



РЕАЛИЗАЦИЯ КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВ УПРАВЛЕНИЯ НА FPGA- МИКРОСХЕМАХ

БАРКАЛОВ А.А., ТИТАРЕНКО Л.А.,
МИРОШКИН А.Н.

Предлагаются результаты исследования реализации композиционных устройств управления на современных микросхемах программируемой логики (Spartan6). Приводится сравнение с реализацией автоматов с жесткой логикой. Определяются диапазоны параметров исходных граф-схем алгоритмов, при реализации которых будут получены устройства с параметрами, которые удовлетворяют заданным критериям.

1. Введение

Базис программируемых вентильных матриц (Field Programmable Gate Array, FPGA) из простой замены логическим элементам на плате стал конкурентом сложным специализированным интегральным схемам (Application-Specific Integrated Circuit, ASIC) [1]. Это стало возможным благодаря увеличению количества эквивалентных элементов на единицу площади кристалла. Наличие специализированных элементов, таких как синхронные блоки встроенной памяти (Embedded Memory Block, EMB), делают базис FPGA очень удобным для реализации сложных цифровых систем. Блоки EMB могут использоваться также и для реализации схем с памятью, таких, как композиционные управляющие автоматы. Целый ряд работ посвящен реализации конечных автоматов с памятью в базисе FPGA [2–6].

Целью данной работы является исследование эффективности базиса FPGA при реализации композиционных микропрограммных устройств управления (КМУУ). Предлагаемая статья рассматривает эффективность реализации ряда структур управляющих автоматов в базисе FPGA фирмы Xilinx. Вторая часть статьи содержит некоторые сведения о синтезе КМУУ. В третьей части описана система автоматизированного проектирования управляющих автоматов. Результаты работы, некоторые графики и выводы представлены в четвертой части. Выводы и направления дальнейших исследований завершат данную статью.

2. Композиционное микропрограммное устройство управления

Конечный автомат – модель, которая состоит из пяти компонент (X, Y, S, f, g) , где $X = \{x_1, \dots, x_L\}$ – входной алфавит, $Y = \{y_1, \dots, y_N\}$ – выходной алфавит, S – множество внутренних состояний автомата, $f : X \times S \rightarrow S$ – функции возбуждения, обеспечивающие переход из текущего состояния s_i в следующее состояние s_j при определенном наборе входных логических сигналов, $g : X \times S \rightarrow Y$ – функции выхода, которые определяют множество активных выходных сигналов в текущем состоянии под воздействием входных условий. Иногда добавляют шестой элемент, S_0 – начальное состояние автомата.

Композиционное микропрограммное устройство управления является одним из способов реализации управляющего устройства [7, 8]. КМУУ использует микросхемы ПЗУ для реализации управляющей памяти и дискретную логику – для схем адресации.

В КМУУ выходные функции зависят только от состояния автомата и формируются при помощи ПЗУ, следовательно, могут быть описаны как $g : S \rightarrow Y$.

Управляющий автомат может быть задан в виде направленного графа переходов, граф-схемы алгоритма или прямой структурной таблицы (ПСТ), каждая строка которой описывает один переход автомата (таблица). Строка ПСТ включает следующие элементы: текущее состояние am , его двоичный код $k(am)$, следующее состояние as и его код $k(as)$, множество логических сигналов X , обеспечивающих переход, функции возбуждения памяти D регистра состояния автомата и номер h строки.

Система функций переходов КМУУ задается формулой:

$$d_i = \bigvee_{h=1}^H (P_i^h \wedge (\bigwedge_{r=1}^R q_r^h) \wedge (\bigwedge_{l=1}^L C_l^h x_l^h)), i = 1, \dots, R, \quad (1)$$

