

**Федеральное государственное автономное
образовательное учреждение
высшего профессионального образования
"СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"**

Институт космических и информационных технологий

Кафедра Вычислительной техники

КУРСОВОЙ ПРОЕКТ
по дисциплине "Теория автоматов"

Специализированный процессор

Руководитель

 15.12.11

доц. каф. ВТ, к.т.н. А.И. Постников

Студент гр. КИ09-06 № 031210461



15.12.11

Д.М. Иванов

Красноярск 2011


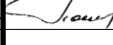
Техническое задание.

1. Разработать функциональную схему специализированного процессора состоящего из операционного и управляющего автоматов предназначенного для выполнения заданной операции.

Задание:

Арифметическая операция:	Сложение с плавающей точкой
Код поступления мантиссы А	Дополнительный
Код поступления порядка А:	Прямой
Код поступления мантиссы В:	Дополнительный
Код поступления порядка В:	Обратный
Код обработки:	Обратный
Код выдачи результата (мантисса и порядок):	Дополнительный
Число разрядов:	4
Число знаковых разрядов:	2
Тип управляющего автомата:	С программируемой логикой и принудительной адресацией.
Проверочные числа	$A=-1, B=-3$

2. Разработать УГО микросхемы спецпроцессора

					КП – 230101.65 – 031210461 ПЗ			
Изм	Лист	№ докум.	Подпись		Специализированный процессор Пояснительная записка	Лит.	Лист	Листов
Выполнил	Иванов Д.М.			15.12.11				
Пров.	Постников А.И.			15.12.11			2	41
Н. контр.						КИ09-06		

РЕФЕРАТ

Настоящий курсовой проект "Специализированный процессор" посвящён разработке функциональной схемы специализированного процессора предназначенного для выполнения арифметической операции сложения с плавающей точкой двоичных чисел представленных в модифицированном коде (исходные операнды и результат представлены в различных кодах в соответствии с техническим заданием).

Данная пояснительная записка к курсовому проекту содержит 41 страницу текста с иллюстрациями, таблицами и формулами, 9 использованных источников и 1 лист графического материала (функциональная схема специализированного процессора).

ОПЕРАЦИОННЫЙ АВТОМАТ, УПРАВЛЯЮЩИЙ АВТОМАТ, ЛОГИЧЕСКИЙ ЭЛЕМЕНТ, РЕГИСТР, СУММАТОР, МУЛЬТИПЛЕКСОР, ПОСТОЯННОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО, ГРАФ-СХЕМА МИКРОПРОГРАММЫ, МИКРОКОМАНДА, КОД ОПЕРАНДОВ, МИКРОПРОГРАММА, ФУНКЦИОНАЛЬНАЯ СХЕМА, СПЕЦИАЛИЗИРОВАННЫЙ ПРОЦЕССОР.

Цель курсового проекта – разработка специализированного процессора.

Задачи, решённые в процессе курсового проектирования:

- определена структура спецпроцессора состоящего из операционного и управляющего автоматов;
- разработана структура операционного автомата;
- выбраны необходимые логические элементы, регистры, сумматоры, мультиплексоры и др., определены связи между ними;
- выбрана структура управляющего автомата с программируемой логикой и принудительной адресацией;
- определены разрядности полей микрокоманды и разрядность микрокоманды в целом;
- определён перечень управляющих и осведомительных сигналов;
- разработаны граф-схема микропрограммы и микропрограмма выполнения заданной операции, предназначенная для занесения в ПЗУ.

Разработанный в процессе работы над курсовым проектом специализированный процессор позволяет выполнять операцию арифметического сложения с плавающей точкой двоичных чисел представленных в модифицированном коде.

СОДЕРЖАНИЕ

Введение	5
1 Разработка операционного автомата специализированного процессора	7
1.1 Сложение чисел с ПТ	7
1.2 Структура операционного автомата спецпроцессора	9
1.3 Разработка и описание функциональной схемы спецпроцессора. Элементы и узлы операционного автомата спецпроцессора	10
2 Разработка граф-схемы микропрограммы	22
2.1 Представление алгоритма в виде граф-схемы	22
2.2 Осведомительные сигналы X используемые в спецпроцессоре	24
2.3 Управляющие сигналы Y используемые в спецпроцессоре	25
2.4 Граф-схема микропрограммы	25
2.5 Словесное описание граф-схемы микропрограммы	25
2.6 Таблица работы операционного автомата	28
3 Разработка функциональной схемы управляющего автомата специализированного процессора	30
3.1 Управляющий автомат с программируемой логикой и принудительной адресацией	30
3.2 Разработка функциональной схемы управляющего автомата спецпроцессора	31
3.3 Разработка микропрограммы выполнения заданной операции	37
4 Условное графическое обозначение микросхемы спецпроцессора	38
Заключение	40
Список использованных источников	41

ВВЕДЕНИЕ

При проектировании различных узлов и устройств цифровой вычислительной техники одной из основных задач является задача рационального конструирования, синтеза таких схем.

В процессе синтеза сложного автомата или устройства представляющего собой совокупность нескольких автоматов решаются следующие задачи [1]:

- определяется объём памяти и устанавливаются те изменения состояний памяти под воздействием входных сигналов;
- осуществляется выбор логических и запоминающих элементов;
- составляются канонические уравнения, сводящие общую задачу синтеза автомата к синтезу схем из логических элементов;
- проводится комбинационный синтез;
- преобразование и дополнение построенных схем с целью обеспечения надёжности их функционирования.

Процессор ЦВМ или другое операционное устройство, выполняющее операции над словами информации можно рассматривать как состоящее из двух блоков или автоматов – операционного и управляющего [3, 4].

Процессор выполняет заданное множество операций F над входными словами D с целью вычисления результатов R . Каждая операция из множества операций F возбуждается соответствующей командой из множества команд K .

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.

Целевая установка курсового проектирования – закрепление основных теоретических положений предмета, приобретение навыков практического решения технических задач логического проектирования узлов и блоков ЭВМ, закрепление и углубление знаний в части проектирования конечных автоматов и устройств на их основе, а также формирование соответствующих практических умений и навыков по разработке алгоритмов выполнения арифметических операций в ЦВМ, разработке структуры операционных автоматов и управляющих автоматов, реализующих эти алгоритмы. Знания, полученные при изучении

дисциплины "Теория автоматов" и в процессе выполнения данного курсового проекта, в дальнейшем являются базой для изучения специализированных дисциплин; будут использованы при курсовом и дипломном проектировании на старших курсах.

Для выполнения курсового проекта необходимо знание основных разделов предмета "Теория автоматов".

Поставленная цель достигается путем самостоятельной разработки алгоритмов выполнения заданной операции, структуры операционного автомата и управляющего автомата (специализированного процессора).

1 Разработка операционного автомата специализированного процессора.

1.1 Сложение чисел с ПТ

Числа, представленные в форме с ПТ, изображаются двумя частями – мантиссой и порядком [3, 4]. При операции алгебраического сложения действия, выполняемые над мантиссами и порядками, различны.

Для того чтобы сложить два числа с ПТ, необходимо выполнить следующие действия.

1. Приём исходных операндов.
2. Выравнивание порядков.
3. Сложение мантисс. Порядок результата будет равен любому из порядков чисел.
4. Нормализация результата.
5. Обнаружение переполнения разрядной сетки.
6. Выдача результата операции.

Рассмотрим подробнее каждое из вышеперечисленных действий.

Приём исходных операндов. Операнды A и B записываются во входные регистры ОА, предназначенные для хранения мантисс и порядков исходных операндов.

"Выравнивание порядков". Для операции сложения чисел необходимым условием является соответствие разрядов операндов друг другу. Значит, прежде всего, нужно уравнивать порядки, что, естественно, повлечёт за собой временное нарушение нормализации одного из слагаемых. Выравнивание порядков означает, что порядок меньшего числа надо увеличить на величину $\Delta p = |p_A - p_B|$, что означает сдвиг мантиссы меньшего числа вправо на количество разрядов, равное Δp . Порядок большего числа уменьшать до меньшего нельзя, т.к. в этом случае потребуются сдвиг мантиссы влево, что приведёт к появлению целой части числа.

Для получения Δp порядков, представленных в ОК, все разряды порядка числа B , включая знаковые, необходимо проинвертировать и сложить на ДСОК с порядком числа A . Знак результата $Sg\Delta p$ укажет какой из порядков больше, а какой меньше. Если $Sg\Delta p = 0$, то $p_A > p_B$, а если $Sg\Delta p = 1$, то $p_A \leq p_B$. Значащие разряды будут содержать представленную в ОК величину $|p_A - p_B|$.

Данная операция производится циклически до тех пор, пока разность порядков не будет равна 0 ($p_A - p_B = 0$).

Сложение мантисс. Сложение мантисс производится аналогично сложению чисел с ФТ.

Нормализация результата. Проводится при нарушении нормализации числа – невыполнении условия

$$q^{-1} \leq |m_A| < 1. \quad (1)$$

В этом случае используется операция сдвига.

Так как условие (1) содержит два неравенства, то может быть нарушено справа или слева. Признак нарушения нормализации числа справа γ (когда величина результата равна или превышает единицу) - наличие разных значений в знаковых разрядах сумматора, т.е.

$$\gamma = Sg1 \oplus Sg2,$$

$$\gamma = \begin{cases} 0, & \text{нарушения нормализации нет;} \\ 1, & \text{нормализация нарушена.} \end{cases}$$

Признак $\gamma = 1$ указывает на необходимость сдвинуть мантиссу числа вправо на один разряд и скорректировать порядок (прибавить к нему 1).

