White Rabbit PTP Core (WRPC)

Preliminary abstract of a draft of a functional specification (rev. 2)

1. Overview

WR PTP Core is a HDL module implementing a complete gigabit Ethernet interface (MAC + PCS + PHY) with integrated PTP slave ordinary clock compatible with White Rabbit protocol. It performs subnanosecond clock synchronization via WR protocol and also acts as an Ethernet “gateway”, providing access to TX/RX interfaces of the built-in WR MAC.

Features:
- Physical layer: 1.25 Gbps 802.3z SerDes with optional support to other media (e.g. 1000BaseT) via external PHY.
- Support for copper-based physical layer via an external PHY.
- Works with Xilinx Spartan-6 GTP or discrete ten-bit interface (TBI) PHY
- PTP clock: 2-step ordinary clock (slave only), Layer 2 protocol, support for WR subnanosecond synchronization.
- PPS/timecode/reference clock output.
- WRF source/sink provided for the user application with configurable MAC address filtering
- optional Wishbone slave port for PTP and Endpoint configuration (direct access to EP regs and CPU memory)
- optional configurable status and control pins
- I2C interface for optional calibration/configuration EEPROM

Fig. 1. WRPC core and its interfaces
2. Platform requirements

The minimal WRPC implementation requirements are listed below:

- Xilinx Spartan-6 FPGA with at least one GTP transceiver
- (or) Xilinx (Spartan3/6) or Altera (Cyclone3) FPGA with TBI PHY
- at least 64 kB (CPU memory) + 12 kB (Endpoint buffers/FIFOs) of FPGA RAM blocks
- max. FPGA footprint: 20% of XC6SLX45T
- external digitally tunable 125 MHz reference clock source
- one FPGA clock manager (DCM/PLL)

3. Core I/O

3.1. 802.1z SerDes interface

1.25 Gbit/s, 8b10b-Encoded Ethernet (802.3z). Provided by Spartan-6 GTP or external TBI PHY. For external TBI PHYs (such as TLK1221, etc.) additional calibration logic must be provided.

3.2. Clock inputs

WRPC requires the following external clocks:

- REFCLK: 125 MHz Ethernet reference clock. Clock rate must be digitally tunable within 125 MHz +- 10ppm with 2.5ppm center frequency accuracy. The source must provide a glitch-free continuous phase shifting capability with resolution of at least 25 ps. REFCLK must meet the jitter requirements of 802.3z.
- DMTDCLK: DMTD offset clock locked to SerDes RX clock (RXCLK). Maximum frequency offset from RXCLK is 1/512.
- FPGA system clock input (clk_sys_i). Clock used by the CPU, WRF interface and Wishbone interconnect. Due to the width of internal datapath, its frequency must be at least REFCLK/2 (i.e. 62.5 MHz)

3.3. PPS and timecode output

WRPC must provide a 1 PPS output (both to inside and outside of the FPGA), meeting the following requirements:

- programmable pulse width.
- synchronous to REFCLK
- issued at the beginning of every full second or a programmable number of REFCLK cycles in advance

Timecode output *(TBD – in case of the FPGA internal logic, it can be as simple as UTC time register accessible via Wishbone. External output – UTC, IRIG-B, any other ideas?)*

3.4. WRF Interface

White Rabbit Fabric I/F, interfacing the WR MAC to the user logic in the FPGA.
3.5. Control/status pins

WRPC shall provide configuration/status pins with a possibility of enabling/disabling each pin using a VHDL generic. The status pins are:

- stat_link_up_o: 1 indicates that Ethernet link is up
- stat_locked_o: 1 indicates that the timing system is locked to the RX clock
- stat_synced_o: 1 indicates that the WRPC has completed the synchronization process. Asserted after all timing signals (REFCLK, PPS) have been properly synchronized and compensated.
- stat_wr_mode_o: 1 indicates that WRPC is synchronized in WR mode

The control pins are:

- ptp_enable_i: 1 enables the CPU core and timing system. When 0, the Endpoint still works, but all PTP-related features are disabled.
- sync_enable_i: 1 enables PTP synchronization process.
- wr_enable_i: 1 forces WRPC to synchronize in WR mode.

(add more pins if you need them)

3.6. Configuration EEPROM interface

Optional I2C master port for accessing the configuration EEPROM. Can be accessed via Wishbone from outside the core and used for storing data other than WRPC configuration.

(define the format of EEPROM contents)

3.7. Wishbone slave port

Classis 32-bit Wishbone slave port. Provides access to:

- Endpoint internal registers
- Part of the PTP CPU memory (for PTP configuration)
- EEPROM I2C controller
- Part of the PPS generator registers (for timecode access)
4. Block diagram

![Block diagram of WRPC Core](image)

**Fig. 2. Block diagram of WRPC Core**

### 4.1. WR GTP PHY

A dual-port pre-configured wrapper for Xilinx GTPs with phase alignment and all the other timing tricks necessary for achieving subnanosecond precision.

**Status:** based on Peter & Henk's work, synthesizes and simulates correctly, not tested in HW yet.  
**Todo:** test in HW

### 4.2. WR Endpoint

WR-compatible MAC.

**Status:** tested in the WR switch, synthesizes on Xilinx.  
**Todo:** test Xilinx port, add programmable address filtering.

### 4.3. WRF priority Mux

Unit multiplexing 2 WRF sources/sinks, used to redirect some packets to the PTP core and pass everything else to the user application. Data from/to user app has priority over PTP packets. If the PTP-side port is unable to receive a packet, it shall be dropped to avoid disturbing the user application traffic.

**Todo:** write & test.

*(for the initial implementation of WRPC, this block is not necessary – we don't have Etherbone or NIC yet, so this unit would be useless anyway. WRF endpoint ports can be then directly connected to the miNIC).*
4.4. WR miNIC

Minimalistic network card implementation. Shares the memory with the CPU through a dedicated port in the system RAM (it’s a dual-port RAM anyway, so the DMA is for free)

Status: done.

4.5. Timing system

A set of tunable oscillators, PLLs and other components required to produce a 125 MHz REFCLK (with fine phase shifting) and the DMTD offset clock (DMTDCLK). The user of the core can use the reference timing system design provided in this document or build his own timing system, provided that it follows these requirements:

- capability of locking to RXCLK (coming from a digital-interpolation CDR, so very jittery)
- contigious, monotonic, glitch-less phase shifting of REFCLK with resolution of 25 ps and INL of 50 ps, programmable via Wishbone (define the software interface for that – Alessandro)
- produces DMTDCLK with offset (with respect to RXCLK) of 1/512 or better (i.e. smaller)
- provides external REFCLK output (outside the FPGA).

Fig. 3. Block diagram of reference WRPC Timing System

Status: working & tested in the WR switch, HPLL already works on S6 (on PFC carrier).
Todo: Test Xilinx ports, fix some bugs in DMPLL, jitter optimization.

4.6. WB Interconnect

Classic Wishbone matrix interconnect, such as WB Conmax from OpenCores.

Status: stable
Todo: add optional access control

4.7. CPU Core

ZPU CPU core.

Status: done.
Todo: define memory layout and CPU <-> Wishbone slave port communication model (*mailboxes in system mem?*)

4.8. GPIO port

CPU-controlled GPIO port, providing access to status and control signals.

Todo: write or reuse something already existing

4.9. PPS generator

Real-time clock & PPS generator.

Status: working and tested
Todo: some register layout fixes as suggested by Alessandro, Xilinx port, timecode output.