

Оптимизация схемы МПА Мура в составе системы на кристалле

В.А. Саломатин, С.А. Цололо

Донецкий национальный технический университет
s.solos@gmail.com

Abstract

Salomatin V.A., Tsololo S.A. Optimization of Moore FSM as a part of “system-on-chip”. Method of Moore’s circuit optimization is proposed. Method based on features of CPLD architecture and Moore’s FSM model. The carried out researches have shown that the method reduces hardware expenses up to 30%.

1. Введение

Важным блоком любой цифровой системы является устройство управления (УУ), которое координирует взаимодействие всех блоков системы [1, 2]. На практике УУ часто реализуется с использованием модели микропрограммного автомата (МПА) Мура [3]. В настоящее время прогресс в области микроэлектроники привел к появлению «систем-на-кристалле» (SoC, system-on-chip) [4], функциональные возможности которых достаточны для реализации сложной цифровой системе на одном кристалле [5]. В SoC произвольная логика может реализовываться с использованием макроячеек PAL (programmable array logic), а табличные функции реализуются с помощью блоков памяти (EMB, embedded memory blocks) [6]. Одной из актуальных задач в этом случае является уменьшение аппаратных затрат в схеме МПА [1]. Решение этой задачи позволяет уменьшить площадь кристалла, занимаемую схемой УУ, при этом возможно увеличение функциональных возможностей системы в рамках одного кристалла [6]. Для решения этой задачи необходимо учитывать как особенности элементного базиса, так и особенности модели МПА. Особенности PAL является большой коэффициент объединения по входу, который достигает нескольких десятков в реальных CPLD (complex programmable logic devices) [7], и ограниченное число термов (элементарных конъюнкций) в одной макроячейке (порядка восьми) [1]. Особенности МПА Мура является наличие псевдоэквивалентных состояний [8] и регулярный характер системы микроопераций, что позволяет реализовать ее на EMB [6]. Целью исследований,

представленных в этой работе, является возможность оптимизации комбинационной схемы автомата Мура за счет использования нескольких источников кода текущего состояния автомата, что возможно благодаря особенностям PAL. Задачей, решаемой в работе, является разработка формализованного метода синтеза микропрограммного автомата Мура, позволяющего оптимизировать число макроячеек PAL в схеме формирования функций возбуждения триггеров памяти автомата. При этом алгоритм управления цифровой системы задан в виде граф-схемы алгоритма (ГСА) [3].

2. Особенности реализации автомата Мура

Пусть алгоритм управления цифровой системы представлен ГСА $\Gamma = \Gamma(V, E)$, где $V = \{b_0, b_E\} \cup E_1 \cup E_2$ — множество вершин, $E = \{ \langle b_q, b_t \rangle \mid b_q, b_t \in V \}$ — множество дуг. Здесь b_0 — начальная вершина ГСА, b_E — конечная вершина ГСА, E_1 — множество операторных вершин, E_2 — множество условных вершин. В вершинах $b_q \in E_1$ записываются наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ — множество микроопераций операционного автомата цифровой системы [9]. В вершинах $b_q \in E_2$ записываются элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Начальная и конечная вершины ГСА соответствуют состоянию $a_1 \in A = \{a_1, \dots, a_M\}$, где A — множество состояний автомата Мура, а каждая вершина $b_q \in E_1$ соответствует одному из элементов множества A [3]. Логическая схема МПА Мура задается системой уравнений

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

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

где $T = \{T_1, \dots, T_R\}$ — множество внутренних переменных, кодирующих состояния $a_m \in A$, $R = \lceil \log_2 M \rceil$; $\Phi = \{D_1, \dots, D_R\}$ — множество функций возбуждения памяти состояний. Системы (1)-(2) формируются на основе прямой структурной таблицы (ПСТ) со столбцами: a_m — текущее состояние; $K(a_m)$ — код состояния $a_m \in A$; a_s — состояние перехода; $K(a_s)$ — код состояния $a_s \in A$; X_h — конъюнкция некоторых элементов множества X (или их отрицаний), определяющая переход $\langle a_m, a_s \rangle$; Φ_h — набор функций возбуждения памяти МПА, принимающих единичное состояние для переключения памяти из $K(a_m)$ в $K(a_s)$; $h = 1, \dots, N_1(\Gamma)$ — номер строки таблицы. В столбце a_m записывается набор микроопераций

$Y(a_m) \subseteq Y$, формируемых в состоянии $a_m \in A$. Естественно, что $Y(a_m) = Y(b_q)$, где вершина $b_q \in E_1$ отмечена состоянием $a_m \in A$.

