

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»  
**Снежинский физико-технический институт-**  
филиал федерального государственного автономного образовательного учреждения высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»  
**(СФТИ НИЯУ МИФИ)**

Кафедра: АИиВС  
(наименование кафедры)

**ОТЧЕТ ПО КУРСОВОМУ ПРОЕКТУ**

по курсу Теория автоматов  
(наименование учебной дисциплины )

на тему «Синтез цифровых управляющих автоматов»

*Группа* : ПАС-30Д  
*Студент* \_\_\_\_\_ Домрачева Ю.К.

*Студент* \_\_\_\_\_ Горшкова А.К.

*Преподаватель* \_\_\_\_\_ Крушной В.В.

Снежинск  
2022 г.

## Содержание

Введение.....	3
1 Канонический метод структурного синтеза.....	4
2 Синтез цифрового автомата по граф-схеме алгоритма.....	8
2.2 Отмеченная ГСА.....	10
2.3 Структурный синтез автомата.....	12
3 Синтез автомата с программируемой логикой.....	14
3.1 Адресация микрокоманд с двумя принудительными адресами.....	14
3.2 Адресация микрокоманд с одним принудительным адресом (инкремент текущего адреса).....	16
3.3 Адресация микрокоманд с одним принудительным адресом (инкремент адреса перехода).....	18
3.4 Естественная адресация.....	20
Заключение.....	23
Список литературы.....	24

## Введение

Данная курсовая работа состоит из трёх разделов. В первом разделе представлен синтез цифрового автомата по каноническому методу структурного синтеза. Результатом этого метода является система логических уравнений и комбинационная логическая схема, реализующих данную систему.

Структурный синтез цифрового автомата выполнен на основании заданных таблицы переходов и таблицы выходов. Для того чтобы реализовать автомат по данному методу выполнено следующее:

- 1) структуризация входов, выходов и состояний автомата;
- 2) кодировка входных, выходных сигналов и внутренних состояний автомата;
- 3) получение кодированных таблиц переходов и выходов структурного автомата;
- 4) получение логических функций по кодированным таблицам;
- 5) построение функционально-логической схемы автомата на основании полученных в результате синтеза логических функций.

Во втором разделе представлен цифровой автомат по ГСА, выполнен структурный синтез автомата.

В третьем разделе представлен синтез автомата с жёсткой и программируемой логикой по трем видам адресации.

## 1 Канонический метод структурного синтеза

Выполнен структурный синтез частичного автомата А, заданного своими таблицами переходов и выходов, представленных в таблице 1.1 и таблице 1.2.

Таблица 1.1 – Таблица переходов

Таблица 1.2 – Таблица выходов

$\delta$	$a_1$	$a_2$	$a_3$	$a_4$
$z_1$	$a_3$	-	$a_2$	$a_1$
$z_2$	-	$a_2$	$a_4$	-
$z_3$	$a_2$	$a_3$	-	$a_3$

$\lambda$	$a_1$	$a_2$	$a_3$	$a_4$
$z_1$	$w_1$	-	$w_2$	$w_3$
$z_2$	-	$w_2$	$w_3$	-
$z_3$	$w_2$	$w_3$	-	$w_3$

В качестве элементов памяти используется Т – триггер.

Произведена кодировка входных, выходных сигналов и внутренних состояний автомата. Количество входных абстрактных сигналов  $F = 3$ , количество входных структурных сигналов  $n = \lceil \log_2 F \rceil = \lceil \log_2 3 \rceil = 2$ , то есть  $X = \{x_1, x_2\}$ . Количество выходных абстрактных сигналов  $G = 3$ , количество выходных структурных сигналов  $m = \lceil \log_2 G \rceil = \lceil \log_2 3 \rceil = 2$ , то есть  $Y = \{y_1, y_2\}$ . Количество внутренних состояний абстрактного автомата  $M = 4$ , следовательно, количество двоичных элементов памяти (триггеров)  $K = \lceil \log_2 M \rceil = \lceil \log_2 4 \rceil = 2$ .

Кодирование входных, выходных сигналов и внутренних состояний представлено в таблицах 1.3 -1.5.

Таблица 1.3 –  
Кодирование входных  
сигналов

$Z$	$x_1$	$x_2$
$z_1$	0	0
$z_2$	0	1
$z_3$	1	0

Таблица 1.4 –  
Кодирование  
выходных сигналов

$W$	$y_1$	$y_2$
$w_1$	0	0
$w_2$	0	1
$w_3$	1	0

Таблица 1.5 –  
Кодирование  
выходных сигналов

$A$	$Q_1$	$Q_2$
$a_1$	0	0
$a_2$	0	1
$a_3$	1	0
$a_4$	1	1

Кодированные таблицы переходов и выходов структурного автомата представлены на таблицах 1.3 – 1.5. В таблицах переходов и выходов исходного абстрактного автомата  $z_i, w_i, a_i$  заменены соответствующие коды. Кодированные таблицы переходов и выходов представлены в таблицах 1.6 и 1.7.

		$a_1$	$a_2$	$a_3$	$a_4$
$Q_1 Q_2$	$x_1 x_2$	00	01	10	11
$z_1$	00	10	-	01	00
$z_2$	01	-	01	11	-
$z_3$	10	01	10	-	10

		$a_1$	$a_2$	$a_3$	$a_4$
$Q_1 Q_2$	$x_1 x_2$	00	01	10	11
$z_1$	00	00	-	01	10
$z_2$	01	-	01	10	-
$z_3$	10	01	10	-	00

В кодированной таблице переходов заданы функции:

$$Q_1^{t+1} = \square_1(x_1, x_2, Q_1^t, Q_2^t),$$

$$Q_2^{t+1} = \square_2(x_1, x_2, Q_1^t, Q_2^t).$$

В кодированной таблице выходов заданы функции:

$$y_1 = \phi_1(x_1, x_2, Q_1^t, Q_2^t), \quad y_2 = \phi_2(x_1, x_2, Q_1^t, Q_2^t).$$

При каноническом методе синтез сведен к получению функций:

$$y_1 = \phi_1(x_1, x_2, Q_1, Q_2);$$

$$y_2 = \phi_2(x_1, x_2, Q_1, Q_2);$$

$$T_1 = f_1(x_1, x_2, Q_1, Q_2);$$

$$T_2 = f_2(x_1, x_2, Q_1, Q_2).$$

Функции  $y_1$  и  $y_2$  получены из кодированной таблицы выходов.

Карты Карно для минимизации выражения для  $y_1$  и  $y_2$  изображены на рисунке 1.1.

