Министерство образования и науки, молодежи и спорта Украины

Севастопольский Национальный технический университет

Кафедра кибернетики

и вычислительной техники

Пояснительная записка

к курсовому проекту по дисциплине

" Цифровые ЭВМ " на тему:

**"Синтез центрального обрабатывающего устройства ЦВМ"**

Выполнил:

ст. гр. М – 41д

Сергиенко Н.Н.

Проверил:

проф. Апраксин Ю. К.

Севастополь

2011

**Содержание**

Введение…………………………………………………………………………...3

1. Постановка задачи……………………………………………………………….4
2. Описание форматов команд и обрабатываемых данных……………………...5
3. ГСА функционирования ЦОУ………………………………………………….9
4. Структурная схема ЦОУ……………………………………………………….13
5. Архитектура внешних выводов процессорного блока………………………16
6. Синтез управляющего устройства…………………………………………….17
   1. Структурная схема управляющего устройства…………………..17
   2. Адресация микрокоманд…………………………………………..18
   3. Кодирование микрокоманд………………………………………..18
   4. Микропрограмма функционирования ЦОУ……………………...20
   5. Разработка принципиальной схемы управляющего устройства..23
7. Определение временных характеристик и моделирование работы ЦОУ…..24

Заключение………………………………………………………….……………...26

Библиографический список……………………………………………….………27

Приложение………………………………………………………………………...28

**Введение**

Данная курсовая работа посвящена изучению принципов структурной и функциональной организации цифровых вычислительных машин и их узлов. Целью является практическое закрепление основных разделов дисциплины «Цифровые ЭВМ».

Объектом курсового проектирования является центральное обрабатывающее устройство (ЦОУ), реализующее заданную совокупность команд из системы команд абстрактной цифровой электронной вычислительной машины (ЦЭВМ). Устройство управления будет организовано в виде управляющего автомата с программируемой логикой. Управляющий автомат с программируемой логикой строится на основе принципа микропрограммного управления, использующего операционно-адресную структуру управляющих слов (микрокоманд).

Устройство управления будет синтезировано в соответствии с естественной адресацией микрокоманд. По разработанной ГСА будет составлена микропрограмма функционирования центрального обрабатывающего устройства.

1. **Постановка задачи**

Разработать центральное обрабатывающее устройство (ЦОУ), реализующее заданную совокупность команд из системы команд абстрактной ЦВМ. Предполагается, что проектированию подлежит процессор с традиционной принстонской архитектурой.

Необходимо реализовать следующий набор команд:

-арифметическая команда – сложение над операндами с плавающей точкой длиной в 8 байт (F8);

-логическая команда – двойной сдвиг вправо над операндом длиной в 8 байт (L4);

-команда передачи управления – условный переход по маске;

-команда обращения к памяти – загрузка в регистр из оперативной памяти целого числа длиной 4 байта (I4);

- команда обращения к устройству ввода/вывода – передача байта из МВВ в процессор.

Необходимо реализовать следующие способы адресации:

-непосредственная;

-прямая;

-косвенная;

-регистровая;

-косвенная через регистр;

Объем оперативной памяти – 512 Мб, ширина выборки – 4 байта. Объем регистровой памяти – 32 регистра, тип регистровой памяти – универсальная.

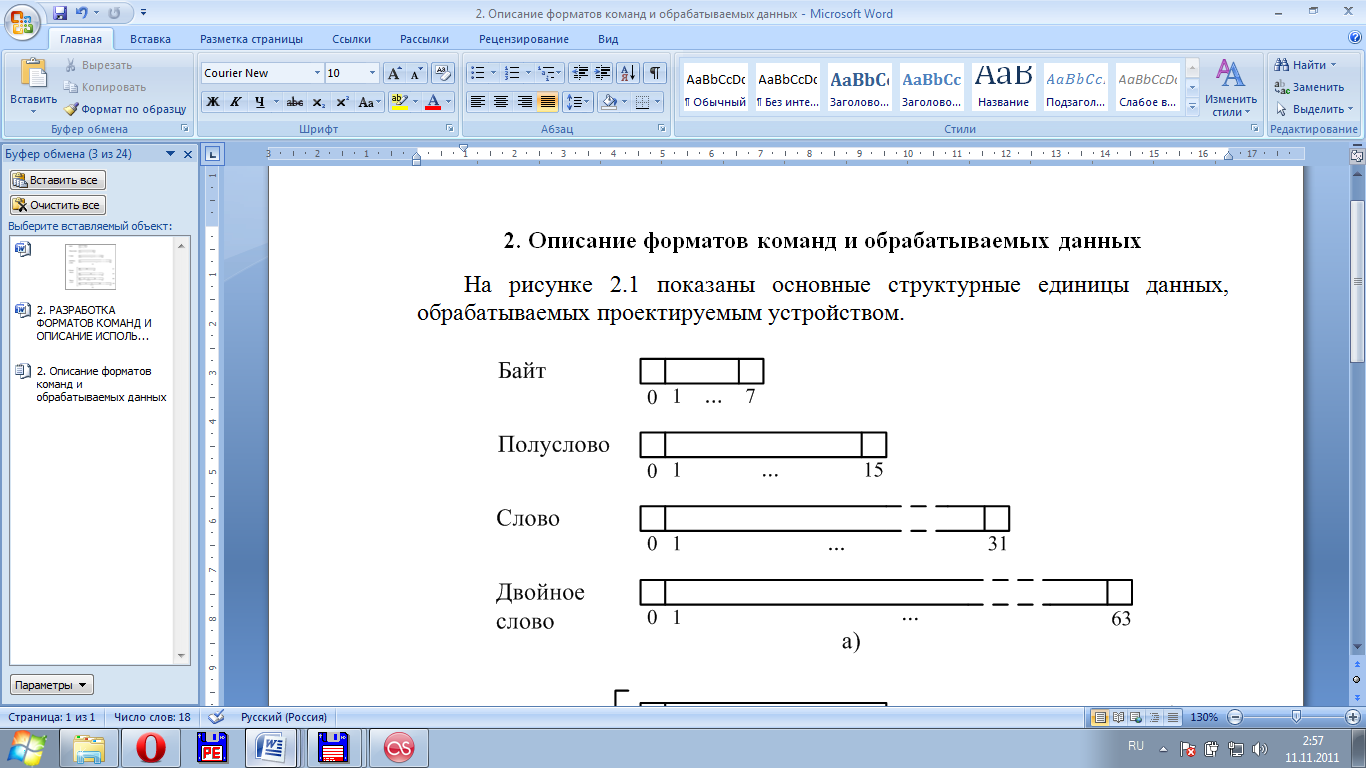
Необходимо реализовать устройство управления (УУ) в виде управляющего автомата (УА) с программируемой логикой, способ адресации микрокоманд – естественная, способ кодирования микроопераций – вертикально-горизонтальный.

В ходе курсового проекта необходимо разработать ГСА функционирования процессора, структурную схему ЦОУ, схему принципиальную электрическую УУ (с перечнем элементов).

Для реализации схемы должны использоваться микросхемы серии К155, ПЗУ – на элементах К155РЕ3.