Как правило, число переходов $N_1(\Gamma)$ больше числа переходов $N_2(\Gamma)$ эквивалентного автомата Мили [3]. Это приводит к увеличению числа PAL в схеме МПА Мура по сравнению с этим показателем эквивалентного автомата Мили. Параметр $N_1(\Gamma)$ можно уменьшить, благодаря наличию псевдоэквивалентных состояний (ПЭС) МПА Мура [10]. Состояния $a_m, a_s \in A$ называется ПЭС, если выходы соответствующих им вершин соединены с выходом одной и той же вершины ГСА Γ . Пусть $\Pi_A = \{B_1, \dots, B_I\}$ – разбиение множества A на классы ПЭС ($I \leq M$). Поставим в соответствие классу $B_i \in \Pi_A$ двоичный код $K(B_i)$ разрядности $R_1 = \lceil \log_2 I \rceil$ и используем переменные $\tau_i \in \tau$ для такого кодирования, где $|\tau| = R_1$. В этом случае МПА Мура представляется в виде структуры U_1 (рис. 1)

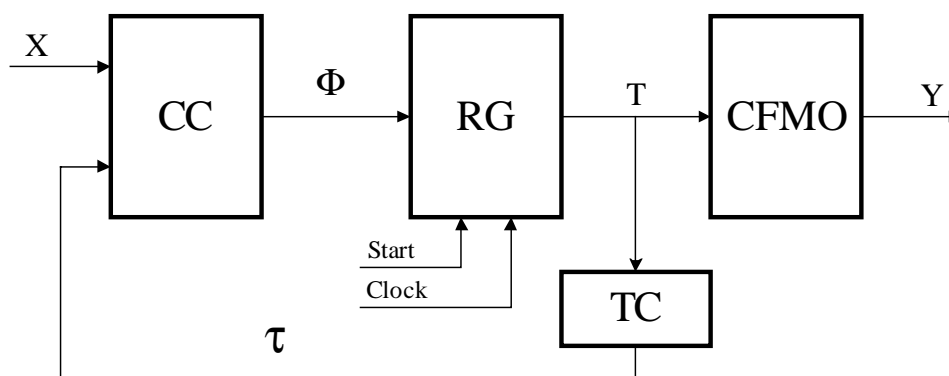


Рисунок 1 – Структурная схема МПА Мура U_1

В МПА U_1 схема CC формирует функции

$$\Phi = \Phi(\tau, X), \quad (3)$$

а схема формирования микроопераций $CFMO$ реализует систему (2). Регистр RG представляет собой память состояний, по сигналу $Start$ в RG заносится нулевой код начального состояния $a_1 \in A$, по сигналу $Clock$ происходит смена кодов в регистре. Преобразователь кодов состояний TC реализует систему функций

$$\tau = \tau(T), \quad (4)$$

при этом код $K(B_i)$ формируется на основе кода $K(a_m)$, где $a_m \in B_i$.

В [10] показано, что для МПА U_1 число переходов уменьшается до $N_2(\Gamma)$. Недостатком МПА U_1 является наличие схемы TC , которая требует дополнительных ресурсов. Отметим, что схема CC реализуется на PAL, а

схемы ТС и CFMO – на блоках памяти EMB. В настоящей работе предлагается метод синтеза МПА Мура, позволяющий уменьшить аппаратные затраты в схеме ТС (при определенных условиях этот блок может не использоваться вообще). Предлагаемый метод основан на следующих особенностях SoC, основанных на технологии CPLD [1, 7]:

- коэффициент объединения по входу макроячейки PAL значительно превосходит максимально возможное число букв в термах системы (1), определяемое $L + R$;
- число выходов EMB может меняться в некотором диапазоне (как правило, 1, 2, 4, 8).

3. Основная идея предлагаемого метода

Используем идею оптимального кодирования состояний МПА Мура [10], смысл которой заключается в таком кодировании ПЭС, чтобы максимально возможное число классов $V_i \in \Pi_A$ соответствовали одному обобщенному интервалу R -мерного булева пространства. Представим множество Π_A в виде $\Pi_A = \Pi_B \cup \Pi_C$, где $V_i \in \Pi_B$, если

$$|V_i| > 1, \quad (5)$$

и $V_i \in \Pi_C$ в противном случае. Очевидно, что схема ТС должна формировать только коды классов $V_i \in \Pi_B$. Закодируем состояния $a_m \in A$ оптимальным образом [10]. Представим множество Π_B в виде $\Pi_B = \Pi_D \cup \Pi_E$, где $V_i \in \Pi_D$, если коды $a_m \in B$ входят один обобщенный интервал пространства кодирования. Теперь преобразованию подлежат только коды состояний $a_m \in A(\Pi_E)$, где $A(\Pi_E) \subseteq A$ – множество состояний, входящих в классы Π_E . Для кодирования классов $V_i \in \Pi_E$ достаточно

$$R_2 = \lceil \log_2(|\Pi_E| + 1) \rceil \quad (6)$$

переменных, образующих множество Z , где $|Z| = R_2$.

