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».

Вариант №22

Выполнил: ст. гр. М-41з

Тарасенко И.А.

Проверил: Апраксин Ю.К.

Севастополь 2012

***Содержание***

[Введение 3](#_Toc216487135)

[1. Постановка задачи 4](#_Toc216487136)

[2. Разработка системы команд реализуемых операционным устройством и описание форматов обрабатываемых данных ………..](#_Toc216487137)6

[3. Объединённая ГСА функционирования ЦОУ](#_Toc216487138) 8

[4. Структурная схема ЦОУ](#_Toc216487139) 10

[5. Синтез УА](#_Toc216487140) 14

[5.1. Общая структура](#_Toc216487141) 14

[5.2. Адресация микрокоманд](#_Toc216487142) 15

[5.3. Кодирование поля микроопераций.](#_Toc216487143) 18

[5.4. Микропрограмма функционирования УА.](#_Toc216487144) 18

[5.5. Разработка принципиальной схемы управляющего автомата](#_Toc216487145) 25

[6. Определение временных характеристик работы ЦОУ](#_Toc216487147) 27

[Заключение](#_Toc216487148) 29

[Библиографический список](#_Toc216487149) 30

***Введение***

Курсовое проектирование преследует цель практического закрепления основных разделов дисциплины «Цифровые ЭВМ», посвященных изучению принципов структурной и функциональной организации цифровых вычислительных машин и их узлов. Эта цель наилучшим образом достигается в процессе проектирования основных блоков компьютера. Процесс проектирования предполагает решение целого ряда задач, начиная от построения первичной спецификации проектируемого устройства до разработки его принципиальной схемы. ЦВМ состоит из многих тысяч электрических компонент и относится к классу сложных систем, анализ и синтез которых базируется на иерархическом подходе к вопросам их организации. Внутренняя организация ЦВМ строится и описывается на трех уровнях: микропрограммном, логическом и электрическом. Принцип микропрограммной организации (микропрограммного управления) является наиболее значительным, поскольку только на его основе можно результативно определить, как и какими средствами выполняются действия в ЦВМ. Организация ЦВМ на уровне логических и электрических схем базируется на принципах теории цифровых автоматов и цифровой электроники. ЦВМ принято строить как композицию операционных устройств, на каждое из которых возлагается выполнение некоторой совокупности операций, необходимых для выполнения заданного списка команд. В функциональном отношении операционное устройство разделяется на две части: операционный автомат (ОА) и управляющий автомат (УА). ОА служит для хранения совокупности слов информации, выполнения набора микроопераций над ними и вычисления логических условий. УА обеспечивает требуемый порядок следования микроопераций на основе заданных микропрограмм.

# *1. Постановка задачи*

В ходе выполнения курсового проектирования необходимо произвести разработку центрального обрабатывающего устройства ЦВМ, сгенерировать устройство управления для разрабатываемого обрабатывающего устройства. Проектируемое ЦОУ должно обеспечивать реализацию хранящейся в ОП программы, состоящей из ограниченного множества типовых команд. К этим командам относится:

- арифметическая команда

- логическая команда

- команда пересылки данных

- команда обращения к устройству ввода/вывода

- команда передачи управления

- команда «стоп».

В командах должны быть реализованы следующие способы адресации операндов:

- непосредственная;

- относительная;

- косвенная;

- регистровая;

- косвенная через регистр.

Процессор, обеспечивающий исполнение каждой команды, должен:

1. осуществить выборку команды из ОП в строгом соответствии с форматом команды,
2. расшифровать код операции в команде,
3. выполнить расшифрованную операцию,
4. подготовить систему к выполнению следующей команды.

***Задание***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №  вар | Команды АЛУ | | | | Дополнит.  команды | | | ОП | | РП | | Способы адресации | | | | | |
| КА | ДА | КЛ | ДЛ | ПУ | ОбрП | ВВ | ЕОП | ШВ | ЕРП | Тип | Н | П | К | Р | КР | О |
| 22 | **×** | I2 | **∨** | L1 | ПИ> | ЗП|F8 | ПМ | 128 | 8 | 8 | Р | + |  | + | + | + | + |
| Устройство управления – автомат с программируемой логикой | | | | | | | | | | | | | | | | | |
| Способ адресации микрокоманд – естественная адресация (**Е**) | | | | | | | | | | | | | | | | | |
| Способ кодирования микроопераций – горизонтально-вертикальный (**ГВ**) | | | | | | | | | | | | | | | | | |

**КА** – арифметическая команда: умножение (\*)

Данные: I2 – операнды целого типа размером 2 байта

**КЛ** – логическая команда: дизъюнкция

Данные: L1 – двоичный вектор длиной 1байт

**ПУ** – команды передачи управления: ПИ> – переход по индексу «больше»

**ОбрП** – команды обращения к памяти: запись в память (ЗП)

Данные : F8

**ВВ** – команды ввода/вывода: ПМ –обратная передача;

**Характеристики ОП**:

**ЕОП** – емкость ОП в мегабайтах = 128

**ШВ** – ширина выборки = 8

**Характеристики РП:**

**ЕРП** – емкость регистровой памяти – 8;

**тип**: **Р** – раздельная РП

***2. Форматы команд***

При выборе форматов команд разработчик исходил из критерия минимального разнообразия, но в тоже время, необходимости реализовать все заданные способы адресации. Команда должна состоять из следующих полей: КОП и поля содержащие операнды. В поле КОП выделяют три поля: код длины команды, код класса команды, номер команды в списке класса.

Ниже представлены форматы для 5 разрабатываемых команд.

*Арифметическая команда - умножение 16-разрядных целых чисел.*

|  |  |  |  |
| --- | --- | --- | --- |
| КОП | R1 | В2 | D2 |
| 0 7 | 8 10 | 11 13 | 14 31 |

Признак результата остается без изменения. Флаги A, S не вырабатываются.

