High Speed Transceiver Debugging
NMI FPGA Network
2015

David Taylor
Xilinx HSIO AE
Agenda

» Transceiver Overview

» Design for debug

» Debug tools and methodology – practical steps

» Advanced debug

» Summary

» Demo
The Insatiable Demand for Bandwidth

- In 2016, global IP traffic will reach 107.3 Exabytes per month
- Global IP traffic has increased eightfold over the past 5 years, and will increase threefold over the next 5 years
- The number of devices connected to IP networks will be nearly three times as high as the global population in 2016
- Traffic from wireless devices will exceed traffic from wired devices by 2014.

107 Exabytes of IP Traffic per Month by 2016
Enhanced Portfolio and More Bandwidth

- Kintex UltraScale has >2Tbps total transceiver bandwidth
- Virtex UltraScale has up to 5.1Tbps total transceiver bandwidth

Increased System Performance

Higher Line Rate and More transceiver counts result in more total bandwidth
Transceiver Challenges

- More flexibility for a given link. Growing number of protocols that need to be supported
- Line rate requirements continue to increase, exacerbating channel impairments
- Pulling functionality off the board and into the FPGA requires more IO
- System level power requirements continue to tighten, making the traditionally high power transceivers a focal point for power savings
- Serial design is labor intensive, requiring proper layout, simulation, and in system tuning

Programmable Systems Integration
Increased System Performance
BOM Cost Reduction
Total Power Reduction
Accelerated Design Productivity

© Copyright 2015 Xilinx
Agenda

» Transceiver Overview

» Design for debug

» Debug tools and methodology – practical steps

» Advanced debug

» Summary

» Demo
Thomas Edison wrote of “bugs” in 1878
- It is speculated that this term was in use prior to Edison

First documented case of the term “debugging” is credited to Admiral Grace Hopper in 1947

The term "bug" is used to a limited extent to designate any fault or trouble in the connections or working of electric apparatus
UltraFast™ Methodology Benefits

Fast Compile Times and Predictable Results
- Require good methodology

Project Schedules Drive Time To Market
- Manage risk affectively
- Minimize Iterations, especially late-stage changes
- Explore options early with estimation and progressive analysis

Proven Recommendations from Successful Customers
- Best Practices with Checklists and Links to Documentation
- Verification Tools and Reports
- Linting and DRC
JTAG! JTAG! JTAG!

➢ JTAG access
  – Provide on-board USB-to-JTAG circuitry (Digilent JTAG-SMT2 module) to enable configuration using a standard USB A-B cable eliminating the need to use an external Xilinx programming cable (e.g. Platform Cable USB II or Parallel Cable IV).

➢ Access to
  – Vivado Logic Analyzer
  – Vivado Serial IO Analyzer

➢ Hardware design for debug
  – Header access
  – GT clocks
Application Design Turn on....

» Measures
  – No link up or no picture
  – Data flow – IP framed
  – BER
  – Picture quality

Functional or parametric failures?
DEBUG.........
Agenda

- Transceiver Overview
- Design for debug
- Debug tools and methodology – practical steps
- Advanced debug
- Summary
- Demo
Debug Flow

On board debugging Start

PLL Locked?

Yes

PLL reset is released?

Yes

No

Check and fix the bugger of the PLL reset controller and the CPLLPD, QPLLPD signal

PLL lock

Ref Clocks

Verify PLL settings

Power supply

Is the near and Fabric loopback correct?

Yes

To debug and fix the FPGA logic.

No

Verify the MGTREFCLK's quality (Frequency, jitter, Frequency)

Is the MGTREFCLK correct?

Yes

Fix the MGTREFCLK issues

No

Verify the parameters and settings of the PLL

Fix the parameters and settings' issues

Is the near end Fabric loopback correct?

Yes

Verify the RX EQ and CDR setting (RXELECIDLEMODE) = 0

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the near and PMA loopback correct?

Yes

To debug and fix the FPGA logic.

No

Verify the RX EQ and CDR setting

Does the all resetdone ports go to high?

Yes

PLL reset is released?

No

Check and fix the bugger of the PLL reset controller and the CPLLPD, QPLLPD signal

Reset sequence

Verify loopbacks

Is the near end Fabric loopback correct?

Yes

Verify the RX EQ and CDR setting

TX/ RXRPMARESET, RXDFEPRESET, RXVESCARESET

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Double check the RXPMARESETDONE _1?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the near and PMA loopback correct?