где d_i – функция возбуждения i -го разряда регистра состояния автомата; P_i^h – переменная, определяющая наличие функции d_i в строке h ПСТ ($P_i^h = 1$, если функция d_i присутствует в строке h , и $P_i^h = 0$ – в обратном случае); q_r^h – выход r -го разряда регистра состояния автомата в строке h ПСТ автомата ($q_r^h = q_r$, если на выходе r -го разряда регистра находится значение логической «1», и $q_r^h = \overline{q_r}$ – в обратном случае); x_l^h – переменная, определяющая значение логического условия x_l в строке h ПСТ автомата ($x_l^h = x_l$ при условии, что переход в строке h выполняется при выполнении условия, и $x_l^h = \overline{x_l}$ – при невыполнении); C_l^h – переменная, определяющая наличие логического условия x_l в строке h ПСТ автомата ($C_l^h = 1$ при

наличии сигнала x_1 и $C_1^h = 0$ – при его отсутствии); H – количество строк ПСТ автомата; R – разрядность регистра состояния автомата; L – количество сигналов во входном алфавите автомата.

Функции перехода формируются только для вершин-выходов операторных линейных цепей (ОЛЦ). Определение ОЛЦ, ее входа, выхода можно найти в [9].

Фрагмент ПСТ управляющего автомата

a_m	$k(a_m)$	a_s	$k(a_s)$	X	Y	D	h
a_6	0110	a_4	0100	$\overline{x_2}$	y_1, y_2, y_4	d_3	1
		a_{11}	1000	x_2		d_4	2
a_4	0100	a_8	0111	1	y_2, y_3	d_3, d_2, d_1	3

Для данного фрагмента ПСТ система функций f имеет вид:

$$\begin{aligned} d_1 &= \overline{q_4 q_3 q_2 q_1}; \\ d_2 &= \overline{q_4 q_3 q_2 q_1}; \\ d_3 &= \overline{q_4 q_3 q_2 q_1 x_2} \vee \overline{q_4 q_3 q_2 q_1}; \\ d_4 &= \overline{q_4 q_3 q_2 q_1 x_2}. \end{aligned} \quad (2)$$

Система функций выходов КМУУ с общей памятью (Common Memory, CM-структура) реализуется на ПЗУ тривиально: в памяти расположена микропрограмма из M_1 слов, где M_1 – количество операторных вершин в ГСА. Разрядность слова при унитарном кодировании микроопераций равна

$$N_1 = |Y| + 2, \quad (3)$$

где константа «2» определяет разряды для хранения внутренних переменных y_0 и y_E .

В случае с КМУУ с разделением кодов (Code Sharing, CS-структура) количество слов M_2 микропрограммы можно найти как

$$M_2 = (G - 1) \cdot 2^{\lceil \log_2 |\alpha_{\max}| \rceil} + |\alpha_{\min}|, \quad (4)$$

где G – количество ОЛЦ α_g в исходной ГСА; $\alpha_{\max}, \alpha_{\min}$ – ОЛЦ соответственно с максимальным и минимальным количеством компонент. Для обеспечения минимального числа слов микропрограммы необходимо ОЛЦ $\alpha_{\min}$ дать максимальный из используемых кодов ОЛЦ, что отнесет соответствующее содержимое УП в область с максимальными адресами.

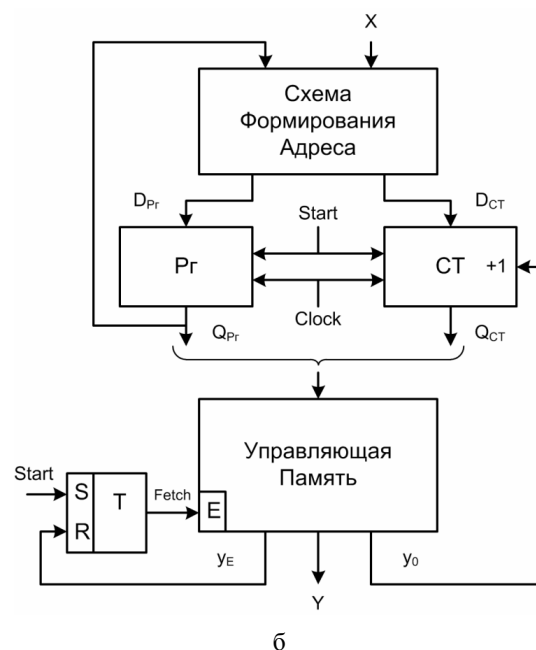
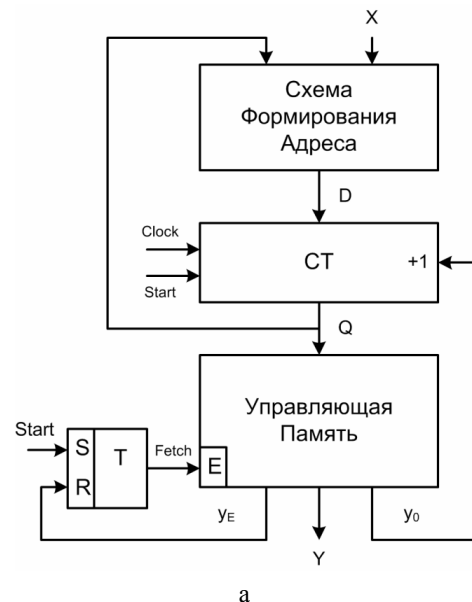