Признак нарушения нормализации числа слева δ (когда результат по модулю меньше q^{-1}) - наличие одинаковых комбинаций в разряде переполнения и старшем значащем разряде s_1 :

$$\delta = Sg2 \oplus s_1,$$

$$\delta = \begin{cases} 1, & \text{нарушения нормализации нет;} \\ 0, & \text{нормализация нарушена} \end{cases}$$

Признак $\delta = 0$ указывает на необходимость сдвигать мантиссу числа влево до тех пор, пока δ не станет равен 1. Порядок должен быть скорректирован на число произведённых сдвигов, т.е. от значения порядка следует вычесть число произведённых сдвигов.

Обнаружение переполнения разрядной сетки. Признак $\gamma = 1$, возникший при сложении мантисс чисел с ПТ не является признаком переполнения, поскольку при нормализации мантиссу можно сдвинуть как вправо, так и влево. При нормализации результата корректируется порядок и, следовательно, может возникнуть переполнение разрядной сетки порядка.

Обнаружение переполнения разрядной сетки всегда проводится после нормализации результата, поскольку коррекция порядка может привести к его переполнению.

Выдача результата. Здесь полученные мантисса и порядок результата либо сигнал о возникновении переполнения выдаются на шину данных цифрового автомата.

1.2 Структура операционного автомата спецпроцессора

Основываясь на вышеизложенном и учитывая требования технического задания, предлагается следующая структура операционного автомата (рис. 1).

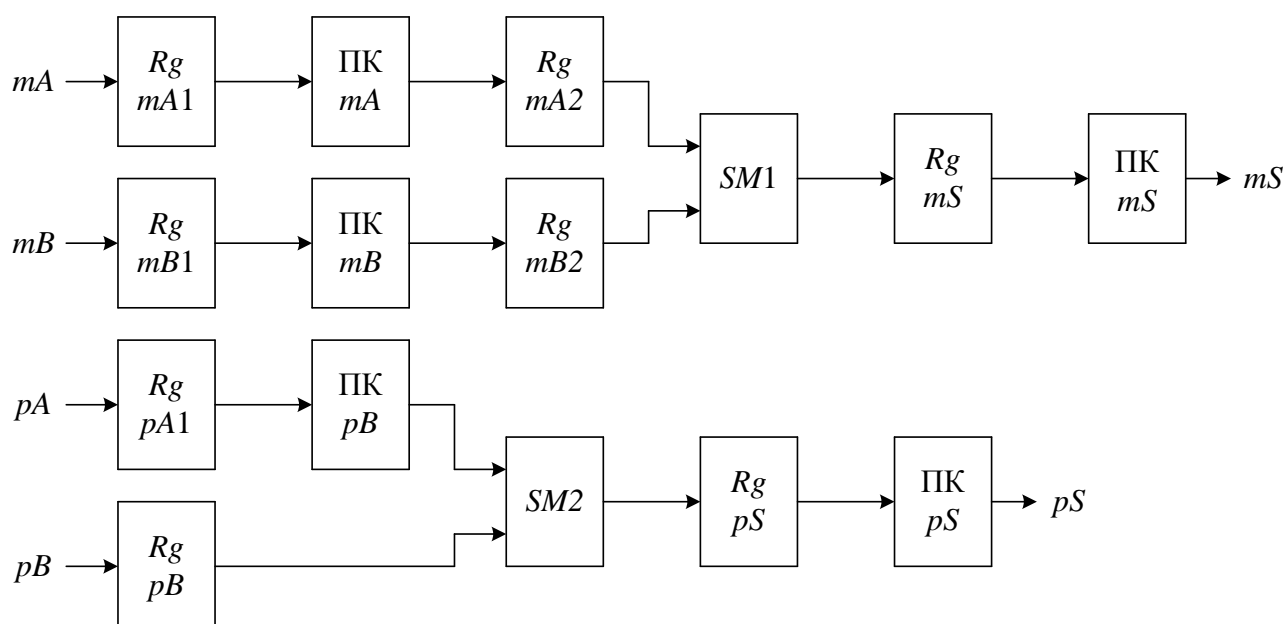


Рисунок 1 – Структурная схема операционного автомата спецпроцессора

Входные регистры $RgmA1$, $RgmB1$, $RgpA1$ и $RgpB$ предназначены для записи и хранения в них исходных операндов A и B .

Поскольку коды исходных операндов (мантиссы A , порядка A и мантиссы B) отличаются от кода обработки в схеме имеются три преобразователя кодов входных операндов $ПКmA$, $ПКmB$ и $ПКpB$, которые обеспечивают перевод в обратный код. Получившиеся в результате преобразования обратные коды мантиссы числа A и мантиссы числа B записываются в регистры $RgmA2$ и $RgmB2$, в которых они хранятся до завершения операции. Также в этих регистрах производится микрооперация сдвига необходимого при проведении выравнивания порядков.

Сумматор $SM1$ предназначен для суммирования мантисс исходных операндов. Результат суммирования мантисс (мантисса результата mS) после выполнения операции суммирования записывается в регистр мантиссы результата $RgmS$.

Сумматор $SM2$ предназначен для вычитания порядков исходных операндов в процессе их выравнивания. Единый выровненный порядок (порядок результата) записывается в регистр порядка результата $RgpS$.

Регистры $RgmS$ и $RgpS$ должны позволять выполнить нормализацию результата, т.е. $RgmS$ должен обеспечивать возможность выполнения микрооперации сдвига, а $RgpS$ – обеспечивать возможность коррекции порядка на ± 1 .

Поскольку код обработки отличен от кодов выдачи результата в схеме присутствуют преобразователи кодов результата $ПКmS$ и $ПКpS$, с выходов которых результат выполненной операции сложения двоичных чисел с плавающей точкой выдаётся пользователю.

1.3 Разработка и описание функциональной схемы спецпроцессора. Элементы и узлы операционного автомата спецпроцессора

В функциональной схеме операционного автомата используются следующие элементы и узлы.

Регистр мантиссы исходного операнда A представлен на рис. 2.

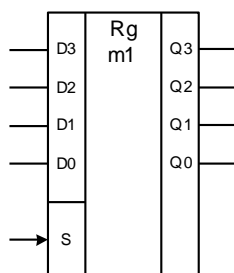


Рисунок 2 – Регистр мантиссы исходного операнда A

$Rgm1$ представляет собой 4-разрядный параллельный регистр.

$D0-D3$ – прямые входы регистра ($D2, D3$ – знаковые разряды, $D0, D1$ – значащие; причём $D3$ – знаковый разряд, $D2$ – разряд переполнения, $D1$ – старший значащий разряд, $D0$ – младший).

Предназначен для записи и хранения мантиссы A поступающей в спецпроцессор в дополнительном коде.

Имеет прямые выходы $Q0-Q3$ ($Q2, Q3$ – знаковые, $Q2, Q3$ – значащие; причём $Q3$ – знаковый разряд, $Q2$ – разряд переполнения, $Q1$ – старший значащий разряд, $Q0$ – младший). В дальнейшем при описании остальных элементов и узлов операционного автомата назначение разрядов совпадает с вышеуказанным.

Режимы работы:

При поступлении на вход S единицы производится запись с входов $D0-D3$ и информация появляется на прямых выходах $Q0-Q3$.

Регистр мантиисы исходного операнда B представлен на рис. 3.

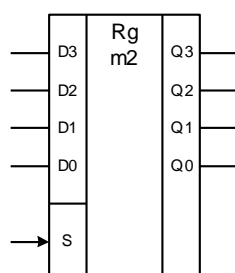


Рисунок 3 – Регистр мантиисы исходного операнда B

$Rgm2$ – 4-разрядный параллельный регистр.

$D0-D3$ – прямые входы регистра.

Предназначен для приема и хранения мантиисы операнда B в дополнительном коде.

Имеет прямые выходы $Q0-Q3$.

Режимы работы:

При поступлении на вход S логической единицы производится запись информации с входов $D0-D3$, которая появляется на прямых выходах $Q0-Q3$.

Регистр порядка исходного операнда A представлен на рис. 4.

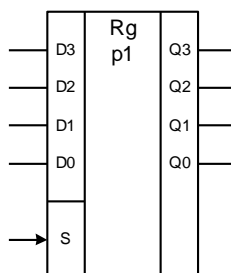


Рисунок 4 – Регистр порядка исходного операнда A

Rgp1 – 4 разрядный параллельный регистр.

D0-D3 – прямые входы регистра.

Предназначен для приема и хранения порядка числа *A* представленного в прямом коде.

Имеет прямые выходы *Q0-Q3*.

Режимы работы:

При поступлении на вход *S* логической единицы производится запись с входов *D0-D3* и информация появляется на прямых выходах *Q0-Q3*.

Регистр мантиисы исходного операнда *A* преобразованного в код обработки (в обратный код) показан на рис. 5.

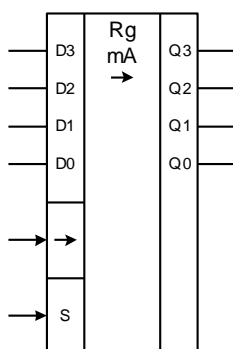


Рисунок 5 – Регистр мантиисы операнда *A* в ОК

RgmA представляет собой 4-разрядный последовательно-параллельный регистр.

D0-D3 – прямые входы регистра.

Предназначен для приема, хранения и сдвига вправо (в сторону младших разрядов) мантиисы *A* представленной в обратном коде.

