

## ОПТИМИЗАЦИЯ ЛОГИЧЕСКОЙ СХЕМЫ АВТОМАТА МУРА НА FPGA

Баркалов А.А., Красичков А.А., Цололо С.А.

**1. Введение**

Развитие микроэлектроники привело к появлению микросхем типа “система-на-кристалле” (SoC, system-on-a-chip), степень интеграции которых достаточна для реализации сложных цифровых систем [1, 2]. Как правило, в состав SoC входят программируемые вентильные матрицы (FPGA, field-programmable logic arrays) и встроенные блоки памяти (DMB, dedicated memory blocks). При этом FPGA используются для реализации нерегулярной логики, например систем булевых функций, а DMB предназначены для реализации различных таблиц [3]. В большинстве случаев FPGA включают универсальные табличные элементы типа LUT (look-up table) с ограниченным числом входов [1]. Такое ограничение приводит к необходимости функциональной декомпозиции реализуемых функций [4], что связано с увеличением числа уровней схемы, влекущим увеличение времени такта, усложнение задач размещения и трассировки [1]. Для устранения этих негативных явлений необходимо уменьшать число аргументов реализуемых функций. В настоящей работе предлагается метод решения этой задачи при реализации устройства управления, представленного в виде микропрограммного автомата Мура [5].

**2. Общие положения и основная идея метода**

Пусть автомат Мура задан прямой структурной таблицей (ПСТ) со столбцами [5]:  $a_m, K(a_m), a_s, K(a_s), X_h, \Phi_h, h$ . Здесь  $a_m$  – исходное состояние МПА,  $a_m \in A$ , где  $A = \{a_1, \dots, a_M\}$ ;  $K(a_m)$  – код состояния  $a_m \in A$  разрядности  $R = \lceil \log_2 M \rceil$ , для кодирования состояний используются внутренние переменные из множества  $T = \{T_1, \dots, T_R\}$ ;  $a_s, K(a_s)$  – соответственно состояние перехода и его код;  $X_h$  – входной сигнал, определяющий переход  $\langle a_m, a_s \rangle$  и равный конъюнкции некоторых элементов (или их отрицаний) множества логических условий  $X = \{x_1, \dots, x_L\}$ ;  $\Phi_h$  – набор функций возбуждения триггеров памяти МПА, принимающих единичное значение для переключения памяти из  $K(a_m)$  в  $K(a_s)$ ,  $\Phi_h \subseteq \Phi = \{\varphi_1, \dots, \varphi_R\}$ ;  $h = 1, \dots, H$  – номер перехода. В столбце  $a_m$  PST записывается набор микроопераций  $Y_q$ , формируемых в состоянии  $a_m \in A$ , где

$Y_q \subseteq Y = \{y_1, \dots, y_N\}$ ,  $q = 1, \dots, Q$ . Эта таблица является основой для формирования систем функций

$$\Phi = \Phi(T, X), \quad (1)$$

$$Y = Y(T), \quad (2)$$

задающих логическую схему МПА.

При реализации МПА Мура в составе SoC система (1) реализуется комбинационной схемой СС, а система (2) – памятью микроопераций ММО (рис. 1).

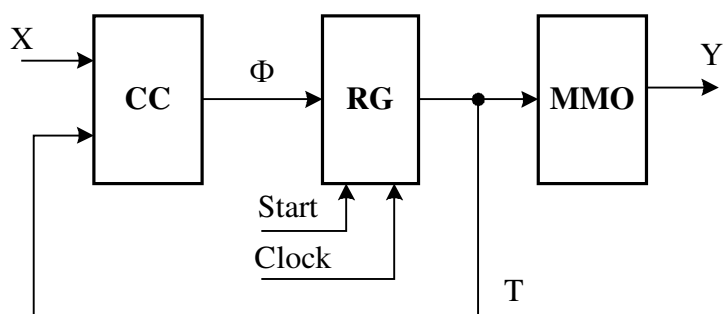


Рисунок 1. – Структурная схема МПА Мура

Здесь схема СС реализуется на FPGA, схема ММО реализуется на DMB, регистр RG используется для хранения кода состояния МПА. Сигнал Start используется для записи в RG кода начального состояния  $a_m \in A$ , синхросигнал Clock разрешает смену состояний МПА. Обозначим эту схему символом  $U_1$ .

