

КОНТРОЛЛЕРЫ
МАЛОКАНАЛЬНЫЕ МНОГОФУНКЦИОНАЛЬНЫЕ
РЕГУЛИРУЮЩИЕ МИКРОПРОЦЕССОРНЫЕ РЕМИКОНТЫ Р-І30

Техническое описание

ЯЛБИ.421457.001 ТОІ

(2Яа.399*550 ТОІ)

Часть І

СОДЕРЖАНИЕ	Стр.
1.1. Алгоритмы лицевой панели	4
1.2 Библиотека стандартных конфигураций.....	5
1.3. Условные обозначения	6
1.4. Режимы работы алгоритмов и безударность	6
1.5. Диапазон изменения параметров настройки.	8
1.6. Применение стандартных конфигураций	8
2. Алгоритмы лицевой панели	8
2.1. ОКО(01) - Оперативный контроль регулирования	8
2.2. ОКЛ(02) - Оперативный контроль логической программы.....	11
2.3. ОНД (03) - Оперативный непрерывно-дискретный контроль	15
2.4. ДИК(04) - Дискретный контроль.....	18
3. Алгоритм ввода-вывода.....	20
3.1. ВИН(05) - Ввод интерфейсный	20
3.2. ИНВ(06) - Интерфейсный вывод	22
3.3. ВАА(07), ВАБ(08) - Ввод аналоговый групп А и Б	24
3.4. ВДА(09) ВДБ(10) - Ввод дискретный групп А и Б	26
3.5. АВА(11) АВБ(12) – Аналоговый вывод групп А и Б.....	28
3.6. ДВА(13) ДВБ(14) - Дискретный вывод групп А и Б.....	30
3.7. ИВА(15) ИВБ(16) - Импульсный вывод групп А и Б	32
3.8. АВР(17) - Аварийный вывод.....	34
4. АЛГОРИТМЫ РЕГУЛИРОВАНИЯ	36
4.1. РАН(20) - Регулирование аналоговое.....	36
4.2. РИМ(21) - Регулирование импульсное.....	40
4.3. ЗДН(24) – Задание	43
4.4. ЗДЛ(25) - Задание локальное	46
4.5. РУЧ(26) – Ручное управление	48
4.6. ПРЗ(27) – Программный задатчик.....	50
4.7. ИНЗ(28) - Интегрирующий задатчик.....	53
4.8. ПОК(29) - Пороговый контроль	55
4.9. АНР(30) - Автонастройка регулятора.....	57
5. Динамические преобразования	65
5.1. ИНТ(33) - Интегрирование.....	65
5.2 ДИФ(34) – Дифференцирование.....	67
5.3. ФИЛ(35) - Фильтр	69
5.4. ДИН(36) - Динамическое преобразование.....	71
5.5. ДИБ(37) - Динамическая балансировка	73
5.6. ОГС(38) - Ограничение скорости	75
5.7. ЗАП(39) - Запоздывание	78
6. Статические преобразования	81
6.1. СУМ(42) – Суммирование.....	81
6.2. СМА(43) - Суммирование с масштабированием.....	83
6.3. УМД(44) -Умножение-деление.....	85
6.4. КОР(45) - Корень квадратный	86
6.5. МОД(46) - Модуль	89
6.6. КУС(47) - кусочно-линейная функция	91
6.7. ОГР(48) - Ограничение	93
6.8. СКС(49) - Скользящее среднее	95
6.9. ДИС (50) - Дискретное среднее	97
6.10. МИН (51) - Минимум.....	99

6.11. МКС (52) - Максимум	101
6.12. СИТ (53) Средний из трех	103
6.13. ЭКС (54) - Экстремум	105
6.14. МСШ (55) - Масштабирование	109
7. Аналого-дискретные преобразования	111
7.1. ПЕР (57) - Переключатель с дискретным управлением	111
7.2. ПЕН (58) - Переключение по номеру	113
7.3. ПОР (59) - Пороговый элемент	115
7.4. НОР (60) - Нуль-орган	117
7.5. ИМП (61) - Импульсатор	120
7.6. ЗАИ (62) - Запрет изменения	123
7.7. ЗАЗ (63) - Запрет знака	127
7.8. СЛЗ (64) - Слежение-запоминание	129
7.9. ЗПМ (65) - Запоминание	131
7.10. БОС (66) - Блокировка обратного счета	133
7.11. ВОТ (67) - Выделение отключения	135
8. Логические операции	137
8.1. ЛОИ (70) - Логическое И	137
8.2. МНИ (71) - Многовходовое И	139
8.3. ИЛИ (72) - Логическое ИЛИ	141
8.4. МИЛ (73) - Многовходовое ИЛИ	143
8.5. ИИЛ (74) - Исключающее ИЛИ	145
8.6. МАЖ (75) - Мажорирование	147
8.7. ТРИ (76) - RS - Триггер	149
8.8. РЕУ (77) - Регистр с записью по уровню	151
8.9. РЕФ (78) - Регистр с записью по фронту	153
8.10. ВЫФ - Выделение фронта (79)	155
9. Дискретное управление	157
9.1. ЭТП (80) - Этап	157
9.2. ТМР (81) - Таймер	161
9.3. СЧТ (82) - Счетчик	163
9.4. ОДВ (83) - Одновибратор	165
9.5. МУВ (84) - Мультивибратор	167
9.6. ПЧИ (85) - Переключение чисел	169
9.7. СЧИ (86) - Сравнение чисел	171
9.8. ВЧИ (87) - Выделение чисел	173
9.9. УДП (88) - Управление двухпозиционной нагрузкой	175
9.10. УТП (89) - Управление трехпозиционной нагрузкой	177
9.11. ШИФ (90) - Шифратор	179
9.12. ДЕШ (91) - Дешифратор	181
10. ГРУППОВОЕ НЕПРЕРЫВНО-ДИСКРЕТНОЕ УПРАВЛЕНИЕ	185
10.1. ШАП (94) - шаговая программа	185
10.2. ГРА (95) - групповое управление аналоговыми сигналами	189
10.3. ГВД (96) - групповое управление входными дискретными сигналами	191
10.4. ГДВ (97) - групповое управление дискретными выходными сигналами	193
10.5. ГРУ (98) - групповое ручное управление	196
10.6. ГРК (99) - групповой контроль	199
11. Стандартные конфигурации	201
11.1. РЕГА(01) - Регулятор аналоговый (стандартная конфигурация)	201
11.2. РЕГИ(02) - Регулятор импульсный (стандартная конфигурация)	203

1. Библиотека алгоритмов и стандартных конфигураций

1.1. Алгоритмы лицевой панели

Код	Шифр	Наименование	Авт. рег.	Лог. упр.	Непр.-дискр, упр.
01	ОКО	Оперативный контроль регулирования	+	-	-
02	СКЛ	Оперативный контроль логической программы	+	-	-
03	СКД	Оперативный непрерывно-дискретный контроль			
04	ДИК	Дискретный контроль	+	+	+
2. Алгоритмы ввода-вывода					
05	ВИН	Ввод интерфейсный	+	+	+
06	ИНВ	Интерфейсный вывод	+	+	+
07	ВАА	Ввод аналоговый группы А	+	+	+
08	ВАБ	Ввод аналоговый группы Б	+	+	+
09	ВДА	Ввод дискретный группы А	+	+	+
10	ВДБ	Ввод дискретный группы Б	+	+	+
11	АВА	Аналоговый вывод группы А	+	+	+
12	АВБ	Аналоговый вывод группы Б	+	+	+
13	ДВА	Дискретный вывод группы А	+	+	+
14	ДВБ	Дискретный вывод группы Б	+	+	+
15	ИВА	Импульсный вывод группы А	+	+	+
16	ИВБ	Импульсный вывод группы Б	+	+	+
17	АВР	Аварийный вывод	+	+	+
3. Алгоритмы регулирования					
20	РАН	Регулирование аналоговое	+	+	+
21	РИМ	Регулирование импульсное	+	+	+
24	ЗДН	Задание	+	-	-
25	ЗДЛ	Задание локальное	+	-	-
26	РУЧ	Ручное управление	+	-	-
27	ПРЗ	Программный задатчик	+	+	+
28	ИНЗ	Интегрирующий задатчик	+	+	+
29	ПОК	Пороговый контроль	+	+	+
30	АНР	Автонастройка регулятора	+	+	+
4. Динамические преобразования					
33	ИНТ	Интегрирование	+	+	+
34	ДИФ	Дифференцирование	+	+	+
35	ФИЛ	Фильтрация	+	+	+
36	ДИН	Динамическое преобразование	+	+	+
37	ДИБ	Динамическая балансировка	+	+	+
38	ОГС	Ограничение скорости	+	+	+
39	ЗАП	Запаздывание	+	+	+
5. Статические преобразования					
42	СУМ	Суммирование	+	+	+
43	СМА	Суммирование с масштабированием	+	+	+
44	УМД	Умножение-деление	+	+	+
45	КОР	Корень квадратный	+	+	+
46	МОД	Модуль	+	+	-
47	КУС	Кусочно-линейная функция	+	+	+
48	ОГР	Ограничение	+	+	+
49	СКС	Скользящее среднее	+	+	-
50	ДИС	Дискретное среднее	+	+	-
51	МИН	Минимум	+	+	+
52	МКС	Максимум	+	+	+
53	СИТ	Средний из трех	+	+	-
54	ЭКС	Экстремум	+	+	+
55	МСШ	Масштабирование	+	+	+
6. Аналого-дискретные преобразования					
57	ПЕР	Переключатель с дискр. управл.	+	+	+

58	ПЕН	Переключатель по номеру	+	+	+
59	ПОР	Пороговый элемент	+	+	+
60	НОР	Нуль-орган	+	+	+
61	ИМП	Импульсатор	+	+	+
62	ЗАИ	Запрет изменения	+	+	+
63	ЗАЗ	Запрет знака	+	+	+
64	СЛЗ	Слежение запоминание	+	+	+
65	ЗПИ	Запоминание	+	+	+
66	БОС	Блокировка обратного кода	+	+	+
67	ВОТ	Выделение отключения	+	+	+
7. Логические операции					
70	ЛОИ	Логическое И	+	+	+
71	МНИ	Многовходовое И	+	+	+
72	ИЛИ	Логическое ИЛИ	+	+	+
73	МИЛ	Многовходовое ИЛИ	+	+	+
74	ИИЛ	Исключающее ИЛИ	+	+	+
75	МАЗ	Мажорирование	+	+	-
76	ТРИ	Триггер	+	+	+
77	РЕУ	Регистр с записью по уровню	+	+	+
78	РЕФ	Регистр с записью по фронту	+	+	+
79	ВЫФ	Выделение фронта	+	+	+
8. Дискретное управление.					
80	ЭТП	Этап шаговой программы	-	+	-
81	ТМР	Таймер	+	+	+
82	СЧТ	Счетчик	+	+	+
83	ОДВ	Одновибратор	+	+	+
84	МУВ	Мультивибратор	+	+	+
85	ПЧИ	Переключатель чисел	+	+	+
86	СЧИ	Сравнение чисел	+	+	+
87	ВЧИ	Выделение чисел	+	+	+
88	УДП	Управление двухпозиционной нагрузкой	-	+	+
89	УТП	Управление трехпозиционной нагрузкой	-	+	+
90	ШИФ	Шифратор	+	+	+
91	ДЕШ	Дешифратор	+	+	+
92	ЛОК	Локальный контроль	+	+	+
9. Групповое непрерывно-дискретное управление					
94	ШАП	Шаговая программа	-	-	+
95	ГРА	Групповое управление аналоговыми сигналами	-	-	+
96	ГВД	Групповое управление дискретными сигналами	-	-	+
97	ГДВ	Групповое управление дискретными выходными сигналами	-	-	+
98	ГРУ	Групповое ручное управление	-	-	+
99	ГРК	Групповой контроль	-	-	+

Примечания: 1. Знак "+" означает наличие данного алгоритма в библиотеке, знак «-» его отсутствие.
2. Алгоритм с кодом 00 является «пустым» (не выполняющим никаких операций).

1.2 Библиотека стандартных конфигураций

Код	Шифр	Наименование	Авт. рег.	Лог. упр.	Непр.- дискр. упр.
01	РЕГА	Регулятор аналоговый	+	-	-
02	РЕГИ	Регулятор импульсный	+	-	-

Примечания: 1. Знак "+" означает наличие данной стандартной конфигурации (СК) в библиотеке, знак «-» его отсутствие.
2. СК с кодом 00 является «пустой»; ее ввод приводит к полному обнулению.

1.3. Условные обозначения

При описании библиотеки алгоритмов используется единая система обозначений.

Сигналы и параметры настройки обозначаются следующим образом:

X , Y - аналоговые сигналы и параметры соответственно на входе и выходе алгоритма;

C , D - дискретные сигналы соответственно на входе на выходе алгоритма;

T - интервалы времени и постоянные времени;

N - числовые сигналы и параметры;

K - масштабные коэффициенты, коэффициенты пропорциональности;

V - скорость изменения сигналов и параметры, задающие скорость изменения;

W - технические единицы;

Z - сигналы, формат которых додается при программировании.

Для обозначения нескольких одноименных сигналов и параметров используется индексация (например, X_1 , Y_6 , K_n и т. д.).

Каскадные входы и выходы на функциональной схеме алгоритма помечаются буквой "K", помещенной в скобках (например, $X_2(K)$, $Y(K)$ и т.д.).

Начальное значение сигнала, которое передается на каскадный выход алгоритма, обозначается Y_0 .

На функциональных схемах используются сокращения:

m - модификатор;

MB - масштаб времени.

1.4. Режимы работы алгоритмов и безударность

В составе библиотеки Р-130 имеется группа алгоритмов, которая реагирует на какие-либо дискретные события или регистрирует достижение аналоговым сигналом установленных уровней ограничения. К этой группе относятся алгоритмы ручного управления, изменения вида задания, переключения, ограничения и т.д.

Указанные алгоритмы инициируют переход предвключенных алгоритмов в один из двух режимов: отключения или запрета. В этих режимах работа алгоритмов изменяется таким образом, чтобы возврат в стандартный режим работы произошел безударно.

Два безударного изменения режимов в Р-130 предусмотрена процедура так называемого обратного счета. Суть ее заключается в том, что команда изменения режима вместе с вычисленным значением начальных условий передается от одного алгоритма к другому против основного потока информации, то есть от входа одного к выходу другого - предвключенного - алгоритма.

Обратный счет реализуется лишь в том случае, когда в последовательной цепочке алгоритмов есть хотя бы один алгоритм, который в состоянии отслеживать и запоминать результат обратного счета. Такие алгоритмы называются следящими.

К следящим относятся, следующие алгоритмы:

РАН - регулирование аналоговое;

РИМ - регулирование импульсное;

ИНТ - интегрирование;

ДИБ - динамическая балансировка.

Ряд алгоритмов становятся следящими при определенных условиях. Например, алгоритм ЗДН становится следящим при наличии динамической или статистической балансировки.

Если алгоритм является следящим, это оговаривается при его описании.

Если в последовательной цепочке нет следящих алгоритмов, обратный счет не реализуется.

В общем случае алгоритм имеет несколько входов и выходов, однако в процедуре обратного счета участвуют лишь определенные входы и выходы, называемые каскадными. Алгоритм может содержать несколько каскадных входов, но каскадный выход всегда один. Это означает, что алгоритм получает начальные условия лишь по одной цепи

Если перед алгоритмами, управляющими режимом, включена цепочка других: алгоритмов имеющих каскадные входы-выходы, в процедуре обратного счета участвуют все эти алгоритмы. Алгоритмы, не имеющие каскадных входов-выходов, блокируют процедуру обратного счета.

Каскадные входы к выходам участвуют в обратном счете лишь в режимах отключения и запрета. В стандартном режиме они наряду с остальными входами и выходами выполняют обычные функции предписанные конкретным алгоритмам.

Особенности обратного счета иллюстрируют следующие примеры.

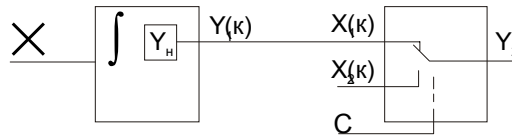
В схеме рис. 1 (а) после следящего алгоритма интегрирования включен алгоритм переключения. Выход интегратора Y_1 , входы X_1 ; X_2 и выход, Y_2 переключателя являются каскадными.

В стандартном режиме (сигнал на входе $C = 0$) справедливы следующие соотношения:

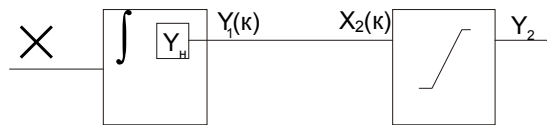
$$Y_1 = Y_n = \int X_1 dt + Y_0; X_1 = Y_1; Y_2 = X_1, \quad (1)$$

где Y_n и Y_0 -соответственно значение сигналов в интегральной ячейке и начальное значение этого сигнала.

Особенности режимов работы



а)



б)

а) работа в режиме отключения;

б) работа в режиме запрета

Рис. 1

Если на вход C поступает дискретный сигнал $C = 1$, переключатель отключает интегратор, при этом на входе X_1 переключателя формируется команда отключения вместе с значением начальных условий $X_1 = Y_2$. Эти сигналы передаются на выход интегратора, в результате чего устанавливается $Y_1 = Y_2$. Интегратор переходит в режим отключения, при этом интегрирование входного сигнала X прекращается, а интегральная ячейка отслеживает сигнал Y_1 , то есть $Y_n = Y_1$. В результате начальные условия для интегратора $Y_0 = Y_2 = X_2$ и включение интегратора произойдет безударно.

В другом примере (рис. 1 б)) до тех пор, пока выходной сигнал интегратора Y_1 не вышел за область ограничения, $Y_2 = Y_1 = X_2$. Как только сигнал X_2 сравнивается с порогом ограничения $X_{\max}$ на каскадном входе X_2 формируется команда запрета на увеличение. Эта команда передается на каскадный выход Y_1 интегратора и увеличение сигнала в интегральной ячейке Y_n блокируется. При перемене знака сигнала X и интегратор, и ограничитель переходят в стандартный режим работы. Аналогично ведет себя схема при достижении порога ограничения $X_{\min}$ - с той лишь разницей, что формируется команда запрета на уменьшение.

Таким образом, при отключении в результате обратного счета интегратору передается два сигнала: дискретный сигнал (команда) отключения и аналоговое значение начальных условий, причем оба сигнала передается по одной и той же связи между алгоритмами и через одни и те же входы-выходы. При запрете передается лишь команда запрета.

В системе состоящей из нескольких алгоблоков, могут встретиться следующие два случая

1. В последовательной цепочке один алгоблок вырабатывает команду запрета, другой - отключения. В этом случае алгоблок, который вырабатывает команду запрета и получает через свой каскадный выход команду отключения, транслирует предвключенному алгоритму команду отключения. Если алгоритм, вырабатывающий команду отключения, получает через свой каскадный выход команду запрета, то он также транслирует предвключенному алгоритму команду отключения. В этом смысле команда отключения приоритетна, над командой запрета.
2. К выходу одного алгоблока могут быть подключены несколько алгоблоков, вырабатывающих команды отключения и (или) запрета. В этом случае приоритет отдается не виду команды, а алгоблоку, ближайшему по номеру к данному алгоблоку. Если алгоблоки, вырабатывающие режимные команды, имеют номера как меньше, так и больше номера данного алгоблока, то приоритетен алгоблок с минимальным номером в группе старших номеров.

Процедура обратного счета учитывает инверсию сигнала на каскадном входе. Например, если в схеме рис. 1 б) на входе X_2 установлена инверсия и достигнуто ограничение по максимуму, интегратору передается команда запрета на уменьшение.

1.5. Диапазон изменения параметров настройки.

Диапазон изменения параметров настройки для большинства алгоритмов одинаков и определяется таблицей 2.

Таблица 2

Параметры настройки	Обозначение	Размерность	Диапазон изменения
Уровень аналоговых сигналов	X	%	-199,9...199,9
Постоянные времени	T*	S, min h	0...819,0 и ∞
Число	N	-	-8191 ... 8191
Масштабный коэффициент	K _м	-	-15,99 ... 15,99
Коэффициент пропорциональности	K _п	-	-127,9 ... 127,9
Скорость изменения	V*	%/ s %/ min %/ h	0-199,9 и ∞
Минимальная длительность импульса	T	S	0,12...3,84
Технические единицы	W	-	-1999 ... 8191

% - Для всего контроллера задается 1-й (младший) или 2-й (старший) диапазон изменения параметров; для младшего диапазона индивидуально в каждом алгоритме задается масштаб (поддиапазон) "секунды" или "минуты", для старшего диапазона - "минуты" или «часы».

Если диапазон изменения параметров является нестандартным, он оговаривается при описании алгоритма.

Диапазон изменения аналоговых, временных и числовых сигналов совпадает с диапазоном изменения соответствующих параметров. Дискретные (логические) сигналы принимают значения 0 или 1.

Если в процессе изменения временных параметров (коэффициентов) устанавливается $T < 0$, это воспринимается алгоритмом как $T = 0$.

1.6. Применение стандартных конфигураций

Стандартные конфигурации (СК) (РЕГА, РЕГИ) позволяют сократить объем программирования. СК состоит из заранее сконфигурированного набора библиотечных алгоритмов и вводится в контроллер в процедуре «приборные параметры» (см. техническое описание 2.399.550 ТО).

С помощью СК можно ввести набор алгоритмов, образующих стандартный аналоговый или импульсный регулятор с функциями оперативного управления (табл. 1).

Ввод любой СК аннулирует всю ранее введенную конфигурацию. Введенная СК с помощью стандартных процедур может быть дополнена или видоизменена. В ней можно устанавливать требуемые параметры настройки.

2. Алгоритмы лицевой панели

2.1. ОКО(О1) - Оперативный контроль регулирования

Назначение

Алгоритм применяется в том случае, если оперативное управление контуром регулирования должно вестись с помощью лицевой панели контроллера. Каждый контур (от 1 до 4) обслуживается своим алгоритмом ОКО. Алгоритм позволяет с помощью клавиш лицевой панели изменять режим управления, режим задания, управлять программным задатчиком, изменять выходной сигнал регулятора (в режиме ручного управления), изменять сигнал задания (в режиме ручного задатчика), а также контролировать сигналы задания и рассогласования, входной и выходной сигналы, параметры программы (при программном регулировании) и т.п.

Как правило » алгоритм ОКО применяется в сочетании с алгоритмами ЗДН, ЗДЛ, РУЧ, РАН, РИМ

Описание алгоритма

Алгоритм ОКО (рис. 2) помещается только в алгоблоки с номерами от 1 до 4. Номер алгоблока, в который помещен алгоритм ОКО, определяет номер контура, обслуживаемого данным алгоритмом ОКО. При одной контуре алгоритм ОКО помещается в первый алгоблок, при двух контурах - в первый и второй алгоблоки и т.д.

Алгоритм имеет модификатор $0 \leq m \leq 15$. Модификатор задает вид и специфические параметры регулятора, а именно:

- является регулятор обычным или каскадным;
- имеет регулятор аналоговый или импульсный выход;
- предусматривается ли переход на внешнее задание;
- предусматривается ли режим дистанционного управления.

Если режимы каскадного регулирования, внешнего задания и дистанционного управления не предусматриваются, переключения в соответствующие режимы блокируются.

Значение модификатора выбирается из табл. 4.

Алгоритм ОКО имеет 10 или 15 входов. Если задан обычный регулятор ($m \leq 7$), имеется только 10 входов, если задан каскадный регулятор ($m \geq 8$), имеется 15 входов. Выходов алгоритм не имеет,

Путем конфигурирования входов определяется, какие сигналы принимаются в качестве сигналов оперативного управления.

Вход $X_{з\text{дн}}$ определяется сигнал, который является сигналом текущего задания и выводится на цифровой индикатор "задание" (см. ТО, ТО2). Обычно этот вход подключается к основному выходу алгоритма ЗДН.

Вход $X_{\text{вх}}$ определяет сигнал, который является входным сигналом (регулируемым параметром) и выводится на цифровой индикатор избирательного контроля в позиции "вх". Обычно этот вход подключается к одному из выходов алгоритма ввода аналогового ВАА или ВАБ.

Вход $X_{\text{Е}}$ определяет сигнал, который является сигналом рассогласования и выводится на цифровой индикатор избирательного контроля в позиции "Е". Обычно этот вход подключается к выходу $Y_{\text{Е}}$ алгоритма регулирования РАН или РИМ.

Входы W_0 и W_{100} являются настроечными. Эти входы задают технические единицы, в которых индицируются сигнал задания, входной сигнал и сигнал рассогласования (для всех трех параметров задаются одни и те же технические единицы). Вход W_0 задает значение, соответствующее 0 % аналогового сигнала, а вход W_{100} значение, соответствующее 100 % сигнала. Текущее индицируемое значение $W_{\text{инд}}$ сигнала задания $X_{з\text{дн}}$, входного сигнала $X_{\text{вх}}$ и сигнала рассогласования $X_{\text{Е}}$ определяется из формулы:

$$W_{\text{инд}} = W_0 - ((W_{100} - W_0) / 100) * X \quad (2)$$

где X - соответственно сигнал $X_{з\text{дн}}$, $X_{\text{вх}}$ или $X_{\text{Е}}$, выраженный в процентах. Если эти сигналы предполагается индицировать в процентах, то устанавливается $W_0 = 0$ и $W_{100} = 100$.

Вход $X_{\text{руч}}$ подключается к основному выходу алгоритма ручного управления РУЧ. Такое соединение обеспечивает изменение с лицевой панели режимов управления, а также ручное изменение выхода.

На вход $X_{\text{вр}}$ (выход регулятора) подается сигнал, характеризующий управляющее воздействие. Для аналогового регулятора это может быть тот же выход алгоритма РУЧ, к которому подключается вход $X_{\text{руч}}$, либо (при наличии датчика положения вала исполнительного механизма) - сигнал на одном из аналоговых входов, к которому подключен датчик положения. Для импульсного регулятора на вход как правило подается сигнал от датчика положения.

Сигнал, поступивший на вход $X_{\text{вр}}$, при оперативном управлении выводится на шкальный индикатор, а также в режиме контроля "вых" - на цифровой индикатор избирательного контроля. Выходной сигнал контролируется в процентах.

На вход Z подается любой (по выбору) сигнал, который требуется индицировать в процессе оперативного управления. Этот сигнал контролируется по цифровому индикатору избирательного контроля в позиции « Z ». Это может быть аналоговый или дискретный сигнал, время (например, время какого-либо таймера) или число (например, оставшееся число повторений программы при программном регулировании). Тип сигнала задается числом на настроечном входе N_z в соответствии с табл.3.

Вход $N_{\text{ок}}$ (ошибка контура) используется в том случае, когда необходим контроль выхода одного или нескольких сигналов за допустимый диапазон. Если $N_{\text{ок}} > 0$, то на лицевой панели контроллера загорается один из ламповых индикаторов «1-4» в зоне "ошибки" (номер этого индикатора соответствует номеру контура, обслуживаемого данным алгоритмом ОКО).

Обычно вход $N_{\text{ок}}$ соединяется с выходом алгоритма порогового контроля ПОК. В этом случае по цифровому индикатору избирательного контроля в позиции "ок " (ошибка контура) можно определить номер сигнала (если их несколько), вышедшего за допустимые границы (см. также описание алгоритма ПОК). Если $N_{\text{ок}} = 0$, то ламповый индикатор в зоне "ошибка" не горит.

Все перечисленные выше входы (от 01 до 10) задают параметры оперативного управления как обычного регулятора, так и каскадного регулятора, если он работает в режиме каскадного управления КУ. В последнем случае входы 01-05 определяют параметры ведущего регулятора в каскадной схеме, а входы 06-10 - параметры регулятора в целом. При каскадном регулировании обычно требуется оперативно управлять также и ведомым регулятором. Такая необходимость возникает, когда каскадный регулятор переводится в режим локального управления ЛУ. Возможности управления регулятором в локальном режиме определяются сигналами, подаваемыми на входы 11-15 алгоритма ОКО.

Вход $X_{\text{здл}}$ обычно соединяется с основным выходом алгоритма локального задатчика ЗДЛ. В этом случае сигнал локального задания в режиме ЛУ контролируется по цифровому индикатору "задание".

Входы $X_{\text{вх,А}}$, $X_{\text{Е,А}}$ зададут соответственно сигналы, контролируемые по цифровому индикатору избирательного контроля в позиции "вх" и "Е" при ЛУ.

Назначение входов $W_{0,А}$ и $W_{100,А}$, такое же, как входов W_0 и W_{100} .

Таким образом, с помощью алгоритма ОКО программируются (назначаются) функции и сигналы оперативного управления контуром регулирования. Алгоритм определяет, какие сигналы будут выведены на индикаторы лицевой панели и в каких технических единицах (два задания, входа и рассогласования) эти сигналы будут индицироваться.

Таблица 3

Nz	Тип сигнала
0	Дискретный
1	Время, младший масштаб
2	Время, старший масштаб
3	Скорость, младший масштаб
4	Скорость, старший масштаб
5	Время импульса
6	Масштабный коэффициент
7	Коэффициент пропорциональности
8	Аналоговый
9	Числовой

Значения $N_z < 0$ и $N_z > 9$ соответствуют $N_z = 0$.

Таблица 4

Вид регулятора	Режимы		Модификатор
	Внешнего задания («ВНШ»)	Дистанционного Управления («ДСТ»)	
Обычный аналоговый	-	-	00
	-	+	01
	+	-	02
	+	+	03
Обычный импульсный	-	-	04
	-	+	05
	+	-	06
	+	+	07
Каскадный аналоговый	-	-	08
	-	+	09
	+	-	10
	+	+	11
Каскадный импульсный	-	-	12
	-	+	13
	+	-	14
	+	+	15

Входы алгоритма ОКО

Таблица 5

Входы			Назначение
№	Обозн.	Вид	
01	$X_{зди}$	Входы	Сигнал задания
02	$X_{вх}$		Входной сигнал (регулируемый параметр)
03	W_0		0 % в технических единицах
04	W_{100}		100 % в технических единицах
05	X_{ϵ}		Сигнал рассогласования
06	$X_{руч}$		Сигнал ручного управления
07	$X_{вд}$		Выходной сигнал регулятора
08	Z		Любой выбранный сигнал
09	N_z		Тип сигнала на входе z
10	$N_{ок}$		Ошибка контура
11	$X_{зди}$		Сигнал задания ведомого регулятора в локальном режиме
12	$X_{вх,А}$		Входной сигнал (регулируемый параметр) ведомого регулятора в локальном режиме
13	$W_{0,А}$		0% в технических единицах для ведомого регулятора в локальном режиме
14	$W_{100,А}$		100% в технических единицах для ведомого регулятора в локальном режиме
15	$X_{\epsilon,А}$		Сигнал рассогласования ведомого регулятора в локальном режиме

**Функциональная схема алгоритма
"Оперативный контроль регулирования ОКО"**

Код 01		m = 00 - 15	МВ - отсутствует
Хздн.	01	Управление заданием и индикация задания, входа, рассогласования стандартного регулятора или ведущего регулятора в каскадном режиме	
Хвх	02		
W0	03		
W100	04		
Хε	05		
Хруч	06	Ручное управление и индикация выхода, произвольного параметра и ошибки	
Хвр	07		
z	08		
Nz	09		
Нок	10		
Хздл	11	Управление заданием и индикация задания, входа, рассогласования стандартного ведомого регулятора в локальном режиме (только при m ≥ 8)	
Хвх,л	12		
W0,л	13		
W100,л	14		
Хв,л	15		

Рис. 2

2.2. ОКЛ(02) - Оперативный контроль логической программы.
Назначение.

Алгоритм ОКЛ применяется в составе модели Р-130, ориентированной на решение задач логического шагового управления. Алгоритм применяется в сочетании с алгоритмом этапа ЭТП, координирует работу алгоритмов ЭТП и позволяет выводить на индикаторы лицевой панели оперативную информацию о ходе выполнения логической программы.

В одном контроллере можно реализовать 4 независимых программы. Для контроля каждой программы используется отдельный алгоритм ОКЛ.

Описание алгоритма

Номер программы, с которой связан алгоритм ОКЛ, равен номеру алгоблока, в котором помещен данный алгоритм ОКЛ. Так как в контроллере можно реализовать до 4 программ, алгоритм ОКЛ помещается только в первые четыре алгоблока.

Все операции, выполняемые алгоритмом ОКЛ, относятся к программе, с которой связан данный алгоритм,

Алгоритм имеет 3 секции (рис.3). Первая секция управляет состоянием программы, вторая организует контроль сигналов по цифровому индикатору, третья координирует работу алгоритмом этапа ЭТП.

Управление состоянием программы может выполняться как с помощью клавиш лицевой панели контроллера, так и с помощью дискретных команд пуска, останова и сброса, поступающих на входы соответственно $C_{п}$, $C_{ст}$, $C_{сбр}$. Эти команды действуют по переднему фронту, причем если управление ведется и от клавиш лицевой панели, и от входов алгоритма, то выполняется последняя поступившая команда.

Если передний фронт команд пуска, останова и сброса поступил одновременно, то действует система приоритетов: команда "стоп" приоритетна над командой "пуск", команда "сброс" приоритетна над другими командами. Если передний фронт одной из указанных команд поступил одновременно с командой от лицевой панели, то выполняется команда, поступившая на дискретный вход алгоритма.

На выходах первой секции формируются дискретные сигналы, характеризующие текущее состояние программы: «пуск» ($D_{п}$), «стоп» ($D_{ст}$), «сброс» ($D_{сбр}$), «ожидание» ($D_{ож}$), «конец программы» ($D_{кп}$).

С помощью лицевой панели в рамках каждой программы можно контролировать до 4 различных сигналов. Контролируемые сигналы подаются на входы Z_i алгоритма ОКЛ. Формат, в котором эти сигналы контролируются, задается индивидуально для каждого из сигналов с помощью параметра $N_{z,i}$; в соответствии с табл. 6. Алгоритм ОКЛ для каждой из программ позволяет определить номер сигнала, который не соответствует допустимому значению. Для этого используется либо алгоритм логического контроля ЛОК (если контролируются дискретные сигналы), либо алгоритм порогового контроля ПОК (если контролируются аналоговые сигналы). В любом случае выход N этих алгоритмов подключается ко входу $N_{оп}$ алгоритма ОКЛ. При таком включении с помощью лицевой панели в режиме "оп" (ошибка программы) можно определить номер сигнала, принявшего недопустимое значение.

К одному алгоритму ОКЛ можно подключить несколько (до 87) алгоритмов этапа ЭТП. Число подключаемых этапов $0 \leq m \leq 87$ задается модификатором.

Этапы подключаются к входам $N_{э,i}$ алгоритма ОКЛ. Номер i в обозначении этих входов присваивается этапу, который с этим входом связан. Так этап, подключенный к входу $N_{э,i}$, получает номер 5. Алгоритм ОКЛ выполняет свои функции, только если к нему подключен хотя бы один алгоритм этапа ЭТП.

Выходы $N_{эт}$, $N_{ш}$ и $T_{ш}$ характеризуют ход выполнения программы. $N_{эт}$ - номер текущего (выполняемого) этапа, $N_{ш}$ - номер выполняемого шага, $T_{ш}$ - время, оставшееся до истечения контрольного времени для выполнения шага.

Алгоритм имеет специальный дискретный выход $D_{кл}$, который свидетельствует о нажатии клавиши "ft" на лицевой панели. Если эта клавиша не нажата, то $D_{кл}=0$ если нажата, то $D_{кл}=1$. Сигнал $D_{кл}$ может использоваться для ручного включения, выключения, сброса и т.п. операций внутри контроллера (однако, этот сигнал нельзя подавать на входы «пуск», «стоп», «сброс» данного алгоритма ЭТП).

Таблица 6

N_z	Тип сигнала
0	Дискретный
1	Время, младший масштаб
2	Время, старший масштаб
3	Скорость, младший масштаб
4	Скорость, старший масштаб
5	Минимальная длительность импульса
6	Масштабный коэффициент
7	Коэффициент пропорциональности
8	Аналоговый
9	Числовой

Значения $N_z < 0$ и $N_z > 9$ соответствует $N_z = 0$.

Входы-выходы алгоритма ОКЛ

Таблица 7

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{\text{п}}$	Входы	Команда пуска
02	$C_{\text{ст}}$		Команда стопа
03	$C_{\text{сбр}}$		Команда сброса
04	$N_{\text{оп}}$		Номер ошибки программы
05	Z_1		1-й контролируемый сигнал
06	$N_{z,1}$		Формат 1-го сигнала
07	Z_2		2-й контролируемый сигнал
08	$N_{z,2}$		Формат 2-го сигнала
09	Z_3		3-й контролируемый сигнал
10	$N_{z,3}$		Формат 3-го сигнала
11	Z_4		4-й контролируемый сигнал
12	$N_{z,4}$		Формат 4-го сигнала
13	$N_{3,1}$		Подключение 1-го этапа
14	$N_{3,2}$		Подключение 2-го этапа
.	.		.
.	.		.
.	.		.
m+12	$N_{3,m}$		Подключение этапа m
01	$D_{\text{п}}$	Выходы	Состояние «пуск»
02	$D_{\text{ст}}$		Состояние «стоп»
03	$D_{\text{сбр}}$		Состояние «сброс»
04	$D_{\text{ож}}$		Состояние «ожидание»
05	$D_{\text{кп}}$		Состояние "конец программы"
06	$N_{\text{эт}}$		Номер выполняемого этапа
07	$N_{\text{эш}}$		Номер выполняемого шага
08	$T_{\text{ш}}$		Время, оставшееся до окончания контрольного времени выполняемого шага
09	$D_{\text{кл}}$		Признак нажатия на лицевой панели клавиши "↵"

Функциональная схема алгоритма
"Оперативный контроль логической
программы ОКЛ"

	Код 02	m = 00 - 81	МВ - отсутствует
Сп	01	Управление состоянием программы	01 Дп
Сст.	02		02 Дст
Ссбр.	03		03 Дсбр
			04 Дож
			05 Дкп
Ноп	04	Контроль сигналов	
Z1	05		
Nz,1	06		
Z2	07		
Nz,2	08		
Z3	09		
Nz,3	10		
Z4	11		
Nz,4	12		
НЭ,1	13	Управление этапами	06 Нэт
НЭ,2	14		07 Нш
.	.		08 Тш
НЭ,m	m+12		09 Дкл

Рис. 3

2.3. ОНД (03) - Оперативный непрерывно-дискретный контроль

Назначение

Алгоритм применяется для организации оперативного управления в задачах, в которых используется как непрерывная, так и дискретная обработка сигналов.

Алгоритм ОКД применяется в сочетании с алгоритмами группового управления и контроля – ГРА, ГВД, ГДВ, ГРУ и ГРК, а также с алгоритмами порогового (ПОК) и логического (ЛОК) контроля.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Структура алгоритма.

Функциональная схема алгоритма содержит три области (рис. 3.1)

- 1) область управления шаговой программой;
- 2) область информационного управления и контроля;
- 3) область контроля ошибок.

Область управления шаговой программой выполняет функцию оперативного логического управления.

Область информационного управления позволяет контролировать аналоговые и дискретные сигналы и изменять эти сигналы вручную.

Область контроля ошибок позволяет установить, выходят ли контролируемые сигналы за допустимые границы (для аналоговых сигналов), или принимают недопустимое состояние (для дискретных сигналов).

Управление шаговой программой.

Функцию управления шаговой программой алгоритм ОКД выполняет вместе с алгоритмом шаговой программы ШАП. Для организации управления выход $N_{ш}$ алгоритма ШАП соединяется со входом N алгоритма ОКД. После такого соединения на лицевую панель выводится оперативная информация о работе шаговой программы и с помощью клавиш лицевой панели можно управлять ходом выполнения шаговой программы.

Управлять шаговой программой можно также с помощью сигналов, поступающих на входы C_d , $C_{ст}$ и $C_{сбр}$, при этом выполняется соответственно пуск, останов и сброс шаговой программы. Эти команды действуют по переднему фронту, причем если управление ведется и от клавиш лицевой панели, и от входов алгоритма, то выполняется последняя поступившая команда. Если передний фронт команд C_d , $C_{ст}$, $C_{сбр}$ поступил одновременно, то действует система приоритетов: команда "стоп" приоритетна над командой "пуск", команда "сброс" приоритетна над другими командами. Если передний фронт одной из указанных команд поступил одновременно с командой от лицевой панели, то выполняется команда, поступившая на дискретный вход алгоритма.

При управлении шаговой программой по цифровому индикатору лицевой панели можно одновременно контролировать один из сигналов, назначение которого программируется. Контролируемый сигнал подается на вход Z алгоритма ОКД. Формат, в котором этот сигнал выводится на цифровой индикатор, определяется параметром настройки, установленном на входе N_z (см. табл. 7.1). Если выбран аналоговый формат, то дополнительно с помощью входов W_0 и W_{100} можно выбрать технические единицы, в которых контролируется сигнал. Параметры W_0 и W_{100} определяют значения в технических единицах, соответствующие 0% и 100% аналогового сигнала, при этом текущее индицируемое значение $W_{инд}$ определяется из формулы:

$$W_{инд} = W_0 + ((W_{100} - W_0) / 100) * Z,$$

где Z - контролируемый аналоговый сигнал, выраженный в процентах.

На выходе алгоритма ОКД формируются дискретные сигналы, отражающие текущее состояние шаговой программы. В состоянии "пуск", «стоп» «сброс», "ожидание", "конец программы" на выходе соответственно $D_{п}$, $D_{ст}$, $D_{сбр}$, $D_{ож}$, $D_{кп}$ формируется сигнал, равный логической 1. Например, в состоянии "сброс" $D_{п} = D_{ст} = D_{сбр} = D_{ож} = D_{кп} = 0$.

Информационное управление и контроль.

Алгоритм ОКД имеет пять информационных входов $Z_1 - Z_5$. Сигналы, поступающие на эти входы, выводятся на цифровой индикатор лицевой панели контроллера (с помощью избирательного контроля все пять сигналов выводятся на один цифровой индикатор). Возможности входов $Z_1 - Z_5$ могут быть расширены с помощью алгоритмов группового контроля и управления ГРА, ГВД, ГДВ, ГРУ или ГРК. Каждый из этих алгоритмов контролирует до 9 сигналов, любой из которых по правилам избирательного управления может быть передан на выход N этих алгоритмов. Если указанный выход N алгоритмов ГРА, ГВД, ГДВ, ГРУ или ГРК соединить с одним из входов $Z_1 - Z_5$ алгоритма ОКД, то каждый из входов $Z_1 - Z_5$ может контролировать до 9 сигналов. Таким образом, в пределах с помощью алгоритма ОКД можно организовать контроль по цифровому индикатору лицевой панели до 45 сигналов.

Формат сигналов, контролируемых по цифровому индикатору, задается в алгоритмах, подключенных к входам $Z_1 - Z_5$ алгоритма ОКД.

Алгоритм ОКД имеет дискретный выход $D_{кл}$ состояние которого отражает состояние клавиши "↵" на лицевой панели контроллера:

пока клавиша "↵" нажата, сигнал $D_{кл} = 1$, если клавиша отпущена $D_{кл} = 0$.

Контроль ошибок.

На лицевой панели контроллера в области ошибок расположены ламповые индикаторы "1-4", состояние которых соответствует состоянию входов $N_1 - N_4$ алгоритма ОКД. Если на вход N_i поступает сигнал $N_i > 0$, инди-

катор с соответствующим номером загорается. При $N_i \leq 0$ индикатор не горит.

Таким образом, с помощью алгоритма ОКД можно контролировать 4 независимых сигнала, появление которых интерпретируется как ошибка управления.

Возможности контроля ошибок можно расширить, если ко входам $N_1 - N_4$ (одному или нескольким) подключить алгоритм порогового контроля ПОК или логического контроля ЛОК. В этом случае все ошибки разбиваются на 4 группы (входы $N_1 = N_4$) в каждой из которых может контролироваться до 32 сигналов. Если хотя бы один сигнал в группе вышел за допустимые границы (алгоритм ПОК) или принял недопустимое состояние (алгоритм ЛОК), загорается соответствующий ламповый индикатор "1 - 4". а на цифровой индикатор в режиме "ОУ" (ошибка управления), выводится номер сигнала в группе, который явился причиной ошибки.

Форматы сигналов

Таблица 7.1

№	Тип сигнала
0	Дискретный
1	Время, младший масштаб
2	Время, старший масштаб
3	Скорость, младший масштаб
4	Скорость, старший масштаб
5	Время импульса
6	Масштабный коэффициент
7	Коэффициент пропорциональности
8	Аналоговый
9	Числовой

Значения $N_z < 0$ и $N_z > 9$ соответствуют $N_z = 0$.

Входы - выходы алгоритма ОКД

Таблица 7.2

Входы-выхода			Назначение
№	Обозн.	Вид	
01	C_n	Входы	Команда пуска
02	$C_{ст}$		Команда останова
03	$C_{сбр}$		Команда сброса
04	$N_{шаг}$		Вход, соединяемый с выходом $N_{шаг}$ алгоритма ЩАП
05	Z		Сигнал, контролируемый в режиме "прог"
06	N_z		Формат сигнала Z
07	W_0		Технические единицы, соответствующие 0% сигнала Z , если Z – аналоговый сигнал
08	W_{100}		То же, что W_0 , но для сигнала, равного 100%
09-13	$Z_1 - Z_5$		Сигналы, контролируемые в режимах соответственно $Z_1 - Z_5$
14-17	$N_1 - N_4$		Сигналы ошибок, контролируемые по индикаторам соответственно 1-4 лицевой панели
01	D_n	Выходы	Состояние программы "пуск"
02	$D_{ст}$		Состояние программы "стоп"
03	$D_{сбр}$		Состояние программы "сброс"
04	$D_{ож}$		Состояние программы "ожидание"
05	$D_{кп}$		Состояние программы "конец программы"
06	$D_{кл}$		Клавиша "↵" на лицевой панели нажата

Функциональная схема алгоритма
"Оперативный непрерывно-дискретный
контроль ОКД"

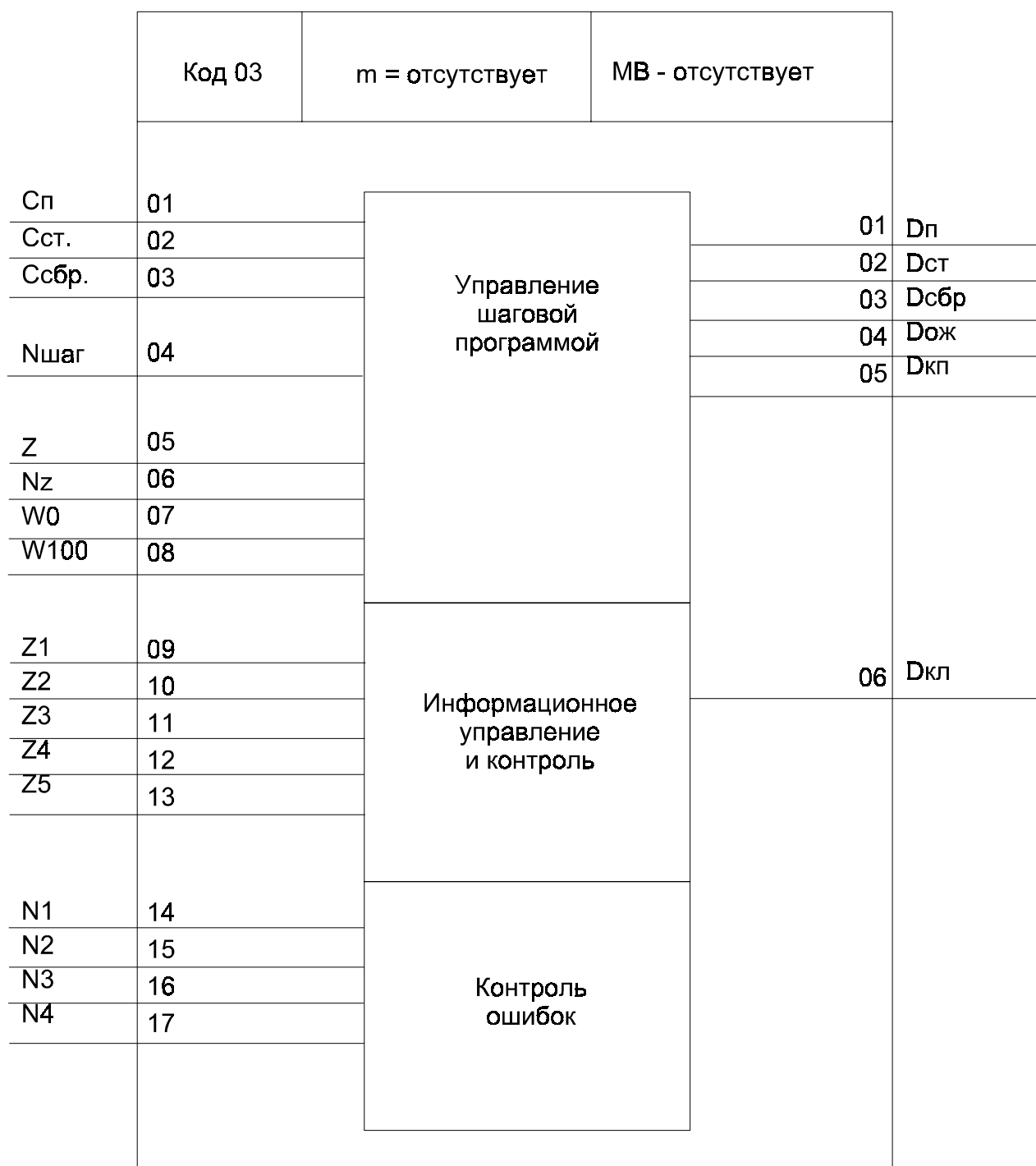


Рис. 3.1

2.4. ДИК(04) - Дискретный контроль

Назначение

Алгоритм применяется в сочетании с алгоритмом ОКЛ используется для выдачи дискретной информации на лицевую панель контроллера (только в варианте логического управления).

Описание алгоритма

Алгоритм имеет $0 \leq m \leq 32$ входов, где m задается модификатором. Выходов алгоритм не имеет (рис. 4).

Если на i - й вход алгоритма поступает дискретный сигнал $C_i=1$, в зоне дискретного контроля лицевой панели контроллера, рассчитанного на логическое управление, зажигается i -й ламповый индикатор.

Всего в составе одного контроллера может быть задействован только один алгоритм ДИК.

Входы алгоритма ДИК

Таблица 8

Входы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	1-й ламповый индикатор
02	C_2		2-й ламповый индикатор
.	.		.
.	.		.
.	.		.
m	C_m		m -й ламповый индикатор

Функциональная схема алгоритма
"Дискретный контроль ДИК"

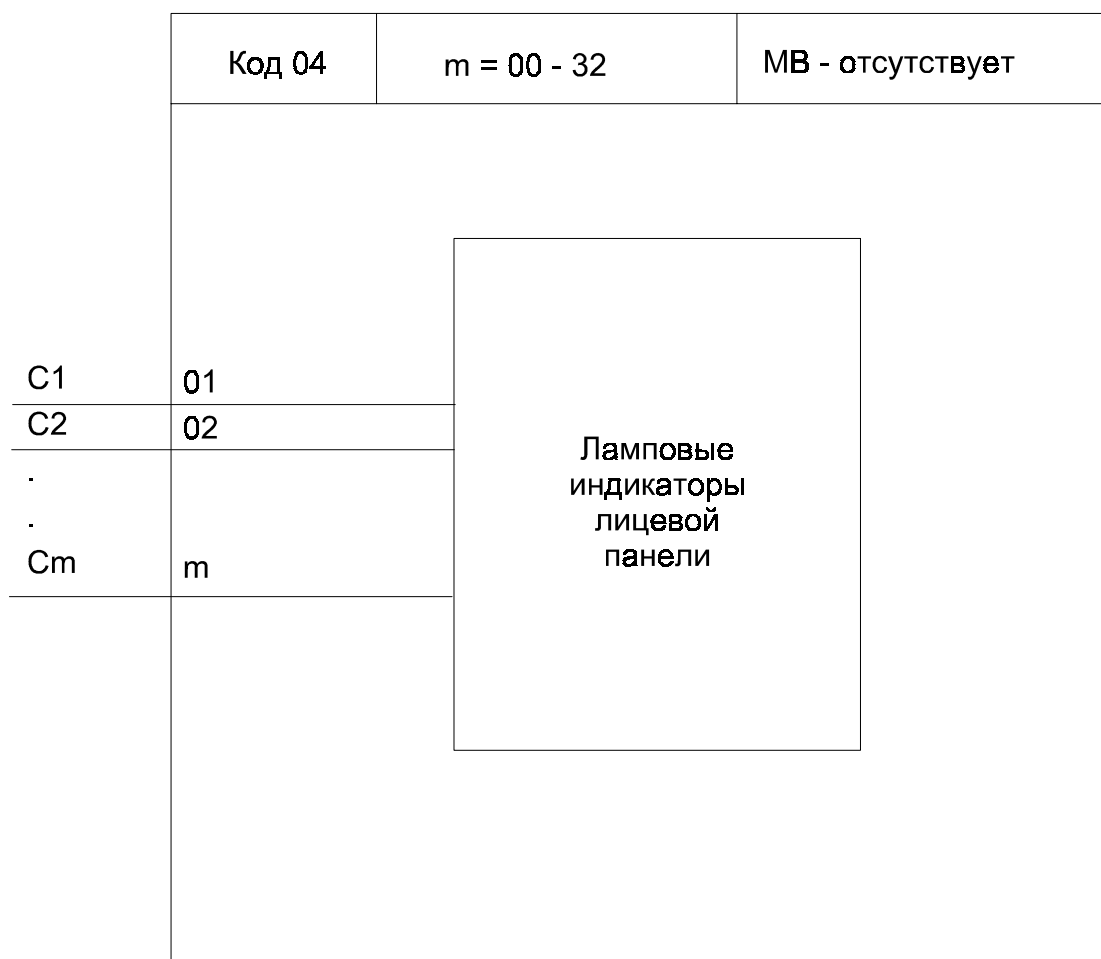


Рис. 4

3. Алгоритм ввода-вывода

3.1. ВИН(05) - Ввод интерфейсный

Назначение.

Алгоритм применяется в тех случаях, когда Ремиконт Р-130 должен принимать сигналы, поступающие на вход его интерфейсного канала. Один алгоритм ВИН организует связь с одним абонентом-источником, от которого можно принимать несколько (до 21) различных сигналов. Если необходима связь с несколькими абонентами-источниками, в контроллере используется несколько алгоритмов ВИН. Алгоритм ВИН применяется при включении Р-130 в кольцевую локальную сеть «Транзит».

Для приема, через интерфейсный канал команд оперативного управления применения алгоритма ВИН не требуется.

Описание алгоритма.

Один алгоритм ВИН применяется для приема сигналов, поступающих от одного абонента-источника. Номер абонента-источника задается на входе $N_{ист}$ ври настройке алгоритма (рис. 5).

Каждый абонент-источник может передавать несколько сигналов. Алгоритм ВИН выделяет из общего пакета этих сигналов требуемые (до номеру) сигналы и размещает их на своих выходах $Y_1 \dots Y_m$. От одного абонента-источника могут приниматься m сигналов, при этом $0 \leq m \leq 21$ и m задается модификатором. Выделенные на выходах Y_i сигналы затем могут обрабатываться другими алгоритмами, связанными по конфигурации с алгоритмом ВИН.

Номер сигнала абонента-источника, выделяемого на выходе Y_i , задается при настройке на входе N_i алгоритма. Например, если на выходах Y_1 , Y_2 и Y_3 должны быть выделены сигналы третьего абонента-источника, имеющие номера соответственно 5, 12 и 2, то устанавливаются параметры $N_{ист}=3$, $N_1=5$, $N_2=12$, $N_3=2$.

Для того чтобы связаться с другим абонентом-источником, задействуется другой алгоритм ВИН, у которого при настройке устанавливается требуемое значение $N_{ист}$.

Входы-выходы алгоритма ВИН

Таблица 9

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$N_{ист}$	Входы	Номер источника
02	N_1		Номер сигнала источника, поступающего на выход Y_1 алгоритма
03	N_2		Номер сигнала источника, поступающего на выход Y_2 алгоритма
.	.		.
.	.		.
.	.		.
$m+1$	N_m		Номер сигнала источника, поступающего на выход Y_m алгоритма
01	Y_1	Выходы	Сигнал источника с номером N_1
02	Y_2		Сигнал источника с номером N_2
.	.		.
.	.		.
.	.		.
02	Y_m		Сигнал источника с номером N_m

Функциональная схема алгоритма
"Ввод интерфейсный ВИН"

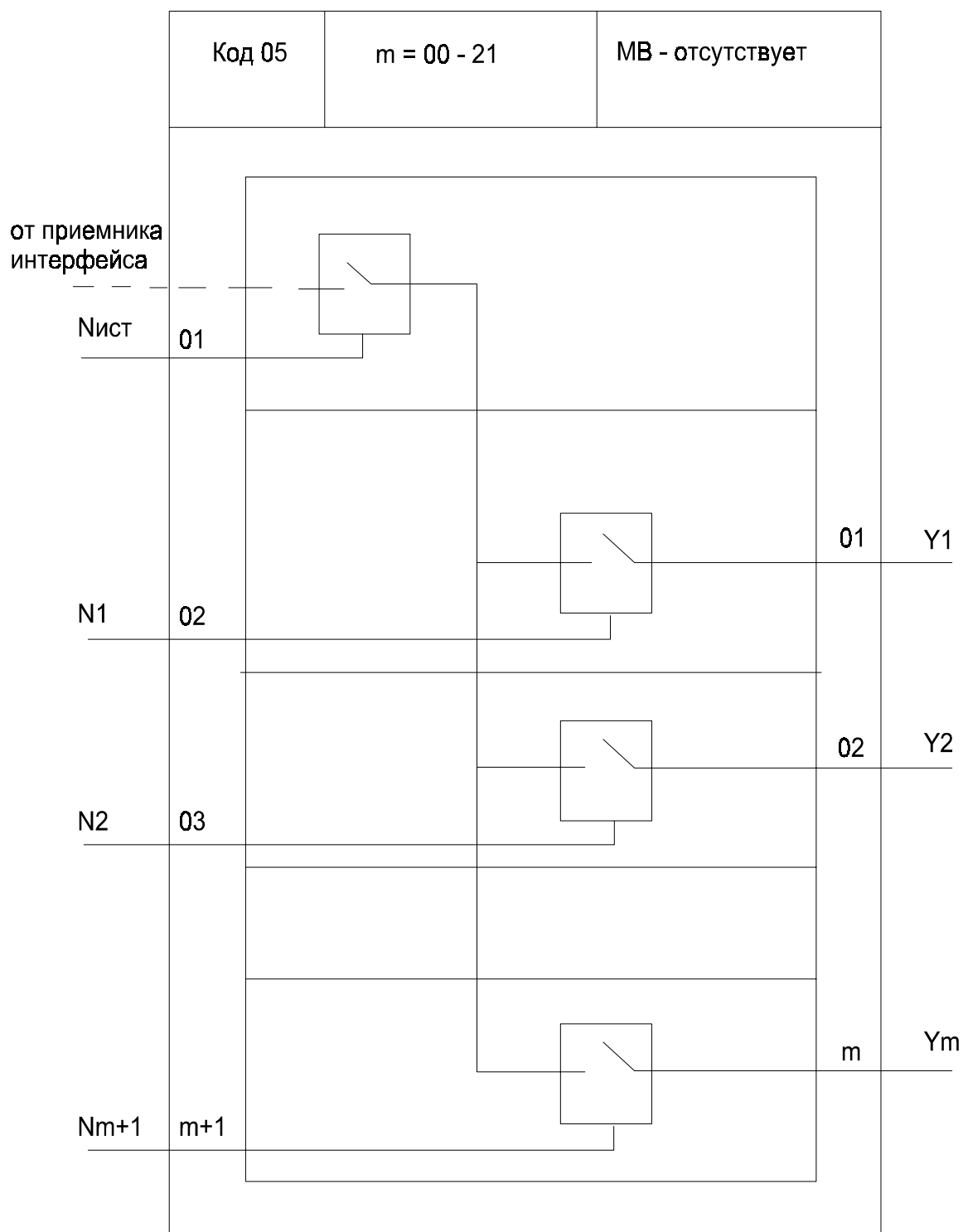


Рис. 5

3.2. ИНВ(06) - Интерфейсный вывод

Назначение

Алгоритм применяется в тех случаях, когда Ремиконт Р-130 должен передавать какие-либо сигналы через канал интерфейса. Алгоритм может передавать несколько (до 30) аналоговых или неупакованных дискретных сигналов. Максимальное число упакованных дискретных сигналов, передаваемых через интерфейс, равно 390.

Описание алгоритма

Сигналы, которые требуется передать через интерфейс, должны быть предварительно путем конфигурирования сформированы на входах X_i алгоритма. Число передаваемых сигналов $0 \leq m \leq 30$ и задается модификатором (рис 6).

В информационном пакете, передаваемом через интерфейс, сигналам приписываются номера, равные номерам соответствующих входов X_i . Так сигнал на входе X_1 получает номер 1, сигнал на входе X_2 - номер 2 и т.д. Весь пакет получает номер источника, и равный системному номеру контроллера, устанавливаемому в процессе его программирования.

На входы X_i могут поступать как аналоговые, так и дискретные сигналы в любом сочетании. С точки зрения числа передаваемых сигналов обычный (не упакованный) дискретный сигнал эквивалентен аналоговому сигналу. Для того, чтобы увеличить число передаваемых дискретных сигналов, они могут быть предварительно упакованы с помощью алгоритма шифрации ШИФ. В этом случае в рамках одного передаваемого сигнала можно передать до 13 дискретных сигналов. Таким образом, пользуясь упаковкой можно передать до $13 \cdot 30 = 390$ дискретных сигналов. На приемной стороне упакованные дискретные сигналы должны быть распакованы с помощью алгоритма дешифрации ДЕШ. В одном контроллере задействуете» лишь один алгоритм ИНВ.

Входы-выходы алгоритма ИНВ

Таблица 10

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Сигнал №1
02	X_2		Сигнал №2
.	.		.
.	.		.
.	.		.
M	X_m		Сигнал №m

Функциональная схема алгоритма
"Интерфейсный вывод ИНВ"

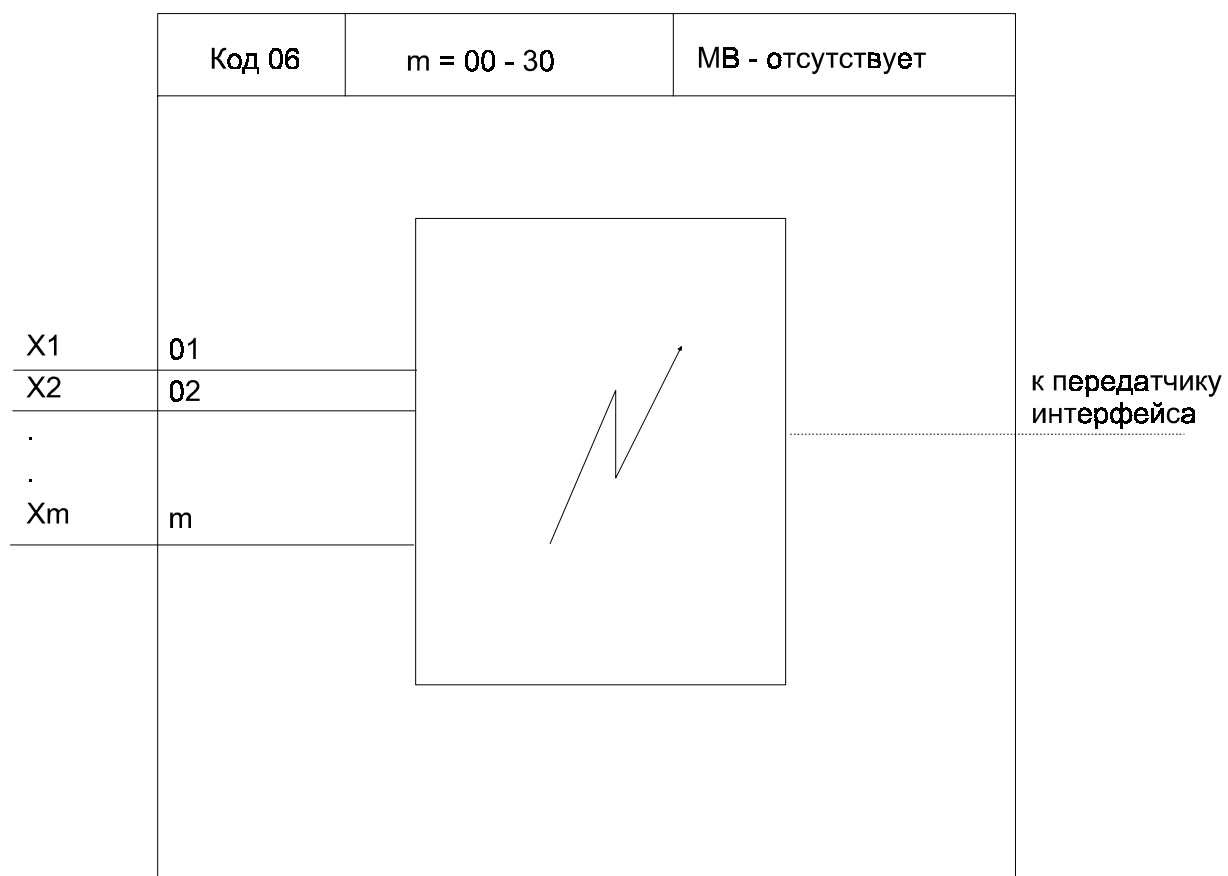


Рис. 6

3.3. ВАА(07), ВАБ(08) - Ввод аналоговый групп А и Б

Назначение

Алгоритм применяется для связи функциональных алгоритмов с аппаратными средствами аналогового входа (с АЦП). Для связи с аналоговыми входами группы А и В используются соответственно алгоритмы ВАА и ВАБ. Каждый алгоритм обслуживает до 8 аналоговых входов.

Помимо связи с АЦП алгоритмы ВАА и ВАБ позволяют корректировать диапазон входного аналогового сигнала в двух точках, соответствующих 0 и 100 % диапазона.

Описание алгоритма

Алгоритм содержит несколько идентичных независимых каналов. Число каналов $0 \leq m \leq 8$ и задается модификатором (рис. 7). Каждый канал связан с соответствующим (по номеру) аналоговым входом контроллера. Эта связь образуется "автоматически", как только алгоритм ВАА и ВАБ вводится в один из алгоблоков контроллера. К входному аналоговому сигналу добавляется сигнал смещения $X_{см}$, полученная сумма умножается на коэффициент K_m . Эти операции позволяют компенсировать смещение нуля и диапазона как АЦП, так и датчика, подключенного к контроллеру.

Выходной сигнал канала равен;

$$Y_i = (X_{ан,вх,i} + X_{см,i}) * K_{m,i}, \quad (3)$$

где $X_{ан,вх,i}$ – аналоговый входной сигнал, поступающий от АЦП на i -й канал.

Если коррекции не требуется, устанавливается $X_{см,i}=0$; $K_{m,i}=1$. В этом случае $Y_i = X_{ан,вх,i}$.

Время обслуживания одного канала аналогового ввода равно 40 ms. Время обновления аналоговых сигналов на выходе алгоритма равно $0,04 * m$ (ms), однако это время при любом m не может быть меньше времени цикла, с которым работает контроллер.