Первый операнд хранится в регистре с номером R1(регистровая), второй операнд хранится в ОП по адресу, который содержится в базовом регистре В2+смещение D2

*Логическая команда - дизъюнкция двоичных векторов длиной 1 байт.*

|  |  |  |
| --- | --- | --- |
| КОП | Im2 | A(y) |
| 0 7 | 8 15 | 16 31 |

Первый операнд хранится непосредственно на месте первого операнда. Второй операнд хранится в ОП по адресу на который указывает адрес А(у)

*Команда пересылки данных – запись в память 64-разрядного числа с плавающей точкой.*

|  |  |  |  |
| --- | --- | --- | --- |
| КОП | R1 | R2(k) | 00 |
| 0 7 | 8 10 | 11 13 | 14 15 |

Первый операнд хранится в регистрах R1 и R1+1, второй операнд хранится в ячейке памяти с адресом хранящимся по адресу находящемуся в регистре R2.

*Команда обращения к устройству ввода/вывода – передача из процессора в МВВ.*

|  |  |
| --- | --- |
| КОП | НУВВ |
| 0 7 | 8 15 |

Адрес устройства хранится в команде (прямая адресация).

*Команда передачи управления – переход по индексу >.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| КОП | R1 | R2 | R3 | R4 |  |
| 0 7 | 8 10 | 11 13 | 14 16 | 17 19 | 20 31 |

Первый операнд содержится в регистре R1, второй в R2. Первый операнд сравнивается со вторым и если он больше то происходит переход по адресу который хранится в базовом регистре B2+смещение D2.

*Приведем таблицу, в которой описывается система команд ЦОУ.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер класса | Номер команды | Название | Содержание | ПР | Флаги | Код |
| 1 | 1 | Запись в ОП | ((R2)):=(R1).(R1+1) | - | A,S | 10010001 |
| 2 | 1 | Умножение | R1:=R1\*((B2)+D2) | =0  >0  <0 | A,S | 10001011 |
| 3 | 1 | Или | ((А(у)):=((А(у)))˅Im2 | =0 | А,S | 10000111 |
| 4 | 1 | ПИ > | (R1>R2)?: СчАК:=((В2)+D2) | - | - | 10001111 |
| 5 | 1 | Команды вывода | ПОРТi := ПОРТD  I=НУВВ | - | - | 10011110 |
| 6 | 1 | стоп | - | - | K | 1000000 |

***3. Объединенная ГСА выполнения команд***

*ГСА функционирования операционного устройства предоставлена на плакате 1 (2012.М-41з.22.01).*

Объединенная ГСА работы управляющего автомата включает в себя следующие подсхемы:

– ГСА выборки команды;

В зависимости от кода операции выполнение программы производится по одной из следующих подсхем:

– ГСА исполнения арифметической команды ;

– ГСА исполнения логической команды ;

– ГСА исполнения команды передачи управление;

– ГСА исполнения команды обращения к памяти;

– ГСА исполнения команды ввода\вывода;

– ГСА исполнения команды «Стоп»;

– Подсхема, сигнализирующая о нарушении работы.

Упрощенная структура алгоритма выборки команды представлена на рисунке 3.1. [1]



Рисунок 3.1 – Схема алгоритма функционирования ЦОУ

Для того, чтобы выполнить любую команду программы, ее необходимо выбрать из ОП на РК. В процессе проектирования у нас получилось два формата команд длиной 2,4 байт, а ширина выборки ОП равна 8 байтам. Корректный адрес команды при таких предположениях должен быть кратен 2 или 4, а его значение не превышать предельно допустимое, определяемое емкостью ОП.

ГСА выборки команды разработана при следующих предположениях: ЕОП=128 МВ, ШВ=8 байтов, в системе команд есть команды длиной 2 байта и 4 байта, команды загружаются на РК полусловами.

Для уменьшения числа обращений в ОП в процессе выборки команды, в структурную схему процессора вводятся два регистра: АР(0:24) и БР(0:63).

АБ – регистр, на котором хранится адрес слова ОП, считанного на буферный регистр (БР),

ТРК – триггер, состояние которого указывает номер полуслова команды, загружаемого в РК.

Операнды, вступающие в операцию, могут находиться в РОН, РПТ и ОП. Все регистры РОН или РПТ адресуются четырехразрядным полем. Разрядность поля адреса зависит от количества регистров в блоке (8, 16 или 32), в нашем случае их 8. Если регистровый блок общий, то адреса РПТ должны отличаться от адресов РОН. Нарушение этого условия в команде называется неправильной спецификацией и отмечается установкой соответствующего бита регистра флагов в единичное состояние. При неправильной спецификации выполнение операции прекращается.

Вычисленный исполнительный адрес операндов, находящихся в ОП, также может быть некорректным. Возможна ситуация «неправильная адресация», когда адрес выходит за пределы фактической емкости ОП. Флаг нарушения адресации в этом случае устанавливается в единичное состояние и операция не выполняется.

Неправильная спецификация операндов, находящихся в ОП, является следствием нарушения целочисленности границы для обрабатываемого типа данных.

***4.*** ***Структурная схема ЦОУ***

*Структурная схема приведена на чертеже 2012.41з.22.02*

Операционное устройство синтезировано на основе известных технических решений с учетом специфики построения целевого процессора и особенностей реализации, заданных условием курсового проекта команд. Операционное устройство состоит из отдельных блоков, электрически связанных между собой информационными и управляющими линиями.

Основными составляющими операционного устройства являются:

Управляющий автомат (УА);

Арифметико-логическое устройство (АЛУ);

Оперативная память (ОП);

Регистровая память (РП);

Регистр команд (РК);

Буферный регистр (БР);

Регистр адреса буфера (АБ);

Счетчик адреса команд (СчАК);

Сумматор адреса(СмА);

ПортД

ПортА

Регистр флагов прерываний (РФ)

Регистр признака результата (РПр);

Триггер TRK;

Триггер нарушения спецификации (S);

Триггер нарушения адресации (А);

*Управляющее устройство (автомат) - УУ.*

УУ обеспечивает выработку управляющих сигналов УУ в зависимости от момента времени и поступающих в УУ сигналов Х, необходимых для осуществления ветвления программы. Работы УУ управляется входными сигналами А – начало работы, Б – прекращение работы. Таким образом, УУ имеет связи с операционным блоком и «внешним миром».