Yes

To debug and fix the FPGA logic.

No

Verify the RX EQ and CDR setting

Does the all resetdone ports go to high?

Yes

PLL reset is released?

No

Check and fix the bugger of the PLL reset controller and the CPLLPD, QPLLPD signal

Far-end Loopback

Verify the MGTREFCLK's quality (Frequency, jitter, Frequency)

Is the MGTREFCLK correct?

Yes

Fix the MGTREFCLK issues

No

Verify the parameters and settings of the PLL

Fix the parameters and settings' issues

Is the far and PMA loopback correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the far and Fabric loopback correct?

Yes

Verify the RX EQ and CDR setting

TX/ RXRPMARESET, RXDFEPRESET, RXVESCARESET

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the near and Fabric loopback correct?

Yes

Verify the RX EQ and CDR setting

TX/ RXRPMARESET, RXDFEPRESET, RXVESCARESET

No

Fix the bugger of the reset controller

Are all correct?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the near and PMA loopback correct?

Yes

To debug and fix the FPGA logic.

No

Verify the RX EQ and CDR setting

Does the all resetdone ports go to high?

Yes

PLL reset is released?

No

Check and fix the bugger of the PLL reset controller and the CPLLPD, QPLLPD signal

Far-end loopbacks

Verify RXBUFRESET

TXUSERRDY

TXPCSRESET

RXPCSRESET

RXMARESET

RXDFEPRESET

RXVESCARESET

TX/ RXRPMARESET

Double check the RXPMARESETDONE _1?

Yes

Fix the bugger of the reset controller

No

Fix the bugger of the reset controller

Is the near and PMA loopback correct?

Yes

To debug and fix the FPGA logic.

No

Verify the RX EQ and CDR setting

Does the all resetdone ports go to high?

Yes

PLL reset is released?

No

Check and fix the bugger of the PLL reset controller and the CPLLPD, QPLLPD signal

Working !
Debug Tools – Practical Steps

- **Functional logic simulation**
  - Is the circuit reset successfully and clocks at the right frequency?

- **Vivado logic analyzer and debug header ports**
  - Primary application debug information of coding errors, reset information and clocks

- **GT BiT features**
  - Loop back, in built PRBS and Eye Scan

- **IBERT – Integrated Bit Error Rate Tester**
  - Standalone IP used for proving basic hardware functionality and margin testing

- **IBIS AMI simulation**
  - Mostly used in design, can be used for correlation

- **Oscilloscopes/spectrum/protocol analyzers**
  - Only way to measure the PCB signals and power independently
1) Functional Logic Simulation

- Vivado logic simulator
- All IP example designs have test bench built in
- Verify application design does what is expected to some level
  - Eg as minimum comes out of reset....
2) Vivado Logic Analyzer Components

- Vivado Logic Analyzer components
  - Debug cores that sample and inject signals into the programmable logic
  - Comprised of two basic items
- Virtual Inputs and Outputs (basic IO)
- Integrated Logic Analyzer (advanced trigger and capture)

Flows for generating and inserting these debug cores into a design and for controlling and viewing these cores once downloaded
Probing Ports on Vivado Logic Analyzer

- C(Q)PLLLOCK
- TX(RX)USR_RDY
- TX(RX)RESETDONE
- TXDATA
- TXBUFSTATUS
- RXDATA
- RXBUFSTATUS
- Plus MORE…

Needs to be pre-implemented in user design
3) GT BiT Features - Loopback

- Loopback testing (Run bit error test under each condition)
  - Near-end fabric loopback (verifies fabric logic inbound and outbound)
3) GT BiT features - Loopback

- Loopback testing (Run bit error test under each condition)
  - Near-end fabric loopback (verifies fabric logic inbound and outbound)
  - Near-end PCS loopback (verifies fabric to GT interface inbound and outbound)
3) GT BiT features - Loopback

Loopback testing (Run bit error test under each condition)
- Near-end fabric loopback (verifies fabric logic inbound and outbound)
- Near-end PCS loopback (verifies fabric to GT interface inbound and outbound)
- Near-end PMA loopback (verifies fabric thru PMA path inbound and outbound)
3) GT BiT features - Loopback

Loopback testing (Run bit error test under each condition)
- Near-end fabric loopback (verifies fabric logic inbound and outbound)
- Near-end PCS loopback (verifies fabric to GT interface inbound and outbound)
- Near-end PMA loopback (verifies fabric thru PMA path inbound and outbound)
- Channel loopback
  - Prefer a loopback thru as much of the channel as possible to verify channel performance
3) GT BiT features - Loopback