Рис. 1. Структурная схема КМУУ: а – с общей памятью (CM); б – с разделением кодов (CS)

Разрядность слова равна соответствующей разрядности слова управляющей памяти КМУУ CM, $N_2 = N_1$.

На рис. 1, а представлена CM-структура КМУУ (базовая структура). Схема формирования адреса (СФА) реализует систему функций переходов (1), а управляющая память (УП) содержит микропрограмму, в которой находятся закодированные одним из известных способов [9] выходные функции (3).

По сигналу Start в СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ разрешает чтение из управляющей памяти. Если считанная из УП микрокоманда не соответствует выходу ОЛЦ, то одновременно с микрооперациями Y формируется сигнал y_0 . Если $y_0 = 1$, то к содержимому СТ прибавляется единица и адресуется следующая компонента текущей ОЛЦ. Если выход ОЛЦ достигнут, то $y_0 = 0$.

При этом адрес входа следующей ОЛЦ формируется схемой СФА. При достижении окончания микропрограммы формируется сигнал u_E , триггер ТВ обнуляется и выборка микрокоманд прекращается.

Кроме указанной структуры исследовалась также структура КМУУ с разделением кодов [8, 10], приведенная на рис. 1, б. Отличается данная структура тем, что источником адреса для СФА является только код текущей ОЛЦ, что в общем случае снижает затраты на реализацию данной схемы.

3. Система автоматизированного проектирования

Для описания схем конечных автоматов используются различные языки описания аппаратуры, наиболее используемыми из которых являются VHDL [11] и Verilog [12]. В целях автоматизации синтеза моделей управляющих автоматов по исходным описаниям алгоритмов была создана система ГенА (Генератор Автоматов). Схема процесса исследования приведена на рис. 2.

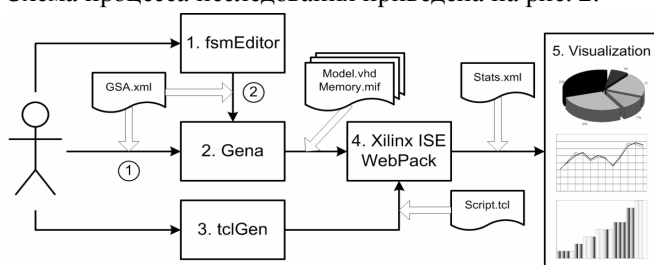


Рис. 2. Схема процесса исследования при помощи САПР

Исходными данными для системы является описание граф-схемы алгоритма управления в формате XML, которое можно получить в любом текстовом редакторе или в специализированной среде fsmEditor, имеющей графический интерфейс и позволяющей представлять ГСА в более удобном для пользователя виде. На рис. 3 представлена ГСА в среде fsmEditor.

