

ИСПОЛЬЗОВАНИЕ М-МОДЕЛИ ДЛЯ ОПТИМИЗАЦИИ ЛОГИЧЕСКОЙ СХЕМЫ АВТОМАТА С «ЖЕСТКОЙ» ЛОГИКОЙ

Баркалов А. А., Самир Нахлави
Донецкий Государственный Технический
Университет, кафедра ЭВМ
тел: 91-07-35
e-mail: mold@dstu.donetsk.ua

Abstract

Barkalov A., Nachlavi S. The application of the M-model for the optimization of the logical circuit of the automaton with hardwired logic. The methods for solving of the basic problems connected with synthesis of automaton with use of M-model (M-automaton) are proposed. The M-automaton' transitions are determined not more then by one logic condition. The example of M-automaton's synthesis on multiplexes and ROMs is given, the field of the effective utilisation of suggested method is designed.

Аннотация

Рассмотрены методы решения основных задач при синтезе автомата с использованием М-модели (М-автомат). В М-автоматах переходы определяются не более, чем одним логическим условием. Приведен пример синтеза М-автомата на мультиплексорах и ПЗУ, определена область эффективного использования М-модели.

Введение

Устройство управления является важнейшим блоком цифровых систем [1] и часто реализуется на практике как автомат с «жесткой» логикой [2]. В настоящее время для реализации логических схем автоматов с «жесткой» логикой (АЖЛ) широко используются программируемые БИС с регулярной структурой типа ПЗУ, ПЛМ, ПЛИС [3]. При этом одной из важнейших задач структурной теории автоматов является построение схемы с минимально возможным числом БИС. Оптимальное решение обычно достигается при совместном использовании методов структурной редукции (увеличение числа уровней в схеме), гетерогенной реализации (использование смешанного элементного базиса) и оптимального кодирования состояний и промежуточных переменных [4]. В работе [5] рассматривается задача синтеза автомата на заказных матричных БИС и для минимизации площади кристалла предлагается М-модель. Характерное свойство М-модели – каждый переход автомата определяется не более, чем одним логическим условием. В настоящее время предлагается использование М-модели для реализации схемы автомата Мили на стандартных БИС и рассматриваются методы решения некоторых задач, возникающих в этом случае.

Метод синтеза М-автомата

Пусть задана граф-схема алгоритма (ПА) Γ (рис. 1), содержащая $L=3$ логических условий $x_i \in X = \{x_1, \dots, x_L\}$ и $N=5$ микроопераций $y_n \in Y = \{y_1, \dots, y_N\}$. Структура логической схемы АЖЛ Мили на основе М-модели включает (рис. 2):

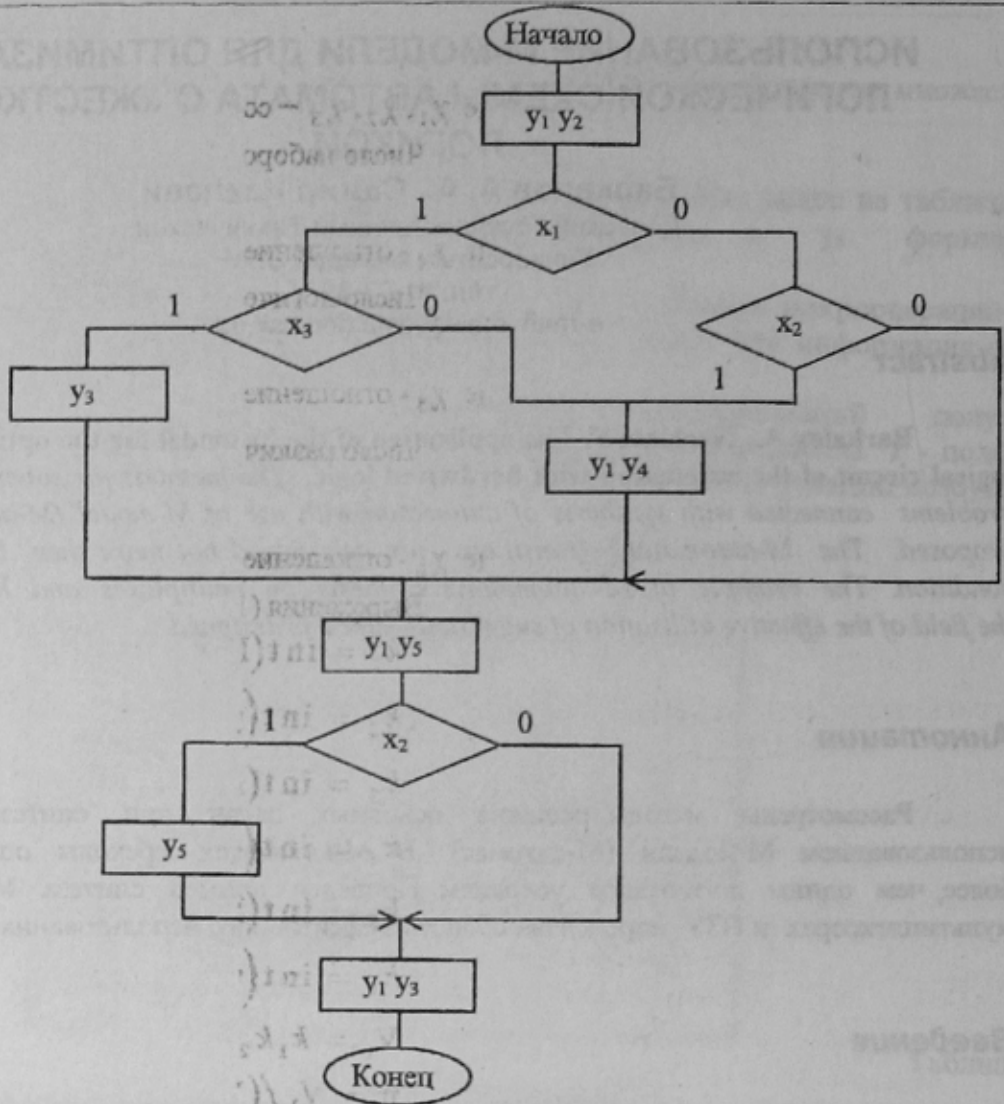


Рис. 1. Граф-схема алгоритма Г.

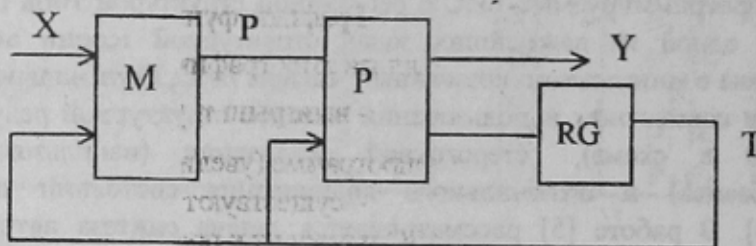


Рис. 2. Структурная схема М-автомата.

1. М-подсхему для замены множества логических условий (ЛУ) X переменной P, которая задается выражением

$$P = \bigvee_{m=1}^M C_{m1} A_m X_i \tag{1}$$

Здесь A_m – конъюнкция внутренних переменных $T_p \in T = \{T_1, \dots, T_R\}$, соответствующая коду состояния $a_m \in A = \{a_1, \dots, a_M\}$ автомата, $R = \text{int} \log_2 M$; c_{m1} – булева переменная, равная единице только, если переход из состояния a_m определяется переменной x_i . Для реализации М-подсхемы целесообразно использовать мультиплексоы.

2. Р-подсхему, реализующую систему булевых функций

$$\begin{aligned} \Phi &= \Phi(T, p); \\ Y &= Y(T, p), \end{aligned} \quad (2)$$

где $\Phi = \{\phi_1, \dots, \phi_R\}$ – множество функций возбуждения памяти автомата, которая представляет собой R-разрядный регистр RG. Система (2) формируется по прямой структурной таблице автомата [2]. В силу регулярности системы (2) для ее реализации целесообразно использовать ПЗУ. Это позволяет уменьшить стоимость схемы по сравнению с известными структурами, в которых для реализации R-подсхемы используется ПЛИМ или ПЛИС.

Для синтеза автомата Мили с использованием M-модели (условимся называть такой автомат M-автоматом) предлагается следующий алгоритм

1. Преобразование исходной ГСА.
2. Оптимальное кодирование состояний.
3. Формирование системы (1).
4. Формирование модифицированной прямой структурной таблицы и системы (2).
5. Синтез логической схемы в заданном элементном базисе.

Рассмотрим пример синтеза логической схемы M-автомата по ГСА Г. Преобразование исходной ГСА необходимо выполнить таким образом, чтобы после каждой операторной вершины было не более одной условной вершины. Преобразование выполняется тривиально – между любой парой условных вершин вводится операторная вершина, не содержащая микроопераций. Для нашего примера преобразованная граф-схема алгоритма (ПГСА) представлена на рис.3.

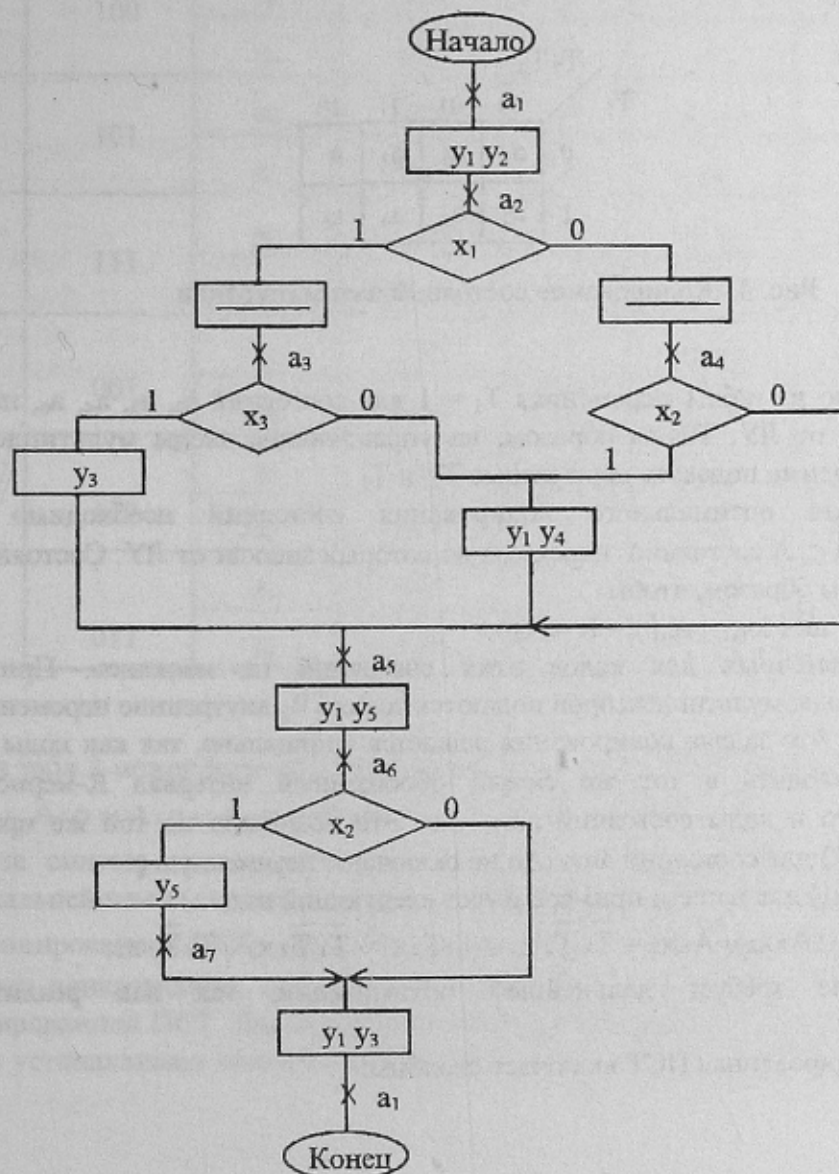


Рис. 3. Преобразованная граф-схема алгоритма

Отметка состояний автомата Мили позволяет получить множество $A = \{a_1, \dots, a_7\}$, для кодирования состояний необходимо $R=3$ внутренние переменные $T_r \in T$. Отметим, что для ГСА Γ (рис. 1) необходимо $M_0 = 5$ отметок и число внутренних переменных для кодирования состояний $R_0 = \text{int} \log_2 M_0 = 3$. Таким образом, в данном случае преобразование ГСА не приводит к увеличению мощности множества внутренних переменных.

Кодирование состояний необходимо выполнить таким образом, чтобы минимизировать число мультиплексоров в M -подсхеме. В соответствии с (1) код состояния определяет логическое условие, заменяемое переменной p . Построим таблицу замены логических условий (табл. 1) со столбцами: x_i – заменяемое логическое условие; a_m – состояние, переход из которого определяется условием x_i ; $k(a_m)$ – код состояния $a_m \in A$. Состояния закодированы в соответствии с картой Карно (рис. 4).

Таблица 1

Таблица замены логических условий

x_i	a_m	$k(a_m)$
x_1	a_2	100
x_2	a_4	111
x_3	a_3	101
x_2	a_6	110

		$T_2 T_3$			
		00	01	11	10
T_1	0	a_1	a_5	a_7	*
	1	a_2	a_3	a_4	a_6

Рис. 4. Кодирование состояний автомата Мили.

Как видно из табл. 1 переменная $T_1 = 1$ для состояний a_2, a_3, a_4, a_6 , переходы из которых зависят от ЛУ. Таким образом, на управляющие входы мультиплексоров M -подсхемы необходимо подавать переменные T_2 и T_3 .

Итак, для оптимального кодирования состояний необходимо выделить подмножество $A_0 \subseteq A$ состояний, переходы из которых зависят от ЛУ. Состояния $a_m \in A_0$ кодируются таким образом, чтобы

$$R_1 = R - \text{int}(\log_2 |A_0|) = R - R_2$$

внутренних переменных для кодов этих состояний не менялись. При этом на управляющие входы мультиплексоров подаются только R_2 внутренние переменные.

Отметим, что задача кодирования решается тривиально, так как коды состояний $a_m \notin A_0$ могут входить в тот же самый обобщенный интервал R -мерного булева пространства, что и коды состояний $a_m \in A_0$. Это возможно по той же причине, что термы системы (2) для состояний $a_m \notin A_0$ не включают переменную p .

Система (1) для нашего примера имеет следующий вид:

$$P = A_2 x_1 \vee A_4 x_2 \vee A_3 x_3 \vee A_6 x_2 = \bar{T}_1 \bar{T}_2 x_1 \vee T_1 T_2 x_2 \vee \bar{T}_1 \bar{T}_2 x_3 \vee T_1 \bar{T}_2 x_2.$$

Эта система не требует дальнейшей оптимизации, так как реализуется на мультиплексоре.

Модифицированная ПСТ включает столбцы:

a_m – исходное состояние М-автомата; $k(a_m)$ – код исходного состояния; a_s – состояние перехода; $k(a_s)$ – код исходного состояния; P – входной сигнал, определяющий переход из a_m в a_s ; Y_h – выходной сигнал, формируемый на переходе из a_m в a_s ; Φ_h – набор функций возбуждения, необходимый для переключения памяти автомата из $k(a_m)$ в $k(a_s)$; $h=1, \dots, H$ – номер перехода. Поскольку Р-система реализуется на ПЗУ, то имеется одно существенное отличие от классических методов построения ПСИ [2]. Для безусловных переходов в ПСТ отводится две строки ($p=1, p=0$) с одинаковым содержимым столбцов $a_m, k(a_m), a_s, k(a_s), Y_h$ и Φ_h .

Для нашего примера модифицированная ПСТ (табл.2) содержит $H=2M=14$ строк, для реализации Р-подсхемы требуется ПЗУ с $S=R+1=4$ – адресными входами и $t=R+N=8$ выходами. Задача реализации Р-подсхемы на ПЗУ во многом аналогична задаче реализации управляющей памяти автоматов с программируемой логикой и достаточно изучена [3]. В качестве элементного базиса для реализации регистра в примере использованы D-триггера.

Таблица 2

Модифицированная ПСТ автомата Мили

a_m	$k(a_m)$	a_s	$k(a_s)$	P	Y_h	Φ_h	h
a_1	000	a_2	100	0	$y_1 y_2$	D_1	1
		a_2	100	1	$y_1 y_2$	D_1	2
a_2	100	a_4	111	0	—	$D_1 D_2 D_3$	3
		a_3	101	1	—	$D_1 D_3$	4
a_3	101	a_5	001	0	$y_1 y_4$	D_3	5
		a_5	001	1	y_3	D_3	6
a_4	111	a_6	110	0	$y_1 y_5$	$D_1 D_2$	7
		a_5	001	1	$y_1 y_4$	D_3	8
a_5	001	a_6	110	0	$y_1 y_5$	$D_1 D_2$	9
		a_6	110	1	$y_1 y_5$	$D_1 D_2$	10
a_6	110	a_1	000	0	$y_1 y_3$	—	11
		a_7	011	1	y_5	$D_2 D_3$	12
a_7	011	a_1	000	0	$y_1 y_3$	—	13
		a_1	000	1	$y_1 y_3$	—	14

Из табл.2 может быть тривиально получена система (2), например,

$$y_1 = A_1 \bar{p} \vee A_1 p \vee A_3 \bar{p} \vee A_4 \bar{p} \vee A_5 \bar{p} \vee A_5 p \vee A_6 \bar{p} \vee A_7 \bar{p} \vee A_7 p.$$

Для синтеза логической схемы необходимо сформировать содержимое ПЗУ, чтобы в дальнейшем выполнить его программирование. Эта задача решается тривиально по модифицированной ПСТ – адрес ПЗУ определяется конкатенацией $k(a_m)*P$, где * –

знак конкатенации, содержимое ячейки ПЗУ определяется столбцами Y_h и Φ_h модифицированной ПСТ. Для нашего примера содержимое ПЗУ приведено в табл.3, где столбец h устанавливает соответствие между словом ПЗУ и строкой модифицированной ПСТ.

Таблица 3

Содержимое ПЗУ Р-подсхемы

Адрес $k(a_m) * P$	Содержимое									h
	y_1	y_2	y_3	y_4	y_5	D_1	D_2	D_3		
0000	1	1	0	0	0	1	0	0		1
0001	1	1	0	0	0	1	0	0		2
0010	1	0	0	0	1	1	1	0		9
0011	1	0	0	0	1	1	1	0		10
0100	0	0	0	0	0	0	0	0		
0101	0	0	0	0	0	0	0	0		
0110	1	0	1	0	0	0	0	0		13
0111	1	0	1	0	0	0	0	0		14
1000	0	0	0	0	0	1	1	1		3
1001	0	0	0	0	0	1	0	1		4
1010	1	0	0	1	0	0	0	1		5
1011	1	0	0	0	1	1	1	0		6
1100	1	0	1	0	0	0	0	0		11
1101	0	0	0	0	1	0	1	1		12
1110	1	0	0	0	1	1	1	0		7
1111	1	0	0	1	0	0	0	1		8

Как видно из табл.3 адреса 0100 и 0101, соответствующие неопределенности на рис.4, не содержат полезной информации. В общем случае число таких ячеек m определяется выражением $m=2^{R+1}-2M$.

Схема М-автомата для нашего примера показана на рис.5, при этом использован мультиплексор с двумя управляющими входами и ПЗУ с $S=t=4$, что привело к наличию в схеме 2-х корпусов ПЗУ. При управлении регистром сигнал Start используется для установки автомата в исходное состояние a_1 , сигнал Clock – для синхронизации.

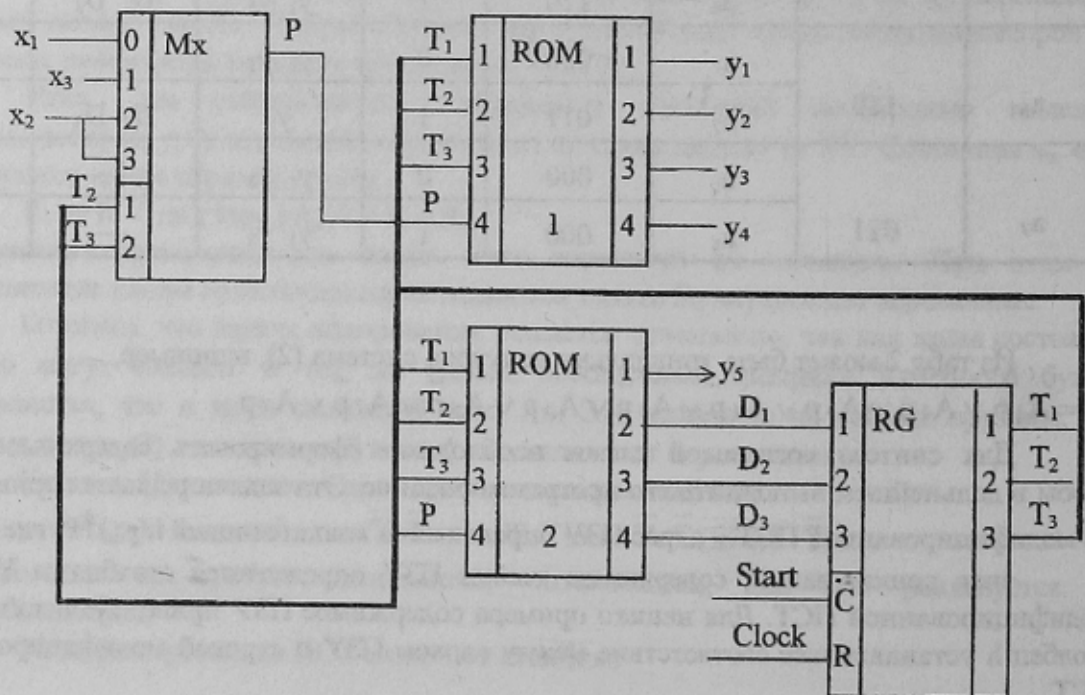


Рис. 5. Логическая схема М-автомата

Заключение

Как показали исследования использование М-модели позволяет на 18-25% уменьшить число БИС в схеме автомата по сравнению с классической двухуровневой схемой с заменой входных переменных [3]. При этом выигрыш тем больше, чем ближе количество внутренних переменных в М-автомате и автомате, синтезируемой по исходной граф-схеме алгоритма.

Литература

1. Stalling W. Computer Organization and Architecture.—New Jersey: Prentice Hall, 1996.—683 pp.
2. Баранов С.И. Синтез микропрограммных автоматов.—Л.: Энергия, 1979.—232с.
3. Bostock G. Programmable Logic Handbook.—London: Collins Professional Books, 1988.—243 pp.
4. Баранов С.И., Скляр В.А. Цифровые устройства на БИС с матричной структурой.—М.: Радио и связь, 1986.—272 с.