Федеральное государственное автономное

образовательное учреждение высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра Вычислительной техники

**КУРСОВОЙ ПРОЕКТ**

**Специализированный процессор**

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель: |  | \_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Постников А.И.  инициалы, фамилия |
| Студент: | ЗКИ21-07Б 171940587  номер группы, зачетной книжки | \_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Гладких Д.А.  инициалы, фамилия |

Красноярск 2023

**РЕФЕРАТ**

Данный курсовой проект «Специализированный процессор» посвящен разработке управляющего и операционного автоматов предназначенных для выполнения арифметических действий с фиксированной точкой двоичных чисел, представленных в обратном коде.

Данная пояснительная записка содержит 28 страниц с иллюстрациями, таблицами и формулами и 1 лист графического материала (функциональная схема специализированного процессора).

ОПЕРАЦИОННЫЙ АВТОМАТ, УПРАВЛЯЮЩИЙ АВТОМАТ, ЛОГИЧЕСКИЙ ЭЛЕМЕНТ, РЕГИСТР, СУММАТОР, МУЛЬТИПЛЕКСОР, ГРАФ-СХЕМА, МИКРОКОМАНДА, ФУНКЦИОНАЛЬНАЯ СХЕМА, СПЕЦИАЛИЗИРОВАННЫЙ ПРОЦЕССОР, НОК, РАЗРЯД.

Цель курсового проекта – закрепление основных теоретических положений предмета, приобретение навыков практического решения технических задач логического проектирования узлов и блоков ЭВМ.

В процессе проектирования были решены следующие основные задачи:

* Определена структура спецпроцессора;
* Выбраны необходимые логические элементы для создания;
* Разработана структура операционного и управляющего автоматов;
* Определен перечень управляющих и осведомительных сигналов;
* Разработана граф-схема микропрограммы;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

КП – 09.03.01 ПЗ

Разраб.

Гладких Д.А.

Провер.

Постников А.И.

Реценз.

Н. Контр.

Утверд.

Специализированный

процессор

Записка пояснительная

Лит.

Листов

28

ВТ

**СОДЕРЖАНИЕ**

Содержание 3

ВВЕДЕНИЕ 4

Постановка задачи 5

1. Разработка операционного автомата спецпроцессора 6

1.1 Анализ поставленной задачи 6

1.2 Структура операционного автомата 7

1.3 Элементы и узлы операционного автомата 8

1.4 Условия и схемы выполнения операций 11

1. Разработка закодированной граф – схемы машинного алгоритма выполнения арифметических операций 13

2.1 Список используемых входных сигналов X 13

2.2 Список используемых управляющих сигналов Y 13

2.3 Граф – схема микропрограммы 14

2.4 Словесное описание граф – схемы микропрограммы 15

2.5 Таблица работы микропрограммы 17

1. Разработка функциональной схемы управляющего автомата микросхемы специализированного процессора, выполняющего заданную арифметическую операцию 19

3.1 Разработка структуры управляющего автомата 19

3.2 Разработка микропрограммы выполнения заданной операции 23

1. Условное графическое обозначение спецпроцессора 24

Заключение 26

Список используемых источников 27

**ВВЕДЕНИЕ**

Процессор ЦВМ или другое операционное устройство, выполняющее операции над словами информации, можно разделить на две части операционный (ОА) и управляющий (УА) автоматы.

Любая команда, операция или процедура, выполняемая в операционном

автомате, описывается некоторой микропрограммой и реализуется за несколько тактов, в каждом из которых выполняется одна или несколько микроопераций.

Интервал времени, отводимый на выполнение микрооперации, называется рабочим тактом или просто тактом цифрового устройства. Если все такты имеют одну и ту же длину, то она устанавливается по самой продолжительной микрооперации.

**ПОСТАНОВКА ЗАДАЧИ**

1. Разработать функциональную схему специализированного процессора, состоящего из операционного и управляющего автоматов, выполняющего заданные арифметические операции над числами.

59 вариант, задание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Операция | Условия | Код | КА | УА |
| 1.5А + 2.75А | A < 0, B > 0, |A|< 0,5, |B|<0.25 | нок | 27 | Е |
| 3,5A – 1,25B | A > 0, B < 0, |A|< 0,25, |B|<0.5 |

Используемые сокращения:

нок - немодифицированный обратный код;

|A| – машинное изображение в формате с фиксированной точкой абсолютного значения числа А;

|B| – машинное изображение в формате с фиксированной точкой абсолютного значения числа А;

Е – управляющий автомат с программируемой логикой на ПЗУ с естественной адресацией;

КА – масштабный коэффициент;

Вариант исходных чисел для проверки работы спецпроцессора:

А = -42; В = 24;

2. Разработать УГО микросхемы спецпроцессора.

