UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Ingeniería Técnica Industrial
Especialidad en Electrónica Industrial

PROYECTO
Fin de Carrera

TÍTULO	Hardware and Firmware Development for an FPGA-based Multipurpose Board Targeted for High Energy Physics Experiments
AUTOR	Manoel Barros Marín
DIRECTOR	Camilo Quintáns Graña
CODIRECTOR	Paschalis Vichoudis
PONENTE	Manuel Alonso Castro Gil
DEPARTAMENTO	Departamento de Ingeniería Eléctrica Electrónica y de Control (DIEEC)

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

(A rellenar por el Tribunal Calificador)

TRIBUNAL CALIFICADOR

PRESIDENTE: _____
_____

VOCAL _____
_____

SECRETARIO _____
_____

FECHA DEFENSA ___ de _____ de _____

CALIFICACIÓN _____

Vocal Presidente Secretario

Fdo.:_____ Fdo.:_____ Fdo.:_____

CÓDIGOS UNESCO

PH-ESE-BE

Hardware and Firmware Development for an FPGA-based Multipurpose Board Targeted for High Energy Physics Experiments

Author:

Manoel Barros Marín

Director:

Camilo Quintáns Graña

Co-director:

Paschalis Vichoudis

Supervisor:

Manuel Alonso Castro Gil

ETSII

CERN (Geneva), 20th March 2012

Adicado a/Dedicado a/Dedicated to:

Pai, sempre botarei en falta as nosas conversas...

Mamá, ya sé que eres la única, pero para mí eres la mejor.

Família.

Amigos/Friends.

Colleagues from PH-ESE-BE.

Camilo, eres un gran profesor, gracias por haberme ayudado tanto durante toda la carrera y en la toma de muchas decisiones.

Paschalis, thanks for giving me this opportunity, for teaching me so many things and for being such a good supervisor.

Table of Contents

Symbols list

A

- A - Ampere
- ACJTAG - Alternating Current JTAG
- ADC - Analog-to-Digital Converter
- AFE - Analog Front End
- ALICE - A Large Ion Collider Experiment
- AMC - Advanced Mezzanine Card
- ASIC - Application Specific Integrated Circuit
- ATLAS - A Toroidal LHC Apparatus

B

- BCC - Bump Chip Carrier
- BE - Back-end
- BER - Bit Error Ratio
- BERT - Bit Error Ratio Test
- BGA - Ball Grid Array
- BOOSTER - Proton Synchrotron Booster
- BPI - Byte Parallel Interface
- BSC - Bit Slip Control

C

- CDR - Clock and Data Recovery
- CDRH - Center for Devices and Radiological Health
- CERN - Conseil Européen pour la Recherche Nucléaire
- CLB - Configurable Logic Block
- CML - Current mode logic
- CMS - Compact Muon Solenoid
- CMOS - Complementary metal oxide semiconductor
- COTS - Components Of The Shelf
- CPLD - Complex Programmable Logic Device
- CPPM - Centre de Physique des Particules de Marseille
- CS - Clock Synthesizer

D

- D - Data
- DAQ - Data AcQuisition
- dBc - decibels relative to the carrier
- DC - Direct Current
- DCI - Digital Controlled Impedance
- DCS - Detector Control System
- DDR - Double Data Rate
- DEC - Decoder
- DES - De-Serializer
- DFE - Digital Feedback Equalization
- DIEEC - Departamento de Ingeniería Eléctrica Electrónica y de Control
- DOH - Digital Opto Hybrid
- DRP - Dynamic Reconfiguration Port

- DSP - Digital signal processor

E

- e - electron
- ECAL - Electromagnetic Calorimeter
- EDA - Electronic Design Automation
- EEPROM - Electrically Erasable Programmable Read-Only Memory
- EMI - ElectroMagnetic Interference
- ESD - ElectroStatic Discharge
- ESE - Electronics Systems for Experiments
- ET - Transverse Energy

F

- FAE - Física de Alta Energía
- FB - FeedBack
- Fc - cutoff Frequency
- FD - Feedback divider
- FE - Front-end
- FEC(1) - Front End Control
- FEC(2) - Forward Error Correction code
- FFT - Fast Fourier Transform
- FIFO - First In First Out
- FMC - FPGA Mezzanine Card
- FPGA - Field Programmable Gate Array
- FSM - Finite State Machine

G

- Gbe - Gigabit ethernet
- GBLD - GigaBit Laser Driver
- Gbps - Gigabit per second
- GBT - GigaBit Transceiver
- GBTIA - GigaBit Transimpedance Amplifier
- GHz - Gigahertz
- GLIB - Gigabit Link Interface Board
- GLRC - GBT Link Reset Control
- GMII - Gigabit Media Independent Interface
- GOH - Gigabit Optical Hybrids
- GOL - Gigabit Optical Link
- GPIB - General-Purpose Instrumentation Bus
- GR - General Reset
- GSps - Giga Samples per second
- GUI - Graphical User Interface

H

- HAL - Hardware Abstraction Layer
- HBM - Human Body Model
- HCAL - Hadron Calorimeter
- HDL - Hardware Definition Language
- HEP - High Energy Physics
- HLT - High-Level Trigger
- Hz - Hertz

I

- I2C - Inter-Integrated Circuit
- iBERT - integrated Bit Error Ratio Test
- IC - Integrated Circuit
- IEC - International Electrotechnical Commission
- IEEE - Institute of Electrical and Electronics Engineers
- I/O - Input/Output
- IOB - Input/Output Block
- IP - Intellectual Property
- IPbus - Internet Protocol bus

J

- JTAG - Joint Test Action Group

K

- k - kilo
- kb - kilobit
- kHz - kiloHertz
- krad - kilorad

L

- LC - Lucent Connector, Little Connector or Local Connector
- LCD - Liquid Crystal Display
- LHC - Large Hadron Collider
- LHCb - Large Hadron Collider beauty
- LHCf - Large Hadron Collider forward
- LINAC - LINear ACcelerator
- LOS - Loss-of-Sync
- LUT - Look Up Table
- LVCMOS - Low Voltage Complementary Metal Oxide Semiconductor
- LVDS - Low-Voltage Differential Signalling
- LVPECL - Low-Voltage Positive Emitter-Coupled Logic

M

- m - metres
- MAC - Media Access Control address
- MB - MegaByte
- Mbps - Megabit per second
- ME - Micro Electronics section
- MGT - MultiGigabit Transceiver
- MHz - MegaHertz
- MII - Media Independent Interface
- MM - Multi Mode
- MMC - Module Management Controller
- MMCM - Mixed-Mode Clock Manager
- Mpt - Megapoints
- Mrad - Megarad
- MSA - Multi-Source Agreement
- mV - millivolt

N

- N - Negative
- nm - nanometre

- ns - nanosecond
- NOR - Not OR

O

- OMA - Optical Modulated Amplitude
- OOB - Out Of Band

P

- p - proton
- P - Positive
- PA - Phase Aligner
- Pb - Lead
- PC - Personal Computer
- PCB - Printed Circuit Board
- PCIe - Peripheral Component Interconnect express
- PCS - Physical Coding Sublayer
- pF - picoFarad
- PFC - Proyecto Fin de Carrera
- PG - Pattern Generator
- PH - Physics Department
- PISO - Parallel Input Serial Output
- PLL - Phase Locked Loop
- PMA - Physical Medium Attachment sublayer
- PMF - Pattern Match Flag
- POF - Plastic Optical Fibre
- ppm - parts per million
- PRBS - PseudoRandom Bit Sequence
- ps - picosecond
- PS - Proton Synchrotron

Q

- QFN - Quad-Flat No-leads

R

- RAM - Random Access Memory
- REFCLK - REFerence CLocK
- RFCLK - Rx Frame CLocK
- RFI - Radio Frequency Interference
- rms - root mean square
- ROHS - Restriction of Hazardous Substances
- ROSA - Receiver Optical Sub-Assembly
- RX - Receiver
- RXRECCLK - RX RECovered CLocK
- RXUSRCLK - RX USeR CLocK
- RXUSRCLK2 RX USeR CLocK 2
- RWCLK - RX Word CLocK

S

- SAS - Serial Attach SCSI
- SATA - Serial Advanced Technology Attachment
- SAW - Surface Acoustic Wave
- SC - Slow Control
- SDR - Single Data Rate

- SER - SERializer
- SERDES - SERializer/DE-Serializer
- SFP+ - Small Form-factor Pluggable plus
- SGMII - Serial Gigabit Media Independent Interface
- SIPO - Serial Input Parallel Output
- SLHC - Super Large Hadron Collider
- SMA - SubMiniature version A
- SPI - Serial Peripheral Interface bus
- SPS - Super Proton Synchrotron
- SRAM - Static Random Access Memory
- SRL - Shift Register Left
- SS - Sequential System
- SVN - apache SubVersioN

T

- TDC - Time to Digital Converter
- TeV - Tera electronVolt
- TFCLK - TX Frame CLocK
- TFT - Thin Film Transistor
- TIA - Telecommunications Industry Association
- TOTEM - TOTal Elastic and diffractive cross section Measurement
- Tp - peaking Time
- TSCLK - TX Serial CLocK
- TTC - Timing Trigger and Control
- TTL - Transistor-Transistor Logic
- TWCLK - TX Word CLocK
- TX - Transmitter
- TXOUTCLK - TX OUTput Clock
- TXUSRCLK - TX USeR CLocK
- TXUSRCLK2 - TX USeR CLocK 2

U

- UI - Unit Interval
- UL - User Logic
- UNED - Universidad Nacional de Educación a Distancia

V

- V - Volt
- VCO - Voltage-Controlled Oscillator
- VGA - Video Graphics Array
- VTRx - Versatile Transceiver
- VXCO - Voltage-Controlled crystal Oscillator

W

- W - Watt
- WLCG - Worldwide LHC Computing Grid

X

- XST - Xilinx Synthesis Technology

Y

Z

Other Symbols

- κ - kaon
- γ - photon
- μ - muon
- μs - microsecond
- μTCA - micro Telecommunications Computing Architecture
- π - p ion
- ν - neutron
- Ω - Ohm

Figures list

Tables list

GLIB Schematic list

HDL Codes list

Peticionario

El trabajo para este Proyecto Fin de Carrera (PFC) ha sido llevado a cabo en el CERN (Conseil Européen pour la Recherche Nucléaire) por Manoel Barros Marín para obtener el título de “Ingeniero Técnico Industrial especialista en Electrónica Industrial” a petición del “Departamento de Ingeniería Eléctrica Electrónica y de Control (DIEEC)” de la Universidad Nacional de Educación a Distancia (UNED).

Petitioner

The work in this thesis (Proyecto Fin de Carrera) (PFC) has been carried out at CERN (Conseil Européen pour la Recherche Nucléaire) by Manoel Barros Marín in order to obtain the diploma of “Ingeniero Técnico Industrial especialista en Electrónica Industrial” from the “Departamento de Ingeniería Eléctrica Electrónica y de Control (DIEEC)” of the Universidad Nacional de Educación a Distancia (UNED).

CERN Technical Student Programme

The Technical Student Programme, offered by CERN (sited on the French/Swiss border near Geneva), is focused on undergraduate students in applied physics, engineering or computing. The programme gives the opportunity to students of spending a training period of 4 to 12 months (14 months at most) during the course of their studies.

During the last year of my Studies in “Ingeniería Técnica Industrial”, I have applied for this programme following the recommendation by one of my professors. My application has been selected by the committee in early 2011 and I was offered a 9-month contract (starting in February 1st, 2011) that has later been extended for another 5 months.

In my opinion, the participation in this programme is a great opportunity in several ways. From the academic/professional point of view, CERN is an outstanding place to work, with excellent facilities and it gives the chance of working hand in hand with experts in their respective fields, while socially, it offers the opportunity to work in a multi-cultural environment with all that this implies.

Resumen

Este Proyecto Fin de Carrera (PFC) describe el desarrollo de hardware y firmware para una tarjeta multipropósito basada en FPGA dirigida a Física de Alta Energía (FAE) que ha sido llevado a cabo durante mi estancia en el CERN

El capítulo 1 introduce el CERN y los experimentos de FAE. El acelerador de 27 km de largo llamado Large Hadron Collider (LHC) así como los seis experimentos del LHC (CMS, ATLAS, LHCb, ALICE, TOTEM and LHCf) son brevemente descritos. Este capítulo está mayoritariamente enfocado en la arquitectura de los sistemas electrónicos de los experimentos arriba mencionados y en los enlaces de comunicación entre los sistemas electrónicos situados en el detector y en la "counter room". Finalmente, las razones que han llevado al desarrollo de la tarjeta presentada en este PFC son comentadas, así como los requerimientos de la nueva tarjeta.

El capítulo 2 describe la Gigabit Link Interface Board (GLIB), la tarjeta que ha sido diseñad y desarrollada durante mi estancia en el CERN. Después de una introducción general sobre la tarjeta y su propósito, las principales características de la tarjeta son brevemente descritas. Además del hardware, firmware y software, también son mencionadas las tarjetas auxiliares implicadas en el proyecto. Al final del capítulo, uno de los proyectos llevados a cabo durante el desarrollo y testeo del a GLIB es seleccionado para una explicación en mayor profundidad.

El capítulo 3 desarrolla la explicación del proyecto seleccionado. Este proyecto emula un enlace entre la electrónica del "Front-end" (FE) y del "Back-end" (BE). Tan pronto como los objetivos del proyecto han sido clarificados, la explicación comienza con una breve descripción de los componentes implicados en el proyecto, continua a través de la configuración de dichos componentes y termina con la presentación de los resultados obtenidos de los test.

El capítulo 4 repasa el estado del proyecto GLIB al momento de escribir este PFC y también presente los futuros desarrollos planeados para los próximos meses.

Abstract

This thesis (PFC) describes the hardware and firmware development of an FPGA-based multi-purpose board targeted for High Energy Physics (HEP) that has been carried out during my stay at CERN.

Chapter 1 introduces CERN and HEP experiments. The 27 km long accelerator namely Large Hadron Collider (LHC) as well as the six LHC experiments (CMS, ATLAS, LHCb, ALICE, TOTEM, and LHCf) are briefly described. This chapter is mainly focused on the electronics systems architecture for the above mentioned experiments and on the communication links between the electronic systems of the detectors and the counter room. Finally, the reasons that have led to the development of the board presented in this thesis (PFC), as well as the requirements for the new board are mentioned.

Chapter 2 describes the Gigabit Link Interface Board (GLIB), .the board that has been designed and developed during my stay at CERN. After a general introduction to the board and its purpose, the main features of the board are briefly described. Besides the hardware, firmware and software, the auxiliary boards implied in the project are mentioned as well. At the end of the chapter, one of the projects carried out during the development and testing of the GLIB is selected for further explanation.

Chapter 3 deals with the explanation of the selected project. This project emulates a link between Front-end (FE) and Back-end (BE) electronics. As soon as the objectives of the project have been clarified, the explanation begins with a brief description of the components implied in the project, continues through the setup of these components and ends with a presentation of the results obtained from the tests.

Chapter 4 summarizes the status of the GLIB project at the moment of writing this thesis and also presents the future courses of action planned for the next months.

Chapter 1 - Introduction

1.1. High Energy Physics experiments at CERN

1.1.1. Introduction to CERN

CERN, the *European Organization for Nuclear Research* sits astride the Franco–Swiss border near Geneva and is one of the world's largest and most respected centres for scientific research [1].

Figure 1-1: CERN logo.

The name is derived from the acronym for the French "*Conseil Européen pour la Recherche Nucléaire*" or European Council for Nuclear Research, a provisional body founded in 1952 with the mandate of establishing a world-class fundamental physics research organization in Europe. When the Organization officially came into being in 1954, the Council was dissolved, and the new organization was given the title European Organization for Nuclear Research, although the name CERN was retained.

CERN is run by 20 European Member States and Israel as Associate Member State (Romania is a candidate to become a Member State). The current Member States are Austria, Belgium, Bulgaria, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Italy, the Netherlands, Norway, Poland, Portugal, the Slovak Republic, Spain, Sweden, Switzerland and the United Kingdom.

Many non-European countries are also involved in different ways. Scientists come from around the world to use CERN's facilities.

CERN business is fundamental physics, finding out what the Universe is made of and how it works.

1.1.2. Introduction to HEP experiments

At CERN, the world's largest and most complex scientific instruments are used to study the basic constituents of matter (the fundamental particles). By studying what happens when these particles collide, physicists learn about the laws of Nature. Teams of physicists from around the world analyse the particles created in the collisions using special detectors in a number of experiments dedicated to the LHC. The instruments used at CERN are particle accelerators and detectors. Accelerators boost beams of particles to high energies before they are made to collide with each other or with stationary targets. Detectors observe and record the results of these collisions [2] [3].

The Large Hadron Collider (LHC)

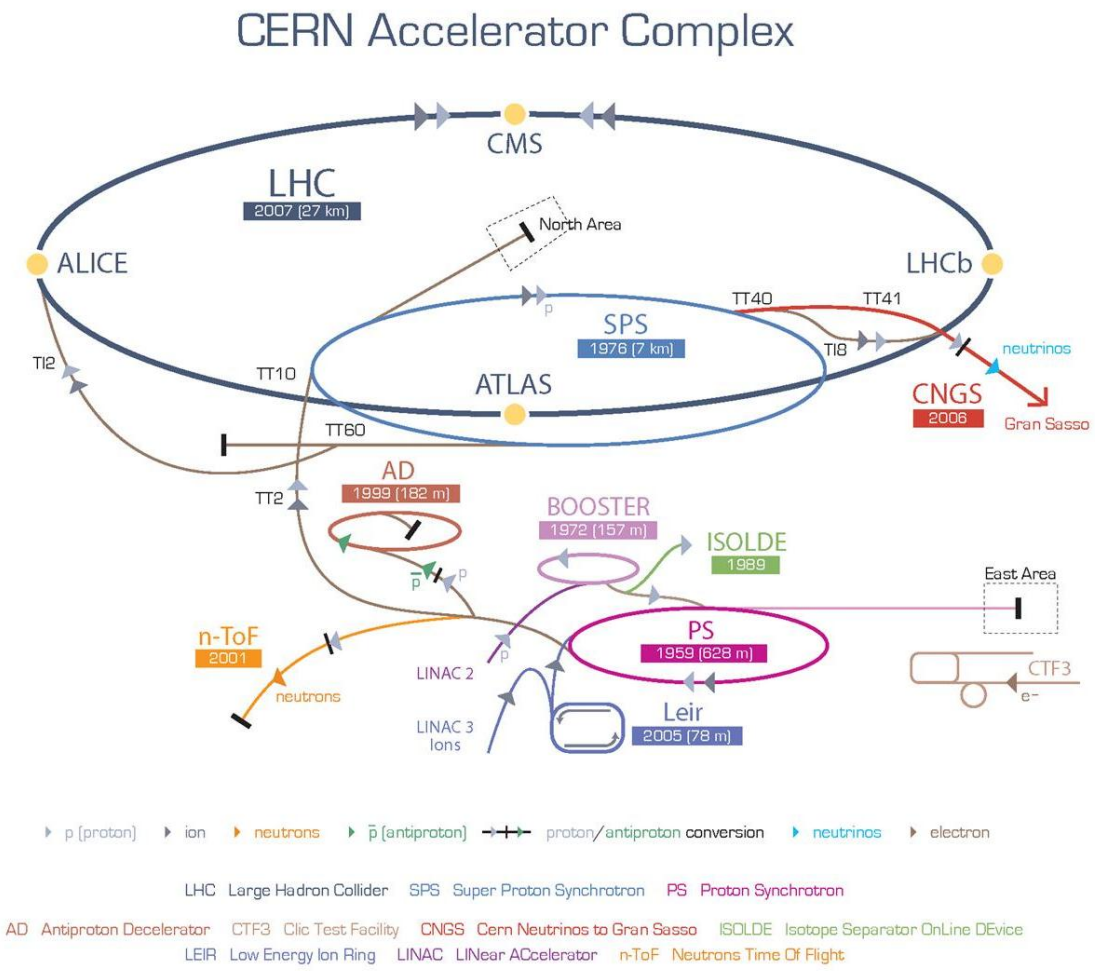


Figure 1-2: CERN particle accelerators and experiments.

The *Large Hadron Collider* (LHC) [4] is the world's largest and highest-energy particle accelerator. With the LHC, it is possible to recreate the conditions just after the Big Bang, by colliding two beams of particles head-on at very high energy. These two beams of subatomic particles called "hadrons" (either protons or lead ions) travel in opposite directions inside the circular accelerator, gaining energy with every lap.

The collider is contained in a circular tunnel, with a circumference of 27 kilometres at a depth ranging from 50 to 175 metres underground. Its construction took a ten year period from 1998 to 2008 by collaboration between over 10,000 scientists and engineers from over 100 countries, as well as hundreds of universities and laboratories.

As it was said before, the LHC is able to work either with protons or with lead ions. Its synchrotron is designed for colliding opposing particle beams at energy up to 7 teraelectronvolts (7 TeV) per nucleon in the case of protons, or 2.76 TeV per nucleon when the particles are lead ions.

Hadrons are accelerated by different pre-accelerators: *linear accelerators* (LINAC) (LINAC 2 for proton or LINAC 3 for lead ions), *Proton Synchrotron Booster* (BOOSTER) (only for proton), *Proton Synchrotron* (PS) and *Super Proton Synchrotron* (SPS). Once the acceleration has reached the desired level, the hadrons are sent to the LHC synchrotron.

CERN Experiments

There are six experiments at the LHC. All of them run by international collaborations, bringing together scientists from institutes all over the world. Each experiment is distinct, characterised by its unique particle detector.

ATLAS (A Toroidal LHC Apparatus)

The *A Toroidal LHC Apparatus* (ATLAS) [5] detector is the largest volume particle detector ever constructed (It is about half as big as the Notre Dame Cathedral in Paris and weighs the same as the Eiffel Tower or a hundred 747 jets). More than 3000 scientists from 172 institutes in 38 countries work on the ATLAS experiment. ATLAS is one of two general-purpose detectors at the LHC. It is used for investigating a wide range of physics, including the search for the Higgs boson, extra dimensions, and particles that could make up dark matter. ATLAS records sets of measurements on the particles created in collisions (their paths, energies, and their identities). This is accomplished in ATLAS through six different detecting subsystems. Another vital element of ATLAS is the huge magnet system that bends the paths of charged particles for momentum measurement.

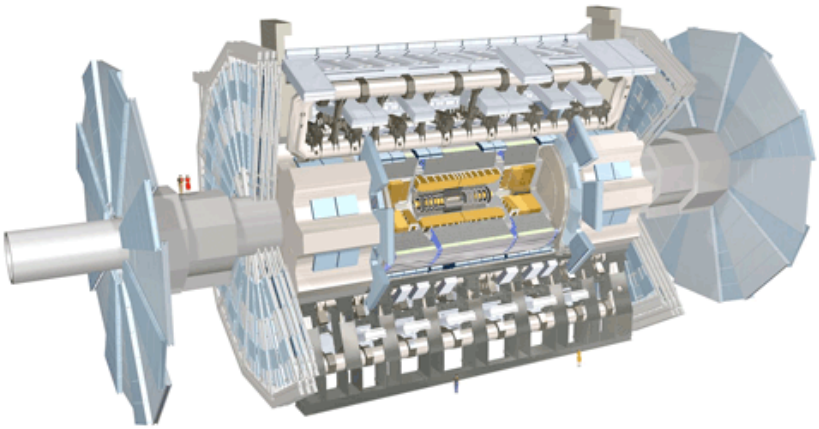


Figure 1-3: ATLAS detector.

Characteristics:

- Size: 46 m long, 25 m high and 25 m wide
- Weight: 7000 tonnes
- Design: barrel plus end caps
- Location: Meyrin, Switzerland

CMS (Compact Muon Solenoid)

More than 3000 scientists collaborate in the *Compact Muon Solenoid* (CMS) [6], coming from 172 institutes in 40 countries. It is a general purpose detector. Although it has the same scientific goals as the ATLAS experiment, it uses different technical solutions and design of its detector magnet system to achieve these. The CMS detector is built around a huge solenoid magnet. This takes the form of a cylindrical coil of superconducting cable that generates a magnetic field of 4 teslas, about 100 000 times that of the Earth. The magnetic field is confined by a steel 'yoke' that forms the bulk of the detector's weight of 14 000 tonnes (or about 465 Boeing 737s). An unusual feature of the CMS detector is that instead of being built in-situ underground, like the other giant detectors of the LHC experiments, it was constructed on the surface, before being lowered underground in 15 sections and reassembled.

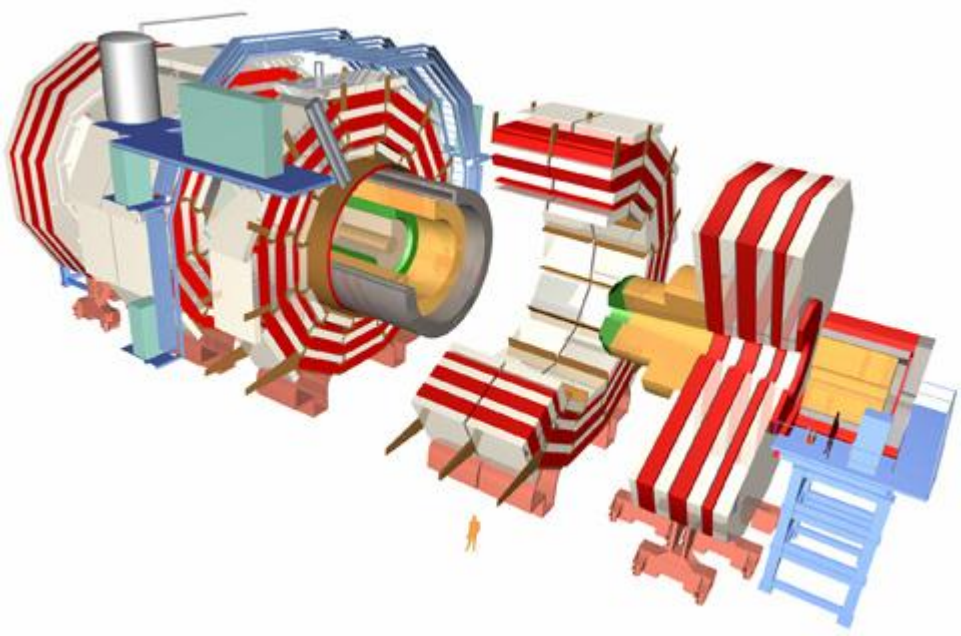


Figure 1-4: CMS detector.

Characteristics:

- Size: 21 m long, 15 m wide and 15 m high
- Weight: 12 500 tonnes
- Design: barrel plus end caps
- Location: Cessy, France

ALICE (A Large Ion Collider Experiment)

A collaboration of more than 1000 scientists from 105 physics institutes in 30 countries works on the *A Large Ion Collider Experiment* (ALICE) (November 2011) [7]. For the ALICE experiment, the LHC collides lead ions to recreate the conditions just after the Big Bang under laboratory conditions. The data obtained allows physicists to study a state of matter known as quark‑gluon plasma, which is believed to have existed soon after the Big Bang. Collisions in the LHC generate temperatures more than 100,000 times hotter than the heart of the Sun.

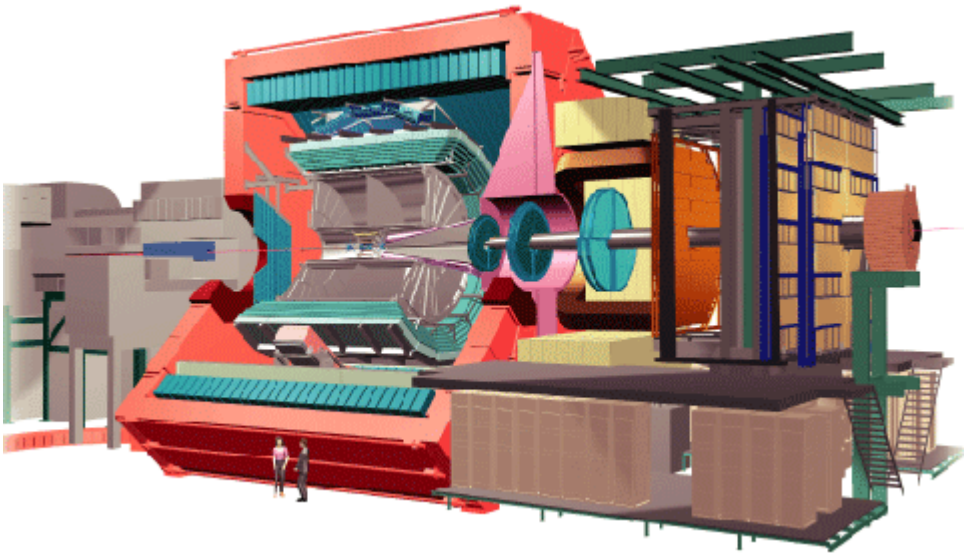


Figure 1-5: ALICE detector.

Characteristics:

- Size: 26 m long, 16 m high, 16 m wide
- Weight: 10,000 tonnes
- Design: central barrel plus single-arm-forward muon spectrometer
- Location: St Genis-Pouilly, France

LHCb (Large Hadron Collider beauty)

The *Large Hadron Collider beauty* (LHCb) [8] collaboration has 700 scientists from 52 institutes around the world. The LHCb experiment will help us to understand why we live in a Universe that appears to be composed almost entirely of matter, but no antimatter. This experiment specialises in investigating the slight differences between matter and antimatter by studying a type of particle called the 'beauty quark', or 'b quark'. Instead of surrounding the entire collision point with an enclosed detector, the LHCb experiment uses a series of sub-detectors to detect mainly forward particles. The first sub-detector is mounted close to the collision point, while the next ones stand one behind the other, over a length of 20 m. An abundance of different types of quark will be created by the LHC before they decay quickly into other forms. To catch the b-quarks, LHCb has developed sophisticated movable tracking detectors close to the path of the beams circling in the LHC.

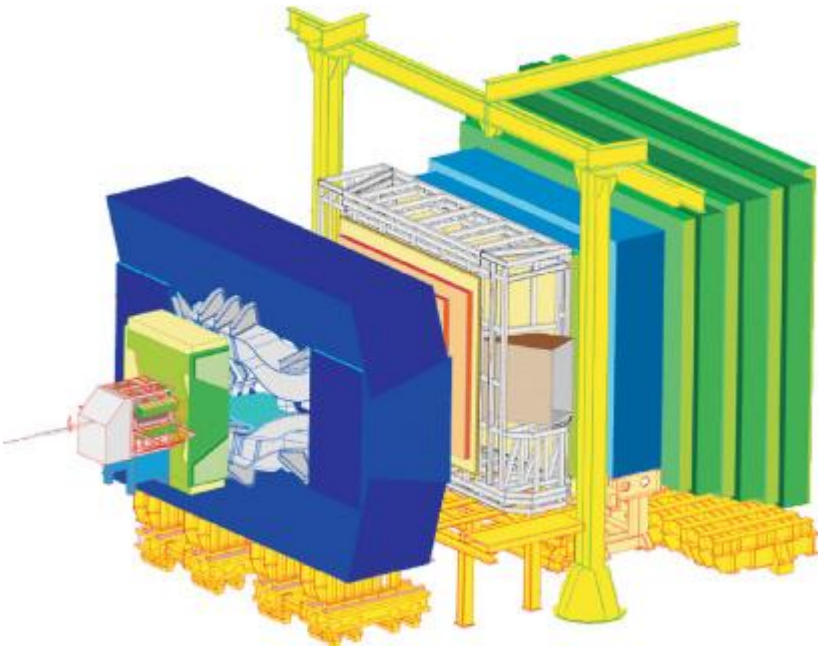


Figure 1-6: LHCb detector.

Characteristics:

- Size: 21 m long, 10 m high and 13 m wide
- Weight: 5600 tonnes
- Design: forward spectrometer with planar detectors
- Location: Ferney-Voltaire, France

LHCf and TOTEM

These two further experiments, *TOTal Elastic and diffractive cross section Measurement* (TOTEM) [9] and *Large Hadron Collider forward* (LHCf) [10], are much smaller in size. They are designed to focus on "forward particles" (protons or heavy ions). These are particles that just brush past each other as the beams collide, rather than meeting head-on.

The **TOTEM** experiment studies forward particles to focus on physics that is not accessible to the general-purpose experiments. Among a range of studies, it will measure, in effect, the size of the proton and also monitor accurately the LHC's luminosity. To do this TOTEM must be able to detect particles produced very close to the LHC beams. Although the two experiments are scientifically independent, TOTEM complements the results obtained by the CMS detector and by the other LHC experiments overall.

TOTEM detector:

- Size: 440 m long, 5 m high and 5 m wide
- Weight: 20 tonnes
- Design: Roman pot and GEM detectors and cathode strip chambers
- Location: Cessy, France (near CMS)

The **LHCf** experiment uses forward particles created inside the LHC as a source to simulate cosmic rays in laboratory conditions. Cosmic rays are naturally occurring charged particles from outer space that constantly bombard the Earth's atmosphere. They collide with nuclei in the upper atmosphere, leading to a cascade of particles that reaches ground level. Studying how collisions inside the LHC cause similar cascades of particles will help scientists to interpret and calibrate large-scale cosmic-ray experiments that can cover thousands of kilometres.

LHCf detector:

- Size: two detectors, each measures 30 cm long, 80 cm high, 10 cm wide
- Weight: 40 kg each
- Location: Meyrin, Switzerland (near ATLAS)

1.1.3. Electronics for HEP experiments

Everything related with *High Energy Physics* (HEP) experiments involves high complexity, and their electronic systems are a good example of that. To be able to deal with so huge complexity, it is necessary to manage these electronic systems as small groups with interactions between them. It is like a chain with many links, and each link has its own characteristics and functionality. In HEP experiments, electronic systems are divided in two main categories taking into account the distance of the device with the detector. These main categories are called Front-end (FE) and Back-end (BE).

Front-end

With the term *Front-end* (FE), we refer to the electronic sensors, and the rest of the electronic systems (signal processing, communications etc.) which are close to the detector. Most of the times these devices have to be custom designed because of their special characteristics:

- They have to **resist very high levels of radiation**. Although it is possible to find radiation hardened devices on the market already developed for other fields (aerospace, military etc.), the levels of radiation found in HEP experiments are so high that these devices are not able to withstand so extreme conditions.

Table 1-1: Radiation limit comparison.

Radiation Limit (rad) (*)	
Commercial grade devices	krad
Space grade devices (SOI and SOS)	Mrad
HEP Front-end devices (ATLAS)	50 Mrad

- Their measurements have to be **very precise** (high sensitivity, high resolution etc.) because they have to measure accurately characteristics of tiny particles.
- They have to be **very fast**. In HEP, events occur a with a very short time difference between them, and in any case, an event which was not measured is lost forever.

(*) 100 rad = 1 Joule per kg

Detection elements

A **detector** is composed by different layers of detection elements. Electronic sensors have a very important role among these detection elements. Each layer is designed for a special kind of measurement with different types of electronic sensors. The technologies employed by the different multipurpose experiments may be different, but the distribution and functionality of the macroscopic elements is the same. This distribution is not randomly disposed. This layout is necessary to be able the identification of the particles generated from hadrons collisions (also called “events”) and for the measurement of its position, momenta or energy.