*Оперативная память – ОП.*

Оперативная память предназначена для хранения в себе кода исполняемой программы, а также для хранения данных с которыми может оперировать программа. ОП имеет два регистра: регистр адреса и регистр данных, необходимых для записи считывания данных. Управляется ОП с помощью сигналов записи и чтения ОП. При работе с оперативной памятью следует учитывать что время необходимое для записи считывания сравнительно больше, нежели время работы ЦОУ, и следовательно возникает необходимость ожидания осведомительного сигнала от ОП о ее занятости (Zоп). Уровень логической единицы в осведомительном сигнале Zоп свидетельствует о том что запись/чтение данных в процессе, и последующая работа программы может привести к потере данных. Zоп=0 о способности ОП к работе.

*Регистровая память – РП.*

РП в рассматриваемом варианте представляется раздельной регистровой памятью, т.е. существует отдельно 8 регистров с плавающей точкой (РПП), и 8 регистров с фиксированной точкой(РПФ). Функционально РПП и РПФ не отличаются, каждая имеет по 2 регистра регистр данных (РСРПП и РСРПФ) необходимых для обмена данными, и регистр адреса (РАРПП и РАРПФ). Каждая из регистровых памятей имеет управляющие сигналы записи и чтения. Запись/чтение осуществляется за 1 машинный такт, и, следовательно, ожидание окончания записи не нужно для работы.

*Арифметико-логическое устройство - АЛУ.*

АЛУ, как видно из названия данного блока, выполняет все арифметически и логические операции в ЦОУ. АЛУ имеет два регистра для операндов, результаты операции, если иное не оговорено помещаются в первый регистр. АЛУ управляется сигналами для соответствующих операций над данными (умножение, дизъюнкция и тд.). Время функционирования для каждой операции различно и при использовании АЛУ следует это учитывать и ожидать результата выполнения команды с помощью опроса оповещающего сигнала (Zалу) работа которого схожа с оповещающим сигналом занятости от ОП.

*АЛУ выполняет следующие функции:*

Сложение с фиксированной точкой ,сигнал инициации – СлФТ.

Вычитание с фиксированной точкой ,сигнал инициации – ВЫЧ.

Умножение с фиксированной точкой, сигнал инициализации – УмнФ

Дизъюнкция - Диз

*Регистр команды – РК.*

Регистр команды хранит в себе текущую команду, выбранную из ОП. Разрядность регистра команд выбирается по максимальной длине команды, встречающейся в системе команд разрабатываемого ЦОУ. Подробнее о форматах команд было изложено в пункте первом данной пояснительной записки.

*Буферный регистр – БР.*

Буферный регистр предназначен для хранения, считанного из ОП слова при выборке очередной команды. Введение буферных регистров приводит к уменьшению числа обращений к ОП, а следовательно и времени выборки команды. Разрядность буферного регистра берется по ширине выборки из ОП.

*Регистр адреса буфера – АБ.*

АБ хранит в себе адрес слова записанного в буферный регистр.

*Триггер разрядности команды - ТРК.*

ТРК используется при выборке команды из ОП и служит для определения разрядности команды (2 или 4 байта).

*Регистр признака результата – РПР.*

В нашем случае на него передаются значения признаков результата с ШАЛУ [0:1] .

Признаки результата: 0 − = 0; 1 − < 0 ; 2 − >0; 3 − переполнение.

РПР служит для хранения признака результата операции осуществленной на АЛУ. Согласно признакам результатов может осуществляться условный переход по программе.

*Регистр флагов – РФ.*

Регистр флагов хранит флаги, формируемые АЛУ в процессе выполнения операции.

*Счётчик адреса команд*

Для того, чтобы выполнить некоторую команду, ее необходимо выбрать из ОП на регистр команды РК. Адрес выбираемой команды хранится на счетчике адреса команд (СчАК). Счетчик адреса команд предназначен для хранения, приема и увеличения на 2 адреса исполняемой команды. СчАК имеет разрядность 27. Единица в младшем разряде СчАКа (26) говорит о нарушении спецификации

*Сумматор адреса*

Сумматор адреса, предназначен для вычисления исполнительного адреса (АИСП) в случае его задания в виде нескольких компонент (например, при относительной адресации АИСП=((B)+D). Разрядность СмА отличаться от СчАК дополнительным старшим разрядом, фиксирующим переполнение, и имеет разрядность 28 .

*ПортА*

Порт А – порт, через который процессор выдает адрес слова ОП или номер УВВ.

*ПортД*

Порт Д – порт процессора, через который он обменивается данными с внешними устройствами (ОП и МВВ).

*Триггер неправильной адресации*

Сигнализирует о нарушении адресации, которое возникает в том случае, когда адрес выходит за пределы фактической емкости ОП, т.е. 128 Мбайт. В случае нарушения адресации флаг А устанавливается в 1 под действием соответствующего управляющего сигнала .

*Триггер неправильной спецификации*

Сигнализирует о нарушении спецификации, которое является следствием нарушения целочисленности границы для обрабатываемой единицы информации. В случае нарушения спецификации флаг S устанавливается в 1 под действием соответствующего управляющего сигнала.

Для реализации проектируемого ЦОУ будем использовать корпус с 132 выходами (27 – шина адрса, 64 шина данных, RD, WR, IN, OUT, +5, GND,READY I/O, READY MEM, CLC, RESET). Такая конструкция позволит не мультиплексировать шины данных и адреса, незадействованные выходы можно использовать для питания и заземления микросхемы.