1. **Разработка операционного автомата спецпроцессора**

**1.1 Анализ поставленной задачи**

Согласно поставленной задаче, выполнение операций операционным автоматом должно осуществляться с помощью обратного кода. Обратный n-разрядный двоичный код положительного целого числа состоит из одноразрядного кода знака (двоичной цифры 0), за которым следует (n−1)-разрядное двоичное представление модуля числа (обратный код положительного числа совпадает с прямым кодом). Обратный n-разрядный двоичный код отрицательного целого числа состоит из одноразрядного кода знака (двоичной цифры 1), за которым следует (n−1)-разрядное двоичное число, представляющее собой инвертированное (n−1)-разрядное представление модуля числа. Следует отметить, что для изменения знака числа достаточно проинвертировать все его разряды, не обращая внимания, знаковый ли это разряд или информационный. Этим обратные коды удобны в применении. В качестве недостатка следует отметить, что в обратных двоичных кодах имеются два кода числа 0.

При проектировании необходимы следующие операции: умножение на два (сдвиг в сторону старших разрядов), деление на два (сдвиг в сторону младших разрядов) и сложение.

**1.2 Структура операционного автомата**

Структурная схема операционного автомата (рисунок 1.1) была разработана по следующим расчетам:

* Для числа A потребуется два регистра. Это связано с тем, необходимо представить число 1.5А и 3.5А. Регистр RgА1 передаёт числа А и 4А, а регистр RgА2 передаёт числа 0.5А и -0.5А. Числа 1.5А и 3.5А получаются из суммы RgA1 и RgA2.
* Для числа В потребуется три регистра. Это связано с тем, что необходимо представить числа 2.75В, 1.25В. Регистр RgВ1 передаёт числа В и 2В, регистр RgВ2 передает 0.25B, регистр RgВ3 передает 0.5B. Число 2.75В получается из суммы RgВ1 и RgВ2 и RgВ3. Число 1.25В получается из суммы RgВ1 и RgВ2.
* Для разделения операций используется четыре сумматора.   
  Сумматор SM1 предназначен для выполнения операций А + 0,5А,   
  4A - 0,5A.

Сумматор SM2 предназначен для выполнения операции 0.25В + 0.5B.

Сумматор SM3 предназначен для выполнения операции   
0.25В + 0.5B + 2B.

Сумматор SM4 предназначен для выполнения операций 1.5А + 2.75B, 3,5А + 1,25B.

* Для выбора какие значения будут поступать на сумматоры используем мультиплексоры MUX1 и MUX2.

