

Особенности кодов настройки LUT FPGA фирмы Intel

Сергей Ф. Тюрин, Пермский национальный исследовательский политехнический университет, Пермь, Россия
Андрей С. Прохоров, Пермский национальный исследовательский политехнический университет, Пермь, Россия



Сергей Ф. Тюрин



Андрей С.
Прохоров

Резюме. В настоящее время в цифровой схемотехнике широко применяются программируемые логические интегральные схемы (ПЛИС), в которых настраиваются функции логических элементов и их связи. Это определяется конфигурационным файлом, который загружается в ячейки конфигурационной памяти (это оперативная память статического типа) ПЛИС из внешней памяти. Сама логика реализуется в так называемых LUT (Look Up Table), представляющих собой мультиплексор, реализующий ячейку памяти и построенный на передающих транзисторах, и представляющий собой дерево, которое активируется на конкретном наборе переменных. Настройка представляет собой данные мультиплексора, поэтому на выход дерева передается значение логической (переключательной) функции на данном конкретном наборе. Оказывается, соответствующий код настройки LUT можно расшифровать и использовать при анализе результатов синтеза в системе Quartus II фирмы Альтера, которая в настоящий момент приобретена фирмой Интел. Теперь фирма Интел специализируется также и на производстве ПЛИС типа FPGA (Field – Programmable Gate Array). Рассматривается пример синтеза простого конечного комбинационного автомата, реализующего так называемую мажоритарную функцию или функцию голосования по большинству голосов (выбор 2 из 3-х). Эта функция равна единице в случае, если большинство переменных равно единице. Выполняется синтез схемы реализации мажоритарной функции в системе Quartus II, формирующей специальный файл BDF (Block Diagram / Schematic File). Исследуются особенности полученной схемы с помощью средства Map Viewer. На соответствующей схеме указываются коды настройки LUT (Logic Cell Comb) на реализацию заданной функции в виде четырехзначных шестнадцатеричных кодов. Расшифровываются коды настройки логических элементов типа LUT ПЛИС типа FPGA, описывающие содержимое соответствующих таблиц истинности функций, зависящих от входного переменного автомата. Показывается изменение кодов в процессе оптимизации схемы, выполняемой системой Quartus II, с возможным изменением порядка следования переменных и соответствия входам LUT с четырьмя входами, но сама логическая функция не изменяется. При использовании ПЛИС Stratix IIGX, имеющих так называемые адаптивные логические модули АЛМ, у которых 6 входов, Quartus II использует 64 битные коды (восьмизначные шестнадцатеричные коды). Рассматриваются особенности соответствующего кодирования.

Ключевые слова: комбинационный автомат, мажоритарная функция или функция голосования по большинству голосов (выбор 2 из 3-х), логический элемент – LUT (Look Up Table), программируемые логические интегральные схемы – ПЛИС типа FPGA (Field – Programmable Gate Array), настройки логических элементов (Logic Cell Comb), адаптивные логические модули – АЛМ.

Формат цитирования: Тюрин С.Ф., Прохоров А.С. Особенности кодов настройки LUT FPGA фирмы Intel // Надежность. 2017, Т.17, №2. С. 11-16. DOI: 10.21683/1729-2640-2017-17-2-11-16

1. Введение

Программируемые логические интегральные схемы – (ПЛИС) типа FPGA (Field-Programmable Gate Array) строятся на основе оперативных запоминающих устройств (ОЗУ), называемых LUT (Look Up Table) – просмотрными таблицами [1-3]. В указанные ОЗУ записывается конфигурационная информация – таблицы истинности требуемых логических функций.

Система QuartusII фирмы Altera (США), выпускающей ПЛИС, позволяет выполнять синтез конечных автоматов. При этом автомат задается не только схемой в виде BDF (Block Diagram / Schematic File), но и на языках описания аппаратных средств VHDL, Verilog, AHDL

и др., а также в виде графа автомата – State Machine File. QuartusII формирует коды настройки логических элементов.

Вызывает интерес расшифровка этих конфигурационных данных (Logic Cell Comb) и сравнение их с заданными логическими функциями автомата. Зададим несложный автомат и исследуем соответствующие коды настройки.

2. Получение таблицы истинности для LUT на четыре входа

Пусть необходимо синтезировать схему реализации переключательной функции (ПФ) трех переменных №132. Построим таблицу истинности (рисунок 1).