# *5. Синтез УА*

### *5.1. Общая структура*

Обобщенная структура УА с программируемой логикой представлена на рисунке 5.1.

Память микропрограмм (ПМП) организована в виде ПЗУ. В качестве АМК используется счетчик (в случае естественной адресации) или регистр (в случае принудительной адресации). Пуск автомата осуществляется подачей в схему управляющего сигнала **В**, разрешающего подачу тактирующих сигналов на ПМП, останов – подачей управляющего сигнала **А**. Сигнал Сброс или Уст (установка) устанавливает на АМК адрес начальной микрокоманды в микропрограмме. 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 конкретной МК. Формирователь адреса (ФА), сравнивая поле ЛУ в МК с двоичным вектором осведомительных сигналов (x1, x2,…,xn), принимает решение о передаче кода НА на АМК.[1]



Рисунок 5.1 – Обобщенная структура УА с программируемой логикой

(В – сигнал, запускающий УА, А – сигнал, останавливающий УА)

* 1. ***Адресация микропрограммы***

При синтезе управляющего автомата осведомительным и управляющим сигналам были поставлены в соответствие следующие коды. Кодировка управляющих сигналов (У) представлена в таблице 1, а осведомительных сигналов – в таблице 2.

Таблица 5.1. Описание микроопераций

|  |  |
| --- | --- |
| y1 | ПортА(0:23):=СчАК(0:23) |
| y2 | АБ(0:23):=СчАК(0:23) |
| y3 | ЧтОП |
| y4 | БР(0:63):=ПортД(0:63) |
| y5 | РК(0:15):=БР[СчАК(24:25)] |
| y6 | РК(16:31):=БР[СчАК(24:25)] |
| y7 | СчАК(0:27):=СчАК(0:27)+2 |
| y8 | ТРК:=1 |
| y9 | ТРК:=0 |
| Y9 | S=1 |
| y10 | СмА(0:27) := 0..0РК(14:31) |
| y11 | РАРП(0:2) := 0.РК(11:13) |
| y12 | ЧтРП |
| y13 | СмА(0:27) := СмА(0:27)+0.РСРП(5:31) |
| y14 | A := 1 |
| y15 | Порт А(0:23) :=СмА(1:24) |
| y16 | Р2(0:15) := ПортД(0:15) |
| y17 | РАРП(0:3):=РК(8:10) |
| y18 | Р1(0:15) := РСРП(0:15) |
| y19 | УмнФ |
| y20 | РСРП(0:15):=Р1(0:15) |
| y21 | ЗпРП |
| y22 | РПр(0:1):=ШАЛУ(0:1) |
| y23 | ПортА(0:23):=0..Рк(16:31) |
| y24 | СмА(0:27):=ПортД(5:31) |
| y25 | ПортА(0:23):=ПортД(5:28) |
| y26 | Р1(0:7):=ПортД(СмА(25:27)) |
| y27 | Р2(0:7):=РК(8:15) |
| y28 | Диз |
| y29 | ПортД(СмА(25:27)):=Р1(0:7) |
| y30 | ЗпОП |
| y31 | СмА(0:27) := 0..0РК(17:31) |
| y32 | РАРП(0:2) := 0.РК(15:17) |
| y33 | Р1(0:31):=РСПР(0:31) |
| y34 | РАРП(0:2) := 0.РК(11:13) |
| y35 | Р2(0:31):=РСПР(0:31) |
| y36 | СЛФТ |
| y37 | РСРП(0:31) := Р1(0:31) |
| y38 | РАРП(0:2):=0.РК(8:10) |
| y39 | РАРП(0:2):=0.РК(11:12).1 |
| y40 | Р2(0:31):=РСРП(0:31) |
| y41 | ВЫЧ |
| y42 | СЧАК(0:23):=СмА(1:24) |
| y43 | ПортА(0:23) := 0..0.PK[8:15] |
| y44 | ПМ |
| y45 | РАРП(0:2):=1.РК(8:10) |
| y46 | ПортД(0:31)РСРП(0:31) |
| y47 | РАРП(0:2):=1.РК(8:9).1 |
| y48 | ПортД(32:63):=РСРП(0:31) |
| y49 | прерывание |
| Y50 | Yk:=1 |
| Y51 | Р1(0:7):=ПортД(0:7) |
| Y52 | Р1(0:7):=ПортД(8:15) |
| Y53 | Р1(0:7):=ПортД(16:23) |
| Y54 | Р1(0:7):=ПортД(24:31) |
| Y55 | Р1(0:7):=ПортД(32:39) |
| Y56 | Р1(0:7):=ПортД(40:47) |
| Y57 | Р1(0:7):=ПортД(48:55) |
| Y58 | Р1(0:7):=ПортД(56:63) |
| Y59 | ПортД(0:7):=Р1(0:7) |
| Y60 | ПортД(8:15):=Р1(0:7) |
| Y61 | ПортД(16:23):=Р1(0:7) |
| Y62 | ПортД(24:31):=Р1(0:7) |
| Y63 | ПортД(32:39):=Р1(0:7) |
| Y64 | ПортД(40:47):=Р1(0:7) |
| Y65 | ПортД(48:55):=Р1(0:7) |
| Y66 | ПортД(56:63):=Р1(0:7) |
| Y67 | K:=1 |
| Y68 | РАРПП(0:2):=РК(11:13) |

Таблица 5.2 Описание условий

|  |  |
| --- | --- |
| x1 | СчАК(26) |
| x2 | АБ(0:23):=СчАК(0:23) |
| x3 | Zоп |
| x4 | ТРК |
| x5 | РК(0:1)=00 |
| x6 | РК(0:7)=8В |
| x7 | РК(0:7)=87 |
| x8 | РК(0:7)=8F |
| x9 | РК(0:7)=91 |
| x10 | РК(0:7)=9Е |
| x11 | РК(0:7)=40 |
| x12 | РК(11:13) = 000 |
| x13 | СмА(0) |
| x14 | СмА(27)=0 |
| x15 | zАЛУ |
| x16 | РПР(0:1)=11 |
| x17 | РК(30:31)=00 |
| x18 | ПортД(0:4)=0..0 |
| x19 | РК(15:17) = 000 |
| x20 | ПрР(0:1)=10 |
| x21 | РК(10) |
| x22 | РСРП(29:31)=000 |
| x23 | РСРП(0:4)=0000 |
| x24 | Zвв |
| X25 | РФ(0…k):=100…0 |
| Х26 | СмА(25) |
| Х27 | СмА(26) |
| Х28 | СмА(27) |