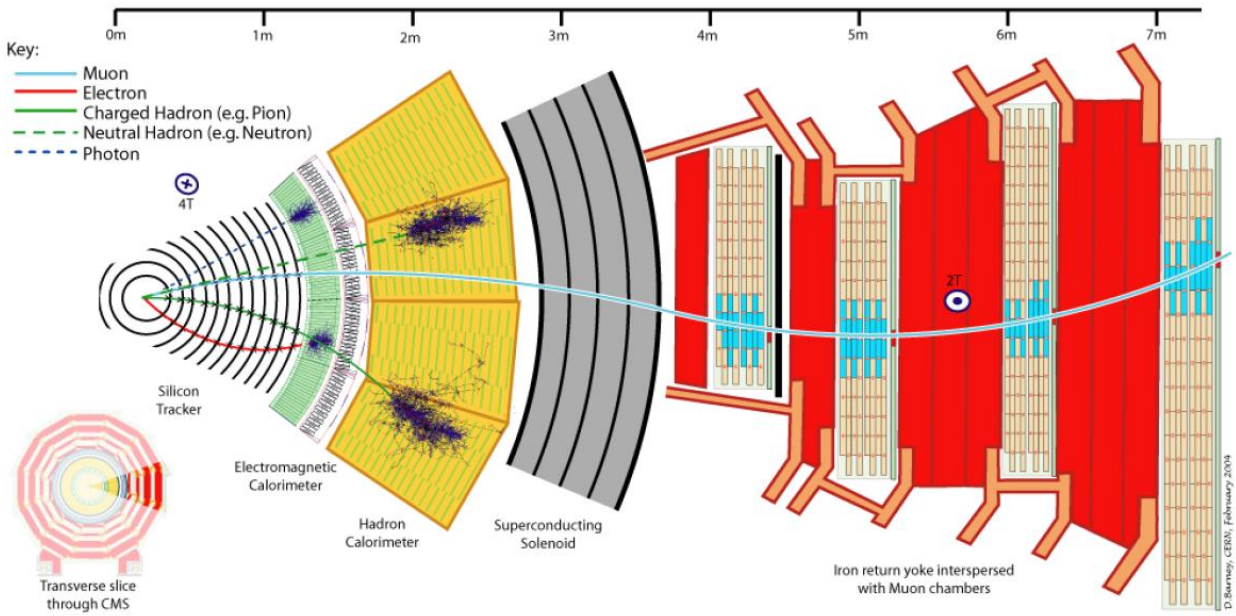


Figure 1-7: Detecting elements of the CMS.

As example, the different layers of the CMS detector are described below (although the interaction point and the magnet are not electronic devices, they are included to complete the layout):

- **The interaction point:** is the point in the centre of the detector at which hadrons collisions occur between the two counter-rotating.
- **Tracker:** The tracker is designed to measure precisely the momentum of charged particles. It does this by having a highly segmented and layered charged particle detector involving millions of channels. Silicon pixel and strip sensors are used on

the inner layers, while “straw” drift chambers are used on the outer layers of the tracker. The presence or not of a charged particle is recorded in each channel and the tracks reconstructed. The radius (R) of curvature of the tracks in the magnetic field (B) reveals the momentum (p) of the particle, since $R \propto \frac{p}{0.3B}$.

- **Electromagnetic Calorimeter (ECAL):** The electromagnetic calorimeter measures the energy of photons and electrons (γ, e). Almost no photons or electrons originating from the proton-proton interaction point pass beyond the ECAL.
- **Hadron Calorimeter (HCAL):** The hadronic calorimeter measures the energy of particles such as protons, p ions, neutrons and kaons (p, π, ν, κ). Almost no hadrons generated from the p-p interaction pass beyond the HCAL.
- **Magnet:** The magnet allows the charge/mass ratio of particles to be determined from the curved track that they follow in the magnetic field.
- **Muon Detector:** The only particles generated from the proton-proton interaction to reach the muon detector are muons and neutrinos (μ, ν). The muons are detected directly but the neutrinos escape undetected. The presence of a neutrino is inferred by missing transverse energy (ET).

As it was said before, each layer has its own kinds of electronic sensors, with different characteristics (silicon sensors, scintillators, ionization chambers etc.), but in all of them, during the detection of an event, radiation deposits energy in a detecting medium (gas, solid or liquid). This energy produces the movement of charges. Every individual particle detection event will appear as a pulse of electric charge.

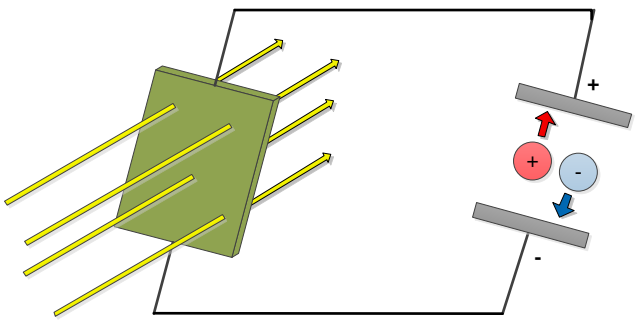


Figure 1-8: Electronic sensor and electrodes.

Signal processing systems

When a particle passes through a sensor, a pulse-like current is produced in the electrodes. This very low current has to be amplified and processed. This task is done by the **FE signal processing systems**. Traditionally, the first stages of the signal processing chain were analogic, conditioning the signal for a further digitization. However, digital devices are displacing the analogic versions. At this moment, analogic and digital devices can be found in first stages of the signal processing chain and both approaches are valid. In this thesis (PFC), the **traditional (analogic) version** is chosen for the example for didactic purposes.

When current signals are exposed at high irradiation levels, it is impossible to distinguish individual pulses because they mix up and look like a noisy continuous current. At low irradiation levels, **pulse-like currents** are mostly distinguishable, and individual events can be studied separately. In HEP experiments, FE systems work at low irradiation levels.

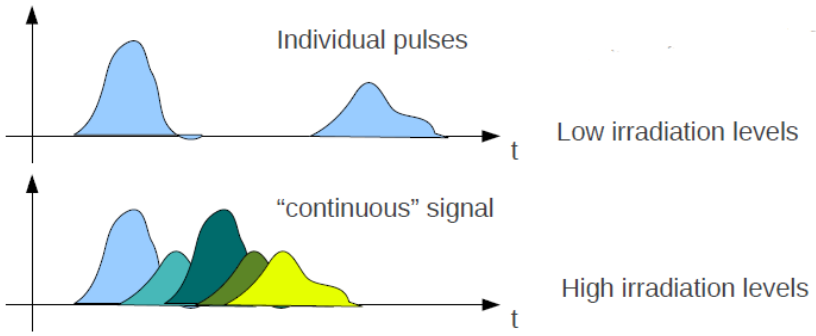


Figure 1-9: Pulse-like currents.

The signal level of these pulses is very low and must be amplified to allow a proper digitization. It is preferable to amplify these signals in the very first stages of the system to avoid the inclusion of noise, which would be amplified with the signal. The **pre-amplifier** is the responsible of increasing the level of the signal, but it also acts as a low pass filter, so high frequency noise is eliminated.

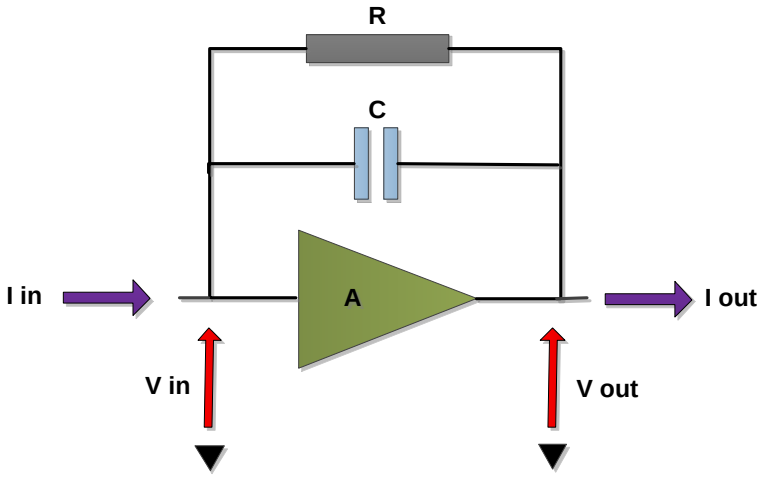


Figure 1-10: Pre-amplifier.

Normally, the next step at this point, once the current pulse has been amplified, is to compensate the time constant of the pre-amplifier. The time constant is too long because of the slow response of the pre-amplifier. This slow response is needed to avoid noise. HEP experiments need signals with a short time constant, so the time constant of the signal has to be compensated. This compensation is done with a "**pole-zero" circuit**. After this compensation, the signal will present shorter response, complying the requirements of the HEP experiment.

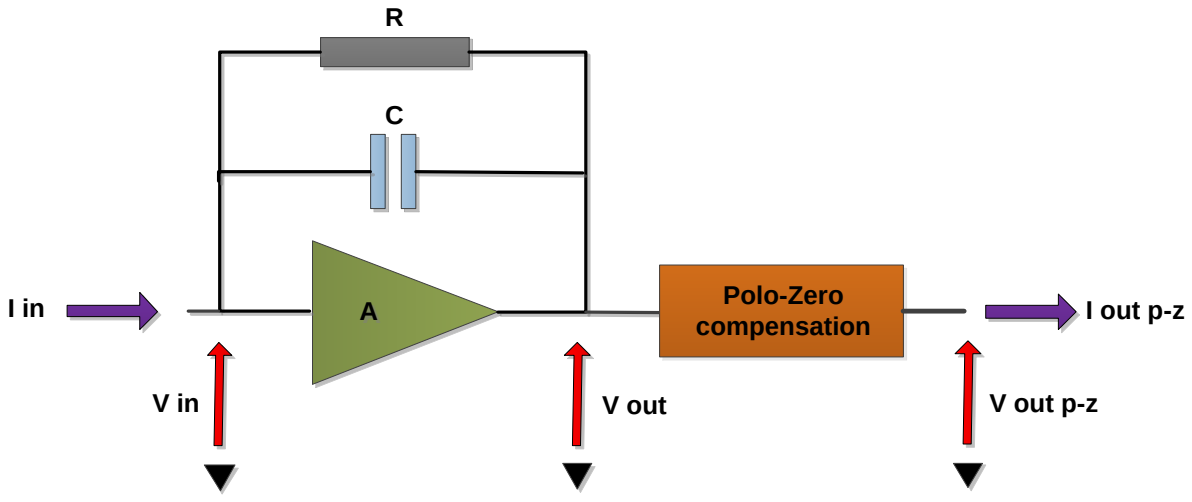


Figure 1-11: Polo-zero compensation.

The next step is the shaping of the signal. It is done by the **shaper circuit.** There are two typical transformations done by the shaper circuit. One of them is that the peak of the pulse is gradually rounded to facilitate measurement of the amplitude at the

peaking time (Tp). If the pulse keeps the same shape when the level of the signal changes, the measurement of the height of the pulse, also provides a measure of the energy. The other typical transformation is that narrow pulses are made wider to reduce the noise of the signal. However, when measuring rapid successions of pulses, if the pulse is too wide, it could be overlapping between pulses, so the duration of the pulse has to be limited.

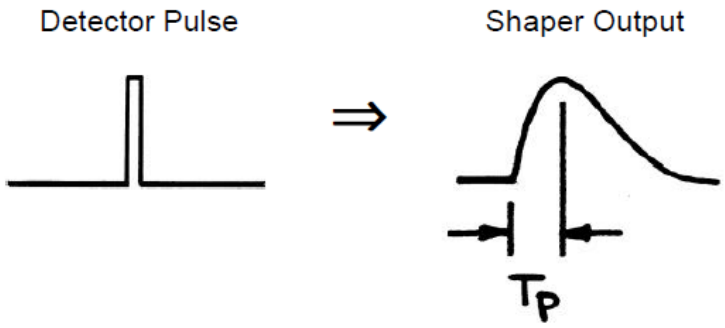


Figure 1-12: Shaping of the signal.

At this point, the signal composed by pulses which correct amplitude level, timing response and shape for being digitized and whose area is proportional to the energy captured by the sensor. Therefore, it is the turn of the **digitization**. In HEP there are needed two different devices for the digitalization process. One is the *analog-to-digital converter* (ADC) which converts the pulse-like signal into a more efficient form of codification (binary code etc.). The other is the *time–to-digital converter* (TDC). The purpose of the TDC is to measure the difference between the signal produced by the event, and a reference signal (for example, the time between collisions) to use it for further calculations.

Level-1 Trigger

Normally, the event produced in HEP experiments occurs at very high frequencies and produce lots of data. In the case of the LHC, the proton beams will cross each other at a frequency of 40 MHz with an average of ~20 inelastic proton-proton events producing approximately 1 MB of data per collision. This data is too much for being stored, and not all events are actually interesting for the physicist. Therefore, the way of filtering this data is by using Triggers. Normally there are two levels of Trigger. In FE is implemented the first step (Level-1 Trigger). This Trigger is implemented by hardware, and it is designed to reduce the rate of events accepted for further processing from 40 MHz to less than 100 kHz. During the time interval that data is selected for passing the first Trigger and is sent to the BE systems, pipeline memories in the FE electronics (FIFOs etc.) are used for temporal storage.

Back-end

With the term *Back-end* (BE), we refer to the electronic devices and computers that are away from the detector. These electronic systems are far away from the radioactive environment of the detector chamber so in most of the cases can be implemented by *Components Of The Shelf* (COTS). Although every experiment at CERN has its own characteristics, the basic purposes of a BE system still the same in all the experiments. The main purposes of BE systems is to be responsible of the control, trigger and data acquisition of the HEP experiment.

Figure 1-13: CERN control centre.

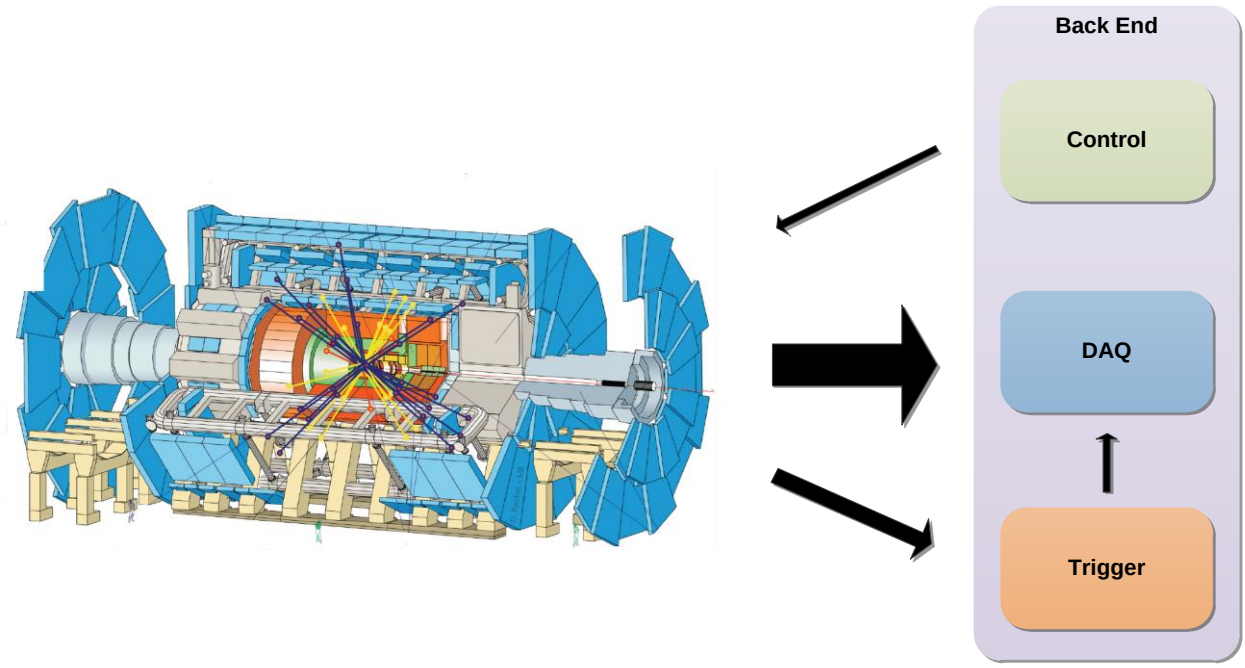


Figure 1-14: Control, DAQ and trigger.

Control

One of the crucial functions of BE is the control. This control is done by the *Detector Control System* (DCS). The task of the DCS is the operation and supervision of all detector components and the general infrastructure of the experiment. This means that the DCS is guarantees the safe operation of the detector to obtain high-quality physics data.

Trigger and Data AcQuisition (DAQ)

Trigger and Data AcQuisition (DAQ) systems of the BE play an essential role in HEP experiments. **Trigger** is responsible for real time selection of the subset of data to be recorded, while **DAQ** is responsible for collecting data from detector systems and recording them to mass storage for offline analysis. Normally, in HEP experiments, both the collision and the overall data rates are much higher than the rate at which one can write data to mass storage, even after the Level 1 Trigger of the FE. For that reason there is other Trigger implemented on BE, the *High-Level Trigger* (HLT), but in this case, this trigger is controlled by software. The HLT is designed to reduce the maximum Level-1 accept rate of 100 kHz to a final output rate of 100 Hz. The output data rate of

the Level 1 Trigger is equivalent to having one event every 10 μs. This time is sufficiently long for the storage of all data from the FE electronics to be feasible in commercial, random-access memories. The HLT process this data by using complex physics selection algorithms on the events read out, in order to accept the ones with the most interesting physics content, but also a small number of discarded event are accepted by the HLT as a sample. To be able to perform this process without losing data, DAQ systems must provide significant buffering capability.

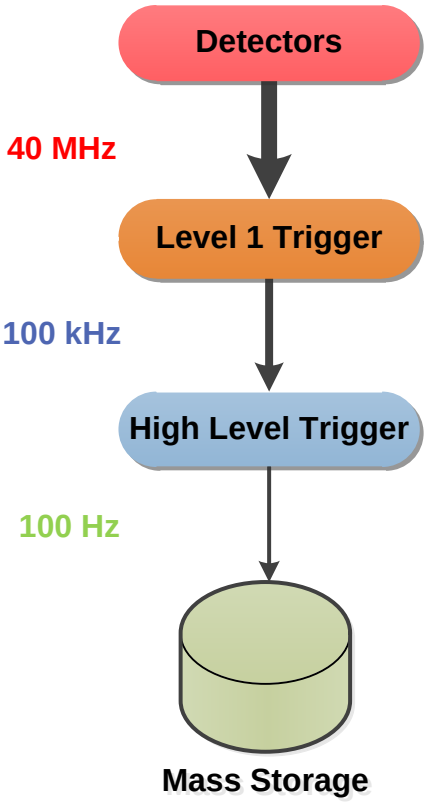


Figure 1-15: Trigger and DAQ data flow.

As it was said before, all data accepted by HLT has to be stored for further offline analysis. It is important to keep all this data because as much data collected as better. To have so much data means that there are more samples for the statistical studies and more opportunities of finding what the physicist are looking for. This data is then either stored directly in one of the mass storage devices that are in situ, in the computers farm near the experiment, or it can be sent online to one of the institutions which collaborates with the experiment to be stored on their mass storage systems.

Figure 1-16: CERN data centre

In the case of CERN and the LHC, the way of sharing online data for storage and analysis is by using the *Worldwide LHC Computing Grid* (WLCG) [11], a global network which at this moment counts with a collaboration of more than 140 computing centres in 35 countries, the 4 LHC experiments, and several national and international grid projects. Regarding number of people a community of more than 8000 physicists around the world use near real-time access to LHC data provided by the WLCG.

Figure 1-17: Worldwide LHC Computing Grid (WLCG).

Links between FE and BE

In HEP experiments, the distance between FE and BE systems is usually in the order of 100 m. For the communication between FE and BE a large number of optical fibres is used.

Current links

Although the current FE-BE link of each experiment at CERN has different characteristics with respect to the other experiments, all of them have the same structure and are based on the same concept.

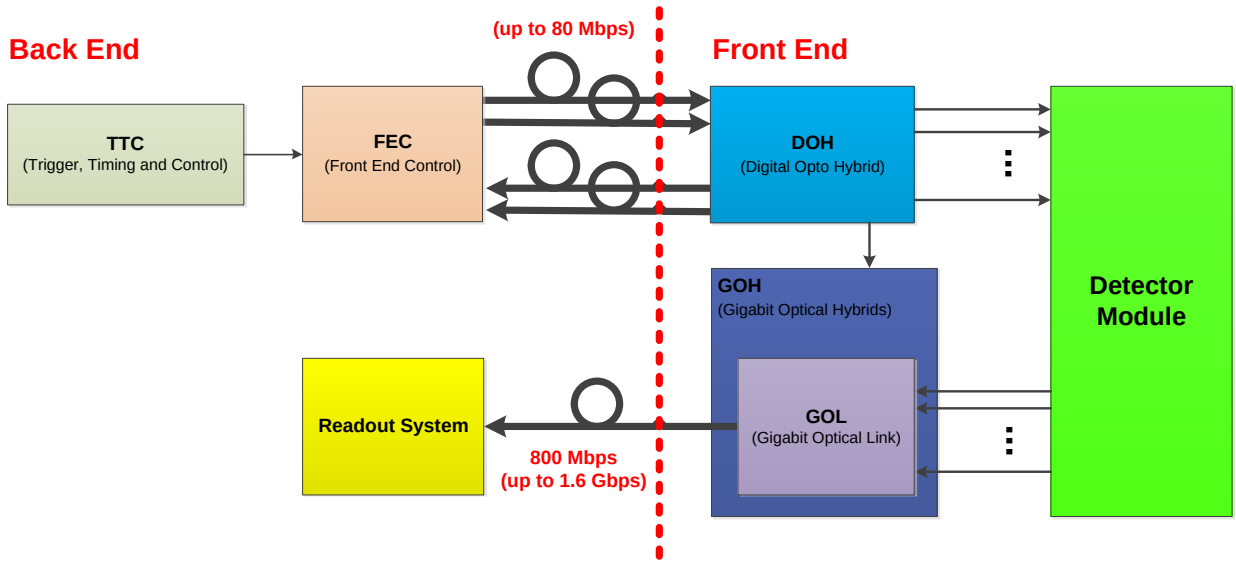


Figure 1-18: Current links FE-BE.

On the BE side, the *Trigger, Timing and Control* (TTC) system is connected to the *Front End Control* (FEC[(1)]) system which communicates with the FE through redundant optical links in both directions, sending and receiving information at a lane rate up to 80Mbps per channel. On the other hand, the *Digital Opto Hybrid* (DOH**)** system is the system responsible of the communication with the FEC[(1)] at the FE side. Regarding the DAQ system, data from the detectors is received through electrical links by the *Gigabit Optical Hybrids* (GOH) system and sent to the *Readout* system placed at the BE side by the *Gigabit Optical Link* (GOL) through optical links at a lane rate of 800 Mbps (up to 1.6 Gbps) and data payload of 640 Mbps (up to 1280 Mbps) respectively.

Future links (GBT)

With the installation of LHC and its associated experiments approaching completion, CERN and its collaborating institutes are now considering an upgrade of the accelerator to achieve higher beam luminosity, the *Super Large Hadron Collider* (SLHC) [12]. Higher luminosity will bring the benefit of improving the statistical accuracy of the measurements but it will also impose more stringent requirements on the performance of the data acquisition systems as well as on their radiation tolerant characteristics. Some of the systems and their components will have thus to be redesigned to cope with higher data rates and higher radiation levels. In particular, the data transmission links will have to be upgraded to wider bandwidths in order to cope with the larger amounts of physics data being produced by the detectors. The only way of achieving such higher data rates employing the technology that is used currently for data transmission links on the LHC would be by increasing the number of links, because of the low bandwidth of each link. Taking into account that currently solutions for transmission links are composed by three different channels (TTC, DAQ and *Slow Control* (SC)), the cost of the installation and maintenance would increase significantly. To increase the bandwidth without paying a penalty on the detector's material budget, it is necessary to use fewer optical links at higher data rates rather than simply increasing the number of links.

With these motivations in mind, the *microelectronic section of CERN* (PH-ESE-ME) started the development of the **GBT project** [13]. The new system implements bidirectional point-to-point optical fibre links between FE-BE at 4.8 Gbps (3.36 Gbps of payload). Such high bandwidth is a big achievement when working in radiation environments, it is achieved by not apply any redundancy technique on the data path between the input of the serializer and the output of the de-serializer like in traditional approaches. Instead, data is transmitted with a *Forward Error Correction code* ($FEC^{(2)}$). The other important improvement with respect to the traditional approach is that GBT links carry simultaneously DAQ, TTC and SC data. The high data rate and the unification of the traditional three channels represent a significant reduction of the overall system cost, easier maintenance and more reliability.

On the **BE side**, the receiver and transmitters of the GBT project will be implemented using COTS components and FPGAs. It is possible because these components work on a radiation free environment. The use of COST components reduces the price of the system, while the use of FPGAs with embedded transceivers will allow concentrating data from several FE sources into a single module, facilitating data merging and leading to compact systems.

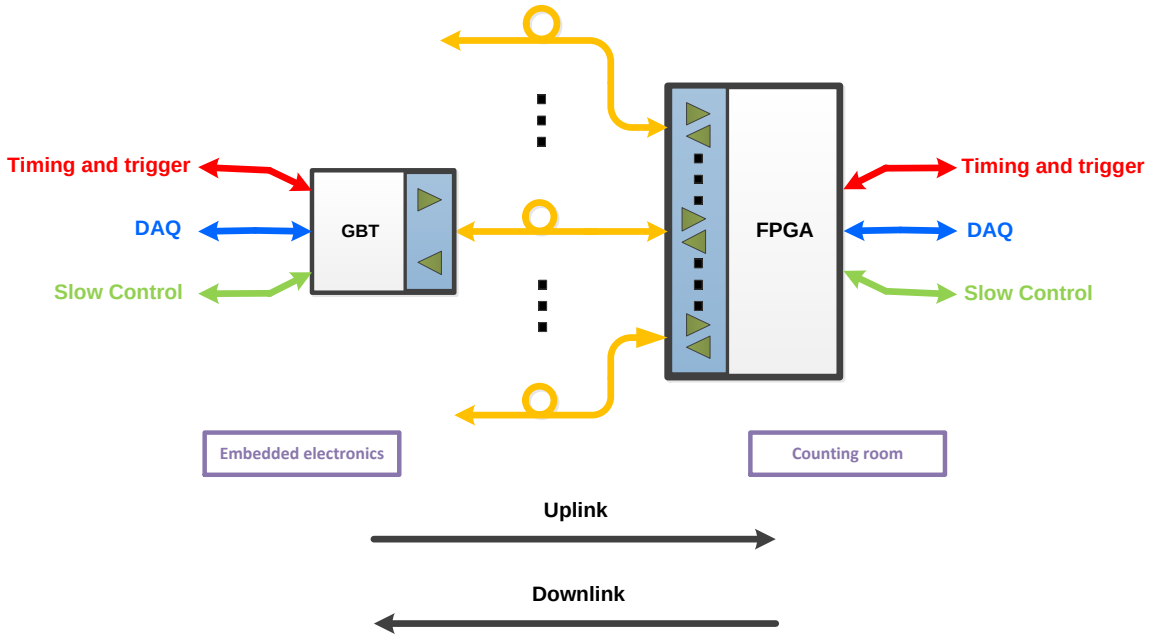


Figure 1-19: GBT based link architecture.

These BE systems mentioned above, have to be fully compatible with the **GBT protocol** which is used for communication between BE and FE systems. To understand how the bandwidth is allocated, it is necessary to consider in detail how the data is encapsulated when transmitted between the counting room and the detectors in what is called the GBT frame.

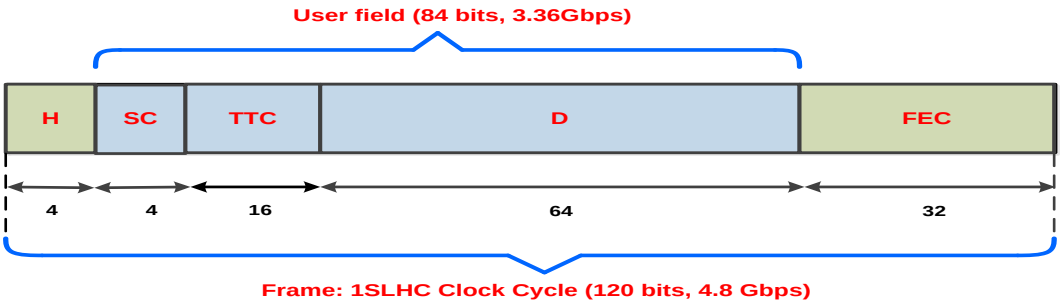


Figure 1-20: GBT frame.

As represented schematically in Figure 1-20, the **GBT frame** is composed of 120 bits which are transmitted during a single bunch crossing interval (≅ 25 ns) resulting in a line data rate of 4.8 Gbps . Of these, 4 bits are used for the frame Header (H) and 32 used for FEC[(2)]. This leaves a total of 84 bits free for data transmission corresponding to a user bandwidth 3.36 Gbps. The user field is distributed as follows: 4 bits for the SC (160 Mbps), 16 bits for TTC (640 Mbps) and 64 bits for Data (D) (2.56 Gbps)

On the **FE side**, the receivers and transmitters embedded on the detectors work in a hostile radiation environment requiring custom made components. The GBT chipset will be thus fabricated in a commercial 130 nm technology. This chipset consists on the GBT itself (GBT13) plus the *GigaBit Laser Driver* (GBLD) and the *GigaBit Transimpedance Amplifier* (GBTIA).

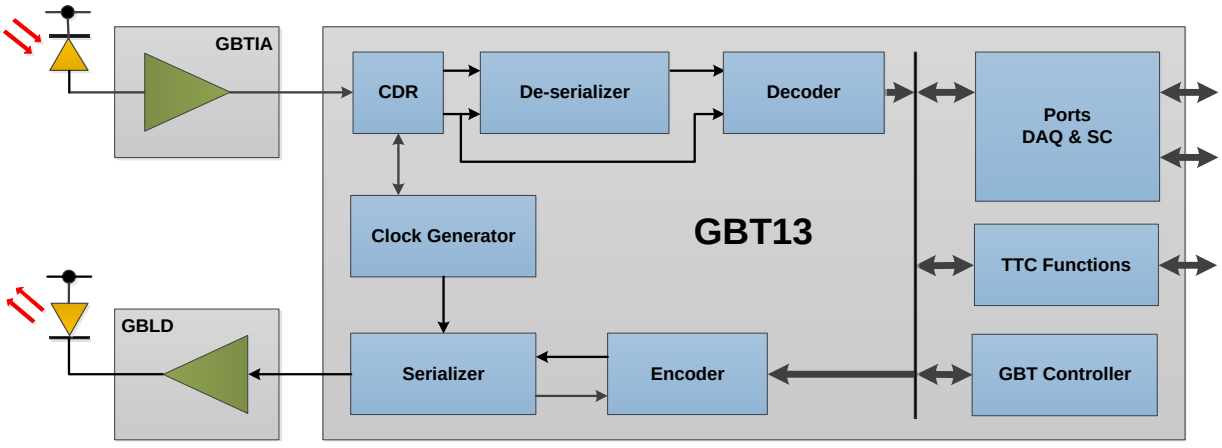


Figure 1-21: GBT chip set block diagram.

The block diagram of the GBT chipset (which is shown in Figure 1-21) is composed of the following parts: The **GBTIA** whose function is to convert the weak photocurrent generated by the photodiode into a digital differential voltage capable of

driving the *Clock and Data Recovery* (CDR) circuit of the GBT13. The **GBLD** uses the serial data produced by the *GBT Serializer* (SER) to modulate the laser diode optical power. The **GBT13**, which acts as a receiver and transmitter for the optical link, as a communications controller for the e-links and as a TTC receiver. The optical link receiver section of the GBT contains the CDR circuit, the *De-Serializer* (DES) and a *decoder* (DEC). The CDR circuit recovers the serial clock (4.8 GHz) from the incoming data while the DES circuit converts the 120 bits serial frame into a parallel word. The decoder detects and corrects any errors (which are within the error correction capabilities of the code) that have been introduced during the transmission. After error detection and correction, the decoder de-scrambles the data that is then feed to the "Ports" the "TTC functions" or the "GBT controller". On the optical link transmitter side, data is scrambled encoded and serialized. The GBT13 contains as well a clock generator that can be used as a clock source for "standalone" operation or as a phase noise reduction circuit for an external reference clock.

The GBT13 chip will connect with the FE ASICs through a set of 32 bi-directional serial links that are called *E-Links*. Typically, the available bandwidth of a GBT will exceed the throughput of single FE device being possible for a single GBT13 to serve several FE ASICs simultaneously. The general FE link topology using the E-links is schematized in Figure 1-22.

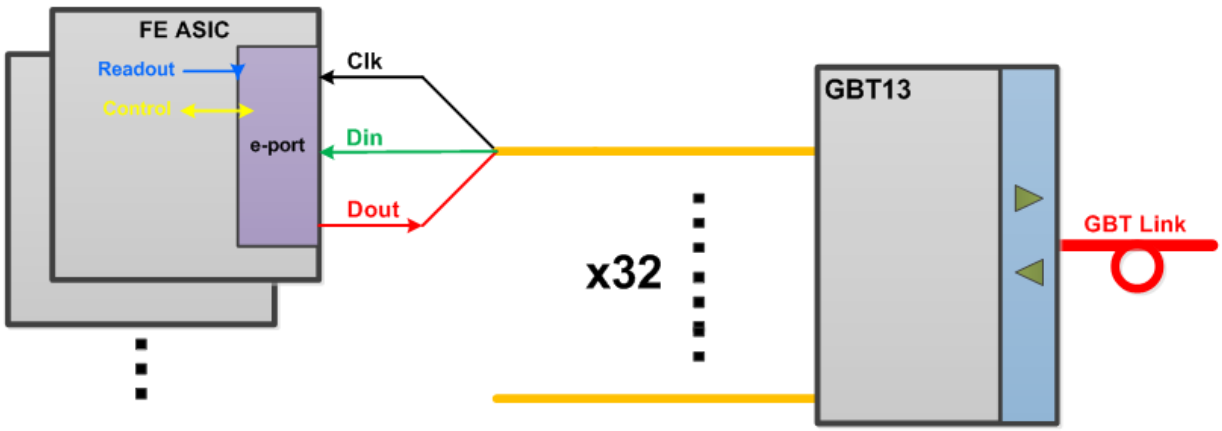


Figure 1-22: E-link.

1.2. Motivation

Together with the development of the GBT chipset which covers the requirements of the GBT link on the FE side, it is necessary to provide a hardware platform for the GBT link on the BE side. For that reason, an FPGA-based board interfacing with the GBT on the BE side has been developed in order to serve as a testing platform for this future link.

1.3. Requirements (Specifications)

The hardware requirements of the board are listed next:

- **GBT Link compliance:** It is required to implement GBT transmitter and receiver instances by using configurable logic on FPGA. Since the GBT link is used for *Trigger*, *Control* and DAQ, the board must comply the specific requirements of each mode of operation. On **Trigger and Control modes** it is necessary to perform a deterministic and low latency communication. On the other hand, on the **DAQ mode** it is necessary to count with enough memory to use as buffer of the received data.
- **GBT chipset simulation:** In some setups, the board is required to emulate the GBT chipset (e.g. for FE module testing purposes). In this mode of operation, the board should rely **only** in the clock recovered from the incoming data frame.
- **Modularity:** Due to the diversity of links (e-link, VTRx etc.) and technologies (LVDS, optical etc.) required on HEP experiments, expansion sockets (e.g. FMC) and pluggable optics in order to expand the I/O capability of the board, as required.
- **Crate operation:** As GBT interface on the BE side of the link, the board will operate inserted in an uTCA crate so it needs to comply the AMC specification.
- **Benchtop operation:** As test platform for the GBT link, the board will operate on a test bench so it is necessary to include at least one communication interface (PCIe, Gbe etc.) on the user side.

- **Clock versatility:** The new board will deal with different protocols, devices etc. Moreover, the inclusion of an FPGA will increase the possibilities of the board. Under these circumstances, it is necessary to take into account the clock versatility of the board.

Besides the hardware requirements, there are also other requirements that would be necessary to comply:

- Simplicity
- Low cost
- Easy maintenance
- User friendly

Chapter 2 - Gigabit Link Interface Board (GLIB)

2.1. Introduction

We have designed and built a platform for the evaluation of optical links in the laboratory as well as for triggering and/or data acquisition systems in beam or irradiation tests of detector modules. The **Gigabit Link Interface Board (GLIB)** [14] is based on an FPGA with *Multi-Gigabit Transceivers* (MGT) operating at rates of up to 6.5 Gbps. This performance matches comfortably the specifications of the GBT and Versatile Link projects with its targeted data rate of 4.8 Gbps. Figure 2-1 highlights the baseline configuration of a GBT - Versatile Link - GLIB system. On the left side, FE ASICs are electrically connected to the GBT ASIC through e-links while the GBT high-speed serial data-streams are converted to/from the optical domain through the *Versatile Transceiver* (VTRx). At the other end, the GLIB converts data to/from the optical domain, implements the GBT protocol and codes/decodes the user payload at the link BE.