переменные			BC	f(abc)	
a	b	c			
0	0	0	0	0	2 ⁰
0	0	1	1	0	2 ¹
0	1	0	2	0	2 ²
0	1	1	3	1	2 ³
1	0	0	4	0	2 ⁴
1	0	1	5	1	2 ⁵
1	1	0	6	1	2 ⁶
1	1	1	7	1	2 ⁷

Рис. 1 – Таблица истинности ПФ №132₁₀

Эта ПФ №132₁₀ – так называемая мажоритарная функция, и она в дизъюнктивной нормальной форме (ДНФ) имеет вид:

$$f(abc) = ab \vee bc \vee ac = \overline{\overline{ab} \vee \overline{bc} \vee \overline{ac}} = \overline{\overline{ab} \cdot \overline{bc} \cdot \overline{ac}}.$$

Строим вручную схему в виде BDF (Block Diagram / Schematic File) – рисунок 2.

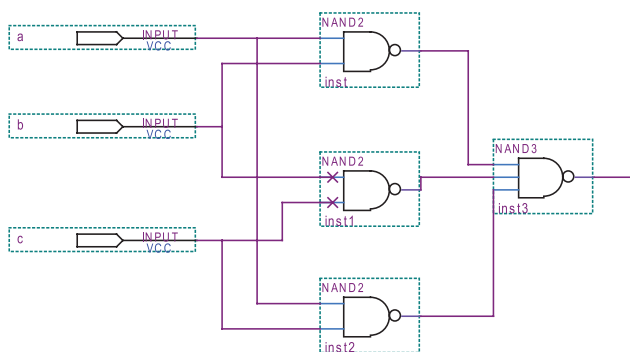


Рис. 2 – Схема электрическая принципиальная реализации мажоритарной функции в виде BDF (Block Diagram / Schematic File) QuartusII

На входах схемы рисунка 2 указана шина «+» источника питания V_{cc} – входы «подтянуты» к напряжению питания, то есть на входах в исходном состоянии – логические единицы. Компилируем проект на ПЛИС, например, EP2C5AF256A7, получаем отчет–файл (рисунок 3).

Проанализируем отчет, предоставляемый средством Map Viewer. Схема RTL (Register transfer level – уровень регистровых передач) представлена на рисунке 4.

Схема RTL фактически повторяет BDF и не содержит конфигурационной информации, но она имеется в виде

Flow Status	In progress - Sat Jan 23 12:05:46 2016
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	Lab1
Top-level Entity Name	Lab1
Family	Cyclone II
Device	EP2C5AF256A7
Timing Models	Final
Met timing requirements	N/A
Total logic elements	1 / 4,608 (< 1 %)
Total combinational functions	1 / 4,608 (< 1 %)
Dedicated logic registers	0 / 4,608 (0 %)
Total registers	0
Total pins	4 / 158 (3 %)
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. 3 – Результат компиляции схемы на ПЛИС EP2C5AF256A7

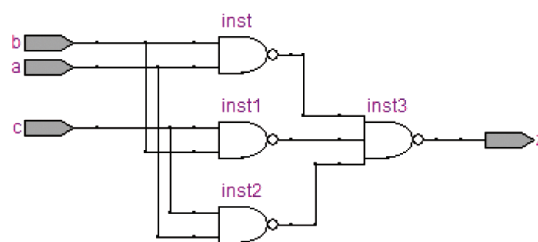


Рис. 4 – RTL файл

шестнадцатеричного кода в отчете Technology Map Viewer (Post Mapping – после размещения на «карте» ячеек ПЛИС) – рисунок 5.

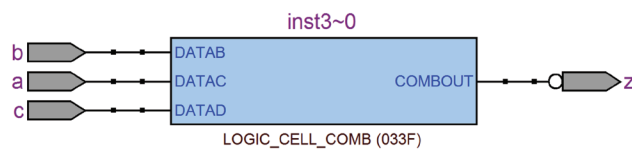


Рис. 5 – Technology Map Viewer (Post Mapping)

Таким образом, QuartusII фирмы Altera «упаковал» нашу схему в один единственный логический элемент ПЛИС. Расшифруем, сопоставляя входные переменные и входы логического элемента на 4 входа, код 033F настройки логического элемента – LOGIC_CELL_COMB (рисунок 6).

Видим, что кодировка, как и положено, идет со старших разрядов таблицы истинности, в которой входы логического элемента идут в порядке D, C, B, A, а переменные, в порядке c, a, b. Поскольку на рисунке 5 указана инверсия по выходу z, то и код 033F – это инверсия требуемой мажоритарной функции $f(abc)$. Как видим (рисунок 5), вход A не используется, поскольку функция зависит от трех переменных и QuartusII посчитал целесообразным задействовать только входы B, C, D.