Пусть ПСТ формируется на основе граф-схемы алгоритма (ГСА)  $\Gamma$  [5], включающей операторные вершины с наборами  $Y_q \subseteq Y$  и условные вершины с логическими условиями  $x_i \in X$ . Назовем состояния  $a_m, a_s \in A$  псевдоэквивалентными состояниями, если они отмечают операторные вершины, связанные со входом одной и той же вершины ГСА [6].

Наличие псевдоэквивалентных состояний может быть использовано для оптимизации схемы СС [6]. Например, состояния  $a_m \in A$  кодируются так, чтобы каждый класс  $B_i \in \Pi_A$ , где  $\Pi_A = \{B_1, \dots, B_l\}$  – разбиение множества  $A$  на классы псевдоэквивалентных состояний, выражался одной конъюнкцией. Это уменьшает число термов в системе (1). Кроме того, состояния  $a_m \in A$  могут быть закодированы так, чтобы некоторые элементы системы (2) выражались одной конъюнкцией [7]. Это позволяет уменьшить емкость памяти ММО, так как некоторые функции  $y_n \in Y$  реализуются на FPGA. Однако эти методы

кодирования не могут быть использованы одновременно, так как они ориентированы на решение разных задач.

В настоящей работе предлагается подход, позволяющий одновременно решать задачи оптимизации схем СС и ММО. Закодируем классы  $V_i \in \Pi_A$  двоичными кодами  $K(V_i)$  разрядности

$$R_1 = \lceil \log_2 I \rceil \quad (3)$$

и используем переменные из множества  $\tau = \{\tau_1, \dots, \tau_{R_1}\}$  для такого кодирования. Тогда МПА Мура может быть представлен в виде схемы  $U_2$  (рис. 2).

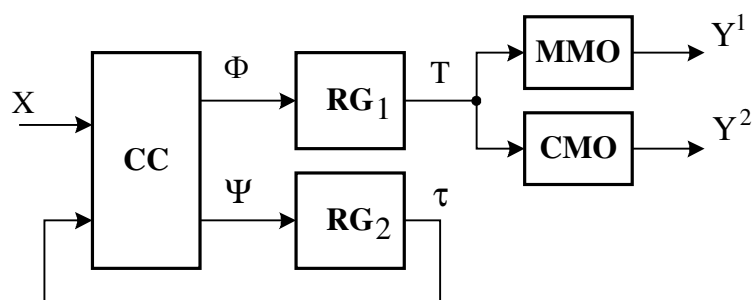


Рисунок 2. – Структурная схема МПА Мура  $U_2$

Здесь схема СС формирует системы функций

$$\Phi = \Phi(\tau, X), \quad (4)$$

$$\Psi = \Psi(\tau, X), \quad (5)$$

формирующие в регистре  $RG_1$  код состояния перехода и в регистре  $RG_2$  код класса разбиения  $\Pi_A$  соответственно. Память ММО используется для хранения микроопераций  $y_n \in Y^1$ , комбинационная схема СМО формирует микрооперации  $y_n \in Y^2$ . Отметим, что  $Y^1 \cap Y^2 = \emptyset$ .

Предлагаемая организация схемы МПА позволяет:

- уменьшить число входных переменных схемы СС от  $R(U_1)$  до  $R_1(U_2)$ ;
- уменьшить емкость ДМВ, используемых для хранения системы микроопераций.

К недостаткам этого метода относится увеличение числа выходных функций, реализуемых схемой СС от  $R(U_1)$  до  $R + R_1(U_2)$ . Однако число аргументов оказывает большое влияние на аппаратные затраты схем на FPGA, чем число реализуемых функций [2].

