

СИНТЕЗ МИКРОПРОГРАММНОГО АВТОМАТА С МНОГОНАПРАВЛЕННЫМИ ПЕРЕХОДАМИ

Зеленёва И.Я.

Донецкий национальный технический университет,
кафедра ЭВМ

Abstract. I.J. Zelenyova. Synthesis of microprogramming automata with multi-branched transitions. Method of synthesis of control automata with high number of logical conditions is proposed. The method is oriented to reducing of hardware amount for limited performance. The method is based on the state's classification and uses paging organization of control memory. Example of automaton synthesis is given.

Введение. Устройство управления (УУ) является одним из важнейших блоков любой цифровой системы, и на практике часто реализуется в виде автомата с жесткой логикой в современном элементном базисе, включающем БИС с регулярной структурой (ПЛИМ, ПЗУ, ПЛИС) [1, 5-6]. Схемы УУ могут быть реализованы на заказных, полужаказных или стандартных БИС. В случае реализации на заказных БИС аппаратные затраты оцениваются суммарной площадью кристаллов. Если же схема реализуется на стандартных БИС, то показателем эффективности является количество корпусов микросхем, необходимых для реализации схемы УУ при заданном быстродействии. Таким образом, для уменьшения стоимости схем необходимо стремиться к уменьшению площади заказных БИС, либо к сокращению числа стандартных БИС, используемых в схеме устройства. Удешевления схемы можно достичь, применяя методы замены входных переменных [2-3], а также многоуровневую реализацию [1], но эти методы ориентированы на граф-схемы алгоритмов (ГСА) с небольшим числом (до трёх) логических условий на переходах из текущего состояния автомата в следующее. Кроме того, увеличение числа уровней в схеме ведёт к снижению её быстродействия, что в современных условиях является отрицательной чертой.

Предлагаемый метод синтеза управляющего автомата с жесткой логикой по ГСА с многонаправленными переходами позволяет достичь снижения стои-

мости схемы автомата без существенного увеличения времени работы устройства. Основу метода составляет разделение множества состояний автомата на два класса по признаку наличия или отсутствия безусловных переходов, — а также использование страничной организации управляющей памяти.

Синтез управляющего автомата по ГСА с многонаправленными переходами. Рассмотрим пример синтеза автомата Мура по произвольной ГСА, в которой не менее 25% вершин — условные (т.к. в данном случае рассматриваются только нелинейные ГСА). При этом переходы из каждого текущего состояния $a_m \in A$ могут зависеть от различного числа логических условий (ЛУ), либо выполняются безусловно. Автомат задается множествами:

$$A = \{a_1, \dots, a_M\}, \quad X = \{x_1, \dots, x_L\}, \quad Y = \{y_1, \dots, y_N\}, \quad (1)$$

где A — множество состояний автомата, X — множество логических условий, Y — множество выходных сигналов. В автомате Мура зависимости для выходных сигналов Y и функций возбуждения памяти Φ выражены формулами:

$$Y^t = f(A^t), \quad \Phi^{t+1} = f(A^t, X), \quad (2)$$

где A^t — состояние автомата в текущий момент времени. Состояние автомата кодируется внутренними переменными $T_r \in T = \{T_0, \dots, T_R\}$, где

$$R = \lceil \log_2 M \rceil. \quad (3)$$

Прямая структурная таблица (ПСТ) автомата Мура S1, построенная с учетом данных условий, приведена в таблице 1. Граф алгоритма, соответствующий табл. 1, имеет следующие характеристики:

- число логических условий в ГСА $L=12$;
- максимальное число L_m логических условий $X(a_m)$ на переходе из одного состояния a_m ; в данном примере $L_m(a_3)=5$, т.к. $X(a_3)=\{x_4, x_5, x_6, x_7, x_8\}$;
- число состояний автомата $M=14$, значит, для их кодирования необходимо $R = \lceil \log_2 14 \rceil = 4$ внутренних переменных $T_r \in T$;
- число выходных сигналов $N=6$;
- число переходов, совпадающее с числом строк ПСТ, $H=28$.

Пусть регистр памяти автомата реализован стандартно на D-триггерах, и в столбце функций возбуждения памяти указаны переменные $D_r \in \Phi$, соответствующие переменным $T_r \in T$.

