5.2. Предварительная разработка подпрограммы умножения

Анализ формул (4) и (5) показывает, что в алгоритме вычисления функции $y(x)$ необходимо выполнять умножение отрицательных чисел. В подпрограмме реализуем ускоренный алгоритм умножения чисел, заданных в дополнительном коде [12].

Алгоритм умножения состоит в следующем. Проверяется содержимое i -го и $(i-1)$-го разрядов множителя Мт (счет ведется справа налево). На первом такте содержимое $(i-1)$-го разряда полагается равным нулю. Если в анализируемых разрядах Мт содержится комбинация 10, то из старших разрядов частичной суммы вычитается множимое Мн, если комбинация 01 – прибавляется множимое. Для комбинаций 00 и 11 никаких операций сложения или вычитания не производится. Умножение на i -й разряд завершается арифметическим сдвигом вправо (знаковый разряд распространяется) на один разряд частичной суммы и множителя. Примеры выполнения операции умножения приведены на рис. 6.

Сдвиг влево на два разряда частичной суммы после завершения цикла умножения (определяется по нулевому содержимому счетчика тактов) необходим для правильной фиксации целой и дробной частей произведения.

В примерах приведено получение всех разрядов результата. В программе младшие разряды произведения будем терять, так как округление производить нецелесообразно, что следует из разд. 3.

Распределение регистров общего назначения БОД показано в табл. 2. Спадающий при сдвиге вправо разряд множителя PQ0 будем хранить в триггере T1 регистра состояния. Триггер T2 РС используется для фиксации признака нулевого результата блока обработки данных.

Исходная блок-схема подпрограммы умножения приведена на рис. 7, преобразованная – на рис. 8. Адреса микрокоманд обозначены через i , $i+1$, $i+2$ и т.д. Предварительная микропрограмма выполнения алгоритма умножения представлена в табл. 5.

Для обращения к подпрограмме MULT из основной программы может быть использована команда CJS (условный переход к подпрограмме), а для возврата из подпрограммы в основную программу – команда CRTN (условный возврат из подпрограммы). Эти команды могут быть реализованы с помощью БИС К1804ВУ4 [9, 10]. Если сигнал на входе $\overline{CCE}$ этой микросхемы имеет высокий уровень, то указанные команды будут выполнены безусловно.

$M_н = A = 1,375_{10}$

0	0 1,0 1 1		$M_н = A$
0	0 1,1 1 0	0	$M_т = B$
+	0 0 0 0 0	0 0 0 0 0	$\Sigma 0$
	1 1 0 1 0 1		$A_{дк}$
	1 1 0 1 0 1	0 0 0 0 0	$\Sigma 1$
	1 1 1 0 1 0	1 0 0 0 0	$R1(\Sigma 1)$
	1 1 1 1 0 1	0 1 0 0 0	$\Sigma 2 = R2(\Sigma 1)$
	1 1 1 1 1 0	1 0 1 0 0	$\Sigma 3 = R3(\Sigma 1)$
+	1 1 1 1 1 1	0 1 0 1 0	$\Sigma 4 = R4(\Sigma 4)$
	0 0 1 0 1 1		A
	0 0 1 0 1 0	0 1 0 1 0	$\Sigma 5$
	0 0 0 1 0 1	0 0 1 0 1	$R1(\Sigma 5)$
0	1 0,1 0 0	1 0 1 0 0	$П = L2(\Sigma 5)$

$M_т = B = \pm 1,875_{10}$

0	0 1,0 1 1		$M_н = A$
1	1 0,0 0 1	0	$M_т = B_{дк}$
+	0 0 0 0 0	0 0 0 0 0	$\Sigma 0$
	1 1 0 1 0 1		$A_{дк}$
	1 1 0 1 0 1	0 0 0 0 0	$\Sigma 1$
+	1 1 1 0 1 0	1 0 0 0 0	$R1(\Sigma 1)$
	0 0 1 0 1 1		A
	0 0 1 0 1 1	1 0 0 0 0	$\Sigma 2$
	0 0 0 1 0 1	1 1 0 0 0	$R1(\Sigma 2)$
	0 0 0 0 0 1	0 1 1 0 0	$\Sigma 3 = R2(\Sigma 2)$
+	0 0 0 0 0 0	1 0 1 1 0	$\Sigma 4 = R3(\Sigma 1)$
	1 1 0 1 0 1		$A_{дк}$
	1 1 0 1 0 1	1 0 1 1 0	$\Sigma 5$
	1 1 1 0 1 0	1 1 0 1 1	$R1(\Sigma 5)$
1	0 1,0 1 1	0 1 1 0 0	$П = L2(\Sigma 5)$