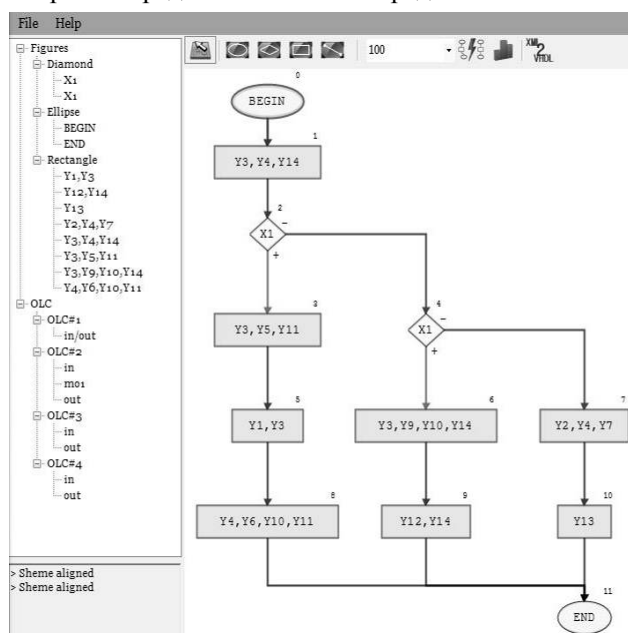


Рис. 3. Визуальное отображение алгоритма управления в среде fsmEditor

Из описания ГСА формируется модель управляющего автомата указанной структуры. Модель представляет собой описание функциональных узлов автомата на языке VHDL, а также при необходимости файлы прошивки памяти. Процессом имплементации модели в среде Xilinx ISE управляет скрипт, который можно создать при помощи утилиты telGen. Необходимо лишь указать расположение файлов модели устройства и тип микросхемы, на которой оно должно быть реализовано.

Модель автомата, файлы прошивки памяти и сценарий передаются в САПР Xilinx ISE, где выполняется имплементация модели с учетом реальных параметров указанной микросхемы. При наличии программатора и соответствующей микросхемы можно реализовать управляющее устройство непосредственно из разработанной САПР, указав в сценарии необходимость физической реализации автомата. Статистика об использовании микросхемы участвует в составлении общей картины эффективности реализации управляющих автоматов. По полученной статистике можно вывести аналитические зависимости, которые помогут еще на этапе оценки параметров ГСА определить структуру управляющего автомата, реализация которого дает результат, удовлетворяющий выдвигаемым к цифровому устройству критериям. Среди таких критериев может быть минимальный объем необходимой памяти, дискретной логики, минимальное время работы устройства, наименьшая стоимость микросхемы, которая позволит реализовать устройство, и т.д.

Визуализация статистических результатов может быть выполнена в любом подходящем программном пакете.

4. Результаты экспериментов

Для эксперимента были выбраны ГСА со следующими параметрами:

- количество вершин от 10 до 500 с шагом 10;
- доля операторных вершин от 50 до 90% с шагом 10%;
- количество микроопераций – 15;
- количество логических условий – 5.

Возможность пакетного создания ГСА с указанием диапазонов необходимых параметров реализована в программе fsmEditor.

Для указанных ГСА были получены результаты имплементации КМУУ с общей памятью и с разделением кодов (рис. 4-6). Также приводится сравнительная характеристика с результатами реализации тех же ГСА в виде автоматов Мили (рис. 7). В качестве элементного базиса выбраны микросхемы Spartan-6 фирмы Xilinx. Измеряемый параметр – количество использованных LUT-элементов микросхемы FPGA.

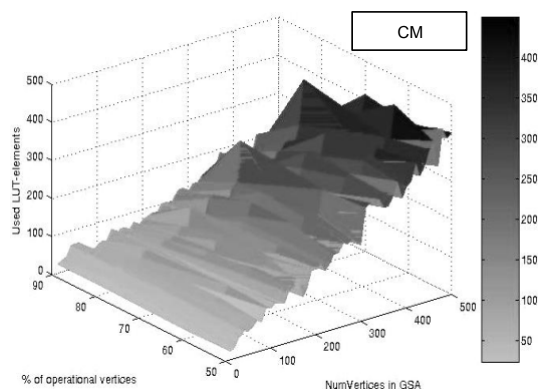


Рис. 4. Зависимость аппаратных затрат при реализации КМУУ с общей памятью на микросхемах серии Spartan6 от количества вершин в исходной ГСА и ее разветвленности