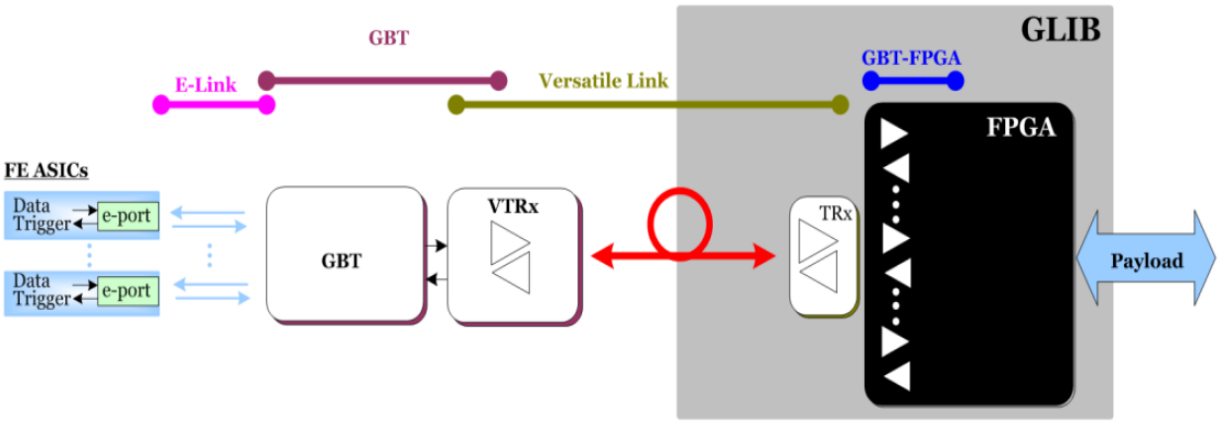


Figure 2-1: The GLIB board in a GBT-Versatile Link system.

The GLIB I/O capability can be further enhanced with two *FPGA Mezzanine Cards* (FMCs). This gives users the flexibility to adapt the GLIB interface to their system, by for instance adding connectivity to the TTC network at the BE, or connecting to e-links at the FE. Figure 2-2 illustrates a case where two GLIB boards are interconnected back-to-back, allowing implementing and experimenting with GBT-based systems well before full-fledged GBT ASICs become available.

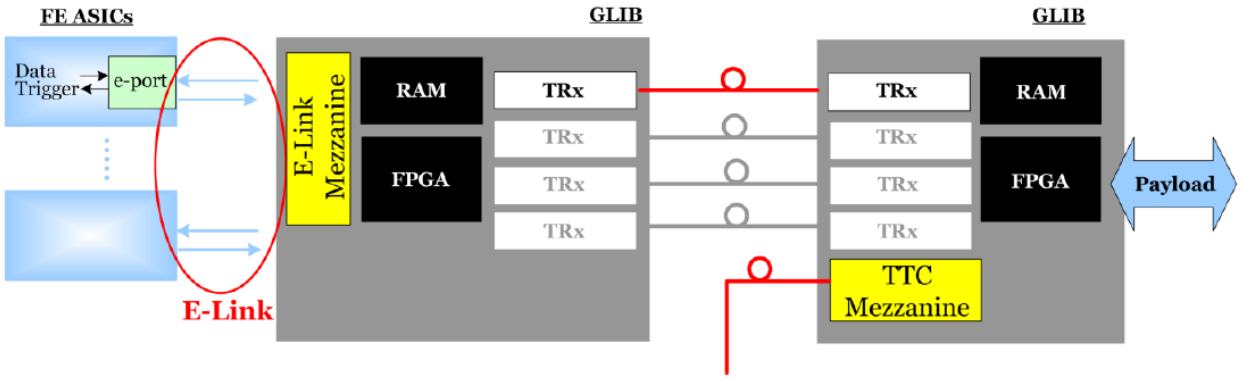


Figure 2-2: Back-to-back interconnected GLIB boards with customization mezzanines (TTC and E-Link in yellow).

2.2. Hardware

The GLIB is a compact board conceived in the form of a double wide *Advanced Mezzanine Card* (AMC) that serves a small and simple system operating either stand-alone or residing inside a micro *Telecommunications Computing Architecture* (µTCA) crate. Figure 2-3 shows a picture of the GLIB prototype highlighting its major components i.e. the Virtex-6 FPGA, the AMC connector, the cage for *Small Form Factor Pluggable Plus* (SFP+) optical modules, the FMC sockets, the SRAM devices, the socket for the *Module Management Controller* (MMC) mezzanine card and the 1000Base-T interface.

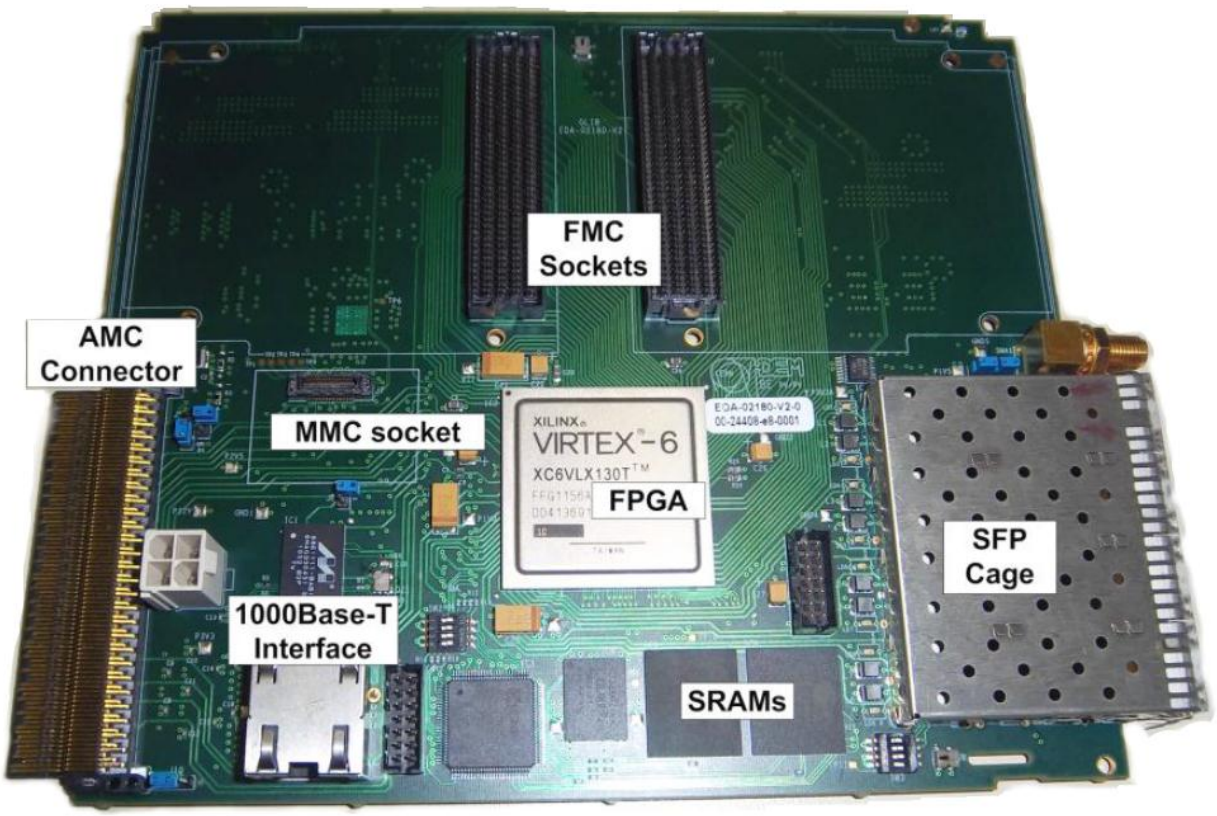


Figure 2-3: Picture of the GLIB prototype.

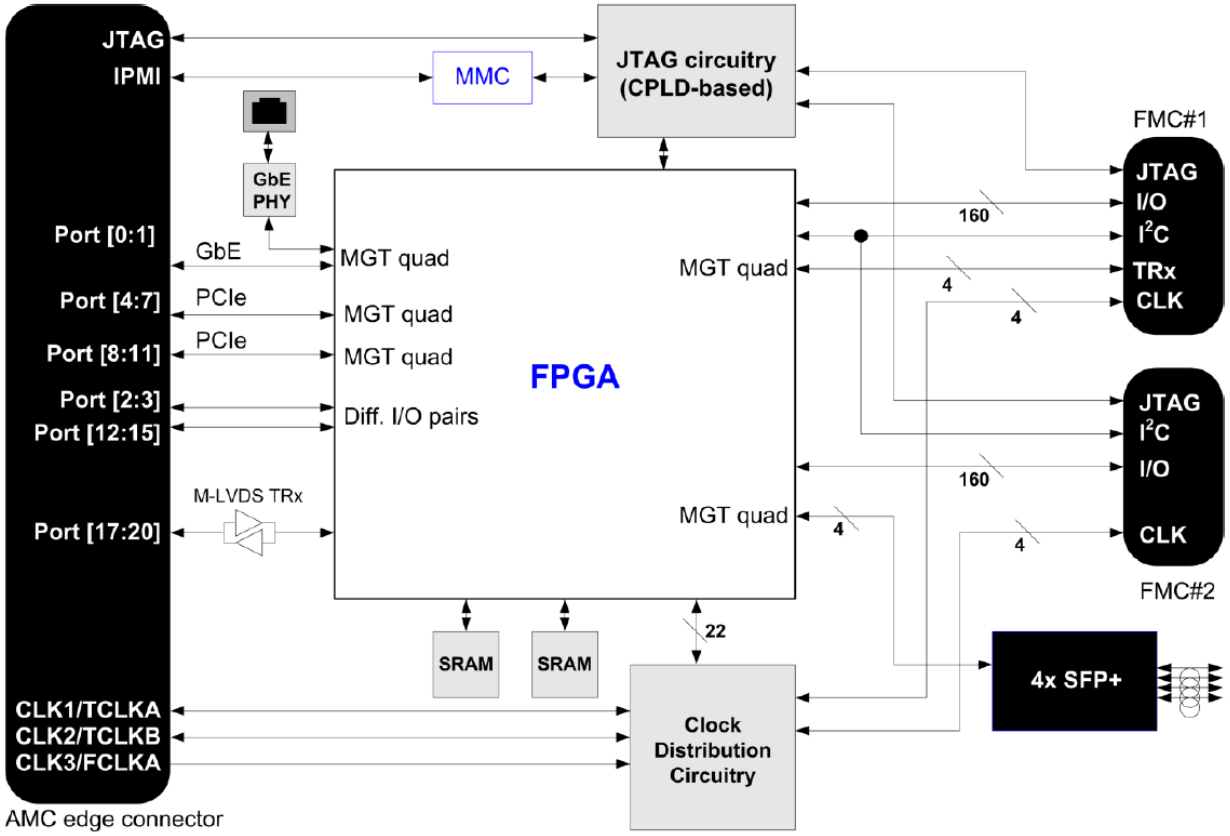


Figure 2-4: The Block diagram of the GLIB AMC card.

It is important to mention that the board shown in Figure 2-3 is the second version of the GLIB. There were various reasons why we have designed a second version, even though the first version was relatively successful. The most important one was to correct various errors in the schematic design (e.g. failure to program the flash memory used for the loading of the FPGA after power-up, problems accessing one of the two SRAM device as well as some other minor issues). Another reason was the decision to use a *Ball Grid Array* (BGA) package for the 1000Base-T physical layer device instead of the originally selected *Bump Chip Carrier* (BCC) because of assembly issues. The last reason was the modification of the MMC socket in order to accommodate the latest version of the MMC mezzanine.

2.3. Clocking scheme

The versatility of the clocking scheme on GLIB is an essential part of the system. Figure 2-5 shows a detailed diagram of the clock circuitry of the GLIB. The MGT reference clocks, fabric clocks and control signals are shown on the FPGA side in red, black and blue, respectively.

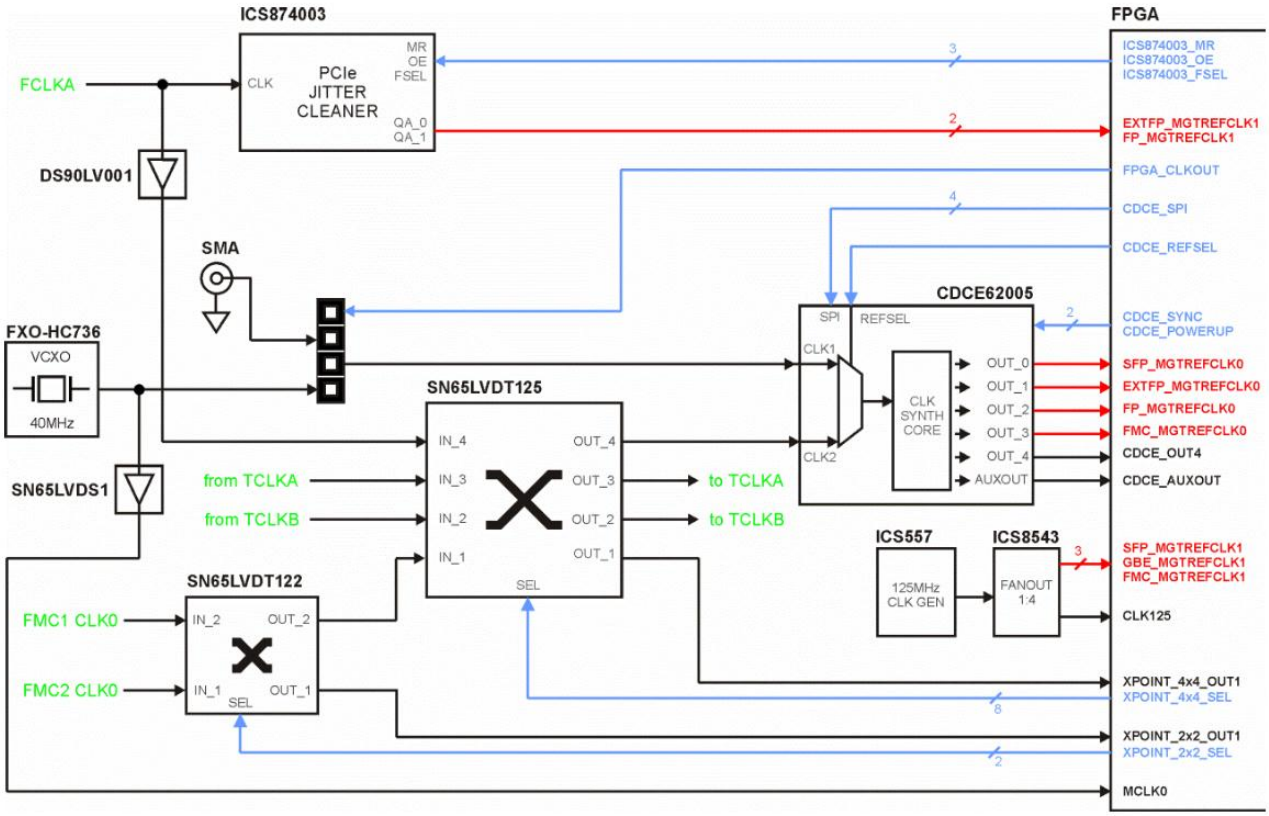


Figure 2-5: The clocking circuitry of the GLIB.

It is important to mention that each MGT Reference clock (REFCLK) can be used to clock the MGT of its neighbouring MGT Quads (see Figure 2-6). For instance, the REFCLK0 of the MGT 114 can also clock the MGT quads 113 and 115. Details about the Virtex-6 clocking resources are commented on Chapter 3 -.

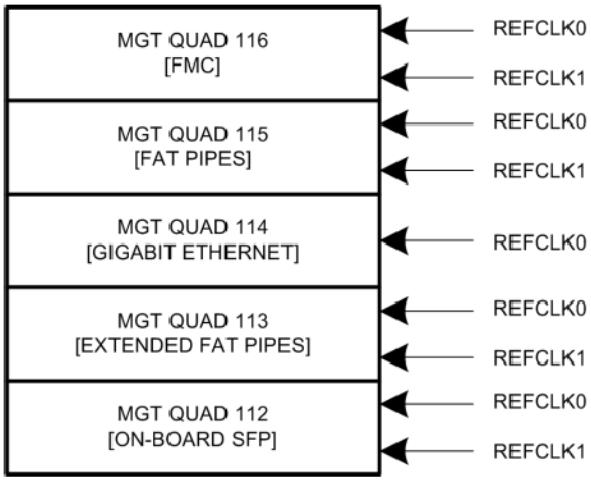


Figure 2-6: The MGT quads.

2.4. High speed optical links

For qualifying the high speed optical connectivity, we have measured the *Bit Error Rate* (BER) as a function of the *Optical Modulation Amplitude* (OMA) seen on the receiver side of the SFP+ module. The OMA is controlled by attenuating the optical output signal delivered by the transmitter side of the SFP+ while the BER is measured by using the built-in BER tester functionality of the FPGA. A *Pseudo-Random Bit Sequence* with 2^{31}-1 bits length (PRBS-31) has been used as the transmitted pattern for the tests. For all measurements, the same SFP+ module has been used. A diagram and a picture of the setup are shown in Figure 2-7 (top left and bottom left, accordingly). From the BER curves at 5 Gbps (Figure 2-7, right) we conclude that the performance is very satisfactory since errors appear only after a very significant attenuation (approx. 14 dB) of the optical signal (the OMA of the SFP+ output was approx. -4 dBm). The small variations between locations are due to differences in the routing of the printed circuit.

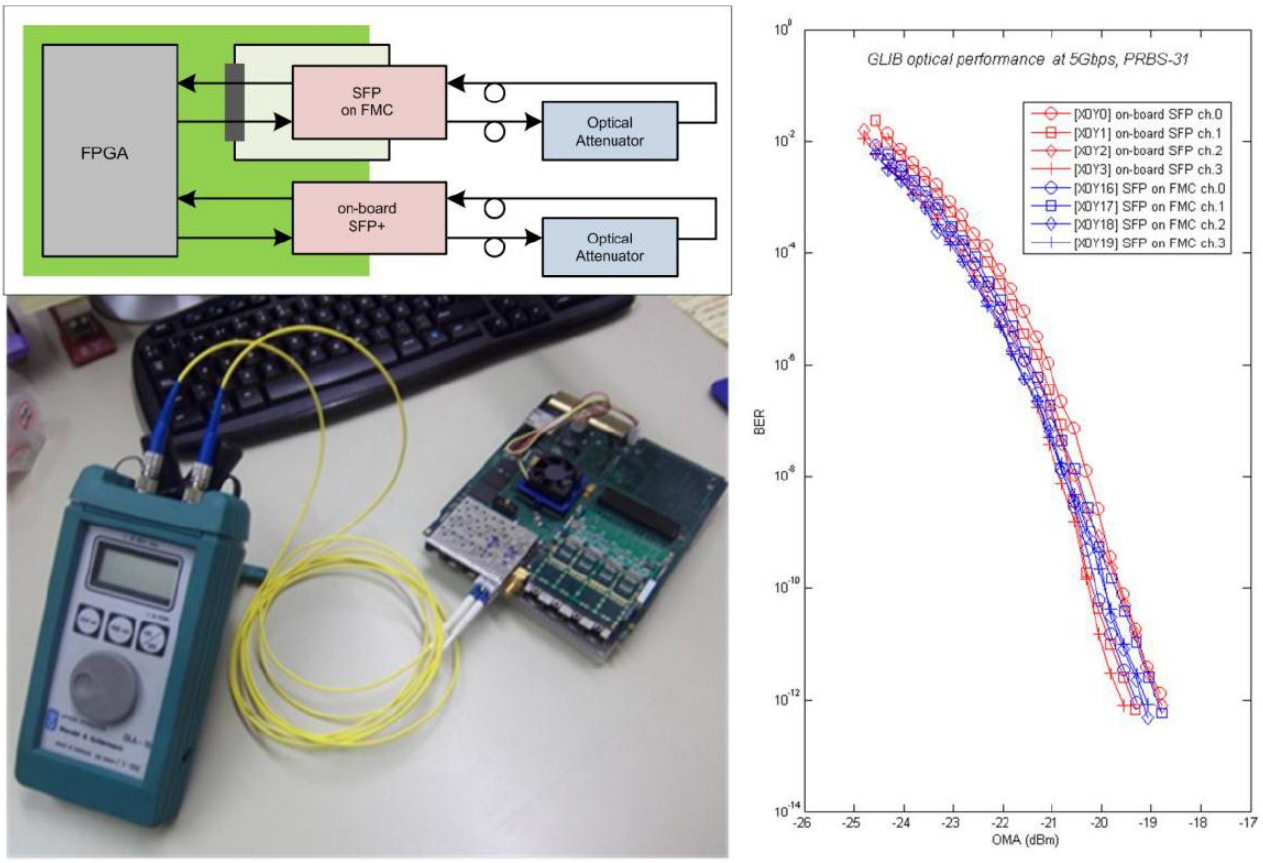


Figure 2-7: GLIB optical performance tests. Top left: A diagram of the test setup. Bottom left: Picture of the test setup. Right: the BER curves at 5 Gbps.

2.5. High speed electrical links

For the qualification of the AMC high speed serial connectivity, a commercial AMC breakout board carrying *Sub-Miniature version "A" radio-frequency connectors* (SMA) has been used. By connecting the GLIB to the AMC breakout board and using SMA cables to create loopback connections (Figure 2-8, left) we have measured that the links can run error-free during tens of hours at 5 Gbps using the built-in BER tester functionality of the Virtex-6 FPGA (Figure 2-8, right). The transmitted pattern used for the measurements was again PRBS-31.

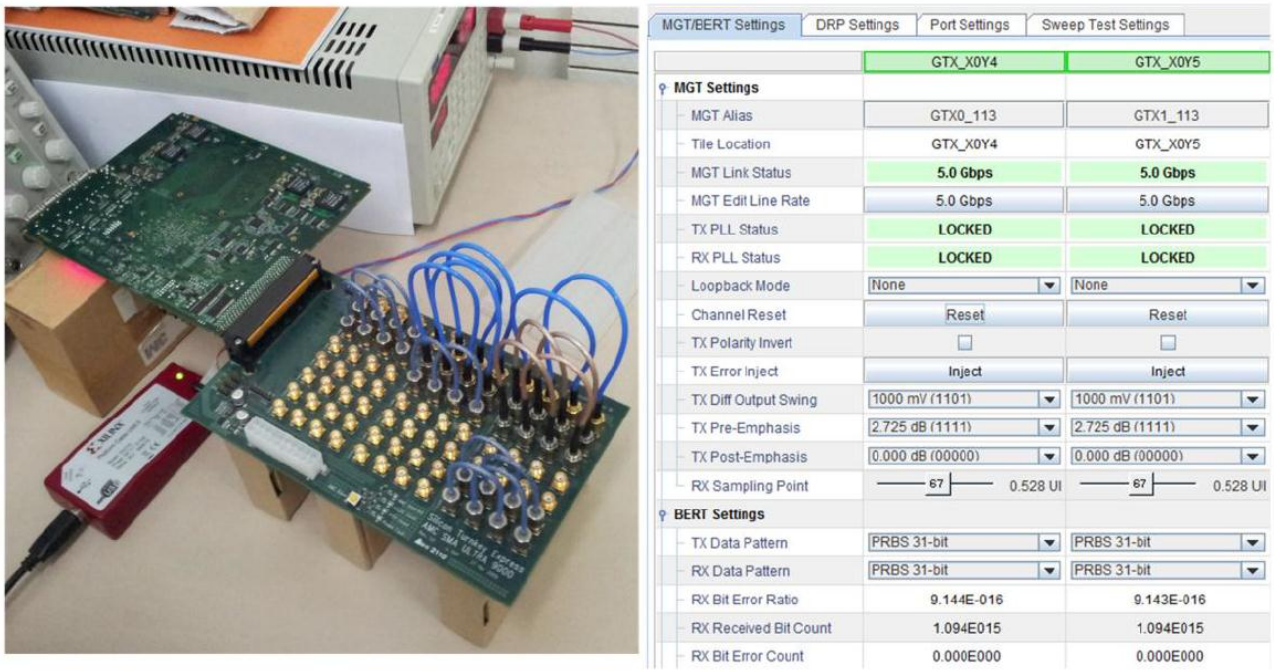


Figure 2-8: AMC high speed serial connectivity test.

The test setup used is considered realistic since the high-speed transmitted signals are travelling un-buffered through 2 AMC and 2 SMA connections (Test setup 1 on Figure 2-9) which is comparable with the connectivity path in a µTCA crate where the signals are travelling un-buffered through only 2 AMC connections (real case on Figure 2-9).

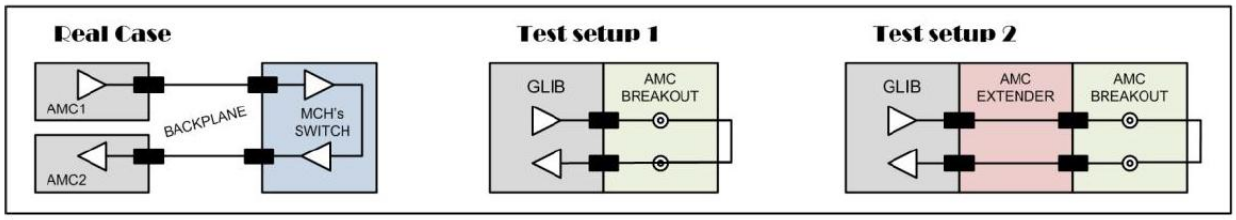


Figure 2-9: AMC high speed serial connectivity test setups.

The performance deteriorated (error-free operation at up to 3.125 Gbps) only after using an unrealistic setup where the high-speed transmitted signals are travelling un-buffered through 4 AMC and 2 SMA connections by introducing an AMC extender between the two boards (Figure 2-10 and Test setup 2 on Figure 2-9).

Figure 2-10: AMC high speed serial connectivity test setup 2.

2.6. Firmware

Concerning the FPGA firmware, the development is currently focused on the stand-alone operation of the board controlled by a Gigabit Ethernet link. The GLIB firmware instantiates a simple *IP-based control protocol* (IPbus) designed for controlling xTCA-based hardware over Gigabit Ethernet that includes all basic transactions needed for this purpose (bitwise, single register and block transactions). The firmware also instantiates all necessary interfaces to the external hardware including I^2C communication with the on-board temperature sensors and the serial memory, SPI communication with clock synthesizer, parallel memory interface with the two SRAMs operating at 160 MHz as well as FLASH memory interface with multiboot and fallback features. Additionally, the firmware instantiates a number of GBT *Hardware Description Language* (HDL) modules that together with the associated MGTs are used for interfacing with GBT protocol based systems. It is important to mention that the firmware architecture is designed in such way that allows the user to add his logic without the need of editing anything outside the user logic block. Figure 2-11 illustrates the firmware architecture.

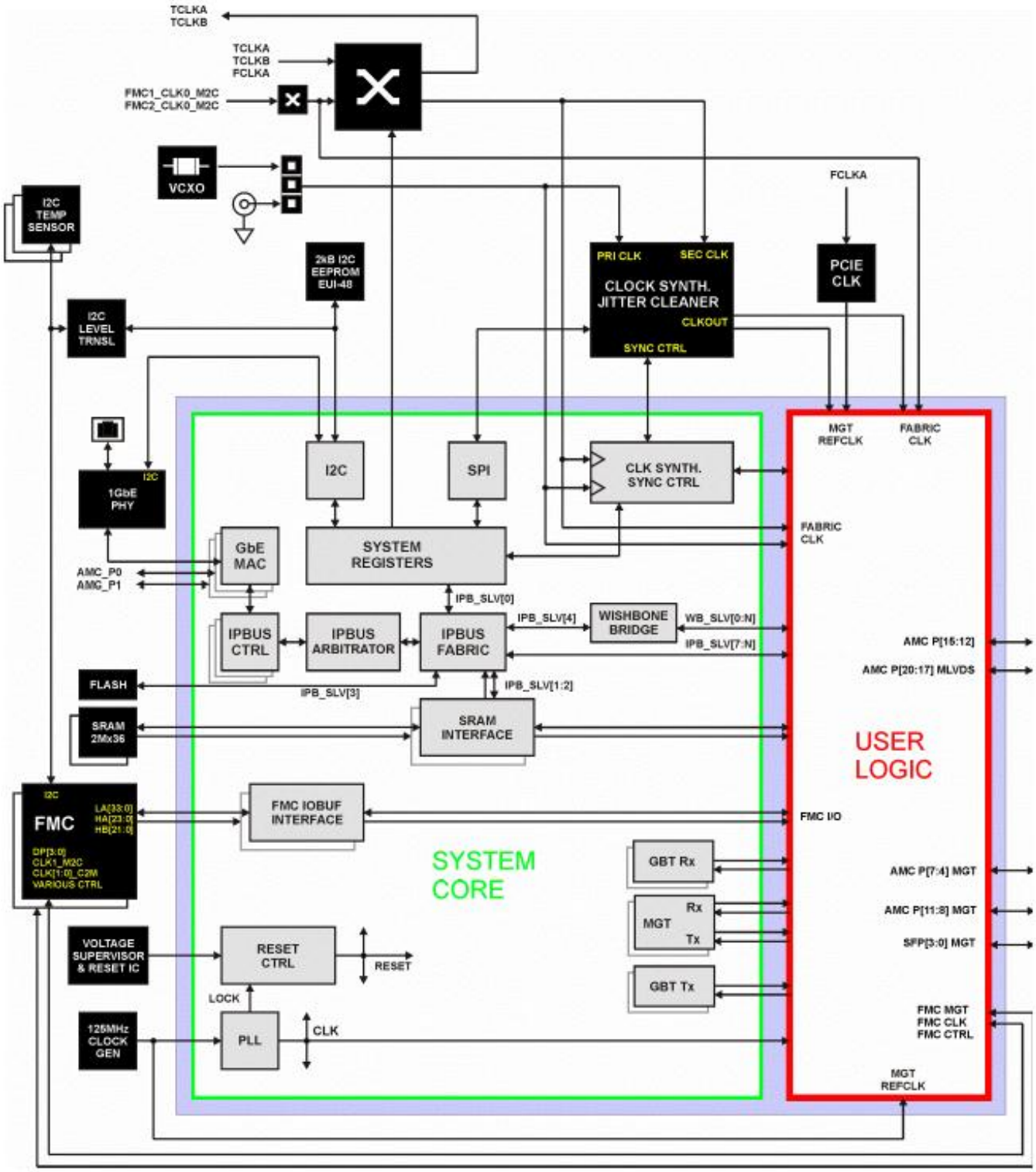


Figure 2-11: Firmware architecture.

2.7. Software

At this point of the software development, the communication between the GLIB board and the computer of the user is based on PyChips [15], a Hardware Access Library (HAL) written in Python [16], which provides functions to enable transactions with the hardware attached to the IPbus (included in the GLIB firmware) via Gigabit Ethernet. The functions provided by PyChips are called by scripts written in Python and executed on the Command Prompt of a Windows operating system (7, Vista, XP etc.).

```
1   from PyChipsUser import *
2   import os
3   glibAddrTable = AddressTable("./glibAddrTable.dat")
4   print
5   print "--========================================--"
6   print "  Opening GLIB with IP 192.168.0.111"
7   print "--========================================--"
8   glib = ChipsBusUdp(glibAddrTable, "192.168.0.111", 50001)
9
10  print
11  print "-> BOARD INFORMATION"
12  print "-> -----------------"
13
14  brd_char    = ['w','x','y','z']
15  brd_char[0] = chr(glib.read("board_id_char1"))
16  brd_char[1] = chr(glib.read("board_id_char2"))
17  brd_char[2] = chr(glib.read("board_id_char3"))
18  brd_char[3] = chr(glib.read("board_id_char4"))
```

Figure 2-12: Python script.

```
E:\glib_project\amc_glib\trunk\software\PyChips_1_4_1\scripts>glib_board_info.py
PyChips v1.4.1

--========================================--
  Opening GLIB with IP 192.168.0.111
--========================================--

-> BOARD INFORMATION
-> -----------------
-> brd : G L I B
-> sys : L A B
-> ver : 1 . 0 . 5
-> date: 19 / 2 / 2012
PyChips v1.4.1

-> hardware ID: 00 04 a3 2e 34 a8

-> BOARD STATUS
-> -----------------
-> sfp1 absent      : 1
-> sfp1 rxlos       : 1
-> sfp1 txfault     : 1
-> sfp2 absent      : 1
-> sfp2 rxlos       : 1
-> sfp2 txfault     : 1
-> sfp3 absent      : 1
-> sfp3 rxlos       : 1
-> sfp3 txfault     : 1
-> sfp4 absent      : 1
-> sfp4 rxlos       : 1
-> sfp4 txfault     : 1
-> ethphy interrupt: 0
-> fmc1 presence    : 0
-> fmc2 presence    : 1
-> fpga reset state: 0
-> cpld bus state   : 0a
-> cdce locked      : 0

-> done

--========================================--
```

Figure 2-13: Windows command prompt.

Apart from the standard IPbus software framework distribution, we are developing a stand-alone JAVA-based Graphical Users Interface (GUI). This light-weight GUI provides to the users the ability to configure the board according to their needs in a simple way under any operating system. Screenshots of the GUI are shown in Figure 2-14: Snapshots from the Graphical Users Interface.

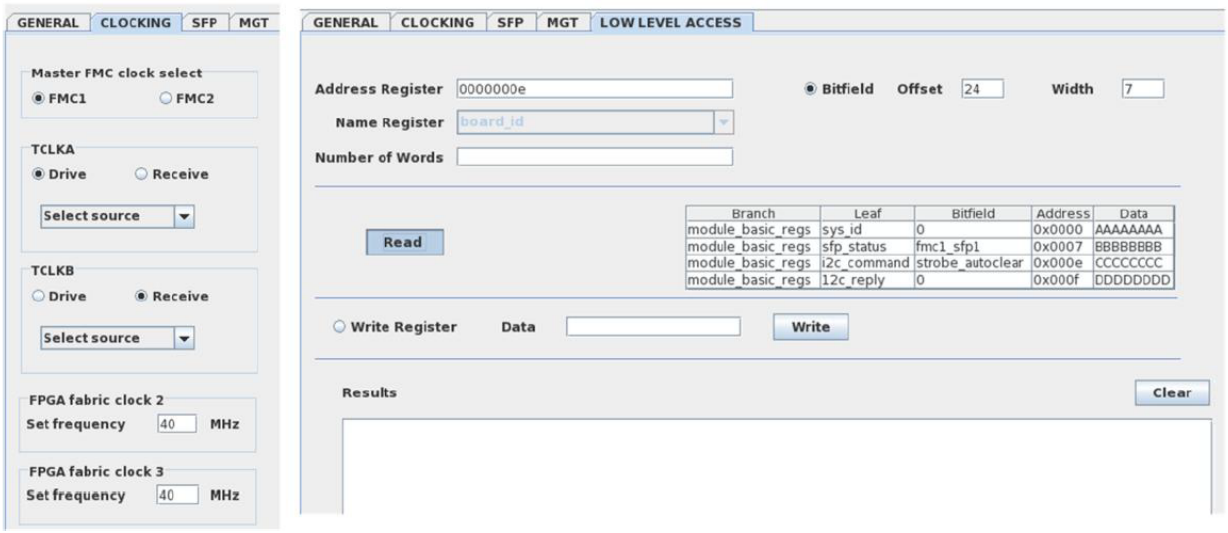


Figure 2-14: Snapshots from the Graphical Users Interface.

2.8. Development of auxiliary cards

Within the frame of the GLIB project, some auxiliary boards (mainly FMCs) are currently under development. One of them is the **Timing, Trigger and Control (TTC) FMC** developed based on a commercial Clock & Data Recovery (CDR) integrated circuit for interfacing the GLIB with the TTC system used currently in the LHC experiments. This board receives the 160 Mbps bi-phase mark encoded bit-stream from an optical source, extracts the clock and the data and forwards them to the FPGA of the carrier card for further decoding of the data stream. Another board under development is the **VTRx FMC**, a mezzanine card that interfaces the various flavours of VTRx with the GLIB system. A third system which is currently under development is an **AMC to PCI Express x4 adapter**, that provides a high bandwidth interconnection between a stand-alone GLIB and a PC.