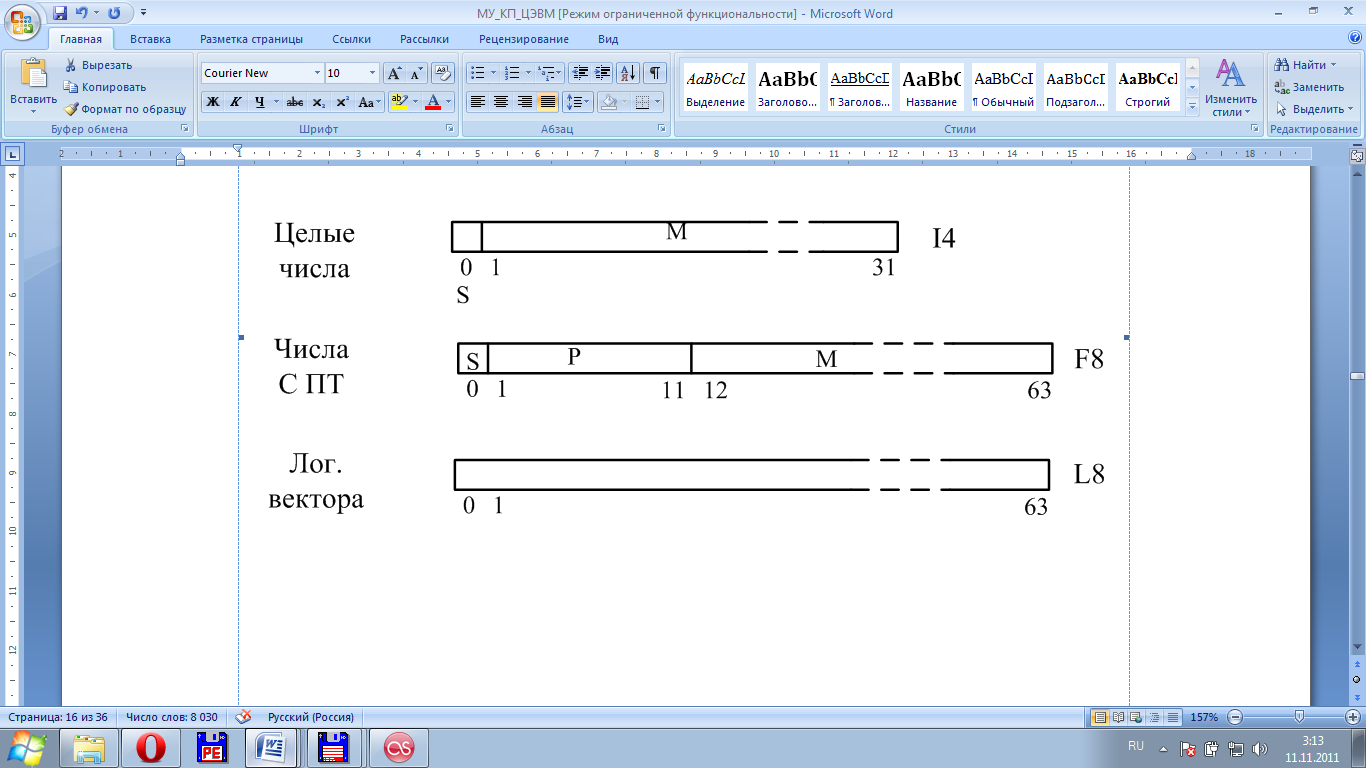
**2. Описание форматов команд и обрабатываемых данных**

На Рис.1 показаны основные структурные единицы данных, обрабатываемых проектируемым устройством.



*Рис. 1 – Основные структурные единицы данных*

Форматы представления данных в процессе обработки, используемые в командах процессора, представлены на Рис. 2.



*Рис. 2 – Форматы данных*

Целые числа могут быть представлены как со знаком (S), так и без знака (для представления адресов). Числа со знаком представляются в дополнительном коде. Диапазон представления целых чисел – [­2n, 2n­1], где n – количество разрядов числа без учета разряда знака.

Двоичные числа с плавающей точкой представляются в виде двух чисел с фиксированной точкой: порядка (Р) и мантиссы (М), при этом Р – целое со знаком, . В случае так называемой нормализованной мантиссы  для двоичной системы счисления. Диапазон представления чисел с плавающей точкой – .

Все форматы команд будут кратны 2 байтам (для облегчения процесса выборки команды из ОП). Адрес команды в ОП должен быть кратен 2 (согласно методу целочисленных границ*).* Под поле КОП в команде отводится 1 байт (принято, что система команд процессора состоит из 128-256 команд, а мы разрабатываем фрагмент).

**Класс 3 – Арифметические операции над числами с плавающей точкой**

Операция: Сложение чисел с плавающей точкой (+).

Формат данных: F8 (двойное слово).

Описание: Первый операнд складывается со вторым операндом, результат записывается на место первого операнда.

Формат команды: Первый операнд находится в одном из РПТ (регистров, предназначенных для хранения чисел в формате с плавающей точкой). Т.к. объем универсальной регистровой памяти – 32 регистра (16 РОН и 16 РПТ), то для кодировки первого операнда нужно 4 бита (24 = 16).

Второй операнд представлен регистром общего назначения (РОН), в котором содержится адрес памяти, по которому хранится число формата F8.

Объем памяти - 512 Мбайт (229 бит), разрядность шины выборки – 4 байта (22 бит), поэтому для адресации памяти необходимо 29-2 = 27бит, следовательно, 32-разрядного РОН достаточно с избытком для реализации косвенной адресации. Для кодировки РОН, как и для РПТ, необходимо 4 бита.



Таким образом, в данной команде реализована регистровая и косвенная через регистр адресации.

Признаки результата: 0 – мантисса результата = 0; 1 – результат < 0;

2 – результат > 0; 3 – переполнение порядка результата.

Прерывания: Спецификация (S); Переполнение порядка (ПП); Потеря значимости (ПЗ); Исчезновение порядка (ИП).

**Класс 5 – Команды сдвигов**

Операция: Двойной сдвиг вправо (2→).

Формат данных: L8 (двойное слово).

Описание: Первый операнд, представляющий слово двойной длины, сдвигается на число разрядов, которое указывает второй операнд.

Формат команды: Первый операнд – двойное слово из памяти, на которое указывает ячейка памяти из блока косвенных адресов, на которую в команде указывает укороченный адрес Ау. Под блок косвенных адресов в ОП отведено 1Кбайт памяти (1Кбайт = 1024байта=210байт), поэтому для адресации Ау в команде отведено 10 бит.

Второй операнд – константа сдвига, задается в команде непосредственно. Т.к. операция производится над 64-битными логическими векторами, то для задания константы сдвига нужно 6 бит (64=26).



Таким образом, в данной команде реализована косвенная и непосредственная адресации.

Признаки результата: Не формируются.

Прерывания: Спецификация (S).

**Класс 5 – Команды обращения к памяти**

Операция: Загрузка регистра.

Формат данных: I4 (слово).

Описание: Производится загрузка в регистр, на который указывает первый операнд, слова из памяти, на которое указывает второй операнд.

Формат команды: Первый операнд – РОН, в который происходит загрузка. Для его кодировки нужно 4 бита. Второй операнд – РОН, в котором находится адрес ОП, из которого происходит загрузка.



Таким образом, в данной команде реализована регистровая и косвенная через регистр адресации.

Признаки результата: Не формируются.

Прерывания: Спецификация (S).

**Класс 7 – Команды ввода-вывода**

Операция: Пересылка данных из порта ввода-вывода в процессор.