Имеет прямые выходы *Q0-Q3*.

Режимы работы:

При поступлении логической единицы на вход *S* производится запись с входов *D0-D3* и информация появляется на прямых выходах *Q0-Q3*.

Так как при выравнивании порядков необходимо одновременно с коррекцией порядка производить сдвиг мантиисы, то при поступлении логической единицы на вход \rightarrow происходит модифицированный сдвиг информации хранящейся в регистре (обратного кода мантиисы числа *A*) в сторону младших разрядов.

Регистр мантиисы исходного операнда *B* преобразованного в код обработки (в обратный код) показан на рис. 6.

$RgmB$ – 4 разрядный последовательно-параллельный регистр.

$D0-D3$ – прямые входы регистра.

Предназначен для приема, сдвига вправо и хранения мантиссы B в обратном коде.

Имеет прямые выходы $Q0-Q3$.

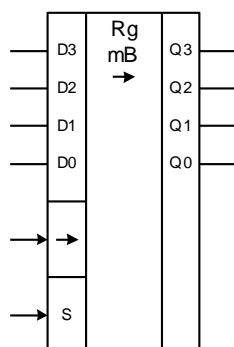


Рисунок 6 – Регистр мантиссы операнда B в ОК

Режимы работы:

При поступлении логической единицы на вход S производится запись с входов $D0-D3$ и информация появляется на прямых выходах $Q0-Q3$.

Так как при выравнивании порядков необходимо одновременно с коррекцией порядка производить сдвиг мантиссы, то при поступлении логической единицы на вход \rightarrow происходит модифицированный сдвиг информации хранящейся в регистре (обратного кода мантиссы числа B) в сторону младших разрядов.

Регистр порядка исходного операнда A преобразованного в код обработки (в обратный код) показан на рис. 7.

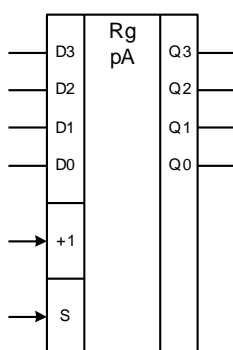


Рисунок 7 – Регистр порядка операнда A в ОК

RgpA – 4 разрядный регистр-счетчик.

D0-D3 – прямые входы регистра.

Предназначен для приема, хранения и прибавления 1 к порядку числа *A* в обратном коде.

Имеет прямые выходы *Q0-Q3*.

Режимы работы:

При поступлении на вход *S* логической единицы производится запись с входов *D0-D3* и информация появляется на прямых выходах *Q0-Q3*.

Так как при выравнивании порядков необходимо одновременно со сдвигом мантиссы производить коррекцию порядка, то при поступлении на вход +1 логической единицы происходит прибавление единицы к содержащемуся в регистре числу.

Регистр порядка исходного операнда *A* преобразованного в код обработки (в обратный код) показан на рис. 8.

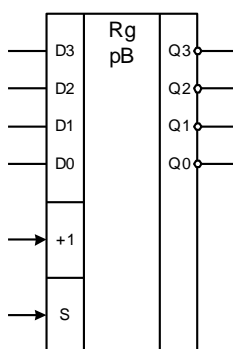


Рисунок 8 – Регистр порядка операнда *B* в ОК

RgpB – 4 разрядный регистр-счетчик.

D0-D3 – прямые входы регистра.

Предназначен для приема, прибавления 1 и хранения порядка *B* поступающего в спецпроцессор в обратном коде.

Имеет инверсные выходы *Q0-Q3*.

Режимы работы:

При поступлении на вход *S* единицы производится параллельная запись с входов *D0-D3* и информация появляется на инверсных выходах *Q0-Q3*.

Так как при выравнивании порядков необходимо одновременно со сдвигом мантиссы производить коррекцию порядка, то при поступлении на вход +1 логической единицы происходит прибавление единицы к содержащемуся в регистре числу.

Регистр мантиисы результата S представленного в коде обработке (в обратном коде) приведён на рис. 9.

$RgmS$ – 4 разрядный универсальный реверсивный регистр. $D0-D3$ – прямые входы регистра.

Регистр предназначен для приема сдвига вправо (в сторону младших разрядов) и влево (в сторону старших разрядов), хранения мантиисы S в представленной в обратном коде.

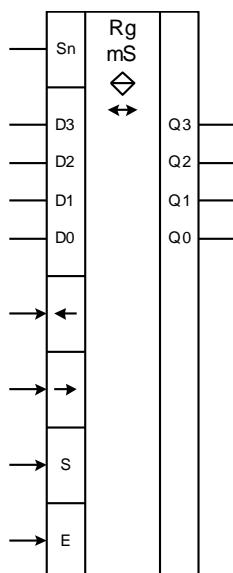


Рисунок 9 – Регистр мантиисы результата S в ОК

Имеет прямые выходы $Q0-Q3$.

Режимы работы:

При поступлении на вход S логической единицы производится запись с входов $D0-D3$ и информация появляется на прямых выходах $Q0-Q3$.

Вход S_n предназначен для подачи вдвигаемой информации при сдвиге содержимого регистра влево (в сторону старших разрядов).

При поступлении на вход \rightarrow логической единицы происходит модифицированный сдвиг информации вправо (в сторону младших разрядов).

При поступлении на вход \leftarrow логической единицы происходит сдвиг информации влево (в сторону старших разрядов), при этом в младший значащий разряд вдвигается знаковый разряд.

Прямые выходы регистра имеют три состояния. При поступлении на вход E логической единицы происходит выдача на прямые выходы регистра информации находящейся в регистре. Если на вход E подан логический ноль – выходы регистра находятся в третьем высокоимпедансном состоянии.

Регистр порядка результата S представленного в коде обработке (в обратном коде) приведён на рис. 10.

$RgpS$ – 4 разрядный универсальный регистр.

$D0-D3$ – прямые входы регистра.

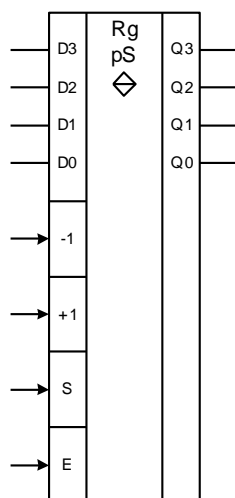


Рисунок 10 – Регистр порядка результата S в ОК

Регистр предназначен для приема, вычитания 1, прибавления 1 и хранения порядка результата S в обратном коде.

Имеет прямые выходы $Q0-Q3$.

Режимы работы:

При поступлении на вход S единицы производится запись с входов $D0-D3$ и информация появляется на прямых выходах $Q0-Q3$.

При поступлении на вход $+1$ единицы происходит арифметическое прибавление единицы к содержащемуся в регистре числу.

При поступлении на вход -1 единицы происходит арифметическое вычитание единицы из содержащегося в регистре числа.

Прямые выходы регистра имеют три состояния. При поступлении на вход E логической единицы происходит выдача на прямые выходы регистра информации находящейся в регистре. Если на вход E подан логический ноль – выходы регистра находятся в третьем высокоимпедансном состоянии.

Сумматоры $SM1$ и $SM2$ входящие в преобразователи кодов мантисс исходных операндов A и B приведены на рис. 11.

$SM1, SM2$ - 4 разрядные сумматор.

$A0-A3$ входы числа A .

$B0-B3$ входы числа B .

C – вход входного переноса.

$S0-S3$ выходы суммы чисел A и B .

P – перенос в следующий разряд.

Сумматоры предназначены для перевода мантисс A и B в обратный код.

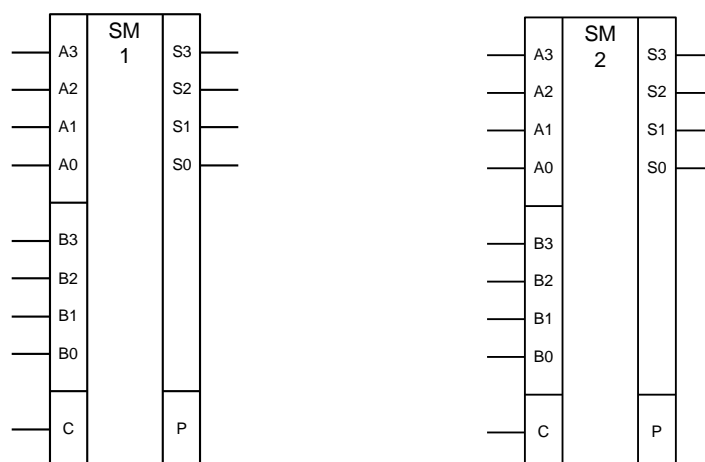


Рисунок 11 – Сумматоры преобразователей кодов мантисс исходных операндов

Режим работы:

На входы $A0-A3$ подается число, которое нужно перевести в обратный код. На входы $B0-B3$ подается знак этого числа. На вход C подается 0. Таким образом на этих сумматорах производится вычитание единицы из мантисс исходных операндов подающихся на входы спецпроцессора в дополнительном коде. На выходах $S0-S3$ получается число в обратном коде.

Выход P не используется.

Сумматоры $SM3$ и $SM4$ предназначенные для суммирования мантисс и вычитания порядков соответственно, операндов A и B приведены на рис. 12.

