How to use the WR PTP Core to make your own WR Nodes

Greg Daniluk

1st White Rabbit Tutorial Workshop
7 October 2017
Barcelona, Spain
Outline

• What is WR PTP Core
• How to make your own WR Node
• WR PTP Core interfaces and parameters
• First configuration of the WR PTP Core
• WR PTP Core shell
White Rabbit network

In this presentation we focus on WR Nodes
WR Node

How to use the WR PTP Core to make your own WR Nodes
WR Node

WR PTP Core is essential part of every WR Node
WR PTP Core overview

- Ethernet MAC HDL module
- ... with WR features
- Provides time to user cores
- Can send and receive user-defined Ethernet frames

How to use the WR PTP Core to make your own WR Nodes

G. Daniluk
WR PTP Core overview

- Implemented in the FPGA
- Using VHDL language
- You don’t need to know WR internals
- You need to know FPGAs to use it
• **LatticeMico32 runs WR PTP daemon**
Where to start?

• Main WRPC wiki page
  https://www.ohwr.org/projects/wr-cores/wiki/wrpc-core

• User manual for the last stable release (v4.1)

• wr-cores git repository
  git://ohwr.org/hdl-core-lib/wr-cores.git

• wrpc-sw git repository (optional)
  git://ohwr.org/hdl-core-lib/wr-cores/wrpc-sw.git
Outline

• What is WR PTP Core
• How to make your own WR Node
• WR PTP Core interfaces
• First configuration of the WRPC
• WR PTP Core shell
Repository structure

- **wr-cores**
  - **bin**
  - **board**  - Board Support Packages
  - **ip_cores**
  - **modules**
    - **wrc_core**  - White Rabbit PTP Core module
  - **platform**  - Platform Support Packages
  - **sim**
  - **syn**  - Xilinx/Altera synthesis project files
  - **testbench**
  - **top**  - Reference designs
3 options for a WR Node

1. Based on officially supported hardware
2. Custom hardware with supported FPGA
3. Non-supported FPGA platform
Officially supported hardware

• Reference design for every stable release
Officially supported hardware

- Reference design for every stable release
- SPEC – PCIe, Xilinx Spartan 6
Officially supported hardware

• Reference design for every stable release
• SPEC – PCIe, Xilinx Spartan 6
• SVEC – VME, Xilinx Spartan 6

G. Daniluk
How to use the WR PTP Core to make your own WR Nodes
Officially supported hardware

- Reference design for every stable release
- SPEC – PCIe, Xilinx Spartan 6
- SVEC – VME, Xilinx Spartan 6
- VFC-HD – VME, Altera Arria V
Officially supported hardware

• Reference design for every stable release
• SPEC – PCIe, Xilinx Spartan 6
• SVEC – VME, Xilinx Spartan 6
• VFC-HD – VME, Altera Arria V
• FASEC – “pizzabox” Xilinx Zynq – coming soon
Officially supported hardware

• SPEC – PCIe, Xilinx Spartan 6
• SVEC – VME, Xilinx Spartan 6
• VFC-HD – VME, Altera Arria V
• FASEC – “pizzabox” Xilinx Zynq – coming soon

For all these, use one of Board Support Packages.
Board and Platform Support Package

• Platform Support Package (PSP)
  • Deterministic GbE Serdes module
  • PLLs for main and DMTD offset clock

• Board Support Package (BSP)
  • WR PTP Core
  • VCO DAC controller
  • Reset logic
  • Differential clock buffers
  • Platform Support Package

BSP glues WRPC with all required FPGA modules for a given hardware

G. Daniluk
How to use the WR PTP Core to make your own WR Nodes
How to make a WR Node from BSP?

- Go to `wr-cores/top/` and pick the reference design for your board
- Add your application-specific HDL modules
- Connect your modules with the WRPC interfaces
- Done!
3 options for a WR Node

1. Based on officially supported hardware

2. Custom hardware with supported FPGA

3. Non-supported FPGA platform
Supported FPGA platforms

• Current release (v4.1):
  • Xilinx Spartan-6
  • Intel Arria V

• To be included in the next release:
  • Xilinx Zynq
  • Xilinx Kintex-7
  • Xilinx Kintex Ultrascale

For all these, use one of the Platform Support Packages.
WR Node on custom hardware(1)

- Draw inspiration from an existing BSP
  - Modify reset logic
  - Modify clocks and DAC interface
- Take PSP for your FPGA family
  - Modify PLL parameters for main and DMTD offset clocks
WR Node on custom hardware (2)

