2016

Математика. Механика. Информатика

Вып.1(32)

### удк 681.32 Диагностические возможности САПР Quartus II фирмы Altera

#### А. Н. Корнилков

Пермский национальный исследовательский политехнический университет Россия, 614990, Пермь, ул. Комсомольский пр., 29 79824947721@yandex.ru; +7-982-49-47-721

Рассматриваются вопросы диагностики проектов в САПР Quartus II фирмы Altera для ПЛИС типа FPGA (Field -Programmable Gate Array). Анализируется встраиваемый логический анализатор SignalTap II, процедуры создания теста, хранения и анализа тестовой информации. Делается вывод о том, что процедуры существующей диагностики собственно логических элементов ПЛИС, описанные в открытых источниках, не позволяет использовать их остаточную функциональность в критических приложениях.

Ключевые слова: paduoэлектронная annapamypa (РЭА); программируемые логические интегральные схемы – ПЛИС; FPGA (Field-Programmable Gate Array); логический элемент – LUT (Look Up Table); логическая ячейка (Logic Cell); САПР Quartus II.

#### Введение

В настоящее время активно развивается направление проектирования радиоэлектронной аппаратуры (РЭА) на программируемых логических интегральных схемах (ПЛИС) [1]. Одним из широко применяемых типов ПЛИС являются FPGA (Field-Programmable Gate Array) [2], содержащие множество реконфигурируемых логических блоков RLB (Reconfigurable Logic Block) [3], содержащих, в свою очередь, логические элементы LUT (Look Up Table) или Logic Cell. Количество таких элементов в некоторых ПЛИС уже более миллиона [3]. В ПЛИС "Speedster22i" фирмы Achronix имеются даже целые логические кластеры (Logic Cluster): "легкие" – Light Logic Cluster (LLC) и "тяжелые" Heavy Logic Cluster (HLC). В последние введены, например, сумматоры. Каждый RLB содержит три LLC и два HLC.

Причем еще более 10 лет назад ведущими специалистами отмечалось [4], что логические элементы занимают все гораздо меньше места, чем конфигурационная память и средства трассировки, а решение проблемы обеспечения ее надежности [5] в критических областях применения остается более сложным вопросом. В этой связи вызывает интерес рассмотрения диагностических возможностей [6], предоставляемых средствами и системами автоматизированного проектирования (САПР) ведущих фирм, производителей ПЛИС. Тем более что имеется потребность в таких приложениях ПЛИС, как и вообще в тестопригодном проектировании (DFT – Design for Testability), основанном на интерфейсе JTAG (Joint Test Action Group) и стандарте "Standard Test Access Port and Boundary-Scan Architecture" [7, 8].

В перспективе ожидается создание высоконадежных, самовосстанавливающихся, в каком-то смысле "интеллектуальных" ПЛИС [9–11]. Одним из популярных САПР является Quartus II фирмы Altera. Учебная литература по Quartus II, изданная в ПНИПУ, например [12], подробно рассматривает процедуры создания проектов, функционального и временного моделирования, но вопросы трассировки, загрузки проектов в ПЛИС и диагностики проекта "в железе" представлены недостаточно. Однако имеется достаточное число источников, описывающих эти процессы весьма неплохо, например [13, 14].

Проанализируем диагностические возможности САПР Quartus II фирмы Altera и попытаемся определить, насколько существующие средства отвечают потребностям восстановления собственно логических элементов в процессе эксплуатации ПЛИС.

<sup>©</sup> Корнилков А. Н., 2016

## 1. Встраиваемый логический анализатор SignalTap II

В состав этой системы, включающей отладочную плату ПЛИС, входят:

• память для записи отсчетов в реальном масштабе времени реализуется на блоках встроенного ОЗУ;

 в качестве средств отображения и анализа может использоваться подключенный к ПЛИС через загрузочный/отладочный кабель компьютер с соответствующим программным обеспечением;

• управление записью в память может реализоваться на логическом ядре ПЛИС;

• подключение к внутренним сигналам и выводам микросхемы осуществляется стандартными средствами САПР и вносит минимальные искажения в наблюдаемые сигналы (рис. 1).



Рис. 1. Средство Signal Tap II

Доступ к мегафункции SignalTap II возможен как с помощью редактора параметризируемых модулей MegaWizard (рис. 2), так и через специализированный пользовательский интерфейс – открытием stp-файла в окне Tools, как показано на рис. 3.



