Как следует из рис. 6, коды состояний $a_m \in B_1$ входят в интервал 00**. Этот интервал можно использовать как код класса $B_1 \in \Pi_A$. Аналогично можно найти коды $K(B_2)$=01**, $K(B_3)$=10** и $K(B_4)$=11**.

Построим таблицу ЗЛУ. В нашем примере это табл. 1.

*Таблица 1.* Таблица ЗЛУ автомата $U_4$

| $B_1$ | $B_1$ | $B_2$ | $B_3$ | $B_4$ |
|---|---|---|---|---|
| $K(B_i)$ | 00 | 01 | 10 | 11 |
| $p_1$ | $x_1$ | $x_3$ | $x_5$ | — |
| $p_2$ | $x_2$ | $x_4$ | $x_6$ | — |

В табл. 1 переменная $x_l \hat{I} X$ записывается на пересечении строки $p_g$ и столбца $B_i$ , если переменная $x_l$ заменяется переменной $p_g$ для состояний $B_i \hat{I} P_A$ . Первая позиция соответствует внутренней переменной $T_1$ , вторая – $T_2$ .

Из табл. 1 можно найти функции (11). Они имеют следующий вид:

$$P_1 = \overline{T}_1\overline{T}_2 x_1 \vee \overline{T}_1 T_2 x_3 \vee T_1\overline{T}_2 x_5$$
$$P_2 = \overline{T}_1\overline{T}_2 x_2 \vee \overline{T}_1 T_2 x_4 \vee \overline{T}_1\overline{T}_2 x_6 \qquad (12)$$

Прямая структурная таблица (ПСТ) автомата $U_4$ имеет следующие столбцы: $B_i$ – класс ПЭС, включающий текущее состояние $a_m \in A$; $K(B_i)$ – код класса $B_i \in \Pi_A$; $a_S$ – состояние перехода; $K(a_S)$ – код состояния $a_S \in A$; $p_h$ – входной сигнал, определяющий переход $a_m$, $a_S$; $Y^1_h$ – выходной сигнал, формируемый на переходе $a_m$, $a_S$; $\Phi_h$ – набор функций возбуждения памяти, равных 1 для загрузки в регистр RG кода $K(a_S)$; h – номер перехода. В рассматриваемом примере таблица имеет H=10 строк (табл. 2).

*Таблица 2.* Прямая структурная таблица автомата $U_4$

| $B_i$ | $K(B_i)$ | $a_S$ | $K(a_S)$ | $p_h$ | $Y^1_h$ | $\Phi_h$ | $h$ |
|---|---|---|---|---|---|---|---|
| $B_1$ | 00 | $a_2$ | 0100 | $p_1$ | $y_1$ | $D_2$ | 1 |
| | | $a_2$ | 0100 | $p_1 p_2$ | $y_2 y_3$ | $D_2$ | 2 |
| | | $a_3$ | 0101 | $p_1 p_2$ | $y_4$ | $D_2 D_4$ | 3 |
| $B_2$ | 01 | $a_5$ | 1001 | $p_1 p_2$ | $y_1 y_4$ | $D_2 D_4$ | 4 |
| | | $a_6$ | 1010 | $p_1 p_2$ | – | $D_1 D_3$ | 5 |
| | | $a_4$ | 1000 | $p_1$ | $y_5$ | $D_1$ | 6 |
| $B_3$ | 10 | $a_7$ | 1100 | $p_1$ | $y_2 y_5$ | $D_1 D_2$ | 7 |
| | | $a_8$ | 1101 | $p_1 p_2$ | $y_5$ | $D_1 D_2 D_4$ | 8 |
| | | $a_9$ | 1110 | $p_1 p_2$ | $y_2 y_3$ | $D_1 D_2 D_3$ | 9 |
| $B_4$ | 11 | $a_1$ | 0000 | 1 | – | – | 10 |

Рассмотрим принципы заполнения столбца $p_h$ . Переход $a_1$, $a_2$ определяется сигналами $x_1$ и $x_1x_2$ . Из табл.1 следует, что $x_1 = p_1$ и $x_2 = p_2$ . Следовательно, в первой строке ПСТ пишется $p_1$, а во второй – $p_1$ $p_2$ . Остальные строки формируются аналогичным образом.

Для построения таблиц элементов LUT блока LUTer1 необходимо проанализировать систему (12). Пусть $S_L$=4. В уравнении для $P_1$ содержится 5 букв. Следовательно, это уравнение должно быть преобразовано с использованием принципа функциональной декомпозиции [9-10].