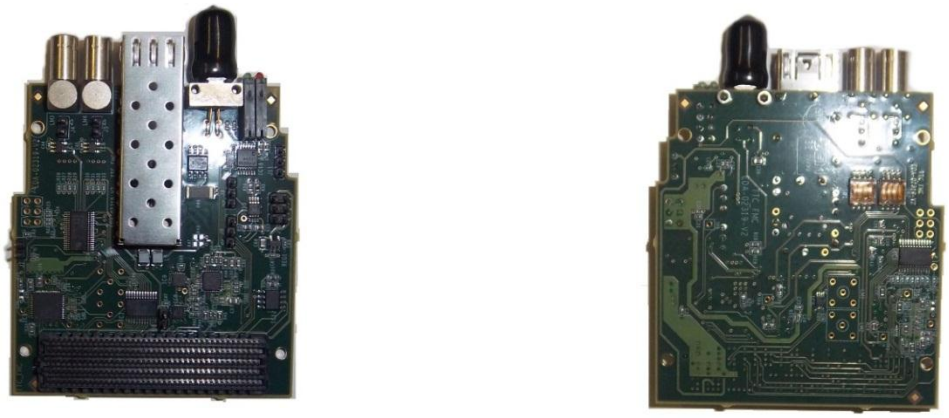


Figure 2-15: TTC FMC prototype (v2).

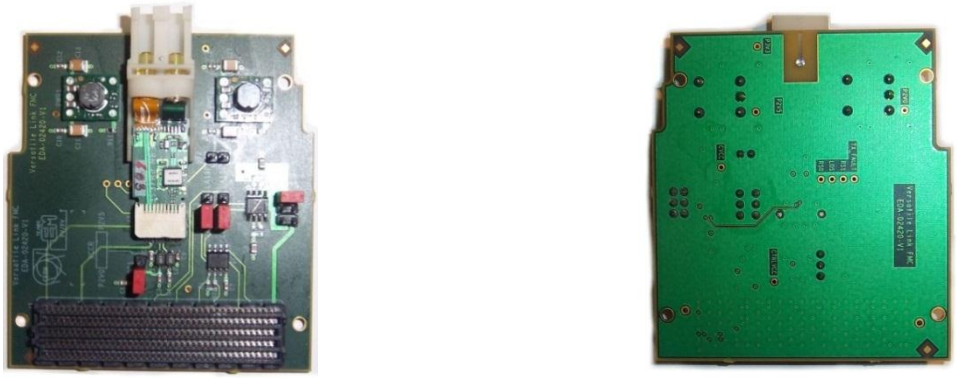


Figure 2-16: VTRx FMC prototype.

2.9. Where this thesis (PFC) is focused on

Although I have been involved on the development and testing of different modules of the firmware for the GLIB project, this thesis (PFC) is focused on a project that emulates realistically a GBT link by using two GLIB module, where one of them operates as a typical BE card while the second one plays the role of a GBT chipset sitting at the FE. There are three main reasons why this project has been selected to be explained more deeply. The first reason is because I was the main developer. The second reason is the importance of a realistic GBT link simulation for the GLIB project. Finally, the third reason is that due to the variety of components implied (HDL modules, ICs, serial transceivers etc.) this project is very didactic and a very good example of HDL modules and IC devices interaction.

Chapter 3 - Back-end/Front-end link simulation

3.1. Objectives

The objective of this project is to perform an error-free communication with low and deterministic latency between a GLIB that is clocked by an external reference clock with another GLIB that recovers the reference clock based on the received data-stream, thus simulating a Back-end/Front-end link where the “BE” board is clocked externally while the “FE” board is clocked from the incoming optical data.

The procedure for simulating a BE-FE communication is the following:

1. The “BE” GLIB transmits GBT packets.
2. The “FE” GLIB starts-up using its local oscillator, recovers the clock from the data received, it feeds the recovered clock to the clock synthesizer and switches its input clock source. Now the board has exactly the same clock as the “Back-end” GLIB.
3. The “FE” GLIB transmits packets and the “BE” GLIB receives them.

The link is up with two boards operating with exactly the same clock frequency.

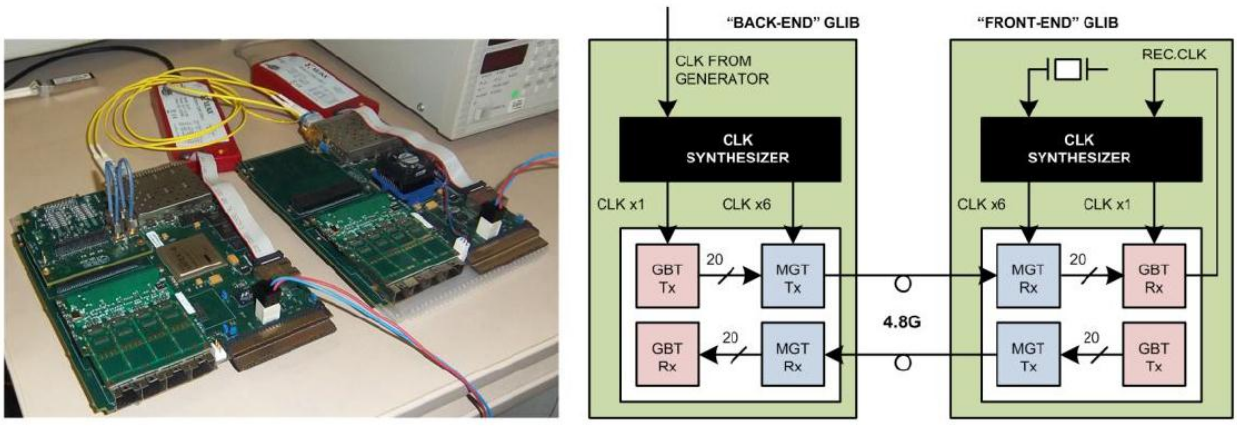


Figure 3-1: Picture (left) and block diagram (right) of the setup with two GLIB boards.

3.2. BE/FE link simulation components

In this section are described the devices and HDL modules implied on the BE-FE simulation project. The features related with the purpose of this project are described more deeply for further understanding. Other features that are not related with this project can be briefly described, just mentioned, or omitted. For a complete description, it is advised to consult the data sheet or documentation provided by the vendor/developer of the device/module. It is highly recommended the lecture of this section for further understanding of the project. It can also be used as reference document.

Table 3-1: Devices and HDL modules utilised on the BE-FE simulation project.

Clock source	Voltage Controlled Crystal Oscillator (FXO-HC736R-40)
	Single ended to LVDS driver (SN65LVDS1)
Crosspoint switches	Crosspoint switch 2x2 (SN65LVDT122)
	Crosspoint switch 4x4 (SN65LVDT125)
Clock synthesizer (CDCE62005)	
FPGA Virtex-6 (XC6VLX130T-1FF1156)	
GTX transceiver	
GBT HDL module	
Other HDL modules and IP cores	See list on page 88
Optical transceiver (TRX10GDP0310)	
Optical fibre cable	
FMC board for external clock feedback (FMC DIO 16CH TTL A)	
AMC breakout board (SMA-AMC Ultra 9000)	
Laboratory equipment	Power supply (TTi TSX 1820P)
	USB to JTAG interface (Xilinx Platform Cable USB II)
	Oscilloscope (LeCroy WaveRunner LT354M)
Software	Project Navigator, Core Generator and ChipScope Pro Analyzer

3.2.1. Clock source

Voltage Controlled Crystal Oscillator (FXO-HC736R-40)

The FXO-HC736R-40 [17] is a 40 MHz *Voltage Controlled Crystal Oscillator* (VCXO) of the family XPRESSO designed and developed by Fox. This VCXO utilizes a family of proprietary ASICs with a key focus on noise reduction technologies. The 3rd order Delta Sigma Modulator reduces noise to the levels that are comparable to traditional Bulk Quartz and SAW oscillators. The ASICs family has ability to select the output type, input voltages, and temperature performance features.

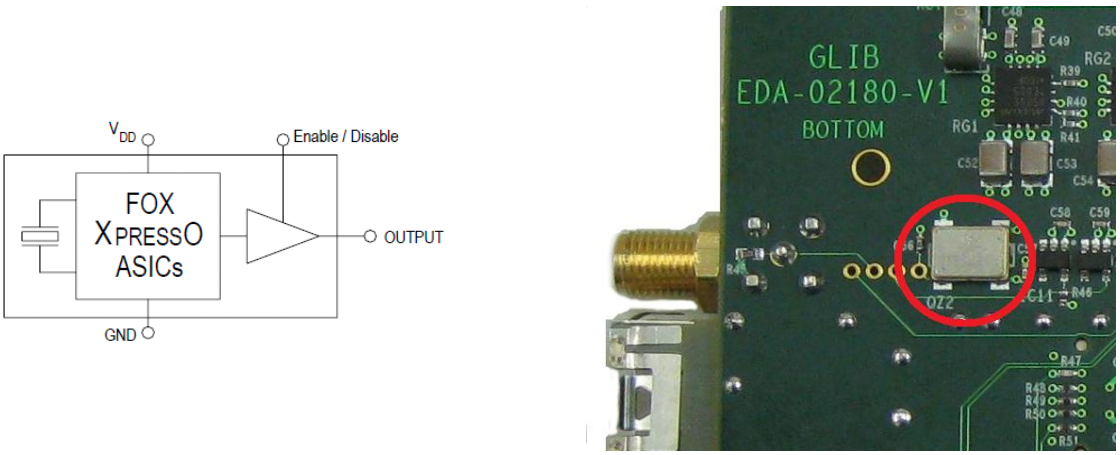


Figure 3-2: VCXO block diagram (right) and picture (left).

Single ended to LVDS driver (SN65LVDS1)

The SN65LVDS1 [18] is a single ended to differential driver in the small-outline transistor package. The output comply with the TIA/EIA-644A standard and provide a minimum differential output voltage magnitude of 247 mV into a 100 Ω load at signalling with rates up to 630 Mbps.

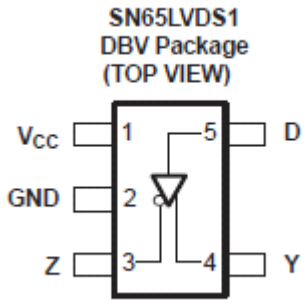


Figure 3-3: LVDS driver diagram.

3.2.2. Crosspoint switches

Crosspoint switch 2x2 (SN65LVDT122)

The SN65LVDT122 [19] is a crosspoint switch that uses *low voltage differential signalling* (LVDS) to achieve signalling rates as high as 1.5 Gbps. The internal signal paths maintain differential signalling for high speeds and low signal skews. This device has a 0 V to 4 V common-mode input range that accepts LVDS, LVPECL, or CML inputs. Two logic pins (S0 and S1) set the internal configuration between the differential inputs and outputs. Additionally, the SN65LVDT122 incorporates a 110 W termination resistor for those applications where board space is a premium. Although this device is designed for 1.5 Gbps, some applications at a 2 Gbps data rate can be supported depending on loading and signal quality.

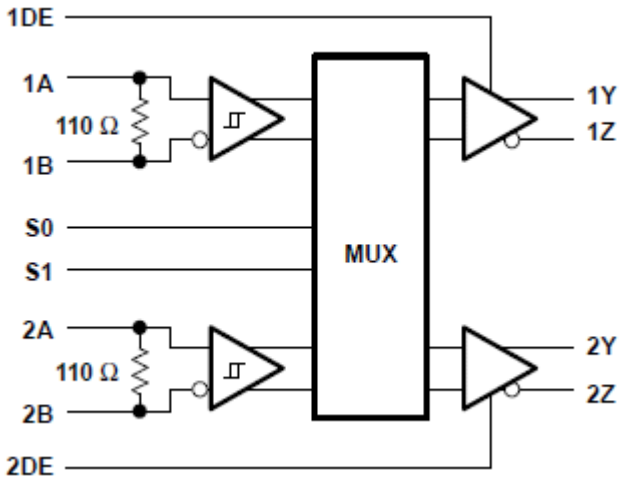


Figure 3-4: Crosspoint switch 2x2 diagram.

Crosspoint switch 4x4 (SN65LVDT125)

The SN65LVDT125 [20] is a 4x4 nonblocking crosspoint switch. LVDS is used to achieve signalling rates of 1.5 Gbps per channel. Each output driver includes a 4:1 multiplexer to allow any input to be routed to any output. Internal signal paths are fully differential to achieve the high signalling speeds while maintaining low signal skews. The SN65LVDT125 incorporates 110 Ω termination resistors for those applications where board space is a premium.

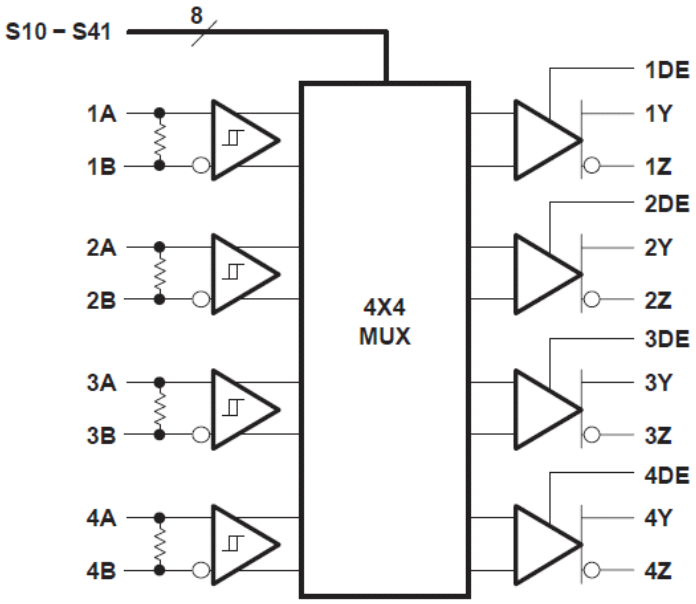


Figure 3-5: Crosspoint switch 4x4 diagram.

3.2.3. Clock synthesizer (CDCE62005)

General description

The CDCE62005 [21] is a high performance clock generator, clock distributor and jitter cleaner (output jitter under 1 ps rms). This device has been selected for the GLIB project because it includes a feature that allow synchronize all output clock with the input reference clock. Doing this, **the phase relationship between the input reference clock and the output clocks is always deterministic**, even after power up (it is also known as *zero-delay*). This feature is needed for fixed latency links because it ensures that the reference clock has deterministic phase.

However, after an intensive test of this zero-delay feature, the result of the test showed that sometimes the synchronization is not performed correctly. The workaround of this problem will be commented in chapter 4.2.

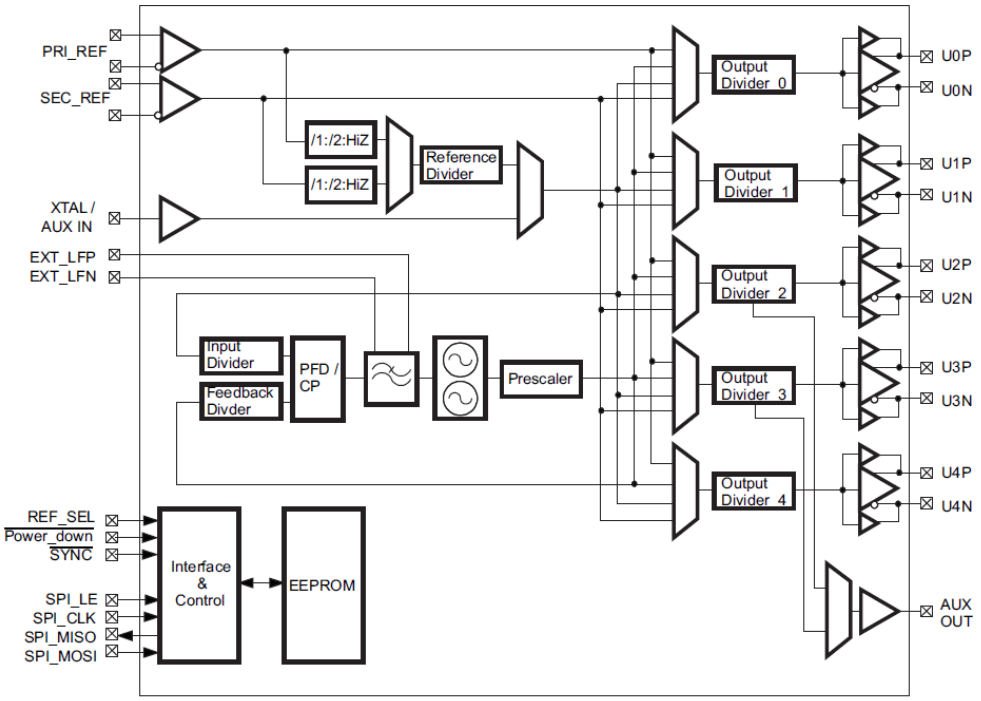


Figure 3-6: CDCE62005 block diagram.

Features

The main features of the CDCE62005 are listed below:

- Multiple operational modes:
  - Clock generation via crystal
  - SERDES start-up mode
  - Jitter cleaning
  - Oscillator holdover mode
- 3 clock inputs:
  - Two universal differential inputs accept frequencies in the range of 40 kHz to 1500 MHz (LVPECL), 800 MHz (LVDS), or 250 MHz (LVCMOS)
  - One auxiliary input accepts crystals in the range of 2 MHz–42 MHz
- Smart Input Multiplexer
- Frequency synthesizer including PLL, multiple VCOs, and loop filter:
  - Dual-VCO architecture supports a wide tuning range 1.750 GHz–2.356 GHz
  - Programmable charge pump gain
  - Programmable loop filter settings
- Fully configurable outputs (frequency, output format, and output skew):
  - Support up to 5 differential, 10 single-ended, or combinations of both
  - Output frequency ranges from 4.25 MHz to 1.175 GHz in synthesizer mode
  - Output frequency up to 1.5 GHz in fan-out mode
  - LVPECL, LVDS, LVCMOS, and special high output swing modes
  - Independent output dividers. Support divide ratios from 1–80 (non-continuous values supported)
  - Independent coarse skew control on all outputs (coarse skew control does not operate for reference input frequencies < 1 MHz)
  - Input/output clocks phase synchronization
  - 0.35 ps rms output jitter performance (from 10 kHz to 20 MHz)
  - Low output phase noise: –130 dBc/Hz at 1 MHz offset, Fc = 491.52 MHz

- Integrated EEPROM
- QFN-48 Package
- ESD Protection Exceeds 2 kV HBM
- Industrial Temperature Range –40 °C to 85 °C
- Typical Power Consumption 1.7 W at 3.3 V
- Applications:
    - Data converter and data aggregation clocking
    - Wireless Infrastructure
    - Switches and Routers
    - Medical Electronics
    - Military and Aerospace
    - Industrial
    - Clock Generation and Jitter Cleaning

Functional description

The CDCE62005 is composed by four primary blocks. These blocks are briefly described below.

Input block

The input block includes two universal differential inputs and an auxiliary input that can be configured to connect to an external crystal via an on chip oscillator block. The smart input multiplexer has two modes of operation, manual and automatic. In manual mode, the user selects the synthesizer reference via the SPI interface. In automatic mode, the input multiplexer will automatically select between the highest priority input clock available.

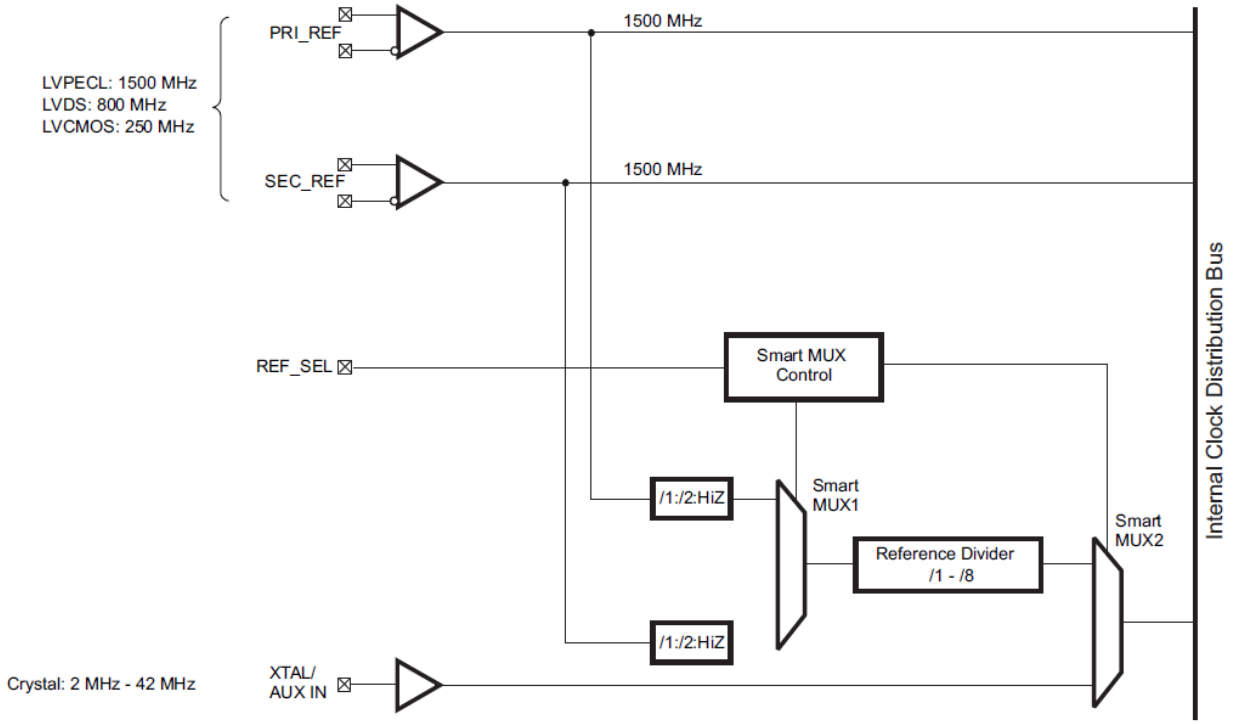


Figure 3-7: CDCE62005 input block.

Synthesizer block

In this block, the clock from the smart mux goes through an input divider, and it is compared with the feedback clock. This task is done by the integrated phase detector. Once the phase has been corrected, this signal is used for feeding one of the two internal *voltage controlled oscillators* (VCO). This is one of the most important features of the CDCE62005 because it allows the user to avoid the use of an extra external VCO like in other clock generators. After the VCO, the frequency of the signal is divided by the pre-scaler before the output block. In this block is also included a partially integrated loop filter but for some frequencies is highly recommended to use external components.

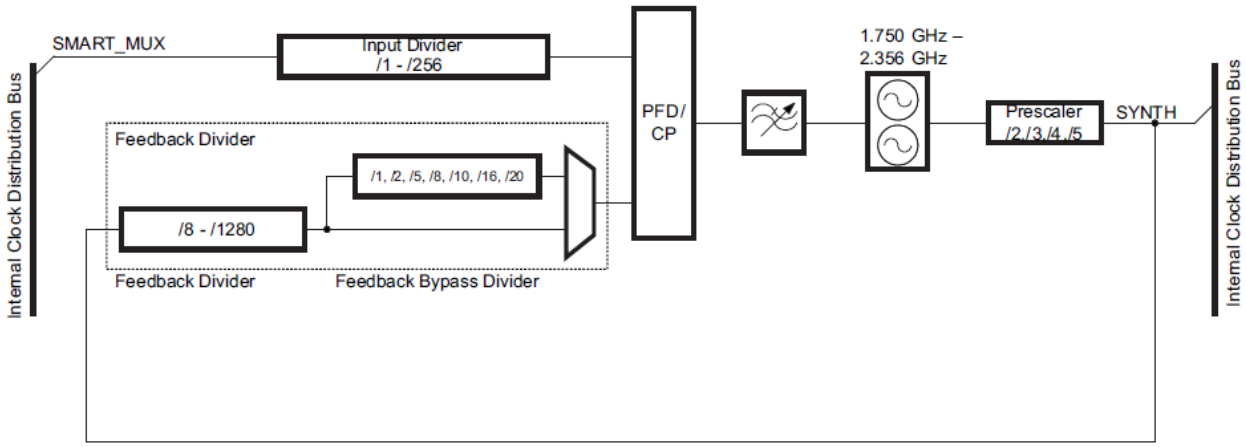


Figure 3-8: CDCE62005 synthesizer block.

Figure 3-9 presents the block diagram of the CDCE62005 in synthesizer mode highlighting the clock path for a single output. It also identifies the following regions containing dividers comprising the complete clock path:

- R: Includes the cumulative divider values of all dividers included from the Input Ports to the output of the Smart Multiplexer
- O: The output divider value
- I: The input divider value (see Synthesizer Block for more details)
- P: The Pre-scaler divider value (see Synthesizer Block of more details)
- F: The cumulative divider value of all dividers falling within the feedback divider

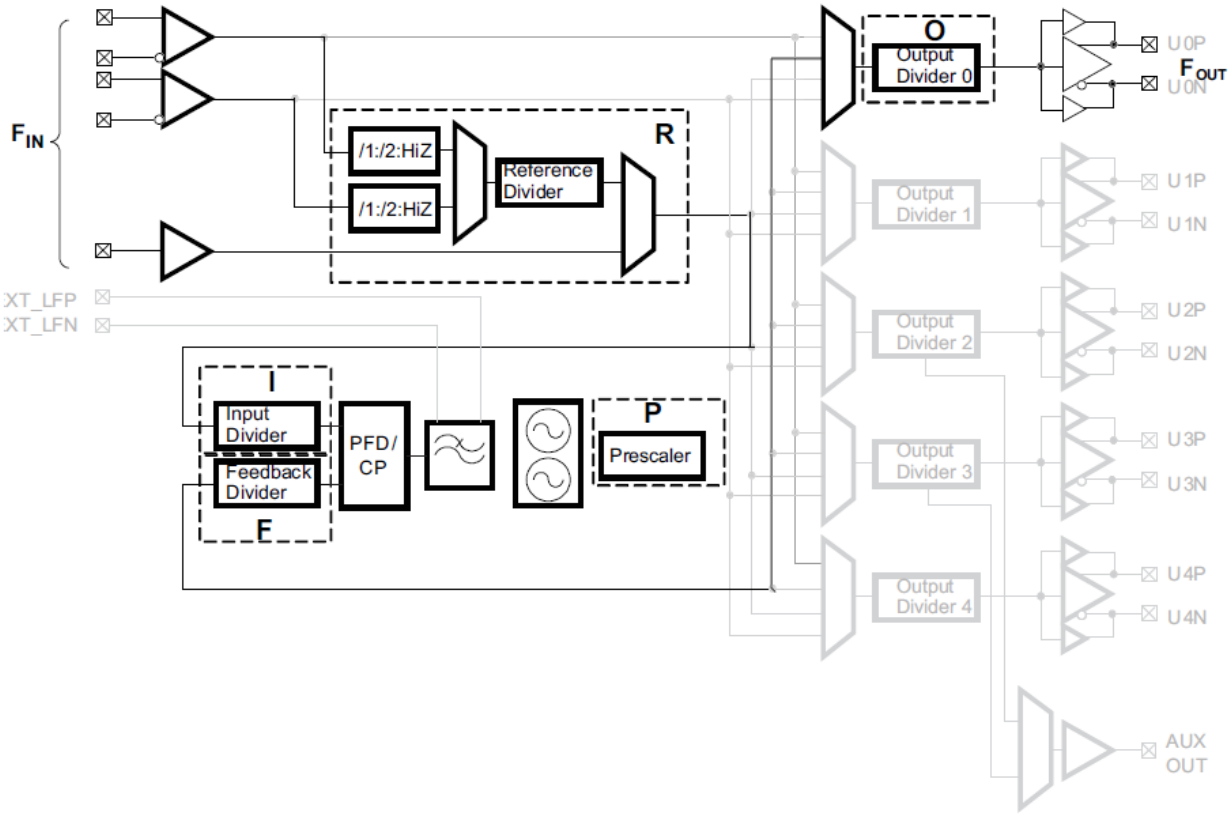


Figure 3-9: CDCE62005 Clock Path – Synthesizer Mode.

Output block

The output block includes five individually programmable outputs (U0, U1, U2, U3, and U4) and one auxiliary output (AUX OUT).

Each of these five identical **programmable outputs** incorporates an output multiplexer, a clock divider module, and a universal output driver that can be configured to provide different output formats (LVPECL, LVDS, and LVCMOS). Each output can also be programmed to a unique output frequency (up to 1.5 GHz) and skew relationship via a programmable delay block. If all outputs are configured in single-ended mode (e.g., LVCMOS), the CDCE62005 supports up to ten outputs.

The **auxiliary output** can be internally feed by either U3 or U4. Regarding electrical levels, this output only supports LVCMOS.

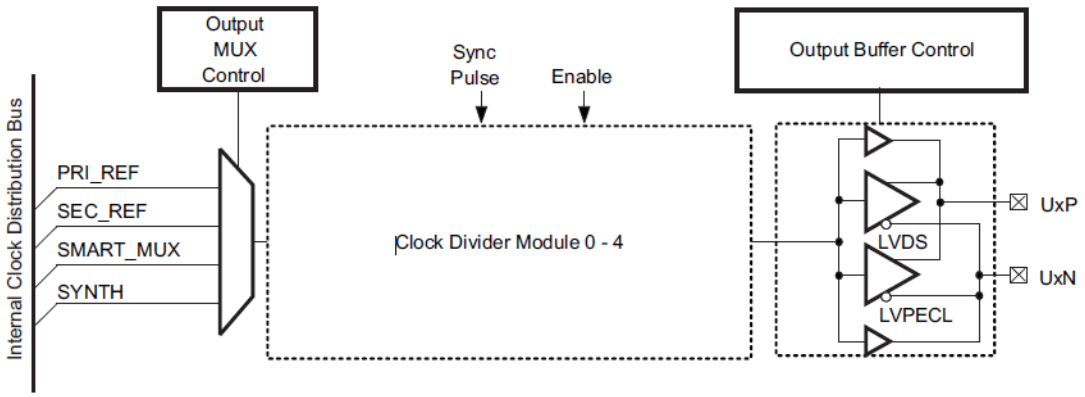


Figure 3-10: CDCE62005 output block.

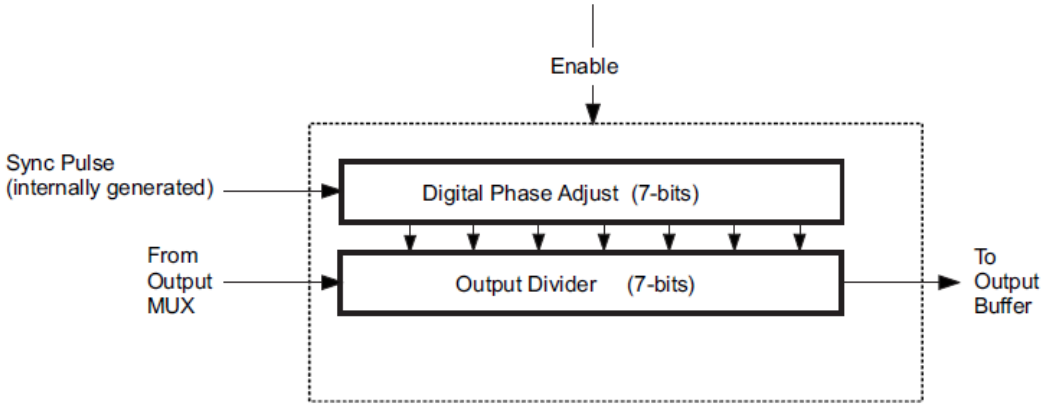


Figure 3-11: CDCE62005 output block (Clock divider detail).

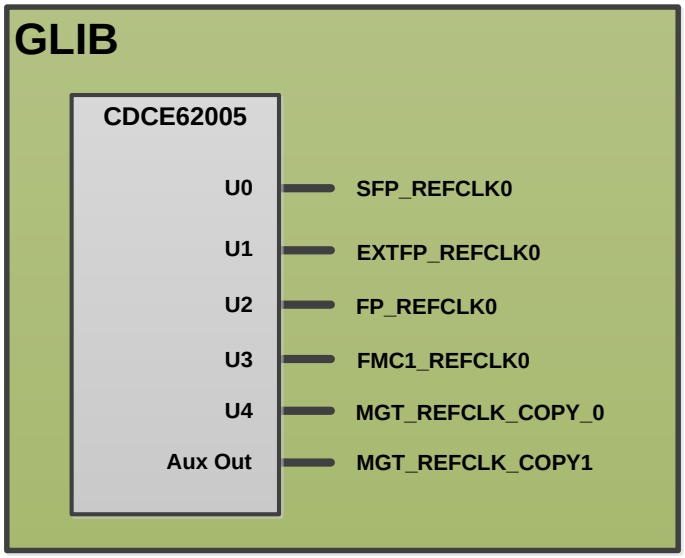


Figure 3-12: GLIB names for the Clock Synthesizer outputs.

Interface and control block

The interface and control block presents an SPI interface that allows the configuration of the CDCE62005 during system operation. This configuration is done by writing the internal registers of the interface and control block. Each output or feature can be configured independently. The CDCE62005 has a total of nine 28 bits wide registers implemented in static RAM. In addition, this configuration can be stored in an on-chip EEPROM and loaded during power up.

Note!!! The internal EEPROM can be locked with an SPI command and the process is irreversible

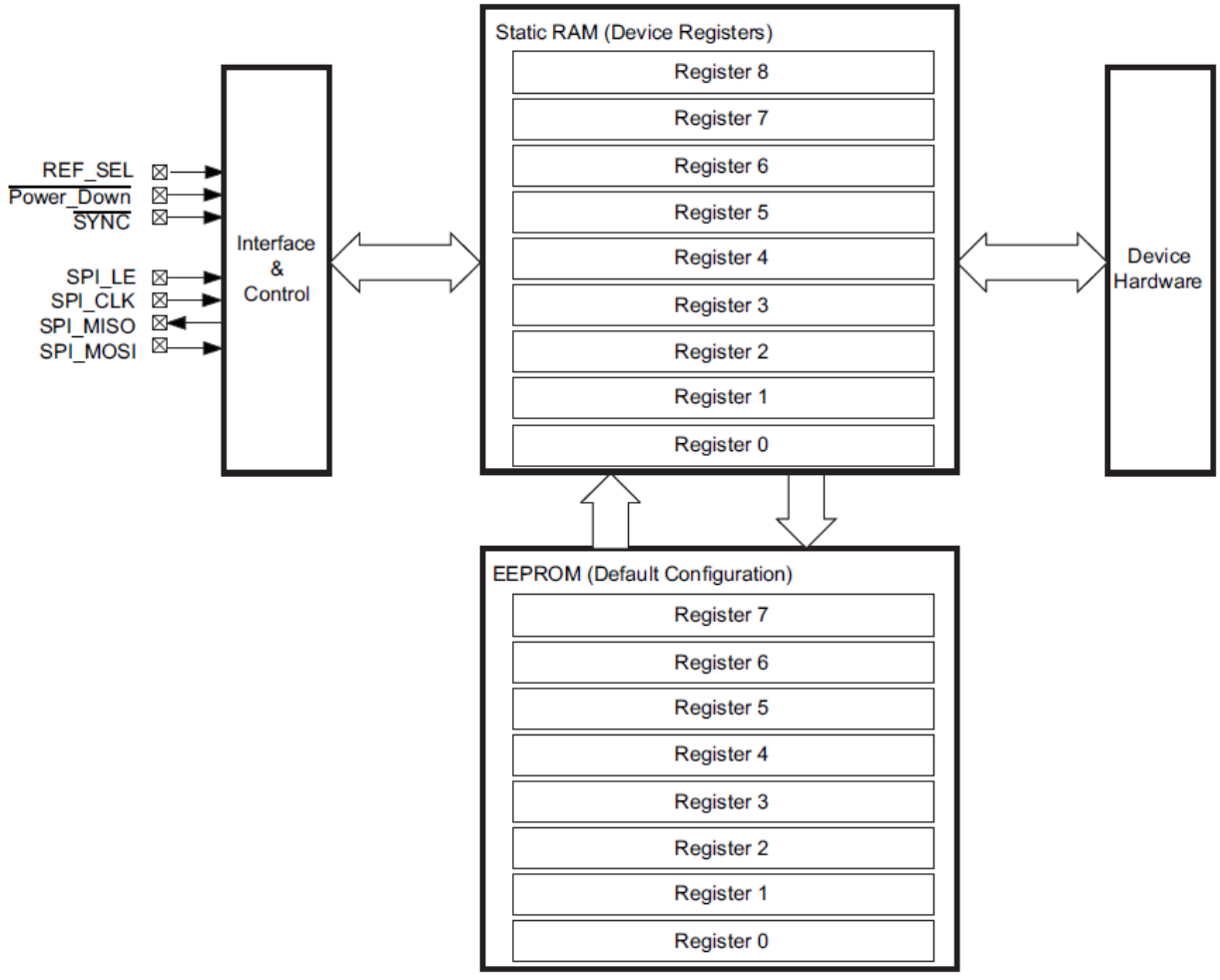


Figure 3-13:CDCE62005 interface and control block.

