

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ РАДИОТЕХНИКИ,
ЭЛЕКТРОНИКИ И АВТОМАТИКИ (ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)»

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КУРСОВАЯ РАБОТА ПО ТЕОРИИ АВТОМАТОВ

Выполнил:

студент факультета ВМС

группы ВВ-11-06

А. *. ****

Шифр: ВВ-1*-06-****

Преподаватель:

М.И. Антик.

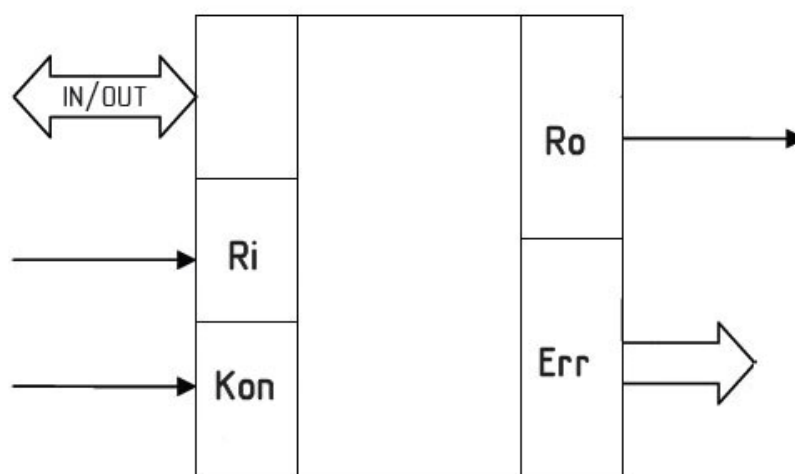
Москва, 2008

Содержание:

1. Интерфейс разрабатываемого устройства.....	3
2. Форматы данных.....	3
3. Математическое обоснование используемых алгоритмов	4
4. Микропрограмма в содержательном виде.....	5
5. Управляющие слова	5
6. Функциональные схемы операционного и управляющего автоматов..	6
7. Тестовые примеры.....	10

© Guzich, 2008

1. Интерфейс разрабатываемого устройства



Условные обозначения:

IN/OUT – шина ввода и вывода данных.

Ri – Ready input – управляющий сигнал, даёт разрешение на приём данных с **IN/OUT**-шины. Действующее значение – 1.

Ro – Ready output – управляющий сигнал, даёт разрешение на выдачу данных на **IN/OUT**-шину. Действующее значение – 1.

КОП – код операции. Управляющий сигнал, принимает следующие значения:

- **0** – автомат работает в режиме умножения двух чисел со знаком в дополнительном коде;
- **1** – автомат работает в режиме сложения двух чисел в плавающем формате.

Er – шина ошибок.

- **00** – ошибок нет;
- **01** – входные числа не нормализованы;
- **10** – переполнение счётчика.

2. Форматы данных

На **IN/OUT**-шину устройства должны поступать данные в следующих форматах:

При **КОП** = 0

31..0
Число со знаком в дополнительном коде

При **КОП** = 1

31	30..24	23..0
Знак	Порядок в смещённом коде	Нормализованная мантисса в прямом коде

3. Математическое обоснование используемых алгоритмов

1. Умножение чисел в дополнительном коде

Умножение чисел со знаком в дополнительном коде состоит из циклического анализа каждого разряда множителя, начиная с младшего, и:

- прибавления множимого к частичной сумме и последующем сдвиге частичной суммы вправо, если очередной разряд множителя равен 1,
- сдвига частичной суммы вправо, если очередной разряд множителя равен 0.

При этом сдвиг частичной суммы вправо на первом шаге осуществляется нулём, а на всех остальных шагах – знаковым разрядом множимого. Особенностью данного способа является то, что в случае отрицательного множителя необходимо делать корректировку полученного результата путём вычитания множимого в дополнительном коде. Стоит также отметить, что при перемножении двух 32-х разрядных чисел получается 64-х разрядное число, передача которого на выходную шину осуществляется за 2 такта.