Преобразуем уравнение следующим образом:

$$p_1 = \overline{T_1}\left(\overline{T_2}x_1 \vee T_2 x_3\right) \vee T_1\left(\overline{T_2}x_5\right) = A \vee B \qquad (13)$$

Уравнение (13) соответствует схеме, включающей два элемента LUT (рис.7). Например, элементу LUT2 соответствует таблица истинности – табл.3. Аналогичным образом преобразовываются все уравнения, реализуемые на элементах LUT.

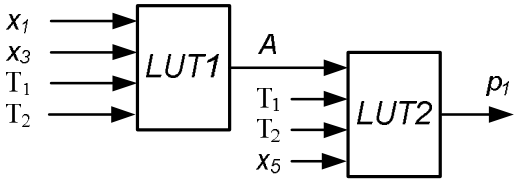


Рис. 7. Реализация функции $p_1$ при $S_L$=4

*Таблица 3.* Таблица элемента LUT2

| A T1 T2 x5 | p1 | A T1 T2 x5 | p1 |
|---|---|---|---|
| 0 0 0 0 | 0 | 1 0 0 0 | 1 |
| 0 0 0 1 | 0 | 1 0 0 1 | 1 |
| 0 0 1 0 | 0 | 1 0 1 0 | 1 |
| 0 0 1 1 | 0 | 1 0 1 1 | 1 |
| 0 1 0 0 | 0 | 1 1 0 0 | 1 |
| 0 1 0 1 | 1 | 1 1 0 1 | 1 |
| 0 1 1 0 | 0 | 1 1 1 0 | 1 |
| 0 1 1 1 | 0 | 1 1 1 1 | 1 |

Таблица блока EMB строится на основе ПСТ. Она включает следующие столбцы: $K(B_i)$, p (адрес ячейки памяти), $Y^1$, Ф (содержимое ячейки памяти); $q$ – номер ячейки памяти ($q = \overline{1,Q}$). В общем случае имеет место следующее равенство:

$$Q = 2^{R1+G}.$$

Переходы из состояний $a_m \in B_i$ задаются при помощи $H(B_i)$ строк таблицы, где:

$$H(B_i) = 2^G.$$

В рассматриваемом примере Q = 16 и $H(B_i)$ = 4. В табл.4 представлен фрагмент таблицы блока EMB для нашего примера.

*Таблица 4.* Таблица блока EMB автомата $U_4$

| $K(B_i)$ | $p$ | $Y^1$ | Ф | $q$ | $h$ |
|---|---|---|---|---|---|
| $T_1 T_2$ | $p_1 p_2$ | $y_1 y_2 y_3 y_4 y_5$ | $D_1 D_2 D_3 D_4$ | | |
| 0 1 | 0 0 | 0 0 0 0 1 | 1 0 0 0 | 5 | 6 |
| 0 1 | 0 0 | 0 0 0 0 1 | 1 0 0 0 | 6 | 6 |
| 0 1 | 0 1 | 0 0 0 0 0 | 1 0 1 0 | 7 | 5 |
| 0 1 | 0 1 | 1 0 0 1 0 | 1 0 0 1 | 8 | 4 |

Таблица 4 задает переходы из состояний $a_2$, $a_3$ Î $B_2$. Столбец $h$ добавлен, чтобы показать соответствие между ПСТ (Табл.2) и таблицей блока EMB.

Таблица блока LUTer2 объединяет $N_2$ таблицы для функций $y_n \in Y^2$. Эта таблица имеет столбцы: $K(a_m)$ – адрес ячейки, $Y^2$ – содержимое ячейки, $q$ – номер ячейки ($q = \overline{1, Q_1}$). Параметр $Q_1$ определяется как

$$Q_1 = 2^R.$$

В рассматриваемом примере имеем $Q_1 = 16$ (табл. 5).