Входы-выходы алгоритма ВАА (ВАБ)

Таблица 11

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$X_{см,1}$	Входы	Смещение канала 1
02	$K_{m,1}$		Диапазон канала 1
03	$X_{см,2}$		Смещение какала 2
04	$K_{m,m}$		Диапазон канала 2
.	.		.
.	.		.
.	.		.
2m-1	$X_{см,m}$		Смещение канала m
2m	$K_{m,m}$		Диапазон канала m
01	Y_1	Выходы	Выход канала 1
02	Y_2		Выход канала 2
.	.		.
.	.		.
.	.		.
m	Y_m		Выход канала m

Функциональная схема алгоритма
"Ввод аналоговый ВАА (ВАО)"

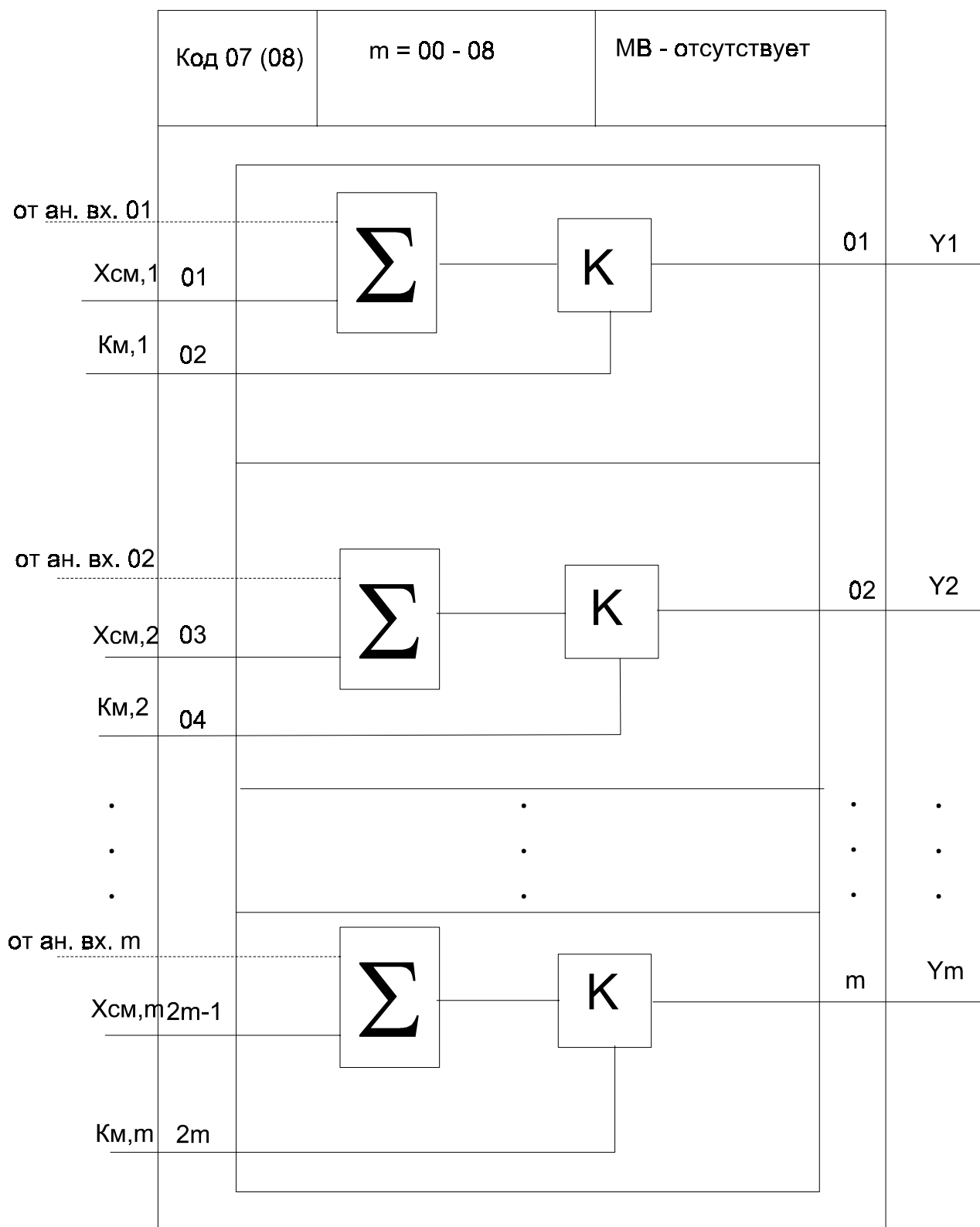


Рис. 7

3.4. ВДА(09) ВДБ(10) - Ввод дискретный групп А и Б

Назначение

Алгоритм применяется для связи функциональных алгоритмов с аппаратными средствами дискретного ввода (с ДЦП). Для связи с входами группы А и Б используются соответственно алгоритмы ВДА и ВДБ. Каждый алгоритм обслуживает до 16 дискретных входов.

Описание алгоритма

Алгоритм содержит несколько идентичных независимых каналов, число которых $0 \leq m \leq 16$ и задается модификатором (рис. 8).

Если напряжение на i -м дискретном входе контроллера равно 0-7 В, сигнал на i -м выходе алгоритма ВДА(ВДБ) равен $D_i=0$. Если на i -й дискретный вход контроллера подано напряжение 24 В, сигнал на i -м выходе алгоритма ВДА (ВДБ) принимает значение $D_i=1$.

Выходы алгоритма ВДА (ВДБ)

Таблица 11.1

Выходы			Назначение
№	Обозн.	Вид	
01	D_1	Выхода	Выход канала 1
02	D_2		Выход канала 2
.	.		.
.	.		.
.	.		.
m	D_m		Выход канала m

Функциональная схема алгоритма
"Ввод дискретный ВДА (ВДБ)"

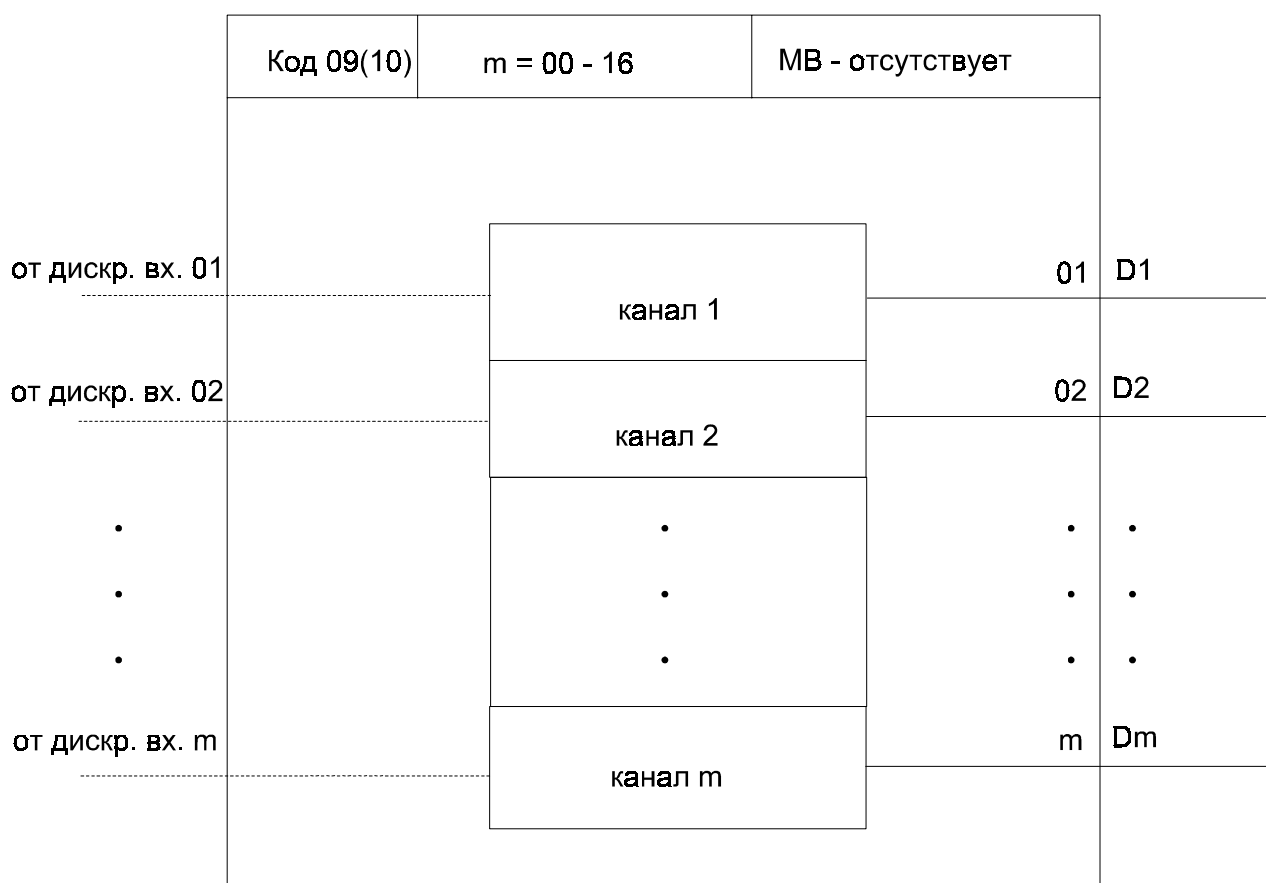


Рис. 8

3.5. АВА(11) АВБ(12) – Аналоговый вывод групп А и Б

Назначение

Алгоритм применяется для связи функциональных алгоритмов с аппаратными средствами аналоговых выходов (с ЦАП). Для связи с аналоговыми выходами группы А и Б используются соответственно алгоритмы АВА и АВБ. Каждый алгоритм обслуживает до 2 аналоговых выходов.

Помимо связи с ЦАП алгоритмы АВА и АВБ позволяют корректировать диапазон выходного аналогового сигнала в двух точках, соответствующих 0 и 100% диапазона.

Описание алгоритма

Алгоритм содержит до 2 идентичных независимых каналов. Число этих каналов $0 \leq m \leq 2$ задается модификатором (рис. 9).

Каждый канал связан с соответствующим (по номеру) аналоговым выходом контроллера. Эта связь образуется «автоматически», как только алгоритм АВА (АВБ) вводится в один из алгоблоков контроллера.

Прежде чем поступить на аналоговый выход, входной сигнал умножается на коэффициент и к полученному произведению добавляется сигнал смещения. Эти операции позволяют компенсировать смещение диапазона и нуля ЦАП.

Выходной аналоговый сигнал (на выходе ЦАП) равен:

$$Y_{\text{ан.вых},i} = X_i * K_{m,i} + X_{\text{см},i}, \quad (4)$$

Если коррекции не требуется, устанавливается масштабный коэффициент $K_{m,i}=1$ и сигнал смещения $X_{\text{см},i}=0$.

Входы алгоритма АВА(АВБ)

Таблица 12

Входы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Вход канала 1
02	$X_{\text{см},1}$		Смещение канала 1
03	$K_{m,1}$		Диапазон канала 1
04	X_2		Вход канала 2
05	$X_{\text{см},2}$		Смещение канала 2
06	$K_{m,2}$		Диапазон канала 2

Функциональная схема алгоритма
"Аналоговый вывод АВА (АВБ)"

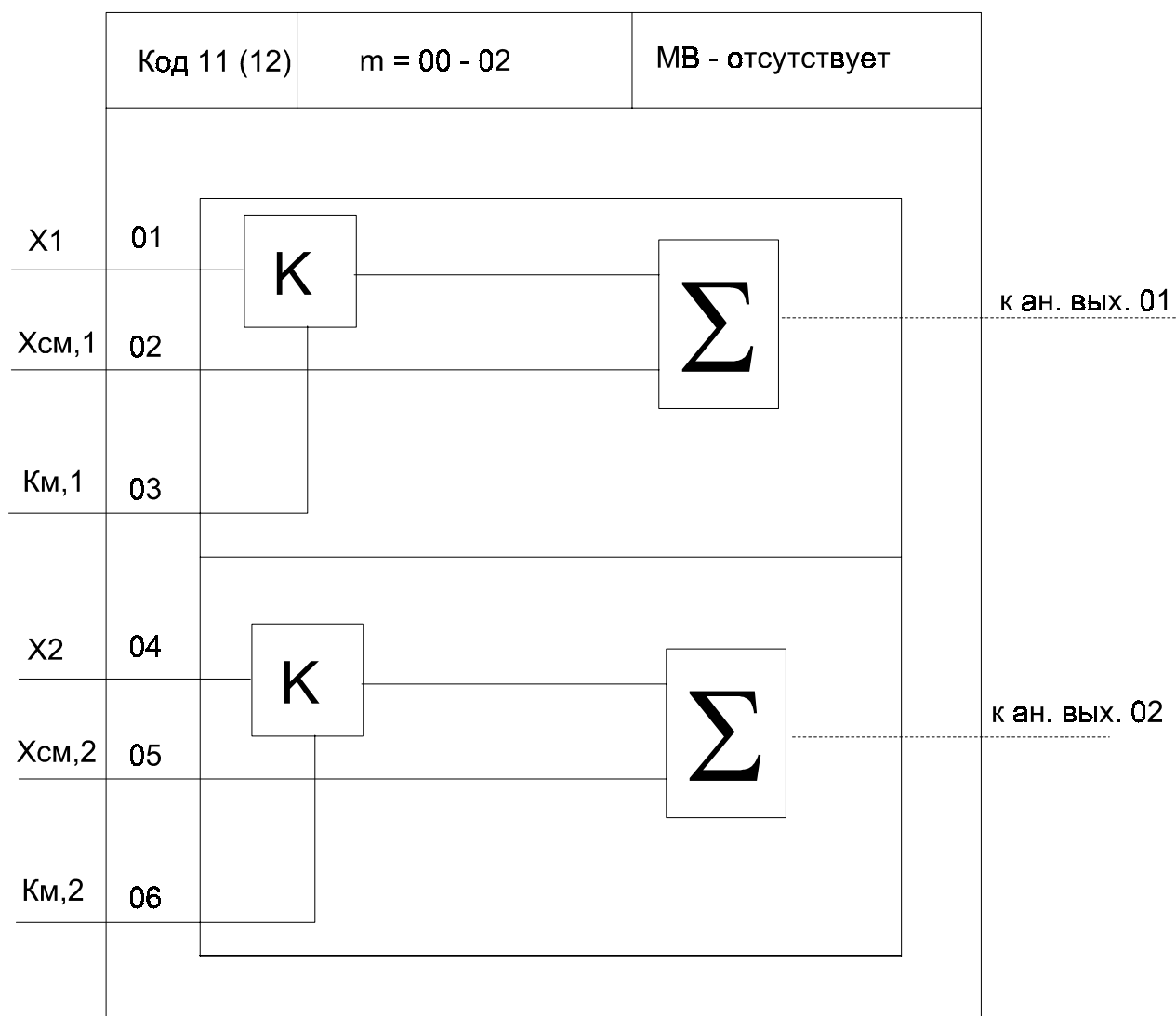


Рис. 9

3.6. ДВА(13) ДВБ(14) - Дискретный вывод групп А и Б

Назначение

Алгоритм применяется для связи функциональных алгоритмов с аппаратными средствами дискретного вывода (с ЦДП). Для связи с выходами А и В используются соответственно алгоритмы ДВА (ДВБ). Каждый алгоритм обслуживает до 16 дискретных выходов.

Описание алгоритма

Алгоритм содержит несколько идентичных независимых каналов, число которых $0 \leq m \leq 16$ и задается модификатором (рис. 10)

Если на i -й вход алгоритма поступает сигнал $D_i=0$, контакты i -го дискретного выхода разомкнуты. Если $D_i=1$, контакты i -го дискретного выхода замыкаются.

Входы алгоритма ДВА (ДВБ)

Таблица 13

Входы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Вход канала 1
02	C_2		Вход канала 2
.	.		.
.	.		.
.	.		.
m	C_m		Вход канала m

Функциональная схема алгоритма
"Дискретный вывод ДВА (ДВБ)"

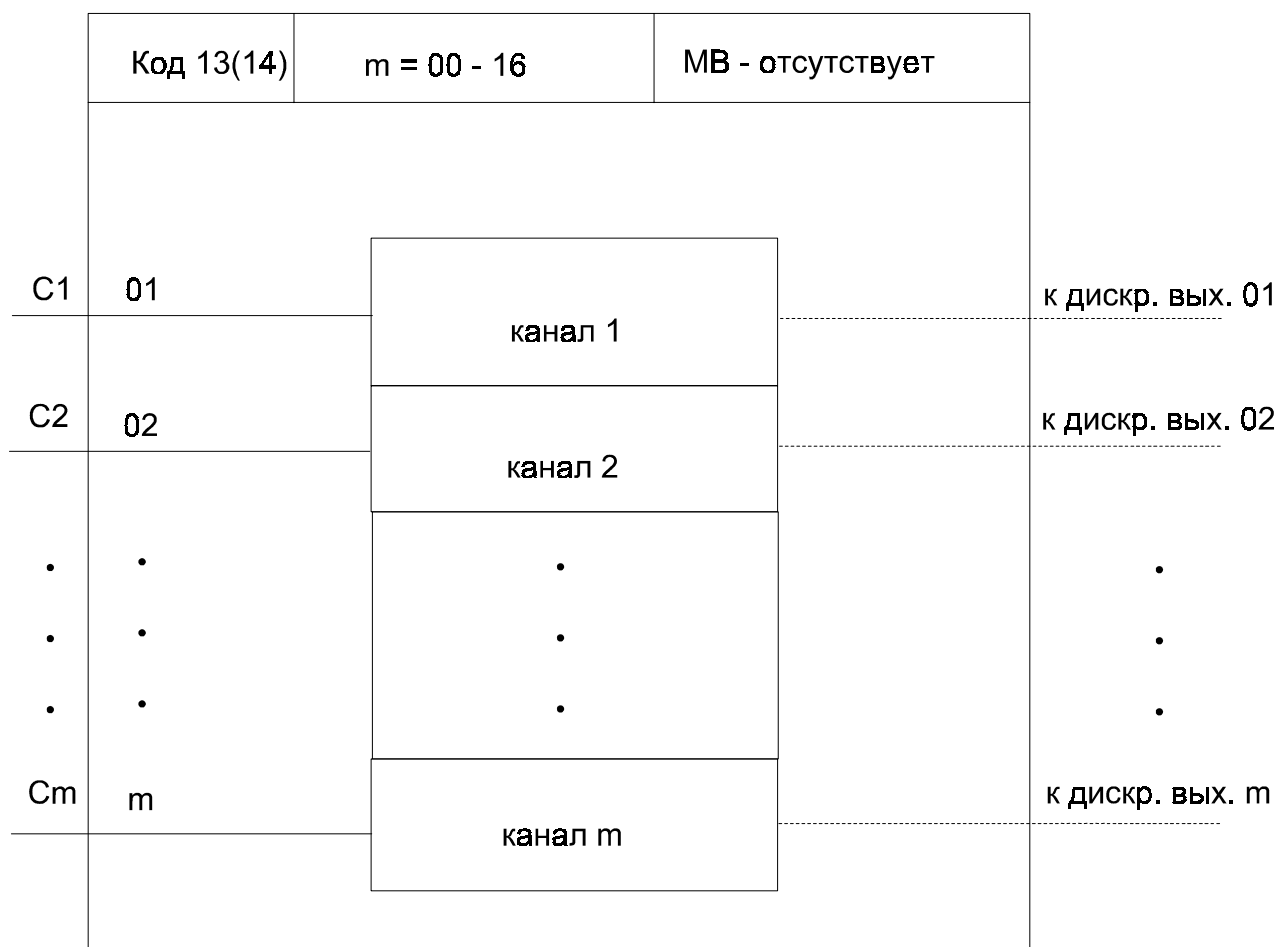


Рис. 10

3.7. ИВА(15) ИВБ(16) - Импульсный вывод групп А и Б

Назначение

Алгоритм применяется в тех случаях, когда контроллер должен управлять исполнительным механизмом постоянной скорости. Алгоритм преобразует сигнал, сформированный алгоблоками контроллера (в частности, алгоритмом импульсного регулирования), в последовательности импульсов переменной скважности.

Алгоритм выдает последовательность указанных импульсов на средства дискретного выхода контроллера. Для связи с выходами группы А и Б используются алгоритмы соответственно ИВА и ИВБ.

Описание алгоритма

Алгоритм содержит несколько (до 4) каналов связи с выходами контроллера. Число этих каналов $0 \leq m \leq 4$ задается модификатором (рис. 11). Если импульсный выход используется как в группе А, так и в группе Б, то общее число импульсных выходов (т.е. сумма модификаторов алгоритмов ИВА и ИВБ) не может быть больше 4.

Каждый канал алгоритма ИВА (ИВБ) содержит широтно-импульсный модулятор (ШИМ), преобразующий входной сигнал X в последовательность импульсов со скважностью Q , пропорциональной входному сигналу: $Q = |X|/100$. При $|X| \geq 100\%$ скважность $Q = 1$. Если $X > 0$, импульсы формируются в выходной цепи "больше", если $X < 0$, то в цепи "меньше". При $X = 0$ выходной сигнал равен нулю.

Параметр T задает минимальную длительность выходных импульсов. Этот параметр устанавливается в диапазоне $0,12 \leq T \leq 3,84$.

Параметр N определяет, к какому контуру регулирования относится данный канал алгоритма ИВА(ИВБ). Задание этого параметра необходимо лишь в том случае, когда требуется, чтобы синхронно с формированием выходных импульсов на лицевой панели контроллера зажигались ламповые индикаторы " $\triangle$ ", " ∇ " («больше», «меньше»). Например, если установлен параметр $N_1 = 1$, то при работе ШИМ1 на лицевой панели будут зажигаться индикаторы при вызове 1-го контура. Если задано $N > 4$ или $N < 1$, то индикаторы зажигаться не будут, какой бы контур ни был вызван на лицевую панель.

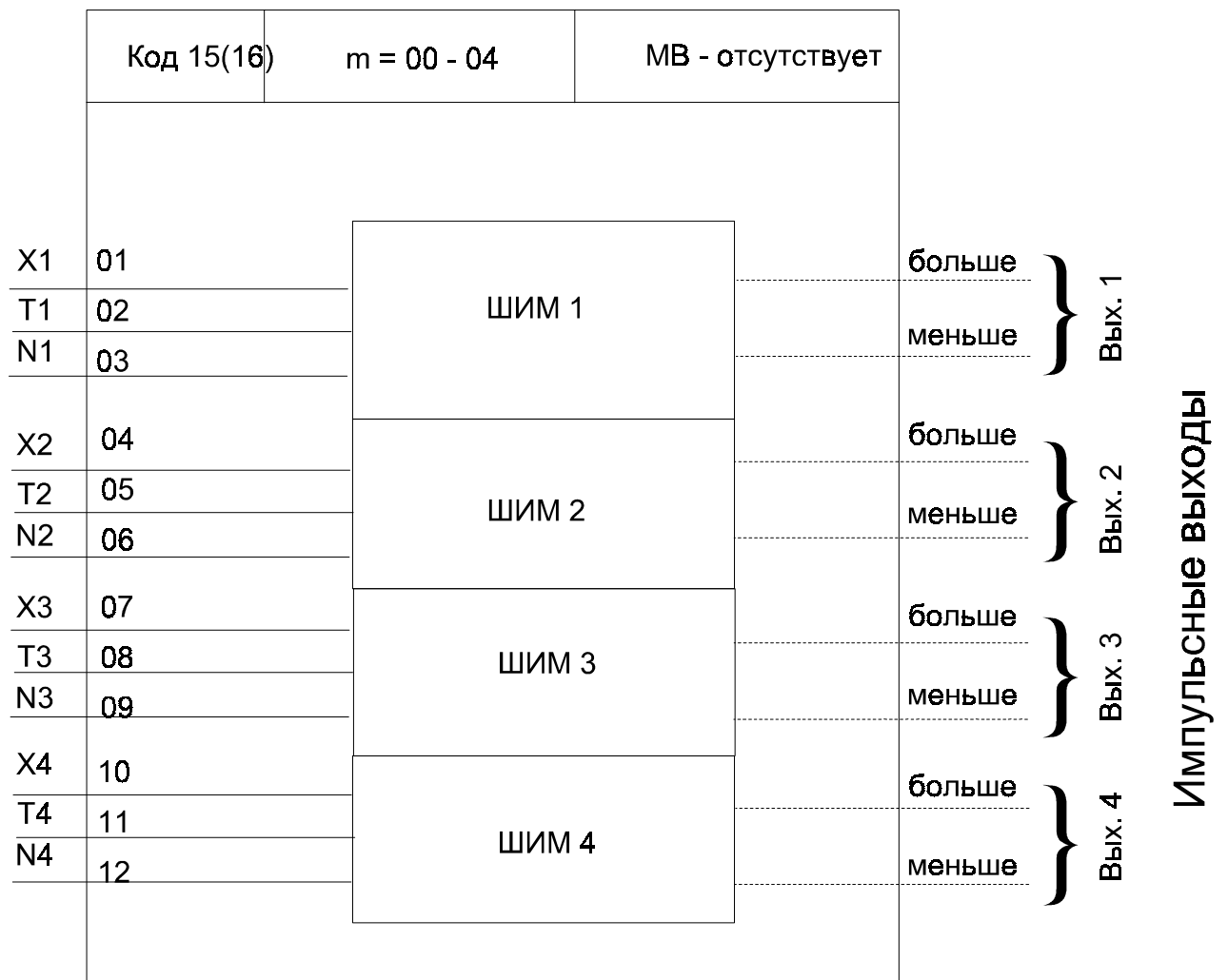
Если для нескольких ШИМ задан одинаковый номер N , то действует следующая система приоритетов: ШИМ группы Б приоритетны над ШИМ группы А; в пределах одной группы приоритетны ШИМ со старшими номерами.

Входы алгоритма ИВА(ИВБ)

Таблица 14

Входы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Сигнал 1-го выхода
02	T_1		Длительность импульса 1-го выхода
03	N_1		Номер, контура, с которым связан 1-й выход
04	X_2		То же, что X_1 , но для 2-го выхода
05	T_2		То же, что T_1 , но для 2-го выхода
06	N_2		То же, что N_1 , но для 2-го выхода
07	X_3		То же, что X_1 , но для 3-го выхода
08	T_3		То же, что T_1 , но для 3-го выхода
09	N_3		То же, что N_1 , но для 3-го выхода
10	X_4		То же, что X_1 , но для 4-го выхода
11	T_4		То же, что T_1 , но для 4-го выхода
12	N_4		То же, что N_1 , но для 4-го выхода

Функциональная схема алгоритма
"Импульсный вывод ИВА (ИББ)"



ШИМ - широтно-импульсный модулятор

Рис. 11

3.8. АВР(17) - Аварийный вывод

Назначение

Алгоритм позволяет алгоритмическими средствами сформировать два независимых сигнала на аварийных выходах контроллера: на выходе "отказ" и на выходе «отключение интерфейса». Алгоритм применяется в тех случаях, когда какая-либо ситуация (т.е. появление каких-либо сигналов, поступивших извне или сформированных внутри контроллера) должна рассматриваться либо как аварийная, либо как сигнал о том, что следует заблокировать связь контроллера с абонентами по интерфейсу каналу.

Алгоритм позволяет также выявить наличие короткого замыкания на дискретных или импульсных выходах контроллера.

Описание алгоритма

Если на вход отказа приходит сигнал $C_{отк}=1$, на аварийном выходе контроллера, формируется сигнал отказа. Сигнал отказа, сформированный алгоритмом, по схеме ИЛИ объединяется с сигналом отказа, сформированным средствами самодиагностики контроллера. При наличии сигнала отказа с помощью аппаратных средств формируется также сигнал "отключение интерфейса", т.е. при отказе связь контроллера по интерфейсу каналу блокируется (рис. 12).

Если на вход отключения интерфейса приходит сигнал $C_{инт}=1$, на аварийном выходе контроллера формируется сигнал отключения интерфейса. Этот сигнал, сформированный алгоритмом, по схеме ИЛИ объединяется с сигналом отключения интерфейса, сформированным средствами самодиагностики контроллера.

Алгоритм имеет два дискретных выхода, свидетельствующих о том, имеется ли короткое замыкание на дискретных или импульсных выходах контроллера. Если хотя бы на одном дискретном или импульсном выходе в группе А возникло короткое замыкание, сигнал $D_{кз,А}=1$, в противном случае $D_{кз,А}=0$. Аналогично выход $D_{кз,А}$ сигнализирует о коротком замыкании в группе Б.

Входы-выходы алгоритма АВР

Таблица 15

Входа-выхода			Назначение
№	Обозн.	Вид	
01	$C_{отк}$	Входы	Команда "отказ"
02	$C_{инт}$		Команда "отключение интерфейса"
01	$D_{кз,А}$	Выходы	Признак короткого замыкания на дискретных (импульсных) выходах группы А
02	$D_{кз,Б}$		То же для группы Б

**Функциональная схема алгоритма
"Аварийный вывод АВР"**

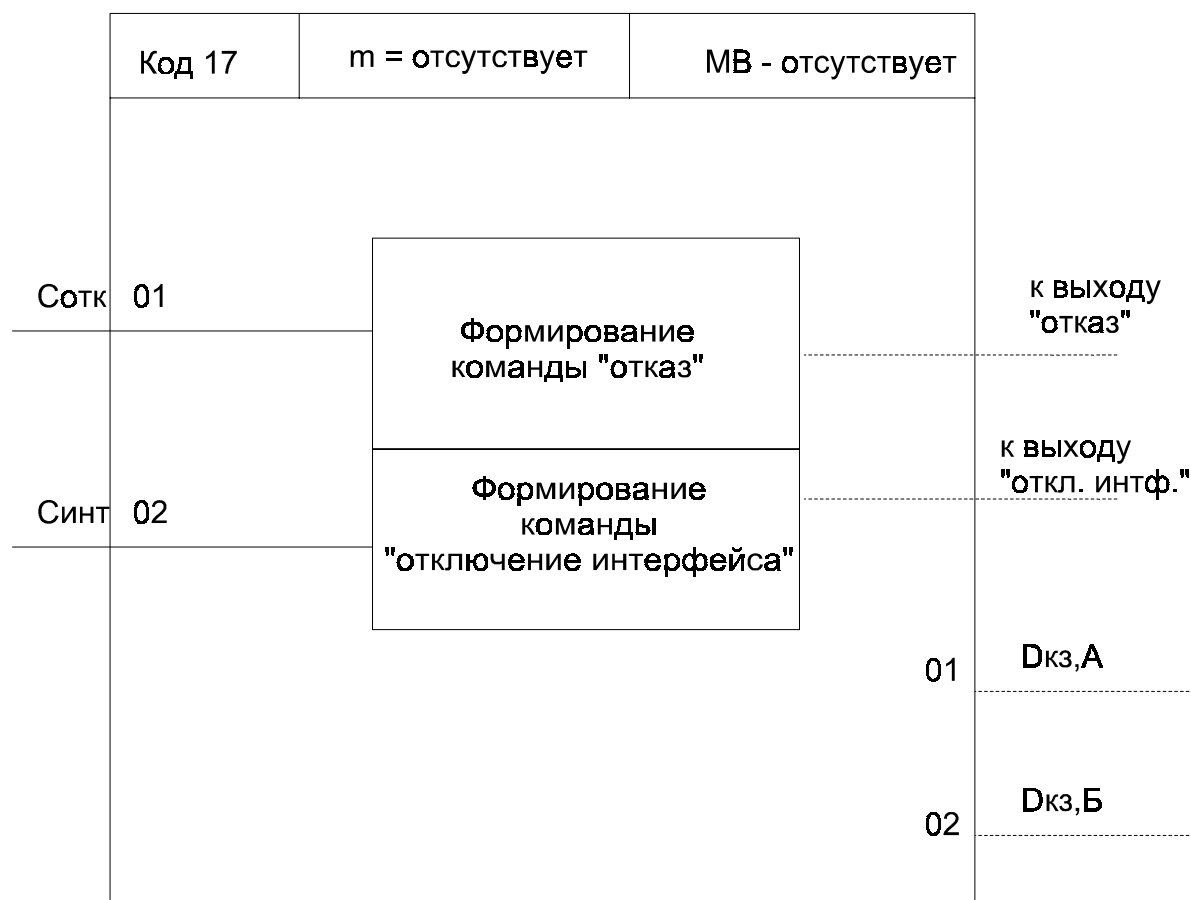


Рис. 12

4. АЛГОРИТМЫ РЕГУЛИРОВАНИЯ

4.1. РАН(20) - Регулирование аналоговое

Назначение

Алгоритм используется при построении ПИД регулятора имеющего аналоговый выход. Алгоритм как правило сочетается с пропорциональным исполнительным механизмом (позиционером) либо используется в качестве ведущего в схеме каскадного регулирования.

Помимо формирования ПИД закона в алгоритме вычисляется сигнал рассогласования, этот сигнал фильтруется, вводится зона чувствительности. Выходной сигнал алгоритма ограничивается по максимуму и минимуму. Алгоритм содержит узел настройки, позволяющий автоматизировать процесс настройки регулятора.

Описание алгоритма

Функциональная схема алгоритма (рис. 13), содержит несколько звеньев. Звено, выделяющее, сигнал рассогласования, суммирует два входных сигнала, при этом один из сигналов масштабируется, фильтруется и инвертируется. Сигнал рассогласования ϵ на выходе этого звена (без учета фильтра) равен:

$$\epsilon = X_1 - K_m * X_2, \quad (5)$$

где K_m – масштабный коэффициент.

Фильтр нижних частот первого порядка имеет передаточную функцию:

$$W(p) = 1 / (T_\phi * p + 1), \quad (6)$$

где T_ϕ – постоянная времени фильтра.

Зона нечувствительности не пропускает на свой выход сигнал рассогласования ϵ значение которого находится внутри установленного значения зоны.

Сигнал ϵ_2 на выходе этого звена равен:

$$\epsilon_2 = 0, \text{ при } |\epsilon| \leq X_D / 2 \quad (7)$$

$$\epsilon_2 = (|\epsilon| - X_D / 2) * \text{sign } \epsilon, \text{ при } |\epsilon| > X_D / 2, \quad (8)$$

где X_D – зона нечувствительности.

ПИД звено выполняет пропорционально-интегрально-дифференциальное преобразование сигнала и имеет передаточную функцию:

$$W(p) = K_p * \{ 1 + 1 / (T_i * p) + K_d * T_i * p / (1 + K_d * T_i * p)^2 \}, \quad (9)$$

где K_p , T_i , K_d – соответственно коэффициент пропорциональности,

постоянная времени интегрирования и постоянная времени дифференцирования, равный $K_d = T_d / T_i$ (при $K_d * T_i > 819$ значение $T_d = \infty$).

Алгоритм РАН может использоваться в качестве ПД или П - регулятора. В ПД - регуляторе устанавливается $T_d = \infty$. Если этот параметр установлен оператором вручную в режиме программирования, то интегральная ячейка анулируется и при переходе в режим работы алгоритм формирует передаточную функцию:

$$W(p) = K_p * \{ 1 + 819 * K_d * p / (1 + 819 * K_d / 8 * p)^2 \}, \quad (10)$$

при этом если $K_d \leq 1$ постоянная времени дифференцирования $T_d = 819 * K_d$ (сек, мин или час в зависимости от выбранного диапазона и масштаба времени), если $K_d > 1$, то $T_d = \infty$.

Если используется автонастройка и в процессе автоматически устанавливается $T_i = \infty$, то также формируется передаточная функция (10), но интегральная ячейка сохраняется и сигнал этой ячейки замораживается к добавляется к выходу ПД звена как постоянное смещение.

Для получения П-регулятора следует установить $T_i = \infty$ и $K_d = 0$. Свойства интегральной ячейки при этом остаются такими же, как в ПД-регуляторе.

Ограничитель ограничивает выходной сигнал алгоритма по максимуму и минимуму. Уровни ограничения устанавливаются коэффициентами $X_{\max}$, $X_{\min}$.

Диапазон параметров настройки - стандартный для Р-13О (см. табл.2). Значение $X_D < 0$ воспринимается алгоритмом соответственно как $X_D = 0$.

Помимо двух сигнальных входов X_1 и X_2 алгоритм имеет 8 настроечных входов, которые задают параметры настройки алгоритма.

Алгоритм содержит узел настройки, состоящий из переключателя режима "работа - настройка", нуль-органа и дополнительного фильтра с постоянной T_ϕ , 1. Свойства нуль-органа описываются выражением:

$$y_{но} = x_{но} \quad \text{при } \epsilon \geq 0;$$

$$y_{но} = -x_{но} \quad \text{при } \epsilon < 0.$$

где $y_{но}$ - сигнал на выходе нуль-органа.

При дискретном сигнале на входе $C_{нас} = 1$ алгоритм переходит в режим настройки и в замкнутом контуре регулирования устанавливаются автоколебания. Параметры этих колебаний (амплитуда и период), которые контролируются на выходе $Y\epsilon$, используются для определения параметров настройки регулятора (методика настройки приведена в описании алгоритма автонастройки АНР).

Для перехода в режим работы устанавливается сигнал $C_{нас} = 0$.

Алгоритм имеет 4 выхода. Выход Y - основной выход алгоритма. На выходе $Y\epsilon$ формируется отфильтро-

ванный сигнал рассогласования. Два дискретных выхода фиксируют момент наступления ограничения выходного сигнала Y . Логика формирования выходных дискретных сигналов определяется следующей таблицей (здесь Y_1 - сигнал на входе звена ограничения):

Таблица 16

Y_1	Y	$D_{\text{макс}}$	$D_{\text{мин}}$
$X_{\text{мин}} < Y_1 < X_{\text{макс}}$	$Y = Y_1$	6	0
$Y_1 \geq X_{\text{макс}}$	$Y = X_{\text{макс}}$	1	0
$Y_1 \leq X_{\text{мин}}$	$Y = X_{\text{мин}}$	0	1

Алгоритм будет правильно работать, только если $X_{\text{макс}} > X_{\text{мин}}$.

Режимы работы

Алгоритм относится к группе следящих (см. п. 1.4).

Команда отключения вместе с значением начальных условия Y_0 поступает на каскадный выход Y алгоритма. В режиме отключения работа алгоритма изменяется следующим образом:

1) интегральная ячейка ПИД звена «заряжается» до значения

$$Y_{\text{и}} = Y_0 - K_{\text{п}} * \epsilon_2, \quad (11)$$

где ϵ_2 - определяется выражением (8)

2) D - составляющая обнуляется;

3) при выполнении необходимых условий входной сумматор переходит в режим обратного счета, при этом на каскадном входе X_1 формируется команда отключения и устанавливается значение начальных условий $X_0 = K_{\text{п}} * X_2$; эти сигналы передаются предвключенному алгоритму.

В режиме запрета алгоритм переходит в двух случаях:

1) по собственной инициативе, когда выходной сигнал ПИД-звена вышел за установленные пороги ограничения;

2) если команда запрета поступает извне на каскадный выход Y .

В любом случае в режиме запрета выполняются следующие операции:

1) блокируется изменение I - составляющей ПИД-звена в запрещенном направлении;

2) при выполнении необходимых условий на каскадном входе X_1 формируется команда запрета, которая передается предвключенному алгоритму.

В остальном работа алгоритма в режиме запрета не изменяется.

При работе в отключенном режиме сигнал Y_0 , "навязанный" алгоритму извне, может выйти за пороги ограничения. Ограничитель не препятствует этому, но после перехода в стандартный режим изменение выходного сигнала возможно лишь в направлении, приближающем выходной сигнал к установленным порогам ограничения. Если выходной сигнал изменяется именно в этом направлении, команда запрета не формируется; в противном случае формируется команда запрета, которая через вход X_1 транслируется предвключенному алгоритму.

После того, как выходной сигнал выйдет из области ограничения и вернется на линейный участок, выходной сигнал вновь может изменяться в обоих направлениях.

Алгоритм содержит узел настройки, состоящий из переключателя режима "работа-настройка", нуля-органа и дополнительного фильтра с постоянной времени $T_{\text{ф},1}$.

Свойства нуля-органа описываются выражением:

$$u_{\text{но}} = x_{\text{но}} \quad \text{при } \epsilon \geq 0;$$

$$u_{\text{но}} = -x_{\text{но}} \quad \text{при } \epsilon < 0.$$

где $u_{\text{но}}$ - сигнал на выходе нуля-органа.

При дискретном сигнале на входе $C_{\text{нас}} = 1$ алгоритм переходит в режим настройки и в замкнутом контуре регулирования устанавливаются автоколебания. Параметры этих колебаний (амплитуда и период), которые контролируются на выходе $u_{\text{е}}$, используются для определения параметров настройки регулятора (методика настройки приведена в описании алгоритма автонастройки АНР).

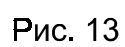
Для перехода в режим работы устанавливается сигнал $C_{\text{нас}} = 0$.

Входы-выходы алгоритма РАН

Таблица 17

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1		Немасштабируемый вход (каскадный)
02	X_2		Масштабируемый вход
03	$K_{\text{м}}$		Масштабный коэффициент
04	$T_{\text{ф}}$		Постоянная времени фильтра
05	X_{Δ}		Зона нечувствительности

06	K_p	Входы	Коэффициент пропорциональности
07	T_n		Постоянная времени интегрирования
08	K_d		Постоянная времени дифференцирования
09	$X_{\text{макс}}$		Уровень ограничения по максимуму
10	$X_{\text{мин}}$		Уровень ограничения по минимуму
11	$C_{\text{нас}}$		Команда перехода в режим настройки
12\	$X_{\text{но}}$		Уровень сигнала на выходе нуля-органа
13	K_I		Коэффициент, устанавливаемый в зависимости от свойств объекта.
01	Y	Выходы	Основной выход алгоритма (каскадный)
02	Y_e		Сигнал рассогласования
03	$D_{\text{макс}}$		Ограничение по максимуму
04	$D_{\text{мин}}$		Ограничение по минимуму



4.2. РИМ(21) - Регулирование импульсное

Назначение

Алгоритм используется при построении ПИД регулятора, работающего в комплекте с исполнительным механизмом постоянной скорости. Алгоритм как правило применяется в сочетании с алгоритмом импульсного вывода ИВА(ИБВ), который преобразует выходной аналоговый сигнал алгоритма РИМ в последовательность импульсов; управляющих исполнительным механизмом. Помимо формирования закона, регулирования в алгоритме вычисляется сигнал рассогласования, этот сигнал, фильтруется, вводится зона нечувствительности. Алгоритм содержит узел настройки, позволяющий автоматизировать процесс настройки регулятора.

Описание алгоритма

Функциональная схема алгоритма содержит несколько звеньев (рис. 14).

Звено, выделяющее сигнал рассогласования, суммирует два входных сигнала задания и регулирования параметра, при этом один из сигналов масштабируется фильтруется и инвертируется. Сигнал рассогласования на выходе этого звена (без учета фильтра) равен:

$$\varepsilon = X_1 - K_m * X_2, \quad (12)$$

где K_m - масштабный коэффициент.

Фильтр нижних частот первого порядка имеет передаточную функцию:

$$W(p) = 1 / (T_\phi * p + 1), \quad (13)$$

где T_ϕ - постоянная времени фильтра.

Зона нечувствительности не пропускает на свой выход сигналы, значения которых находятся внутри установленного значения зоны. Сигнал E_2 на выходе этого звена равен:

$$E_2 = 0 \text{ при } |\varepsilon| \leq X_\Delta / 2; \quad (14)$$

$$E_2 = (|\varepsilon| - X_\Delta / 2) * \text{sign } \varepsilon \text{ при } |\varepsilon| > X_\Delta / 2, \quad (15)$$

где X_Δ - зона нечувствительности.

ПИД²- звено имеет передаточную функцию:

$$W(p) = K_p * T_m / T_i * [1 + T_i * p + T_i^2 * K_d * p / (1/8 * K_d * T_i * p + 1)^2], \quad (16)$$

где T_m - время полного перемещения исполнительного механизма, движущегося с максимальной скоростью.

В сочетании с интегрирующим исполнительным механизмом, имеющим передаточную функцию $W_{\text{им}}(p) = 1 / (T_m * p)$, общая передаточная функция регулятора с алгоритмом РИМ имеет вид:

$$W_{\text{рег}}(p) = W(p) * W_{\text{им}}(p) = K_p * [1 + 1 / (T_i * p) + K_d * T_i * p / (1/8 * K_d * T_i * p + 1)^2], \quad (17)$$

где K_p , T_i , K_d - соответственно коэффициент пропорциональности, постоянная времени интегрирования и коэффициент дифференцирования, равный $K_d = T_d / T_i$ (при $K_d * T_i > 819$ значение $T_d = \infty$).

Помимо двух сигнальных входов алгоритм имеет 6 настроечных входов. Диапазон параметров настройки - стандартный для Р-130 (см. табл.2). Значение $X_\Delta < 0$ воспринимается как $X_\Delta = 0$.

Алгоритм имеет 2 выхода. Выход Y - основной выход алгоритма, на выходе Y_e формируется отфильтрованный сигнал рассогласования.

Режим работы

Алгоритм относится к группе следящих и имеет каскадный выход Y и один каскадный вход X .

Команда отключения поступает извне на выход Y . В режиме отключения звенья D и D^2 обнуляются и при выполнении необходимых условий алгоритм выполняет процедуру обратного счета, формируя на каскадном входе X_1 сигнал $X_0 = K_m * X_2$. На команду запрета алгоритм не реагирует. Поступившие извне команды отключения и запрета вместе с значением начальных условия X_0 транслируется алгоритмом через вход X_1 предвключенному алгоритму.

Алгоритм содержит узел настройки, состоящий из переключателя режима "работа-настройка", нуля органа и дополнительного фильтра с постоянной времени $T_{\phi,1}$.

Свойства нуля органа описываются выражением:

$$Y_0 = X_{\text{но}} \text{ при } \varepsilon \geq 0;$$

$$Y_0 = -X_{\text{но}} \text{ при } \varepsilon < 0,$$

где Y_0 - сигнал на выходе нуля-органа.

При дискретном сигнале на входе $C_{\text{нас}} = 1$ алгоритм переходит в режим настройки и в замкнутом контуре регулирования устанавливаются автоколебания. Параметры этих колебаний (амплитуда и период), которые контролируются на выходе Y_e , используются для определения параметров настройки регулятора (методика настройки приведена в описании алгоритма автонастройки АНР).

Для перехода в режим работы устанавливается сигнал $C_{\text{нас}} = 0$.

Входы-выходы алгоритма РИМ

Таблица 18

Входы-выходы			Назначение
№	Обозн.	Вид	

01	X_1	Входы	Немасштабируемый вход (каскадный)
02	X_2		Масштабируемый вход
03	K_m		Масштабный коэффициент
04	T_Φ		Постоянная времени фильтра
05	X_Δ		Зона нечувствительности
06	K_p		Коэффициент пропорциональности
07	T_i		Постоянная времени интегрирования
08	K_d		Постоянная времени дифференцирования
09	T_m		Время исполнительного механизма
10	-		Не используется
11	$C_{нас}$		Команда перехода в режим настройки
12	$X_{но}$		Уровень сигнала на выходе нуль-органа
13	K_1		Коэффициент, устанавливаемый в зависимости от свойств объекта.
01	Y	Выходы	Основной выход алгоритма (каскадный)
02	Y_ϵ		Сигнал рассогласования

Функциональная схема алгоритма
"Регулирование импульсное РИМ"

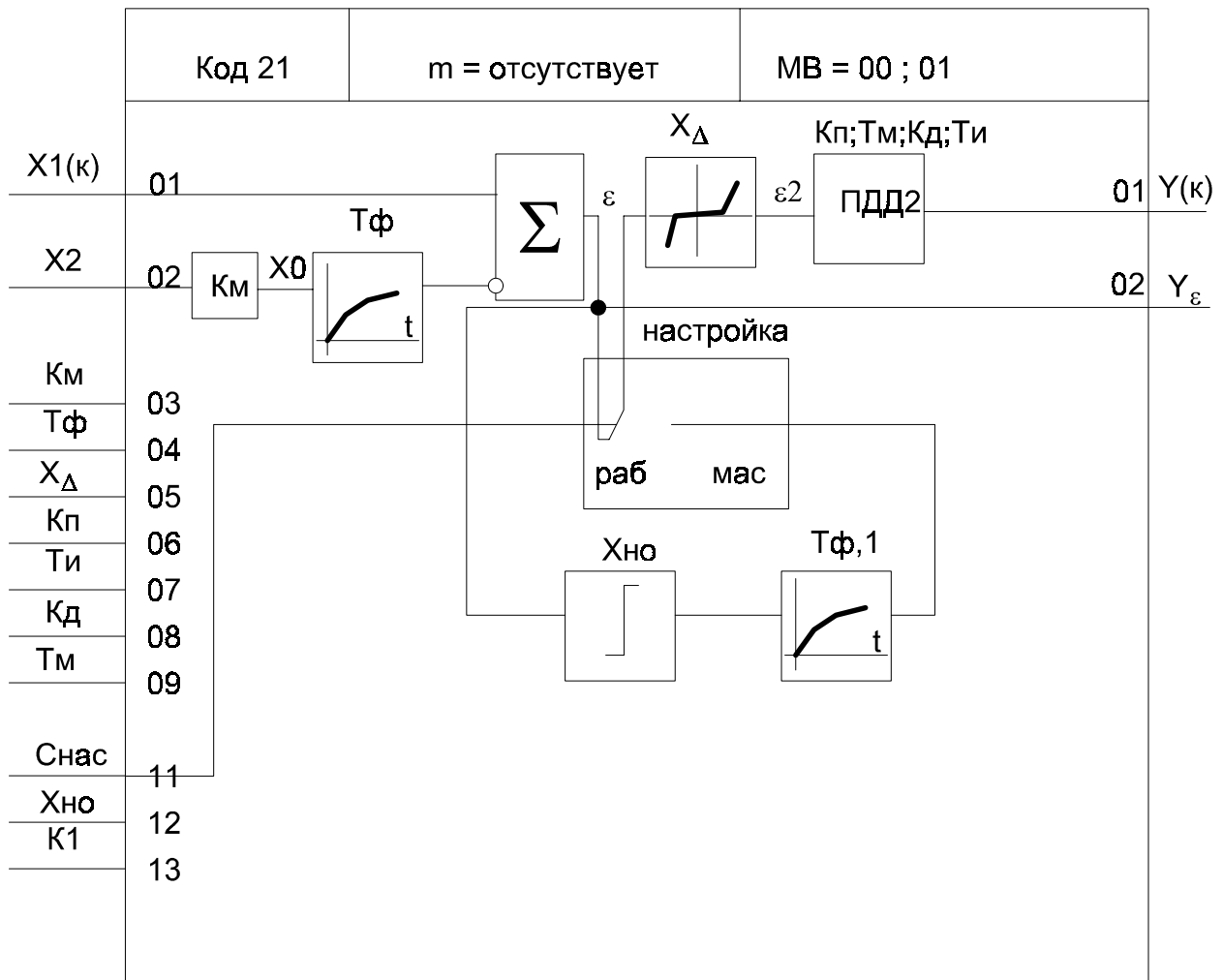


Рис. 14

4.3. ЗДН(24) – Задание

Назначение

Алгоритм применяется для формирования сигнала ручного задания в контуре регулирования. Через этот алгоритм к регулятору подключаются также программные задатчики и сигнал внешнего задания.

Алгоритм применяется в сочетании с алгоритмом ОКО.

Описание алгоритма

Алгоритм содержит узел ручного задания, узел динамической балансировки, переключатель вида задания и переключатель программ (рис. 15).

Алгоритм имеет модификатор $0 \leq m \leq 40$, который определяет число независимых программных задатчиков, подключаемых к регулятору. Если Программные задатчики не используются, то устанавливается $m=0$.

С помощью переключателя вида задания выбирается один из трех видов задания: ручное задание ("РЗ"), программное задание («ПЗ») или внешнее задание ("ВЗ"). Для того, чтобы вид задания можно было изменить с помощью клавиш, расположенных на лицевой панели контроллера, к выходу $Y_{здн}$ подключается вход $X_{здн}$ алгоритма оперативного контроля ОКО. Такое подключение позволяет также с помощью клавиш лицевой панели управлять переключателем программ (см. также описание алгоритма ОКО).

Переключение на программное задание возможно, только если модификатор $m > 0$. Переключение на внешнее задание возможно, только если соответствующий режим предусмотрен в алгоритме ОКО, связанном с данным алгоритмом ЗДН (см. табл.19).

В режиме ручного задания сигнал задания изменяется вручную. В режиме программного задания сигнал задания поступает со входов $X_{пр,i}$ (обычно к этим входам подключаются программные задатчики). В режиме внешнего задания поступает со входа $X_{вн}$. Этот вход может быть подключен к выходу любого алгоблока, в частности – к алгоблоку ввода аналогового или интерфейсного. В этом случае сигнал внешнего задания поступают соответственно через аналоговый вход контроллера или через интерфейсный канал.

Алгоритм ЗДН содержит встроенный механизм статической и динамической балансировки.

Статическая балансировка действует только на узел ручного задатчика. Если статическая балансировка не включена ($C_{сб}=0$), сигнал, вырабатываемый узлом ручного задатчика при переключениях режима задания и отключении алгоритма не изменяется. При включенной статической балансировке ($C_{сб}=1$) отключенный узел ручного задатчика отслеживает либо текущее задание (в режимах «ПЗ» и «ВЗ»), либо начальное значение, поступающее на выход алгоритма (в отключенном режиме). После включения ручного задатчика последнее значение сигнала задания запоминается, но затем это значение может быть изменено вручную.

Если включена динамическая балансировка ($C_{дб}=1$), то при любых переключениях вида задания или отключении алгоритма ЗДН командой обратного счета узел динамической балансировки вырабатывает сигнал компенсации, с помощью которого выходной сигнал $Y_{здн}$ в первый момент после переключения сохраняется неизменным. Затем сигнал компенсации уменьшается (по модулю) до нуля с постоянной скоростью, задаваемой входом $V_{дб}$, при этом выходной сигнал $Y_{здн}$ плавно (безударно) переходит к текущему значению ручного, программного или внешнего задания.

Возможен вариант, когда включена как статическая, так и динамическая балансировка (т. е. $C_{сб}=C_{дб}=1$). В этом случае узел ручного задания работает в соответствии с правилами статической балансировки, а при переключениях на режим программного и внешнего задания, изменения номера программы и включении алгоритма, работающего в режимах «ПЗ» и «ВЗ», действует динамическая балансировка.

Если входы $C_{сб}$ и (или) $C_{дб}$ подключить к алгоблокам, формирующим те или иные дискретные сигналы, то в зависимости от значений этих сигналов процедуры балансировки будут автоматически включаться и выключаться. Например, можно сделать так, чтобы при переходе в локальный режим включилась динамическая балансировка, при отключении контура регулирования каким-либо переключателем включилась статическая балансировка, в ручном режиме балансировка отсутствовала и т. д.

Помимо основного выхода $Y_{здн}$ в алгоритме имеется ряд дополнительных выходов.

Дискретные выходы $D_{руз}$, $D_{внш}$, $D_{прг}$ индицируют установленный вид задания. Если задание ручное, то $D_{руз}=1$, а $D_{внш}=D_{прг}=0$; если задание внешнее, $D_{внш}=1$, а $D_{руз}=D_{прг}=0$; если задание программное, то $D_{прг}=1$, а $D_{руз}=D_{внш}=0$.

Остальные выходы алгоритма ЗДН. имеются только в том случае, если задан хотя бы один программный задатчик, т. е. если модификатор алгоритма $m > 0$.

Число на выходе N_n равно номеру выполняемой программы (т. е. положению переключателя программ).

Число на выходе $N_{уч}$ равно текущему номеру участка программы.

Время T_n – время, оставшееся до конца текущего участка программы.

Если в алгоритмах программного задатчика ПРЗ задано число повторений $N_{повт} > 1$, то дискретный сигнал $D_{кпв}$ (конец повторений) при окончании каждого текущего повторения программы на один цикл работы контроллера (т.е. на время $T_0 = 0,2-2с$) устанавливается $D_{кпв}=1$, после чего $D_{кпв}=0$. Когда окончится последнее повторение, сигналы $D_{кп}=D_{кпв}=1$ постоянно, пока программа не будет сброшена ($D_{кп}$ - конец программы). Если $N_{повт}=1$, то после окончания программы $D_{кп}=D_{кпв}=1$.

Число на выходе $N_{ост}$ равно числу оставшихся повторений программы.

Дискретные сигналы $D_{пс}$, $D_{ст}$, $D_{сбр}$ равны лог. 1, когда программа находится соответственно в состоянии

пуска, останова и сброса.

Режим работы.

Алгоритм является инициатором команды отключения. Эта команда формируется на каскадном входе $X_{\text{вн}}$, если установлен ручной или программный вид задания. Команда отключения вместе с значением начальных условий $X_0=Y_{\text{здн}}$ (т.е. текущим значением сигнала задания) транслируется предвключенному алгоритму.

Выход алгоритма $Y_{\text{здн}}$ каскадный. Если на этот выход поступает команда отключения, то эта команда вместе с значением начальных условий Y_0 через вход $X_{\text{вн}}$ транслируется предвключенному алгоритму.

На входах $X_{\text{пр},i}$ команда отключения не формируется, поэтому на работу программных задатчиков, подключенных к этим входам, команда отключения не влияет. Работа программных задатчиков не зависит от положения переключателя вида задания.

Если в алгоритме установлена статическая или динамическая балансировка, то внешняя команда отключения, поступающая на выход $Y_{\text{здн}}$, инициирует процедуру балансировки и эта команда вместе с значением начальных условий Y_0 транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, поступающую на выход алгоритма, но транслирует эту команду через вход $X_{\text{вн}}$ предвключенному алгоритму (только при наличии статической или динамической балансировки).

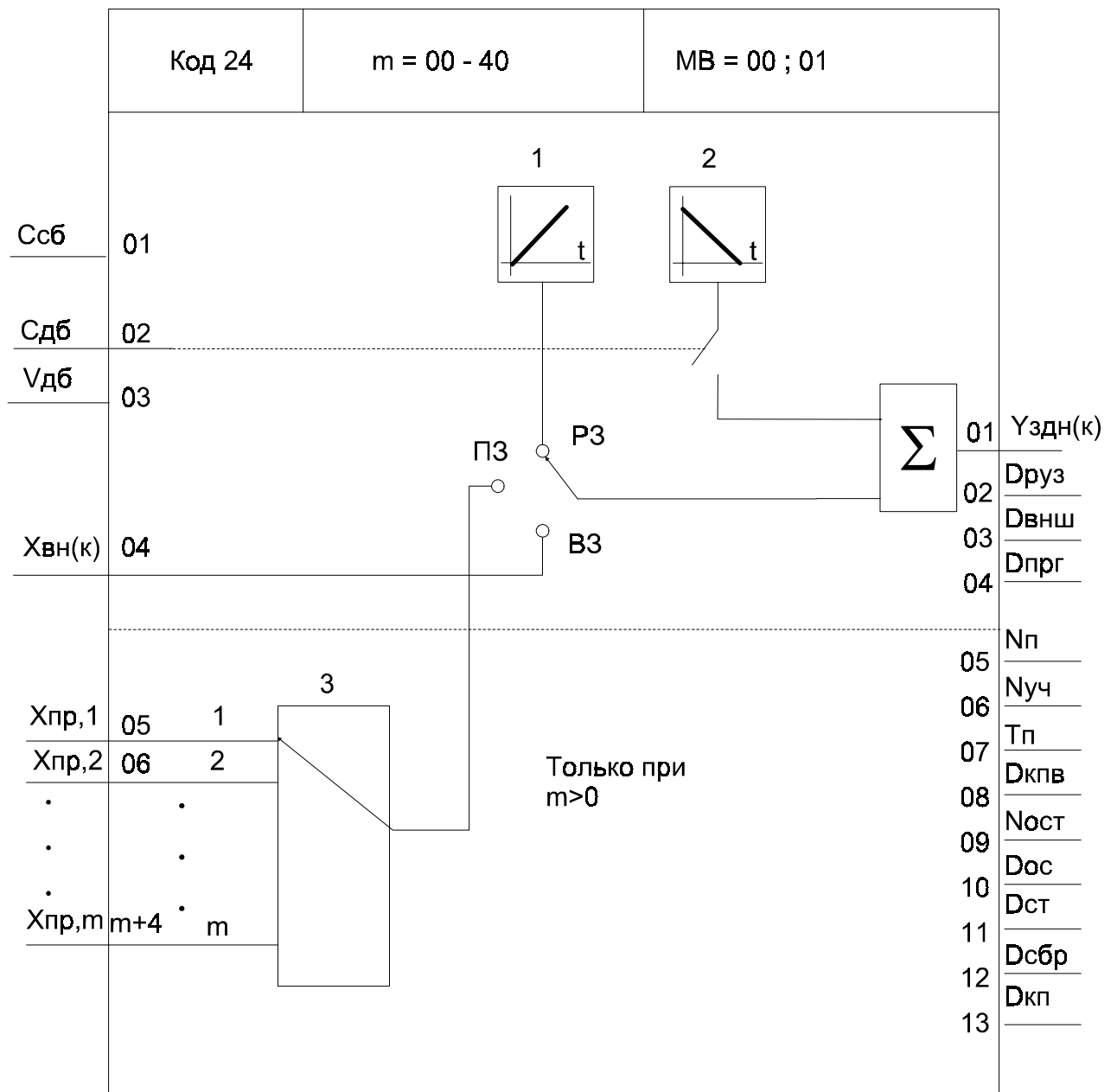
Если статическая и динамическая балансировка отсутствует, внешние команды отключения и запрета алгоритмом не воспринимаются.

Входы-выходы алгоритма ЗДН

Таблица 19

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{\text{сб}}$	Входы	Включение статической балансировки
02	$C_{\text{дб}}$		Включение динамической балансировки
03	$V_{\text{дб}}$		Скорость динамической балансировки
04	$X_{\text{вн}}$		Сигнал внешнего задания (каскадный)
05	$X_{\text{пр},1}$		Вход для 1-го программного задатчика
06	$X_{\text{пр},2}$		Вход для 2-го программного задатчика
•	•		•
•	•		•
m+4	$X_{\text{пр},m}$	Выходы	Вход для m-го программного задатчика
01	$Y_{\text{здн}}$		Основной выход (каскадный)
02	$D_{\text{руз}}$		Режим ручного задания
03	$D_{\text{внш}}$		Режим внешнего задания
04	$D_{\text{прг}}$		Режим программного задания
05	$N_{\text{п}}$		Номер текущей программы
06	$N_{\text{уч}}$		Номер текущего участка
07	$T_{\text{п}}$		Время, оставшееся до окончания текущего участка
08	$D_{\text{кпв}}$		Конец очередного повторения программы
09	$N_{\text{ост}}$		Оставшееся число повторений
10	$D_{\text{пс}}$		Программа в состоянии "пуск"
11	$D_{\text{ст}}$		Программа в состоянии "стоп"
12	$D_{\text{сбр}}$		Программа в состоянии "сброс"
13	$D_{\text{кп}}$		Конец программы

Функциональная схема алгоритма
"Задание ЗДН"



1-узел ручного задатчика;
2-узел динамической балансировки;
3-переключатель программ

Рис. 15

4.4. ЗДЛ(25) - Задание локальное

Назначение

Алгоритм используется в составе каскадного регулятора. Он необходим, если должно предусматриваться переключение каскадного регулятора в локальный режим и ручное изменение задания ведомого регулятора в этом режиме. Алгоритм ЗДЛ используется в сочетании с алгоритмом ОКО.

Описание алгоритма

Алгоритм ЗДЛ включается между ведомым и ведущим регулятором в каскадной схеме. Алгоритм содержит узел ручного задания и переключатель режимов (рис. 16). В положении "КУ" (каскадное управление) сигнал на входе X алгоритма передается на его выход. Обычно вход X подключается к выходу ведущего регулятора, а к выходу Y подключается вход ведомого регулятора. В этом случае в каскадном режиме алгоритм ЗДЛ связывает между собой ведущий и ведомый регулятор.

При переходе в локальный режим связь между ведущим и ведомым регуляторами разрывается и к выходу алгоритма ЗДЛ подключается узел ручного задания. Это задание может изменяться вручную с помощью клавиш лицевой панели (см. также описание алгоритма ОКО).

На дискретном выходе $D_{\text{лок}}$ устанавливается значение $D_{\text{лок}}=0$, если режим каскадный, и $D_{\text{лок}}=1$, если режим локальный.

Режимы работы

Алгоритм ЗДЛ является инициатором команды отключения. Эта команда формируется на каскадном входе X в том случае, когда установлен локальный режим управления. Эта команда вместе с значением начальных условий $X_0=Y$ передается предвключенному алгоритму.

При каскадном режиме ручной задатчик отслеживает текущее значение выходного сигнала алгоритма, так что переключение в локальный режим происходит безударно.

На каскадном выходе Y может поступить внешняя команда отключения или запрета. Если алгоритм работает в режиме "ЛУ", эти команды алгоритмом не воспринимаются. При работе в режиме "КУ" эти команды вместе с значением начальных условий Y_0 через вход X транслируются предвключенному алгоритму.

Входы-выходы алгоритма ЗДЛ

Таблица 20

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Вход	Подключение к ведущему регулятору (каскадный)
01	Y	Выходы	Основной выход (каскадный)
02	$D_{\text{лок}}$		Режим управления локальный

Функциональная схема алгоритма
"Задание локальное ЗДЛ"

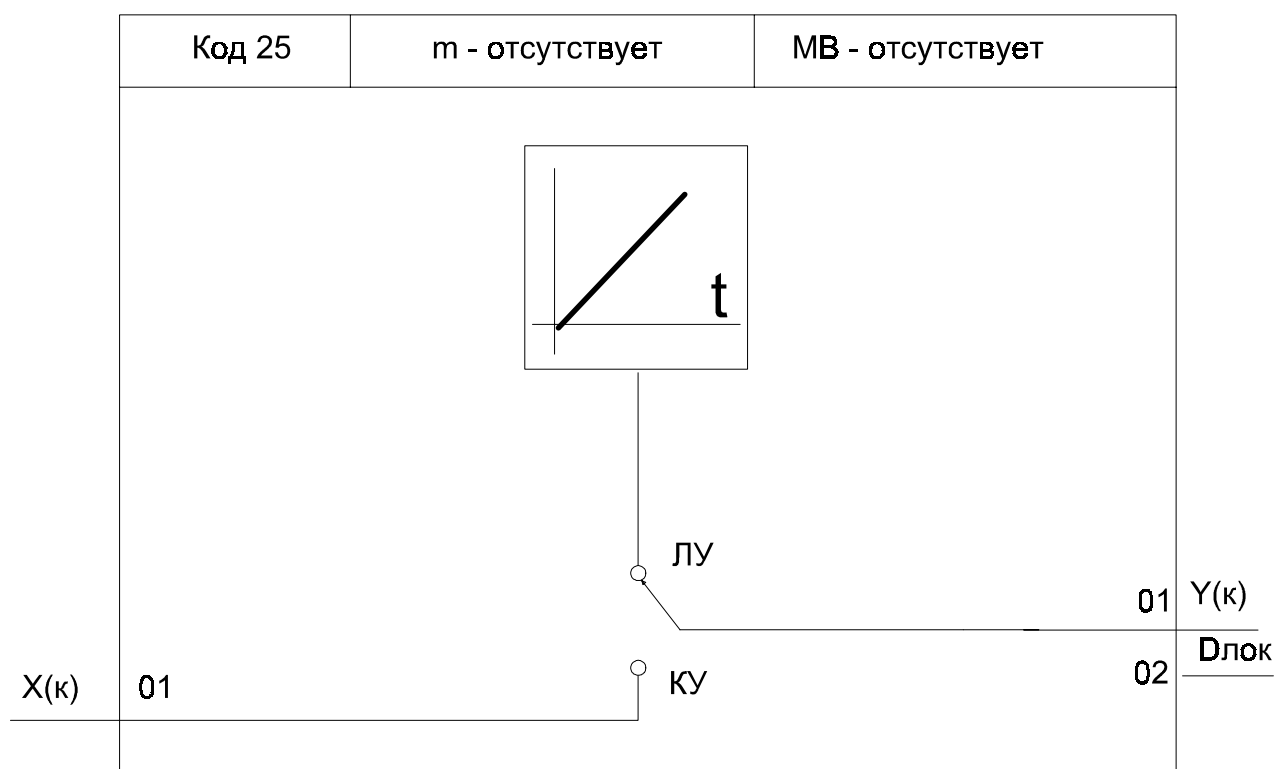


Рис. 16

4.5. РУЧ(26) – Ручное управление

Назначение

Алгоритм предназначен для изменения режима управления регулятора. С его помощью регулятор переключается в дистанционный или ручной режим работы. В ручном режиме выходной сигнал изменяется вручную.

Алгоритм РУЧ применяется в составе как аналогового, так и импульсного регулятора и используется в сочетании с алгоритмом ОКО.

Описание алгоритма

Алгоритм содержит переключатель режима работы и узел ручного управления (рис. 17).

Если на лицевой панели контроллера нажимается клавиша ручного режима, к выходу алгоритма РУЧ подключается узел ручного управления. Если в алгоритме ОКО, связанным с данным алгоритмом РУЧ, задан аналоговый регулятор, то узел ручного управления алгоритма РУЧ работает по принципу «больше» – «меньше». Если в алгоритме ОКО задан импульсный регулятор, то узел ручного управления формирует константу, определяющую среднюю скорость перемещения исполнительного механизма. Эта константа равна –100, –25, 0, 25 или 100 % (см. также описание функций оперативного управления).

