

# 11. Transceiver Link Debugging Using the System Console

#### QII53029-11.1.0

В этой главе описывается, как использовать Transceiver Toolkit (инструмент приёмопередатчика) в программе Quartus<sup>®</sup>II. Transceiver Toolkit в программе Quartus II позволяет вам быстро протестировать функционирование каналов приёма и передачи и поможет вам улучшить целостность сигнала в цепи приёмопередатчика вашего проекта.

Вы можете использовать примеры проекта, доступные на сайте Altera, если хотите немедленно начать использовать Transceiver Toolkit, или вы можете создать собственный проект.

В современных высокоскоростных интерфейсах, не просто соответствовать жестким требованиям по частоте появления ошибочных битов (BER). Вы можете использовать Transceiver Toolkit в программе Quartus II для проверки и улучшения целостности сигнала в цепи приёмопередатчика на вашей плате до того, как закончите весь проект. При этом вы сэкономите время и найдёте наилучшую настройку медианы физической среды (PMA) для вашего высокоскоростного интерфейса.

Эта глава состоит из следующих секций:

- "Введение в Transceiver Toolkit"
- "Примеры проектов отладки цепи приёмопередатчика" на стр. 11-3
- "Установки тестов для цепи отладки" на стр. 11-3
- "Использование сценариев" на стр. 11-14
- "Быстрый проводник по использованию Transceiver Toolkit в программе Quartus II" на стр. 11-18

### Введение в Transceiver Toolkit

Основой среды для Transceiver Toolkit является System Console (системная консоль). Системная консоль выполняет низкоуровневую аппаратную отладку вашего проекта. Системная консоль предлагает доступ к операциям чтения и записи в ваши IP ядра, используемые в проекте.

Используйте системную консоль для начального запуска вашего РСВ и низкоуровневого тестирования.

За дополнительной информацией о системной консоли, обратитесь к главе "<u>Анализ и отладка проекта с помощью системной консоли</u>" в томе 3 Настольной книги Quartus II. Тренировочные курсы по использованию системной консоли на странице <u>Altera Training</u>.

Transceiver Toolkit позволяет вам выполнять задачи на стадии прогона, включая выполнение тестов для высокоскоростных цепей приёмопередатчиков вашего чипа. Transceiver Toolkit позволяет вам тестировать ваши высокоскоростные интерфейсы в реальном времени. Для запуска Transceiver Toolkit в главном окне Quartus II в меню Tools кликните **Transceiver Toolkit**.

### Интерфейс пользователя Transceiver Toolkit

Transceiver Toolkit имеет интуитивный GUI, который открывается из меню Tools программы Quartus II. Он разработан для того, чтобы запускаться из-под среды системной консоли. Интерфейс окна представлен четырьмя панелями. Слева - панель навигации System Explorer, показывает вам информацию о соединения в тестируемой среде, включая проекты, элементы проекта и скрипты.

После того, как вы загрузите проект в системную консоль, вся информация о проекте появляется в System Explorer. На панели Messages отображаются сообщения об ошибках и предупреждения на любое действие, которое вы выполняете на других панелях. Вы можете вводить Tcl команды в панели Tcl Console. Большинство действий, которые вы выполняете с помощью GUI, вы можете выполнить с помощью команд Tcl. Большинство команд Tcl приведены в этой главе.

Transceiver Toolkit Channel Manager GUI (менеджер каналов инструмента приёмопередатчика) состоит из трёх вкладок: Transmitter Channels (каналы передатчика), Receiver Channels (каналы приёмника) и Transceiver Links (цепь приёмопередатчика). Три этих вкладки имеют несколько контрольных кнопок, нажав на которые, вы откроете панель Auto Sweep Control (автоматический контроль развёртки) и панель EyeQ. Из этих трёх вкладок вы также можете открыть панели Control Channel (контроль каналов) и Control Link (контроль цепей), которые позволяют вам рассматривать и модифицировать настройки приёмопередатчика и теста.

Например, вы можете открыть Link Control Channel.

За дополнительной информацией обратитесь к разделу помощи Quartus II About the Transceiver Toolkit.

#### Transceiver Auto Sweep

Вы можете изменять настройки РМА вашего приёмопередатчика и запускать тесты автоматически с помощью автоматической развёртки. Вы можете сохранять историю тестовых запусков и извлекать запись о самых лучших настройках. Вы можете использовать эти настройки в вашем окончательном проекте.

За дополнительной информацией обратитесь к разделу помощи Quartus II *Transceiver Auto Sweep Panel*.