### 3. Пример применения предложенного метода

В настоящей работе предлагается метод синтеза МПА Мура  $U_2$ , включающий следующие этапы:

- формирование разбиения  $\Pi_A$ ;
- кодирование классов  $B_i \in \Pi_A$ ;
- оптимальное кодирование состояний;
- формирование преобразованной ПСТ;
- формирование системы функций  $Y^2$ ;
- формирование таблицы схемы ММО;
- формирование систем функций  $\Phi$  и  $\Psi$ ;
- реализация схемы МПА в заданном элементарном базисе.

Рассмотрим применение этого метода на примере синтеза МПА  $S_1$ , заданного прямой структурной таблицей (Табл. 1).

Таблица 1. Прямая структурная таблица МПА Мура  $S_1$

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X_h$	$\Phi_h$	$h$
$a_1 (-)$	0000	$a_2$	0001	$x_1 x_2$	$D_4$	1
		$a_3$	0010	$\overline{x_1} \overline{x_2}$	$D_3$	2
		$a_2$	0001	$\overline{x_1} \overline{x_3}$	$D_4$	3
		$a_4$	0011	$\overline{x_1} \overline{x_3}$	$D_3 D_4$	4
$a_2 (y_2 y_5 y_6)$	0001	$a_5$	0100	1	$D_2$	5
$a_3 (y_5 y_7)$	0010	$a_5$	0100	1	$D_2$	6
$a_4 (y_7)$	0011	$a_5$	0100	1	$D_2$	7
$a_5 (y_3 y_6)$	0100	$a_5$	0100	$x_3$	$D_2$	8
		$a_7$	0110	$\overline{x_3} x_4$	$D_2 D_3$	9
		$a_8$	0111	$\overline{x_3} \overline{x_4} x_5$	$D_2 D_3 D_4$	10
		$a_6$	0101	$\overline{x_3} \overline{x_4} \overline{x_5}$	$D_2 D_4$	11
$a_6 (y_4 y_5)$	0101	$a_9$	1000	1	$D_1$	12
$a_7 (y_1 y_3)$	0110	$a_9$	1000	1	$D_1$	13
$a_8 (y_1 y_3 y_8)$	0111	$a_9$	1000	1	$D_1$	14
$a_9 (y_1 y_2 y_4)$	1000	$a_1$	0000	1	–	15

Состояния автомата  $S_1$  закодированы произвольным образом, при этом  $R = 4$ , а регистр  $RG$  имеет информационные входы  $D$ -типа.

Формирование разбиения  $\Pi_A$ . Основываясь на определении псевдоэквивалентных состояний, на базе анализа табл. 1 формируется разбиение  $\Pi_A = \{B_1, \dots, B_5\}$ , где  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2, a_3, a_4\}$ ,  $B_3 = \{a_5\}$ ,  $B_4 = \{a_6, a_7, a_8\}$ ,  $B_5 = \{a_9\}$ . Таким образом,  $I = 5$ .

Кодирование классов. Согласно (3), для автомата  $S_1$  параметр  $R_1 = 3$ . Следовательно  $\tau = \{\tau_1, \tau_2, \tau_3\}$ . Закодируем классы  $B_i \in \Pi_A$  произвольным образом:  $K(B_1) = 000$ ,  $K(B_2) = 001$ , ...,  $K(B_5) = 100$ .

Оптимальное кодирование состояний. Закодируем состояния  $a_m \in A$  автомата  $S_1$  так, чтобы максимальное число микроопераций  $y_n \in Y$  представлялись одной конъюнкцией. Для этой цели можно использовать, например, известный алгоритм ESPRESSO [7]. Один из вариантов кодирования представлен картой Карно (рис. 3).