$Q_1 Q_2$	00	01	11	10
00		-	1	
01	-		-	1
11	-	-	-	-
10		1		-

$Q_1 Q_2$	00	01	11	10
00		-		1
01	-	1	-	
11	-	-	-	-
10	1			-

*a* *б*  
Рисунок 1.1 – Карты Карно для  $y_1$  (*a*) и  $y_2$  (*б*)

Результат минимизации представлен в выражении (1):

$$\begin{cases} y_1 = x_2 \bar{Q}_2 \vee \bar{x}_1 Q_1 Q_2 \vee x_1 \bar{Q}_1 \bar{Q}_2; \\ y_2 = x_2 Q_2 \vee \bar{x}_2 Q_1 \bar{Q}_2 \vee x_1 \bar{Q}_2. \end{cases} \quad (1)$$

С помощью таблицы функций возбуждения (Таблица 1.8) выведены выражения для T1 и T2. Используя код исходного состояния автомата и код состояния перехода на основании таблицы входов триггера, найдено требуемое выражение функций возбуждения, представленное на рисунке 1.2, обеспечивающее заданный переход.

Таблица 1.8 – Таблица функций возбуждения

$Q_1 Q_2$ $x_1 x_2$	00	01	10	11
00	10	-	11	11
01	-	00	01	-
10	01	11	-	01

T1T2

$Q_1 Q_2$ $x_1 x_2$	00	01	11	10
00	1	-	1	1
01	-			
11	-	-	-	-
10		1		-

а

$Q_1 Q_2$ $x_1 x_2$	00	01	11	10
00		-	1	1
01				1
11	-	-	-	-
10	1	1	1	-

Рисунок 1.2 – Карты Карно для T<sub>1</sub> (а) и T<sub>2</sub> (б)

$$T_1 = \bar{x}_1 \bar{x}_2 \vee x_1 Q_1 \bar{Q}_2 \quad (2)$$

(2)

На основании полученных в результате синтеза булевых выражений (1), (2) построена функционально-логическая схема автомата, изображённая на рисунке 1.3.

Для этого уравнения 1 и 2 представлены в виде:

$$\begin{cases} y_1 = x_2 \bar{Q}_2 \vee \bar{x}_1 Q_1 Q_2 \vee x_1 \bar{Q}_1 Q_2; \\ y_2 = x_2 Q_2 \vee \bar{x}_2 Q_1 \bar{Q}_2 \vee x_1 \bar{Q}_1 \bar{Q}_2; \\ T_1 = \bar{x}_1 \bar{x}_2 \vee x_1 \bar{Q}_1 \bar{Q}_2; \\ T_2 = Q_1 \bar{Q}_2 \vee \bar{x}_2 Q_2 \vee x_1 \bar{x}_2. \end{cases}$$