- Loopback testing (Run bit error test under each condition)
  - Near-end fabric loopback (verifies fabric logic inbound and outbound)
  - Near-end PCS loopback (verifies fabric to GT interface inbound and outbound)
  - Near-end PMA loopback (verifies fabric thru PMA path inbound and outbound)
  - Channel loopback
    - Prefer a loopback thru as much of the channel as possible to verify channel performance
  - Far-end PMA
  - Far-end fabric loopback, if clocking supports it
3) GT BiT UltraScale Transceivers PRBS

- **PRBS Generator**
- **128/130 Encoder**
- **TX Async Gearbox**
- **TX PI Controller**
- **PRBS Checker**
- **128/130 Decoder**
- **RX Async Gearbox**

**Diagram Details:**
- TX Driver
- TX Clock Dividers
- TX Phase Interpolator
- TX PCS
- TX PMA
- RX Clock Dividers
- RX EQ DFE
- RX OOB
- RX Serial Clock
- PMA Parallel Clock (XCLK)
- RX Async Gearbox
- PRBS Decoder
- RX Elastic Buffer
- RX Sync Gearbox
- RX PI Controller
- RX Status Control
- RX Pipe Control
- FPGA RX Interface
- PCIe Beacons
- SATA OOB
- TX Async Gearbox
- Phase Comp. FIFO
- TX Phase Interpolator Controller (GTy)
- PRBS Generator
- 128/130 Encoder
- PRBS Checker
- RX Async Gearbox
- 128/130 Decoder
4) GT BiT Features – Non Destructive Eye Scan Architecture

**Figure 4-25: PMA Architecture to Support Eye Scan**

Eye Scan HW
4) GT BiT Features – Non Destructive Eye Scan Architecture

- **Silicon**
  - Non-disruptive Eye Scan
  - Up to 16384 sampling points at maximum rate
  - Data pattern independent

- **IBERT**
  - Stand alone design
  - Pattern generation/checking
  - Access to all ports and attributes on the fly
  - Attribute sweep capabilities
  - Access to Eye Scan circuitry and automated eye plot generation
5) IBERT – Integrated Bit Error Rate Tester

Asynchronous Backplane

Backplane Cable loopback 28.2 Gb/s! 0 Errors 1 day 9 hrs! PRBS31! All TX values the same! No Pre-emphasis! DFE Working!

© Copyright 2015 Xilinx
5) IBERT – Integrated Bit Error Rate Tester
6) IBIS-AMI Simulation

Simulation is done in Keysight ADS

25G Equalized Eye Diagram

With TX EQ

With RX EQ

Adaptation

ChannelSim

ChannelSim1

NumberOfBits=1000000

ToleranceMode=Auto

EnforcePassivity=yes

Mode=Bit-by-bit

 simulation is done in Keysight ADS

ComponentName="xilinx_ultrascale_gty_ami_tx" FileName="C:/For Brandon/Simulation_Home_Directory/ADS/ibis-ami_simulation.xsim" PinName="1p" ModelName="xilinx_ultrascale_gty_ami_tx" SetAllData=yes DataTypeSelector=Typ UsePkg=yes BitRate=25 Gbps

Rx

ChannelSim

ChannelSim1

NumberOfBits=1000000

ToleranceMode=Auto

EnforcePassivity=yes

Mode=Bit-by-bit

simulation is done in Keysight ADS

ComponentName="xilinx_ultrascale_gty_ami_rx" FileName="C:/For Brandon/Simulation_Home_Directory/ADS/ibis-ami_simulation.xsim" PinName="2p" ModelName="xilinx_ultrascale_gty_ami_rx" SetAllData=yes DataTypeSelector=Typ UsePkg=yes
7) Signal Analysis

➢ Oscilloscopes
   - Real time/sub-sampling eye and jitter analysis
   - Can have TDR functions for impedance testing

➢ BER and protocol testers
   - Jitter tolerance testing
   - PCIe bus analyzers
   - Ethernet testers etc

➢ Spectrum analysis and VNA
   - SA useful for clock analysis ie PSU jitter components
   - VNA can be used for channel analysis but more specialised
Agenda

- Transceiver Overview
- Design for debug
- Debug tools and methodology – practical steps
- Advanced debug
- Summary
- Demo
Embedded Eye Scan


» MCS.
  - Microbraze Controller System

» FAST
  - Software-based implementation of Eye Scan allows faster TAT and flexibility of continual improvements

» Extensible
  - Capability of running Eye Scans on multiple channels simultaneously.

