

УДК 681.32

Моделирование троированного мажоритарного элемента в Quartus Prime State Machine

А. В. Греков

Пермский военный институт войск национальной гвардии Российской Федерации
Россия, 614112, г. Пермь, ул. Гремячий Лог, 1
temagre@ieee.org; +7-909-118-45-82

Для создания высоконадежных цифровых автоматов на основе программируемых логических интегральных схем исследуется мажоритарный элемент, основанный на полном сумматоре, в среде Quartus Prime State Machine. Для этого к ранее разработанному графу автомата добавляются две новые группы входов. Моделирование отказа одним из трех мажоритаров выполняется путем указания соответствующей константы в одной из трех входных групп. Проведена оценка функционирования разработанного устройства.

Ключевые слова: *Quartus Prime; VHDL; мажоритирование; LUT FPGA; State Machine Editor; Map Viewer*

DOI: 10.17072/1993-0550-2021-1-57-60

Введение

Система проектирования программируемых логических интегральных схем (FPGA) Intel Quartus Prime [1] имеет опцию State Machine для автоматического синтеза схемы в соответствии с графом конечного автомата через промежуточный этап получения программы на языке VHDL.

Представляет интерес исследование возможности синтеза троированного мажоритарного элемента [2, 3].

Для этих целей используется автоматический синтез конечного автомата по графу переходов.

1. Автоматический синтез

Попробуем автоматически синтезировать двоичный сумматор [3, 4] (табл. 1).

Таблица 1. Таблица истинности сумматора

<i>a</i>	<i>b</i>	<i>c</i>	<i>p1(abc)</i>	<i>p2(abc)</i>	<i>p3(abc)</i>
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Логические функции, соответствующие табл. 1, имеют вид:

$$\left\{ \begin{array}{l} p1(abc) = ab \vee ac \vee bc \\ p2(abc) = ab \vee ac \vee bc \\ p3(abc) = ab \vee ac \vee bc \end{array} \right.$$

Попробуем выполнить автоматический синтез схемы соответствующего комбинационного автомата – полного сумматора по высокоуровневому описанию в виде графа автомата [3, 4]. Запускаем опцию State Machine и создаем граф из одной вершины (рис. 1).



Рис. 1. Задание графа из одной вершины в State Machine

Таблицу переходов из состояния 1 в состояние 1 в State Machine задаем с использованием символов следующих логических операций:

- | – "ИЛИ";
- & – "И";
- ~ – "НЕ".

Настройка выполняется в окне Transition Properties (рис. 2).

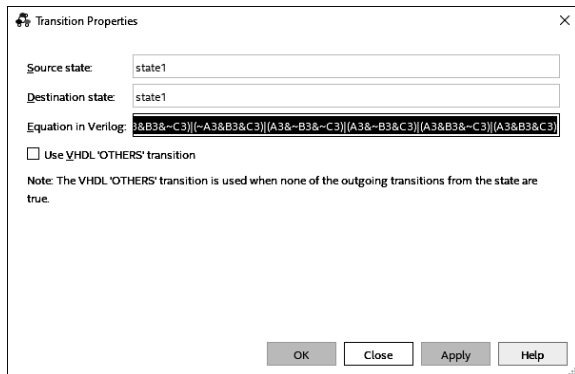


Рис. 2. Таблица переходов комбинационного автомата

Тогда итоговый граф будет выглядеть следующим образом (рис. 3):

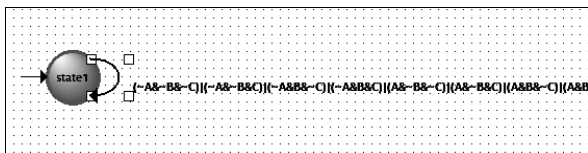


Рис. 3. Граф комбинационного автомата с логической функцией переходов из состояния 1 в состояние 1

Таблицу истинности в State Machine задаем во вкладке Actions окна State Properties (рис. 4).

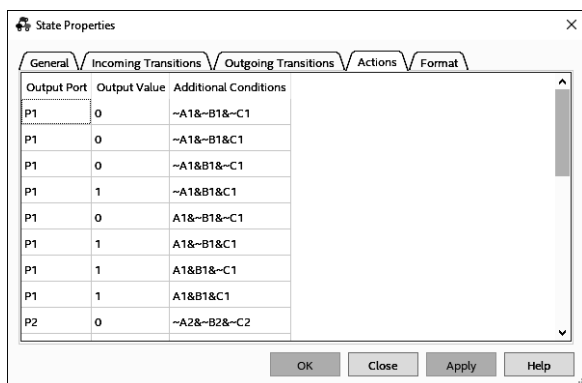


Рис. 4. Задание таблицы истинности

Входные наборы приходится повторять два раза – по числу используемых функций.

2. Компиляция комбинационного автомата, заданного в виде State Machine File

Необходимо сначала получить VHDL-файл (рис. 5).