Если на лицевой панели контроллера нажимается клавиша автоматического режима управления, узел ручного управления отключается и выход Y алгоритма РУЧ соединяется с его входом X или $X_{дст}$. С входом X выход Y соединяется в локальном и каскадном режиме, с входом $X_{дст}$ – в дистанционном режиме.

Вход X в принципе может подключаться к выходу любого алгоблока, но в типовом случае он подключается к выходу алгоритма регулирования. Вход $X_{дст}$ также может лодключаться к выходу любого алгоритма. В частности, если он подключается к выходу алгоритма цифрового ввода, то в дистанционном режиме команда управления исполнительным механизмом поступает через интерфейсный канал.

Алгоритм может быть принудительно переключен в ручной режим по дискретной команде на входе $C_{руч}$. Если сигнал $C_{руч}=1$, то алгоритм переключается в ручной режим и его перевод в автоматический режим блокируется. После того, как сигнал $C_{руч}$ снимается (т.е. $C_{руч}=0$) алгоритм остается в ручном режиме, но может быть вручную переведен в автоматический режим.

Помимо основного выхода Y алгоритм содержит два дополнительных дискретных выхода. Сигнал $D_{руч}=1$, в если переключатель П2 находится в положении "РУ" (ручное управление), в противном случае $D_{руч}=0$. Сигнал $D_{руч}=1$, если переключатель П1 находится в положении "ДУ" (дистанционное управление), в противном случае $D_{дст}=0$.

Режимы работы

Алгоритм РУЧ инициирует команду отключения. В ручном режиме эта команда вместе с значением начальных условий, равным текущему значению сигнала Y , через вход X и $X_{дст}$ транслируется предвключенным алгоритмам. Если установлен автоматический локальный или автоматический каскадный режим, команда отключения формируется на входе $X_{дст}$. Если установлен автоматический режим, команда отключения формируется на входе X . В любом случае значение начальных условий равно текущему значению выходного сигнала Y .

В автоматическом режиме узел ручного управления при аналоговом регуляторе отслеживает текущее значение выходного сигнала Y , а при импульсном регуляторе обнуляется. Благодаря этому переход на ручное управление как для аналогового, так и для импульсного регулятора выполняется безударно.

Алгоритм может быть отключен внешней командой отключения, поступившей на каскадный выход Y . Независимо от установленных режимов управления эта команда вместе с значением начальных условий Y_0 через входы X и $X_{дст}$ транслируется предвключенным алгоритмам. Кроме того, в ручном режиме узел ручного управления при аналоговом регуляторе отслеживает значение начальных условий Y_0 .

Алгоритм не реагирует на команду запрета, поступившую на выход Y , но транслирует эту команду через вход X (в режимах «КУ», "ЛУ*") или вход $X_{дст}$ (в режиме «ДУ») предвключенным алгоритмам.

Входы-выходы алгоритма РУЧ

Таблица 21

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{руч}$	Входа	Переход на ручной режим
02	X		Сигнал локального или каскадного регулятора
03	$X_{дст}$		Сигнал дистанционного управления
01	Y	Выходы	Основной выход (каскадный)
02	$D_{руч}$		Ручной режим
03	$D_{дст}$		Дистанционный режим

Функциональная схема алгоритма
"Задание локальное ЗДЛ"

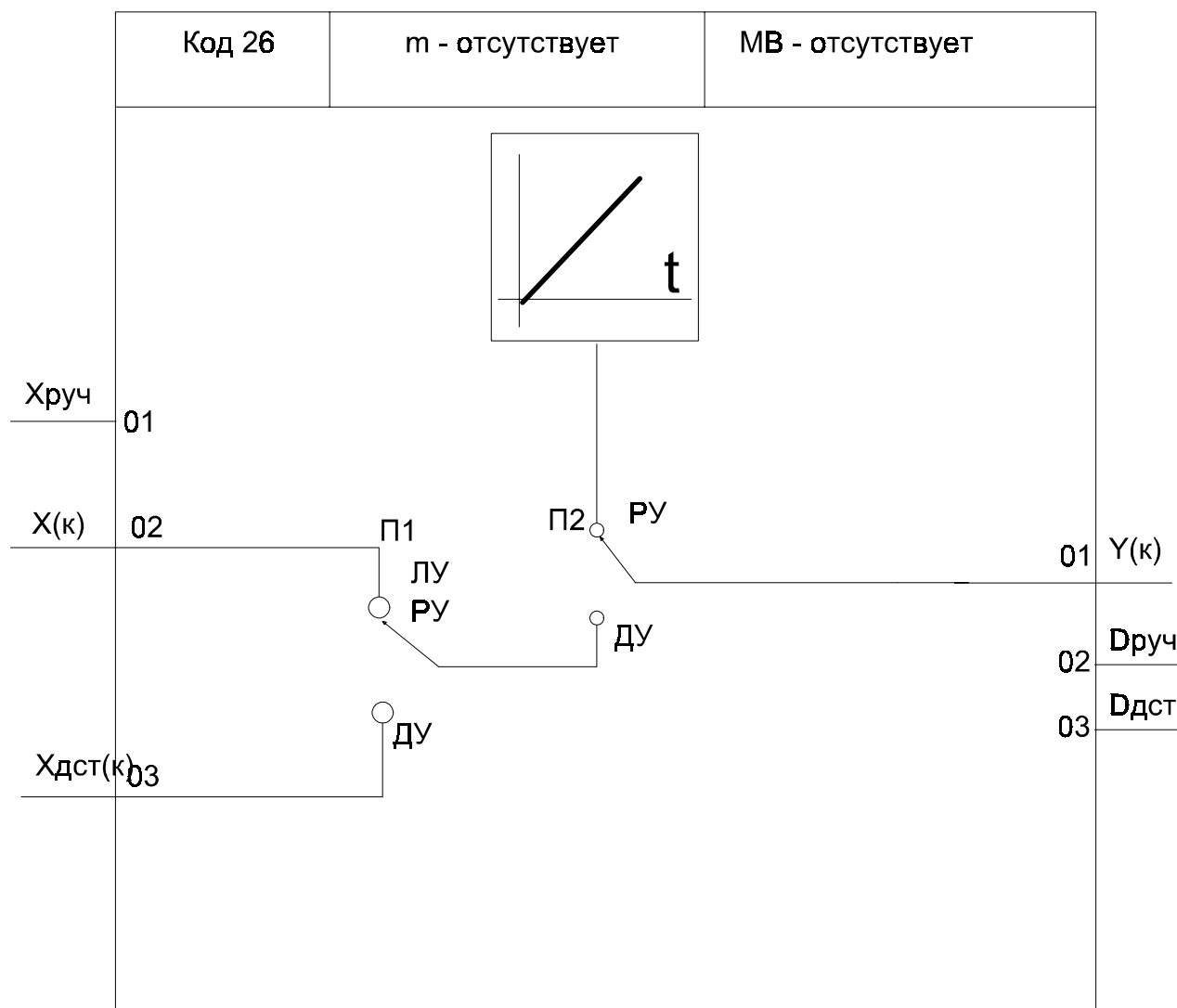


Рис. 16

4.6. ПРЗ(27) – Программный задатчик

Назначение

Программный задатчик формирует кусочно-линейную функцию времени, состоящую из нескольких (до 47) отрезков. Для каждого отрезка задается его продолжительность во времени и конечная ордината. Предусмотрена возможность пускать, останавливать и сбрасывать программу. Программа может выполняться заданное число раз.

Описание алгоритма

В состоянии сброса сигнал Y на выходе алгоритма имеет начальное значение $Y_0 = X_0$. После пуска сигнал Y начинает изменяться в соответствии с заданной программой. Параметры X_i и T_i на настроечных входах задают соответственно конечные ординаты и продолжительность отдельных участков программы (рис. 19).

Если выход Y соединен с одним из входов "ПРЗ" алгоритма задания ЗДН, то пускать, останавливать и сбрасывать программу можно с помощью клавиш лицевой панели. Кроме того, независимо от связи с алгоритмом ЗДН, алгоритм ПРЗ переводится в состояние "пуск", «стоп», «оброс» с помощью дискретных сигналов соответственно C_p , $C_{ст}$ и $C_{сбр}$, поступающих на вход алгоритма. При этом команды «пуск» и «стоп» действуют по переднему фронту (т.е. при переходе сигнала из состояния лог. 0 в лог. 1). Сигнал «сброс» является приоритетным, то есть при наличии лог. 1 на входе «сброс» алгоритм переходит в состояние «сброс» и не может быть переведен в другие состояния ни с помощью лицевой панели, ни с помощью дискретных команд на входе алгоритма. Однако, если алгоритм был переведен в состояние «сброс» с помощью лицевой панели, он может быть пущен дискретной командой «пуск» на входе алгоритма. Если команды «пуск» и «стоп» пришли на вход алгоритма одновременно, то выполняется команда «стоп».

После выполнения последнего участка программа переходит в состояние «конец программы» (КП), при этом выходной сигнал Y замораживается. Если задано количество повторов $N_{пвт}=1$, программа выполняется один раз, после чего переходит в состояние КП. При $N_{пвт}>1$ программа, дойдя до конца, автоматически переходит в начало, оставаясь в состоянии пуска до тех пор, пока не закончится заданное число повторений. После каждого окончания программы дискретный сигнал $D_{кпв}$ (конец повторения) на выходе алгоритма на один цикл переходит в состояние лог.1 и вновь возвращается в состояние лог.0. После того, как все повторения будут выполнены, программа переходит в состояние «конец программы» (КП), при этом сигнал $D_{кпв}$ становится равным лог.1 и остается в этом состоянии до тех пор, пока программа не будет сброшена. Кроме того, в этом состоянии сигнал $D_{кп}$ (конец программы) также становится равным лог.1 (рис. 18).

Остальные выходы имеют следующее назначение. $N_{уч}$ - текущий номер участка; $T_{ост}$ - время, оставшееся до окончания текущего участка; $N_{ост}$ - оставшееся число повторений программы; D_p , $D_{ст}$, $D_{сбр}$ - дискретные сигналы, индицирующие текущее состояние программы (соответственно «пуск», «стоп» и «сброс»).

Если входной сигнал $C_p=1$ и снимается сигнал сброса (т.е. $C_{сбр}$ переходит из состояния $C_{сбр}=1$ в состояние $C_{сбр}=0$), то алгоритм пускается (т.е. эта ситуация рассматривается как приход переднего фронта сигнала C_p). Если $C_p=1$ и снимается сигнал останова (т.е. $C_{ст}$ переходит из состояния $C_{ст}=1$ в $C_{ст}=0$) алгоритм остается в состоянии «стоп».

Входы-выходы алгоритма ПРЗ

Таблица 22

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_p	Входы	Пуск задатчика
02	$C_{ст}$		Останов задатчика
03	$C_{сбр}$		Сброс задатчика
04	$N_{пвт}$		Число повторений программы
05	X_0		Начальное значение задания
06	T_1		Длительность 1-го участка
07	X_1		Ордината 1-го участка
08	T_2		Длительность 2-го участка
09	X_2		Ордината 2-го участка
.	.		.
.	.		.
.	.		.
2m+4	T_m		Длительность m-го участка
2m+5	X_m		Ордината m-го участка
01	Y	Выходы	Основной выход задатчика
02	$N_{уч}$		Номер текущего участка
03	$T_{ост}$		Время, оставшееся до конца текущего участка
04	$D_{кпв}$		Конец текущего повторения программы
05	$N_{ост}$		Оставшееся число повторений

06	$D_{\text{п}}$	Программа в состоянии "пуск"
07	$D_{\text{ст}}$	Программа в состоянии "стоп"
08	$D_{\text{сбр}}$	Программа в состоянии "сброс"
09	$D_{\text{кп}}$	Конец программы

Сигнал, формируемый программным задатчиком

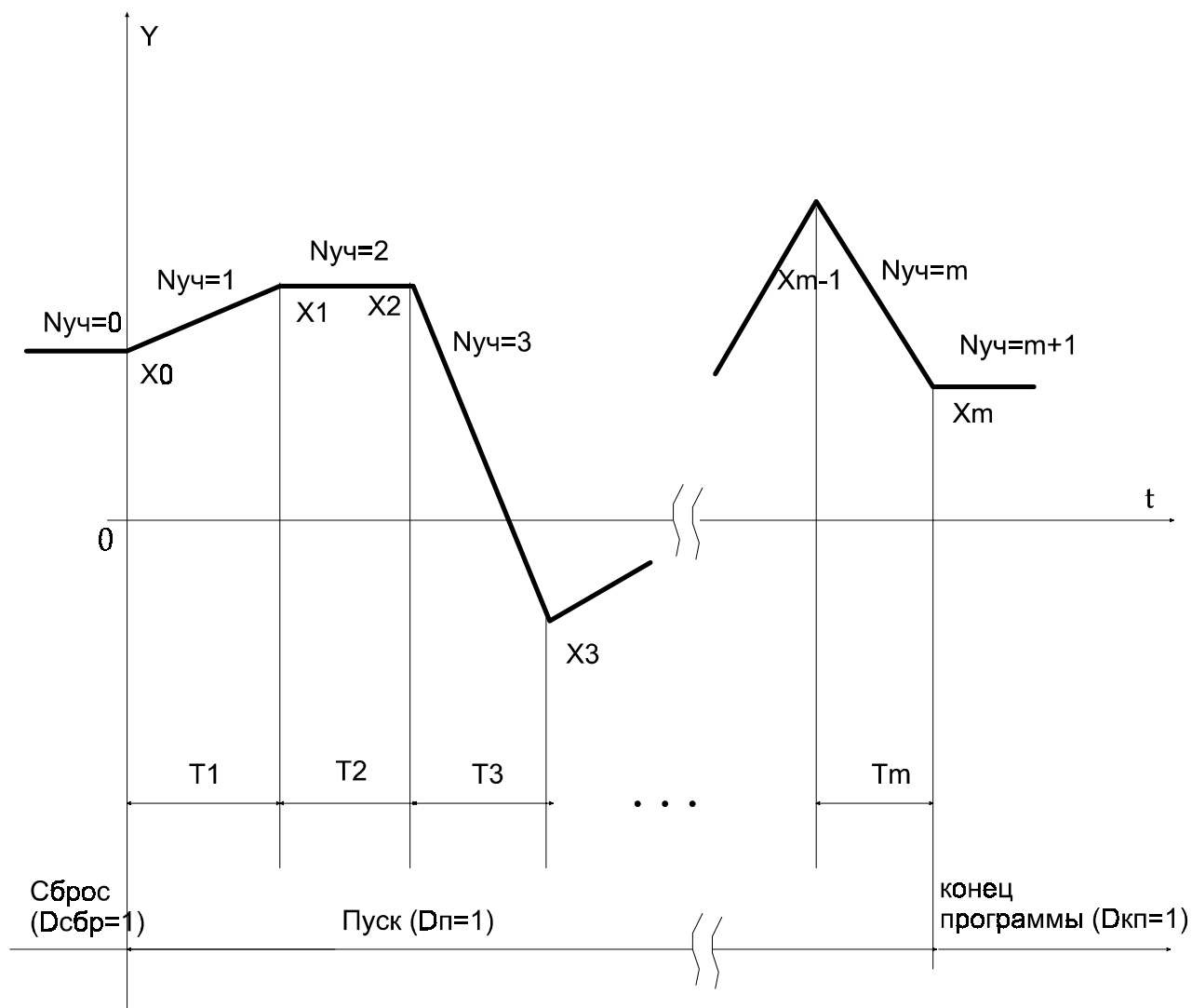


Рис. 18

Функциональная схема алгоритма
"Программный задатчик ПРЗ"

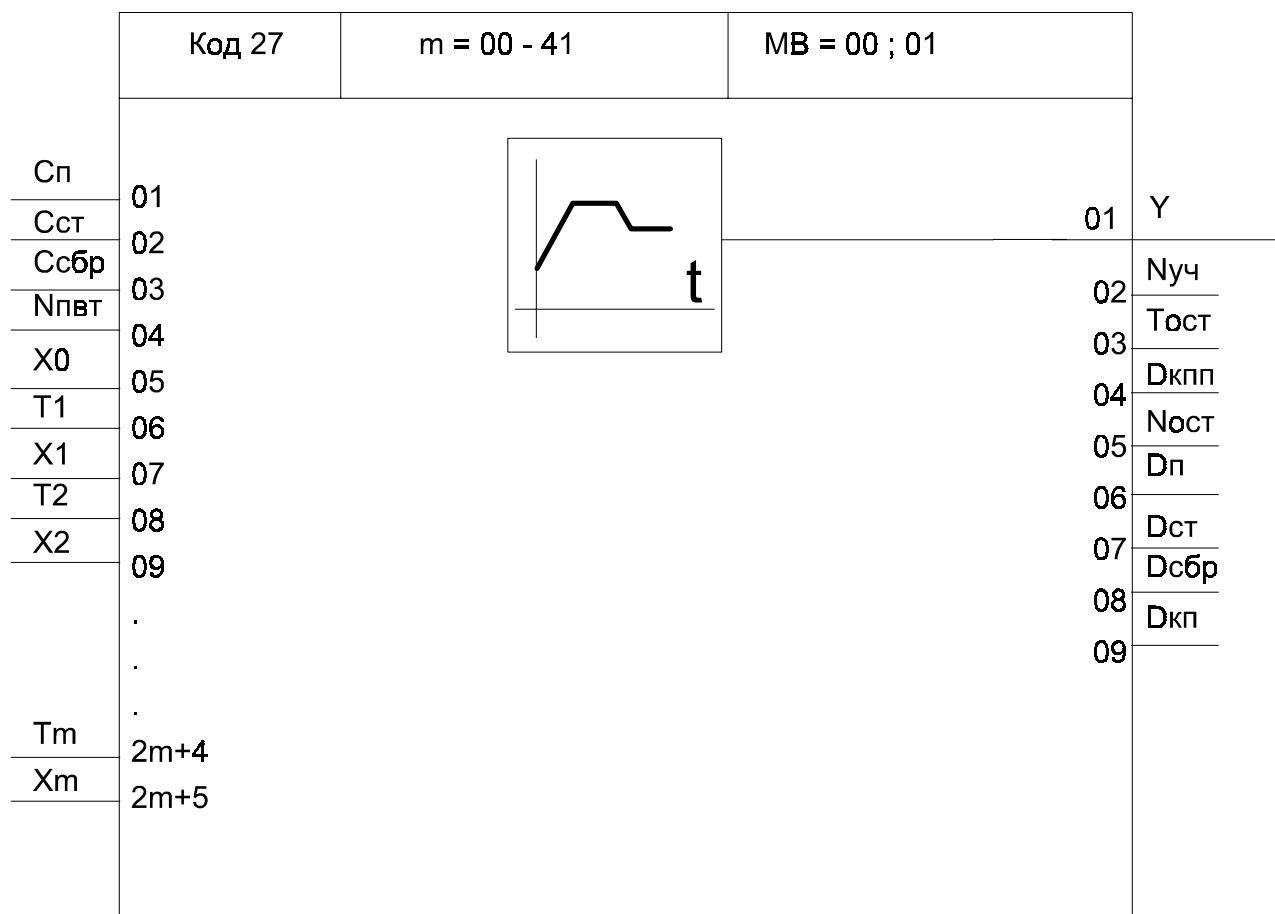


Рис. 19

4.7. ИНЗ(28) - Интегрирующий задатчик

Назначение

Алгоритм используется для дистанционного изменения задания по принципу «больше»-"меньше".

Описание алгоритма

Алгоритм имеет два дискретных входа C_6 , C_m , сигналы на которых управляют изменением выходного сигнала Y . Если команда подается на вход C_6 , выходной сигнал увеличивается, если на вход C_m – уменьшается. При $C_6=C_m=0$ выходной сигнал запоминается (рис. 20).

Скорость изменения задания переменна. В первый момент после поступления команды она минимальна, затем увеличивается. Если команда прерывается и затем вновь подается, скорость опять становится минимальной после чего снова возрастает. Такое свойство позволяет получать минимальное приращение задания и в то же время при необходимости быстро изменить задание на большую величину.

Скорость изменения выходного сигнала определяется выражением:

$$V = V_0 \cdot 2^n, \text{ причем } V \leq V_m, \quad (18)$$

Где V_0 , V_m – начальная (минимальная) и максимальная скорости, задаваемые при настройке алгоритма; показатель степени n определяется из таблицы (в ней t – время, прошедшее от момента поступления последней команды):

t	$0 \leq t \leq 2s$	$2s < t \leq 4s$	$4s < t \leq 6s$	
n	0	1	2	... и т. д.

Если команда C_6 и C_m действуют одновременно, это эквивалентно отсутствию обеих команд.

Режимы работы

Если на вход статической балансировки подан дискретный сигнал $C_{сб}=1$, то алгоритм становится следящим.

В этом случае выход Y алгоритма каскадный и если на этот выход поступает команда отключения, ячейка задания начинает отслеживать значение начальных условий Y_0 . Алгоритм при этом не реагирует на входные команды. Если на выход Y поступает команда запрета, изменение выходного сигнала в запрещенном направлении блокируется.

В этом случае, когда $C_{сб}=0$, алгоритм перестает быть следящим и всегда управляется командами, поступающими на входы C_6 , C_m .

Входы-выходы алгоритма ИНЗ

Таблица 23

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_6	Входы	Команда «больше»
02	C_m		Команда «меньше»
03	V_0		Минимальная скорость
04	V_m		Максимальная скорость
05	$C_{сб}$		Статическая балансировка
01	Y	Выходы	Выход задания (каскадный)

Функциональная схема алгоритма
"Интегрирующий задатчик ИНЗ"

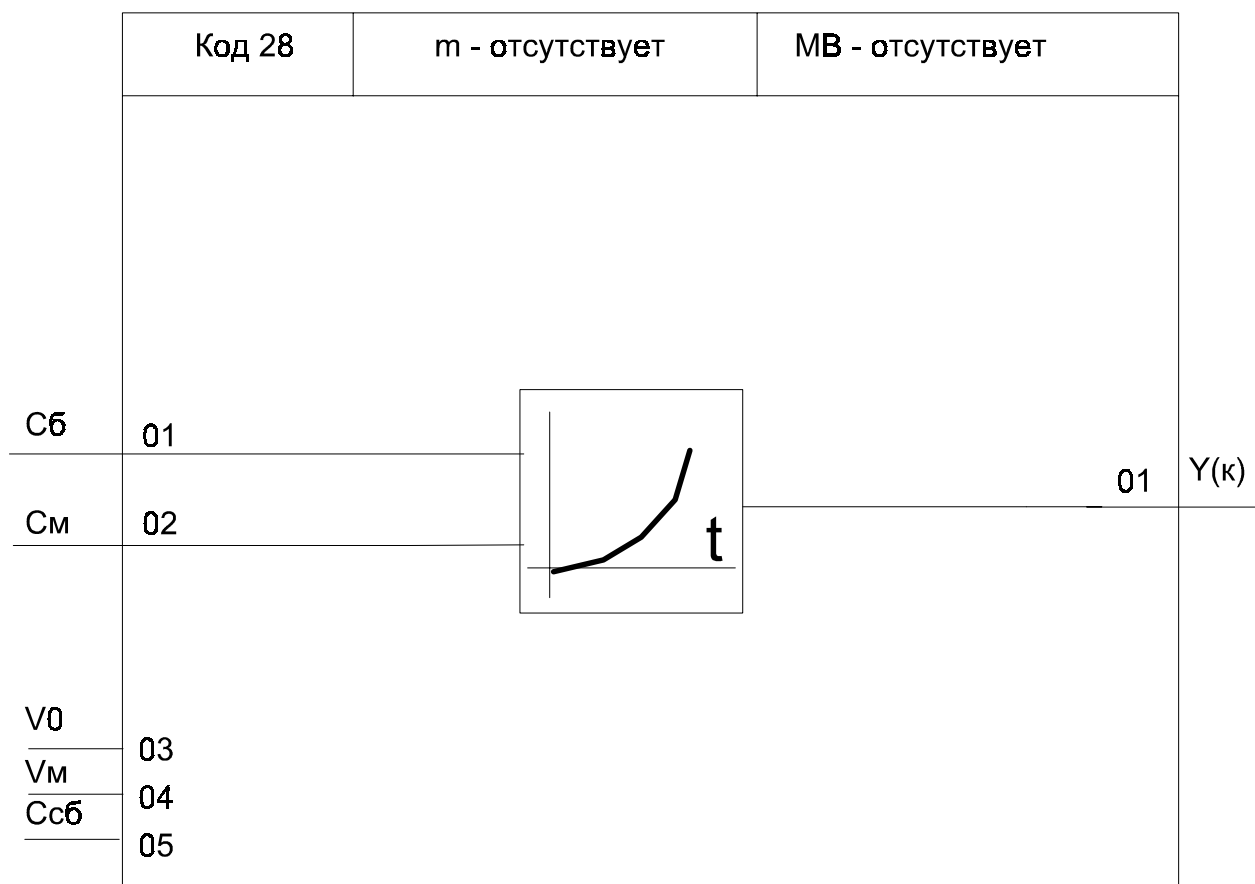


Рис. 20

4.8. ПОК(29) - Пороговый контроль

Назначение

Алгоритм контролирует несколько (до 20) аналоговых сигналов сравнивая каждый из них с двумя индивидуальными для каждого сигнала допустимыми значениями (нижними и верхними). Как правило, алгоритм используется вместе с алгоритмами ОКО или ОКЛ. В атом случае выход любого контролируемого сигнала за допустимые значения приводит к загоранию одного из индикаторов "ошибка контура" или «ошибка программы» на лицевой панели.

Описание алгоритма

Алгоритм содержит $0 \leq m \leq 20$ идентичных каналов, причем m задается модификатором (рис. 21).

В каждом канале входной сигнал X_i сравнивается с двумя уставками: верхней $X_{в,i}$ и нижней $X_{н,i}$. Если $X_{н,i} < X_i < X_{в,i}$, то дискретный сигнал на выходе канала равен логическому 0. В противном случае этот сигнал равен логической 1.

Если хотя бы один из контролируемых сигналов достигнет заданной для него уставки, выходной сигнал $D=1$, иначе $D=0$. На выходе N формируется номер выходного сигнала i , в котором сигнал $X_i \leq X_{н,i}$ или $X_i \geq X_{в,i}$.

В каждом канале предусмотрен фиксированный гистерезис, равный 0,2%.

Если одновременно несколько сигналов достигли уставки, N равно младшему номеру из этих сигналов.

Входы-выходы алгоритма ПОК

Таблица 24

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	1-й контролируемый сигнал
04	X_2		2-й контролируемый сигнал
.	.		.
.	.		.
.	.		.
$3m-2$	X_m		m -й контролируемый сигнал
02	$X_{в,1}$		Верхняя установка 1-го канала
03	$X_{н,1}$		Нижняя установка 1-го канала
05	$X_{в,2}$		Верхняя установка 2-го канала
06	$X_{н,2}$		Нижняя установка 2-го канала
.	.		.
.	.		.
.	.		.
$3m-1$	$X_{в,m}$		Верхняя установка m -го канала
$3m$	$X_{н,m}$		Нижняя установка m -го канала
01	N	Выходы	Номер входного сигнала, достигшего установки
02	D		Признак того, что один из входных сигналов достиг установки

Функциональная схема алгоритма
"Пороговый контроль ПОК"

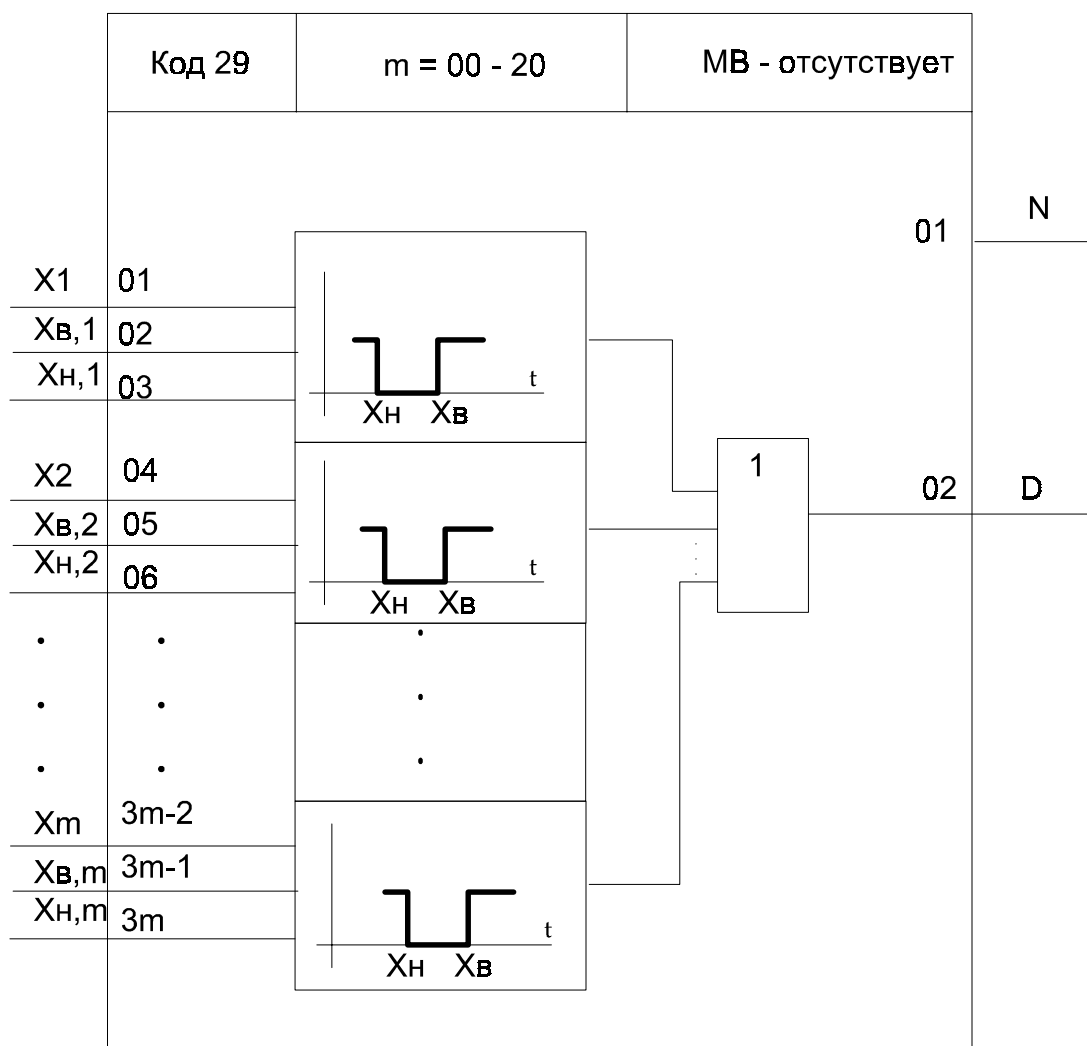


Рис. 21

4.9. АНР(ЗО) - Автонастройка регулятора

Назначение

Алгоритм используется для автоматизации расчета динамической настройки регулятора и применяется совместно с алгоритмами РАН и РИМ.

Описание функциональной схемы

Функциональная схема алгоритма АНР (рис, 22ж) содержит три узла: узел выделения сигнала рассогласования, текущих значений параметров настройки K_n и T_n ; узел анализа параметров колебаний, узел расчета параметров настройки.

Алгоритм содержит узел настройки, состоящий из переключателя режима "работа-настройка", нуля-органа и дополнительного фильтра с постоянной времени $T_{ф.1}$.

Свойства нуля-органа описываются выражением:

$$Y_{но} = X_{но} \text{ при } \epsilon \geq 0;$$

$$Y_{но} = -X_{но} \text{ при } \epsilon < 0;$$

где $Y_{но}$ - сигнал на выходе нуля-органа.

При дискретном сигнале на входе $C_{нас}=1$ алгоритм переходит в режим настройки и в замкнутом контуре регулирования устанавливаются автоколебания.

Параметры этих колебаний (амплитуда и период), которые контролируются на выходе используются для определения параметров настройки регулятора (методика настройки приведена в описании алгоритма автонастройки АНР). Для перехода в режим работы устанавливается сигнал $C_{нас}=0$.

Вход $X_{вх}$ алгоритма подключается к основному выходу алгоритма РАН (РИМ). Используя эту конфигурацию, алгоритм АНР выделяет сигнал рассогласования и текущие значения параметров настройки K_n и T_n , установленные в алгоритме РАН (РИМ), а также значение $X_{но}$, установленное в этом алгоритме.

Выделенный сигнал ϵ поступает на вход узла, анализирующего параметры колебаний. Анализ начинается, когда на вход $C_{пс}$ (пуск) подана дискретная команда $C_{пс}=1$.

На колебания может быть наложен шум, связанный с действием возмущений. Поэтому при анализе колебаний используется специальный помехозащищенный алгоритм, который на основе анализа нескольких периодов колебаний определяет усредненные значения амплитуды Y_k и периода T_k колебаний. Факт окончания анализа фиксируется на дискретном выходе кон. Пока анализ не закончен, кон = 0, если анализ окончен, кон = 1. Текущий номер периода колебаний для данного цикла анализа фиксируется на выходе $N_{тек}$.

В состоянии сброса ($C_{пс}=0$) сигналы $D_{кон}=0$, $N_{тек}=0$, а параметры колебаний Y_k и T_k остаются равными значениям, вычисленным в процессе предыдущего цикла анализа.

Результаты анализа параметров колебаний вместе с выделенными значениями текущих параметров настройки регулятора поступают на вход узла, рассчитывающего параметры настройки. На входе этого узла задаются также два настроечных коэффициента K_2 и K_3 (см. ниже).

На основании этих данных узел расчета вычисляет новые (расчетные) значения $K_{п,р}$ и $T_{п,р}$. Тот факт, что эти новые значения близки к оптимуму, фиксируется на дискретном выходе $D_{опт}$. В состоянии сброса ($C_{пс}=0$) значение $D_{опт}=0$, а величины $K_{п,р}$ и $T_{п,р}$ равны значениям, рассчитанным на предыдущем цикле.

Принципы настройки

При настройке контура регулирования используются следующие предпосылки:

- 1) Настройка основана на анализе автоколебаний в контуре регулирования.
- 2) Для установки автоколебаний алгоритм РАН (РИМ) переводится в режим настройки. При этом контур не должен быть отключен (например, не переведен на ручной режим) и при колебаниях в контуре не должны достигаться пороги ограничения.
- 3) Настройка выполняется для ПИ закона регулирования, при этом определяется два параметра настройки - коэффициент пропорциональности K_n и постоянная времени интегрирования T_n . Если предполагается использовать ПИД закон, то после определений K_n и T_n в алгоритме РАН (РИМ) устанавливается коэффициент дифференцирования $K_d = T_d/T_n = 0,1-0,3$, при этом значение K_n может быть увеличено на 20-40 %.
- 4) Настройка, выполняется итеративным способом: анализируются параметры колебаний для текущих параметров настройки, по ним определяется новые параметры, эти параметры устанавливаются в алгоритме РАН (РИМ), после чего определяются новые параметры настройки, - и так до тех пор, пока новые значения параметров K_n и T_n будут близки к текущим значениям. Обычно, требуется не более, чем 3-5 циклов итерации.

При настройке можно использовать одну из трех методик, предполагающих:

- 1) Точную оценку колебаний.
- 2) Автоматический анализ колебаний,
- 3) Автоматический расчет параметров настроек.

Ручная оценка колебаний.

При ручной оценке колебаний алгоритм АНР не требуется. Параметры колебаний оцениваются по выходу "Уе" алгоритма РАН (РИМ) с помощью пульта настройки ПН-1 и секундомера. Предварительно в этих алгоритмах устанавливается амплитуда нуля-органа $X_{но}$, настроечный коэффициент K_1 и параметры K_n и T_n (см. ниже). Значение $X_{но}$ выбирается таким, чтобы амплитуда колебаний заметно превышала уровень шумов, нало-

женных на колебания, и в то же время оставалась в допустимых (с точки зрения объекта) границах. Величины K_1 , K_n и T_n зависят от свойств объекта, при этом K_1 зависит, кроме того, от заданной степени затухания и определяется из графиков, приведенных на рис. 22 а. Если о шумах и об объекте ничего заранее неизвестно, можно воспользоваться установленными в алгоритмах РАН(РИМ) начальными значениями $X_{но}=1\%$; $K_1=0,3\%$; $K_n=1$; $T_n=15$ (секунды, минуты или часы, в зависимости от временного диапазона контроллера и выбранного в алгоритме РАН (РИМ) масштаба времени).

По замеренным значениям амплитуды Y_k и периода T_k колебаний определяются новые значения параметров настройки K_n и T_n :

$$\begin{aligned} K_n^{i+1} &= K_n^i * K_2 * X_{но} / Y_k; \\ T_n^{i+1} &= T_n^i / K_3; \end{aligned} \quad (18a)$$

где Y_k и K_1 – соответственно амплитуда и период автоколебаний, индексами $i+1$ и i обозначены соответственно новые и текущие параметры настройки, а зависящие от свойств объекта и заданной степени затухания настроечные коэффициенты K_2 и K_3 выбираются из графиков, приведенных на рис. 22а. Если исходная информация об объекте отсутствует, можно принять $K_2=0,92$ и $K_3=3,7$.

Автоматический анализ колебаний

Для автоматического анализа колебаний используется алгоритм АНР, вход $X_{вх}$ которого по конфигурации подключается к выходу Y алгоритма РАН (РИМ) (но не к выходу Y_E этих алгоритмов).

С помощью пульта настройки на вход $C_{пс}$ подается сигнал, равный логической 1. После этого начинается анализ колебаний. Он обычно завершается по прошествии 4-6 периодов, при этом выходной сигнал «конец анализа» устанавливается равным $C_{кон}=1$. Амплитуда и период колебаний с помощью пульта ПН-1 считываются с выходов соответственно Y_k и T_k . Полученные величины подставляются в формулу (18 а), определяются новые значения K_n и T_n , эти значения устанавливаются на входе алгоритма РАН (РИМ) и процедура повторяется. Перед повторением анализа в алгоритме АНР устанавливается $C_{пс}=0$ (т.е. алгоритм переводится в состояние «сброс») и затем вновь устанавливается $C_{пс}=1$ ("пуск").

Если по прошествии 8-10 периодов (текущий номер периода фиксируется на выходе N тек) анализ параметров колебаний не закончится, это означает, что уровень шумов превышает допустимый. В этом случае рекомендуется увеличить амплитуду колебаний, увеличив параметр $X_{но}$ на входе алгоритма РАН (РИМ).

Автоматический расчет параметров настройки

Для автоматического расчета параметров K_n и T_n на входе алгоритма АНР задаются два настроечных коэффициента K_2 и K_3 значения которых зависят от свойств объекта и степени затухания и определяются из графиков рис. 22 а. Если исходные данные об объекте отсутствуют, можно работать с начальными значениями $K_2=0,92$ и $K_3=3,7$, имеющимися в алгоритме АНР при его первом включении.

После пуска алгоритм АНР анализирует амплитуду и период колебаний и, используя эти параметры, рассчитывает новые значения параметров настройки, которые формируются на выходах алгоритма $K_{п,р}$ и $T_{п,р}$. Эти параметры устанавливаются в алгоритме РАН (РИМ), после чего в алгоритме АНР подается новая команда пуска и весь процесс повторяется. Новые значения действительны, когда на выходе $D_{кон}$ установится сигнал $D_{опт}=1$.

Если новые значения $K_{п,р}$ и $T_{п,р}$ будут мало отличаться от предыдущих значений, на выходе алгоритма $D_{опт}$ установится сигнал $D_{опт}=1$, что свидетельствует о достижении оптимальной настройки. В противном случае $D_{опт}=0$.

Выбор настроечных коэффициентов

При настройке регулятора используются три настроечных коэффициента K_1 , K_2 , K_3 , зависящих от свойств объекта управления и заданной степени затухания. Обычно точное значение параметров объекта неизвестно, однако ориентировочно можно оценить отношение запаздывания к постоянной времени объекта (объект близок к звену первого порядка с запаздыванием или к звену второго порядка с запаздыванием).

Для выбора настроечных коэффициентов используются графики, представленные на рис. 22 а (объект первого порядка с запаздыванием) и рис. 22 б (объект второго порядка с запаздыванием).

На этих графиках используются следующие обозначения: τ/T - отношение запаздывания к постоянной времени объекта, полученные из передаточной функции;

τ_p/T_p - те же величины, полученные из кривой разгона объекта.

Здесь предполагается, что для объекта первого порядка передаточная функция имеет вид:

$$W(p) = R * e^{-p\tau} / (Tp + 1), \quad (18 б)$$

а для объекта второго порядка

$$W(p) = K * e^{-p\tau} / (Tp + 1)^2 \quad (18 в)$$

Метод обработки кривой разгона показан на рис. 22 в. На графиках рис. 22 а, б используется параметр Ψ (коэффициент колебательности), равный отношению коэффициента усиления замкнутой системы "регулятор-объект" на резонансной частоте к этому коэффициенту на нулевой частоте (рис. 22 г). Параметр Ψ связан со степенью затухания Y зависимостью, показанной на рис. 22 е. Определение параметра Y показано на рис. 22 д.

Последовательность действий при настройке.

1. Выбирается один из трех вариантов настройки: ручная оценка параметров колебаний, автоматический анализ параметров колебаний, автоматический расчет параметров настройки.

2. Задается параметр u или величина затухания Ψ . Если задается Ψ , из графика рис. 22 е определяется u .
3. Оценивается порядок объекта (первый или второй) и отношение запаздывания к постоянной времени объекта. Если известна кривая разгона, она обрабатывается согласно рис. 22а.
4. Если автоматический расчет параметров настройки не используется, из графиков рис. 22 а (для объекта первого порядка) или рис. 22 б (для объекта второго порядка) определяются настроечные коэффициенты K_1 , K_2 , K_3 и коэффициент K_1 устанавливается на входе алгоритма РАН (РИМ). В этом же алгоритме устанавливается величина $X_{но}$ и алгоритм переводится в режим настройки. Параметры колебаний определяются ими вручную на выходе ε алгоритма РАН (РИМ), или автоматически на выходах Y_k и T_k алгоритма АНР.
5. Если предполагается использовать автоматический расчет параметров настройки, то в дополнение к подпункту 4 на входе алгоритма АНР устанавливаются коэффициенты K_2 , K_3 , а рассчитанные параметры настройки фиксируются на выходах $K_{п.р}$; $T_{и.р}$ алгоритма АНР.
6. Если исходные данные об объекте полностью отсутствуют, коэффициенты K_1 - K_3 можно не определять, а воспользоваться их исходными значениями, заранее установленными в алгоритмах РАН (РИМ) и АНР.

Методика настройки регулятора, основанная на анализе автоколебаний, заимствована из книги «Автоматизация настройки систем управления» под редакцией В.Я. Ротача, Москва, Энергоавтомиздат 1984 г.

Таблица 25

Входы-выходы алгоритма АНР

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$X_{вх}$	Входы	Входной сигнал
02	$C_{пс}$		Команд пуска
03	Y_k		2-й настроечный коэффициент
04	K_3		3-й настроечный коэффициент
01	$D_{кон}$	Выходы	Признак конца анализа
02	$N_{тек}$		Текущий номер периода колебаний
03	Y_k		Амплитуда колебаний
04	T_k		Период колебаний
05	$D_{опт}$		Признак достижения оптимума
06	$K_{п.р}$		Расчитанное значение коэффициента пропорциональности
07	$T_{и.р}$		Расчитанное значение постоянной времени интегрирования

Зависимость коэффициентов
 k_1, k_2, k_3 от степени затухания

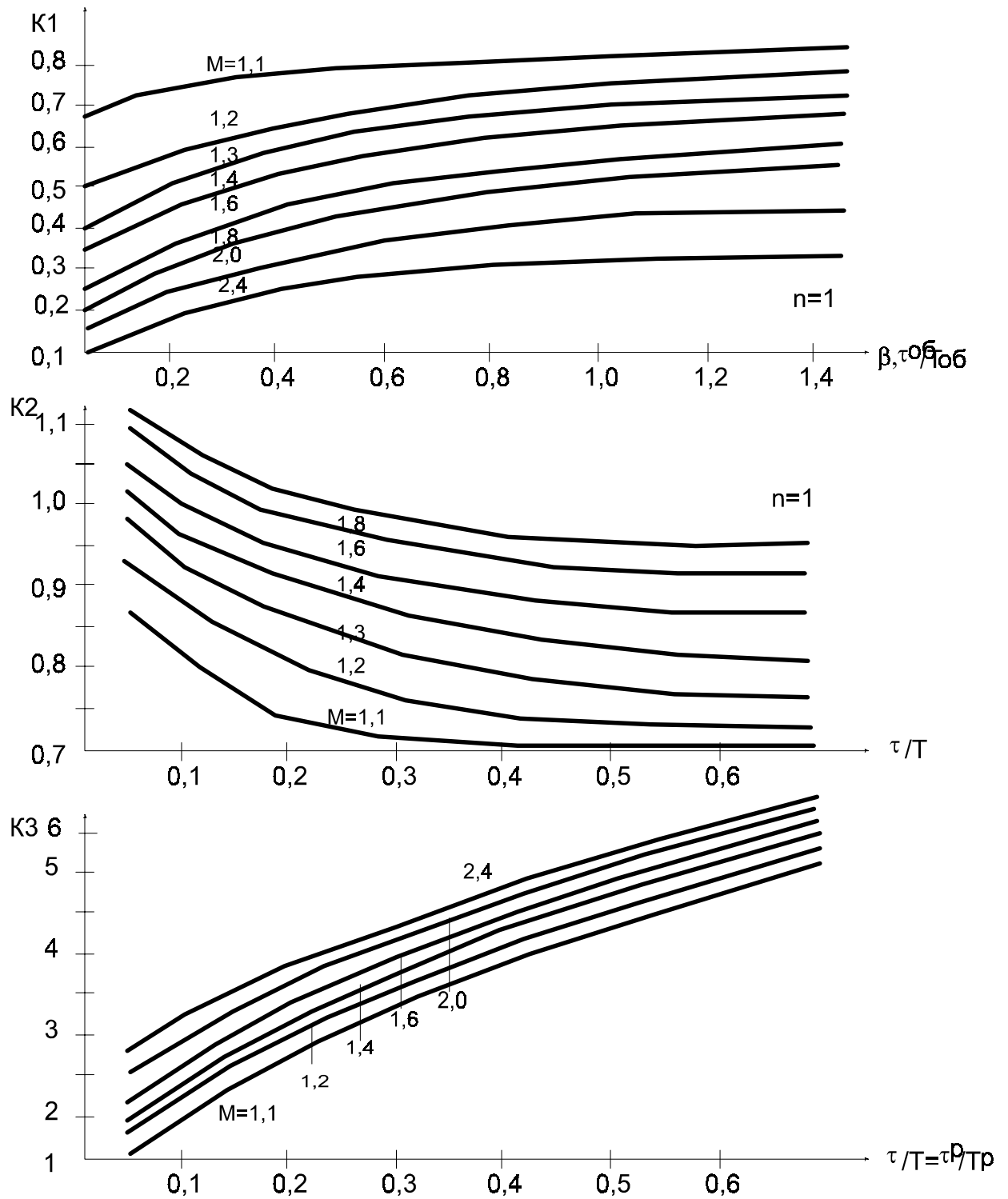


Рис. 22а

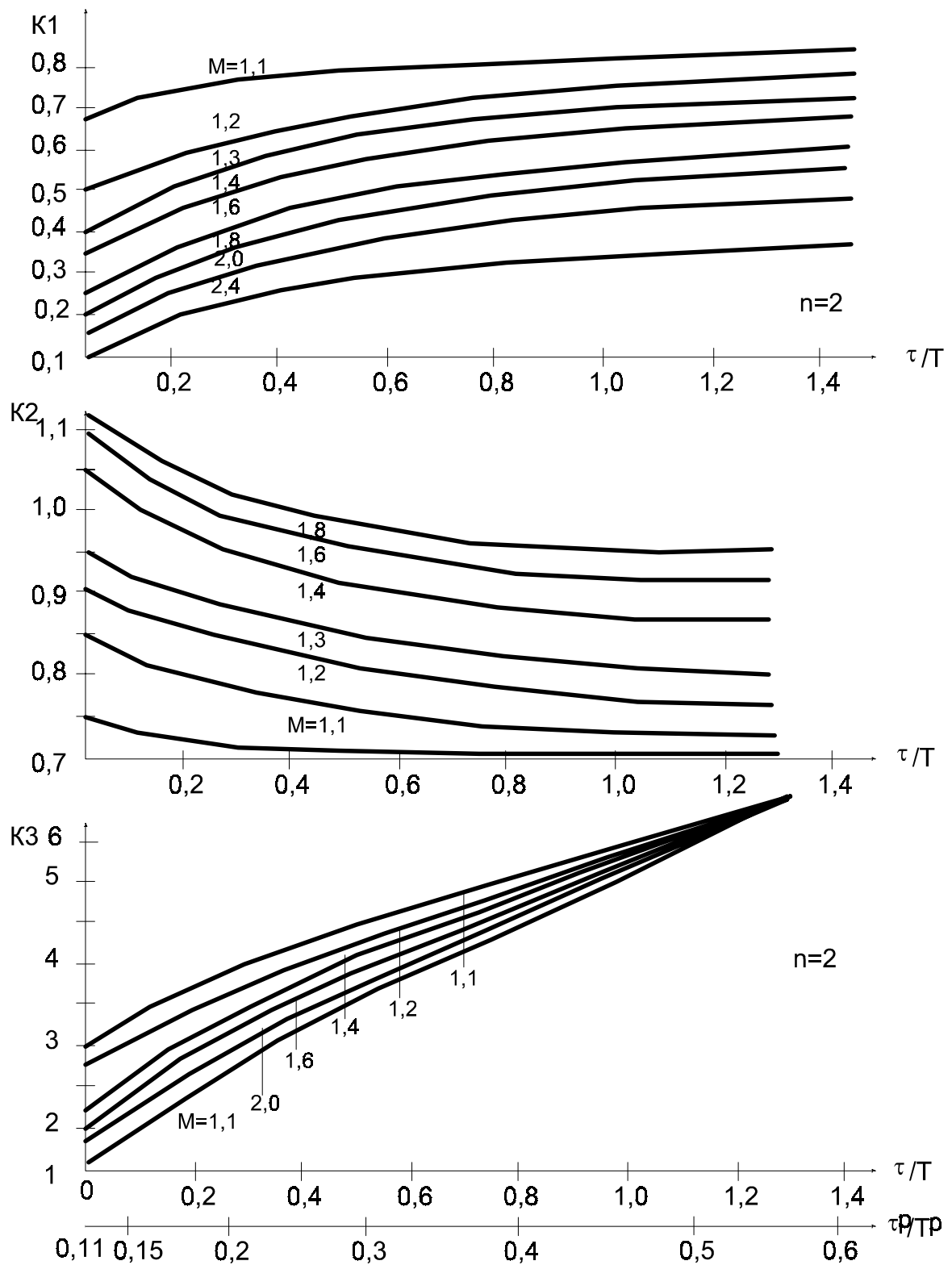


Рис. 226

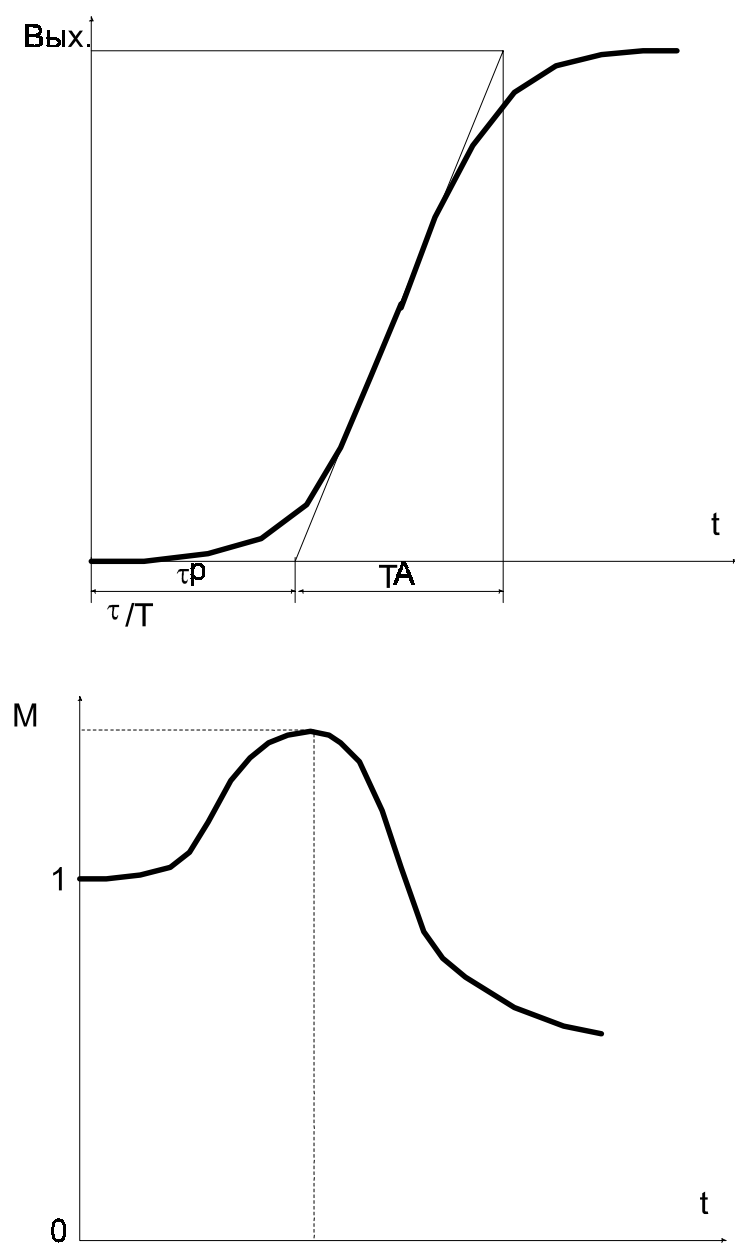


Рис. 22г

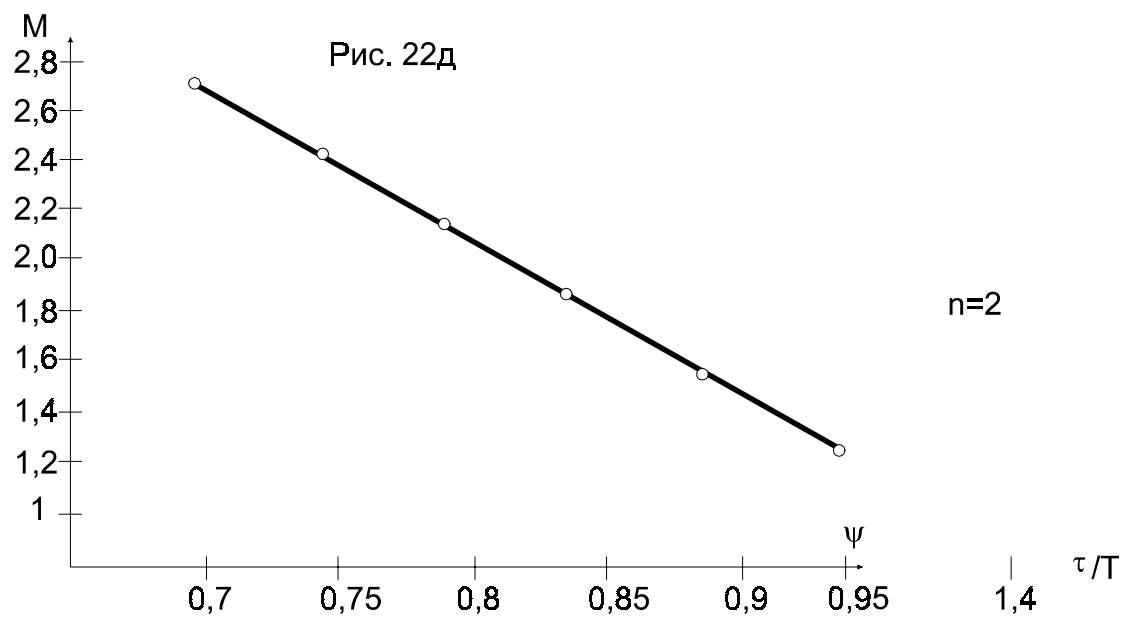
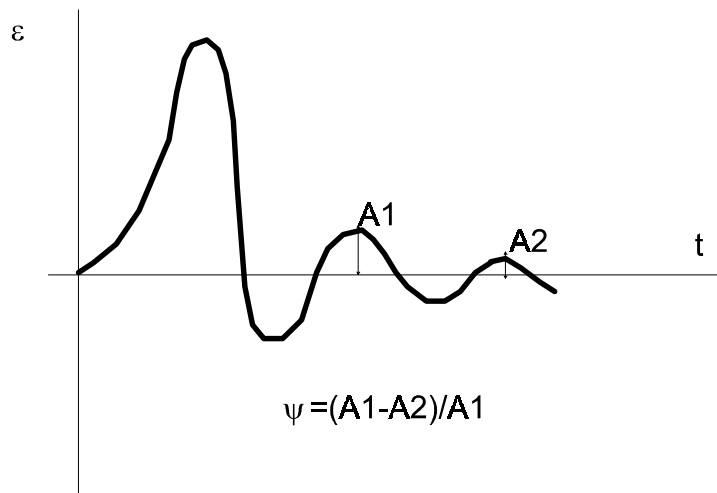


Рис. 22е

Функциональная схема алгоритма
"Автонастройка регулятора АНР"

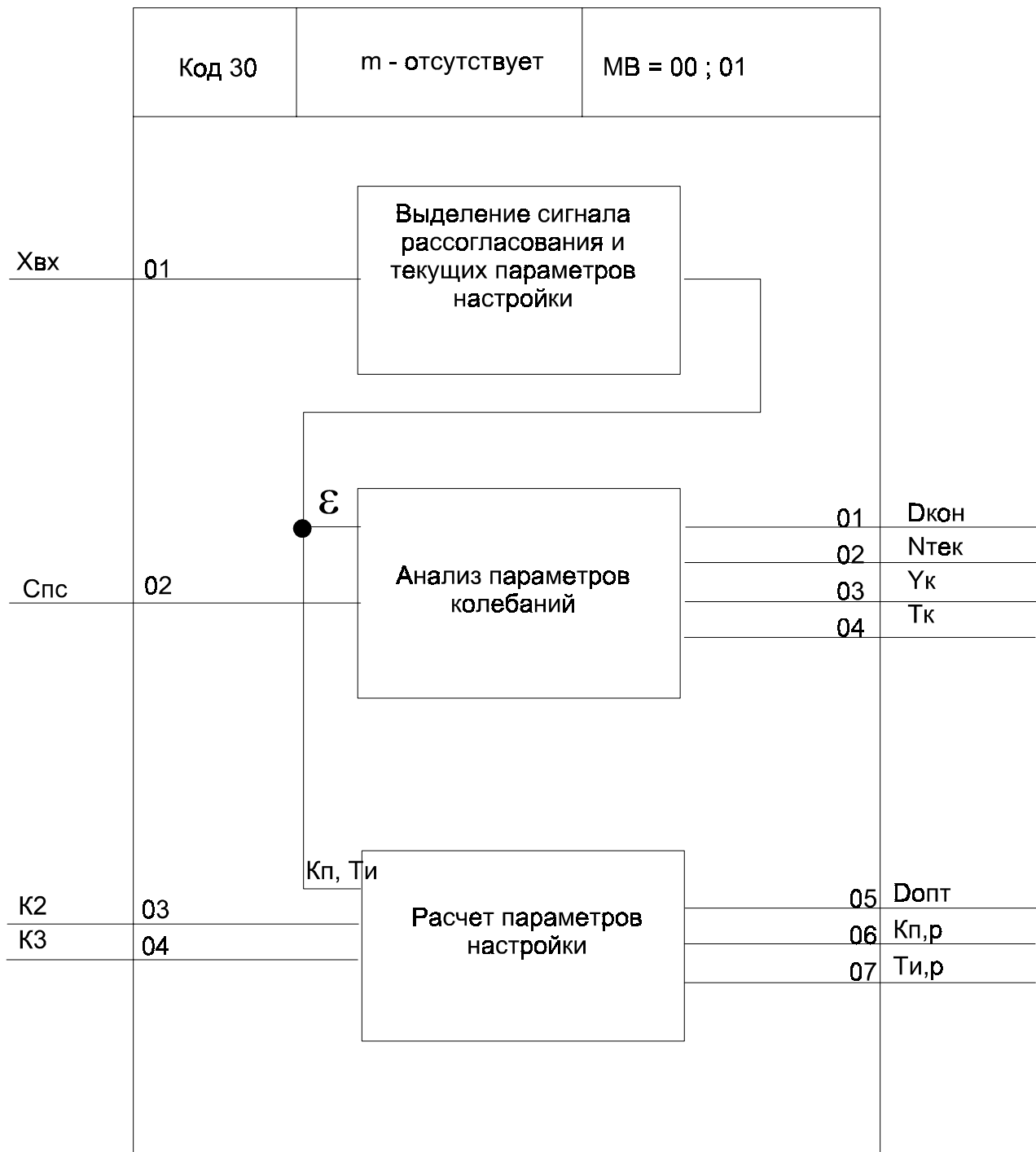


Рис. 22ж

5. Динамические преобразования

5.1. ИНТ(33) - Интегрирование

Назначение

Алгоритм используется для интегрирования и (или) запоминания сигнала.

Описание алгоритма

Передаточная функция алгоритма равна (рис. 23):

$$W(p)=Y(p)/X(p)=1/T_{\text{и}}p, \quad (19)$$

где $T_{\text{и}}$ - постоянная времени интегрирования.

Алгоритм содержит пороговый элемент. Если выходной сигнал $Y < X_{\text{пор}}$, где $X_{\text{пор}}$ - уставка порогового элемента $D=0$. Если $Y \geq X_{\text{пор}}$, то $D=1$.

В алгоритме предусматривается установка начальных условий. Если дискретный сигнал на входе $C_{\text{нач}}=0$, то выполняется основная функция интегратора. При $C_{\text{нач}}=1$ интегратор переходит в режим установки начальных условий, при этом выход $Y=X_{\text{нач}}$. В частном случае при $X_{\text{нач}}=0$ интегратор обнуляется.

Режимы работы

Алгоритм относится к группе следящих.

Вход X и выход Y алгоритма каскадные.

Если на выход Y поступает команда отключения с значением начальных условий Y_0 , интегральная ячейка отслеживает это значение, при этом логический элемент продолжает нормально работать. Через каскадный вход X команда отключения и значение обратного счета $X_0=0$ транслируется предвключенному алгоритму.

Команда запрета, поступившая на выход Y , запрещает изменение выходного сигнала в направлении увеличения (запрет «больше») или уменьшения (запрет «меньше») и одновременно транслируется через каскадный вход X предвключенному алгоритму.

В режиме установки начальных условий ($C_{\text{нач}}=1$) алгоритм не реагирует на команды отключения и запрета, поступающие на выход алгоритма.

Входы-выходы алгоритма ИНТ

Таблица 26

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	$T_{\text{и}}$		Постоянная времени интегрирования
03	$C_{\text{нач}}$		Команда установки начальных условий
04	$X_{\text{нач}}$		Значение начальных условий
05	$X_{\text{пор}}$		Установка порогового элемента
01	Y	Выходы	Основной выход (каскадный)
02	D		Выход порогового элемента

Функциональная схема алгоритма
"Интегрирование ИНТ"

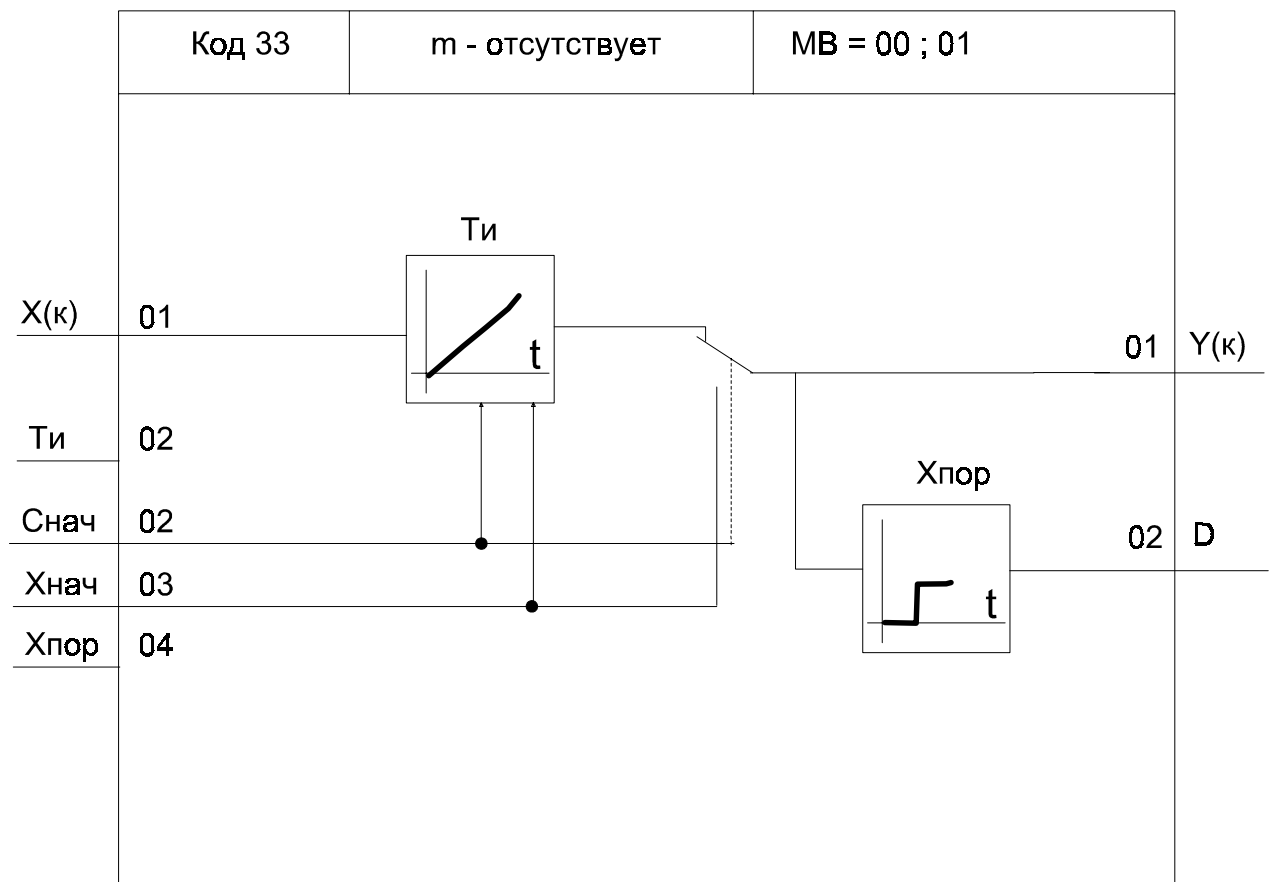


Рис. 23

5.2 ДИФ(34) – Дифференцирование

Назначение

Алгоритм применяется в схемах динамической коррекции для получения сигналов, связанных со скоростью изменения параметра.

Функциональное описание

Алгоритм представляет собой реальное дифференцирующее звено с регулируемым коэффициентом усиления и постоянной времени дифференцирования (рис. 24).

Передаточная функция алгоритма имеет вид:

$$W(p) = Y(p)/X(p) = K_m * T_d * p / (T_d * p + 1), \quad (20)$$

где K_m - масштабный коэффициент (коэффициент усиления); T_d – постоянная времени дифференцирования.