Описание: Передача байта из МВВ, номер (адрес) которого указан в команде, в процессор.

Формат команды: Один операнд, указывающий номер устройства ввода-вывода (0-255).



Таким образом, в данной команде реализована прямая адресация.

Признаки результата: Не формируются.

Прерывания: Нет.

**Класс 6 – Команды передачи управления**

Операция: Условный переход по маске.

Описание: Продвинутый адрес команды замещается адресом перехода, если значение признака результата соответствует коду, указанному в поле М1. В противном случае продолжается выполнение обычной последовательности команд с использованием продвинутого адреса.

Когда во всех четырех разрядах маски находятся единицы, происходит безусловный переход.

Если во всех четырех разрядах маски находятся нули или поле R2 (во втором формате) содержит нули, то команда эквивалентна отсутствию операции (переход не происходит).

Формат команды: Первый операнд - поле М1 используется в качестве 4-разрядной маски. Четыре разряда маски соответствуют слева направо четырем разрядам признака результата (0,1, 2, 3). Второй операнд, РОН – в котором хранится адрес условного перехода (тоже 4 бита).



|  |  |
| --- | --- |
| Значение РПр | Биты РК |
| 00 | 8 |
| 01 | 9 |
| 10 | 10 |
| 11 | 11 |

Таким образом, в данной команде реализована непосредственная и косвенная через регистр адресации.

Признаки результата: Не формируются.

Прерывания: Нет.

***Сведенная таблица системы команд ЦОУ:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Класс*** | ***№ком*** | ***Название и содержание*** | ***ПР****\** | ***Фл****\*\** | ***Код*** | |
| Bin**\*\*\***  01  234 567 | Hex |
| **1** | **0** | **Чтение из ОП в РП**  **РОН1:=[РОН2]** | - | **S** | **00  001 000** | **08h** |
| 1 | Чтение из РП в ОП  [РОН1]:=РОН2 | **-** | S | 00 001 001 | 09h |
| … | … | …. | … | … | … | … |
| **3** | **2** | **Сложение F8**  **РПТ:=РПТ+[РОН]** | **=0**  **<0**  **>0**  **ПП** | **S**  **ПП**  **ИП**  **ПЗ** | **00 011 010** | **1Bh** |
| 3 | Вычитание F8  РПТ1:=РПТ1-РПТ2 | 00 011 011 | 1Ch |
| … | … | … | … | … | … | … |
| **5** | **3** | **Двойной сдвиг вправо**  **(РОН0:1):= (РОН0:1)→с** | **-** | **S** | **01 101 011** | **6Bh** |
| 4 | Двойной сдвиг влево  (РОН0:1):= (РОН0:1)←с | 01 101 100 | 6Ch |
| **6** | **0** | **Переход по маске**  **СчАК:=(М≡РПр)?(R2):(СчАК+2)** | **-** | **-** | **00 110 000** | **30h** |
| 1 | Переход по маске  СчАК:=(М≡РПр)?((В2)+D2):(СчАК+2) | 01 110 001 | 71h |
| **7** | **0** | **Ввод данных с порта**  **ПортD:=(УВВ)** | **-** | **-** | **00 111 000** | **38h** |
| 1 | Вывод данных на порт  (УВВ):=ПортD | 00 111 001 | 39h |
| **0** | **0** | **Cтоп (команда останова)** | **-** | **-** | **00  000 000** | **00h** |

\**Признаки результата: =0 – мантисса равна 0; >0 – результат больше нуля; <0 – результат меньше нуля; ПП – переполнение порядка.*

*\*\*Флаги: S – нарушение спецификации; ПП – переполнение порядка; ИП – исчезновение порядка; ПЗ – потеря значимости.*

\*\*\**При кодировании: КОП( 0:1) – код длины команды (00-2байта, 01 - 4), КОП(2:4) – номер класса; КОП(5:7) – номер команды.*

**3. ГСА функционирования ЦОУ**

ГСА функционирования ЦОУ приведена на чертеже 2011.М41д.21.01.

Разработке подлежат алгоритмы выполнения пяти операций, определенных вариантом задания, и алгоритм выборки команды из ОП. Взаимосвязь указанных алгоритмов отражена на упрощенной граф-схеме алгоритма функционирования ЦОУ (рисунок 3.1). Если код выбранной на РК команды не совпадает с кодом ни одной из реализованных команд, устанавливается флаг прерывания K (резервная команда).



*Рис 3.1 – Схема алгоритма функционирования ЦОУ (здесь ЗР – команда загрузки регистра; МП – команда передачи байта из МВВ в процессор; СлПТ – команда сложения с плавающей точкой; СдвВпр – команда двойного сдвига вправо; УПМ – команда условного перехода по маске)*

Опишем подробней алгоритмы, входящие в состав общей ГСА:

1. **Алгоритм выборки команды** *(на чертеже представлен блоками 1-15):*

1. Если установлен ТП, следовательно, была выполнена команда ПУ и значение СчАК может быть любым, поэтому проверяется, является ли адрес кратным 2 (однобайтных команд быть не может). Если указан адрес одного байта – нарушение спецификации. Иначе, читаем 4 байта из ОП.

1.1. Если был указан адрес 4 байт, проверяем, является ли команда 4 байтной (проверяются первые два бита Порта Д, а следовательно первые два бита КОП). Если команда 4-байтная, загружаем 4 байта в РК и прибавляем к СчАК 4 (Адрес в СчАК останется кратным 4), иначе загружаем в РК 2 байта, прибавляем к СчАК 2 и адрес в СчАК не будет кратен 4, поэтому запоминаем младшие 2 байта в БР. Сбрасываем ТП и переходим к выполнению команды.

1.2. Если был указан адрес 2 байт (в СчАК после команды ПУ) прибавляем к СчАК 2 (в СчАК теперь будет адрес кратный 4), сбрасываем ТП и проверяем, является ли команда двухбайтной(проверяются первые два бита КОП, а следовательно первые два бита РК). Если команда 2 байтная, переходим к выполнению команды, иначе читаем из ОП данные. Так как старшие 2 байта команды у нас уже хранятся в РК, то из Порта Д выбираются старшие 2 байта и помещаются в младшие 2 байта РК, а младшие 2 байта Порта Д сохраняются в буферный регистр. СчАК увеличивается на 2, а следовательно перестает быть кратным 4.

2. Если ТП не был установлен, то в СчАК хранится адрес либо 2 либо 4 байт.

2.1. Если в СчАК адрес 2 байт, то старшая часть РК хранится в БР. Сохраняем в старшие 2 байта РК содержимое БР и увеличиваем СчАК на 2( он становится кратным 4). Дальше выполняем те же действия, что и в пункте 2 б) начиная с проверки длины команды.

2.2. Если же адрес в СчАК кратен 4, то выполняем действия, указанные в пункте 1, без проверки на нарушение спецификации.