Рис 2. Редактор параметризируемых модулей MegaWizard



Рис 3. Запуск интерфейса Signal Tap Logic Analyzer II

SignalTap II позволяет:

•осуществлять запись логических состояний сигналов проекта, используя выбранный внутренний или внешний сигнал тактирования;

•выбирать сигналы проекта для наблюдения;

•подключаться к САПР Quartus II через JTAG-интерфейс;

•захватывать наблюдаемые сигналы в реальном времени.

SignalTap II входит как программный модуль в состав Quartus II, в том числе и в его свободную версию Web Edition. Отметим особо, что для использования SignalTap II в Quartus II Web Edition необходимо разрешить опцию TalkBack: Tools => Options =>Internet Connectivity => TalkBack Options =>Turn on the Quartus II software TalkBack feature.

Основные шаги работы с SignalTap II:

1) установка и настройка;

2) определение момента начала захвата данных (Data Triggering);

3) захват данных (Data Capture);

4) анализ записанных данных (Data Analysis).

Для проведения отладки проекта могут быть использованы несколько встроенных логических анализаторов (файлов с расширением .stp). Для открытия нового файла STP существуют два способа: 1) в меню Tools выбрать пункт Signal-Tap II Logic Analyzer;

2) создать новый файл через меню File =>New => Other Files => SignalTap II File.

В обоих вариантах откроется файл с именем по умолчанию STP1.stp (как обычно, разработчик может переименовать его по своему усмотрению). Можно создать несколько разных stp-файлов, но к проекту единовременно может быть подключен только один из них (пункт меню Assignments => Settings => SignalTap II Logic Analyzer => SignalTap II File name).

Пример файла STP, создаваемого через специализированный пользовательский интерфейс, приведен на рис. 4.



Рис 4. Основное окно Signal Tap Logic Analyzer II

Окно файла STP содержит несколько областей, обеспечивающих настройку тестов, их запуск, отображение и хранение результатов:

•область менеджера тестов (Instance Manager);

•область редактирования параметров (Signal Configuration);

•область выбора наблюдаемых сигналов и просмотра временных диаграмм (Waveform Viewer);

•область настройки JTAG (JTAG Chain Configuration);

•область иерархии;

•область сохранения данных.

#### 2. Создание теста

Область менеджера тестов (Instance Manager) позволяет в рамках одного файла STP создать несколько тестов (но не более 16) для проверки функционирования различных блоков отлаживаемого устройства. Для создания нового теста следует щелкнуть правой

кнопкой в поле области менеджера тестов (рис. 5) и выбрать Create Instance:

stance Harap	- A 6	B C India	iii antquat	-			 714G Chain Configuration	
	Status	ube 0	-	yit Seak t	Testan 0	Large D	Notes	- 540
		Create Indance					Device: None Detected	* San De
		Delete Instance	Del				Sa Str Hanaper & 0	6
	_	Rename Instance	2				1	
		Enable Power Up Trigg	-					
	- 11	Deable Power Lip Trig	-					
	- 11	Duplicate Trigger						
		Instance Status Help						

Рис. 5. Создание теста в области Instance Mamager

Создаваемым тестам по умолчанию присваивается имя auto\_signaltap\_x, которое в дальнейшем может быть заменено разработчиком на содержательное имя. Менеджер тестов позволяет выбрать нужный тест для установок, запуска, отображения, показывает статус каждого теста и занимаемый аппаратный ресурс, указывает на готовность SignalTap II к запуску или необходимость перекомпиляции проекта при изменениях настроек тестов. Область редактирования параметров (Signal Configuration) показана на рис. 7.

В этой области можно задать следующие параметры:

• тактовый сигнал для записи отсчетов в память логического анализатора (Sample Clock);

• количество отсчетов в записываемой выборке (Sample Depth);

• режим формирования захвата (Trigger flow control):

• позицию захвата (Trigger);

• число условий для формирования сигнала захвата (Trigger Level).

Выбор сигнала тактирования осуществляется через Node Finder. При выборе тактового сигнала предпочтение должно быть отдано глобальному сигналу. Также может быть выбран внешний сигнал тактирования, с автоматическим созданием внешнего вывода. Запись данных в память логического анализатора осуществляется по каждому положительному перепаду тактового сигнала.

#### 3. Хранение реакций на тест

Память для записи отсчетов SignalTap II в зависимости от настроек может быть представлена или в виде кольцевого буфера, или в виде сегментированного буфера (рис. 6). В эти буферы непрерывно, с частотой Sample Clock, ведется запись данных.