Если на вход C_0 подается команда обнуления $C_0=1$, то выходной сигнал $Y=0$ независимо от входного сигнала.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ДИФ

Таблица 27

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход
02	K_m		Коэффициент усиления
03	T_d		Постоянная времени дифференцирования
04	C_0		Обнуление
01	Y	Выход	Выход

Функциональная схема алгоритма
"Дифференцирование ДИФ"

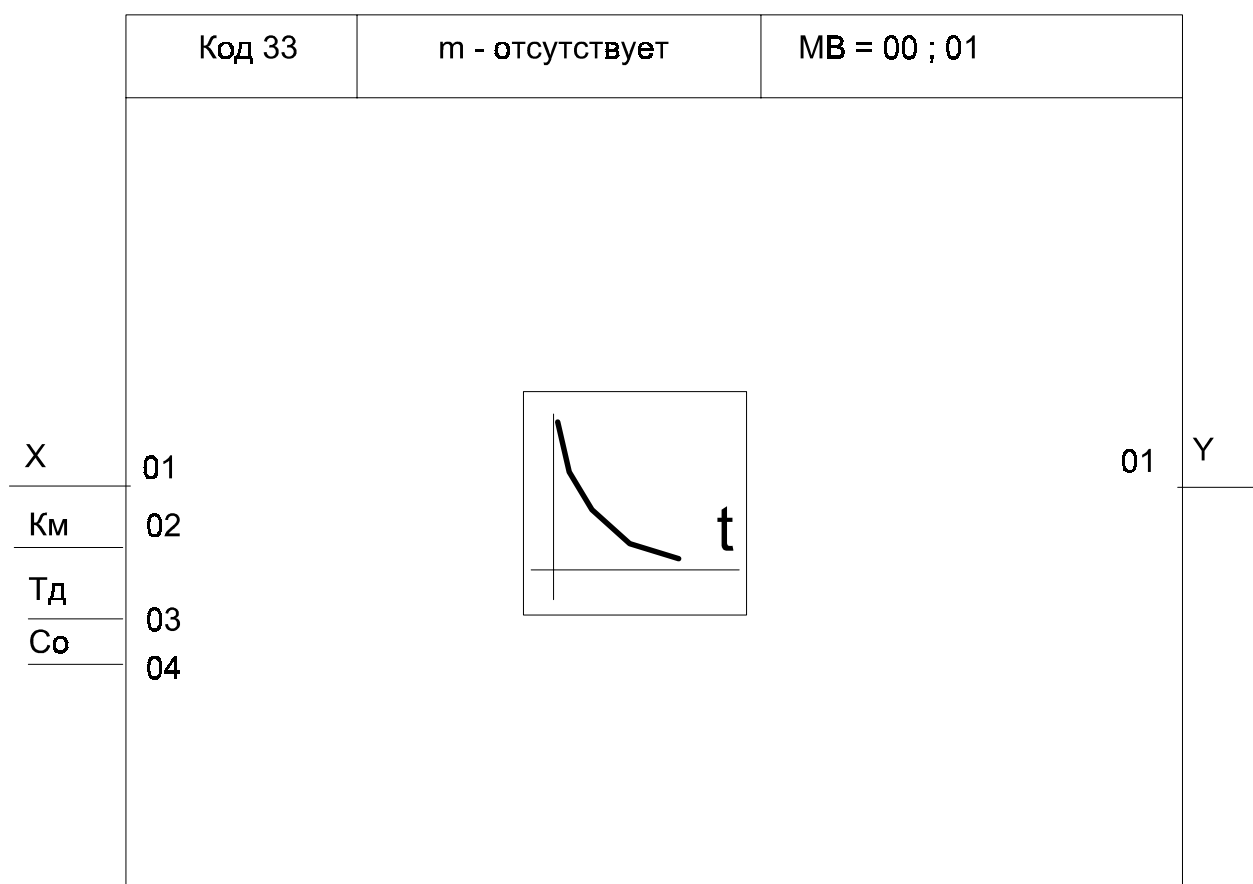


Рис. 24

5.3. ФИЛ(35) - Фильтр

Назначение

Алгоритм используется для фильтрации высокочастотных помех, а также для динамической коррекции. Фильтр, имеющий порядок выше первого, можно получить путем последовательного включения нескольких алгоритмов ФИЛ.

Описание алгоритма

Алгоритм является фильтром нижних частот первого порядка (рис. 25). Его передаточная функция:

$$W(p)=Y(p)/X(p)=1/(T_{\phi}p+1), \quad (21)$$

где T_{ϕ} - постоянная времени фильтра.

Режимы работы

Алгоритм не инициализирует обратный счет, но если по инициативе других алгоритмов на каскадный вход Y поступает команда отключения со значением начальных условий Y_0 или команда запрета. Эти сигналы через каскадный вход X транслируются предвключенному алгоритму. Ячейка фильтра при отключении заряжается до значения, а на команду запрета эта ячейка не реагирует.

Входы-выходы алгоритма ФИЛ

Таблица 28

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	T_{ϕ}		Постоянная времени фильтра
01	Y	Выход	Выход (каскадный)

Функциональная схема алгоритма
"Фильтр ФИЛ"

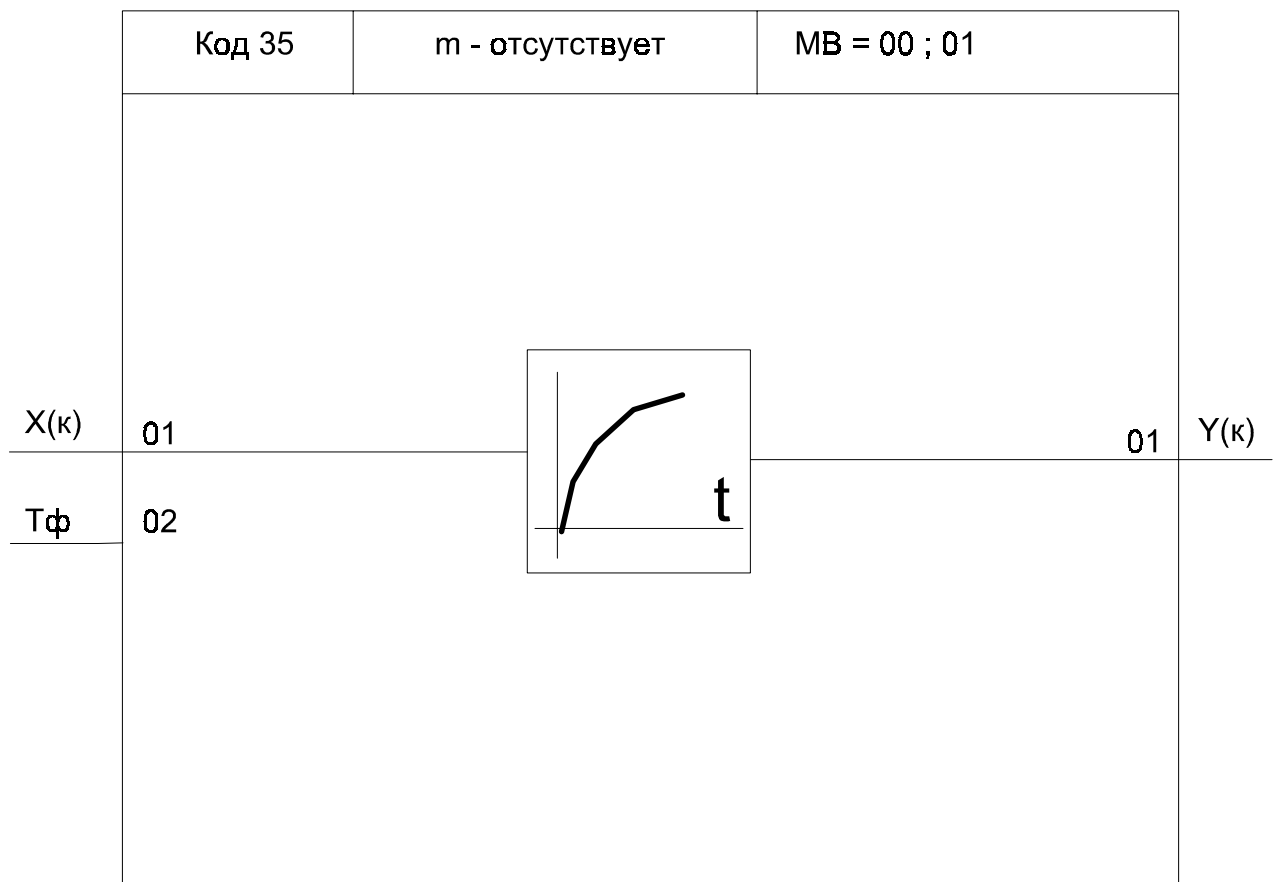


Рис. 25

5 4. ДИН(36) - Динамическое преобразование

Назначение

Алгоритм применяется для динамической коррекции систем управления в тех случаях, когда требуется интегро-дифференцирующее преобразование сигнала.

Описание алгоритма

Алгоритм реализует передаточную функцию (рис. 26):

$$W(p)=Y(p)/X(p)=K_m*(T_2*p+1)/(T_1*p+1), \quad (22)$$

где K_m –коэффициент усиления; T_1 и T_2 - постоянные времени.

Режимы работы

Алгоритм не инициализирует обратный счет, но если по инициативе других алгоритмов на каскадный вход Y поступает команда отключения со значением начальных условий Y_0 , алгоритм выполняет процедуру обратного счета, формируя на каскаде входе X сигнал $X_0=Y_0/K_m$. Этот сигнал вместе с командой отключения транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, но эта команда через вход X транслируется предвключенному алгоритму.

Входы-выходы алгоритма ДИН

Таблица 29

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	K_m		Коэффициент усиления
03	T_1		Постоянная времени знаменателя
04	T_2		Постоянная времени числителя
01	Y	Выходы	Выход (каскадный)

Функциональная схема алгоритма
"Интегрирующий задатчик ИНЗ"

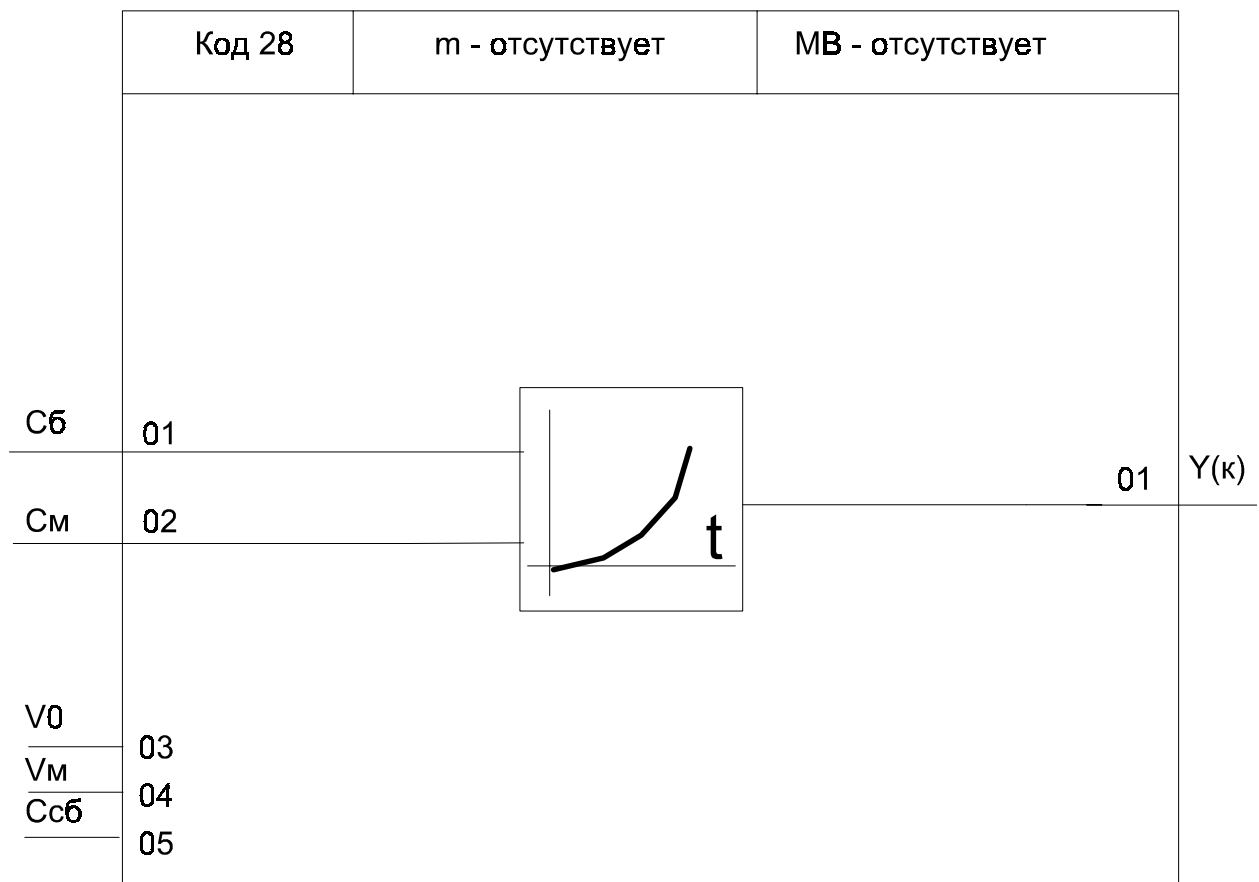


Рис. 20

5.5. ДИБ(37) - Динамическая балансировка

Назначение

Алгоритм применяется для динамической балансировки отключенной цепи. Балансировка обеспечивает безударное включение цепи за счет выработки балансирующего сигнала, который после включения плавно уменьшается до нуля.

При необходимости балансировки задания применять алгоритм ДИБ не требуется, так как механизм динамической балансировки встроен в алгоритм задания ЗДН.

Описание алгоритма

Алгоритм содержит узел динамической балансировки и сумматор (рис. 27). Во включенном состоянии выходной сигнал алгоритма

$$Y = X + Y_6, \quad (23)$$

где X - входной сигнал; Y_6 - выходной сигнал узла балансировки.

Если алгоритм включен и $Y_6 > 0$ то узел балансировки «разряжается» до нуля с постоянной настраиваемой скоростью V_6 . В установившемся режиме $Y_6 = 0$ и $Y = X$.

Режимы работы

Алгоритм ДИБ относится к группе следящих, т. е. он инициализирует процедуру обратного счета в отключенной цепи.

Если на каскадный выход Y алгоритма поступает команда отключения со значением начальных условий Y_0 , звено балансировки "заряжается" до значения

$$Y_6 = Y_0 - X$$

Благодаря этому в первый момент после включения выходной сигнал алгоритма $Y = Y_0$ (т.е. включение выполняется безударно).

Команда отключения через входы алгоритма не транслируется (т.е. алгоритм прерывает распространение команды отключения по цепочке последовательно включенных алгоритмов).

Команда запрета, поступающего на каскадный выход Y не влияет на работу алгоритма, но транслируется через вход X предвключенному алгоритму,

Входы-выходы алгоритма ДИБ

Таблица 30

Входа-выхода			Назначение
№	Обозн	Вид	
01	X	Входы	Вход
02	V_6		Скорость балансировки
01	Y	Выходы	Основной выход (каскадный)
02	Y_6		Выход узла балансировки

Функциональная схема алгоритма
"Динамическая балансировка ДИБ"

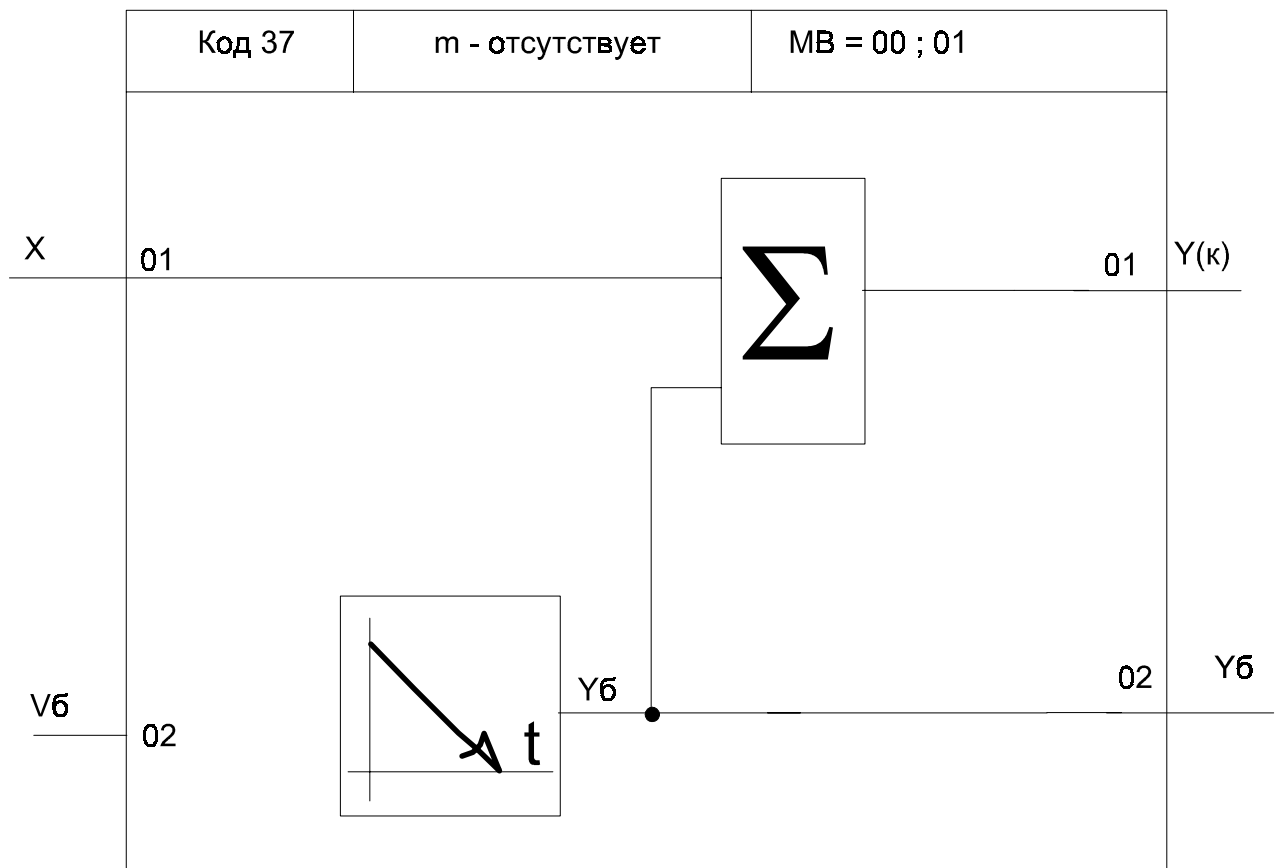


Рис. 27

5.6. ОГС(38) - Ограничение скорости

Назначение

Алгоритм используется в тех случаях, когда необходимо ограничить скорость изменения сигнала.

Описание алгоритма

Алгоритм содержит узел ограничения скорости (рис. 29). Работа алгоритма поясняется рис. 28.

В любой момент времени выходной сигнал Y стремится сравняться с входным сигналом X . Если $Y=X$ и скорость изменения входного сигнала V_X меньше заданного ограничения скорости $V_{огр}$ (т.е. $V_X < V_{огр}$), то выходной сигнал Y изменяется со скоростью $V_Y = V_X$, оставаясь в каждый момент времени равным сигналу X . Если $V_X > V_{огр}$ сигнал Y начинает изменяться со скоростью $V_{огр}$ до тех пор, пока не сравняется с сигналом X .

Алгоритм имеет два дискретных выхода D_6 и D_M . Если скорость V_Y меньше заданного ограничения $V_{огр}$, сигналы на обоих выходах равны нулю. В противном случае появляется сигнал на выходе D_6 или D_M , в зависимости от того, увеличивается (изменяется в направлении "больше") или уменьшается (изменяется в направлении «меньше») выходной сигнал Y . Состояние выходов D_6 и D_M задается следующей таблицей:

Таблица 31

V_Y	D_6	D_M
$ V_Y < V_{огр}$	0	0
$V_Y = V_{огр}$	1	0
$V_Y = -V_{огр}$	0	1

Режимы работы

Алгоритм не иницирует обратный счет, но если по инициативе других алгоритмов на каскадный выход Y поступает команда отключения со значением начальных условий Y_0 , внутренняя ячейка отслеживает этот сигнал, так что при включении алгоритма начальное значение сигнала на выходе Y равно Y_0 .

Алгоритм реагирует на сигнал запрета, запрещая изменение выходного сигнала в направлении увеличения или уменьшения (в зависимости от вида запрета).

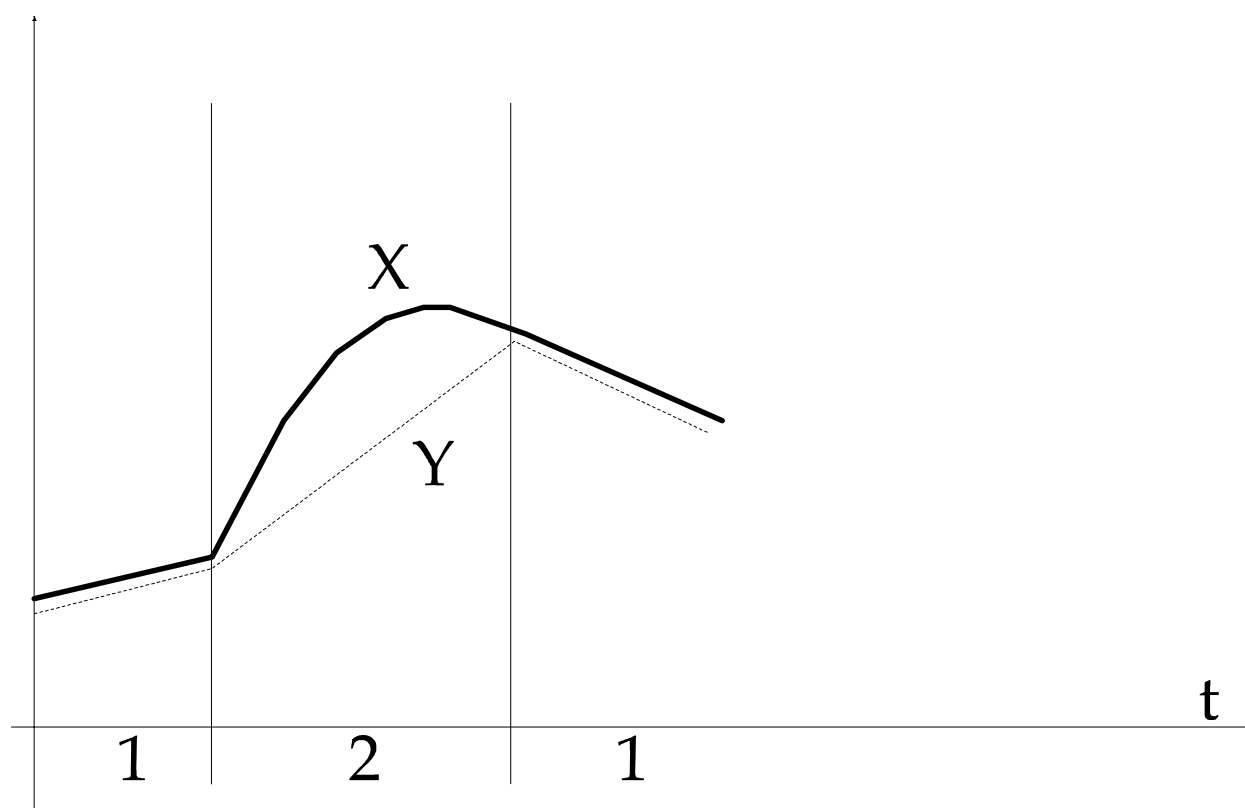
Команды отключения и запрета, а также значение начальных условия Y_0 , транслируются через каскадный вход X предвключенному алгоритму.

Входы-выходы алгоритма ОГС

Таблица 32

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	$V_{огр}$		Заданное значение ограничения скорости
01	Y	Выходы	Основной выход (каскадный)
02	D_6		Изменение с максимальной скоростью в направлении "больше"
03	D_M		То же в направлении "меньше"

Диаграмма работы алгоритма ОГС



1 - $|VY| < V_{огр}$
 2 - $|VY| = V_{огр}$

Рис. 28

Функциональная схема алгоритма
"Ограничение скорости ОГС"

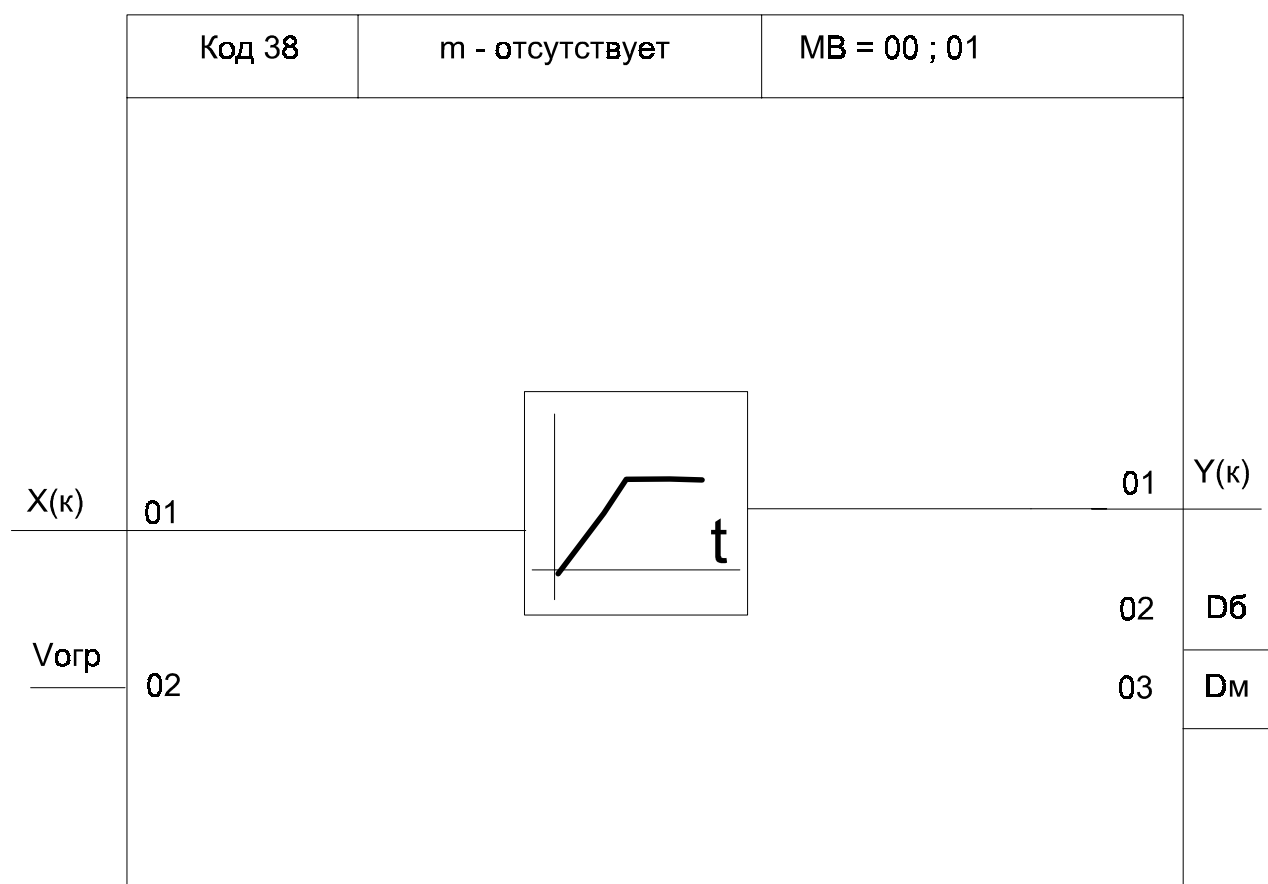


Рис. 29

5.7. ЗАП(39) - Запаздывание

Назначение

Алгоритм моделирует звено чистого запаздывания и используется для введения динамической коррекции или для моделирования динамических свойств объекта управления.

Описание алгоритма

Алгоритм содержит несколько (до 12) ячеек памяти. Число этих ячеек $0 \leq m \leq 12$ задается модификатором (рис. 31).

По каждой команде отсчета происходит «проталкивание» сигналов через ячейки памяти: в первую ячейку записывается текущее значение входного сигнала X , значение ранее записанное в первой ячейке, передается во вторую ячейку, предыдущее значение, записанное во второй ячейке, пересылается в третью ячейку и т.д. Текущее значение последней ячейки является выходным сигналом.

Таким образом, текущее значение входного сигнала X появится на выходе Y через m отсчетов (рис. 30). При этом выходной сигнал Y будет запаздывать относительно сигнала X на время $m \cdot T$, где T - интервал времени между отсчетами (при условии, что этот интервал постоянен). Если $m=0$, то всегда $Y=X$.

Команда отсчета может поступать либо от внутреннего таймера, либо извне - на вход $C_{отс}$. В последнем случае отсчет осуществляется по переднему фронту сигнала $C_{отс}$. Алгоритм имеет дискретный вход $C_{реж}$. Если $C_{реж}=0$, то управление ведется от внутреннего таймера, если $C_{реж}=1$, то от внешней команды $C_{отс}$.

В момент отсчета (в любом режиме) на дискретном выходе $D_{отс}$ на время, равное одному циклу работы контроллера (т.е. на время от 0,2 до 2 с) формируется сигнал $D_{отс}=1$. В остальное время $D_{отс}=0$. Этот выход позволяет - включать алгоритмы ЗАП последовательно, увеличивая время запаздывания без увеличения времени квантования входного сигнала. Для этого выход $D_{отс}$ каждого предыдущего алгоритма ЗАП соединяется со входом $C_{отс}$ каждого последующего алгоритма ЗАП, причем в последующих алгоритмах используется внешнее управление отсчетами.

При управлении от внутреннего таймера интервал времени между отсчетами (период квантования) задается с помощью настроечного входа T . Значение $T \leq T_0$ ($T_0=0,2-2$ с - время цикла работы контроллера), алгоритм воспринимает как $T=T_0$.

При сигнале «сброс» ($C_{сбр}=1$) во все ячейки памяти записывается сигнал на входе X , при этом $Y=X$ (т.е. запаздывание отсутствует). Если используется внешнее управление и сигналы $C_{отс}=1$, $C_{сбр}=1$ то после, снятия сигнала "сброс" происходит однократное проталкивание (т.е. такая ситуация воспринимается алгоритмом как приход переднего фронта сигнала $C_{отс}$). Аналогично ведет себя алгоритм, если при $C_{отс}=1$ осуществляется переход от внутреннего к внешнему управлению (т.е. выполняется переход от $C_{реж}=0$ и $C_{реж}=1$).

При управлении от внутреннего таймера время чистого запаздывания определяется выражением

$$\tau_{зап} = m \cdot T \quad (25)$$

Если $m=0$, то $Y=X$ и $D_{отс}=0$. Для того, чтобы при сохранении требуемого значения $t_{зап}$ уменьшить период квантования входного сигнала, следует уменьшать время T и увеличивать значение модификатора m . Если требуемое значение $m > 12$ следует использовать несколько алгоритмов запаздывания, включенных последовательно. В этом случае общее время запаздывания равно

$$\tau_{зап} = m_1 \cdot T_1 + m_2 \cdot T_2 + \dots + m_i \cdot T_i; \quad (26)$$

где $m_i; T_i$ - соответственно модификатор и параметр T для i -го алгоритма запаздывания, включенного в последовательную цепочку. Если значения m и T у каждого алгоритма одинаковы, общее время запаздывания равно

$$\tau_{зап} = m \cdot T \cdot n, \quad (27)$$

где n - число последовательно включенных алгоритмов запаздывания.

Режимы работы

Алгоритм не инициирует обратный счет, но если по инициативе других алгоритмов на каскадный выход Y пришла команда отключения со значением начальных условий Y_0 , то значение Y_0 записывается во все ячейки памяти и внутренний таймер обнуляется. Команда отключения и величина Y_0 через каскадный вход X транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, поступающую на выход алгоритма. Но эта команда через вход X транслируется предвключенному алгоритму.

Входы-выходы алгоритма ЗАП

Таблица 33

Входа-выхода			Назначение
№	Обозн.	Вид	
01	X	Входы	Вход (каскадный)
02	$C_{сбр}$		Команда сброса
03	$C_{отс}$		Команда отсчета
04	$C_{реж}$		Выбор режима
05	T		Период отсчета

01	Y	Выходы	Основной выход (каскадный)
02	D _{отс}		Признак отсчета

Работа алгоритма запаздывания (ЗАП) при $m=5$

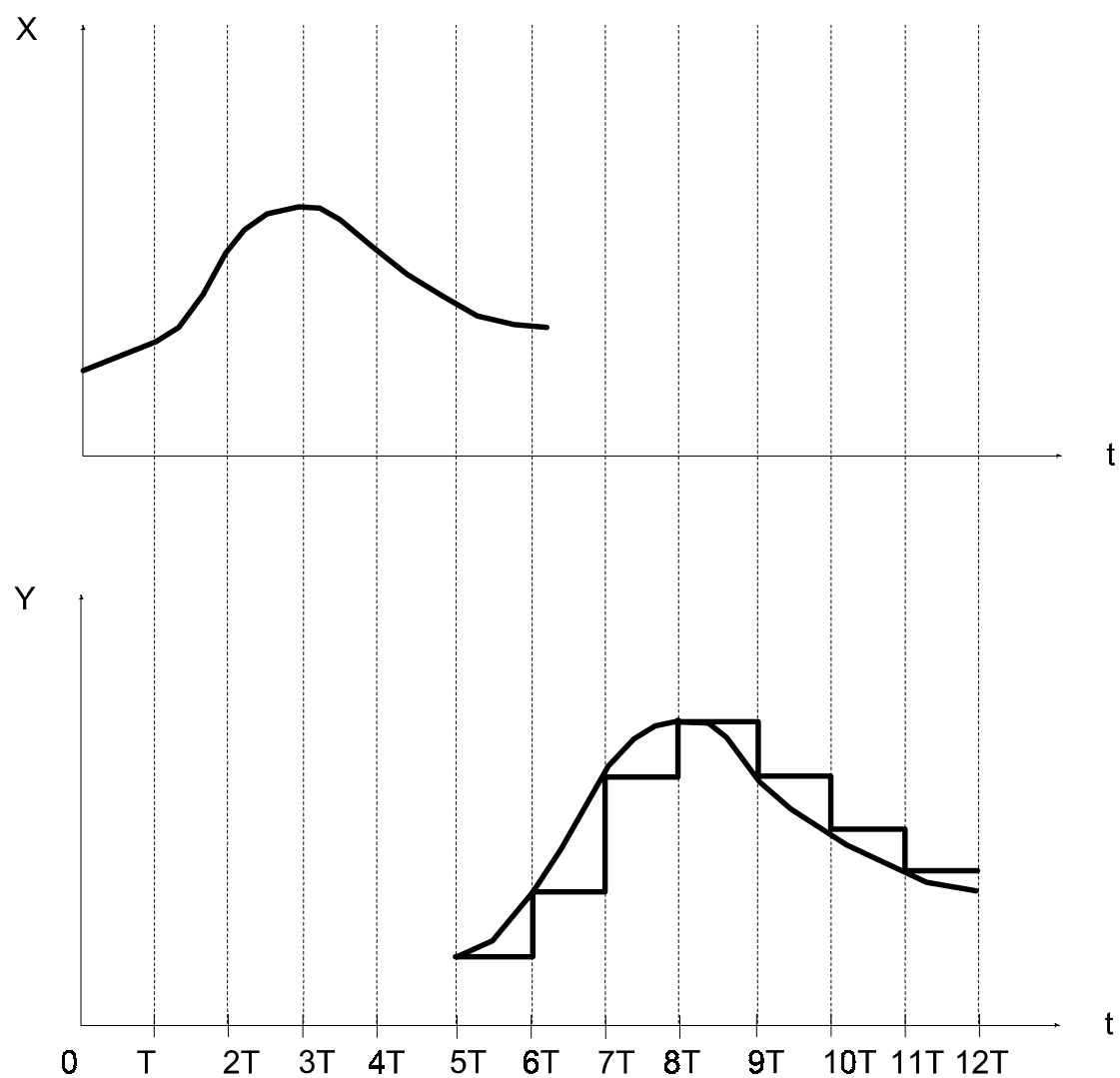
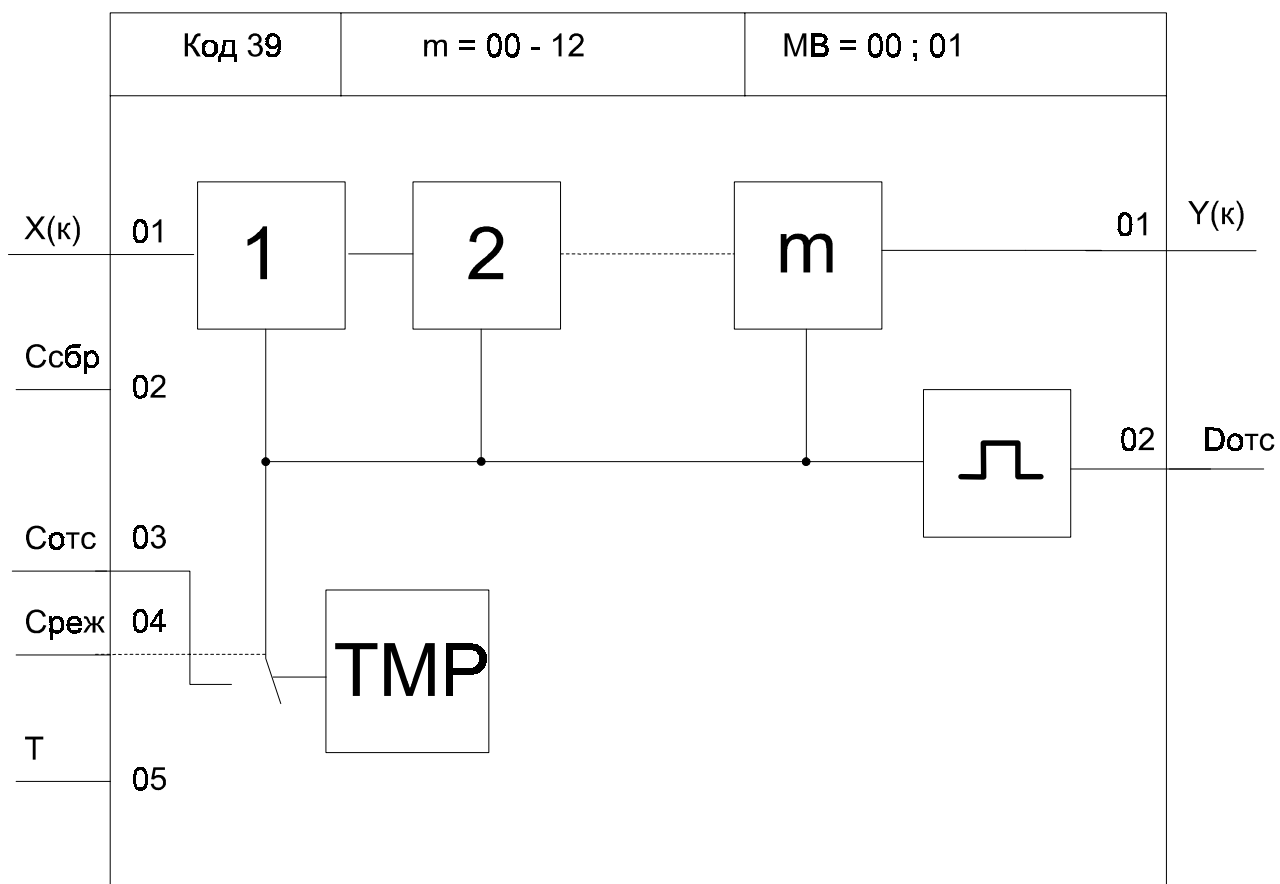


Рис. 30

Функциональная схема алгоритма
"Запаздывание ЗАП"



1;2...m - ячейки памяти;
ТМР - таймер

Рис. 31

6. Статические преобразования

6.1. СУМ(42) – Суммирование

Назначение

Алгоритм используется для суммирования (без масштабирования) нескольких (до 21) сигналов.

Описание алгоритма

Выходной сигнал алгоритма равен (рис. 32)

$$Y = X_0 + X_1 + X_2 + \dots + X_m \quad (28)$$

Число дополнительных (к X_0) суммируемых сигналов $0 \leq m \leq 20$ задается модификатором (при $m=0$ на входе имеется сигнал X_0 , при $m=1$ – сигнал X_0 и X_1 и т. д.).

Режимы работы

Алгоритм не иницирует обратный счет, но если по инициативе других алгоритмов на каскадный выход Y приходит команда отключения со значением начальных условий Y_0 , алгоритм выполняет процедуру обратного счета, формируя на каскадном входе X_0 сигнал

$$X_0 = Y_0 - (X_1 + X_2 + \dots + X_m) \quad (29)$$

Этот сигнал вместе с командой отключения транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, но транслирует ее через вход X_0 предвключенному алгоритму.

Входы-выходы алгоритма СУМ

Таблица 34

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_0	Входы	Основной вход (каскадный)
02	X_1		1-й дополнительный вход
03	X_2		2-й дополнительный вход
.	.		.
.	.		.
.	.		.
m	X_{m-1}		m-й дополнительный вход
01	Y	Выход	Вход (каскадный)

Функциональная схема алгоритма
"Суммирование СУМ"

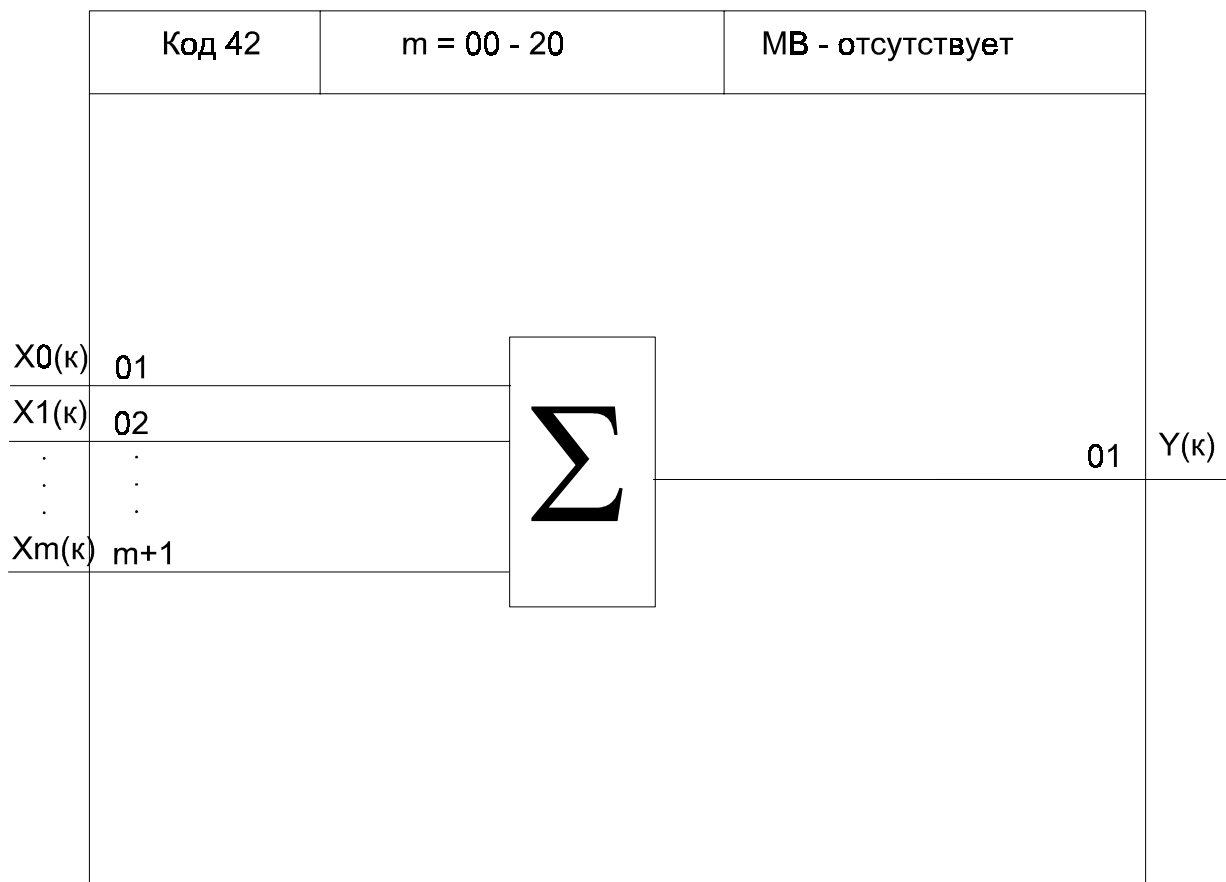


Рис. 32

6.2. СМА(43) - Суммирование с масштабированием

Назначение

Алгоритм используется для получения взвешенной суммы нескольких (до 21) сигналов. В частности, он применяется вместе с алгоритмами регулирования для построения регуляторов соотношения либо для введения статической коррекции.

Описание алгоритма

Число масштабируемых и суммируемых сигналов $0 \leq m \leq 20$ задается модификатором (немасштабируемый сигнал X_0 имеется всегда независимо от значения модификатора). Выходной сигнал алгоритма Y равен (рис. 33):

$$Y = X_0 + \sum_{i=1}^m K_{m,i} * X_i, \quad (30)$$

где K_m - масштабный коэффициент.

Режимы работы

Алгоритм не иницирует обратный счет, но если по инициативе других алгоритмов на каскадный выход Y поступает команда отключения со значением начальных условий Y_0 , алгоритм выполняет процедуру обратного счета, формируя на каскадном входе X_0 сигнал

$$X_0 = Y_0 - \sum_{i=1}^m K_{m,i} * X_i, \quad (30)$$

Этот сигнал вместе с командой отключения через вход X транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, но эта команда через вход X_0 транслируется предвключенному алгоритму.

Входы-выходы алгоритма СМА

Таблица 35

Входы-выходы			Наименование
№	Обозн.	Вид	
01	X_0	Входы	Немасштабируемый вход (каскадный)
02	X_1		1-й масштабируемый вход
03	$K_{m,1}$		Масштабируемый коэффициент по 1-му масштабируемому входу
04	X_2		2-й масштабируемый вход
05	$K_{m,2}$		Масштабный коэффициент по 2-му масштабируемому входу
.	.		.
.	.		.
.	.		.
2m	X_m		m –ый масштабируемый вход
2m+1	$K_{m,m}$		Масштабный коэффициент по m-му масштабируемому входу
01	Y	Выход	Выход (каскадный)

**Функциональная схема алгоритма
"Суммирование с масштабированием СМА"**

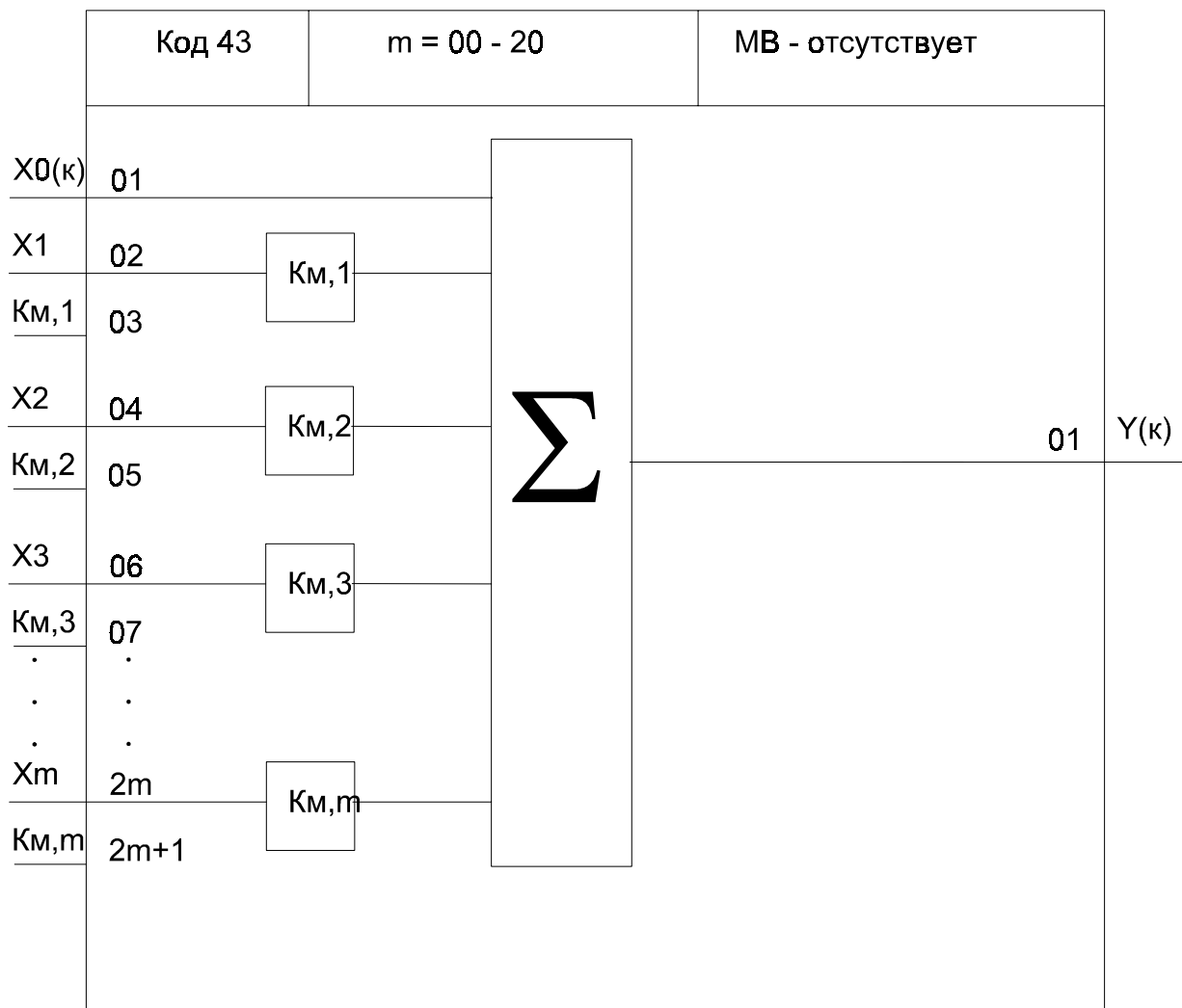


Рис. 33

6.3. УМД(44) - Умножение-деление

Назначение

Алгоритм предназначен для выполнения математических операций умножения и (или) деления.

Описание алгоритма

Алгоритм перемножает два числа и делит полученное произведение на третье число. Выходной сигнал алгоритма равен (рис .34);

$$Y = X_1 * X_2 / X_3 \quad (32)$$

Если необходимо выполнить операцию умножения, на вход X_3 задается константа, значение которой выполняет роль масштабного множителя. Если требуется выполнить операцию деления, константа задается на вход X_2 .

Режимы работы

Алгоритм не иницирует обратный счет, но если по инициативе других алгоритмов на каскадный выход Y поступает команда отключения со значением начальных условий Y_0 , алгоритм выполняет процедуру обратного счета, формируя на каскадном входе X_1 сигнал:

$$X_{1,0} = Y_0 * X_3 / X_2 \quad (32)$$

Этот сигнал вместе с командой отключения транслируется предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, но эта команда через вход X_1 транслируется предвключенному алгоритму.

Входы-выходы алгоритма УМД

Таблица 36

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Первый сомножитель (каскадный)
02	X_2		Второй сомножитель
03	X_3		Делитель
01	Y	Выход	Выход (каскадный)

Функциональная схема алгоритма
"Умножение-деление УМД"

	Код 44	m - отсутствует	МВ - отсутствует
X0(κ)	01	$\frac{X1 * X2}{X3}$	01 Y(κ)
X1	02		
X2	04		
X3	06		

Рис. 34

6.4. КОР(45) - Корень квадратный

Наименование

Алгоритм применяется в схемах статической коррекции. В частности, алгоритм используется для линеаризации характеристики датчиков расхода на сужающих устройствах.

Описание алгоритма

Выходной сигнал алгоритма равен (рис. 35):

$$Y = 10 \sqrt{X} \quad \text{при } X \geq 0 \quad (34)$$

$$Y = -10 \sqrt{|X|} \quad \text{при } X < 0 \quad (35)$$

В этих соотношениях все сигналы выражены в процентах, так что при $X=100\%$ выходной сигнал $Y=100\%$.

Режимы работы

Алгоритм не инициирует обратного счета, но если по инициативе другого алгоритма на каскадный выход Y поступает команда отключения со значением начальных условий Y_0 , алгоритм выполняет процедуру обратного счета, формируя на выходе X сигнал

$$X_0 = 0,01 * Y_0^2 \quad \text{при } Y_0 \geq 0 \quad (36)$$

$$X_0 = -0,01 * Y_0^2 \quad \text{при } Y_0 < 0 \quad (36)$$

Этот сигнал вместе с командой ОТКЛЮЧЕНИЯ через каскадный вход X транслирует предвключенному алгоритму.

Алгоритм не реагирует на команду запрета, однако эта команда через вход X транслируется предвключенному алгоритму.

Входы-выходы алгоритма КОР

Таблица 37

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Вход	Вход (каскадный)
01	Y	Выход	Выход (каскадный)

Функциональная схема алгоритма
"Корень квадратный КОР"

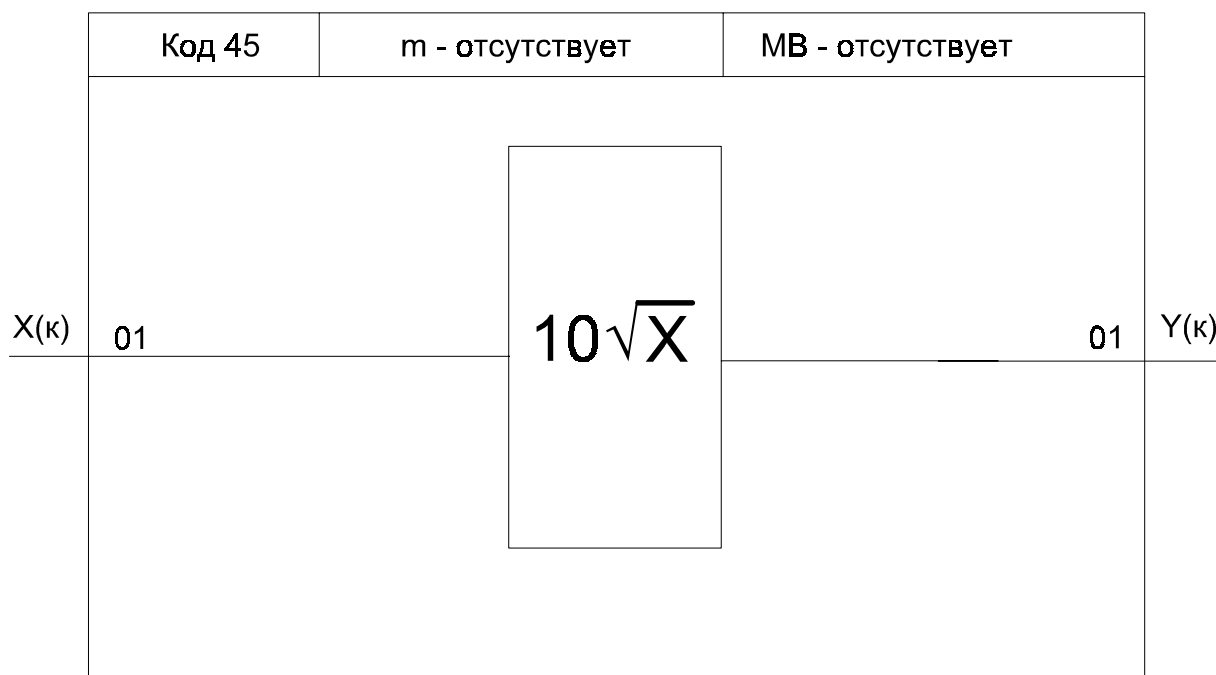


Рис. 35

6.5. МОД(46) - Модуль

Назначение

Алгоритм применяется для выделения модуля сигнала (рис. 36). Один алгоритм содержит несколько (до 20) независимых каналов выделения модуля.

Описание алгоритма

Число независимых каналов, в которых выделяется модуль, равно $0 \leq m \leq 20$, причем m задается модификатором. При $m=0$ алгоритм является “пустым”.

В каждом канале выполняется функция:

$$Y_i = |X_i|$$

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма МОД

Таблица 38

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Входной сигнал 1-го канала
02	X_2		Входной сигнал 2-го канала
.	.		.
.	.		.
.	.		.
m	X_m		Входной сигнал канала m
01	Y_1	Входы	Выходной сигнал 1-го канала
02	Y_2		Выходной сигнал 2-го канала
.	.		.
.	.		.
.	.		.
m	Y_m		Выходной сигнал канала m

Функциональная схема алгоритма
"Модуль МОД"

	Код 46	m = 00 - 20	МВ - отсутствует	
X1	01	·	01	Y1
X2	02	·	02	Y2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Xm	m	·	m	Ym

Рис. 36

6.6. КУС(47) - кусочно-линейная функция

Назначение

Алгоритм применяется для линеаризации нелинейной зависимости (например, датчика или регулирующего органа). Алгоритм применяется также для искусственного введения нелинейности (например, зоны нечувствительности, релейной характеристики и т.п.).

Описание алгоритма

Выходной сигнал Y связан со значением сигнала на входе X зависимостью, показанной на рис. 38. Число отрезков определяется модификатором m и может задаваться в пределах $0 \leq m \leq 49$. Координата конца отрезка определяется парой сигналов X_i, Y_i (абсцисса и ордината конца отрезка), которые задаются соответствующими значениями сигналов на настроечных входах алгоритма.

При $X < X_1$ и $X > X_m$ выходной сигнал $Y = \text{const}$.

На выходе $N_{\text{уч}}$ формируется число, равное номеру текущего участка кусочно-линейной зависимости (рис. 37).

Координаты X_i, Y_i могут находиться в любом из четырех квадрантов. Как правило координата $X_i \geq X_{i-1}$ (i – номер отрезка). Если задано $X_i < X_{i-1}$ то формируется функция, показанная на рис. 37 б. Можно задавать $X_i = X_{i-1}$. В этом случае формируется функция, показанная на рис. 37 в, причем при $X_i = X = X_{i-1}$ выход $Y = Y_i$.

Алгоритм работает правильно, если $|Y_{i+1} - Y_i| \leq 200\%$. При $m=0$ выходной сигнал $Y=0$ независимо от X .

Режимы работы

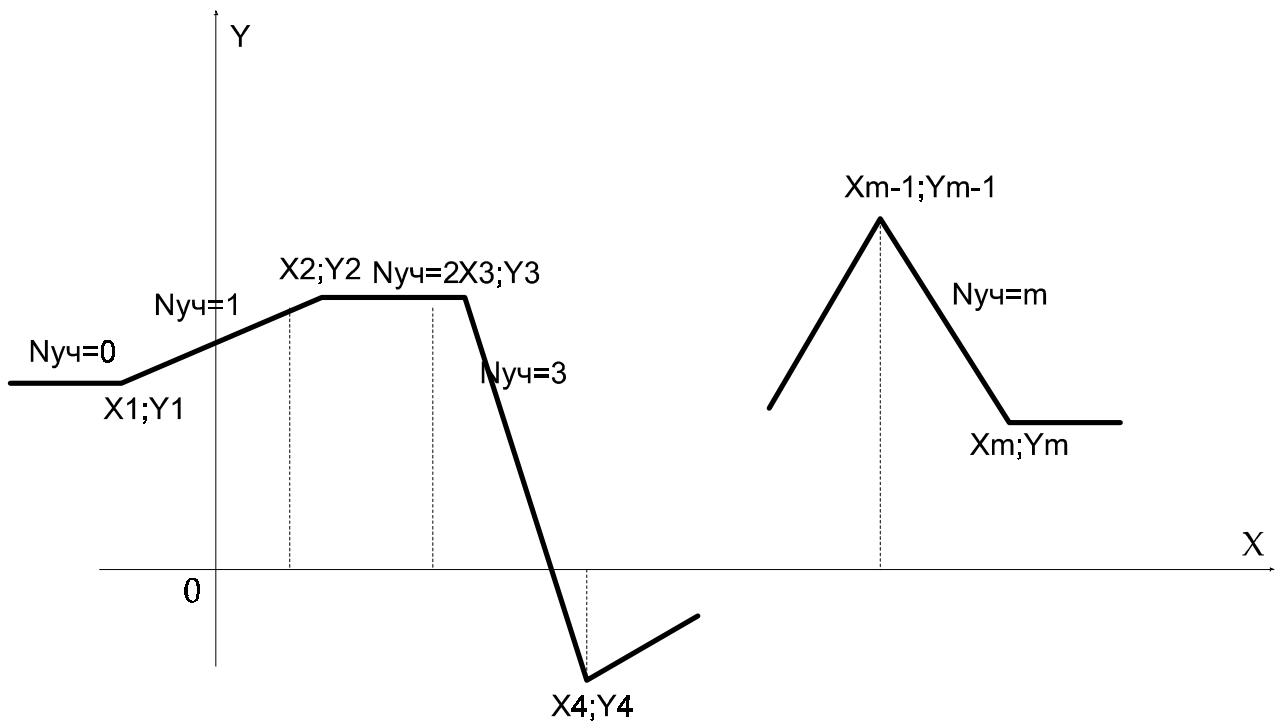
Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма КУС

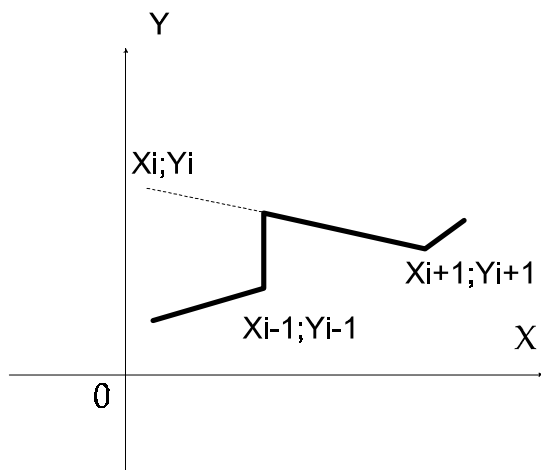
Таблица 39

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход алгоритма
02	X_1		Абсцисса 1-го участка
03	Y_1		Ордината 1-го участка
04	X_2		Абсцисса 2-го участка
05	Y_2		Ордината 2-го участка
.	.		.
.	.		.
.	.		.
$2m$	X_m		Абсцисса m -участка
$2m + 1$	Y_m		Ордината m -го участка
01	Y	Выходы	Основной выход алгоритма
02	$N_{\text{уч}}$		Номер текущего участка

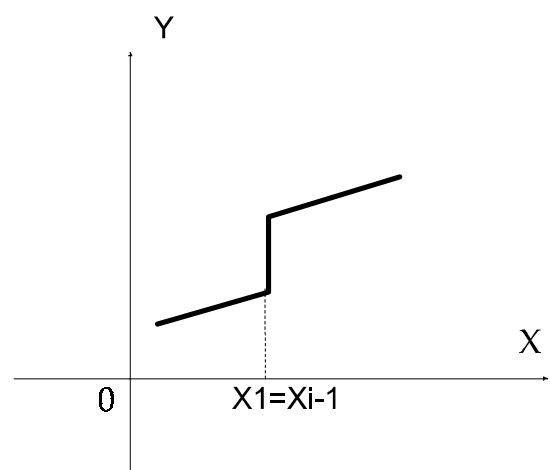
Кусочно-линейные зависимости



а)



б)



в)

- а) общий случай
 б) $X_i < X_{i-1}$
 в) $X_i = X_{i-1}$

Рис. 37

6.7. ОГР(48) - Ограничение

Назначение

Алгоритм используется для ограничения верхней и (или) нижней границы диапазона изменения сигнала.

Описание алгоритма

Алгоритм содержит ограничитель верхнего и нижнего значения сигнала. На двух дискретных выходах D_v и D_n фиксируется достижение сигналом верхней и нижней границы ограничения (рис. 39).

Работа алгоритма определяется следующей таблицей:

Таблица 40

X	Y	D_v	D_n
$X_v > X > X_n$	X	0	0
$X \geq X_v$	X_v	1 0	0
$X \leq X_n$	X_n	0	1

Алгоритм будет правильно работать, только если $X_v > X_n$.

Верхняя и нижняя границы диапазона задаются на настроечных входах X_v и X_n .

Режимы работы

Алгоритм является инициатором режима запрета. Если сигнал Y достиг порога ограничения X_v или X_n , и в цепи есть следящие алгоритмы, на каскадном входе X формируется команда запрета соответственно вверх или вниз. Эта команда транслируется предвключенному алгоритму.

Для последующих алгоритмов алгоритм ОГР не иницирует обратный счет, но если на каскадный выход Y поступит команда отключения со значением начальных условий Y_0 или команда запрета, эти сигналы транслируются предвключенному алгоритму через каскадный вход X. Величина Y_0 может превысить установленные уровни ограничения. Ограничитель этому не препятствует, на выходах D_v , D_n алгоритма при этом формируются дискретные сигналы: $D_v=1$ и $D_n=0$ при $Y_0 \geq X_v$; $D_v=0$ и $D_n=1$ при $Y_0 \leq X_n$.

После включения алгоритма выходной сигнал может находиться за порогами ограничения, однако изменение выходного сигнала возможно лишь в направлении, приближающем его к порогу ограничения. Если выходной сигнал изменяется именно в этом направлении команда запрета не формируется. В противном случае формируется команда запрета, которая через вход X транслирует предвключенному алгоритму.

Входы-выходы алгоритма ОГР

Таблица 41

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Вход (каскадный)
02	X_v		Верхняя граница ограничения
03	X_n		Нижняя граница ограничения
01	Y	Выходы	Основной выход (каскадный)
02	D_v		Достижение верхней границы
03	D_n		Достижение нижней границы

Функциональная схема алгоритма
"Ограничение ОГР"

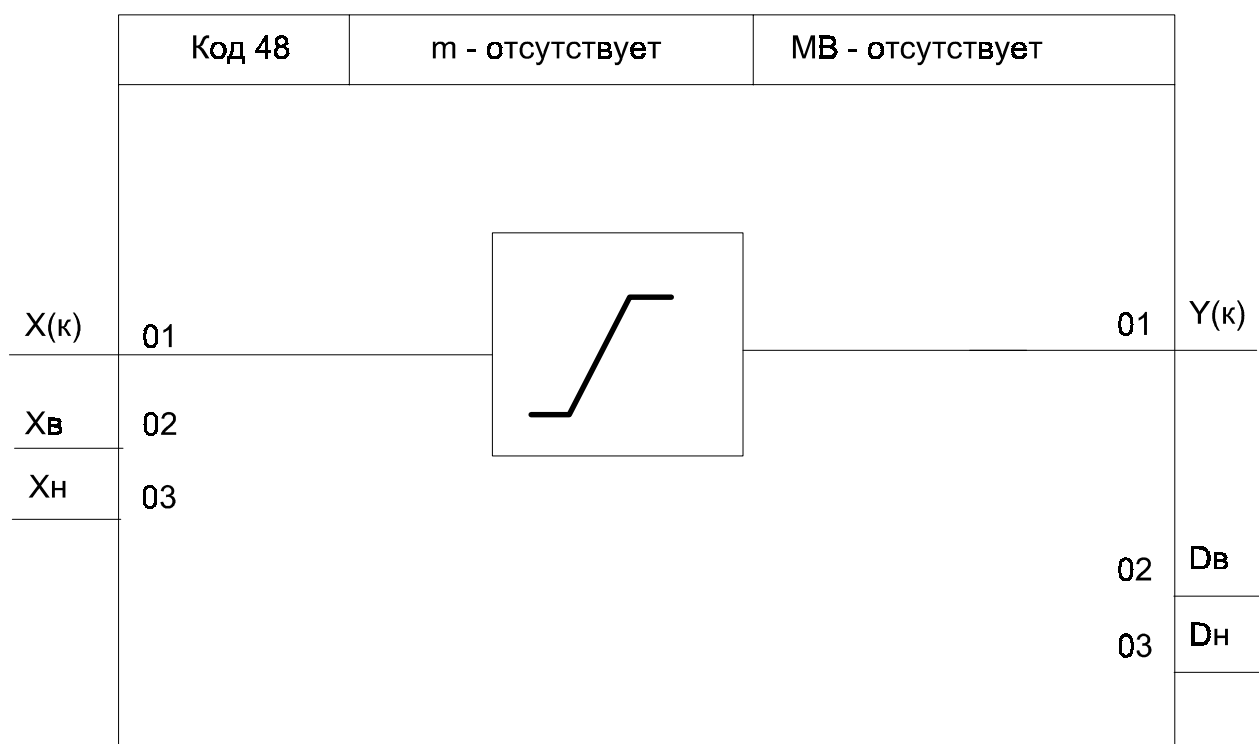


Рис. 39

6.8. СКС(49) - Скользящее среднее

Назначение

Алгоритм применяется для вычисления среднего из нескольких (до 12) последних отсчетов аналогового сигнала.