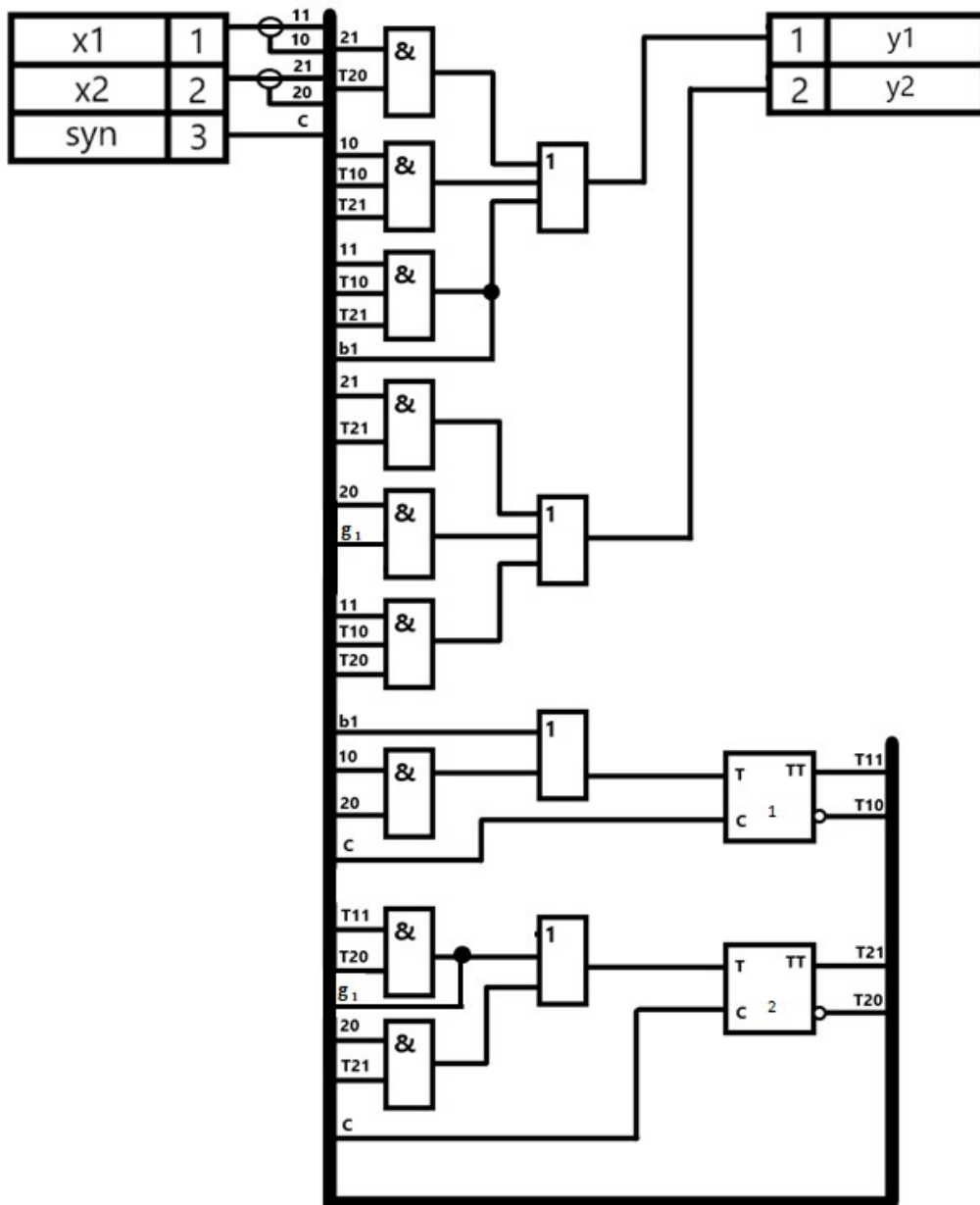


Рисунок 1.3 – Функционально-логическая схема автомата



## 2 Синтез цифрового автомата по граф-схеме алгоритма

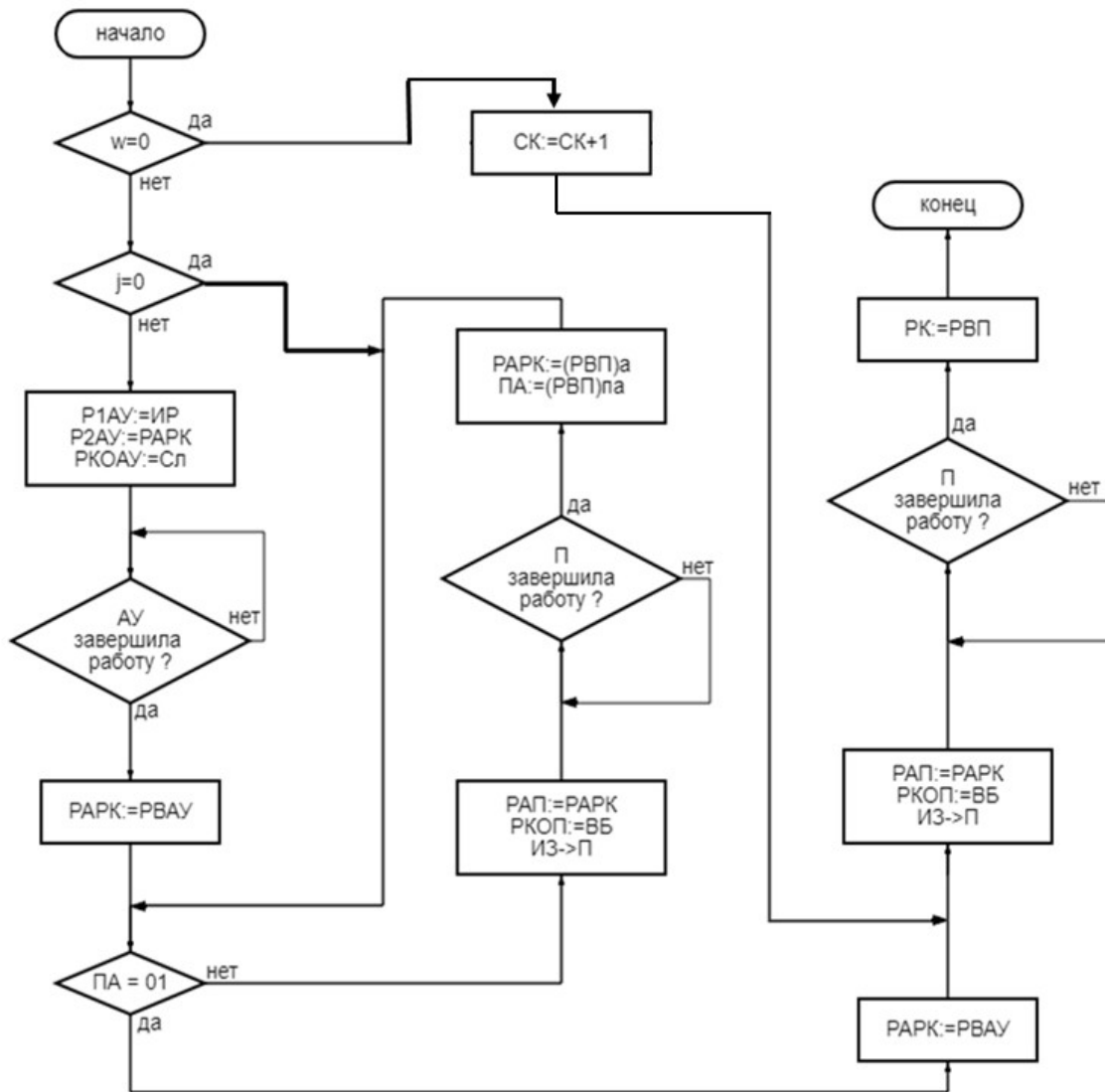


Рисунок 2.1 – Алгоритм функционирования

Синтез микропрограммного автомата по граф – схеме алгоритма, приведенный на рисунке 2 .1, осуществлён в два этапа:

- 1) получение отмеченной ГСА;
- 2) построение графа автомата в виде списковых структур (таблицы переходов).

## 2.1 Кодированная ГСА

Таблица 2.1 – Кодирование логических выражений

	Выражение	Код выражения
$w=0$	$x_1$	1
$j=0$	$x_2$	2
АУ завершило работу?	$x_3$	3
ПА = 01	$x_4$	4
П завершила работу?	$x_5$	5

Таблица 2.2 – Кодирование управляющих сигналов

	Сигнал	Код сигнала
Р1АУ:=ИР Р2АУ:=РАРК РКОАУ:=Сл	$y_1$	01
РАРК:=РВАУ	$y_2$	02
РАП:=РАРК	$y_3$	03
РКОП:=Вб ИЗ->П	$y_4$	04
РАРК:=(РВП) <sub>a</sub> ПА:=(РВП) <sub>па</sub>	$y_5$	05
СК:=СК+1	$y_6$	06
СК:=РАРК	$y_7$	07
РАП:=СК	$y_8$	10
РК:=РВП	$y_9$	11

На рисунке 2.2 представлена полученная кодированная граф-схема алгоритма.

