

УДК 681.32

С.Ф. Тюрин

Пермский национальный исследовательский политехнический университет,
Пермь, Россия

АВТОМАТИЧЕСКИЙ СИНТЕЗ КОМБИНАЦИОННОГО АВТОМАТА С ПОМОЩЬЮ STATE MACHINE EDITOR СИСТЕМЫ QUARTUSII

В системе проектирования программируемых логических интегральных схем (ПЛИС) QuartusII фирмы «Альтера» имеется опция State Machine для автоматического синтеза схемы по графу конечного автомата через промежуточный этап получения программы на языке VHDL. Опции непосредственного ввода таблицы истинности в числовом виде с целью последующего автоматического синтеза схемы автором не были обнаружены, а синтез с помощью опций языков программирования VHDL и других не входил в круг интересов автора, поэтому предпринята попытка использования для этих целей опций автоматического синтеза посредством State Machine QuartusII.

Исследуется автоматический синтез комбинационного автомата в системе QuartusII с помощью State Machine Editor и языка VHDL. Оказывается, эта опция позволяет синтезировать не только конечный автомат с памятью, содержащий более двух состояний, но и комбинационный автомат. В случае комбинационного автомата граф содержит всего одно состояние. Логические функции, описывающие конечный автомат, являются метками множества дуг этого состояния. Метки получают из соответствующей таблицы истинности, которая должна быть получена перед использованием State Machine Editor QuartusII.

Символы логических операций, используемые при задании таблицы переходов-выходов конечного автомата и отображаемые после синтеза средством Map Viewer, имеют существенные отличия. Анализируются логические функции, реализуемые логическими ячейками LUT (Look Up Table), и устанавливаются особенности полученной схемы автомата с помощью средства Map Viewer.

Ключевые слова: комбинационный автомат, логический элемент – ЛЭ, логическая ячейка LUT (Look Up Table), программируемая логическая интегральная схема ПЛИС, FPGA (Field – Programmable Gate Array), язык VHDL, State Machine File, Map Viewer.

S.F. Tyurin

Perm National Research Polytechnic University, Perm, Russian Federation

AUTOMATIC SYNTHESIS OF COMBINATIONAL CIRCUITS USING QUARTUSII STATE MACHINE EDITOR

The system design of programmable logic integrated circuits (FPGAs) QuartusII firm Altera have the option State Machine for automatic synthesis scheme the graph of a finite automaton through the intermediate step of receiving a program in VHDL. Options direct input truth table as a figure for subse-

quent automatic synthesis of circuits by the author have not been found, and the synthesis via the options of programming languages VHDL and others not within the interests of the author, therefore, attempt to use for these purposes the options of automatic synthesis by the State Machine QuartusII.

The paper investigates the automatic synthesis of a combinational finite state machine using QuartusII by the State Machine Editor and language VHDL. It turns out that this option allows to synthesize not only the state machine with the memory containing more than two states, but the combination machine. In this case, the graph of a finite automaton contains only one state. Logic function describing the state machine are marks set of arcs that state. Tags prepared from the corresponding truth table, which must be obtained before using the State Machine Editor QuartusII.

Symbols of logical operations used when specifying a finite state machine and displayed after synthesis tool Map Viewer are substantial differences. Are presented the results of a study the LUT FPGA truth tables.

Keywords: Logic Cell, LUT FPGA, QuartusII, VHDL, State Machine Editor, Map Viewer.

Введение. В системе проектирования программируемых логических интегральных схем (ПЛИС) QuartusII фирмы Альтера [1, 2] имеется опция State Machine для автоматического синтеза схемы по графу конечного автомата через промежуточный этап получения программы на языке VHDL. Вызывает интерес исследование возможности синтеза более простого автомата – комбинационного. Поскольку опции непосредственного ввода таблицы истинности в числовом виде с целью последующего автоматического синтеза схемы автором не были обнаружены, а синтез с помощью опций языков программирования VHDL и др. не входил в круг интересов автора, то предпримем попытку использования для этих целей опций автоматического синтеза посредством State Machine QuartusII.