Пусть t_F – фиксированное число выходов блока EMB и пусть q – число слов в блоке при $t_F = 1$. При реализации схемы CFMO автомата U_1 параметр t_F определяется следующим образом:

$$t_F = \lceil q / M \rceil. \quad (7)$$

При этом интегрально блоки EMB схемы CFMO имеют

$$t_S = \lceil N / t_F \rceil \cdot t_F. \quad (8)$$

выходов. Очевидно, что Δ_t выходов могут не использоваться для представления микроопераций, где

$$\Delta_t = t_s - N. \quad (9)$$

Эти выходы можно использовать для представления переменных $z_r \in Z$. Рассмотрим случай, когда выполняется условие

$$\Delta_t \leq R_2. \quad (10)$$

В этом случае множество Π_E необходимо представить в виде $\Pi_E = \Pi_F \cup \Pi_G$. Множество Π_F включает n_F классов, где

$$n_F = 2^{\Delta_t} - 1, \quad (11)$$

коды которых хранятся вместе с микрооперациями и представляются переменными $z_r \in Z$, где $|Z| = \Delta_t$. Множество Π_G включает

$$n_G = I - n_C - n_D - n_F \quad (12)$$

классов, где $n_C = |\Pi_C|$, $n_D = |\Pi_D|$. Для кодирования классов $V_i \in \Pi_G$ достаточно

$$R_3 = \lceil \log_2(n_G + 1) \rceil \quad (13)$$

переменных, образующих множество τ , где $|\tau| = R_3$.

В этом случае для интерпретации ГСА Γ предлагается автомат Мура U_2 (рис. 2)

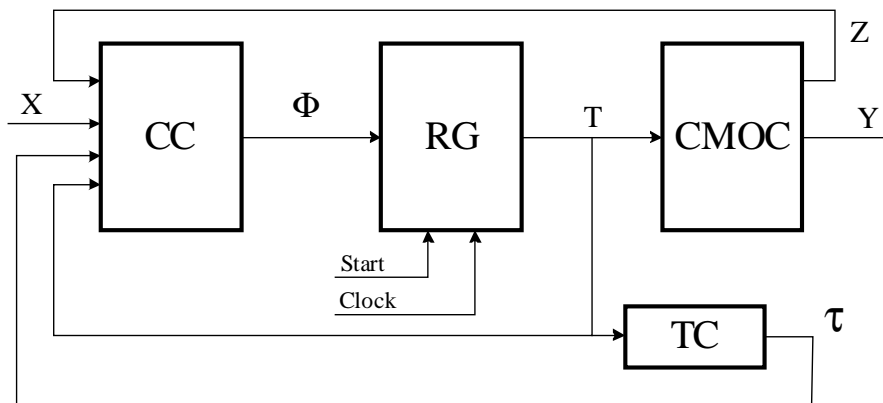


Рисунок 2 – Структурная схема МПА Мура U_2

Автомат U_2 имеет ряд отличий от автомата U_1 :

- схема CC формирует систему функций

$$\Phi = \Phi(T, \tau, Z, X); \quad (14)$$

- вместо схемы CFMO используется схема CMOC, реализующая систему функций (2) и систему функций

$$Z = Z(T) \quad (15)$$

для представления кодов классов $V_i \in \Pi_F$;

- преобразователь кодов TC формирует коды классов $V_i \in \Pi_G$;

- переменные $T_r \in T$ представляют состояния $a_m \in A(\Pi_C)$ и классы $V_i \in \Pi_D$, где $A(\Pi_C)$ – множество состояний, входящих в классы $V_i \in \Pi_C$.

При этом число входов в макроячейках PAL схемы СС увеличивается от $L + R_1$ (автомат U_1) до $L + R + \Delta_t + R_3$ (автомат U_2). Однако это не приводит к росту аппаратурных затрат, так как PAL имеют коэффициент объединения по входу порядка нескольких десятков [7]. Времена циклов автоматов U_1 и U_2 совпадают, причем это время равно времени цикла автомата Мура U_0 , который реализуется по системам (1)-(2). Таким образом, предлагаемый метод позволяет сохранить быстродействие цифровой системы при уменьшении аппаратурных затрат.

Предлагаемый метод синтеза схемы МПА U_2 включает следующие этапы:

1. Формирование отмеченной ГСА Γ .
2. Формирование разбиения $\Pi_A = \Pi_B \cup \Pi_C$.
3. Оптимальное кодирование состояний и формирование множеств Π_D и Π_E .
4. Определение параметра Δ_t и формирование множеств Π_F и Π_G .
5. Кодирование классов $V_i \in \Pi_F \cup \Pi_G$.
6. Формирование таблицы схемы СМОС.
7. Формирование модифицированной ПСТ автомата U_2 .
8. Формирование таблицы схемы ТС.
9. Реализация схемы автомата в заданном элементном базисе.

