

ФЕДЕРАЛЬНОЕ АГЕНТСТВО 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 группы ВВ-2-06
факультета ВМС
Ravenbird

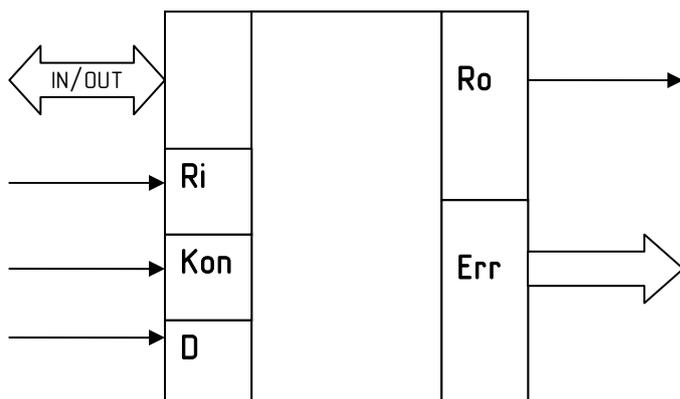
Шифр: ВВ-2-06

Преподаватель:

Антик М.И.

МОСКВА 2008

1. Интерфейс разрабатываемого устройства



IN/OUT – шина данных, одновременно является как шиной входа, так и шиной выхода.

Ri – Ready Input. Управляющий вход. При подаче данных на входы, необходимо подать на вход **Ri** единицу

Ro – Ready Output. Готовность выхода. Поступает единица, когда автомат выдает на выход конечный результат, либо при возникновении ошибки.

Kop – Код операции. Управляющий вход. Подается вместе с числами, указывает какую операцию необходимо совершить над числами.

- 0 – среднее арифметическое двух чисел, представленных в формате с плавающей запятой
- 1 – минимальное в группе чисел в дополнительном коде

D – признак конца последовательности.

D = 0 – Текущее число последовательности не последнее.

D = 1 – На вход подано последнее число последовательности

Err – шина ошибок.

- 00 – ошибок нет
- 01 - на счётчик не поступило число (**Kop** = 1)
- 10 – числа не нормализованы (**Kop** = 0)
- 11 – переполнение (**Kop** = 0)

2. Кодирование данных

Kop = 0

31	30...24	23...0
Знак	Порядок, смещённый код	Нормализованная мантисса, прямой код

Kop = 1

31..0
Число в дополнительном коде

3. Математическое обоснование используемых алгоритмов

Коп = 0, среднее арифметическое двух чисел, представленных в формате с плавающей запятой.

Среднее арифметическое чисел A и B: $(A+B)/2$

Таким образом, проводим сложение двух чисел и деление результата на 2.

Алгоритм:

1. Выравнивание порядков чисел.
 2. Сложение мантисс.
 3. Нормализация результата.
 4. Уменьшение порядка на 1 (т.е. деление числа на 2).
1. Определяется число с большим порядком и циклически увеличивается порядок другого числа с одновременным сдвигом мантиссы вправо.
 2. Происходит перевод чисел из прямого числа в дополнительный для их сложения на сумматоре. При сложении мантисс с одинаковыми знаками, может произойти переполнение, которое приводит к порче знакового разряда. Это переполнение устраняется сдвигом мантиссы результата вправо с одновременным внесением разряда переполнения с сумматора. Итоговый знак определяется исходя из знаков исходных чисел. При увеличении порядка может произойти неустранимое переполнение на счётчике (обнуление счётчика).
 3. Проверка мантиссы и перевод её в прямой код (если требуется), нормализация мантиссы путём циклического сдвига её влево с одновременным уменьшением порядка. Производится контроль обнуления порядка.
 4. Деление числа на 2 путём уменьшения порядка на 1, при этом производится контроль обнуления счётчика.

Коп = 1, минимальное в группе целых чисел в дополнительном коде.

Алгоритм:

1. Принимаем первое число массива за минимальное.
2. Сравниваем следующее число с минимальным; если оно меньше, то принимаем это число за минимальное.
3. Повторение происходит до конца последовательности.

Максимальный размер массива – $2^7 = 128$ чисел. В случае превышения количества чисел, операция заканчивается и на выход поступает минимальное на данный момент число.