### Transceiver EyeQ

Вы можете определить целостность сигнала с помощью средства EyeQ. Средство EyeQ в Transceiver Toolkit позволяет вам создавать U-образную кривую или индикаторную диаграмму (Stratix V), чтобы иметь другие (отличные от BER) способы измерения качества сигнала. После того, как вы запустите средство EyeQ, вы увидите данные в панели **Report** в Transceiver Toolkit, и сможете экспортировать в формат переменных, разделяемых запятой (.csv) для дальнейшего анализа.

За дополнительной информацией обратитесь к разделу помощи Quartus II Working with the Transceiver Toolkit.

За дополнительной информацией обратитесь к AN 605: <u>Использование</u> <u>средства схемы мониторинга качества сигнала на чипе (EyeQ) в</u> <u>приёмопередатчиках Stratix IV</u>.

### Control Links

Вы можете протестировать цепи каналов приёмника и передатчика вашего проекта в ручном режиме с помощью средства контроля каналов. Панель channel

control позволяет вам наблюдать и редактировать вручную настройки для каналов передатчика и приёмника, пока каналы находятся в режиме запуска.

За дополнительной информацией обратитесь к разделу помощи Quartus II Working with the Transceiver Toolkit.

## Примеры проектов отладки цепи приёмопередатчика

Altera предлагает примеры проектов, чтобы помочь вам с настройками и методами работы с Transceiver Toolkit. Чтобы подробнее узнать о версии программы Quartus II, использованной для создания этого примера, выбранном чипе и детальной информации об отладочной плате, обратитесь к файлу **readme.txt** в каждом примере. Каждый пример проверен и протестирован в версии программы Quartus II, указанной в файле **readme.txt**. Однако вы сможете использовать эти примеры и в более поздних версиях программы Quartus II.

Если вы перекомпилировали примеры проекта для другой платы, обратитесь к главе "Изменение назначений выводов" на стр. 11-18, чтобы узнать какие назначения для выводов вам нужно поменять.

Загрузите примеры проектов со страницы <u>Примеры проектов отладки на чипе</u> с веб-сайта Altera.

Быстрый проводник по использованию примеров проекта с Transceiver Toolkit в главе "Работа с примерами проекта" на стр. 11-19.

#### Установки тестов для цепи отладки

Тестирование целостности сигнала для высокоскоростных цепей приёмопередатчика состоит из использования пакетов данных, таких как псевдо случайная двоичная последовательность (PRBS). Несмотря на TO. что последовательность относится К случайной, она обладает специфическими свойствами, которые вы сможете использовать для измерения качества цепи. В примерах проектов, доступных с сайта Altera, пакет данных генерируется генератором пакетов, затем передаётся в передатчик. Приёмопередатчик на дальнем конце возвращает пакет обратно в таком виде, каким он был принят приёмником в приёмопередатчике. Полученные данные проверяются на наличие ошибочных битов.

На рис. 11-1 и 11-2 показаны примеры тестовых установок для инструмента отладки цепи приёмопередатчика. На рисунках приведены установки, которые очень похожи на те примеры проектов, которые вы сможете загрузить со страницы <u>Примеры</u> <u>проектов отладки на чипе</u> с веб-сайта Altera. Вам также нужно иметь передатчик в одном и приёмник в другом чипе FPGA.





На рис. 11-2 показаны простые тестовые установки для второго примера проекта, описанного в этой главе, за исключением того, что здесь набор из четырёх приёмопередатчиков и приёмников, вместо одного.

Figure 11-2. Transceiver Link Debugging Tool Test Setup (Four Channels)



Перевод: Егоров А.В., 2012 г.

Примеры проектов используют инструмент системной интеграции Qsys и состоят из следующий компонентов:

- Собственное ядро PHY IP и ядро с малой задержкой PHY IP
- Avalon-ST генератор пакета данных
- Avalon-ST проверочный блок пакета данных
- Мастер мост JTAG-to-Avalon

Быстрый проводник по использованию примеров проекта с Transceiver Toolkit в главе "Работа с примерами проекта" на стр. 11-19.

#### Собственное ядро РНҮ ІР

Вы можете использовать собственное ядро PHY IP для тестирования приёмопередатчиков всевозможной ширины параллельных данных в этих примерах проектов. Вы можете сконфигурировать собственное ядро PHY IP на 8, 10, 16, 20, 32 или 40-бит. Инструмент развёртки запрещает выравнивание слов в развёртке, которое разрешено в простых временных ограничениях. Вы также можете использовать по необходимости компонент IP адаптер формата данных (Data Format Adapter IP). Также вы можете иметь один или несколько каналов в вашем проекте.