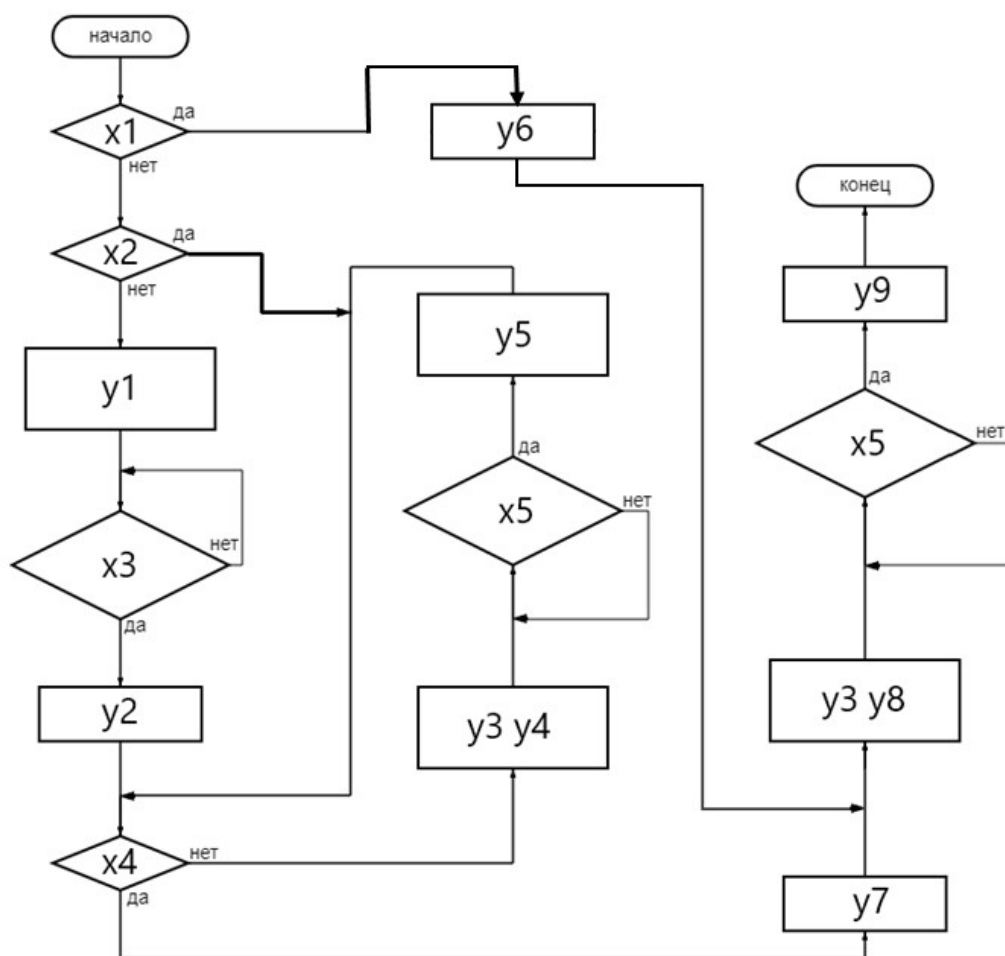


Рисунок 2.2 - Кодированная ГСА

## 2.2 Отмеченная ГСА

На этапе получения отмеченной ГСА входы вершин отмечены символами  $a_1, a_2, \dots, a_n$  по следующим правилам:

- 1) символом  $a_1$  обозначен вход вершины, следующий за начальной, а также вход конечной вершины;
- 2) символами  $a_2, \dots, a_n$  обозначаются входы всех вершин, следующих за операторными;
- 3) если вход вершины обозначен, то только одним символом;
- 4) входы различных вершин, за исключением конечного, обозначаются различными символами.

Таким образом, отмеченная ГСА представлена на рисунке 2. 3.

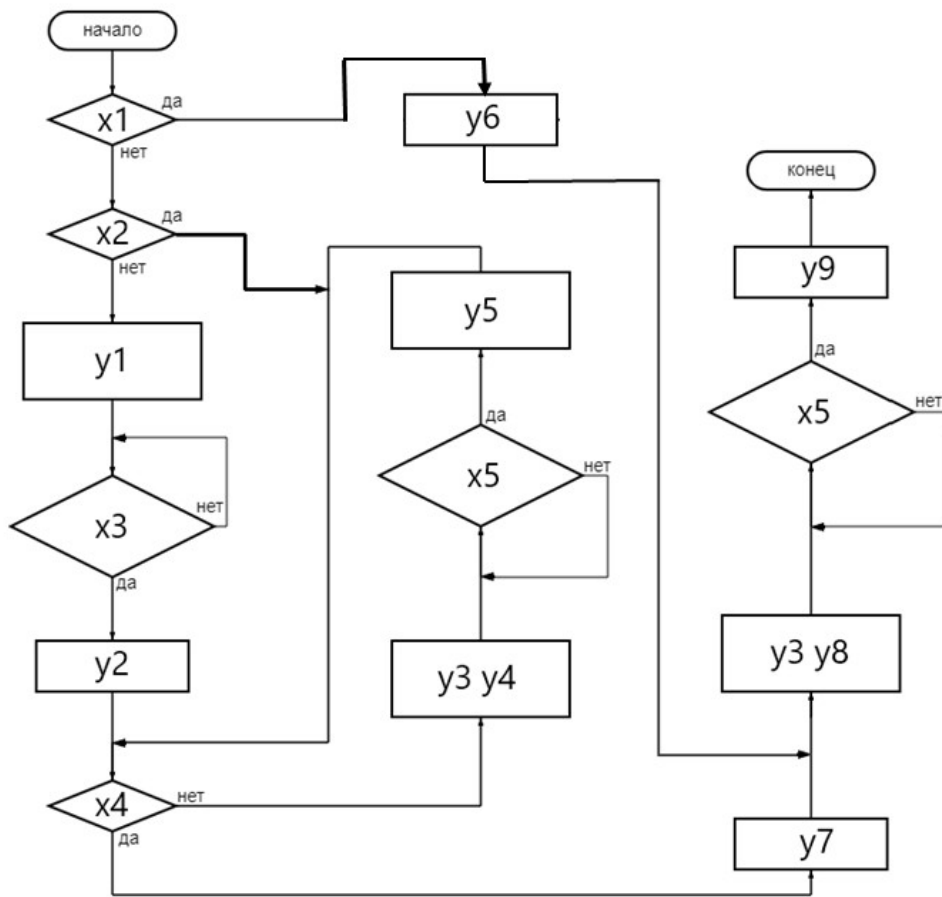


Рисунок 2.3 – Отмеченная ГСА

### 2.3 Структурный синтез автомата

Кодирование состояний автомата приведено в таблице 2.3.

Таблица 2.3 – Кодирование состояний

	00	01	10	11
0	$a_1$	$a_2$	$a_3$	$a_4$
1	$a_5$	$a_6$		

Из таблицы переходов следует, что в автомате количество состояний  $M=6$ , таким образом число элементов памяти составляет:  $m = \lceil \log_2 M \rceil = \lceil \log_2 6 \rceil = 3$ .

Для синтеза использованы D триггеры.

С использованием отмеченной ГСА построена обратная структурная таблица, представленная в таблице 2.4, в которой сначала записываются все переходы в первое состояние, затем во второе и т.д.

В первом столбце указываются все состояния  $a_m$ , из которых осуществляются переходы. Коды этих состояний  $K(a_m)$  после кодирования занесены во второй столбец. В третьем и четвертом столбцах записаны состояния  $a_s$ , в которых произошли переходы, и их коды  $K(a_s)$ . Пятый и шестой столбцы содержат входные  $X(a_m, a_s)$  и выходные сигналы  $Y(a_m, a_s)$ , входящие в пути перехода. В седьмом столбце таблицы перечислены обязательные функции возбуждения  $\Psi(a_m, a_s)$ , вырабатываемые на соответствующих переходах для D-триггера и в восьмом - номер пути перехода для удобства идентификации.