3.2.4. FPGA Virtex-6 (XC6VLX130T-1FF1156)

Figure 3-14: Virtex-6 LXT 130T FF1156 -1.

The Virtex-6 LXT [22] is a High-performance FPGA developed by Xilinx [23], which featured with advanced serial connectivity and many built-in system-level blocks. A summary of the Virtex-6 LXT 130T features is listed next:

- Advanced, high-performance FPGA Logic:
  - Real 6-input look-up table (LUT) technology
  - Dual LUT5 (5-input LUT) option
  - LUT/dual flip-flop pair for applications requiring rich registers mix
  - Improved routing efficiency
  - 64 bits (or two 32 bits) distributed LUT RAM option per 6-input LUT
  - SRL32/dual SRL16 with registered outputs option
- Powerful mixed-mode clock managers (MMCM):
  - MMCM blocks provide zero-delay buffering, frequency synthesis, clock-phase shifting, input jitter filtering and phase-matched clock division
- 36 Kb block RAM/FIFOs:
  - Dual-port RAM blocks
  - Programmable:
    - Dual-port widths up to 36 bits
    - Simple dual-port widths up to 72 bits
  - Enhanced programmable FIFO logic
  - Built-in optional error-correction circuitry
  - Optionally use each block as two independent 18 Kb blocks
- High-performance parallel SelectIO technology:

- 1.2 to 2.5 V I/O operation
- Source-synchronous interfacing using
- ChipSync technology
- Digitally controlled impedance (DCI) active termination
- Flexible fine-grained I/O banking
- High-speed memory interface support with integrated write-levelling capability

- Advanced DSP48E1 slices:
  - 25 x 18, two's complement multiplier/accumulator
  - Optional pipelining
  - New optional pre-adder to assist filtering applications
  - Optional bitwise logic functionality
  - Dedicated cascade connections
- Flexible configuration options:
  - SPI and Parallel Flash interface
  - Multi-bitstream support with dedicated fallback reconfiguration logic
  - Automatic bus width detection
- System Monitor capability:
  - On-chip/off-chip thermal and supply voltage monitoring
  - JTAG access to all monitored quantities
- Integrated interface blocks for PCI Express® designs:
  - Compliant to the PCI Express Base Specification 2.0
  - Gen1 (2.5 Gbps) and Gen2 (5 Gbps) support with GTX transceivers
  - Endpoint and Root Port capable
  - x1, x2, x4, or x8 lane support per block
- GTX transceivers: up to 6.6 Gbps :
  - Data rates below 480 Mbps supported by oversampling in FPGA logic
- Integrated 10/100/1000 Mbps Ethernet MAC block:
  - Supports 1000BASE-X PCS/PMA and SGMII using GTX transceivers

- Supports MII, GMII, and RGMII using SelectIO technology resources
- 2500 Mbps support available

- 40 nm copper CMOS process technology
- 1.0 V core voltage (-1, -2, -3 speed grades only)
- High signal-integrity flip-chip packaging available in standard or Pb-free package options

Table 3-2: Virtex-6 VLT 130T feature summary.

XC6VLX130T		
	Logic Cells:	128,000
Configurable Logic Blocks (CLBs)	**Slices(1):**	20,000
	Max Distributed RAM (Kb):	1,740
	DSP48E1 Slices(2):	480
Block RAM Blocks	**18 Kb(3):**	528
	36 Kb:	264
	Max (Kb):	9,504
	MMCMs(4):	10
	Interface Blocks for PCI Express:	2
	Ethernet MACs(5):	4
Maximum Transceivers	**GTX:**	20
	GTH:	0
	Total I/O Banks(6):	15
	Max User I/O(7):	600

Notes:
(1) Each Virtex-6 FPGA slice contains four LUTs and eight flip-flops, only some slices can use their LUTs as distributed RAM or SRLs.
(2) Each DSP48E1 slice contains a 25 x 18 multiplier, an adder, and an accumulator.
(3) Block RAMs are fundamentally 36 Kb in size. Each block can also be used as two independent 18 Kb blocks.
(4) Each CMT contains two mixed-mode clock managers (MMCM).
(5)This table lists individual Ethernet MACs per device.
(6) Does not include configuration Bank 0.
(7)This number does not include GTX or GTH transceivers.

3.2.5. GTX transceiver

General description

The GTX transceiver [24] of the Virtex-6 is almost identical with the one of Virtex-5. It is a low power and highly configurable multi-gigabit serial transceiver with line rates from 600 Mbps to 6.6 Gbps (5Gbps for speed grade -1). Data is transmitted and received by *Current Mode Logic* (CML) serial drivers/buffers with configurable termination, voltage swing etc. Programmable pre-emphasis/post-emphasis is used by the GTX TX for data transmission. On the other hand, the GTX RX incorporates linear and *digital feedback equalization* (DFE) for optimized signal integrity. Regarding codification and protocols, the GTX transceiver features: encoding, gearbox, comma alignment, channel bonding, clock correction, beacon signalling for PCI Express, Out-of-Band signalling (including COM signal support) for SATA etc. It is important to mention that all these features can be disabled for minimizing data path latency and for fixed latency modes. Architecturally, GTX transceivers are implemented within the Virtex-6 as hard blocks so they have a fixed position within the FPGA. GTX transceivers are grouped by Quads. Each Quad contains a grouping of four GTXE1 primitives. When implementing GTX transceiver, it is highly recommended the utilisation of the CORE Generator tool. This tool includes a wizard for automatically configure GTX transceivers for either standard or custom protocols. Once the GTX has been coarse configured, the user can perform a fine configuration modifying the GTX module manually.

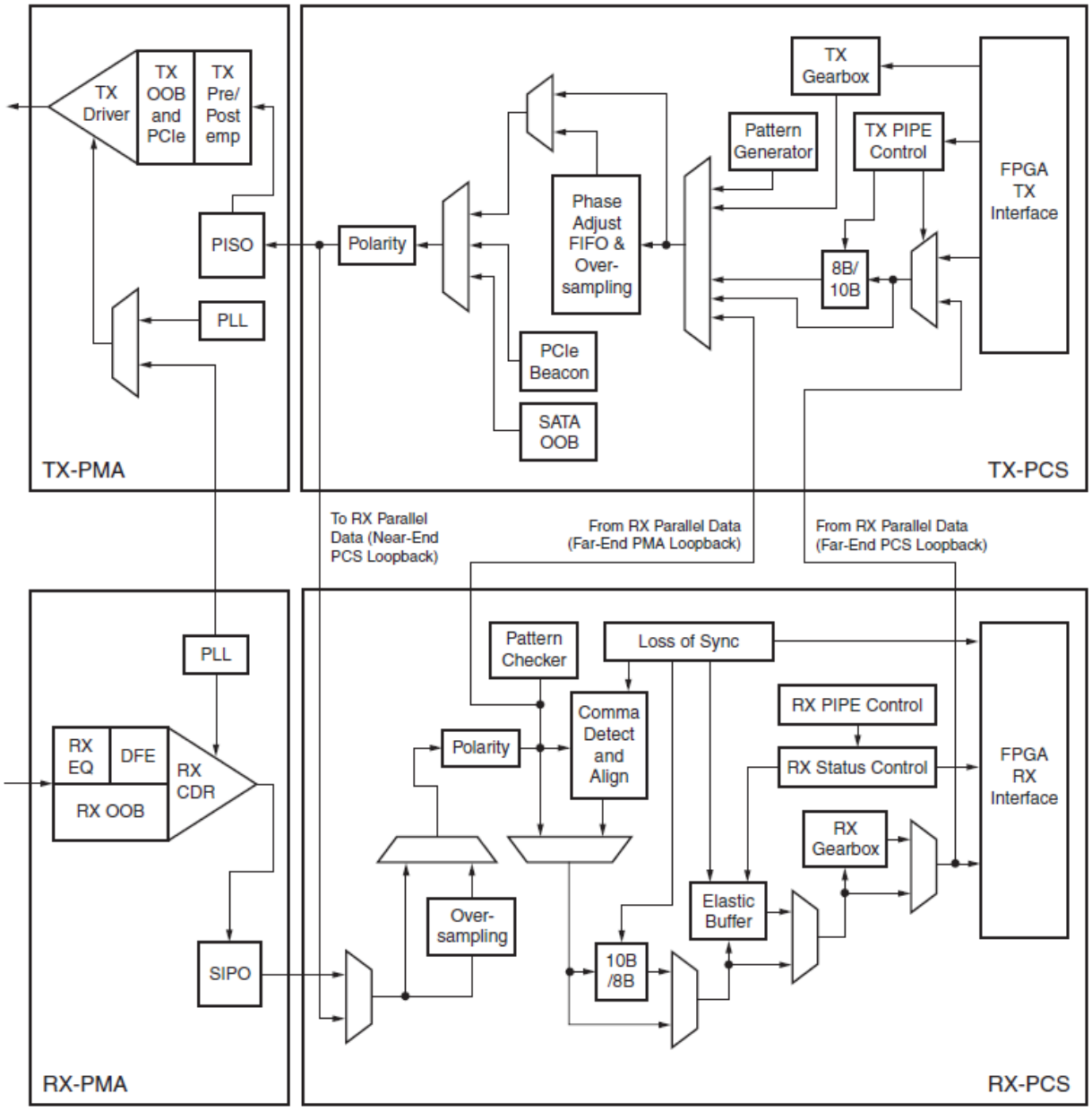


Figure 3-15: GTX transceiver simplified block diagram.

Shared features

Reference clock

Regarding clocking resources, a Quad includes two dedicated reference clock pin pair inputs (MGTREFCLK0 [P/N] and MGTREFCLK1 [P/N]) that can be connected to external clock sources. It also has dedicated reference clock routing, including multiplexing resources, which connects the Quad with its northern and southern Quad (except boundary Quads that are only connected with one Quad). For that reason, each GTX transceiver in a Quad has six clock inputs available:

- Two local reference clock pin pairs, MGTREFCLK [0/1]
- Two reference clock pin pairs from the Quads above, SOUTHREFCLK [0/1]
- Two reference clocks pin pairs or below, NORTHREFCLK [0/1]

Under any circumstance, the maximum number of Quads sourced by an external clock pin pair (MGTREFCLK N/MGTREFCLK P) must not exceed 3 Quads (or 12 GTX transceivers).

Every time that a new external clock pin pair is used, an IBUFDS_GTXE1 primitive must be instantiated. The differential input of the IBUFDS_GTXE1 primitive is connected to the MGTREFCLK [0/1] pins, while the output has to be connected to the GTXE1 primitive by the user (this output could be also used for feeding clock buffers of the logic fabric).

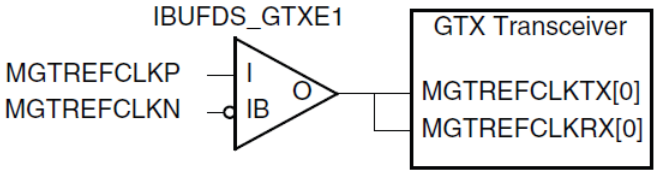


Figure 3-16: GTX single external reference clock.

Below these lines, figures with different reference clocking schemes are shown as example:

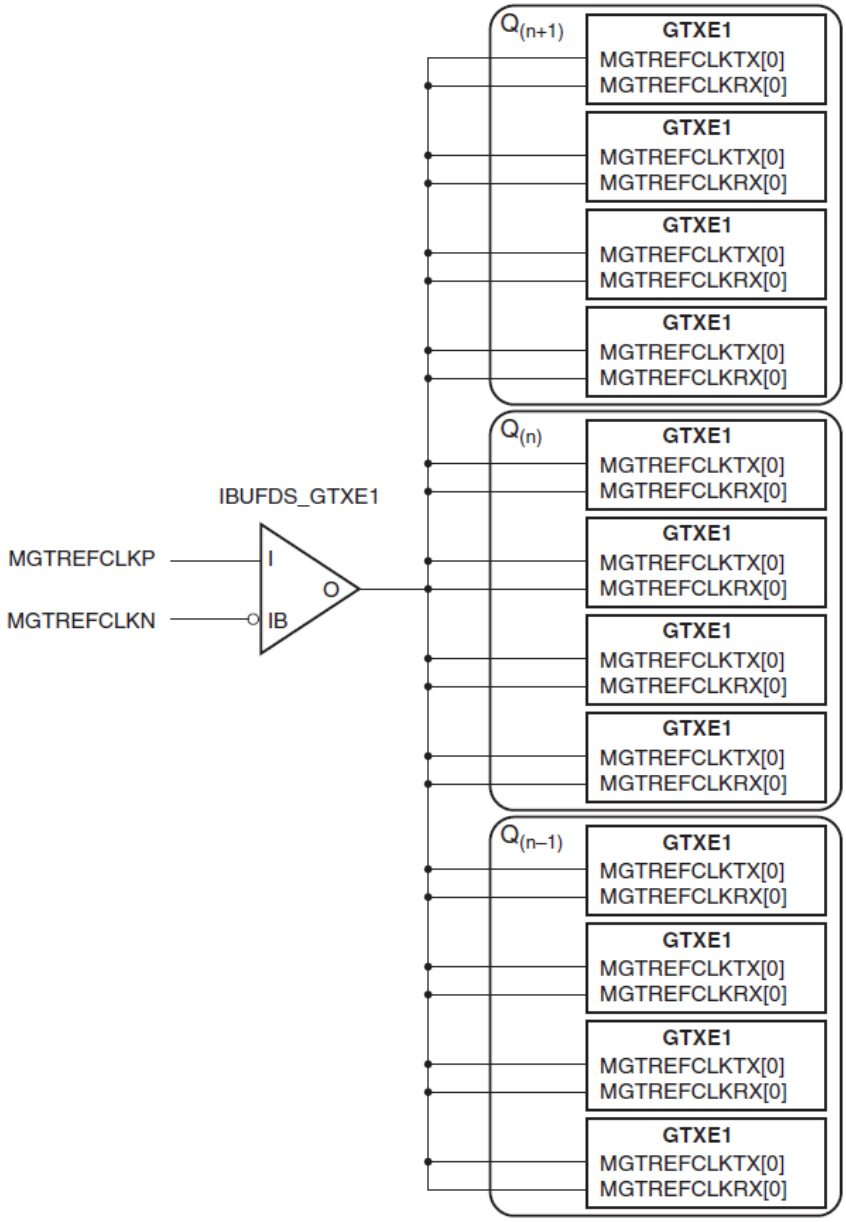


Figure 3-17: Multiple GTX transceivers with shared reference clock.

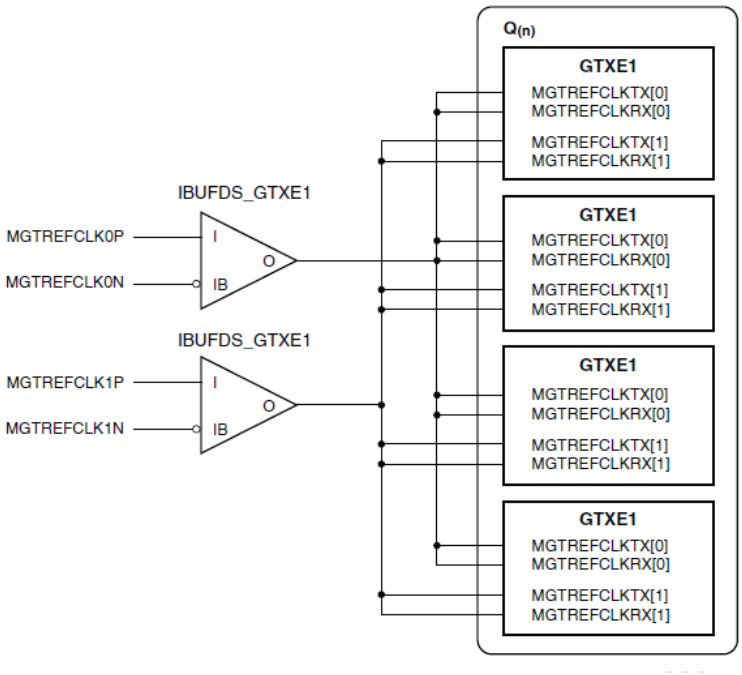


Figure 3-18: Multiple GTX transceivers with multiple reference clocks.

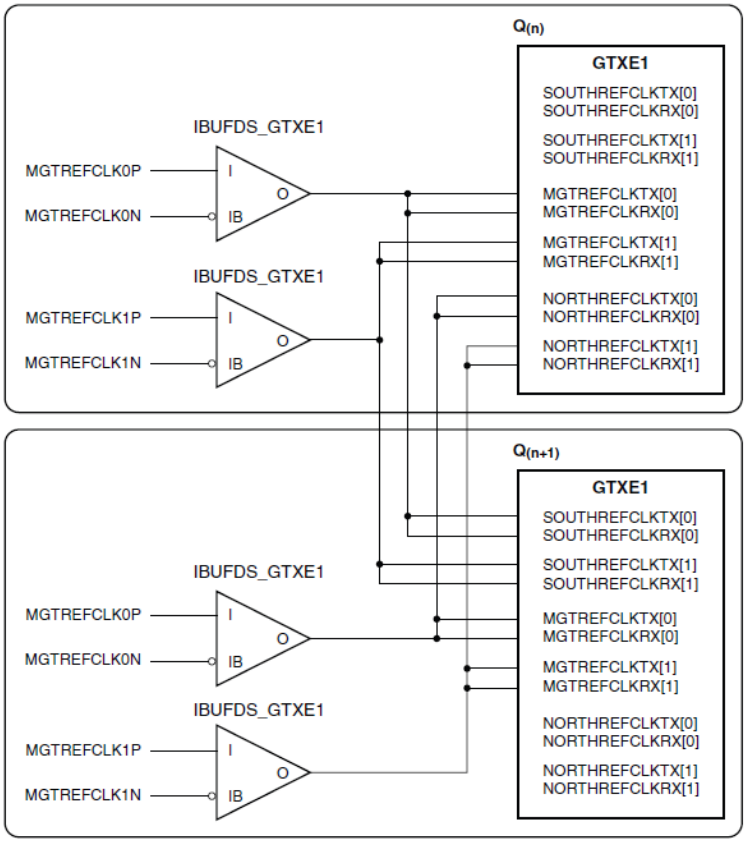


Figure 3-19: Multiple GTX transceivers with multiple reference clocks in different Quads.

PLL

GTX transceivers have different *Phase Locked Loop* (PLL) for TX and RX. They also have different reference clock inputs for these PLLs. This feature allows the user to work with the GTX TX and the GTX RX parts of the GTX transceiver asynchronously, as if they were completely independent devices (fundamental when emulating the Front-end system). One interesting feature of the GTX transceiver is that it is possible to perform static or dynamic selection of the reference clock for the GTX TX PLL and GTX RX PLL (It is needed when working with multi-rate designs that require the reference clock source to be changed on the fly).

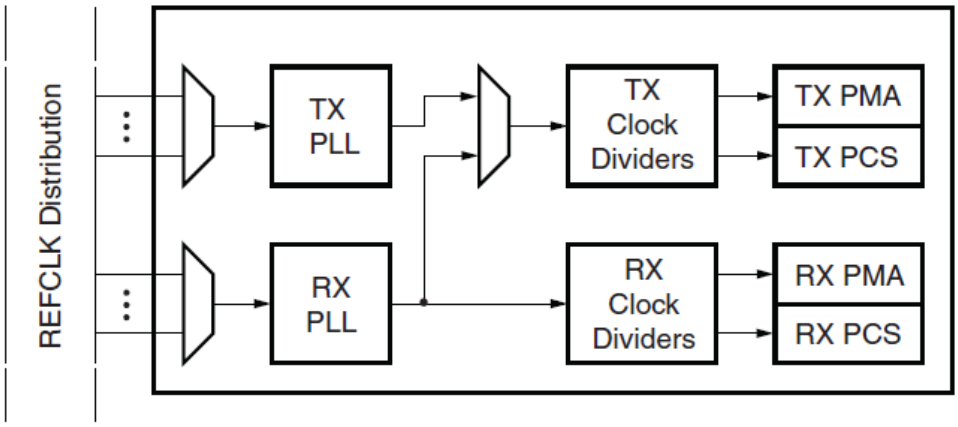


Figure 3-20: GTX PLL architecture.

Both, TX and RX PLLs have the same characteristics. Their nominal operation range is between 1.2 to 2.7 GHz for -1 speed grade devices and 1.2 to 3.3 GHz for -2 and -3 speed -grade devices. These PLLs also have an output divider that can divide the output frequency by one, two, or four. This output divider has to be configured accordingly the lane rate of the application.

Table 3-3: GTX supported line rates per divider setting.

PLL Output Divider	-1 Line rate range (Gbps)	-2/-3 Line Rate Range (Gbps)
1	2.4 to 5.0	2.4 to 6.6
2	1.2 to 2.7	1.2 to 3.3
4	0.6 to 1.35	0.6 to 1.65

Loopback

Loopback modes are specialized configurations of the transceiver data path where the traffic stream is folded back to the source. There are two different categories for loopback test modes:

- **Near-end:** These modes loop transmit data back in the transceiver closest to the traffic generator.
- **Far-end:** These modes loop received data back in the transceiver at the far end of the link.

Loopback testing can be used either during development or in deployed equipment for fault isolation. Typically, a specific traffic pattern is transmitted and then compared to check for errors. The traffic patterns used can be either application traffic patterns or specialized pseudo-random bit sequences. Each GTX transceiver has a built-in PRBS generator and checker.

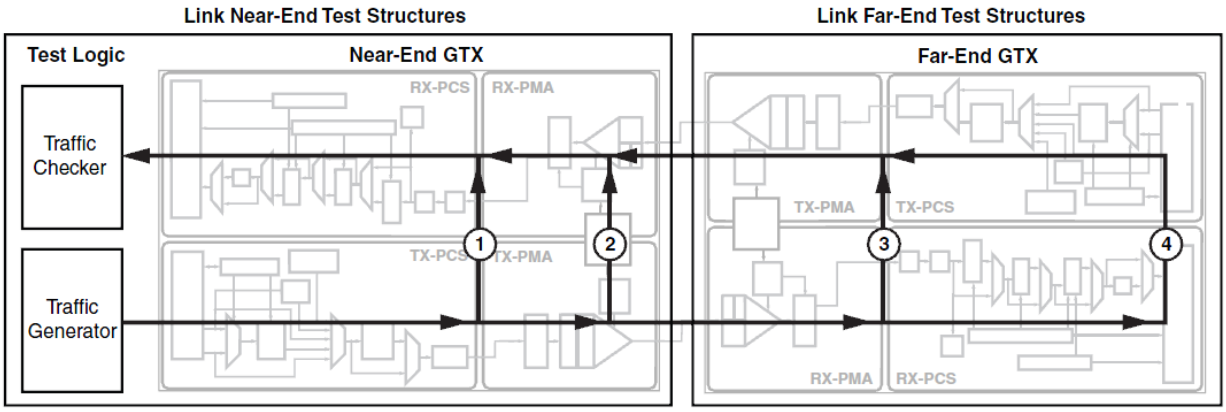


Figure 3-21: GTX loopback modes.

The Figure 3-13 shows the different loopback modes of the GTX transceivers:

- Near-End PCS Loopback (path 1)
- Near-End PMA Loopback (path 2)
- Far-End PMA Loopback (path 3)
- Far-End PCS Loopback (path 4)

Power down

The GTX transceiver supports a range of power-down modes. These modes support both generic power management capabilities as well as those defined in the PCI Express and SATA standards. Other interesting feature is that GTX TX and GTX RX can be powered down separately.

ACJTAG

The GTX transceiver supports ACJTAG, the standard specified by IEEE 1149.6 for testing digital networks.

Dynamic Reconfiguration Port (DRP)

The *dynamic reconfiguration port* (DRP) is a processor friendly synchronous interface that allows the dynamic change of parameters of the GTXE1 primitive.

Transmitter

Each GTX transceiver includes an independent transmitter, which consists of a PCS and a PMA. Parallel data flows from the FPGA into the FPGA TX interface, through the PCS and PMA, and then out the TX driver as high-speed serial data.

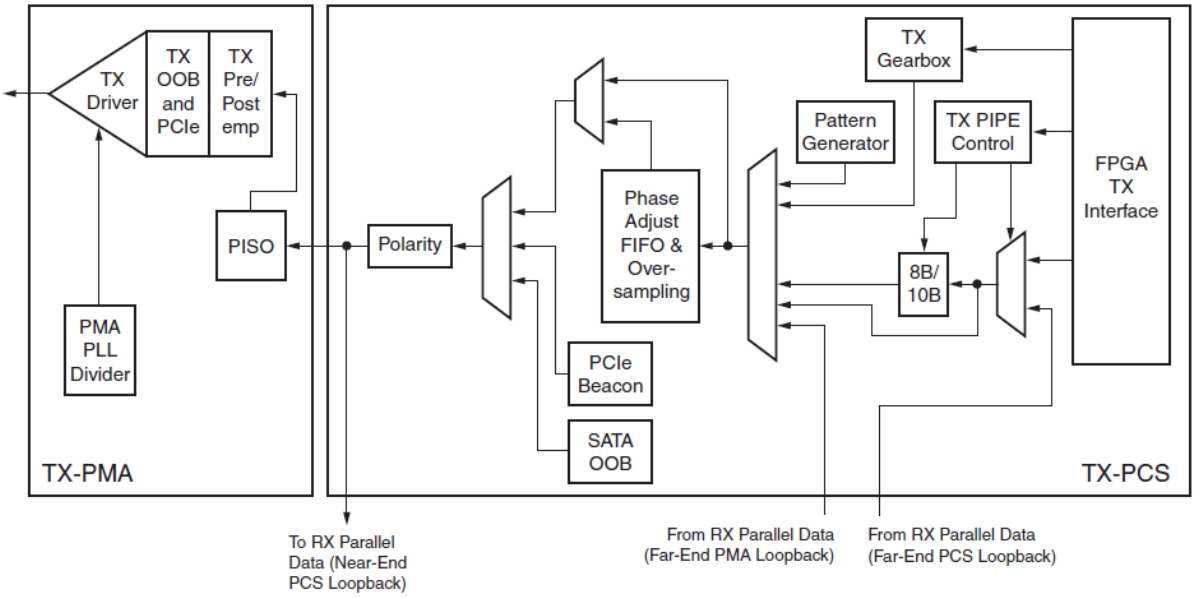


Figure 3-22: GTX TX block diagram.

Under this line are listed the key elements of the GTX TX:

- FPGA TX Interface
- TX Initialization
- TX 8B/10B Encoder
- TX Gearbox
- TX Pattern Generator
- TX Buffer
- TX Buffer Bypass
- TX Oversampling
- TX Polarity Control
- TX Fabric Clock Output Control
- TX Configurable Driver
- TX Receiver Detect Support for PCI Express Designs
- TX Out-of-Band Signalling

FPGA TX Interface

The FPGA TX interface is the FPGA's gateway to the TX data path of the GTX transceiver. Applications transmit data through the GTX transceiver by writing data to the TXDATA port on the positive edge of TXUSRCLK2. The width of the port can be configured to be one, two, or four bytes wide. In some operating modes, a second parallel clock (TXUSRCLK) must be provided for the internal PCS logic in the transmitter.

TX 8B/10B encoder

Many protocols use 8B/10B encoding on outgoing data. 8B/10B is an industry-standard encoding scheme that trades two bits of overhead per byte for improved performance (this encoding is used at CERN on the readout channel). The GTX transceiver includes an 8B/10B encoder to encode TX data without consuming FPGA resources. If encoding is not needed, the block can be disabled to minimize latency.

TX gearbox

Some high-speed data rate protocols use 64B/66B encoding to reduce the overhead of 8B/10B encoding while retaining the benefits of an encoding scheme. The TX gearbox provides support for 64B/66B and 64B/67B header and payload combining. Scrambling of the data is done in the FPGA logic.

TX pattern generator

PRBS are commonly used to test the signal integrity of high-speed links. These sequences appear random but have specific properties that can be used to measure the quality of a link. The GTX transceiver pattern generator block can generate several industry-standard PRBS patterns (PRBS-7, PRBS-15, PRBS-23 and PRBS-31).

TX elastic buffer

The GTX TX data path has two internal parallel clock domains: the PMA parallel clock domain (XCLK) and the TXUSRCLK domain. To transmit data, the XCLK rate must match the TXUSRCLK rate, and all phase differences between the two domains must be resolved. The GTX transmitter includes a TX elastic buffer to resolve phase differences between the PMACLK and TXUSRCLK domains.

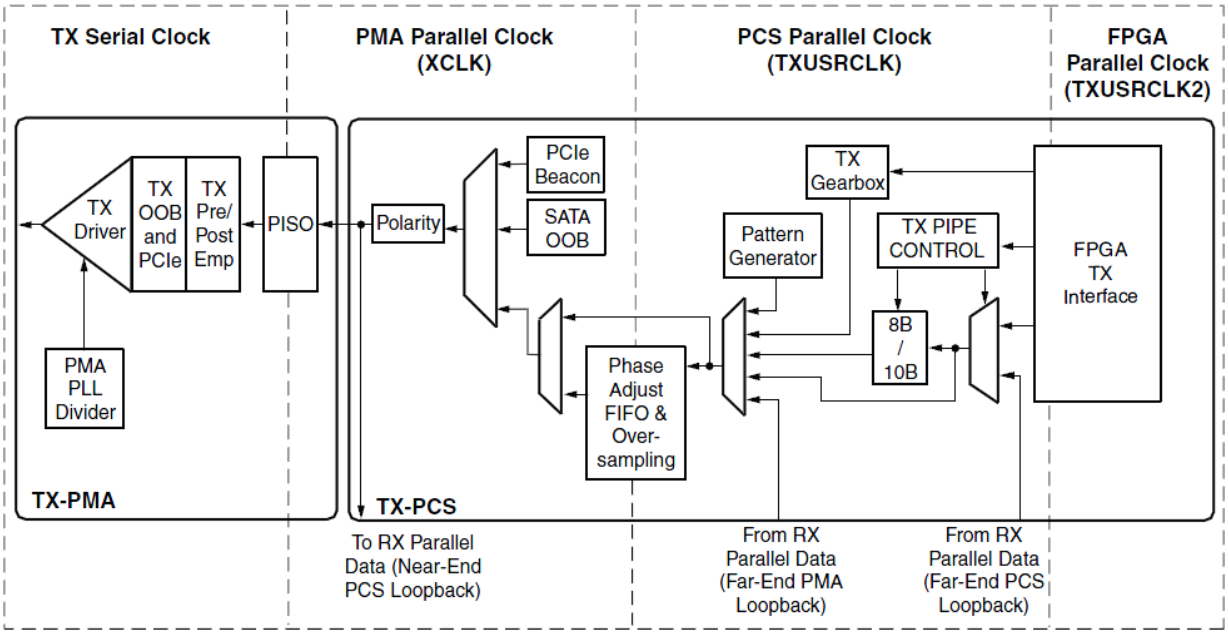


Figure 3-23: GTX TX clock domains.

TX buffer bypass

Bypassing the TX buffer is an advanced feature of the GTX transceiver. **This buffer bypassing is needed for low and fixed latency designs.** In addition to the TX Buffer, the GTX transceiver has a TX phase-alignment circuit. The TX phase-alignment circuit is used to adjust the phase difference between the PMA parallel clock domain (XCLK) and the TXUSRCLK domain when the TX buffer is bypassed. However, phase alignment requires extra logic and additional constraints on clock sources, which make the system error prone if the design is not done correctly.

TX oversampling

Each GTX transceiver includes built-in 5x oversampling to enable serial rates below 600 Mbps. The digital oversampling circuit takes parallel data from the user interface at five times the desired line rate and replicates the data bit five times before advancing to the next bit. The internal data width is automatically set to 20 bits when TX oversampling is enabled.

TX polarity control

The GTX transceiver includes a TX polarity control function to invert outgoing data from the PCS before serialization and transmission. This function is useful in designs where the RX P and RX N signals can be accidentally connected in reverse, or when there is no other possibility of routing the lanes.

TX fabric clock output control

The TX Fabric Clock Output Control block of the GTX TX has two main components:

- **Serial Clock Divider**: Each transmitter PMA module has a D divider that divides down the clock from the PLL for lower line rate support. This divider can be set statically for applications with a fixed line rate or it can be changed dynamically for protocols with multiple line rates.
- **Parallel Clock Divider and Selector**: The parallel clock outputs from the TX Fabric Clock Output Control block can be used as a fabric logic clock. However, the recommended clock for the fabric is the TXOUTCLK from one of the GTX transceivers. It is also possible to bring the MGTREFCLK directly to the fabric and use as the fabric clock. TXOUTCLK is preferred for general applications as it has an output delay control used for applications that bypass the TX buffer for output lane deskewing or constant data path delay.

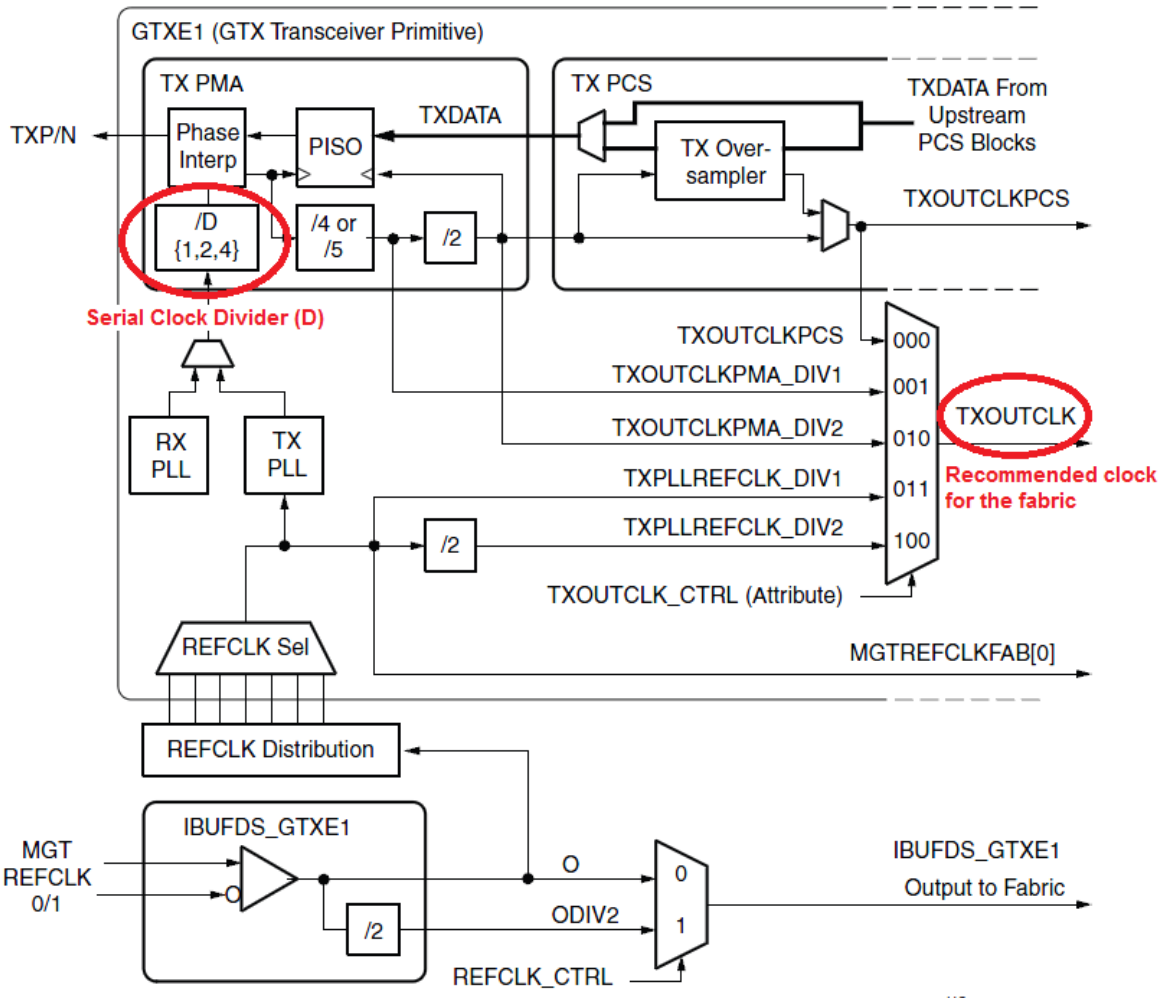


Figure 3-24: GTX TX Fabric Clock Output block detail.