Будем рассматривать множество ГСА, для которых выполняется условие:

$$L_m \ll L \quad (4)$$

Таблица 1 — Прямая структурная таблица автомата Мура

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	$Y(a_m)$	Φ	τ	h
a_0	0000	a_1	1000	1	—	D_0	—	1
a_1	1000	a_2	0001	x_1	y_1y_2	D_3	—	2
		a_5	0010	$\overline{x_1} \overline{x_2}$		D_2	τ_3	3
		a_{11}	0110	$\overline{x_1} x_2$		D_2D_3	τ_2	4
a_2	0001	a_3	1001	1	y_4	D_0D_3	—	5
a_3	1001	a_4	1010	x_3	y_2y_5	D_0D_2	τ_3	6
		a_6	0011	$\overline{x_3}$		D_2D_3	τ_2	7
a_4	1010	a_7	0100	$\overline{x_4} \overline{x_5} \overline{x_6}$	y_3y_4	D_1	—	8
		a_{10}	1100	$\overline{x_4} \overline{x_5} x_6$		D_0D_1	τ_3	9
		a_8	1011	$\overline{x_4} \overline{x_5} x_7$		$D_0D_2 D_3$	τ_2	10
		a_{12}	1101	$\overline{x_4} \overline{x_5} \overline{x_7}$		$D_0D_1 D_3$	$\tau_2 \tau_3$	11
		a_{11}	0110	$x_4 x_8$		D_1D_2	τ_1	12
		a_1	1000	$x_4 \overline{x_8}$		D_0	$\tau_1 \tau_3$	13
a_5	0010	a_6	0011	1	y_1y_2	D_2D_3	—	14
a_6	0011	a_7	0100	1	y_5	D_1	—	15
a_7	0100	a_8	1011	1	y_3	$D_0D_2 D_3$	—	16
a_8	1011	a_9	0101	$x_6 x_7 x_8 x_9$	y_5y_6	D_1D_3	—	17
		a_9	0101	$\overline{x_6}$		D_1D_3	τ_3	18
		a_9	0101	$x_6 \overline{x_7}$		D_1D_3	τ_2	19
		a_9	0101	$x_6 x_7 \overline{x_8}$		D_1D_3	$\tau_2 \tau_3$	20
		a_8	1011	$x_6 x_7 x_8 \overline{x_9}$		$D_0D_2 D_3$	τ_1	21
a_9	0101	a_{10}	1100	1	y_2	D_0D_1	—	22
a_{10}	1100	a_{11}	0110	x_{12}	y_1y_3	D_1D_2	—	23
		a_{12}	1101	$\overline{x_{12}}$		$D_0D_1 D_3$	τ_3	24
a_{11}	0110	a_{12}	0110	1	y_4	D_1D_2	—	25
a_{12}	1101	a_{13}	0111	$x_{10} x_{11}$	$y_1y_3y_5$	$D_1D_2 D_3$	—	26
		a_5	0010	$x_{10} \overline{x_{11}}$		D_2	τ_3	27
		a_0	0000	$\overline{x_{10}}$		—	τ_2	28
a_{13}	0111	a_0	0000	1	y_6	—	—	29

Для решения задачи синтеза автомата Мура по граф-схеме алгоритма, для которой выполняется условие (4), предлагается метод синтеза, направленный на уменьшение аппаратных затрат в схеме автомата и включающий следующие этапы:

1. Разбиение множества состояний автомата на классы — с условными и безусловными переходами.
2. Специальное кодирование состояний в сформированных классах.
3. Формирование адресов переходов для каждого класса состояний.
4. Формирование выходных сигналов.

В табл.1 коды состояний записаны с учетом распределения по подмножествам и специального кодирования.

Нужно заметить, что для универсальности алгоритма необходимо, чтобы переход из состояния a_0 всегда был безусловным. Поэтому, если в ГСА за начальной вершиной следует условная, то перед ней необходимо ввести операторную вершину с нулевым набором микроопераций.

Рассмотрим каждый этап применения данного метода более подробно.

1. Разбиение множества состояний автомата на классы

Множество состояний автомата $A = \{a_1, \dots, a_M\}$ можно разделить на два подмножества, которые обозначим B_0 и B_1 . Подмножество B_0 включает состояния $a_i \in A$, переходы из которых не зависят от логических условий, т.е. $X(a_i) = \emptyset$. Подмножество B_1 включает состояния $a_j \in A$, переходы из которых зависят от произвольного числа логических условий $x_1 \in X$, то есть $x(a_j) \subseteq X$.