Рис. 6. Кольцевой и сегментированный типы буфера

Размер буферов определяется параметром Sample Depth. SignalTap II имеет два режима формирования сигнала захвата:

1) последовательностный режим (Sequ ental) – позволяет использовать и комбинировать стандартные условия захвата;

2) режим формирования условий на основе машины состояний (State-based) – позволяет создавать сложные пользовательские условия захвата.

Выбор положения позиции захвата определяет соотношение выводимых отсчетов до позиции Trigger и после нее и осуществляется посредством использования следующих установок:

• Pre – отображаются данные, записанные до момента захвата;

• Center – отображаются данные, записанные как до, так и после момента захвата;

• Post – отображаются данные, записанные после момента захвата.

При использовании сегментированного буфера заданное положение позиции захвата будет одинаковым для всех сегментов.

В качестве сигнала захвата может использоваться и внешний сигнал (порт Trigger-In), для которого задаются условия формирования сигнала захвата. Для наблюдения внешними инструментальными средствами момента захвата может применяться неиспользуемый в проекте вывод микросхемы (порт Trigger-Out), для которого задается активный уровень сигнала.

Область выбора наблюдаемых сигналов и просмотра временных диаграмм (Waveform Viewer) предоставляет возможности для выбора записываемых в кольцевой буфер сигналов (рис. 10) и формирования условий захвата для заданного количества условий Trigger Level. Также в этой области отображаются записанные в память SignalTap II значения наблюдаемых сигналов (рис. 11) для каждого из создаваемых тестов. Для выбора сигналов нужно перейти на закладку Setup и двойным щелчком правой клавиши в поле колонки Node/Name вызвать Node Finder. В нем нужно установить маску, позволяющую отображать только те сигналы, наблюдение которых возможно с помощью SignalTap II. Не все внутренние сигналы проекта могут наблюдаться средствамиSignalTap II. Сигналы, наблюдение которых невозможно:

- LE Carry Chain Outputs;
- PLL Clock Output;
- LVDS RX/LVDS TX;
- CDR  $\overline{RX}/CDR$   $T\overline{X}$ ;
- сигналы ЈТАС-интерфейса.

Установленные в окне Setup сигналы могут редактироваться: выбранные шины – разгруппировываться, отдельные сигналы – собираться в шины, для любых сигналов могут задаваться мнемонические таблицы соответствия (например, состояния управляющей шины могут отображаться в виде наименований выполняемых операций, значения семисегментного кода – в виде отображаемых цифр и т. п.).

В колонке Data Enable отмечаются сигналы, которые будут захватываться в конкретном эксперименте — это полезно для уменьшения объема требуемой для записи отсчетов памяти.

В колонке Trigger Enable отмечаются сигналы, которые будут участвовать в логике формирования сигнала захвата (Trigger).

Как уже говорилось, количество условий для формирования сигнала захвата (Trigger Level) задается в области редактирования параметров. Простейший способ формирования сигнала захвата (Trigger) в последовательностном режиме – использование базовых условий (Basic), варианты которых предлагаются при щелчке правой кнопкой в колонке Trigger Level соответствующей строки. Каждый Trigger Level выполняется только тогда, когда одновременно выполняются условия, заданные для всех сигналов, участвующих в его формировании.

Для формирования более сложных условий захвата в последовательностном режиме (например, с использованием логических операторов, операторов отношения и операторов задержки) можно использовать усложненные условия (Advanced).

В режиме формирования условий захвата на основе машины состояний разработчик может формировать сложные условия захвата – например, *n*-кратное повторение цепочки некоторых событий. В этом режиме процесс формирования захвата описывается в виде машины состояний, для которой разработчик задает условия перехода из состояния в состояние и действия, производимые в каждом состоянии. Для описания машины состояний используется простой BASIC-подобный язык (см. рис. 7).



Рис. 7. Окно Setup интерфейса Signal Tap LA II

Сигнал захвата Trigger формируется в случае выполнения любого из активных условий Trigger Level. После осуществления захвата (в соответствии с установленной позицией захвата – Pre, Center или Post) запись данных в буфер SignalTap II прекращается и содержимое буфера становится доступным для анализа. Для просмотра временных диаграмм после захвата данных нужно перейти на закладку Data (см. рис. 8, 9).