TX configurable driver

The GTX TX driver is a high-speed current-mode differential output buffer. To maximize signal integrity, the driver includes these features:

- Differential voltage control
- Pre-cursor and post-cursor transmit pre-emphasis
- Calibrated termination resistors

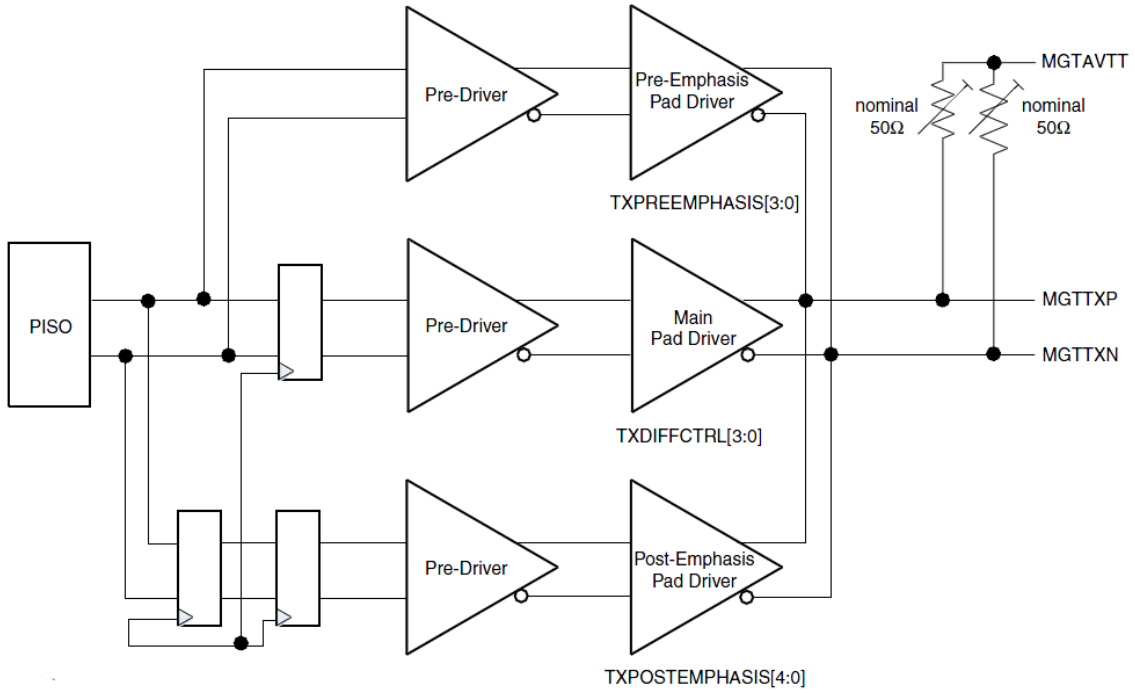


Figure 3-25: GTX TX driver block diagram.

TX receiver detect support for PCI Express designs

For PCI Express specification compliance, the GTX transceiver includes a feature that allows the transmitter on a given link to detect if a receiver is present.

TX Out-of-Band Signalling

Each GTX transceiver provides support for generating the Out-of-Band (OOB) sequences described in the *Serial ATA* (SATA), *Serial Attach SCSI* (SAS) specification, and beaconing described in the PCI Express specification. Each GTX transceiver also supports SATA and SAS auto-negotiation.

Receiver

Each GTX transceiver includes an independent receiver, made up of a PCS and a PMA. High-speed serial data flows from traces on the board into the PMA of the RX, then goes through the PCS, and finally into the FPGA logic.

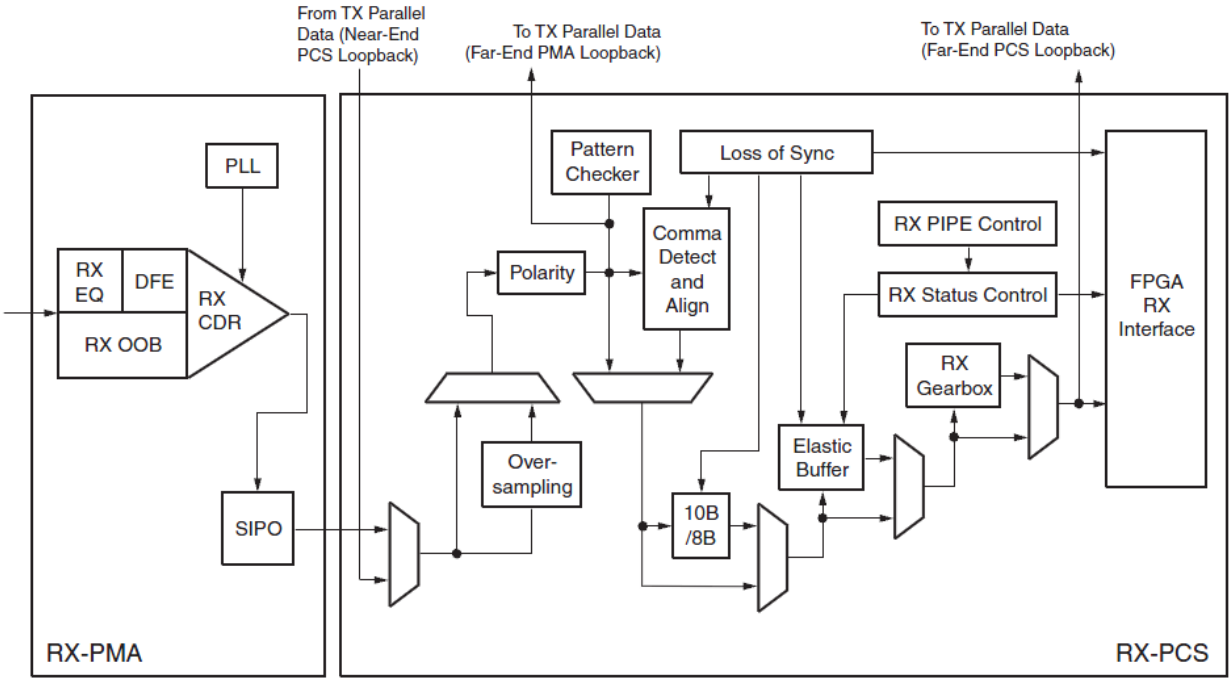


Figure 3-26: GTX RX Block Diagram.

Under this line are listed the key elements of the GTX TX:

- RX Analog Front End
- RX Out-of-Band Signalling
- RX Equalizer
- RX CDR
- RX Oversampling
- RX Fabric Clock Output Control
- RX Margin Analysis
- RX Polarity Control
- RX Pattern Checker
- RX Byte and Word Alignment
- RX Loss-of-Sync State Machine
- RX 8B/10B Decoder
- RX Buffer Bypass
- RX Elastic Buffer
- RX Clock Correction
- RX Channel Bonding
- RX Gearbox
- RX Initialization

RX analog front end

The RX *analog front end* (AFE) is a high-speed current-mode input differential buffer. The AFE includes the following features:

- Configurable RX termination voltage
- Bypassable on-chip coupling capacitors
- Calibrated termination resistors

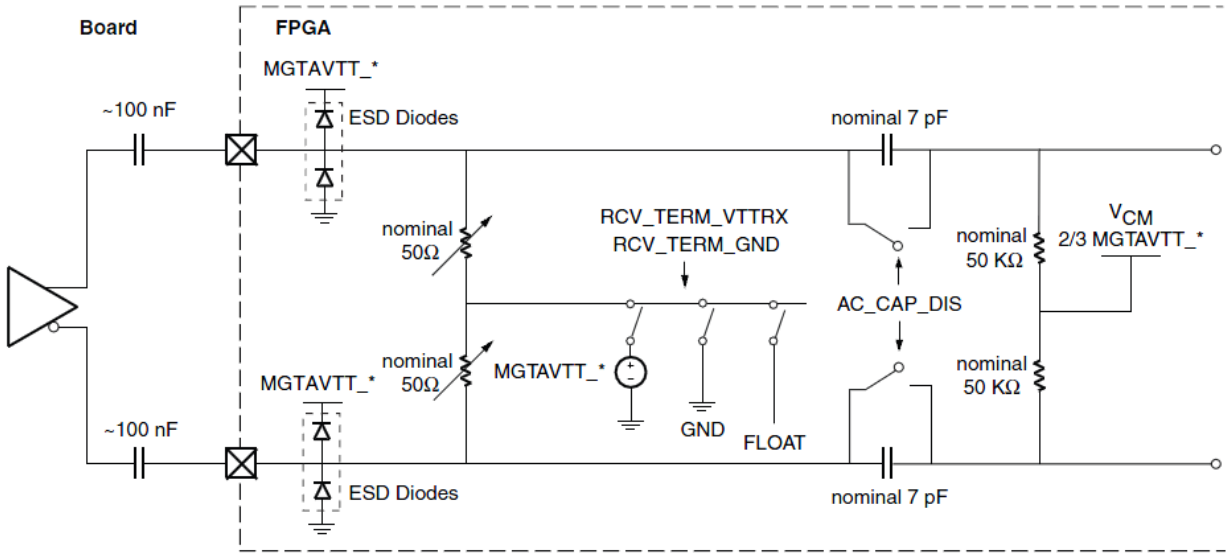


Figure 3-27: GTX RX Analog front end block diagram.

RX Out-of-Band Signalling

The GTX receiver provides support for decoding the *Out-of-Band* (OOB) sequences described in the SATA and SAS specifications and supports beaconing described in the PCI Express specification. The GTX receiver also supports beacons that are PCI Express. The beacon sequence has to be decoded by the FPGA logic.

RX equalizer

The GTX RX has a continuous time RX equalization circuit and a DFE circuit to compensate for high-frequency losses in the channel.

- The **continuous time RX equalization circuit** can be tuned to meet the specific requirements of the physical channel used in the design by compensating for signal distortion due to high-frequency attenuation. It is a 3-stage amplifier with the ability to boost the input signal at low, intermediate, and high frequencies.

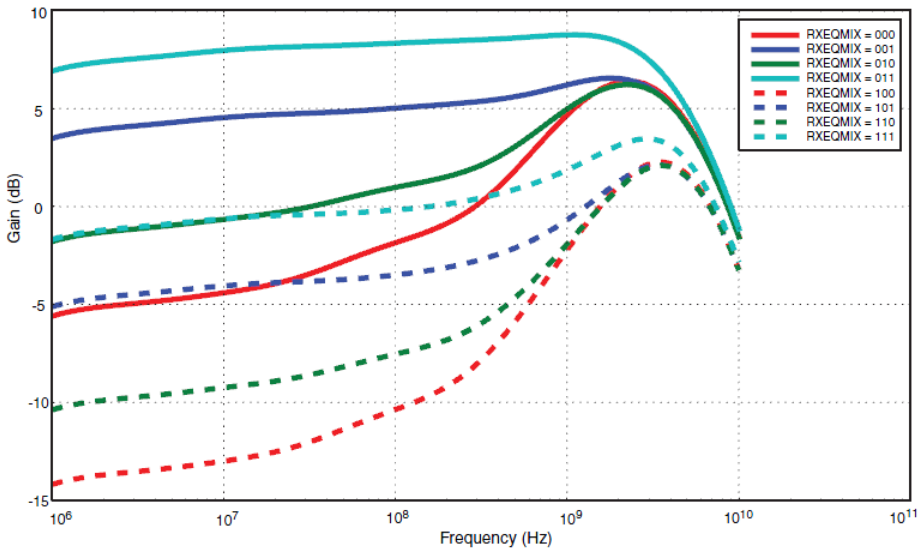


Figure 3-28: GTX RX continuous time equalization - absolute gain (voltage transfer function).

- The **Decision Feedback Equalizer (DFE)** is a discrete-time adaptive high-pass filter, with four-tap architecture. The DFE allows better compensation of transmission channel losses by providing a closer adjustment of filter parameters than when using a linear equalizer. However, a DFE cannot remove the pre-cursor of a transmitted bit. The tap values of the DFE can be either set manually or by the auto-calibration logic.

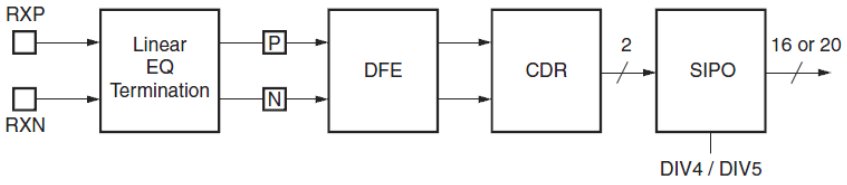


Figure 3-29: GTX RX Decision Feedback Equalizer (DEF).

RX CDR

Each GTX RX presents a CDR circuit that extracts the recovered clock and data from an incoming data stream. First of all, incoming data goes through receiver equalization stages. Once the incoming data has been equalized, it is captured by an edge and a data sampler. The phase for the edge sampler is locked to the transition region of the data stream while the phase of the data sampler is positioned in the middle of the data eye. A third sampler, the scan sampler, normally utilises the same phase as the data sampler and is used by the DFE block.

The CDR represents one of the most important issues when designing fixed latency systems. This issue will be discussed in chapter 3.3.

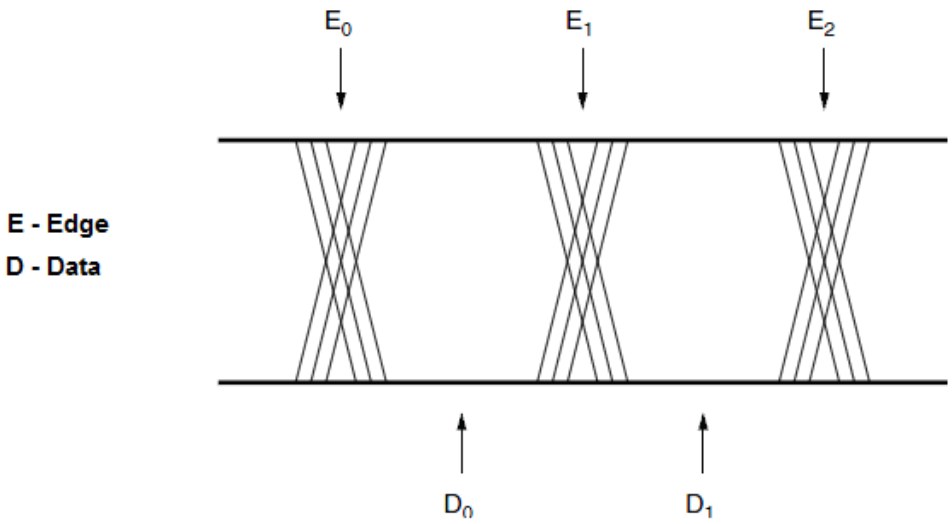


Figure 3-30: CDR sampler positions.

RX oversampling

Each GTX RX includes built-in 5X oversampling to enable serial rates below 600Mb/s. At these low rates, the CDR must operate at five times the desired line rate to stay within its operating limits. The digital oversampling circuit takes parallel data from the SIPO at five times the desired line rate and uses the position of bit value transitions to recover a clock. The transition points are also used to pick an optimal sampling point to recover 4 bits of data from each set of 20 bits presented.

RX fabric clock output control

The RX Fabric Clock Output Control block of the GTX RX has two main components:

- **Serial Clock Divider**: Each receiver PMA module has a D divider that divides down the clock from the PLL for lower line rate support. This divider can be set statically for applications with a fixed line rate or it can be changed dynamically for protocols with multiple line rates.
- **Parallel Clock Divider and Selector**: Parallel clock outputs from the RX Fabric Clock Output Control block can be used as a fabric logic clock. However, the recommended clock for the fabric is the RXRECCLK from one of the GTX transceivers. It is also possible to bring the MGTREFCLK directly to the fabric and use as the fabric clock, but then clock correction is needed. RXRECCLK is preferred for general applications because it is a recovered copy of the transmitter clock.

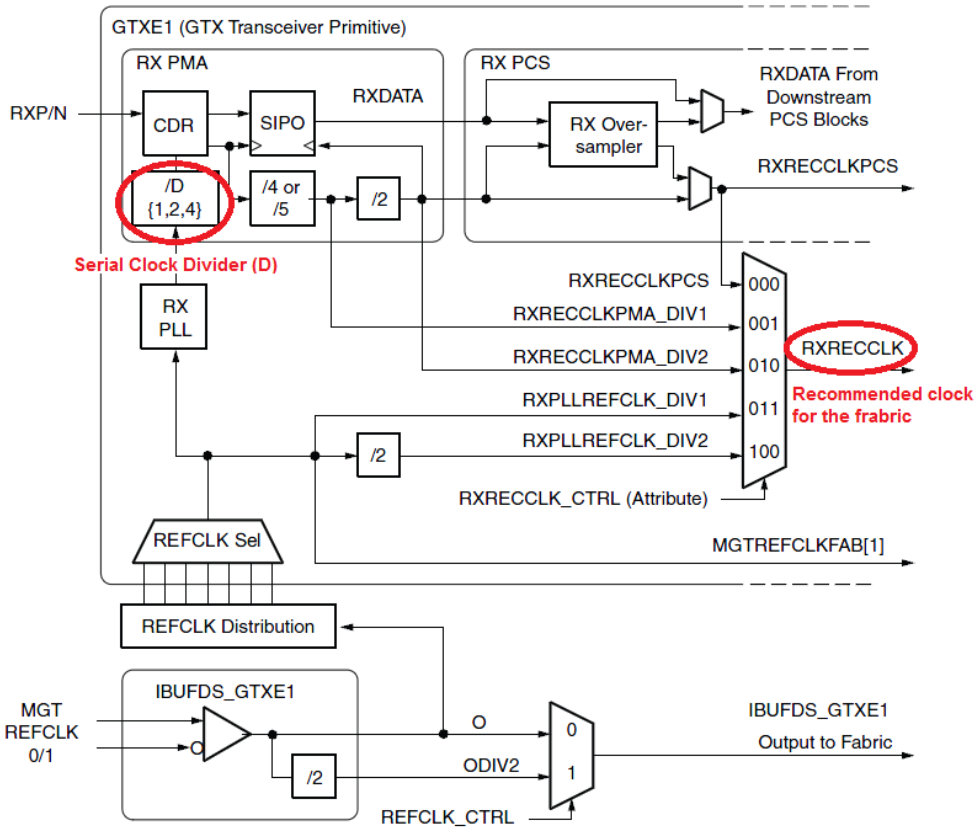


Figure 3-31: GTX RX Fabric Clock Output block detail.

RX margin analysis

As line rates and channel attenuation increase, the receiver equalizers are often enabled to overcome channel attenuation. Channels with equalizers enabled are a challenge to system debug as the quality of the link cannot be determined by measuring the far-end eye diagram. At high line rates, the received eye measured on the printed circuit board can appear to be completely closed even though the internal eye after the receiver equalizer is open. The RX CDR block of the GTX transceiver provides a diagnostic mechanism to estimate the receiver eye margin **after** the equalizer:

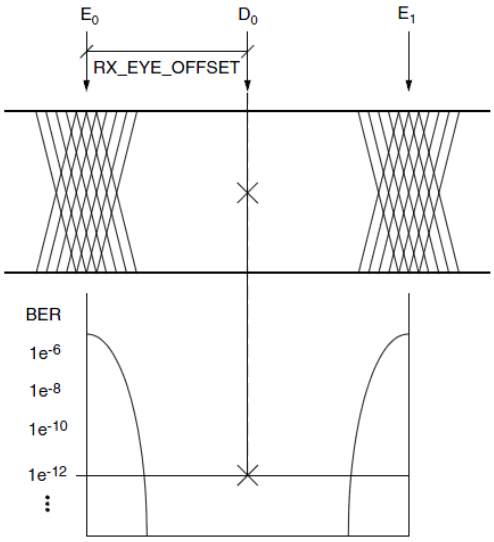


Figure 3-32: GTX RX data sampling position to Bit Error Rate.

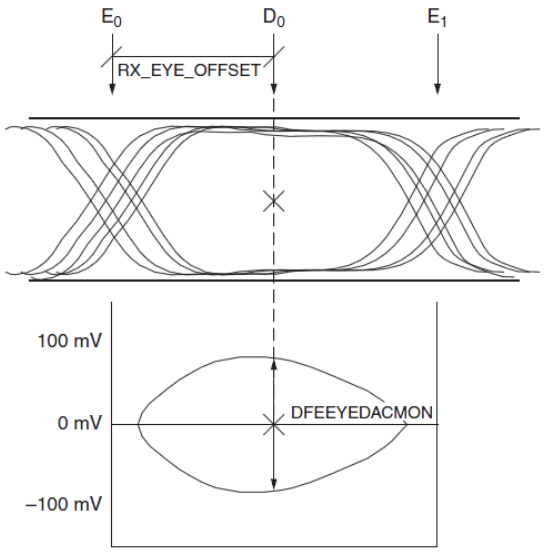


Figure 3-33: GTX RX vertical eye height.

RX polarity control

The GTX RX can invert incoming data using the RX polarity control function. This function is useful in designs where the RX P and RX N signals can be accidentally connected in reverse, or when there is no other possibility of routing the lanes.

RX pattern checker

The GTX receiver includes a built-in PRBS checker. This checker can be set to check for one of four industry-standard PRBS patterns (PRBS-7, PRBS-15, PRBS-23 and PRBS-31). The checker is self-synchronizing and works on the incoming data before comma alignment or decoding. This function can be used to test the signal integrity of the channel.

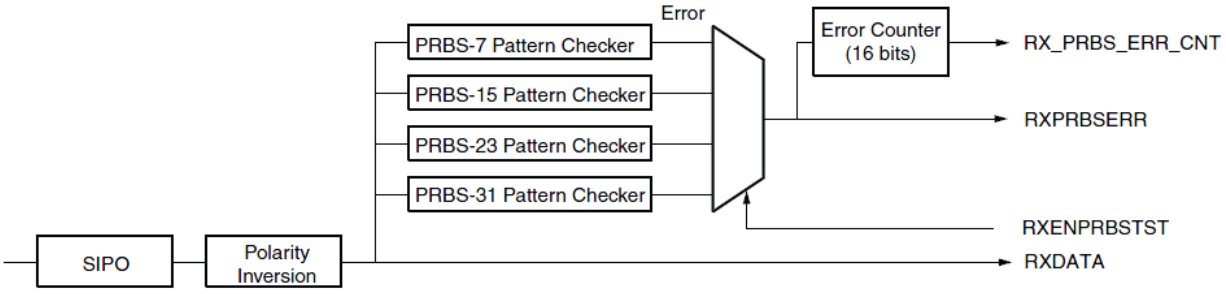


Figure 3-34: GTX RX pattern checker block.

RX Byte and Word alignment

When performing a serial communication, serial data must be aligned to symbol boundaries before it can be used as parallel data. To make alignment possible, transmitters send a recognizable sequence, usually called a comma. The receiver searches for the comma in the incoming data. When it finds a comma, it moves the comma to a byte boundary so the received parallel words match the transmitted parallel words. The receiver of the GTX transceiver includes an alignment block that can be programmed to align specific commas to various byte boundaries, or to manually align data. The block can be bypassed to reduce latency if it is not needed.

RX Loss-of-Sync State Machine

Several 8B/10B protocols make use of a standard *Loss-of-Sync* (LOS) state machine to detect when the channel is malfunctioning. The receiver of the GTX transceiver includes a LOS state machine that can be activated for protocols requiring it. When the state machine is not used, the ports of the LOS state machine can be re-used to monitor the condition of incoming data.

RX 8B/10B decoder

Many protocols require receivers to decode 8B/10B data. 8B/10B is an industry standard encoding scheme that trades two bits of overhead per byte for improved performance (this encoding is used at CERN on the readout channel). The receiver of the GTX transceiver includes an 8B/10B decoder to decode RX data without consuming FPGA resources. The decoder includes status signals to indicate errors and incoming control sequences. If decoding is not needed, the block can be disabled to minimize latency.

RX elastic buffer

The GTX RX data path has two internal parallel clock domains: the PMA parallel clock domain (XCLK) and the RXUSRCLK domain. To receive data, the XCLK rate must be sufficiently close to the RXUSRCLK rate, and all phase differences between the two domains must be resolved. The GTX receiver includes an RX elastic buffer to resolve differences between the PMACLK and RXUSRCLK domains.

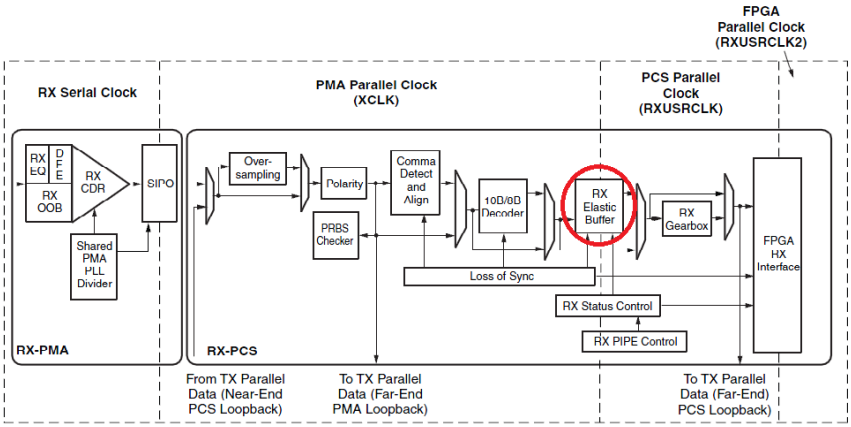


Figure 3-35: RX Clock Domain.

RX buffer bypass

Bypassing the RX buffer is an advanced feature of the GTX transceiver. **This buffer bypassing is needed for low and fixed latency designs.** In addition to the RX Buffer, the GTX transceiver has a RX phase-alignment circuit. The RX phase-alignment circuit is used to adjust the phase difference between the PMA parallel clock domain (XCLK) and the RXUSRCLK domain when the RX buffer is bypassed. Phase alignment causes RXRECCLK from the CDR to be adjusted so that there is no significant phase difference between XCLK and RXUSRCLK. For that reason, the RX elastic buffer can be only bypassed when RXRECCLK is used to source RXUSRCLK and RXUSRCLK2. Other consequences when using buffer bypass are that clock correction and channel bonding are not available. On the Xilinx documentation, it is said that when bypassing the buffer, latency through the RX data path is low and deterministic. That is not completely true regarding deterministic latency as it will be explained in chapter 4. Phase alignment requires extra logic and additional constraints on clock sources that make the system error prone if the design is not done correctly.

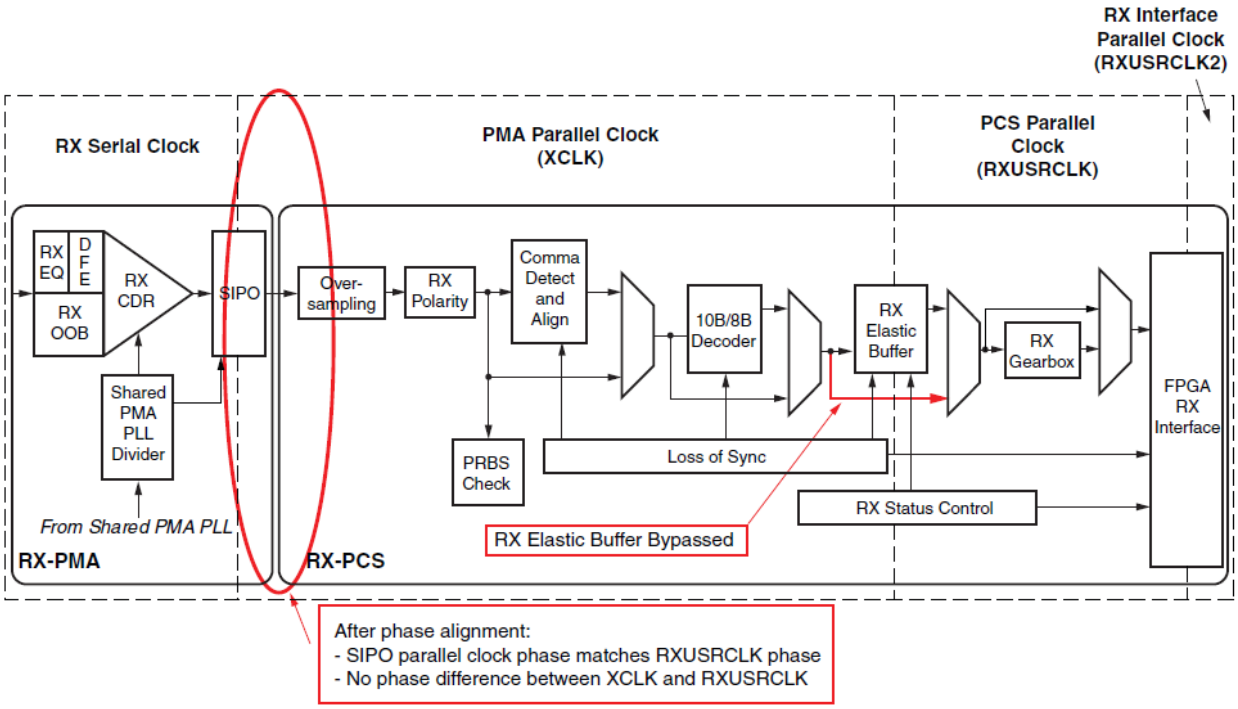


Figure 3-36: GTX RX buffer bypass.

RX clock correction

Clock correction prevents the RX elastic buffer from getting too full or too empty by deleting or replicating special idle characters in the data stream. This feature allows frequency differences between the XCLK and RXUSRCLK domains.

RX channel bonding

The RX elastic buffer can also be used for channel bonding. Channel bonding cancels out the skew between GTX transceiver lanes by using the RX elastic buffer as a variable latency block. The transmitter sends a pattern simultaneously on all lanes, which the channel bonding circuit uses to set the latency for each lane so that data is presented without skew at the FPGA RX interface.

RX gearbox

Some high-speed data rate protocols use 64B/66B encoding to reduce the overhead of 8B/10B encoding while retaining the benefits of an encoding scheme. The RX gearbox provides support for 64B/66B and 64B/67B header and payload combining. Descrambling of the data is done in the FPGA logic.

FPGA RX interface

The FPGA receives RX data from the GTX RX through the FPGA RX interface. Data is read from the RXDATA port on the positive edge of RXUSRCLK2. RXDATA can be configured to be one, two, or four bytes wide. In some operating modes, a second parallel clock (RXUSRCLK) must be provided for the internal PCS logic in the receiver.

Simulation and implementation

Simulation

HDL systems using GTX transceivers can be simulated. This possibility is very interesting for designing the logic around the GTX transceiver, but when configuring the parameters of the transceiver itself, it is highly recommended to do it using iBERT.

Implementation

The GTX transceiver wizard, which is included in CORE Generator, is the recommended tool for generating a GTX transceiver. This wizard generates a wrapper to instantiate a GTX transceiver primitive called GTXE1. The coarse configuration done by the wizard is valid for almost all projects, but when designing a project with a special setup, the GTX transceiver has to be configured either modifying manually the wrapper generated by the wizard or creating a custom .ucf file which overwrites the generics of the wrapper.

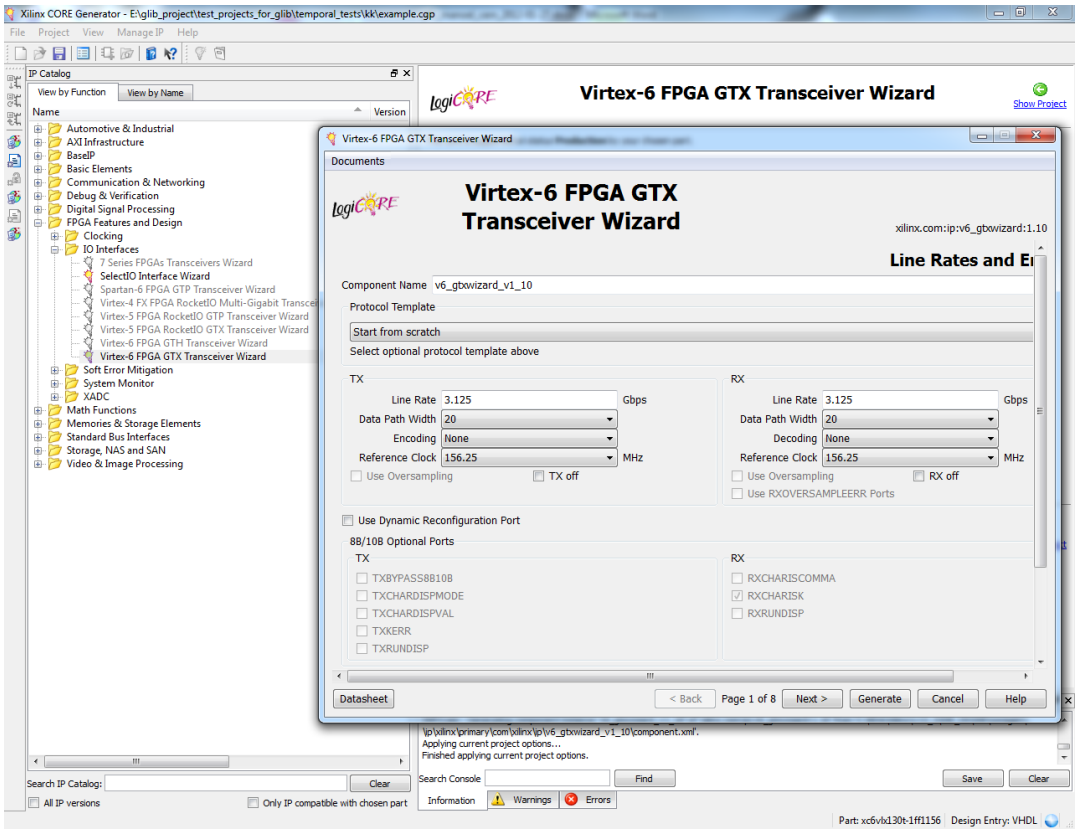


Figure 3-37: GTX transceiver wizard.

iBERT

The Integrated Bit Error Ratio Test (iBERT) is a very useful tool included in ChipScope Pro Analyser (ChipScope). This tool is generated using the iBERT wizard provided by CORE Generator. Once the bitstream has been generated, it has to be loaded into the Virtex-6 using ChipScope. With iBERT, almost all parameters of the GTX transceiver can be modified at run time (PRBS test, pre/post-emphasis etc.). When using iBERT, there is not limit for the number of GTX transceivers used at the same time (actually this limit is set by the power of the computer running iBERT and the number of the GTX transceivers available in the Virtex-6). The control of the parameters is done by using a very user friendly GUI which provides both controls and status indicators. iBERT is the recommended tool for setup and test of the GTX transceivers.