4. Пример применения предложенного метода

Пусть для некоторой ГСА Γ_1 множество $A = \{a_1, \dots, a_{16}\}$, $\Pi_A = \{V_1, \dots, V_7\}$, где $V_1 = \{a_1\}$, $V_2 = \{a_2, a_6\}$, $V_3 = \{a_3, a_5, a_7\}$, $V_4 = \{a_4, a_9, a_{13}\}$, $V_5 = \{a_8\}$, $V_6 = \{a_{10}, a_{15}, a_{16}\}$, $V_7 = \{a_{11}, a_{12}, a_{14}\}$. Таким образом, $\Pi_B = \{V_2, V_3, V_4, V_6, V_7\}$ и $\Pi_C = \{V_1, V_5\}$. Закодируем состояния $a_m \in A$ оптимальным образом [10], используя булево пространство размерности $R = 4$ (рис. 3).

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	a_1	a_2	a_6	a_4
	01	a_3	a_5	a_7	a_9
	11	a_{11}	a_{12}	a_{14}	a_{13}
	10	a_8	a_{10}	a_{15}	a_{16}

Рисунок 3 – Оптимальное кодирование состояний автомата $U_2(\Gamma_1)$

Здесь символ $U_i(\Gamma_j)$ означает, что автомат U_i интерпретирует ГСА Γ_j . Из рис. 3 следует, что $\Pi_D = \{B_2\}$ и $\Pi_E = \{B_3, B_4, B_6, B_7\}$, при этом $K(B_2) = 00*1$, а коды классов $B_i \in \Pi_C$ совпадают с кодами входящих в них состояний: $K(B_1) = 0000$, $K(B_5) = 1000$.

Из (6) следует, что $Y(a_{14}) = Y(a_{15}) = \{y_6, y_{10}, y_8, y_{15}\}$, то есть для представления кодов классов $B_i \in \Pi_E$ необходимы 3 свободных разряда СМОС. Пусть для ГСА Γ_1 $N = 15$ и пусть для реализации схемы СМОС используются блоки ЕМВ с $t_F = 4$ при $q = 16$ ($q = 2^R$). В этом случае $t_s = 4 \cdot 4 = 16$ и $\Delta_t = 1$. Таким образом, условие (10) выполняется и преобразователь ТС необходим. Пусть $\Pi_F = \{B_3\}$, $\Pi_G = \{B_4, B_6, B_7\}$, тогда $R_3 = 2$. Итак, для автомата $U_2(\Gamma_1)$ имеем: $\tau = \{\tau_1, \tau_2\}$, $K(B_i)$. Закодируем классы $B_i \in \Pi_E$ следующим образом: $K(B_3) = K(B_4) = 01$, $K(B_6) = 10$, $K(B_7) = 11$. При этом комбинация $z_1 = \tau_1 = \tau_2 = 0$ означает, что источником кода класса $B_i \in \Pi_A$ является регистр RG.

Пусть переходы между состояниями МПА $U_2(\Gamma_1)$ заданы следующей системой обобщенных формул перехода [6]:

$$\begin{aligned}
 B_1 &\rightarrow a_2; B_2 \rightarrow x_1 a_4 \vee \bar{x}_1 a_6; \\
 B_3 &\rightarrow x_2 a_{10} \vee \bar{x}_2 x_3 a_{11} \vee \bar{x}_2 \bar{x}_3 a_{14}; \\
 B_4 &\rightarrow x_3 a_8 \vee \bar{x}_3 x_4 a_3 \vee \bar{x}_3 \bar{x}_4 a_5; \\
 B_5 &\rightarrow a_5; B_6 \rightarrow x_3 a_7 \vee \bar{x}_3 a_{13}; \\
 B_7 &\rightarrow x_1 a_1 \vee \bar{x}_1 x_2 a_9 \vee \bar{x}_1 \bar{x}_2 x_4 a_{15} \vee \\
 &\quad \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 x_5 a_{12} \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 \bar{x}_5 a_{16}.
 \end{aligned} \tag{16}$$

Пусть микрооперации $y_n \in Y$ распределяются по состояниям автомата $U_2(\Gamma_1)$ следующим образом: $Y(a_1) = \emptyset$, $Y(a_2) = Y(a_3) = Y(a_{16}) = \{y_1, y_3\}$, $Y(a_4) = \{y_2, y_4, y_6\}$, $Y(a_5) = \{y_5, y_6\}$, $Y(a_6) = Y(a_9) = \{y_1, y_7, y_8\}$, $Y(a_7) = \{y_1, y_7, y_8, y_{15}\}$, $Y(a_8) = \{y_3, y_{10}, y_{14}\}$, $Y(a_{10}) = Y(a_{12}) = \{y_9, y_{11}\}$,

$$Y(a_{11}) = Y(a_{13}) = \{y_{11}, y_{12}, y_{13}\}, \quad Y(a_{14}) = Y(a_{15}) = \{y_6, y_{10}, y_8, y_{15}\}, \\ Y(a_{15}) = \{y_5\}.$$

Таблица схемы СМОС включает столбцы a_m , $K(a_m)$, $Y(a_m)$, $K(B_i)$, m , где $K(a_m)$ – является адресом слова ЕМВ. Для МПА $U_2(\Gamma_1)$ эта таблица имеет $M = 16$ строк, которые отображены на рис. 4.

$T_3 T_4$		$T_1 T_2$		00	01	11	10
		00	01	11	10		
$T_1 T_2$	00	–	$Y_1 Y_3$	$Y_1 Y_7 Y_8$	$Y_2 Y_4 Y_6$		
	01	$Y_1 Y_3 Z_1$	$Y_5 Y_6 Z_1$	$Y_1 Y_7 Y_8 Y_{15} Z_1$	$Y_1 Y_7 Y_8$		
	11	$Y_{11} Y_{12} Y_{13}$	$Y_9 Y_{11}$	$Y_6 Y_8 Y_{10} Y_{15}$	$Y_{11} Y_{12} Y_{13}$		
	10	$Y_3 Y_{10} Y_{14}$	$Y_9 Y_{11}$	$Y_6 Y_{10} Y_8 Y_{15}$	Y_5		

Рисунок 4 – Содержимое схемы СМОС автомата $U_2(\Gamma_1)$

Как видно из рис. 4, переменная z_1 добавлена к наборам микроопераций для состояний $a_m \in B_3$.

Модифицированная ПСТ (МПСТ) автомата U_2 включает столбцы: B_i , $K(B_i)$, a_s , $K(a_s)$, X_h , Φ_h , h , где код $K(B_i) = \langle \tau, Z, T \rangle$. Для автомата $U_2(\Gamma_1)$ эта таблица включает $N_2(\Gamma_1) = 17$ строк, что определяется суммарным числом термов системы (16). Переходы для классов $B_1 - B_4$ показаны в табл. 1

Таблица 1.

Фрагмент МПСТ автомата Мура $U_2(\Gamma_1)$

B_i	$K(B_i)$							a_s	$K(a_s)$	X_h	Φ_h	h
	τ_1	τ_2	z_1	T_1	T_2	T_3	T_4					
B_1	0	0	0	0	0	0	0	a_2	0001	1	D_4	1
B_2	0	0	0	0	0	*	1	a_4	0010	x_1	D_3	2
								a_6	0011	\bar{x}_1	$D_3 D_4$	3
B_3	0	0	1	*	*	*	*	a_{10}	1001	x_2	$D_1 D_4$	4
								a_{11}	1100	$\bar{x}_2 x_3$	$D_1 D_2$	5
								a_{14}	1111	$\bar{x}_2 \bar{x}_3$	$D_1 D_2 D_3 D_4$	6
B_4	0	1	0	*	*	*	*	a_8	1000	x_3	D_1	7
								a_3	0100	$\bar{x}_3 x_4$	D_2	8
								a_5	0101	$\bar{x}_3 \bar{x}_4$	D_2, D_4	9

Эта таблица является основой для формирования системы (14). Например, из табл.1 имеем фрагмент ДНФ функции D_4 :

$$D_4 = \bar{\tau}_1 \bar{\tau}_2 \bar{z}_1 \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \vee \bar{\tau}_1 \bar{\tau}_2 \bar{z}_1 \bar{T}_1 \bar{T}_2 \bar{T}_4 \bar{x}_1 \vee \bar{\tau}_1 \bar{\tau}_2 z_1 x_2 \vee \bar{\tau}_1 \bar{\tau}_2 z_1 \bar{x}_2 \bar{x}_3 \vee \bar{\tau}_1 \bar{\tau}_2 \bar{z}_1 x_4.$$

Отметим, что для автомата Мура $U_0(\Gamma_1)$, в котором нет оптимального кодирования состояний, число термов в системе (1) составляет $H_0(\Gamma_1) = 43$. Таким образом, для рассматриваемого примера число термов в системе функций возбуждения памяти уменьшается в $\eta \approx 2.5$ раза.

Таблица преобразователя ТС включает столбцы a_m , $K(a_m)$, $Y(a_m)$, $K(B_i)$, B_i , τ_m , m . В случае МПА $U_2(\Gamma_1)$ эта таблица имеет 9 строк, так как в классы $B_i \in \Pi_G$ входят девять состояний. Однако из рис 3. следует, например, что коды $a_3, a_5 \in B_3$ входят в интервал 010^* . С учетом этого таблица ТС МПА $U_2(\Gamma_1)$ имеет 6 строк (табл. 2).

Эта таблица является основой для формирования системы (4). Так, из табл. 2 имеем, например, $\tau_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \vee T_1 T_3 T_4 \vee T_1 \bar{T}_2 T_4$ (второй терм соответствует строкам 4 и 6 табл. 2)

Реализация схемы МПА U_2 сводится к реализации систем (4) и (14) на PAL и систем (2) и (15) на EMB. Эти задачи достаточно рассмотрены в литературе [1,7] и выходят за рамки нашей статьи.

Таблица 2.

Таблица преобразователя кодов МПА $U_2(\Gamma_1)$

a_m	$K(a_m)$	B_i	$K(B_i)$	τ_m	m
a_3, a_5	010*	B_3	01	τ_2	1
a_7	0111	B_3	01	τ_2	2
a_4, a_9	0*10	B_4	10	τ_1	3
a_{13}	1110	B_4	10	τ_1	4
a_{10}, a_{15}	10*1	B_6	11	τ_1, τ_2	5
a_{16}	1010	B_6	11	τ_1, τ_2	6

5. Исследование эффективности предложенного метода

Для исследования эффективности предложенных методов в данной работе использованы следующие принципы:

- основанный на вероятностном подходе переход от отдельных ГСА к классам ГСА, предложенным Г.И. Новиковым [11] и развитый в работе [8];

- переход от схемы на макроячейках PAL и блоках EMB к матричным схемам, предложенным С.И. Барановым [3];
- переход от абсолютных оценок аппаратных затрат к относительным оценкам для разных структур схемы автомата, позволяющий оценить не отдельные точки, а общие тенденции в пространстве решений.

На рис. 4 представлена матричная реализация МПА U_1 , где символ «&» означает конъюнктивную матрицу, а символ «v» – дизъюнктивную матрицу.

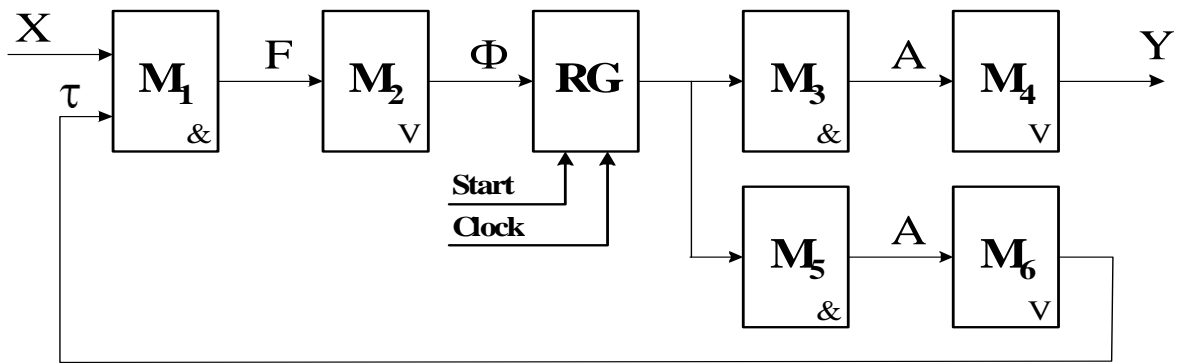


Рисунок 4 – Матричная реализация МПА Мура U_1

Матрицы M_1 и M_2 образуют схему СС, в которой число термов и переменных обратной связи совпадают с характеристиками эквивалентного автомата Мили. Матрицы M_3 и M_4 образуют схему CFMO, выходы схемы M_3 соответствуют конъюнкциям A_m ($m=1, \dots, M$), соответствующих кодам состояний автомата Мура; матрицы M_5 и M_6 образуют схему ТС. Площади $S(M_i)$ матриц M_i ($i=1, \dots, 6$) могут быть определены следующим образом:

$$\begin{aligned}
 S(M_1) &= 2(L + R_1)H_0; & S(M_2) &= H_0R; \\
 S(M_3) &= S(M_5) = 2R \cdot 2^R; & S(M_4) &= 2^R \cdot N; \\
 S(M_6) &= 2^R \cdot R_1.
 \end{aligned}
 \tag{17}$$

В нашем случае эти величины определяются в условных единицах площади.

В работе [11] предложено оценивать классы ГСА параметром P_1 , равным доле операторных вершин ГСА. Используя этот параметр и результаты работы [8] можно найти следующие выражения для аргументов систем (17):

$$\begin{aligned}
L &\approx 0.75 \cdot (1 - P_1)K; \\
R_1 &= \lceil \log_2(3.55 + 0.3 \cdot P_1 \cdot K) \rceil; \\
H_0 &= 4.44 + P_1 \cdot K; \\
R &= \lceil \log_2 P_1 \cdot K \rceil,
\end{aligned} \tag{18}$$

где K – число вершин ГСА. Введем коэффициент k_p , определяющий отношение одной условной единицы площади макроячейки PAL к соответствующему параметру ЕМВ. Теперь площадь матричной реализации схемы МПА U_1 может быть выражена формулой

$$S(U_1) = k_p H_0 (2L + 2R_1 + R) + 2^R (2R + N) + 2^R (2R + R_1) \tag{19}$$

С учетом формулы (17) выражение $S(U_1)$ может быть представлено как функции от аргументов K, P_1, k_p, N .

Аналогичным образом может быть построены матричные схема МПА U_2 . В силу большого коэффициента объединения PAL по входу будем считать, что наличие двух источников кода исходного состояния $a_m \in A$ не влияет на величину площади, занимаемой схемой СС, и эти площади для автоматов U_1 и U_2 одинаковы. Так как переменные $z_r \in Z$ формируются на свободных выходах схемы СМОС, то площади схем СМОС и СФМО также будем считать совпадающими. Кроме того, так как часть состояний преобразовывается на схеме формирования микроопераций, то емкость схемы ТС в автомате U_2 будет меньшей, чем в U_1 . Для проведения исследований введем коэффициент $k_{ТС}$, который характеризует уменьшение емкости схемы ТС благодаря применению предложенного метода, $k_{ТС} \in \{0.2; 0.5; 0.7\}$. Таким образом, площадь матричной реализации схемы МПА U_2 может быть выражена формулой

$$S(U_2) = k_p H_0 (2L + 2R_1 + R) + 2^R (2R + N) + k_{ТС} \cdot 2^R (2R + R_1). \tag{20}$$

С учетом формул (18) выражение $S(U_2)$ может быть представлено как функция от аргументов $K, P_1, k_p, k_{ТС}, N$.

На рис. 5 – 7 представлены некоторые результаты исследований, при этом параметр K менялся в диапазоне от 100 до 1000, параметр $k_p = 0.2$, число микроопераций $N \in \{10, 50, 100\}$, вероятность $P_1 = 0.7$, параметр $k_{ТС} \in \{0.2; 0.5; 0.7\}$.