### *5.3. Кодирование поля микроопераций*

Кодирование микроопераций позволяет сократить размер команды и таким образом уменьшить общий объем используемой устройством памяти. Проведение кодирования методом раздельных полей требует построения графа несовместимых операций. Для наглядности и удобства воспользуемся матрицей совместимости (так как совместимых операций намного меньше, чем несовместимых). Строки и столбцы этой матрицы соответствуют микрооперациям. Элемент матрицы A[i,j] равен 0, если i-я операция совместима с j-й операцией. Если i-я операция несовместима с j-й операцией, элемент таблицы A[i,j] равен. Главная диагональ таблицы не используется, так как считается, что каждая операция совместима сама с собой.

Таблица совместимости находится в приложении А.

По матрице совместимости определяем множества несовместимых операций.

Таблица 5.3. Группы несовместимости

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| #N | МО | Двоичный код | | | | |
| #1 | y1 | 0 | 0 | 0 | 0 | 1 |
| y7 | 0 | 0 | 0 | 1 | 0 |
| Y9 | 0 | 0 | 0 | 1 | 1 |
| y12 | 0 | 0 | 1 | 0 | 0 |
| y15 | 0 | 0 | 1 | 0 | 1 |
| y18 | 0 | 0 | 1 | 1 | 0 |
| y20 | 0 | 0 | 1 | 1 | 1 |
| y24 | 0 | 1 | 0 | 0 | 0 |
| y27 | 0 | 1 | 0 | 0 | 1 |
| y30 | 0 | 1 | 0 | 1 | 0 |
| y35 | 0 | 1 | 0 | 1 | 1 |
| y37 | 0 | 1 | 1 | 0 | 0 |
| y43 | 0 | 1 | 1 | 0 | 1 |
| y51 | 0 | 1 | 1 | 1 | 0 |
| y52 | 0 | 1 | 1 | 1 | 1 |
| y53 | 1 | 0 | 0 | 0 | 0 |
| y54 | 1 | 0 | 0 | 0 | 1 |
| y55 | 1 | 0 | 0 | 1 | 0 |
| y56 | 1 | 0 | 0 | 1 | 1 |
| y57 | 1 | 0 | 1 | 0 | 0 |
| y58 | 1 | 0 | 1 | 0 | 1 |
| y40 | 1 | 0 | 1 | 1 | 0 |
| y50 | 1 | 0 | 1 | 1 | 1 |
| #2 | y2 | 0 | 0 | 0 | 0 | 1 |
| y8 | 0 | 0 | 0 | 1 | 0 |
| y9 | 0 | 0 | 0 | 1 | 1 |
| y11 | 0 | 0 | 1 | 0 | 0 |
| y16 | 0 | 0 | 1 | 0 | 1 |
| y19 | 0 | 0 | 1 | 1 | 0 |
| y21 | 0 | 0 | 1 | 1 | 1 |
| y23 | 0 | 1 | 0 | 0 | 0 |
| y25 | 0 | 1 | 0 | 0 | 1 |
| y34 | 0 | 1 | 0 | 1 | 0 |
| y36 | 0 | 1 | 0 | 1 | 1 |
| y39 | 0 | 1 | 1 | 0 | 0 |
| y44 | 0 | 1 | 1 | 0 | 1 |
| y45 | 0 | 1 | 1 | 1 | 0 |
| y47 | 0 | 1 | 1 | 1 | 1 |
| y48 | 1 | 0 | 0 | 0 | 0 |
| y68 | 1 | 0 | 0 | 0 | 1 |
| y29 | 1 | 0 | 0 | 1 | 0 |
|  | y27 | 1 | 0 | 0 | 1 | 1 |
|  | y59 | 1 | 0 | 1 | 0 | 0 |
|  | y60 | 1 | 0 | 1 | 0 | 1 |
|  | y61 | 1 | 0 | 1 | 1 | 0 |
|  | y62 | 1 | 0 | 1 | 1 | 1 |
|  | y63 | 1 | 1 | 0 | 0 | 0 |
|  | y64 | 1 | 1 | 0 | 0 | 1 |
|  | y65 | 1 | 1 | 0 | 1 | 0 |
|  | y66 | 1 | 1 | 0 | 1 | 1 |
| #3 | y3 | 0 | 0 | 0 | 0 | 1 |
| y4 | 0 | 0 | 0 | 1 | 0 |
| y5 | 0 | 0 | 0 | 1 | 1 |
| y6 | 0 | 0 | 1 | 0 | 0 |
| y10 | 0 | 0 | 1 | 0 | 1 |
| y13 | 0 | 0 | 1 | 1 | 0 |
| y14 | 0 | 0 | 1 | 1 | 1 |
| y17 | 0 | 1 | 0 | 0 | 0 |
| y22 | 0 | 1 | 0 | 0 | 1 |
| y28 | 0 | 1 | 0 | 1 | 0 |
| y31 | 0 | 1 | 0 | 1 | 1 |
| y33 | 0 | 1 | 1 | 0 | 0 |
| y38 | 0 | 1 | 1 | 0 | 1 |
| y42 | 0 | 1 | 1 | 1 | 0 |
| y67 | 0 | 1 | 1 | 1 | 1 |
|  | y32 | 1 | 0 | 0 | 0 | 0 |
|  | y41 | 1 | 0 | 0 | 0 | 1 |
|  | y68 | 1 | 0 | 0 | 1 | 0 |
|  | y46 | 1 | 0 | 0 | 1 | 1 |