• Create your own design based on one of the reference designs in \texttt{wr-cores/top/}

• Add your application-specific HDL modules

• Connect your modules with the WRPC interfaces

• Done!
3 options for a WR Node

1. Based on officially supported hardware

2. Custom hardware with supported FPGA

3. Non-supported FPGA platform
Unsupported FPGA platform

- May be hard

- If GbE SerDes is the same as in a supported FPGA
  - Expand PSP for your FPGA family
  - Instantiate PLLs and SerDes

- If SerDes wrapper does not exist yet
  - Write to white-rabbit-dev mailing list
  - Contact us or one of the companies for help
  - You need to be FPGA & WR expert
  - SerDes has to be properly configured
  - Wrapper for bitslide measurement
Outline

• What is WR PTP Core
• How to make your own WR Node
• **WR PTP Core interfaces**
• First configuration of the WRPC
• WR PTP Core shell
WR PTP Core interfaces

WR PTP Core
- Clocks / reset
- DACs output
- PHY I/F
- Flash/EEPROM
- UART / LEDs

Fabric I/F
- Control WB
- Timecode I/F
- Aux Clk I/F

User core
Fabric interface

- Gives WR Node designers the possibility to send custom Ethernet frames
- WRF Source – for received frames
- WRF Sink – for transmitted frames
- Based on two pipelined Wishbone buses

<table>
<thead>
<tr>
<th>dat[15..0]</th>
<th>meaning of data word</th>
</tr>
</thead>
<tbody>
<tr>
<td>adr[1..0]</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Regular data</td>
</tr>
<tr>
<td>1</td>
<td>Out-of-band data (OOB)</td>
</tr>
<tr>
<td>2</td>
<td>Status word</td>
</tr>
<tr>
<td>3</td>
<td>not used</td>
</tr>
</tbody>
</table>
Fabric interface Tx

How to use the WR PTP Core to make your own WR Nodes
Fabric interface - example Tx cycle

clk

cyc_o

stb_o

adr_o

2 (status word) 0 (data)

dat_o

0000 1A1B 1C1D 1E1F 0A0B 0C0D 0E0F F2B0 0001 0203 0405 0607 0800 0A0B 0C0D 0E0F

dst MAC src MAC ET Payload

ack_i

1

stall_i

G. Daniluk

How to use the WR PTP Core to make your own WR Nodes

31
Fabric interface – Tx status word

• First data word of a Tx frame (adr=2)

  - reserved bits

  • vCRC – Tx frame already contains a valid CRC checksum
  • vSMAC – Tx frame already contains a valid Source MAC
Fabric interface Rx

How to use the WR PTP Core to make your own WR Nodes
Fabric interface - example Rx cycle

 clk
 cyc_i
 stb_i
 adr_i
 dat_i
 ack_o
 stall_o

2 (status word) 0 (data) 1 (OOB)

 dst MAC src MAC ET Payload

1A1B 1C1D 1E1F 0A0B 0C0D 0E0F F2B0 0001 0203 0405 0607 0809 0A0B 0C0D

G. Daniluk
How to use the WR PTP Core to make your own WR Nodes
Fabric interface – Rx status word

• First data word of an Rx frame \( (\text{adr}=2) \)

![Diagram showing Rx status word with 15 bits, 8 bits for packet class, 4 bits for reserved bits, and 1 bit for err.]

• **packet class** – used to split Rx WRPC traffic from user traffic

• **err** – an error occurred while receiving frame
Fabric interface - example Rx cycle

clk

cyc_i

stb_i

adr_i

dat_i

2 (status word)

0 (data)

1 (OOB)

dst MAC

src MAC

ET

Payload

ack_o

stall_o

G. Daniluk

How to use the WR PTP Core to make your own WR Nodes
Fabric interface – Rx OOB

- Last data words appended to the Rx frame (adr=1)
- Carries Rx timestamp of the frame

<table>
<thead>
<tr>
<th></th>
<th>15</th>
<th>11</th>
<th>4</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>word0</td>
<td>Tiv</td>
<td></td>
<td>port ID</td>
<td></td>
</tr>
<tr>
<td>word1</td>
<td></td>
<td></td>
<td>CNTR</td>
<td></td>
</tr>
<tr>
<td>word2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Tiv – timestamp invalid
- port ID – ID of the physical port
- CNTR – Rx timestamp
WR PTP Core interfaces