$П = A \times B = \pm 2,578_{10}$

Рис. 6. Примеры выполнения операции умножения .

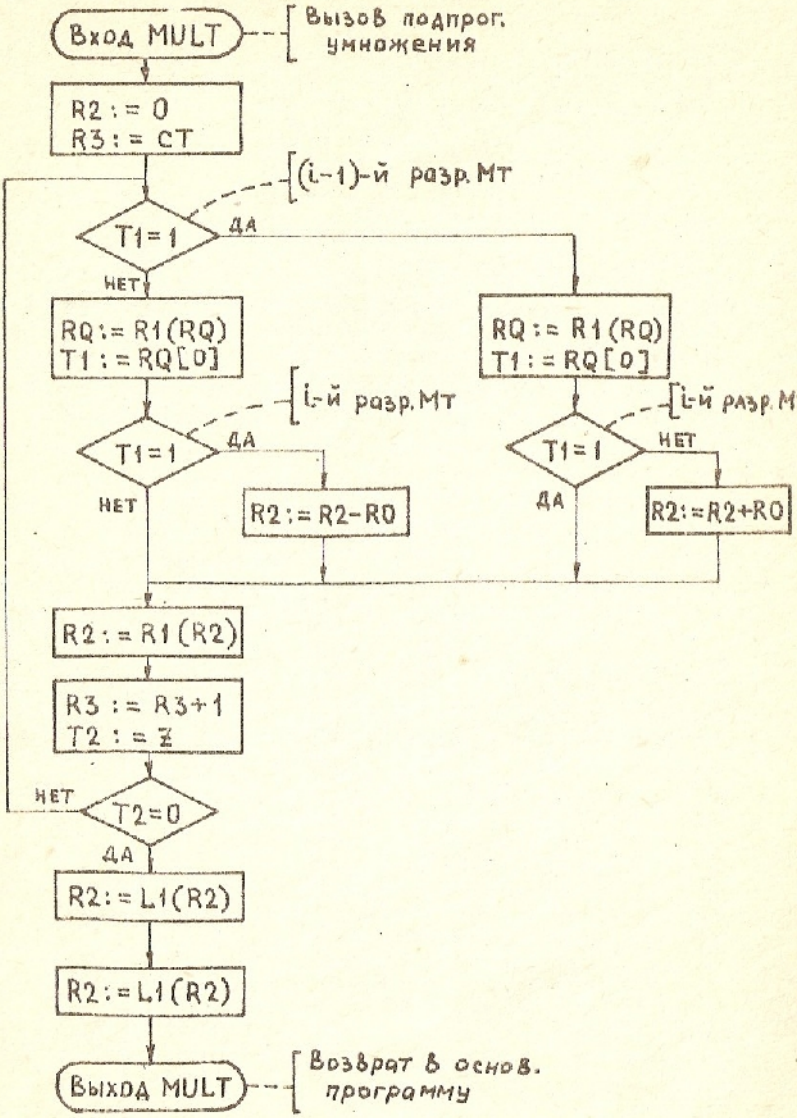

Рис. 7. Исходная блок-схема подпрограммы умножения

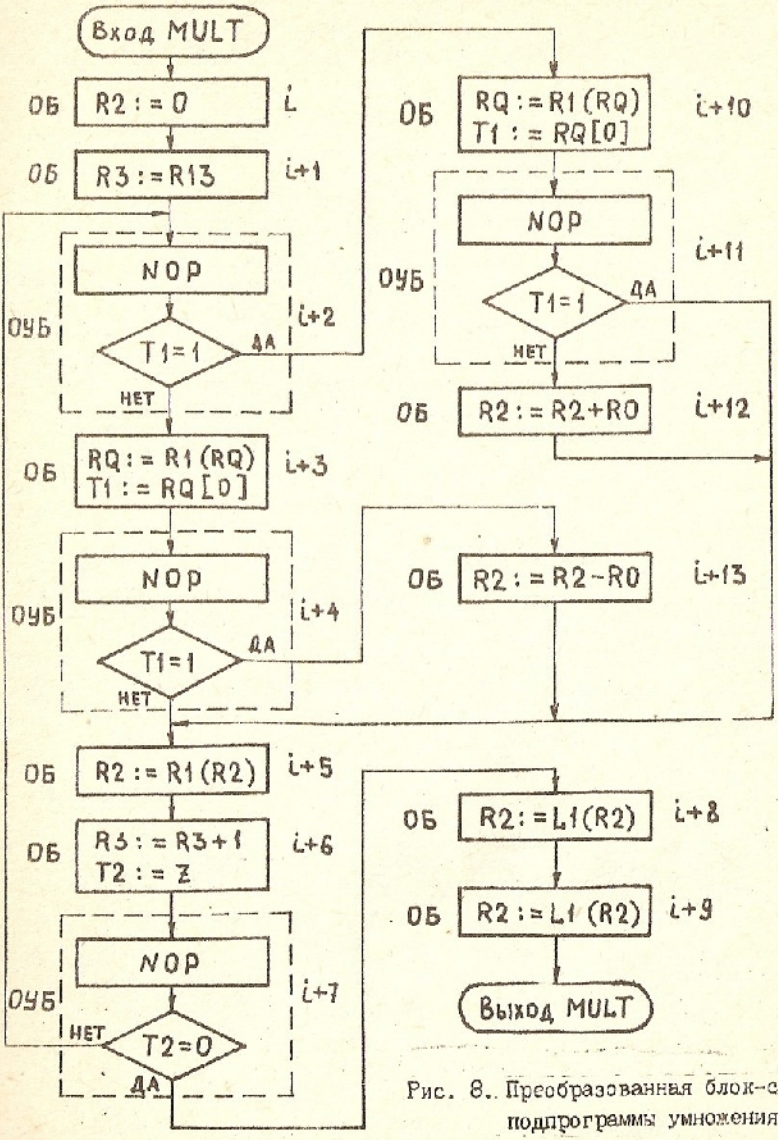

Рис. 8. Преобразованная блок-схема подпрограммы умножения

Таблица 5

Предварительная микропрограмма подпрограммы умножения

Адрес	Микрокоманда	
	Операционная часть	Адресная часть
i	R2:=0	Инкремент
i+1	R3:=R13	Инкремент
i+2	NOP	Если T1=1 i+3, иначе i+10
i+3	RQ:=R1(RQ), T1:=RQ[0]	Инкремент
i+4	NOP	Если T1=1 i+13, иначе i+5
i+5	R2:=R1(R2)	Инкремент
i+6	R3:=R3+1, T2:=Z	Инкремент
i+7	NOP	Если T2=0 i+8, иначе i+2
i+8	R2:=L1(R2)	Инкремент
i+9	R2:=L1(R2)	Возврат в основ. прогр.
i+10	RQ:=R1(RQ), T1:=RQ[0]	Инкремент
i+11	NOP	Если T1=1 i+5, иначе i+12
i+12	R2:=R2+R0	Безусл. переход на i+5
i+13	R2:=R2-R0	Безусл. переход на i+5

5.3. Предварительная разработка программы вычисления заданной функции

Исходная блок-схема алгоритма расчета функции $y = f(x)$ приведена на рис. 9. При ее составлении предполагалось, что переменная x заносится из внешней среды в регистр R1. Результат вычисленной функции будем накапливать в регистре R5. Константы a_{10}, a_9, a_8, ..., a_0 по мере необходимости извлекаются из ПЗУ констант по адресу, указанному в счетчике адреса СА. Операция умножения выполняется с помощью подпрограммы MULT.