Описание алгоритма

Структура и работа алгоритма СКС соответствует алгоритму запаздывания ЗАП (см. рис. 40). Отличие алгоритма СКС заключается в наличие дополнительного сумматора, который усредняет значение сигналов на выходах всех ячеек запаздывания. При этом выходной сигнал Y_{cp} равен

$$Y_{cp} = \sum_{i=1}^m X_i / m, \quad (38)$$

где $1 \leq m \leq 12$ - значение модификатора, X_i - значение сигнала на выходе i -ой ячейки памяти. При $m=0$ выходной сигнал $Y_{cp}=X$ и признак отсчета $D_{отс}=0$.

Если требуемое число отсчетов при усреднении превышает 12, алгоритмы СКС по аналогии с алгоритмами ЗДП включаются последовательно, при этом с помощью алгоритма суммирования с масштабированием СМА сигналы $Y_{cp,n}$ от каждого n -го алгоритма СКС суммируется и по каждому $Y_{cp,n}$ устанавливается масштабный коэффициент

$$K_{m,n} = m_n / \sum_{j=1}^l m_j; \quad (39)$$

где m_n - модификатор n -го алгоритма СКС, l - общее число последовательно включенных алгоритмов СКС. Полученная сумма будет равна искомому среднему значению.

При сигнале $C_{сбр}=1$ таймер обнуляется и во все ячейки записывается текущее значение сигнала X при этом $Y_{cp}=X$.

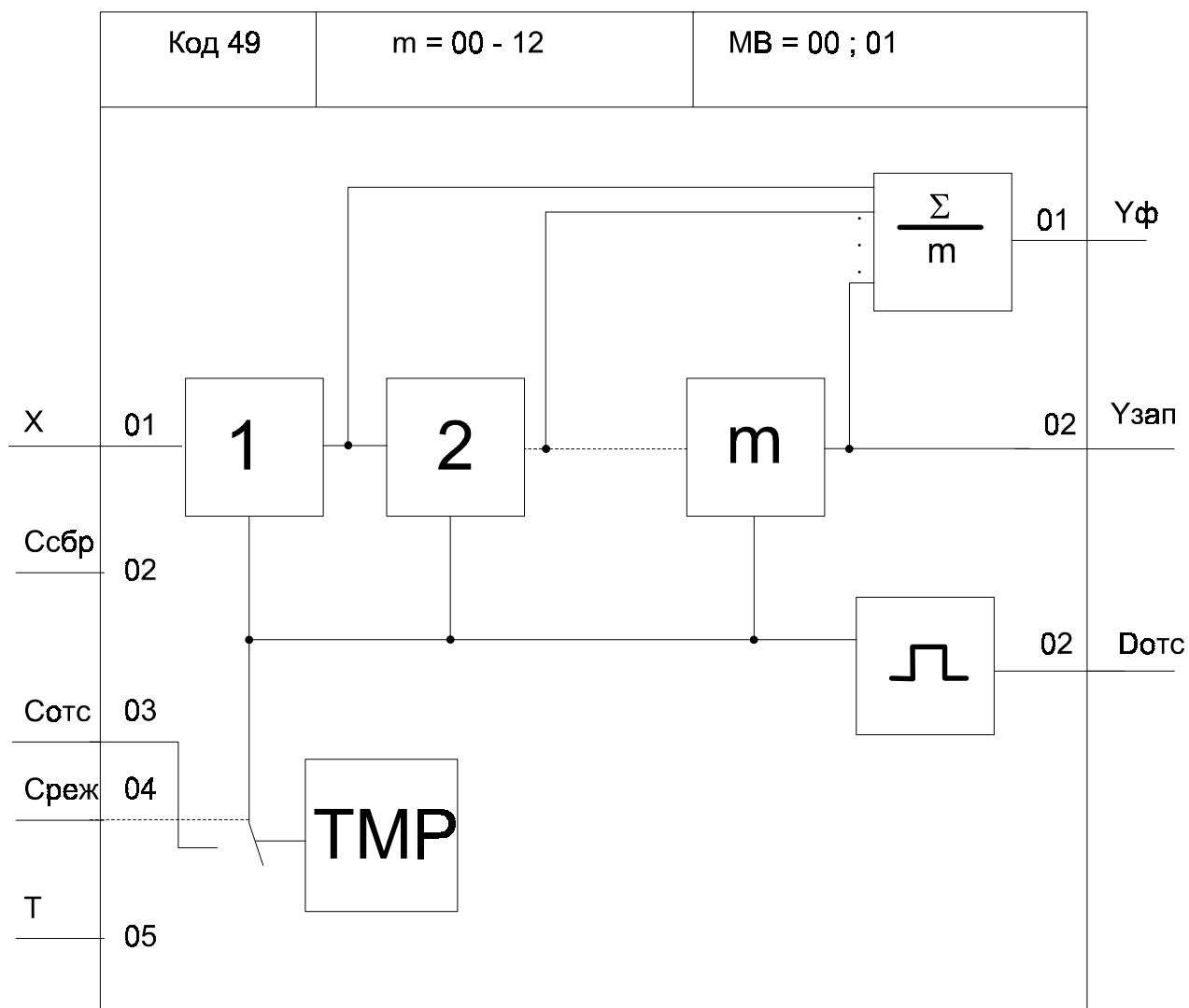
Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма СКС

Таблица 42

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Вход
02	$C_{сбр}$		Команда сброса
03	$C_{отс}$		Команда отсчета
04	$C_{реж}$		Выбор режима
05	T		Период отсчета
01	Y_{cp}	Выходы	Основной выход
02	$Y_{зап}$		Запаздывающее значение входа
03	$D_{отс}$		Признак отсчета

Функциональная схема алгоритма
"Скользящее среднее СКС"



1;2...m - ячейки памяти;
TMP - таймер

Рис. 40

6.9. ДИС (50) - Дискретное среднее

Назначение

Алгоритм применяется для усреднения аналогового сигнала либо на фиксированном отрезке времени, либо за заданное число отсчетов.

Описание алгоритма

Алгоритм содержит одну ячейку памяти, в которой суммируются значения входного сигнала, полученные за m отсчетов, после чего полученная сумма делится на m . Число отсчетов, за которое усредняется входной сигнал, задается модификатором и может находиться в пределах $0 \leq m \leq 99$ (рис. 41).

Команда отсчета формируется либо от внутреннего таймера (при $C_{\text{реж}} = 0$), либо подается извне на вход $C_{\text{отс}}$ (при $C_{\text{реж}} = 1$). Время между отсчетами при работе от внутреннего таймера задается на входе T . При внешнем управлении момент отсчета задается приходом переднего фронта сигнала $C_{\text{отс}}$.

Счетчик C_4 считает текущее число отсчетов N и в момент, когда $N=m$, среднее значение, вычисленное сумматором, запоминается в ячейке "память выхода", обновляя сигнал на выходе $Y_{\text{ср}}$. Тем самым заканчивается очередной интервал усреднения, после чего вплоть до окончания нового интервала сигнал $Y_{\text{ср}}$ не изменяется. На выходе алгоритма N формируется текущий (внутри интервала усреднения) номер отсчета. На выходе $D_{\text{отс}}$ в момент каждого отсчета в течение одного цикла работы контроллера T_0 ($T_0 = 0,2-2$ с) устанавливается $D_{\text{отс}} = 1$, в остальное время $D_{\text{отс}} = 0$. После окончания интервала усреднения сигнал на выходе $D_{\text{кон}} = 1$ (до окончания этого интервала $D_{\text{кон}} = 0$).

Для того, чтобы инициализировать новый интервал усреднения, на вход перезапуска $C_{\text{пер}}$ следует подать команду перезапуска. По переднему фронту сигнала $C_{\text{пер}}$ память сумматора и счетчик числа отсчетов обнуляются, выходной сигнал $Y_{\text{ср}}$ замораживается, устанавливается $D_{\text{кон}} = 0$ и начинается новый интервал усреднения (сигнал $C_{\text{пер}}$ может при этом перейти в ноль или оставаться равным единице). Для того, чтобы интервалы усреднения непрерывно следовали друг за другом, выход $D_{\text{кон}}$ следует соединить с входом $C_{\text{пер}}$.

При команде сброса ($C_{\text{сбр}} = 1$) выполняются те же действия, что и при команде перезапуска, за тем исключением, что при сбросе выходной сигнал $Y_{\text{ср}} = X$, т.е. выходной сигнал равен текущему значению входного сигнала. После снятия сброса начинается новый интервал усреднения.

Если команда сброса снимается, когда при внешнем управлении $C_{\text{отс}} = 1$, эта ситуация воспринимается как приход одной команды отсчета.

Команда сброса приоритетна над командой перезапуска. Если значение модификатора $m=0$, алгоритм работает так же, как при действии команды сброса.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ДИС

Таблица 43

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход
02	$C_{\text{сбр}}$		Команда сброса
03	$C_{\text{отс}}$		Команда отсчета
04	$C_{\text{реж}}$		Выбор режима
05	T		Период отсчета
06	$C_{\text{пер}}$		Команда перезапуска
01	$Y_{\text{ср}}$	Выходы	Основной выход
02	N		Номер текущего отсчета
03	$D_{\text{отс}}$		Признак отсчета
04	$D_{\text{кон}}$		Конец отсчетов

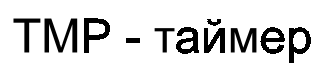


Рис. 41

6.10. МИН (51) - Минимум

Назначение

Алгоритм используется для выделения минимального из нескольких (до 99) сигналов.

Описание алгоритма

На вход алгоритма поступают сигналы, число которых $0 \leq m \leq 99$ и задается модификатором. Выходной сигнал равен минимальному из этих сигналов (рис. 42):

$$Y = \min\{x_1; x_2; \dots; x_m\} \quad (40)$$

При $m=0$ алгоритм является "пустым". Алгоритм имеет дополнительный выход $N_{\text{вх}}$ на котором формируется число, равное номеру входного сигнала, прошедшего на выход (то есть являющегося минимальным).

Если имеется группа равных между собой сигналов, причем эти сигналы являются минимальными, то номер $N_{\text{вх}}$ равен -минимальному номеру сигналов в этой группе.

Алгоритм не имеет каскадных входов-выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма МИН

Таблица 44

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X1	Входы	Сигнал на 1-м входе
02	X2		Сигнал на 2-м входе
.	.		.
.	.		.
.	.		.
m	Xm	Выходы	Сигнал на m -м входе
01	Y		Основной выход
02	$N_{\text{вх}}$		Номер входа с минимальным сигналом

Функциональная схема алгоритма
"Минимум МИН"

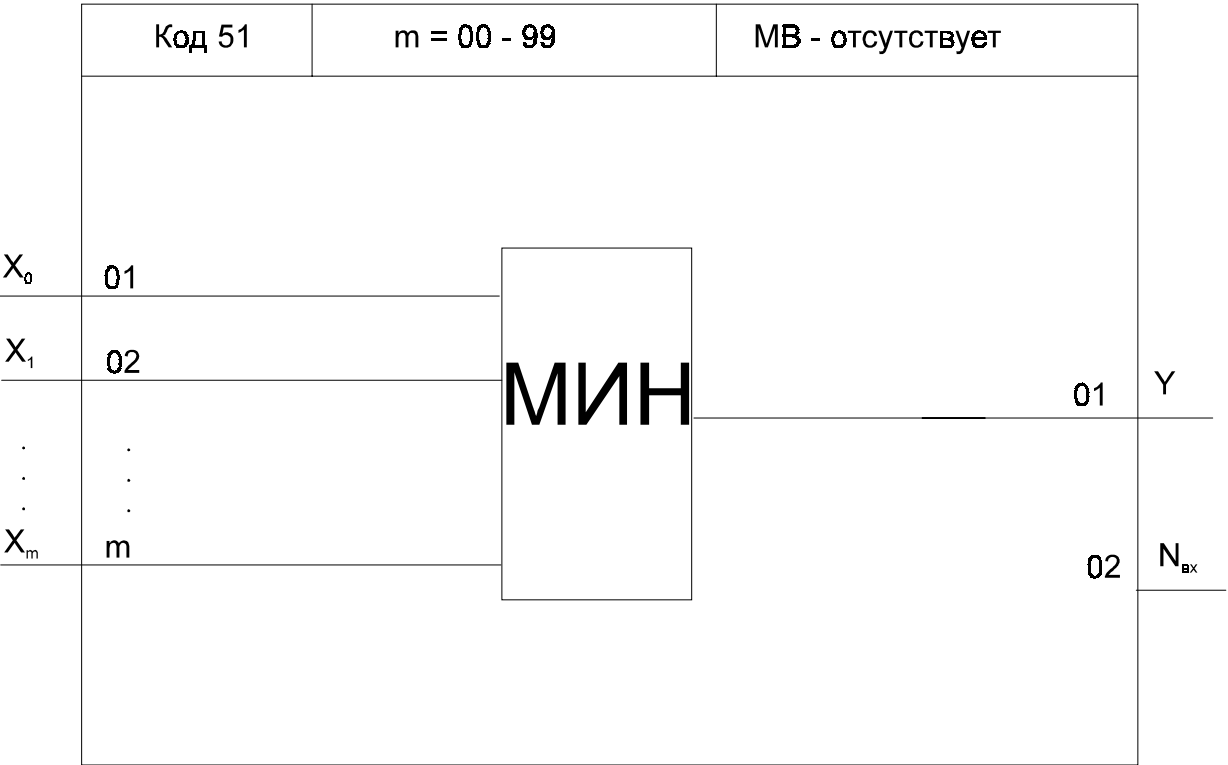


Рис. 42

6.11. МКС (52) - Максимум

Назначение

Алгоритм используется для выделения максимального из нескольких (до 99) сигналов.

Описание алгоритма

На вход алгоритма поступают сигналы, число которых $0 \leq m \leq 99$ и задается модификатором. Выходной сигнал равен максимальному из этих сигналов (рис. 43):

$$Y = \max\{x_1; x_2; \dots; x_m\} \quad (41)$$

При $m=0$ алгоритм является "пустым". Алгоритм имеет дополнительный выход $N_{\text{вх}}$, на котором формируется число, равное номеру входного сигнала, прошедшего на выход (т.е. являющегося максимальным).

Если имеется группа равных между собой сигналов, причем эти сигналы являются максимальными, то номер $N_{\text{вх}}$ равен минимальному номеру сигнала в этой группе.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма МКС

Таблица 45

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	Сигнал на 1-м входе
02	X_2		Сигнал на 2-м входе
.	.		.
.	.		.
.	.		.
m	X_m		Сигнал на m -м входе
01	Y	Выходы	Основной выход
02	$N_{\text{вх}}$		Номер входа с максимальным сигналом

Функциональная схема алгоритма
"Максимум МКС"

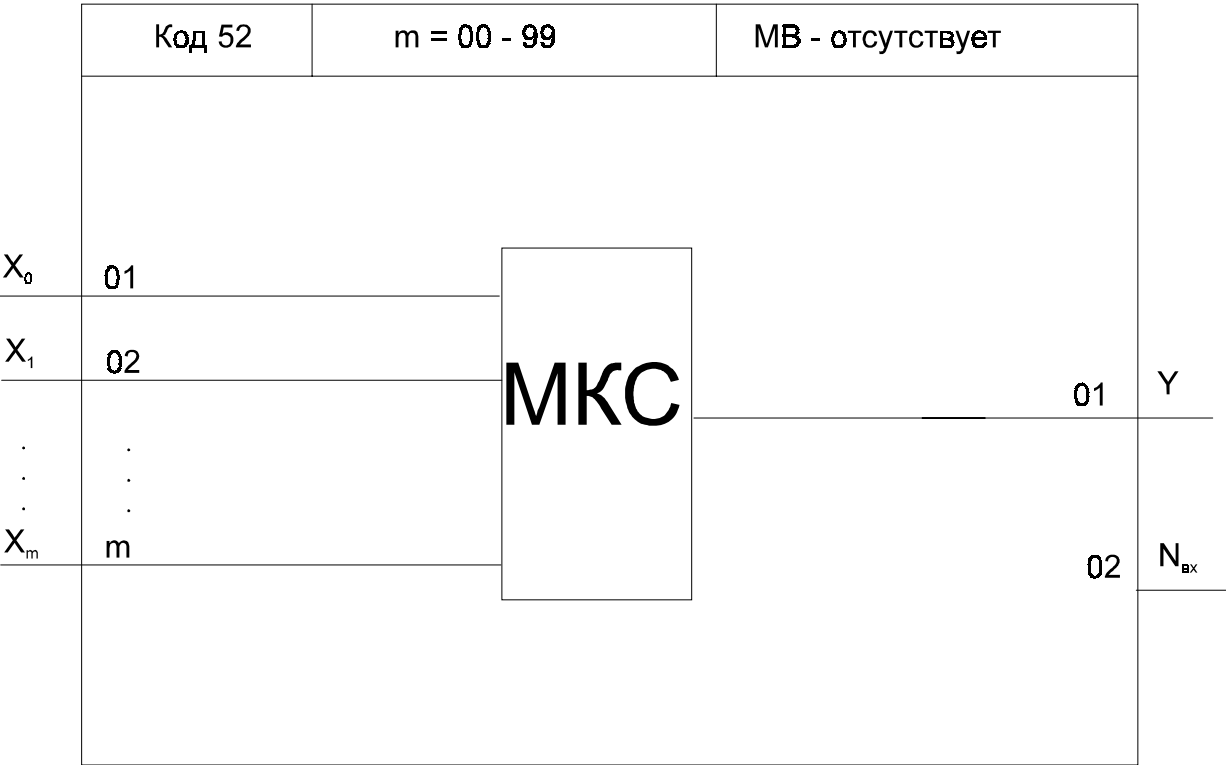


Рис. 43

6.12. СИТ (53) Средний из трех

Назначение

Алгоритм из трех аналоговых сигналов, поступающих на его вход, выбирает средний сигнал. Алгоритм применяется, например, при необходимости повысить надежность цепей аналогового входа. В этом случае один и тот же сигнал (либо сигнал от трех идентичных датчиков) подается на три входа контроллера. Полученные сигналы обрабатываются алгоритмом СИТ.

Описание алгоритма

Алгоритм содержит узел выделения среднего из трех сигналов и логику, фиксирующую недопустимое отклонение одного из сигналов от среднего сигнала (рис. 44).

Выходной сигнал Y равен одному из сигналов X_1 , X_2 или X_3 , а именно тому из них, который меньше большего, но больше меньшего из этих сигналов. Например, если $X_1=40\%$, $X_2=-10\%$, $X_3=5\%$, то $Y=5\%$.

Номер входного сигнала (I ; 2 или 3), прошедшего на выход, формируется на выходе N_0 (в предыдущем примере $N_0=3$).

Логика алгоритма (выходы D и N позволяет зафиксировать отклонилось ли значение одного из крайних сигналов от значения среднего сигнала на величину, большую заданной зоны нечувствительности X_Δ (табл.46):

Таблица 46

Условия	D	N
$ Y-X_1 < X_\Delta$ $ Y-X_2 < X_\Delta$ $ Y-X_3 < X_\Delta$	0	0
$ Y-X_1 \geq X_\Delta$	1	1
$ Y-X_2 \geq X_\Delta$	1	2
$ Y-X_3 \geq X_\Delta$	1	3

Если два или все три входных сигнала равны, то N_0 равно старшее номеру равных сигналов.

Если оба крайних сигнала отличаются от среднего на величину, большую X_Δ , то N равно младшему номеру этих сигналов. Значение $X_\Delta < 0$ воспринимается алгоритмом как $X_\Delta=0$.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма СИТ

Таблица 47

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	1-й сигнал
02	X_2		2-й сигнал
03	X_3		3-й сигнал
04	X_Δ		Зона нечувствительности
01	Y	Выходы	Основной выход
02	N_0		Номер среднего сигнала
03	D		Признак недопустимого отклонения одного из входных сигналов от среднего сигнала
04	N		Номер сигнала, имеющего недопустимое отклонение

Функциональная схема алгоритма
"Средний из трех СИТ"

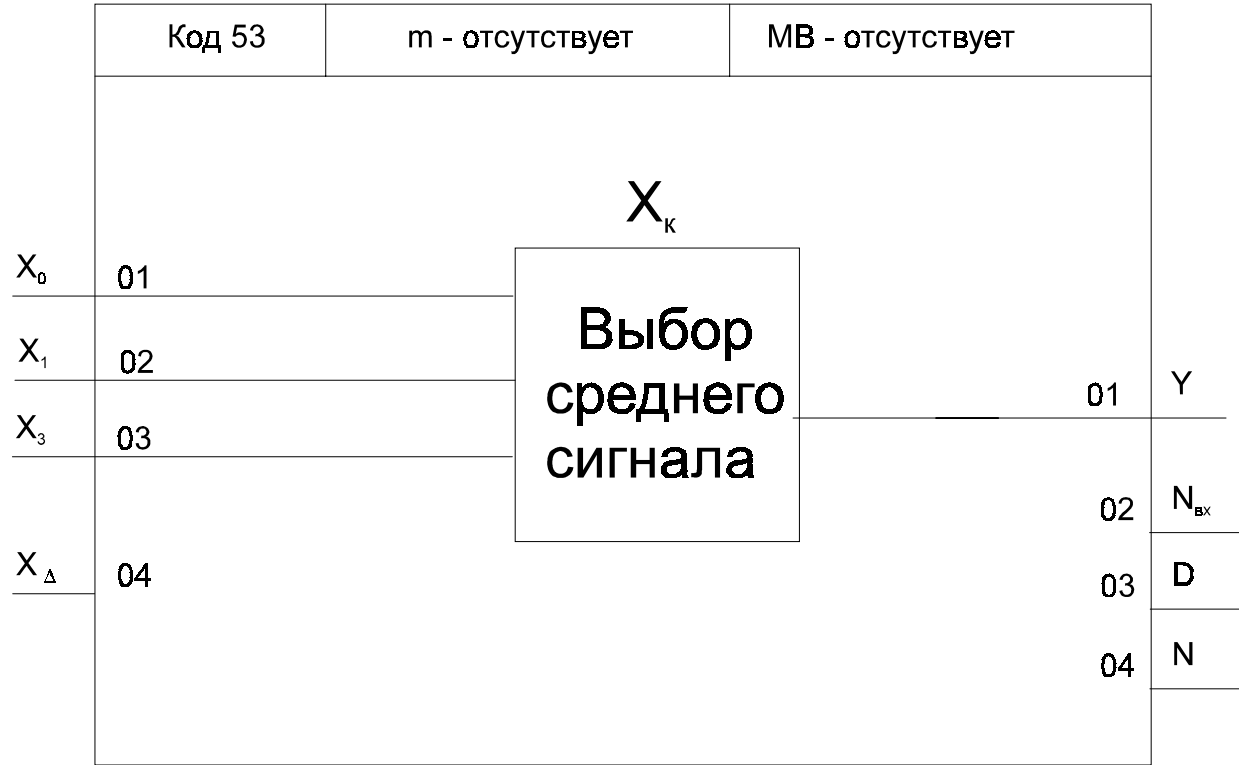


Рис. 44

6.13 ЭКС (54) - Экстремум

Назначение

Алгоритм применяется для поиска и фиксации максимального и (или) минимального значения меняющегося во времени сигнала. В частности, алгоритм используется в задачах оптимизации.

Описание

Алгоритм содержит фильтр нижних частот и узел выделения экстремума ВЭК (рис. 47).

Входной сигнал подается на сигнальный вход алгоритма X и, пройдя через фильтр нижних частот, поступает на вход узла выделения экстремума ВЭК.

На выходах алгоритма $Y_{\text{мкс}}$, $Y_{\text{мин}}$ фиксируется последнее соответственно максимальное и минимальное значение сигнала X_0 , поступающее на вход ВЭК.

Алгоритм работает следующим образом (рис. 45). До тех пор, пока имеется команда сброса ($C_{\text{сбр}}=1$), поиск экстремума не ведется и выходные сигналы $Y_{\text{мкс}}=Y_{\text{мин}}=X_0$. Когда сигнал сброса снимается ($C_{\text{сбр}}=0$), выходные сигналы $Y_{\text{мкс}}$ и $Y_{\text{мин}}$ замораживаются и начинается поиск экстремума.

Как только найден максимум, выходной сигнал $Y_{\text{мкс}}$ становится равным максимальному значению X_0 . Когда алгоритм фиксирует минимум, выходной сигнал $Y_{\text{мин}}$ принимает значение, равное минимальному значению X_0 . Выходы $Y_{\text{мкс}}$ и $Y_{\text{мин}}$ остаются неизменными вплоть до обнаружения очередного экстремума - соответственно максимума и минимума.

В момент обнаружения максимума или минимума на время, равное времени цикла работы контроллера (т. е. на время от 0,2 до 2 с), на дискретных выходах соответственно $D_{\text{мкс}}$ или $D_{\text{мин}}$ формируются дискретные сигналы $D_{\text{мкс}} = 1$, $D_{\text{мин}} = 1$.

На дискретных выходах $D_{\text{пмк}}$ (поиск максимума) и $D_{\text{пмн}}$ (поиск минимума) формируются сигналы, свидетельствующие о направлении поиска. В состоянии сброса поиск не ведется и $D_{\text{пмк}}=D_{\text{пмн}}=0$.

Если в данный момент ведется поиск максимума, то $D_{\text{пмк}}=1$, $D_{\text{пмн}} = 0$, если ведется поиск минимума, то $D_{\text{пмк}} = 0$ и $D_{\text{пмн}} = 1$.

В алгоритме предусмотрены меры по повышению помехозащищенности процесса поиска: имеется фильтр нижних частот, имеется возможность введения зоны нечувствительности и предусмотрена возможность анализа на допустимую скорость изменения сигнала в районе экстремума. Эти меры могут применяться по отдельности, а также в любых сочетаниях.

Фильтр нижних частот имеет передаточную функцию:

$$W_{\phi}(p)=X_0(p)/X(p)=1/(T_{\phi} \cdot p+1), \quad (42)$$

где T_{ϕ} - постоянная времени фильтра.

Влияние зоны нечувствительности показано на рис. 46. Если на настроечном входе алгоритма X_{Δ} установлено значение зоны нечувствительности $X_{\Delta} = 0$, то экстремум фиксируется сразу же, как только знак производной сменился на противоположный (именно такая ситуация представлена на рис. 46). Если введена зона нечувствительности ($X_{\Delta} > 0$), то экстремум фиксируется не сразу, а лишь после того, как, пройдя экстремум, сигнал изменится на величину, большую X_{Δ} . Если же сигнал изменится на меньшую величину и затем вновь начнет изменяться в исходном направлении, экстремум не фиксируется. Значение $X_{\Delta} < 0$ воспринимается алгоритмом как $X_{\Delta}=0$.

На входе $Y_{\text{доп}}$ устанавливается допустимая скорость изменения сигнала, при которой фиксируется экстремум. Если $Y_{\text{доп}} = \infty$, то ограничения на скорость не накладывается. В противном случае экстремум фиксируется лишь тогда, когда после прохождения экстремума (или при наличии зоны нечувствительности - после прохождения этой зоны) скорость изменения сигнала $Y < Y_{\text{доп}}$ (рис. 46).

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ЭКС

Таблица 48

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Входной сигнал
02	T_{ϕ}		Постоянная времени фильтра
03	X_{Δ}		Зона нечувствительности
04	$Y_{\text{доп}}$		Допустимая скорость изменения входного сигнала
05	$C_{\text{сбр}}$		Команда сброса
01	$Y_{\text{мкс}}$	Выходы	Максимальное значение входного сигнала
02	$Y_{\text{мин}}$		Минимальное значение входного сигнала
03	$D_{\text{мкс}}$		Фиксация максимума
04	$D_{\text{мин}}$		Фиксация минимума
05	$D_{\text{пмк}}$		Признак поиска максимума
06	$D_{\text{пмн}}$		Признак поиска минимума

Диаграмма работы алгоритма ЭКС

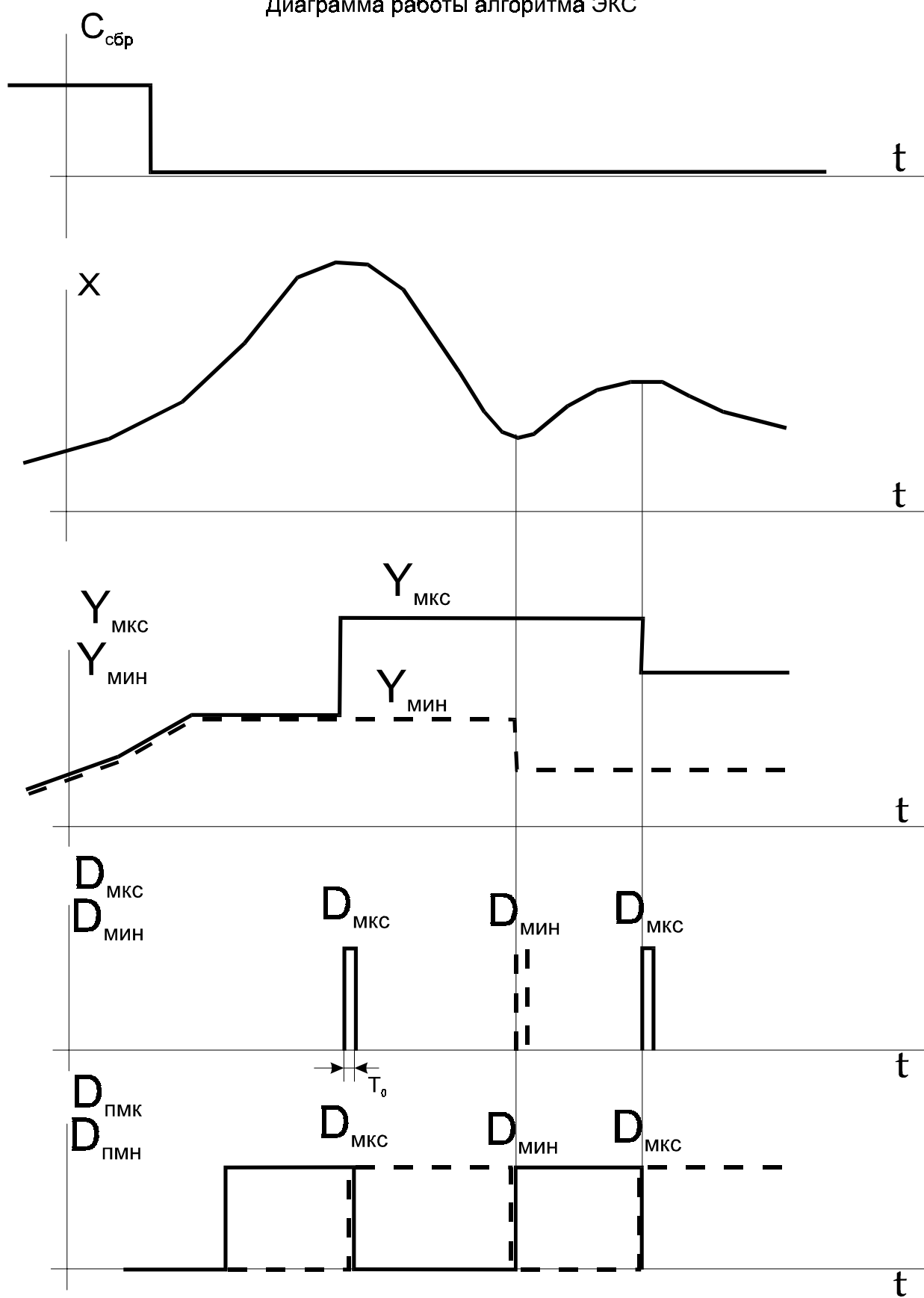
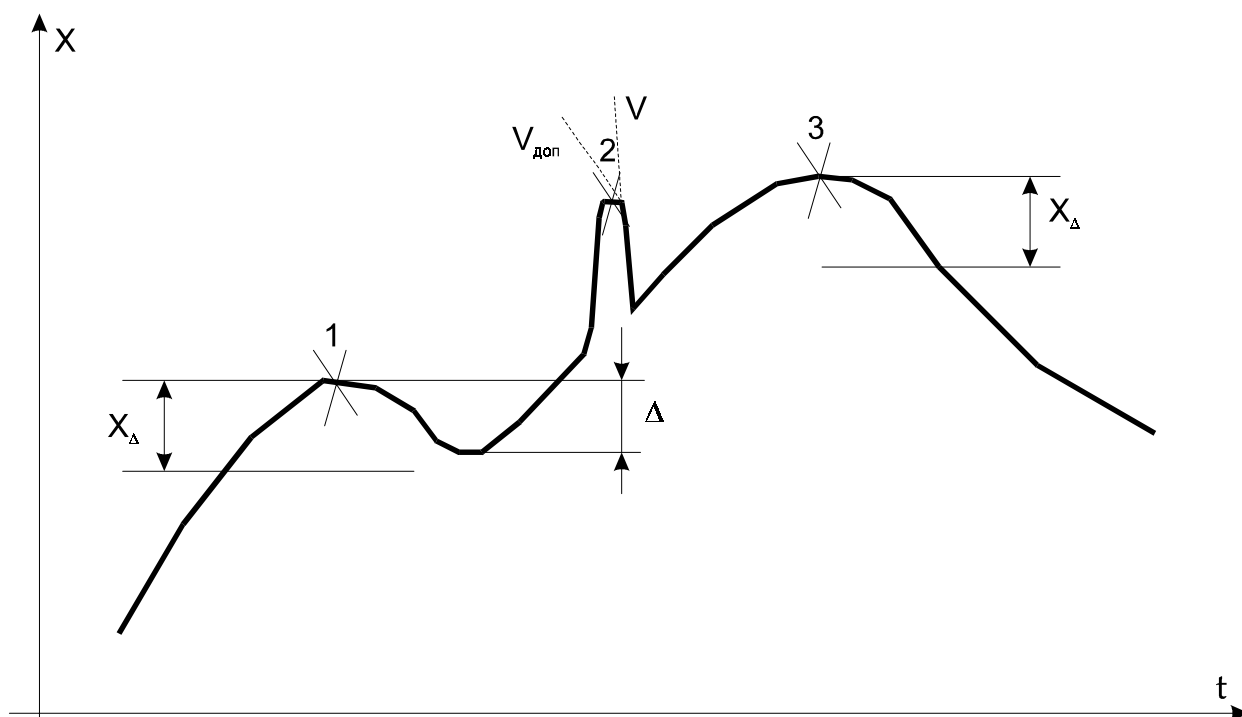


Рис. 45

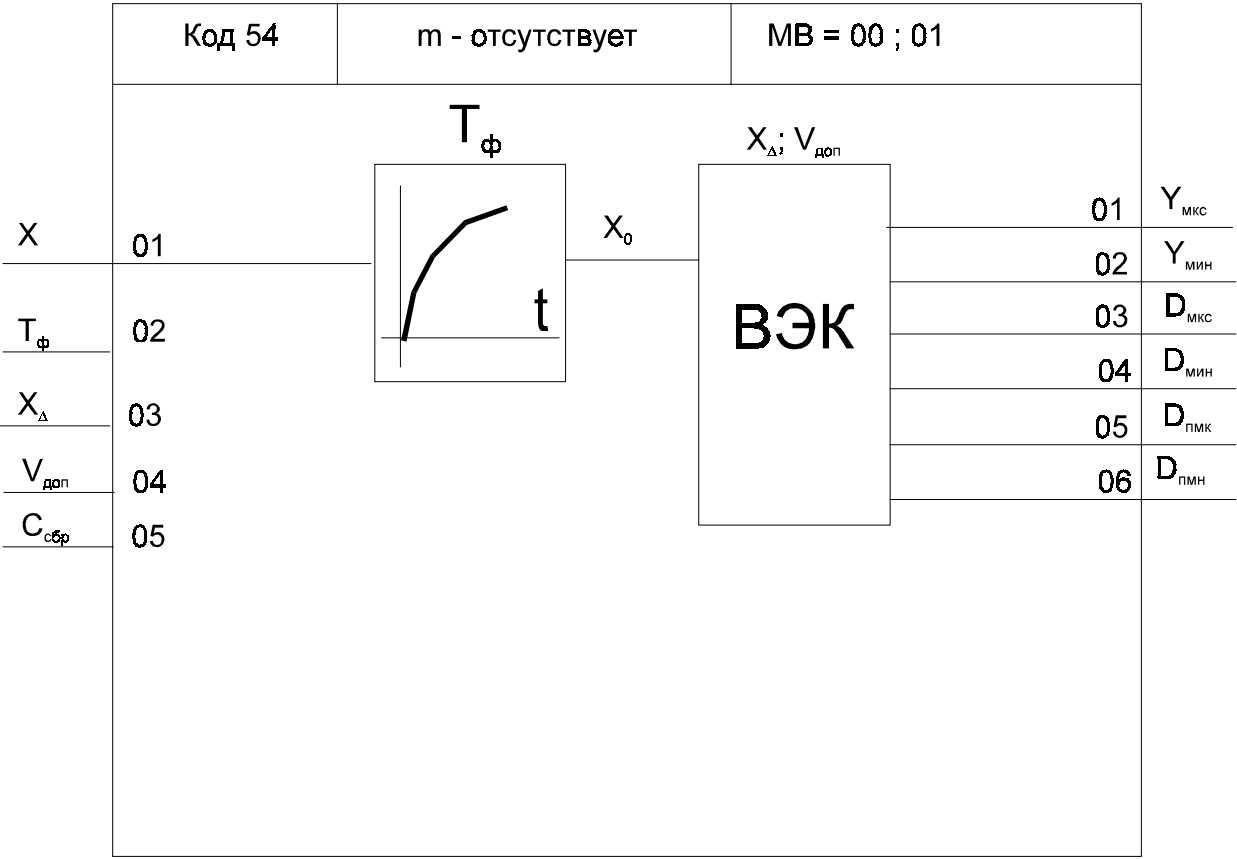
Влияние зоны нечувствительности и допустимой скорости



- 1 - экстремум не фиксируется ($\Delta < X_\Delta$);
- 2 - экстремум не фиксируется ($V > V_{доп}$);
- 3 - экстремум фиксируется ($\Delta > X_\Delta$; $V < V_{доп}$)

Рис. 46

Функциональная схема алгоритма
"Экстремум ЭКС"



ВЭК - узел выделения экстремума

Рис. 47

6.14. МСШ (55) - Масштабирование

Назначение

Алгоритм применяется для масштабирования нескольких (до 20) аналоговых сигналов.

Описание алгоритма

Алгоритм позволяет масштабировать m аналоговых сигналов, где $0 \leq m \leq 20$ задается модификатором (рис. 48). При $m=0$ алгоритм является "пустым". В каждом звене масштабирования выполняется соотношение:

$$Y_i = K_{m,i} * X_i, \quad (43)$$

где Y_i и X_i - соответственно выходной и входной сигналы; $K_{m,i}$ - масштабный коэффициент.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма МСШ

Таблица 49

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	1-й масштабируемый вход
02	$K_{m,1}$		1-й масштабируемый коэффициент
03	X_2		2-й масштабируемый вход
04	$K_{m,2}$		2-й масштабный коэффициент
.	.		.
.	.		.
.	.		.
2m-1	X_m		m-й масштабируемый вход
2m	$K_{m,m}$	Выходы	m- масштабный коэффициент
01	Y_1		1-й выход
02	Y_2		2-й выход
.	.		.
.	.		.
.	.		.
m	Y_m		m -й выход

**Функциональная схема алгоритма
"Масштабирование МСШ"**

	Код 55	m = 00 - 20	МВ - отсутствует	
X_1	01	$K_{M,1}$	01	Y_1
$K_{M,1}$	02			
X_2	03	$K_{M,2}$	02	Y_2
$K_{M,2}$	04		.	.
.	.	.	.	.
.	.	.	.	.
X_m	$2m-1$	$K_{M,m}$	m	Y_m
$K_{M,m}$	2m			

Рис. 48

7. Аналого-дискретные преобразования

7.1. ПЕР (57) - Переключатель с дискретным управлением

Назначение

Алгоритм представляет собой многополюсный переключатель аналоговых сигналов, положение которого определяется дискретными сигналами, поступающими на вход алгоритма. Алгоритм используется для выбора одного из нескольких (до 49) сигналов. Если на аналоговых входах алгоритма заданы константы, то алгоритм может использоваться для дискретной установки требуемой константы

Описание алгоритма

Функциональная схема алгоритма (рис. 49) представляет собой переключатель аналоговых сигналов. Если на всех дискретных входах C , управляющих положением переключателя, сигнал отсутствует (Лог. 0), выходной сигнал $Y=0$. Если на какой-либо из дискретных входов подается дискретный сигнал $C_i=1$, выход алгоритма Y подключается к одноименному (по номеру индекса) аналоговому входу X_i .

Если дискретные сигналы подаются одновременно на несколько входов, приоритетен вход с младшим номером.

Сигнал на выходе Y равен сигналу на выбранном входе. Числа на выходе N указывает номер выбранного входа.

Работа переключателя описывается следующей таблицей:

Таблица 50

Дискретные входы					Выходы	
C_1	C_2	C_3	...	C_m	Y	N
0	0	0	...	0	0	0
1	*	*	...	*	X_1	1
0	1	*		*	X_2	2
.	.	.		.	.	.
.	.	.		.	.	.
.	.	.		.	.	.
0	0	0		1	X_m	m

* - значение сигнала безразлично.

Число переключаемых входов алгоритма задается модификатором $0 \leq m \leq 49$. При изменении модификатора число входов изменяется парами: добавляется (или исключается) переключаемый вход и соответствующий ему дискретный вход. При $m=0$ алгоритм объявляется "пустым".

Режим работы

При наличии в отключенной цепочке следящих алгоритмов алгоритм ПЕР является инициатором режима отключения. Команда отключения формируется на всех невыбранных (отключенных) аналоговых входах. На этих же входах формируется значение начальных условий, равное текущему значению сигнала на выбранном входе (или, что то же самое, текущему значению сигнала Y).

При наличии во включенной цепочке следящих алгоритмов и при условии, что на каскадный выход Y поступает команда отключения с начальными условиями уд, эта команда вместе со значением Y_0 поступает на включенный вход и через него транслируется пред включенным алгоритмам. Условия работы отключенных входов при этом не изменяются.

Внешняя команда запрета, поступающая на выход Y , транслируется пред включенному алгоритму только через выбранный аналоговый вход. В остальном работа алгоритма при поступлении внешней команды запрета не изменяется.

В режиме отключения выход N продолжает указывать на положение переключателя.

Входы-выходы алгоритма ПЕР

Таблица 52

Входы-выходы			Назначение
№	Обозн	Вид	
01	X_1	Входы	1-й переключаемый вход (каскадный)
02	C_1		Команда выбора входа 1
03	X_2		2-й переключаемый вход (каскадный)
04	C_2		Команда выбора входа 2
.	.		.
.	.		.
.	.		.
2m-1	X_m	Входы	m-ый переключаемый вход (каскадный)
2m	C_m		Команда выбора входа m
01	Y		Основной выход (каскадный)

02	N	Порядковый номер выбранного входа
----	---	-----------------------------------

Функциональная схема алгоритма
"Переключатель с дискретным управлением ПЕР"

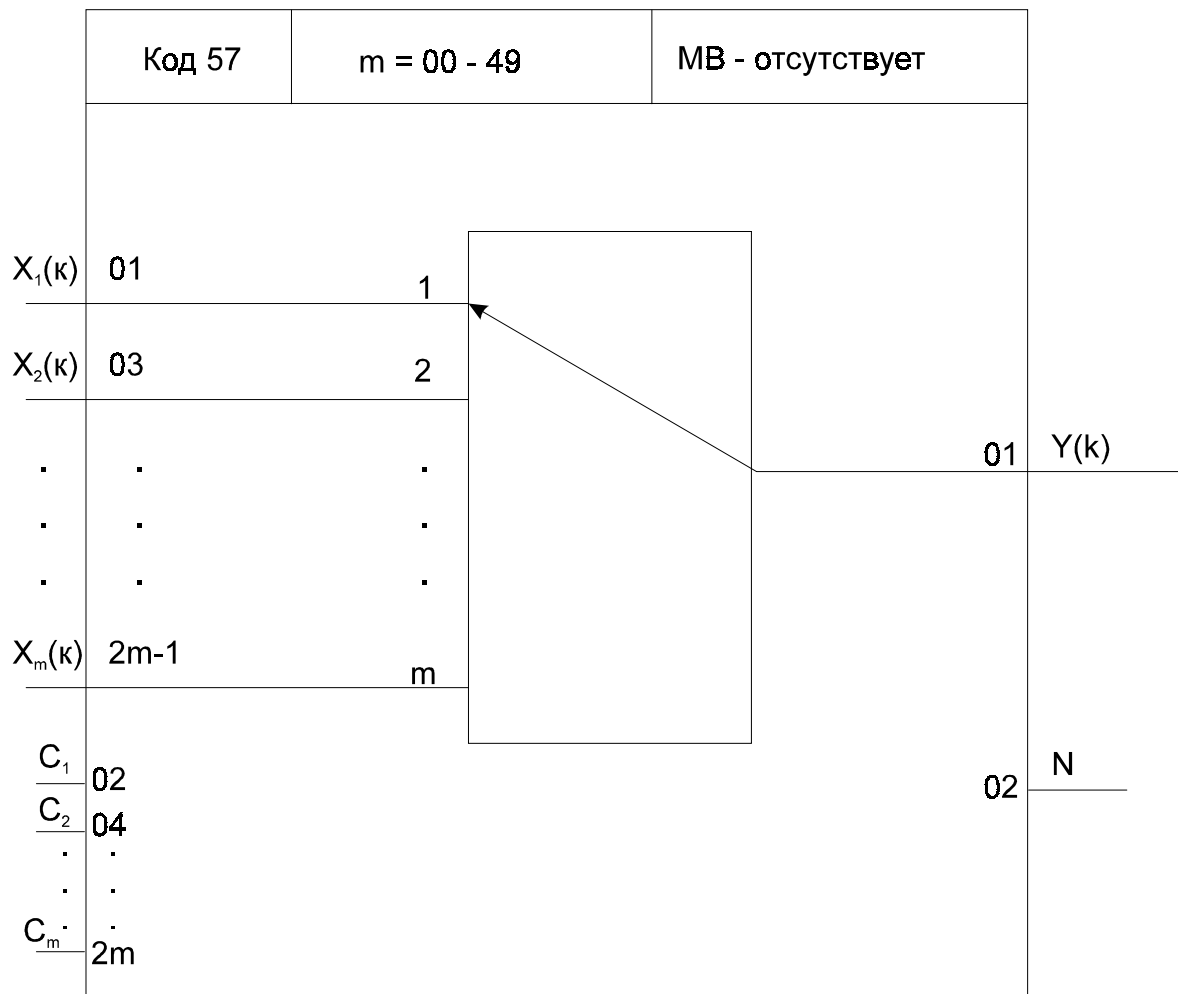


Рис. 49

7.2. ПЕН (58) - Переключение по номеру

Назначение

Алгоритм используется для переключения нескольких (до 98) сигналов. Положение переключателя определяется значением числа, поступающего на специальный вход алгоритма.

Описание алгоритма

Переключатель имеет $0 \leq m \leq 98$ положений, где m задается модификатором. На вход N подается число, определяющее номер подключенного входа (рис. 50).

Работа переключателя описывается таблицей:

Таблица 51

N	$N \leq 0$	$N=1$	$N=1$	...	$N \geq m$
Y	0	X_1	X_2	...	X_m

Особенности различных режимов работы переключателя ПЕН полностью соответствуют переключателю ПЕР (см. п. 7.1).

Входы-выходы алгоритма ПЕН

Таблица 51.1

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X_1	Входы	1-й переключаемый вход (каскадный)
02	X_2		2-й переключаемый вход (каскадный)
.	.		.
.	.		.
.	.		.
m	X_m		m -ый переключаемый вход (каскадный)
$m+1$	N		Управление положением переключателя
01	Y	Выход	Выход (каскадный)

Функциональная схема алгоритма
"Переключатель по номеру ПЕН"

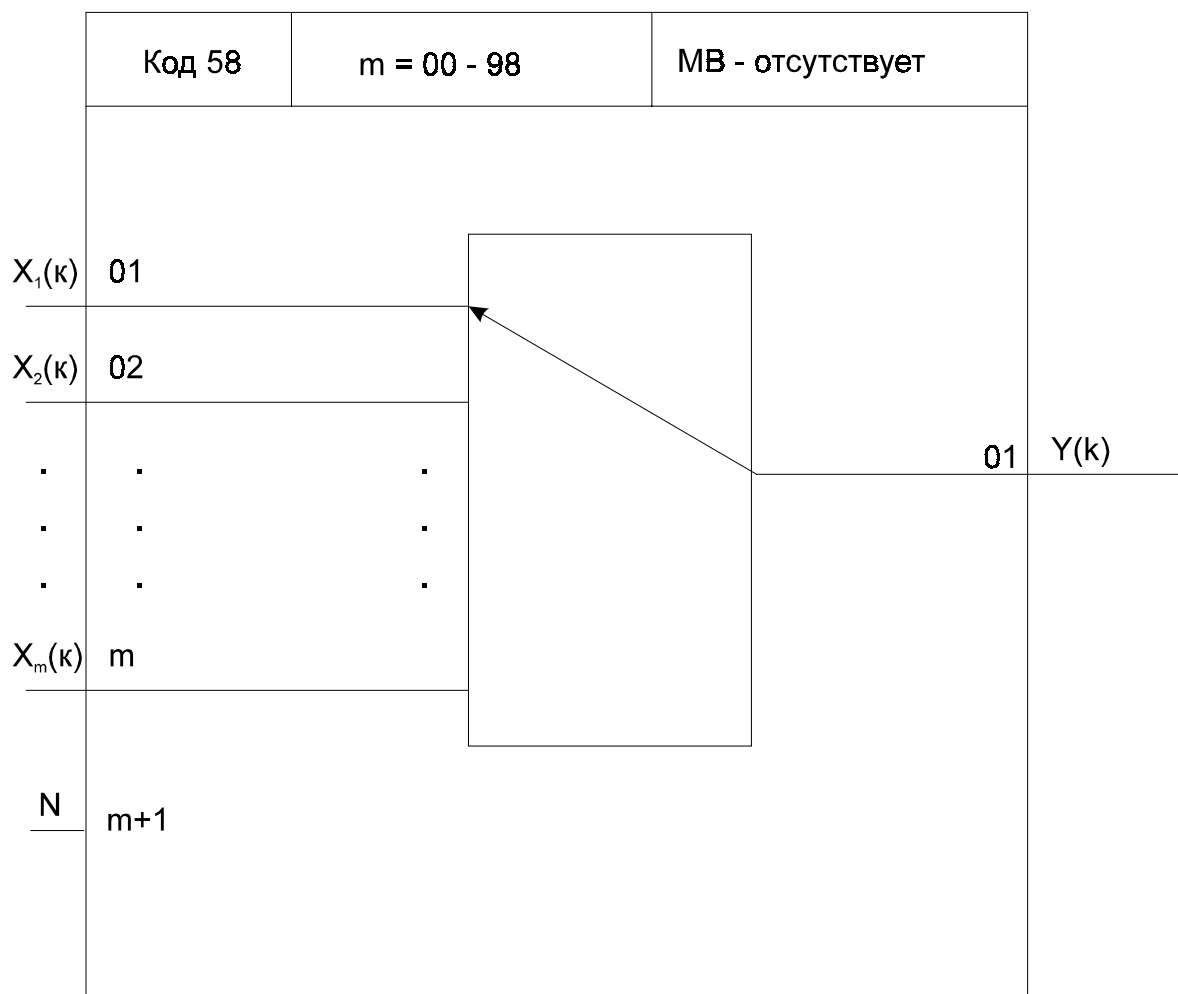


Рис. 50

7.3. ПОР (59) - Пороговый элемент

Назначение

Алгоритм применяется для контроля за выходом сигнала или разности двух сигналов из ограниченной справа области допустимых значений. Каждый алгоритм может содержать несколько (до 20) независимых пороговых элементов.

Описание алгоритма

Алгоритм содержит m независимых ячеек, причем $0 \leq m \leq 20$ и задается модификатором. Каждая ячейка содержит звено сумматора и звено порогового элемента (рис. 51).

Звено сумматора выделяет разность двух сигналов:

$$Z = X_1 - X_2 \quad (44)$$

Разностный сигнал подается на звено порогового элемента с порогом срабатывания X_{cp} и гистерезисом X_{Δ} .

Звено порогового элемента срабатывает, когда $Z \geq X_{cp}$, при этом появляется дискретный сигнал на выходе звена порогового элемента. Логика работы каждой ячейки описывается таблицей:

Таблица 52

Z	D
$Z < X_{cp} - X_{\Delta}$	0
$Z \geq X_{cp}$	1
$X_{cp} - X_{\Delta} \leq Z < X_{cp}$	$D(i-1)$

Здесь $D(i-1)$ - предыдущее значение выходного сигнала.

На настроечных входах X_{cp} , X_{Δ} задается соответственно порог срабатывания и гистерезис. Значение $X_{\Delta} < 0$ воспринимается алгоритмом как $X_{\Delta} = 0$.

Выход D_0 алгоритма является объединением по ИЛИ выходов всех ячеек алгоритма.

Алгоритм не имеет каскадных входов-выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ПОР

Таблица 53

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$X_{1,1}$	Входы	Первый вход первой ячейки
02	$X_{2,1}$		Второй вход первой ячейки
03	$K_{cp,1}$		Порог срабатывания первой ячейки
04	$X_{\Delta,1}$		Гистерезис первой ячейки
05	$X_{1,2}$		Первый вход второй ячейки
06	$X_{2,2}$		Второй вход второй ячейки
07	$X_{cp,2}$		Порог срабатывания второй ячейки
08	$X_{\Delta,2}$		Гистерезис второй ячейки
.	.		.
.	.		.
.	.		.
4m-3	$X_{1,m}$		Первый вход m -ой ячейки
4m-2	$X_{2,m}$		Второй вход m -ой ячейки
4m-1	$X_{cp,m}$		Порог срабатывания m-ой ячейки
4m	$X_{\Delta,m}$		Гистерезис m -ой ячейки
01	D_1	Выходы	Выход первой ячейки
02	D_2		Выход второй ячейки
.	.		.
.	.		.
.	.		.
m	D_m		Выход m-ой ячейки
m+1	D_0		Групповой выход

Функциональная схема алгоритма
"Пороговый элемент ПОР"

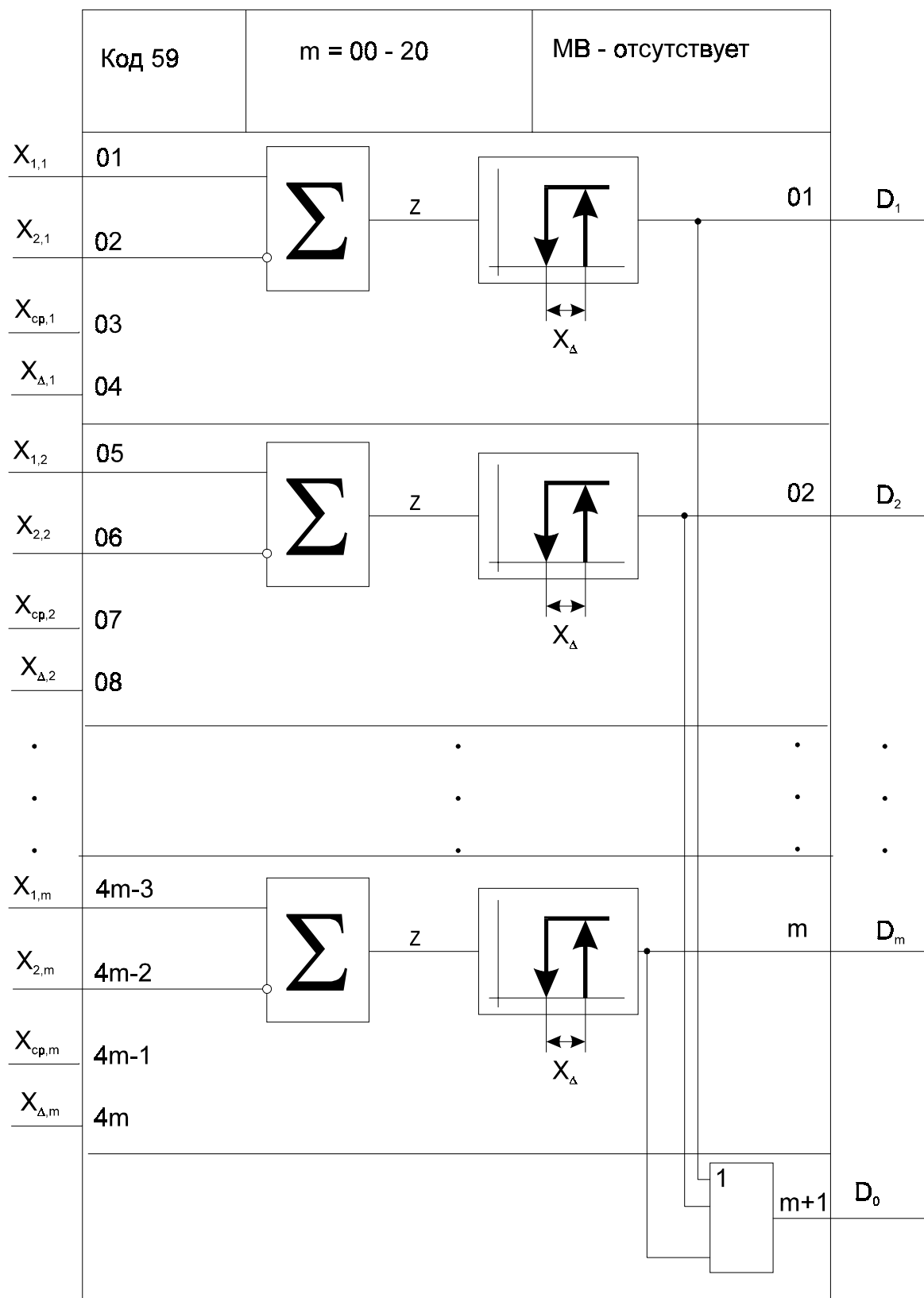


Рис. 51

7.4 НОР (60) - Нуль-орган

Назначение

Нуль-орган используется для контроля за выходом сигнала или разности двух сигналов из ограниченной слева и справа области допустимых значений. Каждый алгоритм может содержать несколько (до 10) независимых нуль-органов.

Описание алгоритма

Алгоритм содержит m независимых ячеек, причем $0 \leq m \leq 10$ и задается модификатором (рис 52).

Каждая ячейка содержит звено сумматора и звено нуль-органа

Звено сумматора выделяет разность двух сигналов:

$$Z = X_1 - X_2 \quad (45)$$

Разностный сигнал подается на звено нуль-органа, имеющее два порога срабатывания X_6 и X_M . Нуль-орган срабатывает, когда $Z \geq X_6$ или $Z \leq X_M$, при этом появляется дискретный сигнал соответственно на выходе D_1 или D_2 . Оба пороговых элемента имеют одинаковый гистерезис.

Логика работы каждой ячейки нуль-органа описывается таблицей:

Таблица 54

Z	D₁	D₂
$Z < X_6 - X_\Delta$	0	*
$Z \geq X_6$	1	*
$X_6 - X_\Delta \leq Z < X_6$	$D_1(i-1)$	*
$Z > X_M + X_\Delta$	*	0
$Z \leq X_M$	*	1
$X_M + X_\Delta \geq Z > X_M$	*	$D_2(i-1)$

* - выходной сигнал не зависит от данного условия, $D(i-1)$ - предыдущее значение выходного сигнала.

На настроечных входах X_6 , X_M , X_Δ задаются пороги срабатывания и гистерезис нуль-органа. Значение $X_\Delta < 0$ воспринимаются алгоритмом как $X_\Delta = 0$.

Выход D_0 алгоритма является объединением по ИЛИ выходов всех нуль-органов.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма НОР

Таблица 55

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$X_{1,1}$	Входы	Первый вход первой ячейки
02	$X_{2,1}$		Второй вход первой ячейки
03	$X_{6,1}$		Первый порог первой ячейки
04	$X_{M,1}$		Второй порог первой ячейки
05	$X_{\Delta,1}$		Гистерезис первой ячейки
06	$X_{1,2}$		Первый вход второй ячейки
07	$X_{2,2}$		Второй вход второй ячейки
08	$X_{6,2}$		Первый порог второй ячейки
09	$X_{M,2}$		Второй порог второй ячейки
10	$X_{\Delta,2}$		Гистерезис второй ячейки
.	.		.
.	.		.
.	.		.
5m-4	$X_{1,m}$		Первый вход m -ой ячейки
5m-3	$X_{2,m}$		Второй вход m -ой ячейки
5m-2	$X_{6,m}$		Первый порог m -ой ячейки
5m-1	$X_{M,m}$		Второй порог m -ой ячейки
5m	X_{Δ}		Гистерезис m -ой ячейки
01	$D_{1,1}$	Выходы	Первый выход первой ячейки
02	$D_{2,1}$		Второй выход первой ячейки
03	$D_{1,2}$		Первый выход второй ячейки
04	$D_{2,2}$		Второй выход второй ячейки
.	.		.
.	.		.
.	.		.
2m-1	$D_{1,m}$		Первый выход m-ой ячейки

2m	$D_{2,m}$	Выходы	Второй выход m -ой ячейки
2m+1	D_0		Групповой выход

Функциональная схема алгоритма
"Нуль-орган НОР"

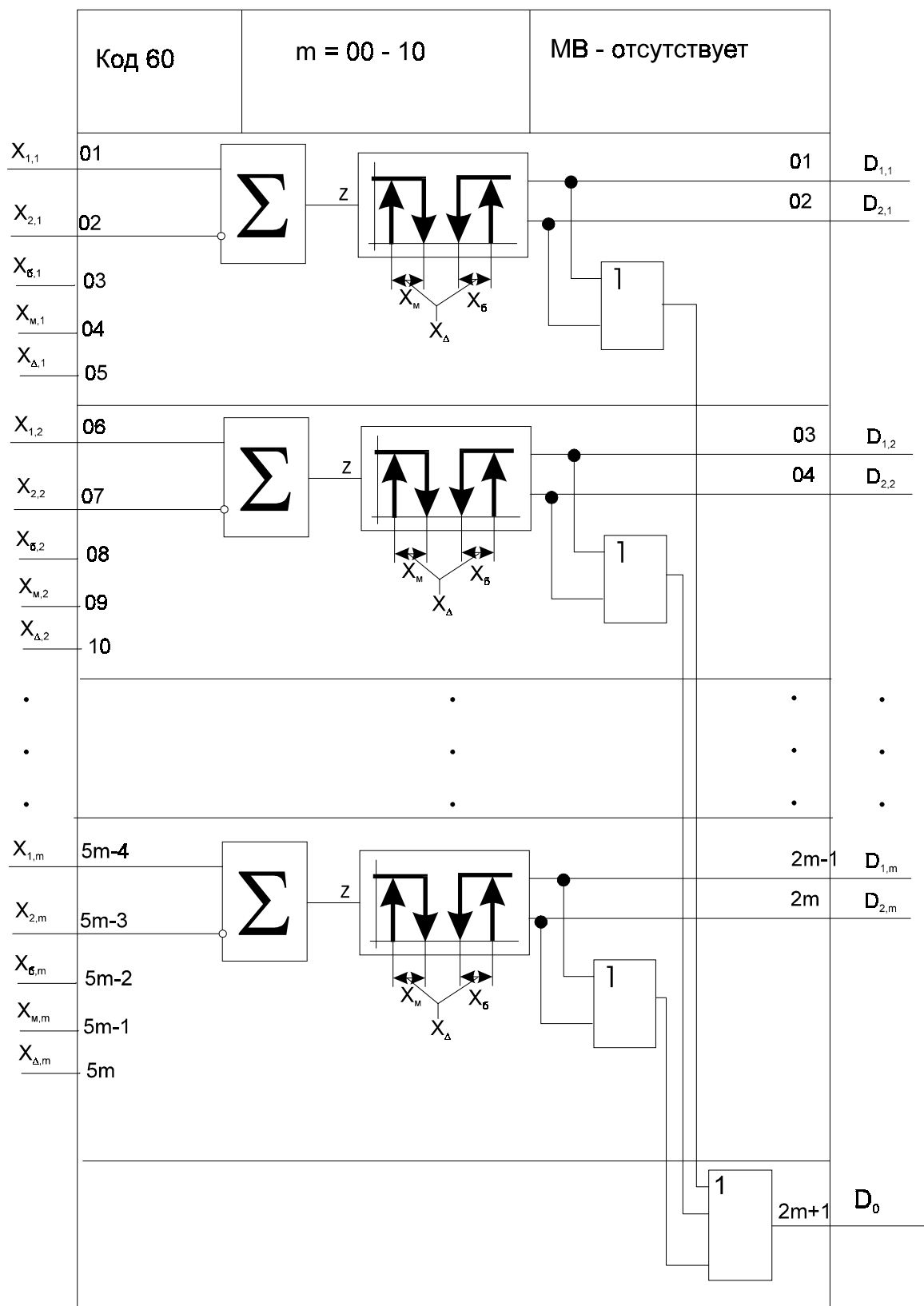


Рис. 52

7.5 ИМП (61) - Импульсатор

Назначение

Алгоритм используется для периодического включения и выключения нагрузки в том случае, когда скважность включения должна быть пропорциональна непрерывному управляющему сигналу.

Описание алгоритма

Алгоритм содержит широтно-импульсный (ШИМ) модулятор с заданным периодом и меняющейся длительностью включения (рис. 54).

Если входной сигнал $X > 0$, то импульсы формируются на дискретном выходе D_6 (больше), если $X < 0$, то на выходе D_m (меньше).

Диаграмма формируемых импульсов представлена на рис. 53. Период следования импульсов задается сигналом на настроечном входе алгоритма T . Длительность выходных импульсов

$$t_n = |X| \cdot T / 100. \quad (46)$$

Если $X = 0$, то $t_n = 0$; если $|X| > 100\%$, то $t_n = T$ (т.е. пауза отсутствует).

Если вычисленное по формуле (46) значение $t_n < T_0$, где T_0 - время цикла контроллера, то реализуемое значение $t_n = T_0$, причем часть импульсов при этом "пропадает", так, что в среднем сохраняется пропорциональность зависимости между скважностью и входным сигналом.

Если установленное значение периода $T < T_0$, то реализуемое значение $t_n = 0$ (т.е. импульсы не формируются).

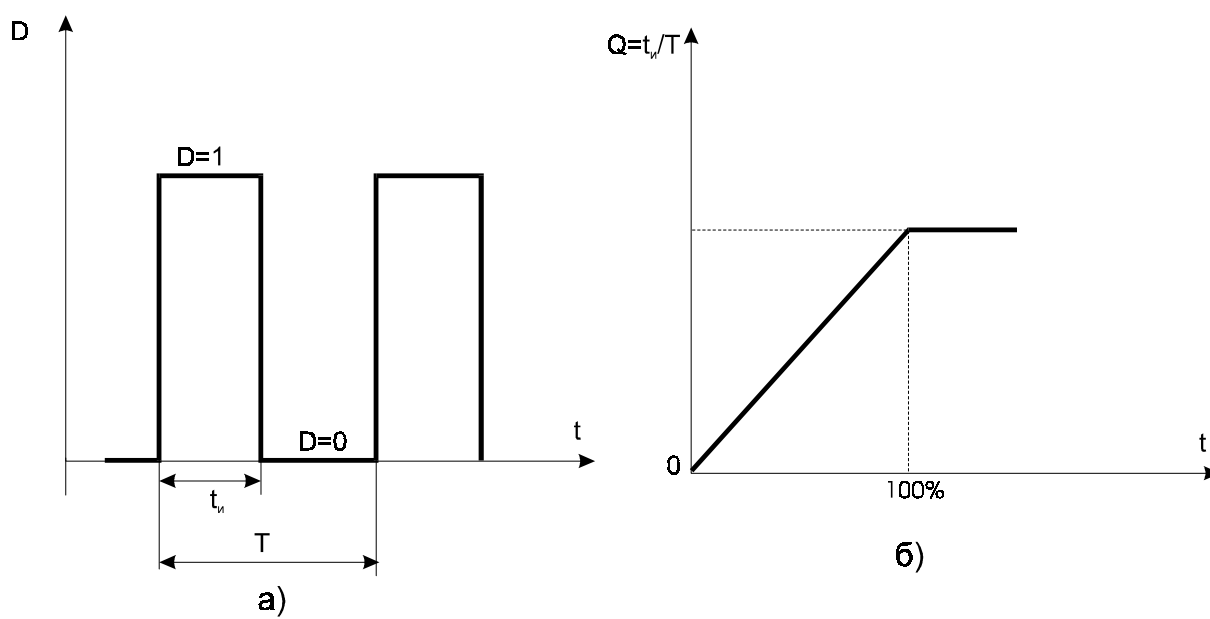
Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ИМП

Таблица 56

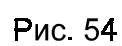
Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Входной сигнал
02	T		Период следования импульсов
01	D_6	Выходы	Выход "больше"
02	D_m		Выход "меньше"

Работа импульсатора



- а) диаграмма выходных импульсов;
б) статическая характеристика

Рис. 53



7.6. ЗАИ (62) - Запрет изменения

Назначение

Алгоритм применяется для запрета увеличения или уменьшения сигнала, например, при действии технологических защит. В частности, запрет может блокировать изменение сигнала задания или выходного сигнала аналогового регулятора.

