.4 – Управление блоком ввода/вывода данных

|  |  |  |
| --- | --- | --- |
| Биты в RGMk | Название | Назначение |
| 67 |  | Разрешение записи в порты ввода/вывода |
| 68 |  | Разрешения чтения из портов ввода/вывода |

В таблицах 4.5 – 4.8 представлены микропрограммы операций *move*, *jmp*, *jz* для различных типов адресации. В таблице 4.9 представлена микропрограмма формирования частного от деления двух целых чисел больших нуля (Дм < Дт) по методу с восстановлением остатка. Микрокоманды обращения к памяти пропущены.

Команда безусловного перехода, при использовании непосредственной адресации, требует лишь записи в регистр СчАК, который для упрощения предполагается разместить в РЗУ. При этом адрес перехода выставлен на шину данных регистром команды. Будем считать, что СчАК – это регистр с последним номером в РЗУ, то есть 1111.

Таблица 4.5 — Микропрограммы операций

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | К1804ВС1 (БОД) | | | | | | К1804ВР2 (БОД) | | | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |
| 66..55 | 54..51 | 50 | 49 48 | 47 46 | 45..42 | 41..38 | 37 | 36..28 | | 27..15 | 14 13 12 | 11 10 | 9 8 7 6 |
| **Команда move R1, mem (прямая адресация)** | | | | | | | | | | | | | | |
| N. | Переход к адресации памяти, запись адреса строки/столбца | | | | | | | | | | | | | |
| N+1. | Загрузка слова с шины данных в R1 | | | | | | | | | | | | | |
| X |  | 1 | 10 | 01 | X | X | 0 | 2178 | | 08 | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ вкл.;  R=DA | Из RG K [23..20]  B=R1 | A=X | YМПС = раб. реж. | R=D  F=R+C0  F → B | | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| **Команда move R1, const (непосредственная адресация)** | | | | | | | | | | | | | | |
| N. | Загрузка слова с шины данных в R1 | | | | | | | | | | | | | |
| X |  | 1 | 10 | 01 | R1 | X | 0 | | 2178 | 08 | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ вкл.;  R=DA | Из RG K [23..20]  B=R1 | A=X | YМПС = раб. реж. | | R=D  F=R+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |

Таблица 4.6 — Микропрограммы операций

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | К1804ВС1 (БОД) | | | | | К1804ВР2 (БОД) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 66..55 | 54..51 | 50 | 49 48 | 47 46 | 45..42 | 41..38 | 37 | 36..28 | 27..15 | 14 13 12 | 11 10 | 9 8 7 6 |
| **Команда move R1, R2 (регистровая адресация)** | | | | | | | | | | | | | |
| N. | Чтение из R2 | | | | | | | | | | | | |
| X |  | 1 | 10 | 10 | X | X | 0 | 2178 | 08 | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ выкл.;  R=A | Из RG K [23..20]  B=R1 | Из RG K [19..16]  A=R2 | YМПС = Roff | R=R1  F=R+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| N+1. | Запись в R1 | | | | | | | | | | | | |
| X |  | 1 | 10 | 00 | X | X | 0 | 2178 | 08 | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ вкл.;  R=A | Из RG K [23..20]  B=R1 | Из RG K [19..16]  A=R2 | YМПС = раб. реж. | R=R1  F=R+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| **Команда jmp label (непосредственная адресация)** | | | | | | | | | | | | | |
| N. | Загрузка константы label в СчАК (R15) с шины данных | | | | | | | | | | | | |
| X |  | 1 | 11 | 01 | 1111 | X | 0 | 2178 | 08 | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | B=15 | A=X | YМПС = раб. реж. | R=D  F=R+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |

Таблица 4.7 — Микропрограммы операций

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | | БОД | К1804ВС1 (БОД) | | | | | К1804ВР2 (БОД) | | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  | |  |
| 66..55 | 54..51 | | 50 | 49 48 | 47 46 | 45..42 | 41..38 | 37 | 36..28 | 27..15 | 14 13 12 | 11 10 | | 9 8 7 6 |
| **Команда jz label (непосредственная адресация)** | | | | | | | | | | | | | | | |
| N. | Проверка признака Z (выход СТ = Z) | | | | | | | | | | | | | | |
| X |  | | 0 | 10 | 11 | X | X | 1 | X | 48 | 110 | 01 | | 1111 |
| N+2 | z=1, СМК. z=0, RGA | |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ выкл.;  R=DA | B=X | A=X | YМПС = Roff | X | CT=NZ | SL, SR = Roff  YВР2=Roff  CT= раб. реж. | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |
| N+1. | Загрузка константы label в СчАК (R15) с шины данных | | | | | | | | | | | | | | |
| X | |  | 1 | 11 | 01 | 1111 | X | 0 | 2178 | 08 | 111 | | 11 | 1111 |
|  | | СМК |  | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | B=15 | A=X | YМПС = раб. реж. | R=D  F=R+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | | Запрет RGM, RGN | Запрет записи признаков в RGM |
| N+2. | Микропрограмма выборки следующей команды | | | | | | | | | | | | | | |

Таблица 4.8 — Микропрограммы операций

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | К1804ВС1 (БОД) | | | | | | К1804ВР2 (БОД) | | | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |
| 66..55 | 54..51 | 50 | 49 48 | 47 46 | 45..42 | 41..38 | 37 | 36..28 | | 27..15 | 14 13 12 | 11 10 | 9 8 7 6 |
| **Команда move R1, [R2]+offset (относительная адресация)** | | | | | | | | | | | | | | |
| N. | Запись в R2 в R1 | | | | | | | | | | | | | |
| X |  | 1 | 00 | 00 | X | X | 0 | 2178 | 08 | | 111 | 11 | 1111 |
|  | СМК |  | YRGA= Roff  RGK A  RGK B | Запись в РЗУ вкл.;  R=A | Из RG K [23..20]  B=R1 | Из RG K [19..16]  A=R2 | YМПС = вкл. | R=R1  F=R+C0  F → B | C0=0 | | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| N+1. | Выполнение операции [R1]+offset и запись результата в RGA | | | | | | | | | | | | | |
| X |  | 1 | 00 | 11 | X | X | 0 | 2068 | X | | 111 | 11 | 1111 |
|  | СМК |  | YRGA= Roff  RGK A  RGK B | Запись в РЗУ выкл.;  R=DA | Из RG K [23..20]  B=R1 | A=X | YМПС = вкл. | R=DA  S=R1  F=R+S+C0  F → Y | I=X | | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| N+2. | Запись из RGA на ША и переход к адресации ОЗУ | | | | | | | | | | | | | |
| X |  | 1 | 10 | 11 | X | X | 0 | X | X | | 111 | 11 | 1111 |
|  | СМК |  | YRGA =  раб. реж.  RGK A  RGK B | Запись в РЗУ выкл.;  R=DA | B=X | A=X | YМПС = вкл. |  | I=X | | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |
| N+3.  N+4. | Запись адреса стобца/строки ОЗУ | | | | | | | | | | | | | |
| N+5. | Чтение слова с ШД и его запись в R1 | | | | | | | | | | | | | |
| X |  | 1 | 10 | 01 | X | X | 0 | 2178 | 08 | | 111 | 11 | 1111 |
|  | СМК |  | YRGA=Roff  RGK A  RGK B | Запись в РЗУ вкл.;  R=DA | Из RG K [23..20]  B=R1 | A=X | YМПС = вкл. | R=D  F=R+C0  F → B | C0=0 | | SL, SR = Roff  YВР2=Roff  CT= Roff | Запрет RGM, RGN | Запрет записи признаков в RGM |