4. Микропрограммы в содержательном виде

m1 – ввод первого числа
m2 – ввод второго числа
m3 – сдвиг первого числа вправо (выравнивание порядков)
m4 – сдвиг второго числа вправо (выравнивание порядков)
m5 – снятие данных сумматора на регистры
m6 – коррекция знакового разряда
m7 – снятие данных сумматора на регистр
m8 – нормализация
m9 – деление на два
m10 – выдача результата
m11 – ошибка 10
m12 – ошибка 11
m13 – ошибка 11
m14 – запись числа на счётчик
m15 – ввод первого числа
m16 – ввод очередного числа
m17 – принятие другого числа за минимальное
m18 – ввод очередного числа
m19 – принятие другого числа за минимальное
m20 – выдача результата
m21 – ошибка 01
m22 – ввод первого числа
m23 – выдача результата
m24 – выдача результата
m02 – обнуление регистра ошибок
m012 – перевод числа в прямой код

5. Управляющее слово:

y1, y2, y11, y12, y17, y18, y21

(0) – хранение; (1) – запись;

(y3, y4), (y5, y6)

(0,0) – хранение; (0,1) – сдвиг вправо; (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;

(y14, y15)

(0,0) – хранение; (0,1) – последовательная запись;

(1,0) – сдвиг влево; (1,1) – параллельная запись;