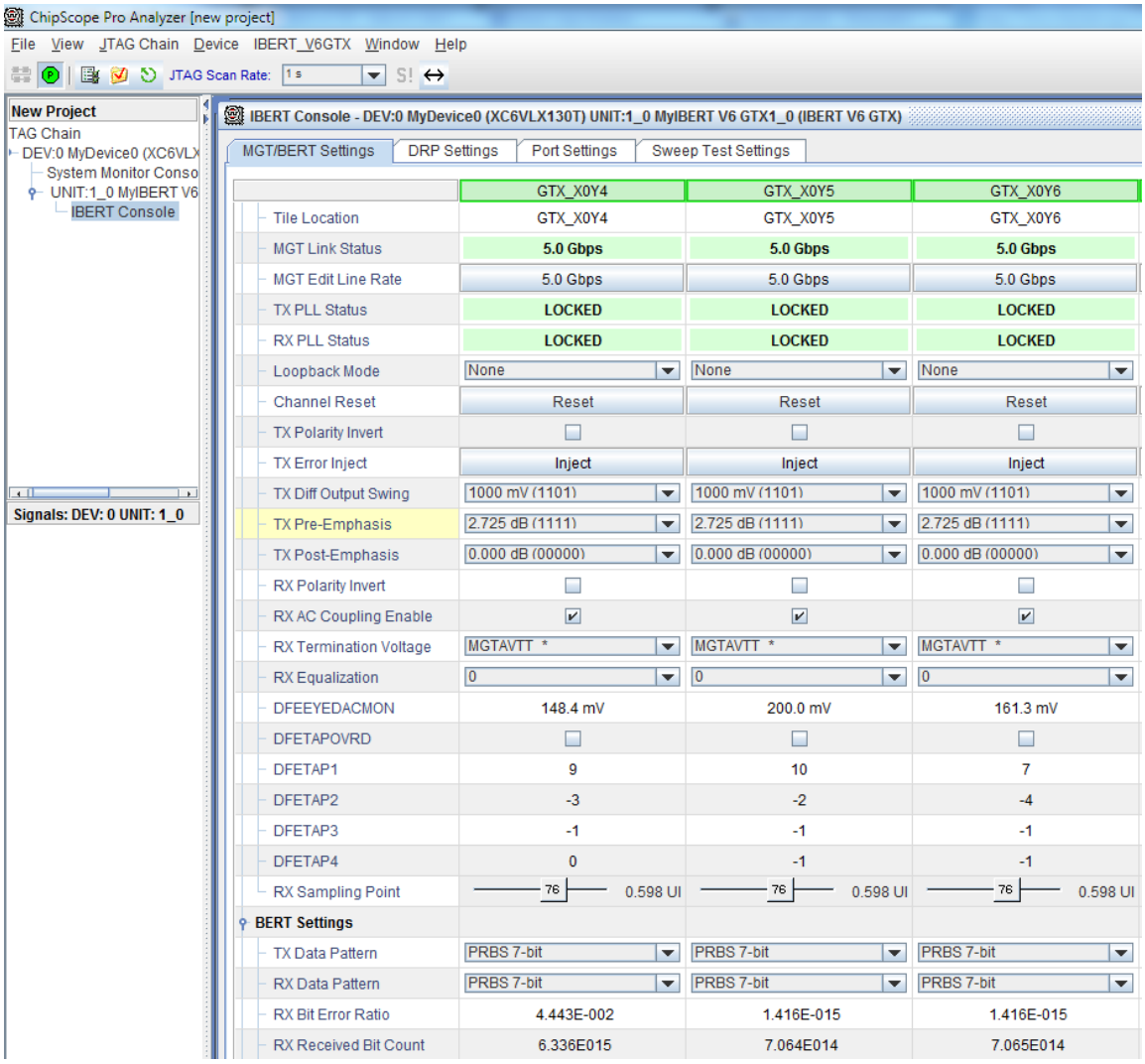


Figure 3-38: iBERT.

3.2.6. GBT HDL module

The idea of a HDL version of the GBT chipset arose because the BE side of the GBT link is planned to be implemented on FPGA based devices. Nevertheless, this HDL version of the GBT chipset would thus be very useful, not only for the counting room GBTs, but also to emulate the GBT chip before its actual release, and to design test platforms for GBT testing and system validation. A team located in Marseille (CPPM) and at CERN implemented the GBT protocol in Altera and Xilinx FPGAs (GBT-FPGA Starter Kit) and made it available to users via SVN [25].

The *GBT-FPGA Starter Kit* [26] provides a set of blocks in charge of each step of the GBT serialization-deserialization procedure, coded in the most generic way. Most of these blocks are common for Altera [27] and Xilinx designs. However, a few blocks had to be designed using vendors core generators: PLLs, Transceivers and dual-port RAMs.

It is important to mention that besides the standard configuration of the GBT HDL module, optimized versions in resources and in latency are being developed by the same team responsible of the standard version [28].

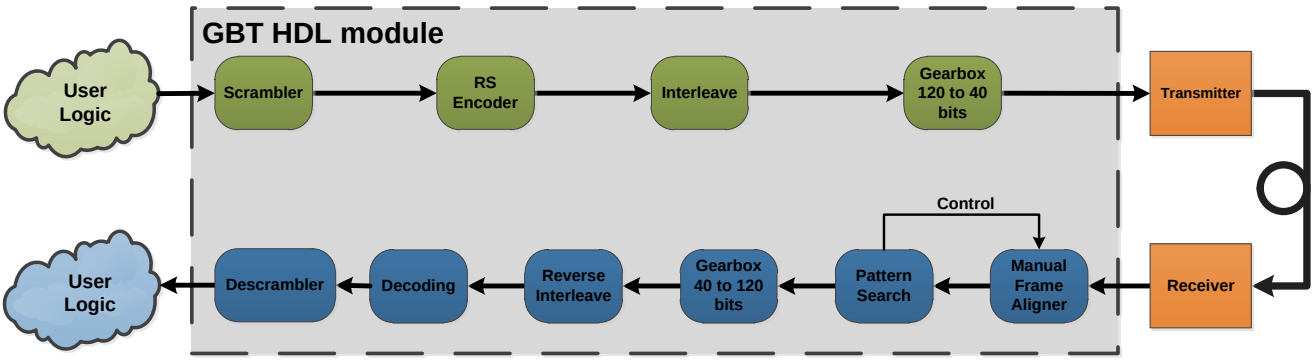


Figure 3-39: GBT HDL module block diagram.

3.2.7. Other HDL modules and IP cores

There are other HDL modules and vendor specific IP cores needed for the BE-FE link simulation project. These elements and their setups are explained in 3.3. These HDL modules and vendor specific IP cores are:

- Clock buffer primitives (BUFG) (IBUFGDS) (IBUFDS_GTXE1)
- CDCE Synchronizer (cdce_synchronizer.vhd)
- General Reset (reset_ctrl.vhd)
- GBT Link Reset Control (glib_link_rst_ctrl.vhd)
- Pattern Generator (pattern_generator.vhd)
- GTX Synchronizers (gtx_tx_sync.vhd) (gtx_rx_sync.vhd)
- Bit Slip Control (bitslip_control.vhd)
- GTX RX Phase Aligner (phase_aligner)
- Bit Error Ratio Test (BERT) (bert.vhd)
- Pattern Match Flag (patter_match_flag.vhd)

3.2.8. Optical transceiver (TRX10GDP0310)

The MERGE OPTICS TRX10GDP0310 [29] is an up to 10 Gbps SFP+ 1310 nm Limiting transceiver for single mode optic fibre. This optical transceiver is in charge of convert the information which comes from the GTX transceivers of the Virtex-6 from electrical to optical format, and back again, at different data rates depending upon the chosen standard.

Figure 3-40: Optical transceiver (MERGE OPTICS - TRX10GDP0310).

The TRX10GDP0310 is compliant with the current SFP+ MSA specifications (SFF-8431 and SFF-8432) and with 10GBASELR/LW per IEEE 802.3ae. It is also compliant with RoHS 6/6 per Directive 2002/95/EC and laser class 1 safety compliant per IEC/CDRH. The transmit path consists of an AC coupled 100 Ω differentially terminated driver coupled to a 1310 nm (single mode optic fibres) distributed feedback laser (DFB). The receiver path consists of a ROSA (receiver optical sub-assembly) for optical electrical conversion, followed by a limiting amplifier to boost the electrical signal. Complete digital optical monitoring is implemented in compliance to SFF-8472 and made accessible via the 2-wire interface providing real time information about all important module parameters and status information.

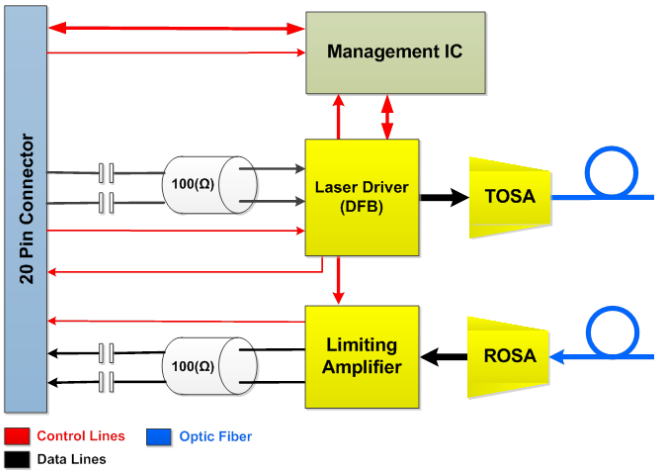


Figure 3-41: MERGE OPTICS - TRX10GDP0310 block diagram.

MERGE OPTICS TRX10GDP0310 Features:

- Compliant to SFP+ Electrical MSA SFF-8431
- Compliant to SFP+ Mechanical MSA SFF-8432
- 10G Ethernet
- Data rate transparent from 9.95 Gbps to 11.3 Gbps
- Support I2C for serial transceiver ID and digital diagnostic monitoring per SFF-8472
- Transmission distance up to 10 km (SM fibre)
- Low power consumption: 0.8 W (Typ.)
- 0 °C to +70 °C case operating temperature range
- Laser Class 1 IEC / CDRH compliant
- RoHS 6/6 compliant
- Compliant with product safety standards

3.2.9. Optical fibre cable

An optical fibre system is similar to the copper wire system. The difference is that optical fibre systems use light pulses to transmit information down fibre lines instead of using electronic pulses to transmit information down copper lines. The main advantages of the optical fibre cable over copper cable are:

- Much faster data transmission over longer distances
- Smaller diameter and less weight
- RFI (Radio Frequency Interference) and EMI (Electromagnetic Interference) immunity
- Less power consumption
- Less signal degradation
- Non-flammability
- Virtually unable to be tapped

There are three types of optical fibre cable commonly used: *plastic optical fibre* (POF), *multimode* (MM) and *single mode* (SM) which is the type selected for this project. This type of fibre gives you a higher transmission rate and up to 50 times more distance than multimode.

Figure 3-42: Optical fibre cable LC-LC (SM).

Table 3-4: Optical fibre characteristics.

Model	LC-LC 50/125 Simplex 2.0 mm LSZH 1m	
Length	1 m	
Connector	LC	LC
Insertion Loss	0.22 dB	0.28 dB
Date	2010.05.17	

3.2.10. FMC board for external clock feedback (FMC DIO 16CH TTL A)

The FmcDIO16chTTLa (EDA-02064-V1-0) [30] is a 2x 8 bits port digital IO card in FMC form-factor designed and developed at CERN. Each 8 bits port can be configured individually as input or output. IOs are TTL compatible. Additional test features can be mounted on the PCB.

It is interesting to mention that complete information about this board and many other electronic devices in addition to HDL modules can be found at *Open Hardware Repository* (OHWR) [31].

Figure 3-43: FMC board for external clock feedback (front).

Figure 3-44: FMC board for external clock feedback (back).

3.2.11. AMC breakout board (SMA-AMC Ultra 9000)

The SMA-AMC Ultra 9000 [32] is a fabric development tool with power, I2C and controls for AMC.0 (.1,.2,.3 and .4). Besides the features mentioned before, this board provides SMA connectors for all AMC lanes so it is possible to perform a full test of any AMC card.

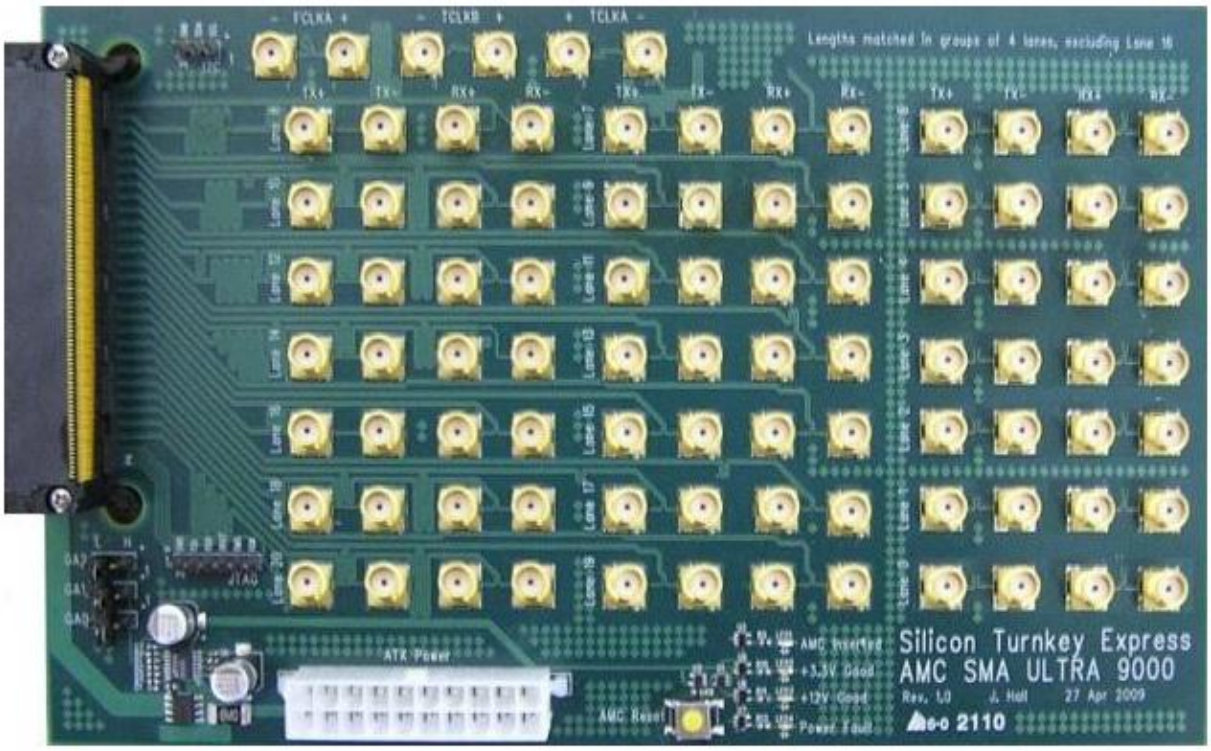


Figure 3-45: SMA AMC9000 ULTRA.

Features:

- 4 Lanes of AMC to SMA
- Control signals to SMA
- 90 SMA connectors
- 170 pin Yamaichi AMC female
- I2C for AMC available
- Power to AMC
- Reset to AMC

3.2.12. Laboratory equipment

Power supply (TTi TSX 1820P)

The TTi TSX 1820P [33] is a power supply for laboratory use. Besides the high quality of this power supply, one of the most notorious features is that the control of different parameters can be done either manually using the controls of the front panel, or remotely through GPIB or RS 232.

Figure 3-46: Power supply.

Features:

- High power 35 V/10 A (350 W)
- Single Output
- RS-232 and GPIB interfaces as standard
- Keyboard setting of all parameters
- Rotary and delta (step) control of V and I
- Watts display
- Non-volatile storage of 25 settings
- Fully programmable with bus readback of V & I

USB to JTAG interface (Xilinx Platform Cable USB II)

The Xilinx Platform Cable USB II provides an interface between USB and JTAG, to easily perform fast and reliable configuration of Xilinx FPGAs, programming Xilinx PROMs and CPLDs, and directly programming third-party SPI flash devices. Moreover, in addition with the cable included provides a means of indirectly programming Platform Flash XL, third-party SPI flash memory devices, and third-party parallel NOR flash memory devices via the FPGA JTAG port. Furthermore, Platform Cable USB II is a cost effective tool for debugging embedded software and firmware when used with applications such as Xilinx's Embedded Development Kit or ChipScope.

Figure 3-47: USB to JTAG interface.

Oscilloscope (LeCroy WaveRunner LT354M)

The LeCroy WaveRunner LT354M is a high-end digital oscilloscope. With 500 MHz of bandwidth, this oscilloscope offers a resolution of 2 ns that is more than enough for this project, where the minimum period measured is 4.1667 ns (240 MHz of frequency).

Figure 3-48: Oscilloscope.

Features:

- 4 Channels
- 500 MHz Bandwidth
- 1 GSps Single-Shot Sampling Rate on all channels
- 50 GSps Repetitive Sample Rate
- Better than 10 ppm Timebase Accuracy with 5 ps resolution
- Up to 2 Mpt waveforms on all channels (with option ML)
- 8.4” TFT LCD Colour Display
- SMART Triggers include Slew Rate and Runt to <= 2.5 ns (optional)
- Analog Persistence feature with History view
- QuickZoom button automatically magnifies signal views
- Wavepilot provides quick access to analysis views of measurements such as: FFTs, Histograms, and TrackViews
- Averaging and Enhanced Resolution up to 11 bits
- Deskew and Rescale
- GPIB, RS-232-C, VGA and Centronics Ports (Standard); Ethernet (Optional)
- Automatic Pass/Fail Testing
- PC Card support for hard drives and memory cards
- Internal Graphics Printer option

3.2.13. Software

All software needed for the development of this project come from ISE, the design suite offered by Xilinx for its FPGA and CPLD devices. Although ISE consists of several programs, only Project Navigator, Core Generator and ChipScope Pro Analyzer are used in this project.

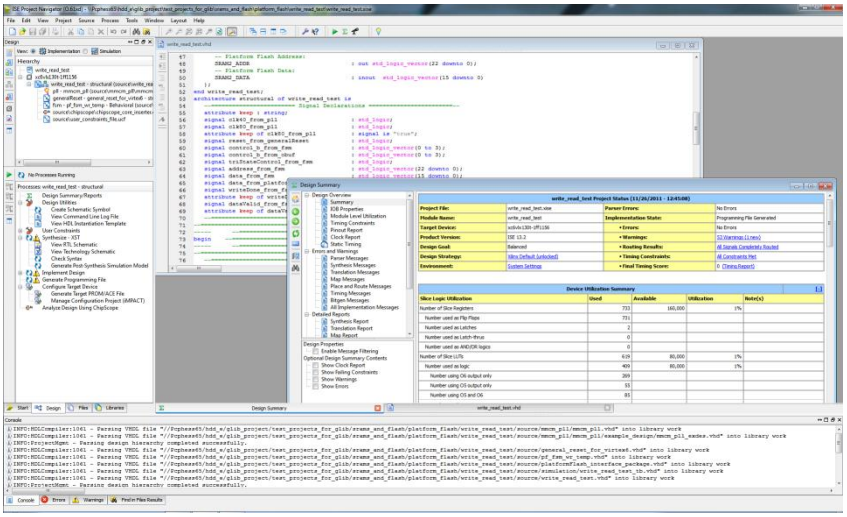


Figure 3-49: Project Navigator.

- Project Navigator is an HDL design tool that allows the edition, synthesis and implementation of the code generated by the user.

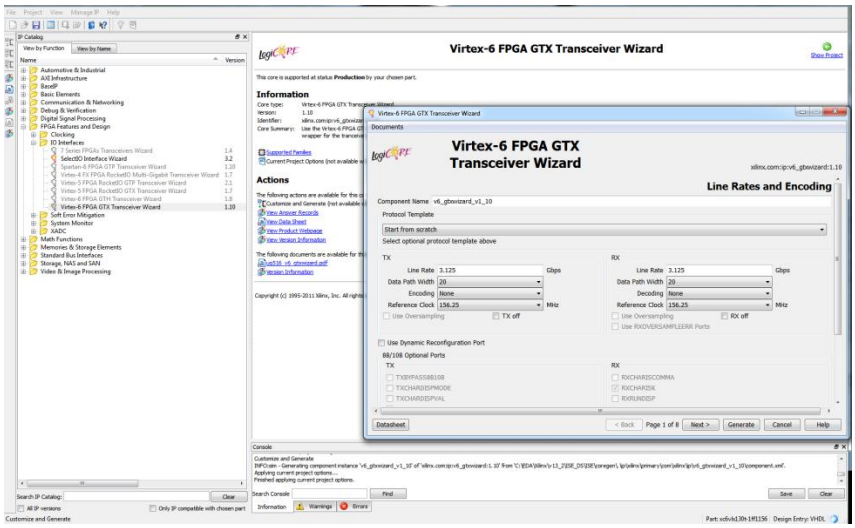


Figure 3-50: Core Generator.

- Core Generator is a program that provides user friendly wizards for the configuration of complex IP cores.

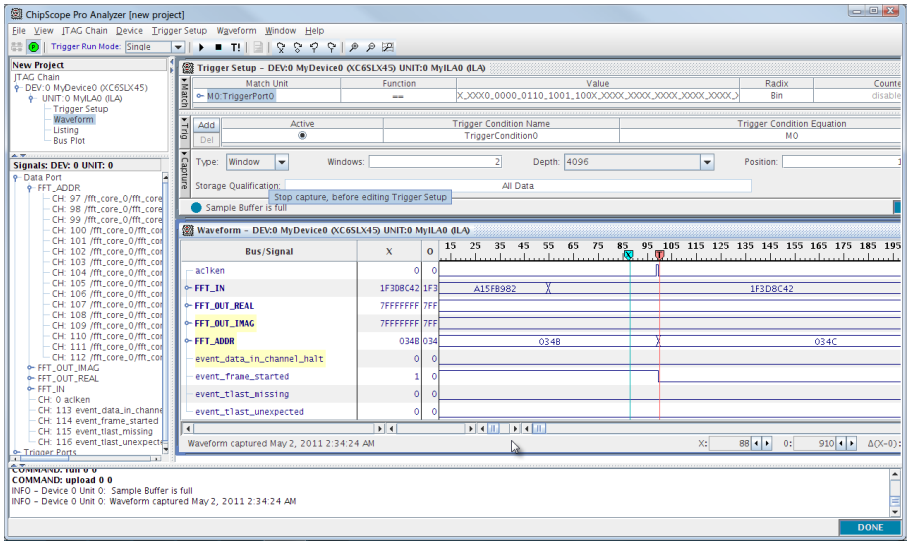


Figure 3-51: ChipScope Pro Analyzer.

- ChipScope Pro Analyzer is an on-chip Signal Analyzer, which is implemented by utilising logic resources of the FPGA. Moreover, this software can also be used for FPGA configuration.

3.3. System setup

In this section are described the system setup and the reasons because this setup has been chosen. There are also briefly described the most notorious issues that our team have found during the development of this project. The system setup is explained in general for both BE and FE systems, but mentioning the corresponding differences when they exist.

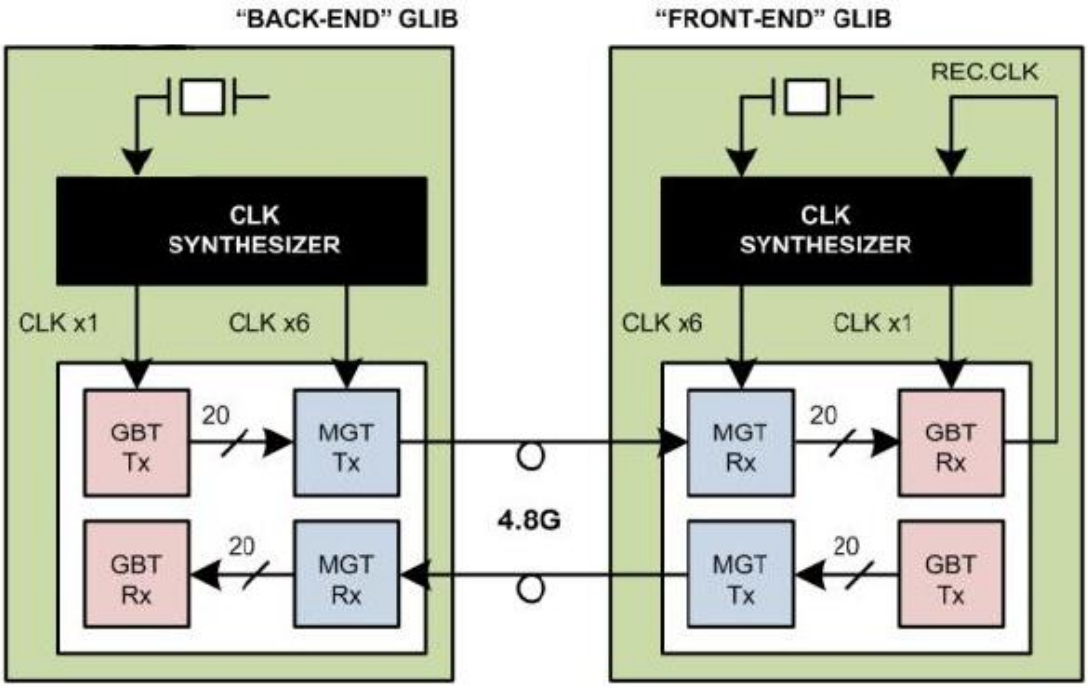


Figure 3-52: General block diagram.

3.3.1. GLIB (BE) and GLIB (FE) block diagrams

GLIB (BE) block diagram

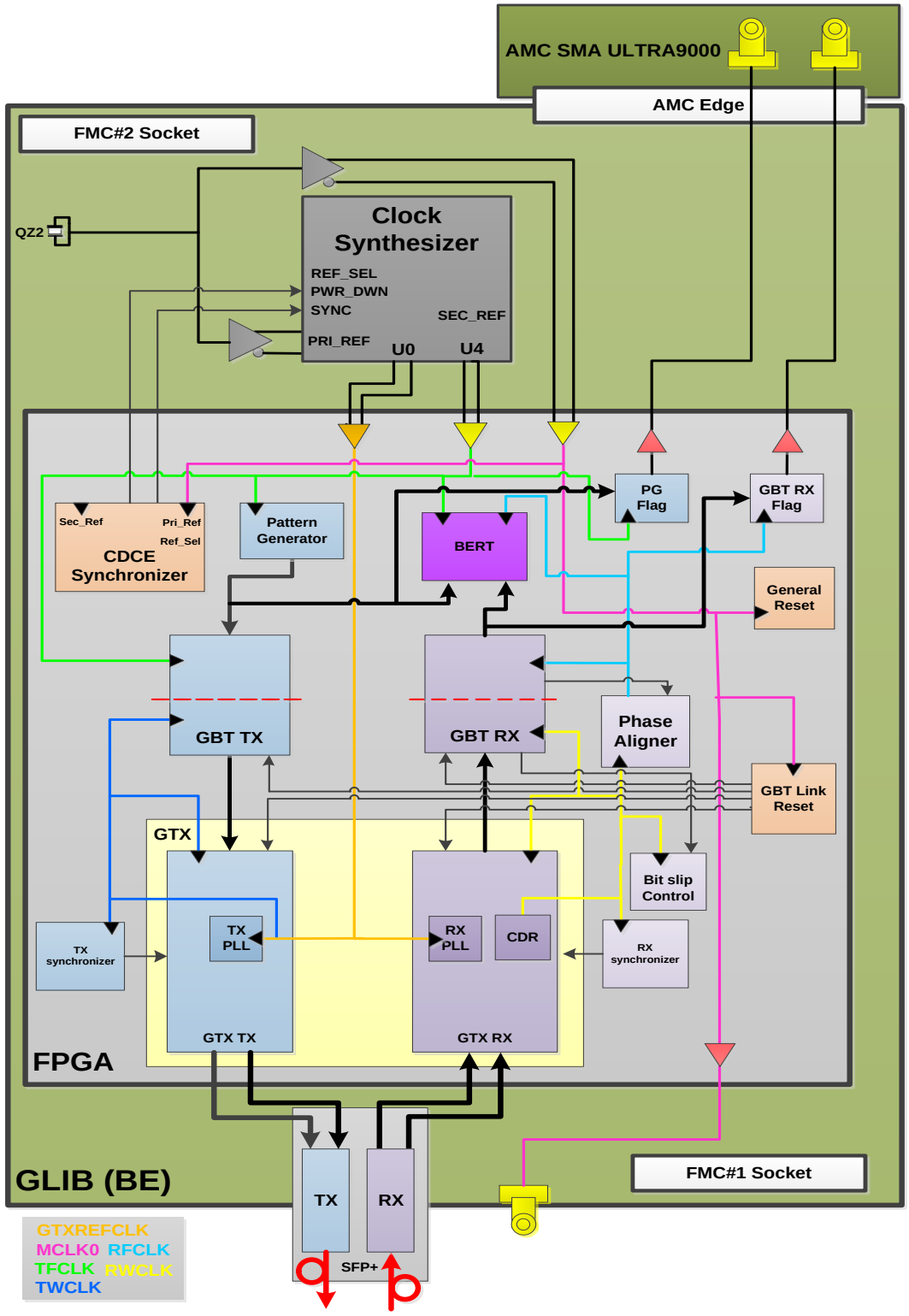


Figure 3-53: GLIB (BE) block diagram.

GLIB (FE) block diagram

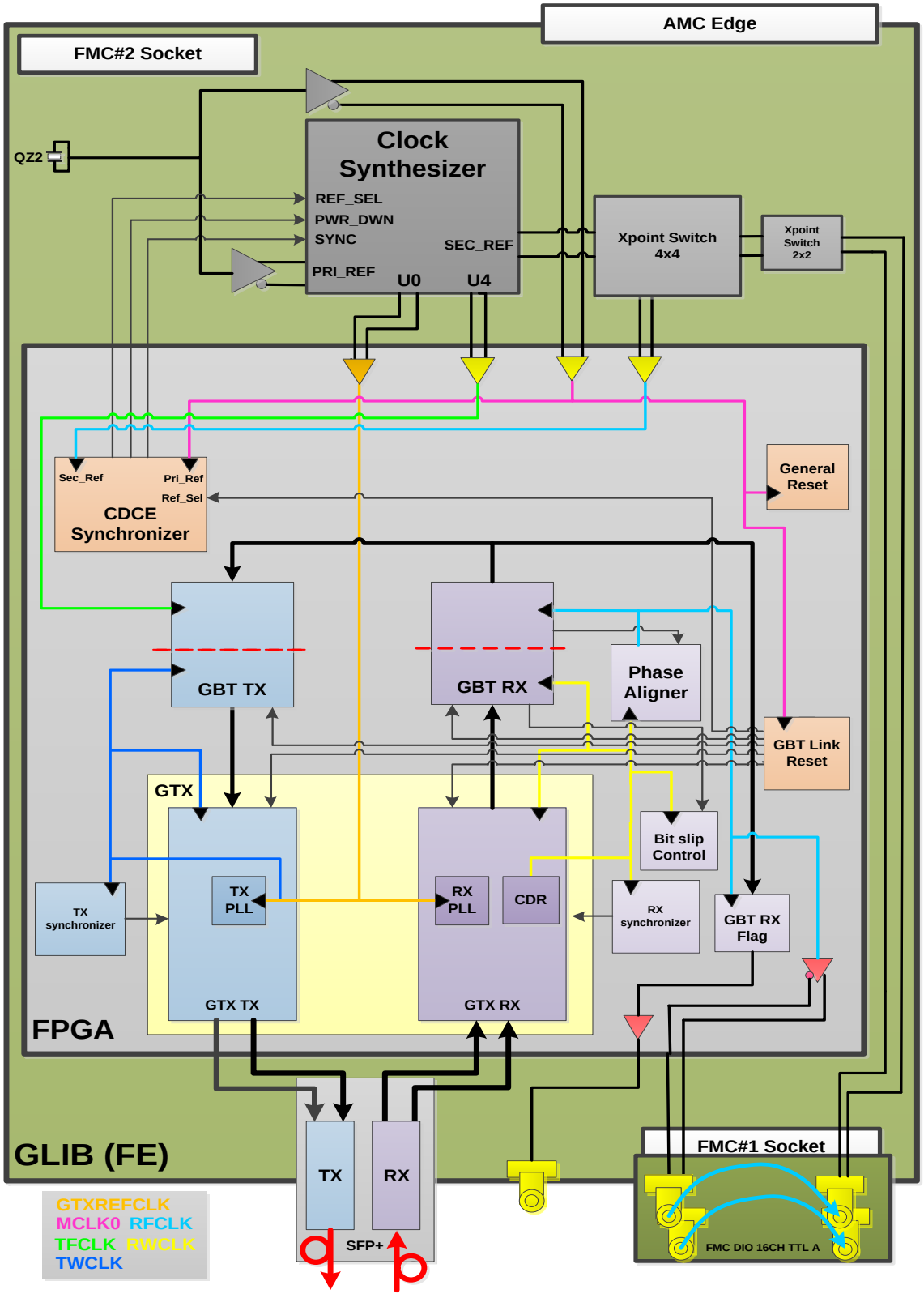


Figure 3-54: GLIB (BE) block diagram.

3.3.2. System clocks and clock domains

Source clocks

The source clock of all devices and modules utilized in this project comes from the 40 MHz crystal oscillator FXO-HC736R-40 (QZ2) of the BE board (on the FE board the crystal oscillator is only used before the clock swap procedure). Due to the single ended output of the FXO-HC736R-40 is not the best configuration for signal transmission, on the GLIB board it is performed the conversion of the single ended output to differential signalling (LVDS in our case) for improving the signal integrity. By the utilisation of differential signalling, data and clocking signals can be transmitted over printed-circuit-board traces or cables at very high rates with very low electromagnetic emissions and power consumption. The GLIB board incorporates two differential drivers (SN65LVDS1) connected to the output of the FXO-HC736R-40. One driver (IC11) goes directly to the FPGA fabric (MCLK0) through an IBUFGDS Xilinx primitive, while the other driver (IC12) is connected to the primary reference clock of the *Clock Synthesizer* (CS) (MCLK1).

Note!!! The crystal oscillator is only used by the CS of the FE board before the clock swap.

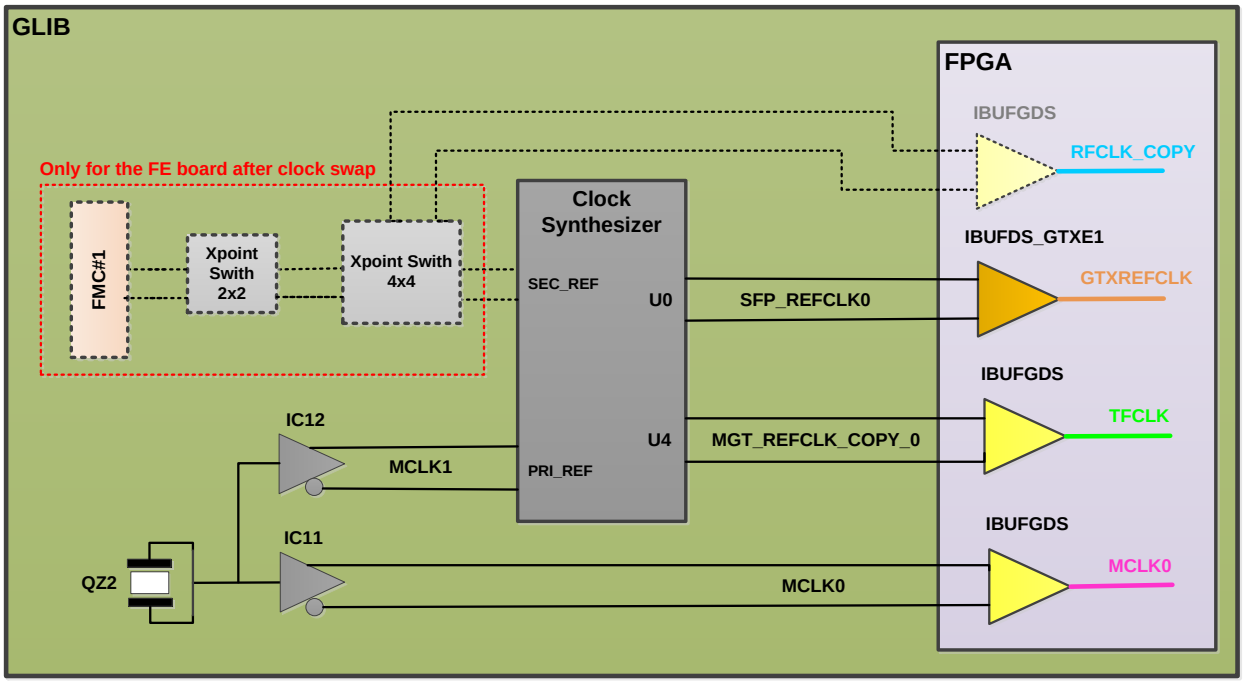


Figure 3-55: Source clocks.

Clock synthesizer

Clock synthesizer setup

Regarding the reference clock BE and FE present different requirements. On the BE board the CS is always clocked by the primary reference clock. However, on the FE board, after power up the CS is clocked by the primary reference clock but when the CDR of the RX is able to manage the clock recovery, it is performed a clock swap and then is the secondary reference clock which is used for clocking the CS.

For this project, only two clock outputs are needed (SFP_REFCLK0 (U0) and MGT_REFCLK_COPY_0 (U4)). SFP_REFCLK0 is used as reference clock of the GTX transceiver connected to SFP1 (GTXREFCLK) and its frequency is configured at 240 MHz. On the other hand, MGT_REFCLK_COPY_0 goes to the logic fabric and it is configured at 40 MHz. Both clocks are synchronized with the reference clock of the CS and the phase relationship between them is deterministic.