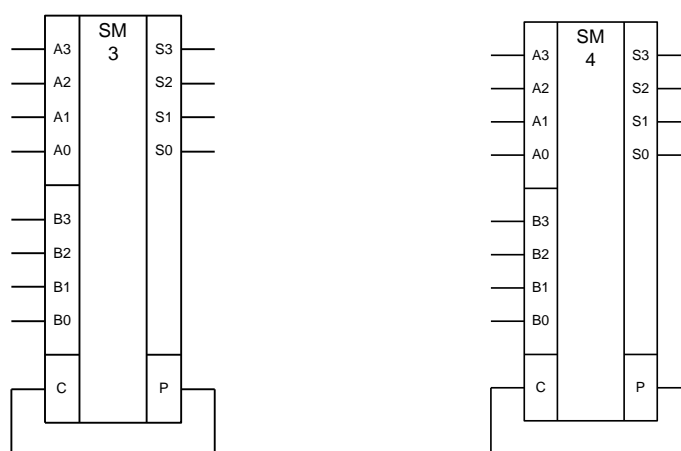


Рисунок 12 – Сумматоры обработки мантисс и порядков операндов

SM3, SM4 - 4 разрядные сумматоры.

A0-A3 и *B0-B3* входы слагаемых.

C – вход входного переноса.

S0-S3 выходы суммы чисел.

P – выходной перенос в следующий разряд.

Выход *P* и вход *C* соединены, чтобы обеспечить циклический перенос при сложении.

Сумматор *SM3* предназначен для сложения мантисс операндов *A* и *B* переведённых в код обработки, в обратный код. На входы *A0-A3* подается первое слагаемое. На входы *B0-B3* подается второе слагаемое. На выходах *S0-S3* сумматора *SM3* получается сумма мантисс чисел обратном коде.

Сумматор *SM4* предназначен для вычитания порядков чисел *A* и *B* представленных в обратных кодах. На входы *A0-A3* подается первое слагаемое. На входы *B0-B3* подается второе слагаемое (инверсия всех разрядов порядка числа *B*). На выходах *S0-S3* сумматора *SM3* получается сумма мантисс чисел в обратном коде.

Сумматоры *SM1* и *SM2* входящие в преобразователи кодов мантиссы и порядка результата приведены на рис. 13.

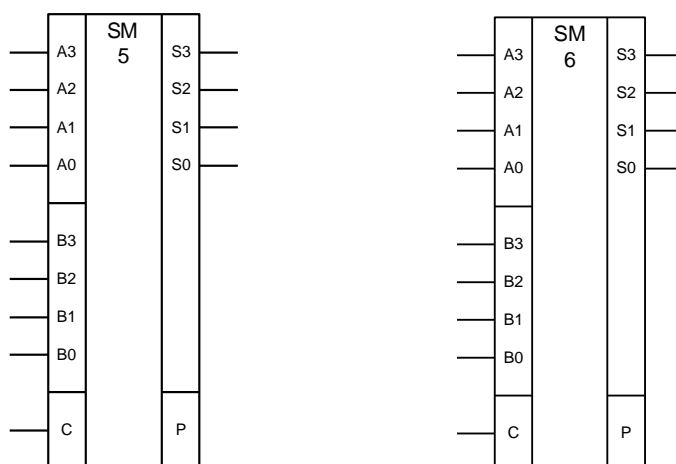


Рисунок 13 – Сумматоры преобразователей кодов мантиссы и порядка результата

SM5, SM6 - 4 разрядные сумматоры.

A0-A3 и *B0-B3* входы слагаемых.

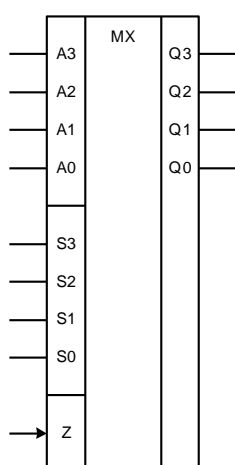
C – вход входного переноса.

S0-S3 выходы суммы чисел.

Сумматоры предназначены для перевода мантиссы и порядка S из обрат-
в дополнительный код.

На входы $A0-A3$ подается число, которое нужно перевести в дополнительный код. На входы $B0-B3$ подается 0. На вход C подается 1. На выходах $S0-S3$ получается число в дополнительном коде.

На рис. 14 представлен мультиплексор, предназначенный для коммутации на выход информации из регистра *RgpA* (входы *A0-A3*) или с сумматора *SM4* (входы *S0-S3*).



MX – мультиплексор $2*4 \rightarrow 4$ разрядов.

Если $Z=0$ на выход мультимплексора $Q0-Q3$ поступают сигналы с $SM4$.

Если $Z=1$ на выход мультиплексора $Q0-Q3$ поступают сигналы с $RgpA$.

Таким образом на выходах мультиплексора будем иметь либо содержимое регистра $RgpA$, в котором после выравнивания порядков содержится порядок результата, либо разность порядков (если на выход мультиплексора коммутируются разряды с выходов сумматора $SM4$).

На рис. 15 приведён мультиплексор, предназначенный для коммутации на Выход ноля.

$MXpS - 2*4 \rightarrow 4$ мультиплексор разрядов.

Предназначен для препятствия записи в регистр rS числа -0 . Так как в обратном коде имеется двузначность нуля ($+0$ и -0), то при переводе порядка

результата из обратного кода в дополнительный необходимо преобразовать -0 в +0.

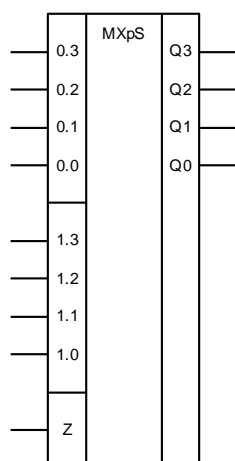


Рисунок 15 – Мультиплексор исключения двузначности нуля

На входы 0.0-0.3 подается значение pS на входы 1.0-1.3 подаются '0'.

Управление происходит сигналом Z .

$Z=0$ на выход мультиплексора $Q0-Q3$ поступают сигналы с 0.0-0.3.

$Z=1$ на выход мультиплексора поступают сигналы с 1.0-1.3 (выбирается +0).

Преобразователь порядка числа A из прямого кода в обратный $D1$ (рис. 16) представляет собой двухразрядный управляемый инвертор позволяющий инвертировать значащие разряды порядка если знак порядка отрицательный и передающий их "как есть" если знак положительный.

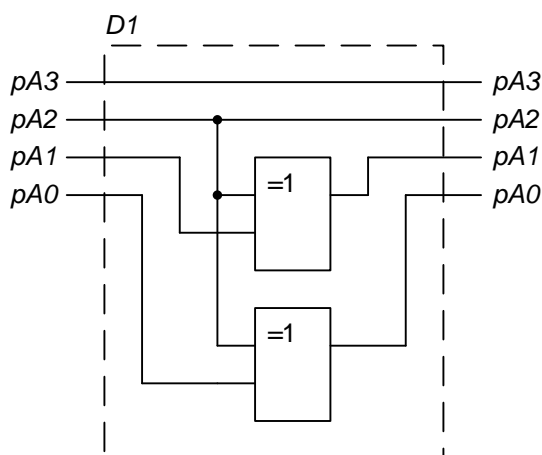


Рисунок 16 – Преобразователь порядка числа A из прямого кода в обратный

Поскольку в обратном коде имеется двузначность нуля, то для обнаружения $+0$ и -0 используется схема, приведённая на рис. 17.

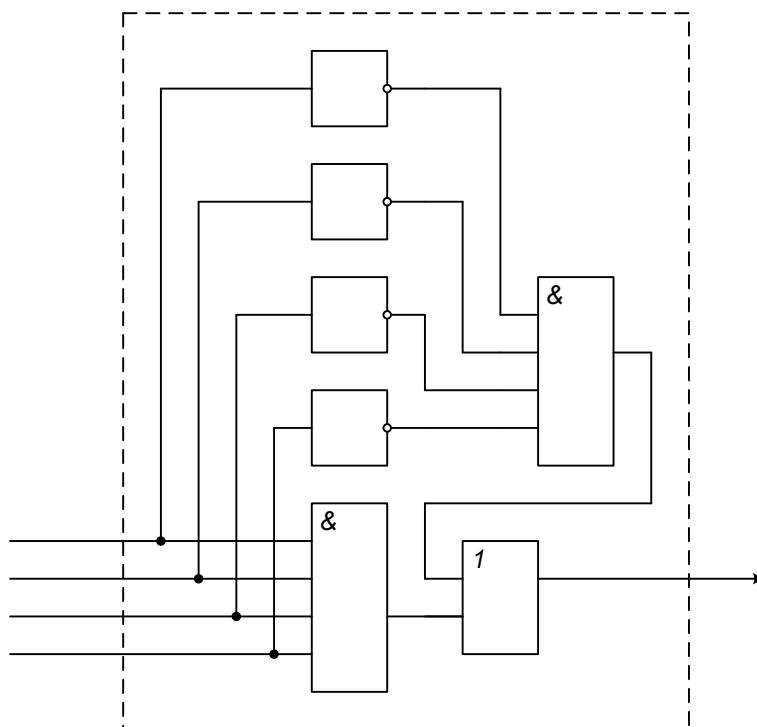


Рисунок 17 – Определитель нуля в обратном коде

По такой схеме построены определители нуля $D2$, $D3$, $D6$, $D9$, используемые в функциональной схеме операционного автомата спецпроцессора.

Кроме этого в схеме используются три двухвходовых сумматора по модулю 2. Это $D4$, $D5$ и $D8$, предназначенные соответственно для обнаружения нарушения нормализации мантиссы результата справа, обнаружения нарушения нормализации мантиссы результата слева и обнаружения переполнения разрядной сетки порядка результата.