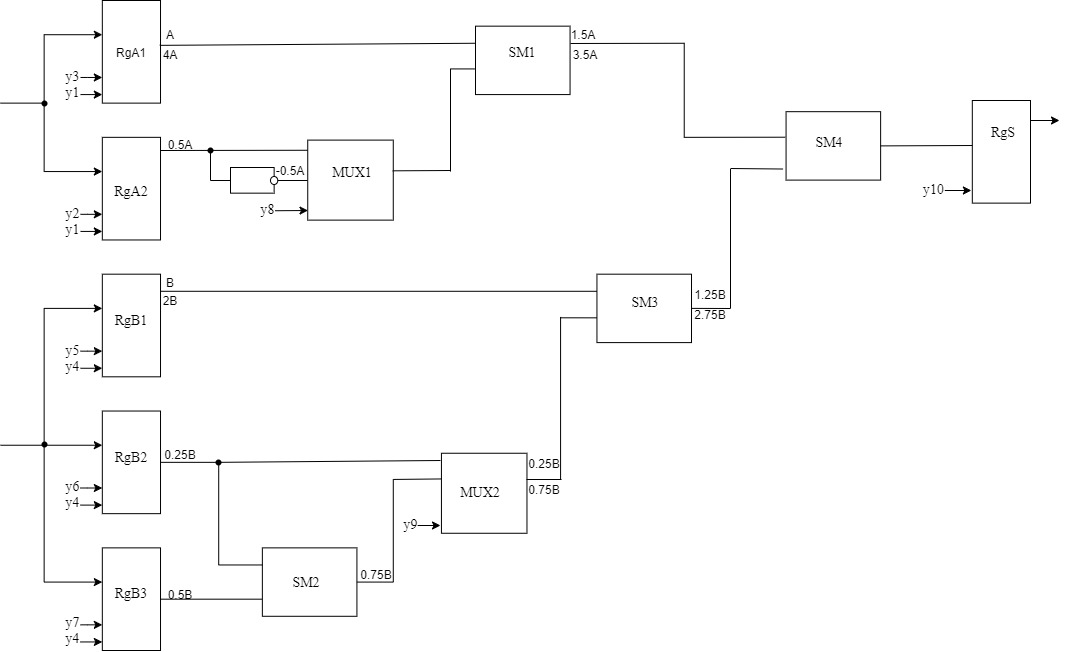


Рисунок 1.1 – Структурная схема ОА спецпроцессора

**1.3 Элементы и узлы операционного автомата**

Для запоминания числа A используются регистры A1 (рисунок 1.2) и А2 (рисунок 1.3). Регистр А1 имеет входы сдвига в сторону старших разрядов для получения числа 4А. Регистр А2 имеет входы сдвига в сторону младших разрядов для получения числа 0.5А. Для запоминания числа B используются регистры B1 (рисунок 1.4), B2 (рисунок 1.5), B3 (рисунок 1.6). Регистр В1 имеет входы сдвига в сторону старших разрядов для получения числа 2В. Регистр В2 имеет входы сдвига в сторону младших разрядов для получения числа 0.25В. Регистр В3 имеет входы сдвига в сторону младших разрядов для получения числа 0.5В. Для запоминания результата – регистр S (рисунок 1.7).

|  |  |  |  |
| --- | --- | --- | --- |
| D:\Загрузки\Копия ptca (3).jpg |  | D:\Загрузки\Копия ptca (6).jpg | D:\Загрузки\Копия ptca (8).jpg |
| Рисунок 1.2 – регистр RgA1 | Рисунок 1.3 – регистр RgA2 | Рисунок 1.4 – регистр RgB1 | Рисунок 1.5 –  регистр RgB2 |
| D:\Загрузки\Копия ptca (9).jpg | D:\Загрузки\Копия ptca (10).jpg |  |
| Рисунок 1.6 – регистр RgB3 | Рисунок 1.7 – регистр RgS |  |
|  |  |  |

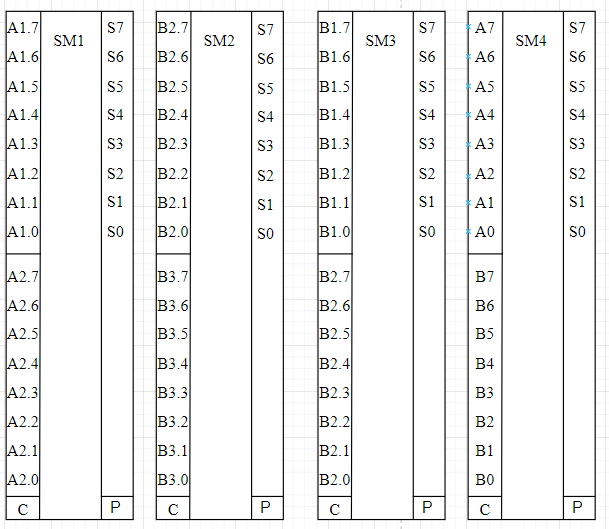
Данные регистры позволяют:

* При подаче на вход *LD* логической единицы, число, установленное на входах *D0 – D7* записывается в регистр.
* При подаче на входы «» и «» логической единицы происходит сдвиг в сторону младших или старших разрядов, соответственно.
* На входах *Q0 – Q7* находится число, содержащееся в регистре.

Мультиплексор MUX1 (рисунок 1.8) используется для выбора между 0.5А и -0.5А в зависимости от того какая операция выполняется. Мультиплексор MUX1 (рисунок 1.9) используется для выбора между сумматором SM2 и числом 0.25B. Данные мультиплексоры подают на выходы Q0-Q7 данные с входов 1.0-1.7, если на вход A подана логическая единица, в противном случае подаются данные с входов 0.0-0.7.

|  |  |  |
| --- | --- | --- |
| D:\Загрузки\Копия ptca (11).jpg |  | D:\Загрузки\Копия ptca (12).jpg |
| Рисунок 1.8 –  мультиплексор MUX1 |  | Рисунок 1.9 –  мультиплексор MUX2 |

Для выполнения операций используются сумматоры. В данном случае используется четыре сумматора. Сумматоры SM1, SM2 и SM3 представлены на рисунке 1.10.



|  |
| --- |
| Рисунок 1.10 – сумматоры SM1, SM2, SM3 и SM4 |
|  |

На входы A0-A7, B0-B7 подаются два числа для суммирования, результат суммирования будет на выходах S0-S7. Вход C – вход входного переноса. Выход P – выход выходного переноса.

**1.4 Условия и схемы выполнения операций**

Проверяются два условия, каждому из которых присвоен свой осведомительный сигнал.

Условия поставленной задачи:

1. A < 0, B > 0, |A| < 0.5, |B| <0.25 осведомительный сигнал X2
2. A > 0, B < 0, |A| < 0.25, |B| <0.5 – осведомительный сигнал X3

Выразим допустимые значения чисел, при учёте условий, указанных выше. Данные выведены в таблице 1.1 для большего удобства в работе с полученными условиями. Если значения на разрядах не подходят условию, проверяется следующее. Если ни одно из условий не выполняется, то выводится сообщение об ошибке ввода.