Таблица 2.4 – Обратная структурная таблица

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$\Psi(a_m, a_s)$	$P$
$a_6$	101	$a_1$	000	$x_5$	$y_9$	-	1
$a_1$	000	$a_2$	001	$\bar{x}_1 \bar{x}_2$	$y_1$	$D_1$	2
$a_2$	001	$a_2$	001	$\bar{x}_3$	-	$D_1$	3
$a_2$	001	$a_3$	010	$x_3$	$y_2$	$D_2$	4
$a_4$	011	$a_3$	011	$x_5$	$y_5$	$D_2$	5
$a_1$	011	$a_4$	011	$\bar{x}_1 x_2 \bar{x}_4$	$y_3 y_4$	$D_1 D_2$	6
$a_3$	010	$a_4$	011	$\bar{x}_4$	$y_3 y_4$	$D_1 D_2$	7
$a_4$	011	$a_4$	011	$\bar{x}_5$	-	$D_1 D_2$	8
$a_1$	000	$a_5$	100	$x_1$	$y_6$	$D_3$	9
$a_1$	000	$a_5$	100	$\bar{x}_1 x_2 x_4$	$y_7$	$D_3$	10
$a_3$	010	$a_5$	100	$x_4$	$y_7$	$D_3$	11
$a_5$	100	$a_6$	101	1	$y_3 y_8$	$D_1 D_3$	12
$a_6$	101	$a_6$	101	$\bar{x}_5$	-	$D_1 D_3$	13

## 2.4 Функционально-логическая схема автомата

На основании полученной обратной структурной таблицы получается функционально-логическая схема автомата, изображенная на рисунке 2.4.

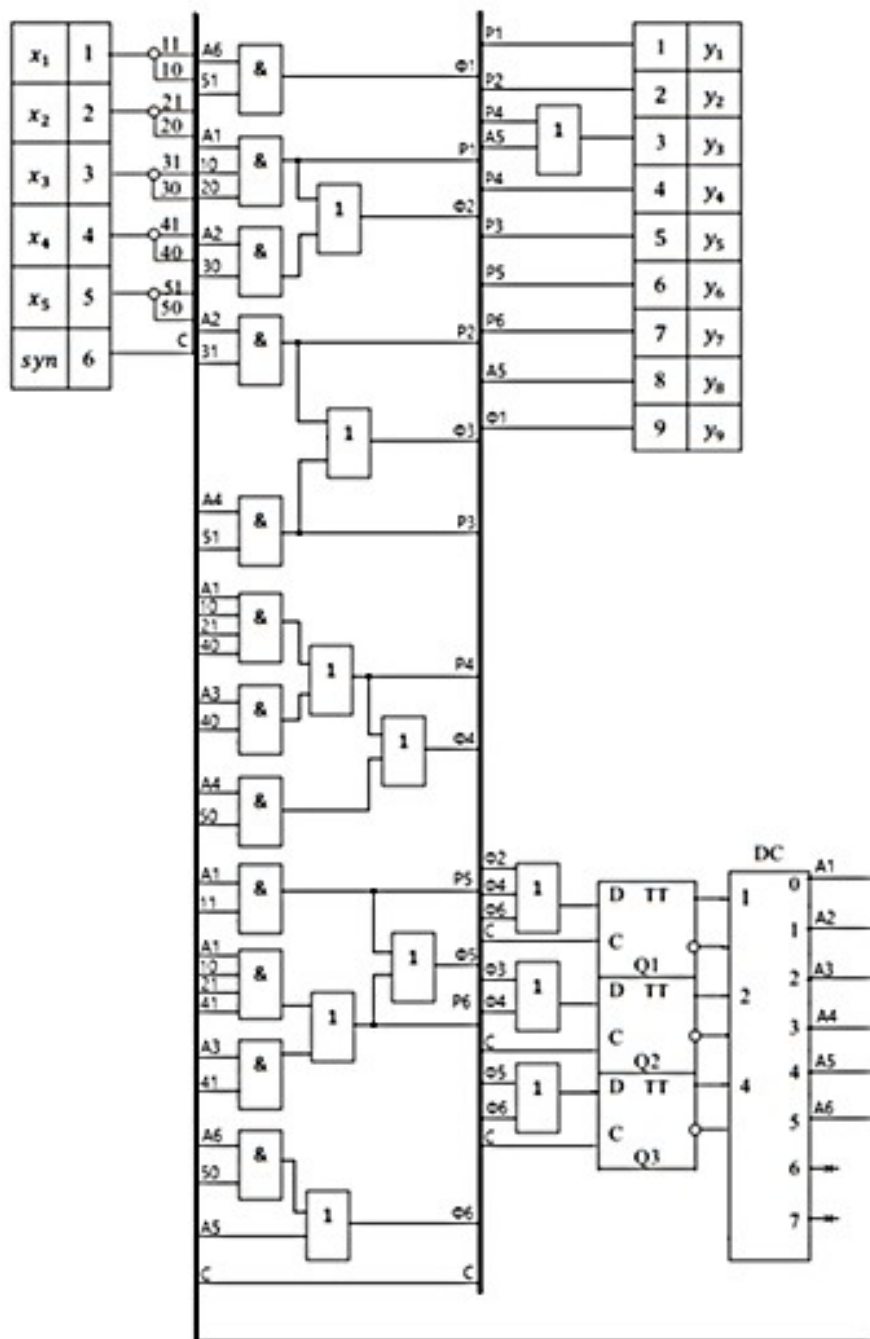


Рисунок 2.4 – Функционально-логическая схема автомата

### 3 Синтез автомата с программируемой логикой

#### 3.1 Адресация микрокоманд с двумя принудительными адресами

Для того, чтобы реализовать цифровой автомат с программируемой логикой с двумя принудительными адресами, составлен набор микрокоманд. Структура каждой микрокоманды включает в себя: микрооперации, выполняемые за один такт, код проверяемого логического условия и два адреса следующих микрокоманд. Адрес следующей микрокоманды определен в зависимости от кода логического условия КЛУ и значения соответствующего логического условия  $x_i$ , либо полем A1, либо полем A2.

Кодировка микропрограммы выполнена в восьмеричной системе счисления. Набор микрокоманд представлен в таблице 3.1.

Длина микрокоманды составила  $N=33$  бит ( $11$  символов  $\times$   $3$  бита). Объем занимаемой ёмкости постоянной памяти при данном способе адресации микрокоманд составил  $E=495$  бит ( $15$  микрокоманд  $\times$   $33$  бит).

Пример формирования микропрограммы с двумя принудительными адресами представлен на рисунке 3.1.

