White Rabbit Node

Functional Specifications

GSI Helmholtzzentrum für Schwerionenforschung GmbH, BEL

Tibor Fleck, Cesar Prados

CERN, European Organization for Nuclear Research, BE-CO

Jean-Claude Bau

March 28, 2011

Contents

[1]This depends on the form factor of the WRN and does not apply for standalone WRNs.

Revision History Table

Version	Date	Authors	Description
0.1	26/10/2010	T.F., C.P.	first draft
0.2	29/10/2010	T.F., C.P.	first draft for discussion
1.0	08/12/2010	T.F., C.P., J.B.	first revision after detailed discussion
1.1	24/03/2011	T.F., C.P., J.B.	further details, adapt to redundancy spec., counter section

List of Acronyms

WR	:	**W**hite **R**abbit
WRP	:	**W**hite **R**abbit **P**rotocol
WRPTP	:	**W**hite **R**abbit **PTP**
WRIB	:	**W**hite **R**abbit **I**nformation **B**lock
WRCT	:	**W**hite **R**abbit **C**onfigurable **T**imer
WRCC	:	**W**hite **R**abbit **C**onfigurable **C**ounter
WRDC	:	**W**hite **R**abbit Configurable **D**own **C**ounter
WRCU	:	**W**hite **R**abbit **C**ounting **U**nit
WRUC	:	**W**hite **R**abbit Configurable **U**p **C**ounter
AWRT	:	**A**bsolute **W**hite **R**abbit notion of **T**ime
WRN	:	**W**hite **R**abbit **N**ode
WRS	:	**W**hite **R**abbit **S**witch
WRCM	:	**W**hite **R**abbit General **C**lock **M**aster Node
WRMM	:	**W**hite **R**abbit **M**anagement **M**aster Node
WRDM	:	**W**hite **R**abbit **D**ata **M**aster Node
FESA	:	**F**ront-**E**nd **S**oftware **A**rchitecture
NIC	:	**N**etwork **I**nterface **C**ontroller
SNMP	:	**S**imple **N**etwork **M**anagement **P**rotocol
DHCP	:	**D**ynamic **H**ost **C**onfiguration **P**rotocol
PTP	:	**P**recision **T**ime **P**rotocol
FMC	:	**F**PGA **M**ezzanine **C**ard
FPGA	:	**F**ield **P**rogrammable **G**ate **A**rray
FEC	:	**F**orward **E**rror **C**orrection
UDP	:	**U**ser **D**atagram **P**rotocol
LCD	:	**L**iquid **C**rystal **D**isplay
LED	:	**L**ight-**E**mitting **D**iode
IRQ	:	**I**nterrupt **Req**uest

IEEE standards A short overview with some description of existing and possibly relevant IEEE specifications. All specifications cited within this specification are set in *italic*.
Specifications marked with *: current versions.
Specifications marked with ∘: versions currently in progress.

IEEE 802: LAN/MAN Overview and Architecture. Standards Committee (LMSC). Develops Local Area Network (LAN) and Metropolitan Area Network (MAN) standards, mainly for the lowest 2 layers of the ISO Reference Model for Open Systems Interconnection (OSI).

- **802-2001:** (R2007) IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture
- **802a-2003:** Amendment 1: Ethertypes for Prototype and Vendor - Specific Protocol Development
- **802b-2004:** Amendment 2: Registration of Object Identifiers
- **802a:** Playpen Ethertypes

IEEE 802.1: LAN/MAN bridging and management. Higher Layer LAN Protocols Working Group. Concerned with 802 LAN/MAN architecture; internetworking among 802 LANs, MANs and other wide area networks; 802 Link Security; 802 overall network management; protocol layers above the MAC and LLC layers.

- ***802.1ab-2009:** Station and Media Access Control Connectivity Discovery
- ∘**802.1AC:** Media Access Control Service revision. Revise the documentation of existing services (IEEE 802.1D, IEEE 802.1Q and ISO/IEC 15802-1) within a single, common service definition standard.
- ***802.1ad-2005:** Provider Bridges (amendment to 802.1Q-1998)
- ***802.1AE-2006:** MAC Security
- ***802.1ag-2007:** Connectivity Fault Management (amendment to IEEE 802.1Q-2005)
- ***802.1ah-2008:** Provider Backbone Bridges
- ***802.1aj-2009:** Two Port MAC Relay (TPMR)
- ***802.1ak-2007:** Multiple Registration Protocol (MRP)