Таблица 1.1 – проверка условий выполнения операций

|  |  |  |
| --- | --- | --- |
| Операция | Условия | Комментарии |
| 1.5A + 2.75B | A < 0 | Знаковый разряд - 1, также есть хотя бы один ноль |
| B > 0 | Знаковый разряд - 0, также есть хотя бы одна единица |
| |A| < 0.5 | один старший разряд равен 1, 1.1\*\*\*\*\*\* |
| |B| <0.25 | два старших разряда равны 1, 0.11\*\*\*\*\* |
| 3.5A - 1.25B | A > 0 | Знаковый разряд - 0, также есть хотя бы одна единица |
| B < 0 | Знаковый разряд - 1, также есть хотя бы один ноль |
| |A| < 0.25 | два старших разряда равны 1, 0.11\*\*\*\*\* |
| |B| <0.5 | один старший разряд равен 1, 1.1\*\*\*\*\*\* |

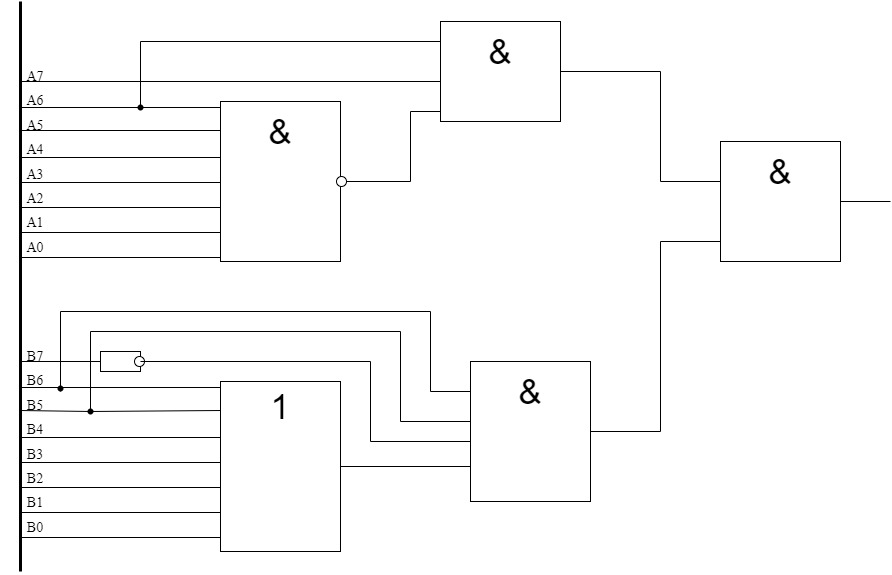


Рисунок 1.13 – проверка условия для операции 1.5A + 2.75B

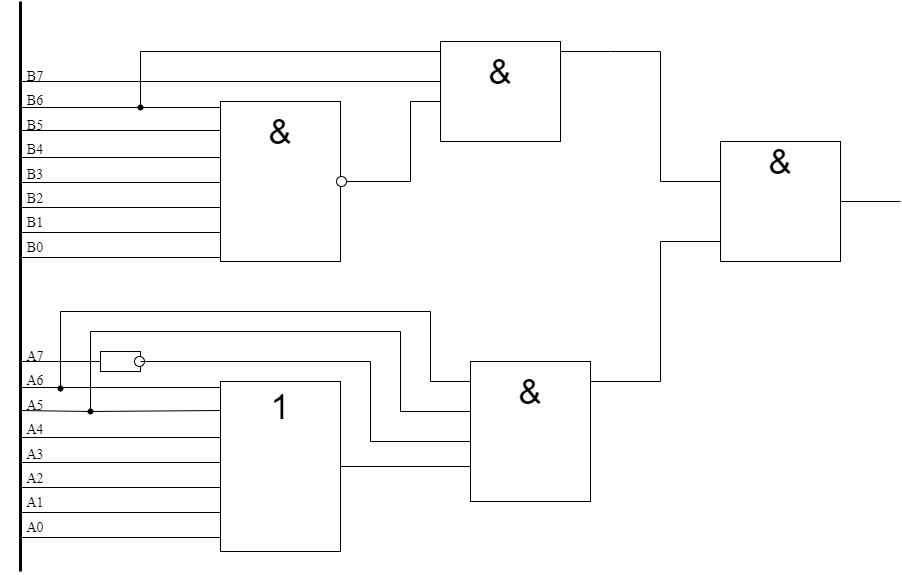


Рисунок 1.14 – проверка условия для операции 3.5A + 1.25B

**2 Разработка закодированной граф – схемы машинного алгоритма выполнения заданных арифметических операций**

* 1. **Список используемых входных сигналов Х**

Для граф – схемы алгоритма выполнения заданных арифметических операций используются следующие осведомительные сигналы:

Х1 – сигнал о наличии на входах чисел A и В

X2 – сигнал о выполнении условия A < 0, B > 0, |A| < 0.5, |B| <0.25

X3 – сигнал о выполнении условия A > 0, B < 0, |A| < 0.25, |B| <0.5

X4 – сигнал проверки переполнения разрядной сетки

* 1. **Список используемых управляющих сигналов Y**

Управляющие сигналы:

– запись чисел А, B в соответствующие регистры;

– сдвиг вправо на один разряд числа A, хранящего в регистре RgA2;

– сдвиг влево на два разряда числа A, хранящего в регистре RgA1;

– сдвиг влево на один разряд числа В, хранящегося в регистре RgB1;

– сдвиг вправо на два разряда числа В, хранящегося в регистре RgB2;

– сдвиг вправо на один разряд числа В, хранящегося в регистре RgB3;

– сигнал для выбора адреса в мультиплексоре MUX1;

Y8 – сигнал для выбора адреса в мультиплексоре MUX2;

Y9– сигнал записи результатов в регистр RgS;

Y10 - сигнал о том, что числа поступившие на входы А и В не соответствуют требуемым условиям (ER);

Y11 – сигнал выдачи результата (RD);

Y12 – сигнал переполнения (OF);

**2.3 Граф – схема микропрограммы**

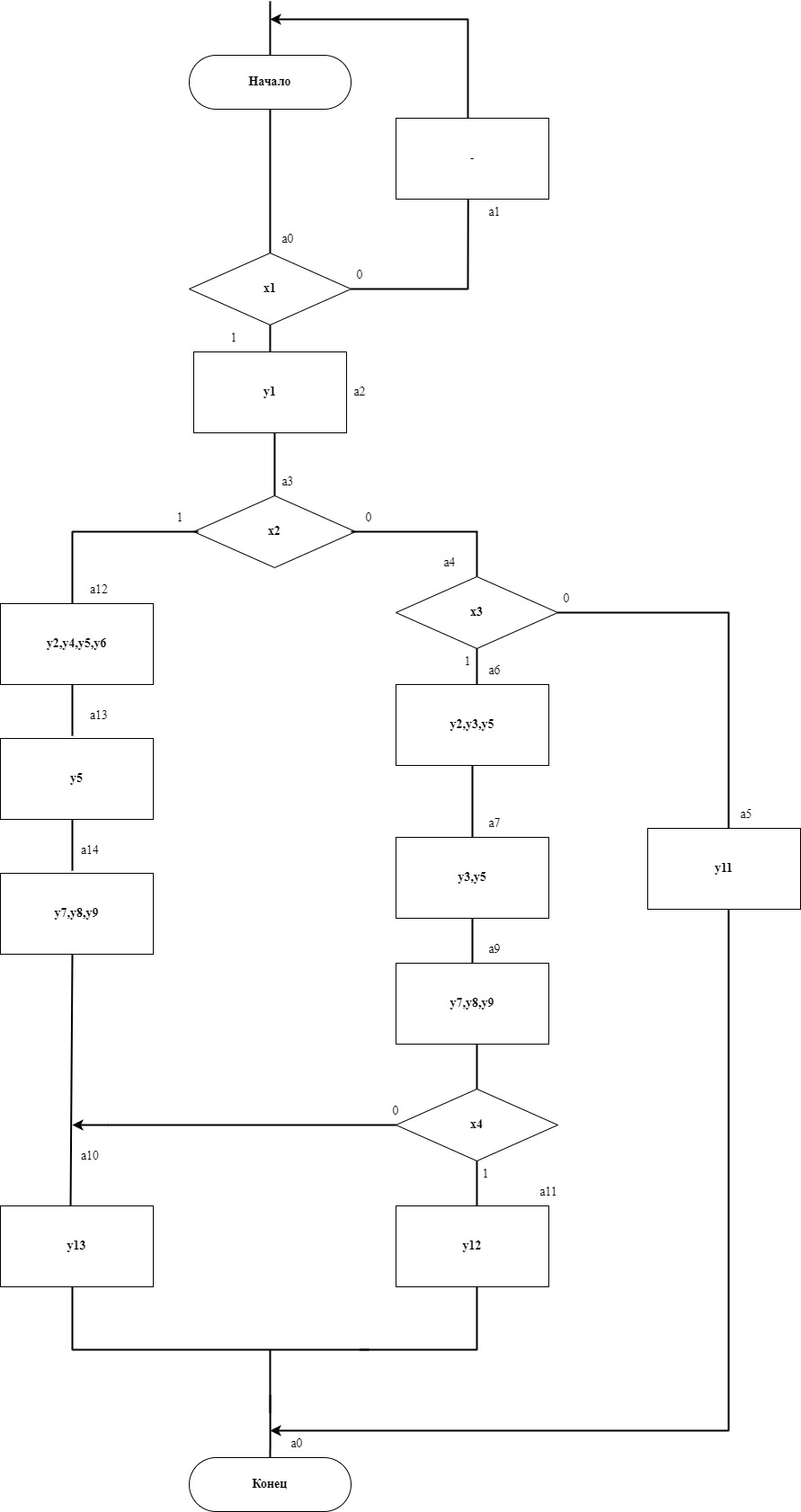


Рисунок 2.1 – граф-схема микропрограммы

**2.4** **Словесное описание граф – схемы микропрограммы**

|  |  |
| --- | --- |
| a0 | Проверка осведомительного сигнала X1. При X1 = 1 переход в а2, иначе а1. |
| a1 | Переход в а0. |
| a2 | Запись чисел А, B с помощью сигнала y1. Переход в состояние а3. |
| a3 | Проверка осведомительного сигнала X2. При X2 = 1 переход в состояние а12, иначе а4. |
| a4 | Проверка осведомительного сигнала X3. При X3 = 1 переход в состояние а6, иначе а5. |
| a5 | Выдача сигнала об ошибке с помощью сигнала y11. Переход в а0. |
| a6 | Сдвиг влево на разряд числа A, хранящегося в регистре RgA1 с помощью сигнала y3. Сдвиг вправо на разряд числа A, хранящегося в регистре RgA2 с помощью сигнала y2. Сдвиг вправо на разряд числа B, хранящегося в регистре RgB2 с помощью сигнала y5. Переход в а7. |
| a7 | Сдвиг влево на разряд числа A, хранящегося в регистре RgA1с помощью сигнала y3. Сдвиг вправо на разряд числа B, хранящегося в регистре RgB2 с помощью сигнала y5. Переход в а8. |
| a8 | Выбор адресов в мультиплексорах с помощью сигналов y7, y8. Запись результата в регистр RgS с помощью сигнала y9, переход в а9. |