kg 20150105 1921.00 #0		duk to root line bar														
Type Alian	Hane	190 -144 -126 -112 -96 -80 -54 -48 -32 -19		35	2 4	64 30	05 11	2 128 144	1 10	176 192	298 27	4 240	256 3	212 218	304 3	9 336
5	. I th deinh dipy be															
5	. J. th. doinh. diller. dat			nnn	JUUU	nnn	nn	www	ທີ	vvvv	ທານ	nn	ллл	m	ທາກ	າມາມ
×.	. h_dcinh_diplais.DATA															
4	h_dbpine.2001_STAR															
4	. and the denie diale	LIC_INVALIATION AND AND AND AND AND AND AND AND AND AN			սո	ມດາກສະຫ			L	ມມາກາ	າມມາ			шш	டாள	ມກມ
3	. J. th. dcinh.dkim.ck															
5	ett.pre															
5	NULARI, BROUNN															
5	Ni, h., a mi, taatako															
5	. IL AN UNAR HOLD															
5	. ami_tent_secon_vaid															
5	mi, in a mi, tproard, 570															
5	NULARI, SPARA															
5	NULANUSUF															
5	NUCLINE															
5	NULANIA															
5			_													
		×														•
Di Deta	To Select															

Рис. 8. Окно Data интерфейса Signal Tap LA II



Рис. 9. Окно составления машины состояний

Формат отображаемых данных может быть задан во всплывающем меню, вызываемом щелчком правой кнопки на выбранном сигнале. Область иерархии (Hierarchy Display) (см. рис. 10) отображает иерархию сигналов, выбранных для просмотра временных диаграмм, и позволяет скрывать любые из них для удобства анализа остальных.

Hierarchy Display:	X 🔽 Data Log: 🛃
▲ 💟 ⇒	4 🕄 auto_signaltap_0
4 <b>V</b> ⇒ mi_in_a:mi_1	signal_set: 2015/01/25 16:19:21 #0
🔽 🌩 h_dk:1n:h_dk	4 🃸 trigger: 2015/01/25 16:19:21 #1
V 🗢 MY_DPRAM:dpram	kog: 2015/01/25 16:21:00 #0
	ig log: 2015/01/25 16:21:01 #0
	ing: 2015/01/25 16:21:01 #1
	kg: 2015/01/25 16:21:02 #0
	ig log: 2015/01/25 16:21:03 #0
	ing: 2015/01/25 16:21:04 #0
	dg: 2015/01/25 16:21:05 #0
	ul los 2015/01/25 16-21-05 #0

Рис. 10. Область иерархии окна Signal Tap Logic Analyzer II

Область сохранения данных (Data Log) (рис. 10) позволяет записывать тестовые последовательности для дальнейшего их просмотра средствами SignalTap II или преобразования в другие форматы (выбор записанного теста в пункте меню Data Log =>File => Export => Сохранение в выбранном формате). Преобразованные тестовые данные могут наблюдаться и анализироваться другими средствами, например, средствами редактора временных диаграмм САПР Quartus II.

Область настройки JTAG (JTAG Chain Configuration), как и для рассмотренных выше средств системной отладки, задает интерфейс компьютера и отлаживаемого устройства (см. рис. 11).

JTAG Chain Configuration:							
Hardware:	▼ Setup						
Device: None Detected	▼ Scan Chain						
<< SOF Manager: 🚊 🕖	ton/YandexDisk/od122.sof						
Attached SOF Files auto_signaltap_0							

Рис. 11. Область настройки JTAG

В этой области, не обращаясь к окну программатора Quartus II, можно производить следующие действия:

• задавать средства программирования (например, можно выбрать загрузочный/отладочный кабель USB Blaster II);

• определять состояние JTAG-цепочки и выбирать нужную для тестирования микросхему;

• выбирать программирующий файл и запускать процесс конфигурации СБИС ПЛ.

После окончания отладки логический анализатор SignalTap II может быть исключен из проекта (пункт меню Assignments =>Settings => SignalTap II Logic Analyzer).

После удаления SignalTap II из проекта следует сохранить существующую разводку проекта (пункт меню Assignments => Back-Annotate Assignments => Pin, sell, routing and device assignments) и перекомпилировать проект.

Установки всех средств системной отладки, используемых в проекте, отображаются в окне отчета Compilation Report =>Analysis & Synthesis => Debug Settings Summary.

Средства системной отладки САПР Quartus II удобны в использовании, и позволяют существенно упростить процесс верификации проектов на основе СБИС ПЛ в реальном аппаратном окружении. Любое из них может быть добавлено к проекту для осуществления физического моделирования и исключено из проекта после его завершения, при переходе от прототипа к серийной продукции.