2. Сложение чисел в формате с плавающей запятой

Сложение чисел в формате с плавающей запятой можно разделить на 3 этапа:

1. Выравнивание порядков чисел в сторону большего
2. Сложение мантисс
3. Нормализация результата

Если на вход автомата приходят ненормализованные числа, то автомат выдаёт ошибку $E_r = 01$.

На первом этапе происходит определение числа с максимальным порядком и циклическое увеличение порядка другого числа, одновременно со сдвигом его мантиссы вправо.

На втором этапе происходит преобразование мантисс из прямого кода в дополнительный и их подача на сумматор. При сложении мантисс одного знака возможно возникновение переполнения и соответствующей ему порчи знакового разряда суммы. Такое переполнение является устранимым путём увеличения порядка результата и сдвига мантиссы вправо. Окончательный знак определяется исходя из знаков исходных чисел.

При увеличении порядка возможно возникновение неустраняемого переполнения, выражающегося в обнулении порядка. В этом случае автомат выдаёт ошибку $E_r = 10$.

На третьем этапе происходит проверка знака мантиссы и её перевод в прямой код, а затем нормализация путём циклического сдвига мантиссы влево вместе с уменьшением итогового порядка. При этом также осуществляется контроль обнуления порядка

4. Микропрограмма в содержательном виде

M1 – ввод первого числа

M2 – ввод второго числа

M3 – обнуление счётчика

M4 – снятие данных с сумматора на регистры

M5 – сдвиг регистров числа вправо, увеличение счётчика на единицу

M6 – выдача 32 старших разрядов числа

M7 – выдача 32 младших разрядов числа

M8 – приём знака, мантиссы и порядка первого числа

M9 – приём знака, мантиссы и порядка второго числа

M10 – сдвиг первого числа вправо для выравнивания порядков

M11 – сдвиг второго числа вправо для выравнивания порядков

M12 – снятие данных с сумматора на регистры

M13 – коррекция знакового разряда

M14 – снятие данных с сумматора на регистр

M15 – нормализация полученного числа

M16 – ошибка 10

M17 – выдача результата

M18 – ошибка 10

M19 – ошибка 01

5. Управляющие слова

Y1, Y2, Y11, Y12, Y22:

(0) – хранение, (1) – запись

(Y3, Y4); (Y5, Y6); (Y14, Y15):

(0, 0) – хранение, (0, 1) – сдвиг вправо, (1, 0) – сдвиг влево, (1, 1) – запись

(Y7, Y8); (Y9, Y10):

(0, 0) – хранение, (0, 1) – инкремент, (1, 0) – декремент, (1, 1) – запись

Y13, Y16:

(0) – 0, (1) – 1

Y17, Y18, Y19:

(0) – буфер закрыт, (1) – буфер открыт

Y20 – первый бит кода ошибки

Y21 – второй бит кода ошибки

6. Функциональные схемы операционного и управляющего автоматов

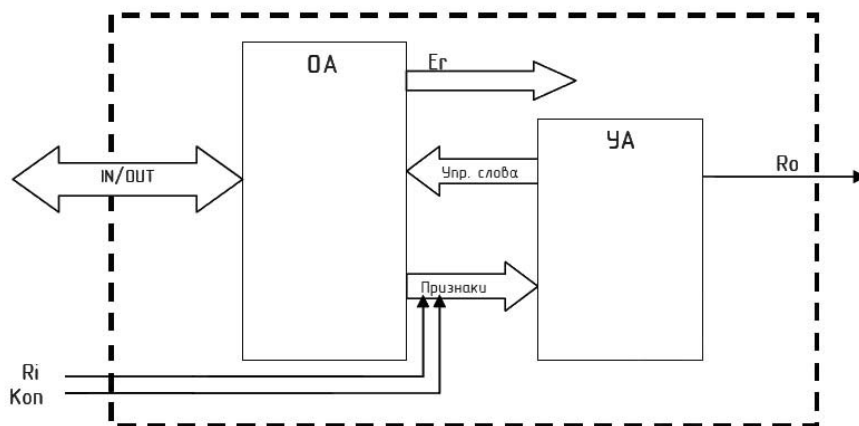


Таблица управляющих сигналов:

MY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Ro
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0								0	0	0	0				0
2	0	1	0	0	1	1	0	0								0	0	0	0				0
3	0	0	0	0	0	0	1	1								0	0	0	0				0
4														1	1	0	0	0	0				0
5					0	1								0	1	0	0	0	0				0
6	0	0	0	0	0	0											1	0	0	0	0	1	1
7																	0	1	0	0	0	1	1
8	1	0	1	1	0	0	1	1	0	0						1	0	0	0				0
9	0	1	0	0	1	1	0	0	1	1						1	0	0	0				0
10	0	0	0	1	0	0	0	1	0	0							0	0	0				0
11	0	0	0	0	0	1	0	0	0	1							0	0	0				0
12	0	0	0	0	0	0	0	0			1	1	0	1	1	1	0	0	0				0
13	0	0	0	0	0	0	0	0			0	0	0	0	1	1	0	0	0				0
14	0	0	0	0	0	0	0	0			0	0	1	1	1	1	0	0	0				0
15	0	0	0	0	0	0	1	0						1	0	1	0	0	0				0
16																	0	0	1	1	0	1	1
17																	0	0	1	0	0	1	1
18																	0	0	1	1	0	1	1
19																	0	0	1	0	1	1	1
	RG_Sa	RG_Sb	RG_A	RG_B	CT_A	CT_B	RG_S	RG_P		RG_G	КОП	BUF 1	BUF 2	BUF 3	RG_Er								