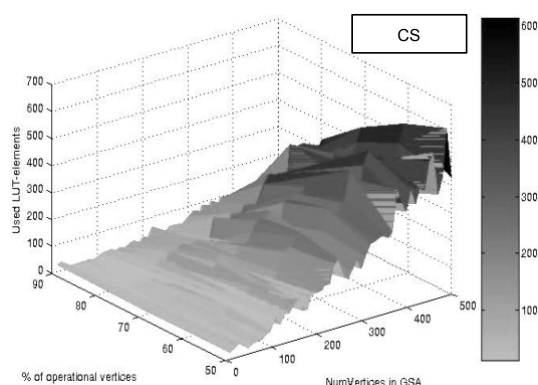


Рис. 5. Зависимость аппаратных затрат при реализации КМУУ с разделением кодов на микросхемах серии Spartan6 от количества вершин в исходной ГСА и ее разветвленности

Рис. 4 и 5 подтверждают, что структуры КМУУ с ростом процента операторных вершин в ГСА требуют меньше аппаратных затрат. На рис. 6 приведена сравнительная характеристика структур CM и CS.

Как видно из рис. 6, при сравнительно небольших ГСА (до 200 вершин) их реализация в виде структуры с разделением кодов дает меньшие аппаратные затраты. С увеличением числа вершин в области ГСА с высоким процентом операторных вершин преимущество CS-автомата остается, а в области с высоким показателем разветвленности ГСА (менее 70% операторных вершин) график CS-структуры выше аналогичного графика CM-структуры, что означает большие аппаратные затраты.

Для сравнения классов автоматов с жесткой логикой и композиционных автоматов были выбраны автомат Мили и CS-структура КМУУ. Сравнение результатов реализации данных автоматов на микросхемах Spartan6 приведено на рис. 6.

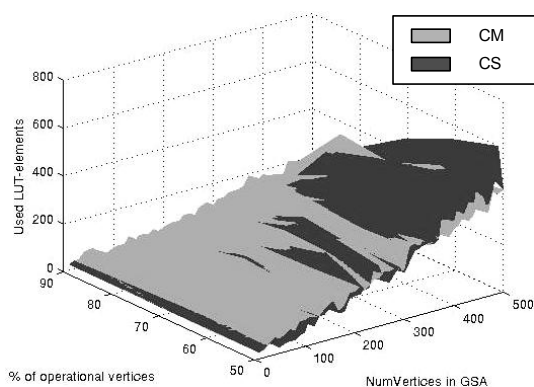


Рис. 6. Сравнение аппаратных затрат при реализации КМУУ с общей памятью и разделением кодов

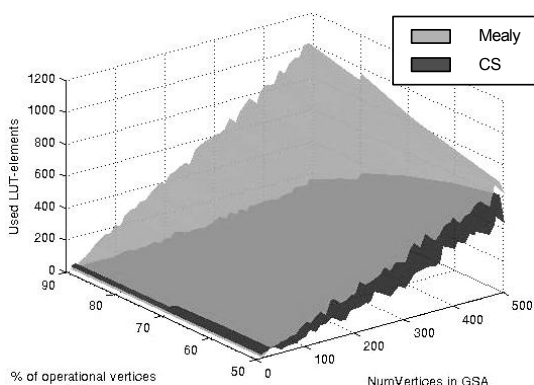


Рис. 7. Сравнение аппаратных затрат при реализации исходных ГСА в виде КМУУ с разделением кодов и в виде автоматов Мили

Реализация автомата Мили как с ростом количества операторных вершин, так и с ростом их доли в ГСА, требует всё больших ресурсов микросхемы. Если при высокой разветвленности ГСА рост требований к параметрам микросхемы у автоматов Мили и КМУУ имеет схожий характер, то при увеличении доли операторных вершин требовательность к ресурсам у автомата Мили растет гораздо быстрее (см. рис. 7).

5. Выводы

Проведенные исследования показали эффективность реализации устройств управления класса КМУУ в сравнении с автоматами с жесткой логикой на современных FPGA микросхемах.