Схема $D7$ представляет собой логический элемент 4И. Предназначен для управления работой мультиплексора выбора порядка $MXpS$. Если порядок результата равен -0 , то данная схема обнаруживает данный факт, формируя логическую единицу на выходе. Эта единица поступает на адресный вход Z мультиплексора $MXpS$ и на выход мультиплексора передаётся $+0$.

Схема $D10$ представляет собой логический элемент 4И. Предназначен для обнаружения кода "11.11" (-0 в обратном коде), который может возникнуть при нормализации результата.

2. Разработка граф-схемы микропрограммы.

2.1 Представление алгоритма в виде граф-схемы

Для представления алгоритмов используется специальный граф-схемный язык, основными элементами которого являются функциональные и вспомогательные блоки, описывающие операции алгоритма [3, 4, 8].

Каждому действию, осуществляемому алгоритмом, придаётся значение микрооперации. Любая микрооперация изменяет состояние функциональных узлов разрабатываемого устройства.

Граф-схема микропрограммы (ГСМ) строится с использованием блоков четырёх типов (рис. 1) и линий связи, связывающих блоки.

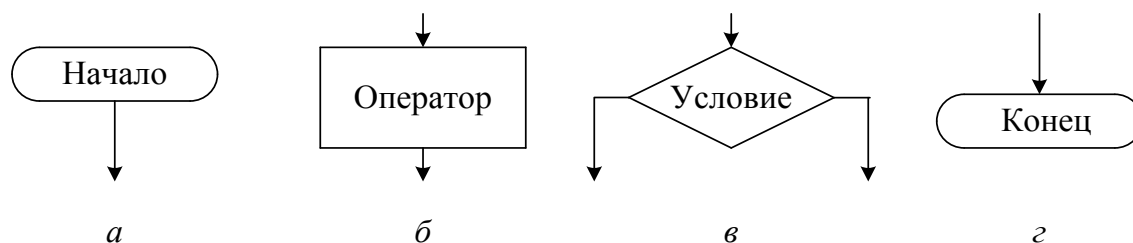


Рисунок 18 – Вершины ГСМ: *а* - начальная; *б* - операторная; *в* - условная; *г* – конечная

Начальный блок отмечает начало алгоритма и имеет единственный выход, из которого исходит линия связи к первому выполняемому блоку графа.

Операторный блок определяет действие – микрооперацию, совокупность совместимых микроопераций.

Условный блок используется для разветвления вычислительного процесса в одном из двух возможных направлений в зависимости от значения проверяемого логического условия.

Конечный блок отмечает конец микропрограммы.

ГСМ считается корректной, если выполняются следующие условия:

- 1) ГСМ содержит конечное число блоков, каждый из которых принадлежит к перечисленным типам;
- 2) имеет один начальный и один конечный блок;
- 3) выходы и входы блоков соединяются с помощью линий связи, направленных от выхода предыдущего блока ко входу последующего;

- 4) каждый выход блока соединён с одним входом;
- 5) из любого блока существует хотя бы один путь к конечному;
- 6) один из выходов условного блока может соединяться с её входом, что недопустимо для операторного блока;
- 7) в каждом условном блоке записывается один из входных сигналов управляющего автомата;
- 8) в каждом операторном блоке записывается оператор (микрокоманда из множества микрокоманд).

ГСМ делятся на содержательные и закодированные [5, 6]. В содержательной ГСМ в условных и операторных вершинах записываются условия и операторы. Пример содержательной ГСМ представлен на рис. 19.

Закодированная ГСМ отличается от содержательной тем, что в условных и операторных вершинах записываются символы x_1, \dots, x_n и y_1, \dots, y_m , отождествляемые с осведомительными и управляющими сигналами УА.

На рис. 20 представлена закодированная ГСМ, полученная на основе содержательной ГСМ рис. 19.

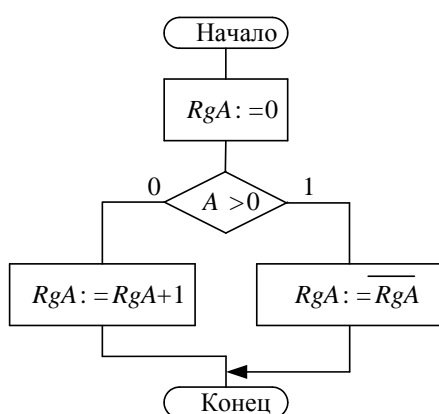


Рисунок 19 – Содержательная ГСМ

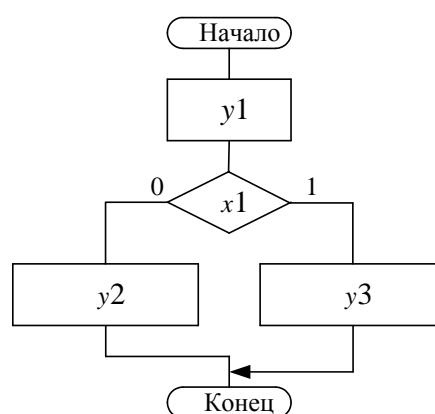


Рисунок 20 – Закодированная ГСМ

Блоки в алгоритме соединяются линиями связи, обозначающими направление потока. Направление потока сверху вниз и справа налево считается стандартным.

В случаях, когда необходимо внести большую ясность в граф схему (например, при соединениях), на линиях используются стрелки. Если поток имеет направление отличное от стандартного, стрелки должны указывать это направление.

Линии в схемах должны подходить к вершине (блоку) либо сверху, либо слева, а исходить либо снизу, либо справа, причём вариант, когда линия входит в вершину сверху, а выходит снизу является предпочтительным. Линии должны быть направлены к центру символа.

Линии связи проводятся параллельно внешним краям листа. Допускается пересечение их или изгиб под углом 90° . Рекомендуемое расстояние между параллельными линиями при изображении ГСМ данного курсового проекта – не менее 5 мм.

В каждый операторный блок (вершину) может войти и выйти только по одной линии связи. Из условного блока выходят две линии связи. Линии могут выходить из правого и левого углов, либо снизу и из одного из углов. Над связями, выходящими из условного блока, пишут слова "Да", "Нет", или ставят цифры "0", "1", причём "0" соответствует "Нет", а "1" – "Да".

2.2 Осведомительные сигналы X используемые в спецпроцессоре

- $X1$ Сигнал о наличии на входах чисел A и B ;
- $X2$ Выходной сигнал схемы обнаружения нуля; $X2=0$ если pA не равна 0 и $X2=1$, если $pA=0$;
- $X3$ Выходной сигнал схемы обнаружения нуля; $X3=0$ если pB не равна 0 и $X3=1$, если $pB=0$;
- $X4$ Признак нарушения нормализации справа;
- $X5$ Признак нарушения нормализации слева;
- $X6$ Выходной сигнал схемы обнаружения нуля; $X6=0$ если mS не равна 0 и $X6=1$, если $mS=0$;
- $X7$ Знак содержимого регистра порядка $RgpS$;
- $X8$ Признак переполнения разрядной сетки; если $X8=1$, то произошло переполнение порядка;
- $X9$ Выходной сигнал схемы обнаружения нуля; $X9=0$ если pS не равен 0 и $X9=1$, если $pS=0$;
- $X10$ Выходной сигнал схемы обнаружения -0; $X10=0$ если pS не равен -0 и $X10=1$, если $pS=-0$;

2.3 Управляющие сигналы Y используемые в спецпроцессоре

Y1, Y2, Y3	Запись информации с входов данных спецпроцессора в соответствующий регистр;
Y4, Y6, Y12, Y13	Модифицированный сдвиг информации, содержащейся в соответствующем регистре.
Y5, Y7, Y9, Y11, Y14, Y19	Запись параллельного кода в соответствующий регистр;
Y8, Y10, Y18	Увеличение кода находящегося в соответствующем регистре на единицу;
Y15, Y20	Выдача результатов из регистров <i>RgmS</i> и <i>RgpS</i> выход спецпроцессора;
Y16	Коммутация мультиплексора; если $Y13=0$ то на выход мультиплексора передается информация с нулевого входа, а если $Y13=1$, то с первого;
Y17	Арифметическое уменьшение на единицу кода, содержащегося в <i>RgpS</i> ;
Y21	Переполнение разрядной сетки;
Y22	Сигнал о наличии на выходах результата;

2.4 Граф-схема микропрограммы

На рис. 21 представлена граф-схема микропрограммы составленной с учётом технического задания и разработанной функциональной схемы специализированного процессора.

2.5 Словесное описание граф-схемы микропрограммы

Блок 1. Проверяем осведомительный сигнал $X1$, наличие входных чисел на шине данных. Если $X1=1$, то переходим на блок 2. Если $X1=0$ опять на блок 1.

Блок 2. Производим запись операндов с шины данных по сигналам $Y1$, $Y2$, $Y3$ в соответствующие регистры. Переходим на блок 3.

Блок 3. Производим запись переведенных в обратный код операндов по сигналам $Y5$, $Y7$, $Y9$ в соответствующие регистры. Производим запись порядка B с шины данных в регистр порядка B по сигналу $Y11$. Переходим на блок 4.

Блок 4. Производим запись данных в регистр порядка суммы по сигналу Y19. Переходим на блок 5.

Блок 5. Проверяем осведомительный сигнал X9. Равенство 0 порядка суммы. Если $X9=0$, то порядок суммы не равен 0 и переходим на блок 6. Если $X9=1$ то порядок суммы равен 0 и переходим на блок 13.