Схема операционного автомата

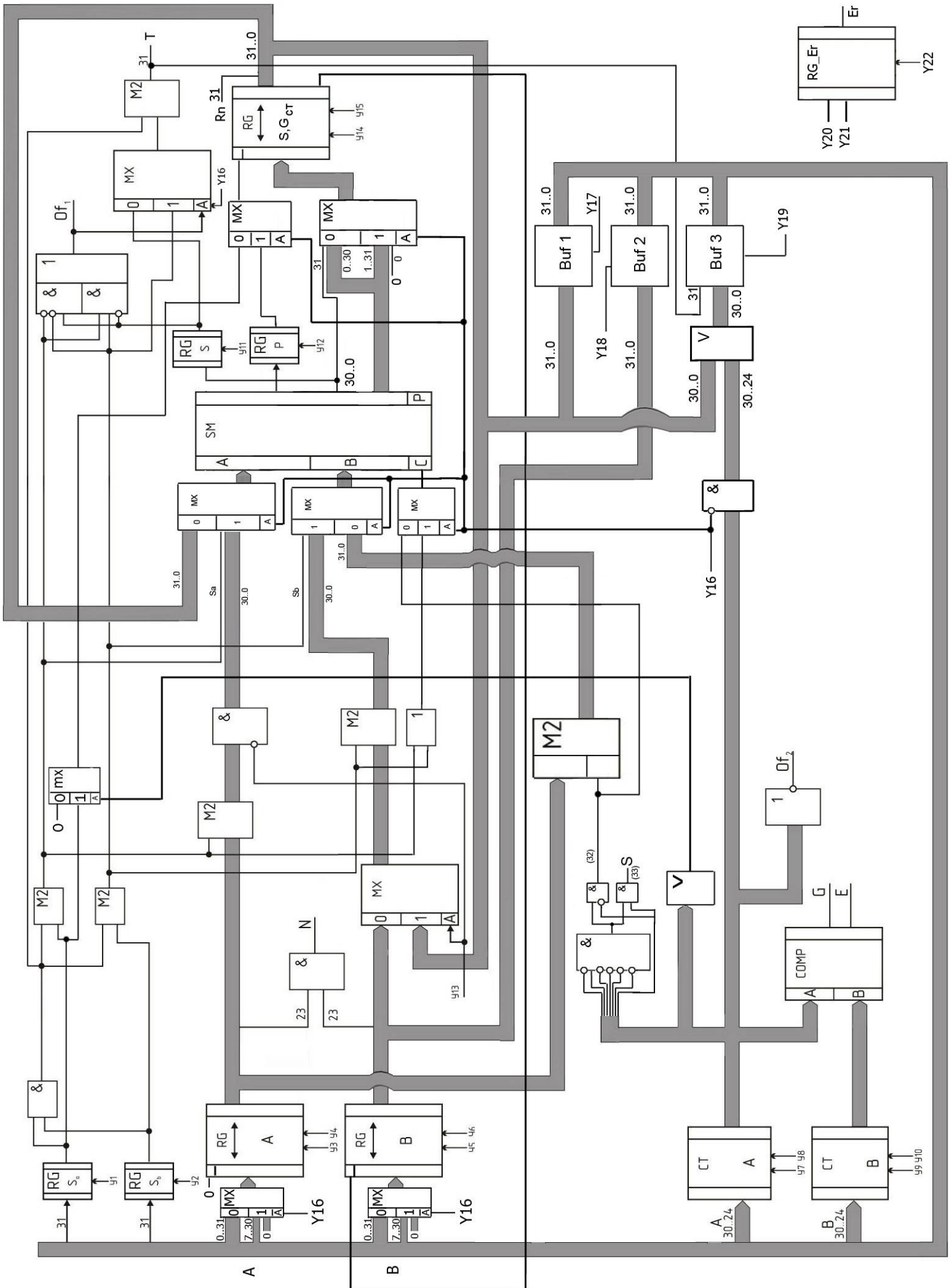


Схема управляющего автомата:

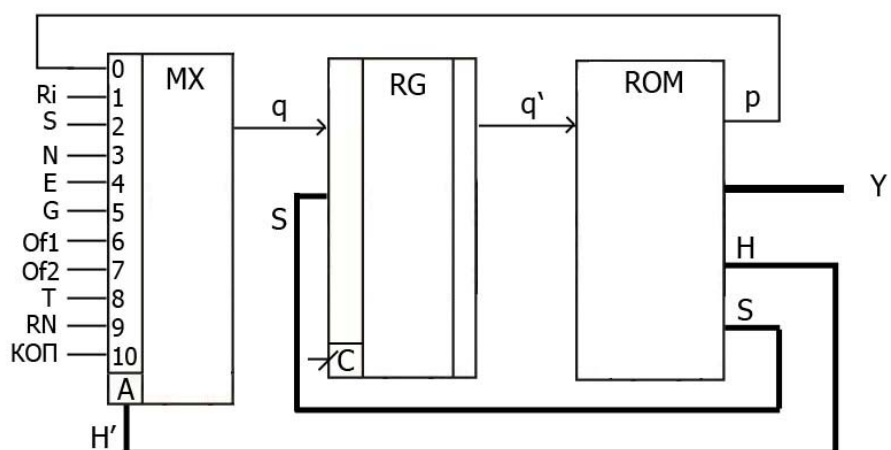
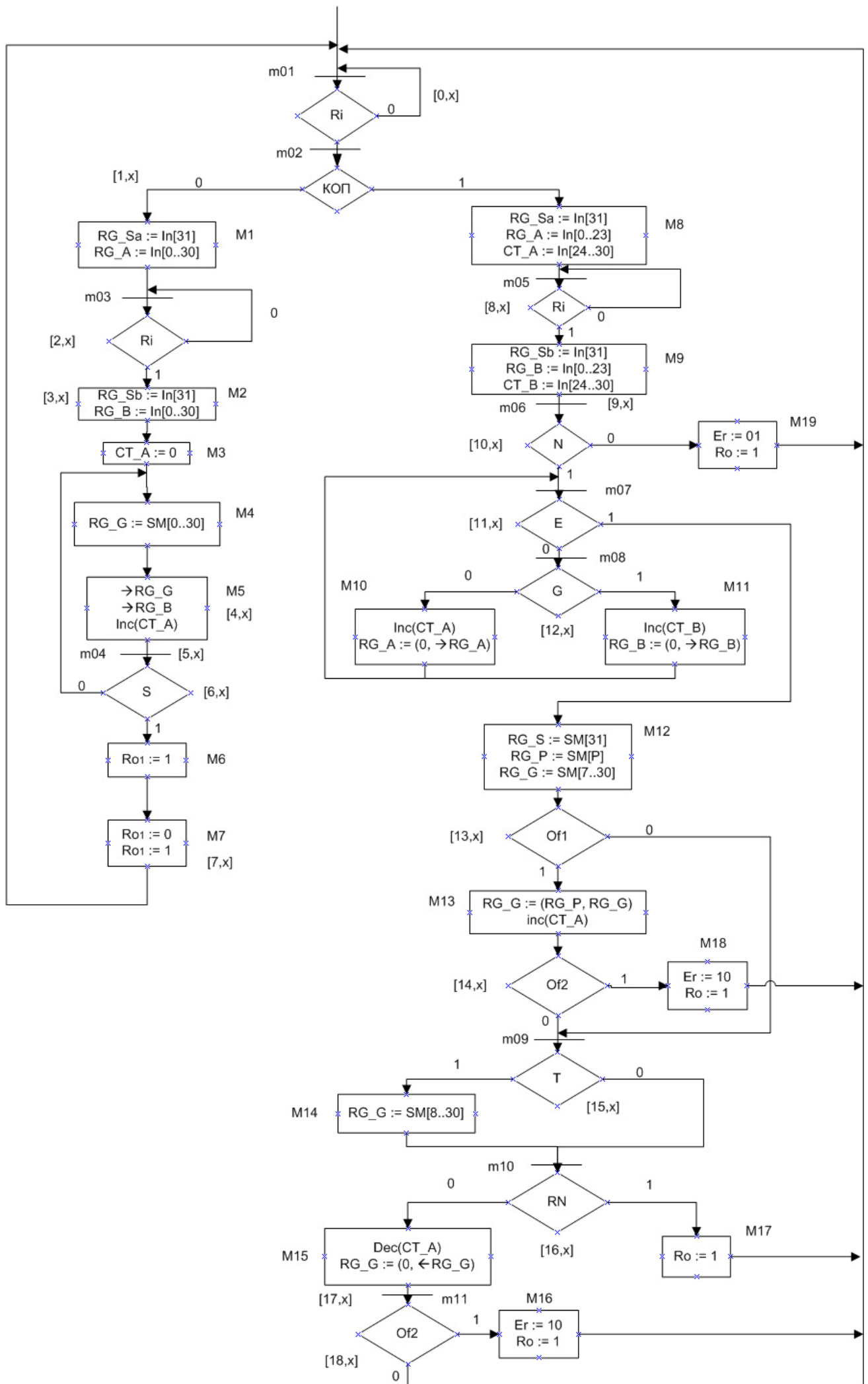