Экспериментально определены диапазоны параметров ГСА, при которых эффективны структуры КМУУ с общей памятью, с разделением кодов.

Проведено сравнительное исследование результатов реализации автоматов с жесткой логикой и композиционных управляющих устройств на микросхемах типа FPGA.

Дальнейшие исследования связаны с анализом других структур КМУУ, их сравнением, составлением аналитических зависимостей, которые позволят определять устройство, удовлетворяющее выдвигаемым к нему критериям.

Литература: 1. *ROM-based finite state machine implementation in low cost FPGAs* / I. Garcna-Vargas, R. Senhadji-Navarro, G. Jimínez-Moreno, A. Civit-Balcells, P. Guerra-Gutiérrez // (2007) IEEE International Symposium on Industrial Electronics, art. no. 4374972. P. 2342-2347. 2. *ROM-based FSM implementation using input multiplexing in FPGA devices* / R. Senhadji-Navarro, I. Garcna-Vargas, G. Jimínez-Moreno, A. Civit-Balcells // (2004) Electronics Letters, 40 (20). P. 1249-1251. 3. *An application of functional decomposition in ROM-based FSM implementation in FPGA devices* / M. Rawski, H. Selvaraj, T. Juba // (2005) Journal of Systems Architecture, 51 (6-7). P. 424-434. 4. *Synthesis and Implementation of RAM-Based Finite State Machines in FPGAs*. / V. Sklyarov // 10th International Conference «Field-Programmable Logic and Applications: The Roadmap to Reconfigurable Computing» Proceedings, FPL 2000 Villach, Austria, August 27–30, 2000. P. 718-727. 5. *Saving power by mapping finite-state machines into embedded memory blocks in FPGAs*. / A. Tiwari, K.A. Tomko // (2004) Proceedings - Design, Automation and Test in Europe Conference and Exhibition, 2, pp. 916-921. 6. *Garcia E. Creating Finite State Machines Using True Dual-Port Fully Synchronous SelectRAM Blocks*, Xcell Journal, 2000, Issue 38. P. 36-38. 7. *Баркалов А.А.* Микропрограммное устройство управления как композиция автоматов с программируемой и жесткой логикой // Автоматика и вычислительная техника. 1983. №4. С.36-41. 8. *Баркалов А.А.* Синтез устройств управления на программируемых логических устройствах. Донецк: ДонНТУ, 2002. 262 с. 9. *Синтез микропро-*

граммных устройств управления / Баркалов А.А., Палагин А.В. К.: ИК НАН Украины, 1997. 135 с. 10. *Оптимизация устройства управления с разделением кодов* / А.А. Баркалов, Л.А. Титаренко, А.Н. Мирошкин // Управляющие системы и машины. 2010. N 1. С. 45-50. 11. *VHDL – язык описания аппаратуры*. Электронный ресурс. Режим доступа на 08.04.2011: <http://www.allhdl.ru/vhdl.php> (Загл. с экрана). 12. *Verilog – язык проектирования аппаратуры*. Электронный ресурс. Режим доступа на 08.04.2011: <http://www.allhdl.ru/verilog.php> (Загл. с экрана).

Поступила в редколлегию 25.02.2011

Рецензент: д-р техн. наук, проф.Кривуля Г.Ф.

Баркалов Александр Александрович, д-р техн. наук, профессор кафедры компьютерной инженерии ДонНТУ, профессор Университета Зеленогурского (Польша). Научные интересы: цифровые устройства управления, научная работа, спорт. Адрес: Украина, 83122, Донецк, ул. Артема, 204А, кв.105. (+38062)301-07-35

Титаренко Лариса Александровна, д-р техн. наук, профессор Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Gыта, e-mail: L.Titarenko@iie.uz.zgora.pl.

Мирошкин Александр Николаевич, ассистент кафедры компьютерной инженерии ДонНТУ. Научные интересы: цифровые устройства управления, научная работа, спорт. Адрес: Украина, 86120, Донецкая область, Макеевка, ул. Курская, 15, кв. 45.