- ***802.1ap-2008:** Management Information Base (MIB) definitions for VLAN Bridges
- ○**802.1aq:** Shortest Path Bridging
- **802.1aq-2007:** Connectivity Fault Management
- ***802.1AR-2009:** Secure Device Identity (DevID)
- ○**802.1AS:** Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. Based on IEEE 1588
- ***802.1AX-2008:** Link Aggregation (previously called 802.3ad)
- **802.1B (1992):** LAN/MAN Management; withdrawn in 2004
- **802.1D-2004:** *Media access control (MAC) Bridges (Incorporates IEEE 802.1t-2001 and IEEE 802.1w). Amendment 802.16k-2007. Covers all parts of the Traffic Class Expediting and Dynamic Multicast Filtering*
- ***802.1H, 1997 Edition:** Media Access Control (MAC) Bridging of Ethernet V2.0 in Local Area Networks (Adopted by the ISO/IEC and redesignated as ISO/IEC TR11802-5:1997(E))
- **802.1H-REV.:** Recommended Practice for MAC Bridging of Ethernet in LANs
- **802.1p: Part of the IEEE standard 802.1D. The 802.1p standard covers traffic class expediting and dynamic multicast filtering part of MAC bridges, known as IEEE 802.1D.** Extends the concept of MAC Bridging in order to define additional capabilities in Bridged LANs: (1) Provision of expedited traffic capabilities to support transmission of time-critical information in a LAN environment (2) Provision of filtering services that support dynamic use of Group MAC addresses in a LAN environment.
- ***802.1Q-2005:** VLAN bridges. ***Part of the IEEE standard 802.1D, defining an architecture for Virtual Bridged LANs and services provided in Virtual Bridged LANs.** *Defines the operation of VLAN Bridges that permit the definition, operation and administration of Virtual LAN topologies within a Bridged LAN infrastructure. No Quality of Service mechanisms are defined in this standard, but an important requirement for providing QoS is included in this standard.*
- **802.1Q-2005/Cor1-2008:** Technical corrections for Multiple Registration Protocol
- ○**802.1Qbc:** Provider Bridging: Remote Customer Service Interfaces

- ∘**802.1Qbe:** Multiple Backbone Service Instance Identifier (I-SID) Registration Protocol (MIRP)
- ∘**802.1Qbf:** PBB-TE Infrastructure Segment Protection
- **802.1Q-REV:** Revision of 802.1Q-2005
- ***802.1Qat:** Stream Reservation Protocol
- ***802.1Qau-2010:** VLAN Amendment 13: Congestion Notification
- ***802.1Qav-2009:** VLAN Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams
- ***802.1Qaw-2009:** VLAN Amendment 9: Management of Data Driven and Data Dependent Connectivity Faults
- ***802.1Qay-2009:** VLAN Amendment 10: Provider Backbone Bridge Traffic Engineering
- ∘**802.1Qaz:** Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes
- ∘**802.1Qbb:** Priority-based Flow Control
- ∘**802.1Qbg:** Edge Virtual Bridging - VEPA (Virtual Ethernet Port Aggregator)
- ∘**802.1Qbh:** Bridge Port Extension / VN-Tag
- **802.1s:** superseded by 802.1Q-2005. Multiple Spanning Trees. Adds the facility for VLAN bridges to 802.1Q to use multiple spanning trees, providing for traffic belonging to different VLANs to flow over potentially different paths within the virtual bridged LAN.
- **802.1X-2004:** Port-Based Network Access Control. (802.1X-2010 approved).

IEEE 802.2: LAN/MAN logical link control.

- **802.2-1998:** Specific requirements – Part 2: Logical Link Control

IEEE 802.3: LAN/MAN CSMA/CDE (Ethernet) access method.

- ***802.3-2008:*** *Carrier Sense Multiple Access with Collision Detection (CMSA/CD) Access Method and Physical Layer Specifications. Sections One, Two, Three, Four, Five.*
- **802.3-2008/Cor1:** Corrigendum 1: Timing Considerations for PAUSE Operation

- **802.3at-2009:** Amendment 3: Data Terminal Equipment (DTE) Power via the Media Dependent Interface (MDI) Enhancements.
- **802.3av-2009:** Amendment 1: Physical Layer Specifications and Management Parameters for 10 Gb/s Passive Optical Networks
- **802.3bc-2009:** Amendment 2: Ethernet Organizationally Specific Type, Length, Value (TLVs)
- **802.3bd:** MAC Control Frame for Priority-based Flow Control

further IEEE standards

- ***1588-D2.2 2007:*** *Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

Other Standards

- **RFC791-1981:** INTERNET PROTOCOL – PROTOCOL SPECIFICATION
- **RFC 2460-IPv6-1998:** Internet Protocol, Version 6 (IPv6) Specification
- **RFC 1213-1991:** Management Information Base for Network Management of TCP/IP-based Internets: MIB-II
- **RFC 2131-1997:** Dynamic Host Configuration Protocol for IPv4
- **RFC 2132-1997:** DHCP Options and BOOTP Vendor Extensions
- ***RFC 768-1980:*** *User Datagram Protocol*
- **Recommendation ITU-T G.8261/Y.1361 - 04/2008:** Timing and synchronization aspects in packet networks; "Synchronous Ethernet".

SNMP V1

- **RFC 1155-1990:** Structure and Identification of Management Information for TCP/IP-based Internets
- **RFC 1156-1990:** Management Information Base for Network Management of TCP/IP-based Internets
- **RFC 1157-1990:** A Simple Network Management Protocol (SNMP)

SNMP V2

- **RFC 1901-1996:** Introduction to Community-based SNMPv2
- **RFC 1905-1996:** Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)
- **RFC 1906-1996:** Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)

SNMP V3

- **RFC 3410-2002:** Introduction and Applicability Statements for Internet-Standard Management Framework
- **RFC 3411-2002:** An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks

- **RFC 3412-2002:** Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
- **RFC 3413-2002:** Simple Network Management Protocol (SNMP) Applications
- **RFC 3414-2002:** User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
- **RFC 3415-2002:** View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
- **RFC 3416-2002:** Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)
- **RFC 3417-2002:** Transport Mappings for the Simple Network Management Protocol (SNMP)
- **RFC 3418-2002:** Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)

- **"Shall"**, which is equivalent to "is required to", is used to indicate mandatory requirements, strictly to be followed in order to conform to the standard and from which no deviation is permitted.
- **"Recommended"** is used to indicate flexibility of choice with a strong preference alternative.
- **"Must"** indicates an unavoidable situation.
- **"Should"**, which is equivalent to "is recommended that", is used to indicate:
  - Among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others
  - That a certain course of action is preferred but not required.
  - That (in the negative form) a certain course of action is deprecated but not prohibited.
- **"May"**, which is equivalent to "is permitted", is used to indicate a course of action permissible within the limits of the standard.
- **"Can"**, which is equivalent to "is able to", is used to indicate possibility and capability, whether material or physical.

1. Introduction

White Rabbit is intended to be the next-generation deterministic network based on synchronous Ethernet, allowing for low-latency deterministic packet routing and transparent, high precision timing transmission. The network consists of White Rabbit Nodes, White Rabbit Switches and supports integration of nodes that are not White Rabbit Nodes, however with restrictions.

White Rabbit Switches are interconnected through GbE fibers or GbE twisted-pair copper cables. Tree topology should be used, but other topology can also be used. Details for the latter situation are described in A.

At least one White Rabbit Node shall have special functionality, however this may be done by configuration only without the need of special White Rabbit Node hardware.

From the functional point of view three special White Rabbit Nodes shall exist that may be implemented in one, two or three individual White Rabbit Nodes. Of this type of nodes only one of each type shall be active at a time:

- A White Rabbit General Clock Master Node (WRCM) shall provide time and frequency reference.
- A White Rabbit Management Master Node (WRMM) shall cover all network related function together with functionality to monitor and configure all White Rabbit Nodes.
- A White Rabbit Data Master Node (WRDM) should be used whenever control information has to be created centrally and fanning-out to all other White Rabbit Nodes is mandatory like in the use case for particle accelerator control.

Detailed definition of these three special White Rabbit Nodes is given in chapter 4.

The receiver within White Rabbit, the White Rabbit Node (WRN), is a device that (i) is syntonized/synchronized to its clock master, (ii) can decode control information coming from the WRDM or any other WRN and (iii) has an absolute notion of time equivalent to that of the WRCM[1]. In dependence of the WRPTP mechanisms used the clock master of an individual White Rabbit Node may be its direct link partner (White Rabbit Switch) or the

[1] with a maximum deviation as specified in TBD

WRCM. The WRCM itself should connect to some GPS receiver (or any other device providing both frequency and time reference) as its clock and time master but may be its own clock master as well.

This document describes the functional requirements and functionality that all White Rabbit Nodes must accomplish.

Typically a White Rabbit Node will be an interface card in one of several specified form factors. In addition standalone White Rabbit Nodes will exist for special use cases.

2. Node Functional Requirements

A White Rabbit Node is a networking device that shall provide:

- Hardware and software support for White Rabbit Protocol.
- Hardware support for Synchronous Ethernet as specified in *ITU-T G.8261/Y.1361*.
- Hardware and software support for PTP.
- OSI layer 2 support.
- Bus communication with a present host card[1].
- Support for a set of protocols in the OSI layer 3.
- Diagnostic, management, debug and test functionalities.
- Control Event Handling.
- Communication with up to two FPGA mezzanine cards.
- A powerful FPGA and at least one soft CPU (specified in detail in a later version of this specification).

2.1 Layer 1 Requirements

- A WRN shall be able to replicate the master clock received from its master with a maximum skew of 100 ps.

2.2 Layer 2 Requirements

- A WRN shall be compliant with Ethernet (IEEE 802.3).
- A WRN shall implement *IEEE 802.1Q*.
- A WRN shall follow the conditions for "Critical Broadcast Traffic" as specified in the WR robustness specification [WRR].
- A WRN shall follow the de-/encoding algorithm specified in [FEC].

[1] Details depend on individual WRN form factors while no bus shall be connected to standalone WRNs

- As soon as all Ethernet frames belonging to one encoded control message have been received by a WRN it shall be possible to decode the original control message in less than $10\mu s$ using algorithms specified in [FEC].
- A WRN shall be possible to encode control messages in less than $10\mu s$ using algorithms specified in [FEC].
- A WRN shall be able to receive and process data up to the limit of the WR link of 1Gbit/s.
- A WRN shall be able to decode control messages up to an upper mean limit of 500MBit/s.

2.3 Upper Layer Requirements

- A WRN shall be able to adjust its phase as specified in [WRS].
- A WRN shall implement an absolute notion of time (AWRT) that shall be synchronized following the WR protocol specification, [WRS].
- A WRN notion of time (AWRT) shall consist of two 32bit values: One 32bit value for the seconds since 01.01.1970 and one 32bit value for the fraction of each second.
- A WRN shall implement the Internet Protocol (IP) as specified in *RFC791-1981*.
- A WRN shall implement UDP as specified in *RFC768-1980*.
- A WRN shall accomplish the White Rabbit Protocol (or the White Rabbit PTP Extension), [WRS].
- A WRN shall implement SNMP as specified in ....
- A WRN shall implement DHCP as specified in *RFC2131-1997*.
- A WRN shall implement EtherBone as specified in [ETB].

2.4 Diagnostics

- A WRN shall implement all network diagnostic functionality to comply with the WR robustness specification, [WRR].
- A WRN shall implement SNMP to report its status to the WRMM.

- A WRN shall implement a monitoring system as specified in [WRR] and can report all relevant monitoring data to the WRMM. A WRN shall be able to report this data also to its bus master (host card).
- A WRN shall implement hardware and software support for sFlow as specified in [WRR].
- In addition to network monitoring and diagnostic functionality a WRN shall implement monitoring and diagnostic functionality for all further functionality as specified in this document including status data of connected FMC cards.

2.5 Management Requirements

- A WRN shall implement all management functionality to comply with the WR robustness specification, [WRR].
- A WRN shall provide mechanism for remote software and firmware updating and upgrading from the WRMM.
- A WRN shall implement SNMP for remote maintenance by the WRMM.
- A WRN shall implement functionality (to be defined) for remote configuration by the WRMM.
- A WRN shall implement DHCP as specified in *RFC 2132-1997*.
- It shall be possible to access all WRN functionality using the host bus communication and its device driver.
- It shall be possible to access all WRN functionality using the WR network link and WR communication as specified in section 2.8.

2.6 Debug and Test Functionalities

- A WRN shall implement both physical interfaces and mechanism for debug purposes.
- A WRN shall implement automated test procedures for verification of its connectivity, memory, power supply and interfaces as described in chapter 3.13 on startup.
- A WRN shall implement functionality to remotely start test procedures as described in chapter 3.13.

2.7 Fault Condition / Error Handling

In a future version of this specification a list of fault conditions and errors will be added with respective links.

- A WRN shall implement mechanisms to detect, log and report its own fault conditions in addition to those fault conditions specified in [WRR].
- A WRN shall implement mechanisms to detect, log and report all sorts of errors (e.g. erroneous configuration) in addition to those fault conditions specified in [WRR].
- A WRN shall implement a possibility to remotely restart its host if the host supports this feature.
- It shall be possible to remotely restart a WRN.
- A WRN shall implement means to detect and report if its own clock is syntonized to its WR clock source (WRS) or not.
- A WRN shall implement means to detect and report its PTP status.
- A WRN shall implement means to report all errors regarding layer 3 protocols.
- A WRN shall count the number of
  - flawed but still decoded frames.
  - non decoded frames.
  - non-received frames. This information can be deduced from the FEC header, compare [FEC].
  - number of lost control messages due to bit errors.
  -

2.8 Datagram Content and WRN Reaction

Datagram in this context describes information that is sent from one WRN (e.g. the WRDM) to one, some or all WRNs. The term 'information block' will be used synonymously and abbreviated as 'WRIB' (White Rabbit Information Block) and may consist of e.g. device commands, device configuration data or data relevant for the WRNs host. The term 'event' that is often used in the field of accelerator controls, may be used to indicate special WRIB content.

- A WRN shall encapsulate all its WRIB content in EtherBone as specified in [ETB].

- A WRN shall be able to interpret all WRIB content that is specified in [EVT].
- A WRN shall be able to receive and filter WRIB content received from the WRDM.
- A WRN shall be able to update its interpretation of WRIB content. This new information may be received from its WRN host card or from special WRIBs. The latter is recommended to originate from the WRDM.
- A WRN shall be able to react on WRIB content matching its own WRIB filter configuration.
- A set of possible actions of a WRN upon receiving special WRIB content shall include:
  - Loading of configurable counters with predefined configurations.
  - Generation of different kinds of front panel trigger pulses [2].
  - Generation of IRQs on the backplane bus to be processed by its Host Card making use of WRCC preset to 0.
  - Trigger actions in connected mezzanine cards.
  - Access to all its remote accessible functionality.
  - Read out of post mortem buffers.
- Device command reaction in a WRN shall include the possibility to initiate internal test procedures.
- A WRN shall provide full access to all its functionality from WRIB content.
- A WRN shall provide full access to all its configuration from WRIB content.
- A WRN shall be able to send the complete WRIB content or parts of it to its host card.
- A WRN shall be able to interpret external trigger inputs like special WRIB content specified in [EVT].
- A WRN shall be able to simulate WRIB content receipt to test device functionality without a WRDM present or even without a WR link established.
- A WRN shall be able to delay individual WRIB content reaction by usage of configurable counters as specified in 2.10.

[2] Coaxial LEMO 00series NIM-CAMAC standard

- A WRN shall be able to create and send out WRIBs.
- A WRN shall be able to schedule events as specified in the Granularity Window Concept Specification [GWC].
- It shall be possible to directly access specific memory spaces within a WRN using special WRIB content as specified in [ETB]. It shall also be possible to configure a WRN on how to react on data written to such memory spaces.

2.9 White Rabbit Configurable Timers (WRCT)

A WRN shall implement configurable timers with 8ns resolution and functionality as specified below[3]. WRCTs may be used to trigger special WRN reaction at predefined absolute times.

2.9.1 Properties

All WRCTs shall

- store at least its last 10 values together with the information if an output has been activated and the time of that output.

2.9.2 Timer I/O

Timer Outputs

The output of a WRCT shall be either of the following:

- Direct output of a single TTL pulse to one of the available LEMO connectors[4].
- Creating an IRQ for its Host Card on the backplane bus.
- Creating a command for the soft CPU.
- Input for a WRCC.

2.9.3 Configuration

It shall be possible to configure a WRCT to

- use one of the possible outputs.

[3]Part of this chapter belongs to implementation but is specified here for the time being to clarify one of the most important WRN functionality.

[4]For other than single TTL pulse output the output of the WRCT shall be input to a WRCC.

2.9.4 Command Reaction

Each WRCT shall react to the following commands with the specified behavior:

- **set:** The set value is written to the WRCT.
- **read:** The current value of the WRCT is read. This must not influence the WRCTs compare to the AWRT.
- **clear:** A WRCT is set to zero by this command and will thus not output anything before a new set command reactivates it.

2.9.5 Behavior

- A WRCT shall constantly compare its value with the AWRT.
- If a WRCT value is equal to the AWRT it shall trigger its output once.
- If a WRCT value is less than the AWRT and if its value has not been equal to AWRT since the last set command it shall trigger its output once.
- It shall be possible to read and set the value of a WRCT.
- If a WRCT is set with a value smaller than the AWRT an error shall be generated. It shall be possible to process this error as specified in section 3.12.

2.10 White Rabbit Configurable Counters (WRCC)

A WRN shall implement configurable counters with 8ns resolution and functionality as specified below[5]:

- Counters with 32bit resolution shall exist as "downcounters" (WRDC) that decrement by one with each (clock) tick as specified in 2.10.4.
- Counters with 16bit resolution shall exist as "upcounters" (WRUC) that increment by one with each (clock) tick as specified in 2.10.4.
- It shall be possible to combine WRCCs.

[5]Part of this chapter belongs to implementation but is specified here for the time being to clarify one of the most important WRN functionality.

2.10.1 Properties

All WRCCs shall

- have two modes: "single shot" or "repetitive".
- have one preload value that shall be initialized with zero.
- have one special activation flag. It shall be configurable if a WRCC reacts on its activation flag or not.
- store the timestamps of at least its last 10 outputs together with the corresponding counters values.

2.10.2 Counter I/O

Clock Inputs

- It shall be possible to configure a WRCC to be clocked with one out of several internal clocks. The maximum allowed frequency for any WRCC input clock shall be 125MHz.

External Clock / Trigger Inputs

- It shall be possible to configure a WRCC to be clocked with an external clock. The maximum allowed frequency for an external clock shall be 50MHz.
- It shall be possible to configure a WRCC to react on external trigger inputs.
- For external clock /trigger inputs it shall be configurable if the signal is high-active or low-active.
- For external clock /trigger inputs it shall be configurable that a WRCC reacts only on the signal if it is active for a minimum amount of time. This time shall be configurable and initialized with 300ns.

Other Inputs

- A WRCC shall react to the following commands: "preload", "start", "stop", "pause", "reset", "clear", "read". The individual WRCC reaction to this commands is specified in 2.10.4.
- This commands shall originate from FPGA logic, the output of another WRCC or shall be initiated by an external trigger depending on the WRCC configuration.
- A WRCC shall react on its activation flag if configured accordingly.

Outputs

The output of a WRCC shall be either of the following:

- Direct output to one of the available LEMO connectors.
- Creating an IRQ for its Host Card on the backplane bus.
- Creating a command for the soft CPU.
- Input for another WRCC.

2.10.3 Configuration

It shall be possible to configure a WRCC to

- use one of the internal or external clock inputs.
- use one of the possible outputs.
- react on its activation flag or not.

2.10.4 Command Reaction

Each WRCC shall react to the following commands with the specified behavior:

- **preload:** The preload value is written to the WRCC. A running WRCC must not react on a preload command.
- **read:** The current value of the WRCC is read. This shall also be possible for a running counter and must not influence it.
- **start:** This starts the WRCC or continues a stopped counter. A WRUC will start incrementing its current value by one with every assigned clock tick while a WRDC will decrement its current value by one with every assigned clock tick.
- **reset:** The WRCC is set to its preload value. This shall also be possible if the counter is running.
- **clear:** The WRCC is set to zero. This shall also be possible if the counter is running.
- **stop:** A running WRCC is stopped and its current value together with the current timestamp is written to the WRCC history buffer.
- **pause:** A running WRCC is stopped.

2.10.5 Behavior

- A WRCC shall increment or decrement by 1 with every clock tick of its assigned input clock.
- A WRCC that is configured to react on its activation flag shall only increment or decrement by 1 on its assigned input clock tick when the activation flag is set. It shall reset its activation flag at the same time.
- A running WRDC shall stop at the value zero and
- A WRDC shall always react when reaching zero.
- A WRUC shall always react when reaching its maximum value of xFFFF according to its configuration.
- A WRCC shall always react when reaching the value of its output compare register according to its configuration.
- A WRN shall implement counters that can be started via some WRIB content ("event").
- It shall be possible to preload configurable counters with a given number of clock ticks. In the case of down counters preloading specifies the delay of the counters output reaction after being started.
- It shall be possible to input configurable counters with different clocks.
- It is recommended that most configurable counters use the standard clock derived from WR.
- It shall be possible to assign different configurable counters to different outputs.
- It shall be possible to combine different counter outputs to one output using e.g. OR, XOR, AND rules.

3. White Rabbit Node Functionality

A WRN is a WR network device that synchronizes/syntonizes its clock to the next master by using WRPTP and Sync Ethernet. It can encode, decode and process Ethernet Frames with control information and communicates with other WRNs. In this chapter WRN functionalities are specified.

3.1 Layer 1 Functionality

- A WRN shall recover the traceable clock encoded by the immediate superior master of the node and shall base its timekeeping on it.

3.2 Layer 2 Functionality

- A WRN shall examine all frames received on the WR Ethernet link.
- A WRN shall provide functionality to tag outgoing packets according to user priorities as specified in IEEE 802.1D.
- A WRN shall generate Raw Ethernet frames.
- A WRN shall provide functionality to encode any WRIB into different frames for transmission as specified in [FEC]. For each WRIB to be transmitted it must be indicated whether or not is has to be encoded.
- A WRN shall numerate encoded frames according to the policy specified in [WRR].
- A WRN shall be able to detect if a received frame is part of an encoded WRIB as specified in [FEC]. In this case a WRN shall provide the functionality to collect all associated frames and restore the original WRIB.
- A WRN is recommended to schedule its transmission according to the Granularity Window Concept [GWC]. In this case a WRN shall still be able to transmit frames at arbitrary times.

3.3 Counting Functionality

Apart from WRCCs as specified in 2.10 a WRN shall implement 'counting units' (WRCU) with 32bit resolution for different usage, e.g. to count lost encoded frames, count the number of sent packets or to be used as diagnostic counters to provide watchdog functionality. These counters may be implemented in software. A WRCU shall implement the following functionality:

- It shall be possible to set, read, clear and increment a WRCU.
- A WRCU shall increment by one only by an increment command.
- A WRCU shall have one compare value. It shall be possible to read and set a WRCUs compare value.
- It shall be possible to generate and process an error as specified in 3.12 when an WRCU equals its compare value.

3.4 Clock Synthesis

- A WRN should be able to synthesize an external clock[1] and fan-out so-called "tuning words" enabling other WRNs to locally reproduce the original clock. The WRCM must implement this functionality.
- A WRN shall be able to locally reproduce (re-synthesize) such clocks with high accuracy regarding frequency and phase.

3.5 White Rabbit Functionality

- A WRN shall carry out what is specified for a Slave Clock in the White Rabbit Specification, in short:
  - Accept the clock hierarchy in the WR network as a Slave Clock.
  - Carry out the WRPTP protocol and the calibration process.
  - Adjust its clock using the offset and delay calculated by its Master Clock.

3.6 Upper Layer Functionality

- A WRN shall provide functionality to schedule upstream traffic in order to fulfill use requirements like e.g. [GWC].
- A WRN shall follow the White Rabbit Flow Control Concept as specified in the White Rabbit Robustness specification [WRR].

[1] Valid frequency range specified in [WRNT].

- A WRN shall be able to automatically readjust its clock according to WRPTP.
- A WRN shall be able to readjust its clock according to WRPTP only when a corresponding command from the WRDM is received.
- It shall be configurable which of this two functionalities will be used for clock adjustments.
- A WRN shall implement MIB for SNMP as described within this document in a later version.
- A WRN shall implement EtherBone as specified in [ETB].
- A WRN shall provide several status registers.

3.7 Soft CPU functionality

A WRN shall provide the following functionalities:

- A WRNs soft CPU shall provide a master wishbone v.? interface as specified in ???.
- A WRNs soft CPU shall provide a slave wishbone v.? interface as specified in ???.
- It shall be possible to program the soft CPU of a WRN using a C/C++ compiler.
- Assisting to configure all WRCCs.
- Assisting in updating the interpretation of WRIB content.
- It is recommended that a WRN soft CPU runs RT LINUX.
- Processes for diagnostic, fault condition and error handling.

3.8 Host Card Communication[2]

- A WRN shall be able to clock the host bus communication with its WR adjusted clock if its host system supports this functionality.
- A WRN shall provide the means to communicate with a Host Card.
- A WRN shall provide status information (Temp, SW FW, Voltages etc...) to its Host Card.

[2]This depends on the form factor of the WRN and does not apply for standalone WRNs.

- A WRN shall provide the means to change its configuration sent from the Host Card.
- A WRN shall propagate its error/general status to the Host Card.
- A WRN shall be able to create bus IRQs.
- A WRN shall be able to communicate its absolute notion of time to its Host Card if a request for time synchronization is received. It shall make use of standard (form-factor dependent) protocol based bus communication. In this case the accuracy of the synchronized Host Card time shall be better than $50\mu s$.
- A WRN shall be able to communicate its absolute notion of time to its Host Card using a combination of trigger pulse and Host Card bus communication:
  - The WRN shall create a single TTL pulse on one of its LEMO connectors if a request for time synchronization is received from its Host Card.
  - Protocol based standard bus communication shall be used to inform the Host Card about the absolute time of this pulse before the pulse is created.
  - It shall be possible to add an individual positive offset to the absolute time that is communicated to minimize constant deviation introduced by the short connection cable.
  - The accuracy of the synchronized Host Card time shall be comparable to or better than 1ns using this mechanism.

3.9 Mezzanine Card Communication

- A WRN shall propagate error status of (a) connected Mezzanine Card(s) to its Host Card.
- A WRN shall propagate measured data of (a) connected Mezzanine Card(s) to its the Host Card.
- A WRN shall propagate configuration data of (a) connected Mezzanine Card(s) to its Host Card.
- A WRN shall propagate configuration data to (a) connected Mezzanine Card(s) that can be provided from different sources (WR network link, Host Card).

3.10 Management Functionality

To be defined

- ...

3.11 Parametrization/Configuration Functionality

- A WRN shall provide functionality to remotely change its parameters and configuration from different sources (WR network link, host bus communication) in a coherent way.
- A WRN shall provide functionality to treat inconsistencies of configuration data or parameters originated from different sources (WR network link, host bus communication).
- A WRN shall be able to generate and propagate a general overview of its parameters and configuration.
- On power-up a WRN shall be able to load a predefined standard configuration set.
- Parametrization of a WRN shall include values for the granularity window and FEC parameters.
- A WRN shall start with some minimal configuration.
- A WRN shall be configured to download its configuration from its host card or from the WRMM.

3.12 Fault Condition Behavior / Error Handling

- A WRN shall be able to detect discrepancies in configuration from different sources (WR network link, host bus communication).
- A WRN shall be able to treat discrepancies in configuration.
- A WRN shall be able to enter in error states and notify to the appropriate master (Host Card, WRDM, WRCM or WRMM).
- A WRN shall implement internal error watchdogs.
- A WRN shall be able to display error information making use of LCD or LEDs.

3.13 Debug and Test Functionalities

- A WRN shall use a startup procedure including configuration and initial self tests. **Startup procedure has to be defined.**
- A WRN shall be able to run test procedures for verification of its connectivity, memory, power supply and interfaces. **Define which tests have to be performed on startup OR remotely started.**
- A WRN shall be able to notify the appropriate master (Host Card, WRDM, WRCM or WRMM) about results of internal test procedures.
- It shall be possible to remotely start automated test procedures within a WRN.

4. WR Node as a Master

- A WRN shall load the default configuration on power-up, and its functionality has been described in chapters 2 and 3.
- A WRN may carry out a special role in a WR network besides the standard WRN, thanks to specific configuration. These roles are:
  - WR General Clock Master Node, it's the time and frequency source for the complete WR network.
  - WR Data Master Node, it's the control data source for the complete WR network.
  - WR Management Master Node, it's the management source for the complete WR network.

The different functionalities of these special types of WRNs are specified below.

4.1 WR General Clock Master Node

- A WRCM shall implement and carry out what applies to a Grandmaster Clock in the White Rabbit PTP Specification, in short:
  - A WRCM shall establish the clock hierarchy in the WR network and establish itself as Grandmaster Clock.
  - A WRCM shall initiate the WRPTP protocol and the calibration process.
  - A WRCM shall calculate the Link Delay between clocks.
  - A WRCM shall issue the WRPTP Sync messages.
  - A WRCM shall implement functionality to resynchronize nodes only at times provided by the WRDM. The WRCM shall give feedback about all nodes synchronization status to the WRDM.
  - A WRCM shall be able to carry on the root role for the Rapid Spanning Tree protocol.
- A WRCM must always be located on top of a WR tree topology.

4.2 WR Data Master Node

- A WRDM shall generate a constant stream of control information in the case of accelerator control usage. It may be used even for general WR applications.
- A WRDM shall determine the duration and beginning of Granularity Windows in the case of accelerator control usage. It may be used even for general WR applications.
- A WRDM shall Code/Decode information according to the FEC.
- A WRDM shall process incoming information from other WRNs and react according to its configuration.

4.3 WR Management Master Node

- A WRMM shall implement the role of a server for network management protocols, namely SNMP.
- A WRMM shall distribute software and firmware updates to WR network devices (WR Switches and WRNs) within the WR network.
- A WRMM shall supervise the updating process and react accordingly in case of errors.
- A WRMM shall implement the DHCP Server.
- A WRMN shall implement sFlow collector as of the Flow Monitor system.
- A WRMN shall monitor the status of all connected WRNs regarding their syntonization and synchronization status.
- A WRMN shall communicate to the WR devices the Flow Control Policy.

Bibliography

[FEC] Forward Error Correction Algorithms to be used in White Rabbit networks (to be published).

[WRR] White Rabbit and Robustness - Specification.

[WRS] White Rabbit Specification (to be discussed).

[GWC] Granularity Window Concept Specification (to be published).

[WRNT] White Rabbit Node Technical Specification (to be discussed).

[EVT] White Rabbit Accelerator Event Specification (to be published).

[ETB] Functional Specification: EtherBone Core, September 2010, CERN BE-CO-HT, J.Lewis.
`http://www.ohwr.org/projects/white-rabbit`

A. Spotting the Master in the WR Network

As it has been explained in Chapter 4, a node can carry out a set of special roles: WRCM, WRDM or WRMM. Each one of these roles can fall only on one WRN in the network. Either all this Master are concentrated in one WRN or spread in 2 or 3 WRNs. If a WRN is configured to such Master functionality, the spot where this Master is placed within the WR network may affect White Rabbit properties and performance[1]. For that reason some good White Rabbit practices must be followed depending on where this individual Masters Nodes are placed: on top of the network or not, and in turns, and if they are concentrated in one WRN or are distributed.

A.1 Tree Topology, the Masters on the top

ToDo → Picture of the Masters on top

No matter if the Master is one or more devices, this configuration doesn't jeopardize the requirements established since:

- The WR traffic shall flow from top-to-bottom.
- The maximum latency shall be defined by the number of WRS layers, although observe the document [WRR]

A.2 Tree Topology, the Master high and low

ToDo → Picture of all Masters in one device but not on top. (Fig. 1)
ToDo → Picture of the Masters spread around the network. (Fig. 2)

Both in the case of Fig. 1 and Fig. 2, the following considerations should be taken into account:

A.2.1 WR latency packets

If the Data Master Node is placed in a non-top position:

[1] This applies only for WRDM and WRMM since WRCM must always be on top of a tree topology

- The delay, of the longest path between the Data Master Node and a node, should not be higher than the time bounds imposed by the Granularity Window.
- The different path delays for the same WR packet must not be higher than ...

.....

.....

.....

This Appendix will be completed after this document has been discussed

B. FESA host card control of a WRN