Рассмотрим теперь Technology Map Viewer (Post Fitting – после «монтажа», очевидно, после оптимизации связей) – рисунок 7.

D	C	B	A	код 033F	f(abc)
c	a	b	~		
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Рис. 6 – Расшифровка кода 033F – Post Mapping

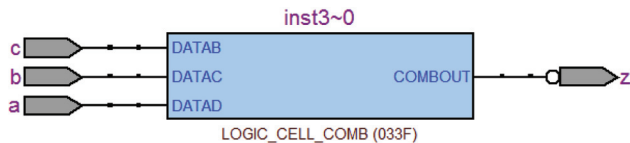


Рис. 7 – Technology Map Viewer (Post Fitting)

D	C	B	A	код 033F	f(abc)
a	b	c	~		
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Рис. 8 – Расшифровка кода 033F Post Fitting – после «монтажа»

Analysis & Synthesis Summary	
Analysis & Synthesis Status	Successful - Sun Jan 24 19:38:29 2016
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	task1-2
Top-level Entity Name	task1-2
Family	Stratix II GX
Logic utilization	N/A
Combinational ALUTs	1
Dedicated logic registers	0
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
DSP block 9-bit elements	0
Total GXB Receiver Channels	0
Total GXB Transmitter Channels	0
Total PLLs	0
Total DLLs	0

Рис. 9 – Результат компиляции на ПЛИС Stratix IIGX

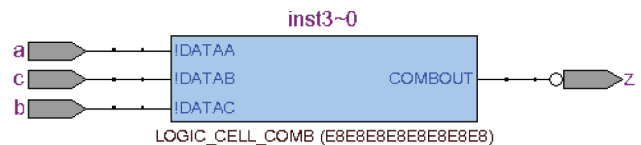


Рис. 10 – Technology Map Viewer (Post Mapping) для схемы на ПЛИС Stratix IIGX

На рисунке 7 видим, что изменилось подключение входов, вероятно это и есть результат оптимизации связей. Расшифруем код с новым подключением переменных (рисунок 8).

Кодировка 033F не меняется, так как функция большинства голосов не зависит от порядка следования переменных. Так и получается (сверху вниз) 033F.

F	E	D	C	B	A	E8E8	f(abc)
~	~	~	b	c	a		
0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1
0	0	0	0	1	0	0	1
0	0	0	0	1	1	1	0
0	0	0	1	0	0	0	1
0	0	0	1	0	1	1	0
0	0	0	1	1	0	1	0
0	0	0	1	1	1	1	0
0	0	1	0	0	0	0	1
0	0	1	0	0	1	0	1
0	0	1	0	1	0	0	1
0	0	1	0	1	1	1	0
0	0	1	1	0	0	0	1
0	0	1	1	0	1	1	0
0	0	1	1	1	0	1	0
0	0	1	1	1	1	1	0

Рис. 11 – Расшифровка части кода E8E8 (Post Mapping) – первая часть таблицы истинности ALUT

3. Получение таблицы истинности для АЛМ на шесть входов

Теперь зададим более сложную ПЛИС – Stratix IIGX, имеющую так называемые адаптивные логические модули АЛМ, у которых не 4 входа, а целых 6 входов [4-6]. Получаем отчёт (рисунок 9).

Видим, что в отчёте на рисунке 9 указаны адаптивные LUT – ALUT, их кодировка уже 64- разрядная – это 16 шестнадцатеричных цифр (рисунок 10).

F	E	D	C	B	A	E8E8	f(abc)
~	~	~	b	c	a		
0	0	0	0	0	0	1	0
0	0	0	0	0	1	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	1	0	1
0	0	0	1	0	0	1	0
0	0	0	1	0	1	0	1
0	0	0	1	1	0	0	1
0	0	0	1	1	1	0	1
0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	0
0	0	1	1	0	1	0	1
0	0	1	1	1	0	0	1
0	0	1	1	1	1	0	1

Рис. 12 – Расшифровка части кода E8E8 с младших разрядов (Post Mapping) – первая часть таблицы истинности ALUT

F	E	D	C	B	A	E8E8	f(abc)
~	~	~	b	c	a		
0	1	0	0	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	1	0	0	1	0
0	1	0	1	0	1	0	1
0	1	0	1	1	0	0	1
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	0	1	1	0
0	1	1	0	1	0	1	0
0	1	1	0	1	1	0	1
0	1	1	1	0	0	1	0
0	1	1	1	0	1	0	1
0	1	1	1	1	0	0	1
0	1	1	1	1	1	0	1

Рис. 13 – Расшифровка части кода E8E8 с младших разрядов (Post Mapping) – вторая часть таблицы истинности ALUT

Так же, как и указано выше, имеется инверсия по выходу. Расшифруем по частям настройку – код E8E8E8E8E8E8E8E8, который по идее также является инверсией мажоритарной функции, учтём дополнительные входы F, E (рисунок 11).