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	$a_1$	$a_4$	$a_5$	$a_7$
	01	$a_2$	$a_3$	*	$a_8$
	11	*	$a_6$	*	$a_9$
	10	*	*	*	*

Рисунок 3. – Оптимальное кодирование состояний

Формирование преобразованной ПСТ. Преобразование исходной ПСТ выполняется следующим образом:

- столбец  $a_m$  заменяется столбцом  $B_i$ ;
- столбец  $K(a_m)$  заменяется столбцом  $K(B_i)$ ;
- если  $a_m \in B_i$ , то в столбце  $B_i$  состояние  $a_m$  заменяется классом  $B_i$ , а в столбце  $K(B_i)$  код  $K(a_m)$  заменяется кодом этого класса;
- если в столбце  $B_i$  есть строки с одинаковыми классами, то остается только одна из них;
- в ПСТ вводятся столбцы  $B_h, K(B_h), \Psi_h$ , при этом в  $h$ -й строке ПСТ записывается класс  $B_j$  такой, что  $a_s \in B_j$ , столбец  $\Psi_h$  содержит функции возбуждения триггеров регистра  $RG_2$ .

В рассматриваемом примере из исходной ПСТ удаляются строки 6, 7 (соответствуют  $a_m \in B_2$ ), строки 13, 14 (соответствуют  $a_m \in B_4$ ), столбец  $\Psi_h$

содержит функции  $D_5, D_6, D_7 \in \Psi$ . Таким образом, преобразованная ПСТ содержит  $H_0 = 11$  строк (табл. 2). В преобразованной ПСТ коды состояний  $a_m \in A$  берутся из карты Карно (рис. 3).

Таблица 2. Преобразованная ПСТ автомата Мура  $S_1$

$B_i$	$K(B_i)$	$a_s$	$K(a_s)$	$B_h$	$K(B_h)$	$X_h$	$\Phi_h$	$\Psi_h$	$h$
$B_1$	000	$a_2$	0100	$B_2$	001	$x_1 x_2$	$D_2$	$D_7$	1
		$a_3$	0101	$B_2$	001	$x_1 \overline{x_2}$	$D_2 D_4$	$D_7$	2
		$a_2$	0100	$B_2$	001	$\overline{x_1} x_3$	$D_2$	$D_7$	3
		$a_4$	0001	$B_2$	001	$\overline{x_1} \overline{x_3}$	$D_4$	$D_7$	4
$B_2$	001	$a_5$	0011	$B_3$	010	1	$D_3 D_4$	$D_6$	5
$B_3$	010	$a_5$	0011	$B_3$	010	$x_3$	$D_3 D_4$	$D_6$	6
		$a_7$	0010	$B_4$	011	$\overline{x_3} x_4$	$D_3$	$D_6 D_7$	7
		$a_8$	0110	$B_4$	011	$\overline{x_3} \overline{x_4} x_5$	$D_2 D_3$	$D_6 D_7$	8
		$a_6$	1101	$B_4$	011	$\overline{x_3} \overline{x_4} \overline{x_5}$	$D_1 D_2 D_4$	$D_6 D_7$	9
$B_4$	011	$a_9$	1110	$B_5$	100	1	$D_1 D_2 D_3$	$D_5$	10
$B_5$	100	$a_1$	0000	$B_1$	000	1	–	–	11

Формирование системы функций  $Y^2$ . Найдем дизъюнктивные нормальные формы (ДНФ) микроопераций  $y_n \in Y$ . Из Рис. 3 имеем с учетом неопределенностей:  $y_1 = T_3 \overline{T_4}$ ;  $y_2 = T_2 \overline{T_3} \overline{T_4} \vee T_1 T_3$ ;  $y_3 = \overline{T_1} T_3$ ;  $y_4 = T_1$ ;  $y_5 = T_2 \overline{T_3}$ ;  $y_6 = T_2 \overline{T_3} \overline{T_4} \vee T_3 T_4$ ;  $y_7 = \overline{T_1} T_3 T_4$ ;  $y_8 = \overline{T_1} T_2 T_3$ .

Для выбора функций  $y_n \in Y^2$  могут быть предложены следующие критерии:

- $y_n \in Y^2$ , если ДНФ  $y_n \in Y$  включает только один терм;
- $y_n \in Y^2$ , если ДНФ  $y_n \in Y$  реализуется на одном LUT-элементе;
- $y_n \in Y^2$ , если для формирования функции  $y_n \in Y$  требуется не больше времени, чем время выборки из DMB.