3. Конец

1. **Алгоритм команды чтения из ОП в РП** *(на чертеже представлен блоками 17-21):*
2. Выполняем чтение адреса ОП, заданного косвенно через РОН: для этого загружаем в РАРП адрес РОН (в «младшей» части универсальной РП – первый бит адреса – 0) и выполняем чтение из регистровой памяти.
3. Если считанный адрес ОП кратен 4, то переход к п.3, иначе прерывание «Спецификация».
4. Загружаем полученный адрес в Порт А и выполняем операцию чтения ОП (по ее окончании – слово из ОП загрузится в Порт Д).
5. Выполняем запись слова ОП в РОН, указанный в первом операнде: в РАРП загружаем адрес РОН, в РСРП – слово из Порта Д и выполняем операцию записи РП.
6. Конец.
7. **Алгоритм команды пересылки данных из порта ввода-вывода в процессор** *(на чертеже представлен блоками 23-24):*
8. Загружаем адрес (номер) устройства ввода-вывода, указанный в коде команды, в порт А.
9. Инициируем операцию пересылки данных из порта ввода-вывода в процессор (операция длительная, поэтому нужно дождаться окончания ее выполнения).
10. Конец.
11. **Алгоритм команды сложения чисел с плавающей точкой** *(на чертеже представлен блоками 26-38)***:**
12. Если адрес РПТ (первый операнд) кратен 2 (64-разрядные числа с плавающей точкой должны храниться в паре РПТ), то переход к п.2, иначе – прерывание «Спецификация».
13. Считываем адрес второго операнда из ОП, который косвенно задан в РОН: в РАРП загружаем адрес РОН и инициируем операцию чтения РП.
14. Если считанный адрес кратен 8 (т.к. 64-разрядное число), то переход к п.4., иначе - прерывание «Спецификация».
15. Выполняем чтение младшей части второго операнда из ОП: загружаем в Порт А адрес, считанный из РОН, и выполняем операцию чтения ОП.
16. В младшую часть Регистра 2 АЛУ загружаем младшую часть второго операнда. Выполняем чтение младшей части первого операнда из первого РПТ регистровой пары. Инкрементируем адрес на Порте А и выполняем операцию чтения ОП (чтение старшей части второго операнда).
17. В младшую часть Регистра 1 АЛУ загружаем младшую часть первого операнда. В старшую часть Регистра 2 АЛУ загружаем старшую часть второго операнда. Выполняем чтение старшей части первого операнда из второго РПТ регистровой пары.
18. В старшую часть Регистра 1 АЛУ загружаем старшую часть первого операнда. Инициируем операцию АЛУ сложения чисел с плавающей точкой.
19. Выполняем запись старшей части результата (Р1АЛУ) по текущему адресу РП (второй РПТ пары). В регистр признака результата загружаем кодированное значение признаков с шины АЛУ. В регистр флагов загружаем значение флагов с шины АЛУ.
20. Выполняем запись младшей части результата (Р1АЛУ) по предварительно уменьшенному на 1 адресу РП (первый РПТ пары).
21. Конец.
22. **Алгоритм команды двойного сдвига вправо** *(на чертеже представлен блоками 40-53)***:**
23. Если укороченный адрес, указанный в коде команды, кратен 4, то переход к п.2, иначе – прерывание «Спецификация».
24. В Порт А загружаем укороченный адрес в младшей части памяти (старшие разряды заполняем 0-ми) и инициируем операцию чтения ОП – получаем адрес первого операнда в ОП.
25. Если адрес кратен 8 (т.к. 64-битный вектор), переход к п.4, иначе – прерывание «Спецификация».
26. Выполняем загрузку считанного из ОП косвенного адреса в Порт А и инициируем операцию чтения ОП. В Регистр 2 АЛУ загружаем второй операнд (константу сдвига), непосредственно заданный в коде команды.
27. В Регистр 1 АЛУ загружаем младшую часть первого операнда, полученную с Порта Д. Инкрементируем значение адреса на Порте А и выполняем чтение ОП – чтение старшей части первого операнда.
28. Догружаем старшую часть первого операнда в Регистр 1 АЛУ и инициируем операцию двойного сдвига вправо.
29. В два приема, подобно чтению первого операнда из ОП, выполняем запись результата в ОП.
30. Конец.
31. **Алгоритм команды условного перехода по маске** *(на чертеже представлен блоками 55-60)***:**
32. Если маска содержит все нули, то переход к п.6.
33. Если адрес РОН, косвенно задающего адрес перехода, содержит все нули, то переход к п.6.
34. Если маска содержит все единицы, то переход к п.5.
35. Если значение регистра признака не соответствует маске (признак в регистре закодирован, поэтому требуется дешифрирования), то переход к п.6.
36. Считать и занести в СчАК адрес перехода, указанный в РОН. Установить ТП.
37. Конец.

**4. Структурная схема ЦОУ**

Структурная схема ЦОУ представлена на чертеже 2011.М41д.21.02.

На структурной схеме представлены следующие основные элементы:

1. **Устройство управления** (**УУ**) ЦОУ - это блок, обеспечивающий выработку необходимых последовательностей управляющих сигналов {y:: y1…y46}, причем, генерируемые управляющим автоматом последовательности управляющих сигналов предопределяются поступающими на вход этого устройства осведомительными сигналами {x:: x1…x24} из операционного устройства, несущими информацию об особенностях реализации каждой микрооперации.
2. **Арифметико-логическое устройство (АЛУ)** – главный элемент процессорного блока. Двуместные операции в АЛУ выполняются над операндами, хранящимися в регистрах Р1(0:63) (первый операнд) и Р2(0:63) (второй операнд) – их разрядность выбрана таким образом, чтобы обеспечить выполнение операций над заданными форматами данных. Результат выполненной операции помещается в Р1.

Каждая операция АЛУ инициируется своим управляющим сигналом, поступающим на вход устройства управления (УУ): СлПТ – операция сложения 64-битных чисел с плавающей точкой, СдвВпр – операция двойного сдвига вправо.

АЛУ вырабатывает три группы осведомительных сигналов: признаки результата (=0 – мантисса результата равна 0; <0 – отрицательный результат; >0 – положительный результат; ПП – переполнение порядка), флаги прерываний (ПП – переполнение порядка; ИП – исчезновение порядка; ПЗ – потеря значимости) и признак занятости (ZАЛУ).

1. **Регистр признака результата (РПр)** – хранит снятый с 4 разрядов шины АЛУ (**ШАЛУ**) и закодированный с помощью шифратора **CD** 2-битный код признака результата арифметической операции. При выполнении команды условного перехода по маске признак результата с РПр сравнивается с маской на схеме сравнения (**СС**), которая вырабатывает осведомительный сигнал при их соответствии.
2. **Регистр флагов (РФ) –** представляет собой регистр, состоящий из 5 триггеров-флагов. Содержит следующие флаги: **ПП** – флаг переполнения порядка; **ИП** – флаг исчезновения порядка; **ПЗ** – флаг потери значимости (эти флаги устанавливаются при прерываниях АЛУ); **S** – флаг прерывания «Нарушение спецификации»; **K** – флаг прерывания «Неизвестная команда».
3. **Оперативная память (ОП)** хранит программы и данные. Ее объем составляет 512 Мб. Первые 1024 байта (1Кбайт) выделены под блок косвенных адресов.

Разрядность шины выборки (ШВ) равна 4 байта – разрядность слова ОП. Отсюда следует, что разрядность регистра слова ОП (РСОП) – 4 байта, а разрядность регистра адреса ОП (РАОП) – 29-2 = 27 бит. Обращение к РСОП осуществляется через Порт Д, а к РАОП – через Порт А.

На вход ОП поступают сигналы, инициирующие следующие операции: чтение из ОП (ЧтОП), запись в ОП (ЗпОП). ОП вырабатывает один осведомительный сигнал – признак занятости (ZОП).