Однако в этом случае мажоритарная функция не реализуется, вернее, реализуется её инверсия. В чём дело? Попробуем кодировку с младших разрядов (рисунок 12).

Таким образом, несоответствие настроек ALUT требуемой функции может быть объяснено «обратной»

F	E	D	C	B	A	NOT f(abc)	f(abc)
~	~	~	b	c	a		
1	0	0	0	0	0	1	0
1	0	0	0	0	1	1	0
1	0	0	0	1	0	1	0
1	0	0	0	1	1	0	1
1	0	0	1	0	0	1	0
1	0	0	1	0	1	0	1
1	0	0	1	1	0	0	1
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	0	1	1	0	1
1	0	1	1	0	0	1	0
1	0	1	1	0	1	0	1
1	0	1	1	1	0	0	1
1	0	1	1	1	1	0	1

Рис. 14 – Расшифровка части кода E8E8 с младших разрядов (Post Mapping) – третья часть таблицы истинности ALUT

F	E	D	C	B	A	NOT f(abc)	f(abc)
~	~	~	b	c	a		
1	1	0	0	0	0	1	0
1	1	0	0	0	1	1	0
1	1	0	0	1	0	1	0
1	1	0	0	1	1	0	1
1	1	0	1	0	0	1	0
1	1	0	1	0	1	0	1
1	1	0	1	1	0	0	1
1	1	0	1	1	1	0	1
1	1	1	0	0	0	1	0
1	1	1	0	0	1	1	0
1	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1
1	1	1	1	0	0	1	0
1	1	1	1	0	1	0	1
1	1	1	1	1	0	0	1
1	1	1	1	1	1	0	1

Рис. 15 – Расшифровка части кода E8E8 с младших разрядов (Post Mapping) – четвёртая часть таблицы истинности ALUT

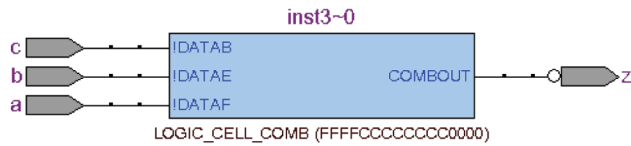


Рис. 16 – Technology Map Viewer (Post Fitting) для схемы на ПЛИС Stratix IIGX

кодировкой – с младших разрядов (с верхней части таблицы на рисунке 12), возможно это связано с большим размером таблицы истинности. Остальные три части

F	E	D	C	B	A	Код	f(abc)
a	b	~	~	c	~	FFFF	
0	0	0	0	0	0	1	0
0	0	0	0	0	1	1	0
0	0	0	0	1	0	1	0
0	0	0	0	1	1	1	0
0	0	0	1	0	0	1	0
0	0	0	1	0	1	1	0
0	0	0	1	1	0	1	0
0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	1	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	1	1	0
0	0	1	1	1	0	1	0
0	0	1	1	1	1	1	0

Рис. 17 – Расшифровка первой части кода FFFF ALUT с младших разрядов (Post Fitting)

F	E	D	C	B	A	Код	f(abc)
a	b	~	~	c	~	CCCC	
0	1	0	0	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	0	1	0	0	1
0	1	0	0	1	1	0	1
0	1	0	1	0	0	1	0
0	1	0	1	0	1	1	0
0	1	0	1	1	0	0	1
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	0	1	1	0
0	1	1	0	1	0	0	1
0	1	1	0	1	1	0	1
0	1	1	1	0	0	1	0
0	1	1	1	0	1	1	0
0	1	1	1	1	0	0	1
0	1	1	1	1	1	0	1

Рис. 18 – Расшифровка второй части кода CCCC ALUT с младших разрядов (Post Fitting)

F	E	D	C	B	A	Код	f(abc)
a	b	~	~	c	~	CCCC	
1	0	0	0	0	0	1	0
1	0	0	0	0	1	1	0
1	0	0	0	1	0	0	1
1	0	0	0	1	1	0	1
1	0	0	1	0	0	1	0
1	0	0	1	0	1	1	0
1	0	0	1	1	0	0	1
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	0	1	1	0
1	0	1	0	1	0	0	1
1	0	1	0	1	1	0	1
1	0	1	1	0	0	1	0
1	0	1	1	0	1	1	0
1	0	1	1	1	0	0	1
1	0	1	1	1	1	0	1