Описание алгоритма

Алгоритм содержит узел запрета, сумматор и узел динамической балансировки (рис. 56).

Узел запрета реагирует на дискретные сигналы запрета в направлении увеличения (C_v - вверх) или уменьшения (C_n - вниз).

Узел балансировки включается, если дискретный сигнал $C_v=1$; в противном случае балансировка отсутствует.

Помимо основного выхода Y алгоритм содержит два дискретных выхода D_v и D_n , сигнализирующих о том, что алгоритм работает в режиме запрета.

При отсутствии балансировки работа алгоритма поясняется графиком на рис. 55. Если команды запрета отсутствуют, то $Y = X$. Если поступила команда C_v и $X \geq 4$, то алгоритм переходит в режим "запрета вверх", при этом $Y = Y_1 = \text{const}$ и $D_v = 1$, где Y_1 - значение Y в момент поступления команды запрета. Этот режим сохраняется до тех пор, пока входной сигнал не уменьшится до значения $X < Y_1$, после чего режим запрета снимается и вновь $Y = X$.

Аналогично, но с реакцией на противоположную тенденцию изменения X , работает алгоритм в режиме "запрета вниз".

Работа алгоритма, при отсутствии балансировки описывается таблицей:

Таблица 57

Команда запрета		X	Y	D _v	D _n	Режим запрета
C _v	C _n					
0	0	*	Y=X	0	0	-
1	0	X ≥ Y ₁	Y=Y ₁ =const	1	0	+
1	0	X < Y ₁	Y=X	0	0	-
0	1	X ≤ Y ₁	Y=Y ₁ =const	0	1	+
0	1	X > Y ₁	Y=X	0	0	-
1	1	X=Y ₁	Y=Y ₁ =const	1	1	+
1	1	X > Y ₁	Y=Y ₁ =const	1	0	+
1	1	X < Y ₁	Y=Y ₁ =const	0	1	+

В этой таблице:

* - значение сигнала безразлично;

Y - значение Y в момент прихода команды запрета.

Если балансировка включена, работа алгоритма изменяется (рис.55).

Если команды запрета отсутствуют, то

$$Y = X + Y_6 \quad (47)$$

$$\text{где } Y_6 = \sigma - V_6 \cdot t \quad (47a)$$

В последнем соотношении Y_6 - сигнал балансировки;

V_6 - скорость балансировки; t - текущее время; σ - начальное значение сигнала балансировки (при $t=0$).

Таким образом, от начала балансировки (т.е. от $t=0$) сигнал V_6 линейно изменяется с постоянной настраиваемой скоростью V_6 , приближаясь к нулю. Когда $Y_6=0$, балансировка заканчивается, после чего сигнал Y_6 остается равным нулю и $Y=X$.

Если поступила команда $C_v=1$ (запрет вверх) и X увеличивается, то алгоритм переходит в режим "запрет вверх", выходной сигнал $Y=Y_1=\text{const}$ и $D_v=1$, т.е. на этом участке алгоритм с балансировкой работает так же, как без балансировки. При этом узел балансировки переходит в режим слежения за разностью сигналов Y и X , т.е. $Y_6=Y-X$.

Как только сигнал X начинает уменьшаться, начинает уменьшаться и сигнал Y , режим запрета отменяется (несмотря на наличие команды запрета и то, что $X > Y$), сигнал балансировки Y_6 замораживается и между сигналами X и Y сохраняется постоянная разница σ , равная разнице между этими сигналами на момент, когда сигнал X начал уменьшаться. После того, как будет снята команда запрета (т.е. $C_v=0$), сигнал балансировки начинает уменьшаться до нуля с постоянной скоростью V_6 согласно выражению (47a).

После того, как Y_6 станет равным нулю, дальнейшее его изменение прекращается и алгоритм возвращается к своему исходному состоянию, когда $Y=X$.

Аналогично, но с реакцией на противоположную тенденцию изменения X , работает алгоритм при наличии балансировки в режиме "запрет вниз".

Работа алгоритма при наличии балансировки описывается таблицей:

Таблица 58

Команда запрета		X	Y	D _в	D _н	Режим запрета
C _в	C _н					
0	0	*	$Y=X+Y_6$	0	0	-
1	0	$\Delta X \geq 0$	$Y=Y_1=\text{const}$	1	0	+
1	0	$\Delta X < 0$	$Y=X$	0	0	-
0	1	$\Delta X \leq 0$	$Y=Y_1=\text{const}$	0	1	+
0	1	$\Delta X > 0$	$Y=X$	0	0	-
1	1	$\Delta X = 0$	$Y=Y_1=\text{const}$	1	1	+
1	1	$\Delta X > 0$	$Y=Y_1=\text{const}$	1	0	+
1	1	$\Delta X < 0$	$Y=Y_1=\text{const}$	0	1	+

В этой таблице * и Y_1 - то же, что в табл. 57, и, кроме того, $\sigma = Y_1 - X_1$, где X_1 - значение X в момент, когда сигнал X начал уменьшаться (для запрета вверх) или увеличиваться (для запрета вниз).

Режимы работы

При наличии в цепочке следящих алгоритмов алгоритм ЗНИ является инициатором режима запрета. Когда алгоритм переходит в этот режим, на каскадном входе X формируется команда запрета, которая транслируется предвключенному алгоритму. Команда запрета может поступить также извне на каскадный выход алгоритма Y. В этом случае алгоритм работает так же, как если бы эта команда поступила на один из входов алгоритма

C_в ; C_н.

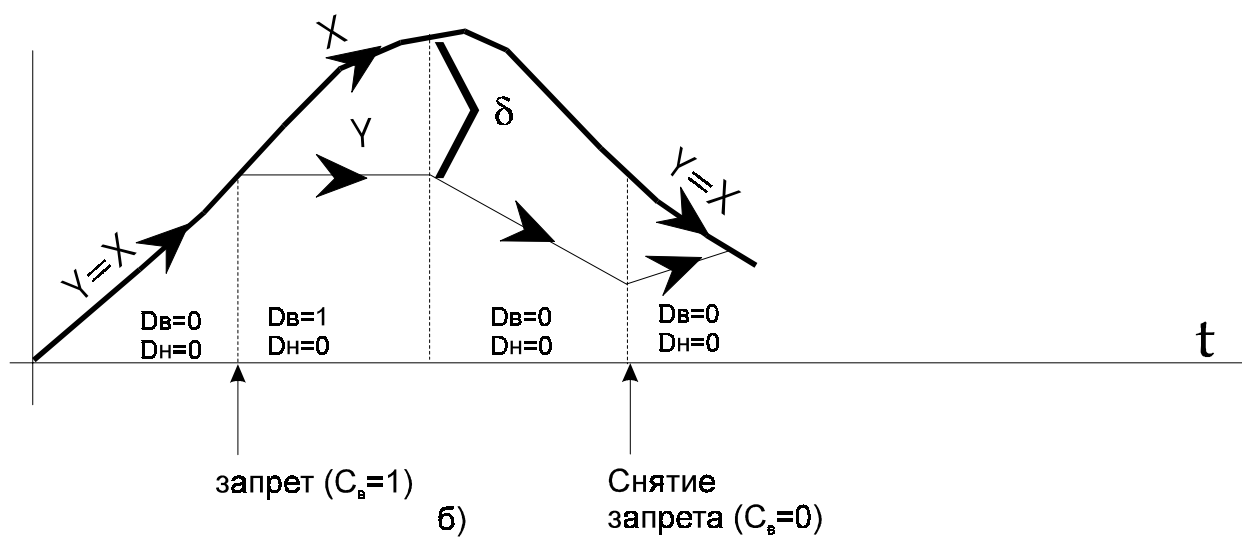
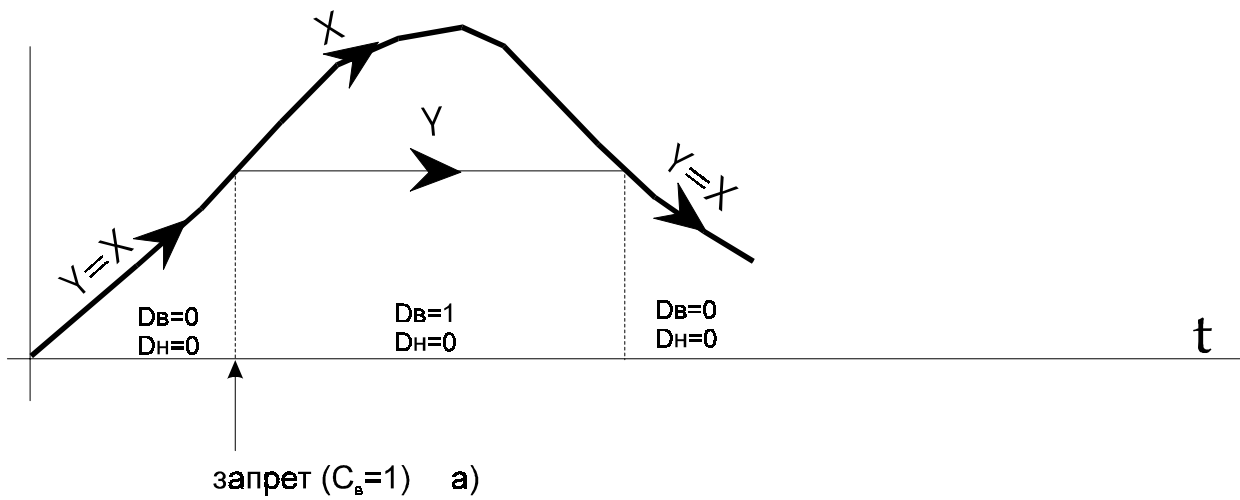
Алгоритм не относится к группе следящих, но если на каскадный выход Y приходит команда отключения с значением начальных условий Y_0 , эта команда вместе с Y_0 через каскадный вход X транслируется предвключенному алгоритму. Динамическая балансировка при этом отсутствует, дискретные выходы не вырабатываются ($D_v=D_n=0$).

Входы-выходы алгоритма ЗАИ

Таблица 59

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	C _в		Команда запрета вверх
03	C _н		Команда запрета вниз
04	C _б		Включение балансировки
05	V _б		Скорость балансировки
01	Y	Выходы	Основной выход (каскадный)
02	D _в		Режим запрета вверх
03	D _н		Режим запрета вниз

Работа алгоритма ЗАИ



- а) - балансировка не включена
 б) - балансировка включена

Рис. 55

Функциональная схема алгоритма
"Запрет изменения ЗАИ"

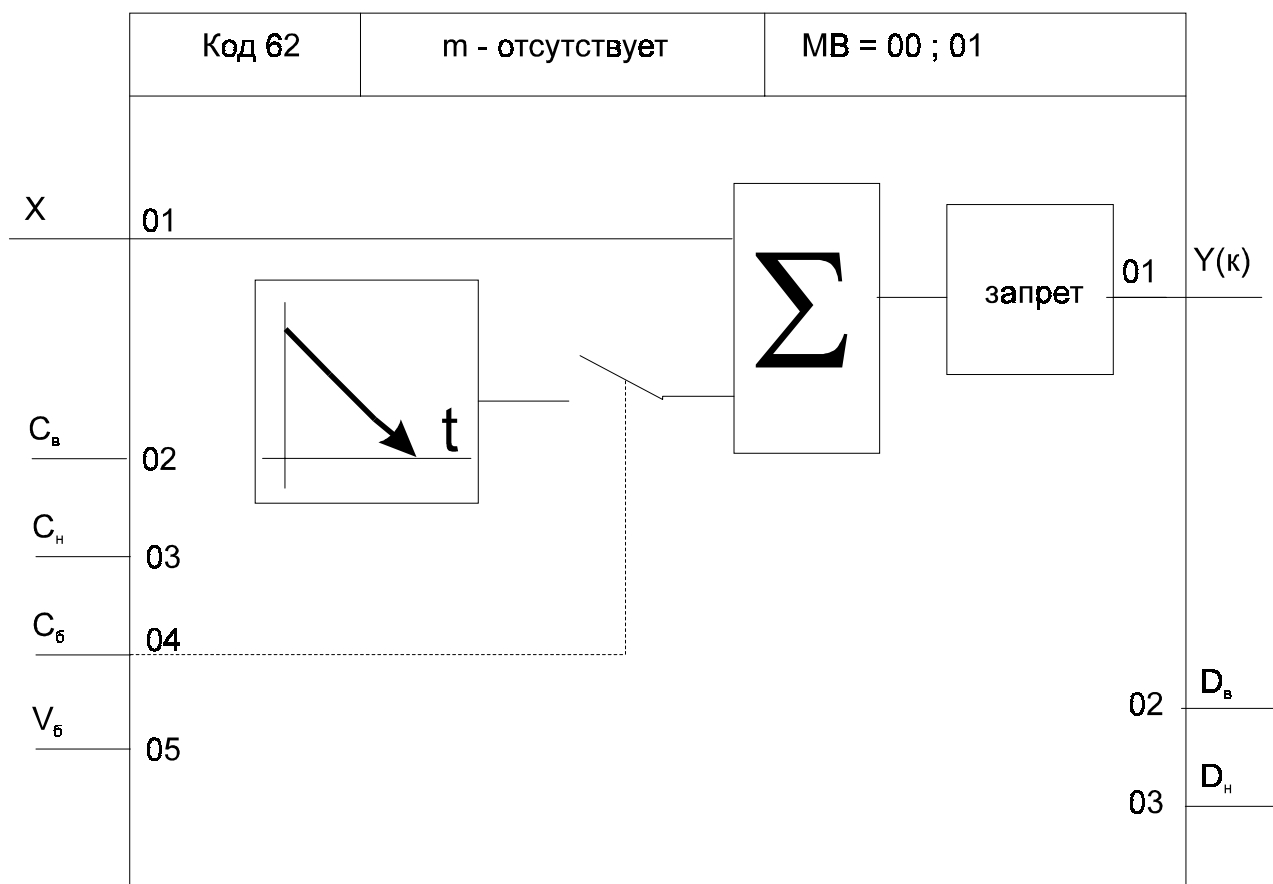


Рис. 56

7.7. ЗАЗ (63) - Запрет знака

Назначение

Алгоритм применяется для запрета перехода сигнала в область положительных или отрицательных значений. В частности, в сочетании с алгоритмом импульсного регулирования алгоритм ЗАЗ используется для запрета перемещения исполнительного механизма в направлении "больше" или "меньше".

Описание алгоритма

Алгоритм содержит узел запрета, управляемый входными дискретными сигналами C_+ и C_- . При действии команды C_+ -запрещается переход выходного сигнала Y в область положительных значений, команда C_- - запрещает изменение Y в область отрицательных значений (рис. 57):

Работа алгоритма описывается следующей таблицей:

Таблица 60

Команда запрета		X	Y	D ₊	D ₋	Режим запрета
C ₊	C ₋					
0	0	*	Y=X	0	0	нет
1	0	X>=0	Y=0	1	0	есть
1	0	X<0	Y=X	0	0	нет
0	1	X<=0	Y=0	0	1	есть
0	1	X>0	Y=X	0	0	нет
1	1	*	Y=0	1	1	есть

* - значение сигнала безразлично

Режимы работы алгоритма ЗАЗ полностью соответствуют режимам работы алгоритма ЗАИ.

Входы-выходы алгоритма ЗАЗ

Таблица 61

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	C ₊		Команда запрета "плюс"
03	C ₋		Команда запрета "минус"
01	Y	Выходы	Основной выход (каскадный)
02	D ₊		Режим запрета "плюс"
03	D ₋		Режим запрета "минус"

Функциональная схема алгоритма
"Запрет знака ЗАЗ"

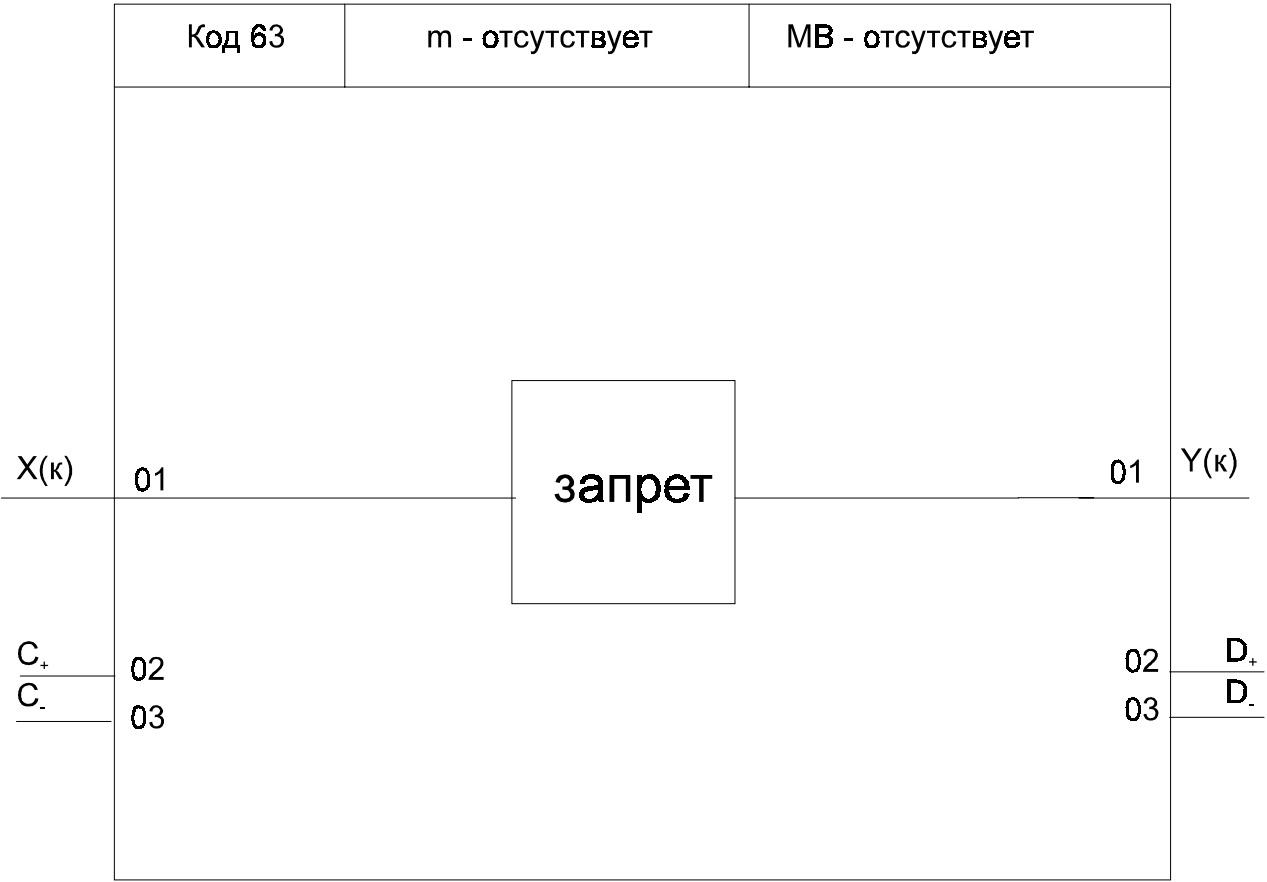


Рис. 57

7.8. СЛЗ (64) - Слежение-запоминание

Назначение

Алгоритм используется для слежения и запоминания нескольких (до 20) аналоговых сигналов. Команда перехода из режима слежения в режим запоминания является общей для всех сигналов.

Описание алгоритма

Алгоритм содержит m ячеек памяти, где $0 \leq m \leq 20$ и задается модификатором (рис. 58). На входы X_i алгоритма подаются аналоговые сигналы, которые нужно запомнить. На вход $C_{сл}$ подается команда слежения.

До тех пор, пока $C_{сл} = 1$, входные сигналы X_i передаются на выходы Y_i , т.е. выход "следит" за входом. При $C_{сл} = 0$ слежение блокируется, текущие значения Y_i запоминаются и перестают зависеть от входных сигналов X_i .

Выходной сигнал $D_{сл} = C_{сл}$, т.е. повторяет входной управляющий сигнал.

Работа алгоритма описывается следующей таблицей:

Таблица 62

$C_{сл}$	Y_i	$D_{сл}$
1	$Y_i = X_i$	1
0	$Y_i = X_{i,0}$	0

В этой таблице $X_{i,0}$ - значение X_i в момент снятия команды $C_{сл}$

Алгоритм не содержит каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма СЛЗ

Таблица 63

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{сл}$	Входы	Команда слежения
02	X_1		1-й сигнал
03	X_2		2-й сигнал
•	•		•
•	•		•
•	•		•
m+1	X_m		m -й сигнал
01	$D_{сл}$	Выходы	Признак слежения
02	Y_1		1-й сигнал
03	Y_2		2-й сигнал
•	•		•
•	•		•
•	•		•
m+1	Y_m		m -й сигнал

Функциональная схема алгоритма
"Слежение-запоминание СЛЗ"

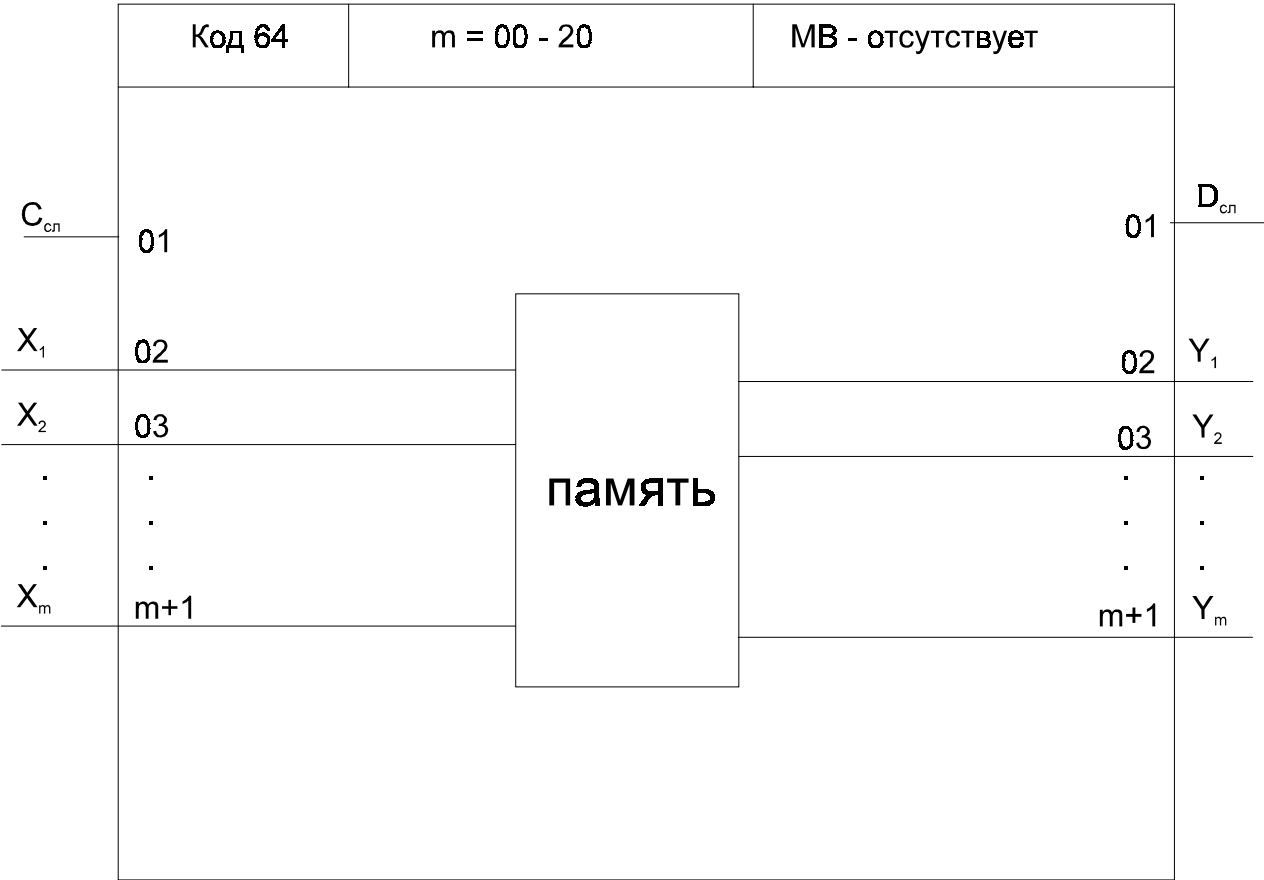


Рис. 58

7.9. ЗПМ (65) - Запоминание

Назначение

Алгоритм используется для запоминания нескольких (до 20) аналоговых сигналов. Запоминание осуществляется по переднему фронту общей для всех сигналов команды записи.

Описание алгоритма

Алгоритм содержит m ячеек памяти, где $0 \leq m \leq 20$ и задается модификатором (рис. 59). На входы X_i алгоритма подаются аналоговые сигналы, которые нужно запомнить. На вход $C_{\text{зап}}$ подается команда запоминания.

Запись осуществляется по переднему фронту сигнала $C_{\text{зап}}$, т.е. в момент перехода $C_{\text{зап}}$ из состояния логического 0 в состояние логической 1. В этот момент текущие значения входных сигналов X_i запоминаются. Запомненные значения остаются неизменными вплоть до прихода нового фронта сигнала $C_{\text{зап}}$.

Состояние ячеек памяти передается на выходы Y_i алгоритма, т.е. сигналы Y_i соответствуют запомненным значениям сигналов X_i .

В момент запоминания на дискретном выходе $D_{\text{зап}}$ формируется одиночный импульс $D_{\text{зап}} = 1$ длительно-стью в один цикл работы контроллера (т.е. на время от 0,2 до 2 с). Остальное время $D_{\text{зап}} = 0$.

Алгоритм не имеет каскадных входов и выходов и блокирует процедуру обратного счета.

Входы-выходы алгоритма ЗПМ

Таблица 64

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{\text{зап}}$	Входы	Команда запоминания (передний фронт)
02	X_1		1-й сигнал
03	X_2		2-й сигнал
.	.		.
.	.		.
.	.		.
m+1	X_m		m-й сигнал
01	$D_{\text{зап}}$	Выходы	Признак запоминания
02	Y_1		1-й сигнал
03	Y_2		2-й сигнал
.	.		.
.	.		.
.	.		.
m+1	Y_m		m -й сигнал

Функциональная схема алгоритма
"Запоминание ЗПМ"

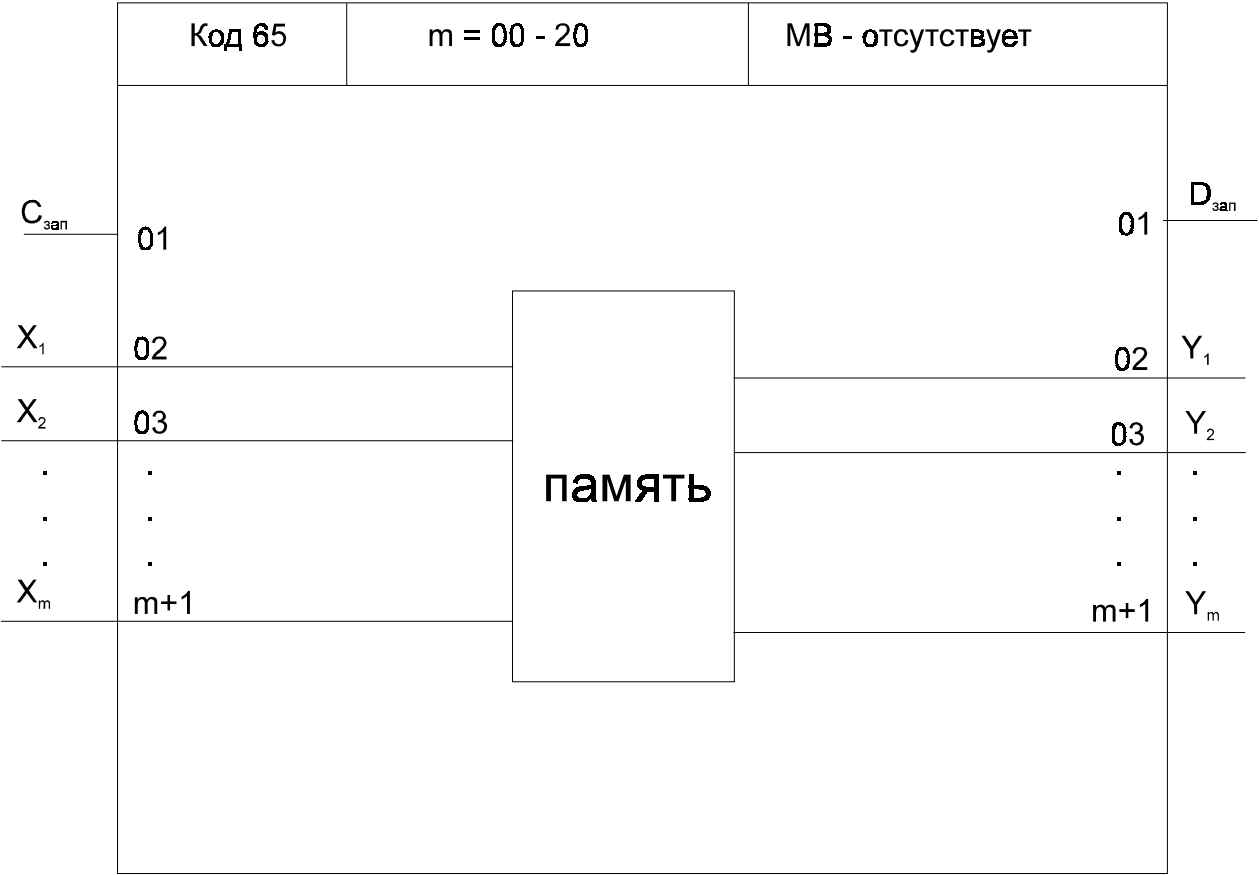


Рис. 59

7.10. БОС (66) - Блокировка обратного счета

Назначение

Алгоритм применяется при необходимости блокировать процедуру обратного счета, связанную с командами отключения и (или) запрета.

Описание алгоритма

Если на выход Y алгоритма не поступают команды отключения и запрета, алгоритм без изменения передает входной сигнал X на выход Y , т.е. $Y = X$ (рис. 60).

Работа алгоритма при отключении и запрете зависит от состояния сигналов на дискретных входах блокировки отключения $C_{отк}$ и блокировки запрета $C_{зап}$.

Если $C_{отк} = C_{зап} = 0$, то команды отключения и запрета транслируются через вход X предвключенному алгоритму, при этом $X = Y_0$, где Y_0 - значение начальных условий, поступающее на выход Y .

При $C_{отк}=1$ и $C_{зап}=0$ команда отключения, поступившая на выход Y_0 блокируется алгоритмом, и на вход X не проходит. Команда запрета как и раньше транслируется через вход X . Если $C_{отк}=0$ и $C_{зап}=1$ блокируется запрета, а команда отключения транслируется через вход X . Если $C_{отк} = C_{зап} = 1$, то блокируется, как команда отключения, так и запрета.

Входы-выходы алгоритма БОС

Таблица 65

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Входы	Основной вход (каскадный)
02	$C_{отк}$		Команда блокировки отключения
03	$C_{зап}$		Команда блокировки запрета
01	Y	Выход	Выход (каскадный)

Функциональная схема алгоритма
"Блокировка обратного счета БОС"

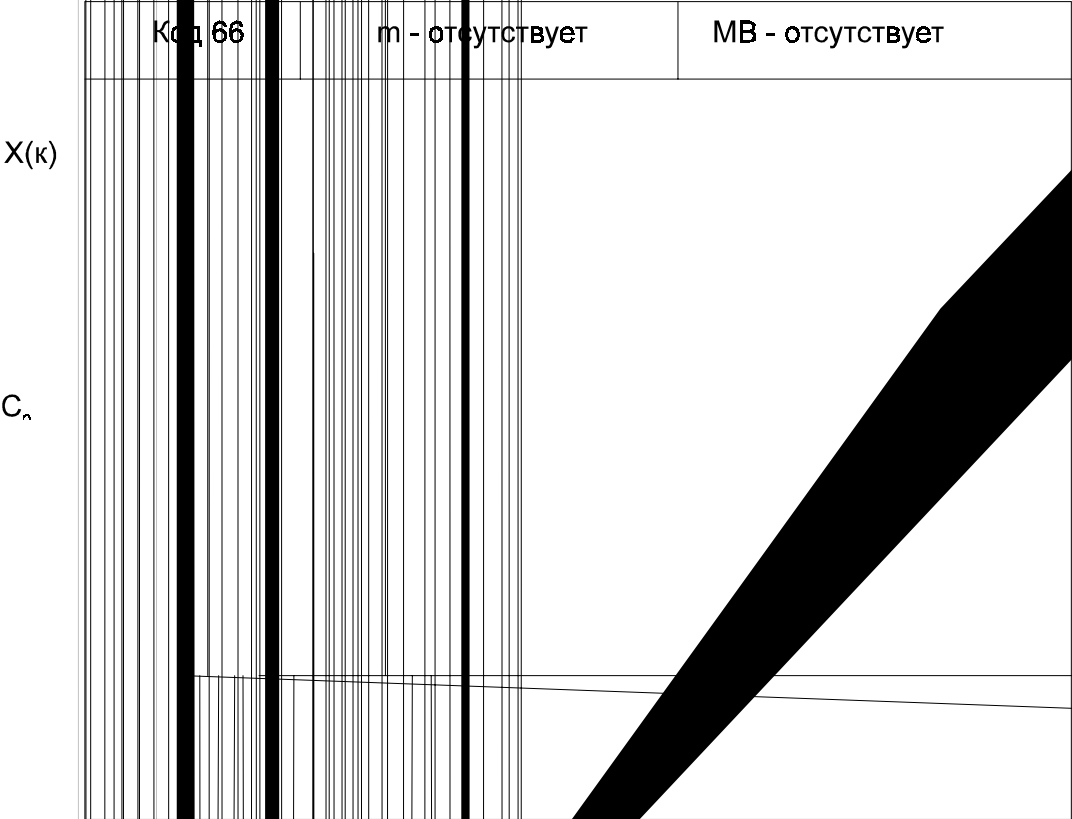


Рис. 60

7.11. BOT (67) - Выделение отключения

Назначение

Алгоритм применяется для того, чтобы выделить признак, свидетельствующий об отключении цепочки алгоритмов командой обратного счета.

Описание алгоритма

Если на каскадный выход Y алгоритма не приходит команда отключения, то входной сигнал X передается на выход Y без изменения, т.е. $Y=X$. При этом $D_{отк} = 0$ (рис. 61).

Если на выход Y поступает команда отключения, она вместе с значением начальных условий Y_0 через каскадный вход X транслируется предвключенному алгоритму, т.е. $X=Y_0$. При этом $D_{отк}=1$.

Команда запрета, поступившая на выход Y , не влияет на работу алгоритма, однако она через вход X транслируется предвключенному алгоритму.

Входы-выходы алгоритма BOT

Таблица 66

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Вход	Вход (каскадный)
01	Y	Выходы	Основной выход (каскадный)
02	$D_{отк}$		Признак отключения

Функциональная схема алгоритма
"Выделение отключения ВОТ"

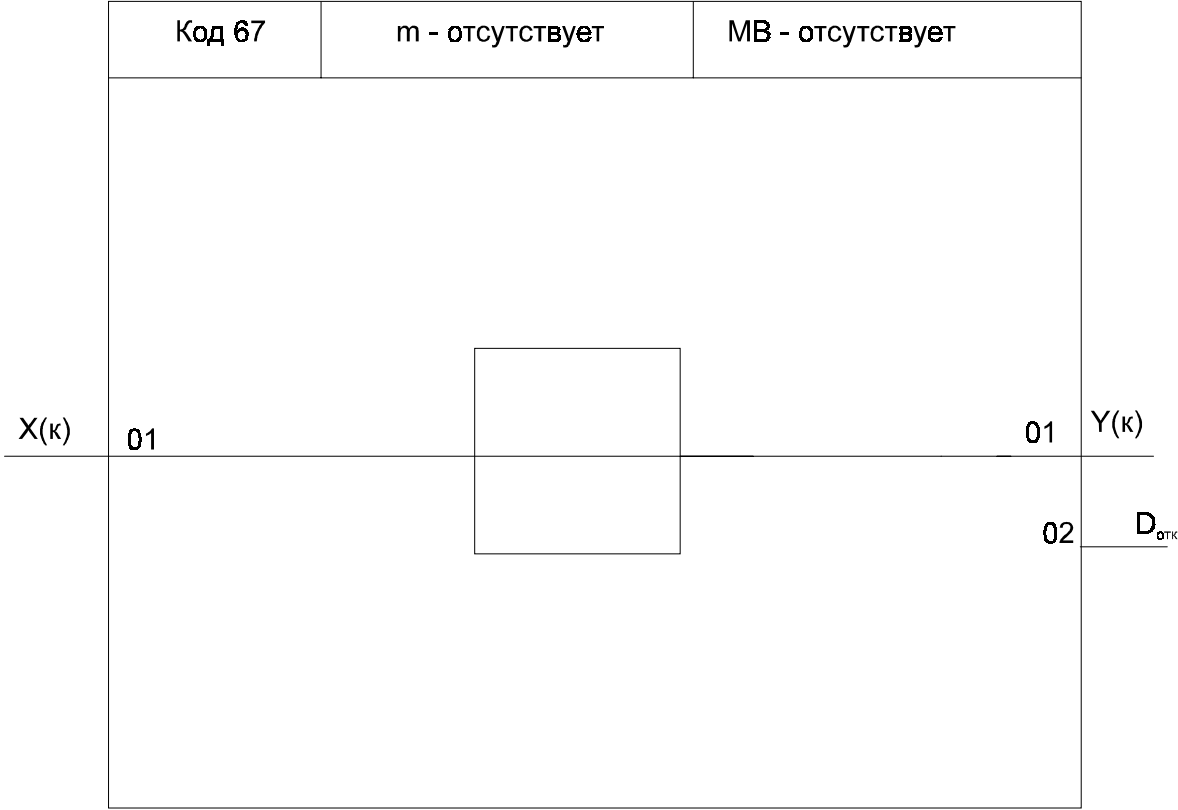


Рис. 61

8. Логические операции

8.1. ЛОИ (70) - Логическое И

Назначение

Алгоритм используется для формирования нескольких (до 20) дискретных сигналов, каждый из которых является логическим объединением по И двух дискретных сигналов.

Описание алгоритма

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 20$ и задается модификатором (рис. 62). При $m=0$ алгоритм является, "пустым".

Каждое звено имеет два входных сигнала $C_{1,j}$ и $C_{2,j}$ и формирует выходной сигнал:

$$D_j = C_{1,j} \wedge C_{2,j} \quad (48)$$

В табличной форме работа алгоритма записывается в виде:

Таблица 67

$C_{1,j}$	0	0	1	1
$C_{2,j}$	0	1	0	1
D_j	0	0	0	1

Входы-выходы алгоритма ЛОИ

Таблица 68

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{1,1}$	Входы	1-й вход 1-го звена
02	$C_{2,1}$		2-й вход 1-го звена
03	$C_{1,2}$		1-й вход 2-го звена
04	$C_{2,2}$		2-й вход 2-го звена
.	.		.
.	.		.
.	.		.
2m-1	$C_{1,m}$		1-й вход m -го звена
2m	$C_{2,m}$	Выходы	2-й вход m-го звена
01	D_1		Выход 1-го звена
02	D_2		Выход 2-го звена
.	.		.
.	.		.
.	.		.
m	D_m		Выход m-го звена

Функциональная схема алгоритма
"Логическое И ЛОИ"

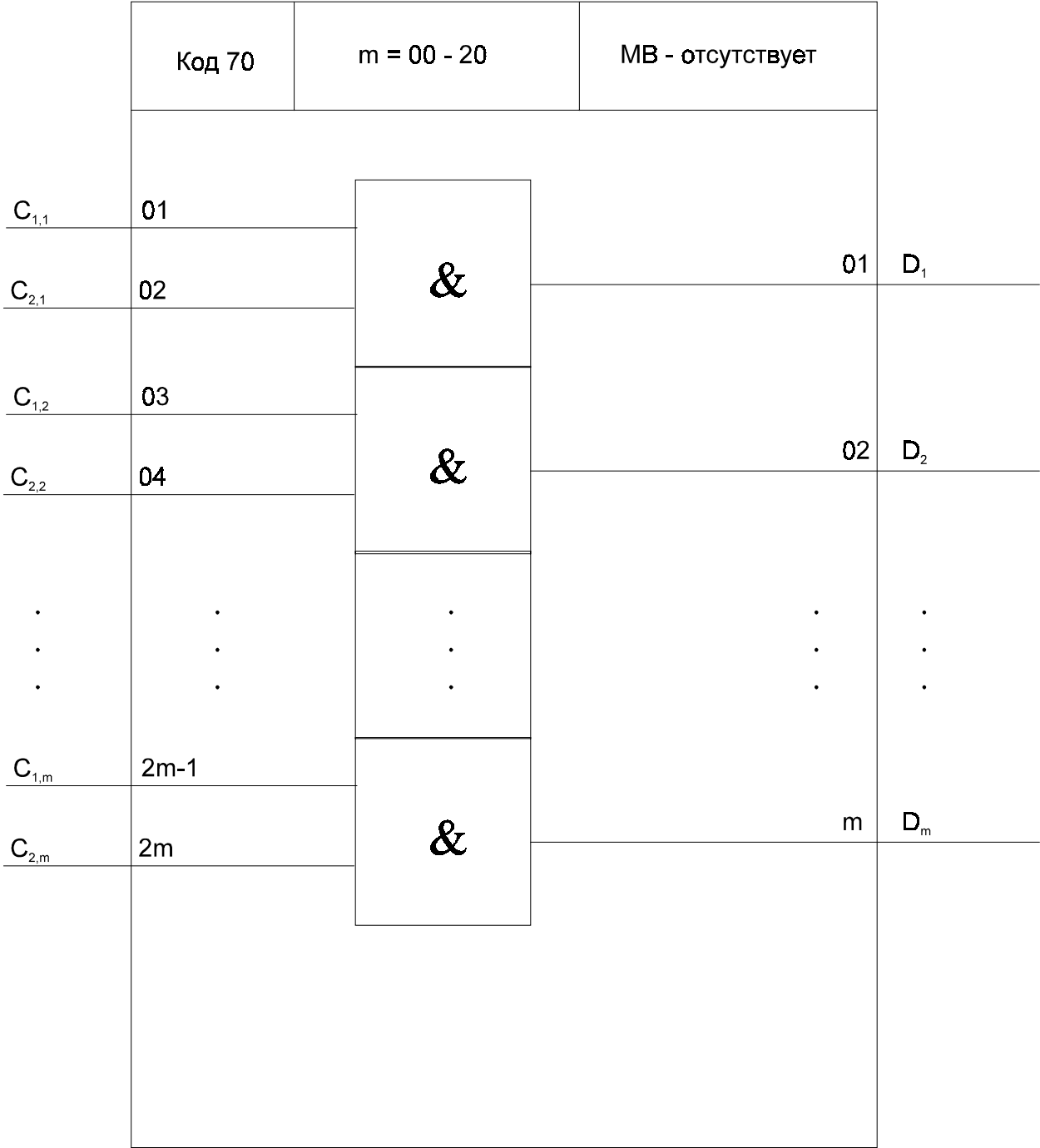


Рис. 62

8.2 МНИ (71) - Многовходовое И

Назначение

Алгоритм используется для логического объединения по "И" нескольких (до 99) дискретных сигналов.

Описание алгоритма

Алгоритм имеет m входных сигналов C_j и один выходной сигнал D . Число входов $0 \leq m \leq 99$ и задается модификатором (рис. 63). При $m=0$ алгоритм является "пустым".

Выходной сигнал: $D = C_1 \wedge C_2 \wedge \dots \wedge C_m$.

В табличной форме работа алгоритма записывается в виде:

Таблица 69

C_1	C_2	C_3	...	C_m	D
1	1	1	...	1	1
0	*	*	...	*	0
*	0	8	...	*	0
*	*	0	...	*	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
*	*	*	...	0	0

* - значение сигнала безразлично.

Входы-выходы алгоритма МНИ

Таблица 70

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	1-й вход
02	C_2		2-й вход
.	.		.
.	.		.
.	.		.
m	C_m		m - й вход
01	D	Выход	Выход

Функциональная схема алгоритма
"Многовходовое И МНИ"

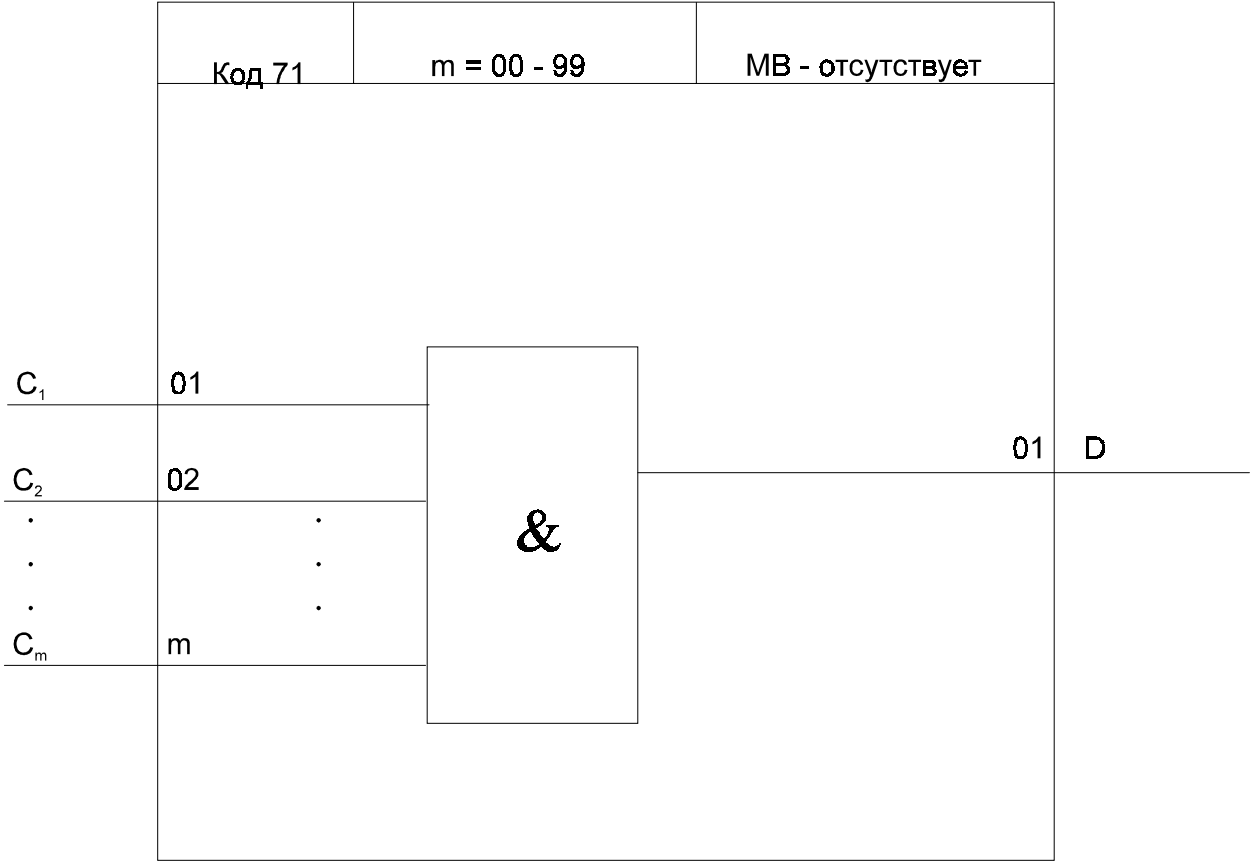


Рис. 63

8.3. ИЛИ (72) - Логическое ИЛИ

Назначение

Алгоритм используется для формирования нескольких (до 20) дискретных сигналов, каждый из которых является логическим объединением по ИЛИ двух дискретных сигналов.

Описание алгоритма

Алгоритм содержит t идентичных независимых звеньев,

где $0 \leq m \leq 20$ и задается модификатором (рис. 64). При $m=0$ алгоритм является "пустым". Каждое звено имеет два входных сигнала $C_{1,j}$ и $C_{2,j}$ и формирует выходной сигнал

$$D_j = C_{1,j} \vee C_{2,j} \quad (49)$$

В табличной форме работа алгоритма записывается в виде:

Таблица 71

$C_{1,j}$	0	0	1	1+
$C_{2,j}$	0	0	0	1
D_j	0	1	1	1

Входы-выходы алгоритма ИЛИ

Таблица 72

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{1,1}$	Входы	1-й вход 1-го звена
02	$C_{2,1}$		2-й вход 1-го звена
03	$C_{1,2}$		1-й вход 2-го звена
03	$C_{2,2}$		2-й вход 2-го звена
.	.		.
.	.		.
.	.		.
2m-1	$C_{1,m}$		1-й вход m-го звена
2m	$C_{2,m}$	Выходы	2-й вход m-го звена
01	D_1		Выход 1-го звена
02	D_2		Выход 2-го звена
.	.		.
.	.		.
.	.		.
m	D_m		Выход m-го звена

Функциональная схема алгоритма
"Логическое ИЛИ"

	Код 70	m = 00 - 20	МВ - отсутствует
$C_{1,1}$	01	1	
$C_{2,1}$	02		01 D_1
$C_{1,2}$	03	1	
$C_{2,2}$	04		02 D_2
.	.	.	.
.	.	.	.
.	.	.	.
$C_{1,m}$	2m-1	1	
$C_{2,m}$	2m		m D_m

Рис. 64

8.4. МИЛ (73) - Многовходовое ИЛИ

Назначение

Алгоритм применяется для логического объединения по ИЛИ нескольких (до 99) дискретных сигналов.

Описание алгоритма

Алгоритм имеет m входов C_j и один выход, причем $0 \leq m \leq 99$ и задается модификатором (рис. 65). При $m=0$ алгоритм является "пустым".

Выходной сигнал:

$$D = C_1 \vee C_2 \vee \dots \vee C_m.$$

В табличной форме работа алгоритма записывается в виде:

Таблица 73

C_1	C_2	C_3	...	C_m	D
0		0	...	0	0
1		*	...	*	1
*		*	...	*	1
*		1	...	*	1
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
*	*	*	...	1	1

* - значение сигнала безразлично

Входы - выходы алгоритма МИЛ

Таблица 74

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	1-й вход
02	C_2		2-й выход
•	•		•
•	•		•
•	•		•
m	C_m	Выход	m -ый вход
01	D		Выход

Функциональная схема алгоритма
"Многовходовое ИЛИ МИЛ"

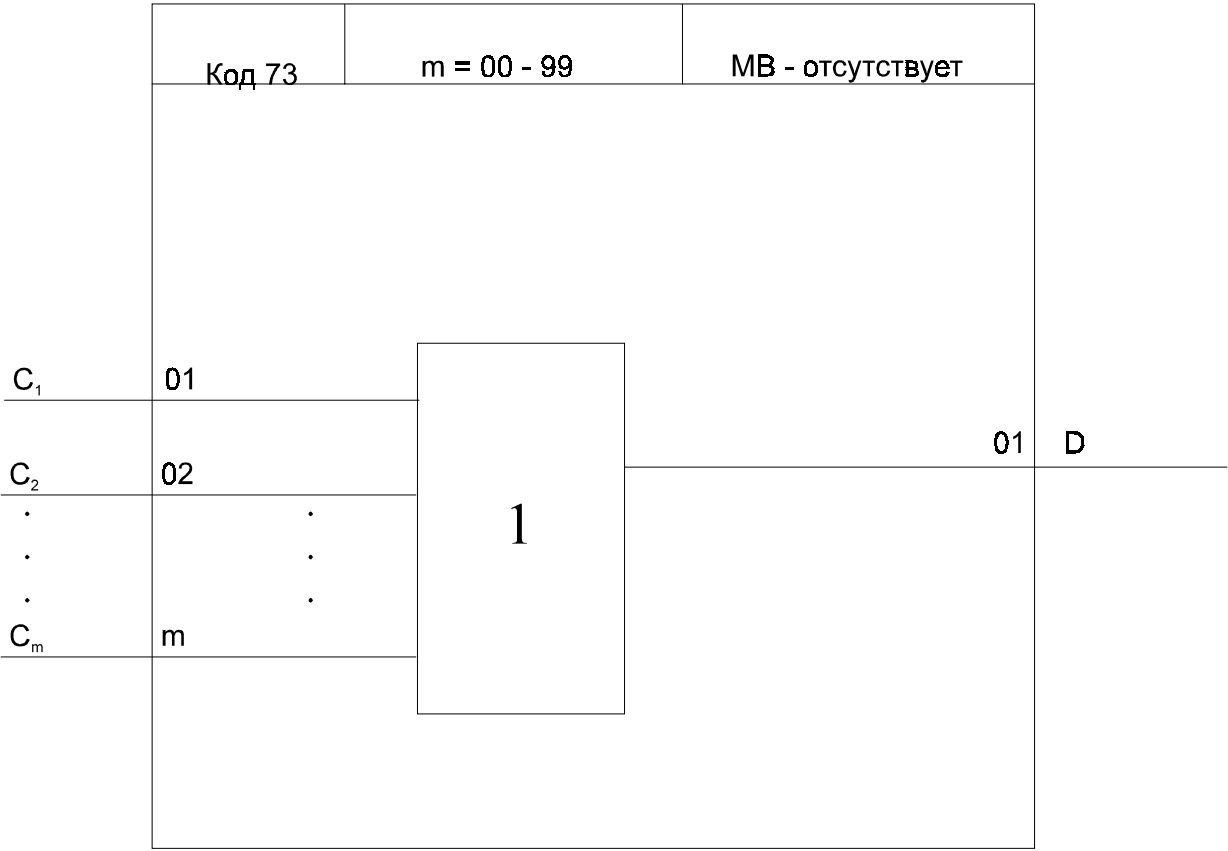


Рис. 65

8.5 ИИЛ (74) - Исключающее ИЛИ

Назначение

Алгоритм используется для формирования нескольких (до 20) дискретных сигналов, каждый из которых является логическим объединением по исключающему ИЛИ двух дискретных сигналов.

Описание алгоритма

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 20$ и задается модификатором (рис. 66). При $m=0$ алгоритм является "пустым".

Каждое звено имеет два входных сигнала $C_{1,j}$ и $C_{2,j}$ и формирует выходной сигнал

$$D_j = C_{1,j} \vee C_{2,j} \quad (50)$$

В табличной форме работа каждого звена представлена в виде:

Таблица 75

$C_{1,i}$	0	0	1	1
$C_{2,i}$	0	1	0	1
D_i	0	1	1	0

Входы-выходы алгоритма ИЛИ

Таблица 76

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{1,1}$	Входы	1-й вход 1-го звена
02	$C_{2,1}$		2-й вход 1-го звена
03	$C_{1,2}$		1-й вход 2-го звена
04	$C_{2,2}$		2-й вход 2-го звена
•	•		•
•	•		•
•	•		•
2m-1	$C_{1,m}$		1-й вход m-го звена
2m	$C_{2,m}$	Выходы	2-й вход m-го звена
01	D_1		Выход 1-го звена
02	D_2		Выход 2-го звена
•	•		•
•	•		•
•	•		•
m	D_m		Выход m-го звена

Функциональная схема алгоритма
"Исключающее ИЛИ ИИЛ"

	Код 74	m = 00 - 20	МВ - отсутствует
$C_{1,1}$	01	=1	
$C_{2,1}$	02		01 D_1
$C_{1,2}$	03	=1	
$C_{2,2}$	04		02 D_2
•	•	•	•
•	•	•	•
•	•	•	•
$C_{1,m}$	2m-1	=1	
$C_{2,m}$	2m		m D_m

Рис. 66

8.6 МАЖ (75) - Мажорирование

Назначение

Алгоритм работает по правилу "два из трех". Алгоритм используется для повышения достоверности дискретных сигналов, поступающих, например, от модулей дискретного ввода контроллера

Описание алгоритма

Алгоритм содержит узел мажорирования, а также логику, определяющую номер сигнала, состояние которого не совпадает с состоянием двух других сигналов (рис. 67)

Работа алгоритма описывается следующей таблицей:

Таблица 77

Состояние входных сигналов	D ₀	D	N
C ₁ =C ₂ =C ₃ =C	C	0	0
C ₂ =C ₃ =C; C ₁ <>C	C	1	1
C ₁ =C ₃ =C; C ₂ <>C	C	1	2
C ₁ =C ₂ =C; C ₃ <>C	C	1	3

Входы-выходы алгоритма

Таблица 78

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C ₁	Входы	1-й сигнал
02	C ₂		2-й сигнал
03	C ₃		3-й сигнал
01	D ₀	Выходы	Основной выход
02	D		Признак несовпадения
03	N		Номер входного сигнала, состояние которого не совпадает с состоянием двух других сигналов

Функциональная схема алгоритма
"Мажорирование МАЖ"

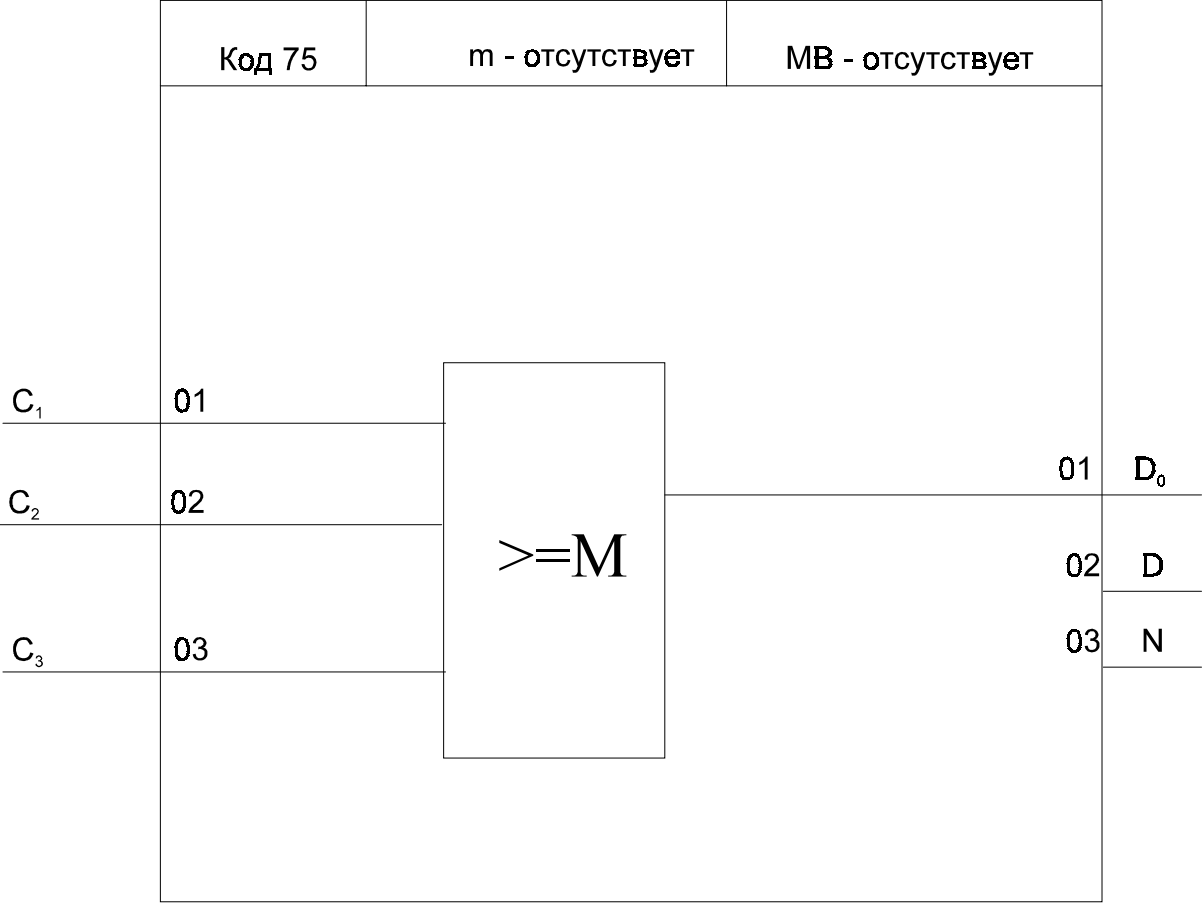


Рис. 67

8.7 ТРИ (76) - RS - Триггер

Назначение

Алгоритм содержит несколько (до 20) независимых RS -триггеров и применяется для запоминания дискретных сигналов.

Описание алгоритма

Алгоритм содержит $0 \leq m \leq 20$ независимых ячеек, каждая из которых представляет собой RS - триггер. Число m определяется модификатором (рис. 68). При $m=0$ алгоритм является "пустым".

Каждая ячейка имеет два дискретных входа и один дискретный выход, S - вход является установочным: если при $C_R = 0$ сигнал $C_S = 1$, то выходной сигнал $D = 1$. Этот сигнал запоминается, т.е. остается равным единице независимо от значения сигнала на входе C_S .

Если $C_R = 1$, триггер сбрасывается, при этом $D = 0$. Работа каждой ячейки алгоритма определяется таблицей:

Таблица 79

C_S	C_R	D
0	0	D_{i-1}
1	0	1
0	1	0
1	1	0

В таблице D_{i-1} - предыдущее значение выхода.

Входы-выходы алгоритма ТРИ

Таблица 80

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{S,1}$	Входы	Вход установки 1-й ячейки
02	$C_{R,1}$		Вход сброса 1-й ячейки
03	$C_{S,2}$		Вход установки 2-й ячейки
04	$C_{R,2}$		Вход сброса 2-й ячейки
.	.		.
.	.		.
.	.		.
2m-1	$C_{S,m}$		Вход установки m-й ячейки
2m	$C_{R,m}$	Выходы	Вход сброса m-й ячейки
01	D_1		Выход 1-ой ячейки
02	D_2		Выход 2-ой ячейки
.	.		.
.	.		.
.	.		.
m	D_m		Выход m-ой ячейки

Функциональная схема алгоритма
"Триггер ТРИ"

	Код 76	m = 00 - 20	МВ - отсутствует
$C_{S,1}$	01	T (RS) ₁	01 D ₁
$C_{R,1}$	02		
$C_{S,2}$	03	T (RS) ₂	02 D ₂
$C_{R,2}$	04		
•	•	•	•
•	•	•	•
•	•	•	•
$C_{S,m}$	2m-1	T (RS) _m	m D _m
$C_{R,m}$	2m		

Рис. 68

8.8. РЕУ (77) - Регистр с записью по уровню

Назначение

Алгоритм используется для запоминания нескольких (до 20) дискретных сигналов. Информация записывается в регистр пока имеется команда записи.

Описание алгоритма

Алгоритм представляет собой m -разрядный регистр, где $0 \leq m \leq 20$ и задается модификатором (рис. 69). Алгоритм имеет m информационных входов, на которые подаются входные сигналы C_i , и один управляющий вход записи $C_{зпс}$.

Если $C_{зпс}=1$, информация записывается в регистр и его выходы $D_i=C_i$ (т. е. $D_1=C_1$; $D_2=C_2$ и т. д.).

Если $C_{зпс}=0$, сигналы D_i запоминаются и не изменяются (т.е. не зависят от C_i).

Сигнал $D_{зпс}=C_{зпс}$, т. е. повторяет входной управляющий сигнал.

Работа алгоритма описывается следующей таблицей:

Таблица 81

$C_{зпс}$	D_i	$D_{зпс}$
1	$D_i=C_i$	1
0	$D_i=C_{i,0}$	0

В этой таблице $C_{i,0}$ - значение C_i в момент снятия команды $C_{зпс}$.

Входы-выходы алгоритма РЕУ

Таблица 82

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{зпс}$	Входы	Команда записи
02	C_1		1-й сигнал
03	C_2		2-й сигнал
•	•		•
•	•		•
•	•		•
$m+1$	C_m		m -й сигнал
01	$D_{зпс}$	Выходы	Признак записи
02	D_1		1-й сигнал
03	D_2		2-й сигнал
•	•		•
•	•		•
•	•		•
$m+1$	D_m		m -й сигнал

Функциональная схема алгоритма
"Регистр с записью по уровню РЕУ"

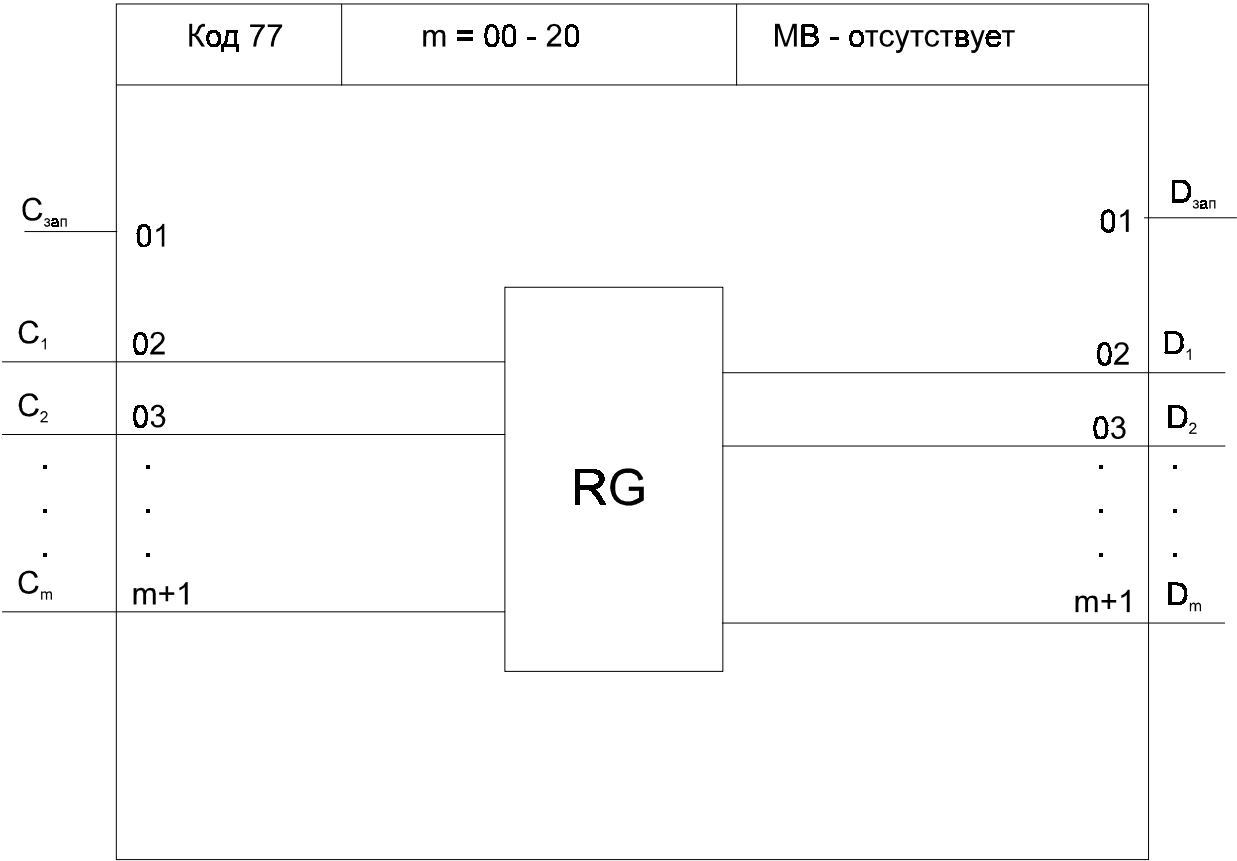


Рис. 69

8.9. РЕФ (78) - Регистр с записью по фронту

Назначение

Алгоритм используется для запоминания нескольких (до 20) дискретных сигналов. Информация фиксируется в регистре в момент прихода переднего фронта управляющего сигнала.

Описание алгоритма

Алгоритм представляет собой m -разрядный регистр, где $0 \leq m \leq 20$ и задается модификатором. Алгоритм имеет t информационных входов, на которые подаются входные сигналы C_i , и один управляющий вход записи $C_{зпс}$ (рис. 70).