*Таблица 5.* Таблица блока LUTer2 автомата $U_4$

| $K(a_m)$ | $Y^2$ | q | $m$ | $K(a_m)$ | $Y^2$ | q | m |
|---|---|---|---|---|---|---|---|
| $T_1\,T_2\,T_3\,T_4$ | $y_6\,y_7\,y_8$ | | | $T_1\,T_2\,T_3\,T_3$ | $y_6\,y_7\,y_8$ | | |
| 0 0 0 0 | 0 0 0 | 1 | 1 | 1 0 0 0 | 0 0 1 | 9 | 5 |
| 0 0 0 1 | 0 0 0 | 2 | – | 1 0 0 1 | 0 0 0 | 10 | 6 |
| 0 0 1 0 | 0 0 0 | 3 | – | 1 0 1 0 | 0 0 0 | 11 | – |
| 0 0 1 1 | 0 0 0 | 4 | – | 1 0 1 1 | 0 0 0 | 12 | – |
| 0 1 0 0 | 0 0 0 | 5 | 2 | 1 1 0 0 | 0 0 0 | 13 | 7 |
| 0 1 0 1 | 0 0 0 | 6 | 3 | 1 1 0 1 | 0 0 0 | 14 | 8 |
| 0 1 1 0 | 0 0 0 | 7 | 4 | 1 1 1 0 | 0 0 0 | 15 | 9 |
| 0 1 1 1 | 0 0 0 | 8 | – | 1 1 1 1 | 0 0 0 | 16 | – |

Столбец $m$ добавлен в табл.3, чтобы показать соответствие между строками табл.3 и состояниями $a_m \in A$.

Последний этап предлагаемого метода связан с использованием стандартных пакетов для реализации схем в базисе FPGA. В данной работе этот этап мы не рассматриваем.

### *Заключение*

Предложенный в работе метод позволяет уменьшить число элементов LUT и блоков EMB в схеме СМПА по сравнению с известными методами. Это достигается за счет оптимального кодирования состояний, позволяющего уменьшить число адресных входов в блоках замены логических условий и EMB. Целью оптимального кодирования ПЭС является представление каждого класса ПЭС одним интервалом кодирующего пространства.

Метод целесообразно использовать, если замена логических условий не приводит к схемам с одним блоком EMB. Анализ библиотеки [13] показал, что к этому классу относится более 20% стандартных автоматов. Кроме того, для этих автоматов число букв в функциях $p_g \in P$ превышает число входов элементов LUT. Предлагаемый метод позволяет в среднем на 32% уменьшить число элементов LUT в блоке ЗЛУ.

Дальнейшее направление наших исследований связано с адаптацией подхода [15-16] к особенностям совмещенного автомата.

### *Список литературы*

1. Baranov S. Logic Synthesis for Control Automata. – Dordrecht: Kluwer Academic Publishers, 1994. – 312 pp.

2. DeMicheli G. Synthesis and Optimization of Digital Circuits. – New York: McGraw-Hill, 1994. – 636 pp.

3. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия – ТЕЛЕКОМ, 2001. – 636 с.

4. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA–based circuits

using Hierarchical Finite State Machines. – Tallinn: TUT Press, 2012. – 240 pp.

5. Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.

6. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-based Systems. – Berlin: Springer, 2014. – 432pp.

7. Cong, J. and Yan. K. (2000). Synthesis for FPGAs with Embedded Memory Blocks, Proceeding of the 2000 ACM/SIGDA 8th International Symposium on FPGAs, pp. 75 – 82.

8. Garcia–Vargas. L., Senhadji–Navarro, R., M. Civit–Balcells, A. and Guerra–Gutierrezz, P. (2007). ROM–Based Finite State Machine Implementation in Low Cost FPGAs, IEEE International Simposium on Industrial Electronics, Vigo, pp. 2342–2347.

9. Nowicka, M., Łuba, T. and Rawski, V/ (1999). FPGA–based decomposition of Boolean functions: algorithms and implementations, Advanced Computer Systems, pp. 502 – 509.

10. Rawski, M., Selvaraj, H. and Łuba, T. (2005). An application of functional decomposition in ROM–based FSM implementation in FPGA devices, Journal of System Architecture 51(6–7): pp. 424 – 434.

11. Rawski, M., Tomaszewicz, P., Borowski, G. and Łuba, T. (2011). Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks on FPGAs, Design of Digital Systems and Devises. LNEE 70, Springer, Berlin, pp. 121 – 144.

12. Tiwari, A. and Tomko, K. (2004). Saving power by mapping finite state machines into embedded memory blocks in FPGAs, Proceedings of Design Automation and Test in Europe, Vol. 2, pp. 916–921.

13. Yang S. Logic synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. – 1991, 43 pp.

14. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. – 1998, №1. – стр. 65-72.

15. Баркалов А.А., Визор Я.Е., Матвиенко А.В, Титаренко Л.А. Оптимизация схемы автомата Мура с двумя источниками кодов // УСиМ. – 2015. – №2.

16. Barkalov A., Titarenko L., Kolopenczyk M. EMB–based design of Mealy FSM.– Proceedings of 12th IFAC Conference on programmable devices and embedded systems. – 2013, pp. 215–220.

Статью представлено к печати 29.09.2016