Рис. 19 – Расшифровка третьей части кода CCCC ALUT с младших разрядов (Post Fitting)

F	E	D	C	B	A	Код	f(abc)
a	b	~	~	c	~	0000	
1	1	0	0	0	0	0	1
1	1	0	0	0	1	0	1
1	1	0	0	1	0	0	1
1	1	0	0	1	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	0	1
1	1	0	1	1	0	0	1
1	1	0	1	1	1	0	1
1	1	1	0	0	0	0	1
1	1	1	0	0	1	0	1
1	1	1	0	1	0	0	1
1	1	1	0	1	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	0	1
1	1	1	1	1	0	0	1
1	1	1	1	1	1	0	1

Рис. 20 – Расшифровка четвёртой части кода 0000 ALUT с младших разрядов (Post Fitting)

таблицы истинности изображены на рисунках 13–15.

Рассмотрим кодировку Post Fitting для схемы на ПЛИС Stratix IIGX (рисунок 16).

Видно, что теперь переменные «сдвинулись» к старшим разрядам F, E. Поэтому и код другой. Проверим реализацию заданной функции (рисунок 17).

4. Выводы

Таким образом, шестнадцатеричные коды – конфигурационные данные (LOGIC_CELL_COMB) LUT

могут быть преобразованы в таблицы истинности соответствующих логических функций. При этом порядок следования переменных («база» переменных) может быть произвольным и меняется при оптимизации схемы – при переходе от схемы Post Mapping к Post Fitting, но сама функция остаётся неизменной.

Несоответствие кодировок (LOGIC_CELL_COMB) для ПЛИС Stratix IIGX с адаптивным логическим модулем ALM на 6 переменных (64 бита – 16 шестнадцатеричных цифр) и ПЛИС, имеющих LUT на 4 переменных (16 бит 4 шестнадцатеричных цифры) может быть объяснено использованием «обратной» кодировки. В этом случае LOGIC_CELL_COMB начинается не со старших разрядов таблицы истинности, а с младших. Описанной расшифровкой целесообразно дополнить лабораторные работы по исследованию реализации цифровых автоматов в системе Quartus [7].

Библиографический список

1. Угрюмов Евгений Павлович. Цифровая схемотехника: учебное пособие / Е. П. Угрюмов. – СПб : БХВ-Петербург, 2004. – 518 с.
2. С. Цыбин. Программируемая коммутация ПЛИС: взгляд изнутри. [Электронный ресурс]. – URL: http://www.kit-e.ru/articles/plis/2010_11_56.php (дата обращения 16.12.2014)
3. An Ultra-Low-Energy, Variation-Tolerant FPGA Architecture Using Component-Specific Mapping [Электронный ресурс]. – URL: <http://thesis.library.caltech.edu/7226/> (Дата обращения 11.11.14 г.)

4. Золотуха Р., Комолов Д. Stratix III – новое семейство FPGA фирмы Altera [Электронный ресурс]. – URL: http://kit-e.ru/assets/files/pdf/2006_12_30.pdf (дата обращения 28.11.2015)

5. Использование ресурсов ПЛИС Stratix III фирмы Altera при проектировании микропроцессорных ядер [Электронный ресурс]. – URL: http://www.kit-e.ru/articles/plis/2010_2_39.php (дата обращения: 27.11.2015).

6. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices [Электронный ресурс]. – URL: https://www.altera.com.cn/content/dam/altera-www/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf (дата обращения: 29.11.2015).

7. Тюрин С.Ф., Громов О.А., Греков А.В. Реализация цифровых автоматов в системе Quartus фирмы Altera: лабораторный практикум. – Пермь : Изд-во ПНИПУ, 2011. – 133 с., 7,25 усл. печ. л.: ил. – Библиогр.: с. 112

Сведения об авторах

Сергей Ф. Тюрин, заслуженный изобретатель Российской Федерации, доктор технических наук, профессор кафедры автоматики и телемеханики Пермского национального исследовательского политехнического университета. Пермь, Россия, e-mail: tyurinsergfeo@yandex.ru

Андрей С. Прохоров, аспирант кафедры автоматики и телемеханики Пермского национального исследовательского политехнического университета. Пермь, Россия, e-mail: nprohor007@yandex.ru.

Поступила 06.03.2016