товуваних у шифрах. Представляється як одне з перспективних рішень з побудови суперблоку перетворення FI шифру MISTY1, яке примітно тим, що реалізує за один цикл диференціальні показники випадкової підстановки відповідного степеня.

Ключові слова: доказова безпека, диференціал, суперблок, випадкова підстановка.

Lysytska I. V.

COMPARING oN EFFECTIVENESS OF SUPERBOXES some MODERN SIPHERS

New method of assessment indicators provable security block symmetric ciphers sets out. With application of this method are analyzed for differential properties superblock three ciphers: cipher AES, the reduced version cipher Muhomor and cipher MISTY1. The results of computational experiments to determine the values of AMDP and MADP AES superblock are presented. Demonstrated that the resistance of large ciphers and, in particular cipher Rijndael (AES) is independent of the differential properties of S-blocks used in the ciphers. It seems like one of the promising solutions for building superblocks transformation FI cipher MISTY1, which is noteworthy that sells for one cycle of differential performance random permutation corresponding degree.

Key words: of provable security, differential, superblock, random permutation.

УДК 004.3

Баркалов А. А.[1], Мальчева Р. В.[2], Солдатов К. А.[3]

[1]Д-р техн. наук, проф. Университета Зеленогурского (Польша)
[2]Канд. техн. наук, доцент Донецкого национального технического университета
[3]Аспирант Донецкого национального технического университета

ОПТИМИЗАЦИЯ СХЕМЫ АВТОМАТА МУРА, РЕАЛИЗУЕМОЙ В БАЗИСЕ ПЛИС

В статье предлагается метод, предназначенный для уменьшения числа входных переменных и промежуточных термов в реализуемых системах булевых функций. Предложенный метод основан на расширении кодов состояний перехода и замене логических условий. Применение предложенного метода позволяет до 20 % уменьшить общее число макроячеек в блоках БЛУ и БФП.

Ключевые слова: автомат Мура, ПЛИС, ГСА, псевдоэквивалентные состояния, замена логических условий.

ВВЕДЕНИЕ

Практически любая цифровая система включает в свой состав устройство управления (УУ) [1]. При реализации схем УУ часто используется модель микропрограммного автомата Мура [2]. В настоящее время программируемые логические интегральные схемы (ПЛИС) [3] широко применяются для реализации схем УУ. Существуют два основных класса ПЛИС: CPLD (Complex Programmable Logic Devices) и FPGA (Field-Programmable Gate Arrays) [4, 5]. Для уменьшения числа макроячеек ПЛИС в схеме УУ необходимо уменьшать число входных переменных и промежуточных термов в реализуемых системах булевских функций (СБФ) [6]. В настоящей работе предлагается метод решения этой задачи для микропрограммного автомата (МПА) Мура. Метод основан на расширении кодов состояний перехода и замене логических условий.

Целью исследований является оптимизация схемы МПА Мура за счет расширения кодов состояний перехода и замены логических условий.

Задачей исследований является разработка метода синтеза МПА Мура, позволяющего уменьшить число макроячеек ПЛИС в схеме автомата. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [1].

ОБЩИЕ ПОЛОЖЕНИЯ И ОСНОВНАЯ ИДЕЯ ПРЕДЛОЖЕННОГО МЕТОДА

Пусть автомат Мура задан прямой структурной таблицей (ПСТ) со столбцами [1]: a_m, $K(a_m)$, a_S, $K(a_S)$, X_h, Φ_h, h. Здесь a_m – исходное состояние МПА; $K(a_m)$ – код состояния $a_m \in A$ разрядности $R_A = \lceil \log_2 M \rceil$, для кодирования состояний используются внутренние переменные из множества $T = \{T_1,...,T_{R_A}\}$; a_S, $K(a_S)$ – соответственно состояние перехода и его код; X_h – входной сигнал, определяющий переход $< a_m, a_S >$, и равный конъюнкции некоторых элементов (или их отрицаний) множества логических условий $X = \{x_1,...,x_L\}$; Φ_h – набор функций возбуждения триггеров памяти МПА, принимающих единичное значение для переключения памяти из $K(a_m)$ в $K(a_S)$, $\Phi_h \subseteq \Phi = \{\varphi_1,...,\varphi_{R_A}\}$; $h = 1,...,H$ – номер перехода. В столбце a_m записывается набор микроопераций Y_q, формируемых в состоянии $a_m \in A$, где $Y_q \subseteq Y = \{y_1,...,y_N\}$, $q = 1,...,Q$. Эта таблица является основой для формирования систем функций:

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

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

задающих логическую схему МПА. Системы (1)–(2) являются основой для реализации схемы МПА Мура, структура которой показана на рис. 1. Условимся обозначать этот МПА символом U_1.

В МПА U_1 блок формирования функций возбуждения памяти (БФП) реализует систему (1). Блок формирования микроопераций (БФМ) реализует систему (2). Память состояний МПА реализуется на регистре (Рг), состоящем из *D*-триггеров [5]. По сигналу Start в Рг записывается нулевой код начального состояния $a_1 \in A$. По сигналу Clock содержимое Рг меняется в зависимости от функций (2).

При реализации схемы U_1 в базисе FPGA схема БФП реализуется на элементах табличного типа (LUT, look-up table). Для реализации БФП используются встроенные блоки памяти (EMB, embedded memory block) [7, 8]. При реализации схемы U_1 в базисе CPLD схема БФП реализуется на макроячейках программируемой матричной логики (PAL, programmable array logic). Для реализации схемы БФМ могут использоваться макроячейки PAL, либо внешние программируемые ПЗУ. Отметим, что существуют микросхемы CPLD, в которых имеются встроенные EMB. К таким CPLD относятся, например, микросхемы Delta 3K [9]. В данной статье мы рассматриваем случай реализации БФМ на программируемых ПЗУ, которые могут быть как встроенными, так и внешними.

Для оптимизации числа термов в системе (1) предлагается использовать наличие классов псевдоэквивалентных состояний автомата Мура [10]. Число блоков ППЗУ можно уменьшить, если входными переменными БФМ будут адресные разряды наборов микроопераций (НМО) [11]. Число входов схемы БФП можно уменьшить за счет замены логических условий $X_l \in X$ некоторыми переменными $P_g \in P$, где $|P| << |L|$[2]. Все эти идеи положены в основу предлагаемого метода. Обозначим предлагаемый МПА символом U_2.

МЕТОД СИНТЕЗА АВТОМАТА U_2

Одной из особенностей МПА Мура является наличие псевдоэквивалентных состояний [2], то есть состояний с одинаковыми переходами под воздействием одинаковых входных сигналов. Такие состояния соответствуют операторным вершинам [1] алгоритма управления, выходы которых связаны со входом одной и той же вершины алгоритма.

Пусть $\Pi_A = \{B_1,...,B_I\}$ – разбиение множества A на классы псевдоэквивалентных состояний. Закодируем классы $B_i \in \Pi_A$ двоичными кодами $K(B_i)$ разрядности:

$$R_B = \lceil \log_2 I \rceil. \quad (3)$$

Пусть исходная ГСА Г включает Q попарно различных наборов микроопераций (НМО) $Y_q \subseteq Y$. Закодируем набор Y_q двоичным кодом $K(Y_q)$ разрядности:

$$R_Y = \lceil \log_2 Q \rceil. \quad (4)$$

Пусть операторная вершина b_t ГСА Г соответствует состоянию $a_m \in B_i$ и пусть в ней записан набор микроопераций Y_q. Тогда код состояния $a_m \in A$ можно представить в виде конкатенации кодов:

$$K(a_m) = K(B_i) * K(Y_q), \quad (5)$$

где * – знак конкатенации.

Пусть $X(a_m) \subseteq X$ – множество логических условий, определяющих переходы из состояния $a_m \in A$. Пусть $L_m = |X(a_m)|$ и $G = \max(L_1,...,L_m)$. Тогда логические условия $x_l \in X$ можно заменить некоторыми переменными $P_g \in P$, где $|P| = G$ [12].

Пусть $X(B_i) \subseteq X$ – множество логических условий, определяющих переходы из состояний $a_m \in B_i$, где $B_i \in \Pi_A$. В силу определения псевдоэквивалентных состояний справедливо равенство $X(B_i) = X(a_m)$, где $a_m \in B_i$. Таким образом, логические условия $x_l \in X$ можно заменить переменными $P_g \in P$ для классов состояний.

Представление кодов состояний в виде (5) и замена логических условий позволяет получить МПА Мура U_2 (рис. 2), предлагаемый в данной работе. Как видно из рис. 2, автомат U_2 включает блок замены логических условий (БЛУ) и блоки БФП и БФМ. Рассмотрим особенности модели U_2.

Блок БЛУ осуществляет замену логических условий $x_l \in X$. Для этого формируется система функций:

$$P = P(X, \tau). \quad (6)$$

Переменные $\tau_R \in \tau$, где $|\tau| = R_B$, используются для кодирования классов $B_i \in \Pi_A$.

Блок БФП реализует систему функций:

$$\Phi = \Phi(P, \tau). \quad (7)$$

Число функций системы (7) определяется как $R_B + R_Y$. Отметим, что в общем случае $R_A < R_B + R_Y$. Однако блок

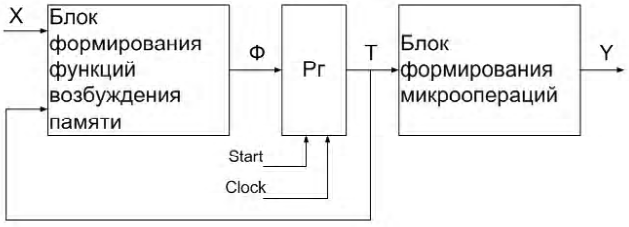

Рис. 1. Структурная схема МПА Мура U_1

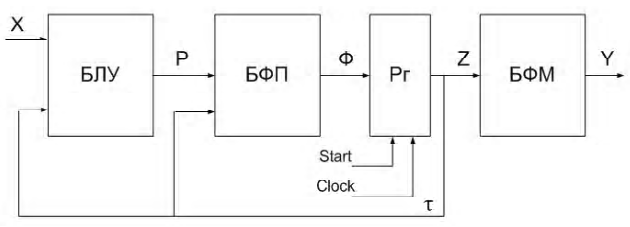

Рис. 2. Структурная схема МПА Мура U_2

БФП в автомате U_1 имеет $R_A + L$ входов, а в автомате U_2 – только $R_B + G$. Кроме того, переход к классам $B_i \in \Pi_A$ позволяет значительно уменьшить число термов в функциях (7) по сравнению с (1).

Блок БФМ реализует систему функций:

$$Y = Y(Z), \tag{8}$$

где переменные $z_R \in Z$ используются для кодирования наборов микроопераций. При этом $|Z| = R_Y \le R_A$. Если выполняется условие:

$$R_Y < R_A, \tag{9}$$

то число блоков ППЗУ в схеме БФМ автомата U_2 уменьшается в m раз, где:

$$m = 2^{R_A - R_Y}. \tag{10}$$

Сравнение автоматов U_1 и U_2 показывает, что автомат U_2 обладает меньшим быстродействием. Это связано с наличием блока БЛУ. Таким образом, предлагаемый метод применим, если он обеспечивает заданное быстродействие управляемой цифровой системы.

Предлагаемый метод синтеза МПА Мура U_2 по отмеченной ГСА Г включает следующие этапы:

1. Формирование разбиения Π_A и кодирование классов $B_i \in \Pi_A$.
2. Формирование таблицы переходов МПА по системе обобщенных формул перехода.
3. Кодирование НМО и определение расширенных кодов состояний $a_m \in A$.
4. Формирование таблицы блока замены логических условий.
5. Формирование прямой структурной таблицы автомата U_2.
6. Формирование таблицы блока БФМ.
7. Реализация схемы автомата в заданном элементном базисе.

Рассмотрим пример применения предложенного метода.

ПРИМЕР ПРИМЕНЕНИЯ ПРЕДЛОЖЕННОГО МЕТОДА

Пусть для некоторой ГСА Γ_1 получено разбиение Π_A, где $\Pi_A = \{B_1,...,B_7\}$. Пусть $A = \{a_1,...,a_{18}\}$ и $B_1 = \{a_1\}$, $B_2 = \{a_2,...,a_6\}$, $B_3 = \{a_7, a_8\}$, $B_4 = \{a_9,...,a_{12}\}$, $B_5 = \{a_{13}, a_{14}\}$, $B_6 = \{a_{15}\}$, $B_7 = \{a_{16}, a_{17}, a_{18}\}$. Таким образом, M=18, R_A=5, I=7, R_B=3. Пусть в ГСА Γ_1 имеется Q=12 попарно различных НМО, тогда R_Y=4. Закодируем наборы $Y_q \subseteq Y$ тривиальным образом: $K(Y_1)$=0000, $K(Y_2)$=0001, … , $K(Y_{12})$=1011. Закодируем классы $B_i \in \Pi_A$ следующим образом: $K(B_1)$=000, … , $K(B_7)$=110. Пусть фрагмент системы обобщенных формул перехода [2] имеет следующий вид:

$$B_2 \to x_3 a_7 \vee \overline{x_3} x_4 a_9 \vee \overline{x_3 x_4} a_{14},$$

$$B_3 \to x_5 a_9 \vee \overline{x_5} x_6 a_{16} \vee \overline{x_5 x_6} a_{14}. \tag{11}$$

Система типа (11) является основой для построения таблицы переходов МПА, имеющей столбцы B_i, a_m, X_h, h. Здесь X_h – конъюнкция логических условий, определяющая переход из состояний $a_S \in B_i$ в состояние $a_m \in A$, $h = \overline{1,H}$ – номер перехода. Для фрагмента (11) таблица переходов имеет 6 строк (табл. 1).

Связь табл. 1 и фрагмента (11) очевидна. Пусть в состоянии a_7 формируется НМО Y_3, в состоянии a_9 – Y_5, в состоянии a_{14} – Y_8, в состоянии a_{16} – Y_6. Это позволяет определить расширенные коды данных состояний: $K(a_7)$=0100010, $K(a_9)$=0110100, $K(a_{14})$=1000111, $K(a_{16})$=1100101. В этих кодах первые три разряда совпадают с кодом $K(B_i)$, где $a_m \in B_i$, а последние четыре разряда определяются кодом НМО.

Пусть для ГСА Γ_1 L=14, при этом $X(B_1)=\{x_1,x_2\}$, $X(B_2)=\{x_3, x_4\}$, $X(B_3)=\{x_5, x_6\}$, $X(B_4)=\{x_1, x_5, x_7\}$, $X(B_5)=\{x_8, x_9, x_{10}\}$, $X(B_6)=\{x_3, x_{11}, x_{12}\}$, $X(B_7)=\{x_{13}, x_{14}\}$. Как следует из анализа этих множеств G=3 и $P=\{p_1, p_2, p_3\}$. Таблица блоков БЛУ имеет 7 строк и 4 столбца (табл. 2).

Из табл. 2 следует система (6). Например, из анализа столбца P_1 можно получить функцию:

$$P_1 = x_1(B_1 \vee B_4) \vee x_3(B_2 \vee B_6) \vee x_8 B_5.$$

Используя коды классов $B_i \in \Pi_A$ можно получить окончательное выражение:

$$P_1 = \overline{x_1}\overline{\tau_1}\overline{\tau_2}\tau_3 \vee x_1\overline{\tau_1}\tau_2\tau_3 \vee x_3\overline{\tau_1}\overline{\tau_2}\tau_3 \vee x_3\tau_1\overline{\tau_2}\tau_3 \vee x_8\tau_1\overline{\tau_2}\tau_3.$$

Отметим, что подобные функции тривиально реализуются на мультиплексоре [12]. Как известно, мультиплексор является стандартным библиотечным элементом САПР [7–9].

Прямая структурная таблица МПА U_2 строится, как расширение таблицы переходов столбцами $K(B_i)$, $K(a_m)$, Φ_h и заменой столбца X_h столбцом P_h. В столбце Φ_h записываются функции $D_r \in \Phi$, принимающие единичные значения на h-м переходе МПА. Для нашего примера табл. 1 преобразовывается в табл. 3 тривиальным образом.

Таблица 1. Фрагмент таблицы переходов автомата Мура

B_i	a_m	X_h	h	B_i	a_m	X_h	h
B_2	a_7	x_3	1	B_3	a_9	x_5	4
	a_9	$\overline{x_3}x_4$	2		a_{16}	$\overline{x_5}x_6$	5
	a_{14}	$\overline{x_3 x_4}$	3		a_{14}	$\overline{x_5 x_6}$	6

Таблица 2. Таблица блока замены логических условий

B_i	P_1	P_2	P_3	i	B_i	P_1	P_2	P_3	i
B_1	x_1	x_2	–	1	B_5	x_8	x_9	x_{10}	5
B_2	x_3	x_4	–	2	B_6	x_3	x_{11}	x_{12}	6
B_3	–	x_5	x_6	3	B_7	–	x_{13}	x_{14}	7
B_4	x_1	x_5	x_7	4	–	–	–	–	

Таблица 3. Фрагмент прямой структурной таблицы МПА U_2

B_i	$K(B_i)$	a_m	$K(a_m)$	P_h	Φ_h	h
B_2	001	a_7	0100010	P_1	D_2D_6	1
		a_9	0110100	$\overline{P_1}P_2$	$D_2D_3D_5$	2
		a_{14}	1000111	$\overline{P_1}\,\overline{P_2}$	$D_1D_5D_6D_7$	3
B_3	010	a_9	0110100	P_2	$D_2D_3D_5$	4
		a_{16}	1100101	$\overline{P_2}P_3$	$D_1D_2D_5D_7$	5
		a_{14}	1000111	$\overline{P_2}\,\overline{P_3}$	$D_1D_5D_6D_7$	6

Эта таблица является основой для формирования системы (7). Так, с учетом минимизации, из табл. 3 можно получить функцию:

$$D_1 = \overline{\tau_1}\,\overline{\tau_2}\tau_3\,\overline{P_1}\,\overline{P_2} \vee \overline{\tau_1}\tau_2\,\overline{\tau_3}\,\overline{P_2}\,.$$

Таблица блока БФМ строится тривиальным образом, и этот этап в данной статье не рассматривается. Последний этап метода связан с применением промышленных САПР фирм-производителей ПЛИС [7–9]. Этот этап мы также здесь не рассматриваем.

ЗАКЛЮЧЕНИЕ

Предложенный метод позволяет гарантировано уменьшить число переходов МПА Мура до величины этого параметра эквивалентного автомата Мили. При этом соответственно уменьшается число термов в функциях возбуждения памяти МПА.

Использование метода замены логических условий позволяет уменьшить число входных переменных в функциях возбуждения памяти. Это особо важно для минимизации числа LUT-элементов при реализации схемы на ПЛИС типа FPGA.

Представление кода состояния в виде конкатенации кодов классов псевдоэквивалентных состояний и наборов микроопераций может привести к увеличению разрядности кода состояния. Однако при выполнении условия (9) такое представление позволяет уменьшить число блоков памяти в схеме формирования микроопераций. Кроме того, предлагаемое представление позволяет закодировать классы так, чтобы оптимизировать число макроячеек в блоке замены логических условий.

Проведенные авторами исследования показали, что применение предложенного метода позволяет до 20 % уменьшить общее число макроячеек в блоках БЛУ и БФП по сравнению с этим параметром для блока БВП автомата U_1. Кроме того, число блоков памяти в БФМ практически всегда уменьшалось в два раза. Отметим, что время цикла МПА U_2 в 1,5 раза больше, чем для эквивалентного автомата U_1.

Научная новизна предложенного подхода заключается в использовании расширенного представления кодов состояний и замены логических условий для уменьшения числа макроячеек ПЛИС и блоков ППЗУ в схеме автомата Мура.

Практическая значимость метода заключается в уменьшении стоимости схемы МПА Мура по сравнению с известными в литературе аналогами.

СПИСОК ЛИТЕРАТУРЫ

1. *Baranov, S.* Logic and System Design of Digital Systems / Baranov S. – Tallinn : TUT Press, 2008. – 328 pp.
2. *Barkalov, A.* Logic Synthesis for FSM-based Control Units / A. Barkalov, L. Titarenko. – Berlin : Springer, 2009. – 233 pp.
3. *Грушницкий, Р. И.* Проектирование систем с использованием микросхем программируемой логики / Р. И. Грушницкий, А. Х. Мурзаев, Е. П. Угрюмов. – С. Пб. : БХВ.-Петербург, 2002. – 608 с.
4. *Maxfield, C.* The Design Warrior's Guide to FPGAs / Maxfield C. – Amsterdam : Elsevier, 2004. – 541 pp.
5. *Соловьёв, В. В.* Логическое проектирование цифровых систем на основе программируемых логических интегральных схем / В. В. Соловьёв, А. С. Климович. – М. : Горячая линия-Телеком, 2008. – 376 с.
6. *DeMicheli, G.* Synthesis and Optimization of Digital circuits / DeMicheli G. – New York : McGraw Hill, 1994. – 541 pp.
7. FPGA, CPLD, and ASIC from Altera [Electronic resource]: база данных содержит сведения о микросхемах ПЛИС фирмы Altera. – Электрон. дан. – Режим доступа : http://www.altera.com. – Загл. с экрана.
8. FPGA and CPLID Solutions from Xilinx, Inc [Electronic resource]: база данных содержит сведения о микросхемах ПЛИС фирмы Xilinx. – Электрон. дан. – Режим доступа: www.xilinx.com. – Загл. с экрана.
9. Cypress Semiconductor [Electronic resource]: база данных содержит сведения о микросхемах ПЛИС фирмы Cypress. – Электрон. дан. – Режим доступа: www.cypress.com. – Загл. с экрана.
10. *Баркалов, А. А.* Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. – 1998. – № 1. – С. 65–72.
11. *Баркалов, А. А.* Матричная реализация автомата Мура с расширением кодов состояний перехода/ А. А. Баркалов, Р. В. Мальчева, К. А. Солдатов // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2010). Выпуск 11 (164). – Донецк : ГВУЗ «ДонНТУ». – 2010. – С. 79–83.
12. *Baranov, S.* Logic Synthesis for Control Automata / Baranov S. – New York : Kluwer Academic Publishers, 1994. – 312 pp.

Стаття надійшла до редакції 23.01.2012.

Баркалов О. О., Мальчева Р. В., Солдатов К. А.

ОПТИМІЗАЦІЯ СХЕМИ АВТОМАТА МУРА, ЩО РЕАЛІЗУЄТЬСЯ В БАЗИСІ ПЛІС

У статті пропонується метод, призначений для зменшення числа вхідних змінних і проміжних термів в реалізованих системах булевих функцій. Запропонований метод заснований на розширенні кодів станів переходу і заміні логічних умов. Застосування запропонованого методу дозволяє до 20 % зменшити загальну кількість макрокомірок в блоках БЛУ та БФП.

Ключові слова: автомат Мура, ПЛІС, ГСА, псевдоеквівалентні стани, заміна логічних умов.

Barkalov A. A., Malcheva R. V., Soldatov K. A.

OPTIMIZATION OF MOORE FINITE STATE MACHINE IMPLEMENTED ON THE PROGRAMMABLE LOGIC

This article is proposed a method which is designed to reduce the number of input variables and intermediate terms of Boolean functions. The method is based on the extended codes of states and replacement of logic conditions. Application of the proposed method allows up to 20% reduction in the total count of macrocells in blocks BLC and BFM.

Key words: Moore FSM, Programmable Logic, GSA, pseudoequivalent states, replacement of logic conditions.