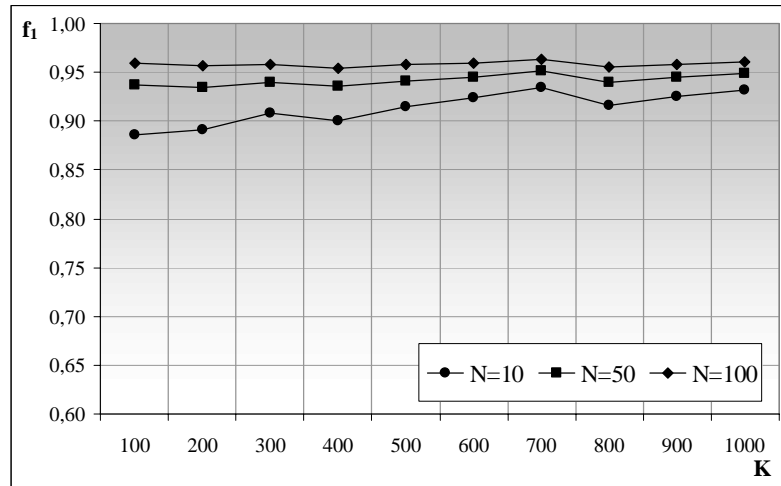


Рисунок 5 – Функция f_1 при $P_1 = 0.7, k_p = 0.2, k_{TC} = 0.7$

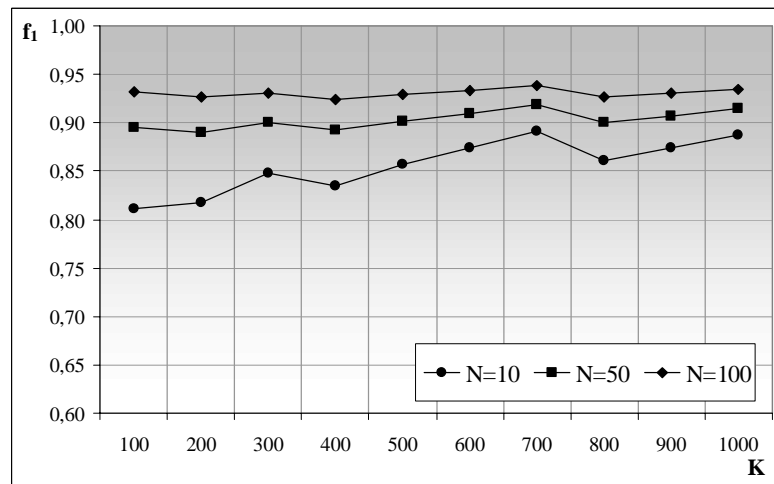


Рисунок 6 – Функция f_1 при $P_1 = 0.7, k_p = 0.2, k_{TC} = 0.5$

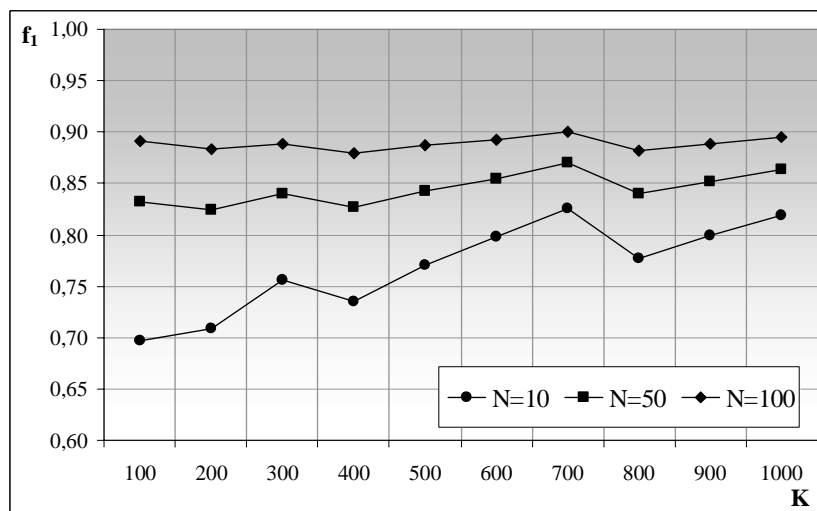


Рисунок 7 – Функция f_1 при $P_1 = 0.7, k_p = 0.2, k_{TC} = 0.2$

Как видно из этих графиков, предлагаемый метод позволяет всегда получить устройства с лучшими характеристиками, чем у автомата U_1 . При этом выигрыш увеличивается по мере уменьшения параметра k_{TC} , а также уменьшения числа микроопераций и числа вершин. Максимальный выигрыш достигался для ГСА, имеющих около 100 вершин, и составлял 30% при $N = 10$ и $k_{TC} = 0.2$. Кроме того, выигрыш увеличивается по мере уменьшения параметра k_p , однако эти графики не показаны.

6. Заключение

Предложенный в работе метод позволяет уменьшить число макроячеек PAL в схеме МПА Мура по сравнению со схемой эквивалентного автомата U_0 . Реализация схемы ТС на PAL позволяет уменьшить число блоков ЕМВ по сравнению с эквивалентным автоматом $U_1(\Gamma)$.

Научная новизна предложенного метода заключается в использовании особенностей автомата Мура (наличие классов псевдоэквивалентных состояний) и элементного базиса (большой коэффициент объединения по входу) для оптимизации числа макроячеек PAL в логической схеме автомата. Практическая значимость метода заключается в уменьшении площади кристалла SoC, занимаемой комбинационной схемой МПА, что позволяет получить схемы, которые обладают меньшей стоимостью, чем известные из литературы аналоги.

Исследования, проведенные авторами, показали, что при выполнении условия (10) автомат $U_2(\Gamma)$ всегда имеет меньшие аппаратные затраты, чем эквивалентный МПА $U_1(\Gamma)$. Максимальный выигрыш может достигать 30%. Необходимо отметить, что автоматы $U_1(\Gamma)$ и $U_2(\Gamma)$ имеют одинаковое быстродействие, то есть выигрыш в аппаратных затратах не приводит к потере производительности.

Литература

1. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
2. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. – Киев: Институт кибернетики НАН Украины, 1997. – 136с.

3. Baranov S. Logic Synthesis for Control Automata. – Kluwer Academic Publishers, 1994. – 312 pp.
4. Maxfield C. The Design Warriors Guide to FPGAs. – Elsevier, 2004. – 541p.
5. Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608с.
6. Barkalov A., Wegrzyn W. Design of Control Units with Programmable Logic. – Zielona Gora: University of Zielona Gora Press, 2006. – 150 pp..
7. Kania D. Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typuPAL. – Zeszyty naukowe Politechniki Śląskiej, Gliwice, 2004. – 240 pp.
8. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. – Донецк: ДНТУ, 2002. – 262 с.
9. DeMicheli G. Synthesis and Optimization of Digital Circuits. – McGraw-Hill, 1994. – 636p.
10. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура. // Кибернетика и системный анализ. – 1998, №1.