Таблица заполнения управляющей памяти:

A	Y	H	S	p	
0	0	m01	1	0	x
0	1	m02	10	1	x
1	0	M1	0	2	0
1	1	M8	0	7	0
2	0	M03	1	2	x
2	1	M2	0	3	0
3	0	M3	0	3	1
3	1	M4	0	4	0
4	0	M5	0	4	1
4	1	m04	2	5	x
5	0	M4	0	4	0
5	1	M6	0	6	0
6	0	M7	0	0	0
6	1	m06	3	8	x
7	0	m05	1	7	x
7	1	M9	0	6	1
8	0	M19	0	0	0
8	1	m07	4	9	x
9	0	m08	5	10	x
9	1	M12	6	11	x
10	0	M10	0	8	1
10	1	M11	0	8	1
11	0	m09	8	13	x
11	1	M13	7	12	x
12	0	m09	8	13	x
12	1	M18	0	0	0
13	0	m010	9	14	x
13	1	M14	0	13	0
14	0	M15	0	15	0
14	1	M17	0	0	0
15	0	m011	7	16	x
15	1				
16	0	m01	1	0	x
16	1	M16	0	0	0



7. Тестовые примеры

1. Умножение чисел в дополнительном коде

а. Множитель – положительное число

$$\begin{array}{r}
 0.1101 \quad = +13 \\
 * 0.1010 \quad = +10 \\
 \hline
 0.0000 \\
 + 0.0000 \\
 0.0000 \\
 ->0.00000 \\
 \hline
 0.1101 \\
 0.11010 \\
 ->0.011010 \\
 \hline
 + 0.0000 \\
 0.011010 \\
 ->0.0011010 \\
 \hline
 + 0.1101 \\
 1.0000010 \\
 ->0.10000010 \quad = +130
 \end{array}$$

$$\begin{array}{r}
 1.0011 \quad = -13 \\
 * 0.1010 \quad = +10 \\
 \hline
 0.0000 \\
 + 0.0000 \\
 0.0000 \\
 ->0.00000 \\
 \hline
 + 1.0011 \\
 1.00110 \\
 ->1.100110 \\
 \hline
 + 0.0000 \\
 1.100110 \\
 ->1.1100110 \\
 \hline
 + 1.0011 \\
 0.1111110 \\
 ->1.01111110 \quad = -130
 \end{array}$$

б. Множитель – отрицательное число

$$\begin{array}{r}
 0.1101 \quad = +13 \\
 * 1.0110 \quad = -10 \\
 \hline
 0.0000 \\
 + 0.0000 \\
 0.0000 \\
 ->0.00000 \\
 \hline
 + 0.1101 \\
 0.11010 \\
 ->0.011010 \\
 \hline
 + 0.1101 \\
 0.001110 \\
 ->0.1001110 \\
 \hline
 + 0.0000 \\
 0.100110 \\
 ->0.01001110 \\
 \hline
 + 1.0011 \quad \text{коррекция} \\
 1.01111110 \quad = -130
 \end{array}$$

$$\begin{array}{r}
 1.0011 \quad = -13 \\
 * 1.0110 \quad = -10 \\
 \hline
 0.0000 \\
 + 0.0000 \\
 0.0000 \\
 ->0.00000 \\
 \hline
 + 1.0011 \\
 1.00110 \\
 ->1.100110 \\
 \hline
 + 1.0011 \\
 0.110010 \\
 ->1.0110010 \\
 \hline
 + 0.0000 \\
 1.0110010 \\
 ->1.10110010 \\
 \hline
 + 0.1101 \quad \text{коррекция} \\
 0.10000010 \quad = +130
 \end{array}$$

2. Сложение чисел в плавающем формате

а. $A = 87 = 0.0000010.101011100000000000000000$
 $B = 54 = 0.0000010.110110000000000000000000$

Порядки исходных чисел равны.

Складываем мантиссы со знаками:

$$\begin{array}{r} 0.101011100000000000000000 \\ + 0.110110000000000000000000 \\ \hline 1.100001100000000000000000 \end{array}$$

Знак числа – отрицательный, а исходные числа были положительные, чего быть не может.

Выполняем корректировку порядка и мантиссы результата:

$$M[C] = 0.110000110000000000000000$$

$$P[C] = P[C] + 1 = 0000011$$

$$C = 0.0000011.110000110000000000000000$$

б. $A = -20 = 1.0000101.101000000000000000000000$
 $B = 5,5 = 0.0000011.101100000000000000000000$

Выравниваем порядки чисел в сторону большего и корректируем мантиссы:

$$B = 0.0000101.001011000000000000000000$$

Складываем мантиссы со знаками в дополнительных кодах:

$$\begin{array}{r} 1,011000000000000000000000 \\ + 0,001011000000000000000000 \\ \hline 1,100011000000000000000000 \end{array}$$

$$C = 1.0000101.100011000000000000000000 = -0.0000101.011101000000000000000000$$

Нормализуем результат:

$$C = 0.0000100.111010000000000000000000 = -14.5$$

в. $A = -181.25 = -0.0001000.101101010100000000000000 = 1.0001000.010010101100000000000000$
 $B = -131.25 = -0.0001000.100000110100000000000000 = 1.0001000.011111001100000000000000$

Складываем мантиссы со знаками в дополнительных кодах:

$$\begin{array}{r} 1.010010101100000000000000 \\ + 1.011111001100000000000000 \\ \hline 10.110001111000000000000000 \end{array}$$

Возникла единица переноса, поэтому корректируем мантиссу и увеличиваем порядок на 1:

$$C = 1.0001001.011000111100000000000000 = -0.0001001.100111000100000000000000 = -312.5$$