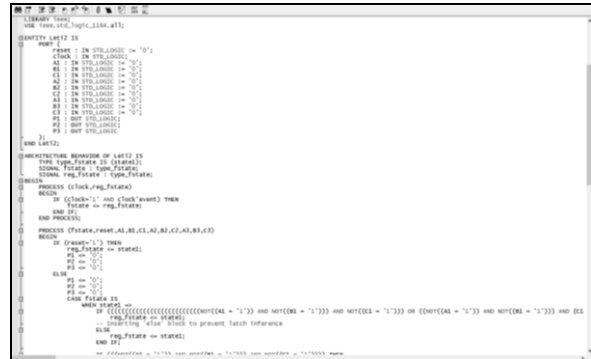


Рис. 5. Фрагмент VHDL-файла, созданного по графу в State Machine

Проводим компиляцию VHDL-файла (рис. 6).

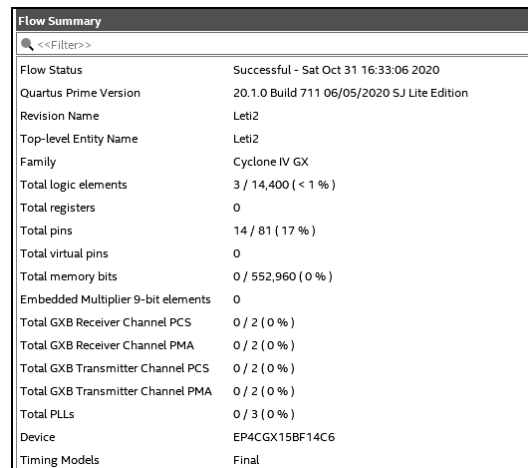


Рис. 6. Результаты компиляции VHDL-файла, созданного по графу в State Machine

Просмотр результатов осуществляется с помощью средства Map Viewer (рис. 7).

На рис. 7 показаны логические элементы (ячейки) 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]. При выделении соответствующей ячейки отображается логическая функция (рис. 8–10).