| a9 | Проверка осведомительного сигнала X4. При X4 = 1, иначе а10. |
| a10 | Выдача результата с помощью сигнала y13, переход в а0. |
| a11 | Выдача сигнала об ошибке с помощью сигнала y12. Переход в а0. |
| a12 | Сдвиг вправо на разряд числа A, хранящегося в регистре RgA2 с помощью сигнала y2. Сдвиг вправо на разряд числа B, хранящегося в регистре RgB1 с помощью сигнала y4. Сдвиг вправо на разряд числа B, хранящегося в регистре RgB2 с помощью сигнала y5. Сдвиг вправо на разряд числа B, хранящегося в регистре RgB2 с помощью сигнала y6, переход в состояние а13 |
| a13 | Сдвиг вправо на разряд числа B, хранящегося в регистре RgB2 с помощью сигнала y5, переход в состояние а14 |
| a14 | Выбор адресов в мультиплексорах с помощью сигналов y7, y8. Запись результата в регистр RgS с помощью сигнала y9, переход в а9.  Выбор адресов в мультиплексорах с помощью сигналов y8, y9. Запись результата в регистр RgS с помощью сигнала y10, переход в а10. |

**2.5 Таблица работы микропрограммы**

Проверочные числа: А = -42, В = 24. Представим их в двоичной системе счисления в немодифицированном обратном коде (далее – нок):

Число А – отрицательное. Запишем его в обратном коде.

Число B – отрицательное. Запишем его в обратном коде.

Таблица 2.1 – потактовая работа автомата

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | RgA1 | RgA2 | RgB1 | RgB2 | RgB3 | RgS |
| 1 | \*.\*\*\*\*\*\*\* | \*.\*\*\*\* \*\*\* | \*.\*\*\*\*\*\*\* | \*\*.\*\*\*\*\*\* | \*\*.\*\*\*\*\*\* | \*\*.\*\*\*\*\*\* |
| 2 | 1.1010101 | 1.1010101 | \*.\*\*\*\*\*\*\* | \*.\*\*\*\*\*\*\* | \*.\*\*\*\*\*\*\* | \*.\*\*\*\*\*\*\* |
| 3 | 1.1010101 | 1.1010101 | 0.0011000 | 0.0011000 | 0.0011000 | \*.\*\*\*\*\*\*\* |
| 4 | 1.1010101 | 1.1010101 | 0.0011000 | 0.0011000 | 0.0011000 | \*.\*\*\*\*\*\*\* |
| 5 | 1.1010101 | 1.1010101 | 0.0011000 | 0.0011000 | 0.0011000 | \*.\*\*\*\*\*\*\* |
| 6 | 1.1010101 | 1.1101010 | 0.0110000 | 0.0001100 | 0.0001100 | \*.\*\*\*\*\*\*\* |
| 7 | 1.1010101 | 1.1101010 | 0.0110000 | 0.0000110 | 0.0001100 | \*.\*\*\*\*\*\*\* |
| 8 | 1.1010101 | 1.1101010 | 0.0110000 | 0.0000110 | 0.0001100 | 0.0000011 |