Автоматический синтез автомата, заданного в виде State Machine File. Как говорится, не мудрствуя лукаво, попробуем автоматически синтезировать не что иное, как двоичный сумматор [1, 2] (рис. 1).

x1	x2	x3	p(x1x2x3)	s(x1x2x3)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Рис. 1. Таблица истинности сумматора

Логические функции, соответствующие таблице (рис. 1) имеют вид:

$$\begin{cases} p(x_1x_2x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3, \\ s(x_1x_2x_3) = \overline{x_1}\overline{x_2}x_3 \vee \overline{x_1}x_2\overline{x_3} \vee x_1\overline{x_2}\overline{x_3} \vee x_1x_2x_3. \end{cases}$$

Попробуем выполнить автоматический синтез схемы соответствующего комбинационного автомата – полного сумматора по высокоуровневому описанию в виде графа автомата [3, 4]. Запускаем опцию State Machine и создаём граф из одной единственной вершины (рис. 2).

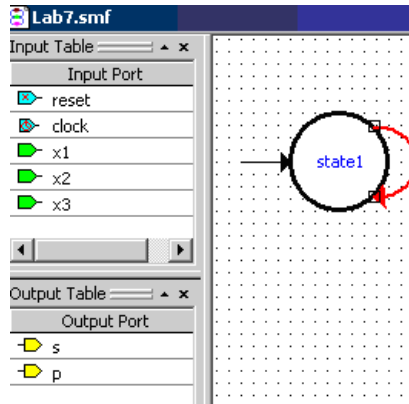


Рис. 2. Задание графа из одной вершины в State Machine

Таблицу переходов (из состояния 1 в состояние 1 – других нет!) в State Machine задаём с использованием символов следующих логических операций:

| – «ИЛИ» (вертикальная черта | в английском алфавите нажимать обратный слэш с шифтом);

& – «И»;

~ – «НЕ».

Делаем следующим образом (рис. 3).

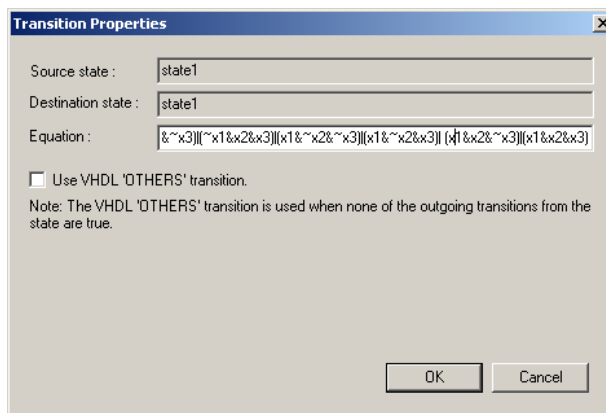


Рис. 3. Таблица переходов комбинационного автомата

Тогда итоговый граф будет выглядеть следующим образом (рис. 4).

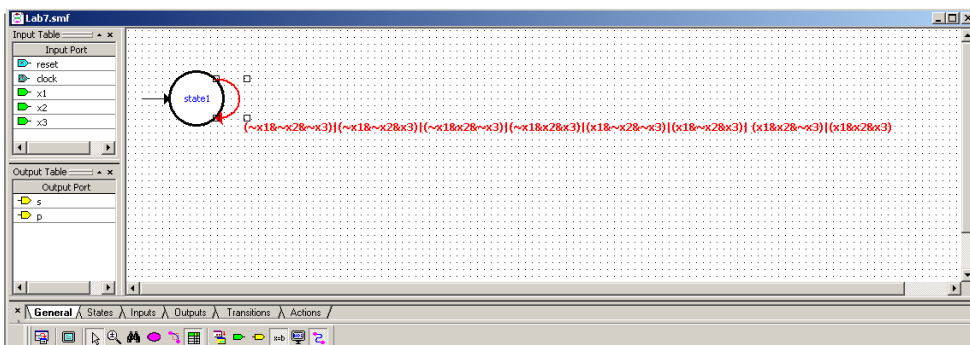


Рис. 4. Граф комбинационного автомата с логической функцией переходов из состояния 5 в состояние 1

Таблицу истинности в State Machine задаём так (рис. 5).