*Произведем кодирование полей условий*

Таблица 5.4. Коды условий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| хбп | 0 | 0 | 0 | 0 | 0 |
| х1 | 0 | 0 | 0 | 0 | 1 |
| х2 | 0 | 0 | 0 | 1 | 0 |
| х3 | 0 | 0 | 0 | 1 | 1 |
| х4 | 0 | 0 | 1 | 0 | 0 |
| х5 | 0 | 0 | 1 | 0 | 1 |
| х6 | 0 | 0 | 1 | 1 | 0 |
| х7 | 0 | 0 | 1 | 1 | 1 |
| х8 | 0 | 1 | 0 | 0 | 0 |
| х9 | 0 | 1 | 0 | 0 | 1 |
| х10 | 0 | 1 | 0 | 1 | 0 |
| х11 | 0 | 1 | 0 | 1 | 1 |
| х12 | 0 | 1 | 1 | 0 | 0 |
| х13 | 0 | 1 | 1 | 0 | 1 |
| х14 | 0 | 1 | 1 | 1 | 0 |
| х15 | 0 | 1 | 1 | 1 | 1 |
| х16 | 1 | 0 | 0 | 0 | 0 |
| х17 | 1 | 0 | 0 | 0 | 1 |
| х18 | 1 | 0 | 0 | 1 | 0 |
| х19 | 1 | 0 | 0 | 1 | 1 |
| х20 | 1 | 0 | 1 | 0 | 0 |
| х21 | 1 | 0 | 1 | 0 | 1 |
| х22 | 1 | 0 | 1 | 1 | 0 |
| х23 | 1 | 0 | 1 | 1 | 1 |
| х24 | 1 | 1 | 0 | 0 | 0 |
| х25 | 1 | 1 | 0 | 0 | 1 |
| х26 | 1 | 1 | 0 | 1 | 0 |
| х27 | 1 | 1 | 0 | 1 | 1 |
| х28 | 1 | 1 | 1 | 0 | 0 |

*Кодирование поля адреса перехода*

Для кодировки адресов используем двоичный позиционный код. Код будет занимать восемь разрядов, так как максимальный адрес равен А+133.

]log 2 133[ = 8

Кодировка микропрограммы приведена в таблице 5.5.