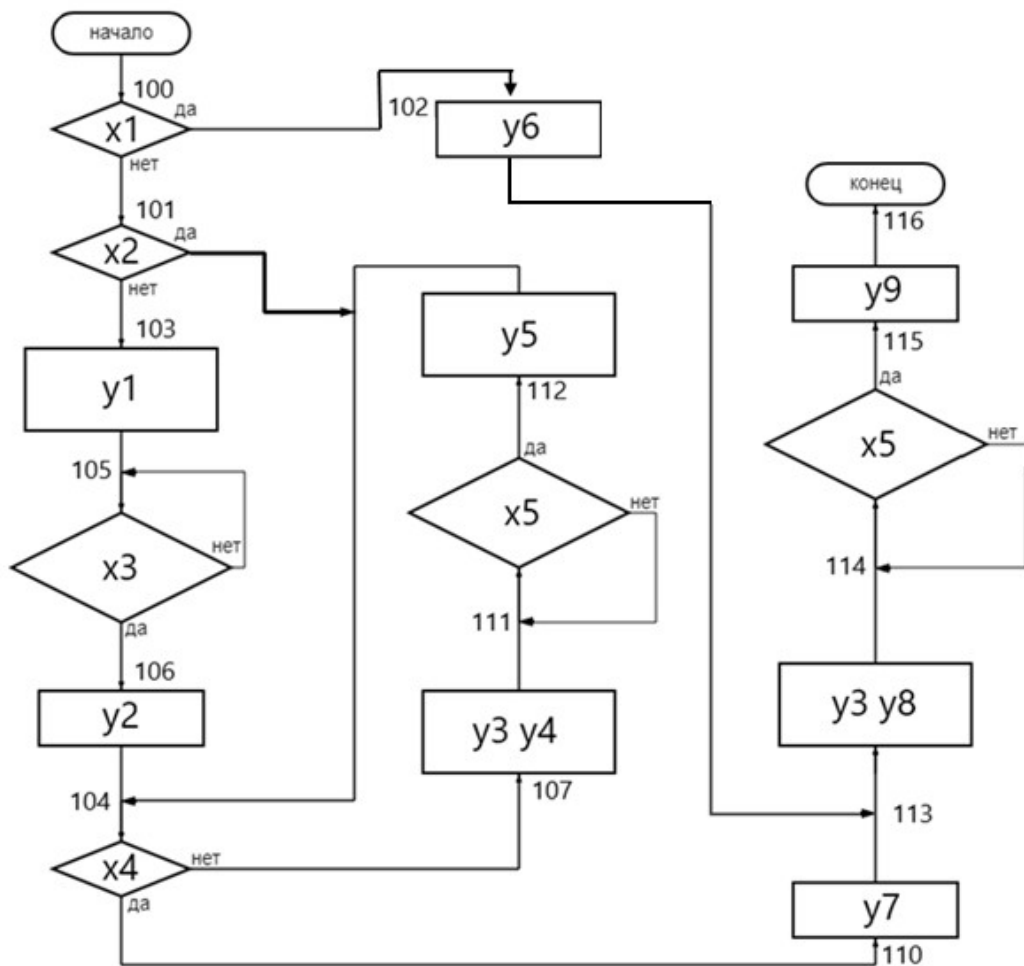


Рисунок 3.1 – Граф структура микропрограммы с двумя принудительными адресами



Таблица 3.1 – Набор микрокоманд

$A_T$	$Y_1$	$Y_2$	КЛУ	$A_1$	$A_2$
100	00	00	1	101	102
101	00	00	2	103	104
102	06	00	0	113	000
103	01	00	3	105	106
104	00	00	4	107	110
105	00	00	3	105	106
106	02	00	4	107	110
107	03	04	5	111	112
110	07	00	0	113	000
111	00	00	5	111	112
112	05	00	4	107	110
113	03	10	5	114	115
114	00	00	5	114	115
115	11	00	0	116	000
116	77	00	0	000	000

stop

3.2 Адресация микрокоманд с одним принудительным адресом (инкремент текущего адреса)

При данном способе адресации для формирования адреса следующей микрокоманды отводится единственное поле адреса перехода  $A_p$ .

Если поле  $КЛУ=0$ , то значение  $A_p$ , безусловно, определяет адрес следующей микрокоманды. Если поле кода логического условия не нулевое, то адрес следующей микрокоманды реализуется условным переходом: если значение проверяемого логического условия  $x_i=0$ , то  $РАПП:=A_p$ , если  $x_i=1$ , то второй адрес перехода определяется инкрементом адреса текущей микрокоманды, хранящегося в регистре адреса постоянной памяти с функцией инкрементного счётчика ( $A_T + 1$ ).

Длина микрокоманды составила  $N=24$  бит. Объём занимаемой ёмкости постоянной памяти при данном способе адресации микрокоманд составил  $E=480$  бит.

Набор микрокоманд представлен в таблице 3.2.

Пример формирования микропрограммы с одним принудительным адресом показан на рисунке 3.2 .

