

Synchronization and Sequencing
of Data Acquisition and Control Electronics
at the European X-Ray Free Electron Laser

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg

zur Erlangung des akademischen Grades

Doktor Ingenieur

genehmigte Dissertation

von
Patrick Geßler

aus
Hamburg

2015

Prof. Dr-Ing. Dr. h.c. Klaus Schünemann
Prof. Dr. F. Mayer-Lindenberg
Prof. Dr.-Ing. Wolfgang Krautschneider

20. November 2015

Abstract

The 3.5km long European X-Ray Free Electron Laser, currently under construction in northern Germany, will deliver bursts of up to 2700 short x-ray pulses every 100ms, providing wavelengths between 0.05 and 6 nm, and a repetition rate of 4.5MHz to several experiment stations. It allows in-depth research in various scientific fields.

In order to set-up the beam, position samples and capture the measured variables, information from the accelerator, diagnostic devices and detectors have to be digitized, converted, processed, transferred, concentrated, distributed, reorganized, controlled and saved. All these steps have to be accurately synchronized and sequenced relative to the actual electron bunch or photon pulse in order to guarantee correct data acquisition timings and unique identification of each bunch passing the beamlines.

This document provides a complete description of the planning, design, realization and evaluation of the European XFEL Timing System, which implements the synchronization and sequencing of the data acquisition and control electronics for the European X-Ray Free-Electron Laser Facility.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Deutsches Elektronen-Synchrotron (DESY) | 1 |
| 1.3 | European X-Ray Free-Electron Laser Facility | 3 |
| 1.4 | Scope of work | 4 |
| 1.5 | Requirements | 4 |
| 1.5.1 | Required functions | 4 |
| 1.5.2 | Bunch structure | 4 |
| 1.5.3 | Frequencies | 5 |
| 1.5.4 | Stability requirements | 5 |
| 1.5.5 | Hardware platform | 6 |
| 1.6 | Structure of this document | 6 |
| 2 | Definition of terms | 7 |
| 2.1 | Timing System | 7 |
| 2.2 | Clock | 8 |
| 2.3 | Trigger | 8 |
| 2.4 | Gate | 8 |
| 2.5 | Syntonization | 8 |
| 2.6 | Synchronization | 9 |
| 2.7 | Deterministic data transmission | 10 |
| 2.8 | Drift | 10 |
| 2.9 | Noise | 10 |
| 2.9.1 | Thermal Noise | 11 |
| 2.9.2 | Shot Noise | 11 |
| 2.9.3 | Flicker Noise and $1/f^n$ -Noise | 11 |
| 2.9.4 | Burst Noise | 11 |
| 2.9.5 | Amplitude, Frequency and Phase Noise | 12 |
| 2.10 | Jitter | 12 |
| 2.11 | Time Interval Error | 12 |
| 2.12 | Dispersion | 13 |
| 2.13 | Inter Symbol Interference (ISI) | 13 |
| 2.14 | Resolution | 13 |
| 3 | Timing system technologies and usage at Light Sources | 15 |
| 3.1 | Distribution of Coordinated Universal Time (UTC) | 15 |
| 3.1.1 | Network Time Protocol (NTP) | 15 |
| 3.1.2 | IEEE1588, SyncE and White Rabbit | 17 |
| 3.2 | Bunch clock distribution system | 19 |
| 3.2.1 | European Synchrotron Radiation Facility (ESRF) | 19 |

| | | |
|----------|---|-----------|
| 3.2.2 | PETRA III | 20 |
| 3.3 | Clock and event distribution systems | 20 |
| 3.3.1 | FLASH Event System | 21 |
| 3.3.2 | Micro-Research Finland (MRF) | 21 |
| 4 | System design | 23 |
| 4.1 | Basic concept | 23 |
| 4.2 | Implementation of the basic concept | 24 |
| 4.3 | Influences on phase stability | 25 |
| 4.3.1 | Temperature induced drift | 25 |
| 4.3.2 | Jitter | 26 |
| 4.3.3 | Electro Magnetic Interference (EMI) | 26 |
| 4.3.4 | Conversion of amplitude into phase variations | 26 |
| 4.4 | Influences on accuracy | 27 |
| 4.4.1 | Resolution | 27 |
| 4.4.2 | Reproducible phase relations | 27 |
| 4.5 | Consequences for the system design | 29 |
| 4.5.1 | Synchronization to power line frequency | 29 |
| 4.5.2 | Optical transmission line and interfaces | 31 |
| 4.5.3 | Drift compensation scheme | 31 |
| 4.5.4 | Link length determination | 34 |
| 4.5.5 | Dedicated low-jitter clock section | 35 |
| 4.5.6 | Differential Signalling Standard | 35 |
| 4.5.7 | Receiver-side clock and trigger synchronization | 35 |
| 4.6 | Selection of components | 37 |
| 4.6.1 | Clock and Data Recovery (CDR) | 37 |
| 4.6.2 | Field Programmable Gate Array (FPGA) | 38 |
| 4.6.3 | Phase detector | 38 |
| 4.6.4 | Adjustable delay | 39 |
| 4.6.5 | Clock dividers and output buffers | 41 |
| 4.6.6 | Switches and buffers | 41 |
| 4.6.7 | Phase Locked Loops (PLLs) | 41 |
| 4.6.8 | Optical transceivers | 41 |
| 4.6.9 | Fibers | 42 |
| 5 | Evaluation | 43 |
| 5.1 | Investigation of critical components | 43 |
| 5.1.1 | Delay, CDR and clock dividers | 43 |
| 5.1.2 | Phase detector | 45 |
| 5.2 | Evaluation board and measurement setup | 47 |
| 5.3 | Implementation of the drift compensation scheme | 49 |
| 5.3.1 | Phase stabilization with fine delays | 49 |
| 5.3.2 | Control of coarse delays | 50 |
| 5.3.3 | Adjustments of non-symmetric delay elements | 50 |
| 5.4 | Measurement results | 51 |
| 6 | MicroTCA hardware platform | 53 |
| 6.1 | Time of transition | 53 |
| 6.2 | Introduction to ATCA and MicroTCA | 54 |
| 6.3 | Features of the MicroTCA standard | 56 |
| 6.3.1 | Passive Backplane | 56 |

| | | |
|-----------|--|-----------|
| 6.3.2 | Management of modules and hot-plugging | 58 |
| 6.3.3 | Centralized switching and distribution | 59 |
| 6.3.4 | Point-to-point connections | 59 |
| 6.3.5 | Redundancy | 59 |
| 6.3.6 | Remote access | 59 |
| 6.4 | xTCA for Physics working group and MTCA.4 standard | 59 |
| 6.4.1 | Distribution of slow clocks, triggers, interlocks and deterministic data | 59 |
| 6.4.2 | Double Size AMCs and Micro Rear Transition Modules | 61 |
| 6.4.3 | High-Speed serial point-to-point interconnects | 62 |
| 6.4.4 | Improvements on noise and jitter | 62 |
| 7 | First generation Timing System board | 63 |
| 7.1 | Hardware | 63 |
| 7.2 | Reference clock and triggers | 65 |
| 7.3 | Timing data stream transmission and drift compensation | 65 |
| 7.4 | Timing receiver | 67 |
| 7.5 | Clock distribution | 67 |
| 7.6 | Triggers, bunch clocks and data outputs | 67 |
| 7.7 | Further options | 68 |
| 7.8 | Experience and consequences for next generation | 68 |
| 7.8.1 | Form factor and module separation | 68 |
| 7.8.2 | Clock buffers | 68 |
| 7.8.3 | Connectors | 69 |
| 7.8.4 | Filtering of power supply | 69 |
| 8 | Second generation Timing System board | 71 |
| 8.1 | Hardware | 71 |
| 8.2 | Reference clock and triggers | 71 |
| 8.3 | Timing receiver | 74 |
| 8.4 | Clock distribution | 75 |
| 8.5 | Trigger, bunch clock and data outputs | 75 |
| 8.6 | Timing data stream transmission and drift compensation | 76 |
| 8.7 | Extending the functionalities | 77 |
| 8.8 | Complete drift compensation measurement | 77 |
| 9 | Firmware | 81 |
| 9.1 | Configuration of integrated circuits on the module | 81 |
| 9.2 | Generation of the timing data stream | 82 |
| 9.3 | Control loop for drift compensation | 82 |
| 9.4 | Synchronization to the timing data stream | 84 |
| 9.5 | Decoding of the timing data stream | 85 |
| 9.6 | Trigger generation | 86 |
| 9.7 | Transmission of deterministic data to local receiving systems | 86 |
| 9.8 | Communication with in-crate CPU via PCIe | 87 |
| 9.9 | Synchronization between Modules | 87 |
| 9.10 | Remote Firmware Upgrade | 88 |
| 10 | Software | 91 |
| 10.1 | CPU technology, operating system and interfacing | 91 |
| 10.2 | Driver | 91 |
| 10.3 | Control Systems and device implementation | 92 |

| | |
|---|------------|
| 10.4 Graphical User Interface (GUI) | 93 |
| 11 Interfacing to consumers | 97 |
| 11.1 Interfaces within the MicroTCA crate | 97 |
| 11.1.1 Low-jitter Clock Distribution | 97 |
| 11.1.2 Signal Distribution on M-LVDS Bus Lines | 99 |
| 11.1.3 Interrupt and Information on PCIe | 101 |
| 11.2 RJ45 connectors for external consumers | 101 |
| 11.3 RJ45 connector for external trigger inputs | 103 |
| 11.4 Future options | 104 |
| 12 Interfacing with other Timing Systems | 105 |
| 12.1 FLASH Timing System | 105 |
| 12.2 Micro Research Finland | 106 |
| 12.3 White Rabbit | 106 |
| 12.4 Bunch clock distribution systems | 107 |
| 13 Conclusion | 109 |
| 14 Acknowledgments | 111 |
| List of Abbreviations | 113 |
| List of Symbols | 115 |
| References | 117 |

Chapter 1

Introduction

1.1 Motivation

Time and order affect almost everybody's life. Whether it is about a point in time, when to get up in the morning and when to meet friends or a customer or how long it takes to walk, drive or fly to a certain destination - time is of importance. Ordering comes into play, as soon as there is more than just one task to do - what should be done first, is there a dependency between different tasks? Usually order becomes even more important, the more tasks have to be done in a limited time, or - the other way around - the shorter the time gets. Even people who think of themselves as not following common conventions of time and order, are still bound to cycles of nature and civilization like the day-and-night cycles, the seasons of the year or opening hours of shops, respectively. They have to synchronize to these cycles - at least to some extent.

The very same is true for complex machines like the European X-Ray Free Electron Laser (XFEL), which is currently under construction in northern Germany. This 3.4km long accelerator facility will provide X-Ray pulses of unprecedented brightness for studies in physics, chemistry, life sciences and materials research. The pulses generated and used for experiments have a duration of less than 100 Femto-seconds and will be repeated up to 2700 times within a time window of 600 Micro-seconds. In order to generate these series of short pulses thousands of components (including lasers, signal generators, diagnostic systems, detectors, motors etc.) have to be synchronized and the order (sequencing) within this short time frame has to be defined in a strict way. The design, development, implementation and verification of a technical solution, which accomplishes the tasks of synchronization and sequencing for the European XFEL is the topic of this work. The realization has been named XFEL Timing System.

1.2 Deutsches Elektronen-Synchrotron (DESY)

The Deutsche Elektronen-Synchrotron (DESY) is a research center within the Helmholtz association and is located in Hamburg, Germany. The focus of the work and the main research fields have changed during the more than 50 years since the DESY had been founded in 1959. The initial interest concentrated on high energy physics, where the inner structure of particles were investigated. Particle collider accelerators were developed in order to produce collisions of different particles, which revealed most of the inner components and effects.

Over the time almost all experiments have been done with the available energy the accelerators were able to provide to the particles. This was the reason, why the biggest accelerator, HERA, reached its planned end in 2007 (although the processing and evaluation of the high amount of detected data is still ongoing).

Besides particle collisions, accelerators are also a source of photons which are radiated when charged particles are accelerated. This always happens in circular accelerators, as the particles have to be kept on a circular path, which requires acceleration towards the center, even if the time for one revolution is constant. The photon production could be even highly intensified by using a special structure called wiggler (see Figure 1.1). In this structure the group of particles (2) - called bunch (in our case consisting of electrons) - will pass a certain number of magnetic fields with alternating polarizations, which are created by permanent and strong magnets (1). Each field will impose a Lorenz-force to the electron bunch, which will slightly divert the trajectory to either the right or left side (depending on the field orientation). Due to the alternating fields, the trajectory becomes a kind of zig-zag course. During that path, many photons are produced (3) and the wave length could be influenced by the distance between the upper and lower row of magnets (the so called gap size) and the energy of the electron bunch (due to the acceleration process). Usual wave lengths of photons produced are between 0.1nm and 10nm (hard to soft X-Rays). The photons generated are guided to special isolated rooms (hutches), where experiments could be carried out. These technologies have been implemented on the DESY premises at accelerators like PETRA III.

The photon pulses produced have a limited but continuous wave length, as the accelerated

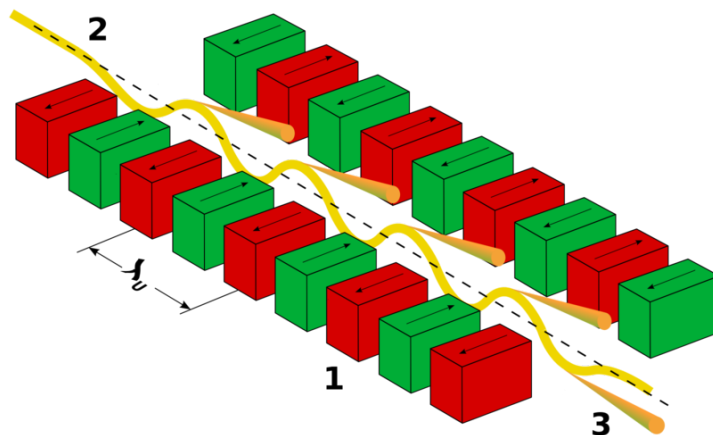


Figure 1.1: *Alternating structure of fixed magnetic fields (1) in order to divert the trajectory of a passing electron bunch (2) onto a zig-zag course. This will create photons (3) due to Bremsstrahlung along the way and leave the structure at the end together with the electron bunches. This structure is called Wiggler or Undulator (depending on the design properties like the magnetic field force and wave length λ_u and the resulting effect). (Picture from Wikipedia)*

electrons will carry non-discrete energies, which are the bases for the photon generation process in the wiggler. Additionally the angle under which the photons leave the wiggler is relatively wide and no interaction of the photons is happening. However, for many experiments (especially for imaging of molecular structures) a coherent and discrete wave length photon pulse with a small angle is of great interest. Then diffraction images of investigated samples could be generated and captured with 2D image detectors. This possibility has been achieved with the development of the Free-Electron-Lasers like the existing FLASH at DESY and the European XFEL under construction - which will be introduced in the next section - and other machines world wide.

Today DESY focuses its work and research area on the following fields:

- Design, construction and operation of accelerators
- Scientific research with photons

- Scientific research in particle and astro-physics

1.3 European X-Ray Free-Electron Laser Facility

The European X-Ray Free-Electron Laser (XFEL) Facility is a European project, internationally funded and built between the DESY premises in Hamburg and the adjacent federal state Schleswig-Holstein in northern Germany (see Figure 1.2). The whole machine will have an ap-

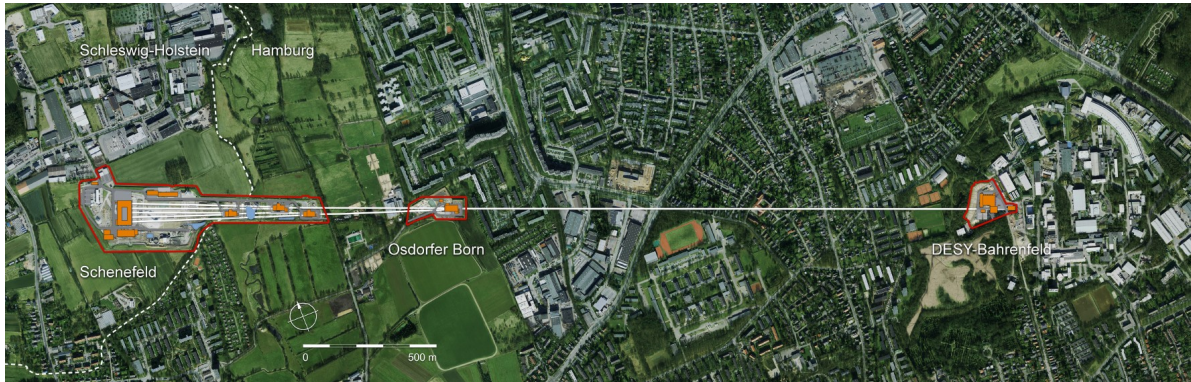


Figure 1.2: *Photo of the location around the European XFEL. Starting from the DESY site in Hamburg (right side), the photons will be available in the experiment area in Schenefeld (left side). (From European XFEL picture data base)*

proximate length of 3.4km and follows the Free-Electron-Laser principle (a summary of history of FELs and the European XFEL in special as well as detailed technical design information and planned research fields could be found in the Technical Design Report (TDR) [1]). On the DESY site electron bunches will be generated, when a laser pulse hits on a cathode. These bunches will be accelerated, where the imposed energy will increase with each accelerating module along the path. After acceleration and compression of the bunches, they will pass a structure called undulator, which is very similar to the previously described wiggler. The main differences are, that (1) the amplitude of the zig-zag course is smaller, which leads to a smaller angle under which the photon pulses will leave the structure, and (2) the generated photons will interact with each other by generating interference and (3) interacting with the electron bunches stimulates the photon generation effect in a way, that (4) laser pulses will be generated, which provide coherent and almost discrete wave length photon pulses. This process is called self-amplified spontaneous emission (SASE) and is the basis for the lasing effect.

The machine can logically be viewed as three parts (see again Figure 1.2 from right to left): (1) the electron accelerator - between DESY Bahrenfeld and Osdorfer Born, (2) the undulator and photon beam lines - between Osdorfer Born and Schenefeld Site - and (3) the experiment stations on the Schenefeld Site. The electron accelerator part consists of a single path, which is shared with all following sections. There the electron bunches are created, accelerated, focused, compressed and monitored. In the first junction point, the bunches are directed into different beam lines. There different undulators (in the beginning three will be available and denoted as SASE 1 to 3) will generate photons with different wave lengths, which are guided to the end of the beam lines to the experiment stations. The electron bunches will be destroyed (dumped) on the way. Per beam line, entering the experiment area, two experiment stations are planned, which add up to six altogether in the beginning. The current layout allows two more undulators, which could be added on an upgrade procedure in the future together with the related experiment stations.

The design and development of the European XFEL started in 2005. In 2009 a new limited

liability legal entity called European X-Ray Free-Electron Laser Facility GmbH was founded in order to adequately represent the international share holders and define the ownership of the machine. In this context DESY was assigned to be responsible for the construction and operation of the accelerator part of the European XFEL. The commissioning of all systems is planned for 2016 and user operation is supposed to start in 2017.

1.4 Scope of work

The work described in this document is focused on the conceptual design, development, setup and evaluation of the European XFEL Timing System, which will be the technical implementation of a facility-wide clock distribution, synchronization and sequencing system for all data acquisition and control electronics at the European XFEL. Based on the author's previous studies and findings [2] and the requirements, which will be presented in the following section, a suitable and economically efficient solution has to be developed and with the support of DESY staff, external collaborators and companies, installation-ready series production level modules have to be prepared.

The following chapters will provide background information, a summary of the different design, development and evaluation phases and further information about interfacing subsystems and integration possibilities for other accelerators. In the acknowledgment section at the end of this document credit is given to all involved persons, institutions and companies, who and which contributed to the realization of this ambitious project.

1.5 Requirements

This section provides an overview of the requirements of the Timing System to be developed. More detailed information and definition of terms will be provided in the following chapters and in the lists of abbreviations and symbols at the end of this document.

1.5.1 Required functions

The Timing System to be developed is required to provide at least the following functions:

- Distribution and generation of derived clocks synchronized and syntonized to the RF master oscillator (see section about frequencies below)
- Distribution of events as the basis for generating time adjustable and synchronized triggers and gates
- Distribution of deterministic data related to fast changing parameters of the European XFEL (e.g. bunch pattern)
- Define the repetition rate of the macro pulse (see following paragraphs)

1.5.2 Bunch structure

The Timing System to be developed has to be compatible with the planned bunch and time structure of the European XFEL and the existing Free-electron LASer in Hamburg (FLASH), as almost all systems under development should be compatible with both machines. Table 1.1 summarizes the important bunch and time structure parameters for both facilities and Figure 1.3 illustrates the parameters in a time diagram:

| | European XFEL | FLASH |
|--|---------------|----------------|
| Length of macro pulse | 600 μ s | 800 μ s |
| Time between macro pulses | 100ms | 100ms |
| Max. number of bunches per macro pulse | 2700 | 800 or 2400 |
| Min. time between bunches | 220ns | 996ns or 332ns |

Table 1.1: Summary of important bunch and time structure parameters for the European XFEL and FLASH.

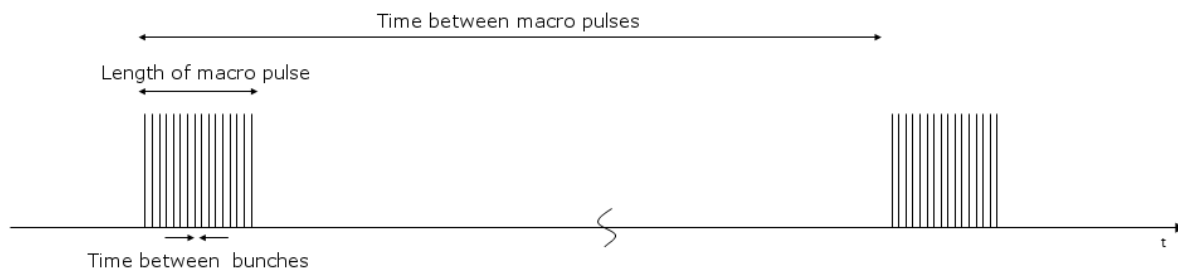


Figure 1.3: Generic time structure of bunch delivery and macro pulse for the European XFEL and FLASH.

1.5.3 Frequencies

The Timing System to be developed has to connect to the master oscillator of the European XFEL or FLASH and therefore has to be compatible and syntonize to the frequency and synchronize to the phase of the 1.3GHz reference of the machine. Furthermore the Timing System has to synchronize to the phase of the 50Hz voltage oscillations on the power line for deriving the 10Hz macro pulse repetition rate (see above).

Besides that, at the receiver side of the distributed timing signals, derived frequencies from the reference have to be generated, which have a fixed phase relation and maintain this relation across macro pulses. The phase relation should be kept even after a system restart or power cut. The rounded values of the most important frequencies to be provided by the receiver are shown in Table 1.2 along with their accurate divider value of the 1.3GHz reference clock:

| Generated Frequency (rounded) | Divider value for 1.3GHz |
|-------------------------------|--------------------------|
| 1.3GHz | 1 |
| 216MHz | 6 |
| 108MHz | 12 |
| 81MHz | 16 |
| 54MHz | 24 |
| 9MHz | 144 |
| 4.5MHz | 288 |
| 1MHz | 1296 |

Table 1.2: List of rounded values of most important frequencies and the accurate divider value for the 1.3GHz reference to be provided by the Timing System receiver side.

1.5.4 Stability requirements

The time jitter of the distributed 1.3GHz reference clock at the receiver side should be within 10ps RMS relative to the reference clock of the master oscillator. For derived clocks at the

receiver side, which can also include user definable phase delays, a slightly higher jitter is acceptable, but should be kept as low as possible. The jitter requirement is mostly motivated by the low jitter requirements of fast analogue to digital converter (ADC) based diagnostics and detectors, which have no additional jitter cleaner in their clock input path and higher jitter would reduce the resulting signal-to-noise ratio beyond acceptable limits. In many cases fast transient signals are captured by ADCs with significant under sampling, where the phase variation between detected signal and sampling clock has to stay within boundaries as small as 10ps

The phases of generated triggers have to be as stable as required to unambiguously identify a certain clock period of the derived clocks at the receiver side (see above) - in the order of 100 ps. This number is motivated by the requirement to safely identify a certain period of the 1.3GHz frequency.

1.5.5 Hardware platform

The chosen hardware platform for most of the fast data acquisition and control electronics will be MicroTCA, the Timing System to be developed has to be compatible to this standard. However, as many subsystems of the European XFEL will not be available in MicroTCA standard, also external interfaces have to be provided in order to provide timing related signals to external components.

1.6 Structure of this document

The structure of this document can be divided into four parts: The first part, consisting of chapters one to three, gives an introduction to the topic, the facility and requirements (1), important terms and definitions (2) and provides an overview about timing system technologies at other light sources (3). The second part (chapters four to ten) is concerned with the actual design (4), evaluation of the design (5), a description of the two generations of developed timing system modules (7 and 8), a description of the firmware (9) and software (10) functionalities. The only exception is chapter six, which provides an introduction to the MicroTCA standard, which is required in order to understand the functionalities of the timing system modules. The third part (chapters eleven and twelve) presents how the developed timing system can be used in terms of interfacing to the consumers (11) and interfacing to other timing systems (12). Finally, the fourth part closes the work with conclusions (13) and the acknowledgments of all persons, who contributed to this project (14).

Chapter 2

Definition of terms

Learning vocabularies and their meaning is crucial in order to understand a foreign language. Although the terms described here are in English language and most of them are even commonly known, it is even more important to introduce them, as their interpretation might be different and partly depends on the context.

This chapter will introduce and briefly describe important terms related to timing systems, their applications and performance measures.

2.1 Timing System

Although the description of the Timing System requires some of the following terms, it will be presented first in order to provide an overview in what sense the other terms relate to the actual topic of this work. This section covers important features of the timing system. More details will be provided later on in other chapters.

As the name already indicates, the timing system is a combination of different components dealing with time related signals. One important aspect of the timing system is the distribution of signals, which allow numerous subsystems along the European XFEL to work synchronously with respect to the electron bunches or photon pulses and other subsystems like analogue-to-digital converters (ADCs), processing units and digital-to-analogue converters (DACs). These systems usually require clocks in order to define the sampling or processing cycles and one or multiple triggers, which define the points in time, when to start sampling or processing. Therefore the Timing System has to be distributed and provide those signals to all subsystems in a synchronized and stable way.

Many parameters of the machine will change during the run-time like the number of bunches, the pattern of bunches, the charge, the path through the machine and so on. Those parameters influence many systems along the machine. There will always be a certain point in time, when one parameter is changed and it is important to provide a deterministic information channel via the timing system, which guarantees facility wide synchronous switch-overs of parameters. The European XFEL will operate continuously over many years. During that time a high data volume is collected from various detectors and diagnostic systems. In order to make use of these collected data, it is crucial, that data sets (e.g. values for a certain bunch) from all sources can be correlated. Therefore a unique ID or synchronized time stamp has to be assigned to all data, which identifies each single data set facility-wide. This information has to be provided by the timing system as well.

The timing system is a facility-wide distributed system, which provides the above mentioned signals and information based on a centralized reference and information provided by the operators of the machine taking its current state and allowed parameters into account.

2.2 Clock

A clock is defined as a periodic signal with one rising edge per period. All other parameters like frequency, phase, signal levels and standards, shape and transport medium are context dependent. Important for clocks provided by the timing systems is the facility-wide synchronization of the phases and syntonization of the frequency. The reference frequency, provided by the master oscillator, is 1.3GHz. There are different frequencies available, but there is always a fixed phase and frequency relation between them maintained.

Common users of clocks are ADCs, DACs, FPGAs, fast serial transceivers.

2.3 Trigger

A trigger is defined as a pulse, where its rising edge indicates a certain point in time. Besides the above described usage of clocks in almost all subsystems a trigger is important to indicate a point in time, when to start sampling data or when to acquire a picture or when to process data. Triggers in the same meaning are also used at oscilloscopes to define a certain point in time.

Triggers can also be periodic. For European XFEL a 10Hz cycle of recurring procedures is planned. Therefore most of the defined triggers will be periodic with 10Hz. In general the time between triggers can be long (maybe even only one time at all) or short (if used as bunch trigger, where trigger pulses can have a period in nano-second range). This is often combined with a burst mode, where the periodicity is only within a certain time window, which can then also be repeated on a regular basis.

2.4 Gate

A gate is very similar to a trigger. The main difference is, that not only the rising edge of the pulse is relevant, but also the falling edge. Therefore gates are usually defined by two certain events, where the first one generates the rising edge of the signal and the second defines the falling edge.

Common application of a gate signal is to define the time frame, where certain signals are to be considered. Examples are cameras where photons are collected on a 2D detector or counters, where pulses from a detector are counted.

2.5 Syntonization

Syntonization defines the property of at least two periodic signals to have the same frequency. In case of the timing system, the important aspect is to syntonize the frequencies at the end points with the reference frequency provided by the master oscillator.

The frequency f of a periodic signal is defined by the inverse of the time of one period t_{period} .

$$f = \frac{1}{t_{period}}$$

The phase of a periodic signal can be defined as

$$\phi(t) = 2\pi ft + \phi_0$$

where ϕ_0 is the initial phase.

If two periodic signals with the same frequency f start with the same phase ϕ_0 , the phase

between them will be maintained for all values of time t .

$$\Delta\phi(t) = \phi_1(t) - \phi_2(t) = (2\pi ft + \phi_0) - (2\pi ft + \phi_0) = 0 \quad (2.1)$$

However, if the frequency is only slightly different, the phase relation between both signals will change over time:

$$\Delta\phi(t) = \phi_1(t) - \phi_2(t) = (2\pi f_1 t + \phi_0) - (2\pi f_2 t + \phi_0) = 2\pi t (f_1 - f_2) \quad (2.2)$$

The time dependent phase difference has some undesirable effects:

- Especially in sampling applications at frequencies between 1MHz to 200MHz the sampling point should be adjusted to capture certain points of a period of a signal (e.g. the peak and baseline). If the phase of the sampling frequency is not constant with respect to the signal to be captured, it is not possible to sample fixed points during one period. If the sampling frequency phase is changing, the desired sample points will not be maintained.
- Usually triggers are used to define a certain point in time during data acquisition or processing. In most applications the next rising edge of a clock after the trigger is used to act - like starting the sampling process. The trigger and the phase of the clock have to maintain a fixed relation in order to provide enough setup and hold time to reliably identify the trigger point. If the phase between the trigger (or more precisely the clock used to generate the trigger) is changing compared to the used clock, a variable time is added to each of the trigger events.
- In high speed sampling application (e.g. in GHz range) the provided frequency is multiplied in order to provide the high sampling frequencies. If the base frequency is slightly higher or lower than the frequency of the signal to be digitized, the number of samples to capture will be higher resp. lower in order to digitize the complete signal.

Syn-tonized frequencies are limited to exactly the same frequency. However, in real applications also other frequencies have to be generated at the timing system endpoints. In this case, another frequency f_2 is generated by dividing and/or multiplying frequency f_1 by an integer value. In this way a fixed frequency relation is maintained over time.

2.6 Synchronization

The term synchronization defines certain properties of at least two signals to happen at the same time (with finite accuracy). Two examples are the rising edge of triggers or clocks at different timing system endpoints.

The problem of time dependent phase differences as a consequence of un-syn-tonized frequencies have been discussed previously. Here the effect of (un-)synchronized phases between two signals with syn-tonized frequencies will be discussed.

If we assume the general case of equation (2.1), where the initial phase ϕ_0 can be different for both signals and denoted as $\hat{\phi}_1$ and $\hat{\phi}_2$ it results in

$$\Delta\phi(t) = \phi_1(t) - \phi_2(t) = (2\pi ft + \hat{\phi}_1) - (2\pi ft + \hat{\phi}_2) = \hat{\phi}_1 - \hat{\phi}_2 \quad (2.3)$$

The previous equation defines, that the phase difference between the two signals is constant, but depends on the initial phases. Assuming, that the two signals could be the clocks of two timing system endpoints or the reference frequency from the master oscillator, this will result in different potential problems:

- If the phase between clocks is arbitrary, the phase has to be adjusted in order to find the correct phase for certain applications (e.g. the sampling points of a signal).
- If the initial phase is arbitrary and different on every system start-up, the previous adjustment has to be repeated every time

Similar synchronization problems occur, if a clock is divided at the timing system endpoints. The point in time, when the dividing process is started, defines the initial phase. Therefore, uncertainty of this point in time will lead to the very same problems discussed above. Solutions to these problems involve synchronization of the initial value of the dividers and point in time to start the process, which will be discussed in the following chapters as well as the synchronization of triggers.

2.7 Deterministic data transmission

There are many parameters to be adjusted in order to operate a large machine like the European XFEL. Most of them are specific to certain subsystems and have no or just limited influence on other systems. Some parameters, however, are crucial and have very strong influence on almost all subsystems. If one of these parameters is changed, other systems have to react on this ideally exactly at the time when it has been changed. Such crucial parameters include the number of bunches to be generated, the charge of the bunches, the way they take through the machine etc. All these parameters are defined by the operators. Changes have to be validated with the machine protection system, that they lead to an allowed operation state and then can be applied system-wide at the same time in a synchronized way. Usual network connections do not provide deterministic transmission of the data in order to guarantee in-time delivery and synchronous evaluation of the parameters. Therefore the timing system has to implement a distribution channel for such data.

Along with other information (e.g. time stamps and unique identifiers) the timing system has to distribute this information and provide it to all interested consumers in a synchronized way. It has to be early enough in order to apply changed parameters to local systems before the actual bunches will be generated and injected. This system is denoted as deterministic data transmission.

2.8 Drift

The term drift will be used to define a time-variant change of certain parameters within a longer time frame (days down to milliseconds). The most important influencing parameter is the phase of clocks. Predominant cause of such drifts are temperature changes, which could influence propagation delays in fibers, copper cables or traces. It affects integrated circuits, which can also induce voltage changes, which indirectly can lead to phase changes. These effects will be discussed more detailed in the following chapters.

2.9 Noise

Noise can be defined as undesired influences on amplitude and phase of a signal. Noise can be divided into two types: deterministic and stochastic noise. Deterministic noise is mostly caused by a component in the system design and, if identified, can be removed or at least reduced in most cases. One example of deterministic noise is an undesired periodic distortion introduced by the switching frequency of a DC/DC converter. If not included in the design in the right way it can influence connected circuits in an undesired way. Deterministic noise is

characterized by non-stochastic effects generated by certain components or technologies. Stochastic noise is visible as a random signal fluctuation. There are different known and for this application partly relevant causes producing stochastic noise like thermal noise [3] [4], shot noise [5], flicker noise, and burst noise which will be described briefly.

2.9.1 Thermal Noise

Thermal noise (also known as Johnson or Nyquist noise) is introduced by thermal agitation of the electrons. These fluctuations do not depend on the current, voltage or frequency. The power spectral density resulting from voltage fluctuation of a resistive conductor is calculated as

$$\overline{v_n^2} = 4k_BTR$$

where R is the resistance of the conductor in Ohms, T the temperature in Kelvin and k_B is Boltzmann's constant in Joules per Kelvin [3] [4].

As the value does not depend on frequency, it represents a horizontal line in the power spectrum and is also referred to as white noise.

2.9.2 Shot Noise

Shot noise arises from the fact, that electrons carry a discrete charge and the number of electrons emitted by a source fluctuates slightly over time. Especially in cases, where potential barriers have to be crossed, the effect becomes visible as a stochastic effect and not as continuous flow of charges. This effect becomes more visible, if the number of electrons is small (meaning low currents), as then the signal to noise ratio becomes smaller. For higher currents the overall noise becomes dominated by other noise sources. This is why it is of less importance for the timing system described in this document. More details on shot noise are given in the original documents from W. Schottky [5] and J.B. Johnson [6].

2.9.3 Flicker Noise and $1/f^n$ -Noise

Based on the previously cited publications [5] and [6], J.B. Johnson measured an effect, which deviated from the theory. For lower frequencies the noise was significantly higher than expected. Based on those measurements W. Schottky later described that effect as "Funkelrauschen", which has been translated into English as flicker noise, as it describes a flicker-like effect on the surface of the cathode used in the experiments. This effect defines a $1/f^2$ noise behavior and strongly depends on the used material.

This effect relates to the so-called $1/f$ -noise, which generally describes noise effects, whose power spectral density attenuates with increasing frequency. Due to that characteristic the effect becomes visible at lower frequencies, where it might dominate over other noise sources. In oscillators and RF mixing techniques, where base band signals are shifted to higher frequencies, existing $1/f$ -noise from the base band might be shifted up to higher frequencies and can dominate the noise behavior.

$1/f$ and $1/f^2$ noise (also denoted as pink resp. red noise) often becomes visible in phase noise measurements as used in this document. Main causes are semiconductor based circuits like amplifiers and buffers.

The original publications on that topic are found in [7] and [8].

2.9.4 Burst Noise

Burst noise is a phenomenon observed in semiconductors (first observed in the operational amplifier 709). Other names for the effect are popcorn noise or random telegraph signal. The

effect creates random offset voltage jumps in the range of microvolts over time periods of milliseconds up to seconds. Some more detailed information about this effect can be found in application notes from integrated circuit manufacturers like [9] and [10].

2.9.5 Amplitude, Frequency and Phase Noise

In general noise effects can influence the stability of signals. For applications, where digital signals are processed, amplitude fluctuations are usually not as critical as phase or frequency fluctuations. However, depending on the transmission and processing technologies used, a conversion between all three types of fluctuations is possible. A simple example is the reception of a transmitted signal. In the receiver an electrical or optical signal has to be detected. Real signals do not follow an ideal rectangular shape (mostly due to band limited communication channels or drivers), but have finite rising and falling edges. Amplitude noise on these edges can lead to a time fluctuation in the detectors, as a transition might be detected too early (if the noise increased the amplitude slightly) or too late (if the noise reduced the amplitude of the rising edge). The phenomenon is usually called AM-to-PM conversion. An example of frequency noise would be a voltage controlled oscillator (VCO), where the frequency can be changed based on a change on the input voltage of the device. Therefore amplitude noise on the input would convert into frequency noise. Finally, an adjustable delay component, where an input voltage can change a signal transmission delay of the signal passing through the device would introduce a phase noise, if the input voltage is carrying noise.

Even if noise on currents and voltages will only introduce amplitude fluctuations, they might be converted into frequency or phase noise via additional components in a general non-linear way. Special care has to be taken in order to reduce influence of noise on system stability.

2.10 Jitter

Jitter is a measure of integrated phase noise. In many cases the phase noise spectrum is not of main interest. It is more important to know the expected RMS value of the phase fluctuations of a signal (within a certain time window). One way of measuring this value is to measure the phase noise spectrum and calculate the RMS jitter value by integrating the phase noise spectrum over the relevant frequency range. A way of defining the relevant frequency range relative to the carrier is to set the lower boundary to the value, where the changes are regarded as drifts (e.g. 1kHz). The higher value might be defined by the physical limitations of the user of the signal. In most cases, the contribution to jitter from higher frequencies is very low, so that the higher boundary is of limited relevance.

An alternative way to determine the jitter of a periodical signal is described in the next section and is defined as Time Interval Error measurement.

2.11 Time Interval Error

The time interval error measurement is an alternative way to measure the jitter of a periodic signal. The measurement can be done in the following way: the signal is measured with an oscilloscope at much higher frequency as the periodic signal to be investigated. Software can then calculate the average period of the digitized signal and afterwards calculate the deviation of each cycle in the data trace. The result represents the individual time interval errors and allows to calculate peak-to-peak deviations, RMS deviations, standard deviations and even histograms and spectral information of the measured data. However, the accuracy of the measurements is limited by the stability and value of the sampling frequency as well as on the vertical resolution and internal noise sources.

2.12 Dispersion

Dispersion in the context of signal transmission over optical fibers describes the dependency of the phase delay on the wave length of the signal. High speed serial signals, like the ones used for timing system signal distribution, require high bandwidth in the order of multiple GHz. All frequency components of the transmitted signal will pass the fiber with different phase delays resulting in a common group delay [11]. The side effect of the dispersion is a broadening of the transmitted data pulses.

The dispersion also depends on the fiber and on the wave length of the carrier. Different possibilities exist in order to reduce or compensate the dispersion effect on the transmitted data signal and will be discussed in a later chapter and can also be found in [11] [12]

2.13 Inter Symbol Interference (ISI)

Inter Symbol Interference (ISI) is closely related to dispersion. If one data word is denoted as symbol and transmitted over a medium, the data word can be recovered on the receiver side by detecting the symbol. However, due to dispersion or other effects, the symbols can be stretched over time on the way through the medium and two or more symbols can be overlapped at the receiver side. This is defined as ISI.

If ISI can not be avoided, there are solutions available in order to reconstruct the individual symbols if the concrete effect of the medium is known. More information on this topic can be found in [13]

2.14 Resolution

The resolution of the timing system, as used in this document, defines the smallest step in time for events, which can be adjusted system-wide without influencing all other events. It therefore depends on the internal time base of the timing system and defines its granularity. In most cases the granularity is defined by the period of the reference clock or a subharmonic of it.

Chapter 3

Timing system technologies and usage at Light Sources

Timing and synchronization is an important aspect of the European XFEL as it is for other light sources, accelerators and industry applications. Before starting to go into the design details of the Timing System for the European XFEL, this chapter will provide an overview of different technologies available in order to implement timing systems as well as some information about which facilities are using them. This overview will not be complete, as many different implementations and variations exist. Therefore it focuses on general concepts and example implementations.

There are different ways of categorizing timing systems and structuring this chapter. The one chosen here is based on the type of time referencing pursued and finally provided by the described timing systems. Following this way, one could divide the timing systems into three categories: (1) distribution of coordinated universal time, (2) bunch clock distribution and (3) clock and event distribution. Those will be described in the following sections along with example implementations providing different levels of accuracy and stability.

3.1 Distribution of Coordinated Universal Time (UTC)

The coordinated Universal Time (UTC) were defined officially in 1961. It provides a universal time base, from which all local time zones can be derived.

Timing Systems of this category have the main purpose of distributing this universal time to numerous end points by maintaining the accuracy and stability required for the dedicated application.

3.1.1 Network Time Protocol (NTP)

The Network Time Protocol (NTP) [14] is a famous member of this timing system category. The task of this protocol is to distribute the universal time within the Internet via packet-based lower level protocols. In this environment variable latencies are expected, as the packets will pass switches and routers on the way, which store and forward the packets with variable time. Also packet losses are possible, as NTP uses state-less UDP (User Datagram Protocol) for transmission. NTP provides a distribution system for UTC to computers connected to the Internet and takes the described boundary conditions into account.

The Network Time Protocol defines a hierarchy of clock sources, called strata (see Figure 3.1). The highest level is Stratum 0 and denotes devices providing the UTC directly like atomic, GPS, radio or other clocks. Those are connected to computers, which usually act as NTP servers. These computers are defined as Stratum 1. They act as the time bases for the next

computers level (Stratum 2), which are sending NTP requests to the Stratum 1 devices. This will go on in the same way between further Stratum layers as shown in Figure 3.1.

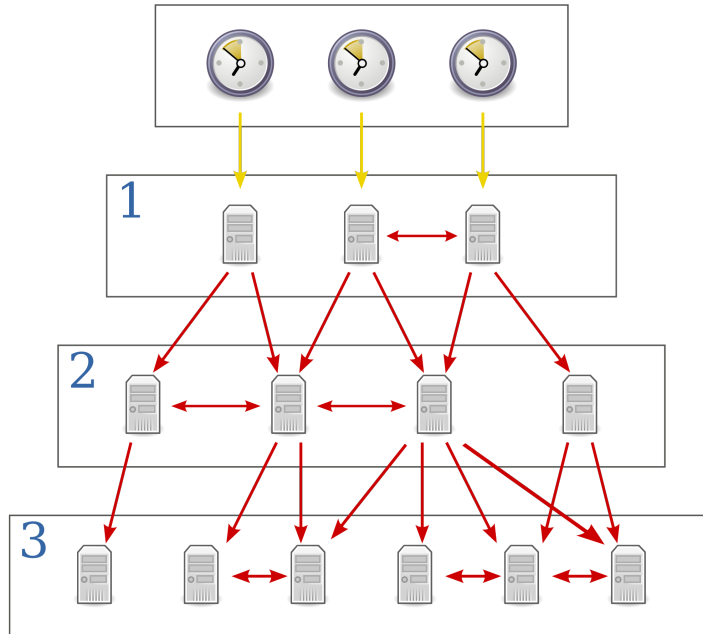


Figure 3.1: *Hierarchy in a Network Time Protocol system. Highest layer (Stratum 0) defines clock sources which are directly connected to the Stratum 1 layer (yellow arrows). Further Stratum layers will communicate with other computers over network (red arrows). (©B.D. Esham)*

NTP uses 64bit time stamps in order to define a point in time. The upper 32 bits define the seconds since January 1, 1900 and the lower 32 bits the fraction of a second, providing a resolution of 233ps. The time stamp size is expected to be doubled in size in the future avoiding roll overs of the seconds part and provide higher resolution.

When a client receives the current time stamp from the NTP server, it gets the absolute time as it was sent by the server. In order to adjust for the transmission delay, it uses a delay measurement depicted in Figure 3.2. The NTP client sends a NTP request to the NTP server and saves its local time stamp (t_1), when the request was sent. The NTP server saves a time stamp when the request was received (t_2). It processes the request and sends back a packet with the send time (t_3) and also the time stamp t_2 . When the client receives the packet it

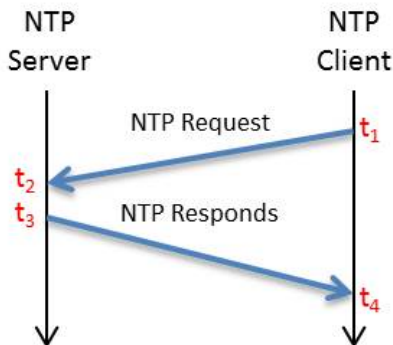


Figure 3.2: *Measurement of time delay between NTP server and client.*

saves the time stamp t_4 and can then calculate the transmission delay by

$$\Delta t_{server-client} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (3.1)$$

It is the time between sending the request and receiving the reply minus the processing time at the server side and then divided by two in order to just count one trip (server to client).

The underlying principle assumes symmetry of the transmission time for request and reply. In real systems this is not true. Deviations from symmetry introduce uncertainties and therefore errors in the transmission delay calculation and the calculated absolute time. This could be improved by using more than one NTP server as reference and calculating averages as well as removing NTP servers from the list, if they deviate too much from the expected value and therefore might be wrong.

Performance measurements showed, that accuracies between tens of micro seconds RMS (LAN) and 60ms RMS (world wide Internet) are typical for NTP [15].

3.1.2 IEEE1588, SyncE and White Rabbit

White Rabbit, known from Charles Lutwidge Dodgson's (alias Lewis Carroll) novel "Alice's Adventures in Wonderland" [16] as a creature always concerned about punctuality, is the code name for a timing system implementation designed at CERN¹ with other laboratories and industry. The concept of this system is to combine established industry standards related to timing and synchronization and extend them in order to achieve stabilities and accuracies beyond their limits. The implementation is built on Gigabit Ethernet as defined in IEEE802.3 in variant 1000BASE-X.

IEEE1588 defines a Precision Time Protocol (PTP), which allows to synchronize clocks of multiple slave nodes in an Ethernet network to a master clock using the coordinated universal time (UTC). In order to achieve this it is crucial to determine the transmission delay between the master and the slave in order to compensate for it (see Figure 3.3). The link delay is

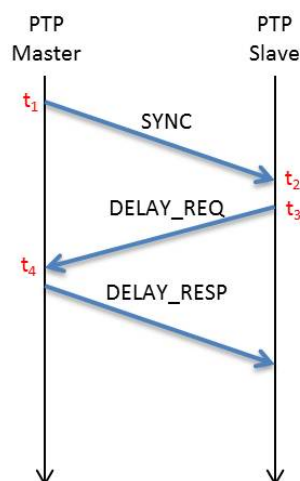


Figure 3.3: Time flow diagram of the link delay measurement procedure of IEEE1588.

measured by special procedures similar to the NTP implementation, where the master sends a packet with a time stamp from its own clock at transmission time (t_1). A slave receives it and saves the time when it was received relative to its own (currently unsynchronized) clock

¹European Organization for Nuclear Research, <http://www.cern.ch>, CH-1211, Gen'ève 23, Switzerland

(t_2). Then the slave sends at time t_3 a packet to the master which will be time stamped by the master at receiving time t_4 and this number then sent back to the slave. Finally the slave is able to calculate the link delay between master and slave as

$$\Delta t_{\text{master-slave}} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (3.2)$$

The accuracy of this system depends mainly on two aspects:

- Symmetry of link delays: the protocol assumes, that the link delay is symmetric in order to determine the transmission delay between master and slave by dividing the round trip time measured by two. In real networks this is not true in all cases.
- Delays of other components: Besides the link delay, there are other sources of variable and fixed delays in the communication channel, which are not considered. Examples are uncertain latency in serializer/de-serializer (PHY) chip and store-and-forward switches in the communication path.

Even if the time stamping of the packets is done in hardware in order to not rely on non-deterministic software layers defining the time information and switches are excluded the accuracy of synchronization is limited to tens of nanoseconds [17].

In ordinary Ethernet networks each communication node has its own reference clock, which is used to prepare and process the data. However, especially on high-speed Ethernet communications channels like 1Gb (1000BASE-X) and 10Gb Ethernet each receiver has to recover the clock of the sender in order to be able to detect the bits correctly. In ordinary networks the data is afterwards transferred in the local clock domain of the node to be further processed. This clock domain crossing introduces uncertainties related to synchronization and the use of local unsynchronized oscillators eliminates synchronization. The solution to this problem is implemented by Synchronous Ethernet (SyncE). In that implementation the node in the network will use the recovered clock for the internal processing and for data transmission. This eliminates the need of crossing clock domains and also provides identical frequencies at all nodes on the network.

White Rabbit combines SyncE and IEEE1588 in order to provide synchronicity and synchronicity in the network. The combination of both protocols allows to reduce the amount of resynchronization procedures of IEEE1588, as the clocks run with exactly the same frequency and therefore only changes in link delay could cause de-synchronization. But the accuracy of the system is still limited to tens of nanoseconds. In order to improve the system to sub-nanosecond precision additional hardware and protocol changes are implemented:

- Reduction of asymmetries: in order to reduce asymmetric behavior, transmission is done with a single optical fiber, where both sides transmit at different wavelengths (usually 1310nm and 1550nm). Delay differences for the different wavelengths are defined by the diffraction indices and are theoretically known and could even be measured for a certain installed fiber for demanding applications.
- Determination of transmitting and receiving latencies: common serializer/de-serializer chips have an uncertainty of latency due to an internal PLL and dividers, which starts up with an arbitrary phase relation. In order to eliminate the latency uncertainty it is measured and compensated for within an FPGA.
- Increasing of time stamp resolution: the accuracy of the IEEE1588 procedure is limited to the single packet boundaries (for 1Gb Ethernet it is 125MHz which is 8ns). In order to increase the resolution a phase comparator is implemented in the FPGA to measure fractions of the packet size (below nanoseconds).

- Related to the previous points, the protocol of the IEEE1588 have been extended in order to allow detection of compatible nodes, provide special data for calibrations and to transmit sub-nanosecond delay information.
- Invention of special White Rabbit switches: As the higher resolution comes with hardware and protocol changes, special switches have been invented implementing the required changes in order to fan-out and distribute accurate timing.
- Besides the measures to increase the resolution and accuracy, the jitter is reduced by including a PLL, which will "clean" the locally syntonized clock.

Measurement results published in [17] [18] [19] showed impressive results, taking into account, that almost all functions are implemented in FPGA. It was shown, that variations of temperatures of a 5km long fiber on a spool from 12.5 centigrade to 85 centigrade resulted in phase changes below 100ps (measured between the master reference clock and slave clocks of the external PLL). Comparing the Pulse per Second (PSS) pulse generated every second by the master and slave FPGAs the variation was higher with around 450ps peak-to-peak.

Although the White Rabbit Timing System is based on industry standards, the extensions to it require significant hardware changes, which lead to the fact, that no commercial hardware like the special switch was available. But as all designs, firmware and software is provided under open hardware and open source licenses and industrial companies are interested.

3.2 Bunch clock distribution system

In most circular accelerator (synchrotron) based light sources the timing relations are relatively simple. The storage ring has a defined circumference where the particle bunches require a time t_{ring} in order to complete a full turn of the ring. The period of the accelerating RF t_{RF} is required to have an integer relation to t_{ring} as

$$t_{ring} = t_{RF} * N \quad (3.3)$$

In this case N defines the number of bunches which could exist at the same time within the ring. Those places are also called buckets which could be filled with a bunch or not.

The main goal of a typical timing system for such facilities is a clock distribution system, which provides the RF reference frequency ($1/t_{RF}$) to all end stations. At the end stations this clock will then be divided and delayed in order to generate all required signals for different applications like clock, gates or triggers for ADCs, TDC and so on. In this case the dividers and delays have to be calibrated for a certain system in order to correspond to the actual filling pattern of the ring.

In order to simplify the local divider synchronization some timing systems also allow to distribute the ring (orbit) frequency ($1/t_{ring}$). This is either implemented as a second distribution cable or by sending a fiducial (or data packet) on the same cable (depends on the implementation). This will be used in order to synchronize the local dividers.

The following paragraphs provide two examples of such timing systems.

3.2.1 European Synchrotron Radiation Facility (ESRF)

At ESRF² the reference frequency of 352.20MHz is generated by the master oscillator and then distributed to the end points. At the end points special devices like the BCDU8 unit are used, which allow flexible dividing and delay adjustments for multiple outputs. In special

²European Synchrotron Radiation Facility, 6 rue Jules Horowitz, 38000 Grenoble, France

cases a second cable with the orbit frequency ($1/t_{ring}$) of 355.04KHz is provided and used to synchronize the dividers of the described unit.

3.2.2 PETRA III

Another example of a bunch clock distribution is used at PETRA III at DESY. This implementation provides a more integrated solution compared to the previous example, as the receiving unit includes already the divider and delay units and also special trigger outputs.

The reference frequency of PETRA III is 500MHz generated by the RF master oscillator.

3.3 Clock and event distribution systems

This class of timing systems is designed to distribute a reference clock (usually provided by the master RF oscillator of the accelerator) and events based on certain hardware or software conditions. The generator and transmitter of the signal are usually called event generator (EVG). In most cases the signal of an EVG is multiplied by a fan-out unit and then distributed to the event receivers (EVR). The receivers have to decode the events and recover the reference clock. The events will be converted into definable actions like providing triggers, generating interrupts, gating of signals and so on. The recovered clock, fractions of it (generated via dividers) or gated versions will be provided at the output. Figure 3.4 illustrates the general system layout.

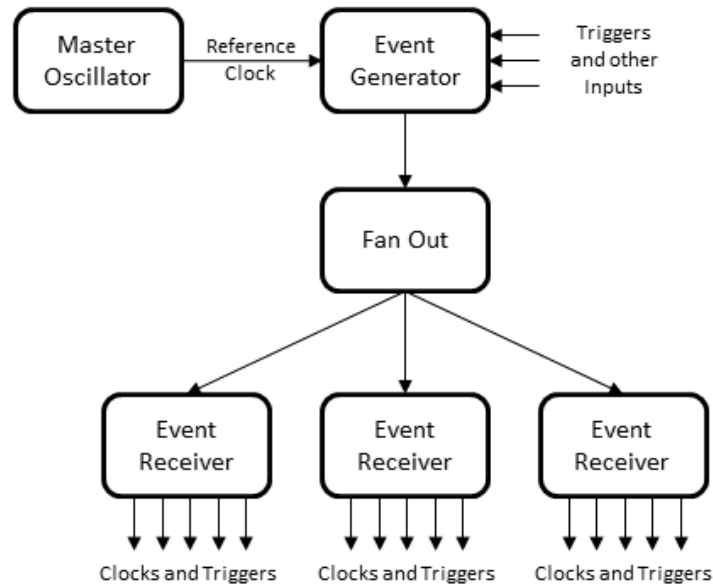


Figure 3.4: Overview of a general event based timing system. The event generator is connected to the master oscillator providing the reference. External triggers define the basis for events. The event stream is generated and distributed via a fan-out to multiple event receivers. They decode the events and provide clocks and triggers to local applications.

Those systems, as used at different facilities, do not have (or do not utilize) a feedback channel in order to measure and compensate for any cable delay or drift. For those systems it is assumed, that drifts are slow enough to not violate the defined requirements for the application,

and initial delays will be calibrated via user definable delays or cable length adjustments. The following paragraphs will present some implementations of this type of timing system.

3.3.1 FLASH Event System

The event system installed at the Free Electron Laser in Hamburg (FLASH) at DESY implements an event and clock distribution based timing system.

The EVG provides inputs for its reference clock (approx. 9MHz), a cycle pulse, which is derived from the 50Hz (or 60Hz) mains frequency as well as external trigger inputs. Based on the derived external cycle pulse (usually 10Hz), external triggers and relative to those via firmware defined points in time, the events are generated and encoded into a data stream. The events are 8-bit words defining up to 256 events. For signal transmission and line encoding the Manchester code [20] is used, where the bit rate is based on the provided reference clock. Transmission to the EVR is possible directly (via fan-out transmitter modules with coaxial cables or multi-mode fibers) and via daisy-chaining of modules, where intermediate EVR act also as repeaters.

The EVR provides up to eight TTL level trigger outputs, four gate outputs, three clock outputs and an output of the encoded clock to connect a further EVR. The trigger outputs could be configured via software to generate a pulse based on a received event with an optional configurable delay. Alternatively, always two triggers could be combined in order to serve up to four gate outputs. In this case the user could configure the rise and fall times of the gate by configuring the two related trigger events. The available clock outputs provide the 9MHz reference clock recovered from the data stream, a derived 1MHz clock and a configurable derived clock. The timing system is implemented as industry pack (IP) modules used on VME carriers and special front panels connected via flat ribbon cables. The EVG as well as the EVR are the same IP modules where firmware, software and connected front panels define the used purpose. The fan-out transmitter module is implemented as an VME module. There are also additional modules on VME available in order to provide further trigger delays and fan-out.

The performance of the system in terms of accuracy and stability is limited by three main factors:

- Length uncompensated distribution of the timing signals: from the EVG to the EVRs the distribution is uni-directional via multiple possible fan-out or repeater elements. Length compensation is done only via manual calibration at the EVR. No length changes due to temperature induced drifts or other factors are measured or compensated for automatically.
- Resolution defined by the reference frequency: The period of the reference clock defines the granularity of event and delay configuration. In this case it is 111ns.
- In this implementation all functions are implemented in FPGAs. The phase stability is therefore dominated by the stability of the FPGA.

Newer generations of hardware installed at FLASH showed, that the stability provided by the timing system is not sufficient in order to accurately define system wide trigger events and provide long term phase stable clocks with higher frequencies.

3.3.2 Micro-Research Finland (MRF)

Micro-Research Finland Oy is a company located in Helsinki in Finland, which is specialized in the design and support of a timing system for research facilities. The system implements a classical clock and event distribution solution in order to provide syn-tonized clocks and syn-chronized triggers at various end-stations related to the dedicated master.

The EVG provides inputs for the reference clock, a cycle clock for mains synchronization and up to 12 trigger inputs (cPCI-EVG-300). Based on the trigger inputs and internal configurable tables the events are generated. Also there are 8-bit words used to define up to 256 event numbers. These events are 8b10b [21] encoded and combined with another 8b10b encoded 8-bit word in order to implement a communication channel between the EVG and EVR for general purpose. These 20 bits words define a frame and are sent over a fiber connection to a fan-out module and the EVR. The bit rate is related and locked to the reference clock input. Line bit rates between 1Gb/s and 2.5Gb/s are supported.

The fan-out module splits up the input signal and distributes it to multiple outputs.

The EVR provides outputs for triggers and clock signals (number and type depends on model and options). The module recovers the reference clock (1/20 of the line clock frequency) and decodes the events from the data stream. Furthermore it receives data (e.g. event tables and others) from the EVG over the special communication word in the data frames. Based on received and locally generated events based on the downloaded tables triggers are generated on the outputs. The EVR allows different configurations in terms of type and additional delays. Also clocks can be divided and provided at the output of the board. In a special RF version of the EVR modules, the high speed serial outputs of the FPGA (Xilinx Multi Gigabit Transceivers - MGT) are provided to the front panel in CML level standard, which allows low-jitter and high resolution output patterns and clocks.

Besides the described protocol the company proposed in 2009 plans for a future class to change data transmission concepts [22]. In order to provide higher data throughput for non-event based information, the protocol will be carrying almost always only such data. The events and synchronization are "slipped" into the data. Also ideas about a feedback channel are mentioned for multi-source event distribution [23] and for line delay compensation [24]. Although the multi-event distribution seems to be available, no delay compensation technology is mentioned on the company's web page.

Results of performance measurements of the timing system are provided by the manufacturer in [25]. The resolution at highest possible transmission speed is 8ns for native outputs. This can be improved with either a special delay module, which allows 10ps steps, with the RF option, which allows 400ps (1/20 of 8ns) or the latest cPCI-EVR-300 module, which allows 200ps (1/40 of 8ns). The short term jitter is noted to be less than 25ps rms and even less (15ps rms and 5ps rms) for special versions. But no further details are given about how these numbers are measured.

Micro-Research Finland provides components of the timing system in different hardware standards, namely VME, PCM, cPCI, and CompactRIO. Two of the users of this timing systems are the Linac Coherent Light Source (LCLS)³ and Diamond Lightsource⁴. Related publications are [26] [27] [28].

³at the SLAC National Accelerator Laboratory in Palo Alto, California, USA

⁴at Harwell Science and Innovation Campus in Oxfordshire, UK

Chapter 4

System design

4.1 Basic concept

The most important feature of the Timing System is to provide stable clocks and triggers at numerous locations along the machine in order to synchronize all data acquisition systems as well as control devices. This includes synchronization among all Timing System endpoints as well as in respect to the electron bunches or photon pulses in the machine.

Synchronization related to clocks implies syntonization, that the frequency is exactly the same or is in a fixed relation compared to the bunches in the machine and their defining subsystems like accelerating fields, lasers and so on. Additionally the phase relation between the provided clocks has to be fixed compared to the bunches and the related subsystem clocks.

Synchronization in the context of triggers is related to the fixed phase relation similar to the clocks. Since triggers are transient signals defining a certain point in time, synchronicity therefore implies a system wide absolute timing within the defined accuracy. Triggers play an important role, as they define the sequencing of the complete machine. In order to allow a flexible, easy to configure and full deterministic system, it is important, that the relations of triggers and critical parameters are centrally defined via the timing system.

Investigating the availability of timing system concepts and solutions as described in chapter 3 and taking into account, that the development of the European XFEL timing system started already in 2007 [2], the most suitable concept is an event distribution system. The reasons for that are:

- The system provides a clock, which is syntonized to a frequency of the machine.
- Triggers are generated based on events and they are centrally defined by the event transmitter.

However, no implementation was available at that time, which would have been able to fulfill the requirements for the European XFEL. Therefore the design of a new event distribution system was started.

In order to provide these features, the decentralized clock and trigger providers of the timing system have to be physically connected with the same reference clock, which defines the bunch frequency and phase. Therefore the design concept is based on the following structure:

The reference clock of the machine is provided by the master oscillator [29], which will be located in the injector area at the beginning of the accelerator on the DESY site. This is a crystal and phase locked loop (PLL) based oscillator providing a 1.3GHz clock output. This clock is connected to the transmitting part of the Timing System, which is located close to the master oscillator. In the Timing System transmitter, a data bit stream is generated, which carries

events and other information. The bit rate is defined as 1.3Gbps¹ and therefore compatible to the 1.3GHz reference clock. The bit stream will be synchronized and synchronized in phase with the reference clock. This data stream will be transmitted to the Timing System receivers located close to the data acquisition and control device subsystems along the machine. On the receiver side, the clock of the data stream will be recovered and the events and further information decoded. Based on the recovered clock further frequencies can be derived by dividing and/or multiplication. The decoded events information will be used to generate actual triggers on the timing receiver output.

4.2 Implementation of the basic concept

The previously described basic concept can be implemented as shown in the simplified diagram in Figure 4.1. The task of the timing system transmitter module is to generate the timing sys-

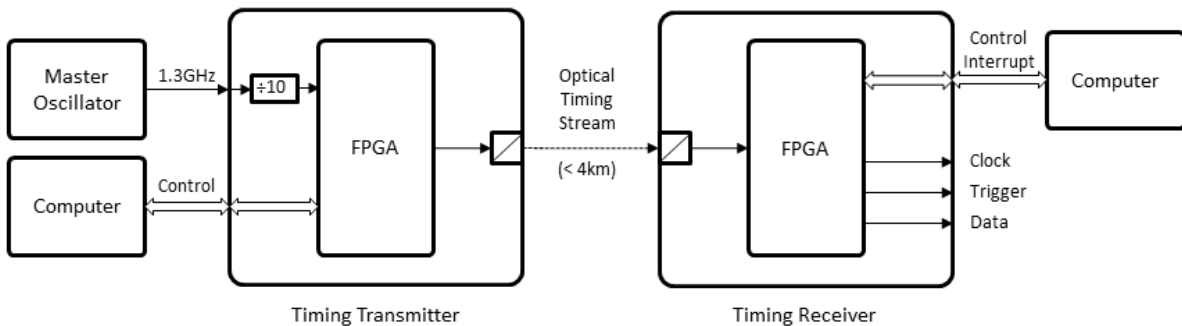


Figure 4.1: Simplified block diagram illustrating the basic concept of the timing system. The Timing Transmitter will generate an optical timing data stream based on configuration information from a computer and the reference clock provided by the master oscillator. This timing stream carries the frequency, phase and further information to generate triggers and provide machine related parameters to the consumers of the Timing Receiver. A computer is used to configure the Timing Receiver and to receive synchronized interrupt signals and other information.

tem data stream synchronized to the the reference frequency and including the information to generate triggers and recovering further information at the receiver side. In order to generate the data stream, the transmitter is connected to the reference clock of 1.3GHz of the master oscillator. The component generating the serial data stream is a serializer module within the FPGA. This component will not accept the 1.3GHz directly, but requires a tenth of the frequency, as it internally uses a PLL to generate the bit rate clock locally. Therefore the 1.3GHz reference has to be divided before entering the FPGA dedicated reference clock input.

Most of the data generated and transmitted to the receiver will be defined by the operators of the machine. Therefore a communication interface between the FPGA and the user has to be implemented as shown in the diagram.

Based on this setup the data stream is generated and converted to an optical signal and transmitted down to multiple timing system receivers (only one is shown in the diagram) which might be up to 4km far away. In the timing system receiver the optical signal is converted back into an electrical signal and applied to the de-serializer input of the FPGA. Internally the 1.3GHz is recovered and provided as a tenth of the frequency to the FPGA user logic. This clock can be provided directly or further divided and/or multiplied with internal PLLs or clock managers to the outside of the FPGA. Those signals are then provided to the output of the

¹Giga-bits per second

module for external consumers.

The triggers and other deterministic data received from the transmitter is decoded in the FPGA and based on that triggers generated. These triggers are available to the outputs of the module as shown in Figure 4.1.

Finally, the deterministic information received can be provided to a connected system on the same channel as it is provided for configuration of the receiver module. Low-latency synchronization is achieved with interrupts or a comparable technology.

4.3 Influences on phase stability

Although the proposed implementation is able to fulfill all functional requirements of the timing system for the European XFEL, the performance of the system is limited and is analyzed briefly in the following sections.

4.3.1 Temperature induced drift

Almost all components involved in the complete system provide a certain delay in the signal flow, which depends partly on the temperature of the component. The component with the most significant influence in terms of drift is the optical fiber used to transfer the data stream from the transmitter to the receiver. Ordinary single mode fibers commonly used in telecommunication applications (9 μ m core diameter and 125 μ m cladding) introduce approximately a phase shift of 30ps/K/km [30]. For each centigrade temperature change the phase of the transmitted signal shifts by 30ps for each Kilometer length of the fiber. If we assume the longest connection for the European XFEL to be 4km and the defined temperature stability in the tunnels to be 10 centigrade peak-to-peak [1], it results in a possible phase drift of 1.2ns. This is far beyond the defined stability requirements. In order to remove (or at least significantly reduce) the influence of the temperature induced drift on the fibers, either passive stabilization through special phase stabilized fibers (PSOF)² have to be used or a drift compensation scheme has to be implemented.

Besides fibers other components like PCB copper traces, RF cables and integrated circuits produce temperature induced phase drifts. As a consequence the following implementation policies are derived:

- Only RF cables will be used, which are as phase stable as possible, in cases where it matters (e.g. between master oscillator and timing transmitter)
- Use of PCB materials with less temperature dependence (e.g. ceramics) should be considered
- Keep traces short
- Select components which have minimal influence on temperature
- Keep temperature on the modules as stable as possible

Besides the direct influence of the temperature variations on components there is also drift effect of integrated circuits due to supply voltage fluctuations. The voltage fluctuations might be caused by temperature fluctuations on the voltage converters or supply part of the system or by other effects. In general care has to be taken to implement a stable voltage supply for the

²PSOF use a special coating on the fibers, which compensate temperature induced time delays by mechanically manipulating the fiber. A datasheet of a PSOF by Furukawa Electric Co Ltd [31] showed, that the temperature induced drift can be reduced to approximately 1ps/K/km

modules. Additionally critical signals should not be routed through drift sensitive components like the FPGA.

4.3.2 Jitter

In the proposed design different components add jitter to the signal passing them. Starting from the divider, the transmitting FPGA, the optical transmitter and receiver and finally the FPGA on the receiver side are contributing. In contrast to drifts, jitter can partially be reduced by means of PLLs with low-bandwidth loop filters [32] [33]. Those will filter high-frequency phase fluctuations and therefore act as a jitter cleaner. However, all jitter sources should be reduced or removed from the signal path, where it matters.

4.3.3 Electro Magnetic Interference (EMI)

Electro magnetic interference includes all external electro-magnetic sources influencing the integrity, performance, quality, and finally the phase stability of the signals, but it also includes how the timing system influences other systems through electro-magnetic effects. Both effects have to be taken into account, investigated and minimized.

Starting with the effect on the timing system (and also other connected systems), the 50Hz frequency of the power line provides a high risk of influencing the timing system and even more sensitive ADC systems. As the timing system will define almost all timing aspects of the machine, it also provides a way of minimizing effects based on the AC phase. The timing system should synchronize its timing bases to the power line in order to avoid unpredictable effects.

Especially long distance connections as required for a large scale timing distribution system increase the risk of picking up interferences from neighboring cables on cable trays, crossing rooms and ground references, passing through or generating ground loops, etc. Due to that and also due to the much lower attenuation of long distant connection, optical fibers were chosen to build up the distribution infrastructure. This also removes the EMI influence on surrounding cables and other systems.

Other EMI sources and more important, reducing their influence, depend on the final technical solution in terms of chosen form factors, surrounded systems and so on. They include filtering of power supplies, shielding, splitting, connection and routing of ground and supply traces and layers on the PCB design, etc.

4.3.4 Conversion of amplitude into phase variations

One effect, which is also involved in previously discussed sections, is the fact, that amplitude variations can convert into phase variations. This effect is also known as AM-to-PM conversion (amplitude modulation to phase modulation).

Usually amplitude variations are uncritical in digital systems, which is concerned about time accuracy like the discussed timing system. However, if an amplitude variation can transform into a phase variation, it can be critical. Those conversions can take place everywhere, where a point in time is defined based on a threshold detection of an incoming signal. Such detection principle is common in digital systems and will also be used in the timing system. Optical receivers at the end of the optical link are based on diodes and a following threshold detection to define a received bit. Also clock buffers and dividers as well as FPGAs will define a logical input value based on the very same principle.

There are different effects to reduce the resulting phase variations. One is to reduce the main source of this effect: reduce amplitude variations. Good candidates are to stabilize and filter the supply voltages and other EMI effects as well as to reduce temperature variations, as

those are the assumed dominant causes of amplitude variations. Another way is to reduce the conversion effect by using differential signal standards. In comparison to single ended signals, where a logic level is defined by an absolute threshold, logic levels of differential signals are defined by the relative polarity change of the two input lines. This has the advantage, that common mode effects (e.g. EMI influencing both lines in the same way) are rejected (or at least reduced). Also scaling of the input amplitudes are not influencing the phase, as long as they are symmetric.

4.4 Influences on accuracy

In contrast to the previously discussed phase stability of the signals, this section will analyze influences on the accuracy of the delivered signals in terms of possible resolution, reproducible phase relations and uncertainties and their sources and dependencies.

4.4.1 Resolution

The resolution of the timing system, as defined in chapter 2, is limited by the distributed system time base and the procedure of event distribution and synchronization. Two basic types of event distribution systems can be defined: a direct event system and a scheduled event system.

The direct event distribution system follows the principle, that events are transmitted to the receiver modules, as they occur and are converted into a trigger. The time, when a trigger is going to be transmitted might be known to the transmitter in advance or it can be signalled by an external device on an input. For direct events the resolution is defined by the time steps, when a new event can be transmitted to the receiver. The data stream generated by the transmitter will serialize the events and all other data to be transmitted to the receiver. Usually it follows a fixed pattern of how many parallel data bits are serialized. If no data is to be sent, a special idle word will be transmitted as a place holder. Therefore the smallest time step to transmit a new event is defined by the word rate (not the serial bit rate). For the presented basic design implementation a direct event would have a best possible resolution of a tenth of the 1.3GHz bit rate (if a ten bit encoding is used and no additional protocol overhead is assumed and no other event or data to be sent is blocking the transmitter) which results in a resolution of 7.69ns.

Scheduled events, however, work in a different way. Those events are known in advance and therefore can be transmitted before they actually happen together with a time information, when they will become valid. With a synchronized time base between the transmitter and all receivers, the receivers can generate the event internally with an accuracy of the time base. In this case the resolution is limited to the resolution of the common time base, which is 1.3GHz (769ps) for the presented base design.

4.4.2 Reproducible phase relations

As discussed in chapter 2 it is often required to fine-adjust the phase of a clock provided by the timing system and used for digitizing a signal originating from the machine in order to sample the correct points (phase) of the signal. Even if all the stability requirements are met, a different phase relation after a power cut would require to again adjust the phase of all the clocks, that need certain phase relations. This is time consuming and can cause a significant down time of the whole machine. Therefore all provided clocks and triggers have to come up with the same phase relation (to the machine and to other clocks) after a power cut or any other interruption of operation.

In order to guarantee reproducible phase relations, different design aspects have to be implemented:

- The link delay has to be constant or known in order to compensate for delay changes.
- Resetting and alignment of complex components
- Clocks derived from the reference at the receiver side have to be synchronized.
- All systems at European XFEL which require synchronization have to be synchronized to the timing system.

The first point is already violated by the uncompensated drift of the optical link between transmitter and receiver. Even if no cable in the connection is changed, the delay is not constant, as the temperature variations will produce an uncertainty of the phase as discussed earlier. The delay variation have to be either reduced to levels below the required stability (special fibers and/or temperature stabilization) or the drift has to be compensated either actively or via compensation at the receiver side with measured link delay changes (or both). A link delay measurement would additionally allow to compensate even cable changes, as the complete link delay is known and can be considered in the phase and time calculations.

Serializer, de-serializer and clock and data recovery components are complex elements and implement many layers of functionalities and synchronization. However, after a startup, restart or reconnection the internal states are not defined in a deterministic way. Internal phase lockings and buffer alignments are at uncertain conditions. In order to generate a deterministic and reliable behavior, these components have to be reset into a defined state. Additionally manual alignment procedures have to be implemented in order to detect correct phase alignment. Those procedures have to be implemented in the timing system design.

Different clock frequencies (derived from the recovered reference) at the receiver are required. They will be generated by dividing and/or multiplying the reference frequency. Frequency dividers, as implemented in the timing system design, are usually based on counters. When started, the counter will count with its input frequency until a certain value is reached, then the binary output value is toggled. In this way lower frequencies can be generated, which are defined by an integer divider. The point in time, when the dividing process is started, as well as the initial counter value and the state of the clock output define the phase relation between the output and input clock. When multiple clocks are generated in this way, the phase relation between the output clocks will depend on the divider processes. Multiplication, on the other hand, does not generate any ambiguities and is therefore only problematic, if it is followed after a divider (like in common PLLs). In order to solve the divider problem, there are at least three ways possible: (1) avoid dividing, (2) synchronizing the dividers or (3) measure the phase and adjust it. Although the first solution is not realistic in general, it is still wise to avoid dividing of clocks whenever possible. One example would be to chose the VCO frequency of an PLL to be the output frequency and not a multiple of it, which would require another divider. This is for sure a performance trade off to be made. Where dividers have to be used, they have to be synchronized. As the timing system has to synchronize the common time base in any case, it can use this time information in order to start or re-synchronize the divider process (either internally in the FPGA or by external divider integrated circuits). A limiting factor in this procedure is defined by the current technologies. The maximum rise time to be generated accurately by an FPGA to synchronize a divider as well as the maximum speed of resetting such divider is limited. Finally it is possible to measure and compare the output phase of a divider or PLL with a phase of another clock or trigger. This procedure works as a phase discriminator where the output phase has to be adjusted, if a mismatch is detected. Implementations of this approach can be realized via dedicated integrated circuits or within

the FPGA (within certain frequency limitations), but this requires more complexity than the divider synchronization approach (e.g. dedicated ICs and splitting or fanout of clock outputs to measure them).

Coming now to the final aspect pointed out earlier in order to guarantee a reproducible phase relation: a phase relation depends at least on phases of two signals. For the results it does not matter, if the phase of a clock from the timing system is changing or if the phase of a signal coming from the machine, which has to be measured, was changing. Both have to have the previous phase relation. The timing system has to guarantee, that the phase relations are reproducible system wide, but in the same way the signals coming from the machine have to do the same. As described above, deriving frequencies from the reference will generate ambiguities. The timing system has to synchronize all of its clocks and triggers based on an internal or external 10Hz cycle. But if other systems generate clocks by dividing the master oscillator clock directly or just locking a lower frequency to it, the phase relation can not be guaranteed after a power cut unless it is synchronized with the timing system. Therefore all systems which require synchronization to the reference have to synchronize their phases with the timing system in order to provide reproducible phases after an unsynchronized startup phase.

4.5 Consequences for the system design

Based on the analysis of stability and accuracy above, different important consequences can be derived which are reflected in Figure 4.2 and will be described in the following paragraphs.

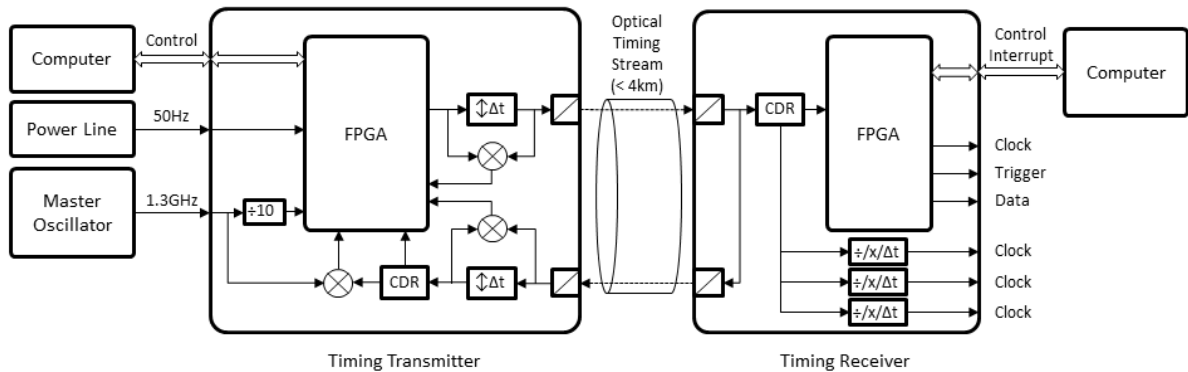


Figure 4.2: Simplified block diagram illustrating the most important features added due to the discussed limitations of the basic concept design. The features include: synchronization to the 50Hz power line zero crossing, implementation of the drift compensation, introduction of a second fiber to implement a feedback channel and a dedicated clock distribution section on the receiver side. CDR denotes Clock and Data Recovery circuits, $\updownarrow \Delta t$ is an adjustable time delay circuit, \otimes represents phase comparator circuits and $\div/x/\Delta t$ represent clock buffers with frequency dividing and multiplication as well as adjustable time delay function.

4.5.1 Synchronization to power line frequency

In order to reduce the influence of the power line 50Hz oscillations the timing system will be synchronized to the zero crossings of the voltage. The resulting 50Hz pulses will be applied to an input of the timing transmitter module. Based on these pulses the desired 10Hz (25Hz as a later upgrade option for the European XFEL) will be derived and used to synchronize the

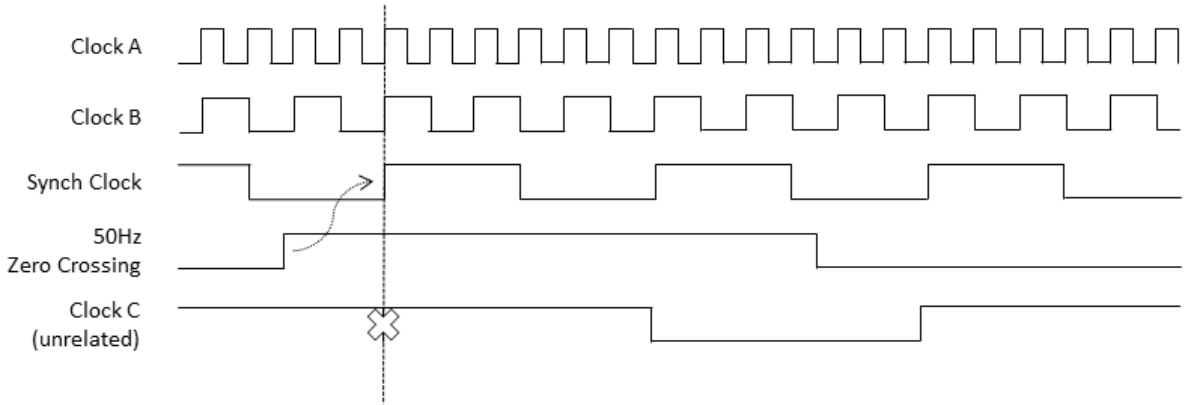


Figure 4.3: This time diagram demonstrates the divider and clock synchronization principle. Bases for the synchronization are a relatively low frequency synch clock derived by dividing the 1.3GHz (in the design a frequency of about 100kHz is planned) and a signal defining the zero crossing of the 50Hz power line frequency. The synchronization of the clock phases (reset of the dividers) is applied with the next rising edge of the synch clock following the rising edge of the 50Hz zero crossing (highlighted by the arrow). Clocks A and B are integer multiples of the synch clock frequency and therefore will not experience any phase jumps once they are in phase with the synch clock. However, the depicted clock C is a subharmonic of the synch clock and a phase jump is possible, as shown.

timing system.

As the zero crossing of the power line will define the synchronization of all distributed dividers, some boundary conditions have to be defined. Synchronizing a divider means, that the rising edge of all output clocks (independent of divider value) are aligned at a certain point in time every 10Hz. As phase jumps in clocks are undesirable, as they can cause problems in systems like FPGAs, PLLs, ADC and so on, they have to be avoided. This can be realized, if the 50Hz (or the divided 10Hz) will be resynchronized with a frequency derived from the 1.3GHz reference. In this case this chosen clock will run continuously without phase jumps. This also holds for clocks being a multiple of the chosen clock, as their rising edges will always be aligned. For lower frequencies, however, this is not (necessarily) true. If the rising edges do not overlap, the divider synchronization will generate a phase jump. These relations are depicted in Figure 4.3. The mathematical relation and valid frequencies are defined by the absolute time between two synchronization pulses. As the synchronization is aligned by the synch clock, the time between two synchronization points is always defined by

$$\Delta t_{synch} = N \times \frac{1}{f_{synclk}} \quad (4.1)$$

$$N \in \mathbb{N}^*$$

Therefore all other frequencies, where the time defined by equation (4.1) for any chosen N results in M complete integer periods, thus

$$M \times \frac{1}{f_{userclk}} = N \times \frac{1}{f_{synclk}} \quad (4.2)$$

$$N, M \in \mathbb{N}^*$$

no phase jumps will be observed. In this case the number of user clock periods is defined by

$$M = N \times \frac{f_{userclk}}{f_{synchclk}}. \quad (4.3)$$

$$N, M \in \mathbb{N}^*$$

As a consequence all frequencies being a multiple of the synchronization clock fall into that category by definition. Therefore defining the derived frequency to resynchronize the 10Hz signal will define the lowest frequency, which never produces phase jumps. For the European XFEL timing system, this frequency is defined to be $\frac{1.3GHz}{12960} \approx 100kHz$. In this case all important frequencies generated by dividing the 1.3GHz clock as defined in Table 1.2 will never experience any phase jumps.

4.5.2 Optical transmission line and interfaces

As main distribution medium optical fibers with optical transceivers will be used. It has been chosen instead of copper cable distribution because fibers provide

- less signal attenuation (less than 1dB/km³ compared to roughly 400dB/km for coaxial RG58⁴)
- an optimal EMI behavior
- a higher density of cables (less space required in the tunnels and on cable trays)
- lower costs

However, there are also some disadvantages. Besides the obvious higher risk to break the fiber cable than a copper cable, the temperature induced drift is stronger than on copper cables. Furthermore the signal will experience a chromatic dispersion due to wavelength dependent phase delay as well as higher order of dispersion.

4.5.3 Drift compensation scheme

One important aspect of the timing system is the drift compensation scheme in order to significantly reduce any delay variations of the transmission channel between the transmitter and a receiver module. The timing data stream, as generated by the FPGA and phase locked to the reference clock of the master oscillator is shown as the right hand side output of the FPGA in the Timing Transmitter in Figure 4.2 and the 1.3GHz reference is connected on the left side. The data stream will pass a variable delay element, which is one of the actuators in the drift compensation scheme. Afterwards it passes the electro-optical transmitter, is transmitted on an optical fiber and then converted back to an electrical signal on the receiver side. The connected clock and data recovery (CDR) circuit extracts the clock information as well as provides the jitter cleaned data stream output in-phase to the input signal. The extracted clock is used as local reference to generate further frequencies and provide them to local consumers. The data stream will be decoded in the FPGA in order to generate trigger, synchronize dividers (see below) and make use of other deterministic information. Additionally the data stream will be sent back to the transmitter. Therefore it passes again an electro-optical transmitter, an optical fiber, optical-to-electrical receiver, another delay circuit used as an actuator and a CDR circuit. The important part of the drift compensation is now to compare the provided

³e.g. 0.38dB/km at 1310nm for a single mode fiber E9/125 from LEONI conform to ITU-T Rec. G.652 [34] and IEC60793-2-50, as used for the European XFEL

⁴e.g. LL1030AF-PUR improved RG58 from elspec provides attenuation of 326dB/km at 1GHz

clock phase of the CDR output to the reference provided by the master oscillator and use the delay actuator in order to keep the phase constant. The effect of the time delay and phase at the receiver side can be derived in the following way:

Referring to the individual delays related to the signal path shown in Figure 4.2, the accumulated delay of the loop from the FPGA of the transmitter via the receiver and back to the phase comparator on the transmitter is defined by (see List of Symbols at the end of the document)

$$T_{Loop} = t_{D_1} + t_{EO_T} + t_{F_{TR}} + t_{OE_R} + t_{CDR_R} + t_{EO_R} + t_{F_{RT}} + t_{OE_T} + t_{D_2} + t_{CDR_T} \quad (4.4)$$

Based on the assumption, that the two delay elements are configured in a way, that they provide the same delay

$$t_{D_1} = t_{D_2} = t_D$$

and further assuming, that the delays of the components with the same functionality are equal

$$t_{EO_T} = t_{EO_R} = t_{EO},$$

$$t_{OE_T} = t_{OE_R} = t_{OE},$$

$$t_{CDR_T} = t_{CDR_R} = t_{CDR},$$

$$t_{F_{TR}} = t_{F_{RT}} = t_F$$

equation (4.4) can be written as

$$T_{Loop} = 2t_D + 2t_{EO} + 2t_F + 2t_{OE} + 2t_{CDR}. \quad (4.5)$$

The delay from the transmitter to the receiver is then defined as

$$T_{Receiver} = t_D + t_{EO} + t_F + t_{OE} + t_{CDR} = \frac{1}{2}T_{Loop}. \quad (4.6)$$

In case of the drift compensation, the initial delay is not of importance - it will be discussed in more detail in the next section. The drift compensation has to ensure, that any delay variation is compensated. Therefore we rewrite equations (4.5) and (4.6) into

$$dT_{Receiver} = dt_D + dt_{EO} + dt_F + dt_{OE} + dt_{CDR} \quad (4.7)$$

$$dT_{Loop} = 2dt_D + 2dt_{EO} + 2dt_F + 2dt_{OE} + 2dt_{CDR}. \quad (4.8)$$

As the phase comparator, as implemented on the transmitter side, is able to detect any change of the phase received from the receiver side relative to the reference, a connected controller is able to adjust the two actuator delays in a way, that the phase change is compensated. As stated earlier, if the two delays are changed in a way, that their delays are always the same, equation (4.8) yields

$$dT_{Loop} = 0 = 2dt_D + 2dt_{EO} + 2dt_F + 2dt_{OE} + 2dt_{CDR}$$

$$\Rightarrow 2dt_D = -2dt_{EO} - 2dt_F - 2dt_{OE} - 2dt_{CDR}$$

$$\Leftrightarrow dt_D = -dt_{EO} - dt_F - dt_{OE} - dt_{CDR}$$

Inserted into equation (4.7)

$$dT_{Receiver} = -dt_{EO} - dt_F - dt_{OE} - dt_{CDR} + dt_{EO} + dt_F + dt_{OE} + dt_{CDR} = 0 \quad (4.9)$$

shows, that the delay change, as induced by drift, is removed.

It should be emphasized, that this only holds, if $dt_{Receiver}$ is equal to $\frac{1}{2}dt_{Loop}$. That means, that the two paths (transmitter to receiver and receiver to transmitter) are symmetric. Basis for that was the assumption of equal delay behavior for components with same functionality. This assumption has to be proven to be valid within the requirements by measurements in a later chapter. At this point only two further improvements of components will be discussed which are most likely to generate the strongest non-equal drift behavior: the delay components and the two optical fibers.

As the delay elements are designed to generate a variable delay, it is quite likely, that both elements generate slightly different delays. In order to monitor and finally compensate for non-symmetric effects, each element is connected to additional phase comparators as shown in Figure 4.2. Both phase comparators will measure the phase difference between the input and the output of the delay elements. If the delays are equal, the resulting phase differences will be equal as well. This can be checked in the controller implemented on the transmitter board. If there is a difference detected, it can be used to adjust the individual control signals to the delay adjustments implementing another control loop.

For the signal transmission two fibers are planned (one for transmitter to receiver and one for the way back). This is mostly influenced by commercially available technologies and the goal to minimize the costs. However, if two fibers are used, there is a risk, that the initial delay as well as the more important drift behavior differs between the fibers. Relatively old investigations published in [35] and [30] showed, that fibers in a commercially available multi fiber cable showed different signal propagation delays for the individual fibers. It shows, that full symmetry can not be assumed. As a consequence, especially for long distance connections $>2\text{km}$ a different approach can be chosen: to avoid unequal behavior of two fibers, a single fiber should be used instead. There are different principles to achieve that, which will be discussed briefly:

- Using two different wavelengths for both directions
- Using different polarizations for both directions
- Time multiplexing of the transmissions on the fiber
- Using circulators to add and drop the signals at both ends

Besides the commonly used optical transceivers providing sockets for two fibers, there are also transceivers available, which provide a connection for only one fiber. Internally the module will transmit the signal at one optical wavelength while it receives a signal at another one. On the other side a different transceiver is required, which does the same approach with swapped wavelength. This would solve the problem about using two fibers. However, this solution will not work for the proposed implementation, as the group delay of an optical fiber depends on the center wavelength (due to the refractive index dependence) and is systematically different for both wavelengths [36]. Therefore the drift would be different for the two directions and the resulting $dt_{Receiver}$ value is not zero.

Using different polarizations for the forward and backward channel would require special optical components to split and combine the different polarizations, and polarization maintaining fibers (PMF). However, those fibers show different group delays for the two polarization axis and therefore are not usable for the desired application. Time division multiplexing would also not provide a satisfactory result, as the transmission from transmitter to receiver would have to be stopped for some time to allow the transmission in the opposite direction, which would violate the closed loop approach and additionally would stop the recovered clock at the receiver side for some time, which is certainly not desirable.

Therefore the most suitable approach is to use circulators at both ends as depicted in Figure 4.4. The optical output on the transmitter side is inserted into an optical circulator. This signal exits

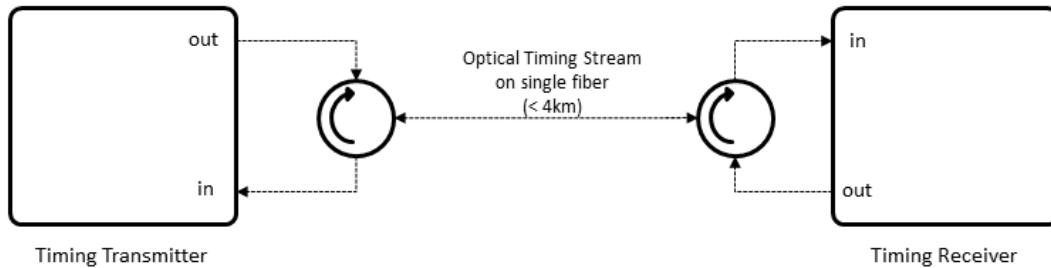


Figure 4.4: This simplified block diagram illustrates the use of two circulators in order to implement a full-duplex long-distance connection on a single fiber. As only a single fiber is used, the production and temperature induced propagation delay is identical for both directions. The implementation uses two optical circulators in order to add and drop signals on both ends. The timing stream from the Timing Transmitter output is entering the first circulator and will leave it at the next clockwise output on the long-distance fiber connection. It will enter the second circulator and leaves it on the upper output to be connected to the input of the Timing Receiver. The output of the receiver will enter the right circulator from the bottom side and leaves it on the left to be transmitted back to the Timing Transmitter. It enters the left circulator on the right side and leaves it on the bottom output and is then connected to the input of the Timing Transmitter and closes the loop. The connections between the input and outputs of the the two sides and the circulators have to be kept at equal lengths and as short as possible.

the circulator at the next port clockwise, is then transmitted over the long distance connection to the receiver side. There it enters a second circulator and leaves the next port clockwise and ends at the receiver's optical input. The receiver's output will enter the circulator on the remaining input and leave it on the next port clockwise to finally reach the transmitter's circulator. There it leaves the port to the input of the transmitter side. In this setup it is important, that the connections between the transmitter and circulator do have the same length and should be short. The same holds for the receiver side.

4.5.4 Link length determination

In order to be able to implement a system wide synchronization, it is required to determine the link delays between the transmitter(s) and the receivers. The accuracy of the determination defines best possible accuracy of synchronization in terms of resolution. The design aim in terms of resolution is one nano second. As the period of the reference clock, as well as the time for one bit of the data stream is $\frac{1}{1.3GHz} \approx 769ps$, resolving the link delay in reference clock cycles is sufficient. As the drift compensation ensures a fixed link delay in full reference clock periods, the delay measurement can be implemented in the following way: A special synchronization word is send by the transmitter FPGA. At that time an internal counter is started counting at a rate of parallel data word transmission (a tenth of the reference clock). When the word is received by the same FPGA after it passed all steps as shown in Figure 4.2, the counter is stopped. Its value represents the number of full word cycles required for the data to pass the full loop.

In general the data word does not have to arrive on the receiving input of the FPGA synchronized to the word boundaries of the transmitter output. Only alignment on bit level is ensured by the drift compensation. To reach the ten times better accuracy, as it is required,

the remaining reference clock period steps are determined by counting the bit offset in the data word received.

Based on the measured number of n whole words and m bits within the last word, the link delay between the transmitter and the receiver sides is calculated by

$$\begin{aligned}\Delta t_{Link} &= \frac{1}{2} \left(n \cdot \frac{10}{1.3GHz} + m \cdot \frac{1}{1.3GHz} \right) \\ \iff \Delta t_{Link} &= \left(5n + \frac{1}{2}m \right) \frac{1}{1.3GHz}\end{aligned}\quad (4.10)$$

The link delay correction will be implemented on the receiver side with a resolution of 1.3GHz periods. Therefore link delays have to be avoided, which are not a multiple of that period, as this can not be corrected on the receiver side. In order to ensure that, condition m in equation (4.10) has to be a multiple of 2. If the measured m is not fulfilling that requirement, both delay elements have to be adjusted equally to correct that.

4.5.5 Dedicated low-jitter clock section

In order to avoid routing of sensitive clock signals through the receiving FPGA and therefore reducing drift and jitter influence, the clock section will be implemented outside of the FPGA on the receiver side. This clock section includes

- Clock recovery from the timing data stream
- Clock distribution and dividers
- PLLs
- Optional resynchronization Flip-Flops for FPGA signals

A clock and data recovery integrated circuit is inserted in the path between the optical receiver and the FPGA. This chip provides the recovered 1.3GHz clock from the data stream. This component will use an internal PLL which is synchronized to the data stream clock. A configurable loop filter will be optimized to reduce short term jitter while maintaining phase synchronization in longer terms to stay synchronized with the transmitter. The output is connected to the following configurable clock divider and buffer. If dividers are enabled, they are synchronized by the FPGA (see below). The resulting clock can be multiplied by a PLL or directly provided to the outputs of the receiver. The previously described boundary conditions for using a PLL have to be met in order to ensure a correct synchronized output.

4.5.6 Differential Signalling Standard

In order to reduce common mode effects on signal lines, which can influence the phase stability (as described previously) all signals for clocks, trigger and data are implemented in differential standards like LVDS, LVPECL and CML. Differential standards are also chosen for the outputs of the receiver module. The same is true for the inputs of the transmitter module, except for the reference clock input, which will be a single ended SMA input, compatible to the output of the master oscillator.

4.5.7 Receiver-side clock and trigger synchronization

As mentioned above, external dividers, but also the logic to generate clocks and triggers within the FPGA have to be synchronized to the transmitter. The general implementation of the synchronization depends on the following steps:

- Recovery of the timing data stream word clock (a tens of the 1.3GHz) within the receiver FPGA
- Detecting the bit offset within the received data words relative to the real word boundaries (to determine the resolution on the 1.3GHz level)
- Resetting an internal counter if a synchronization word has been received (this will be used to delay the divider synchronization)
- Decoding the measured link delay compensation sent by the transmitter
- Resetting an internal counter if a synchronization signal is produced (this will be the internal time reference)
- Decode and process information about events and other parameters sent by the transmitter
- Calculate and generate clocks and triggers based on data, internal synchronized counter, link delay and bit offset

The deserializer component within the receiver FPGA will recover the clock of the data stream as well as converting the data into parallel data words. The clock will be divided to the word rate (a tenth of the 1.3GHz) and provided to the logic.

In general, the phase of the word clock can be at an arbitrary bit boundary of the received data (as discussed in the link delay measurement section above). Therefore the real boundary of the received data can be determined by finding the bit offset within the parallel data words provided by the deserializer. For all calculations and output signals, this additional offset has to be considered.

The next step is to detect the special synchronization word within the data stream. If it has been detected, an internal counter running with the recovered 130MHz will be reset. This counter is required to implement a delay used in the following way:

Within the data stream a link delay packet is periodically distributed and will be received and decoded. This will include the time delay to apply in order to compensate for the absolute link delay. This is required, in order to align all receivers to the same phase. A maximum link delay can be set as measured or externally defined. From this value the actual link delay will be subtracted and the delay then added to the point in time, when the synchronization word has been received. Using the counter together with a high speed serializer, a synchronization signal for the external dividers can be generated with a resolution of 1.3GHz (or even 2.6GHz). One divided clock of the external dividers will be provided to the FPGA to drive the internal logic for the next steps.

A further counter, running with the synchronized clock from the external divider, will be reset with the synchronization signal as well. This counter provides the FPGA internal synchronized time reference.

In order to generate triggers, the event and additional information have to be received and decoded from the data stream. The received events provide a time stamp, when the actual event becomes valid. To determine this time, the counter will be used. The actual time to generate a certain trigger based on an event can even be further delayed in order to phase adjust the trigger to the actual hardware. Finally a serializer will be used on the FPGA output in order to provide a higher resolution in time.

The concrete implementation depends on the chosen hardware. Therefore a more detailed description of the this procedure is presented in a later chapter.

4.6 Selection of components

The remaining part of this chapter provides an overview of the survey and selection process of suitable components, which will allow the implementation of the timing system. The concrete investigation of the selected circuits as well as the system implementation will be discussed in later chapters.

4.6.1 Clock and Data Recovery (CDR)

The task of the clock and data recovery circuit is to recover the clock from the data stream and provide jitter cleaned data bits. This component will be used in the FPGAs in order to receive, decode and synchronize based on the data stream on the receiver as well as on the transmitter side. Additionally a dedicated circuit is required for the receiver as well as for each transmitter path (see section 4.5.3 above).

The selection of the CDR within the FPGA will be discussed in the next section. During the investigation of the market of CDR integrated circuits for the drift compensation and low-jitter clock distribution scheme different types were found. Those can be seen as two classes: circuits including the deserialization of the data and those which do not deserialize the data. This first class provides the data as parallel bits at the output. The provided clock is recovered from the data stream, but then divided in order to be synchronized to the parallel data (in our case 130MHz). The other class does not deserialize the data. Instead, it uses the recovered data clock (in our case 1.3GHz) and resynchronizes the data bits to it before they are provided on the data output.

For the XFEL timing system the second class is the most suitable. The reasons for that are:

- The phase comparison for the drift compensation is based on 1.3GHz. Using 130MHz would reduce the resolution and accuracy.
- All typical frequencies required on the receiver side can be generated with dividers from 1.3GHz. This is not the case for 130MHz.
- The phase of the recovered 1.3GHz is fixed related to the data stream bit boundaries, which is a must for the timing system. All circuits of the first class, which were found, did not guarantee a reproducible phase relation between the 130MHz and the phase of the word boundaries in the data stream.

Investigating the last point, it turned out, that for the deserializing CDR circuits, there are two types of devices available. One, which does not care about the actual data bits and just starts to deserialize groups of eight or ten bits to parallel words. In this case the recovered and divided clock is perhaps in phase to one data bit at the input. But the phase related to the actual data word boundaries as sent by the transmitter is never guaranteed and the 130MHz therefore almost useless. The other type does care about the data bits and implements a 8b10 decoder as well as word alignment logic. This will guarantee correctly aligned data words. However, due to the internal alignment logic (fifos and phase shifting) the phase of the output clock is again not related to the phase of the data words at the input, but depends on the startup conditions.

Out of investigated integrated circuits from Silicon Labs⁵ (Si5018,Si5020,Si5023), Analog Devices⁶ (ADN2812, ADN2817/ADN2818,ADN2865) and Micrel⁷ (SY87721), the most suitable turned out to be the ADN2812 from Analog Devices. It had the lowest additive jitter with less

⁵<http://www.silabs.com> (data sheets for the mentioned circuits are available on that site)

⁶<http://www.analog.com> (data sheets for the mentioned circuits are available on that site)

⁷<http://www.micrel.com> (data sheets for the mentioned circuits are available on that site)

than one picosecond (defined as 0.001 UIrms in [37]) with a small footprint, no requirement of external reference clock and externally definable PLL loop filter and low complexity. A block diagram is shown in Figure 4.5.

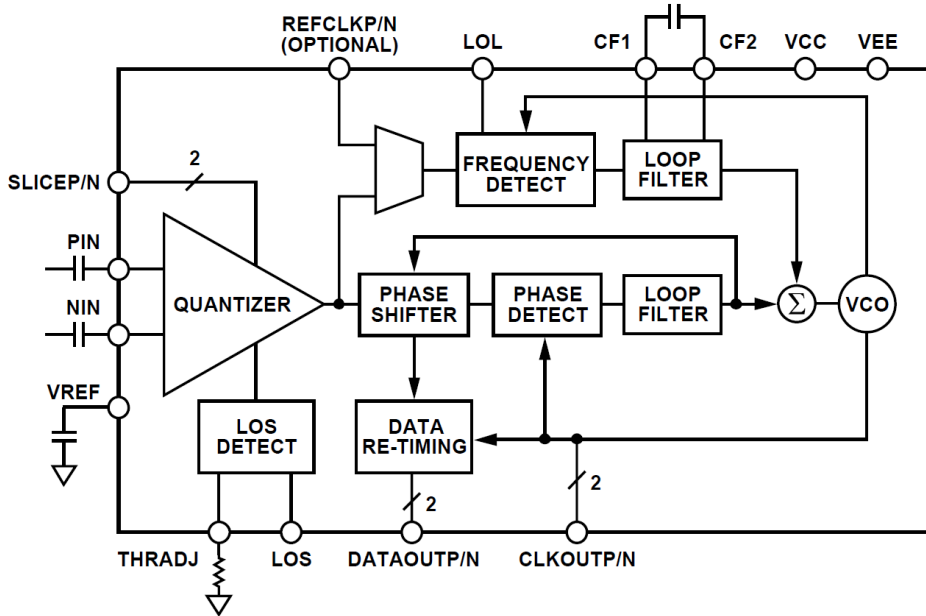


Figure 4.5: Functional block diagram of the chosen clock and data recovery integrated circuit ADN2812 from Analog Devices as taken from the datasheet [37]

4.6.2 Field Programmable Gate Array (FPGA)

The core component on the transmitter as well as on the transceiver side will be a Field Programmable Gate Array (FPGA). It has to implement many functions in order to allow communication with external systems, generate the timing stream, calculate link length, controlling the drift compensation, decoding of data stream, trigger generation and many more. Important aspects of the device are availability of high speed serial interfaces (for timing data streaming, communication with external systems like PCI Express), low jitter and drift behavior.

Looking at the two main manufacturers of such devices (Altera⁸ and Xilinx⁹), both seem to provide devices suitable for the timing system. However, Xilinx was chosen for the implementation. The main reason is, that in-house experience with FPGAs from Xilinx in terms of programming and hardware design exists and compatibility, reusing of firmware blocks and easier maintenance will be possible.

First tests have been started with the Virtex II Pro series. Further implementations as described in following chapters are using Virtex 5 and Spartan 6 series. Different features relevant for the timing system implementation will be described in those chapters.

4.6.3 Phase detector

Phase detectors are required at the transmitter side in order to implement the drift compensation scheme. The phase difference between the reference 1.3GHz clock and the clock recovered from the data stream received from the receiver module has to be compared. Additionally the

⁸<http://www.altera.com>

⁹<http://www.xilinx.com>

phase between input and output of each adjustable delay has to be measured.

Two possible integrated circuits have been selected for closer investigations. Those are the AD8302 from Analog Devices and the HMC439 from Hittite¹⁰. Both have been used in various other projects already. The maximum input frequency for the HMC439 is 1.3GHz, which will be the frequency to be used for this application. Earlier measurements at DESY showed, that temperature changes generate non-reproducible relations in the phase measurements. The AD8302, on the other hand, has the disadvantage of providing only 180 deg accuracy of the phase measurements, as the other 180 deg will result in the same output voltage range, but with the opposite slope (see Figure 4.6). Finally this device was chosen for later designs.

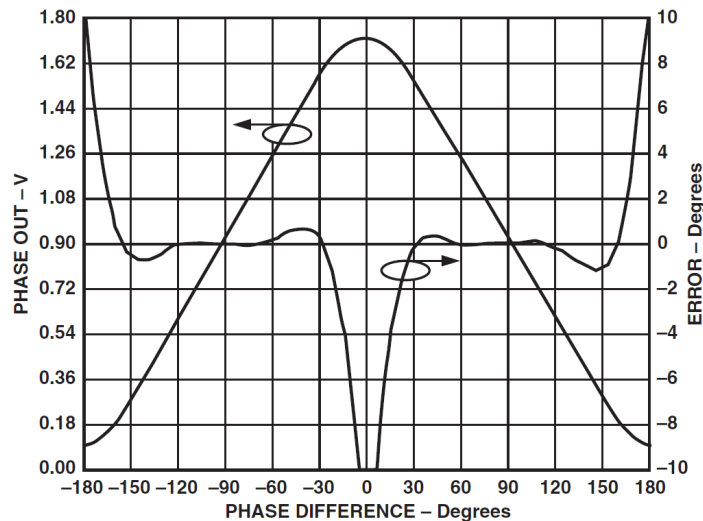


Figure 4.6: Output voltage (left vertical scale) vs. phase difference at the input of the AD8302 phase detector from Analog Devices (at 1900MHz and input levels at -30dBm). The right vertical scale shows the non-linear error. [38]

4.6.4 Adjustable delay

In order to compensate for link delay changes caused by drifts, adjustable delay elements are required. This adjustable delay has to cover a range of ± 4 ns (derived from an assumed maximum temperature range of 10 centigrades, 4km maximum length, a worst case 50ps/km/centigrade delay dependency and a factor of two as safety margin). On the other hand, the smallest increment of the delay should be as small as possible, in order to reduce limitations of accuracy. Technologies used for clock phase shifting like vector modulators or delays in clock buffers and dividers (e.g. AD95xx or LMK01xxx) can not be used. Even if they would accept serial data instead of pure clocks, the range is too low (one period of about 770ps in case of the vector modulator and 0.5 of the period for most clock buffers) or the steps are too coarse (e.g. 150ps for LMK01xxx).

The only electronically adjustable delay lines to be found are the MC10EP195 from On Semiconductor¹¹ and the compatible circuit SY89296 from Micrel. The first one only supports frequencies up to 1.2GHz and is therefore not suitable for the application here. The chosen chip from Micrel provides a delay range of 10ns in steps of 10ps. This can be set up via a parallel digital interface adjusting individual multiplexers to insert fixed delay lines as shown in Figure 4.7, and the relation between digital input and propagation delay is shown in Figure 4.8(a).

¹⁰<http://www.hittite.com>

¹¹<http://onsemi.com>

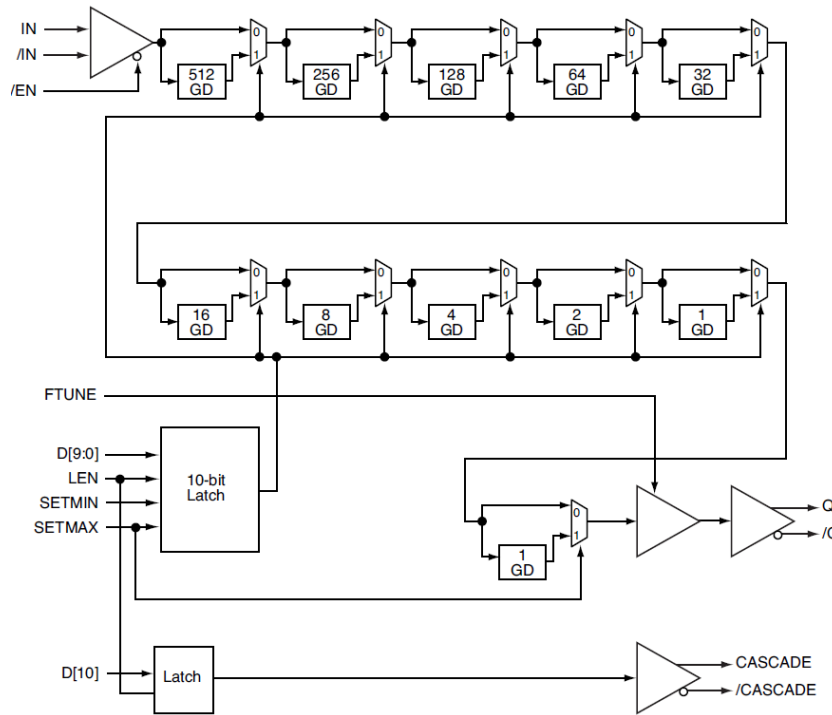
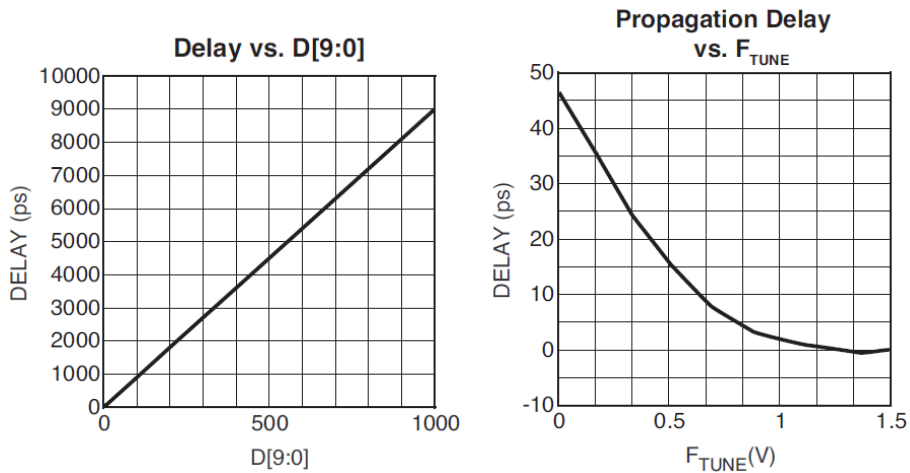


Figure 4.7: Functional block diagram of the selected adjustable delay circuit SY89296 from Micrel [39].



(a) Relation between digital coarse adjustment and the output propagation delay (b) Relation between analogue fine adjustment and the output propagation delay

Figure 4.8: Dependency of the signal propagation delay on fine and coarse adjustment inputs. The fine delay allows a delay range of about 45ps, while the coarse digital delay is able to provide delays between 3.2ns and 14.8ns in 10ps steps [39]

Additionally the device provides a fine delay, which covers a delay range of approximately 45ps. This is adjusted with an analog voltage between 0V and 1.5V (see Figure 4.8(b)). Influences on the serial data, when delay steps are adjusted and the effect on other components

in the chain will be discussed in later chapters.

4.6.5 Clock dividers and output buffers

Different clock buffer and divider circuits have been identified and will be investigated in later chapters. Divider circuits are the AD95xx series from Analog Devices, the LMK010xx series from Texas Instruments¹². Both provide a way to synchronize the internal divider based on an external pulse, which is required for the timing system. They also add only minimal jitter of less than 1ps rms.

4.6.6 Switches and buffers

In order to allow more flexibility and also reusing of certain signals for different purposes, multiplexers, switches and buffers are required. For these purposes different circuits have been identified and selected for the timing system implementation.

A 4-by-4 cross point switch, which can be used as buffer, multiplexer and switch, is the SY58040 from Micrel. Related circuits are the SY8028 4-by-2 multiplexer and the SY8020 1-by-4 buffer. A very recently developed integrated circuit by Integrated Device Technology (IDT)¹³ is a 16-port bi-directional LVDS cross point switch. This device (IDT8V54816) will be used in the timing system module described in chapter 8.

4.6.7 Phase Locked Loops (PLLs)

Phase locked loop circuits in the timing system design are envisioned for two applications: when the timing module will be tested or used without external reference, a PLL together with a crystal oscillator of a lower frequency can be used to generate the 1.3GHz reference internally and therefore allow a self-contained solution. The second application of a PLL is to generate clocks of unusual frequencies for users. The standard scheme would allow to generate clocks with frequencies derived by dividing the recovered 1.3GHz reference clock, which might be even further restricted by the divider circuit. A PLL would allow to generate an output frequency based on a derived and synchronized clock from the divider. An identified circuit for the first case is the LMX2531 from National Semiconductor¹⁴. The output frequency range is limited to a small range and will be selected to be around 1.3GHz. Due to that and the selected crystal oscillator, the circuit is able to generate the reference clock with a very low jitter. For the second application the integrated circuit Si5326 from Silicon Laboratories¹⁵ has been chosen. It allows flexible frequency generation and phase shifting fully remote controllable with a serial interface.

4.6.8 Optical transceivers

Optical transceivers are used in order to convert the electrical timing data stream into an optical signal and back on both sides of the fiber connection. The chosen standard for the transceiver modules is called Small Formfactor Pluggable (SFP) [40]. Many companies (like Avago¹⁶, Cisco¹⁷ and Finisar¹⁸) provide transceivers based on that standard. The transceivers

¹²<http://www.ti.com>

¹³<http://www.idt.com>

¹⁴<http://www.national.com>

¹⁵<http://www.silicon.com>

¹⁶<http://avagotech.com>

¹⁷<http://www.cisco.com>

¹⁸<http://www.finisar.com>

are available in single mode (around 1550nm and 1310nm) and multimode (850nm) versions. Multimode devices are cheaper, but only support distances up to some hundred meters. The single mode devices can communicate over maximum distances of around 10km (the actually supported distances depend on the product). Therefore single mode modules have to be used for the long distant connection at the European XFEL. For shorter connections also multimode modules might be suitable¹⁹.

Different bit rate standards are available. A common speed is 1.25Gbps, which is the bit rate of Gigabit Ethernet (IEEE 802.3 Clause 36 - 1000BASE-SX/LX/ZX). But also faster speeds are available up to around 8Gbps.

The preferred choice based on the available data is to use a newer generation 4Gbps single mode SFP module with 1310nm optical wave length. Newer generations of devices provide higher integration and smaller structures and tend to be more resistant to radio active radiation and provide less additive jitter. Modules with a maximum bit rate of about 4Gbps ensure reliable operation at the targeted 1.3Gbps application bit rate, which can not be ensured with the 1.25Gbps module class used for Gigabit Ethernet. However, in order to reduce costs, they will be investigated for function and reliability. Finally, using 1310nm optical wave length provides advantages on optical single mode fibers as planned to use for the timing system signal distribution. Those fibers generate only minimal dispersion of the transmitted signal at this wavelength [34].

4.6.9 Fibers

The final component of the timing system design discussed here is the fiber connection. Used to transfer the timing stream from transmitters to receivers and (in general) a second fiber to transmit the signal back to the transmitter. The required length of the connections are in between 1m and 4km. A rough estimation of fibers required for all timing system transmitter channels and receivers are in the order of 200 (assuming 100 timing receivers). With an estimated average length of 1.5km, the accumulated length is in the order of 300km.

For all long distance connections (longer than 300m) single mode fibers have to be used. As conventional telecommunication optical transceivers will be used, ordinary 9/125um (as described previously) single mode fibers are a suitable and cheap choice and have been chosen for the timing system fixed cabling in the tunnels. This has an additional advantage, as most of the other applications (data networks, machine protection system, fast feedback systems, etc) will use the same type of fibers. This reduces costs (due to economies of scale, standard connectors and splicing technologies, more flexibility of grouping fibers for different applications in cables and patch panels) and also allows more flexibility in reusing fibers for other applications.

Alternatives for long distant connections are specialized fibers like phase stabilized optical fibers (PSOF) or polarization maintaining fibers (PMF). Polarization as well as polarization mode dispersion are not of significance for the timing system performance, this is why PMF would not improve the performance. The use of PSOF would reduce the drift effect roughly by a factor of ten (see section 4.5.2). However, the drift compensation is able to compensate the drift in the same way as on SMF and therefore the performance is not expected to improve significantly. The costs on the other hand would be much higher.

Important for the drift compensation is the symmetry between forward and backward connection. Therefore the fiber pair used should be in the same bundle, have the same length and therefore almost the same delays and drift behavior.

¹⁹In SFP are also copper cable based connection cables available. They carry the name Twinax cables and can be bought in length up to 17 meters. This is an alternative for short distance connections as well.

Chapter 5

Evaluation

The first evaluation step focuses on the investigation of the previously selected critical components. This includes functional tests and determination of additive jitter. Some of the measurements have been described in [2] and only the results are presented here. Other aspects will be described in more detail.

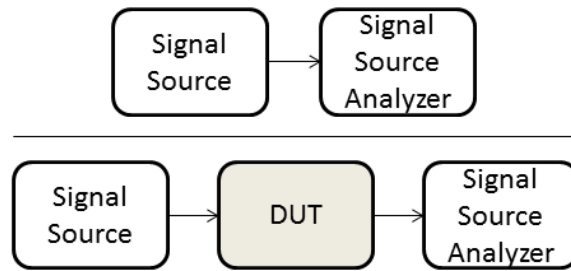
In the next step an evaluation board including these critical components were designed in order to set up a complete transmitter-receiver-transmitter loop to implement and test the drift compensation scheme. The prototype board as well as the measurement setup, the implementation and results of the measurements will be presented and have also been presented and published in [41].

5.1 Investigation of critical components

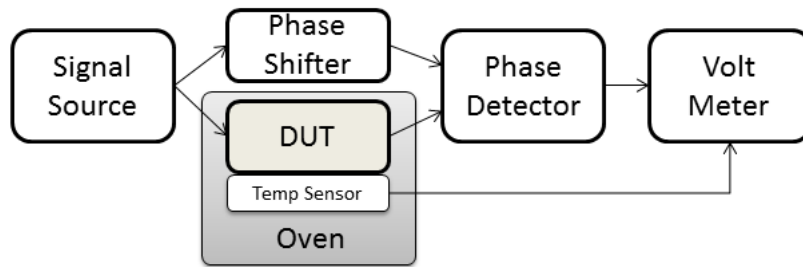
5.1.1 Delay, CDR and clock dividers

Important aspects to be investigated are additive jitter, temperature dependent drift, deterministic phase relations, behavior at temporarily missing input signals and influence of data patterns compared to pure clock cycles.

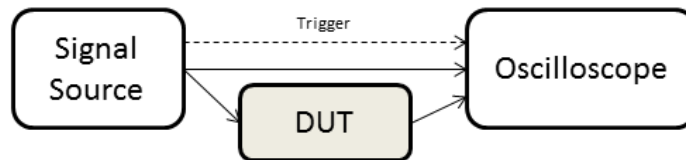
In order to investigate the components (devices under test - DUT), evaluation boards for each of them have been bought from the manufacturers. The setup for investigation of jitter, drift and phase relations and behavior at missing input signal are depicted in Figure 5.1. In case of the jitter measurement (Figure 5.1(a)), the used signal generator (a data timing generator DTG5274 from Tektronix and SMA100A from Rohde und Schwarz) was connected directly to the signal source analyzer (E5052B from Agilent). The resulting phase noise spectrum were measured and the integrated jitter calculated and saved for future comparison. After that the different components (DUT) are connected, the resulting phase noise and integrated jitter could be determined. The results, compared to the pure source, represent a measure of the additive jitter and provides insights of deterministic noise components (if present and relevant). The setup for drift measurements is shown in Figure 5.1(b). The SMA100A is used as signal source and connected to the DUT. Then the output phase is compared to the phase of the source without passing the DUT (in this case an evaluation board of the AD8302 phase comparator from Analog Devices Inc. [38] has been used). In order to adjust the two phases at the comparator, a phase shifter has also been added. The output of the phase comparator is captured by an ADC (in this case a Voltmeter Fluke87) and saved over a longer time frame. In parallel a temperature sensor is attached to the DUT and captured and saved as well in order to correlate phase and temperature later on. In order to investigate the phase change based on temperature of the DUT, it is placed in an oven and temperatures between 20 centigrade and 25 centigrade are generated.



(a) Setup for jitter measurements. Initially the source is connectd directly to the signal source analyzer in order to capture and store the phase noise spectrum for later comparison. Then the DUT is included into the signal path and the resulting phase noise spectra measured and the integrated timing jitter calculated.



(b) Setup for drift measurements. The phase change over time and/or dependent on the temperature is detected by comparing the source’s direct signal and the one through the DUT. In order to set the initial phase at the detector to its optimal position, a phase shifter has been used. Phase detector output voltage is measured and recorded over time with a volt meter. Additionally, the DUT is placed in an oven along with a temperature sensor, which is also connected to the volt meter.



(c) Setup of the transient signal and data measurement. The source is connected directly to the oscilloscope as well as passing the DUT in order to compare the input and output of the DUT. Additionally the source generates a trigger in order to indicate special conditions to be investigated.

Figure 5.1: Three different types of measurements have been carried out: short term jitter, temperature dependent phase drift and transient/data signal behavior.

Finally Figure 5.1(c) shows the setup to investigate transient and deterministic behavior like start-up conditions, missing input signal, data pattern variation and phase relations. In this case the DTG5274 output is attached to an oscilloscope and the DUT in parallel. The output of the DUT is also connected to the scope in order to compare signals. Further more, the source will generate a trigger signal based on certain conditions to be investigated. This signal will be used to trigger the oscilloscope.

The results of the measurements are presented in the following table.

Table 5.1: Results of the measurements of critical components. All components have been measured individually with dedicated evaluation boards from the manufacturers.

| | Adjustable Delay SY89296 | CDR ADN2812 | Clock Divider LMK01000 | Clock Divider AD9516 |
|------------------------------|-----------------------------|------------------|----------------------------|----------------------------|
| Additive Jitter ^a | <1ps | <1ps | 0.35-0.57ps ^b | 0.4-3ps ^b |
| Phase Drift | 3.3-15ps/K ^b | 5ps/K | 0.18-0.51ps/K ^b | 0.09-0.15ps/K ^b |
| Deterministic Phase | yes | yes ^c | yes ^c | yes ^c |
| Compatible to Data | yes | yes | not applicable | not applicable |

^a Integration of phase noise between 10Hz and 10MHz.

^b Proportional to the configured delay.

^c The component has to be reset/synchronized after the input signal has been disconnected or switched off and is then up again. Otherwise a different phase relation is possible.

The results show, that all investigated components provide a low additive jitter. Taking further into account, that the CDR includes a PLL with, currently not optimized, external loop filters, which allow jitter attenuation of the received timing signal, the chosen components will be suitable in terms of jitter requirements. Another important finding can be deduced from the drift measurements: The delay and CDR introduce a significant phase drift depending on temperature. Therefore the two supposedly symmetric delay elements should be tied together in terms of temperature (for example by using the same heat sink) and should experience only slow temperature changes in order to be able to compensate for them (e.g. using a heat sink and keep airflow stable). The temperature of the CDR should be kept constant as far as possible or phase correction based on measured temperature and a calibration table should be considered.

Furthermore the measurements revealed, that all components provide a deterministic phase relation between input and output, if a re-synchronization has been performed, after start-up or an input signal was missing. Therefore a signal detection (or signal loss detection) has to be implemented and used to reset or re-synchronize the related components. Finally it has been verified, that the delay element as well as the CDR are working properly with the type of data pattern intended for later data transmission.

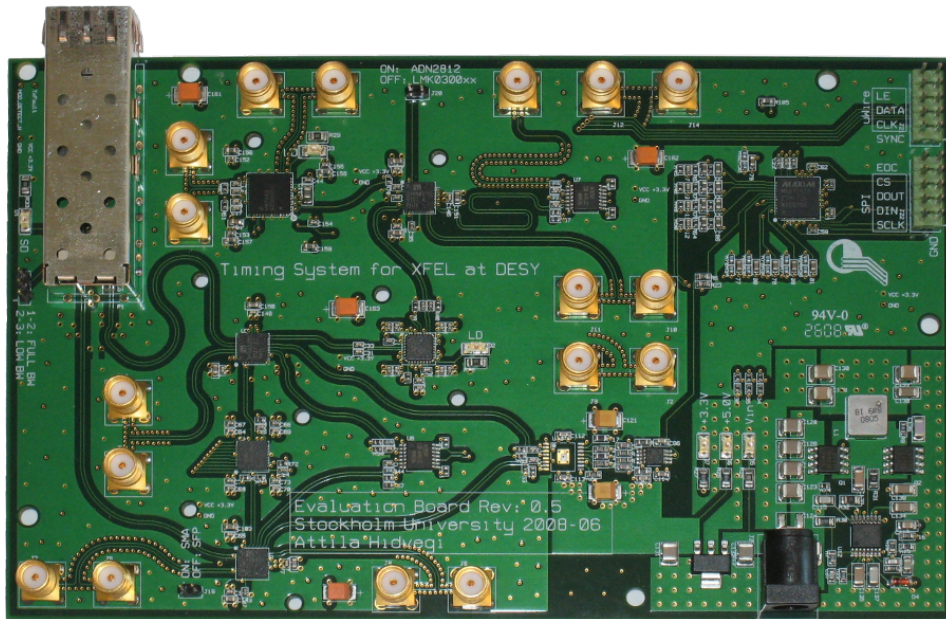
5.1.2 Phase detector

The phase detector is intended to be used at three positions within the drift compensation scheme (see Figure 4.2): at the end of the full loop to compare the recovered phase with the reference and to compare the input and output phase of each delay component. In the first case both inputs of the phase comparator are always continuous clocks and they will be even kept at an phase difference of about 90 degrees by the drift compensation algorithm. However, in the other case, the signal is the timing data stream and not a continuous clock. The measurement described here will investigate the influence of the signal pattern on the phase detector output and design consequences based on that. The setup for this test is the same as shown in Figure 5.1(c) The source was configured to generate arbitrary data patterns with a bit rate of 1.3Gbps. The phase detector measures the phase between input and output of the delay component (which was configured to a mid-range delay). The oscilloscope showed the output of the phase detector.

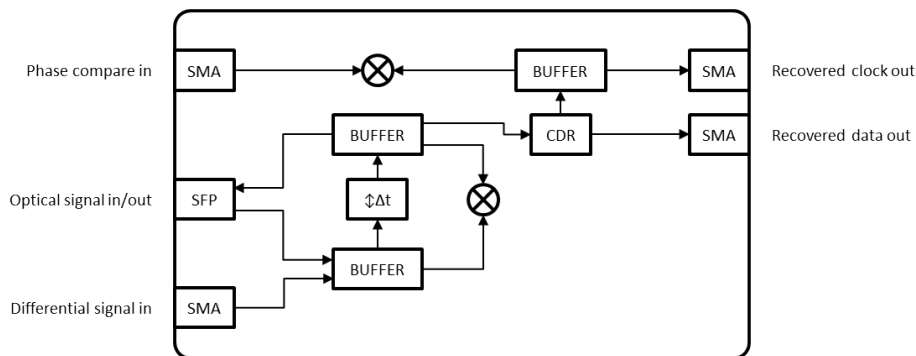
The result of this test was, that a wide-band and noisy signal is visible at the output of the phase detector and no phase relation can be determined. Repeating the same test, but with a

regular (clock-like) data pattern generated by alternating zeros and ones, the phase detector output provides low-noise and reliable phase relations.

This test shows, that a regular clock-like pattern has to be included in the data stream in order to reliably determine the phase relation between the input and output of the delay component. This is feasible, as a valid clock-like data word in the 8b10b encoding table exists (D21.5) and the bandwidth required for the actual information to be transmitted is low.



(a) Photo of the Evaluation board. On the left side the SFP socket and SMA connectors provide a signal input and output interface. The center of the board implements all components required for introducing a phase delay, recovering the clock and measure phase errors. The right side implements supporting electronics for control and power supply.



(b) Simplified block diagram of the evaluation board. Only the components and connections, which are relevant for the described measurements are shown. Also supporting connections and components are omitted.

Figure 5.2: Evaluation board to investigate combinations of components used in the drift compensation scheme. The board was designed by A. Hidvegi from Stockholm University.

5.2 Evaluation board and measurement setup

After investigating the individual components, the next evaluation step is concerned with the complete drift compensation scheme¹. Therefore an evaluation board was designed as shown in Figure 5.2(a). The block diagram of this board is shown in Figure 5.2(b). This board includes an SFP slot to accommodate an optical transceiver module in order to receive and transmit optical signals as well as SMA based inputs and outputs, the delay component, the CDR, phase detectors, an LMK01000 clock buffer, power supply and a flexible ADC, DAC and digital input/output circuit, which can be remotely controlled via a serial peripheral interface (SPI).

In order to setup a complete drift compensation loop, three boards are required and have to be connected as shown in Figure 5.3(a). The block diagram can be slightly simplified (removing all unused components and regrouping) and is shown in Figure 5.3(b). The signal source for this measurement setup is a DTG5274 providing the data signal to the first evaluation board to be routed through the first buffer and variable delay circuit ($\uparrow \Delta t$ - SY89296) before it passes the second buffer and is transmitted optically. A phase comparator (\otimes - AD8302) is connected to the delay in order to measure the difference between the input and output phase.

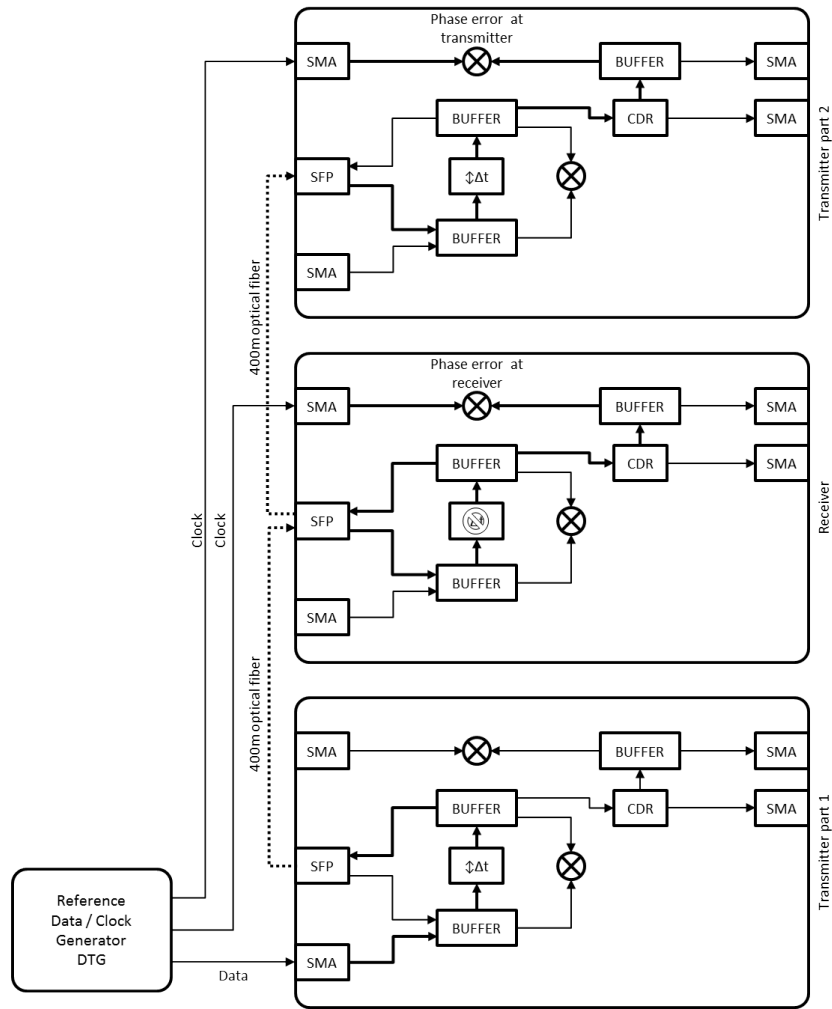
The optical fiber connection has a length of 400m for each direction around an experimental hall at DESY close to the roof on the inner side. The temperature of the cable is measured at two points (sunny and shadow side) in order to estimate the average temperature changes. Both sides of the fiber connections are at a single patch panel and the complete setup is on one table.

The signal at the end of the first fiber enters the second evaluation board, bypasses the delay element after the buffer and is then split in the next buffer into two paths: one is routed back to the optical transceiver to send the signal back and the other path connects the signal to a CDR (ADN2812). The clock output of the CDR is connected to another buffer and the signal path ends at an on-board phase detector. This detector compares the signal phase to the phase of the source, which is connected to a dedicated input. This allows to measure the resulting phase stability at the receiver side, once the phase stabilization is active.

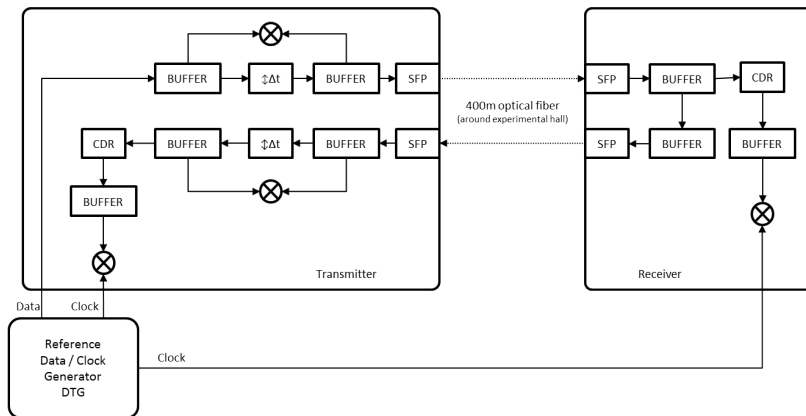
The optical signal sent back uses another fiber within the same cable in opposite direction as defined in the system design. The signal at the end of the fiber is connected to the third evaluation board. It passes a buffer and delay circuit and is then routed through a second buffer to a CDR. Also this clock output passes another buffer before it ends at a phase detector to compare the phase to the one of the reference source, which is connected to the board on a dedicated input as implemented on the second board. This will be used to measure, if the link length is kept constant within the drift compensation loop. As on the first board, there is a phase comparator connected to the delay element in order to compare the phase difference between input and output of the delay.

On all boards the serially accessible ADC, DAC and digital IO circuit (MAX12578, not shown in the block diagrams) is used to monitor all phases, temperatures and states of components as well as controlling the delay elements. The communication protocol used is SPI (serial peripheral interface) and the physical connections are attached to an FPGA on another board. This FPGA is programmed to implement the SPI communication on the evaluation board side and a communication interface to a CPU (via PCIe) on the other side. The implementation of the algorithms as well as visualization and control is performed on the CPU. A screen shot of the graphical user interface (GUI) of the prepared software is shown in Figure 5.4.

¹The described measurements have also been published in [41] in 2009.



(a) Connectivity of the three evaluation boards and the data and timing generator. Two boards implement the transmitter function and one the receiver. The most important signal paths are highlighted. Supporting connections like power supply, control and data acquisition are not shown.



(b) Simplified block diagram of the measurement setup. Only relevant components are shown and the components are grouped by function.

Figure 5.3: Block diagram of the measurement setup. It consists of three evaluation modules and the data and timing generator, which are all located on a single desk. The generator provides the data signal as well as two times the same synchronized reference clock. The optical fiber is approximately 400m long and installed on the roof inside one experimental hall.

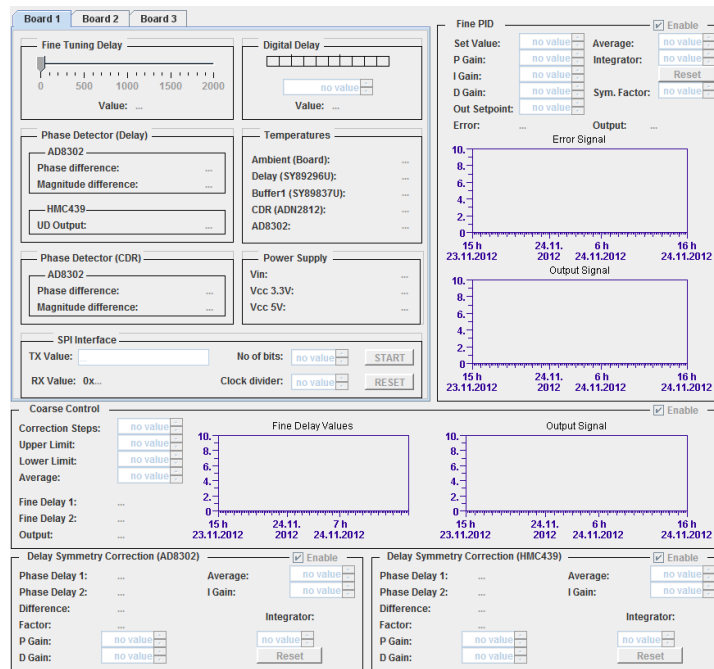


Figure 5.4: Screen shot of the prepared GUI for the control loop software implementation.

5.3 Implementation of the drift compensation scheme

The implementation in the functional sense can be divided into three parts:

1. A closed-loop controller to stabilize the phase using the fine delay
2. A controller readjusting the coarse delay, if the fine delay is close to its limits
3. A symmetry adjustment, if the two delay elements do not behave in exactly the same way

These three steps are described more detailed in the following sections.

5.3.1 Phase stabilization with fine delays

The used delay circuit implements two kinds of delays (see 4.6.4 on page 39): a fine delay and a coarse delay. The fine delay allows a range of about 45ps adjustable via a voltage between 0V and 1.5V without interfering with the signal integrity. The coarse delay is implemented via dedicated delay tabs, which can be added or removed from the signal path. Changes of coarse delay will interfere with the signal integrity in the sense of transmitted data, as a part of the signal will be lost, if a delay tab will be removed from the path (the signal in this tab will be lost) or an undefined signal over a certain time will be added, if a delay tab is added to the path. The consequence is to only use the fine delay as long as the signal integrity has to be maintained (during data transmission) and to use the coarse delay only, if it is required (no margin left on the fine delay) and there is currently no data to be transmitted.

The goal of this part of the implementation is to keep the phase difference between the recovered clock (CDR) on the transmitter side and the reference clock constant (see Figure 5.3) by equally adjusting the fine tuning of the two delay elements. This can be modeled as a classical control loop [42].

The set point of the control loop is defined in terms of the desired stabilized phase difference.

The optimal point can be derived from Figure 4.6, which describes the relation between the phase difference and the output voltage of the phase detector. The highest sensitivity (i.e. the maximum voltage change per degree phase change) along with the highest dynamic range is about -90 degrees or $+90$ degrees phase difference. In the following measurements a phase difference between the reference and the recovered clock of $+90$ degrees have been used.

Depending on the actual measured phase, the phase error compared to the $+90$ degrees set point will be calculated and provides the input to the controller. For this implementation it is modeled as a digital PID controller. The output of the controller represents the fine delay value to be applied to the two delay circuits, which are included in the plant. This adjustment will result in a link delay and therefore in a phase change of the recovered clock which is represented by the output of the plant. This is the signal used to close the loop.

The determination and optimization of the PID parameters were accomplished without a system identification, but using heuristic methods similar to the Ziegler-Nichols approach [43]. The loop time of the controller was chosen to be in between 1Hz and 10Hz.

5.3.2 Control of coarse delays

Outside of the previously described control loop, a program is monitoring the current fine delay values. As the fine delay range is limited, reaching the limit would result in uncompensated phase change, if fine delays higher than the limit would be required. Therefore a coarse delay adjustment is required, when the fine delay is close to the limit. This is implemented via software at a configurable boundary. The coarse delay will be adjusted in such a way, that the fine delay will be at half of the range.

In order to reduce phase jumps, the program will also reset the integrator value of the I part of the previously described PID controller. This will significantly speed up the convergence of the controller, as the I controller will usually show to some extent a low pass filter characteristic. The point in time the switch-over will take place will later be synchronized with gaps of data transmission.

5.3.3 Adjustments of non-symmetric delay elements

It has been shown earlier, that the delay circuits show temperature dependent delay behavior. But even if the two delays are locked to the same temperature by using a single shared heat sink, still non-symmetric delay behavior might be possible. Therefore the implementation for the measurements includes a third part, which will reduce those effects and can again be described as a PID control loop. It is performed by comparing the phase differences each delay chip introduces (which basically is the whole delay, which is introduced by one circuit, modulo the period). Assuming the delay of both circuits are the same, the phase between input and output for each will have the same difference. If not, the difference will represent an error signal, which will be used as input for the controller. The output of the controller represents the fraction of the delay adjustment to be removed from one of the delay circuits and added to the other (part of the plant). Doing this will change the just described phase relations between the two delays, which represent the output of the plant. This signal will therefore change the second input of the subtracting node and closes the loop.

Introducing this control loop can lead to three problems:

1. This loop is connected to the first one. This introduces a risk of dynamic effects between them like oscillations. Care must be taken to avoid such effects
2. If there is a significant difference between the two delays, the resulting fine delay range will be much smaller (the effective delay range is the nominal range subtracted by the delay difference). In order to reduce this effect, also a coarse delay adjustment has to be

implemented. This on the other hand would introduce a different temperature induced delay behavior, as this depends on the active delay tabs, which are then different for the two delay circuits.

3. Finally, the transfer function of the two phase comparators (see Figure 4.6) is neither fully linear (around 0 degrees and ± 180 degrees) nor is it free of ambiguities (-180 degrees to 0 degrees is the same as 0 degrees to $+180$ degrees). All these cases influence the controller and have to be implemented. For the measurements described here, a simplified implementation have been used and care was taken, that the conditions were valid.

5.4 Measurement results

The main goal of this measurement has been to evaluate the performance of the drift compensation scheme (i.e. if the drift compensation scheme and the chosen components are able to reduce phase drifts induced by temperature changes on the fiber connection significantly below the desired 10ps RMS for the recovered reference clock transmitted over the fiber).

The measurement detected the phase error between the reference clock of the transmitter and the recovered clock, when (1) the drift compensation is disabled and (2) when the drift compensation is enabled. Both measurements have been successively performed over a time of 12 hours. The temperature on the fiber connection for both measurements have been in between 17 centigrades and 24 centigrades. The measured maximum phase error in case of no drift compensation has been 63ps and with the drift compensation enabled, only 3.3ps. In these measurements, influence of asymmetric behavior and equal length of the used fibers, asymmetric behavior of the delay components, influence of ADC and DAC conversion and PID control parameter choice has not been analyzed, included or optimized. However, the measured performance showed, that even in this state the system is able to sufficiently reduce the introduced phase drift.

Chapter 6

MicroTCA hardware platform

Timing related signals play a key role in data acquisition and control systems. Most of the hardware is available or designed as modules used inside numerous enclosures (crates). Therefore the used crate standard should support distribution and configuration of timing related signals in an optimal way (e.g. low-noise, low-jitter, high-bandwidth, flexibility, etc.). This chapter provides an introduction to the used standard for the European XFEL and describes some details about timing related aspects. Further information about how the timing system will provide the relevant signals or interface to the other components will be discussed in later chapters.

6.1 Time of transition

Modular crate systems are commonly used at large scale research experiments like particle accelerators. Those systems consist of a mechanical frame called crate, which allows a number of electronic modules to be inserted. Usual modules are computing units, digitizers (analog-to-digital converters, ADCs), digital-to-analog converters (DACs), digital input and output units and numerous specialized types of modules. The communication and data transfer between modules (classically between the computing unit and the other available modules) are mediated via a backplane, which is connected with all boards via sockets with multiple pins (see Figure 6.1).



Figure 6.1: Example of a crate (in this case in the VME standard) consisting of a metal frame, a backplane with connectors to connect the inserted modules and a power supply with fan unit (source: <http://www.caen.it>)

In the past, two of the most commonly used crate standards were CAMAC (Computer Automated Measurement And Control) and VME (Verso Module Eurocard). Newer and smaller standards like cPCI (compact Peripheral Component Interconnect) and PXI from National Instruments followed. But these standards share a common principle: communication and data transfer between modules are accomplished via parallel buses and dedicated interrupt lines. As parallel buses were always known to be the communication with highest bandwidth, this was the best solution to implement. However, a common parallel bus has also disadvantages, which nowadays outweigh the advantages and some of them should be mentioned briefly:

- On a common bus only one connected module can access the bus at a time. That implies, that transfer of data has to be time multiplexed. If there is more than one master on a bus, only one can take control of the bus at a time, even if a second master would like to transfer data between completely different modules.
- Buses in the mentioned standards are implemented as single ended signals with relatively high voltage levels (around three to five Volts). If multiple lines of the bus are switched in parallel, electromagnetic effects could influence sensitive analog components on modules like ADCs.
- The transmission bandwidth on the bus is limited due to stubs for each connected module and bus arbitration schemes. Especially compared to current high-speed serial serdes (serialize / deserialize) technologies, which allow more than 10 GHz of switching speed on a single differential line.

An important part of the implemented solution in new generations of crates is therefore to switch from buses to multiple serial point-to-point connections. As they are point-to-point, the termination is at the endpoints of a differential communication channel and therefore reflections are minimized. No stubs exist and no arbitration of the channel is required. Therefore high communication speed can be achieved. Bundling of such channels allows a further increase of bandwidth. Dedicated point-to-point connections allow communication between different modules at the same time. Configurable switches allow changing of communication partners.

6.2 Introduction to ATCA and MicroTCA

The PCI Industrial Computer Manufacturers Group (PICMG) is a consortium of mostly industrial companies defining new standards, providing specifications and support in order to establish those standards in industry and many other fields. In 2002 the new standard PICMG 3.0 has been ratified and later extended, which became known as Advanced Telecommunication Computing Architecture (ATCA). This standard defines a crate system fully based on serial point-to-point connections and was primarily designed as the next generation standard for telecommunication systems, but found also many applications outside of this sector (see Figure 6.2).

An ATCA crate can support up to 14 modules (called blades), which could implement different functionality, and communicate via the backplane. The communication channels are denoted as fabrics and support different standard protocols like ten megabit up to gigabit Ethernet on the base interface and Fiber Channel, 10 Gigabit Ethernet (XAUI), InfiniBand, PCI Express, or Serial Rapid IO on the main communication pipes. Even newer standards have been adopted like 40 Gigabit Ethernet, where also the single line speed has been increased on

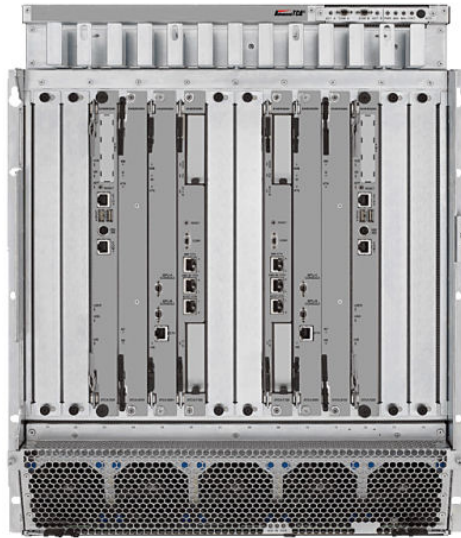


Figure 6.2: *Advanced Telecommunication Computing Architecture (ATCA) System*
(source: <http://www.picmg.org>)

the backplane.

The dimensions of a blade is 280mm x 322mm and in that respect relatively large. It allows powerful processing and switching modules, which are important for telecommunication applications. However, in many cases it is desirable to allow further modularity in smaller scales. In ATCA this is implemented as carrier boards, which accept smaller modules, which are denoted as Advance Mezzanine Cards (AMCs) (see Figure 6.3).

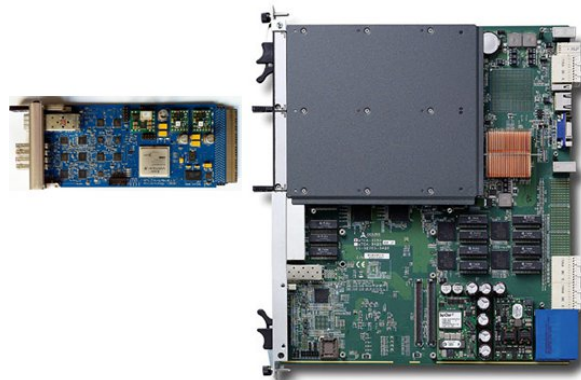


Figure 6.3: *Advanced Mezzanine Card (AMC) with a carrier module for ATCA*

Besides the large scale telecommunication installations, there are also many applications, where only a small system is required. Although it is possible to use a crate with less slots filled with one or two AMC carries to allow all required functions via different AMCs, this solution has some overhead costs. The solution identified was a new crate standard, which had been defined as MicroTCA (μ TCA). That crate allows to directly insert AMCs and provides a backplane directly to those modules (see Figure 6.4).

All required management functionality (usually implemented on a carrier blade) has been moved into a special module in a dedicated slot. This important module is called Mi-



Figure 6.4: Example of a conventional 19" MicroTCA crate from Elma. It provides 12 slots for AMCs plus two MCHs and power supplies (source: <http://www.elma.com>)

croTCA Carrier Hub (MCH) (see Figure 6.5). In that configuration up to 12 AMCs can be used in one crate with only one required MCH. This technology combines lower costs and smaller scale with comparable flexibility and advantages of the ATCA standard.

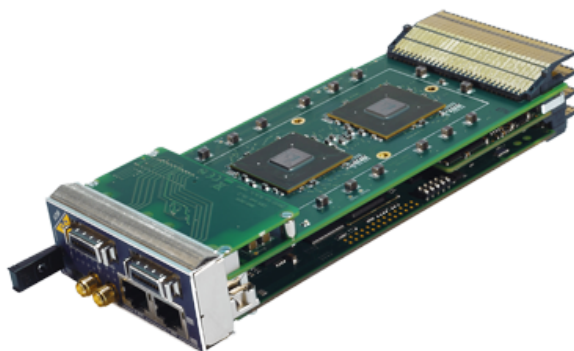


Figure 6.5: A MicroTCA Carrier Hub (MCH) from N.A.T. This module provides all functions, which are usually implemented on a carrier module in an ATCA crate. It also implements switches for 1Gb Ethernet and other protocols and allows full remote access. (source: <http://www.nateurope.com>)

6.3 Features of the MicroTCA standard

MicroTCA inherits almost all features of the ATCA system and only some aspects has been mentioned in the previous section. The main features of this standard will be described briefly in the following paragraphs:

6.3.1 Passive Backplane

In order to allow any kind of function and communication, a backplane is required, which connects all inserted AMCs in a defined way. A simplified overview of a backplane connectivity is shown in Figure 6.6.

All signal channels are defined as ports and numbered from zero to 20. Each port has an input (RX) and output (TX) differential line. The first two ports support Gigabit Ethernet and are connected to the MCH (port 0 to MCH1 and port 1 to an optional redundant MCH2). Ports 2 and 3 are used to be connected directly to neighboring modules. They are mostly used to connect a CPU AMC with hard discs via SATA (Serial Advanced Technology

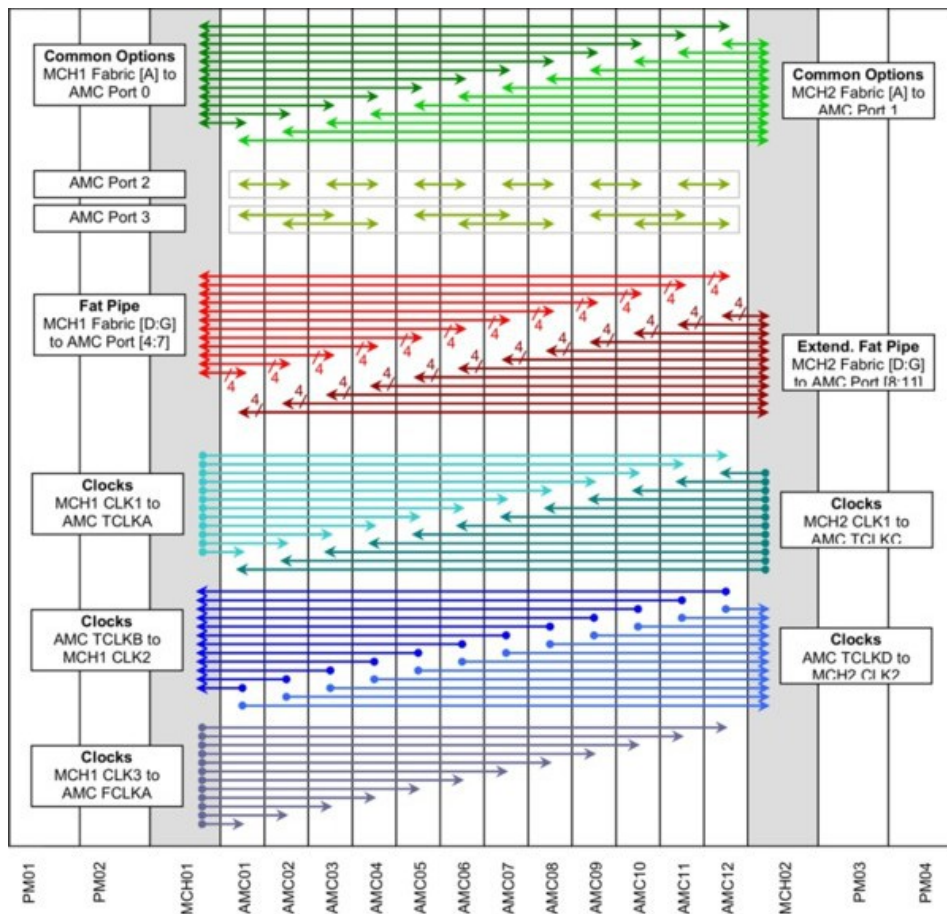


Figure 6.6: Simplified diagram of a MicroTCA backplane connectivity as implemented by Schroff on a 12 slot MicroTCA crate. (source: <http://www.schroff.de>, modified by this author)

Attachment) on those ports. The following ports 4 to 7 are named fat pipe and consist of four channels which can be bonded together to increase bandwidth. They are connected to the MCH and support different protocols as defined for ATCA (e.g. PCIe, SerialRocketIO, XAUI, etc.). The following ports 8 to 11 are named extended fat pipe and can fulfill the same purpose, but they are connected to an optional second MCH2. The ports 12 to 20 are defined as extended options region and are not connected in most conventional MicroTCA crates. Besides the high-speed data channels, there are also five differential clock lines defined: TCLKA-TCLKD and FCLKA. The TCLKA-D lines are user defined telecom clock lines, where TCLKA and TCLKB are connected to MCH1 and TCLKC and TCLKD are connected to an optional second MCH. In this case TCLKA and TCLKC are defined to be from MCH to AMC and TCLKB and TCLKD to be from AMC to MCH. FCLKA is a reference clock and mostly used as reference for serial data transmission on the fat pipes.

6.3.2 Management of modules and hot-plugging

An important feature of the MicroTCA system is that almost all aspects are managed. This includes

- Power supply for modules
- Switching of communication channels and clocks between AMCs
- Fan speeds
- Temperatures
- Hot-plugging
- Firmware management

When a new AMC is inserted into a MicroTCA crate, the module will only receive 3.3V in order to power up a small management micro controller on the module, which is called Module Management Controller (MMC). Then the MCH will recognize, that a new module is available and will start communication on a dedicated low-speed communication channel with the MMC. In a protocol called IPMI (Intelligent Platform Management Interface) the MMC provides information about its module (e.g. required current, number and type of communication channels, revisions, features, sensor information). Each AMC has a handle at the bottom of the front panel. This handle allows easier removal of the module but also has a switch included. If the handle is pushed in, the switch is closed which indicates, that the module should be enabled. The state of the switch is detected by the MMC and the MCH will be informed to enable the board. The MCH will check if the power supply of the crate is able to provide enough current for the board and if other conditions are met in order to safely power the module. If everything is fine, the power supply will be informed to switch on the 12V supply voltage for the new module. Additionally the CPU (central processing unit) AMC will be informed in order to start the integration process (load drivers and start related software).

If a board should be removed, the procedure is the opposite. The handle will be pulled, CPU will be informed and the MCH will wait for a positive respond and then the 12V is disabled. LEDs on the front panel and MCH indicate the state of the module and when the board can be safely removed.

Besides that, there are many more functions provided by the management, which are described in the MTCA.0 [44] and AMC.0 [45] specifications.

6.3.3 Centralized switching and distribution

Besides the management function of the MCH, it also implements different kinds of switches and buffers in order to control and provide communication between AMCs. One base switching function is implemented for Gigabit Ethernet. Switching of the fat pipe channels depend on the chosen technology. Usually for each technology there is a different switch required. Therefore a mix of different standards within one crate is usually not supported.

Besides switching, the MCH is also in charge of distributing clocks (frequencies) within the MicroTCA crate. Telecommunication clock A and B (in redundant cases also clock C and D) (called TCLKA - TCLKD) are available for user defined purposes. FCLKA is used as reference clock for communication within the crate (e.g. 100MHz used to provide a reference clock for the serdes components for the fat pipe protocols).

6.3.4 Point-to-point connections

As written in the description about the passive backplane, limited point-to-point connections at port 2 and 3 are defined on the backplane. They are often used to attach storage to processor AMCs, but could also be used for a different purpose.

6.3.5 Redundancy

In applications, where high availability and minimal down-time is required, defined redundancy allows replacement of critical broken components. The standard and most crates allow to install at least a second MCH and power supply. Combined with the management functionality and special handling within the AMCs, almost all components of the crate could be replaced automatically in case of failures.

6.3.6 Remote access

Important for applications, where no direct access to the hardware is possible at all time (for example in the tunnel of the accelerator) remote access of all functions of the crate and its components is important. MicroTCA allows to access all management functions and health monitoring remotely via network.

6.4 xTCA for Physics working group and MTCA.4 standard

Although the above described properties and features of the MicroTCA standard makes it a good choice for large scale experiments like the European XFEL, there are some aspects missing or at least not ideal. This is why an interest group, namely "xTCA for Physics", within PICMG was formed, which worked on extending the MicroTCA (and also the ATCA) standard in order to support such applications in a better way. More than 40 participants from industry and laboratories (including DESY as a strong driver) joined this group and prepared the extension of the MTCA specification which became an official standard in 2011 as MTCA.4. The most important new features are summarized briefly in the following paragraphs and new features are shown in Figure 6.7. More details can be found in [46].

6.4.1 Distribution of slow clocks, triggers, interlocks and deterministic data

Important in data acquisition and control applications is the distribution of relatively slow but accurate and synchronized signals within the crate. Typical signals in that class are triggers (defining a point in time, when something happens - like start data acquisition), clocks in

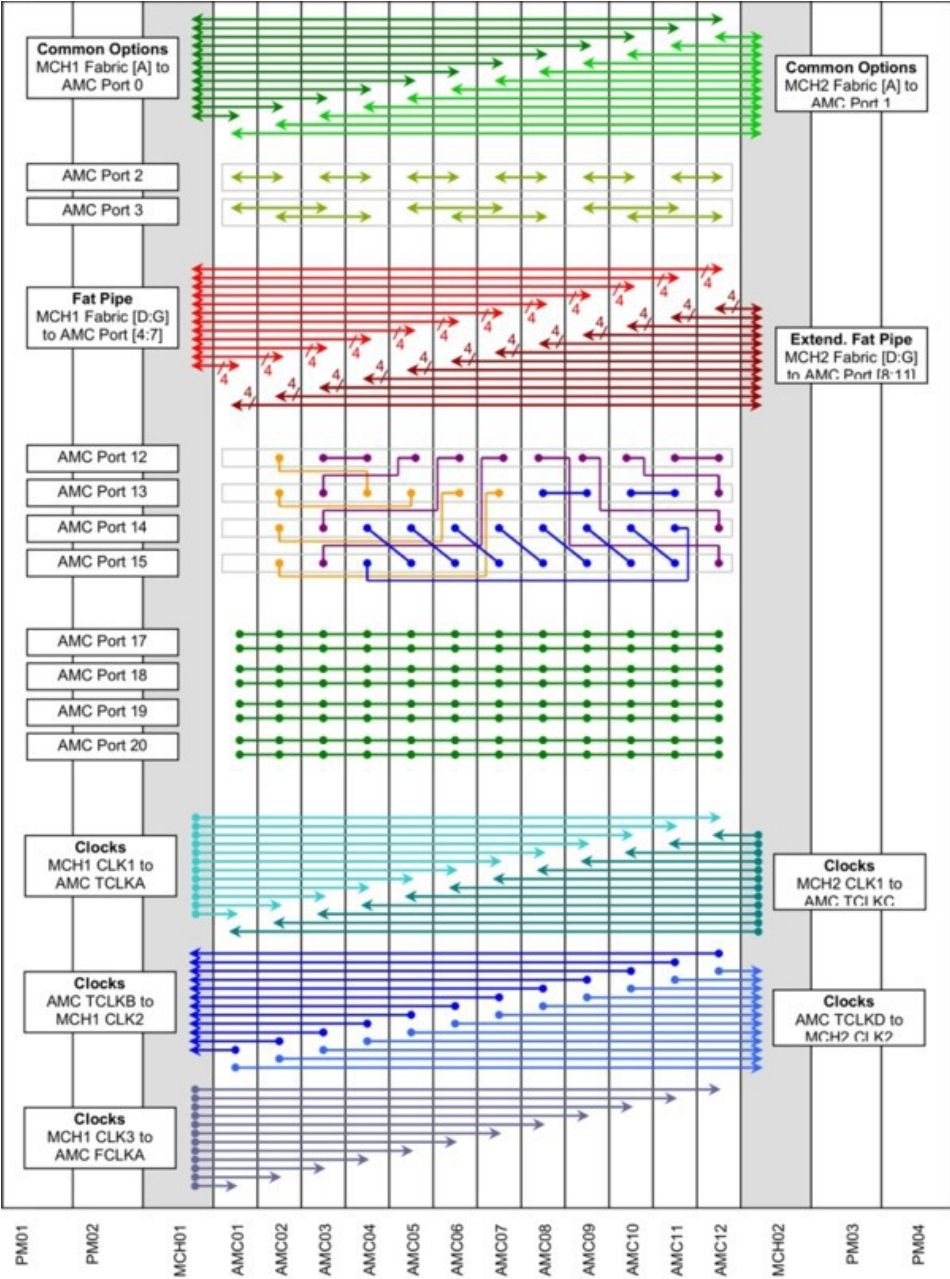


Figure 6.7: Simplified backplane interconnection of a MicroTCA.4 crate. Ports 12 to 17 has been defined to fulfill special purposes. (source: <http://www.schroff.de>, modified by this author)

kHz or lower MHz range (where each rising edge defines a certain event), interlocks (could be defined as a special kind of asynchronous trigger raised under certain error conditions) and deterministic data (information which has to be distributed in a synchronized way).

These features have been implemented in the following way: the ports 17 to 20 on the backplane, which were left open for user defined purposes, are defined as bus lines. As each port has a RX and TX differential pair, eight differential bus lines are defined in the Multipoint Low Voltage Differential Signalling (M-LVDS) standard. With terminations on the backplane and only short stubs via the connectors to the transceivers on the AMCs, frequencies up to 250MHz are specified. The bus architecture is sufficient for this application. In terms of performance it even has some advantages compared to point-to-point and switched solutions. As all AMCs attached to the bus act in the same time and signal domain, the signals are synchronized among the consumers across all bus lines. This is helpful for synchronous data transmission, where data and clock are transmitted to all AMCs. It also allows wired-or as well as wired-and configurations, which are advantageous for interlock applications, where all interlock signal producers could share one bus line and one or more of them could indicate an error condition on the backplane.

6.4.2 Double Size AMCs and Micro Rear Transition Modules

Another limitation of the MTCA and AMC standard is the available Printed Circuit Board (PCB) area. The size of a usual AMC is 73.8mm x 181.5mm. This is rather small as most applications require FPGA, memory and special chips like ADCs, DACs with their analogue interfaces or similar besides the DC-DC converters. The AMC.0 specification also defines a double width¹ form factor which results in a size of 148.8mm x 181.5mm. This improves the layout possibilities significantly. However, it is still small and not very flexible when different kinds of analogue interfacing and signal conditioning are required before a signal will be applied to an ADC input.

The solution defined in MTCA.4 foresees a second module with almost the same size as a double width AMC, but will be inserted in a crate from the rear side directly opposite to the related AMC. Two high density differential connectors (30 pairs each) provide the interface between the front AMC and the Micro Rear Transition Module (μ RTM) as shown in Figure 6.8. Besides the described analogue interfacing and signal conditioning on a μ RTM, there are also

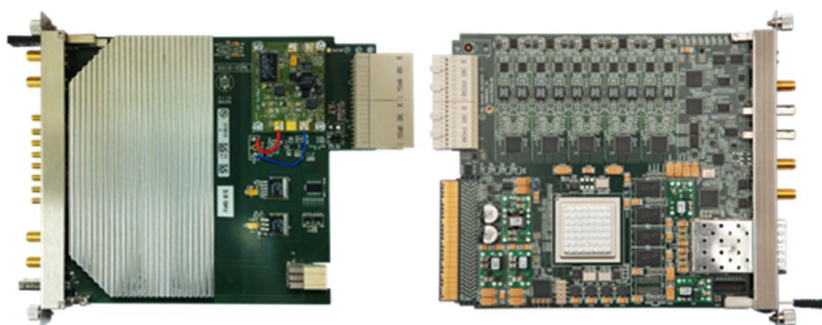


Figure 6.8: Example of an AMC and μ RTM pair. Both modules are connected via a 60 differential pair connection with individual shielding, which also allows transmission of sensitive analog signals. The right AMC is a 10 channel digitizer (SIS8300 from Struck Innovative Systems) and the left side μ RTM is a down converter module developed at DESY.

other possible applications like implementation of special interfaces like optical transceivers,

¹Double width in this context means double height of the PCB area. It is called double width, because an AMC in a carrier module for ATCA would occupy two slots and therefore needs more width.

actuators for control applications (DAC, drivers and so on), processing devices, and much more.

6.4.3 High-Speed serial point-to-point interconnects

Besides the main communication channels within the crate like 1GbE, PCIe and so on, there is a requirement of low-latency and high-speed communication channels between modules within a crate. Those are used for fast feedback systems, to concentrate data in one place in order to process it in real time or distribute information synchronously.

This has been implemented as a proposal in the MTCA.4 standard by ports 12 to 15 and allows to connect multiple AMCs as shown in Figure 6.7. This configuration provides different distribution or concentration possibilities as well as a way to daisy-chain modules in the crate.

6.4.4 Improvements on noise and jitter

The extensions on the MTCA specifications provides a standard suitable for physics applications. However, sensitive analog measurement systems require a low-noise environment and low-jitter reference clocks in order to make use of the high-resolution ADC and analogue interfaces. Especially the power supplies and clock reference and distribution units (implemented in the MCH) designed for telecommunication applications are not suitable in order to satisfy the requirements. This was realized by the manufacturers of such devices and new developments started to provide such low-noise and low-jitter components and brought more and more advanced products which now fulfill the requirements even for rather ambitious analogue applications.

Chapter 7

First generation Timing System board

The design and test of the first generation timing AMC module was a crucial step in the timing system development, since different important aspects could only be investigated with such hardware. Some of them are:

- Timing distribution within the MicroTCA crate
- Integration of the complete system
- Performance and selection of final components
- Design and implementation of FPGA firmware
- Influences of the MicroTCA environment
- Test of the transmitter and receiver

7.1 Hardware

The first generation timing system board was designed as an AMC module in single width, mid-size form factor, using the schematic and layout designed by Attila Hidvegi of the Physics department of Stockholm University. A picture of the module is shown in Figure 7.1.

The module implements the transmitter part (including reference clock input, protocol generation, optical transmission, adjustable delay elements, phase detectors) as well as the receiver part (including clock recovery, protocol decoding, trigger generation, clock distribution, dividers, phase shifters and output drivers). This combination has been chosen for different reasons. One is, that only one module had to be designed instead of two separate ones, which reduced costs and development time. It also allows to use one single module as a stand-alone timing reference in applications, where a timing receiver is required, but no central synchronization mandatory. However, the cost for a single module is higher than the estimated cost for a dedicated receiver or transmitter module. Therefore a different approach was chosen for the second generation timing system board, which will be described in the next chapter. A more detailed view of functional regions of the board are depicted in Figure 7.2 The timing data stream is usually transmitted via optical fibers. The chosen SFP transceiver is placed in the corresponding slot on the upper left side of the pictures. Below are four so called Har-Link connectors (manufactured by the company Harting) assembled, in order to provide multiple differential inputs and outputs to and from various components on the module. This type of connector has been chosen, as it provides a relatively large number of differential connections at a small size. However, it turned out, that it has more disadvantages (non-optimal impedance matched connections and expensive connectors and cables) than advantages. This is why the

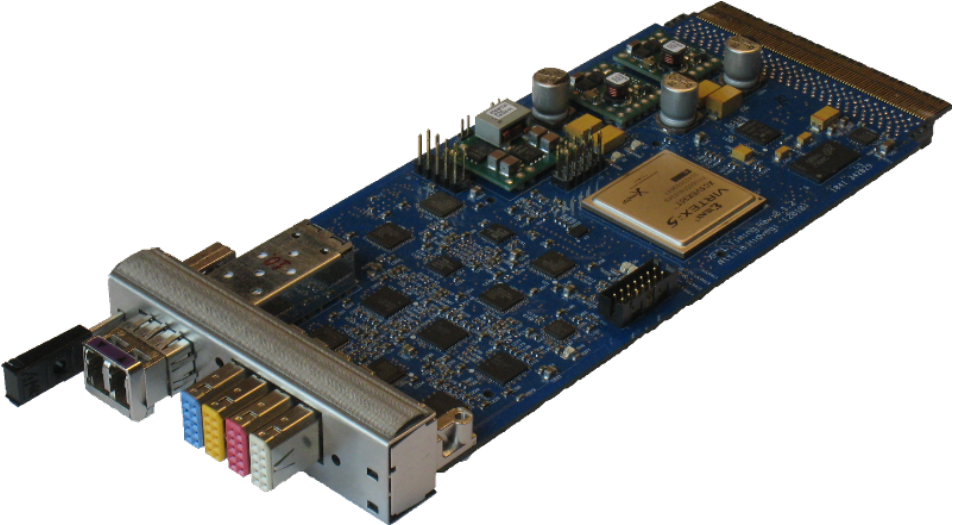


Figure 7.1: Picture of the first generation Timing System AMC module in single width and mid-size form factor. The module combines the transmitter as well as the receiver part and allows flexible testing of functionalities and performance.

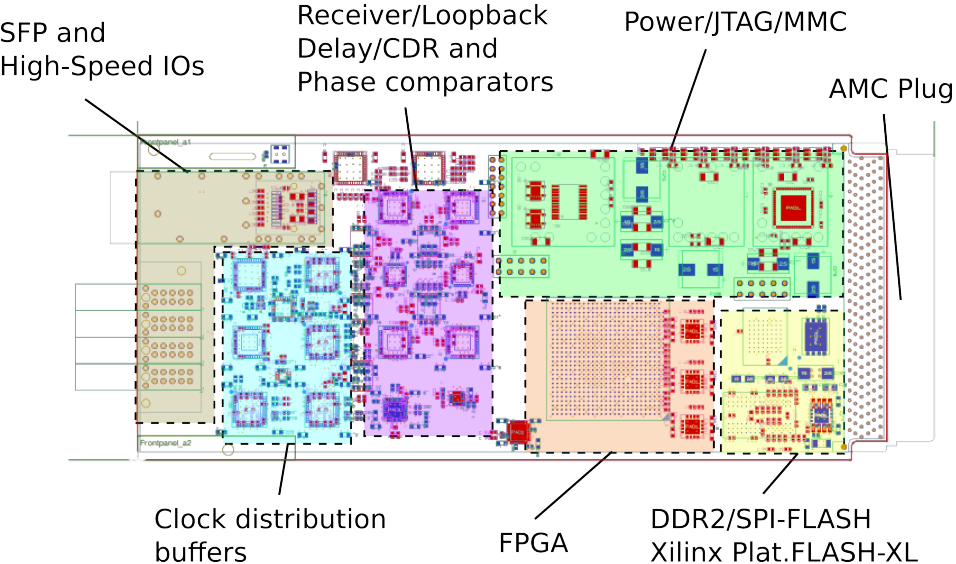


Figure 7.2: Functional layout of the Timing System AMC module (view from the top of the PCB). Different regions of the board are highlighted and their main function defined.

connectors were replaced in the second generation module.

An even more detailed view is shown in the block diagram in Figure 7.3. This diagram will be used as reference during the following detailed sections of this chapter. It should be mentioned, that many components have been included in the design in order to allow more flexible testing, verification and debugging (e.g. cross-point switches, additional phase detectors).

7.2 Reference clock and triggers

When the module is used as the transmitter or in an independent stand-alone mode (as receiver, with no connection to a transmitter - in a lab environment for example), the synchronization to internal or external references is important.

The module provides external inputs as well as internal oscillators in order to provide a reference for the FPGA and other components. An external clock can be applied through input pins (2x Reference Clock In on the right side in Figure 7.3). The clock will pass a multiplexer, divider and clock buffer component. There it can be divided if required and then connected to the FPGA (in this case a Xilinx Virtex 5). The clock is also available at many other components of the module (phase detectors, clock distribution, etc.). An external trigger can be applied at inputs on the bottom right side of the diagram (2x Trigger/Data In/Out) or from the backplane via the MLVDS bus lines.

A stable internal reference is provided by a voltage controlled crystal oscillator (VCXO on the right side close to the outputs) and connected through a MUX block to a PLL. There many frequencies can be generated and provided to the FPGA via a clock divider buffer. Internal trigger signals will be derived from the reference clock within the FPGA.

7.3 Timing data stream transmission and drift compensation

Three key features of the timing module are (1) the generation and (2) distribution of the timing data stream as well as (3) the compensation of phase drifts of this stream between the transmitter and receiver side.

The generation of the timing data stream is implemented in the Virtex 5 FPGA and is described in Chapter 9. Based on internal and/or external clock and trigger signals as well as on pre-defined configuration parameters the timing data stream is generated. It then leaves the FPGA on either of the two dashed lines on the top left side in Figure 7.3 through the flip-flop (FF) block or directly to the cross-point switch (MUX). The FF can be used in order to allow re-synchronization with an external reference clock to remove small jitter, which can be added by the FPGA. This signal would then also be routed to the same cross-point switch as the direct connection.

Afterwards the signal passes the first delay component as described in the system design description in a previous chapter. It then passes a second cross-point switch (MUX) before it reaches the SFP output, where it is converted into an optical signal and transmitted to the receiver side. There it is further used and also sent back to the transmitter side (see description below). When received it is sent to the next cross-point switch (MUX), passes the second delay component and another cross-point switch (MUX). From that point, the signal takes two different ways: one path (upper left dashed line) connects to the FPGA input in order to be decoded. The other (lower left dashed line) connects the signal to the clock and data recovery (CDR) component. The recovered clock (dotted line on the bottom) then passes the clock buffer, divider and delay circuit before it is connected via the dotted line on the right to a phase detector comparing the phase to the reference provided through the MUX for the reference clock input.

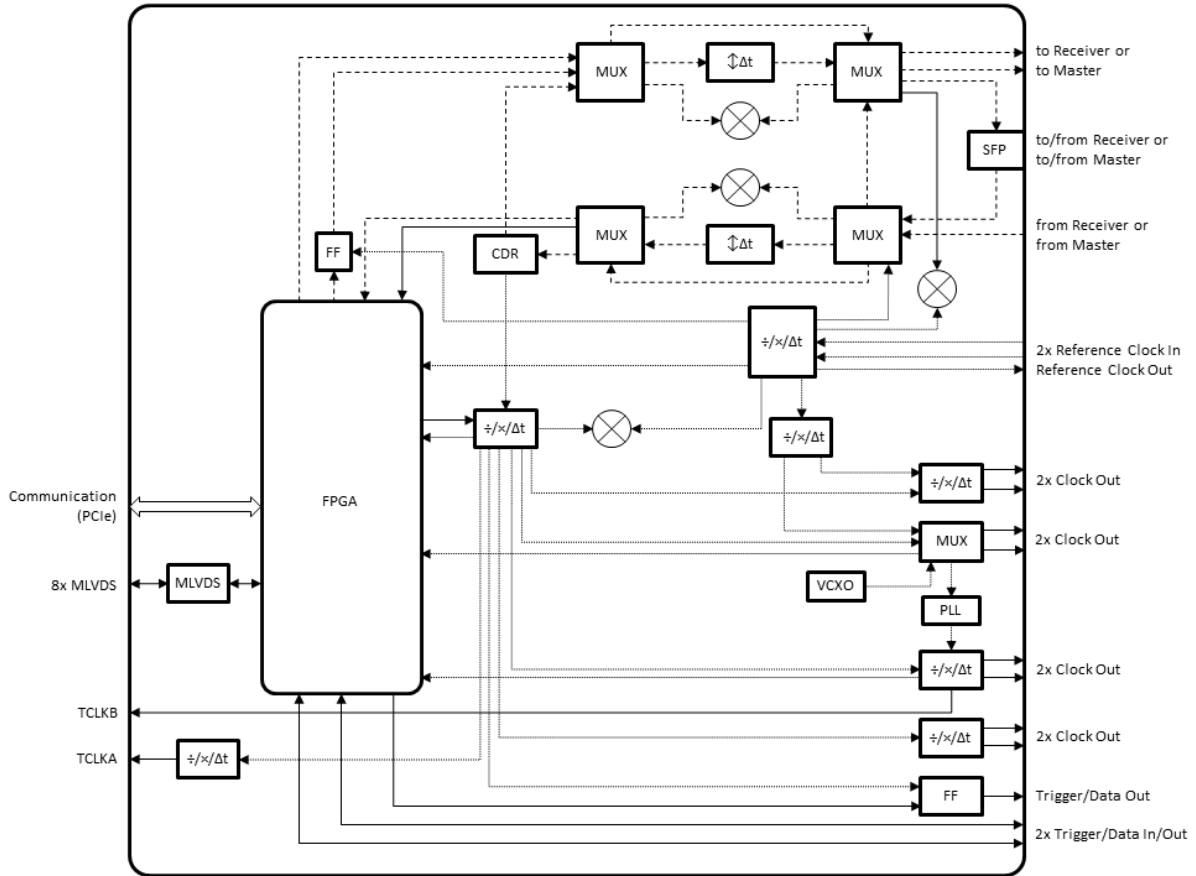


Figure 7.3: Simplified block diagram of the timing system module. The upper part of the diagram implements the transmitter or receiver aspect (depending of the chosen multiplexer configuration) of the timing system. The lower part is related to clock, trigger and data generation and interfacing. CDR denotes a Clock and Data Recovery circuit, MUX represents multiplexers, VCXO represents a voltage controlled crystal oscillator, PLL is a phase locked loop circuit, MLVDS defines an eight channel MLVDS buffer, $\div/\times/\Delta t$ represents clock buffers with frequency dividing and multiplication as well as adjustable time delay function (different circuits have been implemented for closer investigation), FF is a flip-flop, SFP is a small-formfactor pluggable socket for optical transceiver modules. Dashed lines highlight signal lines, which carry timing data signals and dotted lines highlight low-jitter clock paths. The right side of the diagram indicates front panel connections of the module. Except for the SFP all connections are implemented on four Har-Link connectors. The left side interfaces are implemented on the Advanced Mezzanine Card (AMC) connector.

Further phase detectors are connected across the two delay components as required by the system design as described earlier.

7.4 Timing receiver

When the module is used as a receiver, the timing data stream enters through the SFP module shown at the upper right position of Figure 7.3. It then passes through the cross-point switch (MUX), bypasses the delay component to reach the second cross-point switch (MUX). There it is connected to the FPGA (dashed line on the upper left of the MUX) and also to the clock and data recovery (CDR) component (lower left of the MUX). The FPGA decodes the data stream and derive many types of signals like triggers, data outputs and synchronization signals. The CDR component, on the other hand, provides a data output, which will be transmitted back to the transmitter by passing through another cross-point switch (MUX), bypassing the other delay component and passing yet another cross-point switch (MUX) before it reaches the SFP. Besides the data output, the CDR also recovers the important reference clock and provides it via the dotted line to a central clock buffer, divider and delay integrated circuit where it is used for the clock distribution section (described below).

7.5 Clock distribution

A clock distribution takes place at the lower right quadrant of Figure 7.3. The clock reference is usually provided by the CDR based on the received timing data stream as described above. But it can be provided by another external reference or by the FPGA.

The clock buffer, divider and delay circuits allow independent division ratios of the reference clock and the dividers will be synchronized through the FPGA in order to implement phase alignment between different clocks and other timing receivers. These clocks will then be distributed to different clock circuits providing clock outputs and trigger resynchronization (FF) to the front panel as well as to the two clock outputs on the backplane (TCLKA and TCLKB). The different components have been chosen in order to do a deeper analysis of the functionality, performance and stability. In the second generation module this variety is removed. Most components allow further synchronized divisions of the input clocks as well as variable delays in order to adjust the phase to external conditions. The synchronization of the dividers is implemented via additional signals from the FPGA to the clock circuits (not shown in the diagram). The configuration of all components is implemented in the FPGA and controlled by software as described in detail in the following chapters.

7.6 Triggers, bunch clocks and data outputs

Besides the high-performance clock outputs, the timing modules also provide outputs directly connected (or only resynchronized through flip-flops) to the FPGA. Those signals are very flexible as the function can be defined by FPGA programming, but also subject to some higher jitter and drifts introduced by the FPGA. Signals generated by the FPGA are triggers, gates, bunch clocks, flexible slow clocks and serial data. All types can be configured and made available on the front panel connections (lower right side of the diagram) as well as on all eight backplane M-LVDS bus lines (on the left side in the diagram).

7.7 Further options

Due to the flexible design (additional cross-point switches and other components included for easier investigations) the module supports additional functionalities which were not planned in the design described previously. Some of them are:

- Besides the SFP timing data stream interface, also a copper cable connection is possible for loop back or to other modules.
- The internal voltage controlled oscillator as well as the PLL can be used to provide a flexible frequency reference to front panel outputs.
- When used as a receiver, the timing stream can be routed through one or both delay components and used to simulate phase drifts (that was performed in measurements presented below).

7.8 Experience and consequences for next generation

Based on the tests and measurements as well as experience gained using the module in the field, different consequences for the next generation module design were defined.

7.8.1 Form factor and module separation

It turned out, that fitting the complex design on a single width AMC module was not an easy task and the manufacturing costs were relatively high. As the quantity of required modules for the European XFEL is in the order of 200, cost efficiency is an influencing factor. Therefore reducing complexity and increasing the module size would help to reduce the costs of the module.

The current module accommodates only one SFP slot and can therefore only provide the timing data stream with drift compensation for a single receiver module. In order to provide timing distribution to many receiver modules, the fan out has to be increased. On a wider module more space would be available to fit more than one output stage. However, if the module is used as a receiver, those outputs are not required and would increase the costs and power consumption. To be able to still keep a single module solution for receiver and transmitter and at the same time keeping the costs low, an add-on (mezzanine) module with transmitter functionality would be desirable. For double width AMC modules also a RTM would help to provide further space for even more output channels.

7.8.2 Clock buffers

The presented tests and measurements of the clock dividers and buffers showed, that the noise and jitter performance of the investigated integrated circuits are almost identical. The LMK01xx series provide more almost identical outputs than the AD95xx series, which is preferable in terms of channel configuration. But the AD95xx series provides more flexible delay configurations for clock phase adjustments. So both components would be suitable from that point of view.

The decision for the AD95xx series was taken based on the observation, that the LMK01xx series showed divider instabilities on external synchronization, when the input clock was 1300MHz and the divider value higher than 2 (it is at the edge of the allowed operating conditions, but still within them referring to the data sheet).

Furthermore it was decided to use LVDS signalling standards for the clock and trigger outputs, instead of CML.

7.8.3 Connectors

Although Har-Link connectors were often used in different projects and products and offer a compact solution favorable for MicroTCA front panels it has been decided to move away from that connector. Reasons are the bad quality of the cables (only single shielded cables possible) with not matched impedances of 100Ohms and high costs of such cables. Also the handling of the connectors showed some problems. Therefore it has been decided to move to standard RJ45 connectors in the next generation of timing modules.

The test in the laboratory also showed, that in many cases a conversion from differential to single-ended signals were required in order to provide signals compatible to other devices. This will be implemented in an external level converter. But a voltage supply from the timing board side is desirable in order to avoid additional external power supplies for these converters. Therefore such feature will be added in the next version.

7.8.4 Filtering of power supply

Finally the jitter measurements showed the presence of a deterministic component, which major contribution was identified as the DC-DC converters of the on-module power supplies. Further improvement on reducing them would reduce the jitter on clocks and trigger outputs. This will therefore be implemented in the next version of the XFEL Timing System module.

Chapter 8

Second generation Timing System board

Based on the experience with the first generation timing system board and the derived consequences for the redesign as well as additional wishes from potential users and cost saving options, the second generation module has been developed.

8.1 Hardware

A major difference compared to the first generation module is the splitting of different functionalities into separate PCBs, which can be connected to each other depending on the usage of the module. In this concept the timing system module is split into at least two parts: a double size AMC module (see Figure 8.1) and a transmitter mezzanine board (see Figure 8.2).

The new timing system board is designed as a double width mid-size AMC module. The board includes all components required in order to be used as a timing receiver module. Among others, these are: SFP slots, clock and data recovery, clock distribution and divider components and a FPGA. A more detailed view of the main components, their functions and interconnections are shown in the block diagram in Figure 8.3. Additionally some components related to the transmitter functionality are included as well. Those are the additional SFP slots on the front panel, clock fan-out multiplexer / buffer components and connections to the FPGA. All other components, like the delay and clock recovery elements as well as the phase detectors are included on the mezzanine module (see block diagram in Figure 8.4). The configuration is similar to the implementation in the first generation module. However, a major difference is the additional micro controller next to the components for each channel. This device will take over the role of the closed loop controller, which has been implemented on the FPGA previously. This, as well as other main functionalities, will be described in more detail in the following sections.

8.2 Reference clock and triggers

Also the new version of the board allows to generate an internal reference clock, which can be used for testing or in environments, where no external reference clock is available or required. This is implemented as before with a 40MHz crystal oscillator, which is directly connected to a PLL (VCXO) component, which is shown almost in the center of Figure 8.3. Alternatively an external clock can be connected via the SMA connector in the middle of the right side of the diagram. It turned out, that a single-ended SMA input is preferable as reference clock input,

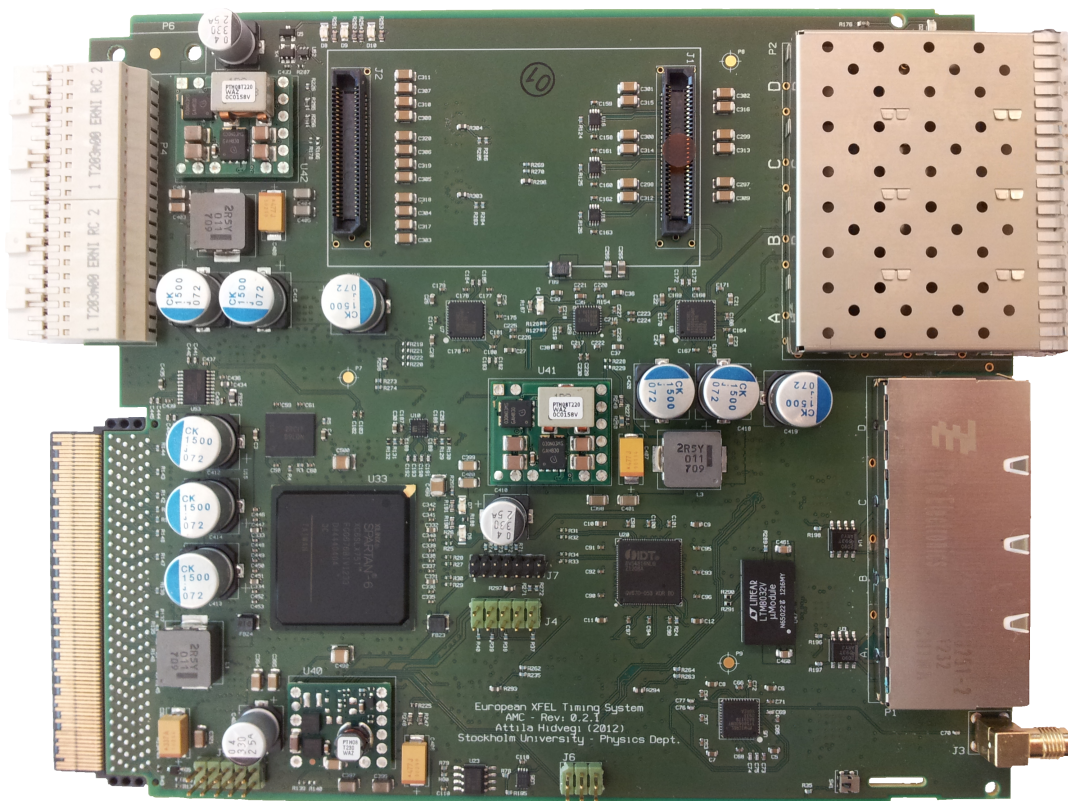


Figure 8.1: The second generation Timing System AMC module view of the top side. It implements all components required to be used as receiver module. Almost all components required to add the functionality for the transmitter are excluded and can be added as an add-on mezzanine module at the slot on the top center position.

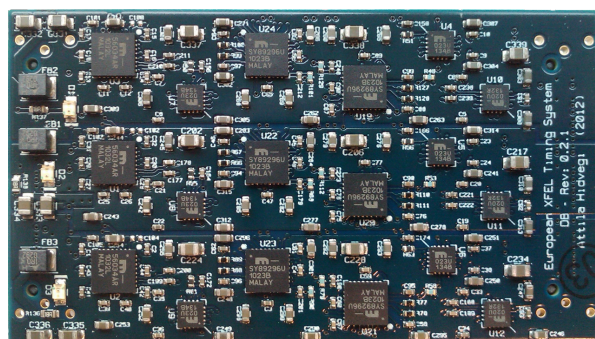


Figure 8.2: The mezzanine module for the timing AMC module. It implements all components required for three transmitter outputs and drift compensation for the timing system.

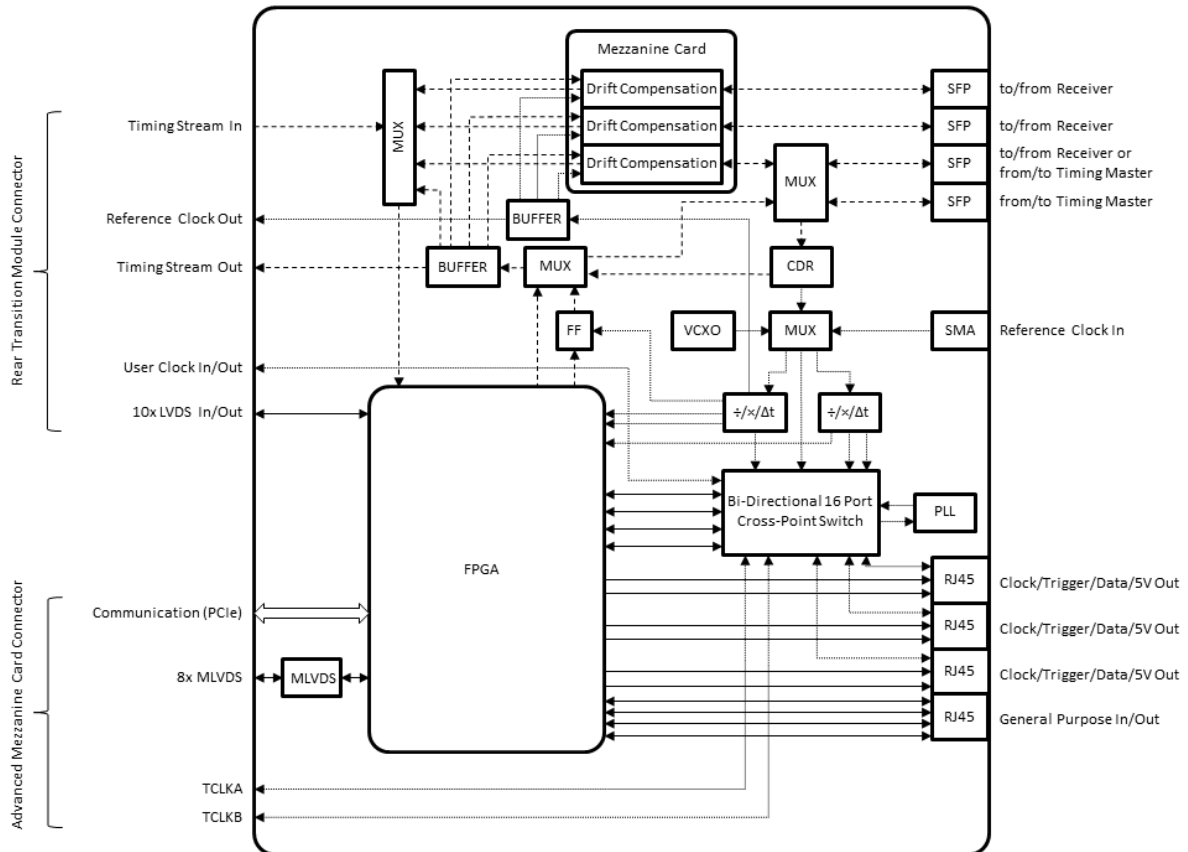


Figure 8.3: Simplified block diagram of the main components and interconnections of the timing system AMC module. Power supply, management, configuration and so on and related signals at the connectors are not shown in this diagram. The block diagram of the transmitter mezzanine card is not shown here. CDR denotes a Clock and Data Recovery circuit, MUX represents multiplexers, BUFFER represents clock buffers with multiple outputs, VCXO represents a voltage controlled crystal oscillator, PLL is a phase locked loop circuit, MLVDS defines an eight channel MLVDS buffer, $\div/\times/\Delta t$ represents clock buffers with frequency dividing and multiplication as well as adjustable time delay function, FF is a flip-flop, SFP are small-formfactor pluggable sockets for optical transceiver modules, SMA and RJ45 are the corresponding socket types. Dashed lines highlight signal lines, which carry timing data signals and dotted lines highlight low-jitter clock paths.

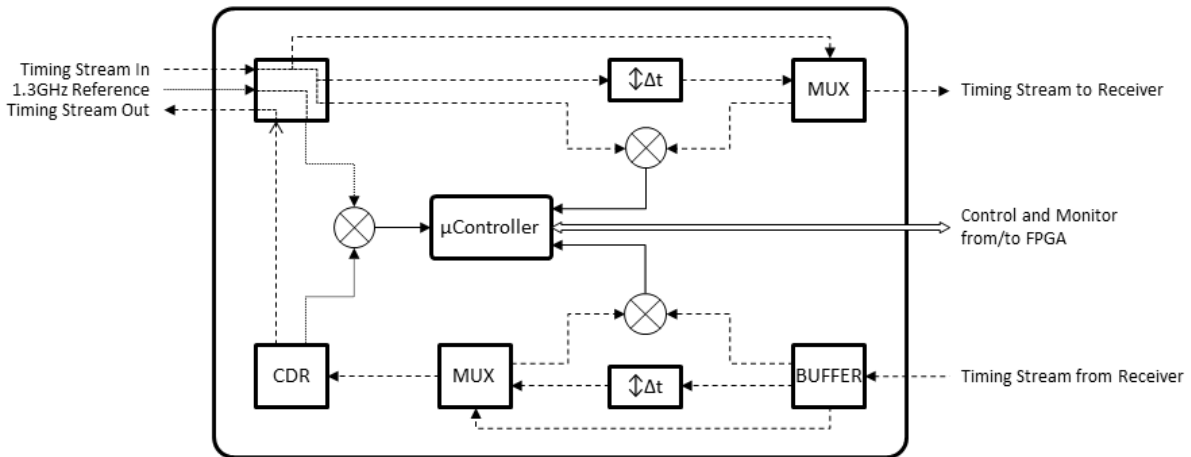


Figure 8.4: Block diagram of the main components of the transmitter mezzanine module. Only one of the three channels is depicted here. CDR denotes Clock and Data Recovery circuits, $\updownarrow \Delta t$ is an adjustable time delay circuit, \otimes represents phase comparator circuits, MUX represents multiplexers (which is also the case for the upper left block with the dashed lines) and BUFFER represents clock buffers with multiple outputs. The two dashed lines on the top and bottom can be used optionally instead of the signal path through the adjustable delay components.

as most sources are compatible to that interface (in contrast to differential inputs).

Both references are connected to a neighboring cross-point switch (MUX), which is used to multiplex and therefore select the reference input and forward the clock to different components like the clock distribution, FPGA and transmitters. As can be seen, there is even another input to this cross-point switch (MUX). The source of this signal is the clock recovery output of the received timing data stream (CDR), which will be discussed in more detail in the next section. Depending on the usage of the module (e.g. transmitter, stand-alone module, receiver) the corresponding reference inputs will be selected (e.g. SMA input, internal oscillator, recovered clock).

An external trigger input (as required to synchronize the data stream generation to an external phase like the 50Hz power line frequency) is provided on a RJ45 socket on the lower right side of the diagram. The four differential pairs are connected to general purpose FPGA pins and can be configured in the firmware.

8.3 Timing receiver

The pre-dominant application of the module (if not equipped with an mezzanine card) is the receiver functionality. The starting point of the signal flow is the optical timing data stream receiver in the lowest SFP slot on the upper right side of the diagram in Figure 8.3. From there, the signal enters the cross-point switch (MUX) next to it. The output of the bottom side of the switch connects the received signal to the clock and data recovery (CDR) circuit, which recovers the clock and forwards it at the bottom to the previously described reference clock multiplexer (dotted line). Additionally the CDR will provide the data stream on the left sided output, which will pass a cross-point switch (MUX). One output (upper right side of the MUX) will send the signal back to the first switch (MUX next to the lowest SFP slot) in order to transmit the data back to the transmitter module via the SFP module. The second output of the switch (left side of the MUX) will connect the data stream to the FPGA by passing through the buffer to the left and the upper left cross-point switch (MUX).

Looking more closely to the SFP slot described above, it can be seen, that the neighboring SFP slot is connected to the same cross-point switch (MUX). Therefore the SFP slot can be indeed used alternatively. The reason for that is the possibility for a redundant timing link to a second timing transmitter. If the receiver detects, that no signal is received from the original input, it can switch over to the second input, if it is connected.

8.4 Clock distribution

Starting point of the clock distribution part of the module is the cross-point switch next to the SMA input on the middle right side of Figure 8.3, which selects the reference clock input (described above). From that switch the clock enters the clock distribution unit, which mostly consists of two identical and synchronized clock circuits (implementing buffers, dividers and delays) and a 16 port bi-directional cross-point switch.

The clock circuits are fully user configurable and can divide the reference clock by a factor between 1 and 32 or between 1 and 1024 depending on the outputs, which can also be delayed individually. Synchronization of the dividers will be accomplished by a dedicated SYNCH input of the circuit (not shown in the diagram), which is controlled by the FPGA and will be described in more detail in the firmware chapter. The high-precision derived clocks (dotted lines) will be provided to the 16 port bi-directional cross-point switch and the FPGA. The FPGA can also generate clocks with a lower precision. These clocks are connected to the 16 port switch as well. Additionally a dedicated low-noise PLL circuit is connected with its input and output connected to the switch. Finally also clocks from the MicroTCA backplane (TKLCA and TCLKB) can be configured as inputs or outputs to and from the 16 port switch. From the cross-point switch connections are available to all relevant systems and outputs like the RJ45 front panel connectors (upper three sockets on the lower right of the block diagram), the clock lines to the first MCH on the backplane (TKLCA and TCLKB), the FPGA and the PLL.

The PLL connected to the 16 port switch has been added for two possible applications: further cleaning of clocks and generation of special frequencies. The cleaning of clocks was required, as complaints from future users of the system have been received, that missing clock cycles on clocks provided by the timing receiver would cause problems at certain user applications. Missing clock cycles will be added by the divider synchronization principle used at the clock divider circuit and will be described in more detail in the measurement section and the firmware chapter. The PLL can be used to generate a clock by multiplication from a lower frequency, where no missing clock cycles are present.

8.5 Trigger, bunch clock and data outputs

Besides the high-precision clocks described previously, other important signals provided by the timing system are triggers, bunch clocks and serial data. All these signals will be generated by the FPGA (see firmware chapter for more details). Connections on the front panel are implemented as RJ45 connectors (as mentioned in the previous sections above and are shown on the lower right side of the diagram). The lowest connector has a special implementation to be used as input or output and is not intended to be used as usual output. The other three connectors, however, follow the same pin out (as shown in Table 11.5 on page 101). The connector has four differential pairs, where one is connected to the 16 port cross-point switch to provide a high-precision clock (see above). Two other pairs are connected via a LVDS driver to the FPGA, which provides the signals described here. The final pair provides a switchable and current limited 5V and ground connection (not shown in the diagram). This can be used

for external level converters or simple consumers (see Chapter 11).

Besides the front panel connection, these signals are also connected to the eight M-LVDS bus lines on the backplane (shown on the left side of the diagram). Via the configurable M-LVDS transceiver circuit, the connectivity to the backplane can be enabled or disabled per line.

More information about these signals are provided in the firmware and interfacing to consumers chapters.

8.6 Timing data stream transmission and drift compensation

In order to transmit the timing data stream to multiple receiver modules and compensate possible phase drifts, the mezzanine module is required and has to be connected to the AMC module (see Figure 8.1 and 8.2).

Starting the signal flow description at the reference clock on the AMC module as described above, the reference clock is connected to the FPGA (via a cross-point switch and clock circuit). The FPGA generates the timing data stream with frequency and phase locked to the reference clock and based on configuration information. This data stream will leave the FPGA on the two dashed lines on the top side of the FPGA. Similar as in the first generation module, the signal can be used directly or resynchronized with an external flip-flop (FF) shown next to the FPGA, which is connected to the reference clock as well. Both signals, the direct path and the output of the FF, will end at a cross-point switch (MUX). The signal will exit the cross-point switch on the left side and reach the mezzanine card just after the buffer component. As the mezzanine implements the phase measurement and drift compensation elements, it also requires the reference clock. This is shown in dotted lines on a path from the clock circuit through another buffer.

The following details are shown in the block diagram of the mezzanine card in Figure 8.4. The interfacing between the AMC and the mezzanine module are two connectors, which are shown on the right and left side of the block diagram. The timing data stream from the FPGA (via the described signal path) is entering the module on the upper left side as dashed line. It passes a multiplexer, where the main signal path leaves the circuit on the upper right side and is connected to the first delay component. Afterwards it reaches a second multiplexer before it connects to the output on the right side of the module. Taking a brief look to the block diagram of the AMC module, it can be seen, that the signal is connected to one of the SFP sockets (one for each transmitter channel of the mezzanine module), where the signal is transmitted to a connected receiver. As described above, the receiver will send back the stream, where it is received by the very same SFP module and connected back to the mezzanine module. Back at the diagram of the mezzanine module, the returned data stream from the receiver enters the right connector on the lower side. Symmetric to the upper part of the diagram, the signal passes two multiplexers and a delay component. The following element is a clock and data recovery circuit (CDR). The clock output (dotted line) is routed to a phase detector and compared with the phase of the reference clock (dotted line coming from the middle left input). This phase difference is important for the drift compensation implementation, which is implemented on a micro controller (in the center of the diagram). Besides the overall phase information, the controller also receives the phase differences between the inputs and outputs of the delay components (shown as dashed lines with a phase detector above and below the delay components). The information will be applied in different closed control loops and the outputs will act on the two delay components, as described in previous chapters.

Besides this, the data output of the clock and data recovery circuit (dashed line), will pass through the multiplexer and leave at the lower left output. Looking again to the block diagram of the AMC module, the dashed line enters the multiplexer on the upper left side of the diagram and connects to the FPGA. In this way the FPGA is able to receive data returned by

the receivers (required to calculate link delays) as well as to receive messages from the receivers (required for registration and hand shaking between transmitter and receiver - described later in the firmware chapter). There is only one FPGA input available. Therefore the connection is switched with the multiplexer, defined by the FPGA.

8.7 Extending the functionalities

The previous descriptions have covered most of the possible paths, but not all of them. One important group of signals will be described here in more detail: the connection to the rear transition module (RTM) connector.

The signals of the interface include:

- the timing data stream output (depending on the configuration provided by the FPGA or received by any timing data input)
- an input for a timing data stream
- the reference clock
- a connection to the 16-port bi-directional cross-point switch
- multiple differential FPGA pin connections

These connections allow RTM module developments, which can implement all functions implemented in the AMC module and even more. Planned (partially under construction or even ready) modules are:

- A transmitter fan-out module (accommodating up to three transmitter mezzanine modules and providing up to 9 transmitter outputs with full drift compensation)
- An optical trigger, clock and serial data output module to provide those signals to klystrons, which require galvanic isolation
- A long distance (up to 1km) trigger and clock output module based on RS422
- A trigger, clock and serial data fan-out module

8.8 Complete drift compensation measurement

In order to measure the full performance of the drift compensation under realistic conditions, a setup has been prepared as shown in Figure 8.5 consists of two Timing modules, where one acts as the Transmitter and the other as the Receiver module. Both are connected via two optical fibers (one for transmitting from the transmitter to the receiver and one to return the signal from the receiver to the transmitter) with a length of 4km. The fibers are within one single cable on a spool. This connection represents slightly more than the largest assumed distance between transmitter and receiver for the European XFEL. Additionally the spool is placed in a way, that different temperatures can be applied to the fiber in order to simulate temperature induced drifts. In this case the spool either placed outside of the building to use the day and night cycles or inside an oven (climate chamber). Additionally, a reference 1.3GHz clock is provided by a Rohde & Schwarz SMB100A clock generator and applied to the transmitter module through the SMA input.

In order to measure the phase relation between transmitter and receiver, both modules generate a 108MHz clock with their internal clock distribution and divider section, which is making use

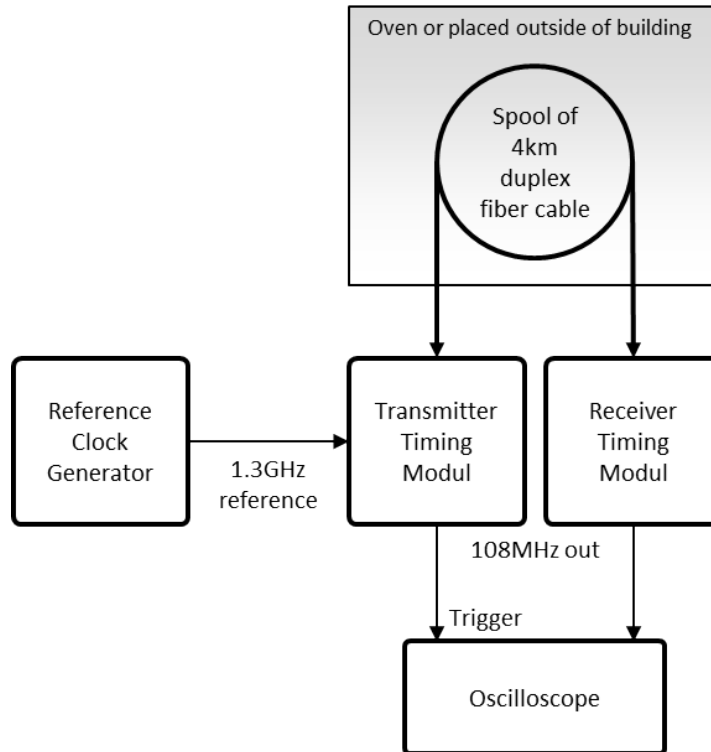


Figure 8.5: Block diagram of the drift measurement setup. Two Timing modules are placed in a MicroTCA crate. One is used as (Master) Transmitter and the other as Receiver. The Transmitter received a reference 1.3GHz clock from a clock generator (Rohde & Schwarz SMB100A). The Transmitter and Receiver are connected via two optical fibers within one cable on a cable spool. The length of each fiber is 4km. The spool has been placed outside of the building or into an oven in order to experience a temperature change. A temperature sensor is placed at the spool (not shown). Both modules generate an 108MHz output clock, which is connected to the oscilloscope. The rising edge of the 108MHz of the Transmitter is used for triggering the oscilloscope.

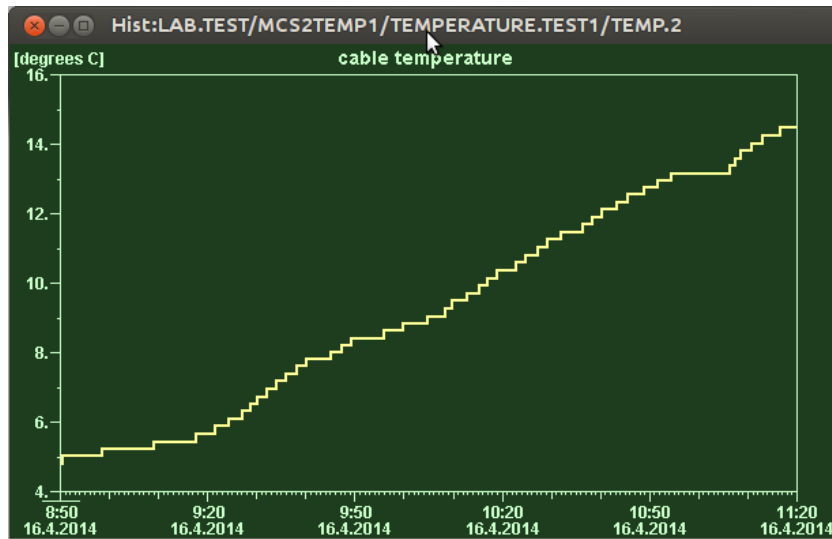


Figure 8.6: *Temperature of the fiber spool during the whole measurement time of 2.5h. The temperature changed from slightly less than 5 centigrades up to almost 15 centigrades resulting in 10 centigrade temperature difference.*

of the FPGA based divider synchronization. In this way also this component is included in the measurement. Both clocks are connected to an oscilloscope, where the rising edge of the 108MHz clock of the transmitter is used to trigger the oscilloscope.

In this measurement, the fiber spool was experiencing a temperature change between slightly less than 5 centigrades up to almost 15 centigrades, which results in a change of almost 10 centigrades (affecting the propagation delay for both fiber connections in the same way). The temperature change can be seen in Figure 8.6. The resulting phase shift was observed by the phase detector on the transmitter and the drift compensation implementation is therefore adjusting the fine and coarse delay of the two delay components (the adjustment of each delay component can be seen in Figure 8.7). In this case a coarse delay adjustment of around 2ns was required. Also corrected, but not shown in the figures is the symmetry detection and correction as described in earlier chapters.

The final results can be seen in the oscilloscope screen shot shown in Figure 8.8. The 108MHz of the transmitter module used for triggering the scope is not shown. The acquisition mode of the oscilloscope was set to infinite persistence. The trace in the middle shows the region of the rising edge of the 108MHz of the receiver module. The gray slit-like rectangular box in the middle represents the part of the edge with the largest phase fluctuations and had therefore been chosen for determining the performance parameters. The peak-to-peak phase fluctuation during the whole 2.5h measurement time is 78.5ps. This includes short term jitter as well as the residual long term drifts. Additionally the histogram of the gray region is calculated and shown at the bottom. It shows a Gaussian distribution with a resulting σ of only 9.71ps as an estimation for the average residual phase fluctuation, which is perfectly fulfilling the related design requirements.

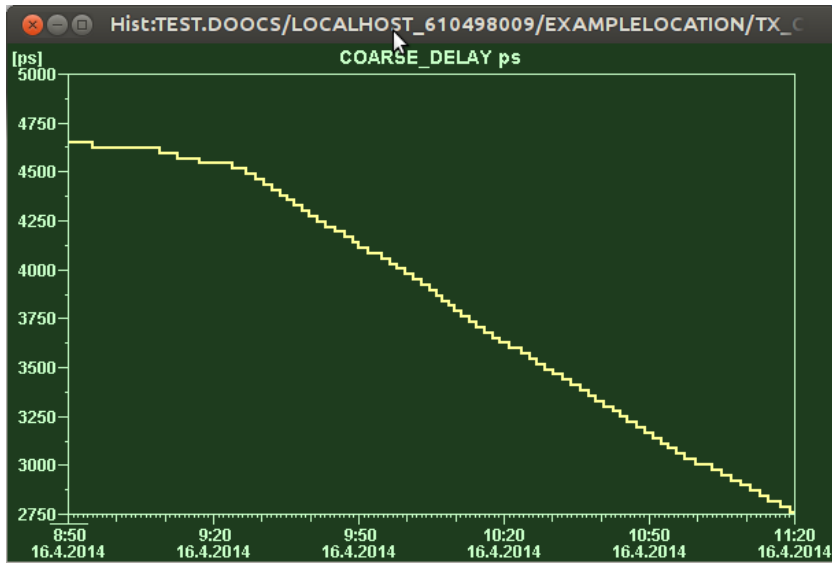


Figure 8.7: The resulting coarse delay corrections calculated by the drift compensation algorithm on the micro controller and applied to the delay components are shown over the whole measurement time (the fine delay adjustments are not shown). The adjustment results into almost 2ns coarse delay.

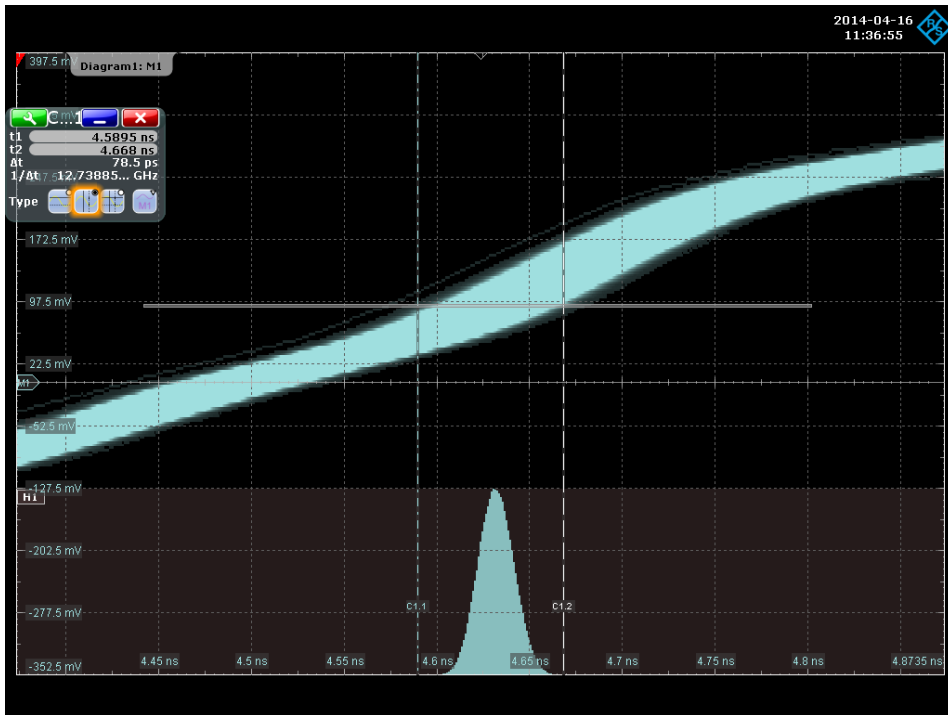


Figure 8.8: Screen shot of the oscilloscope measuring the over-all phase fluctuation. The trigger of the oscilloscope is the rising edge of the 108MHz clock provided by the transmitter module (not shown in the picture). The shown trace is the rising edge of the 108MHz clock generated by the receiver module (based on the internally recovered 1.3GHz reference and divided by 12 and synchronized by the synchronization process). The acquisition was set to infinite persistence. The histogram shown at the bottom was calculated based on the small slit in the middle of the screen and represents the widest part of the edge. The measured peak-to-peak variation is 78.5ps. The resulting σ of the distribution is about 9.71ps.

Chapter 9

Firmware

Except for the critical clocks, which are routed directly between the clock recovery and clock output buffer and driver circuits, a FPGA represents the central component of the timing system module. The Xilinx Virtex 5 FPGA in the case of the first generation timing system module and a Xilinx Spartan 6 FPGA in the second generation module implement the main functionalities. On the second generation module the optional mezzanine board, which provides most of the components for the three transmitter channels, includes an Atmel AVR 8 bit RISC micro controller for each channel. All of these complex devices require firmware in order to implement the functionality required for the timing system, this firmware is described in the following sections in more detail.

9.1 Configuration of integrated circuits on the module

Many of the integrated circuits used on this module are complex elements and require external configuration or at least allow monitoring of important parameters. The interfaces for these components are different: starting from analogue voltages (like the fine tuning input of the adjustable delay components), over binary digital (e.g. coarse delay configuration) and SPI and I²C protocols. The implementation of these interfaces differs slightly between the first and the second generation module. In the first generation module all components are located on the AMC module and the FPGA implements the communication interfaces. In order to generate digitally definable voltages and to reduce the number of digital lines to the FPGA, the first two interfaces are connected through another integrated circuit, which provides those signals (a serially programmable I/O component). This chip provides an SPI interface. Therefore all components are connected to the FPGA via serial communication on SPI and I²C protocols. The firmware includes an implementation of these standards and allows the FPGA or a connected in-crate CPU to configure and read out all parameters.

In the case of the second generation module many components (related to the transmission of the timing data stream and drift compensation) are located on a mezzanine module. The configuration of those components as well as the later described drift compensation are implemented in the Atmel AVR micro controllers on the mezzanine board. Each micro controller provides a I²C interface to the FPGA and as all other components left on the AMC module only requires either I²C or SPI interfaces, no additional programmable I/O circuits are required. As in the first generation module, the FPGA implements interfaces to these serial protocols and provides full access to the in-crate CPU, as will be described later in this chapter.

9.2 Generation of the timing data stream

The timing data stream incorporates many different information types like synchronization information, unique identifiers, events, information about the bunch structure, beam parameters, sections to be passed, machine protection information etc. Most of the data are subject to change very quickly. Therefore the assembly of the data stream is implemented within the FPGA and based on user defined values transmitted via the in-crate CPU and also on internal variables.

The user defined values are set-up and might include calculations on software side and is then provided to the FPGA firmware by loading the data into a memory block within the FPGA via the PCIe communication. Synchronized to an external or internal trigger and the reference clock, the timing stream is generated and provided to the transmitter section.

For transmission of the timing data special comma characters are inserted in order to define synchronization information, idle times and start of messages. Also check sums in CRC8 format are calculated and added to the data stream. Finally the data words are serialized and encoded in 8b10b encoding¹. The serialization and 8b10b encoding are implemented in a special hardware block within the FPGA, which is called GTP (Gigabit Transceiver). A complete description of this component is provided in the respective user guides [47], [48] and [49]. The GTP is a complex module, which provides many configuration parameters. Important for the timing system implementation is to maintain a full deterministic synchronization of the timing data stream related to the reference clock and the externally or internally provided trigger to start the transmission. In standard configuration of the GTP different buffers and FIFOs are used in order to provide a flexible interfacing to the GTP from FPGA user modules with relaxed synchronization requirements. In the timing system firmware implementation intensive studies and tests of the GTP module were performed in order to remove all of these non-deterministic buffers, FIFOs and phase uncertainties.

9.3 Control loop for drift compensation

The drift compensation, as described in previous chapters, consists of phase detectors and adjustable delay elements. However, in order to implement the drift compensation, a closed control loop has to be implemented in order to adjust the delay according to measured phase changes. Also here the implementation differs between the first and second generation timing system module. In case of the first generation, the control loop was implemented on a soft core processor (Micro Blaze) within the FPGA. The focus of the implementation was verifying, that the drift compensation can be implemented, and measuring the performance. The design was not integrated in the complete firmware of the first generation timing board. The main reason is, that the control loop has been moved to a micro controller for the second generation version and therefore the Micro Blaze solution is no longer required and the high amount of work for integration was not justified. The first generation module can still be used as timing receiver or stand-alone module without the drift compensation implementation.

The implementation of the closed loop control is based on multiple control loops. The first level is a generic PID controller² as described in previous chapters.

The phase difference between the reference clock and the clock derived from the received timing

¹8b10b encoding defines a technique, where words of 8 bits will be encoded into 10 bit words based on a fixed conversion rule. Two important goals of this conversion are, that the resulting serial data stream is DC balanced (over a longer time frame the number of logical ones is equal to the number of logical zeros) and special comma words are unique in the stream (e.g. can never be generated by a combination of any other data words) and therefore provides an indicator for word boundaries in the stream via comma detection.

²Proportional, Integral and Differential controller. More information about control theory basics and principles to determine the respective parameters can be found in [43] and [42].

data stream (from the connected receiver module) is measured with a phase comparator. The output voltage (see also Figure 4.6 on page 39) is converted into a digital signal and used as the input (phase error) to the PID controller (a set point of -90 deg is assumed). The PID controller will calculate a digital output value, which will be converted into an analogue signal and applied to the fine delay input of both adjustable delay circuits. A change on this input will introduce a delay change, which will cause a phase change of the received clock compared to the reference clock, which is again the input value for the control loop.

A very important aspect of this control loop has to be pointed out: the system is not linear over the full range of parameters! As the reference and recovered clocks are periodic with the same frequency, also the measured phase error will show periodicity if the phase difference becomes larger than 2π . Taking again a look at Figure 4.6 on page 39, two other difficulties can be seen: (1) the phase detector output voltage is not linear between -40 and $+40$ as well as between 150 and 180 deg, both on the positive and negative sides. And (2) the output voltage range for phase differences between -180 and 0 deg is the same but with inverted slope for phase differences between 0 and 180 deg. These two aspects lead to the consequence, that the operating point should be at either $-$ or $+$ 90 deg and the range should be limited to ± 60 deg around this working point in order to act on a linear range. Another challenge is the actuator in the control loop: the fine delay adjustment of the delay circuit. Taking a look to Figure 4.8(b) on page 40, the fine tuning input of the delay circuit shows a non-linear and range limited input to output behavior. While the integrator part of the controller will remove steady state errors, care has to be taken, that the controller output will stay within the operational range of the fine tuning delay input ($0V$ to $1.5V$). Assuming that the system is in the desired operating point (e.g. -90 deg phase difference and in the middle of the fine delay range by adjusting the digital delay input), the PID parameters can be found by different methods like system identification, numerically or analytically by using the data sheet information or by using heuristic methods like that from Ziegler-Nichols. However, if the link delay becomes larger, than the fine tuning of the delays can handle, then the coarse delay has to be readjusted. This is implemented in a second control loop.

The task of the second control loop is to track the output value of the PID control loop. If a value close to the maximum or minimum has been reached, the loop will readjust the coarse delay of both delay circuits in a way, that the new value brings the fine tuning back to its center. This principle is simple to implement, however, adjusting the coarse delay during signal transmission causes an interruption of the signal for a time window defined by the switched delay taps during the delay adjustment (see the block diagram of the delay circuit in Figure 4.7 on page 40). Therefore the digital delay adjustment has to be synchronized to the data transmission. The procedure chosen in the implementation is as follows: The European XFEL time structure has a periodicity of $10Hz$ (optionally $25Hz$), where a synchronization to the $50Hz$ voltage oscillations of the power line is implemented. In the same $10Hz$ periodicity the fine delay will be checked and the time gap before the synchronization is used to adjust the coarse delay if required. Missing data packets are not a problem as there are only idle frames transmitted during that time anyway. The possible short interruption of the signal transmission (up to $10ns$ in the worst case) are no problem for the clock recovery and will be filtered by the PLL loop filters. The coarse adjustment is implemented in the Atmel AVR micro controller as well. The synchronization between the data stream and the digital delay adjustment is signalled by the FPGA on a dedicated interrupt line to the micro controller.

Finally a third control loop is implemented in the drift compensation system. This loop is concerned with the symmetry of the link delay. Assuming, that the two delay components are not completely identical in overall delay at the same input configuration, this asymmetry can be measured with the two phase detectors across each delay circuit (see block diagram in Figure 8.4 on page 74). If the delay were identical, the two phase comparators would provide

the same phase difference. If their delay were slightly different (possibly changing over time) the phase comparators provide slightly different output values. The task of the third control loop therefore is to readjust the fine (and maybe even the coarse) delay inputs of both circuits in a way, that the two phase detectors provide the same output.

Also some important aspects of the controller have to be pointed out here: (1) although the set point for this control loop is fixed (the difference between both phase detector outputs and therefore the set point should be zero) the output voltage of the phase detectors can be anywhere in the range of operation. Therefore the previously discussed non-linear parts of the range are included. (2) the comparison of the overall delay of the two delay components is based on comparisons of output values of two phase detectors. Therefore an equal behavior of these two chips has to be assumed (or at least much less different, than the delay behavior of the monitored delay circuits). (3) finally, the phase detection between input and output of each delay circuit can only be measured, when a clock-like signal is seen by the phase detector. In contrast, if a random data signal is transferred through the delay component and the input and output phases are measured by the phase detector, a broad band signal is measured at the output of the phase comparator and no assumption of the delay behavior can be derived from that. Therefore a special data pattern is sent at a defined time during the 10Hz cycle if no other data have to be transmitted. This pattern is valid in the 8b10b encoding and has also the property of being clock-like, so that the phase detectors provide a low-noise meaningful output signal. The same interrupt signal as described above will be used, to inform the micro controller about the time, when this pattern is visible.

9.4 Synchronization to the timing data stream

An important feature of the timing receiver is to synchronize clock and trigger phases to the central transmitter and therefore to the machine. In order to accomplish that, the serial receiver and protocol decoder within the FPGA have to be fully deterministic. However, that is not the case in the usual configuration, as reliable data transfer and clock separation for easier system design is the main purpose of this type of module. This includes internal PLLs, ring buffers, FIFOs and different clock domain crossings. Therefore a detailed analysis of the serial gigabit transceiver (GTP) modules and their configuration was required (see Figure 9.1). The solution consists of three steps:(1) the user clock and the recovered clock have to be identical (the two regions shown on the right side in Figure 9.1 denoted as RXRECCLK and RXUSRCLK). (2) The buffers and fifos in the chain have to be disabled and (3) all automatic alignment and synchronization features have to be disabled as well. This provides a deterministic timing of the GTP module. However, as all automatic features are disabled, manual detection and alignment had to be implemented in user logic. Due to that, accurate and deterministic measurement of the phase up to 1.3GHz clock cycles is possible.

Besides the synchronization of the FPGA internal logic for trigger, clock and data, a paramount task is the synchronization of the FPGA external divider components of the high-precision clock distribution. The dividers have to be synchronized by a reset, which has to be accurately timed within the correct 1.3GHz clock cycle. In order to accomplish that, two other important detection and synchronization features had to be developed: (1) providing a high-precision and low-jitter reset pulse to the dividers and (2) adjusting the reset pulse phase to the clock phase. The most accurate output of the FPGA is a GTP, which has been used to generate the reset pulse and it has been configured to be deterministic analog to the receiving part described above. The adjustment of the phase of the reset pulse can be performed by phase shifting the reference clock of the GTP, which is provided by the divider chip (which is therefore already locked to the 1.3GHz reference). There are two important aspects of this reference clock to be mentioned: it is not affected by any reset pulse (can be configured in that way in the divider

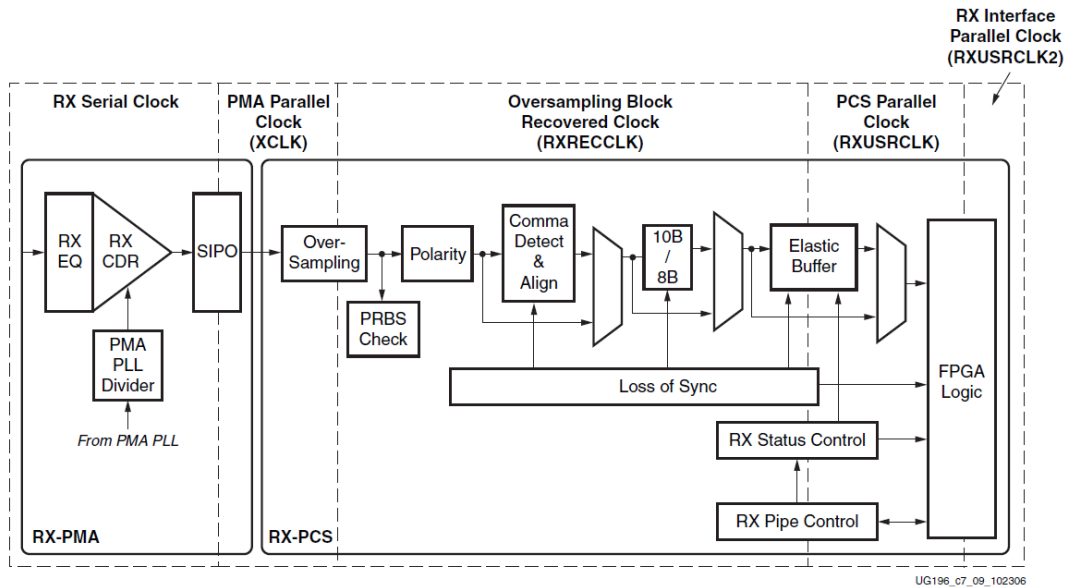


Figure 9.1: This picture shows a block diagram of the high-speed serial transceiver (GTP) of the Xilinx Virtex 5 FPGA. The left side shows the serial interface to the FPGA pins, and on the right the parallel interface within the FPGA is depicted. In between different sections will be passed during clock recovery, serial-to-parallel conversion, comma detection, and interfacing to the user clock domain. (source: Xilinx Virtex 5 GTP user guide [47])

chip) and it can be phase shifted by the divider component (an internal programmable delay), which is the solution to adjust the phase in the correct way (e.g. that the reset pulse edge appears in the middle of the correct 1.3GHz clock cycle).

9.5 Decoding of the timing data stream

Decoding of the timing data stream is required in order to take actions based on the packets received. The implementation is based on a finite state machine, which detects a start of frame (SOF) encoded as a unique comma character, which is followed by a length-word, which defines the number of words to be expected before the CRC is expected. After the length word, a command code word is received, which is the basis for the action to be taken. Depending on the command, a certain number of data bytes will be received and finally the full packet is checked for a CRC error.

An important packet is the SYNCH frame. It initiates a resynchronization and ensures always a correct absolute alignment of derived clock phases and triggers while taking link length and additional local cable lengths and positions into account. It synchronizes the phases of the external (outside of the FPGA) clock dividers via dedicated digital outputs connected to the synchronization input of clock divider circuits and resets the divider counter within the FPGA (see above). The internal counter is used to generate triggers based on scheduled events (see below).

Received events will be forwarded to the trigger units (see below) and, if this event has been registered, the trigger logic prepared.

Complex data received as tables (e.g. bunch pattern) are stored in pre-defined memories or registers for later use by dedicated modules.

9.6 Trigger generation

Two types of triggers are supported by the timing system implementation: (1) scheduled triggers based on events and (2) immediate triggers. The basis for the first class of triggers are events, which are defined in the software of the timing master transmitter. Those events are defined by an event number and a time stamp, when the event will be valid (relative to the synchronization time every 10Hz). There can be many events defined and all these tuples are transmitted in the timing data stream to all timing receivers. Within the timing receivers the timing data stream will be decoded as described above and the received events evaluated. Via software a user can configure trigger outputs for each timing receiver module. If a trigger is defined to be a scheduled trigger, the related event number has to be set. If this event is received by the receiver module, it will compare the attached time stamp continuously with the internal counter (see the synchronization section above). If the values are identical, the event becomes valid and the configured trigger will be started.

Alternatively, a trigger can be defined to be an immediate trigger. In this case it will also be assigned to an event, but this as time an immediate event. Immediate events are sent by the master transmitters as well via the timing data stream. However, the event will have no assigned time stamp and is therefore valid as soon, as it has been received by the receiver module. The main purpose of this type of trigger is to allow asynchronous triggers. They are not pre-defined via software of the master transmitter, but are implemented by external trigger inputs to the master transmitter. One source of such signals can be an interface from another timing system to transfer events to the XFEL timing system or from the machine protection system, but basically any asynchronous system can be used as input if this makes sense.

When configuring the trigger outputs at the timing receiver modules, further parameters can be defined by the user. This includes an additional delay, before the trigger is raised, and the width of the trigger pulse. Further more triggers can be defined as gates, where two trigger definitions can be used in order to define the beginning and end of a gate signal. It is also possible to define a bunch trigger, where a trigger for each bunch - which is configured in the machine and transmitted in the timing stream as larger tables - is raised. This can even be combined with a mask, in order to only select a subset of bunches if desired by the user.

9.7 Transmission of deterministic data to local receiving systems

Besides clocks and triggers to be provided at outputs of the timing receiver, also the transmission of deterministic data is of importance. These data are a subset of all the information transmitted in the timing data stream. Besides the timing modules also other systems (usually including FPGAs or controllers) can make use of the information transmitted. For example each system can include the macro pulse number (also known as general event number or train number) in all produced and further transmitted or saved data to uniquely identify the relation to other measured data later on. Or some systems can make use of the bunch pattern and its parameters in order to make assumptions about pulse energies, if pulses are present or which path the bunches will take through the machine. Also the shot ID or pattern index would be useful for some systems in order to switch internal tables, if adaptive algorithms are used or different parameter sets are defined for different bunch patterns.

Therefore two ways of transmission of this deterministic data have been implemented in the firmware. The first way has been designed for in-crate communication and uses the M-LVDS bus lines of the MicroTCA.4 backplane (usually the first two). The first line is supposed to carry a 108.3MHz clock, which is the most often used clock with fixed relation to the 1.3GHz reference clock. The second bus line will then carry - synchronous to the 108.3MHz clock - the serial data stream. The information in the data stream is almost identical to the original timing

data stream. However, no 8b10b encoding, but a UART based coding of the 8 bit words has been chosen. As the bit rate is roughly a factor of 12 slower than on the timing data stream, the time required to transmit all information takes longer (approximately 10ms), but has been completed significantly in advance of the first bunch.

The second way of transmission is based on an UART communication at 115.2kBaud compatible to standard UART based serial interfaces. Due to the significantly reduced data rate, only limited information will be transmitted via this interface (e.g. no table and event information). Consumers of this interface (among others) are Beckhoff PLC systems and Klystron control units. Slightly different protocols have been defined and implemented in the firmware in order to simplify the receiving and decoding in the receiver of this data as well as to ensure correct synchronization to the data stream.

9.8 Communication with in-crate CPU via PCIe

The communication interface between software and firmware is implemented via PCI Express (although the hardware would also support a gigabit Ethernet interface). At the European XFEL the standard communication protocol for in-crate communication on the Fat-Pipe channels (Port 4 to 7) has been chosen to be PCI Express (PCIe). These channels connect the AMC modules via the PCIe switch in the MCH. The first generation timing module supports up to four lanes of PCIe Gen 1. The chosen Spartan 6 FPGA chosen for the second generation module supports only one lane of PCIe Gen 1, which still satisfies the required bandwidth requirements.

The implementation on the FPGA side is based on a hardware and IP wrapper module implementing the PCIe endpoint, which provides a low level interface to the PCIe communication. Based on that a bridge to an FPGA internal parallel bus system has been developed by DESY colleagues and integrated in the design. A bus system provides a simple way to attach different modules in the FPGA to this interface and communicates with them from the software side (see next chapter).

Besides the different modules connected to the bus, the interface includes also different registers, which are standardized among different FPGA projects and are called *Standard Register Set*. They include information about the project, the version and last date of change to be read out by the driver and related software to ensure compatibility to the module.

Another aspect of the PCIe communication with the in-crate CPU is the signalling of asynchronous events via interrupts. In the same way, as triggers are defined and generated, this principle is also used to send interrupts to the CPU module. This way ensures, that the latency between the issue of the trigger related interrupt and the reaction from the CPU is as short as possible. In this way not only hardware modules can be triggered by the timing system, but also software can be triggered and processes evaluated synchronized to the timing system. Related to the interrupts important parameters can be read from the module (e.g. macro pulse number, bunch pattern, etc.).

9.9 Synchronization between Modules

The previous descriptions related to synchronization were mainly focused on synchronizing a single receiver module to the connected master through the received timing data stream. In this section the synchronization among all modules across the European XFEL will be described as well as the procedure, how the modules have to be registered to the transmitter output channels in order to identify and address the related modules.

In order to implement a system wide absolute time synchronization including all timing mod-

ules, a knowledge of all link delays in the system is required. If this is known, the maximum link delay time can be taken as reference and all other systems delayed in order to match the time with that module. In reality the delay will even be set to a higher value than the longest link delay and is rather defined by the time of the first scheduled trigger. In that way a system wide absolute time synchronization would be possible.

As described earlier, the transmitter sends the timing data stream to all connected receivers, where the signal is returned to the transmitter. This signal can be connected to the FPGA on the transmitter side and the FPGA is able to measure the overall link delay (from transmitter to receiver and from receiver to transmitter). If this value is divided by two, this yields the link delay to the connected receiver. If a second receiver is connected, the input to the FPGA on the transmitter side can be switched to the second input and the link delay measured in the same way.

As the timing data stream is just created once and transmitted to all receivers in the very same way, there is no way to delay the data stream to individual receivers depending on their link delay and the desired system delay. Therefore the delay has to be implemented on the receiver side (as it has been described previously). But then the question is, how to inform each receiver about the measured link delay. As the data stream is just returned by the receiver and no additional information has been included by the receiver, the transmitter has no information about the receiver module.

One possible solution would be the following: if a receiver module has just been connected to a transmitter (or one of the two sides of the communication channels are enabled after a restart) it is detected by both FPGAs (on both ends). In this case the receiver is not just returning the data stream it receives, but is sending directly to the transmitter a registration request including its own identification (e.g. the network or control system address of the related software of this module). If the transmitter detects this registration request, it saves this information in a table. The transmitter software will inform the receiver software through the now known address of the receiver, that it has registered at a transmitter and tells it also its own address. In this procedure the transmitter will later also inform the receiver about its link delay and other possible information like the signal path from the master transmitter down to the receiver module and so on. If the transmitter is not the master transmitter, it will also inform the higher level(s) of transmitters about the new receiver and the respective delay, so that the full tree information is known at the master transmitter.

With this information available at the master transmitter, the longest link delay is known and can be used to verify, that the time of the first scheduled trigger is late enough to allow a system wide synchronization (if required). In any case the internal counters and therefore the system clock can be system wide synchronized to a level defined by the link delay measurement resolution and accuracy and the number of hierarchy levels.

9.10 Remote Firmware Upgrade

When the modules are installed at their destinations in the tunnels, shaft buildings, rooms and hutches and a firmware upgrade is available, it is unacceptable to collect the modules and reprogram the EEPROMS or FLASH memories on a laboratory desk. Therefore remote firmware upgrade has been implemented and will be described briefly.

The firmware for the FPGAs is located in a SPI interfaced FLASH memory on the AMC module. Two ways to remotely upload a new firmware exist: (1) via IPMI and the Hardware Platform Management (HPM) protocol through the MCH and MMC or (2) via PCIe through the FPGA.

In the first case the firmware image will be uploaded via a standard HPM 1.0 protocol, which sends the image over Ethernet to the MCH, where it is forwarded to the MMC on the AMC

module. The MMC will then reprogram the SPI FLASH memory. As the communication between the MCH and MMC is relatively slow, the upload might take many minutes, which should not be a problem in most cases, as the module is still operational while the upload is taking place.

In the second case, the firmware can be uploaded with a special software, which provides an interfaces to the PCIe driver and the FPGA directly. Within the FPGA a module will program the FLASH memory with the new firmware version. As the communication speed is much higher, the upload takes just some minutes and is limited by the programming speed of the FLASH memory.

Chapter 10

Software

Besides the core Timing System function implementation in the firmware of the FPGA and micro controller, the software running on the in-crate CPU module plays also an important role in the overall functionality. Main tasks of the software are to configure the hardware module, calculate and upload table information like bunch patterns, monitoring of timing system parameters, receiving hardware interrupts from the timing system and distribute them and other accelerator related information to other software running on the same CPU module and much more.

This chapter will provide an overview of the different layers of software - from the driver up to the graphical user interface - and provide a brief introduction into accelerator control system software concepts.

10.1 CPU technology, operating system and interfacing

The CPU module in the MicroTCA crate is - as for the implementation and setup of the European XFEL - based on the 3rd generation of Intel's Core i 64 bit CPUs and chipsets or compatible to these technologies. Therefore standard operating systems like Linux or Windows can be used without special modifications. At the European XFEL the Ubuntu Linux 64bit long-term support (LTS) distribution is be used.

The connection to the Timing System module (located in the same crate as the CPU module) is connected to the CPU via PCIe through the backplane and the MCH. The chipset and operating system will detect the timing module, but in order to communicate with the board, a driver is required.

10.2 Driver

A driver is a piece of software, which resides within the core (kernel) of the operating system and adds low-level functions to it as well as interfaces to user programs which can make use of these functions. The main purposes of the driver for the Timing System are: (1) to establish a communication channel between the operation system and the actual module, (2) to implement time-critical or securing functions related to the hardware communication or configuration and (3) to receive and handle interrupts of the module.

The current implementation of the driver for the Timing System module has been developed at DESY. It is automatically loaded by the operating system, if a Timing System module has been detected and already reads basic information from the hardware like version and supported functionality information.

The access for user programs to the hardware and the detected information is provided through

special files in the file system. The most important file is located in the `/dev`-folder. It is named `x1timersX` or `x2timersX` depending on the used generation of the module. The trailing `sX` denotes the MicroTCA slot number, where the Timing System module is located and the `X` is therefore replaced by the slot number. The communication with the hardware by the driver is accomplished by opening the file and using `read()`, `write()`, `ioctl()` and other special operating system functions within the user program. The different functions expect and provide a defined structure consisting of different information including address, base-address-register-number (BAR), data, data width and more. In this way register oriented reading and writing as well as block based reading and writing and special functions can be performed. In order to access the information, which have been gathered by the driver at loading time, another special file with the same name exists in the `/proc`-folder. It can be read as a regular text file and provides (in human readable format) version, date and type information of the driver and the firmware defined by developed and previously mentioned Standard Register Set.

10.3 Control Systems and device implementation

Besides the driver, a special software program is required in order to configure and monitor the Timing System module and do required calculations and data formatting. In principle, any programming language, which is able to interact with the special driver file to communicate with the hardware can be used. However, in an environment of accelerator and X-ray source for many experimental stations like the European XFEL a standardization is required in order to ensure compatibility, simplify support, provide interfaces to a standardized graphical user interface, remote access and so on. The solution to this is to use a software framework, which implements already many different common functions like network communication for remote access and data transmission, state machines, techniques to start, stop or upgrade user implemented programs, specialized libraries for data processing and handling and much more. Such frameworks and concepts of data handling are called Control System. Unfortunately, there is no such thing as a world wide single Control System. A number of different Control Systems are available, which differ slightly in the underlying concepts and technologies and the time, when they have been designed or upgraded. Therefore a selection has to be made, which will always be a trade-off of different aspects and depends on the requirements and boundary conditions. Already at the European XFEL there will be at least two different Control Systems used: (1) the Distributed Object-Oriented Control System (DOOCS) [50], developed in the 90s at DESY and since then upgraded and integrated with another Control System from DESY called TINE [51], and (2) Karabo, a combination of control system and scientific computing framework under development at the European XFEL GmbH [52]. DOOCS will be mainly used in the electron accelerator part of the European XFEL, while Karabo will be mainly used at the Photon (X-ray) beamlines and at the experiments. However, both systems are aware of each other and a defined interface between them exists in order to provide an European XFEL wide Control System functionality. Further information related to control systems can be found in [53]

As the European XFEL Timing System has been developed within the electron accelerator control system group at DESY, DOOCS is the native control system, where the software for the Timing System module has been written so far. As the Timing System is an integral part of the whole European XFEL, a software version in Karabo will follow in future.

The current version of the software (called device server) implements (besides other functions):

- Configuration of hardware
- Monitoring of parameters

- Calculation and preparation of timing protocol related data
- Distribution of information through the control system network
- Validation and reconfiguration of hardware based on received configuration request via the control system

The program as such does not have an integrated graphical user interface. It rather implements network based interfaces for monitoring and control through the control system.

10.4 Graphical User Interface (GUI)

The graphical user interface denotes an application, which provides the user with a graphical representation of information and allows the user to interactively control the underlying program functions (similar to the front panel of electronic devices, which provides indicators like LEDs or displays and buttons or knobs). In the context of most control systems, the GUI is a part of the control system and provides an interface to the network control interfaces to all device servers within the network accessible range and therefore adds the GUI to the Timing System related server. The latest version of the native GUI for DOOCS is called jDDD (Java DOOCS Data Display) [54]. It can run in two modes: edit and run. In edit mode, the user (or designer) can create customized GUI representation (called Panel), which basically means, that different widgets (like buttons, displays, sliders, boxes, plots, etc.) can be added to an empty screen area and arranged and grouped with graphical elements like boxes, lines and pictures. This allows, that different panels can be designed, that can interact with the very same device server (like the one for the timing system). The run mode will then lock all placed and arranged components on the panel and will activate it, so that the communication with the related device server(s) can start and the monitoring and control can be performed.

For the Timing System different panels have been generated and continuously updated with changes and added functions in firmware and software. The different panels provide different views on the underlying server and are optimized for different usages (like debugging, lab test, in-accelerator-test, etc.) and also organized based on the level of knowledge of the expected user (expert or normal user). In figures 10.1 to 10.4 some example panels for the Timing System are shown with a brief description below the pictures.

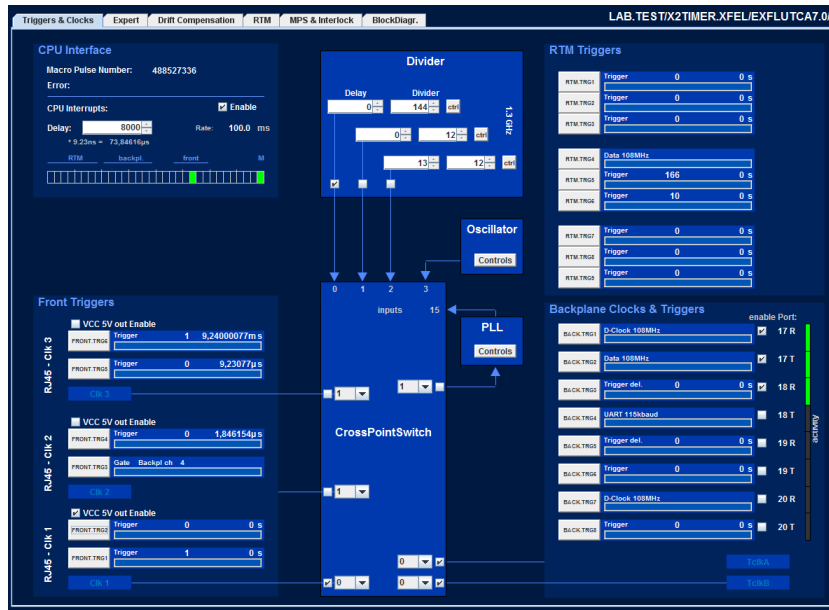


Figure 10.1: Screen shot of the main XFEL Timing System panel. It allows to configure all provided clock, trigger and data outputs. In the center the clock paths and divider can be defined. The lower left configured the front panel RJ45 connectors. The right side the connections to the MLVDS (lower part) and the connections on an optional RTM are configured. The top left areas configures the interrupts to the CPU.

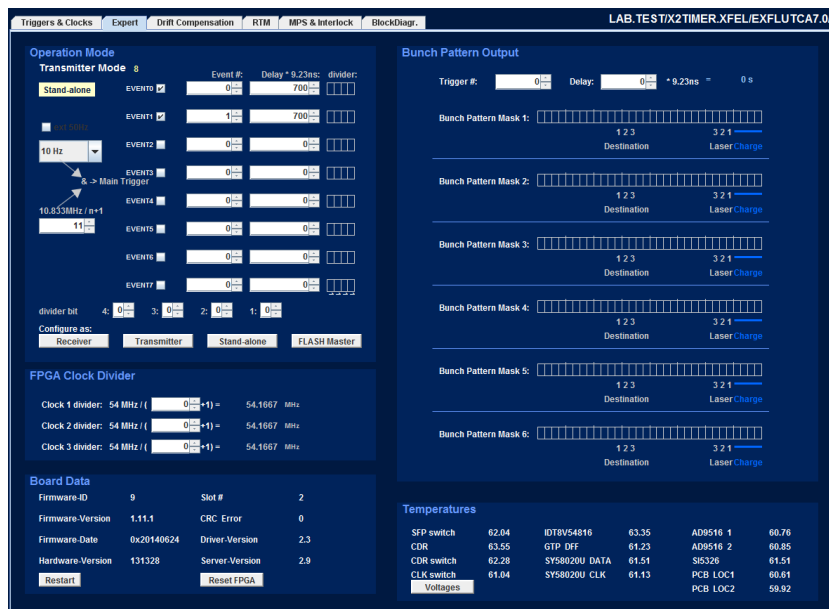


Figure 10.2: This screen shot shows the expert panel of the Timing System. It allows to configure the main operation mode of the module (transmitter, receiver, stand-alone, etc) on the upper left side. The lower left side shows important version information of the firmware, driver and software and related parameters and also allows to configure special clock parameters. The upper right side configures bunch pattern masks and on the lower side temperature values of all sensors on the module are displayed.

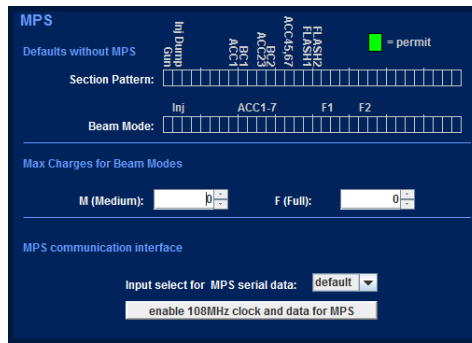


Figure 10.3: This panel shows the configuration and information about the Machine Protection System (MPS), which provides machine critical information to the Timing System through a special low-latency connection. This includes limitation of the number and/or charge of bunches, allowed sections etc.

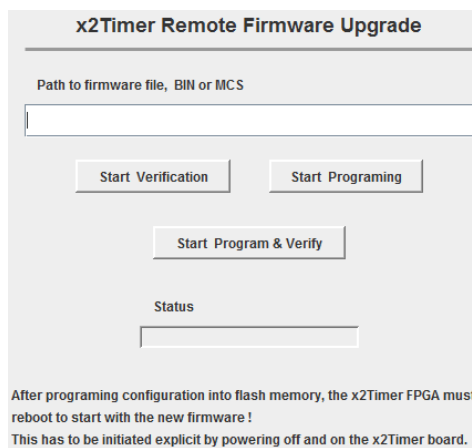


Figure 10.4: This panel provides an interface to the remote firmware upgrade feature of the firmware and software. The firmware image file path has to be provided and then an upgrade and verification procedure can be initiated.

Chapter 11

Interfacing to consumers

How the timing data stream is generated and transmitted to many receiver modules while actively maintaining phase stability from the beginning to the end has been described. Based on that data stream, all clocks, triggers, gates and deterministic information are derived at the timing receiver modules. This chapter will focus on the concrete interfacing between these generated signals at the receiver modules and possible consumers. This includes consumers within and outside of the MicroTCA crate.

11.1 Interfaces within the MicroTCA crate

An overview of the MicroTCA.4 in-crate timing distribution has been given in chapter 6. It includes three main aspects: (1) a configurable and low-jitter clock distribution through the MCH, (2) direct signal distribution to other modules via an eight-channel M-LVDS bus and (3) a PCIe communication channel through the MCH to transfer further data and interrupt messages. All three options will be described in the following paragraphs.

11.1.1 Low-jitter Clock Distribution

11.1.1.1 Distribution Principle

The main clock distribution within a MicroTCA crate is through a central clock switch in the MCH. All module slots provide five differential clock lines defined as Telecommunication Clock A to D (TCLKA-TCLKD) and Fabric Clock (FCLK). FCLK is usually 100MHz and generated by the MCH as reference clock for the common PCIe communication and therefore not usable for special clock distribution from the Timing Module. The TCLKA-TCLKD lines are user definable. They are the preferred way to provide clocks from the timing module to other modules within the crate. While the TCLKA and TCLKB differential lines are connected to the primary MCH, the TCLKC and TCLKD lines are connected to an optional redundant MCH. Therefore the commonly usable, thus more important, lines are the TCLKA and TCLKB.

The clock lines can either be used as inputs or outputs. However, the direction and possible clock distribution possibilities depend on the clock module used within the MCH. This also holds for the quality of the signal distribution in terms of noise, jitter and drift. Common implementations within the MCH are built on FPGA or CPLD type of devices, which are cheap and allow flexible configuration possibilities. However, the signal integrity is not optimal compared to dedicated clock switching circuits. But MCH manufacturing companies are aware of that fact and have already developed products with optimized clock distribution modules. Taking the clock module version 4.1 of an MCH from the company N.A.T.¹ as an example

¹<http://www.nat-europe.com>

(FPGA based switching), all input-to-output combinations between any of the TCLKA and TCLKB lines from all modules are possible. The configuration could be performed in a text file and uploaded via a web interface (or via a program) to the device. A typical configuration would connect the TCLKA and TCLKB outputs from the timing receiver module to the TCLKA and TCLKB inputs of all other modules. But depending on the application also other connections are possible.

11.1.1.2 Possible Clocks

The possible clocks to be distributed via the TCLKA and TCLKB lines depend on the generation of the timing receiver module. While the first generation module connected the two clock outputs through two different clock buffer and divider circuits, the second generation module provides a much more flexible connectivity through the 16-port bi-directional cross-point switch (see also the block diagrams 7.3 on page 66 and 8.3 on page 73). In both cases the two clock outputs could be provided with the lowest-jitter clock from the module generated directly by the clock distribution circuits of the module. In case of the second generation module even signals from the FPGA and a flexible PLL could be used.

11.1.1.3 Properties of the Interface

As mentioned above, the clock distribution via the TCLKA and TCLKB via the MCH provides the best way in terms of maintaining the high clock quality of low-noise and low-jitter as well as minimal phase drift and high maximum frequencies. A reason for that is, that the distribution is point-to-point for each signal path. The output of the timing module (say TCLKA) is only connected to the input of the clock switch in the MCH. This avoids any stubs on the signal paths, which could worsen the signal integrity. It also allows proper termination of the signal at the end of the path (in this case on the clock switch module) and matched drivers. The same is true for the connections from the switch to any receiver module. However, the quality also depends on the implementation of the clock switch within the MCH clock module. Care should be taken in selecting clock modules to provide the best performance.

Besides the MCH, also the MicroTCA backplane plays a role in the clock distribution. The connection between each module and the clock switch in the MCH will be passing through traces on the backplane. At least two aspects have an influence on the clocks distributed: (1) the length of the traces will influence the signal transmission time and therefore phase delay and (2) the routing of the traces could influence the signal integrity in terms of not matched line impedance, cross-talk and bandwidth.

The first aspect could be of great importance, if the traces between the MCH and all modules are not equally long (which is true for most commercially available crates and backplanes). In this case the phase of a clock is dependent on the slot and therefore is different between slots and modules. If this is a problem for a certain application, either a special backplane with matched lengths should be used, special slots selected or solutions within a module developed (e.g. input delays within an FPGA).

In the second case it should be checked, if the signal distortions are introduced by the backplane and - if required - a special backplane can be used. In general a characterization of a planned standard crate for a certain application or project makes sense. Not only for signal traces, but also in terms of grounding and EMI performance.

11.1.2 Signal Distribution on M-LVDS Bus Lines

11.1.2.1 Signal Connectivity

A more direct connection between the timing receiver module and other modules within the same crate is possible through eight Multipoint Low Voltage Differential Signalling (M-LVDS) bus lines. The connectivity is defined in the MTCA.4 specification [46] and implemented on the AMC slot ports 17 to 20 (on RX and TX pairs). In this standard the mentioned ports of each AMC slot are connected to the corresponding port of all other AMC slots (e.g. Port RX17 of all modules is connected, TX17 of all modules is connected, etc.). If one module (e.g. the timing system receiver) is transmitting a trigger on port RX18 (the RX and TX have no meaning in this type of bus standard) all other modules are able to receive the trigger on their RX17 port and use it for triggering.

In most cases the module could be configured, if a certain port is used as an output or just as input. If no module provides an output to a bus line, the line is idle and provides a constant logic "0" to all module inputs. On the other hand, if more than one module is trying to define the logic level on the bus, the dominant logic state "1" has priority over a logic "0". This implies, that if at least one module has an output logic level of "1", the bus line is "1", otherwise "0". This allows an implementation of a wired NOR logic of modules. An application of this is planned for the machine protection system, where it could be used for interlocks.

11.1.2.2 Signal Types and Applications

The timing system receiver module (first and second generation) is connected to all of the eight bus lines via a M-LVDS bus driver circuit and connects the bus to the FPGA. The types of signals to be transmitted on the bus lines are therefore only limited by the FPGA firmware. The main types of signals, which could be selected and configured via software are:

- Triggers
- Gates
- Clocks
- Deterministic Data

The different types had been presented and described in more detail in previous chapters. More important to discuss here are the default assignments of the types on the bus lines. The envisioned default configuration of the eight bus lines is as shown in Table 11.1. The

| Bus line | Default usage |
|----------|---------------------------------------|
| RX17 | Data Clock (108.33MHz) |
| TX17 | Data (108.33Mbps) |
| RX18 | First Bunch Trigger |
| TX18 | Pre-Trigger (30ms before first bunch) |
| RX19 | User Trigger |
| TX19 | User Trigger or Interlock |
| RX20 | User Trigger or Interlock |
| TX20 | User Trigger or Interlock |

Table 11.1: *Default assignment of the eight bus lines. This assignment provides a solution to most applications. However, it can always be reconfigured by the user.*

first two bus lines implement a source synchronous serial data transmission from the timing

receiver module to all listening modules and provides the deterministic information based on the original timing data stream. In this case the second bus line (TX17) carries the serial data bits running at a line speed of 108.33Mbps (representing one data bit every 12 periods of the 1.3GHz reference clock). An optimal phase relation between the clock and data line will be generated by the timing module (roughly half a period distance between the falling edge of the clock and the data transition point). As the signals are routed next to each other, this timing relation will be maintained almost exactly and no setup and hold time violations will be observed. The line encoding follows a UART protocol, where one data word consists of eight data bits with a leading start, trailing stop and even parity bit as shown in the time diagram in Figure 11.1.

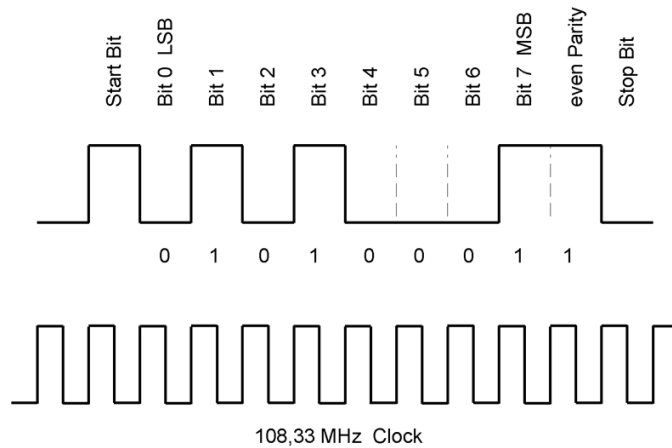


Figure 11.1: Time diagram of one data word transmitted via the M-LVDS bus line on the MicroTCA.4 backplane. The transmission is at a rate of 108.33Mbps ($\frac{1.3GHz}{12}$) and is source synchronous. Therefore a synchronous and phase aligned 108.33MHz clock is provided on a second bus line. On the data line each byte is encoded in an UART compatible frame including a start bit (logic "1"), a parity bit and a stop bit (logic "0"). The idle line carries logic "0".

In order to transfer the different information, multiple bytes have to be combined to a data frame. Table 11.2, 11.3 and 11.4 show the frame structure and the possible values of the fields.

| Length | Command | Data | CRC8 |
|--------|---------|-----------|------|
| 1B | 1B | 0B - 253B | 1B |

Table 11.2: Frame structure of the serial data transmitted over the M-LVDS bus.

| Value | Function |
|-------|--|
| 0 | No command and no data to be sent. Only Length field and CRC8 is sent. |
| N | Length, command, N-1 data bytes and CRC8 are sent. |
| 254 | Length, command, 253 data bytes and CRC8 are sent. Maximum frame size. |
| 255 | Reserved |

Table 11.3: Usage of the length field in the serial protocol over the M-LVDS bus.

| Value | Function |
|-------|--|
| 0 | No command and no data to be sent. Only Length field and CRC8 is sent. |
| 255 | Reserved |

Table 11.4: Available commands and the related data bytes.

11.1.3 Interrupt and Information on PCIe

Besides the TCLK and M-LVDS bus connections and described interfaces, the PCIe communication channel between the timing module and the in-crate CPU module is used to distribute timing related information. The timing module will generate message signalled interrupts (MSI) [55] sent over PCIe, which can be configured and received by the timing modules driver within the CPU. Based on that, the driver is able to execute pre-defined code related to events of the timing system with a very low latency. Besides executing a special code, the driver can also use inter process communication mechanisms in order to inform other processes about the reception of certain events (interrupts). This provides a convenient method to synchronize software processes to the machine with the CPU.

11.2 RJ45 connectors for external consumers

11.2.0.1 Signal Connectivity

The second generation timing module provides three IEC 60603-7 8P8C modular connectors (commonly denoted as RJ45 sockets) with timing related signals, which can be used to provide timing to external systems like cameras, laser systems, PLCs and many more. The RJ45 connectors are shielded connectors providing four differential signal pairs. This is illustrated in Table 11.5.

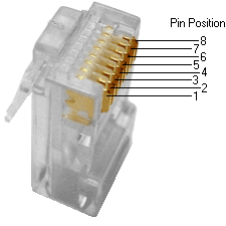
| Pins on plug face | Pin | Pair | Usage |
|---|-----|------|---|
|  | 1 | A+ | High precision Clock |
| | 2 | A- | |
| | 3 | B+ | 5 Volts |
| | 4 | C+ | Trigger, low precision Clock or Serial Data |
| | 5 | C- | |
| | 6 | B- | Ground |
| | 7 | D+ | Trigger or low precision Clock |
| | 8 | D- | |

Table 11.5: Pin out of the three upper RJ45 (8P8C) connectors (the socket pins are reversed). The configurable high precision clock is routed through the clock distribution and therefore offers the lowest jitter and drift possible. The other two outputs are routed through or generated by the FPGA and therefore may carry higher jitter and drifts. The output on pair 4-5 can also be configured to provide the serial data stream. The pair 3-6 provides 5V power with up to 250mA, which can be used by level converters at the end of a connected cable. (source of the picture: www.wikipedia.org)

11.2.0.2 Signal Types and Applications

The connector provides three timing signals in LVDS standard: one high-precision configurable clock output and two outputs for configurable triggers or low precision clocks, as generated

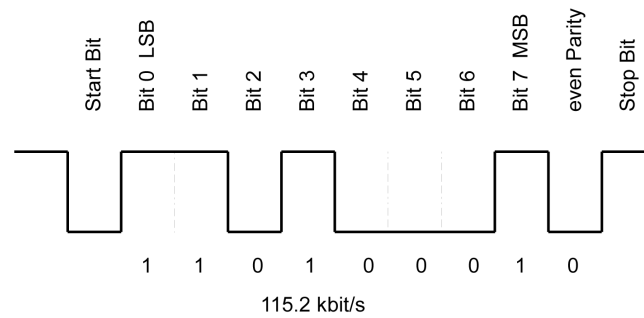


Figure 11.2: Time diagram of one data word transmitted via the RJ45 connector output differential line. The data is transmitted asynchronously at a rate of 115.2kbps (e.g. no additional clock is provided). On the data line each byte is encoded in an UART compatible frame including a start bit (logic "0"), a parity bit and a stop bit (logic "1"). The idle line carries the logic "1". With a level converter, the signal can be converted in RS232 or RS422 signal level and directly received with standard UART receivers like PLCs, PCs etc.

by the FPGA. The latter two can also be configured to carry a serial data stream, which will be described in more detail below. Finally, the connector offers a 5V and ground pair, which provides a power supply with a maximum current of 250mA. This can be used by level converters at the end of a connected cable.

The configuration of the clock and trigger outputs as well as the selection of the data stream is accomplished via the software and the related graphical user interface. While the clock and trigger signals follow exactly the descriptions above, the serial data follows a slightly different pattern and will be described briefly:

Similar to the serial data on the M-LVDS lines, the bytes are also encoded in UART frames as shown in Figure 11.2. As the logic level on idle state is "1" (inverted compared to the M-LVDS based level), the start bit has to be the logic "0" and the stop bit is therefore the logic "1". Except for this, the frame definition is identical to the M-LVDS serial data and is as described in Table 11.2 and 11.3. However, as the bit rate is about a factor of 1000 slower, the data, which can be transferred is significantly reduced. Two special serial data modes are available and can be enabled in the software. These protocols follow special requirements for the klystron controllers and the Beckhoff PLC interfacing at the European XFEL project and include only the macro pulse number and beam mode.

11.2.0.3 Level Converter

The chosen LVDS signal level allows distribution of the different signals over longer distance (up to 300m had been tested with no signal integrity problems - only a voltage drop of the 5V supply line appeared) with minimum signal distortion. Standard modern double shielded CAT7 network cables can be used, which provide good isolation at a low price. Even special cable configurations and a passive splitter could be built or bought from telecommunication equipment suppliers. However, in many cases consumers still use single ended signal levels like TTL or CMOS on coaxial cables with 50 Ohms termination on connectors like BNC, SMA or LEMO. In order to provide clocks and/or triggers for such devices a level converter and connector adapter is required. Such a module has been developed and is shown in Figure 11.3. This compact module is meant to be connected at the end of a RJ45 patch cable. It is supplied by the 5V power from the timing module. A LVDS receiver converts the three differential signals into LVTTTL levels. After that three driver circuits convert the signals into three TTL level

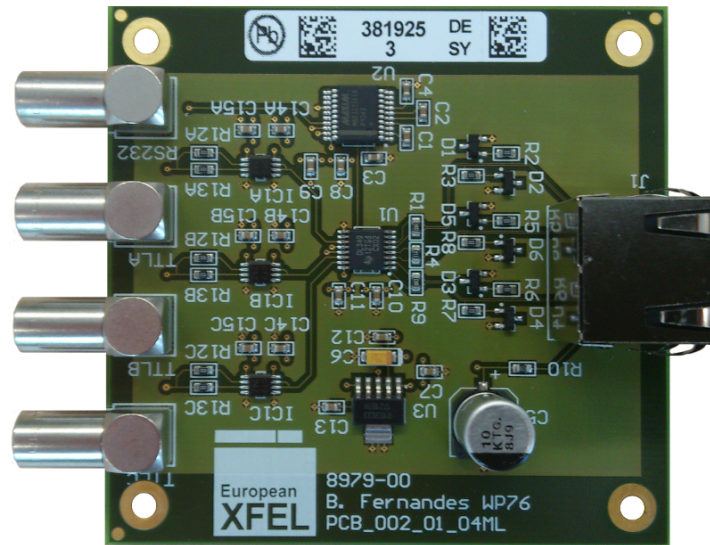


Figure 11.3: Picture of the developed level converter compatible to the RJ45 connector of the second generation timing system module. It converts all three LVDS signals into TTL signals on LEMO connectors. Additionally the middle differential pair is also converted into a RS232 compatible signal level to be connected to serial data receivers like PLC systems.)

signals on LEMO connectors. Additionally, the middle differential signal is converted also to an RS232 conform signal. If the serial data output is enabled, the RS232 signal can be directly connected to the RX pin of an RS232 compatible receiver (i.e. a PLC system).

11.3 RJ45 connector for external trigger inputs

The lowest of the four RJ45 sockets on the timing module front panel has a different pin out and also a different purpose and is depicted in Table 11.6.

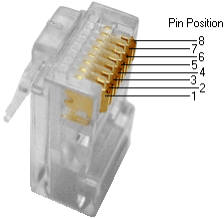
| Pins on plug face | Pin | Pair | Usage |
|---|-----|------|---------------|
|  | 1 | A+ | FPGA in/out 1 |
| | 2 | A- | |
| | 3 | B+ | FPGA in/out 2 |
| | 4 | C+ | FPGA in/out 3 |
| | 5 | C- | |
| | 6 | B- | FPGA in/out 2 |
| | 7 | D+ | FPGA in/out 4 |
| | 8 | D- | |

Table 11.6: Pin out of the lowest RJ45 (8P8C) connector (the socket pins are reversed compared to the connector). All differential pairs are in LVDS standard connected to the FPGA at general purpose I/O pins. Based on configuration, the pins can be input or outputs. This connector will mostly be used for trigger inputs if external hardware triggers are required. (source of the picture: www.wikipedia.org)

All four differential pairs follow the LVDS signal standard and are directly connected to the FPGA. The direction of signalling can be configured via firmware and software. Especially

the input direction is of importance, as no other digital signal is available on the front panel. If used as input, they could serve as external trigger. In the timing transmitter, this trigger is used to synchronize the timing system to the 50Hz power line frequency as described in earlier chapters. Additionally external triggers can be used to generate asynchronous immediate events, which will be transmitted to the receiver modules as soon as a related external trigger has been detected.

11.4 Future options

Besides the described interfaces, there exist further options possible. Related to the RJ45 connectors on the front panel, the one level converter module has been presented. Additional converters could be prepared if other levels (e.g. NIM levels) are required. But the largest flexibility is available via a Rear Transition Module (RTM), which will be inserted into the MicroTCA crate from the back side and connects to the timing receiver through a special RTM connector. As described previously, some RTMs are planned and under construction. Each RTM can provide special interfaces to possible consumers and therefore can increase the number of supported devices and allows to connect even complex future devices, which might not be compatible to current implementations.

Chapter 12

Interfacing with other Timing Systems

The developed and presented timing system for the European XFEL fulfills the requirements shown in Chapter 1 and is the first to implement all timing related aspects of MicroTCA.4. Besides the usage at the European XFEL, this chapter discusses the integration of the timing system at other facilities and the interfacing to their currently used timing systems.

12.1 FLASH Timing System

The European XFEL is built using almost the same technologies as used for the FLASH facility. However, the current timing system at FLASH, which has been briefly described in Chapter 3, is significantly different from the timing system presented in this document. As the requirements for newly developed and installed hardware increased, the current FLASH timing system does not provide sufficient performance.

In order to provide the same timing distribution performance for hardware as for the European XFEL, it has been decided to upgrade the timing system of FLASH, but keep most of the existing timing system in place in order to reduce upgrade time and costs. Therefore both timing systems will coexist and have to be interfaced with each other in order to provide a common timing distribution system.

The complete integration and interfacing solution for FLASH is shown in Figure 12.1 and is described briefly. As the XFEL Timing System is superior in terms of accuracy and stability

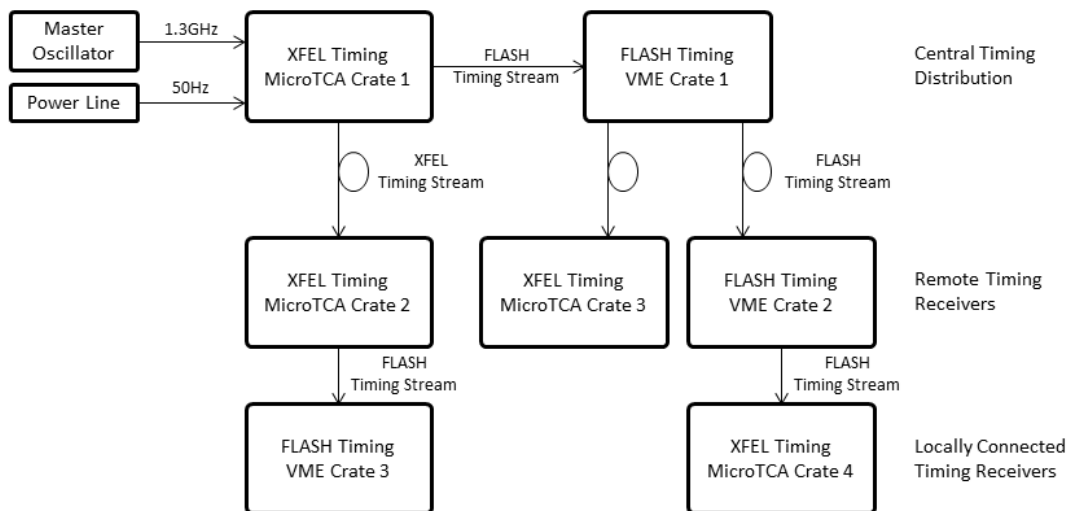


Figure 12.1: Block diagram of the interfacing possibilities between FLASH and XFEL Timing modules in MicroTCA and VME crates.

compared to the FLASH system, it will become the master system also for FLASH (see MicroTCA Crate 1). Within the FPGA of the timing master, all signals are generated as described in previous chapters. Additionally, the FPGA will generate a timing stream fully compatible to the existing FLASH system and provide it on one of the RJ45 LVDS outputs. This will be converted into a TTL signal compatible to the FLASH fan out board on VME standard and directly connected to it (see VME Crate 1). All connected timing receivers following the old standard will not see any difference (see VME Crate 2). This implementation also allows local sub-distribution, where the XFEL timing receiver board (see MicroTCA Crate 2) provides a FLASH conform timing signal for neighboring VME crates (see VME Crate 3).

Besides the replacement of the timing master, there are also applications, where new MicroTCA based hardware will be installed along the machine or replace some older systems, where old timing system endpoints are available. If those systems do not require the higher stability and other features provided by the XFEL Timing System, it is easier and also cheaper to reuse the existing FLASH timing signals. In order to do this, the XFEL Timing System receiver board will be connected to the TTL level FLASH timing stream via a TTL to LVDS adapter to the input RJ45 socket (see MicroTCA Crate 3 or MicroTCA Crate 4 in Figure 12.1). The FPGA will then be switched into FLASH receiver mode and then provides only less performance clocks and triggers and limited information on the bus lines.

12.2 Micro Research Finland

The timing system from Micro Research Finland (MRF) has been described briefly in chapter 3. As mentioned, SLAC is one of the users of this timing system. As they are developing and now using also hardware based on MicroTCA.4, they are looking for an compatible AMC, which provides the timing signals within the crate. Unfortunately Micro Research Finland does not offer an AMC module for their timing system. A possible candidate is the European XFEL timing system receiver board, as it is an AMC, which supports all MicroTCA.4 timing interfacing signals within the crate and is also very similar from the timing stream point of view. In this case the XFEL timing receiver module could be directly connected to the data stream via a SFP module, as any other native EVR from Micro Research. However, as the timing data stream and bit rate are different from the XFEL timing stream, the on board signal distribution and FPGA programming are significantly different.

In the XFEL Timing System configuration, the clock recovered from the data stream will directly take the path through dedicated clock buffers and dividers and the dividers will be synchronized with the FPGA. This mechanism would only be possible for the MRF case, if the bit rate is below 1.6GBPS for the second generation and below 1.3GBPS for the first generation timing board. This is caused by the clock buffer and divider chips used in the two versions and also limited by the achievable accuracy and stability of the divider synchronization by the FPGA. The preferred way of clock recovery and distribution is therefore through the FPGA and then cleaning through the PLL in the clock buffer and divider chip. As the implementation in the event receivers (EVR) from MRF follows a very similar way, comparable results should be achievable. The following aspects have to be programmed in the FPGA: decoding of the data stream, trigger generation and clock generation conform to the MRF defined data stream. However, the firmware for that interfacing functionality has not been implemented yet.

12.3 White Rabbit

The White Rabbit project, also described in chapter 3, uses Gigabit Ethernet for timing distribution. It requires a special switch in order to provide full performance. The reference

switch design, commercialized by Seven Solutions, is based on a MicroTCA MCH-type module [56]. This would allow a conform White Rabbit solution within the crate including all modules supporting Gigabit Ethernet via the backplane and compatible to the White Rabbit Core implementation. However, until now no MicroTCA based full MCH integration seems to exist.

However, there might be scenarios, where a MicroTCA.4 system should be integrated into a White Rabbit based synchronization system. This could be achieved by directly connecting the XFEL Timing Receiver module with a White Rabbit switch. In this case the European XFEL Timing Receiver would act as a White Rabbit node and implements the White Rabbit Core within the FPGA. The 125MHz reference frequency can either be generated directly with the FPGA or via the PLL within the clock buffer chip in the clock distribution on the board. The timing module then distributes the time stamp information on the bus lines within the crate as described in chapter 11. The module would also allow to distribute the synchronized 125MHz clock within the crate via the MCH on TCLKA or TCLKB. However, the firmware for that interfacing functionality has not yet been implemented.

12.4 Bunch clock distribution systems

If a MicroTCA.4 system with the European XFEL timing system receiver should be integrated into an existing bunch clock distribution system at other facilities, this can be performed in two different ways: (1) the reference RF clock of the facility can be connected to the RF reference input of the module. Internal clock divider and delay components of the dedicated chips as well as FPGA functions can be used to divide and generate clocks and triggers synchronously to the bunches in the machine. The optional orbit clock or trigger input (connected via LVDS converter to one of the inputs of the lowest RJ45 connector) allow synchronization of the clock dividers and therefore an easier calibration. All signals will then be provided to the other modules within the crate. The alternative (2) assumes, that the bunch clock will be directly provided to the LVDS converter, which is connected to the lowest RJ45 input, or to the SMA clock input of the timing board. Also this clock, together with signals generated within the FPGA, can then be distributed within the crate.

Besides the described signals, the timing board can generate time stamps based on the CPU system time and distribute them within the crate.

Chapter 13

Conclusion

In summary, based on clock distribution, synchronization and sequencing requirements for the XFEL and FLASH a technical solution called XFEL Timing System had been developed. In this process a conceptual design has been worked out, critical components identified, candidates selected and their performance evaluated. The development of an evaluation board allowed a detailed evaluation and analysis of the most important conceptual aspects of the XFEL Timing System design. Before a definition of the complete design of the first generation timing module was possible, an intensive investigation, discussion and standardization process of the destination hardware platform MicroTCA was required in order to provide the necessary distribution channels for accurate and low-noise timing related signals within the MicroTCA crates. In this process, the new MicroTCA.4 standard (optimized for physics applications) has been defined within the PICMG consortium a group of industrial companies and research laboratories. Based on that standard and the experience from previous evaluations, the first generation timing module has been designed and produced. As important as the actual hardware was the design and implementation of the required protocols, firmware and software. Intensive tests and measurements showed, that the module is working properly and first production units had been used as stand-alone timing modules in different places since then. However, it also revealed elements to be improved and fields to be optimized in terms of function, performance and cost efficiency. The result was the second generation timing module and first prototypes became available beginning in 2013. The new design follows a concept of modularity, simplification and optimization of re-programmable logic elements to the actual demands identified during the firmware developments of the first generation module. Measurements and tests confirmed a successful implementation and the first series production had been received to be installed in the extension of the FLASH project within 2013/2014 and to be used for European XFEL developments. Besides further firmware, software and add-on hardware developments the final step in the process is the on-going industrialization process of the XFEL Timing System hardware to provide a commercially available product for easier and possibly even more cost efficient sourcing for the European XFEL, FLASH and other interested and potential customers world-wide.

Besides the in detail described technical functions and achievements, there are two aspects worth mentioning. The development of the XFEL Timing System involved many different disciplines like high-frequency electronic design and measurement techniques, high-speed serial data communication and synchronization, low-noise and high-speed printed circuit board design, FPGA firmware design, diver and software programming and more. Furthermore, the time is limited until a stable and production ready integrated solution has to be available in order to install the system in the European XFEL and FLASH and to be available as reference module for tests of other interfacing subsystems. Therefore a group of people was directly involved in the project located at DESY and the Stockholm University. Only due to their

combined effort, skills and discipline of communication, documentation and participation in regular meetings the project was successful in that limited time. Also the impressive active collaboration effort and participation within the working group in the PICMG consortium taking care on the development of the MicroTCA.4 standardization is important to be mentioned. Weekly telephone conferences and yearly workshops brought people together from America, Europe and Asia. The work within that group in terms of developing complementary ideas, simulating, designing and evaluating concepts and producing a new standard was an important and successful process. Since then an on-going close collaboration in the development of new products and industrialization based on that new standard had been started as well as further developments on firmware, software and protocol guide lines. All this is very important, when a transition to a new standard and technology takes place and a significant amount of new hardware and knowledge is required. Based on the observed continuous increase of interest from new users and the increasing number of producing companies and products, we seem to be on a good track.

Additionally to the previously mentioned very positive experience of the collaboration within the project, the standardization of MicroTCA.4 and the resulting influence on the success of this project, there is at least one other lesson learned. This is, that in complex projects with many stake holders not all aspects of requirements could be specified at the beginning, but that some aspects develop over time in an iterative process. As a consequence, conceptual designs, planned functions and limitations should be written down, communicated and feedback collected as early as possible to avoid creating artificial critical paths related to possible redesigns at a late stage.

Finally, the next steps related to the XFEL Timing System project should be outlined. The next most important milestone is the ongoing installation and commissioning of the timing system at FLASH2, the extension of the FLASH machine. This involves the procurement, setup and integration of XFEL Timing System modules and MicroTCA crates as well as bringing the firmware and software into the official release state. Besides that, the industrialization of the hardware and possible firmware is ongoing and expected to be finished within 2014. This process includes redesign steps related to the design tools and in order to optimize the production. In 2014/2015 the procurement and setup of European XFEL hardware will be started, followed by first installations as well as further upgrade of FLASH and incremental replacement of the old timing system. In 2015/2016 the complete installation and commissioning for European XFEL is planned in order to prepare the start of user operation.

Chapter 14

Acknowledgments

Many people contributed directly or indirectly to this project and to my personal development. I would like to take the time to pay credit and thank them.

I would like to start thanking Kay Rehlich, who used to be one of my supervisors during my work at DESY and who assigned me the task of the first conceptual design of the European XFEL Timing System and being officially responsible for the project. Since then he always supported me and we have a close relation with many interesting and inspiring discussions which also influenced my personal development. I also would like to thank Holger Schlarb, who used to be my other supervisor during my time working at DESY. He gave me the chance to take over responsibility for the firmware and software development and related coordination in the group of optical synchronization and special diagnostics as well as being responsible for the participation in an EU FP7 collaboration project. That allowed me to gain a much deeper understanding of the physics and timing related requirements of free-electron lasers, coordination and international collaboration. Thank also goes to Christopher Youngman, Markus Kuster and Andreas Schwarz and their trust and confidence in me to set up and take over the joint DAQ and detector electronics group and who allowed and supported me to continue my work on the XFEL Timing System. Furthermore I would like to thank my doctoral thesis supervisor Klaus Schünemann, who allowed me to start this project and supported me since then. I also thank Friedrich Mayer-Lindenberg for being in the examination committee of this dissertation, for the collaboration between his working group and my group at the European XFEL, and finally his lectures related to FPGAs and DSPs, which became an important part of my current work. I also like to thank the remaining professors Sybille Schupp, Wolfgang Krautschneider and Arne Jacob, for being in the examination committee.

Great thank also goes to the colleagues and collaborators directly involved in the project. Besides Kay Rehlich these are in alphabetical order: Arthur Aghababyan, who is implementing and supporting the software, Kai-Erik Ballak, who participates in the testing, evaluation and measurement of the timing modules, Christian Bohm, who is the responsible person within the collaboration (European XFEL in-kind contribution) on the side of the Stockholm University, Bruno Fernandes, who implemented the level converter adapter in order to provide timing related signals to consumers outside of the MircoTCA crate and develops FPGA firmware modules to interface with the timing system within the crate, Attila Hidvegi, who is responsible for the schematic and PCB design of the timing and evaluation modules and also participates in the firmware and software development, measurements and evaluation, Holger Kay, who is implementing the firmware for the FPGA, Vahan Petrosyan, who is responsible for the firmware of the used micro controllers, Lyudvig Petrosyan, who is writing and supporting the linux driver for the timing system, Gevorg Petrosyan, who is supporting the firmware development of the FPGA and Christoph Stechmann, who is taking part in the measurements, tests and installation.

Besides this team, I would also like to thank all stake holders of the XFEL Timing System, who provided valuable feedback and suggestions for improvements. One of these persons is Matthias Werner, who not only identified critical issues, but also provided input and ideas for possible alternative solutions.

I also like to thank the members of the PICMG xTCA for Physics working group, who defined the new MicroTCA.4 standard, which was required in order to provide standard ways to distribute timing related signals within the MicroTCA crate and where industry and research laboratories could build on. Here I would like to single out Ray Larsen as the chair of the hardware committee and a person who provided good ideas and inspired me.

There are certainly more people, who contributed to this project, who I have not mentioned. I would like to thank all of them for their support and provided work, which finally lead to this successful implementation of the XFEL Timing System.

Finally I would like to thank my beloved wife and children for all that support and time they allowed me to work on this project and not being able to participate in the family activities.

List of Abbreviations

A

| | |
|------|---|
| ADC | Analog Digital Converter |
| AMC | Advanced Mezzanine Card |
| ATCA | Advanced Telecommunication Computing Architecture |
| AWGN | Additive White Gaussian Noise |

B

| | |
|-----|----------------|
| BER | Bit Error Rate |
|-----|----------------|

C

| | |
|------|--|
| CERN | European Organization for Nuclear Research |
| CML | Current Mode Logic |
| cPCI | Compact Peripheral Component Interconnect |
| CPLD | Complex Programmable Logic Device |

D

| | |
|------|----------------------------------|
| DAC | Digital Analog Converter |
| DESY | Deutsches Elektronen-Synchrotron |

E

| | |
|------|---|
| ESRF | European Synchrotron Radiation Facility |
| EVG | Event Generator |
| EVR | Event Receiver |

F

| | |
|-------|--------------------------------|
| FEC | Forward Error Correction |
| FFT | Fast Fourier Transformation |
| FLASH | Free Electron Laser in Hamburg |
| FPGA | Field Programmable Gate Array |
| F/W | Firmware |

G

| | |
|-----|--------------------------|
| GbE | Gigabit Ethernet |
| Gen | Generation |
| GTP | Gigabit Transceiver |
| GUI | Graphical User Interface |

H

| | |
|-----|----------|
| H/W | Hardware |
|-----|----------|

I

| | |
|--------|-------------------------------------|
| IFFT | Inverse Fast Fourier Transformation |
| ISI | Inter Symbol Interference |
| I-TECH | Instrumentation Technologies |

L

| | |
|------|------------------------------------|
| LAN | Local Area Network |
| LVDS | Low Voltage Differential Signaling |

LIST OF ABBREVIATIONS

M

| | |
|------|--|
| MCH | MicroTCA Carrier Hub |
| MISO | Multiple In Single Out |
| MTCA | Micro Telecommunication Computing Architecture |
| MRF | Micro Research Finland |

N

| | |
|-----|-----------------------|
| NI | National Instruments |
| NTP | Network Time Protocol |

P

| | |
|-------|---|
| PAPR | Peak-to-Average Power Ratio |
| PCB | Printed Circuit Board |
| PCI | Peripheral Component Interconnect |
| PCIe | Peripheral Component Interconnect Express |
| PICMG | PCI Industrial Computer Manufacturer Group (www.picmg.org) |
| PLL | Phase Locked Loop |
| PSOF | Phase Stabilized Optical Fiber |
| PTP | Precision Time Protocol |

R

| | |
|----|-----------------|
| RF | Radio Frequency |
|----|-----------------|

S

| | |
|-------|-----------------------------|
| SER | Symbol Error Rate |
| SFP | Small Form factor Pluggable |
| SNR | Signal-To-Noise Ratio |
| S/W | Software |
| SyncE | Synchronous Ethernet |

T

| | |
|------|-------------------------------|
| TCLK | Telecommunication Clock |
| TCP | Transmission Control Protocol |
| TDC | Time to Digital Converter |
| TDD | Time Division Duplex |
| TTL | Transistor Transistor Logic |

U

| | |
|-----|----------------------------|
| UTC | Coordinated Universal Time |
| UDP | User Datagram Protocol |

V

| | |
|-----|-----------------------|
| VME | Versa Module Eurocard |
|-----|-----------------------|

X

| | |
|------|---------------------------|
| XFEL | X-Ray Free Electron Laser |
|------|---------------------------|

List of Symbols

| | |
|----------------------------|--|
| λ_u | Undulator wave length |
| f | Frequency |
| t_{period} | Time of one period |
| ϕ_0 | Initial phase |
| $\Delta\phi(t)$ | Time dependent phase difference |
| $\phi_1(t)$ | Time dependent phase of component 1 |
| $\phi_2(t)$ | Time dependent phase of component 2 |
| \bar{v}_n^2 | Power spectral density |
| k_B | Bolzman constant |
| R | Resistance |
| T | Temperature |
| t_1 | Time when request was sent |
| t_2 | Time when request was received |
| t_3 | Time when reply was sent |
| t_4 | Time when reply was received |
| $\Delta t_{master-slave}$ | Link delay between master and slave |
| $\Delta t_{server-client}$ | Link delay between server and client |
| t_{ring} | Orbit period |
| t_{RF} | Period of RF frequency |
| N | Number of bunches |
| $\updownarrow \Delta t$ | Adjustable time delay circuit |
| \otimes | Phase comparator circuits |
| $\div / \times / \Delta t$ | Clock buffers with dividing, multiplication and adjustable time delay |
| Δt_{synch} | Time between two successive synchronization pulses |
| $f_{syncklk}$ | Frequency of synchronization clock |
| $f_{userclk}$ | Frequency of clock to be synchronized |
| T_{Loop} | Loop time from transmitter to receiver and back to master |
| t_{D_1} | Time delay of first delay component |
| t_{EO_T} | Time delay of electrical-to-optical converter at transmitter |
| $t_{F_{TR}}$ | Time delay of signal propagation through fiber (transmitter to receiver) |
| t_{OE_R} | Time delay of optical-to-electrical converter at receiver |
| t_{CDR_R} | Time delay of clock and data recovery component at receiver |
| t_{EO_R} | Time delay of electrical to optical converter at receiver |
| $t_{F_{RT}}$ | Time delay of signal propagation through fiber (receiver to transmitter) |
| t_{OE_T} | Time delay of optical-to-electrical converter at transmitter |
| t_{D_2} | Time delay of second delay component |
| t_{CDR_T} | Time delay of clock and data recovery component at receiver |
| $T_{Receiver}$ | Time delay from transmitter to the receiver |
| $dT_{Receiver}$ | Variation of time delay from transmitter to receiver |

LIST OF SYMBOLS

| | |
|-------------|-------------------------------------|
| dT_{Loop} | Variation of loop time delay |
| n | Number of whole data words |
| m | Number of bits within the last word |
| σ | Sigma of Gaussian distribution |

Bibliography

- [1] M. Altarelli et. al. *XFEL Technical Design Report*. DESY XFEL Project Group, Hamburg, 2007. ISBN 979-3-935702-17-1.
- [2] Patrick Gessler. *Entwicklung eines pikosekunden stabilen Timingsystems fuer das europaeische Roentgenlaserprojekt XFEL*. TU Hamburg-Harburg and DESY, 2007.
- [3] J. Johnson. Thermal agitation of electricity in conductors. *Phys. Rev.*, 32:97, 1928.
- [4] H. Nyquist. Thermal agitation of electric charge in conductors. *Phys. Rev.*, 32:110, 1928.
- [5] W. Schottky. Über spontane stromschwankungen in verschiedenen elektrizitätsleitern. *Annalen der Physik*, 57(23):541–567, 1918.
- [6] J.B. Johnson. Bemerkunge zur bestimmung des elektrischen elementarquantums aus dem schroteffekt. *Annalen der Physik*, 67(1):154–156, 1922.
- [7] J.B. Johnson. The schottky effect in low frequency circuits. *Physical Review*, 26(1):71–85, 1925.
- [8] W. Schottky. Small-shot effect and flicker effect. *Physical Review*, 28(1):74–103, 1926.
- [9] *Noise Analysis in Operational Amplifier Circuits*. Texas Instruments, 2007. <http://www.ti.com/lit/an/slva043b/slva043b.pdf>.
- [10] *Operational Amplifier Noise Prediction*. Intersil, 1996. <http://www.intersil.com/content/dam/Intersil/documents/an51/an519.pdf>.
- [11] G. P. Agrawal. *Nonlinear Fiber Optics - Third Edition*. Academic Press, 2001. ISBN 0-12-045143-3.
- [12] A. E. Siegman. *Lasers*. University Science Books, 1986. ISBN 0-935720-11-3.
- [13] M. Salehi J. G. Proakis. *Communication Systems Engineering - Second Edition*. Prentice Hall, 2002. ISBN 0-13-095007-6.
- [14] D. Mills et. al. *Network Time Protocol Version 4: Protocol and Algorithms Specification (RFC5905)*. IETF RFC, 2010.
- [15] D. Mills. *Network Time Protocol Performance Analysis*. University of Delaware, 2004. <https://www.eecis.udel.edu/mills/database/brief/perf/perf.pdf>.
- [16] Lewis Carroll. *Alice’s Adventures in Wonderland*. 1865.
- [17] G. Daniluk and T. Wlostowski. White rabbit: Sub-nanosecond synchronization for embedded systems. *Proceedings of the 43rd Annual Precise Time and Time Interval Systems and Applications Meeting*, pages 45–60, 2011.

- [18] P. Moreira et. al. White rabbit: Sub-nanosecond timing distribution over ethernet. *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pages 45–60, 2009.
- [19] M. Lipinski et. al. White rabbit: a ptp application for robust sub-nanosecond synchronization. *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication*, pages 25–30, 2011.
- [20] W. Stallings. *Data and Computer Communications (9th Edition)*. Prentice Hall, 2010. ISBN 0-13-2172178.
- [21] A. X. Widmer and P. A. Franaszek. A dc-balanced, partitioned-block, 8b/10b transmission code. *IBM Journal of Research and Development*, 27(5):440, 1983.
- [22] Jukka Pietarinen. *Timing System Evolution - Progress Towards Synchronous Data Distribution*. EPICS Meeting Vancouver, 2009. <http://www.mrf.fi/pdf/presentations/MRF.EPICS2009.pdf>.
- [23] Jukka Pietarinen. *Timing System with Two-Way Signaling, cRIO-EVR*. EPICS Meeting Padova, 2008. <http://www.mrf.fi/pdf/presentations/MRF.EPICS.2008.Padova.pdf>.
- [24] Jukka Pietarinen. *Latest Timing System Developments*. EPICS Meeting Shanghai, 2008. <http://www.mrf.fi/pdf/presentations/MRF.EPICS.2008.Shanghai.pdf>.
- [25] Jukka Pietarinen. *MRF Timing System*. CERN Timing Workshop, 2008. <http://www.mrf.fi/pdf/presentations/MRF.CERN.Feb2008.pdf>.
- [26] Y. Chernousko et. al. Diamond timing system developments. *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems*, pages 244–246, 2003.
- [27] Y. Chernousko et. al. Review of the diamond light source timing system. *Proceedings of the Russian Partical Accelerator Conference*, pages 144–146, 2010.
- [28] J. Dusatko et. al. The lcls timing event system. *Proceedings of the Beam Instrumentation Workshop*, pages 379–383, 2010.
- [29] Master oscillator for the european xfel. *Proceedings of of the International Particle Accelerator Conference*, pages 2771–2773, 2014.
- [30] D. McDowell R.J. Pasquinelli. *Fiber Optic Delay Tracking Experiment*. Fermi National Accelerator Laboratory, 2000. BD RFI Note No.001.
- [31] *Specification of Phase Stabilized Optical Fiber Cable*. Furukawa Electronic Co Ltd, 2004. Datasheet.
- [32] M. Lorek S. Thyagarajan. *Jitter Reduction Techniques for Phase Locked Loops in Deep-Submicron Technologies*. EECS UC Berkeley, 2012. http://www.eecs.berkeley.edu/~sivavth/EE241_Midtermreport_Lorek_Siva.pdf.
- [33] T.H. Smilkstein. *Jitter Reduction on High-Speed Clock Signals - Technical Report No. UBC/EECS-2007-96*. EECS UC Berkeley, 2007. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-96.pdf>.
- [34] *ITU-T G.652 Characteristics of a single-mode optical fibre cable*. International Telecommunication Union, 2000. http://www.iet.unipi.it/m.luise/HTML/AdT/ITU_G652.pdf.

-
- [35] J.P.H. Sladen E. Peschard. Phase compensated fibre-optic links for the lep rf reference distribution. *Proceedings of the IEEE Particle Accelerator Conference*, 3:1960–1962, 1989.
- [36] D. Marcuse. *Principles of Optical Fiber Measurement*. Academic Press, 1981. ISBN 0-12-470980-X.
- [37] Analog Devices. *Continuous Rate 12.3 Mb/s to 2.7 Gb/s Clock and Data Recovery IC with Integrated Limiting Amp - ADN2812*. Analog Devices, Inc., 2012.
- [38] Analog Devices. *LF 2.7 GHz RF/IF Gain and Phase Detector - AD8302*. Analog Devices, Inc., 2002.
- [39] Micrel. *2.5V/3.3V 1.5MHz Precision LVPECL Programmable Delay with Fine Tune Control - SY89296U*. Micrel, Inc., 2006.
- [40] SFF Committee. *INF-8074i Specification for SFP Transceiver*. SFF Committee, 2001. <ftp://ftp.seagate.com/sff/INF-8074.PDF>.
- [41] P. Gessler et.al. A pico-second stable and drift compensated high-precision and low-jitter clock and trigger distribution system for the european xfel project. *Proceedings of the Particle Accelerator Conference*, pages 4168–4170, 2009.
- [42] G. Zito I.D. Landau. *Digital Control Systems - Design, Identification and Implementation*. Springer, 2006. ISBN 978-1846280559.
- [43] H. Werner. *Control Systems 1*. TUHH, 2005. Lecture Notes.
- [44] *MicroTCA.0 Specification Rev. 1.0*. PICMG, 2006. <http://www.picmg.com>.
- [45] *AdvancedMC.0 Specification Rev. 2.0*. PICMG, 2006. <http://www.picmg.com>.
- [46] *MicroTCA.4 Enhancements for Rear I/O and Precision Timing Specification Rev. 1.0*. PICMG, 2011. <http://www.picmg.com>.
- [47] Xilinx. *Virtex-5 FPGA RocketIO GTP Transceiver User Guide*. Xilinx, Inc., 2009.
- [48] Xilinx. *Spartan-6 FPGA GTP Transceivers User Guide*. Xilinx, Inc., 2010.
- [49] Xilinx. *Virtex-5 FPGA User Guide*. Xilinx, Inc., 2012.
- [50] *DOOCS: a Distributed Object Oriented Control System - Home page*. DESY, 2013. <http://doocs.desy.de>.
- [51] *TINE (Three-fold Integrated Networking Environment) - Home page*. DESY, 2012. <http://tine.desy.de>.
- [52] B. Heisen et. al. Karabo: An integrated software framework combining control, data management, and scientific computing tasks. *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems*, pages 1465–1468, 2013.
- [53] W. Chou et. al. *Beam Dynamics Newsletter No. 47*. Internal Committee for Future Accelerators, 2008.
- [54] E. Sombrowski et. al. jddd, a state-of-the-art solution for control panel development. *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems*, pages 1189–1192, 2011.

- [55] *PCI Express Base Specification 1.1*. PICMG, 2005.
- [56] *White Rabbit Switch v3.3 - WRS-3/18*. Seven Solutions, 2013.
http://www.sevensols.com/whiterabbitsolution/files/7SP-WRS-3_18.pdf.