В данном примере подмножества B_1 и B_0 сформированы следующим образом:

$$\begin{aligned} B_0 &= \{a_0, a_2, a_5, a_6, a_7, a_9, a_{11}, a_{13}\} \\ B_1 &= \{a_1, a_3, a_4, a_8, a_{10}, a_{12}\} \end{aligned} \quad (5)$$

В дальнейшем, в схеме автомата, функции переходов и выходных сигналов, зависящие от состояний $a_i \in B_1$, можно реализовать на ПЗУ, а функции, зависящие от $a_j \in B_0$ — на ПЛМ.

2. Специальное кодирование состояний в подмножествах

Как было определено ранее, число состояний автомата Мура в данном примере $M=14$, разрядность кода состояний $R=4$. Закодируем состояния $a_i \in B_0$ двоичными переменными $\{T_0 \dots T_3\} \in T$ так, чтобы старший разряд кода $K(a_i)$ был равен 0, а состояния $a_j \in B_1$ — так, чтобы старший разряд кода $K(a_j)$ был равен 1. Таким образом, старший разряд кода можно использовать, как признак класса.

Кодирование состояний, входящих в класс B_0 , приведено в таблице 2. Здесь для каждого исходного состояния $a_i \in B_0$ в поле FA указан код следующего состояния, т.е., по сути — адрес безусловного перехода, а также в поле FУ указан код набора микроопераций (МО), соответствующий данному исходному

состоянию автомата. Таким образом, эта информация представлена в формате, удобном для хранения в ПЗУ, причем компактность обеспечивается за счет максимального кодирования наборов микроопераций [1].

Таблица 2 — Таблица прошивки ПЗУ для состояний класса B_0

a_i	Адрес ПЗУ			Адрес перехода			Код набора МО			
	$K(a_i)$			FA			FY			
	a_1	a_2	$a_3 a_4$	T_n	T_1	$T_2 T_3$	z_1	z_2	$z_3 z_4$	
a_0	0	0 0 0		1	0 0 0		0 0 0 0			
a_2	0	0 0 1		1	0 0 1		0 0 1 0			
a_5	0	0 1 0		0	0 1 1		0 0 0 1			
a_6	0	0 1 1		0	1 0 0		0 1 0 1			
a_7	0	1 0 0		1	0 1 1		0 1 1 0			
a_9	0	1 0 1		1	1 0 1		1 0 0 0			
a_{11}	0	1 1 0		1	1 0 1		0 0 1 0			
a_{13}	0	1 1 1		0	0 0 0		0 1 0 0			

В данном примере ПСТ содержит $N_y = 12+1 = 13$ различных наборов МО (с учетом одного нулевого набора, соответствующего состоянию a_0), поэтому для максимального кодирования наборов необходимо

$$R_y = \lceil \log_2 N_y \rceil = \lceil \log_2 13 \rceil = 4 \quad (6)$$

двоичных переменных, которые образуют множество $Z = \{z_1, z_2, z_3, z_4\}$. Кодирование наборов МО приведено в таблице 3.

Таблица прошивки ПЗУ для состояния класса B_1 (табл.4) в каждом слове хранит в поле FA адреса всех условных переходов для каждого исходного состояния $a_j \in B_1$ и код соответствующего набора МО в поле FY. Информация в табл.4 организована способом страничной адресации памяти [4]. Адрес слова ПЗУ определяется, как конкатенация кода исходного состояния $a_m \in B_1$ и двоичного кода номера перехода из данного состояния. Код $K(a_m)$, как было определено выше, представляет собой набор $\{T_0, \dots, T_R\} \in T$, где T_0 — это признак условия. Для всех $a_m \in B_1$ $T_0 = 1$, значит $\lceil T_0 = 0$ можно использовать, как управляющий сигнал для соответствующих схем ПЗУ, а набор $\{T_1, \dots, T_R\}$ использовать в качестве адреса страницы. Разрядность двоичного номера перехода, закодированного набором $\{\tau_1, \dots, \tau_k\} \in \tau$, определяется как

$$K = \lceil \log_2 H_m \rceil, \quad (7)$$

где H_m — максимальное число переходов из состояния $a_m \in B_1$ в ПСТ. Таким образом адрес слова ПЗУ для состояний класса B_1 формируется, как

$$A_1 = \{T_1, \dots, T_R\} \cup \{\tau_1, \dots, \tau_k\}. \quad (8)$$

В данном примере $H_m=6$ (строки 8-13 табл.1), значит, $K=\lceil \log_2 6 \rceil = 3$ и $A_1 = \{T_1, T_2, T_3\} \cup \{\tau_1, \tau_2, \tau_3\}$.