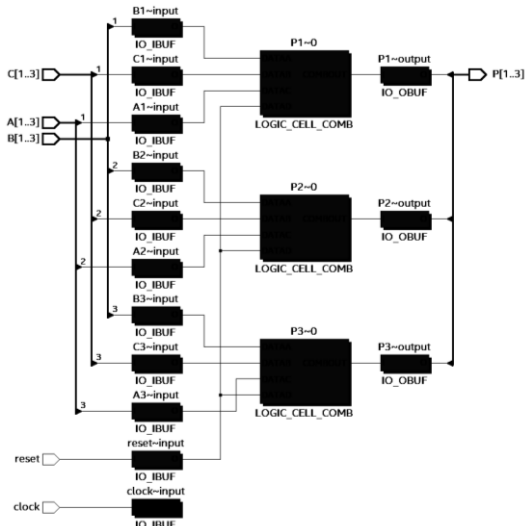


Рис. 7. Map Viewer схемы автомата, созданной по графу State Machine

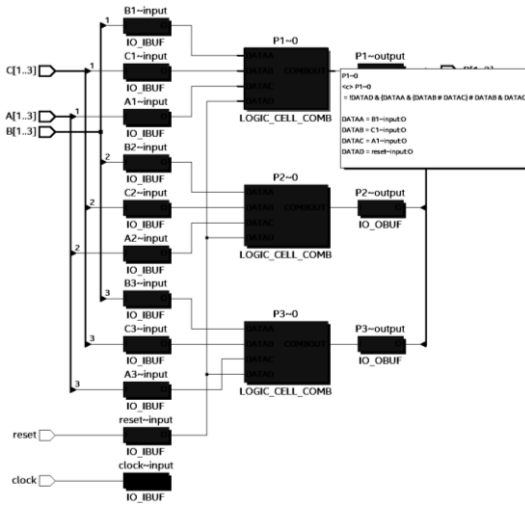


Рис. 8. Map Viewer с отображением логической функции p1

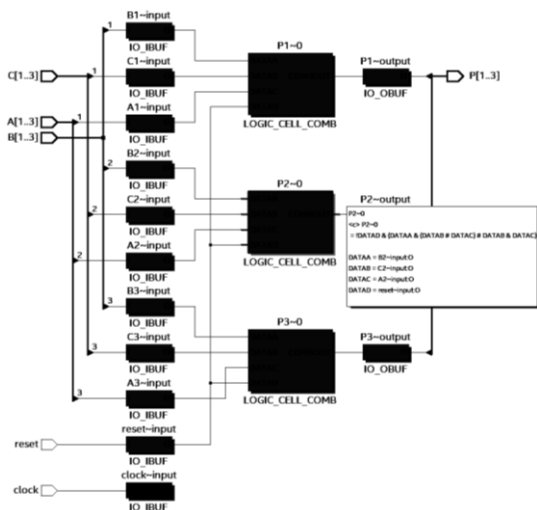


Рис. 9. Map Viewer с отображением логической функции p2

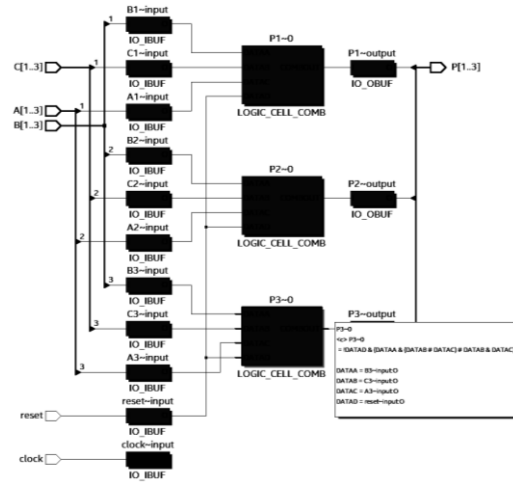


Рис. 10. Map Viewer с отображением логической функции p3

Необходимо отметить имеющееся различие символов операций с теми, что были использованы при настройке графа в State Machine. Так, инверсия на рис. 8–10 обозначена восклицательным знаком, дизъюнкция – решеткой. Очевидно, что \$ на рис. 8 – это сумма по модулю два (Исключающее ИЛИ).

Шестнадцатеричные коды настройки логических элементов (ячеек) LOGIC CELL COMB показаны на рис. 11.

3. Расшифровка шестнадцатеричных кодов логических функций комбинационной схемы, автоматически синтезированной в State Machine File

Расшифруем логическую функцию и для настройки 00E8 получаем (рис. 11).

	DATAD	DATAC	DATAB	DATAA	OUT
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	1
8	0	1	1	1	1
9	1	0	0	0	0
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

Рис. 11. Таблица истинности функции p1, настройка 00E8

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

$$p1(abc)=abVbcVbc.$$

Действительно, функция соответствует табл. 1 с учетом нулевого сигнала сброса.

Временные диаграммы окна Waveform Editor с заданными значениями входных сигналов показаны на рис. 12.

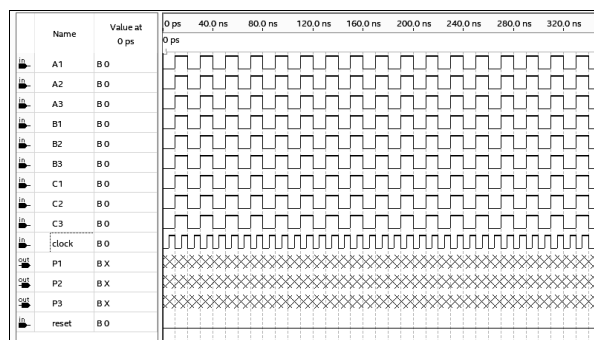


Рис. 12. Окно Waveform Editor с заданными значениями входных сигналов

Выводы

Таким образом, использование графа автомата с одним состоянием позволяет провести автоматический синтез схемы комбинационного автомата по таблице истинности, задаваемой в виде формул соответствующих конъюнкций. Причем ее необходимо повторять столько раз, сколько имеется выходных функций.

Символы логических операций, используемые при задании графа, имеют существенные отличия от символов, отображаемых средством Map Viewer по результатам синтеза.

Simulation of a tripled majority voter by Quartus Prime State Machine

A. V. Grekov

Perm Military Institute of National Guard Troops; 1, Gremyachy Log st., Perm, 614112, Russia
temagre@ieee.org; +7-909-118-45-82

A triple majority element based on a full adder in the Quartus Prime State Machine is investigated to create highly reliable FPGA-based digital automata. For this purpose, two new groups of inputs are added to the previously developed automaton graph. Modeling the failure of one of the three majorities is performed by specifying the corresponding constant in one of the three input groups. The performance indicators of the developed device are evaluated.

Keywords: *Quartus Prime; VHDL; triple majority element; LUT FPGA; State Machine Editor; Map Viewer.*

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.

Список литературы

1. *Intel Quartus Prime Pro Edition Help version 20.3 > enum_encoding VHDL Synthesis Attribute.* Available at: https://www.intel.com/content/www/us/en/programmable/quartushelp/current/index.htm#hdl/vhdl/vhdl_file_dir_enum_encoding.htm (Accessed 20 November 2020).
2. *Тюрин С.Ф.* Надежность систем автоматизации: учеб. пособие. Перм. нац. исслед. политехн. ун-т. Пермь: Изд-во ПНИПУ, 2012. 262 с.
3. *Grekov A.V., Tyurin S.F.* Throughput Logic Simulation by Quartus Prime State Machine. Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIconRus 2020-January. P. 127–129. DOI: 10.1109/EIconRus49466.2020.9039474.
4. *Тюрин С.Ф., Греков А.В., Громов О.А.* Реализация цифровых автоматов в системе Quartus фирмы Altera: учеб. пособие. Пермь: Изд-во Перм. гос. техн. ун-та, 2011. 134 с.
5. *Тюрин С.Ф.* Анализ настроек логических элементов при проектировании конечного автомата в системе QUARTUS II // Наука и технологические разработки. 2015. Т. 94, № 2. С. 17–27.