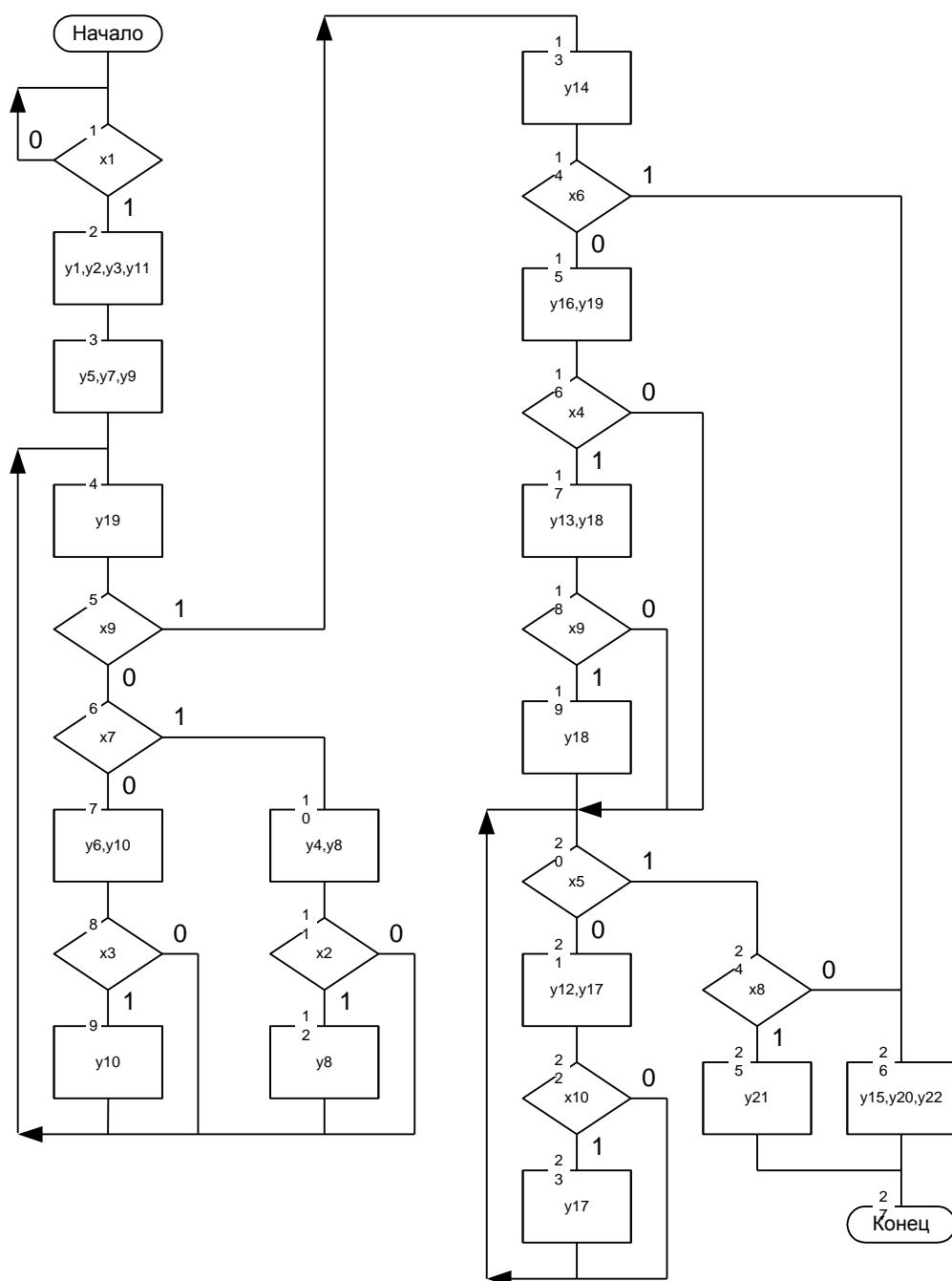


Рисунок 21 – Граф схема микропрограммы арифметической операции сложения с плавающей точкой специализированного процессора

Блок 6. Проверяем осведомительный сигнал $X7$. Знак содержимого регистра порядка суммы. Если $X7=0$, то порядок суммы положителен, переходим на блок 10. Если $X7=1$, то порядок суммы отрицателен, переходим на блок 7.

Блок 7. По сигналу $Y6$ осуществляем сдвиг мантиссы B вправо на один разряд. По сигналу $Y10$ осуществляем увеличение порядка B на единицу. Переходим на блок 8.

Блок 8. Проверяем осведомительный сигнал $X3$. Равенство 0 порядка B . Если $X3=0$, то порядок B не равен 0 и переходим на блок 4. Если $X3=1$ то порядок B равен 0 и переходим на блок 9.

Блок 9. По сигналу $Y10$ осуществляем увеличение порядка B на единицу. Переходим на блок 4.

Блок 10. По сигналу $Y4$ осуществляем сдвиг мантиссы A вправо на один разряд. По сигналу $Y8$ осуществляем увеличение порядка A на единицу. Переходим на блок 11.

Блок 11. Проверяем осведомительный сигнал $X2$. Равенство 0 порядка A . Если $X2=0$, то порядок A не равен 0 и переходим на блок 4. Если $X2=1$ то порядок A равен 0 и переходим на блок 12.

Блок 12. По сигналу $Y8$ осуществляем увеличение порядка A на единицу. Переходим на блок 4.

Блок 13. По сигналу $Y14$ осуществляем запись данных в регистр мантиссы суммы. Переходим на блок 14.

Блок 14. Проверяем осведомительный сигнал $X6$. Равенство 0 мантиссы суммы. Если $X6=0$, то мантисса суммы не равна 0 и переходим на блок 15. Если $X6=1$ мантисса суммы равна 0 и переходим на блок 26.

Блок 15. По сигналу $Y16$ осуществляем коммутацию мультиплексора, чтобы на выход подавалась информация с 1 входа. Производим запись данных в регистр порядка суммы по сигналу $Y19$. Переходим на блок 16.

Блок 16. Проверяем осведомительный сигнал $X4$. Признак нарушения нормализации справа. Если $X4=0$, то нарушения нормализации нет и переходим на блок 20. Если $X4=1$ то нормализация нарушена и переходим на блок 17.

Блок 17. По сигналу $Y13$ осуществляем сдвиг мантиссы суммы вправо на один разряд. По сигналу $Y18$ осуществляем увеличение порядка суммы на единицу. Переходим на блок 18.

Блок 18. Проверяем осведомительный сигнал $X9$. Равенство 0 порядка суммы. Если $X9=0$, то порядок суммы не равен 0 и переходим на блок 20. Если $X9=1$ то порядок суммы равен 0 и переходим на блок 19.

Блок 19. По сигналу $Y18$ осуществляем увеличение порядка суммы на единицу. Переходим на блок 20.

Блок 20. Проверяем осведомительный сигнал $X5$. Признак нарушения нормализации слева. Если $X4=1$, то нарушения нормализации нет и переходим на блок 24. Если $X4=0$ то нормализация нарушена и переходим на блок 21.

Блок 21. По сигналу $Y12$ осуществляем сдвиг мантиссы суммы влево на один разряд. По сигналу $Y17$ осуществляем уменьшение порядка суммы на единицу. Переходим на блок 22.

Блок 22. Проверяем осведомительный сигнал $X10$. Равенство -0 порядка суммы. Если $X10=0$, то порядок суммы не равен -0 и переходим на блок 20. Если $X10=1$ то порядок суммы равен -0 и переходим на блок 23.

Блок 23 По сигналу $Y17$ осуществляем уменьшение порядка суммы на единицу. Переходим на блок 20.

Блок 24. Проверяем осведомительный сигнал $X8$. Признак переполнения разрядной сетки. Если $X8=1$, то произошло переполнение порядка, переходим на блок 25. Если $X8=0$, то переполнения нет, и переходим на блок 26.

Блок 25. Осуществляем выдачу сигнала $Y21$. Сигнал о переполнении разрядной сетки. Переходим на блок 27.

Блок 26. По сигналу $Y15$ осуществляем выдачу информации с регистра мантиссы суммы на шину данных. По сигналу $Y20$ осуществляем выдачу информации с регистра порядка суммы на шину данных. Осуществляем выдачу сигнала $Y22$. Сигнал о наличии на выходах результата. Переходим на блок 27.

Блок 27 Конец программы

2.6 Таблица работы операционного автомата

Проверочные числа: $A=-1$; $B=-3$. Представим их в двоичной системе счисления в модифицированном коде с плавающей точкой в кодах соответствующих техническому заданию:

$$[mA]_{\text{дк}} = 11.10 \quad [pA]_{\text{пк}} = 00.01;$$

$$[mB]_{\text{дк}} = 11.01 \quad [pB]_{\text{ок}} = 00.10;$$

В таблице работы операционного автомата отображено содержимое всех регистров операционного автомата в каждом такте его работы при выполнении операции сложения проверочных чисел.

Время затрачиваемое на выполнение сложения заданных проверочных чисел составляет 19 тактов.

Об окончании работы спецпроцессора по сложению двух чисел с плавающей точкой сигнализирует наличие логической единицы на выходе Y_{22} – сигнал готовности результата.