	Output Port	Output Value	Additional Condi...
1	s	0	$\sim x1 \& \sim x2 \& \sim x3$
2	s	1	$\sim x1 \& \sim x2 \& x3$
3	s	1	$\sim x1 \& x2 \& \sim x3$
4	s	0	$\sim x1 \& x2 \& x3$
5	s	1	$x1 \& \sim x2 \& \sim x3$
6	s	0	$\sim x1 \& x2 \& \sim x3$
7	s	0	$x1 \& x2 \& \sim x3$
8	s	1	$x1 \& x2 \& x3$
9	p	0	$\sim x1 \& \sim x2 \& \sim x3$
10	p	0	$\sim x1 \& \sim x2 \& x3$
11	p	0	$\sim x1 \& x2 \& \sim x3$
12	p	1	$\sim x1 \& x2 \& x3$
13	p	0	$x1 \& \sim x2 \& \sim x3$

Рис. 5. Задание таблицы истинности

Итак, входные наборы приходится повторять два раза – столько функций. Конечно, гораздо удобнее было бы задавать в числовом виде, а не формулами.

Компиляция комбинационного автомата, заданного в виде State Machine File. Получаем в начале VHDL-файл (рис. 6).

```

1 // Copyright (C) 1991-2009 Altera Corporation
2 // Your use of Altera Corporation's design tools, logic functions
3 // and other software and tools, and its AMPF partner logic
4 // functions, and any output files from any of the foregoing
5 // (including device programming or simulation files), and any
6 // associated documentation or information are expressly subject
7 // to the terms and conditions of the Altera Program License
8 // Subscription Agreement, Altera MegaCore Function License
9 // Agreement, or other applicable license agreement, including,
10 // without limitation, that your use is for the sole purpose of
11 // programming logic devices manufactured by Altera and sold by
12 // Altera or its authorized distributors. Please refer to the
13 // applicable agreement for further details.
14
15 // Generated by Quartus II Version 9.0 Build 132 02/25/2009 SJ Web Edition
16 // Created on Thu Feb 19 18:40:44 2015
17
18 // synthesis message_off 10175
19
20 `timescale 1ns/1ns
21
22 module Lab10 (

```

Рис. 6. Фрагмент VHDL-файла, созданного по графу в State Machine

Проводим компиляцию VHDL-файла (рис. 7).

Flow Summary	
Flow Status	Successful – Thu Feb 19 18:41:10 2015
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	Lab10
Top-level Entity Name	Lab10
Family	Cyclone II
Device	ET2C5AF256A7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	2/4,608 (<1%)
Total combinational functions	2/4,608 (<1%)
Dedicated logic registers	0/4,608 (0%)
Total registers	0
Total pins	7/158 (4%)
Total virtual pins	0
Total memory bits	0/119,808 (0%)
Embedded Multiplier 9-bit elements	0/26 (0%)
Total PLLs	0/2 (0%)

Рис. 7. Результаты компиляции VHDL-файла, созданного по графу в State Machine

Далее просматриваем результаты с помощью средства Map Viewer (рис. 8).

На рис. 8 указаны шестнадцатеричные коды настройки логических элементов (ячеек) LOGIC CELL COMB. Каждая логическая ячейка содержит постоянное запоминающее устройство ПЗУ, называемое производителями LUT (Look Up Table), вероятно, потому, что туда загружается таблица истинности требуемой логической функции [5–10]. При выделении соответствующей ячейки отображается логическая функция (рис. 9,10).