1. **Модуль ввода-вывода (МВВ)** служит для обмена данными между процессором и внешними устройствами. Организация ввода-вывода осуществляется по принципу программного управления, т.е. по инициативе процессора и под его полным контролем. Для задания номера устройства ввода-вывода (**НУВВ**) используется специальный 8-разрядный регистр, что позволяет обращаться к одному из 256 устройств. Ввод-вывод данных для внешних устройств осуществляется с помощью 32-битного регистра данных (**Д**). Обращение к Д осуществляется через Порт Д, а к НУВВ – через Порт А.

На МВВ подается управляющий сигнал чтения (Чт). МВВ формирует осведомительный сигнал занятости - (Zвв).

1. **Порт А** – порт адреса процессора, предназначен для хранения адреса ОП и номера устройства МВВ при обмене данными. Имеет разрядность 27 бит, что позволяет адресовать 227 слов ОП. Реализован в виде реверсивного счетчика, что позволяет легко выполнять операции инкрементирования и декрементирования адреса при считывании-записи данных, превышающих разрядность ШВ (в соседних адресах ОП).
2. **Порт Д** – порт данных процессора. Служит для обмена данными между процессором и ОП, МУВВ. Имеет разрядность, равную разрядности ШВ – 32 бита.
3. **Регистровая память (РП)** организуется как сверхоперативная память с прямой адресацией. РП допускает выборку слова в течение одного такта.

Объем РП – 32 регистра. Тип РП – универсальная: один блок как для регистров общего назначения (РОН), так и для регистров, предназначенных для хранения чисел в формате с плавающей точкой РПТ. Половина (16 регистров) РП отводится под РПОН, вторая – под РППТ. Отсюда следует, что разрядность регистра адреса РП (**РАРП**) должна быть ⎡log2(32)⎤ = 5 бит, при чем для адресации РПОН в старшем разряде указывается 0, РППТ – 1. Разрядность регистра слова РП (**РСРП**), как и разрядность одного регистра, определяется разрядностью машинного слова, т.е. равна 4 байта.

На входы РП поступают сигналы, инициирующие следующие операции: запись в РП (ЗпРП), чтение из РП (ЧтРП).

1. **Регистр команд (РК)** – хранит текущую команду в соответствии с разработанными форматами команд. На РК выбирается из ОП команда, подлежащая исполнению. Разрядность РК определяется максимальной длиной формата команды – 32 разряда.

Тип операции определяется ее кодом (КОП) - первым байтом регистра в соответствии с разработанными форматами команд при декодировании на дешифраторе **DC**.

1. **Буферный регистр (БР)** вводится в схему с целью уменьшения числа обращений к ОП (долгосрочная операция) в процессе выборки команды. Имеет разрядность 16 бит – разница между разрядностью выборки и минимальной длиной команды.
2. **Счетчик адреса команды (СчАК)** - предназначен для хранения адреса очередной выбираемой из ОП команды. Обеспечивает адресацию к любому байту ОП. Его разрядность определяется как ⎡log2EОП⎤ = ⎡log2512⎤ = 29 бит. При этом, младшие k = ⎡log2ШВ⎤ = ⎡log24⎤ =2 бита определяют адрес байта в слове, а старшие 27 разрядов – адрес слова ОП.
3. **Триггер перехода (ТП)** устанавливается в 1 в случае формирования адреса перехода при исполнении команды передачи управления и сбрасывается в нулевое состояние, если исполняемая команда относится к другой группе. Учет состояния ТП требуется при выборке команд из ОП.

Помимо основных блоков на структурной схеме ЦОУ показаны микрооперации обмена данными между ними, установки и сброса определенных разрядов в них и т.д. Все микрооперации инициируются управляющими сигналами *{y}* (y1-y45). Также на схеме показаны сформированные на блоках осведомительные сигналы *{x}* (x1-x24). Список управляющих и осведомительных сигналов приведен в приложении А.

**5. Архитектура внешних выводов процессорного блока**

Количество функциональных выводов процессорного блока (рис. 5.1) – 68 (можно без избытка использовать стандартный 68-выводной корпус).

Процессор имеет 27 внешних выводов адреса (Аi) и 32 внешних выводов данных (Di). Выводы R/~WR, IN, OUT используются соответственно для сигналов «чтение» (ЧтОП), «запись» (ЗпОП), «ввод» (МП) и «вывод» (ПМ). Вывод +5V используется для питания, GND – земля, RESET – сброс, START – запуск, STOP – остановка, CLC – синхронизация.



*Рис. 5.1 –УГО процессорного блока*

***Нумерация контактов схемы:***

|  |  |
| --- | --- |
| 1:27 – адресные выводы;  28:59 – выводы данных;  60 – питание;  61 – земля;  62 – «ввод»;  63 – «вывод»; | 64 – запуск;  65 – остановка;  66 – сброс;  67 – синхронизация;  68 – чтение- не запись. |

**6. Синтез управляющего устройства**

***6.1. Структурная схема управляющего устройства***

Обобщенная структура УА с хранимой в памяти логикой изображена на рисунке 6.1.



*Рис. 6.1 – Обобщенная структурная схема УУ ЦОУ*

Память микропрограмм (ПМП) организована в виде ПЗУ. В качестве РАМК используется счетчик. Пуск автомата осуществляется подачей в схему управляющего сигнала **В**, разрешающего подачу тактирующих сигналов, останов – подачей управляющего сигнала ***Сброс*** или выработке управляющего сигнала конца ***yk***. Управляющий сигнал ЧтМК выбирает из ПМП на регистр МК (РМК) очередную МК. Схема формирователя сигналов МО (ФСМО) расшифровывает поле МО и вырабатывает управляющие сигналы, инициирующие выполнение процессором конкретных МК. Поле логического условия (ЛУ) подается на адресные входы мультиплексора (MS). MS выбирает по адресу ЛУ соответствующий осведомительный сигнал (x1, x2,…,x19), которым выбирается адрес следующей МК (либо продвинутый адрес, либо адрес перехода - поле адреса РМК записывается в СчАМК).

***6.2. Адресация микрокоманд***

Использование принципа естественной адресации МК предполагает использование двух различных форматов МК: **В.М** – для операционных МК и   **В.Х.А** – для управляющих МК. Здесь В, М, Х, А – поля микрокоманд: В – одноразрядное поле бит-маркера (В=0 для операционных МК и В=1 для управляющих МК), М – поле для представления кода МО, включенных в МК, Х – поле кода (номера) проверяемого логического условия, А – поле адреса МК, исполнение которой осуществляется в случае истинности проверяемого логического условия.

Исполнительный адрес МК вычисляется по следующему правилу:



Здесь *i* – индекс осведомительного сигнала *xi*, значение которого равно 1.

Адресация МК предполагает расстановку адресов, начиная с базового адреса А+0, для каждой вершины ГСА по пути невыполнения проверяемых логических условий. Правило вычисления каждого следующего адреса – инкрементирование предыдущего (естественная адресация). Если очередная дугаприводит к вершине с уже проставленным адресом, то на этой дуге отмечается дополнительная МК безусловного перехода. Адреса МК показаны на ГСА функционирования ЦОУ (чертеж 2011.М41д.21.01).

***6.3. Кодирование микрокоманд***

Способ кодирования микрокоманд, заданный по варианту, – вертикально-горизонтальное. При использовании этого принципа кодирования МК необходимо выделить подмножества совместимых МО и обеспечить при этом несовместимость сформированных подмножеств (ни одна МО одного подмножества не должна быть совместима с какой-либо МО любого другого подмножества).