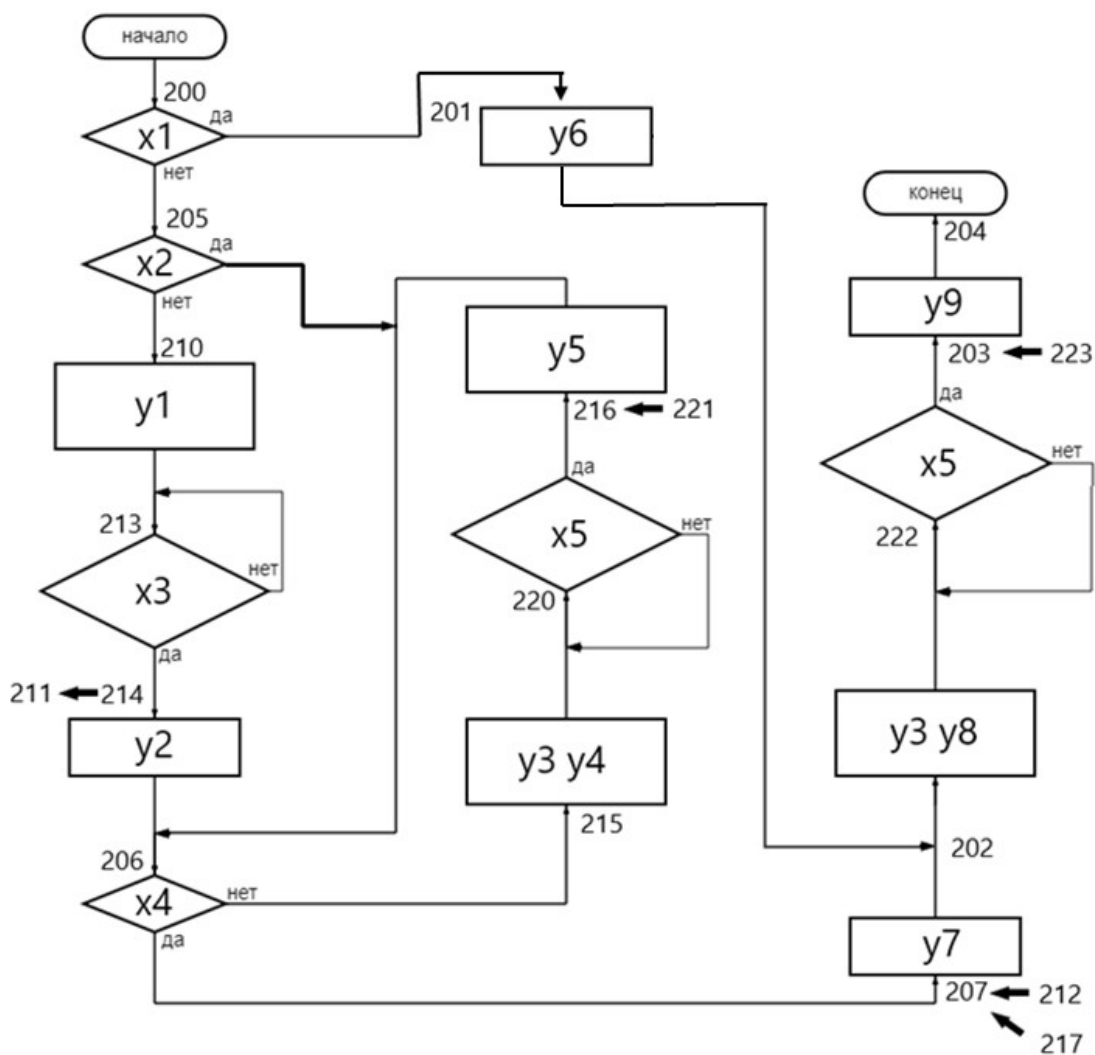


Рисунок 3.2 – Структурный граф микропрограммы с одним принудительным адресом (инкремент текущего адреса)

Таблица 3.2 – Набор микрокоманд

$A_T$	$Y_1$	$Y_2$	$KЛУ$	$A_p$	
200	00	00	1	202	
201	06	00	0	202	
202	03	10	5	222	
203	11	00	0	204	
204	77	00	0	000	stop
205	00	00	2	210	
206	00	00	4	215	
207	07	00	0	202	jump
210	01	00	3	213	
211	02	00	4	215	
212	00	00	0	207	jump
213	00	00	3	213	
214	00	00	0	211	jump
215	03	04	5	220	
216	05	00	4	215	
217	00	00	0	207	
220	00	00	5	220	
221	00	00	0	216	jump
222	00	00	5	222	
223	00	00	0	203	jump

### 3.3 Адресация микрокоманд с одним принудительным адресом (инкремент адреса перехода)

Адресация с инкрементом адреса перехода отличается от адресации с инкрементом текущего адреса лишь способом формирования адреса следующей микрокоманды при условном переходе. Если значение проверяемого логического условия  $x_i = 0$ , то  $РАПП := A_p$ , если  $x_i = 1$ , то второй адрес перехода определяется инкрементом адреса перехода  $A_p+1$ , записанного в регистре микрокоманды.

Длина микрокоманды составила  $N=23$  бит. Объём занимаемой ёмкости постоянной памяти при данном способе адресации микрокоманд составил  $E=391$  бит.

Набор микрокоманд представлен в таблице 3.3.

Пример формирования микропрограммы с одним принудительным адресом показан на рисунке 3.3.

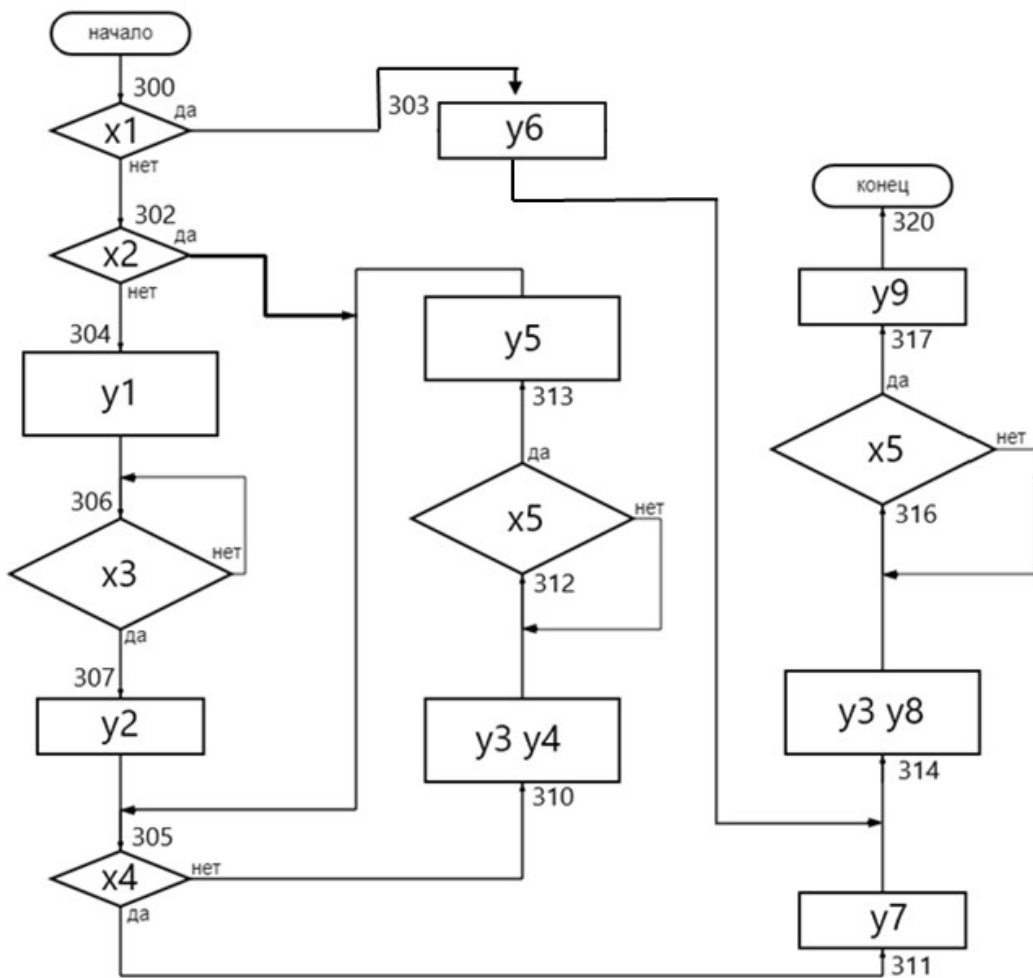


Рисунок 3.3 – Структурный граф микропрограммы с одним принудительным выходом (инкремент адреса перехода)

Таблица 3.3 – Набор микрокоманд

$A_T$	$Y_1$	$Y_2$	$KЛУ$	$A_p$
300	00	00	1	302
301	xx	xx	x	xxx
302	00	00	2	304
303	06	00	0	314
304	01	00	0	306
305	00	00	4	310
306	00	00	3	306
307	02	00	4	310
310	03	04	5	312
311	07	00	0	314
312	00	00	5	312
313	05	00	4	310
314	03	10	5	316
315	xx	xx	x	xxx
316	00	00	5	316
317	11	00	0	320
320	77	00	0	000

stop

### 3.4 Естественная адресация

В данном способе вводится два типа микрокоманд:

- 1) операторная микрокоманда, с признаком  $d=0$ ;
- 2) условная микрокоманда, с признаком  $d=1$ .