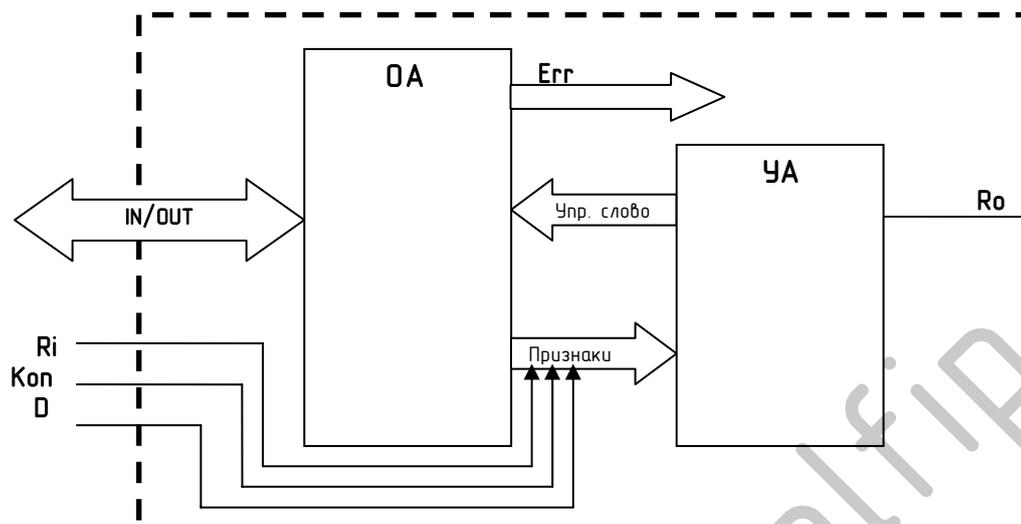
y19, y20

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

y22 – первый бит кода ошибки

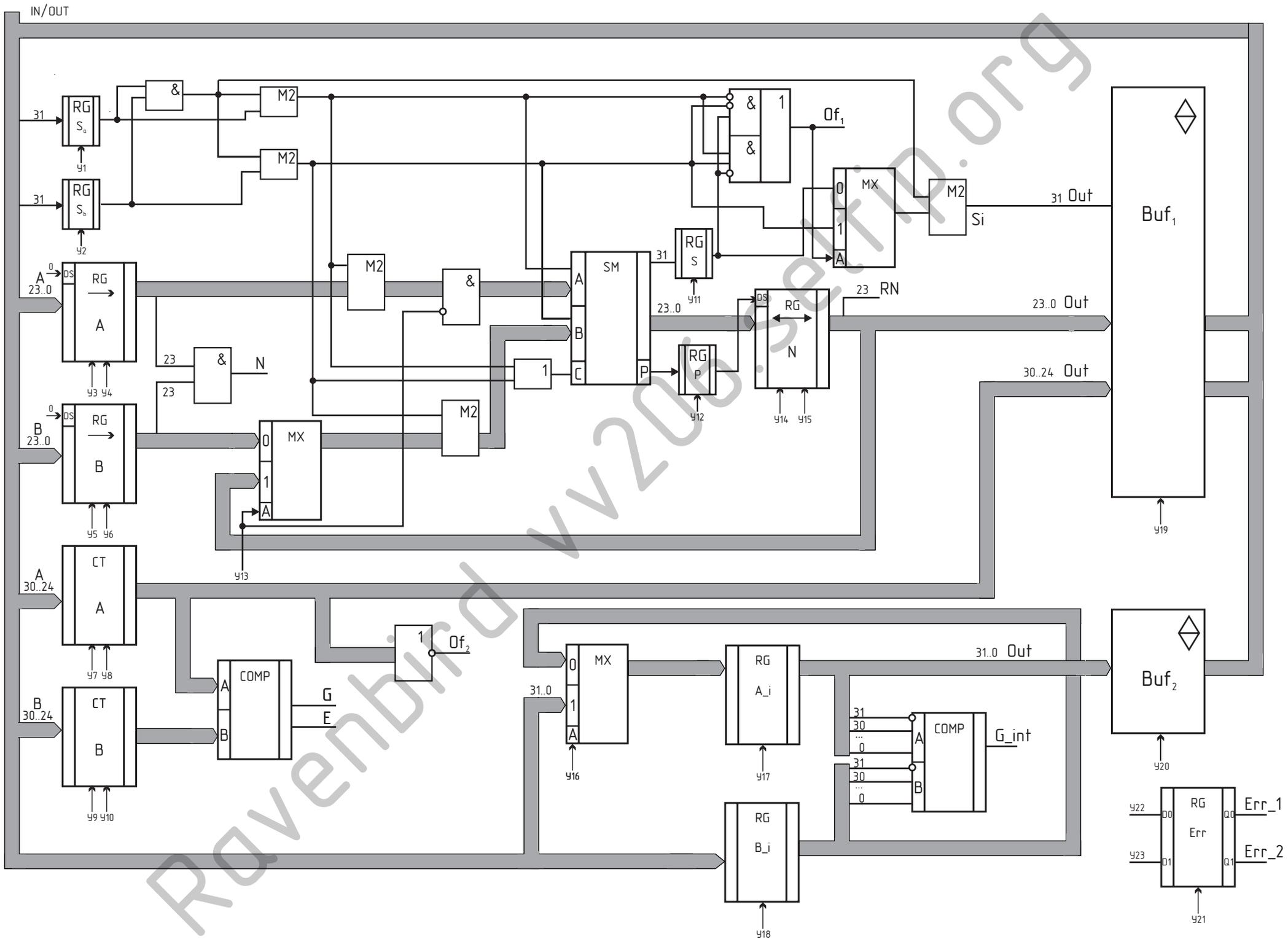
y23 – второй бит кода ошибки

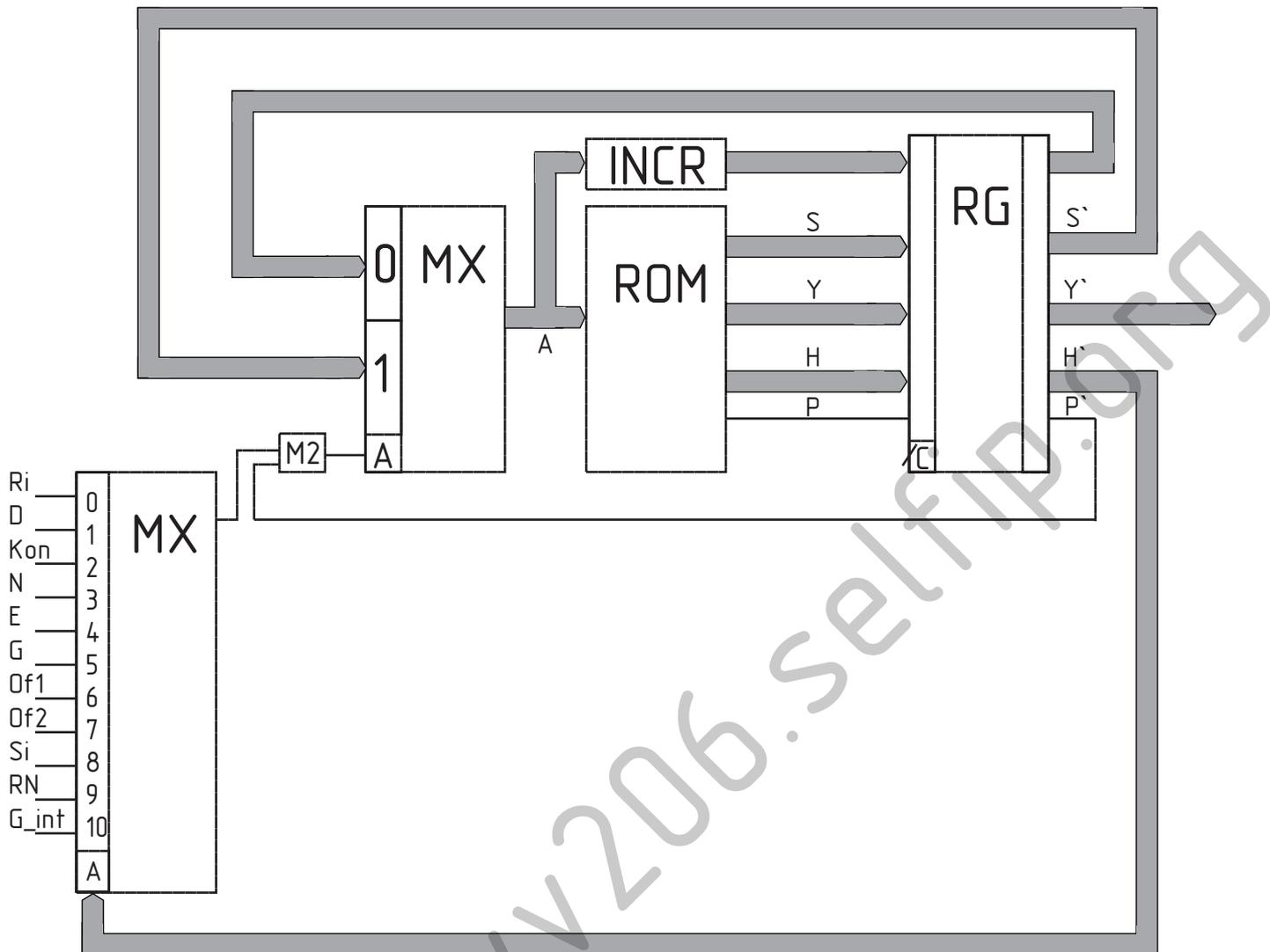
6. Функциональные схемы операционного и управляющего автомата