Используем первый критерий и получим  $Y^2 = \{y_1, y_3, y_4, y_5, y_7, y_8\}$ , следовательно,  $Y^1 = \{y_2, y_6\}$ .

Формирование таблицы схемы ММО. Эта таблица имеет входы  $T$ , выходы  $y_n \in Y^1$  и строится тривиальным образом.

Формирование систем функций  $\Phi$  и  $\Psi$ . Системы (4) и (5) зависят от термов

$$F_h = \left( \bigwedge_{r=1}^{R_1} \tau_r^{l_{hr}} \right) X_h \quad (h = 1, \dots, H_0), \quad (6)$$

где  $l_{hr} \in \{0, 1\}$  – значение  $r$ -го разряда кода  $K(B_i)$  из  $h$ -ой строки преобразованной ПСТ,  $\tau_r^0 = \overline{\tau_r}$ ,  $\tau_r^1 = \tau_r$  ( $r = 1, \dots, R_1$ ). При этом функции  $D_r \in \Phi \cup \Psi$  определяются в виде:

$$D_r = \bigvee_{h=1}^{H_0} C_{rh} F_h \quad (r = 1, \dots, R + R_1), \quad (7)$$

где  $C_{rh}$  – булевская переменная, равная единице, если и только если в  $h$ -й строке ПСТ записана переменная  $D_r$  ( $h = 1, \dots, H_0$ ;  $r = 1, \dots, R + R_1$ ).

Например, из табл. 2 имеем  $D_1 = F_9 \vee F_{10} = \overline{\tau_1} \tau_2 \tau_3 x_3 x_4 x_5 \vee \tau_1 \tau_2 \tau_3$ .

Реализация схемы МПА  $U_2$ . Выполнение этого этапа сводится к реализации систем (4), (5),  $Y^2$  на FPGA и к реализации таблицы схемы ММО на DMB. Методы решения этих задач достаточно полно рассмотрены в литературе [1, 4] и выходят за рамки данной статьи.

#### 4. Заключение

Предлагаемый в работе метод позволяет уменьшать число аргументов в системе функций возбуждения памяти микропрограммного автомата Мура. В основе оптимизации находится кодирование классов псевдоэквивалентных состояний и уменьшение числа LUT-элементов возможно при выполнении условия

$$R_1 < R. \quad (8)$$

Так как результат оптимизации схемы формирования функций возбуждения не зависит от способа кодирования состояний, то состояния могут быть закодированы так, чтобы уменьшить число встроенных блоков памяти DMB, необходимых для реализации системы микроопераций. При этом часть микроопераций реализуется на LUT-элементах. Задача разбиения множества микроопераций на классы, реализуемые на DMB и на LUT-элементах, остается открытой. В качестве критерия сравнения результатов синтеза автоматов  $U_1$  и  $U_2$  авторы использовали число эквивалентных вентилях, требуемых для реализации схем автоматов. Результаты исследований показали, что при выполнении условия (8) и реализации на LUT-элементах микроопераций, ДНФ которых выражаются одним термом, аппаратные затраты в схеме  $U_2$  на 15-22% меньше, чем в схеме эквивалентного автомата  $U_1$ .

## Литература

1. Maxfield C. The Design Warriors Guide to FPGAs. – Elseveir, 2004. – 541p.
2. Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608с.
3. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
4. T. Sasao. Switching Theory for Logic Synthesis. – Kluwer Academic Publishers, 1999. – 362p.
5. Baranov S. Logic Synthesis of Control Automata. – Kluwer Academic Publishers, 1994. – 312p.
6. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура. // Кибернетика и системный анализ. – 1998, №1. – с. 65-72.
7. DeMicheli G. Synthesis and Optimization of Digital Circuits. – McGraw-Hill, 1994. – 636p.