Таблица 4.9 — Микропрограммы операций

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | | К1804ВС1 (БОД) | | | | | | | | К1804ВР2 (БОД) | | | | | | |
|  |  |  |  | |  |  | |  | |  |  | |  |  | |  | |  | |
| 66..55 | 54..51 | 50 | 49 48 | | 47 46 | 45..42 | | 41..38 | | 37 | 36..28 | | 27..15 | 14 13 12 | | 11 10 | | 9 8 7 6 | |
| **Команда div (деление, ДМ < ДТ)** | | | | | | | | | | | | | | | | | | | | |
| N. | Запись делителя с шины данных в регистр RG0 | | | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0000 | | X | 0 | 2178 | | 08 | | 111 | | 11 | | 1111 |
|  | СМК |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=0  (Дт) | | A=X | YМПС= вкл | R=D  F=R+C0  F → B | | C0=0 | | SL, SR = Roff  YВР2=Roff  CT= Roff | | Запрет RGM, RGN | | Запрет записи признаков в RGM |
| N+1. | Запись делимого с шины данных в регистр RG1 | | | | | | | | | | | | | | | | | | | |
| X |  | 1 | 10 | | 01 | 0001 | | X | | 0 | 2178 | X | | 111 | | 11 | | 1111 | |
| N+2. | Запись числа циклов деления с шины данных в регистр RG4 | | | | | | | | | | | | | | | | | | | |
| X |  | 1 | 10 | | 01 | 0100 | | X | | 0 | 2178 | X | | 111 | | 11 | | 1111 | |
| N+3. | Запись нуля в регистр частного RG3 | | | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0011 | | X | 0 | 218 | | X | | 111 | | 11 | | 1111 |
|  | СМК |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=X | | A=X | YМПС= вкл | F=0000 | | I=X | | SL, SR = Roff  YВР2=Roff  CT= Roff | | Запрет RGM, RGN | | Запрет записи признаков в RGM |
| N+4. | Проверка делителя на равенство нулю: RG0 RG0 и переход по z | | | | | | | | | | | | | | | | | | | |
| N+11  (End) |  | 0 | | 10 | 10 | | 0000 | | 0000 | 0 | 368 | | 58 | | 110 | | 01 | | 1111 |
|  | RGMk  z=0,  СМК  z=1 |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ выкл.;  R=A | | B=0 | | A=0 | YМПС= вкл. | R=R0  S= R0  F=RvS  F → B | | CT= | | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |

Продолжение таблицы 4.9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | | К1804ВС1 (БОД) | | | | | | | | К1804ВР2 (БОД) | | | | | |
|  |  |  |  | |  |  | |  | |  |  | |  |  |  | |  | |
| 66..55 | 54..51 | 50 | 49 48 | | 47 46 | 45..42 | | 41..38 | | 37 | 36..28 | | 27..15 | 14 13 12 | 11 10 | | 9 8 7 6 | |
| N+5. | Сдвиг регистра делимого RG1 влево на 1 разряд | | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0001 | | X | 0 | 4108 | | 08 | 110 | | 01 | | 1111 |
|  | СМК |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=1  (Дм) | | A=X | YМПС= вкл | S=B  F=S+C0  2F → B | | C0=0 | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |
| N+6. | Сдвиг регистра частного RG3 влево на 1 разряд | | | | | | | | | | | | | | | | | | |
| X |  | 1 | 10 | | 01 | 0011 | | X | | 0 | 4118 | 08 | | 110 | 11 | | 1111 | |
| N+7. | RGQ = RG Дм – RG Дт и переход по знаку результата | | | | | | | | | | | | | | | | | | |
| N+10 |  | 0 | | 10 | 01 | | 0001 | | 0000 | 0 | 2208 | | 40378 | 110 | | 01 | | 1111 |
| адрес перехода | F3=1  СМК,  F3=0  RGMk |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=1 | | A=0 | YМПС= вкл. | R=A  S=B  F=S-R-1+ +C0  F → Q | | CT=  C0=1 | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |
| N+8. | Инкремент регистра частного RG3 | | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0011 | | X | 0 | 2108 | | X | 110 | | 11 | | 1111 |
|  | СМК |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=3 | | A=X | YМПС= вкл. | S=B  F=S+C0  F → B | | I=X | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM, RGN | | Запрет записи признаков в RGM |

Окончание таблицы 4.9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | К1804ВУ1 (БМУ) | | | БОД | | К1804ВС1 (БОД) | | | | | | | К1804ВР2 (БОД) | | | | | |
|  |  |  |  | |  |  | |  | |  |  |  |  |  | |  | |
| 66..55 | 54..51 | 50 | 49 48 | | 47 46 | 45..42 | | 41..38 | | 37 | 36..28 | 27..15 | 14 13 12 | 11 10 | | 9 8 7 6 | |
| N+9. | Пересылка содержимого регистра RGQ в RG1 | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0001 | | X | 0 | 2118 | X | 110 | | 01 | | 1111 |
|  | СМК |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=1  (Дм) | | A=X | YМПС= вкл. | S=Q  F=S+C0  F → B | I=X | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |
| N+10. | Декремент RG4 и переход по признаку z | | | | | | | | | | | | | | | | | |
| N+5 |  | 0 | | 10 | 01 | | 0100 | | X | 0 | 2208 | 58 | 110 | | 01 | | 1111 |
| адрес перехода | z=0  СМК,  z=1  RGMk |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=4 | | A=X | YМПС= вкл. | R=0  S=B  F=S-R-1+ +C0  F → B | CT=  C0=0 | SL, SR = Roff  YВР2=Roff  CT= вкл. | | Запрет RGM,  RGN вкл. | | Запрет записи признаков в RGM |
| N+11. | Останов | | | | | | | | | | | | | | | | | |
| X |  | 1 | | 10 | 01 | | 0011 | | 0011 | 1 | 2108 | 08 | 111 | | 11 | | 1111 |
|  | RGMk |  | | YRGA=Roff  RGMk A  RGMk B | Запись в РЗУ вкл.;  R=DA | | B=3 | | A=3 | YМПС= вкл. | S=B  F=S+C0  F → B | C0=0 | SL, SR = Roff  YВР2=Roff  CT= Roff | | Запрет RGM, RGN | | Запрет записи признаков в RGM |

При выборке очередной команды для фиксации адреса перехода, находящего в СчАК, на следующий такт времени используется внутренний регистр блока обработки данных *RGA*. Этот регистр подключен к выходу АЛУ, поэтому он должен быть настроен на выдачу необходимого адреса. При этом на вход регистра *RGA* подается нулевой потенциал.

На микропрограммном уровне условный переход реализуется задействованием СУСС 1804ВР2. Нулевой потенциал на входе разрешает запись признаков во внутренний регистр RGN. При этом код операции *I12-I0* задается таким образом, чтобы выход CT показывал значение нужного признака. Данный выход подключен к младшему разряду адреса ПЗУ микропрограммного управления. Для реализации ветвления тут используется различное информационное наполнение соседних ячеек ПЗУ. При наличии единичного значения признака читается нечетная ячейка, которая содержит управляющую информацию, необходимую для выборки следующей команды по адресу перехода.

**4.2 Разработка программ арифметических операций**

От разрабатываемого спецкомпьютера требуется возможность находить значения функции *ln(x)* со значения аргументов в диапазоне [1.2, 1.9]. Алгоритм её вычисления был разработан в разделе 1, блок-схема изображена на рис. 1.6.

Ниже приведен текст программы для вычисления *ln(x)*. Значение аргумента функции берётся из памяти или задаётся константой. Регистры общего назначения обозначены как *R*0 *– R*15.

mov R0, 1 // i=1

mov R1, imax

mov R2, 1 // a=1

mov R3, x

dec R3 // a1=x-1

mov R4, 1 // a2=2

mov R5, R3 // a3=x

div R5, R4 // a3= a1/a2

cycle\_start:

sum R2, R5 // a=a+a3

mov R6, R0

sub R6, R1

jz end // if(i==imax)

inc R0

mov R6, R3

mov R7, R0

dec R7

mov R8, -1

mul R6, R8

mul R6, R7

div R6, R0 //a3(-1)(x-1)(i-1)/i

jmp cycle\_start

end:

**4.3 Разработка служебного микропрограммного обеспечения**

Для диагностики функционирования микроЭВМ необходимо иметь определенный набор программ для контроля блоков.

Функциональный контроль (ФК) решает следующие задачи: определение факта наличия неисправности в блоке и определение места неисправности. Вывод в правильности функционирования ОЗУ делается либо по результатам выполнения очередного элементарного теста (останов по ошибке), либо по конечному результату выполнения полного теста.

Методы ФК основаны на сравнении с эталонными сигналами входных реакций (сигналов) тестируемой схемы на заданные входные воздействия. Одним из основных узлов системы ФК является генератор тестов, предназначенный для формирования последовательности тестирующих и эталонных сигналов по заданному закону. В понятие теста включают состав, параметры и порядок следования электрических сигналов, подаваемых на испытуемую схему с целью измерения какого-либо параметра или контроля работоспособности. Переменной составляющей теста являются наборы входных и эталонных сигналов.

Наборы входных сигналов, задаваемые в виде машинных слов (кодов), определяют порядок обращения к элементам памяти и последовательность выполняемых операций. Математические адреса элементов памяти могут не совпадать с их физическими координатами на кристалле, это следует учитывать при анализе отказов БИС ЗУ.

Коды эталонных сигналов должны соответствовать выходным кодам исправной БИС ЗУ при заданных входных воздействиях, т. е. эквивалентность выходных и эталонных сигналов, определяемая путем логического сравнения, указывает на правильность функционирования БИС. Электрические режимы функционирования контролируемой схемы, как правило, не изменяются в пределах теста и выбираются в соответствии с НТД на конкретное изделие. Специализированные тесты с переменным режимом, хотя и более эффективны, но сложны в реализации и мало распространены. Заключение о правильности функционирования ЗУ делают либо по результатам выполнения очередного элементарного теста («останов по ошибке»), либо по конечному результату выполнения полного теста.

Контроль функционирования может быть совмещен с измерением (контролем) статических и динамических ‚параметров ЗУ, если позволяет точность и быстродействие аппаратуры контроля. Принципиальных методических отличий от уже рассмотренных методов измерения параметров ЗУ в этом случает нет.

Эффективность ФК решающим образом определяется построением теста. Существуют различные способы генерации тестовых последовательностей для контроля ЗУ.

По количеству циклов обращения к тестируемой схеме, выраженному через ее информационную емкость, алгоритмы ФК условно делятся на три типа: N, N2, NЗ/2 (N — емкость ЗУ, бит). Линейные алгоритмы, в том числе и «Диагональ», используются, как правило, для предварительной оценки ОЗУ на отсутствие катастрофических неисправностей.

Программа теста ОЗУ по алгоритму "Бегущая 1 (0)":

test proc

mov R1, T

mov R2, 0

mov R3, 1

mov R6, N

m1: mov R4[R2], R1

m2: cmp R3, R2

je m3

mov R5[R3], R1

m3: dec R6

cmp R3, R6

je m4

inc R3

jmp m2

m4: mov R3, 1

mov R7, R5[R3]

m5: cmp R7, R3

jz error

m6: dec R6

cmp R6, R3

je m7

inc R3

cmp R2, R3

jne m5

jmp m6

m7: mov R7, R5[R3]

cmp R7, R3

jz error

mov R4[R2], R1

dec R6

cmp R2,R6

je m8

m8: inc R2

mov R3, 0

jmp m1

ret

test endp

error: endp

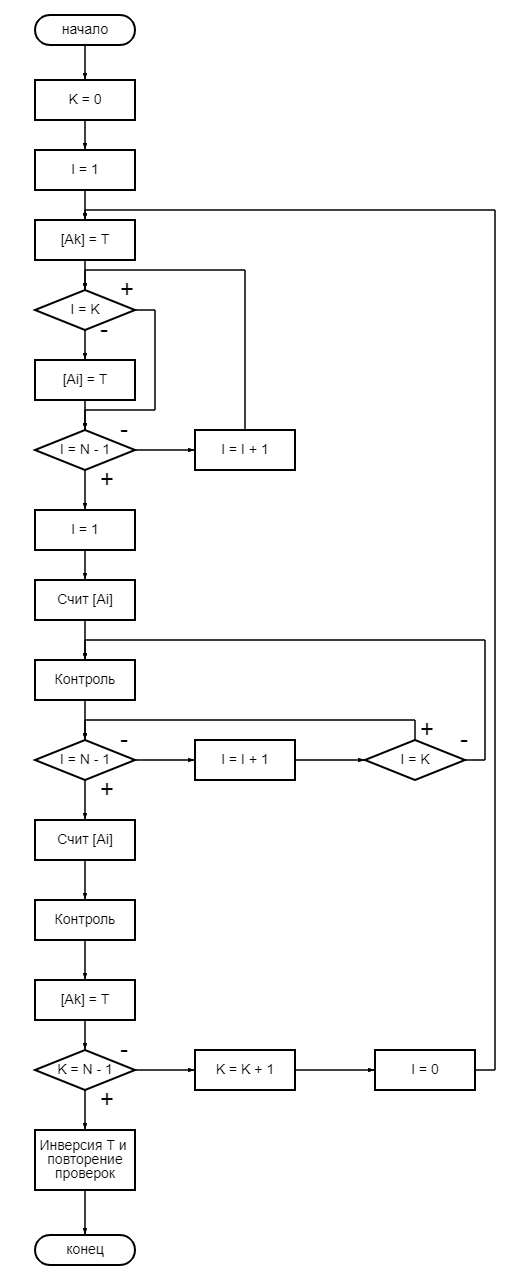


Рисунок 4.1 — Блок-схема алгоритма теста памяти «Бегущая 1 (0)»

**ЗАКЛЮЧЕНИЕ**

В ходе проектирования микроЭВМ предъявлялись требования, изложенные в задании.

Семейство микропрограммируемых устройств К1804 позволяет создавать системы с микропрограммным управлением на базе быстродействующих БИС. Эти устройства выполняют как операции условного перехода к подпрограмме, т. е. по адресу, извлекаемому из определенного источника, так и операции условного возврата из неё.

В данном курсовом проекте были рассмотрены основные принципы организации и функционирования микропроцессорных устройств с разрядно-модульной организацией. Был спроектирован спецкомпьютер на комплекте интегральных схем К1804. Данный компьютер отвечает стандартным нормам и содержит в себе основные блоки, которые должны быть у компьютеров данного класса.

В завершение можно отметить, что данная работа позволила более глубоко изучить эту тему, получить ценные знания и навыки в области разработки и проектирования специализированных компьютеров. Кроме того, курсовой проект позволил лучше понять важность компьютеров в современном мире в решении разного рода задача и их роль в совершенствовании различных процессов и технологий.

Был рассмотрен вопрос о том, что такое специализированные компьютеры и как они используются в различных отраслях науки и промышленности. Были выделены и проанализированы основные характеристики таких компьютеров, а также описан процесс их проектирования и изготовления.

**ЛИТЕРАТУРА**

1. Кобяк И. П. Спецкомпьютер с неймановской базовой архитектурой. Мн. БГУИР, 2013.
2. Кобяк И. П. Основы проектирования компьютерных устройств. Мн. БГУИР, 2008.
3. Баранов В. В., Бекин Н. В., Гордонов А. Ю. и др.; Под редакцией А. Ю. Гордонова и Ю. Н. Дьякова. Полупроводниковые БИС запоминающих устройств: Справочник – М.: Радио и связь, 1986.
4. Хвощ С. Т., Варлинский Н. Н., Попов Е. А.; Под редакцией Хвоща С. Т. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник – Л.: Машиностроение. Ленингр. отд-ние, 1987.
5. Кобяк И. П. Архитектура компьютерных систем: Методическое пособие в 2 ч – Мн.: БГУИР, 2001.

# **ПРИЛОЖЕНИЕ А**

*(обязательное)*

Схема структурная

# **ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Схема принципиальная

# **ПРИЛОЖЕНИЕ В**

*(обязательное)*

Ведомость документов