Далее в приложениях находятся (или не находятся ☺):

- блок-схема алгоритма (без нумерации блоков для УА)
- схема операционного автомата
- схема управляющего автомата
- таблица ROM управляющего слова
- таблица ROM памяти (не заполнена – т.к. писал карандашом и сдал, теперь лень восстанавливать)
- нет тестовых примеров (писал так же карандашом и сдал)





Управляющий автомат.

Регулярная адресация, конвейерный вариант взаимодействия.

В этой схеме при разветвлении процесса вычислений пара альтернативных адресов получается следующим образом: один адрес всегда на единицу больше, чем текущий (т.е. изменяется «регулярным» образом), второй адрес - произвольный и содержится вместе с микрокомандой в микроинструкции.

Элементом, «вычисляющим» адрес, служит счетчик, управляемый входным для УА сигналом. При различных значениях входного сигнала счетчик выполняет две функции: либо прибавляет единицу к значению, которое хранилось в счетчике и являлось текущим адресом, либо загружается значением адреса из управляющей памяти.

В схему введен элемент M2, позволяющий инвертировать значение входного сигнала, что облегчает распределение микроинструкций по ячейкам управляющей памяти.

Для конвейерного варианта взаимодействия регулярное изменение адреса приходится организовывать так, чтобы не увеличивать число мест конвейера.