Таблица 1 – потактовая работа операционного автомата

Такт	$Rgm1$	$Rgm2$	$Rgp1$	$RgmA$	$RgmB$	$RgpA$	$RgpB$	$RgmS$	$RgpS$
1	*	*	*	*	*	*	*	*	*
2	11.10	11.01	00.01	*	*	*	00.10	*	*
3	11.10	11.01	00.01	11.01	11.00	00.01	00.10	*	*
4	11.10	11.01	00.01	11.01	11.00	00.01	00.10	*	11.10
5	11.10	11.01	00.01	11.01	11.00	00.01	00.10	*	11.10
6	11.10	11.01	00.01	11.01	11.00	00.01	00.10	*	11.10
7	11.10	11.01	00.01	11.10	11.00	00.10	00.10	*	11.10
8	11.10	11.01	00.01	11.10	11.00	00.10	00.10	*	11.10
9	11.10	11.01	00.01	11.10	11.00	00.10	00.10	*	00.00
10	11.10	11.01	00.01	11.10	11.00	00.10	00.10	*	00.00
11	11.10	11.01	00.01	11.10	11.00	00.10	00.10	10.11	00.00
12	11.10	11.01	00.01	11.10	11.00	00.10	00.10	10.11	00.00
13	11.10	11.01	00.01	11.10	11.00	00.10	00.10	10.11	00.10
14	11.10	11.01	00.01	11.10	11.00	00.10	00.10	10.11	00.10
15	11.10	11.01	00.01	11.10	11.00	00.10	00.10	11.01	00.11
16	11.10	11.01	00.01	11.10	11.00	00.10	00.10	11.01	00.11
17	11.10	11.01	00.01	11.10	11.00	00.10	00.10	11.01	00.11
18	11.10	11.01	00.01	11.10	11.00	00.10	00.10	11.01	00.11
19	11.10	11.01	00.01	11.10	11.00	00.10	00.10	11.01	00.11

В регистрах $RgmS$ и $RgpS$ содержится результат в обратном коде. Но перед тем как числа выдаются на выход спецпроцессора, они подаются на преобразователь кодов (сумматоры), которые переводят результат в дополнительный код. На выходы данным спецпроцессора выдаётся результат представленный в дополнительном коде, т.е. $[mS]_{дк} = 11.10$ $[pS]_{дк} = 00.11$;

3 Разработка функциональной схемы управляющего автомата специализированного процессора

3.1 Управляющий автомат с программируемой логикой и принудительной адресацией

Принудительная адресация микрокоманд состоит в том, что в каждой микрокоманде указываются все возможные адреса следующих микрокоманд. Адрес следующей микрокоманды может задаваться безусловно, т. е. независимо от значений осведомительных сигналов, или выбираться по условию, определяемому текущими значениями осведомительных сигналов [3, 4, 8].

В разветвляющейся ГСМ содержатся условные вершины, и в зависимости от значений опрашиваемых в них осведомительных сигналов X адрес следующей микрокоманды выбирается из двух или более возможных адресов.

В связи с тем, что структура УА с микропрограммным управлением стандартна (рис. 22), усилия разработчиков направляются не на получение структурной схемы, а на составление кодированной микропрограммы, которая записывается в ячейки ПЗУ.

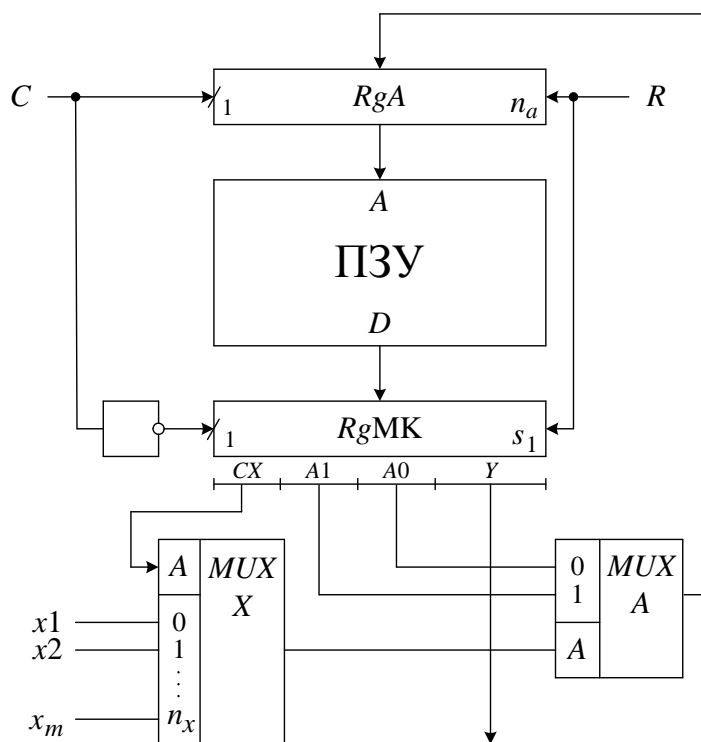


Рисунок 22 – Структура УА с принудительной адресацией

Если принять, что в каждой микрокоманде опрашивается только один осведомительный сигнал x_i , то формат микрокоманды будет выглядеть как показано на рис. 23.

Микрокоманда содержит четыре поля: CX – поле управления выбором опрашиваемого входного осведомительного сигнала x_i ; $A1$ – поле адреса следующей микрокоманды, если опрашиваемый осведомительный сигнал $x_i = 1$; $A0$ – поле адреса следующей микрокоманды, если опрашиваемый осведомительный сигнал $x_i = 0$; Y – поле операционной части микрокоманды (поле выходных управляющих сигналов).

1	CX	n_x	1	$A1$	n_a	1	$A0$	n_a	1	Y	n_y
---	------	-------	---	------	-------	---	------	-------	---	-----	-------

Рисунок 23 – Формат микрокоманды УА с принудительной адресацией

Разрядность этих полей определяется следующим соотношением:

$$n_x = \lceil \log_2 n_{yc} \rceil,$$

где n_{yc} – число входных осведомительных сигналов x ,

$$n_a = \lceil \log_2 (n_{ув} + n_{оп}) \rceil,$$

где $n_{оп}$ – число операторных вершин, а $n_{ув}$ – число условных вершин ГСМ.

Число разрядов поля Y при горизонтальном кодировании операционной части микрокоманды определяется по следующему соотношению.

$$n_y = k,$$

где k – число выходных управляющих сигналов.

3.2 Разработка функциональной схемы управляющего автомата спецпроцессора

Поскольку структура УА с принудительной адресацией стандартна, процесс разработки функциональной схемы заключается в определении разрядностей полей микрокоманды и выборе разрядности регистра адреса, регистра микрокоманд, разрядности мультиплексоров и организации ПЗУ.

Основываясь на разработанной граф-схеме микропрограммы выполнения операции сложения с плавающей точкой определим разрядность отдельных полей микрокоманды и самой микрокоманды в целом.

Разрядность поля управления выбором опрашиваемого входного осведомительного сигнала:

$$n_x = \lceil \log_2 10 \rceil = 4.$$

Разрядность полей адреса следующей микрокоманды:

$$n_a = \lceil \log_2 26 \rceil = 5.$$

Число разрядов поля Y при горизонтальном кодировании операционной части микрокоманды:

$$n_Y = 22.$$

Разрядность микрокоманды в целом составляет:

$$S = n_x + 2 \cdot n_a + n_Y = 36.$$

Таким образом, для реализации управляющего автомата спецпроцессора потребуется ПЗУ организации $2^5 \times 36$, т.е. ПЗУ должно иметь 5 адресных входов, что позволяет обращаться к 32 ячейкам памяти. Разрядность ячеек должна составлять 36 разрядов. Условное графическое обозначение такого ПЗУ представлено на рис. 24.

На вход CS (выборка кристалла) постоянно подан разрешающий потенциал и при поступлении на адресные входы нового адреса происходит выдача информации из ячейки на выход.

В качестве регистра адреса выбран 5-разрядный параллельный регистр с возможностью асинхронного сброса. В каждом такте работы спецпроцессора в нём хранится адрес текущей микрокоманды. При установке в регистре адреса "00000" (при сбросе) происходит адресация к нулевой ячейке ПЗУ, в которой записывается первая микрокоманда микропрограммы.

Условное графическое обозначение регистра адреса приведено на рис. 25.

Режимы работы регистра адреса:

Вход R – обнуление регистра.

Вход S – синхронизация. При поступлении синхроимпульса происходит запись адреса следующей микрокоманды в регистр.

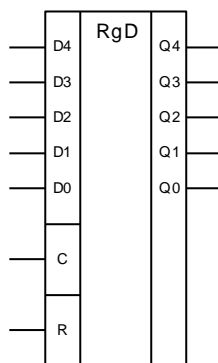


Рисунок 25 – Регистр адреса текущей микрокоманды

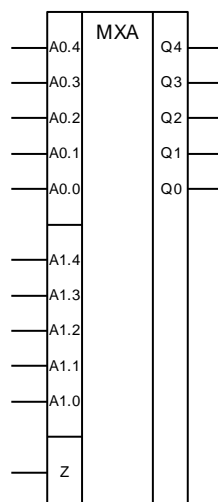


Рисунок 26 – Мультиплексор выбора адреса следующей микрокоманды

Для управления выбора текущего опрашиваемого осведомительного сигнала X выбран мультиплексор MXX , позволяющий коммутировать на выход один из 16-ти информационных одноразрядных сигналов (рис. 27). Для управления коммутацией используются 4 управляющих (адресных) входа $Z3-Z0$.

Поскольку в разрабатываемом спецпроцессоре всего 10 осведомительных сигналов предназначенных для коммутации, то информационные входы с 11 по 15-й не используются.

Регистр текущей микрокоманды (рис. 28) представляет собой параллельный 36-разрядный регистр. Параллельная запись информации (текущей микрокоманды) производится по сигналу C . При подаче логической единицы на вход R – производится обнуление (сброс) данного регистра.