Таблица 3 — Кодирование наборов микроопераций

Y_i	Микрооперации	код Y_i				Y_i	Микрооперации	код Y_i			
		z_1	z_2	z_3	z_4			z_1	z_2	z_3	z_4
Y_1	—	0	0	0	0	Y_7	$y_5 y_6$	0	1	1	1
Y_2	$y_1 y_2$	0	0	0	1	Y_8	y_2	1	0	0	0
Y_3	y_4	0	0	1	0	Y_9	$y_1 y_3$	1	0	0	1
Y_4	$y_2 y_5$	0	0	1	1	Y_{10}	y_4	1	0	1	0
Y_5	$y_3 y_4$	0	1	0	0	Y_{11}	$y_1 y_3 y_5$	1	0	1	1
Y_6	y_5	0	1	0	1	Y_{12}	y_6	1	1	0	0
Y_7	y_3	0	1	1	0	—	—	—	—	—	—

Таблица 4 — Фрагмент таблицы прошивки ПЗУ для состояний класса B_1

a_j	Адрес слова ПЗУ			Содержимое памяти	
	T_0	$T_1 T_2 T_3$	$\tau_1 \tau_2 \tau_3$	ФА	FY
...	
a_4	1	0 1 0	0 0 0	(a_7) 0100	0 1 0 0
	1	0 1 0	0 0 1	(a_{10}) 1100	0 1 0 0
	1	0 1 0	0 1 0	(a_8) 1011	0 1 0 0
	1	0 1 0	0 1 1	(a_{12}) 1101	0 1 0 0
	1	0 1 0	1 0 0	(a_{11}) 0110	0 1 0 0
	1	0 1 0	1 0 1	(a_1) 1000	0 1 0 0
	1	0 1 0	1 1 0	***	***
	1	0 1 0	1 1 1	***	***
...	

В таблице 4 приведен фрагмент содержимого управляющей памяти для состояния a_4 , содержащего наибольшее число переходов. Очевидна некоторая избыточность в использовании ПЗУ, однако схему устройства в целях снижения стоимости, предполагается реализовать на стандартных микросхемах, в которых разряд-

ность адреса достаточно велика, но разрядность слова небольшая. Формат слова ПЗУ из табл.4 ориентирован на данные условия. Кроме того, такой формат слова позволяет использовать ту же микросхему и для реализации автомата Мили, причем эффективность использования ПЗУ в этом случае повышается, так как вместо повторяющейся информации в поле FУ в каждом слове, описывающем один переход автомата, хранятся разные коды наборов МО, соответствующие состояниям перехода.

Строки переходов для каждого исходного состояния $a_m \in B_1$ адресуются, как показано в столбце "τ" ПСТ (табл.1). Функции $\{\tau_1, \tau_2, \tau_3\} \in \tau$ в виде

$$\tau = f(T, x) \quad (8)$$

и реализуются на ПЛМ с малым числом выходов и промежуточных шин. Например, формула для τ_1 имеет вид:

$$\tau_1 = T_0 \bar{T}_1 T_2 \bar{T}_3 x_4 x_8 + T_0 \bar{T}_1 T_2 T_3 x_4 \bar{x}_8 + T_0 \bar{T}_1 T_2 T_3 x_6 x_7 x_8 \bar{x}_9.$$

Структурная схема устройства, полученного предложенным методом, приведена на рисунке 1.

В схеме ROM1 содержит коды МО и адреса A_1 условных переходов из состояний класса B_1 и срабатывает при $T_0 = 1$; ROM2 функционирует аналогично при $T_0 = 0$ и предназначен для состояний класса B_0 . PLA формирует адрес слова $\{\tau_1, \dots, \tau_k\}$ на странице ROM1. Регистр RG имеет два поля приема информации A_1 и A_0 , и в зависимости от значения признака условия T_0 осуществляет выбор адреса в текущем такте.