Вы используете Qsys для генерирования собственного ядра PHY IP. Это ядро в примерах проектов, которые вы загрузили со страницы <u>Примеры проектов отладки на</u> <u>чипе</u> с веб-сайта Altera, было сгенерировано для чипов Stratix IV и Stratix V.

Для использования собственного ядра PHY IP в Transceiver Toolkit выполните следующие пункты:

- 1. Установите следующие параметры для вашего проекта:
  - Количество линий (Number of lanes)
  - Размер связанных групп (Bonded group size)
  - Последовательный коэффициент (Serialization factor)
  - Скорость передачи данных (Data rate)
  - Входная тактовая частота (Input clock frequency)
- 2. Включите Интерфейсы данных Avalon (Avalon data interfaces)
- 3. Запретите **8В/10В**
- 4. Установите Режим выравнивания слов (Word alignment mode) вручную
- 5. Запретите согласование скорости с FIFO (rate match FIFO)
- 6. Запретите блок упорядочения байтов (byte ordering block)

За подробной информацией о настройках протокола, используемых в собственном ядре РНУ IP, обратитесь к главе "Custom PHY IP User Core" в <u>Altera</u> <u>Transceiver PHY IP Core User Guide</u>.

#### Контроллер реконфигурации приёмопередатчика

Этот IP необходим для контроля настроек PMA и изменения прочих настроек приёмопередатчика в чипах Stratix V. Этот IP должен подключаться ко всем PHY IP (custom или low\_latency), контролируемым Transceiver Toolkit. Порты *reconfig\_from\_xcvr* и *reconfig\_to\_xcvr* должны быть соединены вместе.

Перевод: Егоров А.В., 2012 г.

За подробной информацией обратитесь к <u>Altera Transceiver PHY IP Core User</u> <u>Guide</u>.

Должны быть включены следующие настройки:

- Разрешить аналоговый контроль (Enable Analog controls)
- Разрешить блок EyeQ (Enable EyeQ block)
- Разрешить блок AEQ (Enable AEQ block)

# Ядро с малой задержкой РНҮ IP

Ядро с малой задержкой РНҮ ІР используется в следующих случаях:

- Чтобы получить более 8,5 Гб/с в чипах GT
- Чтобы использовать прямой режим РМА, например, когда используются шесть каналов в одной четвёрке.

Чтобы удовлетворять требованиям проекта, используйте такой же набор параметров, какой вы использовали в собственном ядре PHY IP.

Режим фазовой компенсации FIFO должен быть встроен выше определённой скорости передачи данных. Transceiver Toolkit выдаёт сообщение, когда вы выходите за пределы скорости передачи данных. В прямом режиме РМА, который поддерживает малый диапазон скоростей передачи, вы должны выключить фазовую компенсацию FIFO.

РНҮ с малой задержкой имеет настройку возвратной петли, установленную в режим последовательной обратной петли.

За подробной информацией о настройках протокола, используемых в собственном ядре РНҮ IP, обратитесь к главе "Low Latency PHY IP User Core" в <u>Altera</u> <u>Transceiver PHY IP Core User Guide</u>.

# Генератор пакетов данных Avalon-ST

Генератор производит стандартные пакеты данных, которые вы сможете использовать для тестирования. Пакеты включают в себя псевдослучайные двоичные последовательности (prbs): 7, 15, 23, 31; а также пакет с высокой и низкой частотами.

Этот компонент (Avalon-ST Data Pattern Generator) используется в процессе тестирования для генерирования пакетов данных. Вы можете использовать различные варианты популярных пакетов для тестирования целостности сигнала (SI) приёмопередатчика. Компонент генератора пакета данных является компонентом IP ядра Altera. Вы сможете использовать любой пакет, но у вас должен быть проверочный блок, контролирующий должным образом свойства принятого пакета.

Когда вы используете генератор пакетов данных Avalon-ST, его ширина может отличаться от ширины ядер Custom PHY IP и Latency PHY IP, таким образом, вам потребуется адаптер формата данных. Компонент Avalon-ST Data Pattern Generator (генератор) доступен в дереве библиотеки компонентов Qsys. Компонент Avalon-ST Data Pattern Checker (проверочный блок) доступен под **Debug and Performance** в дереве библиотеки компонентов Qsys. Адаптер формата может быть автоматически вставлен и должным образом сконфигурирован с помощью команды **Insert Avalon-ST Adapters**.