По переднему фронту дискретного сигнала $C_{зпс}$ (т.е. в момент перехода сигнала $C_{зпс}$ из состояния логического 0 в состояние логической 1) информация на входах C_i записывается в регистр, после чего алгоритм не реагирует на изменение сигналов на входах C_i . Записанная в регистре информация передается на соответствующие выходы алгоритма. D_i .

В момент записи на один цикл работы контроллера формируется лог.1 на выходе $D_{зпс}$. В остальных случаях $D_{зпс} = 0$.

Входы-выходы алгоритма РЕФ

Таблица 83

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{зпс}$	Входы	Команда записи (передний фронт)
02	C_1		1-й сигнал
03	C_2		2-й сигнал
.	.		.
.	.		.
.	.		.
m+1	C_m	Выходы	m-й сигнал
01	$D_{зпс}$		Момент записи
02	D_1		1-й сигнал
03	D_2		2-й сигнал
.	.		.
.	.		.
.	.		.
m+1	D_m		m-й сигнал

Функциональная схема алгоритма
"Регистр с записью по фронту РЕФ"

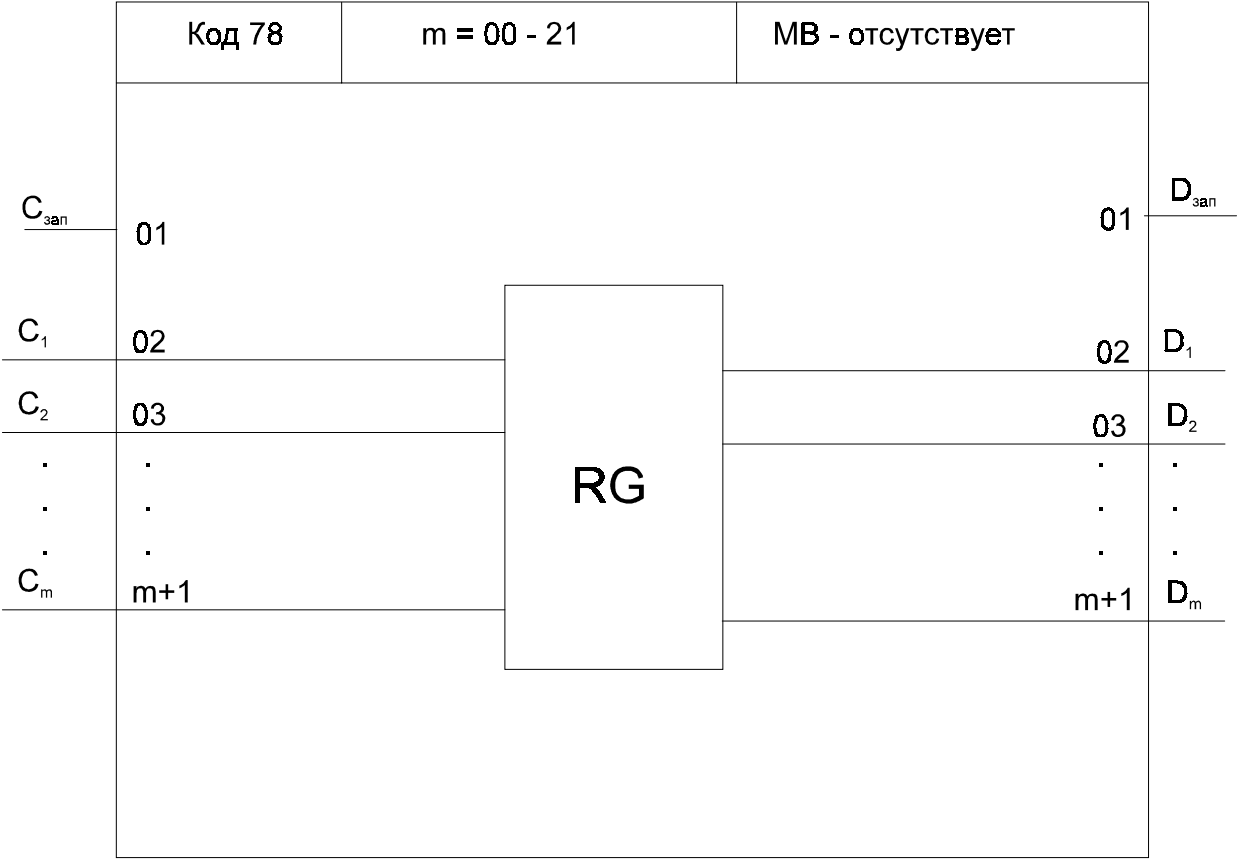


Рис. 70

8.10. ВЫФ - Выделение фронта (79)

Назначение

Алгоритм применяется для выделения переднего или заднего фронта дискретного сигнала.

Описание алгоритма

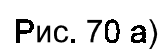
Если на входе алгоритма (рис. 70 а)) дискретный сигнал С изменяет свое состояние с логического 0 на логическую 1 (передний фронт), то на выходе алгоритма формируется сигнал D = 1 на время, равное времени одного цикла работы контроллера (от 0,2 до 2 с). Остальное время D = 0.

Для выделения заднего фронта на входе алгоритма устанавливается инверсия.

Входы-выходы алгоритма ВЫФ

Таблица 83.1

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C	Вход	Входной сигнал
01	D	Выход	Выходной сигнал



9. Дискретное управление

9.1. ЭТП (80) - Этап

Назначение

Алгоритм применяется для организации логической шаговой программы, т.е. программы, которая должна выполнить определенную последовательность действий.

Алгоритм ЭТП содержит несколько (до 20) шагов (рис. 72). Переход от одного шага к другому выполняется либо по условию (т.е. по дискретной команде), либо по истечении определенного времени. В контроллере может быть задействовано несколько алгоритмов ЭТП. Возможно условное или безусловное разветвление шаговой программы.

Алгоритм ЭТП применяется в сочетании с алгоритмом ОКЛ.

Описание алгоритма

Один этап состоит из $0 \leq m \leq 20$ шагов, причем задается модификатором алгоритма ЭТП.

Принцип работы каждого шага одинаков. Шаг имеет 3 входа и 1 выход. Вход C - условие выполнения шага, вход T - контрольное время шага, вход N - параметр, определяющий последующий ход выполнения программы, D - выход шага (рис. 71).

Шаг работает следующим образом. Если условие C окажется выполненным (т.е. установится $C = 1$) раньше, чем истечет контрольное время T , то шаг выполняется, при этом устанавливается $D = 1$ и программа переходит к выполнению следующего по номеру шага.

Если до истечения контрольного времени T условие не будет выполнено (т.е. останется $C = 0$), то поведение программы зависит от параметра N . При этом возможны следующие 4 варианта:

- 1) если $N = 0$, то устанавливается выход $D = 1$ и программа переходит к выполнению следующего шага;
- 2) если $N > 0$, при этом в программе имеется этап с номером N и он подключен к алгоритму ОКЛ, то состояние выхода текущего шага не изменяется, программа переходит к первому шагу этапа с номером N и продолжает выполняться;
- 3) если $N > 0$, но этапа с номером N не существует или он не подключен к алгоритму ОКЛ (например $N = 99$), то выполнение программы прекращается и она переходит в состояние ожидания;
- 4) если $N < 0$, состояние выхода текущего шага не изменяется, то программа переходит к выполнению следующего шага, но предварительно обнуляются выходы всех шагов этапа, имеющего номер N .

После выполнения последнего шага какого-либо этапа и в том случае, когда в этом этапе не запрограммирован переход к другому этапу, программа переходит в состояние "конец программы".

В исходном состоянии алгоритм этапа номера не имеет. Этот номер приписывается этапу после того, как выход N_m алгоритма этапа по конфигурации подключается к одному из входов $N_{3,4}$ алгоритма ОКЛ. Например, если алгоритм этапа подключается к входу $N_{3,1}$ алгоритма ОКЛ, данный этап получает номер 4.

Блок-схема работы шага представлена на рис. 71

От алгоритма ОКЛ, подключенного к выходу N_m алгоритма этапа, поступают команды пуска, останова и сброса. Команда пуска инициирует начало выполнения очередного шага. При останове счетчик контрольного времени останавливается. При сбросе все выходы $D_1 \dots D_m$ шагов обнуляются и выполнение программы прекращается. После пуска из состояния "сброс" программа переходит к выполнению первого шага первого этапа.

Указанные свойства помимо последовательного (шаг за шагом) выполнения программы на каждом шаге программы позволяют осуществить:

- 1) выдержку времени (устанавливается $C = N = 0$, T - заданное время);
- 2) условное разветвление программы (устанавливается $T = 0$, $N = N_3$ - номер этапа, к которому осуществляется переход, C - условие перехода);
- 3) безусловный переход к другому этапу, что требуется, например, в том случае, когда число шагов должно быть больше 20 (устанавливается $C = T = 0$, $N = N_3$ - номер этапа, к которому требуется перейти);
- 4) обнуление этапа (устанавливается $C = 0$, $T = 0$, $N < 0$, при этом $|N| = N_3$ - номер этапа, который должен быть обнулен).

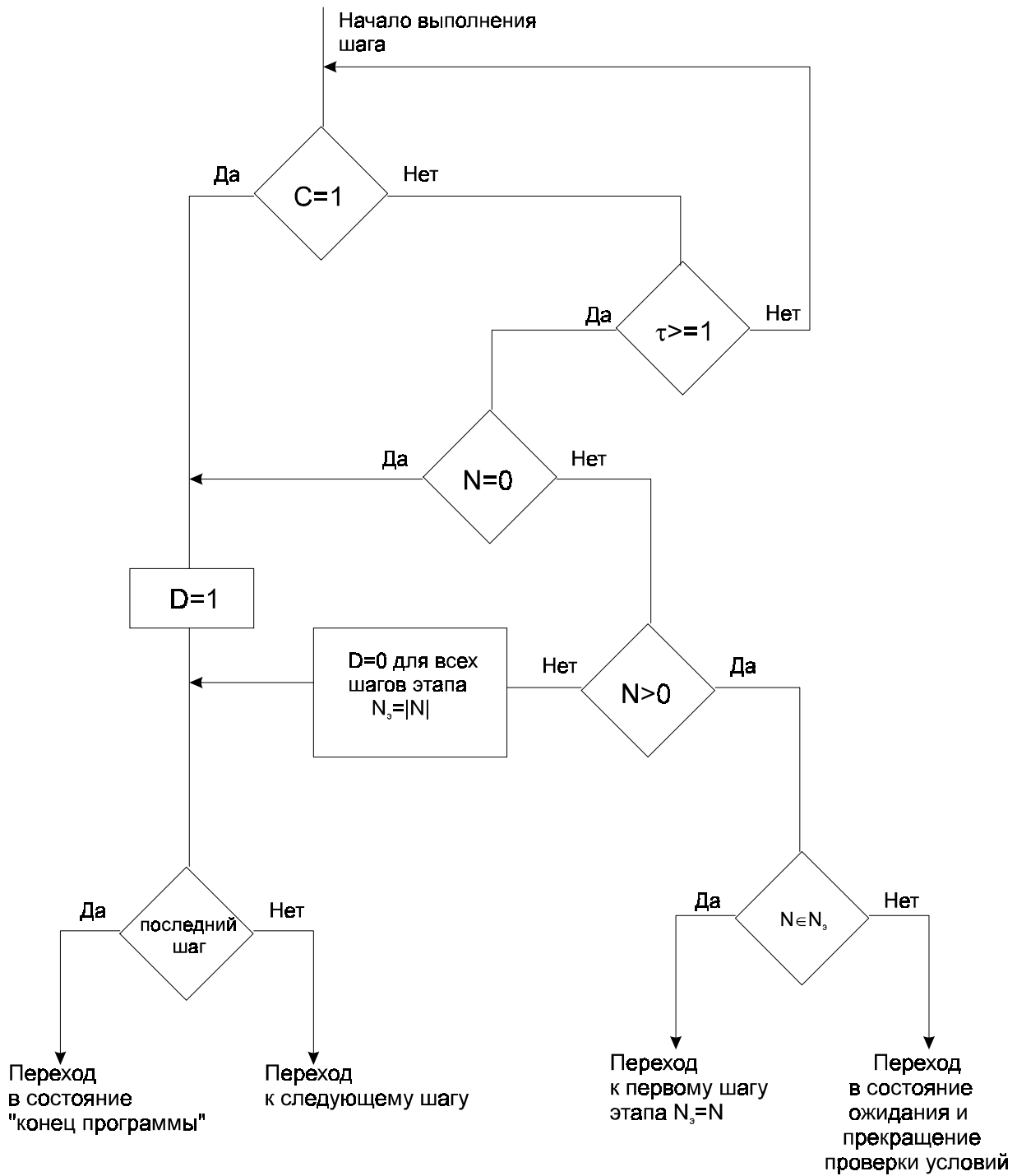
Входы-выходы алгоритма ЭТП

Таблица 84

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Условие выполнения 1-го шага
02	T_1		Контрольное время 1-го шага
03	N_1		Параметр 1-го шага
04	C_2		Условие выполнения 2-го шага
05	T_2		Контрольное время 2-го шага
06	N_2		Параметр 2-го шага
.	.		.
.	.		.

.	.		.
$3m-2$	C_m		Условие выполнения шага m
$3m-1$	T_m		Контрольное время шага m
$3m$	N_m		Параметр шага m
01	$N_{ш}$	Выходы	Номер выполняемого шага
02	$T_{ш}$		Время, оставшееся до истечения контрольного времени выполняемого шага
03	N		Параметр, равный параметру N_i выполняемого шага
04	$D_{ш}$		Признак того, что выполняемый шаг закончен (установлено $D=1$)
05	D_1		Выход 1-го шага
06	D_2		Выход 2-го шага
.	.		.
.	.		.
.	.		.
$m+4$	D_m		Выход шага m

Блок-схема работы шага



t - время, прошедшее от начала выполнения текущего шага;
 N_s - среди этапов, подключенных к алгоритму ОКЛ, есть этап с номером N

Рис. 71

Функциональная схема алгоритма
"Этап ЭТП"

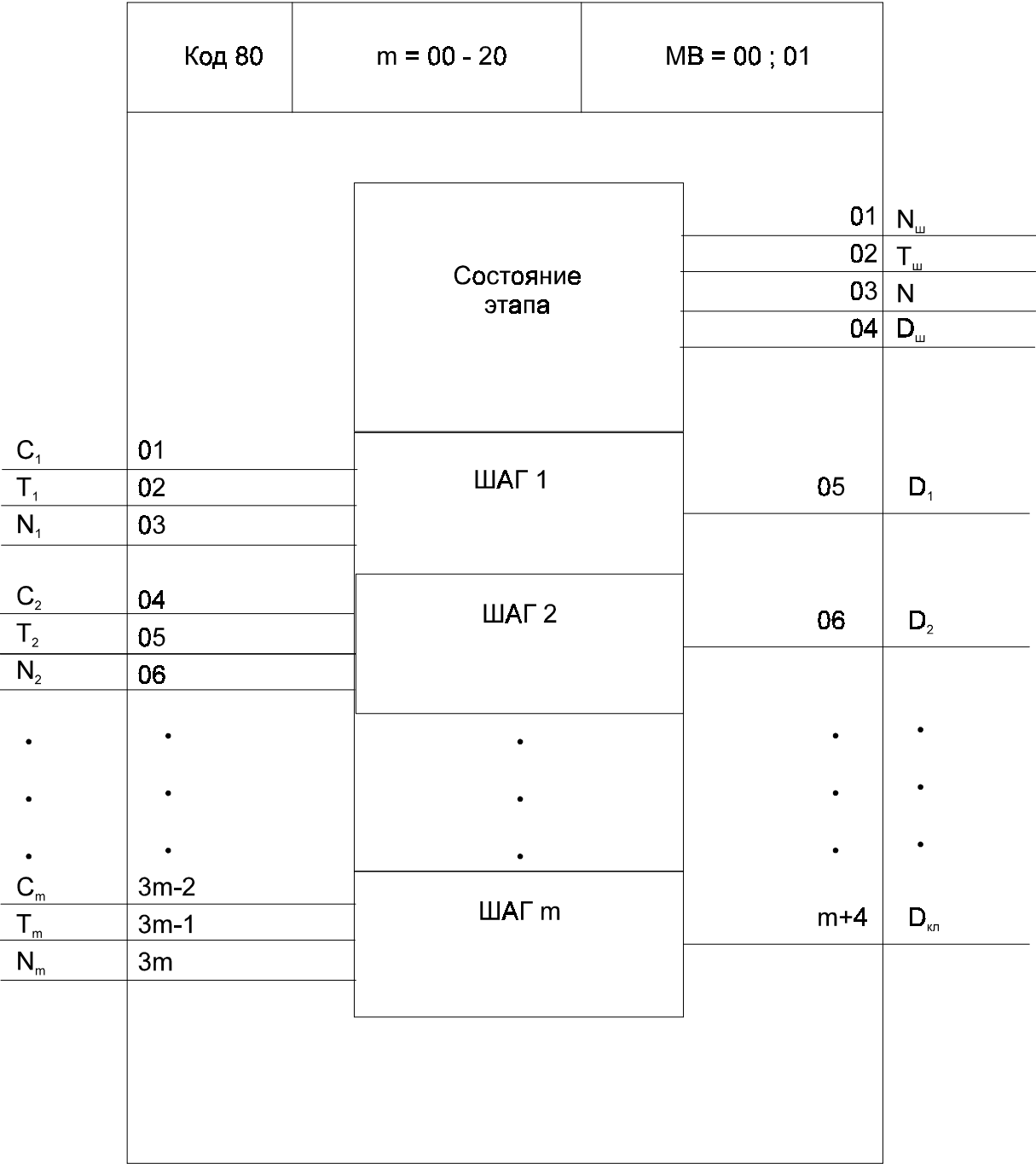


Рис. 72

9.2. TMP (81) - Таймер

Назначение

Алгоритм содержит несколько (до 20) таймеров, объединенных общими командами "стоп" и "сброс". В каждом таймере индивидуально настраивается время срабатывания таймера.

Алгоритм используется для задания выдержек времени.

Описание алгоритма

Алгоритм содержит одно звено таймера и m нуль-органов, где $0 \leq m \leq 20$ и задается модификатором (рис. 72).

Алгоритм имеет два дискретных входа "стоп" и "сброс".

В состоянии "стоп" отчет времени прекращается, в состоянии "сброс" звено таймера обнуляется и время "останавливается" так же, как в состоянии "стоп". Команда "сброс" приоритетна по отношению к команде "стоп".

Если команды "стоп" и "сброс" отсутствуют, алгоритм находится в состоянии "пуск". В этом состоянии производится отчет времени.

Если в i -ом нуль-органе значение текущего времени сравнивается со значением настроечного параметра T_i , выходной сигнал звена D_i переходит в состояние Лог.1 (табл.85).

Таблица 85

Текущее время	Выход нуль-органа
$t < T_i$	$D_i = 0$
$t \geq T_i$	$D_i = 1$

На выходе T алгоритма формируется сигнал текущего времени, отсчитываемого от начала пуска, поступившего после сброса. В состоянии "стоп" сигнал на выходе T замораживается, в состоянии "сброс" - обнуляется.

Входы-выходы алгоритма

Таблица 86

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{ст}$	Входы	Команда "стоп"
02	$C_{сбр}$		Команда "сброс"
03	T_1		Время срабатывания 1-го нуль-органа
04	T_2		Время срабатывания 2-го нуль-органа
•	•		•
•	•		•
$m+1$	T_m		Время срабатывания m -го нуль-органа
01	T	Выходы	Текущее время
02	D_1		Выход 1-го нуль-органа
03	D_2		Выход 2-го нуль-органа
•	•		•
•	•		•
$m+1$	D_m		Выход m -го нуль-органа

Функциональная схема алгоритма
"Таймер TMR"

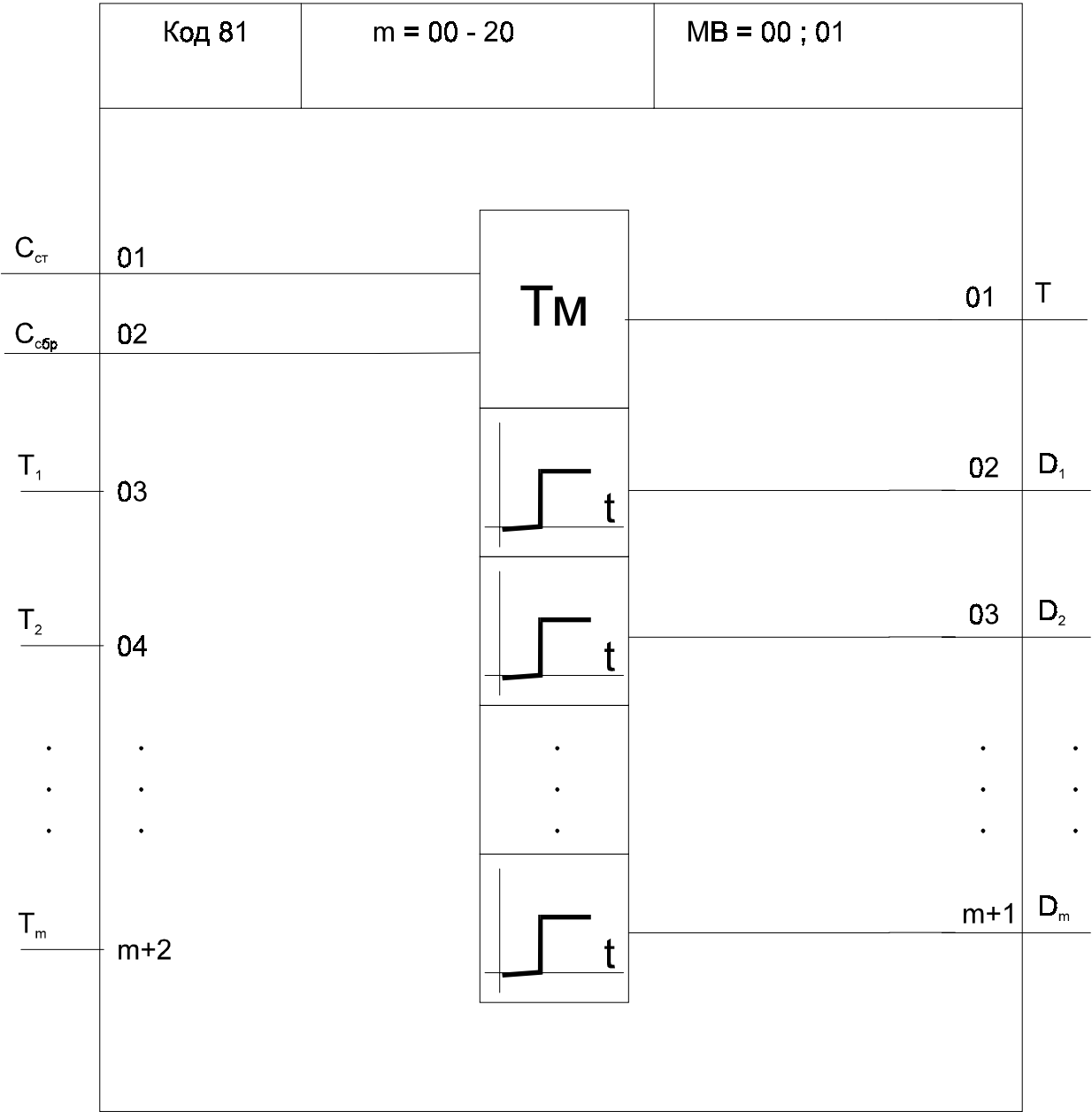


Рис. 72

9.3. СЧТ (82) - Счетчик

Назначение

Алгоритм представляет собой реверсивный счетчик и используется для подсчета числа дискретных событий (переходов из состояния лог. 0 в состояние лог.1 или обратных переходов), а также для сравнения подсчитанного числа с заданными числами с помощью числовых нуль-органов. Один алгоритм может содержать несколько (до 20) таких нуль-органов.

Описание алгоритма

Каждый переход дискретного сигнала на входе C_B из состояния лог.0 в состояние лог.1 вызывает приращение содержимого счетчика на единицу. Аналогичный переход на входе C_M уменьшает содержимое счетчика на единицу. Одновременный приход сигналов на входы C_B и C_M воспринимается как отсутствие обоих сигналов (рис. 73).

Если в момент снятия сигнала "сброс" на входе C_B или C_M имелся сигнал лог.1, эта ситуация воспринимается как наличие перехода соответствующего сигнала из нуля в единицу.

Работа счетчика разрешается, если отсутствуют сигналы "стоп" и "сброс" на входах $C_{ст}$ и $C_{сбр}$. Если имеется сигнал на входе $C_{ст}$, значение счетчика замораживается. Если имеется сигнал на входе $C_{сбр}$, в счетчик записывается начальное значение числа N_0 . Команда "сброс" приоритетна над командой "стоп".

Алгоритм содержит нуль-органы, число которых задается модификатором m и устанавливается в диапазоне $0 \leq m \leq 20$.

В каждом нуль-органе на входе задается пороговое число N_i . Пока содержимое счетчика $N < N_i$, сигнал на дискретном выходе данного нуль-органа равен нулю. Если $N \geq N_i$, на дискретном выходе появляется сигнал лог.1.

Если число в счетчик должно Записываться не по переднему, а по заднему фронту сигналов C_B или C_M , на соответствующем входе при его конфигурировании устанавливается инверсия.

Входы-выходы алгоритма СЧТ

Таблица 87

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_B	Входы	Команда на увеличение числа
02	C_M		Команда на уменьшение числа
03	$C_{ст}$		Команда "стоп"
04	$C_{сбр}$		Команда "сброс"
05	N_0		Начальное значение числа
06	N_1		Пороговое число 1-го нуль-органа
07	N_2		Пороговое число 1-го нуль-органа
.	.		.
.	.		.
.	.		.
m+5	N_m	Выходы	Пороговое число 1-го нуль-органа
01	N		Число, подсчитываемое счетчиком
02	D_1		Выход 1-го нуль-органа
03	D_2		Выход 2-го нуль-органа
.	.		.
.	.		.
.	.		.
m+1	D_m		Выход m-го нуль-органа

Функциональная схема алгоритма
"Счетчик СЧТ"

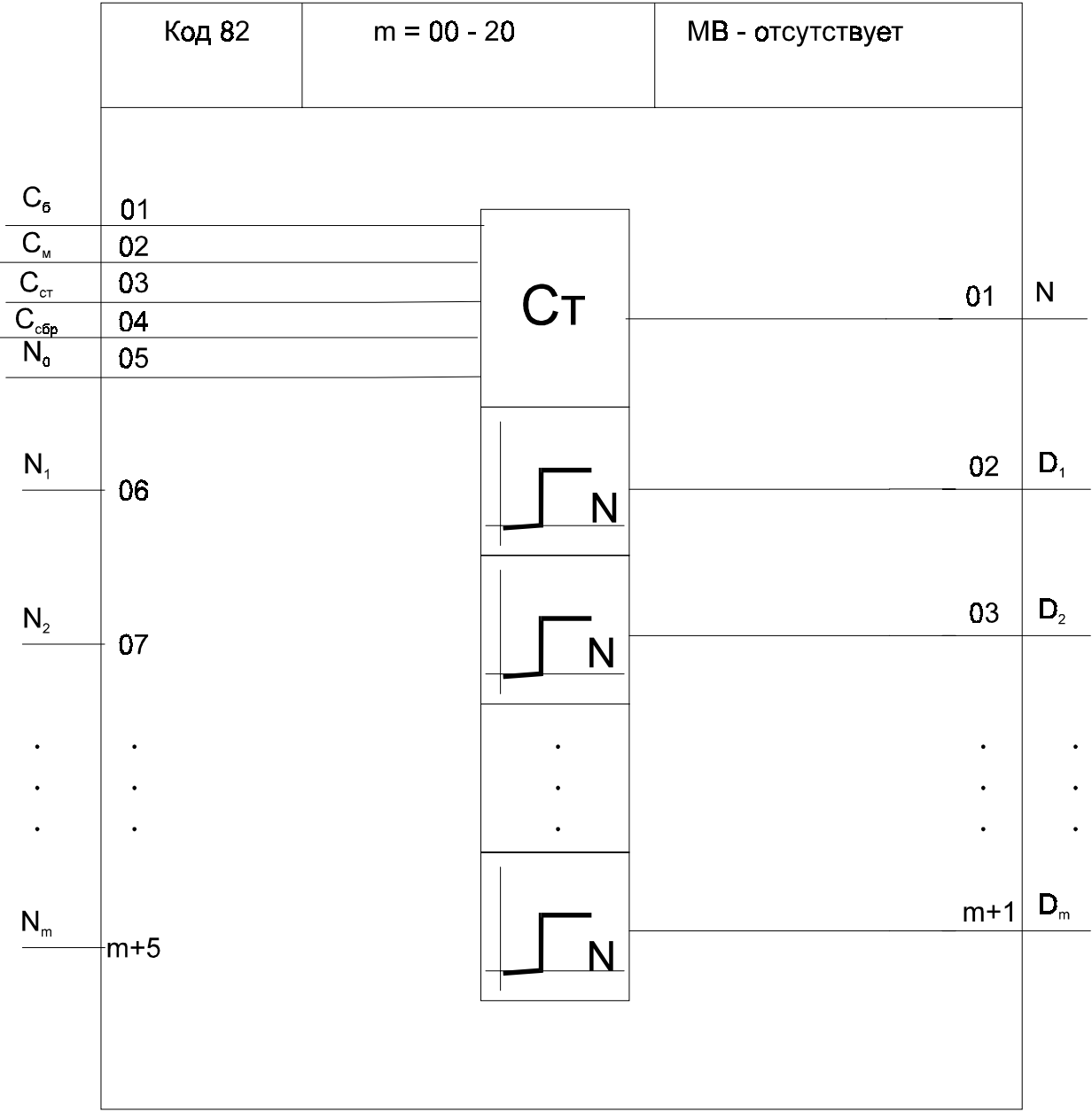


Рис. 73

9.4. ОДВ (83) - Одновибратор

Назначение

Алгоритм применяется в тех случаях, когда необходимо сформировать одиночный импульс заданной длительности.

Описание алгоритма

Одновибратор запускается по переднему фронту сигнала на входе C_n (пуск), т.е. когда на входе C_n дискретный сигнал переходит из состояния лог.0 в состояние лог.1 (рис. 74). Перед пуском выходной дискретный сигнал D отсутствует. После пуска появляется сигнал на выходе D , причем этот сигнал находится в состоянии лог.1 в течение времени $t = T$, где T - параметр настройки. По истечении времени T сигнал на выходе вновь переходит в нулевое состояние, после чего одновибратор можно вновь пустить.

На выходе T_n формируется текущее время, отсчитываемое от момента пуска. После отработки импульса $T_n = 0$.

Если одновибратор должен запускаться по заднему фронту, сигнал на входе C_n инвертируется.

Сигнал на входе $C_{сбр}$ (сброс) в любой момент времени обнуляет оба выхода алгоритма и обнуляет счетчик времени. При наличии команды "сброс" алгоритм не может быть запущен. Алгоритм также не может быть повторно запущен командой "пуск" до тех пор, пока не закончится формирование выходного импульса.

Если в момент снятия команды "сброс" на входе C_n сигнал равен лог.1, такая ситуация рассматривается как команда "пуск".

При $T \leq 0$ выходной импульс не формируется.

Входы-выходы алгоритма ОДВ

Таблица 88

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_n	Входы	Команда "пуск"
02	$C_{сбр}$		Команда "сброс"
03	T		Длительность импульса
01	D	Выходы	Основной выход
02	T_n		Текущее время импульса

Функциональная схема алгоритма
"Одновибратор ОДВ"

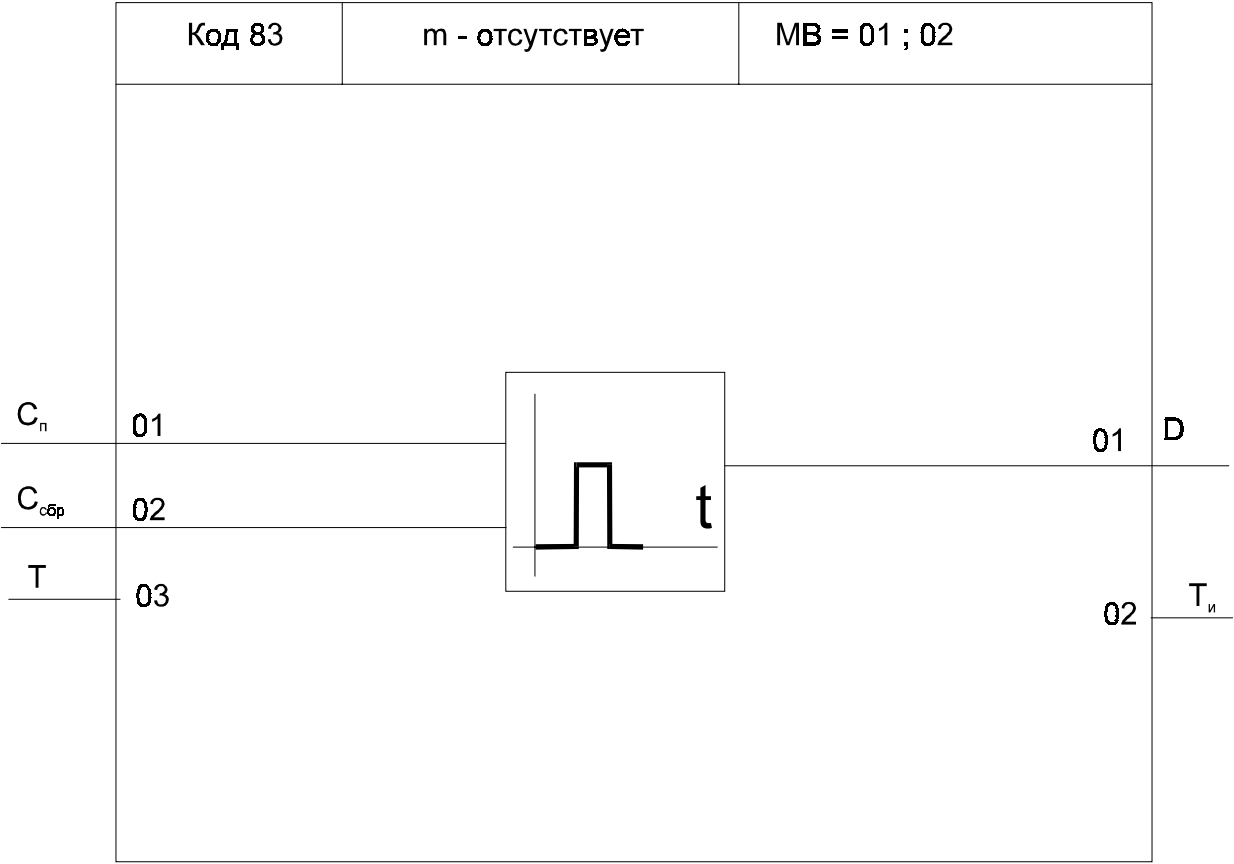


Рис. 74

9.5. МУВ (84) - Мультивибратор

Назначение

Алгоритм применяется для периодического включения оборудования (двигателя, нагревателя, обеспечения мигающей сигнализации и т.п.).

Описание алгоритма

Мультивибратор запускается по переднему фронту сигнала, на входе C_n (пуск), т.е. при изменении C_n из состояния лог.0 на состояние лог.1. После пуска на основном выходе алгоритма формируется последовательность импульсов. Длительность этих импульсов задается настроенным входом T_1 , длительность паузы - входом T_0 (рис. 75).

На выходе T_n формируется время, прошедшее от момента смены состояния выхода.

Сигнал $C_{сбр}$ (сброс) прерывает последовательность импульсов, устанавливая оба выхода алгоритма в состояние лог.0. Сигнал $C_{сбр}$ приоритетен над сигналом C_n .

Если сигнал $C_{сбр}$ снимается в момент, когда присутствует сигнал C_n , такая ситуация воспринимается как команда пуска.

Состояние основного выхода D алгоритма в режиме пуска при различных значениях T_1 и T_0 определяется таблицей:

Таблица 89

T_1	T_0	Выход D
$T_1 > 0$	$T_0 > 0$	
$T_1 \leq 0$	Безразлично	0
$T_1 > 0$	$T_0 \leq 0$	1

Входы-выходы алгоритма МУВ

Таблица 90

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_n	Входы	Команда "пуск"
02	$C_{сбр}$		Команда "сброс"
03	T_1		Длительность импульса
04	T_0		Длительность паузы
01	D	Выход	Основной выход
02	T_n		Время текущего импульса или паузы

Функциональная схема алгоритма
"Мультивибратор МУВ"

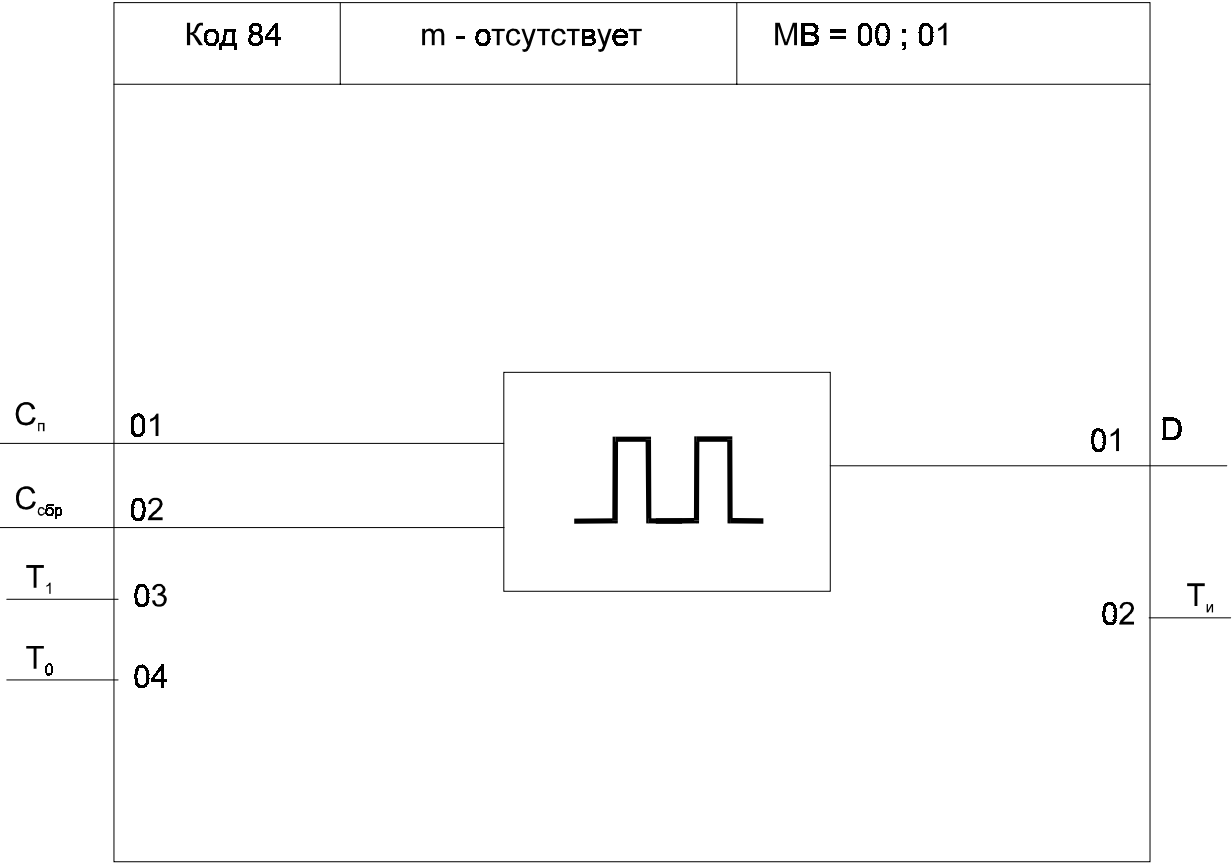


Рис. 75

9.6. ПЧИ (85) - Переключение чисел

Назначение

Алгоритм предназначен для выбора одного из нескольких (до 98) чисел.

Описание алгоритма

Алгоритм представляет собой переключатель на m входов, причем $0 \leq m \leq 98$ и задается модификатором (рис. 76).

На вход переключателя поступают числовые переменные N_i . Переменная, выбранная переключателем, поступает на выход.

Положением переключателя управляет сигнал на входе N_0 , причем значение N_0 равно номеру входа, подключаемого к выходу (табл.91).

Таблица 91

N_0	≤ 0	1	2	...	$\geq m$
N	0	N_1	N_2	...	N_m

Режимы работы алгоритма такие же, как у алгоритма ПЕР.

Входы-выходы алгоритма ПЧИ

Таблица 92

Входы-выходы			Наименование
№	Обозн.	Вид	
01	N_1	Входы	Первое число
02	N_2		Второе число
.	.		.
.	.		.
.	.		.
m	N_m		m -ое число
$m+1$	N_0	Выход	Число, задающее положение переключателя
01	N		Выбранное число

Функциональная схема алгоритма
"Управление двухпозиционной нагрузкой УДП"

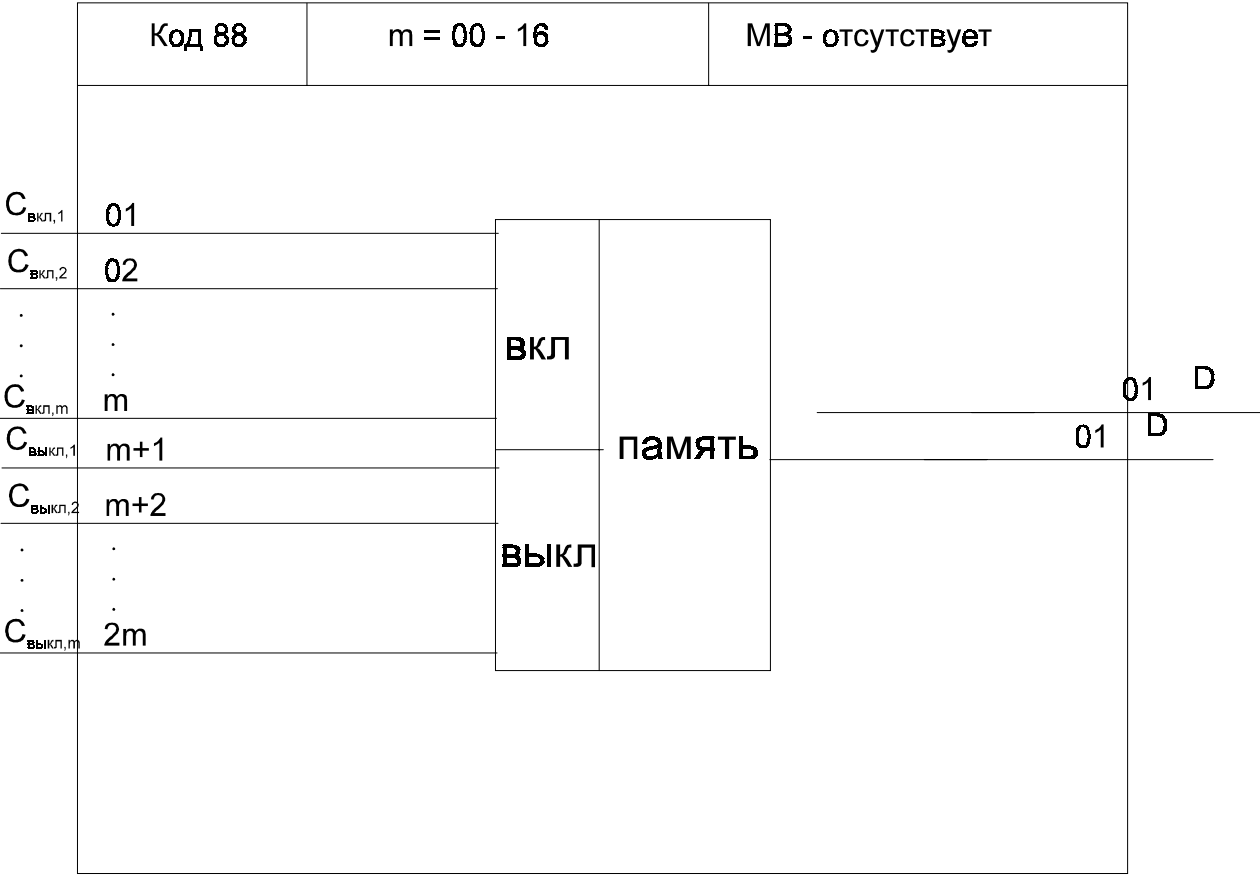


Рис. 79

9.7. СЧИ (86) - Сравнение чисел

Назначение

Алгоритм применяется в том случае, когда требуется зафиксировать достижение числовой переменной заданного значения.

Алгоритм содержит несколько (до 20) независимых каналов сравнения.

Описание алгоритма

Алгоритм содержит m независимых каналов сравнения, причем $0 \leq m \leq 20$ и задается модификатором (рис. 77). При $m=0$ алгоритм является "пустым".

Каждый канал сравнивает между собой два числа, поступающий на входы N_i и $N_{0,i}$, и вырабатывает дискретный сигнал на выходе D_i .

Если $N_i < N_{0,i}$, то $D_i = 0$. При $N_i \geq N_{0,i}$ значение $D_i = 1$.

Входы-выходы алгоритма СЧИ

Таблица 93

Входы-выходы			Назначение
№	Обозн.	Вид	
01	N_1	Входы	Выход 1-го канала
02	$N_{0,1}$		Уставка 1-го канала
03	N_2		Выход 2-го канала
04	$N_{0,2}$		Уставка 2-го канала
.	.		.
.	.		.
.	.		.
$2m-1$	N_m		Выход m -го канала
$2m$	$N_{0,m}$		Уставка m -го канала
01	D_1	Выходы	Признак сравнения 1-го канала
02	D_2		Признак сравнения 2-го канала
.	.		.
.	.		.
.	.		.
m	D_m		Признак сравнения m -го канала

Функциональная схема алгоритма
"Сравнение чисел СЧИ"

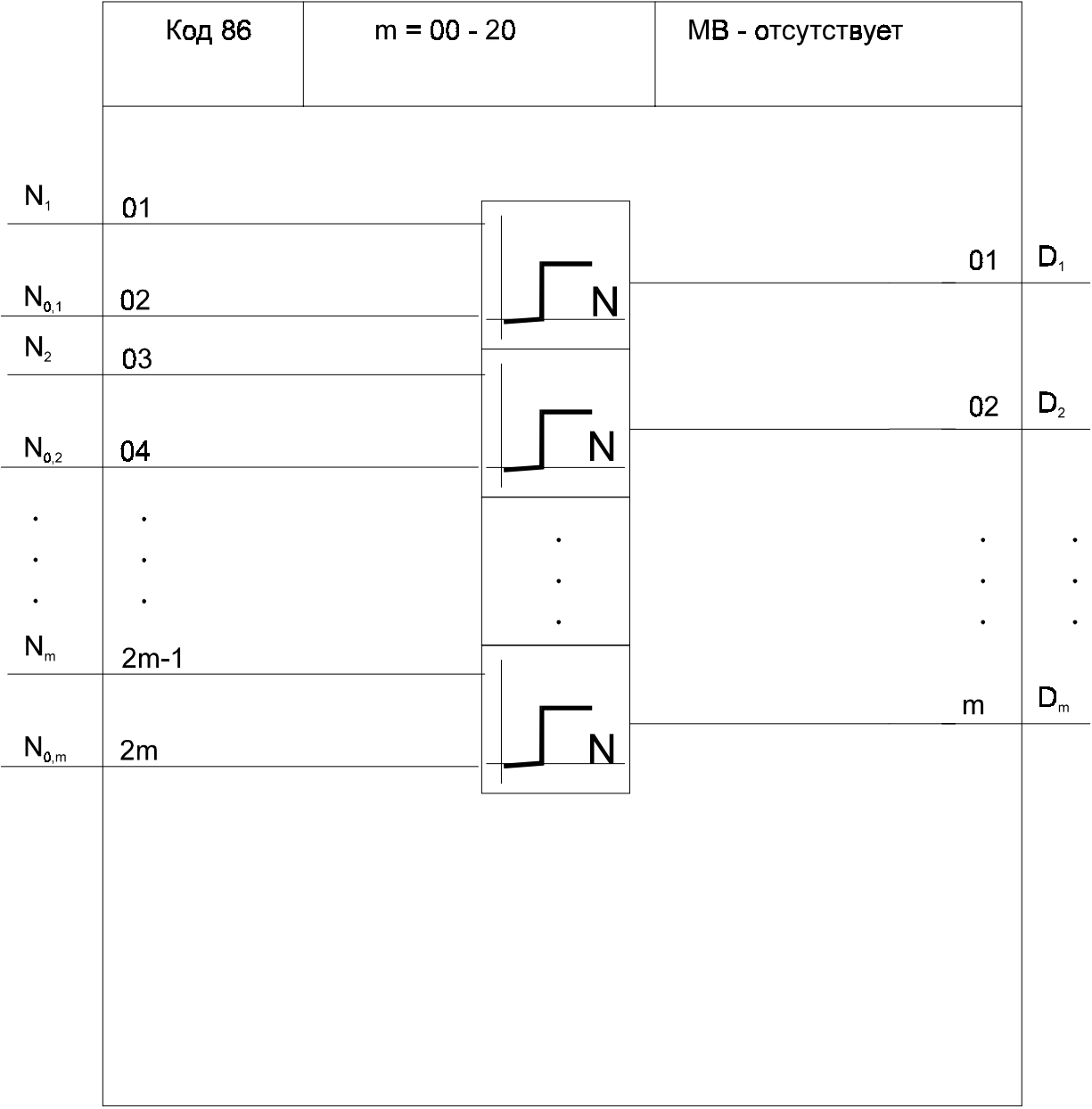


Рис. 77

9.8. ВЧИ (87) - Выделение чисел

Назначение

Алгоритм применяется для выделения из натурального ряда чисел (номеров) заданной группы следующих подряд чисел.

Алгоритм содержит несколько (до 20) независимых каналов выделения.

Описание алгоритма

Алгоритм содержит m независимых каналов выделения чисел, причем $0 \leq m \leq 20$ и задается модификатором (рис. 78). При $m=0$ алгоритм является "пустым".

Каждый канал сравнивает числовую переменную, поступающую на вход N_i , с двумя пороговыми значениями: нижним $N_{н,i}$ и верхним $N_{в,i}$.

Если $N_{н,i} \leq N_i \leq N_{в,i}$, то выходной дискретный сигнал $D_i = 1$, в противном случае $D_i = 0$.

Входы-выходы алгоритма ВЧИ

Таблица 94

Входы-выходы			Назначение
№	Обозн.	Вид	
01	N_1	Входы	Выход 1-го канала
02	$N_{в,1}$		Верхняя уставка 1-го канала
03	$N_{н,1}$		Нижняя уставка 1-го канала
04	N_2		Выход 2-го канала
05	$N_{в,2}$		Верхняя уставка 2-го канала
06	$N_{н,2}$		Нижняя уставка 2-го канала
•	•		•
•	•		•
•	•		•
$3m-2$	N_m		Выход m -го канала
$3m-1$	$N_{в,m}$		Верхняя уставка m -го канала
$3m$	$N_{н,m}$		Нижняя уставка m -го канала
01	D_1	Выходы	Признак выделения 1-го канала
02	D_2		Признак выделения 2-го канала
•	•		•
•	•		•
•	•		•
m	D_m		Признак выделения m -го канала

**Функциональная схема алгоритма
"Выделение чисел ВЧИ"**

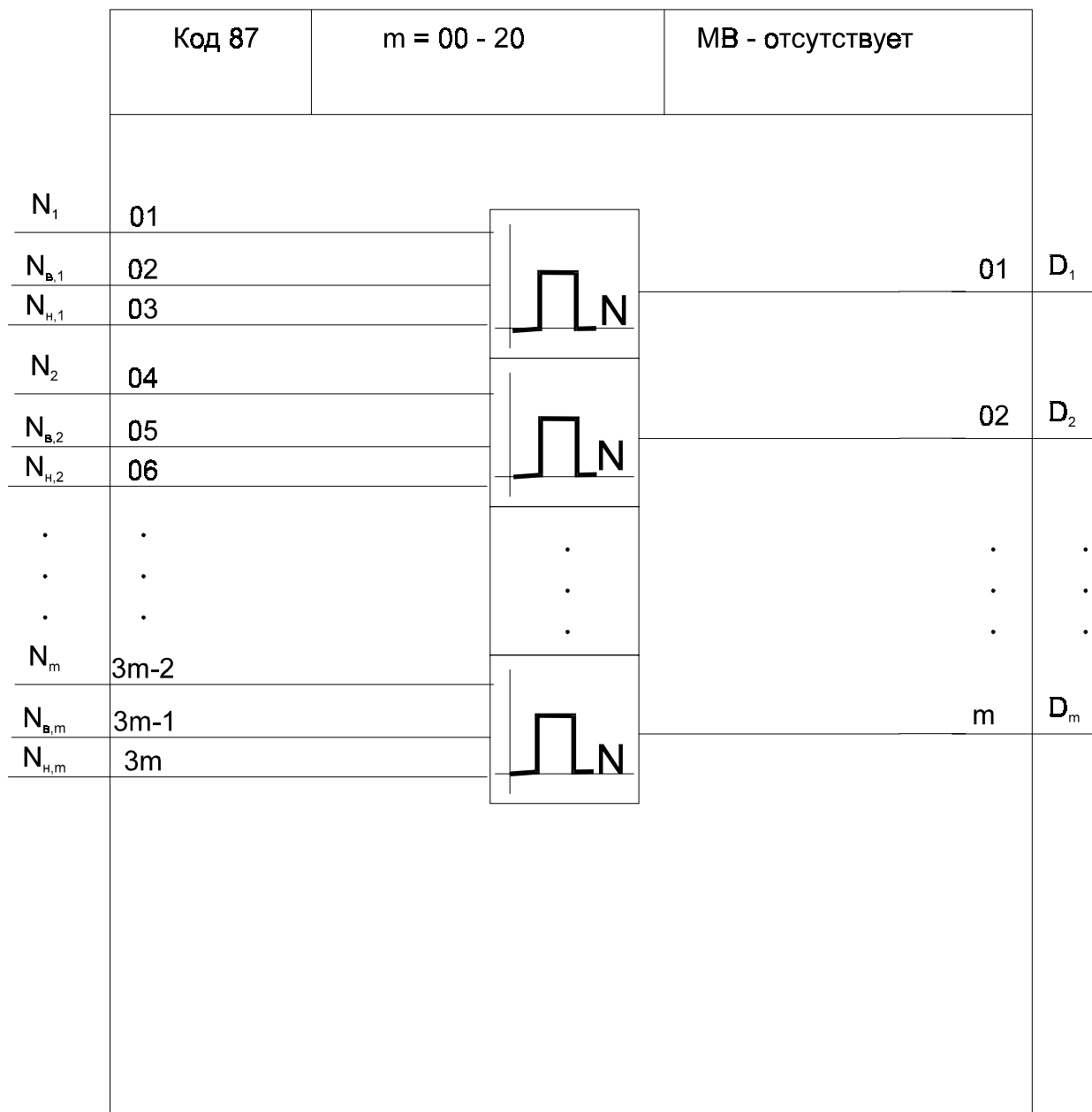


Рис. 78

9.9. УДП (88) - Управление двухпозиционной нагрузкой

Назначение

Алгоритм применяется для управления двухпозиционным исполнительным устройством в тех случаях, когда на одно исполнительное устройство (например, клапан, нагреватель и т. п.) приходят команды из нескольких точек (шагов) логической программы.

Описание алгоритма

Алгоритм имеет две группы входных сигналов: $C_{\text{вкл},j}$ (команды включения) и $C_{\text{выкл},j}$ (команды выключения). В каждой группе число сигналов одинаково и равно $0 < m \leq 16$. Число m задается модификатором (рис. 79).

По переднему фронту (т. е. В момент перехода из состояния лог. 0 в состояние лог. 1) любого сигнала в группе включения устанавливается сигнал $D=1$, при этом состояние остальных входов в группе включения и выключения безразлично (т. е. Они могут находиться в состоянии лог.0 или лог.1). Установленный выходной сигнал запоминается.

Аналогичные действия вызывает приход переднего фронта любого сигнала в группе выключения с той лишь разницей, что на выходе устанавливается (и запоминается) сигнал $D=0$.

При одновременном действии команд на включение и выключение приоритетна команда на выключение.

В момент действия команды на включение (по любому входу) на один цикл формируется выходной сигнал $D_{\text{вкл}}=1$, после чего устанавливается $D_{\text{вкл}}=0$. Аналогично, но в момент действия команды на выключение, формируется выходной сигнал $D_{\text{выкл}}$.

Если входные сигналы должны действовать не по переднему, а по заднему фронту, на соответствующих входах устанавливается инверсия.

Входы-выходы алгоритма УДП

Таблица 95

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{\text{вкл},1}$	Входы	1-я команда на включение
02	$C_{\text{вкл},2}$		2-я команда на включение
•	•		•
•	•		•
•	•		•
m	$C_{\text{вкл},m}$		m-я команда на включение
m+1	$C_{\text{выкл},1}$		1-я команда на выключение
m+2	$C_{\text{выкл},2}$		2-я команда на выключение
•	•		•
•	•		•
•	•		•
2m	$C_{\text{выкл},m}$		m-я команда на выключение
01	D	Выходы	Основной выход
02	$D_{\text{вкл}}$		Команда в группе включения
03	$D_{\text{выкл}}$		Команда в группе выключения

Функциональная схема алгоритма
"Управление двухпозиционной нагрузкой УДП"

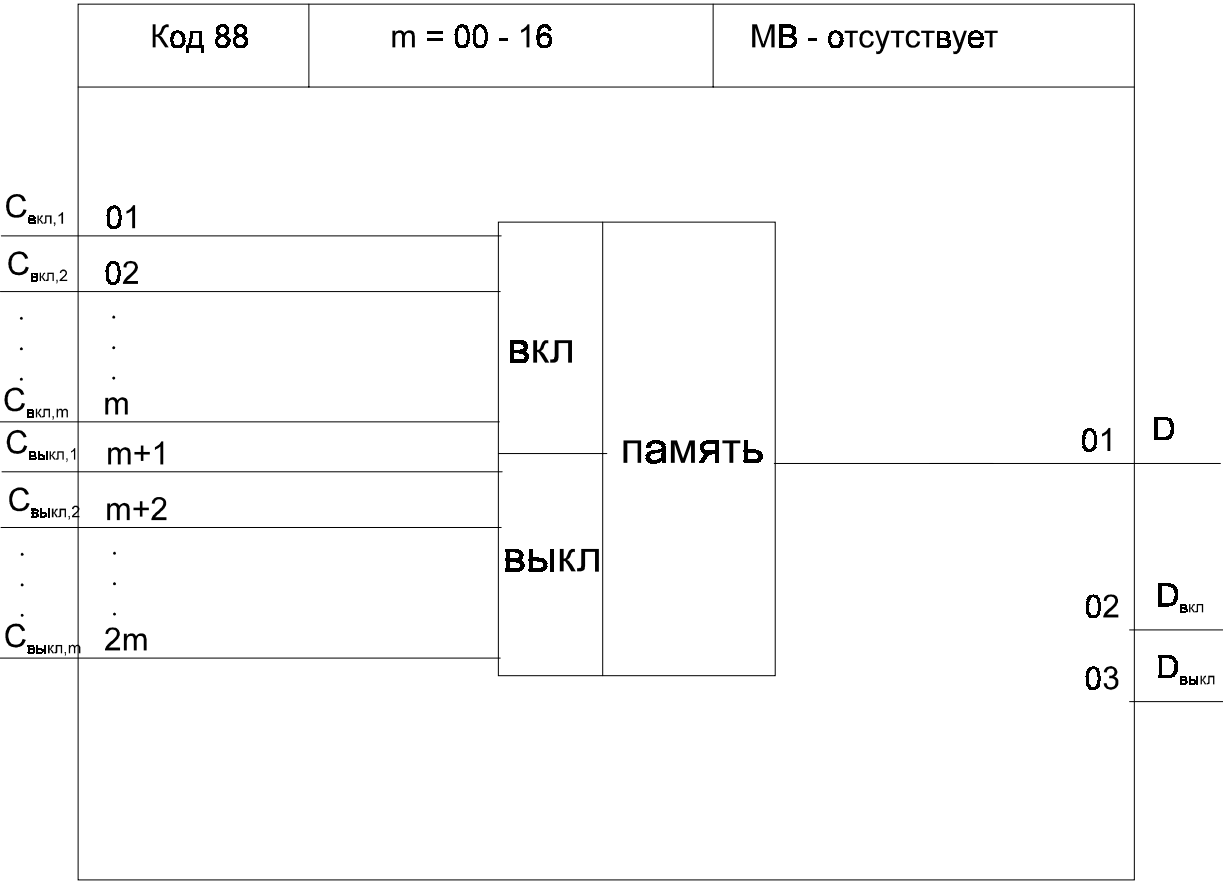


Рис. 79

9.10. УТП (89) - Управление трехпозиционной нагрузкой

Назначение

Алгоритм применяется для логического управления трехпозиционным исполнительным устройством (например, клапаном с моторным управлением) в тех случаях, когда на исполнительное устройство приходят команды из нескольких (до 16) точек (шагов) логической программы.

Описание алгоритма

Алгоритм имеет три группы входных сигналов: группа $C_{отк,j}$ (команды на открытие), группа $C_{зкр,j}$ (команда на закрытие) и группа $C_{ост,j}$ (команда на останов). Каждая группа имеет одинаковое число сигналов $0 \leq m \leq 16$. Число m задается модификатором (рис. 80).

По переднему фронту (т. е. в момент перехода из состояния лог.0 в состояние лог.1) любого сигнала в группе открытия устанавливается выходной сигнал на открытие $D_{отк} = 1$, а выходной сигнал на закрытие $D_{зкр} = 0$, при этом состояние остальных входных сигналов в любой из групп безразлично (т.е. они могут находиться в состоянии лог.0 или лог.1). Установленные значения $D_{отк}$ и $D_{зкр}$ запоминаются.

Аналогично действуют сигналы в группе закрытия с той лишь разницей, что на выходе устанавливаются сигналы $D_{отк} = 0$ и $D_{зкр} = 1$.

По тому же правилу действуют сигналы в группе останова, при этом на выходе устанавливаются сигналы $D_{отк} = D_{зкр} = 0$.

Если в разных группах поступают команды одновременно, действует система приоритетов: группа закрытия приоритетная над группой открытия, группа останова приоритетная над остальными группами.

В момент действия команд в группах открытия, закрытия и останова на один цикл устанавливаются сигналы (лог.1) на выходах соответственно $D_{от}$, $D_{зк}$, $D_{ос}$.

Если команды должны действовать не по переднему фронту, а по заднему фронту, на соответствующих входах устанавливается инверсия.

Входы-выходы алгоритма УТП

Таблица 96

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{отк,1}$	Входы	1-я команда на открытие
02	$C_{отк,2}$		2-я команда на открытие
•	•		•
•	•		•
•	•		•
m	$C_{отк,m}$		m-я команда на открытие
m+1	$C_{зкр,1}$		1-я команда на закрытие
m+2	$C_{зкр,2}$		2-я команда на закрытие
•	•		•
•	•		•
•	•		•
2m	$C_{зкр,m}$		m-я команда на закрытие
2m+1	$C_{ост,1}$		1-я команда на останов
2m+2	$C_{ост,2}$		2-я команда на останов
•	•		•
•	•		•
•	•		•
3m	$C_{ост,m}$		m-я команда на останов
01	$D_{отк}$	Выходы	Выходная команда на открытие
02	$D_{зкр}$		Выходная команда на закрытие
03	$D_{от}$		Признак команды в группе открытия
04	$D_{зк}$		Признак команды в группе закрытия
05	$D_{ос}$		Признак команды в группе останова

**Функциональная схема алгоритма
"Управление трехпозиционной нагрузкой УТП"**

Код 89		m = 00 - 16		МВ - отсутствует	
$C_{отк,1}$	01	открыть	память	01	$D_{отк}$
$C_{отк,2}$	02				
·	·				
·	·				
$C_{отк,m}$	m	заккрыть	память	02	$D_{зкр}$
$C_{зкр,1}$	m+1				
$C_{зкр,2}$	m+2				
·	·				
·	·	остановить	память	03	$D_{ост}$
$C_{зкр,m}$	2m				
$C_{ост,1}$	2m+1				
$C_{ост,2}$	2m+2				
·	·	остановить	память	04	$D_{зк}$
·	·				
$C_{ост,m}$	3m				
				05	$D_{ос}$

Рис. 80

9.11. ШИФ (90) - Шифратор

Назначение

Алгоритм применяется для преобразования набора из нескольких (до 13) дискретных сигналов в один числовой сигнал. В частности, алгоритм применяется в тех случаях, когда через цифровой выход Р-130 (интерфейс) необходимо передать несколько дискретных сигналов. Шифрация позволяет передать несколько (до 13) дискретных сигналов в виде одного пакета и тем самым, повысить скорость обмена цифровой информации. В этом случае алгоритм шифрации применяется в сочетании с алгоритмом цифрового вывода ВИН, причем на приемной стороне используется алгоритм цифрового ввода в сочетании с алгоритмом дешифрации ДЕШ.

Описание алгоритма

Алгоритм преобразует m входных дискретных сигналов X_j в один числовой сигнал Y . Число $0 \leq m \leq 13$ задается модификатором (рис. 81). При $m=0$ алгоритм является "пустым".

Преобразование (шифрация) ведется по формуле:

$$N = C_1 * 2^0 + C_2 * 2^1 + \dots + C_m * 2^{m-1} \quad (51)$$

Входы - выходы алгоритма

Таблица 97

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	1-й вход
02	C_2		2-й вход
.	.		.
.	.		.
.	.		.
m	C_m		m -ый вход
01	N	Выходы	Выход

Функциональная схема алгоритма
"Шифратор ШИФ"

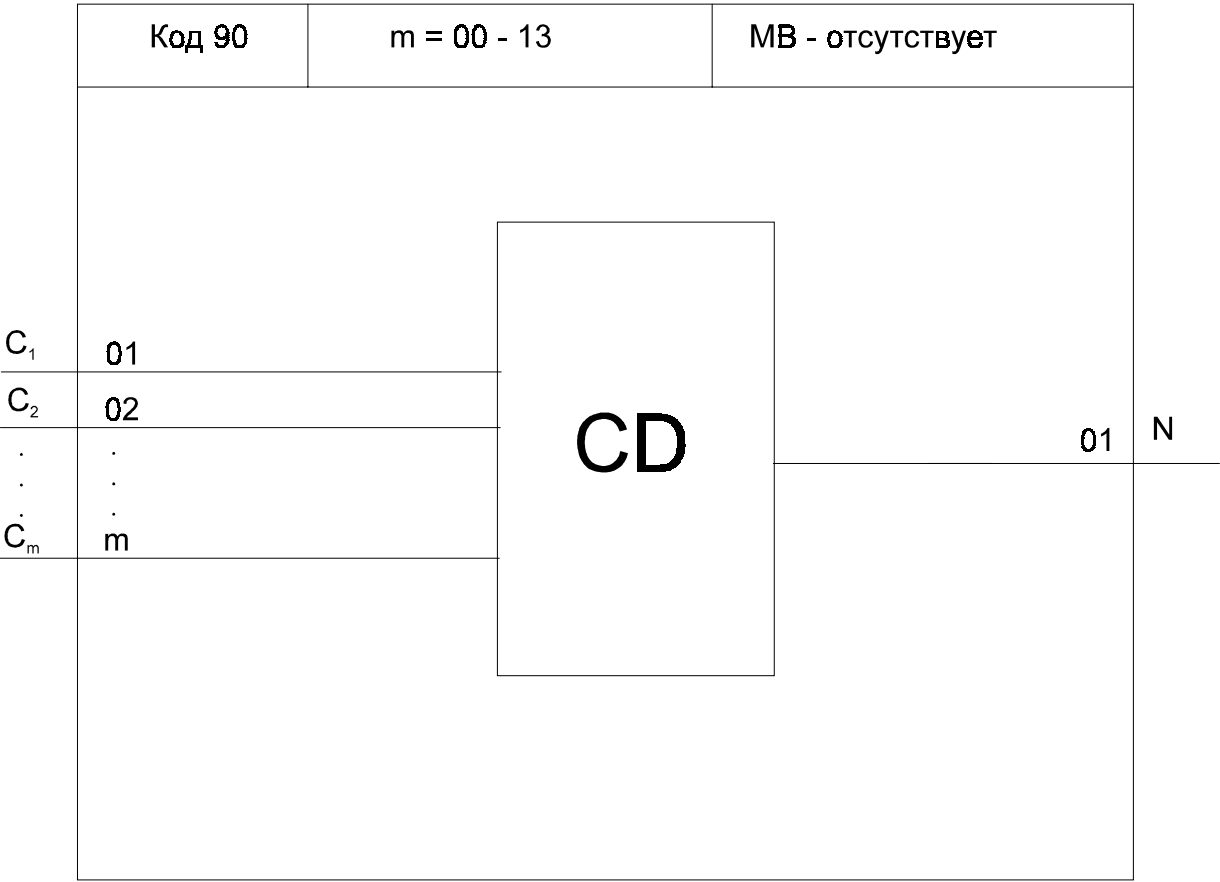


Рис. 81

9.12. ДЕШ (91) - Дешифратор

Назначение

Алгоритм применяется для преобразования числового сигнала в набор дискретных сигналов. В частности, этот алгоритм применяется в сочетании с алгоритмом цифрового ввода в тех случаях когда по интерфейсу принимаются зашифрованные дискретные сигналы (см. алгоритм шифрации ШИФ).