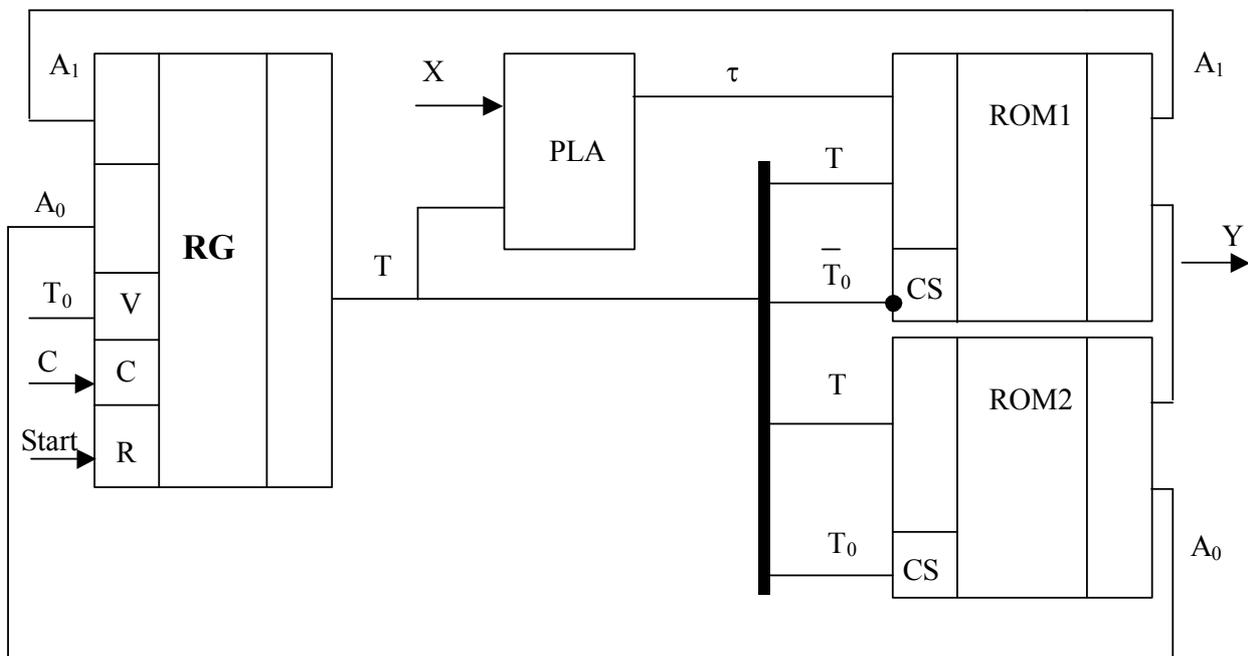


Рисунок 1 — Структурная схема устройства

Заключение. Схема устройства (рис.1), синтезированного с применением предложенного метода, позволяет, как показали исследования, сократить площадь кристалла PLA на 38-43% в сравнении с одноуровневой реализацией. Современные стандартные PROM по своим параметрам позволяют реализовать алгоритмы средней степени сложности, используя лишь по одной микросхеме для классов B_0 и B_1 . Использование в схеме стандартных PROM в качестве управляющей памяти делает данное устройство универсальным и гибким, подобно автоматам с программируемой логикой. Однако данное устройство работает быстрее, т.к. сохраняет основное свойство автоматов с жесткой логикой — выполнение анализа полного набора логических условий за один такт.

Возможна дальнейшая оптимизация данного устройства за счет введения в схему счетчика для формирования адреса слова на странице управляющей памяти.

Литература

1. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. — Донецк: РВА ДонНТУ, 2002. — 262 с.
2. Баркалов А.А., Зеленёва И.Я. Оптимизация способа замены логических условий автомата с двунаправленными переходами // Автоматика и вычислительная техника. — 2000, №5. — С. 58–64.
3. Баркалов А.А., Зеленёва И.Я. Оптимизация логической схемы устройства управления с заменой переменной // Управляющие системы и машины. — 2001, №1. — С. 75–78.
4. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. — НАН Украины, ИК им. В.М. Глушкова. — Киев, 1997.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П., Проектирование систем на микросхемах программируемой логики. — СПб.: БХВ – Петербург, 2002. — 608с.
6. Соловьёв В.В. Проектирование цифровых систем на основе логических интегральных схем. — М.: Горячая линия. — Телеком, 2001. — 636 с.

Сдано в редакцию: 28.03.2003г.

Рекомендовано к печати: д.т.н., проф. Баркалов А.А.