Словесное описание потактовой работы автомата:

1. Опрос X1
2. Запись A в RgA
3. Запись B в RgB
4. Опрос X2
5. Опрос X3
6. Сдвиг RgA2, RgВ2, RgВ3 в сторону младших разрядов, RgВ1 в сторону старших разрядов
7. Сдвиг RgВ2 в сторону младших разрядов
8. Запись числа 1.5А + 2.75В в регистр RgS

Учитывая условия задачи, программа должна выполнить операцию -1.5А + 2.75В

Если A = -42, а B = 24, то 1.5А + 2.75В =1.5\*(-42) + 2.75\*24 = 3

Переведём 3 в обратный двоичный код:

Полученный результат преобразования совпадает со значением, полученным в регистре конечного значения RgS.

1. **Разработка функциональной схемы управляющего автомата микросхемы специализированного процессора, выполняющего заданную арифметическую операцию**
   1. **Разработка структуры управляющего автомата**

При естественной адресации отпадает необходимость во введении одного из адресных полей в микрокоманды. Если микрокоманды следуют в естественном порядке, то процесс адресации реализуется счетчиком адреса микрокоманды, но если опрашивается осведомительный сигнал, то адрес следующей микрокоманды при x = 1 берется из поля A1 текущей микрокоманды, иначе выполнятся следующая по счету микрокоманда.

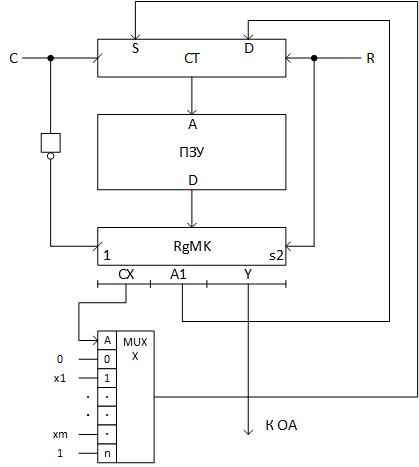


Рисунок 3.1 – структура УА с естественной адресацией для разветвляющейся ГСМ

Автомат состоит из следующих элементов:

* 1. CT – регистр-счетчик с параллельной записью информации.
  2. ПЗУ – постоянное запоминающее устройство. Предназначено для хранения всех микрокоманд. На вход А подается адрес, на котором находится нужная микрокоманда, на выходе D появляется микрокоманда по заданному адресу A.
  3. RgMK – регистр, который хранит текущую микрокоманду. Вход R нужен для обнуления значений.
  4. MUXX – мультиплексор, отвечающий за распределение выбора осведомительного сигнала. На вход А подается адрес осведомительного сигнала х, на выходе появляется значение осведомительного сигнала х по адресу А.

Микрокоманда в автомате с естественной адресацией содержит 3 поля:

* CX – поле управления выбором опрашиваемого осведомительного сигнала х.
* A1 – поле с адресом микрокоманды в случае, если осведомительный сигнал принимает значение 1.
* Y – поле управляющих сигналов.

Разрядность определяется с помощью следующих формул:

;

;

,

где – количество операторных вершин, равное 10;

– количество условных вершин, равное 4;

– количество входных сигналов, равное 4;

– количество дополнительно введенных вершин, равное 1;

k – число управляющих сигналов, равное 12.

Разрядность микрокоманды:

.

Таким образом, необходим ПЗУ с организацией . Это означает, что ПЗУ будет иметь 4 адресных входа, что позволит обратиться к 16 ячейкам. Также ПЗУ будет иметь 19 выходов для вывода микрокоманды (рисунок 3.2).

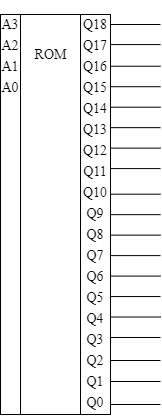


Рисунок 3.2 – УГО ПЗУ

УГО регистра RgMK и УГО регистра-счетчика CT представлено на рисунке 3.3:

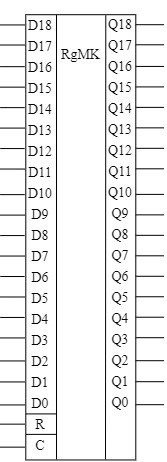
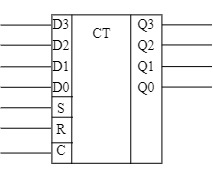
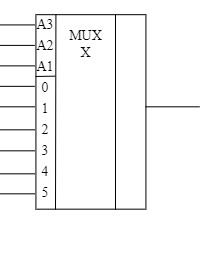
  

Рисунок 3.3 – УГО регистра RgMK и регистра-счетчика CT и мультиплексор MUXX

У регистра RgMK D0 – D18 – входы микрокоманды с ПЗУ, Q0 – Q21 – выходы микрокоманды, R – вход сброса, C – вход синхронизации.