*Приведем ниже таблицу кодировки микропрограммы.*

Таблица 5.5. Кодировка микропрограммы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МК опер | В | #1 | | | | | #2 | | | | | #3 | | | | | Адрес МК | Микрокоманда | | | |
| МУ усл | В | Лог усл | | | | | Адрес | | | | | | | | Х | Х | В | Х | А | |
| А+0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | А+0 |  | x1 | A+10 | |
| А+1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | А+1 |  | x2 | A+5 | |
| А+2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | А+2 |  | y1y2y3 | | |
| А+3 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | А+3 |  | x3 | A+3 |  |
| А+4 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 0 | 1 | 0 | А+4 |  | y4 |  |  |
| А+5 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | А+5 |  | x4 | A+12 |  |
| А+6 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 0 | 1 | 1 | А+6 |  | y5 |  |  |
| А+7 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | А+7 |  | x5 | A+13 |  |
| А+8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |  |  |  | А+8 |  | y7y8 | | |
| А+9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | А+9 |  | xбп | A+1 |  |
| А+10 | 0 | 0 | 0 | 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  | А+10 |  | Y9 |  |  |
| А+11 | 0 | 1 | 0 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  | А+11 |  | y50 |  |  |
| А+12 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 0 | 0 | А+12 |  | y6 |  |  |
| А+13 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |  |  |  |  |  | А+13 |  | y7y9 |  |  |
| А+14 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | А+14 |  | x6 | A+22 |  |
| А+15 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | А+15 |  | x7 | A+41 |  |
| А+16 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | А+16 |  | x8 | A+98 |  |
| А+17 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | А+17 |  | x9 | A+114 |  |
| А+18 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | А+18 |  | x10 | A+127 |  |
| А+19 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+19 |  | x11 | A+11 |  |
| А+20 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 1 | 1 | 1 | А+20 |  | y67 |  |  |
| А+21 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+21 |  | xбп | A+11 |  |
| А+22 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 0 | 1 | А+22 |  | y10 |  |  |
| А+23 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | А+23 |  | x12 | A+26 |  |
| А+24 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |  | А+24 |  | y11y12 | | |
| А+25 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 0 | А+25 |  | y13 |  |  |
| А+26 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | А+26 |  | x13 | A+39 |  |
| А+27 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | А+27 |  | x14 | A+30 |  |
| А+28 | 1 | 0 | 0 | 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  | А+28 |  | Y9 |  |  |
| А+29 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+29 |  | xбп | A+11 |  |
| А+30 | 0 | 0 | 0 | 1 | 0 | 1 |  |  |  |  |  | 0 | 0 | 0 | 0 | 1 | А+30 |  | y15y3 | | |
| А+31 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | А+31 |  | x3 | A+31 |  |
| А+32 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | А+32 |  | y16y17y12 | | |
| А+33 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |  |  |  |  |  | А+33 |  | y18y19 | | |
| А+34 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | А+34 |  | x15 | A+34 |  |
| А+35 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | А+35 |  | y20y21y22 | | |
| А+36 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+36 |  | x16 | A+0 |  |
| А+37 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | А+37 |  | x25 | A+0 |  |
| А+38 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+38 |  | xбп | A+11 |  |
| А+39 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 1 | А+39 |  | y14 |  |  |
| А+40 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+40 |  | xбп | A+11 |  |
| А+41 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | А+41 |  | x17 | A+44 |  |
| А+42 | 0 | 0 | 0 | 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  | А+42 |  | Y9 |  |  |
| А+43 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+43 |  | xбп | A+11 |  |
| А+44 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |  |  |  |  |  | А+44 |  | y23y3 | | |
| А+45 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | А+45 |  | x3 | A+45 |  |
| А+46 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | А+46 |  | x18 | A+49 |  |
| А+47 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 1 | А+47 |  | y14 |  |  |
| А+48 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+48 |  | xбп | A+11 |  |
| А+49 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | А+49 |  | y24y25y3 | | |
| А+50 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | А+50 |  | x3 | A+50 |  |
| А+51 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | А+51 |  | x26 | A+62 |  |
| А+52 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | А+52 |  | x27 | A+68 |  |
| А+53 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | А+53 |  | x28 | A+73 |  |
| А+54 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+54 |  | y51y27y28 | | |
| А+55 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+55 |  | x15 | A+55 |  |
| А+56 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | А+56 |  | x26 | A+80 |  |
| А+57 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | А+57 |  | x27 | A+86 |  |
| А+58 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+58 |  | x28 | A+96 |  |
| А+59 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | А+59 |  | y59y22y30 | | |
| А+60 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | А+60 |  | x3 | A+60 |  |
| А+61 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+61 |  | xбп | A+0 |  |
| А+62 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+62 |  | x27 | A+75 |  |
| А+63 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | А+63 |  | x28 | A+66 |  |
| А+64 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+64 |  | y55y27y28 | | |
| А+65 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+65 |  | xбп | A+55 |  |
| А+66 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+66 |  | y56y27y28 | | |
| А+67 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+67 |  | xбп | A+55 |  |
| А+68 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | А+68 |  | x28 | A+71 |  |
| А+69 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+69 |  | y53y27y28 | | |
| А+70 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+70 |  | xбп | A+55 |  |
| А+71 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+71 |  | y54y27y28 | | |
| А+72 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+72 |  | xбп | A+55 |  |
| А+73 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+73 |  | y52y27y28 | | |
| А+74 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+74 |  | xбп | A+55 |  |
| А+75 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | А+75 |  | x28 | A+78 |  |
| А+76 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+76 |  | y57y27y28 | | |
| А+77 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+77 |  | xбп | A+55 |  |
| А+78 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | А+78 |  | y58y27y28 | | |
| А+79 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+79 |  | xбп | A+55 |  |
| А+80 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+80 |  | x27 | A+91 |  |
| А+81 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | А+81 |  | x28 | A+84 |  |
| А+82 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | А+82 |  | y63y22y30 | | |
| А+83 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+83 |  | xбп | A+59 |  |
| А+84 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | А+84 |  | y64y22y30 | | |
| А+85 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+85 |  | xбп | A+59 |  |
| А+86 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | А+86 |  | x28 | A+89 |  |
| А+87 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | А+87 |  | y61y22y30 | | |
| А+88 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+88 |  | xбп | A+59 |  |
| А+89 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | А+89 |  | y62y22y30 | | |
| А+90 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+90 |  | xбп | A+59 |  |
| А+91 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | А+91 |  | x28 | A+94 |  |
| А+92 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | А+92 |  | y65y22y30 | | |
| А+93 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+93 |  | Xбп | A+59 |  |
| А+94 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | А+94 |  | y66y22y30 | | |
| А+95 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+95 |  | xБп | A+59 |  |
| А+96 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | А+96 |  | y60y22y30 | | |
| А+97 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | А+97 |  | xбп | A+59 |  |
| А+98 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 0 | 1 | 1 | А+98 |  | y31 |  |  |
| А+99 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | А+99 |  | x19 | A+102 |  |
| А+100 | 0 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |  | 1 | 0 | 0 | 0 | 0 | А+100 |  | y32y12 | | |
| А+101 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 0 | А+101 |  | y13 | | |
| А+102 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | А+102 |  | x13 | A+130 |  |
| А+103 | 0 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |  | 0 | 1 | 1 | 0 | 1 | А+103 |  | y38y12 | | |
| А+104 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 1 | 0 | 0 | А+104 |  | y33 |  |  |
| А+105 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |  |  |  |  |  | А+105 |  | y34y12 | | |
| А+106 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |  |  |  |  |  | А+106 |  | y35y36 | | |
| А+107 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+107 |  | x15 | A+107 |  |
| А+108 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | А+108 |  | y37y38y21 | | |
| А+109 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |  |  |  |  |  | А+109 |  | y39y12 | | |
| А+110 | 0 | 1 | 0 | 1 | 1 | 0 |  |  |  |  |  | 1 | 0 | 0 | 0 | 1 | А+110 |  | y40y41 | | |
| А+111 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | А+111 |  | x20 | A+113 |  |
| А+112 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 1 | 1 | 0 | А+112 |  | y42 |  |  |
| А+113 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+113 |  | xбп | A+0 |  |
| А+114 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | А+114 |  | x21 | A+117 |  |
| А+115 | 0 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |  | 1 | 0 | 0 | 1 | 0 | А+115 |  | y68y12 | | |
| А+116 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | А+116 |  | x22 | A+119 |  |
| А+117 | 0 | 0 | 0 | 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  | А+117 |  | Y9 |  |  |
| А+118 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+118 |  | xбп | A+11 |  |
| А+119 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+119 |  | x23 | A+132 |  |
| А+120 | 0 |  |  |  |  |  | 0 | 1 | 0 | 0 | 1 |  |  |  |  |  | А+120 |  | y25 |  |  |
| А+121 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |  |  |  |  |  | А+121 |  | y45y12 | | |
| А+122 | 0 |  |  |  |  |  |  |  |  |  |  | 1 | 0 | 0 | 1 | 1 | А+122 |  | y46 |  |  |
| А+123 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  |  |  |  |  | А+123 |  | y47y12 | | |
| А+124 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |  |  |  |  |  | А+124 |  | y48y30 | | |
| А+125 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | А+125 |  | x3 | A+125 |  |
| А+126 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+126 |  | xбп | A+0 |  |
| А+127 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |  |  |  |  |  | А+127 |  | y43y44 | | |
| А+128 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+128 |  | x24 | A+128 |  |
| А+129 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | А+129 |  | xбп | A+0 |  |
| А+130 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 1 | А+130 |  | y14 |  |  |
| А+131 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | А+131 |  | xбп | A+11 |  |
| A+132 | 0 |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 | 1 | A+132 |  | y14 |  |  |
| A+133 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | A+133 |  | xбп | A+11 |  |

### *5.5. Разработка принципиальной схемы управляющего автомата*

Структура принципиальной схемы состоит из:

ПЗУ микропрограммы работы данного операционного устройства.

