



**Detailed Specification of the  
FAIR Accelerator Control System Component  
„Timing Receivers“**

*Document Name*  
**F-DS-C-06e**

*Date yyyy-mm-dd*  
**2014-03-02**

### **Abstract**

This document describes the Detailed Specification of the accelerator control system component "Timing Receivers". It defines the development, production, acceptance tests and delivery of all Timing Receivers for the FAIR facility. This document covers all items of PSP codes 2.14.10.3.3.1 and 2.14.10.3.3.2.

*FAIR Division/Group/Supplier*  
**Controls Department**

*Prepared by:*  
**GSI Timing Team**

**Table of Contents**

1.	Purpose and Classification of the Document .....	4
1.1.	Responsibilities .....	4
1.2.	Classifications of Requirements .....	4
2.	Scope of the Technical System .....	5
2.1.	System Overview .....	5
2.2.	Limits of the System and Environment .....	7
2.2.1.	Compatibility to PCIe FTRN PEXARIA5 .....	7
2.2.2.	Interfaces .....	8
2.2.3.	Limits .....	8
2.2.4.	Environment .....	10
2.3.	Basis of Concept .....	10
2.3.1.	Functional Requirements .....	11
2.3.2.	Non-functional Requirements .....	13
2.3.3.	General Constraints .....	15
2.3.4.	Architectural Principles .....	15
3.	Technical Specifications .....	15
3.1.	Common Features .....	16
3.1.1.	Hardware .....	16
3.1.2.	Start up, Gateware, Firmware and Configuration .....	16
3.1.3.	Core Modules .....	17
3.1.4.	Configuring Actions .....	17
3.1.5.	Waveform Generators .....	17
3.1.6.	Timestamp Latch Unit .....	17
3.1.7.	Signal Generation .....	18
3.1.8.	Monitoring .....	19
3.1.9.	Interface to the Host .....	19
3.1.10.	Drivers and Test Software .....	19
3.2.	FTRN Form Factors .....	21
3.2.1.	Common Features for all Form Factors .....	22
3.2.2.	VME64x FTRNs .....	26
3.2.3.	MTCA FTRNs .....	26
3.2.4.	PMC Mezzanine WR Receivers .....	28
3.2.5.	PCIe FTRNs .....	28
3.2.6.	Standalone FTRNs .....	29
3.3.	Mezzanine Modules .....	29
3.4.	WR Starter Kit .....	29
3.5.	Device Drivers .....	29
3.6.	Other Issues .....	29
3.6.1.	WR Network on Copper .....	29
3.6.2.	Phase of BuTiS w2 Clock .....	30
3.6.3.	microSD Slots .....	30
3.6.4.	Host Bus Bridging .....	30
3.6.5.	Bluetooth .....	31
3.6.6.	Fine Delay on FTRNs .....	31
3.7.	Open Issues .....	31
3.7.1.	Rate Divider .....	31
3.7.2.	Run-time Gateware Check .....	31
3.7.3.	Local and Remote Operation of Standalone FTRNs .....	31
3.7.4.	Distribution of High-Precision Sampling Clocks .....	31
4.	Quality Assurance, Tests and Acceptance .....	32
4.1.	FTRN Life Cycle .....	32
4.1.1.	Installation .....	32

**Document Title:** Detailed Specification Timing Receivers

4.1.2.	Commissioning.....	33
4.1.3.	Operation .....	33
4.1.4.	Maintenance .....	33
4.2.	Development Methodology .....	34
4.2.1.	Basic Concept.....	34
4.2.2.	Relationship Between Partners .....	34
4.2.3.	Design Concept.....	34
4.2.4.	Hardware Follows Gateware .....	34
4.2.5.	Reporting .....	34
4.2.6.	Good Practice .....	35
4.2.7.	Infrastructural Changes and Reconciliations .....	35
4.2.8.	Road Map .....	36
4.2.9.	Intellectual Property and Licensing Issues.....	38
4.3.	Quality Assurance System of the Supplier .....	38
4.4.	FAT.....	38
4.5.	SAT .....	39
5.	Documentation .....	40
6.	Warranty.....	40
7.	Scope of Delivery .....	40
8.	Acronyms and Definitions .....	43
I.	Attached Documents .....	45
II.	Related Documentation .....	45
III.	Document Information .....	47
III.1.	Document History .....	47

**List of Tables**

Table 1:	List of FTRN and supporting HDL/software functional requirements .....	13
Table 2:	List of Timing Receiver Non-functional Requirements .....	15
Table 3:	FTRN drivers (status February 2014). .....	21
Table 4:	Form factors and their I/Os. ....	22
Table 5:	Common features for all form factors.....	24
Table 6:	Important electronic components.....	24
Table 7:	Properties of single-ended signals.....	25
Table 8:	Properties of differential signals.....	25
Table 9:	VME64x form factor.....	26
Table 10:	MTCA form factor. ....	28
Table 11:	PMC form factor. ....	28
Table 12:	Road Map.....	38
Table 13:	FAT Items.....	39
Table 14:	Deliverables.....	42

**List of Figures**

Figure 1:	Hosted FTRN system architecture.....	6
Figure 2:	System architecture of a standalone FTRN. ....	7
Figure 3:	Software Boundaries on the FTRN host system. ....	8
Figure 4:	Boundaries on the FTRN HDL. Only the host bus bridge is in-scope. (this figure shows only most relevant modules).....	9
Figure 5:	Example: A FEC receives a MSI from the TLU.....	18

## 1. Purpose and Classification of the Document

The purpose of this document is to describe

1. *what kind of* products are needed,
2. a solution based on the White Rabbit (WR) technology [4] and
3. the solution from the technical and management point of view.

This document defines primary requirements and specifications of White Rabbit based FAIR Timing Receivers, according to the requirements of the FAIR accelerator control system [2].

The work package is divided into different phases, see section 4.2.8. At the end of each phase, certain milestones must be achieved. A confirmation of milestone completion must be issued by FAIR before the contractor may progress to the next phase. Since the White Rabbit technology is still under development, some milestone details in the next phase can only be assessed after the preceding milestone has been reached.

Whenever regulations and requirements are specified in the General Specifications, Technical Guidelines or Common Specifications of the Control System they are only referenced in this document. The related documents are listed in Appendix II.

No legal or contractual conditions are treated in this document. All related information is given in the General Specifications for FAIR [1].

### 1.1. Responsibilities

The Timing Group (CSCOTG) of the GSI Helmholtz Centre for Heavy Ion Research GmbH (GSI) Darmstadt is responsible for changes to this document. In this document, this group is designated as **GSI Timing Team**.

For initial information please contact the administration of the Controls Department.

Further information on the organigram, names of responsible persons and task leaders, as well as the agreed document release and approval procedure is summarized in the organizational note 'Controls Project for FAIR'.

The contractor is expected to have experience in development and operation of accelerator control system equipment.

### 1.2. Classifications of Requirements

The following definitions of requirement classifications are being used throughout the document:

- **"Must"** or **"shall"** or **"is required to"** are used to indicate mandatory requirements, strictly to be followed in order to conform to the standard and from which no deviation is permitted.
- **"Must not"** or **"shall not"** mean that the definition is an absolute prohibition of the specification.

- “**Should**” or “**is recommended**” are used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others or that a certain course of action is preferred but not required.
- “**Should not**” or “**is not recommended**” mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighted before implementing any behavior described with this label.
- “**May**”, which is equivalent to “**is permitted**”, is used to indicate a course of action permissible within the limits of the standard.

## 2. Scope of the Technical System

### 2.1. System Overview

The FAIR General Machine Timing (GMT) system is a component of the accelerator control system and has the responsibility of synchronizing accelerator actions throughout the whole accelerator facility [3]. The GMT utilizes the White Rabbit PTP (WR-PTP) protocol [4], [10], [17].

The GMT will be based on a network with a tree topology with a timing master on top, consisting of switches and Timing Receivers. All participants synchronize their internal oscillators and local time to the clock master node using White Rabbit (WR).

This document focuses on these Timing Receivers, which are called FAIR Timing Receiver Nodes (FTRN). FTRNs receive and decode broadcast network messages in real time. These are sent by a data master, which is a component of the timing master, located on top of the WR network topology using the EtherBone [14] protocol.

Each FTRN is configured by third party software, typically Front-end System Architecture (FESA) [6], to execute a set of actions known in advance. These actions are scheduled for execution by the data master, which broadcasts commands containing absolute timestamps (such as Temps Atomique International – TAI) denoting the time of action execution and references to an action. When a broadcast command is received, FTRNs enqueue the designated action to be executed locally at the specified time.

Furthermore, all FTRNs shall be able to send and receive over the WR network low priority network messages for management, diagnostics and maintenance. Support for sending high priority network messages shall also be present, e.g. for beam dump notification.

Additional bus interfaces and local controls are also foreseen in many form factors.

FTRNs have the following features:

- clock and time synchronization with a clock master according to the WR specifications

- ability to record arrival time of a digital pulse (e.g. LVTTTL)
- ability to generate a digital pulse or a pulse sequence at a given time
- ability to generate clock signals with a given frequency and phase
- ability to generate specified interrupts at given times

Figure 1 depicts the system architecture of a FTRN and its hosting platform.

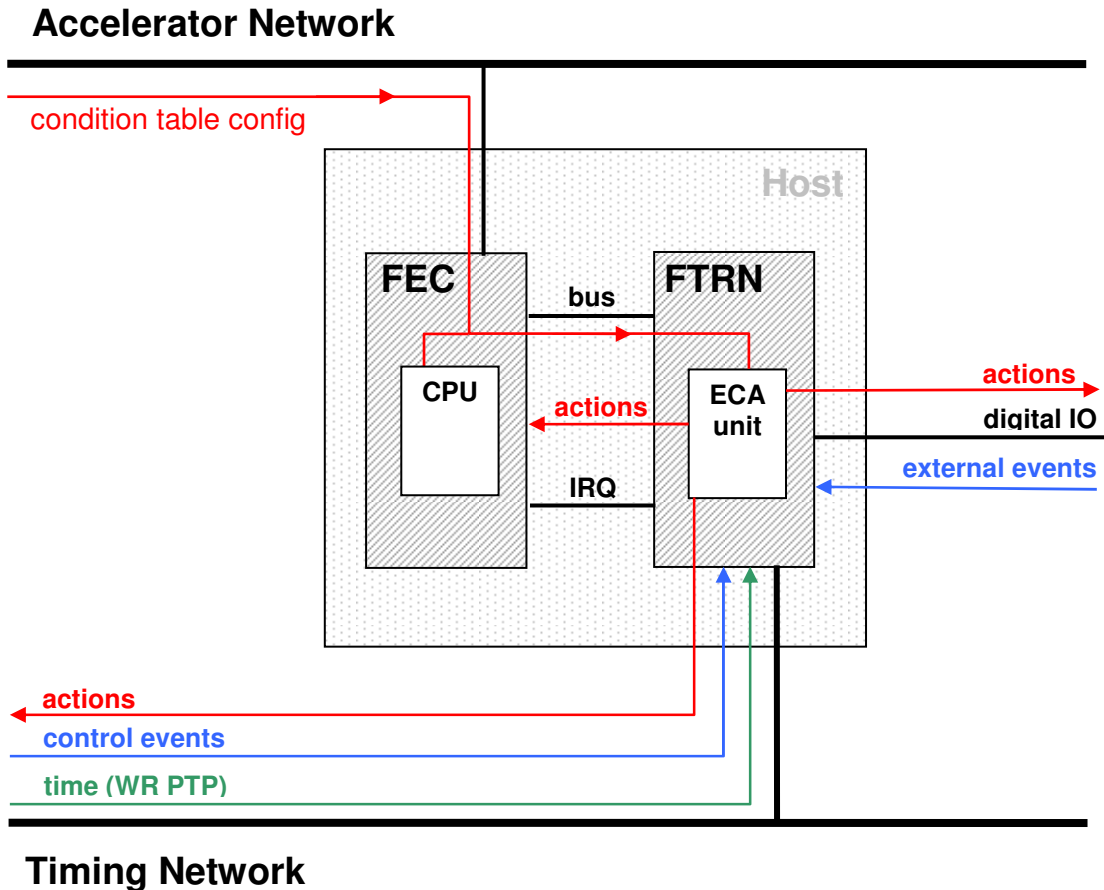


Figure 1: Hosted FTRN system architecture

The FTRN and a **front-end controller** (FEC) are embedded in a host. The FEC is connected to the ACCelerator NETwork (ACCNET) while the FTRN is connected to the dedicated timing network. FEC and FTRN are interconnected via the host system bus and interrupt lines. FTRN time synchronization is achieved by the WR-PTP protocol over the timing network. Coded instructions for FTRNs are typically generated by the data master (not shown) and are called **commands**. Commands come in using EtherBone as messages broadcasted over the timing network or via FEC request. Execution of actions follows the **Event-Condition-Action (ECA)** model [7]: **Conditions** define how FTRNs map the event<sup>1</sup> of an incoming **command** to an **action**. **Actions** drive accelerator equipment by generating digital output, FEC bus interrupt or timing network messages. FTRN actions are typically configured via the FEC CPU. The **ECA**

<sup>1</sup>In the document the word „event“ is used in the original sense: „event“ = „something happens“

**unit** executes actions on time. The time of execution is specified by a timestamp that is part of the command received from the data master.

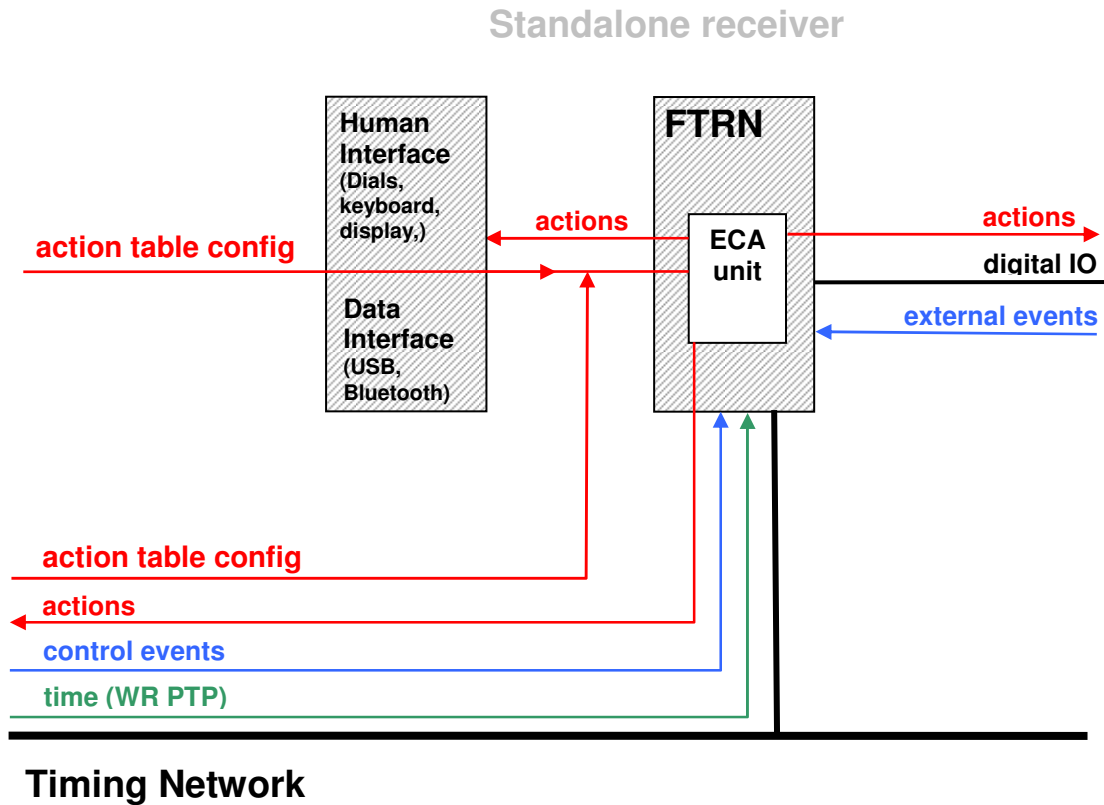


Figure 2: System architecture of a standalone FTRN.

FTRNs shall be available in different form factors for operation within different types of crates or PCs and as standalone devices. The latter are shown in Figure 2. Here, the FEC is replaced by a human interface (knobs, display,...). A standalone FTRN is not embedded in a host system.

## 2.2. Limits of the System and Environment

This section describes what is not in the scope of the work-package and the constraints imposed by the environment.

### 2.2.1. Compatibility to PCIe FTRN PEXARIA5

The existing FTRN PEXARIA5 [25] and its mezzanine board PEXARIA5DB [33] is the reference implementation for all in-scope FTRNs, see section 3.2.

### 2.2.2. Interfaces

Everything on the far side of the main FTRN interfaces is out-of-scope. The four main interfaces are

- the WR network layer
- the bus interface
- the host independent mezzanine board, if supported by the carrier
- the host dependent I/O connectors (e.g. LEMO Series 00)

### 2.2.3. Limits

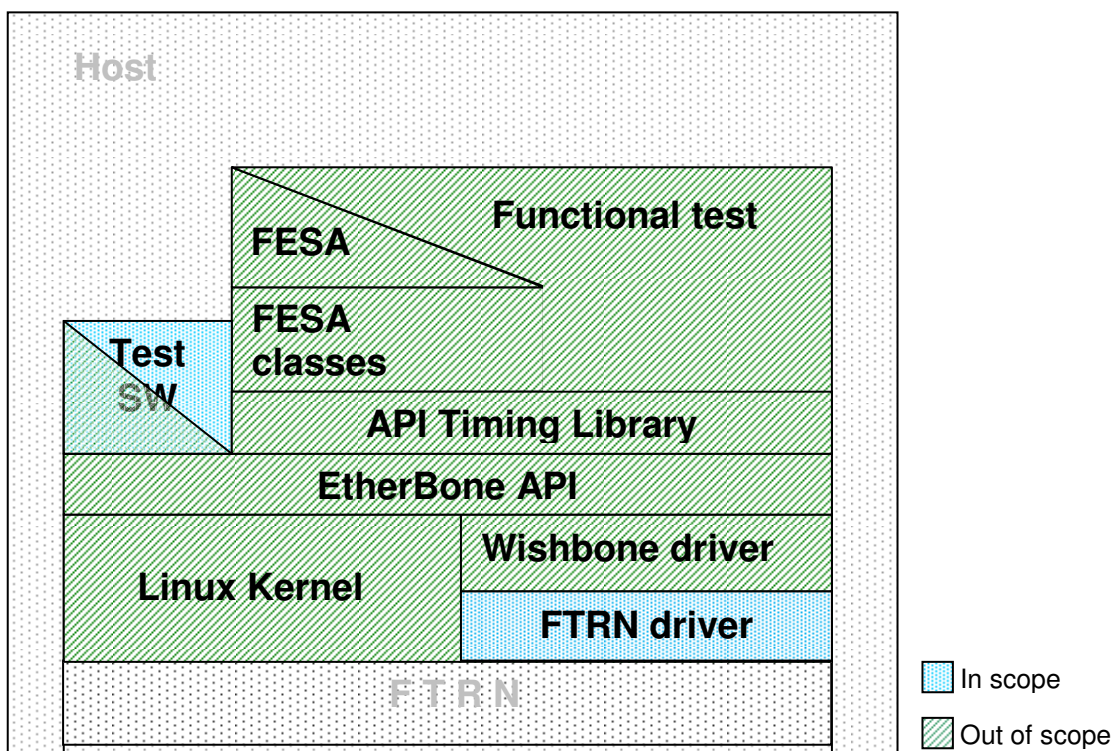


Figure 3: Software Boundaries on the FTRN host system.

#### Software

Figure 3 shows the software stack of the FEC. The FTRN Linux device driver uses resources of the Linux Kernel (driver subsystem) to communicate with the FTRN over the host bus (which is form factor specific). It provides low level access to the FTRN including the Wishbone bus [22] and interrupt handling. Furthermore, it has an interface to a form factor independent Wishbone driver.

The Wishbone driver provides a userland software Wishbone (WB) interface including a 32-bit memory map, cycle error and control lines.

Abstraction to the communication layer is realized by the EtherBone API, which provides register I/O to Wishbone devices and serves as the interface towards userland applications. It provides abstract methods to access components of the FTRN WB bus.



## Document Title: Detailed Specification Timing Receivers

An API Timing Library (ATL) implements an interface towards third party user software such as a FESA class. The ATL abstracts the FTRN by hiding its Wishbone architecture and address layout. It provides high level functionality of the FTRN. Moreover, the ATL only provides the functionalities of the FTRN which is required by the concept of the control system. This implies that other functionalities already implemented by the FTRN may intentionally be hidden.

The Test SoftWare (TSW) shall be developed and implemented for testing gateway and hardware of a FTRN. The TSW uses the EtherBone API as an interface to the FTRN. The main contractor implements TSW for form factor Independent features (TSWI). The contractor implements TSW for form factor Dependent features (TSWD).

The main contractor (GSI) has already implemented the EtherBone API, Wishbone driver and FTRN drivers for PCIe, VME and USB interfaces. Moreover, all FTRN form factors can be accessed through the EtherBone API via the White Rabbit network and USB.

Of the software stack of the FEC, only the form factor Dependent Test SoftWare TSWD and the FTRN driver are in-scope. All software must be developed for Scientific Linux 6 or later and support 32bit and 64bit. All software must be compatible to real-time patches.

### HDL

Boundaries on FTRN Hardware Description Language (HDL) are illustrated in Figure 4. The sole task of the contractor is to implement host bus to WB master and slave bus interfaces and to connect it to one of the WB crossbars. Other HDL is out-of-scope.

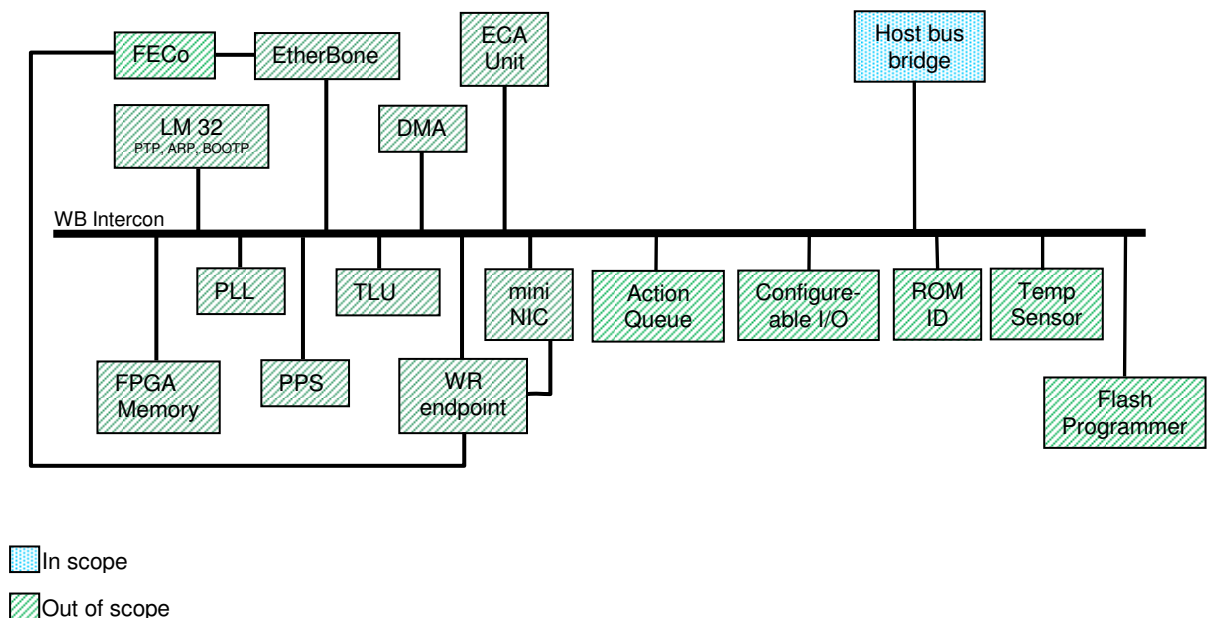


Figure 4: Boundaries on the FTRN HDL. Only the host bus bridge is in-scope.  
(this figure shows only most relevant modules)

All WR switches are out-of-scope; however they are needed as fundamental infrastructure for the implementation of nodes. When a FTRN is connected to a

## **Document Title:** Detailed Specification Timing Receivers

switch, it shall establish a link according to the WR protocol. Furthermore, it shall integrate into the General Machine Timing System seamlessly, as described later in this document. Thus, FTRNs shall be developed in close coordination with the GSI Timing Team. Hardware and software developments shall be implemented in accordance with the CERN Open Hardware licensing scheme [5] and GPL [9] respectively.

### **2.2.4. Environment**

The environment of the FTRNs beyond the four main interfaces consists of

- the WR timing network (~3000 FTRNs, switches ...)
- the host environment: crates, PCs or other devices in which FTRNs are integrated, e.g. receivers in VME form factor operate within VME crates
- accelerator equipment connected to FTRN I/O connectors

The network environment is based on Gigabit Ethernet and has some special features such as

- deterministic delivery of control messages,
- reliable delivery, with packet loss rate less than  $10^{-12}$ ,
- geographical extensions of 2 km and
- approximately five layers of WR switches.

The host environment of the FTRN depends on the form factor. Limits on

- power supply
- cooling performance
- mechanical hosting
- bus communication

are imposed by the form factor standards (e.g. VME, MTCA ...).

The FTRN I/O connectors can send or receive digital signals. I/O standards are defined in section 3.2.

FTRNs will operate in the environmental conditions specified for their host systems. Exposure to significant radiation levels is not considered. Environmental endurance for standalone FTRNs is yet to be specified.

Section 2 assumes that FTRNs are connected to WR switches using optical links.

## **2.3. Basis of Concept**

The FAIR timing system is specified in the FAIR Detailed Specification “General Machine Timing System” [3].

Relevant to Timing Receivers are the following technical specification documents:

- WR specifications [10]: describe the WR network, in particular the clock synchronization protocol, known as WR-PTP [4], [17].

**Document Title:** Detailed Specification Timing Receivers

- WR node functional specifications [11]: describe the Timing Receiver functionality for a general purpose node.
- FAIR Common Specifications “Accelerator Control System” [2].

### 2.3.1. Functional Requirements

Entries in the following table are divided into **In-Scope** and **Out-of-Scope** items. The hardware provided by the contractor must support implementing the out-of-scope items. Items out-of-scope will be provided by the main contractor. Informative: The GSI Timing Team has already done some preparatory work on items TR\_012, TR\_013 and TR\_040.

Number	Description of the Requirement
TR_000	<b>Scheduled digital signal generation</b> – Digital signals (e.g. TTL, LVDS) must be available on connectors. FTRNs shall act as programmable wave form generators with a WB slave interface. When requested by the ECA unit, they shall produce a pulse or a pulse sequence as well as clock signals of variable phase and frequency. It must be possible to run slow serial protocols (I2C, SPI, UART, MIL).
TR_009	<b>Actions</b> – FTRNs shall support at a minimum the following actions: host system messaging (TR_010-TR_013), dispatching network messages (TR_180), and waveform generation (TR_130, TR_000).
TR_010	<b>Action queue</b> – The FTRN shall maintain a time stamped buffer for messages from the ECA unit to the FEC. The queue shall be readable from the FEC, which then marks entries as processed.
TR_011	<b>Overflow detection</b> – The FTRN device driver must detect action queue overflow and report this to the API timing library.
TR_012	<b>Interrupt generation</b> – FTRNs shall generate interrupt signals to the FEC of the host system, when the action queue is not empty. Interrupt masking shall be supported by a register in the FTRN. This enables performing actions on the host system synchronized with accelerator operations.
TR_013	<b>Polling mode</b> – The ATL must implement a “polling mode”, if the host bus does not support interrupts.
TR_020	<b>ECA unit</b> – The Event-Condition-Action (ECA) unit processes incoming commands and initiates appropriate actions. Conditions and actions in the ECA are programmed over the WB bus / FEC by FESA. They specify which commands are transformed to actions on this FTRN, taking into account local information.
TR_030	<b>Action Interface</b> – The ECA unit outputs actions conforming to the action interface format. Receiving components accept actions in this format.
TR_040	<b>Host Bus Bridge</b> – A host bus to Wishbone bridge shall be implemented. This allows the FEC to configure WB slaves in the FTRN.
TR_050	<b>USB-Interface</b> – All FTRNs must have a USB slave interface. As a WB master, it configures WB slaves in the FTRN.

**Document Title:** Detailed Specification Timing Receivers

TR_060	<b>Manual controls</b> – In case the FTRN is not connected to a host-system, a Human Interface Device (HID) like manual knobs or displays should allow configuring WB slaves in the FTRN. The assignment of HID components to configuration parameters must be programmable.
TR_070	<b>Logging</b> – FTRNs shall implement an interface that allows to record their command and action history for post mortem analysis and other purposes.
TR_080	<b>Sharing</b> – The ATL must support simultaneous access from multiple userland applications on the FEC.
TR_100	<b>External events</b> – A FTRN should be able to handle events from an external interface like a digital input. As an example, this allows triggering actions by beam monitors. As part of the gateway, a configurable component should be developed that generates Wishbone “writes” upon a signal received over a digital input line.
TR_110	<b>Delays</b> – It shall be possible to delay all input and output signals.
TR_120	<b>I/O selection</b> – All bi-directional connectors shall be flexibly configured to be either an input or an output.
TR_130	<b>Configuration block of waveform generators</b> – The waveform generator produces signals on the output connectors. A waveform is configured with an FPGA-like composition of AND/OR/MUX/... components linking action inputs (from the ECA unit) to digital outputs. Examples of waveforms include “Single Gate”, “Single Burst”, “Toggling Gate”, “Toggling Burst” and “1/N Bursts of N Pulses” as described in [29]. Whenever the configurable components are connected in series, there is a register between them.
TR_140	<b>Status register</b> – In order for the FEC to verify correct operation, FTRNs shall have an accessible status register and shall issue an interrupt if particularly critical conditions occur (e.g. timing synchronization was lost). The ATL must inform the front-end software if a Timing Receiver ceases to work correctly or the communication is disrupted.
TR_150	<b>LEDs</b> – Various LEDs shall be provided to allow easy installation, operation start up and diagnostics. In particular, it is necessary to have general error LEDs and WR network LEDs showing status of the link, of the clock synchronization and of the PLL. Additional LEDs or a tiny front display should provide additional information. See section 3.1.
TR_160	<b>Reset</b> – FTRNs shall be resettable via EtherBone. A reset button on the front panel is not required.
TR_170	<b>Gateway and firmware update</b> – It shall be possible to update the gateway and firmware via EtherBone (WR network, host bus bridge, USB,...) or directly (e.g. via JTAG).
TR_180	<b>Dispatching network messages</b> – A FTRN shall be able to send messages of high priority via the timing network. This is triggered by an action.
TR_190	<b>Timestamp latching</b> – A FTRN shall be capable of latching a timestamp, if it is configured accordingly. Latching a timestamp can

**Document Title:** Detailed Specification Timing Receivers

	be triggered by an electrical signal (LVTTTL, LVDS...) at one of the input connectors.
TR_195	<b>Timestamp availability</b> – It shall be possible to query a timestamp containing the actual time via EtherBone.
TR_200	<b>BuTiS [20] clock generation</b> <ol style="list-style-type: none"> <li>A FTRN should be able to synthesize a <b>w0</b> signal that has the same clock and phase as the BuTiS <b>T0</b> clock [20]. For this, it receives tuning words from the data master (connected to BuTiS). The accuracy of the generated w0 clock shall be sufficient for the identification of <b>c2</b> clock cycles, see TS_430 and TS_410 [3].</li> <li>A FTRN should be able to synthesize a <b>w2</b> clock signal related to the BuTiS <b>c2</b> clock. That clock signal should be frequency and phase matched to the <b>c2</b> clock generated by a BuTiS receiver.</li> </ol> <p>Info: The WR clock of the GMT is phase locked to the BuTiS clock.</p>
TR_210	<b>Timestamp distribution</b> – It shall be possible to provide timestamps via a serial protocol on an output connector (LVTTTL, LVDS...). As an example, this could be encoded between the generated w0 pulses (“At the next tone, it will be ...”).
TR_220	<b>Form factor dependent I/O</b> – Some form factors shall implement I/O connectors on the carrier board. Those will be specified together with the GSI Timing Team before the Printed Circuit Board (PCB) layout.
TR_230	<b>Identification of hardware, gateware and firmware</b> – Every FTRN and mezzanine shall provide a unique serial number, model and revision. Every gateware and firmware (bitstream, soft-CPU code...) shall provide information including type, version number, author, fitness for the particular hardware and possible unauthorized changed. The ATL provides means of verifying the compatibility between the components.
TR_240	<b>Persistence</b> – All FTRN configurations shall be saveable locally. This is required to fulfill requirement TS_190 of the GMT [3].
TR_250	<b>WR</b> – All FTRNs shall be compliant to the WR specifications [10].
TR_260	<b>EtherBone</b> – All FTRNs shall be compliant to EtherBone [14].
TR_270	<b>Forward Error Correction (FECo)</b> – All FTRNs shall support FECo for data received from and send to the timing network.
TR_280	<b>IP Addresses</b> – All FTRNs obtain their IP addresses via BOOTP.
TR_290	<b>Verification of gateware and firmware</b> – It should be possible to detect run-time modification of firmware and gateware. Such a run-time check shall be investigated, see section 3.7.2.

Table 1: List of FTRN and supporting HDL/software functional requirements.

**2.3.2. Non-functional Requirements**

Entries in the following table are divided into **In-Scope** and **Out-of-Scope** items. The hardware provided by the contractor must support implementing the out-of-scope items. Items out-of-scope will be provided by the main contractor.

**Document Title:** Detailed Specification Timing Receivers

Informative: The GSI Timing Team has already done some preparatory work on items TR\_350 and TR\_360.

Number	Description of the Requirement
TR_310	<b>Maintenance</b> <ol style="list-style-type: none"> <li>All form factors embedded in a host system shall support loading of gateware, firmware and configuration by the host system.</li> <li>All form factors shall support loading their gateware and firmware from flash and shall support gateware and firmware update via EtherBone.</li> </ol>
TR_320	<b>Safety</b> – FTRNs shall be able to send safety related information to the host bus, to the WR network as high priority messages and on the I/O connectors as digital signals. This is of interest for special cases like transmitting beam dump notifications to the post mortem system.
TR_325	<b>I/O GMT compliance</b> – The signal properties at I/O connectors of FTRNs shall be compliant to fulfil the requirements of the GMT [3]. The PCB layout and electronic circuit design connecting I/O connectors and FPGA pins shall consider the rise-time at outputs, the edge-detection at inputs and the intrinsic delays. Specifically, the following requirements of the GMT shall be supported. <ol style="list-style-type: none"> <li>The jitter should be 100 ps or less. Jitter shall be 500 ps or less (TS_390).</li> <li>The precision shall be 1 ns or better (TS_400).</li> <li>The accuracy shall be sufficient for unambiguous identification of BuTiS c2 clock cycles (TS_410).</li> <li>BuTiS clock signals and their properties (TR_200 and TS_430).</li> </ol> <p>The terms jitter, precision and accuracy are defined within the specifications of the GMT [3]. The intrinsic delays for I/O signals between the connectors and the FPGA pins as well as the delays due to signal processing in the FPGA shall be determined and made known to the GSI Timing Team.</p>
TR_330	<b>Real time reliability</b> – FTRNs shall have prior knowledge of the intrinsic delays for actions connected to relevant commands, in order to guarantee timely execution or to signal a missed deadline. Sources of intrinsic delays include HDL execution, signal propagation in PCBs and electronic circuits connecting FPGA pins to I/O connectors.
TR_335	<b>Local delay compensation</b> – FTRNs shall be able to delay or pre-trigger the execution of actions within limits defined by the GMT. Local delays are typically requested by third party software such as a FESA class to compensate for timing issues outside the scope of the main interfaces, see section 2.2.2.
TR_340	<b>Clock synchronization</b> – Each FTRN shall provide a PPS output in order to verify the clock synchronization between WR devices. A synchronization of 1 ns or better shall be achieved.
TR_350	<b>Scalability</b> – It shall be possible to operate many FTRNs on the same host bus



**Document Title:** Detailed Specification Timing Receivers

TR_360	<b>Debugging</b> – It shall be possible to use signal analyzers to debug HDL and debuggers on soft CPU processes.
TR_370	<b>Monitoring</b> – It shall be possible to monitor FTRNs on the WR network by means of a management interface.
TR_380	<b>QR-code</b> – Each FTRN shall have a QR sticker, recognized by notebook / smartphone for configuration or diagnosis.
TR_390	<b>Design concept</b> – Functionality should be implemented in the FPGA, not via dedicated electronics outside the FPGA.
TR_400	<b>Hardware follows gateway</b> – The design of the hardware must suffice gateway already developed by the GSI Timing Team, see section 4.2.3. The suitability of the hardware concept by the contractor will be confirmed by the GSI Timing Team by approval of the engineering specifications.

Table 2: List of Timing Receiver Non-functional Requirements

**2.3.3. General Constraints**

The following constraints apply.

- The contractor shall work closely together with GSI Timing Team.
- All development and production tools must be compatible with GSI infrastructure.
- VHDL shall be the preferred HDL programming language
- Only widely supported tools shall be used
- All project material shall be kept under version control

**2.3.4. Architectural Principles**

The Software Architecture Guideline for the Control System [21] fully applies.

**3. Technical Specifications**

The FTRNs shall be based on and compliant to WR technology.

FTRN engineering specifications shall be written by the contractor based on [2], [19], [10], [11], [23], and possible future functional extensions. They shall be written for each one of the form factors mentioned in this chapter. They are reviewed and approved by the control system timing supplier (FAIR).

They shall be worked out considering the life cycle of the FAIR Timing System (see section 4.1).

### **3.1. Common Features**

#### **3.1.1. Hardware**

All FTRNs shall feature a unique FTRN carrier board serial number, which shall be obtained via a one-wire temperature sensor.

All FTRNs have a unique vendor specific MAC address. Private MAC addresses must not be used. On each FTRN, the MAC address is stored in an EEPROM on the carrier board. The type of EEPROM is specified by the GSI Timing Team and the supplier will deliver FTRNs with MAC addresses and make the MAC address for each device known to the GSI Timing Team.

A “Mean Time Between Failure” (MTBF) calculation for each FTRN type shall be performed by the contractor.

The mechanical design shall allow quick replacement of FTRN carrier boards or mezzanine boards by a technician using standard tools (e.g. screwdriver) without the need for special tools or soldering.

Other mandatory common features are listed in section 3.2.1.

The hardware design of the different FTRN form factors should avoid redundant developments; all circuits and components should be as similar as possible.

#### **3.1.2. Start up, Gateware, Firmware and Configuration**

FTRNs need two kinds of start-up data: gateware and firmware (mandatory) and configuration (optional). The gateware and firmware is intended to be delivered and modified only by developers, while the configuration consists of parameters saved by the user.

All form factors support loading their gateware and firmware from flash and support gateware and firmware update via EtherBone.

All form factors embedded in a host system shall support loading of gateware, firmware and configuration by the host system.

All form factors shall support saving/loading configuration data on/from FTRN EEPROM.

FTRNs shall provide two different regions in the flash memory. One stores the gateware and firmware loaded on normal power up. The other stores a factory default gateware and firmware which is loaded either when the normal power-up gateware is broken or when one of the buttons on the PCB is pressed. Both types of gateware and firmware shall include the following:

- Wishbone firmware programmer
- Programmer for on-board gateware and firmware flash
- EtherBone networking
- Host bus bridging support



In case a FTRN features a host bus bridge chip, a hardwired connection to the FPGA programmer is required (see also section 3.6.4). WR calibration data and MAC address are stored in an on-board EEPROM on the carrier board.

### **3.1.3. Core Modules**

FTRNs in all form factors share a set of core modules provided by the WR community and by the GSI Timing Team. An overview of the provided core modules can be found in Figure 4.

### **3.1.4. Configuring Actions**

Configuration of actions is mainly performed over the bus, accessible from FEC, human interface, timing network.

Configuration of actions shall always be based on EtherBone.

All configurable gateway components relevant for implementation of actions (e.g. pulse generators) shall be Wishbone devices.

It shall be possible to schedule action executions simultaneously by means of commands with the same execution timestamp, provided the actions are handled by distinct ECA channels or timing receivers.

It shall be possible to disable/enable the ECA unit.

### **3.1.5. Waveform Generators**

All FTRNs shall have waveform generators. Each waveform generator shall provide the features described in Table 1.

### **3.1.6. Timestamp Latch Unit**

An external signal from an input port shall trigger the Timestamp Latch Unit (TLU) to store a timestamp of the current time. The TLU has multiple channels. Timestamps are stored in a FIFOs and can be retrieved via EtherBone.

The TLU can also be used as a configurable Message Signalled Interrupt (MSI) source. By this, it can be used to trigger another component upon an incoming digital signal as shown in Figure 5.



Figure 5: Example: A FEC receives a MSI from the TLU.

### 3.1.7. Signal Generation

I/Os are connected to the FTRN's FPGA and only digital signals are used. Dedicated circuits for each I/O serve to configure the direction and provide a buffer to protect the FPGA. In the following, some background information is given.

#### System Clocked

Here, the logic is implemented in the same clock domain as the 125 MHz system clock. This is the most general and flexible case. If a appropriate VHDL component is implemented, direction, source and sink of signals can be configured on-the-fly. Sources and sinks could include items such as I/O connectors, trigger input to the TLU, output of ECA action channels or input and output of simple logic operations (AND, OR, NOT). Due to the 125 MHz clock, signals can be generated and processed within 8 ns. As an example the length of gate signals is a multiple of 8 ns and the period of generated clock signals have a multiple of 16 ns.

#### Driven by Differential I/O Buffers

System clocked signals can possibly be individually phase shifted by using the output buffers of differential signals on the FPGA, which can be clocked to 1.25 GHz. Using this technique, the implementation of fine delay with 1ns resolution is straight forward for input as well as for output signals. For output, the fine delay value can be changed on-the-fly within one 125 MHz clock cycle for all outputs of the concerned ECA channel. For input, signals of different connectors are sampled individually.

#### Driven by WR PLL

If higher granularity or phase shifting is required, the PLL of the WR clock can be used. This is as special case and imposes limitations on the number of signals that can be generated and it is not guaranteed that this resource is available at all. This method allows generating clock signals with a period of 1 ns multiples and a defined phase relative to the 125 MHz system clock. Besides restrictions in

---

**Document Title:** Detailed Specification Timing Receivers
 

---

the PLL itself, the generated signals must be connected in a clock domain different from the 125 MHz system clock, which prevents relevant I/O to be configurable on-the-fly. As an example, this method can be applied for generating the BuTiS w2 clock. Another example would be the implementation of a fine-delay option for output signals with a granularity of 125 ps. As a drawback, a set-up time of up to 10  $\mu$ s has to be considered when changing the phase. Furthermore, connections between PLL and output connectors are defined by VHDL and cannot be changed after the synthesis. This prevents using concerned connectors for other purposes.

### **Driven by Delay Lines with Differential Output Buffers**

Combining the existing 1ns delay at the differential output buffers with delay chains might possibly allow fine delay with granularity of down to 25 ps. This needs to be investigated, but it is most likely only available on the FPGAs of the ArriaV GX family.

### **Driven by a Dedicated PLL**

For more freedom in the frequency of generated clock signals, a dedicated PLL can be used. However, the generation of clock signals from the system clock must suffice certain ratios and the specified frequency range of involved oscillators. Like for the WR PLL, this imposes restrictions on the configurability of I/Os. The availability of this resource is even more restricted and it depends on the schematics of a FTRN, to which I/O connector the signal can be connected. It must be stated that the implementation of such signals is clearly outside the specifications of GMT and FTRNs. It is mentioned here only for completeness.

## **3.1.8. Monitoring**

FTRNs should implement a management interface (e.g. SNMP).

## **3.1.9. Interface to the Host**

On some kinds of host systems, the FEC needs a bus controller in order to communicate with FTRNs. Normally this bus controller is integrated in the CPU module, which is also the bus master.

The engineering specification shall clarify this issue, especially for VME, MTCA and PMC, because FTRN drivers depend on the bus controller. However, the FTRN should depend on as few of the peculiarities of a given bus controller as possible, adhering strictly to the relevant standards and thus maximizing compatibility.

## **3.1.10. Drivers and Test Software**

In combination with the common Wishbone driver, the form factor specific FTRN drivers make FTRNs controllable over Linux operating systems. FTRN drivers for the following form factors are required, see section 2.2.3. Table 3 gives an overview on the FTRN drivers. Entries are divided into **In-Scope** and **Out-of-Scope** items.

**Document Title:** Detailed Specification Timing Receivers

<b>Number</b>	<b>Description of the Requirement</b>
TR_1000	<b>VME</b> – A FTRN driver for a VME host bus (see [26]) has been developed by the GSI Timing Team. Status (February 2014): The driver has been derived from the CERN VME64x driver; release candidate.
TR_1010	<b>USB</b> – A FTRN driver for the USB bus common to all form factors shall be developed. Status (February 2014): done and released
TR_1020	<b>PCI/PCIe</b> – A FTRN driver for the PCI/PCIe bus common to many form factors (see [24], [25] and section 3.2.3) shall be developed. Status (February 2014): done and released
TR_1030	<b>VME64x</b> – A FTRN driver for the specified VME64x form factor (see section 3.2.2) is required. It shall be confirmed, if the existing VME driver (TR_1000) can be used. If not, the FTRN driver for VME64x shall be derived from the VME driver TR_1000.
TR_1040	<b>PMC</b> – A FTRN driver for the specified PMC form factor (see section 3.2.4) is required. It shall be confirmed, if the existing PCIe driver (TR_1020) can be used. If not, the FTRN driver for PMC shall be derived from the PCIe driver TR_1020.
TR_1050	<p><b>MTCA</b> – A FTRN driver for the specified MTCA form factor (see section 3.2.3) is required.</p> <ul style="list-style-type: none"> <li>It shall be confirmed, if the existing PCIe driver (TR_1020) can be used. If not, the FTRN driver for MTCA shall be derived from the PCIe driver TR_1020.</li> <li>Crate management (IPM) is considered to be part of the MTCA standard. It shall be confirmed, that all requirements by MTCA.0 and the LIBERA<sup>®</sup> platform are fulfilled. If not, they have to be derived or developed.</li> </ul>
TR_1060	<p><b>Performance of host bus to Wishbone bridges</b> – Performance of bridges is tested using the form factor independent Test SoftWare (TSWI). Tests always include the whole stack</p> <ol style="list-style-type: none"> <li>Userland test program</li> <li>EtherBone API</li> <li>Wishbone driver</li> <li>FTRN driver</li> <li>Kernel</li> <li>Bus access</li> <li>Host-bus to Wishbone bridge on the FTRN</li> <li>Wishbone device on the FTRN</li> </ol> <p>Tests are done for issues including reliability, addressable range and latency. The following numbers serve as a guideline.</p> <ul style="list-style-type: none"> <li>Robustness. An error-rate better than <math>10^{-12}</math> must be achieved, see [3], TS_070.</li> <li>Latency. Examples are given in section 3.6.4.</li> <li>Address Range. The full 32 bit address range of the FTRN's Wishbone bus must be accessible without re-configuring the host bus bridge.</li> </ul>
TR_1070	<b>Test SoftWare (TSW)</b> – Test Software for testing the FTRN's hardware shall be developed and implemented on top of the EtherBone API. It shall be possible to perform tests using the

**Document Title:** Detailed Specification Timing Receivers

	standard FTRN gateware and firmware. The contractor only develops the form factor dependent TSW for the in-scope form factors.
<b>TR_1080</b>	<b>Environment</b> – All software must support 32bit and 64bit operating systems. Scientific Linux 6 or later with real-time patches must be supported.

Table 3: FTRN drivers (status February 2014).

### 3.2. FTRN Form Factors

The typical front-end controller at FAIR will be the Scalable Control Unit (SCU) with integrated FTRN functionality as described in [24]. It is expected that about 1000-1500 SCUs will be installed. Moreover, FTRNs of the three form factors VME [26], PCIe [25] and standalone [16] have already been developed at GSI of which about 500 will be employed by different users at FAIR. Hence, the FTRNs delivered by the control system component “Timing Receivers” are only a minor fraction of the total number of FTRNs on the FAIR campus. The FTRNs delivered by the work package “Timing Receivers” shall be compatible to the FTRNs already developed at GSI.

The existing FTRN PEXARIA5 [25] and its mezzanine board PEXARIA5DB [33] is the reference implementation from which all in-scope FTRNs shall be derived. The design of the FTRNs by the contractor should be compatible to the designs of the main contractor in many aspects including parts of the schematics as well as electronic components. As an example, all FTRNs delivered by the contractor and by the main contractor will be based on the Arria V GX FPGA family from Altera.

Two other detailed specifications, “Motion Control FECs” [27] and “Serial Equipment Control FECs” [28] mention timing receiver interfaces for those two types of FECs. If an additional Timing Receiver form factor for those FECs would be required, the development and production of this form factor must be agreed upon in written form between the contractor and the contracting body.

Table 4 gives an overview of the form factors, their FMCs and their digital I/Os. Entries are divided into **In-Scope** and **Out-of-Scope** items.

Number	Description of the Requirement
TR_1200	<b>PCIe</b> – General purpose I/O shall be implemented on a mezzanine board. About 5 bi-directional I/O signals shall be available.
TR_1210	<b>Standalone (Handheld)</b> – At least 6 general purpose I/O shall be implemented on a mezzanine board. In addition a display is required. It shall be investigated, if a user interface via dials/switches is appropriate.
TR_1220	<b>Standalone (Rackmount)</b> – At least 6 general purpose I/O, display, dials, knobs,... shall be implemented on the front panel.
<b>TR_1230</b>	<b>VME64x</b> – This form factor shall be equipped with a mezzanine board. It shall feature the following connection on the front panel: <ul style="list-style-type: none"> <li>Carrier: 2 bi-directional LEMO 00 connectors for LVTTTL signals and one JTAG connector in micro-USB format. One of these connectors shall satisfy TR_1260.</li> <li>Mezzanine: 10 bi-directional LEMO 00 connectors for</li> </ul>

**Document Title:** Detailed Specification Timing Receivers

	<p>LVTTL signals. In addition 2 uni-directional LVDS output and 2 uni-directional LVDS input signals shall be provided via a 5x2 IDC connector.</p> <ul style="list-style-type: none"> <li>• A small display on the front panel.</li> <li>• A USB interface in micro-USB format (should be implemented on the carrier board).</li> </ul>
<b>TR 1240</b>	<b>PMC</b> – This form factor itself is a mezzanine board and I/O should be implemented on the “main” PCB. 4-5 bi-directional LEMO 00 for LVTTL signals shall be implemented on the front panel.
<b>TR 1250</b>	<b>MTCA</b> – As for PMC, the I/O should be implemented on the “main” PCB. The following is required. <ul style="list-style-type: none"> <li>• 5-6 bi-directional LEMO 00 for LVTTL signals on the front panel.</li> <li>• For LIBERA<sup>®</sup> systems only: I/O signals generated by the GMT; specific for clock and trigger distribution on the LIBERA<sup>®</sup> backplanes, see TR_5070.</li> <li>• For LIBERA<sup>®</sup> systems only: Distribution of external high precision ADC sampling clocks, see TR_5080.</li> </ul>
<b>TR 1260</b>	<b>Clock Input</b> – One single-ended I/O on the front panel shall be connected to a clock input of the FPGA. If the I/O is used as output, the connection shall be disabled.

Table 4: Form factors and their I/Os.

**3.2.1. Common Features for all Form Factors**

The design of the FTRNs for the different form factors should avoid diversity as much as possible. The aim is to keep hardware, firmware and gateway as portable as possible between the different form factors. This should be achieved by choosing identical electric components and similar circuit diagrams wherever possible. Compatibility to PEXARIA5 and other form factors by the main contractor is a severe issue for all form factors and the design of the form factors shall be oriented towards the GSI designs, maximizing reusability as much as possible.

All FTRNs shall have the following features independent of their form factor.

<b>Number</b>	<b>Description of the Requirement</b>
<b>TR 2000</b>	<b>Reference design</b> – All FTRNs shall be derived from the PCIe PEXARIA5 [25].
<b>TR 2010</b>	<b>SFP cage</b> – for the connection to the WR network.
<b>TR 2020</b>	<b>WR dircuitry</b> – This can be either implemented directly on the carrier or by a separate add-on board like the WREX1 [31].
<b>TR 2030</b>	<b>Main JTAG connector</b> – A micro-USB connector on the PCB of the carrier board is mandatory. If the space permits, it should be placed on the front panel.
<b>TR 2040</b>	<b>Secondary JTAG connector</b> – Not mandatory, but may be implemented.
<b>TR 2050</b>	<b>USB</b> – A separate micro-USB connector, preferably on the front panel. It is connected to a on-board USB controller.
<b>TR 2060</b>	<b>Connector for logic analysis</b>



**Document Title:** Detailed Specification Timing Receivers

<b>TR_2070</b>	<b>MAC address</b> – A MAC address with Organizationally Unique Identifier (OUI) of the manufacturer is required for each FTRN.
<b>TR_2080</b>	<b>Power</b> – By default, FTRNs are powered via the host-bus (except standalone FTRN).
<b>TR_2090</b>	<b>External power</b> – Capability to be used as standalone receiver. An additional connector for external power using 12V should be included in the PCB layout of all FTRNs. If not, the contractor must deliver a hardware interface which allows powering the FTRN by a low voltage DC source (5V, 12V), possibly via an adapter to the host bus interface; the quantity of hardware interfaces to be delivered per form factor is specified by the main-contractor.
<b>TR_2100</b>	<b>Front panel LEDs</b> – Depending on the available space, 4-8 LEDs shall be implemented on the front panel and connected to the FPGA via a single-ended line, see TR_2140. Examples are <ul style="list-style-type: none"> <li>• Power / WR Link Activity</li> <li>• WR Link Up</li> <li>• WR Lock</li> <li>• PPS</li> <li>• Module o.k.</li> <li>• Bus Access – active, if there is a read or write access</li> <li>• IRQ – active, if an interrupt is requested to the host system</li> </ul>
<b>TR_2110</b>	<b>On-Board LEDs</b> – The LEDs shall be connected to the FPGA via single-ended lines, see TR_2140. <ul style="list-style-type: none"> <li>• 8 user defined LEDs on the carrier</li> <li>• 4-8 user defined LEDs on the mezzanine</li> </ul>
<b>TR_2115</b>	<b>Power LEDs</b> – Power LEDs for all voltages on carrier and mezzanine shall be implemented. They are connected to the relevant voltages.
<b>TR_2120</b>	<b>I/O LEDs</b> – For bi-directional I/O, two LEDs per I/O line shall be implemented on the front panel, see TR_3100. For uni-directional I/O, one LED per I/O line shall be implemented on the front panel, see TR_3570. The LEDs shall be connected to the FPGA via single-ended lines.
<b>TR_2130</b>	<b>LED colour</b> – The colour of LEDs matters and is defined by the main contractor.
<b>TR_2140</b>	<b>LED functionality</b> – Defining and implementing the functionality of LEDs (except power LEDs) is done by the GSI Timing Team.
<b>TR_2150</b>	<b>Push buttons and HEX switch</b> – All form factors shall have two push buttons and one HEX switch on the PCB. They are connected to the FPGA via a single-ended line. Defining and implementing the functionality is done by main contractor (“Reset Button”).
<b>TR_2160</b>	<b>CRC check</b> – The main contractor presently (February 2014) investigates the possibility of gateway CRC verification, see section 3.7.2. If this possibility is confirmed, the FPGA CRC_ERROR pin shall be connected to a dedicated pin defined by the GSI timing team.
<b>TR_2170</b>	<b>Front Panel Text</b> – Front panel text such as engraving must be approved by the main contractor.

**Document Title:** Detailed Specification Timing Receivers

Table 5: Common features for all form factors

The following electronic components shall be used. Table 6 sometimes uses the word “should” instead of “shall. This is only done to avoid a dead-lock in case a component is deprecated or no longer available. The components “shall” be used, as long as they are available.

Number	Description of the Requirement
<b>TR 2500</b>	<b>FPGA</b> – All FTRNs should use 5AGXMA3D4F27I3N FPGAs, but dye revision A must not be used. If required, a model with a different pin-out may be used, but speed grade and temperature range shall not be changed. The connection of signals to FPGA pins should follow the schematics described here [32]. The use of Arria II GX families such as the EP2AGX125EF29C5N is no longer considered.
<b>TR 2510</b>	<b>Flash memory</b> – Flash memory for gateware and firmware. Serial flash N25Q512A13GF840E should be used. The design should use the circuitry described in [32]. Informative: For Arria II GX FPGAs, M25P128-VME6GB should be used.
<b>TR 2520</b>	<b>ROM</b> – The 1-wire ROM DS18B20U+ should be used as temperature sensor with serial number. Each carrier and each mezzanine board shall be equipped with such a temperature sensor. The design should use the circuitry described in [32] [33].
<b>TR 2530</b>	<b>EEPROM</b> – The 1-wire EEPROM DS28EC20P should be used for hardware model, hardware revision number, calibration data and MAC address. Each carrier and each mezzanine shall be equipped with such an EEPROM. The design should use the circuitry described in [32] [33].
<b>TR 2540</b>	<b>USB controller</b> – The USB peripheral controller CY7C68013A-56BAXC should be used This is required as serial interface for the WR console and for providing a Wishbone to USB bridge at the same time. The design should use the circuitry described in [32].
<b>TR 2550</b>	<b>Mezzanine connectors</b> – The connectors QMS-052-05.75-L-D-A should be used to connect mezzanine boards to carrier boards. The design should follow the example described in [32] [33].
<b>TR 2560</b>	<b>Power supply</b> – DC-DC regulators such as LTM4619, LTM4620 or LTM8023 should be used. The exact type depends on the power requirements of the FTRN and the voltages provided by the host system. The design should follow the examples described in [32] [33].
<b>TR 2570</b>	<b>WR circuitry</b> – The WR circuitry shall use components identical to the ones used by the SCU3 [24] or the WREX1 [31] board. The circuitry may be implemented on-board or via a add-on board.

Table 6: Important electronic components.

Commonly used single-ended I/Os are used to trigger external equipment or as input to the ECA or the TLU as described in section 2.1. These signals shall be available via the front panel of all form factors and to be implemented as follows.

Number	Description of the Requirement
--------	--------------------------------



**Document Title:** Detailed Specification Timing Receivers

<b>TR 3000</b>	<b>Reference design</b> – The implementation of single-ended I/Os should be similar to the one on the mezzanine PEXARIA5DB1 [33].
<b>TR 3010</b>	<b>I/O is digital.</b>
<b>TR 3020</b>	<b>Signal type</b> – Signal level shall be LVTTTL single-ended.
<b>TR 3030</b>	<b>Connectors</b> – LEMO 00 type shall be used.
<b>TR 3040</b>	<b>Bi-directional</b> – All I/Os shall be bi-directional.
<b>TR 3050</b>	<b>Connection to FPGA</b> – <ul style="list-style-type: none"> <li>• I/Os are connected to the FPGA, but with buffers in between. If an I/O is implemented on a mezzanine board, the buffer shall be implemented on the mezzanine board too.</li> <li>• For each I/O, the buffer is connected with two pairs of differential signals to two pairs of differential FPGA pins.</li> <li>• Termination should be controlled via a single-ended signal.</li> <li>• Direction should be controlled via a single-ended signal.</li> </ul>
<b>TR 3060</b>	<b>Load</b> – As output, it shall be possible to drive loads of 50 $\Omega$ .
<b>TR 3070</b>	<b>Termination</b> - As input, a configurable termination – on or off – is required.
<b>TR 3080</b>	<b>Bandwidth</b> - I/Os shall support frequencies up to 200 MHz.
<b>TR 3090</b>	<b>Signal quality</b> - I/Os shall support the requirements of the GMT, see TR_325.
<b>TR 3100</b>	<b>LEDs</b> – Two LEDs shall be used per I/O. One indicates the direction, the other one indicates activity. See TR_2120.

Table 7: Properties of single-ended signals.

To use longer cables or to profit from the precision provided by the GMT, single-ended are not sufficient but differential signals must be used. However, this is not required on all form factors, see Table 1. Differential signals are described in the following.

<b>Number</b>	<b>Description of the Requirement</b>
<b>TR 3500</b>	<b>Reference design</b> – The implementation of differential signals should be similar to the one on the mezzanine PEXARIA5DB1 [33].
<b>TR 3510</b>	<b>I/O is digital.</b>
<b>TR 3520</b>	<b>Signal type</b> – Signal level shall be LVDS (pairs)
<b>TR 3530</b>	<b>Connectors</b> – not specified.
<b>TR 3540</b>	<b>Uni-directional</b> – All I/Os should be uni-directional
<b>TR 3550</b>	<b>Connection to FPGA</b> – I/Os are connected to the FPGA, but with buffers in between. If an I/O is implemented on a mezzanine board, the buffer shall be implemented on the mezzanine board too.
<b>TR 3560</b>	<b>Load</b> – As output, it shall be possible driving loads up to 50 Ohms.
<b>TR 3570</b>	<b>LEDs</b> – One LED shall be used per I/O to indicate the activity. See TR_2120.

Table 8: Properties of differential signals.

### 3.2.2. VME64x FTRNs

Number	Description of the Requirement
TR 4000	<b>Reference design</b> – A reference design derived from PEXARIA5 [25] does not exist. On-board circuitry should be derived from PEXARIA5. The host-bus to Wishbone bridge should be derived from the VME board VETAR2 [26].
TR 4010	<b>Size</b> – FTRNs in VME form factor shall be 1 slot x 6U x 160 mm VME64x modules that operate in VME64x crates.
TR 4020	<b>Compatibility</b> – It shall be possible to operate VME64x FTRNs in standard VME crates with modified rails. The FTRN shall implement two additional HEX switches to select its VME address.
TR 4030	<b>Power</b> - All voltages required on the VME carrier board shall be generated from the voltages provided by a <u>standard</u> VME backplane.
TR 4040	<b>Single-ended I/O</b> – The majority of I/O shall be implemented on a mezzanine board. A small number (about 2) shall be implemented on the carrier board.
TR 4050	<b>P1 and P2</b> – P1 / P2 VME bus connectors with 5 rows. All pins should be connected to the FPGA, if possible.
TR 4060	<b>P0</b> – A P0 connector is not required.
TR 4070	<b>Front panel and handles</b> – Front panel with holes for I/O of carrier and mezzanine. Hot plug 64x handles are required.
TR 4080	<b>Minimum access modes</b> – At least A32 (AM: 9) must be implemented. Other VME access modes (A16, A24, BLT, MBLT) shall be implemented, if requested by the GSI Timing Team.
TR 4090	<b>VME SBC</b> – The FTRNs in VME form factor are expected to be compatible with VME Single Board Computer (SBC) endowed with the Tundra® TSI 148 PCI/VME bridge . The VME SBC configures the VME Slave FTRNs in order to set the IRQ level/vector, and accesses the WB bus. The VME interface of the FTRNs interfaces a WB master and WB slave. The WB master is used for accessing the WB bus and the WB slave exposes the MSI interrupts to the VME SBC. Support for MSI implies support for legacy interrupts.
TR 4100	<b>Other VME SBCs</b> – The FTRNs in VME form factor are also expected to be compatible with other standard VME SBCs like RIO4 (CES) or IPV from IOxOS. Here, a FTRN driver or Test SoftWare (TSWD – see TR_1060/1070) is not required.

Table 9: VME64x form factor.

### 3.2.3. MTCA FTRNs

Number	Description of the Requirement
TR 5000	<b>Reference design</b> – The MTCA form factor shall be derived from the PEXARIA5 FTRN as a reference design [25].
TR 5010	<b>Size</b> – FTRN in MTCA form factor shall be developed and implemented, single-width, mid-height, with respect to three use cases.

**Document Title:** Detailed Specification Timing Receivers

<b>TR 5020</b>	<b>Use Case MTCA.0</b> – It shall be possible to configure, monitor a FTRN via PCIe, including IRQ – from any third party userland application such as a FESA class. IRQ generation is the only action towards the userland application that can be triggered by the FTRN via the MTCA.0 backplane. All other actions to the other interfaces of the FTRN must be fully supported. The FTRN must be compatible with MTCA.0, including read/write access as well as IRQ generation via the PCIe bus.
<b>TR 5030</b>	<b>Use case GMT on LIBERA<sup>®</sup> system</b> – On LIBERA <sup>®</sup> systems [15], the FTRN shall be used to distribute clock and trigger signals supported by the GMT to other modules in the same crate. The FTRN can distribute actions generated by the ECA unit, such as digital signal and clock generation, on the LIBERA <sup>®</sup> platform. It can also be used for latching time stamps at the TLU.
<b>TR 5040</b>	<b>Use case external clocks on LIBERA<sup>®</sup> system</b> – Special clock signals received via an external input must be distributed in a LIBERA <sup>®</sup> crate. This feature is not supported by the GMT and requires a dedicated solution. This feature is only required for LIBERA <sup>®</sup> systems, see TR_5080.
<b>TR 5050</b>	<b>No vendor lock-out</b> – It must be possible to use the FTRN in standard MTCA.0 crates of other manufactures present on the European market, such as Elma <sup>®</sup> , Schroff <sup>®</sup> or powerBridge <sup>®</sup> .
<b>TR 5060</b>	<b>LIBERA<sup>®</sup> integration</b> – An interface to the LIBERA <sup>®</sup> system [15] shall be specified, implemented and successfully integrated into the FTRN by the contractor. The FTRN must be compatible to LIBERA <sup>®</sup> Hadron, the LIBERA <sup>®</sup> LLRF as well as LIBERA <sup>®</sup> Single Pass systems. The form factor must support LIBERA <sup>®</sup> specific extensions of MTCA.0. The FTRN supports all trigger and clock ports. From the FTRN point of view, all such signals are TX only. The FTRN shall support the management features required by the LIBERA <sup>®</sup> systems. With respect to the interface to LIBERA <sup>®</sup> , it is the task of the contractor to clarify technical, legal and other issues with I-Tech.
<b>TR 5070</b>	<b>Clock and trigger signals supported by the GMT</b> – On the LIBERA <sup>®</sup> platform, the following shall be implemented. <ul style="list-style-type: none"> <li>• Distribution of clocks via the backplane. Frequencies up to 250 MHz must be supported. It must be possible to correlate all samples of up to 16 ADCs can synchronized within 4 ns.</li> <li>• Distribution of other clock signals via the backplane.</li> <li>• All clock signals distributed via the backplane must provide jitter &lt; +/- 200 ps and a precision better than 1 ns.</li> <li>• Distribute trigger signals via the backplane.</li> </ul>
<b>TR 5080</b>	<b>Distribution of high-precision ADC sampling clocks <u>not supported by the GMT</u></b> – For the case of linac RF, sampling clocks with an accuracy of 10 ps or better have to be distributed to the ADC modules hosted by the LIBERA <sup>®</sup> platform. At present (February 2014) such an accuracy cannot be achieved via the FPGA on a FTRN. The following must be implemented for the

**Document Title:** Detailed Specification Timing Receivers

	<p>LIBERA<sup>®</sup> platform.</p> <ol style="list-style-type: none"> <li>1. The 325.224 MHz clock from the Linac Master Oscillator, which is available via an external RF signal source, is used to derive sampling clocks for ADCs.</li> <li>2. The required precision must at least be 1° of the phase of the 325.224 MHz clock and is in the one-digit picosecond range.</li> <li>3. A low jitter phase aligned ADC sampling clock of 118.263 MHz (Master Oscillator * 4/11) must be derived.</li> <li>4. The 118.263 MHz clock must be distributed as sampling clock to the ADCs.</li> </ol> <p>The solution to this requirement is still an open issue, see 3.7.4.</p>
TR 5090	<b>Single-ended I/O</b> – Digital I/O lines on the front panel can be implemented using a mezzanine module or directly on the carrier board.
TR 5100	<b>Responsibility</b> – It is the responsibility of the contractor to integrate clock and other digital signals generated on the FTRN with the LIBERA <sup>®</sup> platform and to identify any serious incompatibility issue in an early stage of development.

Table 10: MTCA form factor.

**3.2.4. PMC Mezzanine WR Receivers**

While in all form factors mentioned so far the WR standard and timing functionality is implemented on a main board, PCI Mezzanine Card (PMC) receivers are piggy back boards for other controllers (e.g. VME, I/O controllers). The main function of the PMC form factor board is to generate interrupts on the PCI bus at a given time.

Number	Description of the Requirement
TR 6000	<b>Reference design</b> – The implementation of PMC form factor shall be derived from the PEXARIA5 FTRN as a reference design [25].
TR 6010	<b>Single-ended I/O</b> – The PMC FTRN shall implement single-ended digital I/Os. The digital I/Os should be implemented directly on the PMC mezzanine.
TR 6020	<b>Signal type</b> – Signal level shall be LVTTTL.
TR 6030	<b>Host bus</b> – A PCI bridge with 33 MHz clock and 32 bit width is sufficient.
TR 6040	<b>Power</b> – The PMC mezzanine must support the two supply voltages 3.3V and 5V.

Table 11: PMC form factor.

**3.2.5. PCIe FTRNs**

FTRNs in PCIe form factor shall be developed featuring

- PCIe bus connector
- Prepared for a mezzanine I/O card
- Every PCIe FTRN must include a mezzanine module for digital I/O.

A prototype of the PCIe form factor is already available [25].

### 3.2.6. Standalone FTRNs

A host-independent FTRN is called **standalone FTRN**. Such FTRNs shall be provided in two versions: Rack mountable modules and individual units.

- Rack mountable standalone FTRNs shall feature a small display and means for input like switches, buttons, dials or keypad. Display, switches, buttons, I/O... should be implemented using add-on boards. Rack mountable standalone FTRNs shall be developed as 19" crates and the height should not exceed 2U. It must be possible to access the FTRNs remotely. It must be possible to disable remote access via the front panel.
- Handheld FTRNs could be mounted to racks by cap-rails or used as mobile devices. The form factor of such a FTRN should be similar to the EXPLODER2C developed by the department of Experiment-Electronics (CSEE) at GSI [16]. Power supply voltage shall be 12V over external pins. Fans must not be included and passive cooling should be sufficient.

### 3.3. Mezzanine Modules

For each form factor supporting mezzanine cards, one mezzanine card for signal I/O must be delivered.

### 3.4. WR Starter Kit

A WR starter kit is provided by the GSI Timing Team to the contractor to allow a quick setup of a WR demo for development and integration of gateway, firmware and hardware. The WR starter kit shall include at least two WR nodes and optionally one WR switch.

### 3.5. Device Drivers

See section 3.1.10.

### 3.6. Other Issues

This section is mainly informative except section 3.6.4, which needs to be respected.

#### 3.6.1. WR Network on Copper

WR on copper is not developed. If radiation becomes an issue, radiation hard optical fibres shall be used.

### **3.6.2. Phase of BuTiS w2 Clock**

As listed in Table 1, FTRNs shall be able to synthesize a clock signal related to the BuTiS c2 clock. The phase of the synthesized w2 and w0 clock signals must line up with the PPS pulse per default. Phase shifting of the w2 clock signal with respect to the w0 pulse should be possible by a discrete value, if possible. For details on the BuTiS system, see also [20].

### **3.6.3. microSD Slots**

The possibility of loading gateway and firmware from microSD is not required. Gateway and firmware are loaded from the flash memory on the FTRN carrier board.

### **3.6.4. Host Bus Bridging**

Bridge chips between the FPGA and the host bus interface shall not be used. If the contractor prefers the usage of such a chip, this must be approved by the main contractor in written form.

As a guideline, the following test has been done involving the full stack (see TR\_1060). A userland program [35] has been used to read timestamps from the TLU. Within one EtherBone cycle, three 32 bit registers are read and one 32 bit register is written. A few million cycles are performed to determine the average time for the full cycle containing the four operations. The following presents the status as of February 2014.

#### **PCIe**

The present driver does not yet support non-blocking read requests or DMA. On a Dell T3500 (Debian Wheezy) the cycle time is about 8.3  $\mu$ s (120kHz) for the PCIe FTRN [25].

#### **USB**

The cycle time is about 134  $\mu$ s (7.5 kHz) for the FTRNs of form factors PCI, VME and standalone [25][26][16], independent of the host computer. This could be improved, although this would break backward compatibility to form factors prior to PCIe FTRN [25]. Older designs miss a clock signal between USB chip and FPGA.

#### **VME – RIO4**

Using a RIO4 SBC from CES and a VME FTRN [26], a cycle time of about 16.5  $\mu$ s (60kHz) is achieved using A32 access mode. BLT or MBLT access modes will improve the performance for this use case.

#### **VME – MEN A20**

Using an A20 SBC from MEN and a VME FTRN [26], a cycle time of about 32.7  $\mu$ s (30kHz) is achieved using A32 access mode.

### **3.6.5. Bluetooth**

A Bluetooth option for configuring FTRNs is not required.

### **3.6.6. Fine Delay on FTRNs**

Fine delay of 1ns offers a benefit over the 8ns granularity provided the WR clock domain. Fine delay features for input and output are directly implemented in the FPGA, see section 3.1.7. Circuitry using fine delay chips is considered inappropriate by the GSI Timing Team.

## **3.7. Open Issues**

Open issues shall be cleared before design completion between the contractor and the main contractor.

### **3.7.1. Rate Divider**

This is out-of-scope: If the rate of executed actions by the FTRN does not fit the purpose of the application, it must be investigated what would be the best solution.

- Rate divider functionality as proposed in [29].
- Implementation of „lossy“ ECA action channels.
- Polling of Action Queues (instead of IRQs).
- Rate reduction by the ATL.
- Rate divider for digital output channels.

### **3.7.2. Run-time Gateway Check**

This is out-of-scope: Although FTRNs are not intended for use in environment with an increased radiation level, it shall be investigated, if the correctness of the gateway can be verified with techniques like scrubbing or CRC tests.

### **3.7.3. Local and Remote Operation of Standalone FTRNs**

This is out-of-scope: A concept for local and remote operation of rack mountable standalone FTRNs shall be investigated.

### **3.7.4. Distribution of High-Precision Sampling Clocks**

High-precision clocks with an accuracy of better than 10 ps must be distributed on LIBERA® platform for the case of linac-rf. The present ideas shall be investigated.

1. The ADC sampling clock is distributed via the backplane. Then, it has to be routed through the FTRN, as it is mounted in the only slot, which can be used for distribution of clock signals on the backplane. The neighboring GDX slot could be used to implement a front panel with a suitable plug for high-precision clocks. The following options exist

- a) SMA plug as input for the Linac Master Oscillator. The sinusoidal clock signal is routed via a module in the GDX slot. The combination of GDX slot module and FTRN is used for deriving the 118.263 MHz sampling clock and further conversion of the signal into a standard suitable for distribution on the backplane such as LVDS. The module in the GDX slot is connected to the FTRN via a short cable.
  - b) Plug suitable for LVDS signals. Conversion to the 118.263 MHz sampling clock as well as conversion to LVDS is done externally. In this scenario, no PLL or circuitry for conversion to LVDS is required. The combination of GDX slot module and FTRN is only used for routing an external LVDS signal to the backplane.
  - c) The FTRN provides on plug on the front panel suitable for LVDS signals that are directly routed to the backplane. On the one hand, one general purpose I/O on the front panel of the FTRN is “lost”, on the other hand the detour via two additional plugs and the GDX module is not required, which would be beneficial.
2. An external clock distribution module derives the 118.263 MHz sampling clock from the 325.224 MHz clock. The distribution module provides a 118.263 MHz output for each ADC. The present design of the ADCs must be modified; each ADC board must have an additional input for the ADC sampling clock. The clock distribution module may be implemented as a 1U rack mountable crate.

## 4. Quality Assurance, Tests and Acceptance

The system to be built must adhere to the guidelines and recommendations for software developments in the FAIR accelerator control system context, as referenced in the FAIR Common Specification Accelerator Control System F-CS-C-01e. The supplier of the work package must identify the relevant standards and recommendations before start of the development. Details must be fixed as part of the technical design concept in the initialization phase.

The following sections complement and refine the corresponding chapters of the FAIR Common Specification Accelerator Control System F-CS-C-01e.

### 4.1. FTRN Life Cycle

This document does not describe the engineering specifications. Therefore, this section outlines the typical life cycle from which some aspects of the engineering specifications shall be derived.

#### 4.1.1. Installation

Installation is the process of physically installing a FTRN and connecting it to the environment, ready for commissioning.

- Requires a technician from FAIR/GSI to be present at the host system.



- All of the following components must be easily accessible/mountable
  - FTRN carrier in host system
  - Mezzanine board on FTRN carrier
  - Network cables
  - I/O lines (plugs)
  - External power for standalone FTRNs
- Preloaded gateway and firmware is required.

#### **4.1.2. Commissioning**

Commissioning is required to integrate the FTRN into the operational system.

- Requires an engineer from FAIR/GSI to be present at the host system.
- Programming and verification of gateway and firmware via EtherBone.
- Verify functional behaviour
  - WR-PTP
  - EtherBone
  - Host communication
  - Management interface
  - I/O signals (e.g. level, shape, timing)
- Calibration of local action fine delays.

#### **4.1.3. Operation**

Operation of a FTRN covers its intended use in the GSI/FAIR environment. This includes:

- To be done remotely by operators
- Operation of typical accelerator equipment (beam instrumentation, power supplies, kicker ...) and the I/O signals that are exchanged between mezzanine and equipment.
- On-time execution of actions (1ns).
- A single command can result in zero, one or multiple actions
- Network management, status and error reporting
- Add and remove FTRNs on-the-fly to the network
- Recovery from blackout
- Remote reset of single or multiple FTRNs

#### **4.1.4. Maintenance**

Maintenance is required to fix issues, broken hardware or adapt to new requirements. The following points shall be supported during accelerator operation.

- Compilation, synthesis, validation and deployment of new gateway and firmware.
- Replacement of hardware (FTRN carrier and mezzanine board).
- Change, update and verification of gateway and firmware images.

## 4.2. Development Methodology

### 4.2.1. Basic Concept

The contractor shall develop a basic concept to meet the requirements stated in this and all related documents. This concept shall be approved by the GSI Timing Team prior to any contractor development activity.

### 4.2.2. Relationship Between Partners

The contractor shall work in close cooperation with the GSI Timing Team. The link between the GSI Timing Team and the contractor is established via contact persons. The GSI Timing Team has the leading role in this relationship. The contractor may cooperate with White Rabbit developers in other facilities and institutions only with a written mandate by the GSI Timing Team.

### 4.2.3. Design Concept

“Things should be done in the FPGA, not via dedicated electronics!” The underlying design philosophy is to implement as much as possible functionality in the FPGA and as little as possible via dedicated hardware. The connection between the FPGA and I/O, ICs, host bus or other components shall be as direct as possible.

### 4.2.4. Hardware Follows Gateware

The main contractor has already completed the development of gateware according to the milestones in the release chain of the accelerator control system. FTRNs in four different form factors – SCU [24], standalone [16], VME [26] and PCIe [25] – that all support the same gateware and firmware have already been developed. For maximizing reusability and minimizing the development as well as the maintenance effort, the hardware of in-scope FTRNs must follow the existing design at GSI. The contractor must design the hardware to match the existing gateware and firmware via “copy-and-paste” of the schematics of the GSI designs.

The responsibility of the gateware and firmware is with the main contractor except the host bus bridge, that is entirely developed by the contractor.

### 4.2.5. Reporting

The contractor shall report to the GSI Timing Team, delivering written progress reports on a monthly basis specifying

- blocking issues,
- points for decision and
- points for information.

GSI/FAIR will confirm milestone achievements.

#### 4.2.6. Good Practice

The developers of the contractor are expected to share the spirit of open hardware and open software development practices and to work in close contact with the GSI Timing Team. In addition, the contractor is expected to follow the developments and issues in the open hardware White Rabbit project [4], [13].

Important practices to follow include:

- Circuit blocks, gateway, firmware and software shall be as re-usable as possible; specialized solutions shall be well motivated
- Ports for logic analysers are mandatory
- All debug facilities (pins, etc.) used during the development phase shall be kept in the final layout. If they have impact on the functionality, the GSI Timing Team shall be informed.

Introduction of components that:

- produce new dependencies
- are difficult to obtain, discontinued or about to be discontinued
- are particularly expensive
- constitute central elements (e.g. FPGAs, CPUs)

shall be approved by the GSI Timing Team but in general should be avoided wherever possible.

Both GSI/FAIR and the contractor will guarantee the continuous availability (except festivities) of contact persons able to cover all issues and to respond promptly if contacted, making any effort to avoiding delays related to unavailability of key persons.

#### 4.2.7. Infrastructural Changes and Reconciliations

The contractor will monitor infrastructure changes coming from the WR developers' community and evaluate their impact on the development made for GSI/FAIR. In case changes with relevant impact occur, the contractor may recommend:

- to reconcile its hardware / HDL / firmware / software
- to agree on a partial or total roll back of the changes with the WR developers' community
- to start a new HDL / software / firmware branch

In any case, the decision shall be approved by GSI/FAIR on the basis of detailed feedback and recommendations provided by the contractor.

### 4.2.8. Road Map

Within this document, the items of the list below are also referred to as *milestones*. Most relevant points are listed below.

Number	Milestone
TR_11000	Consultation with the GSI Timing Team (contractor).
TR_11010	Comprehensive overview technical concept describing elements, solutions, technologies, components common to all form factors. The concept includes a solution path for the implementation of all form factors and their mezzanine cards. The concept is presented in a brief document of approximately 10 pages (contractor).
TR_11020	Concept approval (GSI/FAIR). The main contractor provides schematics of existing FTRNs like PEXARIA5 to the contractor. Copyright of existing FTRNs is not transferred, but the contractor is allowed to derive its work from the work by GSI, see section 4.2.9.
TR_11030	Engineering specifications for all form factors. The engineering specifications not only refine the detailed specifications, but also update them taking into account the developments in the WR community and especially the developments of the PEXARIA5 and other form factors at GSI. The engineering specifications should be presented in written form including hardware, gateway and software (contractor).
TR_11040	Approval of the engineering specifications (GSI/FAIR). The approved engineering specifications are also included in the test criteria for FAT and SAT.
TR_11050	<p>Design of deliverables (without PCB). This step includes design of hardware, gateway and drivers. The design of hardware includes selection of electric components, schematics for all form factors and mezzanine add-on boards and a market survey to check the availability of the electric components (contractor). The design of the hardware must suit the gateway already provided by the main contractor. The following sequence is repeated until the schematics are approved by the main contractor.</p> <ol style="list-style-type: none"> <li>1. The contractor provides new or updated schematics.</li> <li>2. Gateway and firmware (except the host bus to WB bus bridge) is implemented and synthesized. In that process, the compatibility with existing prototypes developed at GSI ([16], [25], [26] , [24]) will be carefully checked (main contractor).</li> <li>3. The gateway of the host bus to WB bridge is implemented and integrated into a VHDL mockup (contractor). The mockup is provided by the main contractor.</li> <li>4. The gateway of the host bus to WB bridge is integrated into the form factor independent part of the VHDL design</li> </ol>

**Document Title:** Detailed Specification Timing Receivers

	<p>(contractor + main contractor). If issues in the schematics are detected, they are reported. If issues in the host bus to WB bridge are detected, they are reported. Issues in the schematics or host bus bridge must be resolved by the contractor.</p> <ol style="list-style-type: none"> <li>5. All issues (errors, incompatibilities ...) are resolved and the schematics is modified accordingly (contractor).</li> <li>6. Revision of the schematics (main contractor).</li> </ol>
TR_11060	Design of the PCB (contractor) and revision of the PCB (main contractor).
TR_11070	Design approval (main contractor).
TR_11075	<p>The form factor independent part of the gateway and firmware is available by the main contractor (see TR_11050). The form factor Independent Test SoftWare (TSWI) is available by the main contractor. Successful tests are performed on the reference platform including gateway and firmware including the following.</p> <ol style="list-style-type: none"> <li>1. USB bridge via EtherBone</li> <li>2. Connection from timing network via EtherBone</li> <li>3. 1-wire temperature sensor</li> <li>4. 1-wire EEPROM for configuration data</li> <li>5. Flash for gateway and firmware</li> <li>6. ...</li> </ol>
TR_11080	<p>Production of prototypes. This includes the following steps.</p> <ol style="list-style-type: none"> <li>1. At least five modules per form factor are delivered.</li> <li>2. Manufacturing of hardware and verification of JTAG chain (contractor).</li> <li>3. Testing of prototype with gateway and firmware via JTAG, WR network interface, USB WR debug console, USB to WB master bus interface. The host bus to WB bridge is not tested (main contractor).</li> <li>4. Testing of prototype with gateway and firmware including the host bus to WB bridge (contractor).</li> <li>5. The form factor Dependent TestSoftware (TSWD) must be delivered by the contractor for the in-scope form factors.</li> </ol>
TR_11090	FAT, approval for series production (GSI/FAIR) with the help of the contractor. FAT is mainly performed on the GSI/FAIR site. Best effort by the main contractor and the contractor is required to minimize the time for FAT. The time for testing the final prototype shall not exceed six months.
TR_11095	Break-down of form factors within the total number of FTRNs to be delivered (GSI/FAIR).

**Document Title:** Detailed Specification Timing Receivers

TR_11100	Series production (contractor). Automated Optical Inspection (AOI) is foreseen at the mass producer.
TR_11110	SAT A (contractor and GSI/FAIR)
TR_11120	SAT B (contractor and GSI/FAIR)

Table 12: Road Map

**4.2.9. Intellectual Property and Licensing Issues**

All hardware, gateware, firmware and software development shall be implemented and published in full accordance with the CERN Open Hardware License (CERN OHL) scheme [5] and GPL [9] respectively.

The contractor shall not raise any intellectual property claims on Timing Receivers, on hardware and software components and development tools implemented within this work package.

The contractor shall investigate, if the CERN OHL causes issues for the case of the MTCA form factor.

**4.3. Quality Assurance System of the Supplier**

Basis for the acceptance test model are the FTRN detailed specification, the *life cycle* listed in this document and the approved engineering specifications.

It is the task of the contractor to set up a test stand where many FTRNs can be tested in a short time.

It is in the responsibility of the GSI Timing Team to provide a test-bed system on the GSI/FAIR site. The test-bed represents a complete timing system, containing a timing master and multiple layers of switches to which FTRNs of various form factors are connected. The test-bed allows testing

- scenarios like the ones described in the *life cycle*,
- interoperability and the interaction between different form factors,
- the fulfillment of timing constraints,
- simultaneous generation of gate signals on Timing Receivers and
- fulfillment of the functional requirements listed in Table 1.
- fulfillment of the non-functional requirements listed in Table 2.

All regulations and procedures defined in the FAIR Common Specification Accelerator Control System [2] fully apply.

**4.4. FAT**

The Factory Acceptance Test (FAT) constitutes the basis for the decision to start series production.

Since FAT requires a WR network, and therefore a timing master and several switches which are not part of this work package, the test bed will be

**Document Title:** Detailed Specification Timing Receivers

implemented at GSI/FAIR. The contractor will deliver at least five FTRNs for each form factor. The contractor shall also deliver a form factor specific test facility, which allows mass testing of FTRNs for each form factor.

Items verified by FAIR include the following.

Number	FAT Item
TR_12000	The form factor Dependent Test SoftWare (TSWD) for the in-scope form factors is verified.
TR_12010	FTRN prototypes work as requested.
TR_12020	FTRNs pass the tests by the approved TSWD and the form factor independent TestSoftWare (TSWI).
TR_12030	Mandatory: FTRNs shall pass burn-in-tests of one week under environmental conditions, while undergoing tests by the TSWD and TSWI.
TR_12040	Optional: FTRNs shall pass burn-in-tests of one month under environmental conditions, while undergoing tests by the TSWD and TSWI.
TR_12050	FTRNs shall pass a test in a climatic chamber as described in TR_10120.
TR_12060	FTRN integration into the WR timing network is successful.
TR_12070	FTRNs interoperate with a prototype of the FAIR accelerator control system, including: <ol style="list-style-type: none"> <li>1. clock master (in charge of providing exact timing)</li> <li>2. data master (in charge of defining real time operation constraints and distributing commands within these)</li> <li>3. management master (in charge of network supervision and slow control)</li> </ol>
TR_12080	The form factor specific test facilities are verified.
TR_12090	Verification of production quality via Automated Optical Inspection (AOI).
TR_12100	The main contractor may perform other tests too.
TR_12110	Optional: FTRNs shall pass a vibration / shaking test.

Table 13: FAT Items.

## 4.5. SAT

The Site Acceptance Test (SAT) will be split into two parts.

SAT A. All FTRNs shall be individually tested for functionality and compliance using the form factor specific test facilities. A subset of FTRNs must be tested using the test bed described in section 4.3. The FTRNs shall pass tests including the ones described in section 4.4 (FAT).

**Document Title:** Detailed Specification Timing Receivers

SAT B. The integration of the FTRNs into the overall FAIR accelerator control system is done. Tests without and with beam are performed.

All components must run inside the FAIR accelerator environment and must be capable to carry out their foreseen functions in a real world scenario.

The procedures as described in the Common Specifications for the Accelerator Control System [2] fully apply.

## 5. Documentation

The delivery of documentation according to the list of documents in the FAIR Common Specification Accelerator Control System [2] fully applies.

The full set of development works, including CAD files, all source codes, tools and development environment, has to be published under the CERN open hardware license scheme [5].

The documentation shall enable technicians and developers to perform the following operations with WR Timing Receivers:

- perform basic function tests
- installation
- gateway and firmware upgrade
- setup of a software and firmware development environment, typically in C/C++
- setup of a gateway development environment, typically in VHDL
- modification of gateway, firmware and software

## 6. Warranty

The conditions and warranty period defined in the contract.

## 7. Scope of Delivery

The contractor shall deliver the following goods:

Number	Description of the Deliverable
TR 10000	The following form factors are in-scope: VME64x, MTCA, PMC.
TR 10010	Technical concept for all in-scope form factors of the FTRNs.
TR 10020	Engineering specifications of the in-scope FTRNs, including specification of hardware, host bus to WB bus bridge, interface to other resources provided by the host system (e.g. MTCA) and other software and a project plan.
TR 10030	Fully disclosed development work for all in-scope Timing Receivers as well as mezzanine cards. The development work must include all documentation of the development including



**Document Title:** Detailed Specification Timing Receivers

	source codes and schematics. Furthermore, all documents of the production process like all files in the native format of the layout tool, layout files in a general open CAD format, PCB gerber files, bill of materials, list of materials, assembly information... is required. Appropriate CAD data including files in "rpt" format are required to allow AOI by the main contractor.
<b>TR 10040</b>	<p>Development and implementation of host bus to WB master and slave bridge. The development shall be done by deriving from the bridges already existing at GSI. Host bus to WB master and slave bus bridges include</p> <ol style="list-style-type: none"> <li>1. HDL code</li> <li>2. connection to the WB crossbar of the gateway provided by the main contractor,</li> <li>3. FTRN drivers for 32Bit and 64Bit Linux operating systems. Scientific Linux 6 and later including real-time patches must be supported. For more details of the software stack see section 2.2.3 (software) and section 3.1.10 (drivers).</li> <li>4. Relevant hardware and gateway for all other signals and protocols provided by the host system (e.g. MTCA).</li> </ol>
<b>TR 10050</b>	<p>Form factor Dependent Test SoftWare (TSWD) for all in-scope form factors based on the EtherBone library for all in-scope form factors. Tests shall also allow the measurement of the latency, robustness and address range (see TR_1060). Test shall verify the functionality of both gateway and hardware including the following components.</p> <ol style="list-style-type: none"> <li>1. host bus bridge via EtherBone</li> <li>2. All LEDs</li> <li>3. All single-ended I/O (max frequency, direction, termination...)</li> <li>4. All differential I/O if present (max frequency, direction, termination...)</li> <li>5. All signals to the backplane on a LIBERA© platform.</li> <li>6. Compatibility of MTCA with standard MTCA.0 crates.</li> <li>7. Relevant crate management features in LIBERA© and standard MTCA.0 crates.</li> <li>8. ...</li> </ol>
<b>TR 10060</b>	Tools for testing the host bus interface for all in-scope form factors, including tools to synthesize gateway, driver and software.
<b>TR 10070</b>	HDL code and tools sufficient to synthesize gateway for testing the hardware of all in-scope mezzanine cards.
<b>TR 10080</b>	Form factor specific test facilities, allowing automated tests of

**Document Title:** Detailed Specification Timing Receivers

	<p>FTRNs in a short time.</p> <ol style="list-style-type: none"> <li>1. For VME64x, only software is required</li> <li>2. For MTCA, software and a host system including backplane, crate controller and CPU is required. The host system must include a backplane suitable for testing distribution of clock and trigger signals required for LIBERA<sup>®</sup> systems.</li> <li>3. For PMC, only software is required. The integration of a PMC module into a VME CPU with a Tundra TSI 148 chip (such as MEN0A20) must be demonstrated by the contractor.</li> </ol>
<b>TR 10090</b>	300 FTRNs. Each form factor supporting mezzanine modules (e.g. VME) must be equipped with mezzanine module. If the WR circuitry is implemented with a WREX mezzanine [31], it must be equipped with such a mezzanine.
<b>TR 10100</b>	<p>All hardware modules - separately for carrier and mezzanines –</p> <ol style="list-style-type: none"> <li>1. must be produced according to commonly accepted industry standards, similar to the directives and guidelines by the IPC [34]. This includes placing of components and soldering is done by machinery - not by hand – to allow AOI.</li> <li>2. It must be delivered as a product, following the relevant guidelines like low voltage directive, CE, declaration of conformity to CE.</li> <li>3. must have a sticker with its serial number. The serial number is the one from the 1-wire temperature sensor.</li> <li>4. must pass a burn-in-test of at least 24 hours.</li> <li>5. test protocol describing the tests, the result of the tests performed by the Test SoftWare (TSWD + TSWI) and the result of the burn-in-test. The result must be documented for each piece of hardware individually identified by its serial number.</li> </ol>
<b>TR 10110</b>	A “Mean Time Between Failure” (MTBF) calculation for each FTRN type shall be performed by the contractor.
<b>TR 10120</b>	A fraction of 10 % of all hardware modules must pass a 24 hour test for artificial aging in a temperature chamber at 60-70 degree centigrade. While being baked, they must continuously undergo and pass tests by the Test SoftWare (TSWD + TSWI). A test protocol is required for each involved module, identified by its serial number.
<b>TR 10130</b>	Documentation for all deliverables.

Table 14: Deliverables

## 8. Acronyms and Definitions

The following acronyms and definitions are used in this document. For a full list of acronyms, please refer to the list of abbreviations for controls.

- **Action:** This is an activity executed by a FTRN. Activities are scheduled locally by the ECA and are performed by receiving components attached to the ECA. Examples of actions are raising an IRQ line or sending a digital output pulse via a LEMO 00 connector.
- **Condition:** Conditions define how received commands are mapped to actions by the ECA.
- **Command:** An input to the ECA. Typically, commands are received from the data master via the timing network.
- **Firmware:** compiled software code. Firmware is executed by a (Soft-)CPU. For WR designs, a Soft-CPU is frequently embedded in the gateway.
- **Gateway:** synthesized HDL code. Gateway is configuration for programmable logic and loaded into the FPGA of a timing node, such as a FTRN. In many cases, gateway includes a Soft-CPU such as the LM32 [18].
- **GSI Timing Team:** The Timing Group within the Controls Department at the GSI Helmholtzzentrum für Schwerionenforschung GmbH (GSI), Darmstadt, Germany.
- **Form Factor Specific Test Facility:** Delivered by the contractor, allows mass testing of FTRNs for each form factor
- **Test-Bed:** provided by the GSI Timing Team. The test-bed system is a complete timing system including a timing master, a timing network and Timing Receivers.
- **Clock Master:** in charge of providing exact timing
- **Data Master:** in charge of defining real time operation constraints and distributing commands within these
- **Management Master:** in charge of network supervision and slow control
- **Timing Master:** Includes the clock master, data master and management master
- **ACCNET** - ACCelerator NETwork
- **AOI** - Automated Optical Inspection
- **API** - Application Programming Interface
- **ATL** - API Timing Library
- **CPU** - Central Processing Unit
- **ECA unit** - Event Condition Action unit
- **FAT** - Factory Acceptance Test
- **FEC** - Front-End Controller

- FECo - Forward Error Correction
- FESA - Front-End Software Architecture
- FMC - FPGA Mezzanine Card
- FTRN - FAIR Timing Receiver Node
- GMT - General Machine Timing system
- GSI - GSI Helmholtzzentrum für Schwerionenforschung GmbH
- HDL - Hardware Description Language
- HID - Human Interface Device
- MAC - Media Access Control address
- MTBF - Mean Time Between Failure
- OS - Operating System
- OUI - Organizationally Unique Identifier, part of a MAC address
- PCB - Printed Circuit Board
- PMC - PCI Mezzanine Card
- PTP - Precision Time Protocol
- PWM - Pulse Width Modulation
- SAT - Site Acceptance Test
- SCU - Scalable Control Unit
- SFP - Small Form-factor Pluggable
- SPEC - Simple PCI express card
- SVEC - Simple VME FMC Carrier
- TAI - Temps Atomique International
- TLU - Timestamp Latch Unit
- TSWD - Test SoftWare, form factor Dependent
- TSWI - Test SoftWare, form factor Independent
- USB - Universal Serial Bus
- VHDL - Very High Speed Integrated Hardware Description Language
- WB - WishBone bus
- WR - White Rabbit
- WR-PTP - White Rabbit PTP

## I. Attached Documents

List of abbreviations for controls (Abbreviations\_Controls.pdf).

## II. Related Documentation

- [1] F-GS-B-01e, FAIR General Specifications
- [2] F-CS-C-01e, FAIR Common Specification “Accelerator Control System”
- [3] F-DS-C-05e, FAIR Detailed Specification “General Machine Timing System”
- [4] WR-PTP Repository: White Rabbit core collection – <http://www.ohwr.org/projects/wr-cores>
- [5] CERN Open Hardware License – <http://www.ohwr.org/projects/cernohl>
- [6] F-DS-C-01e, FAIR Detailed Specification “FEC software framework (FESA)”
- [7] The architecture of an active database management system, Dennis McCarthy and Umeshwar Dayal, SIGMOD ‘89 Proceedings of the 1989 ACM SIGMOD international conference on Management of data
- [8] F-DS-C-03e, FAIR Detailed Specification “Settings management system”
- [9] GNU General Public License – <http://www.gnu.org/copyleft/gpl.html>
- [10] WR Specifications – <http://www.ohwr.org/documents/21>. Draft for comments: <http://www.ohwr.org/attachments/306/WhiteRabbitSpec.pdf>.
- [11] WR Node Functional Specifications – <http://www.ohwr.org/>
- [12] WR PCIe Node SPEC, repository – <http://www.ohwr.org/projects/spec>
- [13] Open Hardware Repository – <http://www.ohwr.org>
- [14] Etherbone core for WR, repository – <http://www.ohwr.org/projects/etherbone-core>
- [15] The LIBERA<sup>®</sup> system from i-Tech – <http://www.i-tech.si/accelerators-instrumentation>
- [16] EXPLoder2C, [http://www.gsi.de/work/organisation/wissenschaftlich technologische abteilungen/experiment elektronik/digitalelektronik/digitalelektronik/module /font end module/exploder/exploder2c.htm](http://www.gsi.de/work/organisation/wissenschaftlich_technologische_abteilungen/experiment_elektronik/digitalelektronik/digitalelektronik/module_font_end_module/exploder/exploder2c.htm).
- [17] M.Sc. Thesis of T. Wlostowski, Warsaw University of Technology, 2010/2011 – <http://www.ohwr.org/documents/80>
- [18] LatticeMico32 soft CPU for Xilinx platforms – <http://www.ohwr.org/projects/lm32>

**Document Title:** Detailed Specification Timing Receivers

- [19] FAIR Technical Note, No 20100510, "Integration and Linking between FAIR Timing System and BuTiS".
- [20] BuTiS – Bunchphase Timing System  
[http://www-bd.gsi.de/dokuwiki/lib/exe/fetch.php?media=meetings:moritz\\_butis\\_tech\\_nikforum\\_2010\\_10\\_28.pdf](http://www-bd.gsi.de/dokuwiki/lib/exe/fetch.php?media=meetings:moritz_butis_tech_nikforum_2010_10_28.pdf)
- [21] F-DG-C-03e "Software Architecture Guideline"
- [22] Wishbone B4 Documentation  
[http://cdn.opencores.org/downloads/wbspec\\_b4.pdf](http://cdn.opencores.org/downloads/wbspec_b4.pdf)
- [23] F-TG-ET-01e, FAIR Technical Guideline "Electrical Design Rules and Regulations"
- [24] F-DS-C-02e, FAIR Detailed Specification "Equipment Interface and Control".
- [25] PEXARIA5.  
[https://www.gsi.de/work/fairgsi/common\\_systems/csee\\_electronics/digitalelektronik/digitalelektronik/module/pci\\_pci\\_e/pexaria/pexaria5.htm](https://www.gsi.de/work/fairgsi/common_systems/csee_electronics/digitalelektronik/digitalelektronik/module/pci_pci_e/pexaria/pexaria5.htm)
- [26] VETAR2.  
[https://www.gsi.de/work/fairgsi/common\\_systems/csee\\_electronics/digitalelektronik/digitalelektronik/module/vme/vetar/vetar2.htm](https://www.gsi.de/work/fairgsi/common_systems/csee_electronics/digitalelektronik/digitalelektronik/module/vme/vetar/vetar2.htm)
- [27] F-DS-C-16e, FAIR Detailed Specification "Motion control FECs"
- [28] F-DS-C-17e, FAIR Detailed Specification "Serial equipment control FECs"
- [29] Technical Note: LOBI FAIR Timing Receiver Node Requirements
- [30] WR VME Node SVEC, repository – <http://www.ohwr.org/projects/svec>
- [31] WREX1 White Rabbit add-on board – see  
<http://indico.cern.ch/event/238322/contribution/7/material/slides/0.pdf>
- [32] Schematics of PEXARIA5.
- [33] Schematics of PEXARIA5DB.
- [34] <http://www.ipc.org>.
- [35] [https://www-acc.gsi.de/svn/bel/timing/trunk/development/eb\\_demo/eb\\_tlu\\_demo.c](https://www-acc.gsi.de/svn/bel/timing/trunk/development/eb_demo/eb_tlu_demo.c).

### III. Document Information

#### III.1. Document History

Version	Date	Description	Author	Review / Approval
0.1	27. Sep. 2011	Draft version	Sergio Mauro	1 <sup>st</sup> draft
1.0	07. Oct. 2011	Draft	S. Mauro	draft status
1.7c	27. Oct. 2011	Working draft	GSI Timing Team	Waiting for comments
1.8a	28. Oct. 2011	Working draft	GSI Timing Team	
1.8d	28. Oct. 2011	Pre release	GSI Timing Team	Waiting for clearance
1.8e	31. Oct. 2011	Checked by J. Fitzek	CCT	Implement Change
1.8f	31. Oct. 2011	Last changes done	GSI Timing Team	Waiting for clearance R. Bär
2.0	31. Oct. 2011	Final version	Kreider, CCT	CCT
2.1	17. Nov. 2011	Renaming of referenced guidelines	CCT	
3.0	02. Aug. 2012	Incorporated FAIR review comments	Beck, CCT	SD, Timing Team
3.1beta	05. Sep. 2012	Merged with F-TN-C-001 and updated	Beck	CCT
3.1	13. Sep. 2012	remarks by CCT and Ralph Bär	Beck	CCT
3.1.1	14 Nov 2013	created patch	Beck	
3.1.2	10 Feb 2014	continue ...	Beck	
3.1.2	11 Feb 2014	Included comments by CL und LOBI, update to status quo	Beck	
3.2.C	12 Feb 2014	PEXARIA5 now reference design	Beck	R. Bär
3.2.D0	13 Feb 2014	check consistency, include comments by T. Hoffmann	Beck	LOBI
3.2.D	14 Feb 2014	Include comments by K. Lang “final” check by Beck	Beck	LOBI + CCT
3.2.E	17 Feb 2014	Include comments by J. Fitzek	Beck	
3.2.E2	18 Feb 2014	Include comments by C. Prados	Beck	Timing Team

**Document Title:** Detailed Specification Timing Receivers

3.2.E3	21 Feb 2014	Include comments from LOBI	Beck	CSEE
3.2.RC1	26 Feb 2014	Comments by Nik, CL	Beck	CL
3.2 RC2	28 Feb 2014	adding latency of host bus bridge	Beck	
3.2 RC3	02 Mar 2014	The bag is closed! Final Reading	Beck	Beck
3.2	02 Mar 2014	Version 3.2 now final	Beck	