У регистра-счетчика CT D0 – D3 – входы адреса следующей микрокоманды, S – вход параллельной записи, C – сход синхронизации, R – вход сброса, Q0 – Q4 – выходы, содержащие микрокоманду.

Мультиплексор, отвечающий за распределение выбора осведомительного сигнала, представлен на рисунке 3.3. Здесь, на входы A0 – A2 подается адрес осведомительного сигнала x, на выходе появляется значение данного сигнала. Входы 0-5 – входы осведомительных сигналов.

**3.2 Разработка микропрограммы выполнения заданной операции**

Таблица 3.1 – кодировка осведомительных сигналов

|  |  |  |  |
| --- | --- | --- | --- |
| Вых. MUX X | A2 | A1 | A0 |
| «0» | 0 | 0 | 0 |
| x1 | 0 | 0 | 1 |
| x2 | 0 | 1 | 0 |
| x3 | 0 | 1 | 1 |
| x4 | 1 | 0 | 0 |
| «1» | 1 | 0 | 1 |

Таблица 3.2 – микропрограмма

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Адрес ПЗУ | | | | | CX | | | A1 | | | | Y | | | | | | | | | | | |
|  | 4 | 3 | 2 | 1 | 0 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| A0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | \* | \* | \* | \* | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | \* | \* | \* | \* | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A4 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A5 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A6 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A7 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | \* | \* | \* | \* | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | \* | \* | \* | \* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| A10 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | \* | \* | \* | \* | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| A11 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | \* | \* | \* | \* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| A12 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A13 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| A14 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

1. **Условное графическое обозначение спецпроцессора**

Для обозначения спецпроцессора будет использоваться условно графическое обозначение, указанное на рисунке 4.1.

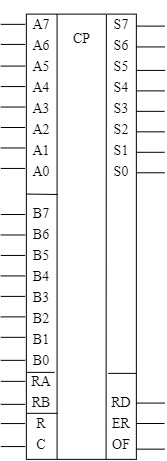


Рисунок 4.1 – УГО спецпроцессора

Микросхемы разработанного специализированного спецпроцессора имеет 31 информационный вывод.

Назначение входов спецпроцессора:

* A0-A7 – число A (A7 – знаковый разряд, A6 – старший значащий разряд, A0 – младший значащий).
* B0-B7 – число B (B7 – знаковый разряд, B6 – старший значащий разряд, B0 – младший значащий разряд).
* C – тактовые синхроимпульсы.
* R – сброс.
* RA – наличие на информационных входах спецпроцессора операнда А
* RB – наличие на информационных входах спецпроцессора операнда B

Назначение выходов спецпроцессора:

* S0-S7 – результат операций (S7 – знаковый разряд, S6 - старший значащий разряд, S0 – младший значащий разряд).
* ER – сигнал о входных данных, не удовлетворяющих ни одному из условий.
* RD – готовность результата.
* OF –переполнения разрядной сетки.

**ЗАКЛЮЧЕНИЕ**

При выполнении курсового проекта были закреплены навыки разработки операционного и управляющего автоматов. Также были закреплены знания по дисциплине «Прикладная теория цифровых автоматов».

В ходе работы над курсовым проектом было несколько возможных решений по выполнению поставленной задачи. Например, для каждой операции из условия сделан сумматор, и исходя из условий выбирается значение из нужного сумматора. Однако, это не является единственным решением данной задачи. Эту задачу можно было решить, используя один сумматор, подавая на него разные значения в зависимости от условия.

Поставленная задача решена полностью. Спроектированный специальный процессор выполняет операции, указанные в задаче, согласно условиям.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Постников, А. И. Основы теории цифровых автоматов : учеб. пособие / А. И. Постников. – Красноярск : КГТУ, 2000. – 296 с.
2. ГОСТ 2.743-91. Обозначения условные графические в схемах. Элементы цифровой техники.
3. Постников, А. И. Теория автоматов. Управляющие автоматы с программируемой логикой: методические указания к лабораторным работам для студентов направления подготовки 230100.62 – "Информатика и вычислительная техника" и специальности 230101.65 – "Вычислительные машины, комплексы, системы и сети" [Электронный ресурс] / сост. А.И. Постников. – Красноярск: Сиб. федер. ун-т, 2011. – 48 с.
4. СТО 4.2-07-2014 "Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности"
5. «Студопедия. Понятие операционного и управляющих автоматов» [Электронный ресурс] – Электрон. дан. URL: http://studopedia.ru/2\_34646\_ponyatie-operatsionnogo-i-upravlyayushchih-avtomatov.html (дата обращения 28.05.2017).