Структура операторной микрокоманды: признак микрокоманды, микрооперации, выполняемые за один такт. Структура условной микрокоманды: признак микрокоманды, код проверяемого логического условия, адрес следующей микрокоманды.

При естественной адресации адрес следующей микрокоманды -  $A_T + 1$ . При условном переходе, если проверяемое логическое условие  $x_i = 0$ , то следующей выполняется микрокоманда, адрес которой указан в адресном поле текущей микрокоманды; если  $x_i = 1$ , то следующей выполняется микрокоманда с инкрементом текущего адреса ( $A_T + 1$ ).

Для данного способа адресации длина микрокоманды составила  $N=16$  бит, а объём занимаемой ёмкости постоянной памяти –  $E=288$  бита.

Набор микрокоманд представлен в таблице 3.4.

Пример формирования микропрограммы с одним принудительным адресом показан на рисунке 3.4.

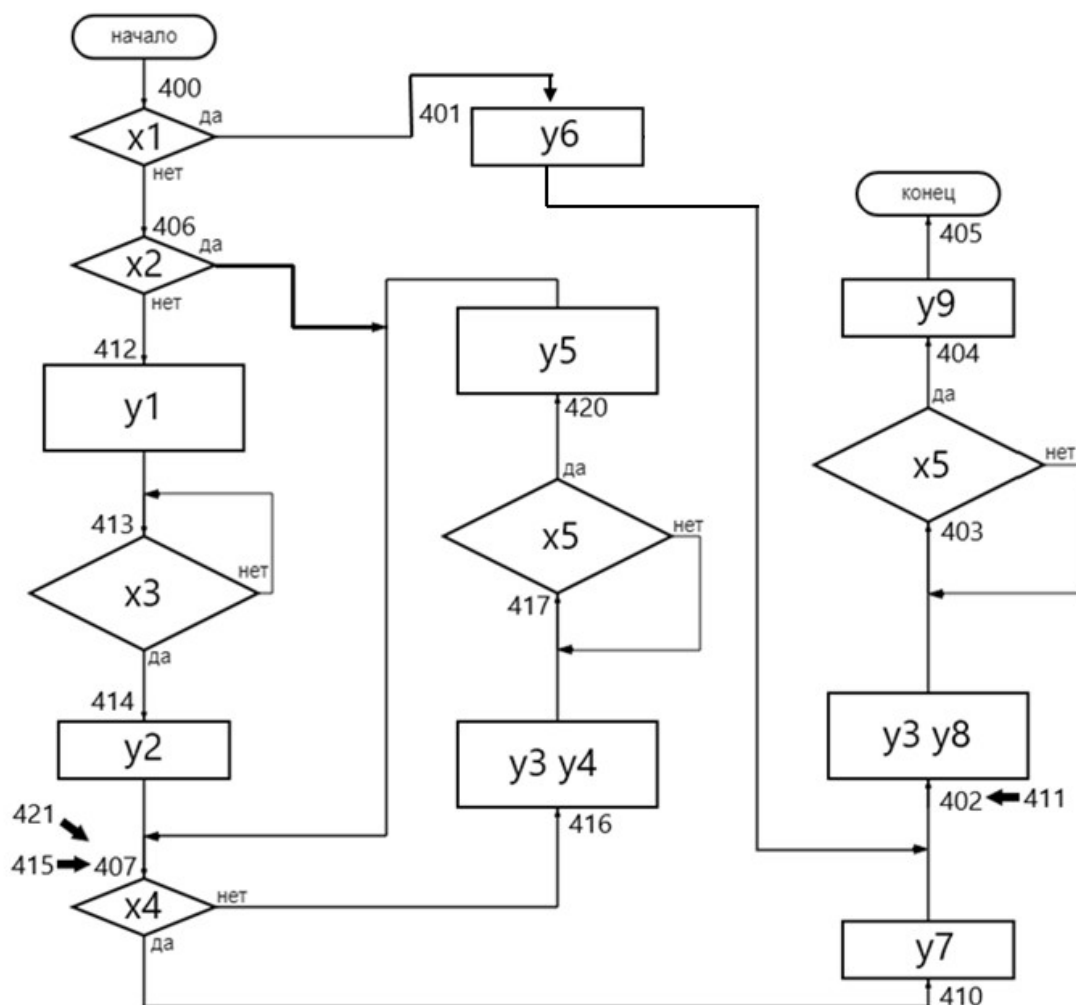


Рисунок 3.4 – Естественная адресация

Таблица 3.4 – Набор микрокоманд

$A_r$	$d=0$	$Y$		
	$d=1$	$KЛУ$	$A_p$	
400	1	1	406	
401	0	0	006	
402	0	3	010	
403	1	5	403	
404	0	0	011	
405	0	7	000	stop
406	1	2	412	
407	1	4	416	
410	0	0	007	
411	1	0	402	jump
412	0	0	001	
413	1	3	413	
414	0	0	002	
415	1	0	407	jump
416	1	0	004	
417	1	5	417	
420	0	0	005	
421	1	0	407	

## Заключение

В данной курсовой работе:

- 1) выполнен синтез цифрового автомата по каноническому методу структурного синтеза с построением функционально - логической схемы автомата;
- 2) Реализован цифровой синтез по ГСА с жесткой логикой. При реализации пользовались составленной обратной структурой таблицей переходов, по которой построена функционально - логическая схема с использованием полной дешифрации состояний;
- 3) Реализован синтез автомата с программируемой логикой тремя способами, наиболее оптимальной оказалась естественная адресация, т.к. объем занимаемой емкости памяти минимален  $E = 288$  бит, но проигрывает по быстродействию



## Список литературы

- 1 Крушный В. В. Синтез цифровых управляющих автоматов: Учебное пособие. – М.: НИЯУ МИФИ, 2011. – 164 с.
- 2 ГОСТ 2.105
- 3 Д.Э. Розенталь Русский язык. Грамматика.