» Familiality
  - Compilable C code for the statistical eye algorithm that can be reused in other implementations.

» Resource Utilization for 4ch
  - 910 LUTs
  - 870 FFs

» Youtube demo available
  - http://www.youtube.com/watch?v=4YcxACij8Fc

» EyeScan over PCI express Gen3 link at 8 Gbps
GTH will support 10GBase-KR (up to 16.3Gb/s)
- 11 Tap DFE + 3 Stage CTLE + 3 Tap TX FIR
GTY will support CEI-28G-LR (up to 28.21Gb/s)
- 15 Tap DFE + 3 Stage CTLE + 3 Tap TX FIR
The receiver equalization in GTH and GTY are fully adaptive
UltraScale Transceiver Signal Integrity
Receiver CTLE and DFE

3-Stage CTLE

High-Freq. Boost

Long-Tail Cancellation

Gain

Pulse Response

Amplitude

Time

0.4

0.3

0.2

0.1

0.0

11 Tap in GTH

15 Tap in GTY

DFE Region

Long Tail

© Copyright 2015 Xilinx
Auto-Adaptive Receiver Equalization in Action

Transceiver Architecture

- Hard PCS Logic
- PLL
- RX CDR
- RX DFE
- RX CTLE
- RX AGC
- TX Driver Pre-emphasis
- TX PKG
- RX PKG
- 2-D Eye Scan

Open Eye at Transmitter

Lossy Channel

Closed Eye at Receiver

Open Eye Post-Equalization

2-D Eye Scan
## Auto Adaptive Equalization

<table>
<thead>
<tr>
<th></th>
<th>Number of Settings</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTLE</td>
<td>Thousands</td>
<td>Adjustable boost amplitude and peaking frequency</td>
</tr>
<tr>
<td>DFE</td>
<td>Thousands -&gt; millions</td>
<td>Number of taps with adjustable tap weights for each tap</td>
</tr>
<tr>
<td>Combinations</td>
<td>Millions -&gt; Trillions</td>
<td>Increase exponentially with tap numbers</td>
</tr>
</tbody>
</table>

### Variations

- **Chip Variation**
- **Temperature Variation**
- **Voltage Variation**
- **Channel Length Variation**
- **Channel Length Variation**

---

© Copyright 2015 Xilinx
Correlation Debug Analysis

Hardware measured

*eye scan result post processed

Simulation

Control loop
Scope Debug Analysis
Spectrum Analyser Debug Analysis

Expand A Portion
Of A Sideband
The Spectrum Analyzer Output is in dBc / Hz, dBc = dB relative to carrier

Phase noise specifications
- Are understood by oscillator vendors – dBc/Hz
- Can be converted to rms jitter using conversion programs or spreadsheets
- RMS to peak-peak estimates can be made using an assumption of Gaussian noise distributions ie (10E-12 x14, 10E-15 x16)
- Direct effect on achievable BER

Need to be thinking in terms of ‘low ps’ rms jitter
Via Debug Analysis

Figure 9 - Daughter Card Via Stubs

Figure 33 - SMA Connector Pin Stub Diagram

Figure 34 - Reference Channel Insertion-Loss with Trimmed SMA
Layout Debug Analysis

- Crosstalk
- Track and plane proximity
- Signal coupling
  - Near and far end
  - Adjacent tracks
  - Via and connector coupling
  - Use ground screening pins/vias

- Power plane
  - Signal contamination
  - High di/dt switching supplies
Agenda

» Transceiver Overview

» Design for debug

» Debug tools and methodology – practical steps

» Advanced debug

» Summary

» Demo
Summary (1)

➤ Serial mostly just works
  – Everywhere, with 10G now mainstream

➤ However some more common problems –

➤ Clocks
  – Is Reference Clock present and with the appropriate frequency and location?

  – What’s the Reference Clock Jitter?

  – Are the buffers over/underflowing?

  – Verify on pins or with ILA
Resets
  – Are the resets completing?
  – Does it simulate?

Signal integrity and PDS
  – Reflections, loss and coupling
  – Power plane IR drops

Serial mostly just works
  – Follow the design guides