Другой вариант применения алгоритма - в задачах дискретного управления при необходимости преобразования числа, поступающего на вход N , в определенную комбинацию дискретных сигналов на выходе.

Описание алгоритма

Алгоритм преобразует числовую переменную N в m дискретных выходных сигналов D_j . Значение $0 \leq m \leq 13$ и задается модификатором (рис. 82). При $m=0$ алгоритм является "пустым".

Дешифрация осуществляется в соответствии с зависимостью:

$$N = D_1 * 2^0 + D_2 * 2^1 + \dots + D_m * 2^{m-1} \quad (52)$$

Например, если на стороне передатчика дискретные сигналы C_i с помощью алгоритма ШИФ были зашифрованы, а на стороне приемника с помощью алгоритма ДЕШ дешифрованы, то на выходе алгоритма ДЕШ сигналы $D_i = C_i$.

Входы-выходы алгоритма ДЕШ

Таблица 98

Входы-выходы			Назначение
№	Обозн.	Вид	
01	X	Вход	Вход
01	D_1	Выходы	1-й выход
02	D_2		2-й выход
.	.		.
.	.		.
m	D_m		m-й выход

Функциональная схема алгоритма
"Дешифратор ДЕШ"

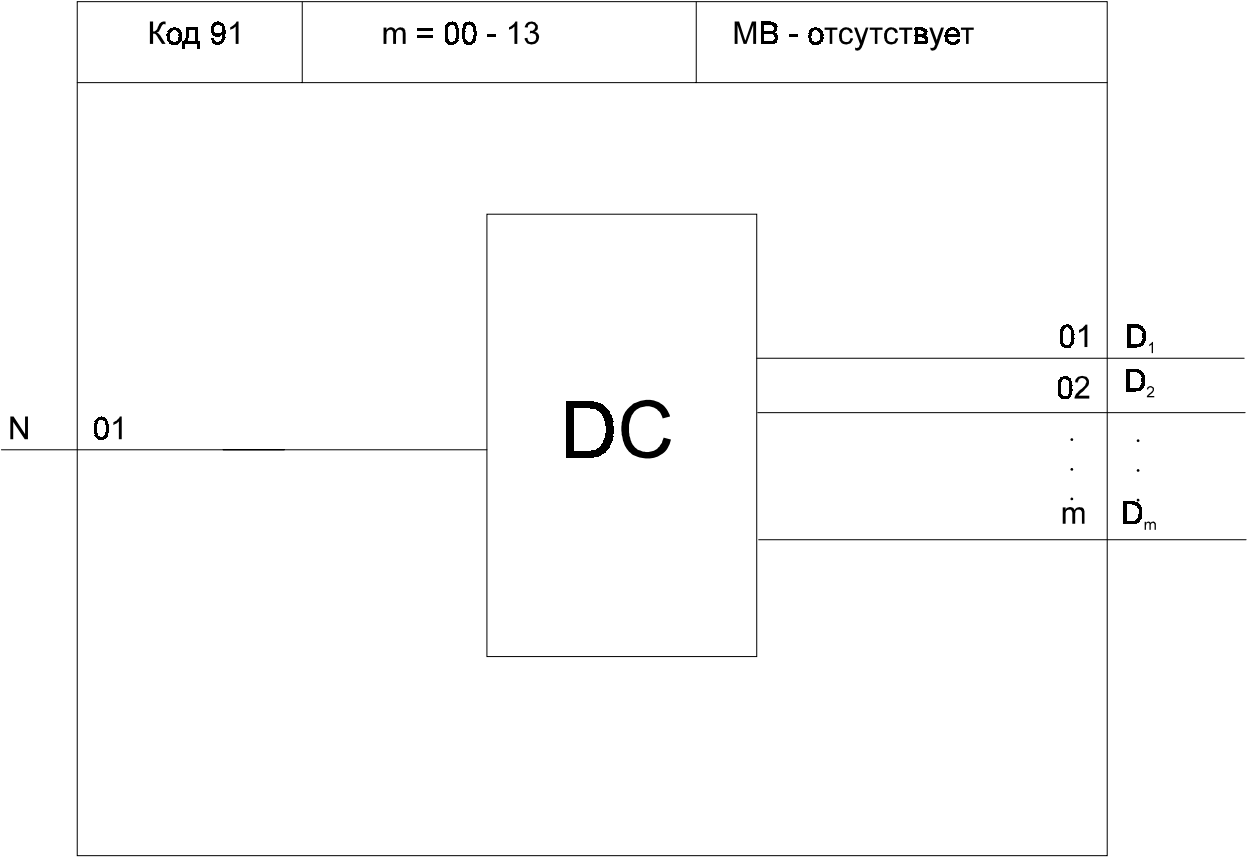


Рис. 82

9.13. ЛОК (92) - Логический контроль

Назначение

Алгоритм применяется для контроля за состоянием нескольких (до 99) дискретных сигналов. Как правило, алгоритм ЛОК используется в сочетании с алгоритмами оперативного контроля ОКЛ и ОКО.

Описание алгоритма

На вход алгоритма подаются m дискретных сигналов, причем $0 \leq m \leq 99$ и задается модификатором (рис. 83). При $m=0$ алгоритм является "пустым".

Если все входные сигналы равны логическому 0, выходные сигналы $N=D=0$.

Если хотя бы один входной сигнал отличен от нуля, то $D=1$, а число N показывает номер входа с этим сигналом. Если несколько входных сигналов отличны от нуля, то N - младший номер из этой группы сигналов.

Обычно выход N соединяется с входом $N_{оп}$ алгоритма ОКЛ или $N_{ок}$ алгоритма ОКО. В этом случае с помощью лицевой панели можно определить номер отличного от нуля входного сигнала, поданного на алгоритм ЛОК.

Входы-выходы алгоритма ЛОК

Таблица 99

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Сигнал на 1 -м входе
02	C_2		Сигнал на 2 -м входе
.	.		.
.	.		.
.	.		.
m	C_m		Сигнал на m -м входе
01	N	Выходы	Номер входного сигнала, отличного от нуля
02	D		Признак отличия от нуля одного из входных сигналов

Функциональная схема алгоритма
"Логический контроль ЛОК"

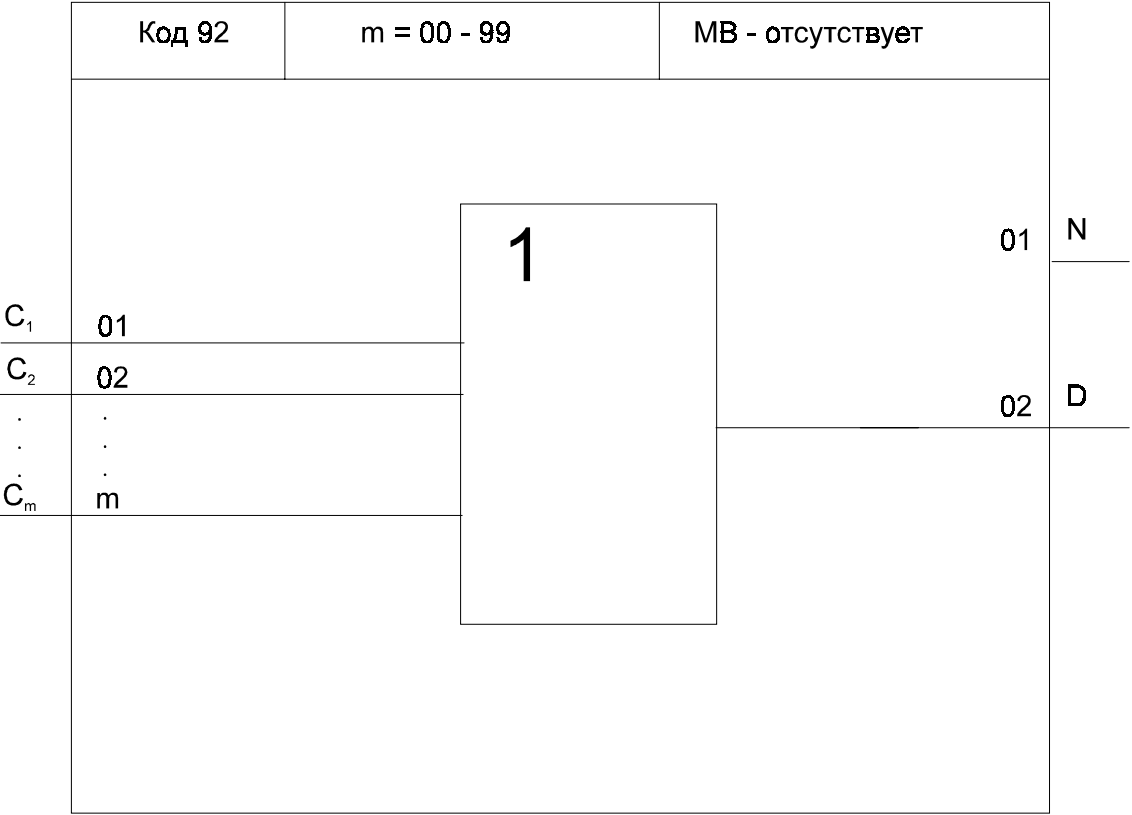


Рис. 83

10. ГРУППОВОЕ НЕПРЕРЫВНО-ДИСКРЕТНОЕ УПРАВЛЕНИЕ

10.1. ШАП (94) - шаговая программа

Назначение

Алгоритм применяется для организации логической последовательности включения и отключения дискретных команд с возможностью контроля выполнения команд.

Алгоритм применяется в сочетании с алгоритмом ОИД.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Описание алгоритма.

Алгоритм ШАП имеет m шагов, где $0 \leq m \leq 9$ и задается модификатором (рис.84). Если программа находилась в состоянии "сброс" и была пущена, начинает выполняться 1-й шаг алгоритма. Вслед за ним программа может перейти к следующему шагу или перейти к другому заданному шагу.

Когда выполняется последний шаг программы, она переходит в состояние "конец программы".

При выполнении шага на его выходе D_i формируется сигнал $D_i=1$. Этот сигнал сохраняется, пока программа не будет переведена в состояние "сброс".

Работа Шага.

Все шаги алгоритма ШАП функционально идентичны.

Каждый шаг имеет 3 входа и один выход.

Вход C шага контролирует условие выполнения шага. Считается, что при $C = 0$ условий нет, а при $C = 1$ условия имеются.

На входе T шага задается время, в течение которого продолжается контроль условий C . Если до истечения времени T условие на входе C оказалось выполненным (т.е. $C = 1$), формируется выходной сигнал шала $D = 1$ и программа переходит к следующему по номеру шагу. Если выполняемый шаг был последним, то программа переходит в состояние "конец программы".

Если время T истекло, а условие на входе C осталось невыполненным (т.е. $C = 0$), поведение программы определяется параметром на входе N шага. При этом возможны следующие четыре варианта.

- 1). Если $N = 0$, то программа, невзирая на то, что $C = 0$, формирует выходной сигнал $D = 1$ и переходит к следующему по номеру шагу. Этот вариант используется для реализации выдержки времени.
- 2). Если $0 < N \leq N_m$, где N_m - максимальный номер шага, заданный модификатором, то выход шага не формируется и программа переходит к выполнению шага N . Этот вариант используется для организации условных и безусловных переходов.
- 3). Если $N > 0$ и одновременно $N > N_m$ (т.е. шага с номером N не существует), то программа переходит в состояние "ожидание". Этот вариант используется для останова программы и привлечения внимания оператора к тому, что требуемые условия не выполнены.
- 4). Если $N < 0$, то происходит обнуление шага $|N|$ (т.е. в шаге $|N|$ устанавливается $D=0$), на выходе данного шага сигнал остается равным нулю и программа переходит к выполнению следующего шага, если данный шаг не последний, и в состояние "конец программы", если данный шаг последний. Если при указанных выше условиях установить $N < -N_m$, то данный шаг никакого действия не выполняет.

Блок-схема работы шага представлена на рис.85.

Выходы алгоритма

Помимо выходов шагов D_i алгоритм ШАП имеет пять дополнительных выходов.

Выход $N_{ш}$ соединяется со входом $N_{шаг}$ алгоритма ОКД. При контроле сигнала на выходе $T_{ост}$ с помощью пульта настройки на пульте индицируется номер текущего (выполняемого в данный момент) шага.

На выходе $T_{ост}$ формируется время, оставшееся до истечения контрольного времени текущего шага.

На выходе N формируется параметр, равный параметру N на входе текущего шага.

На выходе $D_{ш}$ устанавливается сигнал $D = 1$ в тот момент, когда на выходе текущего шага формируется сигнал $D = 1$.

На выходе $T_{тек}$ формируется текущее время, прошедшее от начала работы текущего шага. Очевидно, что $T_{тек} = T - T_{ост}$, где T - контрольное время установленное на входе текущего шага.

Блок-схема работы шага

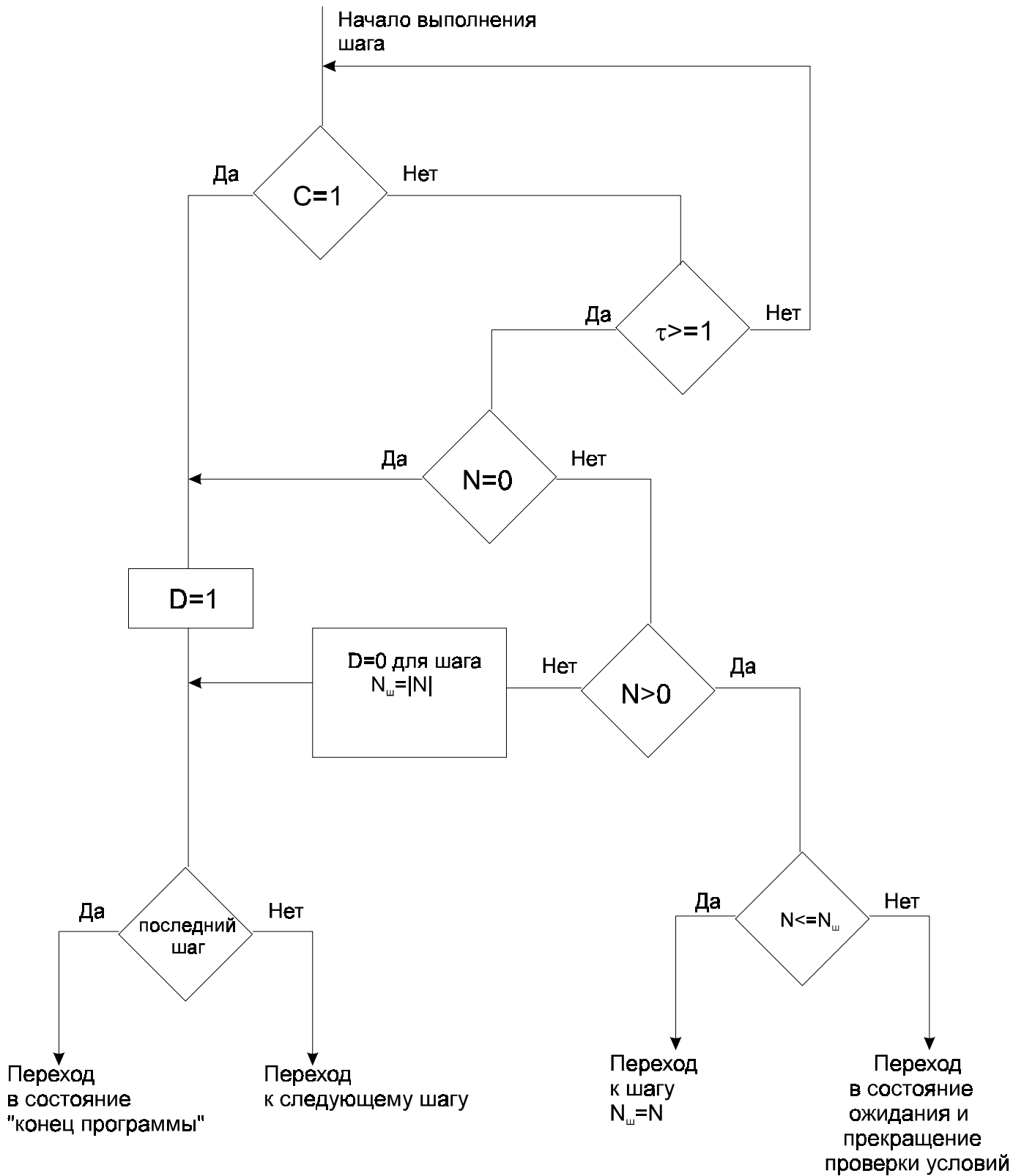


Рис. 84

Входы - выходы алгоритма ШАЛ

Таблица 100

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Условие выполнения 1-го шага
02	T_1		Контрольное время 1-го шага
03	N_1		Параметр 1-го шага
04	C_2		Условие выполнения 2-го шага
05	T_2		Контрольное время 2-го шага
06	N_2		Параметр 2-го шага
.	.		.
.	.		.
.	.		.
3m-2	C_m		Условие выполнения m-го шага
3m-1	T_m		Контрольное время m-го шага
3m	N_m		Параметр m-го шага
01	$N_{ш}$	Выходы	Номер выполняемого шага
02	$T_{ост}$		Время, оставшееся до истечения контрольного времени выполняемого шага
03	N		Параметр, равный параметру N на входе выполняемого шага
04	$D_{ш}$		Признак включения выхода выполняемого шага ($D=1$)
05	$T_{тек}$		Текущее время, прошедшее от начала работы выполняемого шага

Функциональная схема алгоритма
"Шаговая программа ШАП"

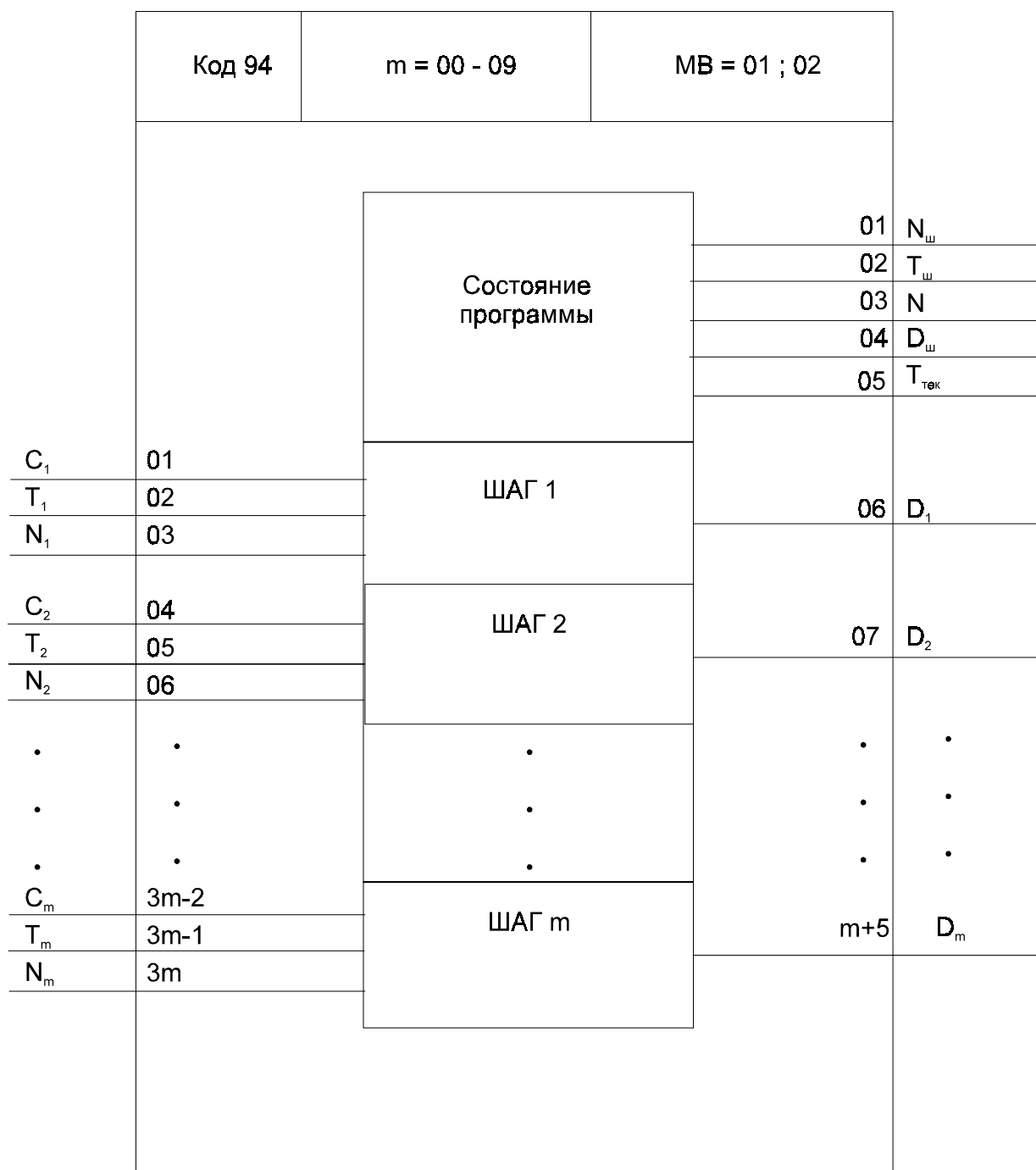


Рис. 85

10.2. ГРА (95) - групповое управление аналоговыми сигналами

Назначение

Алгоритм используется для управления и контроля за аналоговыми сигналами.

Алгоритм применяется в сочетании с алгоритмом ОКД и совместно с ним позволяет в каждом из девяти звеньев алгоритма ГРА переключать цепь внешнего аналогового сигнала на ручной задатчик и изменять сигнал этого задатчика вручную с помощью клавиш лицевой панели контроллера.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Основные функции

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 9$ и задается модификатором (рис. 86).

Каждое звено содержит переключатель, с помощью которого выход звена Y подключается либо к внешней цепи X (это может быть выход любого алгоблока, константа или коэффициент), либо к встроенному узлу аналогового ручного задатчика.

В каждом звене на входах W_0 и W_{100} задаются такие технические единицы, в которых аналоговый выходной сигнал данного канала должен контролироваться по цифровому индикатору лицевой панели (см. описание алгоритма ОКД).

Алгоритм содержит четыре выхода, характеризующие параметры того звена, которое с помощью клавиш лицевой панели выбрано для управления и индикации.

Выход N подключается к одному из входов $Z_1 - Z_5$ алгоритма ОКД. После такого соединения с помощью клавиш лицевой панели можно выбирать одно из звеньев алгоритма ГРА, изменять режим работы этого звена, в режиме ручного задания "руз" изменять сигнал задания вручную, а также по цифровому индикатору контролировать выходной сигнал Y выбранного звена. (Специально соединять выходы Y с алгоритмом (ОКД не требуется). При контроле сигнала на выходе N с помощью пульта настройки на пульте индицируется номер выбранного звена. Все сигналы по цифровому индикатору лицевой панели индицируются в процентах.

На выходе $D_{руз}$ формируется признак режима, в котором работает выбранное звено. При внешнем режиме ("внш") $D_{руз}=0$ в режиме ручного задатчика ("руз") $D_{руз} = 1$.

На выходе W_0 и W_{100} формируются сигналы, равные одноименным параметрам, установленным на входе выбранного звена.

В режиме "внш" внутренний задатчик каждого звена отслеживает сигнал X на входе "своего" звена. В результате переключение в режим "руз" выполняется безударно.

Входы - выходы алгоритма ГРА

Таблица 101

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$W_{0,1}$	Входы	Технические единицы звена 1, соответствующие 0% аналогового сигнала
02	$W_{100,1}$		То же, что $W_{0,1}$, но для 100% аналогового сигнала
03	X_1		Внешний сигнал звена 1
.	.		.
.	.		.
.	.		.
$3m-2$	$W_{0,m}$		Технические единицы звена m , соответствующие 0% аналогового сигнала
$3m-1$	$W_{100,1}$	Выходы	То же, что $W_{0,m}$, но для 100% аналогового сигнала
$3m$	X_m		Внешний сигнал звена m
01	N		Номер выбранного звена
02	$D_{руз}$		Признак ручного задания в выбранном режиме
03	W_0		Технические единицы выбранного звена, соответствующие 0% аналогового сигнала
04	W_{100}		То же, что W_0 , но для 100% аналогового сигнала
05	Y_1		Выход звена 1
06	Y_2		Выход звена 2
.	.		.
.	.		.
.	.		.
$m+4$	Y_m		Выход звена m

Функциональная схема алгоритма
"Групповое управление аналоговыми сигналами ГРА"

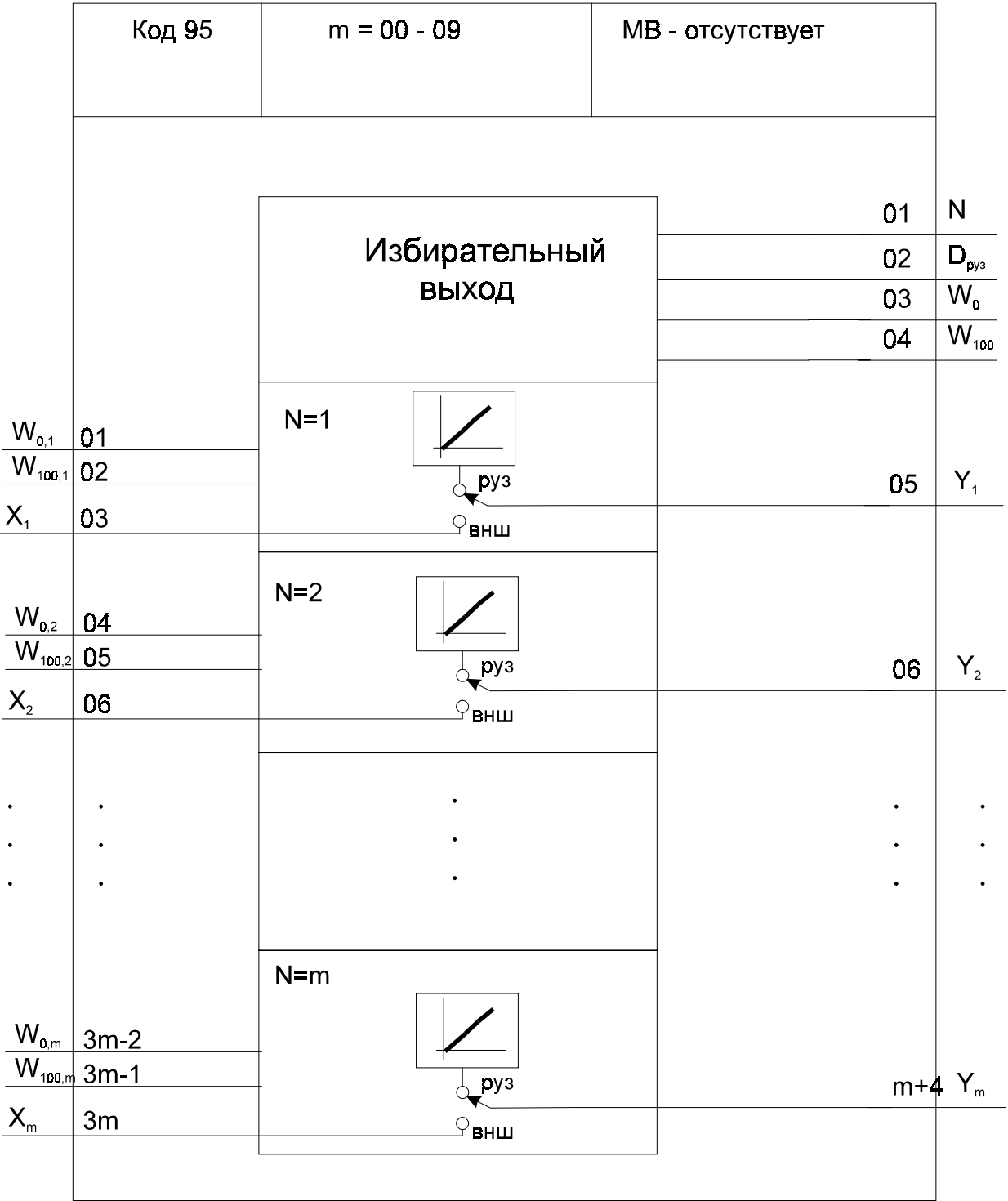


Рис. 86

10.3. ГВД (96) - групповое управление входными дискретными сигналами

Назначение

Алгоритм позволяет избирательно подменять до 9 внешних дискретных сигналов дискретными сигналами, состояние которых можно изменять вручную.

Алгоритм применяется в сочетании с алгоритмом ОКД.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Описание алгоритма

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 9$ и задается модификатором (рис.87).

Каждое звено содержит переключатель, с помощью которого выход звена D подключается либо к внешней цепи C (это может быть выход любого алгоблока, константа или коэффициент), либо к встроенному узлу дискретного ручного задатчика. С помощью этого задатчика сигнал на выходе D выбранного звена устанавливается в состояние $D = 0$ или $D = 1$.

Алгоритм содержит два выхода, характеризующие параметры того звена, которое с помощью клавиш лицевой панели выбрано для управления и индикации»

Выход N подключается к одному из входов $Z_1 - Z$ алгоритма ОКД. После такого соединения с помощью клавиш лицевой панели можно выбирать одно из звеньев алгоритма ГВД, изменять режим работы этого звена, в режиме ручного задания "руз" изменять состояние выхода D выбранного звена вручную, а также по цифровому индикатору контролировать состояние этого выхода:

при $D=0$ и $D=1$ на цифровой индикатор выводится значение соответственно 0 и 1 (специально соединять выходы D с алгоритмом ОКД не требуется). При контроле сигнала на выходе с помощью пульта настройки индицируется номер выбранного звена.

На выходе $D_{руз}$ фиксируется признак режима, в котором работает выбранное звено. При внешнем режиме ("внш") $D_{руз} = 0$, в режиме ручного задатчика ("руз") $D_{руз} = 1$.

В режиме "внш" внутренний задатчик каждого звена отслеживает состояние дискретного сигнала C на входе "своего" звена. В результате переключение в режим "руз" выполняется безударно.

Входы-выходы алгоритма ГВД

Таблица 102

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Внешний сигнал звена 1
02	C_2		Внешний сигнал звена 2
.	.		.
.	.		.
.	.		.
m	C_m	Выходы	Внешний сигнал звена m
01	N		Номер выбранного звена
02	$D_{руз}$		Признак ручного задания в выбранном режиме
03	D_1		Выход звена 1
04	D_2		Выход звена 2
.	.		.
.	.		.
.	.		.
$m+2$	D_m		Выход звена m

Функциональная схема алгоритма
"Групповое управление входами дискретными ГВД"

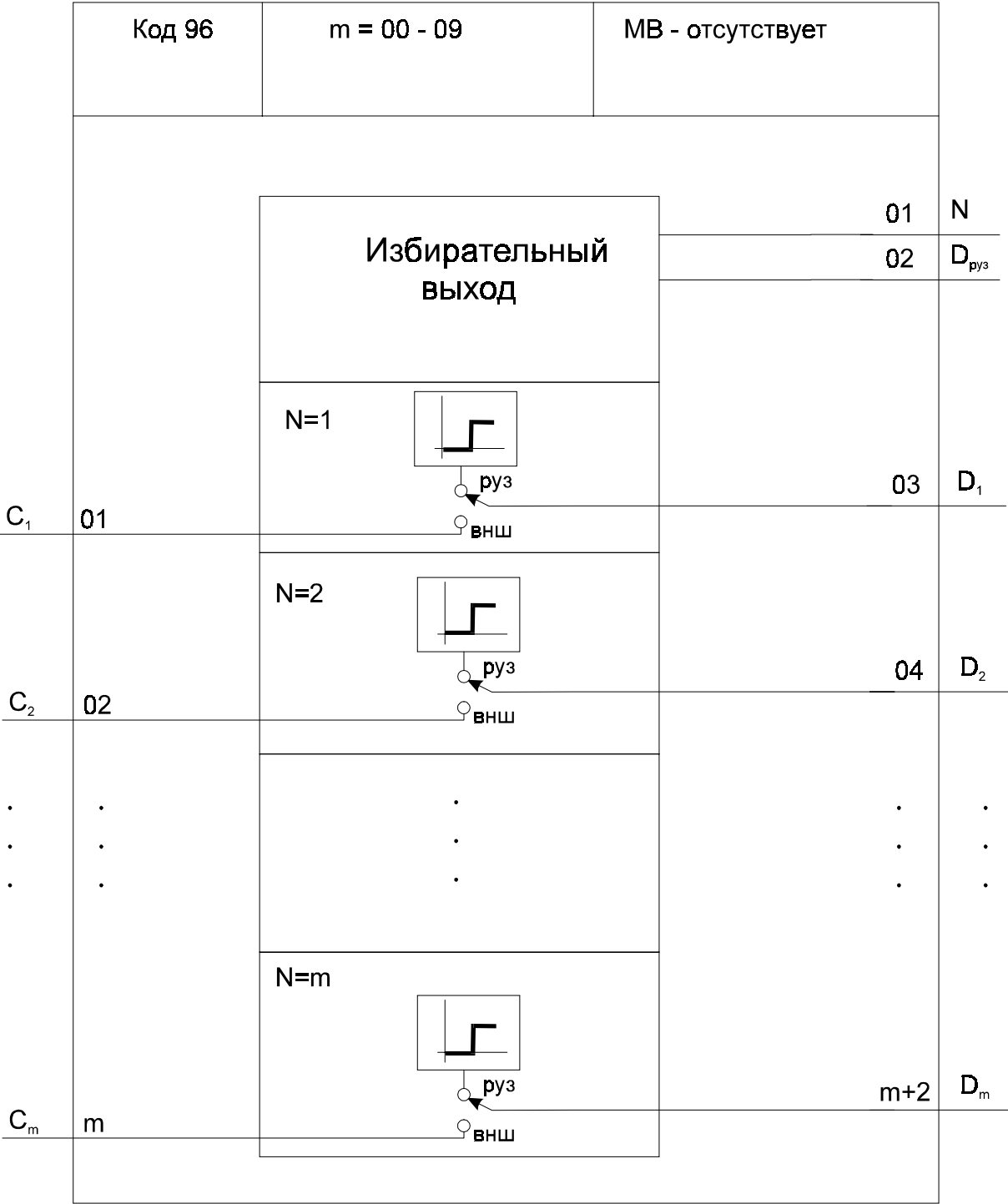


Рис. 87

10.4. ГДВ (97) - групповое управление дискретными выходными сигналами

Назначение

Алгоритм позволяет избирательно отключать до 9 дискретных выходов контроллера от алгоблоков, формирующих эти выходы, и включать или отключать дискретные выходы вручную.

Алгоритм применяется в сочетании с алгоритмом ОКД.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Описание алгоритма

Структура и возможности алгоритма ГДВ полностью соответствуют алгоритму ГВД (рис.88). Разница заключается лишь в том, что ручное управление дискретным выходом ведется с помощью другой комбинации клавиш на лицевой панели контроллера (используются клавиши  + ,  вместо клавиш  + , , которые применяются при работе с алгоритмом ГВД). Кроме того вместо наименований режимов - внешний сигнал ("внш") и ручной задатчик ("руз"), используемых в алгоритме ГВД, при применении алгоритма ГДВ используются наименования: автоматический режим ("авт"), ручной режим ("руч").

Функциональная схема алгоритма
"Групповое управление дискретными выходами ГДВ"

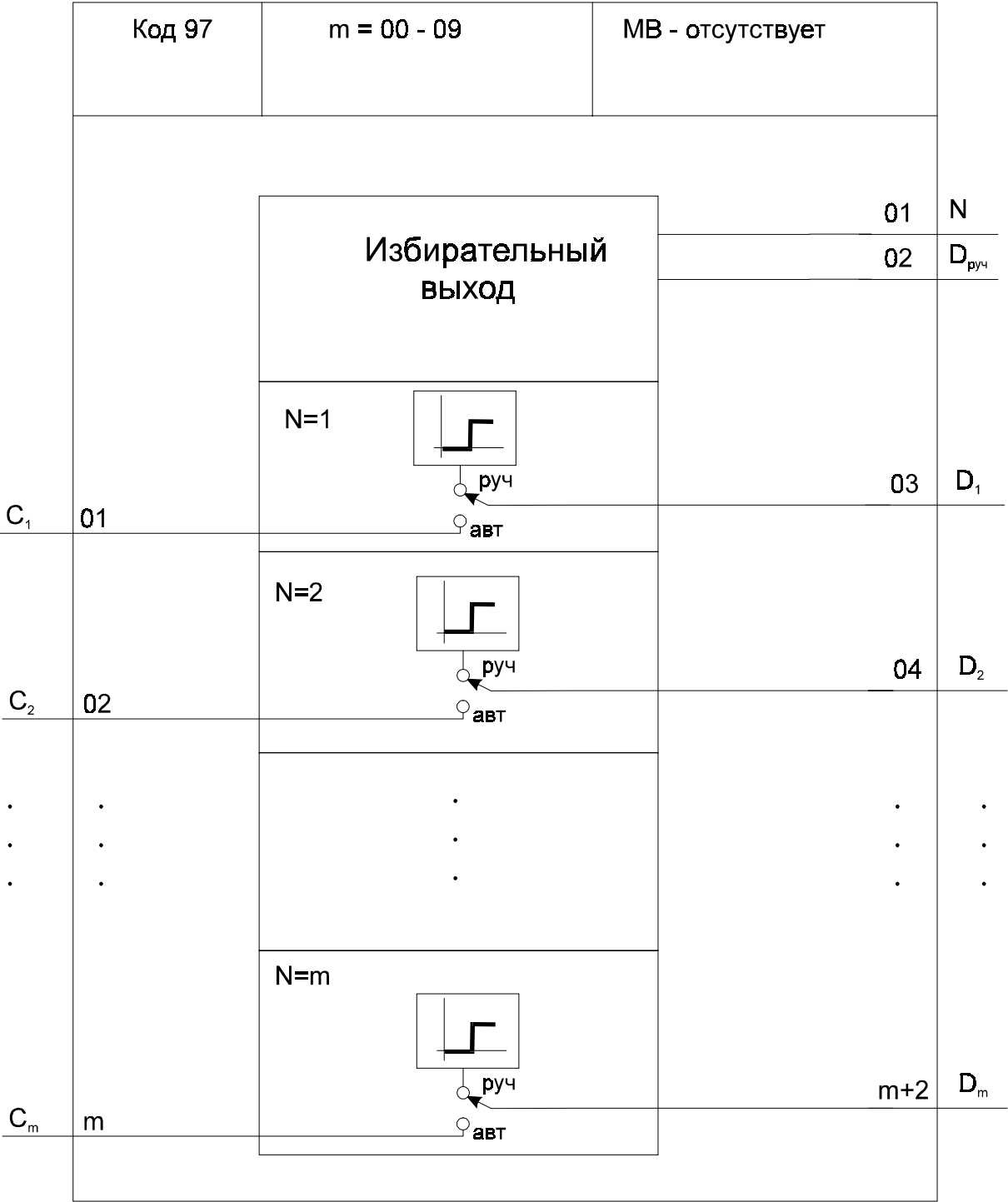


Рис. 88

Входы - выходы алгоритма ГДВ

Таблица 103

Входы-выходы			Назначение
№	Обозн.	Вид	
01	C_1	Входы	Вход звена 1
02	C_2		Вход звена 2
.	.		.
.	.		.
.	.		.
m	C_m		Вход звена m
01	N	Выходы	Номер выбранного звена
02	$D_{руч}$		Признак ручного режима в выбранном режиме
03	D_1		Выход звена 1
04	D_2		Выход звена 2
.	.		.
.	.		.
.	.		.
m+2	D_m		Выход звена m

10.5. ГРУ (98) - групповое ручное управление

Назначение

Алгоритмы используются для избирательного ручного управления аналоговыми и (или) импульсными выходами контроллера.

Алгоритм применяется в сочетании с алгоритмом ОКД и совместно с ним позволяет в каждом из девяти звеньев алгоритма ГРУ, переключать аналоговый или импульсный выход контроллера на режим ручного или автоматического управления и управлять исполнительным механизмом вручную с помощью клавиши лицевой панели контроллера.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Описание алгоритма

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 9$ и задается модификатором (рис.89).

Каждое звено содержит переключатель с помощью которого устанавливается автоматический ("авт") или ручной ("руч") режим управления.

Каждое звено имеет три входа. С помощью входа $C_{имп}$, на котором может задаваться только константа, устанавливается вид регулятора: при $C_{имп} = 0$ регулятор имеет аналоговый выход, при $C_{имп} = 1$ - импульсный выход.

На вход $X_{им}$ при импульсном выходе подается сигнал, характеризующий положение исполнительного механизма.

На вход X подается сигнал, который в автоматическом режиме поступает на выход Y звена, а в ручном режиме заменяется на сигнал ручного управления.

Алгоритм имеет 4 выхода, характеризующие параметры выбранного звена.

Выход N подключается к одному из входов $Z_1 - Z_5$ алгоритма ОКД. После такого соединения с помощью клавиш лицевой панели можно выбирать одно из звеньев алгоритма ГРУ, изменять режим работы этого звена, в ручном режиме "руч" управлять исполнительным механизмом вручную, а также по цифровому индикатору контролировать положение исполнительного механизма (специально подавать сигнал, характеризующий положение исполнительного механизма, на алгоритм ГРУ не требуется). При контроле сигнала на выходе N с помощью пульта настройки на пульте индицируется номер выбранного звена. То, такой сигнал характеризует положение исполнительного механизма и выводится на цифровой индикатор лицевой панели, зависит от вида регулятора. При $C_{имп} = 0$ (аналоговый регулятор) на цифровой индикатор выводится сигнал Y выбранного звена, при $C_{имп} = 1$ (импульсный регулятор) на цифровой индикатор выводится сигнал, поступающий на вход $X_{им}$ выбранного звена. В любом случае эти сигналы индицируются в процентах.

На выходе $D_{руч}$ формируется признак режима, в котором работает выбранное звено, при автоматическом режиме $D_{руч} = 0$, при ручном режиме $D_{руч} = 1$.

На выходах $D_{имп}$ и $Y_{им}$ сигналы равны сигналам соответственно $C_{имп}$ и $X_{им}$ на входе выбранного звена.

На выход Y каждого звена в автоматическом режиме поступает входной сигнал X соответствующего звена независимо от типа регулятора.

Режимы работы

Каждое звено алгоритма ГРУ инициирует команду отключения. Эта команда формируется, когда звено работает в ручном режиме. При наличии следящих алгоритмов по конфигурации связанных с выходами X звеньев алгоритма ГРУ, на входе этих звеньев формируется команда отключения со значением начальных условий, равных текущему значению сигнала Y на выходе соответствующего звена.

В результате переход на автоматический режим выполняется безударно. В автоматическом режиме узел ручного управления при аналоговом регуляторе отслеживает выходной сигнал Y "своего" звена, при импульсном регуляторе этот узел обнуляется. В обоих случаях переход на ручной режим выполняется безударно.

Входы - выходы алгоритма ГРУ

Таблица 104

Входы-выходы			Назначение
№	Обозн.	Вид	
01	$C_{имп,m}$	Входы	Признак импульсного регулирования звена 1
02	$X_{им,m}$		Положение исполнительного механизма при импульсном регулировании звена 1
03	X_1		Входной сигнал звена 1
.	.		.
.	.		.
.	.		.
3m-2	$C_{имп,m}$		Признак импульсного регулирования звена m
3m-1	$X_{им,m}$		Положение исполнительного механизма при импульсном регулировании звена m

3m	X_m	Выходы	Входной сигнал звена m
01	N		Номер выбранного звена
02	$D_{руч}$		Признак ручного режима в выбранном звене
03	$D_{имп}$		Признак импульсного регулирования в выбранном звене
04	$Y_{им}$		Положение импульсного механизма при импульсном регулировании в выбранном звене
05	Y_1		Выход звена 1
06	Y_2		Выход звена 2
.	.		.
.	.		.
m+4	Y_m		Выход звена m

Функциональная схема алгоритма
"Групповое ручное управление ГРУ"

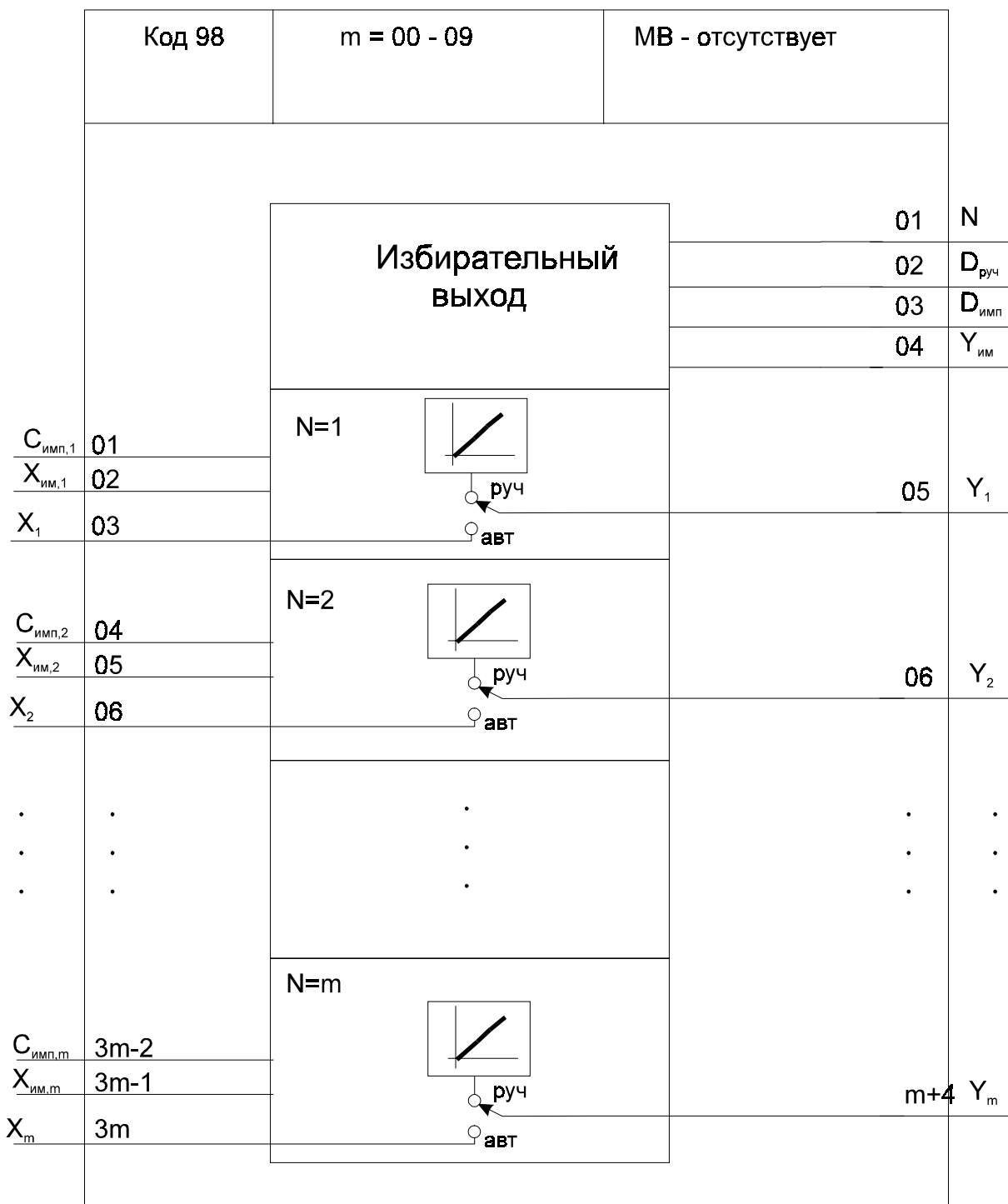


Рис. 89

10.6. ГРК (99) - групповой контроль

Назначение

Алгоритм предназначен для организации избирательного контроля сигналов различного формата - аналоговых, временных, числовых и т. д.

Алгоритм применяется в сочетании с алгоритмом ОКД.

Алгоритм входит в состав библиотеки лишь непрерывно-дискретной модели контроллера.

Функции алгоритма

Алгоритм содержит m идентичных независимых звеньев, где $0 \leq m \leq 9$ и задается модификатором (рис.90).

Каждое звено предназначено для контроля одного сигнала, которое подается на вход Z звена. Формат этого сигнала задается индивидуально в каждом звене параметром на входе N_z (см. табл. 7.1 в описании алгоритма ОКД). Если формат аналоговый, для него дополнительно на входах W_0 и W_{100} каждого звена устанавливаются технические единицы, в которых контролируется сигнал, при этом величины W_0 и W_{100} равны техническим единицам, соответствующим 0 % и 100 % аналогового сигнала (см. также формулу на листе 326 в описании алгоритма ОКД).

Алгоритм содержит пять выходов, характеризующих параметры выбранного звена. Выход N подключается к одному из входов $Z_1 - Z_3$ алгоритма ОКД. После такого соединения с помощью клавиш лицевой панели можно выбрать звено алгоритма ГРК, сигнал на входе которого требуется вывести на цифровой индикатор лицевой панели. При контроле сигнала на выходе N с помощью пульта настройки на пульте индицируется номер выбранного звена.

На остальных выходах Z , N_z , W_0 , W_{100} алгоритма ГРК формируются сигналы, равные одноименным параметрам на входах выбранного звена.

Входы - выходы алгоритма ГРК

Таблица 105

Входы-выходы			Назначение
№	Обозн.	Вид	
01	Z_1	Входы	Контролируемый сигнал звена 1
02	$N_{z,1}$		Формат контролируемого сигнала звена 1
03	$W_{0,1}$		Технические единицы, соответствующие 0% контролируемого сигнала звена 1
04	$W_{100,1}$		То же, что $W_{0,1}$, но для 100% контролируемого сигнала
.	.		.
.	.		.
.	.		.
4m-3	Z_m		Контролируемый сигнал звена m
4m-2	$N_{z,m}$		Формат контролируемого сигнала звена m
4m-1	$W_{0,m}$		Технические единицы, соответствующие 0% контролируемого сигнала звена m
4m	$W_{100,m}$		То же, что $W_{0,m}$, но для 100% контролируемого сигнала
01	N	Выходы	Номер выбранного звена
02	Z		Контролируемый сигнал выбранного звена
03	N_z		Формат контролируемого сигнала выбранного звена
04	W_0		Технические единицы, соответствующие 0% контролируемого сигнала выбранного звена
05	W_{100}		То же, что W_0 , но для 100% контролируемого сигнала выбранного звена

Функциональная схема алгоритма
"Групповой контроль ГРК"

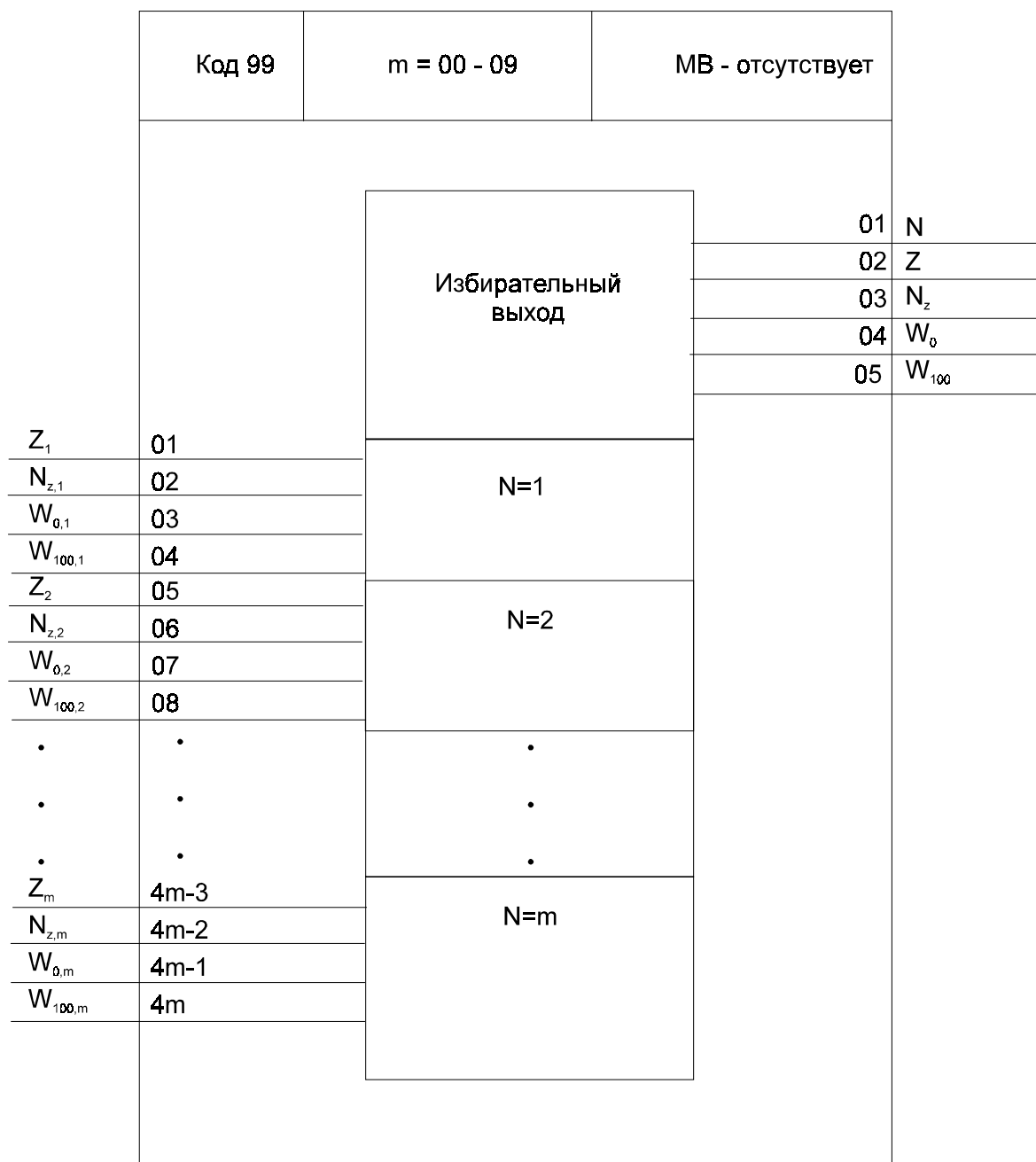


Рис. 90

11. Стандартные конфигурации

11.1. РЕГА(01) - Регулятор аналоговый (стандартная конфигурация).

Назначение

Стандартная конфигурация "Регулятор аналоговый" РЕГА" (в дальнейшем РЕГА) предназначена для построения контура регулирования с аналоговым выходным сигналом.

РЕГА включена в состав библиотеки только регулирующей модели контроллера.

функциональное описание.

РЕГА состоит из шести алгоритмов (см. табл.106).

Алгоритмы связаны между собой конфигурацией, показанной на рис. 94

Регулятор помимо алгоритма регулирования содержит ручной задатчик, алгоритм ручного управления, алгоритм оперативного контроля. Входной сигнал подается на аналоговый вход контроллера (вход 01, группа А), а выходной сигнал формируется на аналоговом выходе контроллера (выход 01, группа А).

РЕГА помещается только в первый контур. Остальные три контура (алгоблоки 02-04) оставлены свободными, поэтому при необходимости в них могут помещаться нужные алгоритмы.

РЕГА помимо функции регулирования обеспечивает также функции оперативного управления в первом контуре, а именно ручное изменение сигнала задания, переход на ручной режим и ручное изменение выхода, контроль входного сигнала (регулируемого параметра) и сигнала рассогласования, а также контроль сигнала задания и выходного сигнала.

Алгоритмы РЕГА имеют параметры настройки, представленные в табл.107. После ввода РЕГА эта конфигурация может видоизменяться или дополняться другими алгоритмами, в ней могут изменяться параметры настройки в соответствии со стандартными правилами программирования.

Таблица 106

Состав стандартной конфигурации "Регулятор аналоговый РЕГА"

№ алгоблока	Наименование алгоритма	Код	Модификатор	Масшт. врем.
01	Оперативный контроль контуров регулирования ОКО	01	00	-
02	-	00	-	-
03	-	00	-	-
04	-	00	-	-
05	Ввод аналоговый, группа А,В,АА	07	08	-
06	Задание ЗДН	24	00	00
07	Регулирование аналоговое РАН	20	-	00
08	Ручное управление РУЧ	26	-	-
09	Аналоговый вывод, группа А,АВ,А	11	02	-

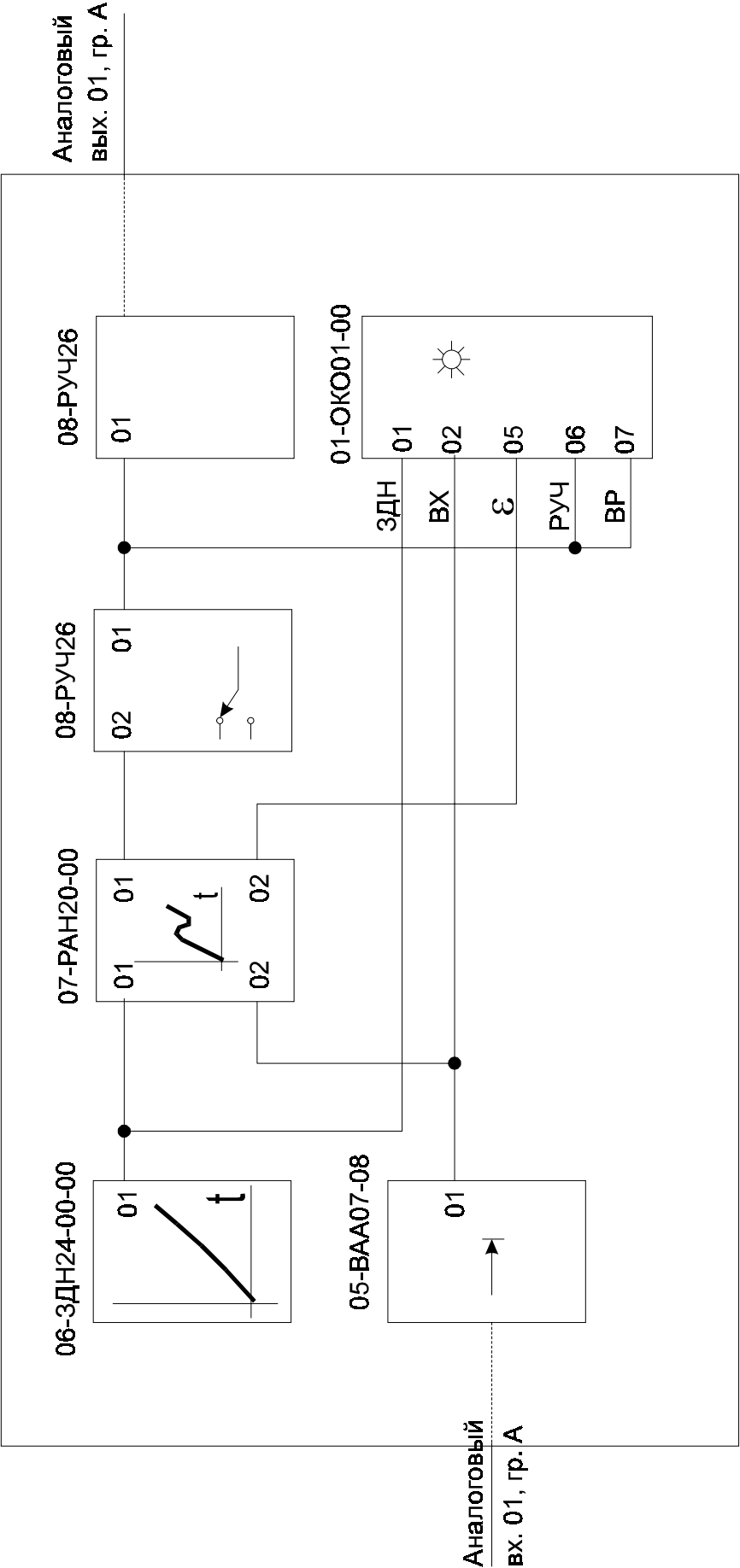
Таблица 107

Параметры настройки

Алгоритм	№ входа	Наименование	Значение
РАН	03	Масштабный коэффициент Км	1,0
	06	Коэффициент пропорциональности Кг	1,0
	07	Постоянная времени интегрирование Т _и , сек	15
	09	Ограничение по максимуму Х _{мксД}	100,0
ОКО	04	Технические единицы, соответствующие 100%	100

Остальные параметры настройки равны нулю.

Стандартная конфигурация 01 "Регулятор аналоговый РЕГА"



Первая цифра -№ алгоритма, затем шифр алгоритма, его код, модификатор и масштаб времени в алгоблоках 02 - 04 - "пустой" алгоритм (код 00)
Рис. 91

11.2 РЕГИ(02) - Регулятор импульсный (стандартная конфигурация)

Назначение

Стандартная конфигурация "Регулятор импульсный РЕГИ" (в дальнейшем РЕГИ) предназначена для построения контура регулирования с импульсным выходным сигналом.

РЕГИ включена в состав библиотеки только регулирующей модели контроллера.

Функциональное описание

РЕГИ состоит из 6 алгоритмов (см. табл.108). Алгоритмы связаны между собой конфигурацией, показанной на рис. 92.

Регулятор помимо алгоритма регулирования содержит ручной задатчик, алгоритм ручного управления, алгоритм оперативного контроля. Входной сигнал и сигнал, характеризующий положение исполнительного механизма, подаются на аналоговые входы контроллера (соответственно вход 01 и 02 группа А). Выходной сигнал формируется на импульсном выходе контроллера (выход 01, группа А).

РЕГИ помещается только в первый контур. Остальные три контура (алгоблоки 02-04) оставлены свободными, поэтому при необходимости в них могут помещаться нужные алгоритмы.

РЕГИ помимо функций регулирования обеспечивает также функции оперативного управления в первом контуре, а именно ручное изменение сигнала задания, переход на ручной режим и ручное изменение выхода, контроль входного сигнала (регулируемого параметра), и сигнала рассогласования, а также контроль сигнала задания и выходного сигнала.

Алгоритм РЕГИ имеет параметры настройки, представленные в табл. 109. После введения РЕГИ в алгоблоке 09 на входе 03 параметр настройки "номер контура N_k " должен быть установлен на значение $N_k = I$.

После ввода РЕГИ конфигурация может видоизменяться и дополняться другими алгоритмами, в ней может изменяться конфигурация и параметры настройки в соответствии со стандартными правилами программирования.

Таблица 108

Состав стандартной конфигурации "Регулятор импульсный РЕГИ"

№ алгоблока.	Наименование алгоритма	Код	Модификатор	Масшт. времени
01	Оперативный контроль контуров регулирования ОКО	01	05	-
02	-	00	-	-
03	-	00	-	-
04	-	00	-	-
05	Ввод аналоговый, группа А,В,АА	07	08	-
06	Задание ЗДН	24	00	00
07	Регулирование импульсное РИМ	21	-	00
08	Ручное управление РУЧ	26	-	-
09	Импульсный вывод, группа А,ИВА	15	02	-

Таблица 109

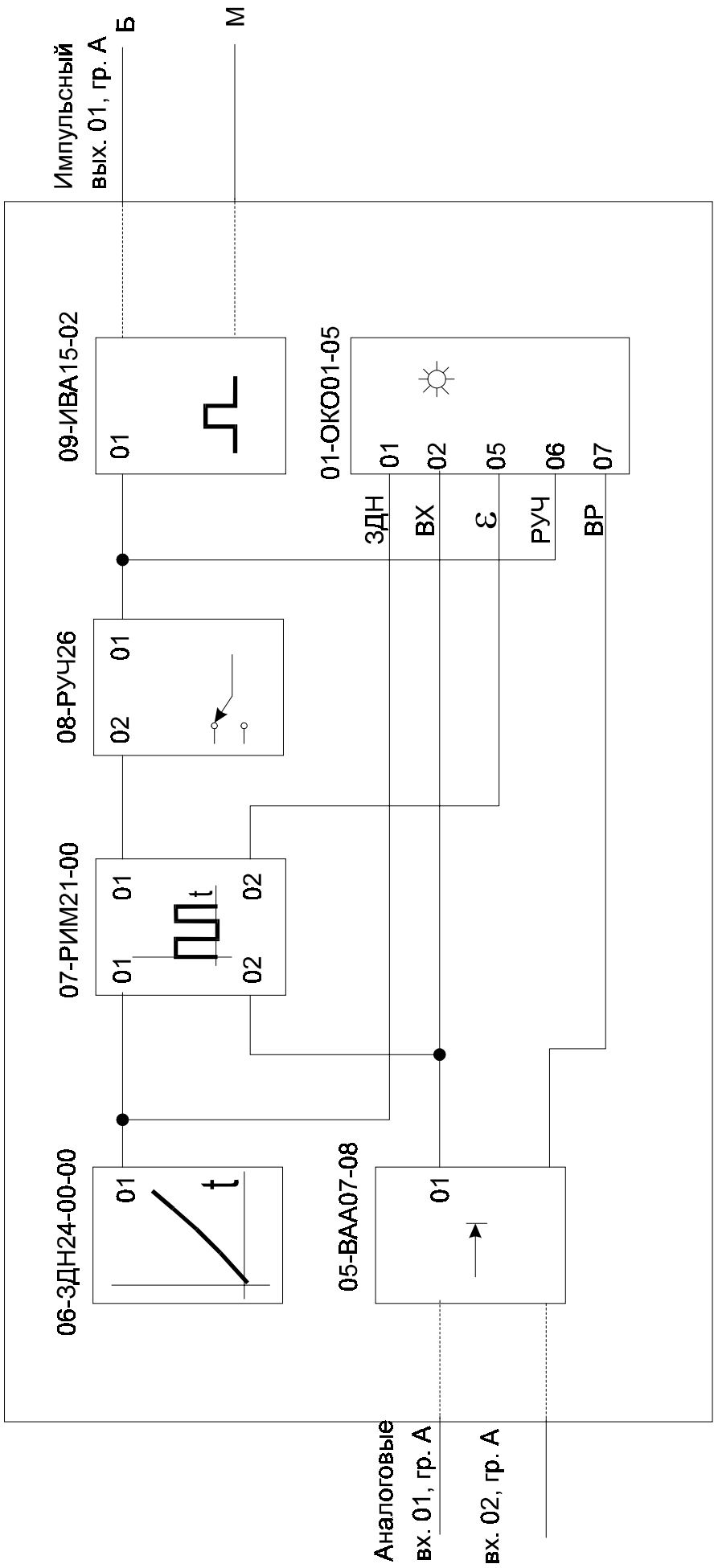
Параметры настройки

Алгоритм	№ входа	Наименование	Значение
РИМ	03	Масштабный коэффициент K_m	1,0
	06	Коэффициент пропорциональности K_p	1,0
	07	Постоянная времени интегрирования $T_{и}$, сек	15
	09	Время исполнительного механизма $T_{из}$, сек	60
ОКО	04	Технические единицы, соответствующие 100%	100
ИВА	02	Минимальная длительность импульса $T_{имп}$, сек	0,12
	03	Номер контура *	1

*-устанавливается оператором

Остальные параметры настройки равны нулю.

Стандартная конфигурация 02 "Регулятор импульсный РЕГИ"



Первая цифра -№ алгоблока, затем шифр алгоритма, его код, модификатор и масштаб времени в алгоблоках 02 - 04 - "пустой" алгоритм (код 00)
Рис. 92