Полученные в процессе данные (каждый полученный log) можно сохранять на диске отдельным файлом, для этого требуется в поле с интересующей диаграммой кликнуть правой кнопкой мыши и в появившемся меню выбрать Create Signal Tap List File (см. рис. 12).



Рис. 12. Создание отдельного отчета для выбранной диаграммы

После окончания отладки логический анализатор SignalTap II может быть исключен из проекта (пункт меню Assignments => Settings => SignalTap II Logic Analyzer). После удаления SignalTap II из проекта следует сохранить существующую разводку проекта (пункт меню Assignments => Back–Annotate Assignments => Pin, sell, routing and device assignments) и перекомпилировать проект.

#### Выводы

Средства системной отладки САПР Quartus II удобны в использовании и позволяют существенно упростить процесс верификации проектов на основе ПЛИС в реаль-

ном аппаратном окружении. Любое из них может быть добавлено к проекту для осуществления физического моделирования и исключено из проекта после его завершения, при переходе от прототипа к серийной продукции. Однако при всех широких возможностях доступных опций, анализ показал, что глубина существующей диагностики собственно логических элементов ПЛИС, описанная в открытых источниках, не позволяет использовать их остаточную функциональность в критических приложениях. То есть, при массированных отказах даже этого огромного множества элементов может быть недостаточно. Поэтому вопрос совершенствования такой диагностики, например с целью нахождения работоспособного поддерева LUT [10], является актуальным.

#### Список литературы

- 1. Основные производители современных ПЛИС-компьютеров и комплектующих к ним. URL: https://parallel.ru/fpga/-vendors.html (дата обращения: 27.10.2015).
- 2. *Цыбин С.* Программируемая коммутация ПЛИС: взгляд изнутри. URL: http://www.kite.ru/articles/plis/2010\_11\_56.p hp (дата обращения: 16.10.2015).
- 3. *Speedster22i* Configuration User Guide. URL:

http://www.achronix.com/wpcontent/uploads /docs/Speedster22i\_Configuration\_User\_Gui de UG033.pdf (дата обращения:29.10.2015).

- 4. *Yervant Z.* Gest editors' introduction: Design for Yield and reliability / Z.Yervant, G.Dmytris // IEEE Design & Test of Computers. May–June 2004. P. 177–182.
- ГОСТ Р 53480-2009. Надежность в технике. Термины и определения. IEC 60050 (191):1990-12 (NEQ). М.: Стандартинформ, 2010.
- 6. ГОСТ 20911-89. Техническая диагностика. Термины и определения. М.: Стандартинформ, 2019.
- 7. *IEEE* Std 1149.1-1990, "Test Access Port and Boundary-Scan Architecture", IEEE, USA, 1995.
- 8. *Авионика* марсохода Curiosity поможет найти следы жизни на красной планете. URL:http://www.russianelectronics.ru/leaderr/pe chat/60585/ (дата обращения: 20.10.2015).
- 9. *Тюрин С.Ф., Плотникова А.Ю*. Концепция "зеленой логики" // Вестник Пермского на-

ционального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2013. № 8. С. 61–72.

- Tyurin S.F., Grekov A.V., Gromov O.A. The principle of recovery logic FPGA for critical applications by adapting (3). P. 328–332. DOI: 10.5829/idosi.wasj.2013.26.03.13474.
- 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.
- 12. 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.
- Бабурин Д. Sky dynamics. Сайт об интересных технических проектах и разработках. URL: http://skydynamics.ru/electronicsfor-all/plis-prosto-o-slozhnom-chast-1.html (дата обращения: 16.10.2015).
- 14. Стешенко В.Б. ПЛИС фирмы ALTERA: элементная база, система проектирования и языки описания аппаратуры. 3-е изд. М.: Издат. дом, ДОДЕКА XXI, 2007. 576 с.

# The diagnostic capabilities of CAD Quartus II from Altera

#### A. N. Kornilkov

Perm National Research Polytechnic University, Russia, 614990, Perm, Komsomolsky Av., 29 79824947721@yandex.ru; +7-982-49-47-721

The issues of diagnosis objects in CAD Quartus II Altera's FPGA-type FPGA (Field-Programmable Gate Array). We analyze the embedded logic analyzer SignalTap II, the creation of test procedures, storage and analysis of test data. It is concluded that the existing diagnostic procedures for proper FPGA logic elements described in the public domain, does not allow the use of the residual functionality of critical applications.

**Key words:** *FPGA (Field-Programmable Gate Array); Logic Elemen; Logic Cel; LUT (Look Up Table); reliability; diagnostic; CAD Quartus II.*