В разд. I показано, что для вычисления искомой функции необходимо учесть одиннадцать членов ряда (см. рис. 2). Для организации циклической части программы необходим счетчик циклов. Его роль возложена на регистр R6. Перед вычислением функции в этот регистр из ПЗУК засылается константа $n = 10$.

Преобразованная блок-схема алгоритма расчета функции приведена на рис. 10, а предварительная микропрограмма – в табл. 6.

5.4. Разработка предварительной программы функционирования спецвычислителя

Упрощенный алгоритм функционирования спецвычислителя приведен на рис. II.

Вход 1 соответствует началу работы системы при поступлении сигнала "Исходное состояние". В этом случае спецвычислитель выдает во внешнюю среду сигнал "Готовность" и переходит в состояние ожидания сигналов "Данные" или "Контроль".

Вход 2 соответствует входу в программу по сигналу "Контроль". В режиме контроля R9 = I $x = x_{эт} = x_м = 1{,}5708$.

Вход 3 соответствует входу в программу по сигналу "Данные". В этом режиме работы спецвычислителя R9 = 0, а переменная x вводится в регистр R1 из внешней среды.

В случае возникновения ошибки в расчетах при контроле или выходе входной переменной x за пределы допустимого диапазона в 33-м разряде слова микрокоманды формируется сигнал "Ошибка" [1].

При нормальном окончании работы единица устанавливается в 34-м разряде слова микрокоманды и во внешнюю среду выдается сигнал "Результат". Его запоминание, хранение и анализ происходят за пределами проектируемого устройства.

Заканчивается работа спецвычислителя переходом по нулевому адресу, что соответствует исходному состоянию системы. Сигнал "Готов-