Регистр микрокоманды – он необходим для сохранения, на время необходимое для перехода на следующий адрес (перехода в следующее состояние), информации, потому что код микрокоманды появляется на выходах ПЗУ на достаточно короткое время, которого недостаточно для обработки этого кода остальными компонентами проектируемого устройства.

Дешифраторы множеств Y – необходимы для расшифровки кода Y в множестве в операционной команде.

Мультиплексоры условий переходов.

Микросхема ПЗУ в данном устройстве имеет разрядность данных 32х8 разрядов. Для достижения необходимого объема памяти, шины ПЗУ соединяются параллельно.

Сигналы выбора микросхемы определяется с помощью дешифратора, анализируя старшие разряды.

Дешифратор кода множества реализован с использованием микросхемы типа K155ИД3.

В качестве регистра будут использоваться три микросхемы К155ИР13.

Мультиплексор также пришлось формировать из двух мультиплексоров типа К155КП1 (на 16 входов). При этом адресные входы этих ИМС соединены следующим образом: разрешающие сигналы генерируются логикой и подается на входы выбора микросхемы. При данном включении обеспечивается полное соответствие номеров условий и физических номеров входов мультиплексоров.

В принципиальной схеме должна быть предусмотрена возможность установки схемы в нулевое состояние. Это реализуется подведением соответствующего сигнала ко входам обнуления в регистрах.

Принципиальная схема представлена на плакате 2012.М41.022.003.

Список элементов:

DD1- DD9, DD5- DD36 – ПЗУ К155РЕ3

DD10 – дешифратор типа К155ИД7

DD11,DD12 – счетчик К155ИЕ7

DD13 – DD15 - регистры типа К155ИР13

DD16, DD17- мультиплексор К155КП1

DD18 – DD21 - дешифраторы типа К155ИД3

DD22 - DD24 - К155ЛА3

DD25 – DD34 - К155ЛН1

DD38 – Триггер типа RS

Таблица 8.1. Основные параметры и характеристики ТТЛ

|  |  |
| --- | --- |
| Параметр, характеристика | Серия микросхемы К155 |
| Входной ток, мА, не более:  низкого уровня  высокого уровня | -1,6  0,04 |
| Время задержки распространения сигнала, нс, при:  включении  выключении | 18,5  18,5 |
| Частота переключения, МГц, не более | 15 |
| Максимальное напряжение питания, В | 6 |
| Максимальное напряжение на входе, В | 5,5 |
| Минимальное напряжение на выходе, В | -0,4 |
| Максимальная емкость нагрузки, Пф | 200 |

# *6. Определение временных характеристик 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.

Рабочая частота FР выбирается из гостированного ряда частот {F} при условии, что FР≤0,8F.

Временная диаграмма работы ОУ с учетом реальных задержек в схемах этого устройства может быть правильно построена, если учесть следующие временные соотношения:

, где  − максимальное время выборки слова из ПМП;

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

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

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

; , где  − время срабатывания регистра АМК (РАМК) в случае принудительной адресации МК,  − время срабатывания счетчика АМК (СчАМК) при естественной адресации;

, где − цикл выборки слова из ПЗУ, на основе которого построена ПМП.

*Воспользовавшись справочной литературой получил такие значения:*

= 65 нс



= 40 нс



= 35 нс



Значение  выбирается в предположении, что самая длительна операция это операция СчАК:=СчАК+2, так как СчАК имеет 27 разрядов, следовательно состоит из 7-ми регистров К155ИЕ7,

 = 7\*35 = 245 нс

= 35 нс



= 65 + 40 + 35 +245 +35 = 420 нс

Рассчитаем значение тактовой частоты F=1/TОУ = 0,0024 \* 1000 МГц

Рабочая частота FР выбирается из гостированного ряда частот {F} при условии, что FР≤0,8F.

Fp = 0.8 \* 2.4 МГц = 1,9 МГц

По ГОСТу самая приближённая частота Fp = 1 МГц

*На рисунке 6.1 приведена временная диаграмма процесса исполнения МК*

Рисунок 6.1 – Временная диаграмма исполнения МК (МКi 🡪 МКj)



# *Заключение*

В ходе курсового проектирования было спроектировано простейшее операционное устройство, реализующее систему из пяти команд. Была разработана его структурная схема. Для управляющего автомата была построена микропрограмма его функционирования и спроектирована его принципиальная схема. Были рассчитаны временные характеристики синтезированного устройства.

Это операционное устройство обеспечивает все пункты поставленной задачи на курсовое проектирование, указанное в пункте "Постановка задачи" данной пояснительной записки.

В ходе курсового проектирования были закреплены основные разделы курсов «Архитектура ЭВМ» и «Цифровые ЭВМ», приобретены навыки проектирования узлов ЭВМ и изготовление соответствующей конструкторской документации.

***Библиографический список***

1. Методические указания к курсовому проектированию по курсу “Теория и проектирование ЦВМ”.
2. Шило В.Д. Популярные цифровые микросхемы: Справочник. – М.: Радио и связь, 1988.-352 с.
3. «Параметры интегральных микросхем 155 серии» -http://www.chipinfo.ru/dsheets/ic/155/index.html
4. Цифровые и аналоговые микросхемы: справочник/ С.В. Якубовский [и др.] ; под ред. С.В. Якубовского – М.: Радио и связь, 1989. – 496 с.