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

Greg Daniluk, Hongming Li

2nd White Rabbit Tutorial Workshop
28 June 2018
THU China
Outline

• Hardware requirements
• 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

How to use the WR PTP Core to make your own WR Nodes
Hardware requirements for a WR node

• Current, all WR nodes are based on FPGA. (Xilinx or Intel)
  • FPGA with high speed transceiver
  • Xilinx Spartan-6/Zynq/Kintex-7/Kintex Ultrascale

• With FPGA related circuit:
  • Power/Power filter
  • Flash
  • Debug interface(JTAG/UART...)

• With extra clock circuit:
  • DAC
  • VCXO(TC-VCXO, OC-VCXO, Atomic clock)
  • PLL
Clock circuits

OSC Power

DAC

TCVCXO

PLL chip

VCXO

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
Gigabit SFP circuit

GTP/GTX interface

GTP/GTX reference clock from clock circuit

SFP interface

RX/TX differential signal

TX/RX differential signal

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
PCB layout guidelines

• High speed serial line
  • TX and RX (differential signal)
  • Careful design of impedance matching

• To reduce temperature effect
  • Equal length of TX and RX
  • Short distance of TX and RX
  • short PPS/clock output routine
Variety of WR nodes

Not so hard to design your own WR node

SPEC
AFC/AFCK
CUTEDP

MINI
- Quite small
- Easy to Integrate

board used in LHAASO project
board used in nuclear Medicine
Outline

• Hardware requirements
• **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 Node

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

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
WR PTP Core inside

• 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 (v4.2)**

- **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

• Hardware requirements
• 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
3 options for a WR Node

1. Based on officially supported hardware

2. Custom hardware with supported FPGA

3. Non-supported FPGA platform
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
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
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
• CUTE-WR-DP – Xilinx Spartan 6 (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

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
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 (v4.2):**
  - Xilinx Spartan-6
  - Intel Arria V
  - Xilinx Zynq
  - Xilinx Artix-7
  - Xilinx Kintex-7

• **To be included in the next release:**
  - 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 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

• Hardware requirements
• 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
  - dat[15..0]
  - adr[1..0]  
    | meaning of data word |
    |----------------------|
    | 0                    |
    | Regular data        |
    | 1                    |
    | Out-of-band data (OOB) |
    | 2                    |
    | Status word         |
    | 3                    |
    | not used             |
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
dat_o
ack_i
stall_i

2 (status word)
0 (data)
dst MAC
src MAC
ET
Payload

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
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

G. Daniluk, H. Li
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 (adr=2)

\[
\begin{array}{c}
15 & 8 & 4 & 0 \\
\hline
\text{packet class} & & & \\
\hline
- \text{reserved bits} & & & \\
\hline
\text{err} & & & \\
\end{array}
\]

• packet class – used to split Rx WRPC traffic from user traffic
• err – an error occurred while receiving frame
Fabric interface - example Rx cycle
Fabric interface – Rx OOB

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

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

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
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

WR PTP Core

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

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

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
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

- Tx and Rx signals connected directly to the GbE Serdes
- Passes transmitted and received Ethernet frames
- GbE Serdes has to come from the wr-cores repository
- GTP / GTX / GTH wrappers for Xilinx platform

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
Flash, EEPROM, UART, LEDs

- Flash / EEPROM
  - SPI interface for external Flash chip
  - ... or I²C interface for external EEPROM chip
  - Used to store calibration parameters and configuration

- 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

G. Daniluk, H. Li
How to use the WR PTP Core to make your own WR Nodes
Outline

• Hardware requirements
• 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 on request by manufacturers for SPEC and SVEC boards
Flash configuration for storage

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

• See WR PTP Core User Manual for instructions
Calibration

• Need to be 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/EEPROM

• See the presentation on calibration for more details
Outline

• Hardware requirements
• 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:
  • sdb fs 0
  • ptp start/stop
  • mode slave/master/gm
  • gui
  • init show/erase/add
  • sfp show/erase/add
  • ip set/get
Outline

• Hardware requirements
• 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
Backup
Repository structure

- **wr-cores**
  - bin
  - board
  - ip_cores
  - modules
    - wrc_core
  - platform
  - sim
  - syn
  - Testbench
  - top

- Board Support Packages
- White Rabbit PTP Core module
- Platform Support Packages
- Xilinx/Altera synthesis project files
- Reference designs
Fabric interface – Tx status word

- First data word of a Tx frame ($\text{adr}=2$)

- $v\text{CRC}$ – Tx frame already contains a valid CRC checksum
- $v\text{SMAC}$ – Tx frame already contains a valid Source MAC
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)