Определим отношения совместимости МО:

1-2

2-1,5,6,7,8,11

3-4

4-3,14,13,20,21,22,32,33,34,35

5-2,6,7

6-2,5,7,11

7-2,5,6,8,9,10,11

8-2,7

9-7,10

10-7,9

11-2,6,7

12-13

13-12,23,24,25,20,21,22

14-4

15-16,17

16-15,17

17-15,16,28,29,30,31,21

18-19

19-18

20-21,22,13,4

21-4,13,20,22,17,31

22-4,13,20,21,35

23-13,24,25

24-13,23,25

25-13,23,24

26-27

27-26

28-17,29,30

29-17,28,30

30-17,28,29

31-17,21

32-4

33-4,34

34-4,33

35-4,22

36-37

37-36

38-39

39-38,40,41

40-39,41

41-39,40

42-43

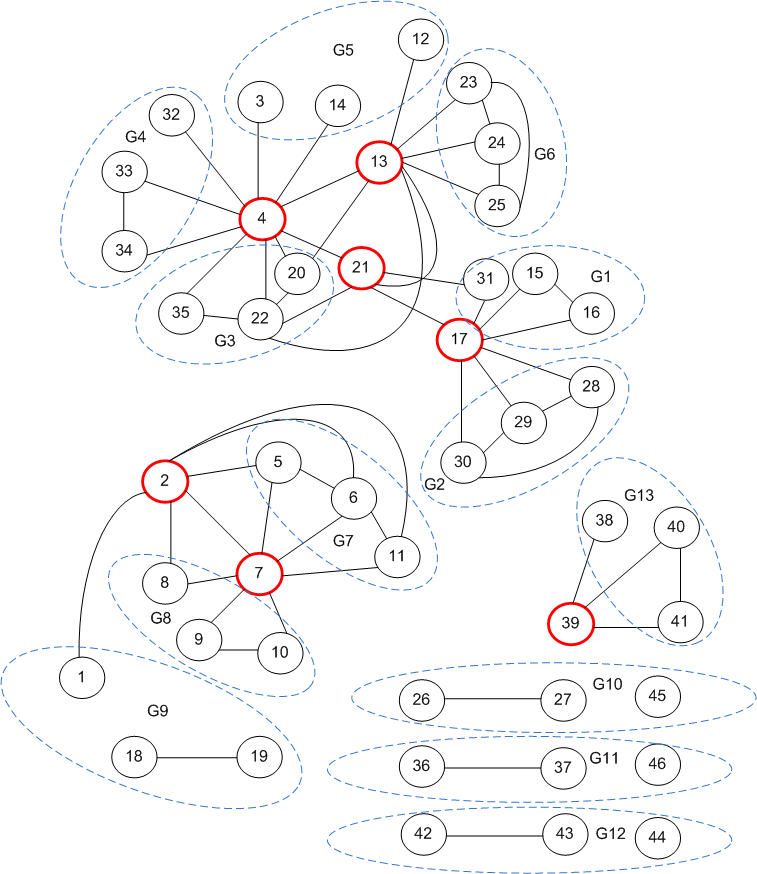
43-42

44-

45-

46-

Для более наглядного представления отношений совместимости МО построим по полученным отношениям совместимости построим граф, в котором наиболее связанные вершины выделим в универсальную группу для получения несовместимых подмножеств совместимых МО, при чем подмножества должны иметь одинаковую (или примерно одинаковую) мощность (Рис. 6.2).



*Рис. 6.2 – Граф совместимости МО*

Как видно из рисунка, после определения МО *y2, y4, y7, y13, y17, y21, y39* в универсальную группу (УГ) были получены следующие 13 несовместимых подмножеств совместимых МО:

*G1 = {y15, y16, y31};*

*G2 = {y28, y29, y30};*

*G3 = {y20, y22, y35};*

*G4 = {y32, y33, y34};*

*G5 = {y3, y12, y14};*

*G6 = {y23, y24, y25};*

*G7 = {y5, y6, y11};*

*G8 = {y8, y9, y10};*

*G9 = {y1, y18, y19};*

*G10 = {y26, y27, y45};*

*G11 = {y36, y37, y46};*

*G12 = {y42, y43, y44};*

*G13 = {y38, y40, y41}.*

При использовании вертикально-горизонтального кодирования МО используются три поля: М1 – горизонтальный код УГ – его разрядность равна мощности УГ, т.е. 7; М2 – вертикальный код подмножества (позиционный код его номера) – его разрядность равна m2 = ⎡log2{Gi}⎤ = ⎡log213⎤ =4; М3 – горизонтальный код МО в подмножествах – его разрядность равна максимальной мощности подмножества совместимых МО, т.е. 3 (во всех подмножествах одинакова).

Представим таблицы, по которым будет производиться кодирование МО:

*Универсальная группа:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № разряда | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| МО УГ | 2 | 4 | 7 | 13 | 17 | 21 | 39 |

*Несовместимые подмножества:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Подмн-во | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Код | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 |