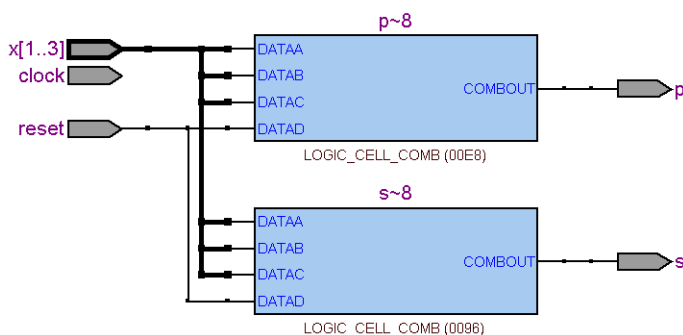


Рис. 8. Map Viewer схемы автомата, созданной по графу State Machine

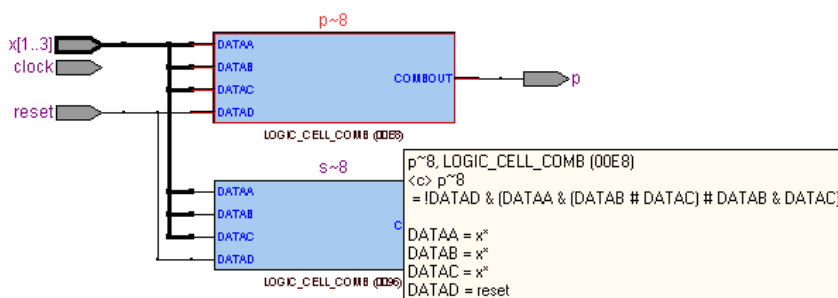


Рис. 9. Map Viewer с отображением логической функции p

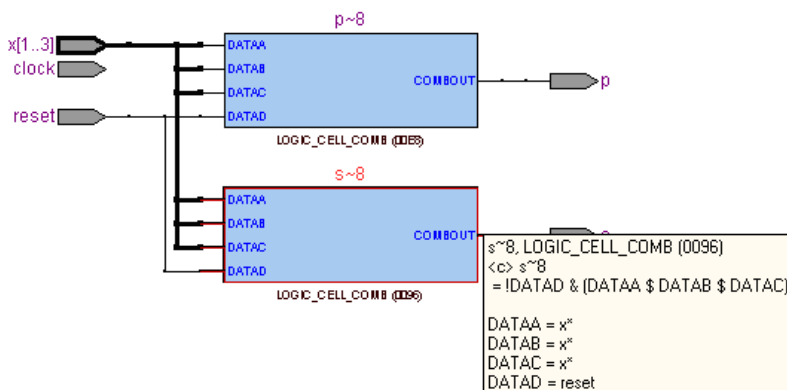


Рис. 10. Map Viewer с отображением логической функции s

Вызывает некоторое недоумение различие символов операций с теми, что были использованы при настройке графа State Machine. Так, инверсия на рис. 9, 10 обозначена восклицательным знаком, дизъюнкция – решёткой. Очевидно, что \$ на рис. 10 – это сумма по модулю два (Исключающее ИЛИ).

Расшифровка шестнадцатеричных кодов логических функций комбинационной схемы, автоматически синтезированной в State Machine File. Расшифруем логическую функцию. Так, для настройки 0096 получаем (рис. 11).

D Reset	C X3	B X2	A X1	f(DCBA)=
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Рис. 11. Таблица истинности функции s, настройка 0096

Видим, что по умолчанию задан активный уровень сигнала сброса Reset – единичный. Таким образом, действительно задана функция сложения по модулю два – функция суммы:

$$s(x_1x_2x_3) = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3.$$

Расшифруем логическую функцию и для настройки 00E8 получаем (рис. 12).

D Reset	C X3	B X2	A X1	f(DCBA)=
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Рис. 12. Таблица истинности функции p, настройка 00E8

Таким образом, действительно задана мажоритарная логическая функция – функция переноса:

$$p(x_1x_2x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3.$$

Так и есть, функции соответствуют рис. 1 с учётом нулевого сигнала сброса.

Выводы. Таким образом, использование графа автомата с одним состоянием позволяет провести автоматический синтез схемы комбинационного автомата по таблице истинности, задаваемой фактически в виде формул соответствующих конъюнкций. Причём её необходимо повторять столько раз, сколько имеется выходных функций. Символы логических операций, используемые при задании графа и отображаемые после синтеза средством Map Viewer, имеют существенные отличия.

В дальнейшем целесообразно установить имеющиеся ограничения на количество входных переменных и выходных функций при подобном задании комбинационного автомата, а также попытаться найти возможности задания таблиц истинности в более удобном – числовом виде, возможно, и с использованием языков описания аппаратных средств типа VHDL.

Библиографический список

1. Аляев Ю.А., Тюрин С.Ф. Дискретная математика и математическая логика. – М.: Финансы и статистика, 2006. – 357 с.
2. Тюрин С.Ф., Аляев Ю.А. Дискретная математика: практическая дискретная математика и математическая логика. – М.: Финансы и статистика, 2010. – 394 с.
3. Тюрин С.Ф., Громов О.А., Греков А.В. Реализация цифровых автоматов в системе Quartus фирмы Altera: лабораторный практикум. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2011. – 133 с.
4. Quartus II Help v11.1 > enum_encoding VHDL Synthesis Attribute [Электронный ресурс]. – URL: http://quartushelp.altera.com/11.1/mergedProjects/hdl/vhdl/vhdl_file_dir_enum_encoding.htm (дата обращения: 12.01.2015).
5. Цыбин С. Программируемая коммутация ПЛИС: взгляд изнутри [Электронный ресурс]. – URL: http://www.kit-e.ru/articles/plis/2010_11_56.php (дата обращения: 16.12.2014).