- Clocks / reset
- DACs output
- PHY I/F
- Flash/EEPROM
- UART / LEDs

WR PTP Core

Fabric I/F
- Control WB
- Timecode I/F
- Aux Clk I/F

User core
Control Wishbone interface

- Wishbone Slave interface
- Direct access to all the internal WRPC registers
- Should be connected to the host system via appropriate bridge (PCIe, VME)
Timecode interface

- WR-synchronized time for user-defined modules
- 1-PPS output
- time valid output
- TAI time – seconds since 1 January 1970
- Cycles counter – number of clock cycles since the start of a second
Auxiliary clock interface

• Synchronizes on-board Aux clocks to the WR clock
• Inputs the Aux clock signal
• Drives DAC to tune the Aux VCO
• Aux clock of same frequency as the Ref clock (62.5 MHz or 125 MHz)
WR PTP Core interfaces

- Clocks
- DACs output
- PHY I/F
- Flash/EEPROM
- UART / LEDs

WR PTP Core

Fabric I/F
Control WB
Timecode I/F
Aux Clk I/F
Clocks and DACs

• Clocks
  • System clock (62.5 MHz)
  • Reference clock (125/62.5 MHz)
  • DMTD offset clock (62.5 MHz)

• DACs output
  • Main and helper 16-bit data word
  • Main and helper load signals
PHY interface

• Set of Tx and Rx signals connected directly to the GbE Serdes
• Passes transmitted and received Ethernet frames
• 8 or 16-bit data word (depending on the FPGA platform and reference clock)
• GbE Serdes has to come from the wr-cores repository
  GTP / GTX / GTH wrappers for Xilinx platform
Flash, EEPROM, UART, LEDs

- Flash / EEPROM
  - SPI interface for external Flash chip
  - Optional I²C interface for external EEPROM chip
  - Used to store calibration parameters and configuration
  - Currently Flash is preferred

- SFP EEPROM
  - I²C interface for SFP identification

- UART
  - 115200 bps interface for accessing a simple WRPC Shell

- LEDs
  - Link, Activity LEDs for Ethernet socket

- 1-Wire thermometer
  - PCB temperature reporting
  - Pseudo-unique MAC address generation
WR PTP Core interfaces

WR PTP Core

Clocks / reset
DACs output
PHY I/F
Flash/EEPROM
UART / LEDs

Fabric I/F
Control WB
Timecode I/F
Aux Clk I/F

User core
WRPC parameters

• Parametrization done with VHDL generics

• \texttt{g\_simulation} – speed up initializations for simulation
• \texttt{g\_aux\_clks} – number of Aux clocks to be synchronized
• \texttt{g\_pcs\_16bit} – SerDes data word width (8/16 bits)
Outline

• What is WR PTP Core
• WR PTP Core interfaces and parameters
• How to use WR PTP Core for your WR Node
• First configuration of the WR Node
• WR PTP Core shell
Before first start

• Flash configuration for storage
• Calibration
Flash configuration for storage

• **SDBFS** - very simple filesystem used in Flash

• **Files:**
  - `mac-address` – MAC address of a WR port
  - `sfp-database` – Calibration values
  - `wr-init` – WRPC Shell commands executed on boot time

• **SDBFS structure written by manufacturers for SPEC and SVEC boards**

Request manufacturers to assign official MAC address during production.
Flash configuration for storage

• For other boards and custom hardware
• Empty SDBFS image has to be written to Flash
  • using JTAG cable
  • from a host system

• See WR PTP Core User Manual for instructions
Calibration

- Done for a given hardware and firmware
- To measure fixed Tx/Rx hardware delays
- Otherwise sub-ns synchronization is not guaranteed
- Calibration parameters stored in Flash

- See the presentation on calibration for more details
Outline

• What is WR PTP Core
• WR PTP Core interfaces and parameters
• How to use WR PTP Core for your WR Node
• First configuration of the WR Node
• WR PTP Core shell
WRPC Shell

- Available through UART and Wishbone interface
- Useful for early stage debugging
WRPC Shell

• Most important commands:
  • ptp  start/stop
  • mode slave/master/gm
  • gui
  • init show/erase/add
  • sfp  show/erase/add
  • ip    set/get
Outline

• What is WR PTP Core
• WR PTP Core interfaces and parameters
• How to use WR PTP Core for your WR Node
• First configuration of the WR Node
• WR PTP Core shell
Summary

• WR PTP Core implements White Rabbit for the node
• Provides WR time for user-defined HDL modules
• The simplest way to use it is through Board and Platform Support Packages

• You don’t need to know WR internals to use it
• You need to know FPGAs to use it