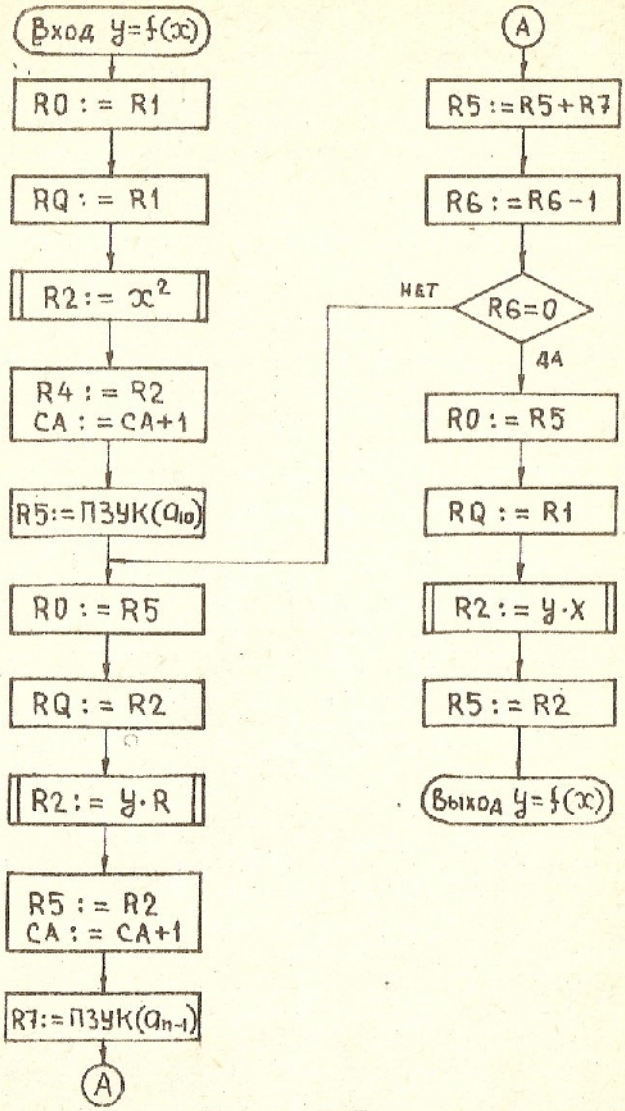

Рис. 9. Исходная блок-схема алгоритма расчета функции $y = f(x)$

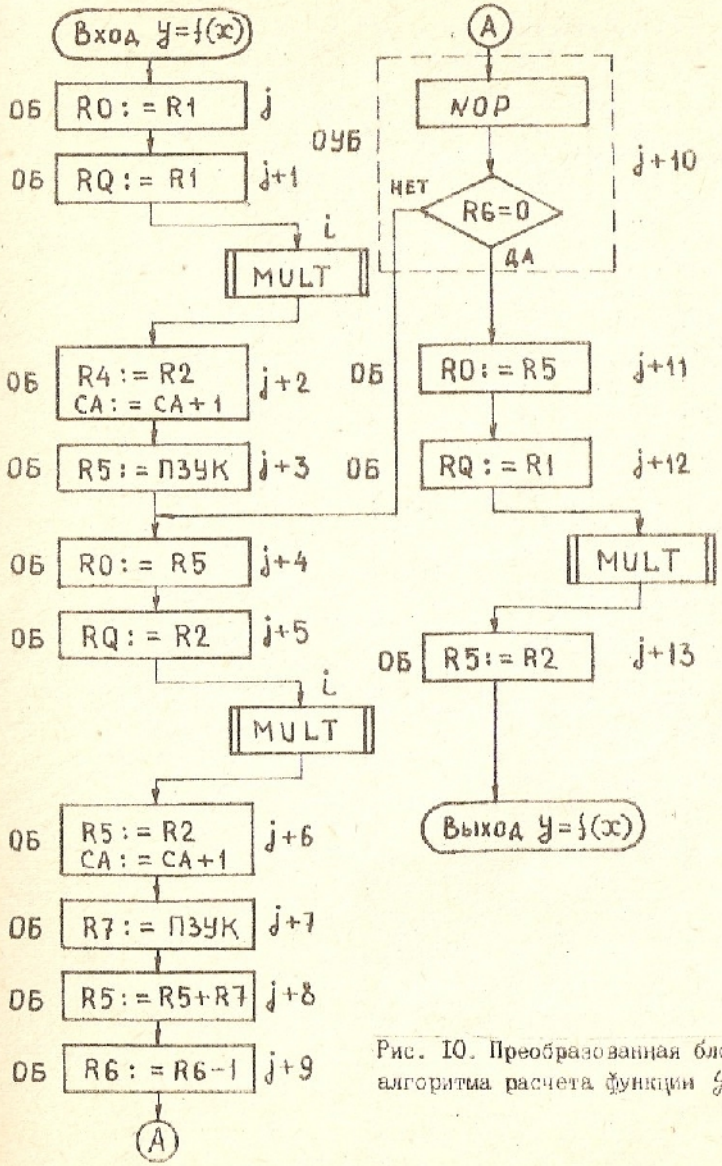

Рис. 10. Преобразованная блок-схема алгоритма расчета функции $y = f(x)$

Таблица 6

Предварительная микропрограмма вычисления функции $y = f(x)$

Адрес	Микрокоманда	
	Операционная часть	Адресная часть
j	R0 := R1	Инкремент
j+1	RQ := R1	Вызов MULT
j+2	R4 := R2, CA := CA+1	Инкремент
j+3	R5 := ПЗУК	Инкремент
j+4	R0 := R5	Инкремент
j+5	RQ := R2	Вызов MULT
j+6	R5 := R2, CA := CA+1	Инкремент
j+7	R7 := ПЗУК	Инкремент
j+8	R5 := R5+R7	Инкремент
j+9	R6 := R6-1	Инкремент
j+10	NOP	Если R6=0 j+11, иначе j+4
j+11	R0 := R5	Инкремент
j+12	RQ := R1	Вызов MULT
j+13	R5 := R2	Безус. переход на K+9