6. Тюрин С.Ф. Расшифровка кодов настроек автомата, заданного в виде схемы системы QUARTUS II // Проектирование и технология электронных средств. – 2015. – № 4. – С. 25–28.

7. Тюрин С.Ф. Анализ настроек логических элементов при проектировании автомата в системе QUARTUSII // В мире научных открытий. – 2015. – № 4 (64). – С. 437–453.

8. Тюрин С.Ф. Дерево транзисторов для реализации систем логических функций // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2015. – № 2 (14). – С. 37–45.

9. Tyurin S.F., Grekov A.V., Gromov O.A. The principle of recovery logic FPGA for critical applications by adapting // World Applied Sciences Journal. – 2013. – № 26(3) – P. 328–332. DOI: 10.5829/idosi.wasj.2013.26.03.13474

10. Tyurin S.F., Gromov O.A. A residual basis search algorithm of fault-tolerant programmable logic integrated circuits // Russian Electrical Engineering. – 2013. – 84 (11). – P. 647–651. DOI: 10.3103/S1068371213110163

References

1. Aliaev Iu.A., Tiurin S.F. Diskretnaia matematika i matematicheskaia logika [Discrete mathematics and mathematical logic]. Moscow: Finansy i statistika, 2006. 357 p.

2. Tiurin S.F., Aliaev Iu.A. Diskretnaia matematika: prakticheskaia diskretnaia matematika i matematicheskaia logika [Discrete Mathematics: Practical discrete mathematics and mathematical logic]. Moscow: Finansy i statistika, 2010. 394 p.

3. Tiurin S.F., Gromov O.A., Grekov A.V. Realizatsiia tsifrovyykh avtomatov v sisteme Quartus firmy Altera: laboratornyi praktikum [The implementation of digital machines in the Quartus system Altera company: laboratory-based practical]. *Permskii natsional'nyi issledovatel'skii politekhnicheskii universitet*, 2011. 133 p.

4. Quartus II Help v11.1 > enum_encoding VHDL Synthesis Attribute, available at: http://quartushelp.altera.com/11.1/mergedProjects/hdl/vhdl/vhdl_file_dir_enum_encoding.htm (accessed 12 January 2015).

5. Tsybin S. Programmiruemaia kommutatsiia PLIS: vzgliad iznutri [Software switching PLD: an outward glance], available at: http://www.kit-e.ru/articles/plis/2010_11_56.php (accessed 16 December 2014).

6. Tiurin S.F. Rasshifrovka kodov nastroek avtomata, zadannogo v vide skhemy sistemy QUARTUS II [Deciphering of the machine setting codes, specified in the form of QUARTUS II schemes]. *Proektirovanie i tehnologiiia elektronnykh sredstv*, 2015, no. 4, pp. 25-28.

7. Tiurin S.F. Analiz nastroek logicheskikh elementov pri proektirovanii avtomata v sisteme QUARTUSII [Analysis of logic elements settings while the machine design in QUARTUSII system]. *V mire nauchnykh otkrytii*, 2015, no. 4 (64), pp. 437-453.

8. Tiurin S.F. Derevo tranzistorov dlia realizatsii sistem logicheskikh funktsii [The transistors tree for the logic functions systems implementation]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniia*, 2015, no. 2 (14), pp. 37-45.

9. Tyurin S.F., Grekov A.V., Gromov O.A. The principle of recovery logic FPGA for critical applications by adapting. *World Applied Sciences Journal*, 2013, no. 26(3), pp. 328-332. DOI: 10.5829/idosi.wasj.2013.26.03.13474

10. Tyurin S.F., Gromov O.A. A residual basis search algorithm of fault-tolerant programmable logic integrated circuits. *Russian Electrical Engineering*, 2013, no. 84 (11), pp. 647-651. DOI: 10.3103/S1068371213110163

Сведения об авторе

Тюрин Сергей Феофентович (Пермь, Россия) – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор кафедры автоматизации и телемеханики Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: tyurinsergfeo@yandex.ru).

About the author

Tyurin Sergey Feofentovich (Perm, Russian Federation) is Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: tyurinsergfeo@yandex.ru).

Получено 06.07.2015