В табл. 2 приведено распределение входов мультиплексора выбора текущего опрашиваемого сигнала X .

Перечень осведомительных сигналов X использующихся в разрабатываемом спецпроцессоре представлен в табл 3.

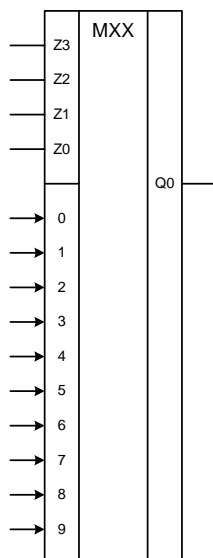


Рисунок 27 – Мультиплексор выбора текущего опрашиваемого осведомительного сигнала

Таблица 2 – назначения адресных входов мультиплексора MXX сигнала X

Z3	Z2	Z1	Z0	Сигнал
0	0	0	0	X1
0	0	0	1	X2
0	0	1	0	X3
0	0	1	1	X4
0	1	0	0	X5
0	1	0	1	X6
0	1	1	0	X7
0	1	1	1	X8
1	0	0	0	X9
1	0	0	1	X10

Таблица 3 – Перечень осведомительных сигналов X

X1	Сигнал о наличии на входах чисел А и В;
X2	Выходной сигнал схемы обнаружения нуля;
X3	Выходной сигнал схемы обнаружения нуля;
X4	Признак нарушения нормализации справа;
X5	Признак нарушения нормализации слева;

X6	Выходной сигнал схемы обнаружения нуля;
X7	Знак содержимого регистра порядка RgpS;
X8	Признак переполнения разрядной сетки;
X9	Выходной сигнал схемы обнаружения нуля;
X10	Выходной сигнал схемы обнаружения -0;

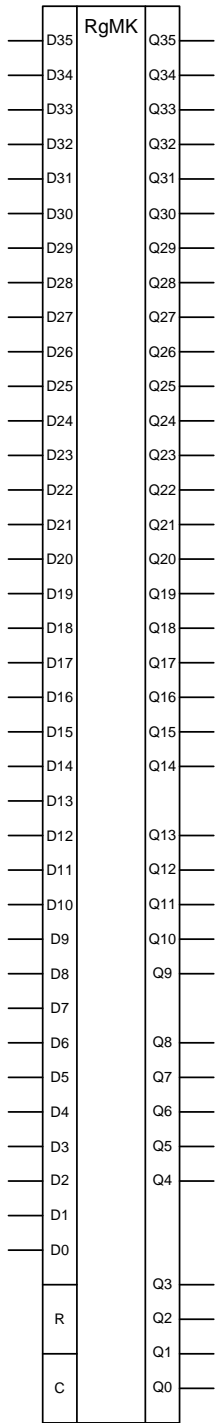


Рисунок 28 – Регистр текущей микрокоманды

3.3 Разработка микропрограммы выполнения заданной операции

Разработанная микропрограмма (табл. 4) заносится в ПЗУ. В каждой ячейке ПЗУ записана отдельная микрокоманда. Выполнение микропрограммы начинается с микрокоманды, размещённой в нулевой ячейке (в ячейке с адресом "00000"). Занесение в регистр адреса "00000" осуществляется подачей сигнала *R* (рис. 22)

Таблица 4 – Микропрограмма

Адрес ПЗУ	CX	Адрес A1	Адрес A0	Y(1-22)
00000	0000	00001	00000	0000000000000000000000
00001	****	00010	00010	1110000000100000000000
00010	****	00011	00011	0000101010000000000000
00011	****	00100	00100	00000000000000000001000
00100	1000	01100	00101	0000000000000000000000
00101	0110	01001	00110	0000000000000000000000
00110	****	00111	00111	0000010001000000000000
00111	0010	01000	00011	0000000000000000000000
01000	****	00011	00011	0000000001000000000000
01001	****	01010	01010	0001000100000000000000
01010	0001	01011	00011	0000000000000000000000
01011	****	00011	00011	0000000100000000000000
01100	****	01101	01101	0000000000000100000000
01101	0101	11001	01110	0000000000000000000000
01110	****	01111	01111	00000000000000001001000
01111	0011	10000	10011	0000000000000000000000
10000	****	10001	10001	0000000000001000010000
10001	1000	10010	10011	0000000000000000000000
10010	****	10011	10011	00000000000000000010000
10011	0100	10111	10100	0000000000000000000000
10100	****	10101	10101	00000000000010000100000
10101	1001	10110	10011	0000000000000000000000
10110	****	10011	10011	00000000000000000010000
10111	0111	11000	11001	0000000000000000000000
11000	****	00000	00000	00000000000000000000010
11001	****	00000	00000	000000000000000010000101

4 Условное графическое обозначение микросхемы спецпроцессора

В соответствии с разработанной функциональной схемой спецпроцессора выбрано следующее условное графическое обозначение микросхемы (рис. 29).

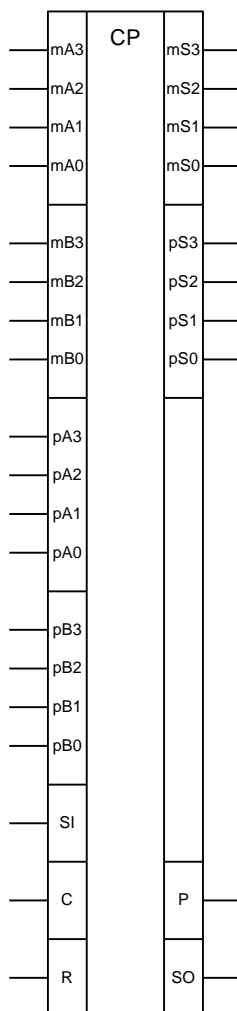


Рисунок 29 – Условное графическое обозначение микросхемы спецпроцессора

Микросхема разработанного специализированного процессора имеет 29 информационных выводов.

Назначение входов спецпроцессора:

- $mA0$ - $mA3$ - мантисса A ($mA3$, $mA2$ знаковые разряды, $mA1$ – старший значащий разряд, $mA0$ – младший значащий разряд);
- $mB0$ - $mB3$ - мантисса B ($mB3$, $mB2$ знаковые разряды, $mB1$ – старший значащий разряд, $mB0$ – младший значащий разряд);

- $pA0$ - $pA3$ - порядок A ($pA3$, $pA2$ знаковые разряды, $pA1$ – старший значащий разряд, $pA0$ – младший значащий разряд);
- $pB0$ - $pB3$ - порядок B ($pB3$, $pB2$ знаковые разряды,);
- SI – сигнал о наличии на входах чисел A и B (готовность исходных операндов);
- C – вход синхронизации;
- R – сброс.

Назначение выходов спецпроцессора:

- $mS0$ - $mS3$ - мантисса суммы ($mS3$, $mS2$ знаковые разряды, $mS1$ – старший значащий разряд, $mS0$ – младший значащий разряд);
- $pS0$ - $pS3$ - порядок суммы ($pS3$, $pS2$ знаковые разряды, $pS1$ – старший значащий разряд, $pS0$ – младший значащий разряд);
- P – сигнал о возникновении переполнения разрядной сетки;
- SO – сигнал о наличии на выходах результата.

Заключение

Результатом выполнения данного курсового проекта является разработанная функциональная схема специализированного процессора состоящего из двух автоматов – операционного и управляющего и выполняющего заданную арифметическую операцию (операцию арифметического сложения в обратном коде двоичных чисел представленных в модифицированном коде с плавающей запятой).

Поставленная задача решена полностью. Специализированный процессор способен обрабатывать любые два 8-разрядных числа (4 разряда мантиссы и 4 разряда порядка числа).

В качестве варианта реализации операционного автомата спецпроцессора выбран вариант с двумя сумматорами обратного кода, один из которых предназначен для обработки мантисс исходных операндов, а другой – для обработки порядков. Несмотря на то, что принятый вариант реализации с двумя сумматорами по сравнению с вариантом, когда и мантисса и порядок обрабатываются на одном сумматоре, имеет несколько большие аппаратурные затраты, время выполнения операции суммирования меньше за счёт того, что отсутствует мультиплексирование информации подаваемой на входы сумматора.

Оформление работы выполнено с учётом нормативных документов [2, 9].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Глушков, В.М. Синтез цифровых автоматов. – М.: Физматгиз, 1962. – 476 с.
2. ГОСТ 19.701-90 (ИСО 5807-85) "Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения"
3. Постников, А. И. Теория автоматов и машинная арифметика [Текст] : учеб.пособие для студентов вузов / А. И. Постников, Е. А. Вейсов ; Краснояр. гос. техн. ун-т. - Красноярск : ИПЦ КГТУ, 2006. - 376 с.
4. Постников, А. И. Основы теории цифровых автоматов : учеб. пособие / А. И. Постников. – Красноярск : КГТУ, 2000. – 296 с.
5. Карпов, Ю. Г. Теория автоматов : учебник / Ю. Г. Карпов. – СПб. : Питер, 2002. – 224 с.
6. Горбатов, В. А. Теория автоматов : учебник для студентов вузов / В. А. Горбатов, А. В. Горбатов, М. В. Горбатова. - М. : АСТ : Астрель, 2008. - 559 с.
8. Лазарев, В. Г. Синтез управляющих автоматов / В. Г. Лазарев, Е. И. Пийль. - 3-е изд., перераб. и доп. - М. : Энергоатомиздат, 1989. - 327 с.
9. СТО 4.2-07-2010 "Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности"