

Лабораторная работа № 6

Синтез микропрограммного автомата

Цель работы: изучение методов абстрактного и структурного синтеза микропрограммных автоматов

1. Понятие о микропрограммном автомате

При описании узлов и устройств цифровой обработки их часто представляют в виде композиции управляющей и операционной частей или *управляющего* и *операционного* автоматов. Операционный автомат (ОА) выполняет конкретные операции преобразования информации (шифраторы, дешифраторы, регистры и т.д.). Функцией управляющего автомата (УА) является координация работы операционных устройств.

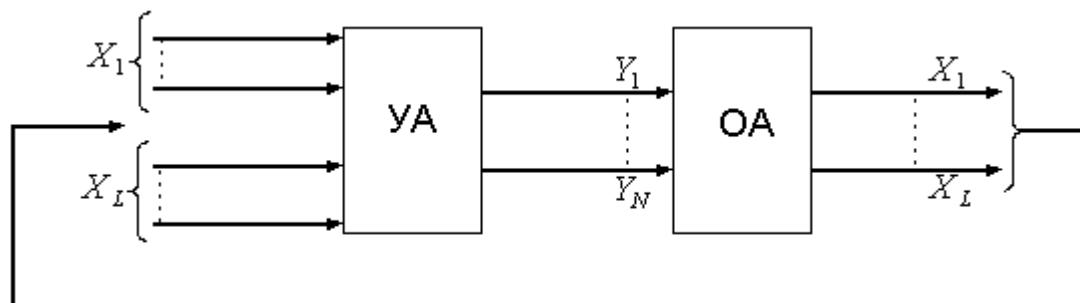


Рис. 6.1. Структура микропрограммного автомата

Задача УА — выработка распределенной во времени последовательности сигналов, определяющих порядок работы операционного автомата (рис. 6.1).

Любая операция, выполняемая ОА, может быть представлена совокупностью микроопераций.

Микрооперацией называется элементарный неделимый акт обработки информации в ОА, происходящий в течение одного момента автоматного времени (т.е. за один такт).

Выполняемые ОА микрооперации обозначаются обычно буквами из множества $Y = \{y_1, \dots, y_N\}$. Эти микрооперации выполняются под воздействием управляющих сигналов из УА, которые обычно обозначаются также, как и микрооперации - y_1, \dots, y_N .

Микрооперации, выполняемые одновременно, называются **микрокомандой** и обозначаются $Y_t = \{y_{t1}, \dots, y_{tnt}\}$, где индекс "t" обозначает микрооперации, выполняемые за один такт автоматного времени.

Порядок выполнения микрокоманд определяется функциями перехода α_{ij} -логическими функциями двоичных переменных из множества $X = \{X_1, \dots, X_L\}$ входных переменных УА. Естественно, что

каждая микрокоманда Y_i может быть связана со многими функциями перехода - множеством $\{\alpha_{i1}, \dots, \alpha_{iT}\}$.

Совокупность микрокоманд и функций перехода образуют **микропрограмму**.

Таким образом, конечный автомат, реализующий микропрограмму работы дискретного устройства, называется **микропрограммным автоматом (МПА)**.

Реализация МПА в виде совокупности ОА и УА была предложена В.М.Глушковым, но она не является единственной. Впервые структура МПА была разработана английским ученым Майклом Уилксом в 1963 году, она называется схемой Уилкса

2. Языки описания МПА

МПА - это специальный автомат, поэтому он задается начальными автоматными языками, к которым относятся: содержательная граф-схема алгоритма (микропрограмма), граф-схема алгоритма (ГСА), логическая схема алгоритма (ЛСА), временные диаграммы. Для описания МПА на абстрактном уровне используются и стандартные языки, из которых наиболее удобными являются графы, таблицы, матрицы переходов.

Содержательная граф-схема алгоритма. Содержательная ГСА - это ориентированный граф, содержащий начальную и конечную вершины, а также вершины, в которых микрооперации и условия записаны в содержательных терминах. Пример содержательной ГСА приведен на рис.6.2..

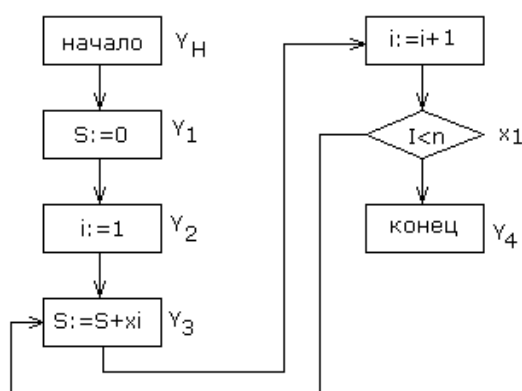


Рис..6. 2. Микропрограмма вычисления суммы

Граф-схема алгоритма (ГСА). ГСА - ориентированный граф, но в его вершинах информация представляется в закодированной форме. Обычно операторные вершины кодируются буквами Y_j , условные - X_j , а микрооперации - Y_j (см .рис.6.2):

$$\begin{aligned}
Y_1: & S:=0; \\
Y_2: & i:=l; \quad x_1:i \leq n; \\
Y_3: & S:=S+x_j; \\
Y_4: & i:=i+1.
\end{aligned}$$

В ГСА одинаковые микрооперации и условия обозначаются одними и теми же символами.

По граф-схеме алгоритма можно строить функции перехода α_{ij} , используя следующие правила.

Если переход из вершины Y_i в вершину Y_j проходит только через условные вершины, соответствующая функция перехода записывается в виде:

$$\alpha_{ij} = \bigwedge_{r=1}^R e_{ijr},$$

где x_{ir} - логическое условие, записанное в r -й вершине;

R - число условных вершин на пути $Y_i - Y_j$;

e_{jr} - принадлежит $\{0;1\}$ и приписывается выходу условной вершины;

$$x_{ir} = \begin{cases} x_{ir}, & \text{если } e_{ir} = 1; \\ \bar{x}_{ir}, & \text{если } e_{ir} = 0. \end{cases}$$

Если на переходе от Y_i к Y_j имеется несколько путей, проходящих через условные вершины, то функция перехода формируется как дизъюнкция конъюнкций, соответствующих всем путям перехода:

$$\alpha_{ij} = \bigvee_{h=1}^H \alpha_{ij}^h,$$

где H - число всевозможных путей от Y_i к Y_j ;

α_{ij}^h - конъюнкция, соответствующая h -му пути от Y_i к Y_j .

Например, для ГСА (рис.6.3) формулы перехода имеют вид:

$$\alpha_{12} = \bar{x}_1 \vee x_1 \bar{x}_1 \vee \bar{x}_1 x_1 \dots x_1 \bar{x}_1 \vee \dots$$

$$\alpha_{13} = x_1 x_2;$$

$$\alpha_{14} = x_1 x_2.$$

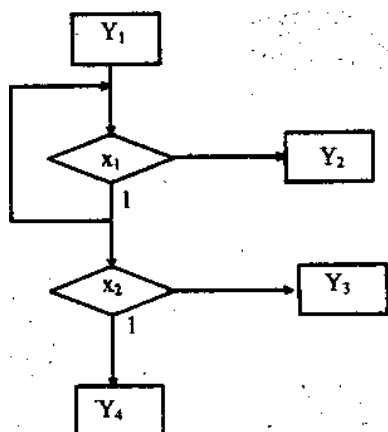


Рис. 6.3. Пример построения функций перехода по ГСА

Логическая схема алгоритма (ЛСА). ЛСА - конечная строка, содержащая символы операторов Y_i и условий X_j и верхние и нижние стрелки, пронумерованные числами натурального ряда.

Например: $Y_H Y_I Y_2 \downarrow^1 Y_3 Y_4 x_I \uparrow^2 w \uparrow^1 \downarrow^2 Y_K$

Правила построения ЛСА:

- 1) ЛСА содержит один оператор начала Y_H и один оператор конца Y_K .
- 2) Перед оператором Y_H и после оператора Y_K стрелки не ставятся.
- 3) После логического условия x_i всегда ставится верхняя стрелка \uparrow .
- 4) В ЛСА не может быть одинаковых нижних стрелок.
- 5) Каждой верхней стрелке соответствует одна нижняя.
- 6) Каждой нижней стрелке может соответствовать несколько верхних стрелок.

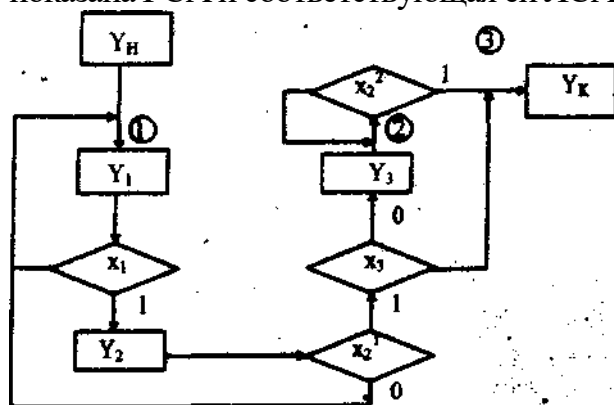
Переход от ГСА к ЛСА выполняется следующим образом

- 1) В ГСА отмечаются входы всех вершин, к которым подходит более одной стрелки, натуральными числами от 1 до S .
- 2) Записывается оператор начала Y_H ; если после начальной вершины в ГСА следует отмеченная S вершина, то после Y_H ставится нижняя стрелка с номером S . Далее записывается следующий оператор.
- 3) После условия x ставится верхняя стрелка \uparrow с номером S в том случае, если выход условной вершины по нулю отмечен, в противном случае номер не ставится. Если единичный выход условной вершины отмечен, то после верхней ставится нижняя стрелка с соответствующим номером, после чего записывается символ оператора, следующего за условной вершиной по выходу "1".
- 4) Процедура записи продолжается, пока в строке не появится нижняя стрелка, записанная ранее, или символ " Y_K " до окончания ЛСА. В этих случаях вместо нижней стрелки и Y_K ставится тождественно ложное условие $w \uparrow$.
- 5) В записанной строке находятся верхние стрелки без номеров, они отмечаются натуральными числами, начиная от $S+1$.

6) Записываются новые строки ЛСА, каждая из которых начинается с нижней стрелки с номером $S+1, S+2, \dots$, вслед за которыми записываются символы операторов и условий, соответствующие нулевым выходам условных вершин.

7) Все строки объединяются в одну путем их последовательной записи, причем последним символом в ЛСА будет оператор конца с нижней стрелкой $\downarrow Y_K$.

8) Проверяется соответствием верхних и нижних стрелок. На рис.6.4 показана ГСА и соответствующая ей ЛСА.



$$Y_H \downarrow Y_1 x_2 \uparrow Y_2 x_2^1 \uparrow x_3 \uparrow w \uparrow Y_3 \downarrow x_2^2 \uparrow \downarrow Y_K$$

Рис.6.4. Пример построения ЛСА по граф-схеме алгоритма

Формулы перехода. Формулы перехода указывают всевозможные пути перехода от оператора Y_i к другим операторам и имеют вид :

$$Y_i \rightarrow \bigvee_{t=1}^T \alpha_{it} Y_t$$

где α_{it} - функция перехода от Y_i к Y_t . Для построения формул перехода удобны ГСА.

Матричная схема алгоритма (МСА). МСА - это квадратная матрица, строки и столбцы которой отмечаются символами операторных вершин, а в поле матрицы записываются функции перехода α_{it} .

Например, для ГСА на рис. 6.4 МСА имеет вид:

$$\begin{array}{c}
 Y_H \quad Y_2 \quad Y_3 \quad Y_4 \quad \leftarrow \text{вершины перехода} \\
 \begin{array}{l}
 Y_H \\
 Y_1 \\
 Y_2 \\
 Y_3
 \end{array}
 \begin{bmatrix}
 1 & . & . & . \\
 \bar{x}_1 & x_1 & . & . \\
 \bar{x}_2^1 & . & x_2^1 \bar{x}_1 & x_2^1 x_3 \\
 . & . & x_2^2 & \bar{x}_2^2
 \end{bmatrix}
 \end{array}$$

↑ исходные вершины

3. Синтез микропрограммного автомата по граф-схеме алгоритма (ГСА)

3.1. Синтез абстрактного автомата

Синтез абстрактного автомата, или *абстрактный синтез*, заключается в построении абстрактного автомата на основе имеющегося физического описания работы автомата и состоит из двух этапов:

- 1) описание автомата на одном из стандартных автоматных языков и определение числа внутренних состояний;
- 1) минимизация внутренних состояний автомата.

Стандартными автоматными языками являются таблицы переходов и выходов, графы и матрицы переходов.

Микропрограммные автоматы, как правило, задаются на начальном языке в виде содержательной граф-схемы алгоритма или микропрограммы. Для построения граф-схемы (ГСА) все микрооперации кодируются символами y_1, y_2, \dots , а условия – символами x_1, x_2, \dots ; причем одинаковые микрооперации и условия кодируются одинаково.

По закодированной ГСА можно построить граф автомата. Для определения числа внутренних состояний автомата проводится отметка ГСА следующим образом:

для автомата Мили

- 1) символом a_1 отмечается вход первой вершины, следующей за начальной, и вход конечной вершины;
- 1) входы вершин, следующих за операторными, отмечаются разными символами a_2, a_3, \dots ;

для автомата Мура

- 1) символом a_1 отмечаются начальная и конечная вершины ГСА;
- 1) каждая операторная вершина отмечается единственным символом a_2, a_3, \dots .

Далее составляются слова вида:

- 1) $a_m X(a_m, a_s), Y(a_m, a_s) a_s$;
- 2) $a_m X(a_m, a_s) a_s$;
- 3) $a_m Y(a_m, a_s) a_s$;
- 4) $a_m a_s$.

Слова первого, второго и третьего видов сопоставляются переходам от метки a_m к метке a_s через условные и операторные вершины, условные вершины и операторные вершины соответственно. Слова четвертого вида соответствуют переходу

между операторными вершинами. Очевидно, что слова 1), 2) и 3) определяются для автомата Мили, а слова 2) и 4) – для автомата Мура.

На рис.6.5 приведен пример отметки ГСА для автомата Мили и автомата Мура. Состояния автомата Мура обозначены символами b_1, b_2, \dots , а для автомата Мили сохранены обозначения a_1, a_2, \dots . Синтезируемый абстрактный автомат Мили имеет 5 состояний, а автомат Мура – 7.

Переход автомата из состояния a_m в состояние a_s ($a_m \rightarrow a_s$) определяется функцией перехода $X(a_m, a_s)$, представляющей конъюнкцию всех условий, записанных в условных вершинах на пути $a_m \rightarrow a_s$. Содержимое операторных вершин $Y(a_m, a_s)$ определяет выходной сигнал на переходе $a_m \rightarrow a_s$.

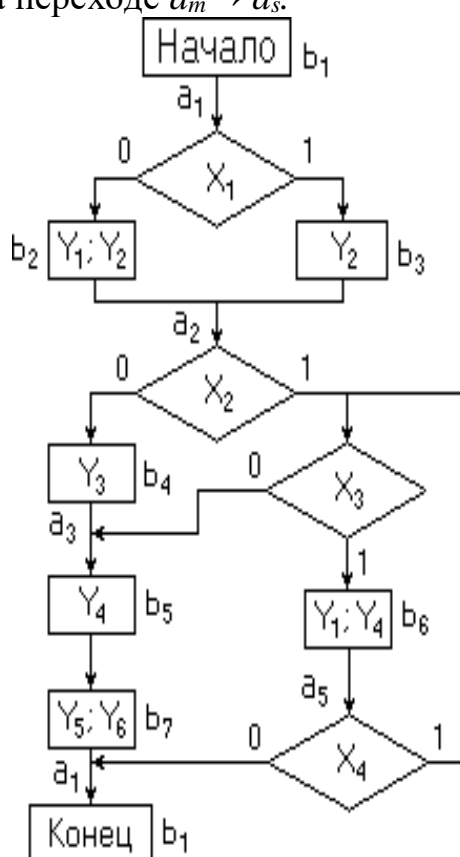


Рис. 6.5. Пример отметки ГСА для автомата Мили (а) и автомата Мура (б)

При построении графа автомата Мили началу дуги приписывается входной сигнал $X(a_m, a_s)$, а концу – выходной сигнал $Y(a_m, a_s)$. Если выполняется переход вида 2), на месте выходного сигнала ставится “0” или прочерк, при переходе вида 3) входной сигнал $X(a_m, a_s)$ полагается равным единице. При построении автомата Мура содержимое операторных вершин ГСА

приписывается вершинам графа, а дуги помечаются символами входных сигналов. Если выполняется переход вида 4), то $X(a_m, a_s) = 1$. На рис.6.6 показан граф автомата Мили (рис.6.5)

Рис.6.6

При синтезе микропрограммного автомата удобнее использовать таблицу переходов в измененном виде. *Прямая таблица переходов* - таблица, в которой последовательно перечисляются все переходы сначала из первого состояния, затем из второго и т.д. В ряде случаев оказывается удобным пользоваться *обратной таблицей переходов*, в которой столбцы обозначены точно так же, но сначала записываются все переходы в первое состояние, затем во второе и т.д. В качестве примера приведена прямая таблица переходов микропрограммного автомата Мили на рис.6.6 – табл.6.1.

Таблица.6.1

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$
a_1	a_2	x_1	y_2
	a_2	$\neg x_1$	y_1, y_2
a_2	a_3	$\neg x_2$	y_3
	a_3	$\neg x_3 x_2$	-
	a_5	$x_3 x_2$	y_1, y_4
a_3	a_4	1	y_4
a_4	a_1	1	y_5, y_6
a_5	a_1	$\neg x_4$	-
	a_5	$x_3 x_4$	y_1, y_4
	a_3	$\neg x_3 x_4$	-

3.2. Структурный синтез МПА.

Структурный синтез микропрограммного автомата состоит в построении структурной схемы автомата и выполняется в соответствии с общим *алгоритмом канонического метода* синтеза:

- 1) выбирается тип и определяется количество автоматов памяти (АП) по формуле

$$R > \lceil \log_2 M \rceil ,$$

где M – число состояний автомата.

- 1) кодируются состояния автомата;
- 2) составляется структурная таблица автомата;
- 3) по структурной таблице составляется система логических уравнений, связывающих функции выходов и возбуждения с состояниями автомата и входными сигналами;
- 4) проводится минимизация системы логических функций;
- 5) на выбранной элементной базе синтезируется комбинационная схема автомата и строится его структурная схема.

Отличие процедуры синтеза МПА заключается в принципе построения структурной таблицы. В общем случае эта таблица содержит столбцы состояний a_m и a_s , входных сигналов $X(a_m, a_s)$, выходных сигналов $Y(a_m, a_s)$, сигналов возбуждения $F(a_m, a_s)$ и кодов состояний. Каждая строка такой таблицы соответствует одному пути перехода на графе автомата.

При построении функции возбуждения используются таблицы переходов или выходов автоматов памяти. В качестве АП может использоваться любой автомат Мура, но наиболее просто функции возбуждения строятся для элементарных АП, к которым относятся триггеры.

Таблицы переходов и входов D –, T –, RS – и JK-триггеров приведены в работе №5.

Пример

На рис.6.7 приведен граф автомата Мили S_6 . Требуется составить структурную таблицу МПА и систему канонических уравнений.

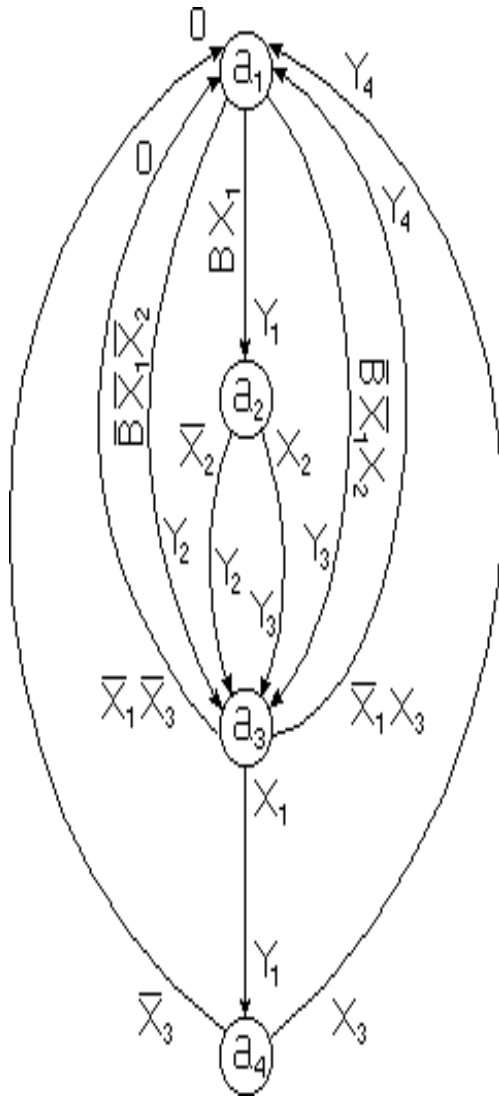


Рис.6.7. Граф автомата Мили S_6

Решение. Автомат S_6 имеет 4 состояния, для кодирования которых необходимо $R = \log_2 4 = 2$ триггера. В качестве автомата памяти будем использовать RS-триггер.

Для кодирования можно использовать метод единичного кодирования (лаб. работа №4), тогда коды состояний будут такими:

$$K(a_1) = 00, \quad K(a_2) = 01, \quad K(a_3) = 10, \quad K(a_4) = 11.$$

Структурная таблица отличается от таблицы переходов тем, что содержит коды состояний и сигналы возбуждения, формируемые на переходе (a_m, a_s). Для RS – триггера в столбце сигналов возбуждения указывается сигнал S, если триггер переключается из состояния 0 в состояние 1, и сигнал R, если из состояния 1 в состояние 0.

Табл.6.2. представляет собой расширенную таблицу переходов и содержит помимо кодов состояний сигналы возбуждения, подаваемые на входы соответствующих триггеров.

Таблица 6.2

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$
a_1	$0\ 0$	a_2	$0\ 1$	$\overline{B} \overline{x_1}$	y_1	S_2
		a_3	$1\ 0$	$B \overline{x_1} x_2$	y_2	S_1
		a_3	$1\ 0$	$B \overline{x_1} x_2$	y_3	S_1
a_2	$0\ 1$	a_3	$1\ 0$	$-$	y_2	$S_1\ R_2$
		a_3	$1\ 0$	x_2	y_3	$S_1\ R_2$
a_3	$1\ 0$	a_1	$0\ 0$	$\overline{x_1} \overline{x_3}$	$-$	R_1
		a_1	$0\ 0$	$\overline{x_1} x_3$	y_4	R_1
		a_4	$1\ 1$	x_1	y_1	S_2
a_4	$1\ 1$	a_1	$0\ 0$	$\overline{x_3}$	$-$	$R_1\ R_2$
		a_1	$0\ 0$	x_3	y_4	$R_1\ R_2$

Ниже приводится система канонических уравнений автомата S_6 .

$$y_1 = \overline{\tau_1} \overline{\tau_2} B \overline{x_1} V \overline{\tau_1} \overline{\tau_2} x_1,$$

$$y_2 = \overline{\tau_1} \overline{\tau_2} B \overline{x_1} x_2 V \overline{\tau_1} \overline{\tau_2} x_1 x_2,$$

$$y_3 = \overline{\tau_1} \overline{\tau_2} B \overline{x_1} x_2 V \overline{\tau_1} \overline{\tau_2} x_1 x_2,$$

$$y_4 = \overline{\tau_1} \overline{\tau_2} \overline{x_1} x_3 V \overline{\tau_1} \overline{\tau_2} x_3.$$

$$R_1 = \overline{\tau_1} \overline{\tau_2} V \overline{\tau_1} \overline{\tau_2} x_1,$$

$$R_2 = \overline{\tau_1} \overline{\tau_2} V \overline{\tau_1} \overline{\tau_2},$$

$$S_1 = \overline{\tau_1} \overline{\tau_2} B \overline{x_1} V \overline{\tau_1} \overline{\tau_2},$$

$$S_2 = \overline{\tau_1} \overline{\tau_2} B \overline{x_1} V \overline{\tau_1} \overline{\tau_2} x_1.$$

Далее выполняются п.п.5 и 6 алгоритма канонического метода синтеза структурного автомата.

При синтезе МПА по граф-схеме алгоритма предполагается, что автомат имеет наименьшее число состояний, так как это обусловлено

самим принципом построения граф-схемы. Однако в некоторых задачах можно сократить число состояний автомата путем выявления 0-, 1-, 2-, ..., k -эквивалентных состояний. По существу этот метод мало отличается от метода минимизации автоматов общего типа, однако здесь используется более слабое отношение эквивалентности, что дает возможность провести минимизацию.

0-эквивалентными называются состояния, которые при одинаковых входных сигналах характеризуются одинаковыми выходными сигналами. 1-эквивалентными называются состояния, переход из которых при одних и тех же входных сигналах происходит в одни и те же 0-эквивалентные состояния и т.п.

Метод минимизации заключается в разбиении множества состояний автомата на эквивалентные классы. Строятся классы 0-эквивалентных состояний (π_0), 1-эквивалентных, 2-эквивалентных и т.д. Множество 1-эквивалентных классов обозначается (π_1) и т.д., пока на каком-то $(k+1)$ -м шаге не окажется, что $\pi_k = \pi_{k+1}$, т.е. разбиение на (k) -эквивалентные и $(k+1)$ -эквивалентные классы совпадает. Последнее разбиение π_{k+1} будет соответствовать минимальному автомату.

4. Задания выполнению работы

4.1. Синтезировать структурную схему автомата S_6 , используя в качестве АП D – триггеры.

4.2. Синтезировать структурную схему автомата S_6 , используя в качестве АП T – триггеры.

4.3. Построить автомат Мура S_7 , эквивалентный автомату Мили S_6 ; составить таблицу переходов МПА.

4.4. Синтезировать структурную схему автомата S_7 , в качестве АП использовать D – триггеры.

4.5. Решить задачу 4.4 для RS – триггера.

4.6. Решить задачу 4.4. для T – триггера.

4.7. Построить граф автомата Мура, реализующего микропрограмму на рис.6.8., его таблицу переходов и синтезировать схему на RS – триггерах.

4.8. Построить граф и таблицу переходов автомата Мили, реализующего микропрограмму на рис.6.8. Синтезировать схему автомата на D-триггерах.

4.9. Решить задачу 4.7 для T-триггеров.

4.10. Решить задачу 4.7 для D-триггеров

4.11. Решить задачу 4.8 для T-триггеров

4.12. Решить задачу 4.8 для RS-триггеров

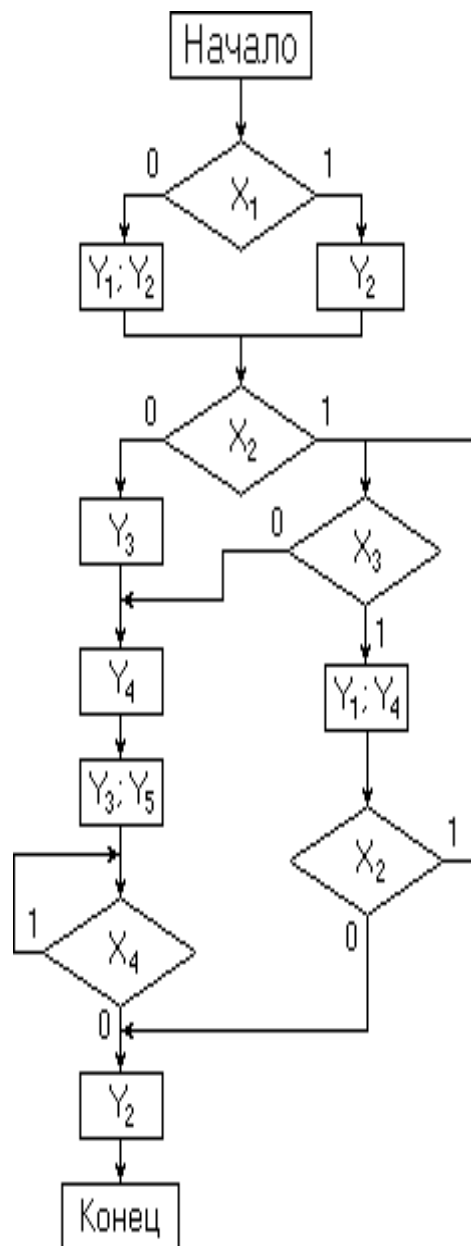


Рис.6.8. Микропрограммный автомат

5. Контрольные вопросы

- 5.1. Назовите функции операционного и управляющего автоматов.
- 5.2. Приведите основные положения принципа микропрограммного управления.
- 5.3. Объясните структуру МПА, полученную В.М. Глушковым.
- 5.4. Способы задания микропрограммных автоматов.
- 5.5. Что называется микропрограммой?

- 5.6. Объясните отличие содержательной ГСА и ГСА.
- 5.7. Что такое логическая схема алгоритма?
- 5.8. Приведите правила построения ЛСА.
- 5.9. Как происходит переход от ГСА к ЛСА?
- 5.10. Что такое функция перехода?
- 5.11. Что такое формула перехода?
- 5.12. Что такое матричная схема алгоритма?
- 5.13. Приведите основные этапы абстрактного синтеза.
- 5.14. В чем отличие абстрактного синтеза автоматов Мили и Мура?
- 5.15. Как строится таблица переходов или граф МПА?
- 5.16. Какую структуру имеет структурная таблица переходов МПА?

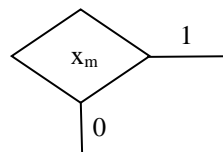
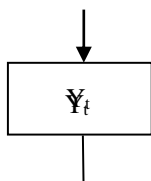
Правила составления логической схемы алгоритма (ЛСА)

1. Обозначения:

\mathbf{Y}_t – символ, обозначающий безусловный оператор, который записывается в прямоугольник (t – номер оператора);

\mathbf{Y}_n – начальный оператор, \mathbf{Y}_k – конечный оператор;

\mathbf{x}_m – символ, обозначающий условный оператор, записываемый в ромб, который обязательно должен иметь обозначения выходов или одного из них (единичного или нулевого), (m – номер условного оператора).



2. Отметка граф-схемы алгоритма (ГСА).

Входы всех вершин ГСА, к которым подходит более одной дуги (стрелки), а также вход последней вершины оператора \mathbf{Y}_k , даже если к ней подходит одна дуга, отмечаются «жирными» точками на этой дуге с числовой отметкой в виде натуральных чисел $1, 2, \dots, S$, от начального оператора \mathbf{Y}_n к конечному оператору \mathbf{Y}_k .