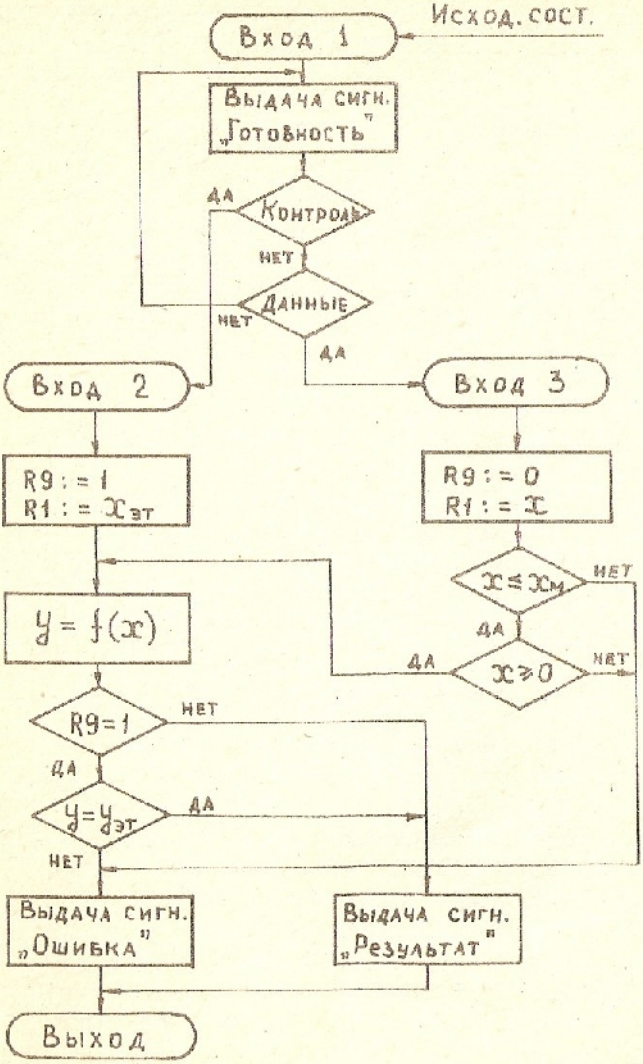

Рис. II. Упрощенный алгоритм функционирования спецвычислителя

ность", который при этом вырабатывается, подтверждает готовность системы к работе.

Подробная блок-схема алгоритма работы спецвычислителя приведена на рис. 12. Текст предварительной микропрограммы функционирования спецвычислителя представлен в табл. 7.

6. Разработка функциональной схемы спецвычислителя

На основе структурной схемы спецвычислителя (см. рис. 3) и предварительной разработки программного обеспечения можно разработать функциональную схему спецвычислителя. Синтезу подлежит только комбинационная схема преобразователя начального адреса, остальные связи следуют из технического описания микропроцессорного набора и структурной схемы из рис. 3.

Синтез схемы ПНА может быть выполнен на основе таблицы истинности ее работы и затруднений не вызывает.

Предварительная разработка программного обеспечения показывает, что в МПП необходимо иметь 50 ячеек с числом разрядов 39 (см. табл. 3). В ПЗУ констант необходимо иметь 16 ячеек разрядностью в 30 бит. Для организации ПЗУ рекомендуется использовать микросхемы К556РТ4 (организация 256 х 4) или К556РТ5 (организация 512 х 8) [14, 16]. Выходы этих микросхем выполнены с открытым коллектором. Для согласования по уровням с нагрузочными микросхемами необходимо определить величину внешнего резистора, расчет которого можно найти, например, в [12].

Фрагменты организации 32 – разрядного блока обработки данных на основе микросхем КР804ВС1 с цепями ускоренного переноса показаны на рис. 13 и 14. Для ускорения распространения переноса используются микросхемы КР804ВР1 [9, 15, 16]. Одна схема ускоренного переноса (СУП) позволяет организовать параллельные цепи переноса в блоке обработки данных разрядностью до 16. При разрядности БОД больше 16 используется каскадное включение СУП [9].

Регистр микрокоманд рекомендуется реализовывать на микросхемах КР804ИР1 [9, 16]. Одна такая микросхема представляет собой четырехразрядный параллельный регистр на D -триггерах. Наличие выходов информации с тремя состояниями позволяет использовать регистр для работы на общую шину.

В качестве счетчика адреса ПЗУК, мультиплексоров и коммутаторов могут быть выбраны стандартные микросхемы серии КР55.