*МО в подмножествах будут кодироваться по своему «месту» в векторе.*

***6.4. Микропрограмма функционирования ЦОУ***

Микропрограмму в символьных обозначениях и кодах представим в виде следующей таблицы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес МК | Код адреса | МК | | | Кодированная МК | | | | | | | | | | | | | | |
| B | X | A | B | X | | | | | А | | | | | | | - | - |
| MO | | УГ | | | | | | | №п-мн. | | | | №МО | | |
| A+0 | 0000000 | 1 | x1 | A+15 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | - | - |
| A+1 | 0000001 | 1 | x2 | A+19 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | - | - |
| A+2 | 0000010 | 0 | y3,y4 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| A+3 | 0000011 | 1 | x4 | A+3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | - | - |
| A+4 | 0000100 | 1 | x2 | A+32 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+5 | 0000101 | 1 | x6 | A+34 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | - | - |
| A+6 | 0000110 | 0 | y7,y9,y10 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| A+7 | 0000111 | 1 | x7 | A+25 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | - | - |
| A+8 | 0001000 | 1 | x9 | A+36 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | - | - |
| A+9 | 0001001 | 1 | x11 | A+39 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | - | - |
| A+10 | 0001010 | 1 | x16 | A+54 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | - | - |
| A+11 | 0001011 | 1 | x19 | A+71 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | - | - |
| A+12 | 0001100 | 1 | x24 | A+14 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | - | - |
| A+13 | 0001101 | 0 | y45 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| A+14 | 0001110 | 0 | y46 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| A+15 | 0001111 | 1 | x5 | A+17 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+16 | 0010000 | 1 | xбп | А+2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | - | - |
| A+17 | 0010001 | 0 | y44 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| A+18 | 0010010 | 1 | xбп | А+14 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | - | - |
| A+19 | 0010011 | 0 | y1,y2 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| A+20 | 0010100 | 1 | x3 | A+7 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | - | - |
| A+21 | 0010101 | 0 | y3,y4 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| A+22 | 0010110 | 1 | x4 | A+22 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | - | - |
| A+23 | 0010111 | 0 | y2,y5,y6,y7 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| A+24 | 0011000 | 1 | xбп | A+7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | - | - |
| A+25 | 0011001 | 0 | y12,y13 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| A+26 | 0011010 | 1 | x8 | A+28 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | - | - |
| A+27 | 0011011 | 1 | xбп | A+17 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+28 | 0011100 | 0 | y4,y14 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| A+29 | 0011101 | 1 | x4 | A+29 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | - | - |
| A+30 | 0011110 | 0 | y15,y16,y17 | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| A+31 | 0011111 | 1 | xбп | A+0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+32 | 0100000 | 0 | y2,y7,y8 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| A+33 | 0100001 | 1 | xбп | A+20 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | - | - |
| A+34 | 0100010 | 0 | y2,y6,y7,y11 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| A+35 | 0100011 | 1 | xбп | A+7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | - | - |
| A+36 | 0100100 | 0 | y18,y19 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| A+37 | 0100101 | 1 | x10 | A+37 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | - | - |
| A+38 | 0100110 | 1 | xбп | A+0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+39 | 0100111 | 1 | x12 | A+42 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | - | - |
| A+40 | 0101000 | 0 | y12,y13 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| A+41 | 0101001 | 1 | x13 | A+43 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | - | - |
| A+42 | 0101010 | 1 | xбп | A+17 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+43 | 0101011 | 0 | y4,y14 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| A+44 | 0101100 | 1 | x4 | A+44 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | - | - |
| A+45 | 0101101 | 0 | y4,y13,y20,y21,y22 | | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| A+46 | 0101110 | 1 | x4 | A+46 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | - | - |
| A+47 | 0101111 | 0 | y13,y23,y24,y25 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| A+48 | 0110000 | 0 | y26,y27 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| A+49 | 0110001 | 1 | x14 | A+49 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+50 | 0110010 | 0 | y17,y28,y29,y30 | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A+51 | 0110011 | 0 | y17,y21,y31 | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| A+52 | 0110100 | 1 | x15 | A+0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+53 | 0110101 | 1 | xбп | A+14 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | - | - |
| A+54 | 0110110 | 0 | x17 | A+56 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | - | - |
| A+55 | 0110111 | 1 | xбп | A+17 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+56 | 0111000 | 0 | y4,y32 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| A+57 | 0111001 | 1 | x4 | A+57 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | - | - |
| A+58 | 0111010 | 1 | x18 | A+60 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | - | - |
| A+59 | 0111011 | 1 | xбп | A+17 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | - |
| A+60 | 0111100 | 0 | y4,y33,y34 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| A+61 | 0111101 | 1 | x4 | A+61 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | - | - |
| A+62 | 0111110 | 0 | y4,y22,y35 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| A+63 | 0111111 | 1 | x4 | A+63 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | - | - |
| A+64 | 1000000 | 0 | y36,y37 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| A+65 | 1000001 | 1 | x14 | A+65 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | - | - |
| A+66 | 1000010 | 0 | y38,y39 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| A+67 | 1000011 | 1 | x4 | A+67 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | - | - |
| A+68 | 1000100 | 0 | y39,y40,y41 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| A+69 | 1000101 | 1 | x4 | A+69 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | - | - |
| A+70 | 1000110 | 1 | xбп | A+0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+71 | 1000111 | 1 | x20 | A+75 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | - | - |
| A+72 | 1001000 | 1 | x21 | A+75 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | - | - |
| A+73 | 1001001 | 1 | x22 | A+76 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | - | - |
| A+74 | 1001010 | 1 | x23 | A+76 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | - | - |
| A+75 | 1001011 | 1 | xбп | A+0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| A+76 | 1001100 | 0 | y12,y13 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| A+77 | 1001101 | 0 | y42,y43 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| A+78 | 1001110 | 1 | xбп | A+0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |

***6.5. Разработка принципиальной схемы управляющего устройства***

Память МП имеет размерность 79х15. Для реализации ПМП (MPM) была взята ИМС К155РЕ3. Микросхема представляет собой электрически программируемое посредством пережигания плавких перемычек постоянное запоминающее устройство (ППЗУ) емкостью 256 бит (32x8). Для реализации ПМП понадобится 3 линейки по 2 микросхемы К155РЕ3 (DD1-DD6). При этом старший бит каждого слова ПМП и старшие 17 слов последней линейки не используются и могут иметь произвольные значения.

На адресные входы всех микросхем подаются разряды 6...2 с регистра адреса микрокоманд (РАМК, MIA) – адресация внутри линейки. Старшие адресные разряды (MIA(1:0)) подаются на дешифратор, реализованный в виде комбинационной схемы с помощью логических элементов 2И-НЕ микросхем К155ЛА3 (DD17-DD18). Выходы дешифратора активируют одну из трех линеек ПМП.

Регистр микрокоманд имеет разрядность 15 бит. Для реализации его в схеме электрической принципиальной использовались 2 регистра К155ИР13 (DD10-DD11). В исходном состоянии по всем адресам и разрядам записан логический ноль. По стробу СС1 данные с памяти микропрограмм поступают на входы регистра микрокоманд (РМК, MIR).

Регистр адреса микрокоманд имеет разрядность 7 бит и для реализации его в схеме используется два 4-разрядных счетчика К155ИЕ7 (DD12-DD13), при чем первый тактируется сигналом СС3, а второй – выходом прямого переноса первого. По стробу СС3 в РАМК подаются данные из регистра микрокоманд (РМК[12:6]) или значение адреса в РАМК увеличивается на 1, в зависимости от информационного бита-маркера (РАМК[0]).

В качестве формирователя разрешающего сигнала *Хх* использовались два мультиплексора К155КП1 (DD8, DD9). В качестве старшего адресного бита используется бит РМК(5), который подается в прямом и инверсном виде на стробирующие входы мультиплексоров. Сигналы РМК(1:4) подаются на адресные входы обеих микросхем. Для логического объединения инверсных выходов мультиплексоров для формирования разрешающего сигнала *Хх* загрузки адреса в РАМК используется логический элемент 2И-НЕ одной из микросхем К155ЛА3.

В качестве формирователя сигналов микроопераций использовались 10 микросхем К155ЛЕ1 (DD21-DD30) (для формирования сигналов подмножеств), 2 микросхемы К155ЛИ1 (DD19-DD20) (для формирования сигналов МО УГ) и дешифратор К155ИД3 (DD7) (для декодирования номера подмножества, разрешающий сигнал – с участием тактового сигнала).

В схеме предусмотрен разъем, состоящий из двух разъемов типа РППМ-26 на 50 и 28 выходов. Вблизи разъема предусмотрел емкостной фильтр из расчета 0.1мкф \* 32 ≈3.3мкФ (стандартный номинал). Емкостные фильтры на четыре группы ИМС реализованы в виде четырех конденсаторов на 0.015мкФ (≈(32/4)\*0.002 мкФ).

Схема электрическая принципиальная УУ с перечнем элементов представлена на чертеже 2011.М41д.21.03.

**7. Определение временных характеристик**

**и моделирование работы ЦОУ**

Временные характеристики работы ЦОУ рассчитываются, исходя из алгоритма функционирования ЦОУ (алгоритма исполнения МК) и схемотехнических особенностей реальных схем, составляющих процессор.

Промежуток времени, достаточный для реализации процессором любой МК, называется тактом. Другими словами, такт – это период синхросерии, обеспечивающей стабильную работу операционного устройства (ОУ). Так как любое ОУ, в том числе и проектируемое ЦОУ, состоит из управляющего устройства (управляющего автомата) и обрабатывающего блока (операционного автомата), такт операционного устройства в случае последовательной работы УУ и ОБ определяется по формуле:

,

где − время срабатывания управляющего устройства,

− время срабатывания обрабатывающего блока, определяемое по времени исполнения самой длительной МО [1]. Расчетное значение тактовой частоты определяется величиной F=1/TОУ.

Временная диаграмма работы ОУ с учетом реальных задержек в схемах устройства строится в соответствии со следующими временными соотношениями:

, где  − максимальное время выборки слова из ПМП;

нс

, где  − время, необходимое для записи слова в РМК;

нс

, где  − время срабатывания ФСМО (схема этого устройства и время срабатывания зависит от используемого способа кодирования МО),  − суммарное время срабатывания всех схем, включенных в ФА;

нс

, где − время исполнения в ОБ *k*-ой микрооперации (в случае синхронизации с постоянным тактом);

нс

; , где  − время срабатывания счетчика АМК (СчАК);

нс

нс



нс – интервал δ, гарантирующий загрузку «правильного слова» в РМК.

 нс

Расчетное значение тактовой частоты определяется величиной F=1/TОУ.



Рабочая частота Fр выбирается из гостированного ряда частот {F} при условии, что FР≤0,8F [4]:



Диаграмма функционирования процессора при выполнении фрагмента программы А+19-А+20-А+7 представлена на рисунке 7.1.



*Рисунок 7.1 – Временная диаграмма процесса исполнения МК*

**Заключение**

В ходе курсовой работы было спроектировано ЦОУ, способное выполнять определенную часть из общей системы команд. Устройство управления представляет собой управляющий автомат с программируемой логикой и принудительным способом адресации. Результатом данной курсовой работы является:

1. Определенные форматы команд и данных и сформированная система команд;
2. Граф-схема функционирования ЦОУ;
3. Структурная схема ЦОУ;
4. Архитектура внешних выводов ЦОУ;
5. Форматы микрокоманд, кодирование микрокоманд и микропрограмма функционирования управляющего автомата;
6. Принципиальная схема УА;
7. Рассчитанные временные характеристики ЦОУ.

**Библиографический список**

1. Синтез центрального обрабатывающего устройства ЦВМ: Методические указания к курсовому проектированию по дисциплине «Цифровые ЭВМ» для студентов направления 6.050102 – «Компьютерная инженерия» дневной и заочной форм обучения/ Разраб. Ю.К.Апраксин, Т.В.Волкова. − Сева­стополь: Изд-во СевНТУ, 2008. − 36с.
2. Синтез управляющих автоматов с программируемой логикой: Методические указания к циклу лабораторных работ по дисциплине «Цифровые ЭВМ» для студентов направления 6.050102 – «Компьютерная инженерия» дневной формы обучения/ Разраб. Ю.К.Апраксин, Т.В.Волкова. − Сева­стополь: Изд-во СевНТУ, 2009. − 44 с.
3. Горобец А.И. Справочник по конструированию радиоэлектронной аппаратуры (печатные узлы) / А.И. Горобец, А.И. Степаненко, В.М. Коронкевич. - К.: Технiка, 1985
4. В. Б. Тарабрин. Интегральные микросхемы: Справочник. – М.: Радио и Связь, 1983г. - 640с.
5. Сайт ChipInfo [Электронный ресурс]. – Режим доступа: http://www.chipinfo.ru/dsheets/ic/155/index.html

**Приложение А**

Список управляющих и осведомительных сигналов:

|  |  |  |  |
| --- | --- | --- | --- |
| y1 | РК(0:15):=БР(0:15) | х1 | ТП |
| y2 | СчАК(0:28):=СчАК(0:28)+2 | х2 | СчАК(27) |
| y3 | ПортА(0:26):=СчАК(0:26) | х3 | РК(0:1)=00 |
| y4 | ЧтОП | х4 | zОП |
| y5 | РК(16:31):=ПортД(0:15) | х5 | СчАК(28) |
| y6 | БР(0:15):=ПортД(16:31) | х6 | ПортД(0:1)=00 |
| y7 | ТП:=0 | х7 | РК(0:7)=0x08 |
| y8 | РК(0:15):=ПортД(16:31) | х8 | РСРП(30:31)=00 |
| y9 | РК(0:31):=ПортД(0:31) | х9 | РК(0:7)=0х38 |
| y10 | СчАК(0:27):=СчАК(0:27)+4 | х10 | Zвв |
| y11 | РК(0:15):=ПортД(0:15) | х11 | РК(0:7)=0х1B |
| y12 | РАРП(0:4):=0.РК(12:15) | х12 | РК(11)=1 |
| y13 | ЧтРП | х13 | РСРП(29:31)=000 |
| y14 | ПортА(0:26):=РСРП(3:29) | х14 | zАЛУ |
| y15 | РСРП(0:31):= ПортД(0:31) | х15 | РФ(0:4)=00000 |
| y16 | РАРП(0:4):= 0.РК(8:11) | х16 | РК(0:7)=0х6B |
| y17 | ЗпРП | х17 | РК(23:24)=00 |
| y18 | ПортА(0:26):=0..0.РК(8:15) | х18 | ПортД(29:31)=000 |
| y19 | МП | х19 | РК(0:7)=0х30 |
| y20 | Р2АЛУ(0:31):=ПортД(0:31) | х20 | РК(8:11) = 0000 |
| y21 | РАРП(0:4):=1.РК(8:11) | х21 | РК(12:15) = 0000 |
| y22 | ПортА(0:26):=ПортА(0:26)+1 | х22 | РК(8:11)=1111 |
| y23 | Р1АЛУ(0:31):=РСРП(0:31) | х23 | РПр(0:1) = РК(8:11) |
| y24 | Р2АЛУ(32:63):=ПортД(0:31) | х24 | РК(0:7)=0х00 |
| y25 | РАРП(0:4):=1.РК(8:10).1 |  |  |
| y26 | Р1АЛУ(32:63):=РСРП(0:31) |  |  |
| y27 | СлПТ |  |  |
| y28 | РСРП(0:31):=Р1АЛУ(32:63) |  |  |
| y29 | РПр(0:1):=CD(ШАЛУ(0:3)) |  |  |
| y30 | РФ(0:2):=ШАЛУ(3:5) |  |  |
| y31 | РСРП(0:31):=Р1АЛУ(0:31) |  |  |
| y32 | ПортА(0:26):=0..0.РК(15:22) |  |  |
| y33 | ПортА(0:26):=ПортД(3:29) |  |  |
| y34 | Р2АЛУ(0:63):=0..0.РК(26:31) |  |  |
| y35 | Р1АЛУ(0:31):=ПортД(0:31) |  |  |
| y36 | Р1АЛУ(32:63):=ПортД(0:31) |  |  |
| y37 | СдвВпр |  |  |
| y38 | ПортД(0:31):=Р1АЛУ(32:63) |  |  |
| y39 | ЗпОП |  |  |
| y40 | ПортА(0:26):=ПортА(0:26)-1 |  |  |
| y41 | ПортД(0:31):=Р1АЛУ(0:31) |  |  |
| y42 | СчАК(0:28):=РСРП(3:31) |  |  |
| y43 | ТП:=1 |  |  |
| y44 | РФ(0:4):=00010 |  |  |
| y45 | РФ(0:4):=00001 |  |  |
| y46 | yK |  |  |