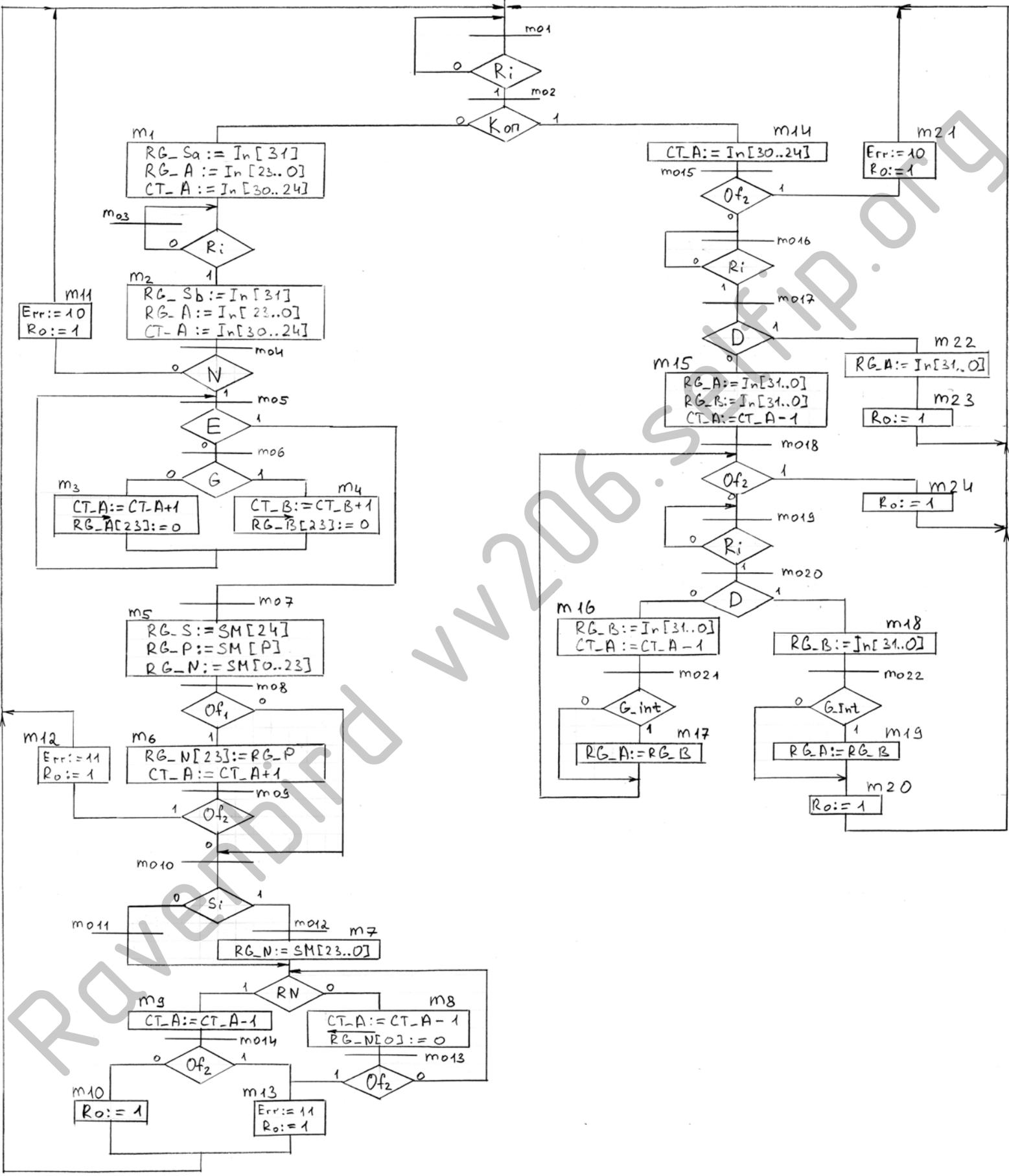


Таблица заполнения управляющей памяти №1

mly	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11	y12	y13	y14	y15	y16	y17	y18	y19	y20	y21	y22	y23	Ro
m1	1	0	1	1	0	0	1	1	0	0									0	0	0			0
m2	0	1	0	0	1	1	0	0	0	1	1								0	0	0			0
m3	0	0	0	1	0	0	0	0	1	0	0								0	0	0			0
m4	0	0	0	0	0	0	1	0	0	0	1								0	0	0			0
m5	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1			0	0	0			0
m6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1			0	0	0			0
m7	0	0						0	0	0	0	0	0	0	1	1			0	0	0			0
m8	0	0						1	0	0	0	0	0	0	1	0			0	0	0			0
m9	0	0						1	0	0	0	0	0	0	0	0			0	0	0			0
m10	0	0						0	0	0	0				0	0			1	0	0			1
m11	0	0						0	0	0	0				0	0			1	0	1	1	0	1
m12	0	0						0	0	0	0				0	0			1	0	1	1	1	1
m13	0	0						0	0	0	0				0	0			1	0	1	1	1	1
m14								1	1										0	0	0			0
m15								1	0								1	1	1	0	0	0		0
m16								1	0									0	1	0	0	0		0
m17								0	0								0	1	0	0	0	0		0
m18																		0	1	0	0	0		0
m19																	0	1	0	0	0	0		0
m20																		0		0	1	0		1
m21																		0	0	1	1	0	1	1
m22																	1	1	0	0	0	0		0
m23																		0		0	1	0		1
m24																		0		0	1	0		1
m02																				0	0	1	0	0
m012													1							0	0	0		0

Таблица заполнения управляющей памяти №2

A	Y	S	H	P
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				

RUENBING VV206.Selfip.org