Генератор пакета данных Avalon-ST генерирует пакеты данных, соответствующие индустриальному стандарту. Пакеты данных генерируются в исходный потоковый порт Avalon шириной 32 или 40-бит.

Перевод: Егоров А.В., 2012 г.

За дополнительной информацией о системе внутренних соединений в SOPC Builder и Qsys обратитесь к главе "Система структуры внутренних соединений для потоковых интерфейсов" в <u>руководстве пользователя SOPC Builder</u>.

На рис. 11-3 показана страница wizard, которую вы используете для установки параметров для адаптера формата данных Avalon-ST.

#### Figure 11–3. Avalon-ST Data Format Adapter

Parameter settings Input Interface Parameters Data Symbols Per Beat: Include Empty Signat: AUTO Common to Input & Output Common to Input & Output Channel Signal Width (bits): D Max Channel O Include Packet Support When packets are supported, the starto/packet, endo/packet, and empty signals are used. Error Signal Width (bits): P Error Signal Description Ready Latency: D D D D D D D D D D D D D D D D D D D	MegaCore	Avalon-ST [	Data Format Adapto	er 🛛	bout Documentation
Input Interface Parameters         Data Symbols Per Beat:         Data Symbols Per Beat:         Include Empty Signat:         AUTO         Common to Input & Output         Channel Signal Vidth (bits):         0         Max Channet         0         Include Packet Support         When packets are supported, the startorpacket, endorpacket, and empty signals are used.         Error Signal Width (bits):         0         Data Bits Per Symbol:         0	Parameter Settings				
Input Interface Parameters Data Symbols Per Beat: 4 Include Empty Signat: AUTO Common to Input & Output Channel Signal Width (bits): 0 Max Channel Include Packet Support When packets are supported, the starto/packet, endotpacket, and empty signals are used. Error Signal Width (bits): 0 Error Signal Description: Ready Latency: 0 Data Bits Per Symbol: 0					
Common to Input & Output Channel Signal Width (bite): 0 Max Channet 0 Include Packet Support when packets are supported, the starto/packet, endotpacket, and empty signals are used. Error Signal Width (bits): 0 Error Signal Description: 0 Ready Latency: 0 Data Bits Per Symbol: 0 	Input Interfac Data Symbols Include Empty	e Parameters Per Beat: 4 Signal: AUTO	Output Interface Parameters Data Symbols Per Beat: 1 Include Empty Signal: AUTO	I.	
Channel Signal Width (bite): 0 Max Channel 0 Include Packet Support When packets are supported, the starto/packet, endotpacket, and empty signals are used. Error Signal Width (bits): 0 Error Signal Description: 0 Ready Latency: 0 Data Bits Per Symbol: 0 	Common to I	input & Output			
Max Channet 0 Include Packet Support When packets are supported, the startofpacket, endotpacket, and empty signals are used. Error Signal Width (bits): PError Signal Description: Ready Latency: Data Bits Per Symbol: 9	Channel Signa	l Width (bita): 0			
Include Packet Support When packets are supported, the startofpacket, endofpacket, and empty signals are used. Error Signal Width (bits): Description: Ready Latency: Data Bits Per Symbol: Description: Description	Max Channet	0			
When packets are supported, the startofpacket, endofpacket, and empty signals are used. Error Signal Vidth (bits): p Error Signal Description: Ready Latency: p Data Bits Per Symbol: g	🔲 Include P	acket Support			
endofpacket, and empty signals are used. Error Signal Width (bits):  D Error Signal Description: Ready Latency:  D Data Bits Per Symbol:  D	When p	ackets are supported, th	e startofpacket,		
Error Signal Width (bits): p Error Signal Description: Ready Latency: p Data Bits Per Symbol: 8	endorp	acket, and empty signals	are used.		
Error Signal Description:	Error Signal W	fidth (bits): D			
Ready Latency: p Dats Bits Per Symbol: 8	Error Signal D	escription:			
Data Bits Per Symbol: 8	Ready Laters				
	Data Bits Per 9	r. p Symbol: le			
		. 10			

#### Проверочный блок

Блок проверки данных Avalon-ST (Avalon-ST Data Pattern Checker) является поставляемым компонентом Qsys. Он сравнивает поток входящих данных со стандартным тестовым пакетом, который может быть: PRBS7, PRBS15, PRBS23, PRBS31, высокочастотным и низкочастотным. Он используется совместно с генератором пакетов Avalon-ST для тестирования цепей приёмопередатчика.

Блок проверки данных Avalon-ST доступен под **Debug and Performance** в дереве библиотеки компонентов Qsys. Проверочный блок является основой проверки пакетов данных. Пакеты данных доступны от исходного потокового порта Avalon шириной 32 или 40-бит.

За дополнительной информацией обратитесь к главе "Avalon Streaming Data Pattern Generator and Checker Cores" в <u>Embedded Peripherals User Guide</u>.

Используйте примеры проектов в качестве отправной точки для работы с отдельной отладочной платой целостности сигналов. Также вы можете изменять и дополнять примеры проектов специально под ваш проект приёмопередатчика. Когда вы используете Transceiver Toolkit, вы можете проверить целостность сигнала в цепи приёмопередатчика до завершения вашего проекта.

Используйте примеры проекта для быстрого тестирования функционирования каналов приёмников и передатчиков в вашем проекте без необходимости создания каких-либо собственных проектов с генераторами данных и проверочными блоками. Вы сможете быстро изменить настройки приёмопередатчика в примерах проектов, чтобы посмотреть, какой эффект это оказывает на характеристики цепи приёмопередатчика. Также вы сможете использовать Transceiver Toolkit для изолирования и верификации цепей приёмопередатчика без захвата отладчиком прочей логики в вашем проекте.

#### Компиляция примеров проектов

Для начала вам необходимо загрузить примеры проектов, открыть программу Quartus II версии 10.0 или выше и разархивировать проект. Если у вас есть такая же отладочная плата с тем же чипом, как упомянуто в файле **readme.txt** вашего примера, вы сможете прямо запрограммировать ваш чип поставляемым с примером программным файлом. Если вы хотите перекомпилировать проект, вам необходимо внести собственные изменения в конфигурацию в Qsys, регенерировать Qsys и перекомпилировать проект в программе Quartus II, чтобы получить новый программный файл.

Если у вас есть отладочная плата, описанная в файле readme.txt, но на ней вам необходимо выбрать соответствующий стоит другой чип, то чип И некоторые ранние перекомпилировать проект. Например, отладочные платы высылались с инженерными вариантами чипов.

Если у вас другая плата, вы должны отредактировать необходимое назначение выводов и перекомпилировать примеры проектов.

#### Изменение назначений выводов

Вы можете редактировать назначения выводов для различных отладочных наборов. В следующих параграфах приведены примеры назначений выводов для отладочного набора Stratix IV GX.

За подробной информацией о других отладочных наборах обратитесь к файлу **readme.txt**, находящемуся в примерах проектов. Дополнительная информация о выводах микросхемы находится в соответствующем руководстве пользователя, которое можно найти на странице <u>All Development Kits</u> сайта Altera.

В табл. 11-1 показаны назначения выводов для набора разработчика целостность сигнала приёмопередатчика Stratix IV (DK-SI-4SGX230N). Вы должны сделать эти назначения перед перекомпиляцией вашего проекта.

Top-Level Signal Name	I/O Standard	Pin Number on DK-SI-4SGX230N Board
REFCLK_GXB2_156M25 (input)	2.5 V LVTTL/LVCMOS	PIN_G38
S4GX_50M_CLK4P (input)	2.5 V LVTTL/LVCMOS	PIN_AR22
GXB1_RX1 (input)	1.4-V PCML	PIN_R38
GXB1_TX1 (output)	1.4-V PCML	PIN_P36

Table 11-1.	Stratix IV GX To	n-Level Pin	Assignments	(DK-SI-4SGX230N)
		P-Level I III	Assignments	DK-SI-4SUAZSON)

В табл. 11-2 показаны назначения выводов для набора разработчика Stratix IV GX (DK-DEV-4SGX230N). Вы должны сделать эти назначения перед перекомпиляцией вашего проекта.

#### Table 11–2. Stratix IV GX Top-Level Pin Assignments (DK-DEV-4SGX230N)

Top-Level Signal Name	I/O Standard	Pin Number on DK-DEV-4SGX230N Board
REFCLK_GXB2_156M25 (input)	LVDS	PIN_AA2
S4GX_50M_CLK4P (input)	2.5 V LVTTL/LVCMOS	PIN_AC34
GXB1_RX1 (input)	1.4-V PCML	PIN_AU2
GXB1_TX1 (output)	1.4-V PCML	PIN_AT4

Похожим образом вы изменяете назначения выводов на собственной плате и перекомпилируете пример проекта.