To perform the CS configuration, the SPI port is used to program the desired values on the internal registers. However, this operation is performed only once because during the SPI configuration, the setup is stored on the internal EEPROM. Once the desired values are stored on the EEPROM, the registers are automatically configured after power up.

Table 3-5: CDCE registers values.

CDCE registers values		
Register	**Value**	**Description**
Register 0	0xEB840720	U0=240 MHz, LVDS, phase shift 90deg
Register 1	0xEB020321	U1=160 MHz, LVDS
Register 2	0xEB020302	U2=160 MHz, LVDS
Register 3	0xEB840303	U3=240 MHz, LVDS
Register 4	0xEB140334	U4= 40 MHz, LVDS, R4.1=1
Register 5	0x013C0CF5	LVDS in, DC term, PRIM REF enable, SEC REF enable, smartMUX off etc.

Table 3-4: CDCE registers values (cont.).

Register 6	0x33041BE6	VCO1, PS=4, FD=12, FB=1, ChargePump 50uA, Internal Filter, R6.20=0, AuxOut= enable; AuxOut= U2
Register 7	0xBD800DF7	C2=473.5 pF, R2=98.6 kΩ, C1=0 pF, C3=0 pF, R3=5 kΩ etc., SEL_DEL2=1, SEL_DEL1=1
Register 8	0x20009978	Various

Clock synthesizer phase determinism

As it was mentioned before, the phase of the output clock is synchronized with the phase of the 40 MHz reference clock. This synchronization process ensures that the phase difference between both clocks is deterministic. The phase synchronization is performed by driving low the SYNC pin of the CS. This synchronization pin can be controlled asynchronously, but on the user guide, it is recommended to synchronize the signal that controls this pin with the reference clock of the CS. The clock swap and the control of the synchronization procedure are carried out by the *CDCE Synchronizer* module (cdce_synchronizer.vhd). The synchronization procedure is controlled by an internal *Finite State Machine* (FSM). The timing sequence of the Power_Down and SYNC pins of the CS. During the synchronization procedure is shown next.

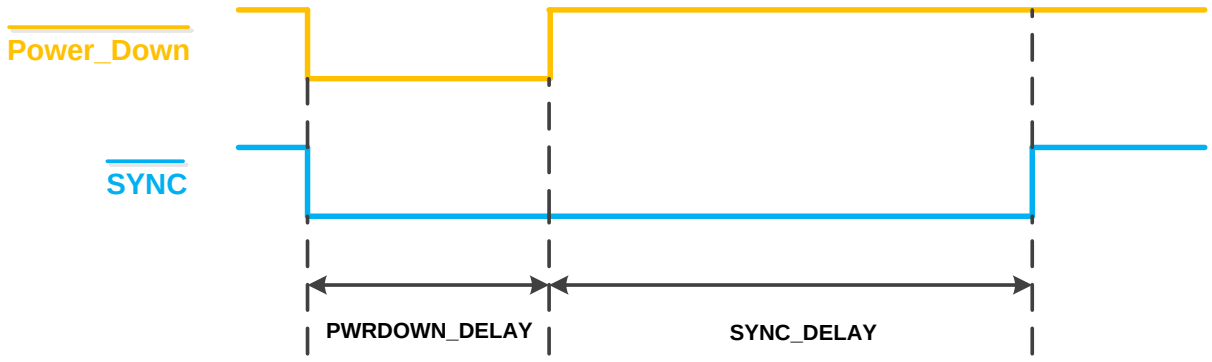


Figure 3-56: CS synchronization timing diagram.

Nevertheless, for the reference clock swap, the *CDCE Synchronizer* only features combinatorial logic so the control of the procedure is performed by other module, the *GBT Link Reset Control* (GLRC) (which will be described later).

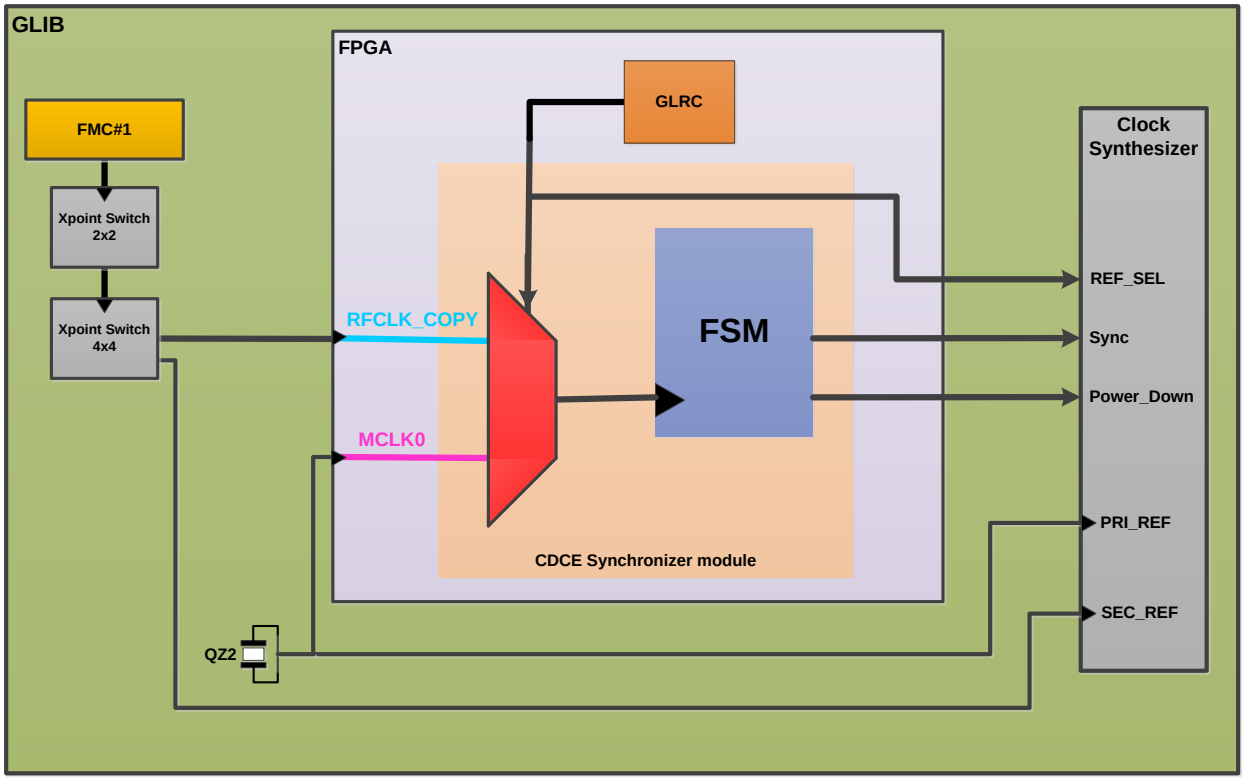


Figure 3-57: CS synchronization and clock swap block diagram.

The main combinatorial element of the CDCE Synchronizer is the clock multiplexor, a BUFGMUX Xilinx primitive for the Virtex-6. This multiplexor includes two inputs. One input is connected to MCLK0, which is synchronous with the primary reference clock of the CS. The other input is connected to a 40 MHz clock (RFCLK_COPY) which comes from the GTX RX Phase Aligner and is synchronous with the secondary reference clock of the CS. The output of the multiplexor is used as reference clock for the FSM. This clock multiplexor is needed for the correct synchronization after the clock swap procedure (the FSM has to be clocked by a clock synchronous with the reference clock of the CS). On the BE board, the CS is always clocked by the primary reference clock so the output of the multiplexor is always the primary reference clock. However, on the FE board, after the clock swap procedure, the phase of the output clocks of the CS must be aligned with the secondary reference clock. For that reason when performing the clock swap, the input connected to the

secondary reference clock is selected in the clock multiplexor and forwarded to the FSM.

It is important to mention that the phase synchronization process is not always accomplished correctly. Sometimes there is 1ns of error in the phase difference. This error is very rare (error < 5 after 500 powers up and not always happens). The issue is admitted by Texas Instruments and is due to the internal functioning of the CS. The workaround of this issue is commented on the section 4.2.

Clock domains

There are seven different clock domains within the FPGA of each board in this project. Any synchronous element either can be completely included in one or can belong to several of these clock domains. It is important to mention that in both boards, during normal operation (after the CS reference clock swap on the FE board) these clock domains are synchronized and phase deterministic with the 40 MHz crystal oscillator (QZ2) of the BE board.

- **MCLK0 clock domain:** this clock domain is sourced by MCLK0.
- **GTX reference clock domain:** this clock domain is sourced by *GTX Reference Clock* (GTXREFCLK), which actually is the fabric version of SFP_REFCLK0 after an IBUFDS_GTXE1.
- **TX frame clock domain:** this clock domain is sourced by *TX Frame Clock* (TFCLK), a 40 MHz clock, which actually is the fabric version of MGT_REFCLK_COPY_0 after an IBUFGDS.
- **TX word clock domain:** this clock domain is sourced by *TX Word Clock* (TWCLK), a 240 MHz clock, which comes from the port TXOUTCLK of the GTX transceiver through a BUFG. It is important to mention that this clock is actually a forwarded version of GTXREFCLK.
- **TX serial clock domain:** this clock domain is clocked by *TX Serial Clock* (TSCLK), a 2.4 GHz clock, which is only used by the serial part of the GTX transceivers. However, it is important to mention that serial data is sampled in *Dual Data Rate* (DDR) mode so the actual lane rate is 4.8 Gbps
- **RX frame clock domain:** this clock domain is sourced by *RX Frame Clock* (RFCLK), a 40 MHz clock, which comes from the *RX Phase Aligner*. Nevertheless, only on the FE board there is also a forwarded copy of this clock (RFCLK_COPY).
- **RX word clock domain:** this clock domain is sourced by *RX Word Clock* (RWCLK), a 240 MHz clock, which is the fabric version of RXRECCLK after a BUFG.

3.3.3. Reset & Clock Swap

The reset procedure is very important for any *Sequential System* (SS) (synchronous or asynchronous). This procedure has to be performed before the normal functioning. It ensures that all flip-flops of the system are in a known state. Moreover, hardware component and modules synchronization can be accomplished by connecting all of them to the same reset signal. In this project, two HDL modules are responsible of the reset procedure.

General reset

Although the GLIB board features a power up reset by hardware, this reset is not executed after FPGA configuration. To overcome this problem, the *General Reset* (GR) module is included in this project. This module is clocked by MCLK0 because this clock is operational after power up.

The reset module consists of a SRL16 Xilinx primitive for Virtex-6 and a 32 bits counter. The SRL16 is a synchronous shift register implemented by using logic resources of the FPGA. This primitive has been selected for this purpose because includes the feature that its FFs are reset to a known state automatically after FPGA configuration.

The reset procedure performed by the GR is described next. The SRL16 is used as active low reset for the 32 bits counter just after configuration. This is done by connecting the input of the SRL16 to a logic '1', the output to the reset signal of the 32 bits counter (which has an asynchronous active low reset) and setting the preload with '0's (which are the reset pulse). When all '0's of the preload have been shifted out, the reset pulse finishes and then the output of the SRL1 becomes always '1' (this reset procedure is done only once after configuration). Once the 32 bits counter has been properly reset, it starts counting until reach the end of count chosen by the user. As soon as this occurs, the counter asserts an active high pulse, which is the actual reset for the rest of the system.

GBT Link reset & clock swap

For the correct synchronization of the BE-FE transceivers, the TX and RX parts of the GBT Link of each board must be reset following a specific sequence, as well as the clock swap must be performed at the correct moment. The *GBT Link Reset Control* (GLRC) module is responsible for this sequence. This module is also clocked by MCLK0.

The sequence performed by GLRC is described next:

1. The TX of the BE board is reset.
2. The RX of the FE board is reset (at this point the CDR must be able to recover the clock and data from the serial frame).
3. The clock swap is performed (the GTX RX of the FE board needs to be aligned again).
4. The TX of the FE board is reset.
5. The RX of the BE board is reset (at this point the CDR must be able to recover the clock and data from the serial frame).

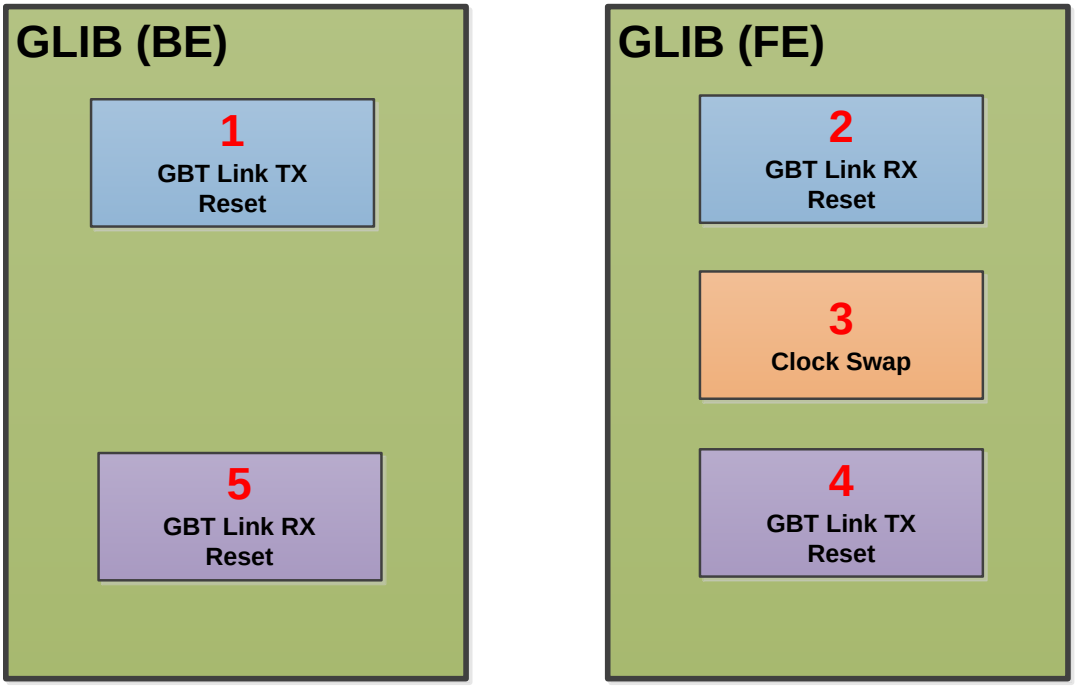


Figure 3-58: GBT Link Reset and Clock Swap sequence.

3.3.4. GBT Link TX setup

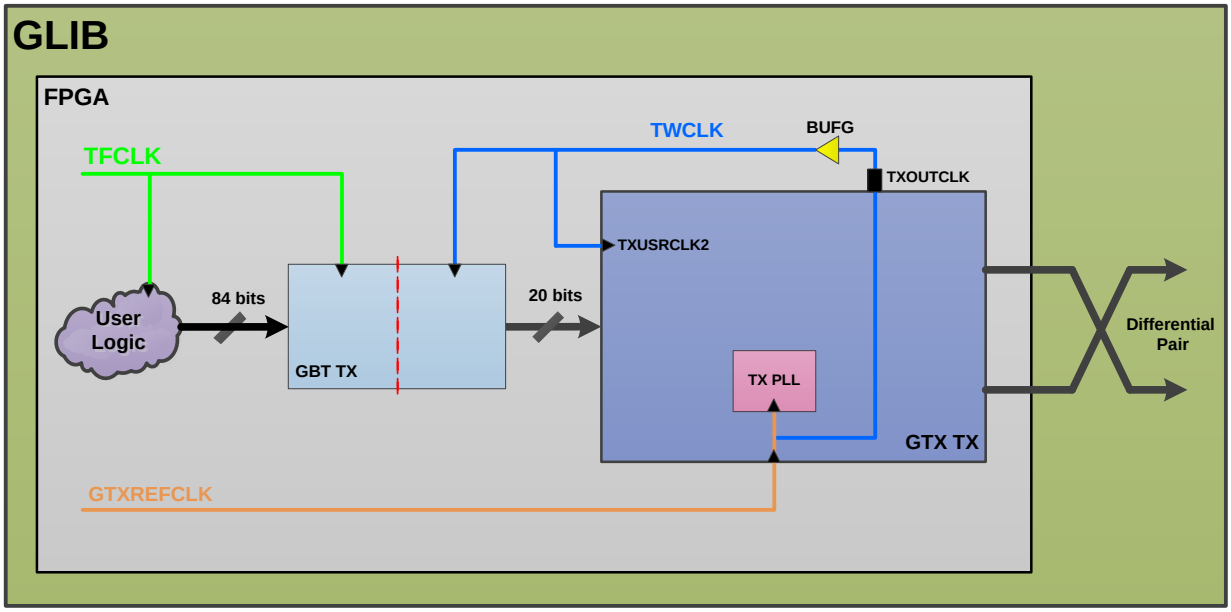


Figure 3-59: GBT Link TX block diagram.

GBT TX

Due to the low and deterministic latency requirements, the optimized version of the GBT module has been chosen for this project. The difference between the standard and the optimized version is the utilisation of registers for the internal gearbox instead of dual port memories.

Modified files for GBT TX opt:

- Scrambler_opt.vhd
- Scrambling_opt.vhd

The placement of the GBT TX module is between the User Logic (UL) and the GTX TX. Each interface of the GBT TX belongs to a different clock domain so this module is clocked by two reference clocks (TFCLK and TWCLK).

- The part that interfaces with the UL presents an 84 bits data path clocked by TFCLK.
- The part that interfaces with the GXT TX has been modified with respect to the original GBT module. In the original GBT project, the 40 bits interface between GBT TX and GTX TX is clocked by a 120 MHz clock whereas for this project the

frequency of this clock has been increased to 240 MHz (TWCLK). To be able to keep the same data rate (4.8 Gbps) the width of the data path between the GBT TX and the GTX TX has been reduced to 20 bits.

Modified files for GBT TX 20 bits interface:

- Mux_120_to_20bits_opt.vhd

GTX TX

The correct configuration of the GTX is one of the keys for the success of the low latency and deterministic communication. There are 4 main points that were taken into account when choosing the correct setting of the GTX TX:

- Clock resource optimization.
- Internal clock domains unification to avoid latency indeterminism.
- Elastic buffer bypassing and internal clock alignment to achieve deterministic latency operation.
- Unused internal blocks bypassing to achieve the lowest latency possible.

The clock resource optimization and the internal clock domain unification are achieved by choosing an appropriate reference clock frequency and data path width. The frequency chosen for the TX PLL reference clock is 240 MHz (GTXREFCLK) and the data width selected for the TX FPGA interface is 20 bits. Using this configuration it is possible to forward the TX PLL reference clock through TXOUTCLK to TXUSRCLK2 using just a BUFG (TXUSRCLK is feed internally by TXUSRCLK2). Thereby, TXUSRCLK2 and TXUSRCLK have been unified becoming a single clock domain, and it is not necessary to use any internal PLL (MMCM) of the FPGA.

For bypassing the buffer, it is necessary to perform the phase alignment between TXUSRCLK and XCLK. After the phase alignment procedure, this two clock regions are merged as well so the parallel data path of the GTX TX only comprises one clock domain instead of three. This alignment procedure is performed by the *GTX TX Synchronizer* (TXSYNC) module. This module is provided by Xilinx and can be found within the example folder, which is automatically generated by Core Generator when the GTX module is generated.

The GBT TX module features the components necessary to handle the GBT frame (encoder, gearbox etc.). Consequently, the number of internal blocks of the GTX TX transceiver can be minimized to improve the latency. Figure 3-60 shows the actual configuration of the GTX TX for this project while the latency of each block (in terms of integer numbers of TXUSRCLK clock cycles) is showed on Table 3-6.

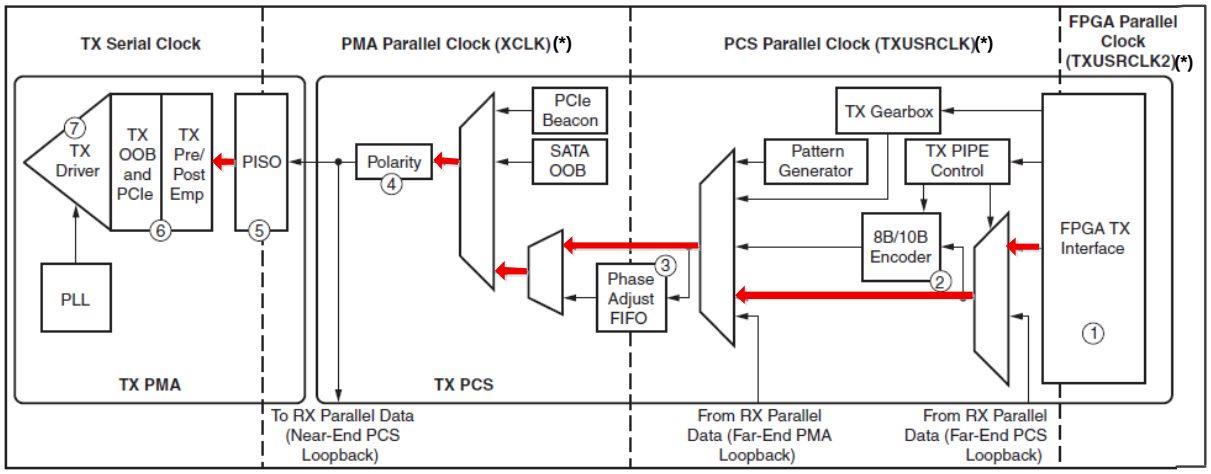


Figure 3-60: GTX TX internal configuration.

Table 3-6: GTX TX latency.

Block Number	Block Name	Setup	TX Latency (TWCLK cycles) (*)
1	FPGA TX Interface	Data Width= 20 bits	1 cycle
2	8B/10B Encoder	Bypassed	0 cycles
3	TX Elastic Buffer	Bypassed	1 cycle
4+5+6+7	PMA + Interface		3 cycles
Total TX Latency	5 cycles		

The screenshots of the different pages of the Core Generator wizard for the GTX configuration are shown on Appendix B.

(*) TXUSRCLK2, TXUSRCLK and XCLK are forwarded versions of TWCLK

3.3.5. Optical transceivers setup

Although the GLIB board features a cage for SFP+ optical modules, for this project only one is needed. On both boards (BE and FE), the SFP+ optical transceiver (TRX10GDP0310) is connected to SFP1 (J15). The connection of the transceivers is performed by the utilisation of two single mode optical cables.

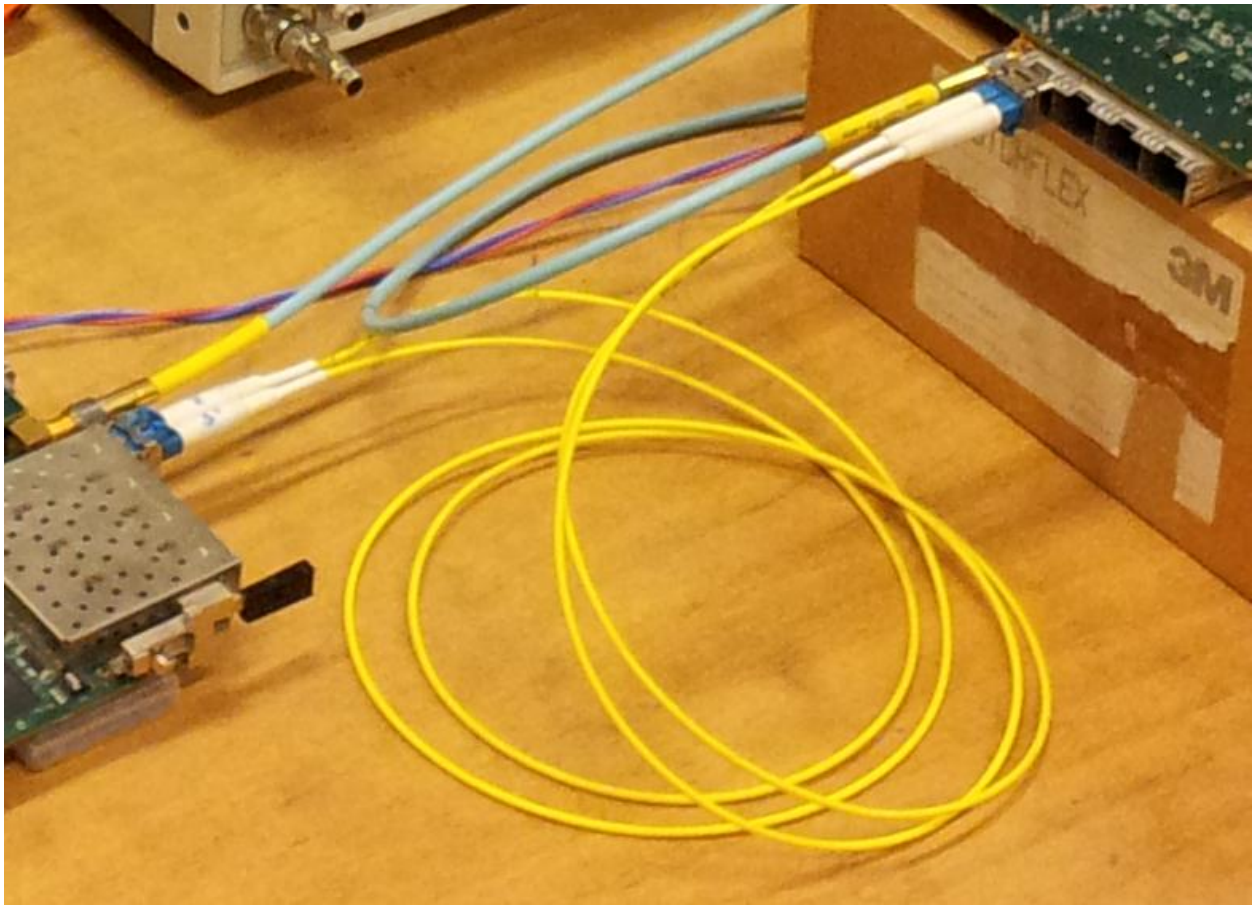


Figure 3-61: Optical transceivers connection through optic fibres detail.

3.3.6. GBT Link RX setup

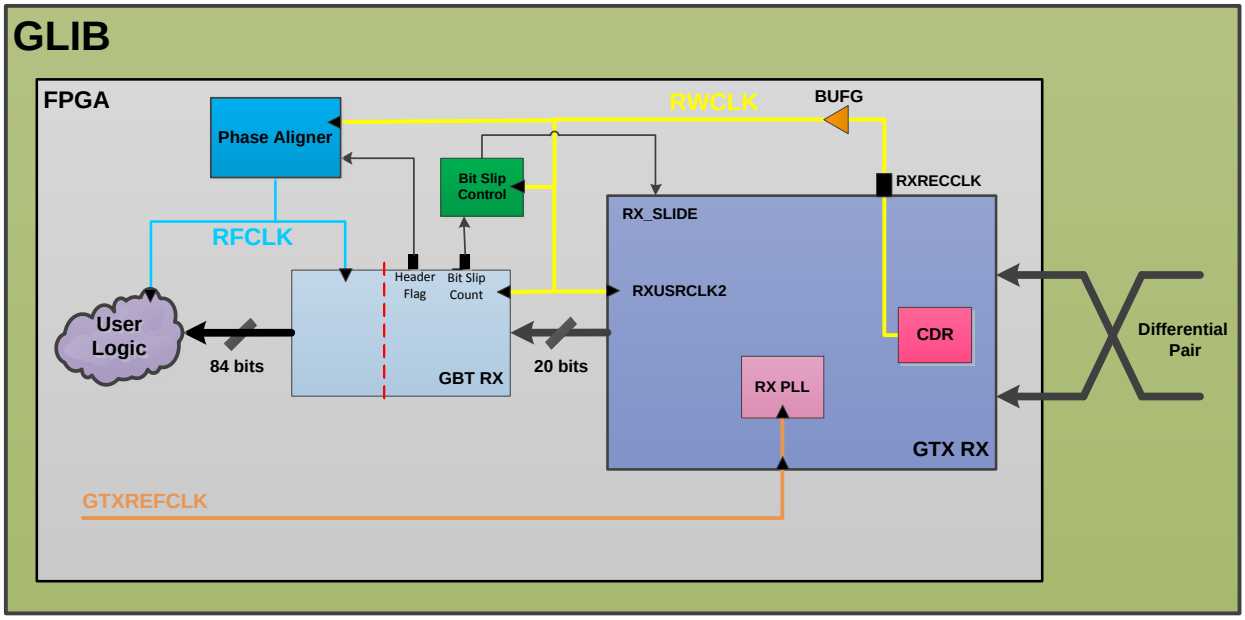


Figure 3-62: GBT Link TX block diagram.

GTX RX

The configuration of the GTX RX is very similar to the configuration of the GTX TX. The main difference is that the GTX RX performs clock recovery from the incoming frame so RWCLK comes from the internal CDR of the GTX RX instead of the CS.

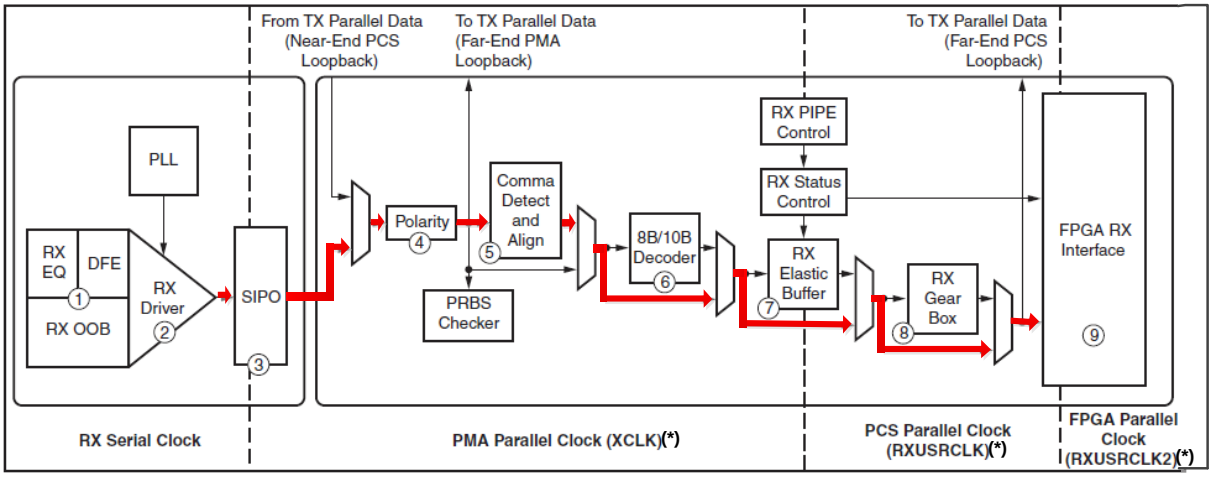


Figure 3-63: GTX RX internal configuration.

Table 3-7: GTX RX latency.

Block Number	Block Name	Setup	RX Latency (RWCLK cycles) (*)
1+2+3+4	PMA + Interface		4 cycles + 7 UI ± 1 UI (#)
5	Comma Detect	Enabled (Manual Alignment)	2 cycles
6	8B/10B Encoder	Bypassed	0 cycles
7	RX Elastic Buffer	Bypassed	0 cycles
9	FPGA RX Interface	Data Width= 20 bits	2 cycles
Total RX Latency	8 cycles + 7 UI ± 1 UI (#)		

The screenshots of the different pages of the Core Generator wizard for the GTX configuration are shown on Appendix B.

The clock recovery procedure is affected by an issue that may originate a latency indeterminism of one RWCLK period. The problem arises when TSCLK is divided by ten in order to obtain RXRECCLK. Due to the clock period of RXRECCLK is ten times the period of TSCLK and that the internal *Serial Input Parallel Output* (SIPO) module of the GTX RX samples data in DDR mode, there are twenty possible edges to lock with. This means that there are twenty possible phases for RXRECCLK. This phase indeterminism produces a variation in latency of the data when crossing from the serial to the parallel domain.

Fortunately, the GTX RX transceiver includes an internal block, the “RX Byte and Word Alignment”, which can perform a bit slip with phase shift of the RXRECCLK (RWCLK) when the RX_SLIDE port is driven high. It is important to mention that the generic RX_SLIDE_MODE must be set to “PMA” otherwise only the bit slip is performed. The bit slip of the data would not need to be executed by GTX RX because the GBT RX also presents this feature. However, the phase shifting feature allows controlling the phase of RXRECCLK (RWCLK). This control leads the latency of the incoming data frame to a deterministic behaviour.

(*) RXUSRCLK2, RXUSRCLK and XCLK are forwarded versions of RWCLK

(#) The Unit Interval (UI) in this case is 200 ps (5 Gbps)

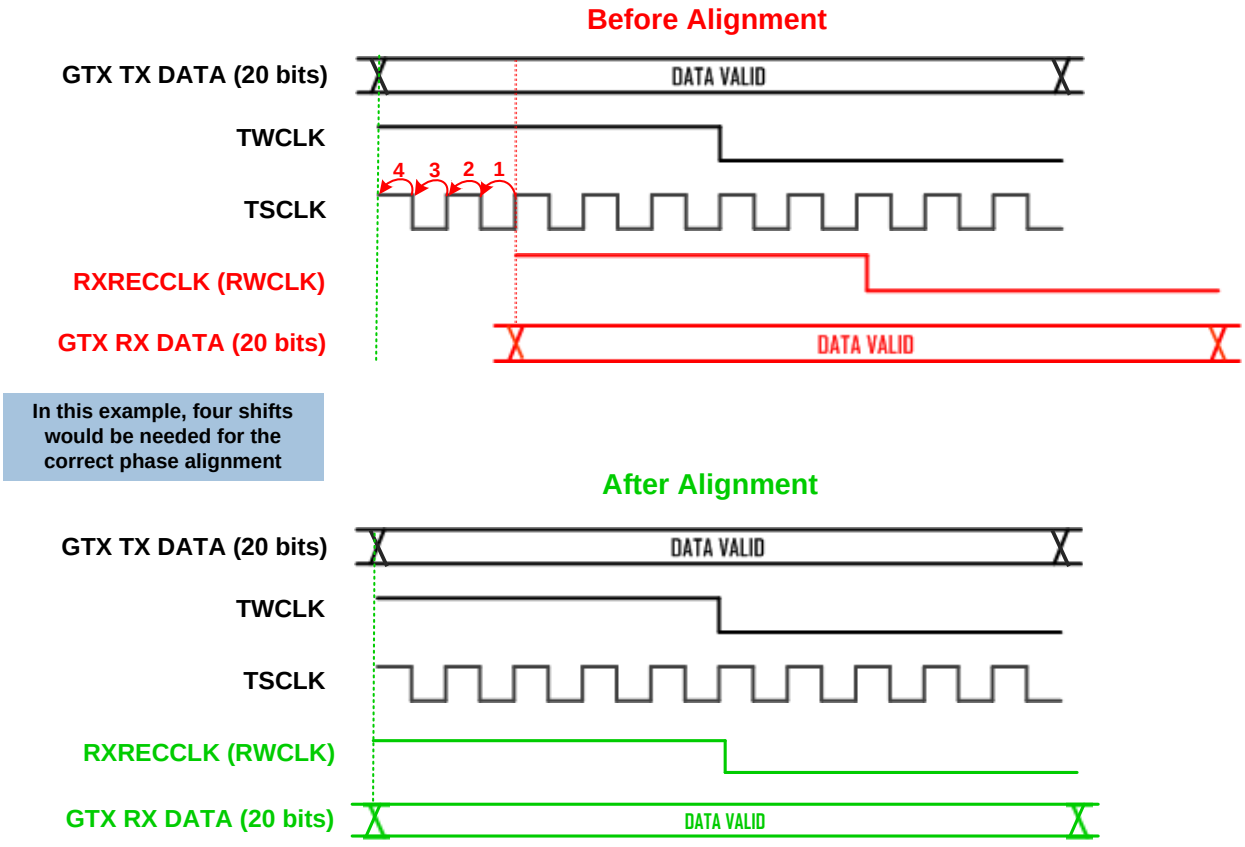


Figure 3-64: RXRECCLK (RWCLK) phase alignment timing diagram.

Bit slip control module

The *Bit Slip Control* (BSC) module is responsible of controlling the RX_SLIDE port of the GTX RX. The number of shifts needed for the correct phase alignment is provided by GBT RX. The control of the RX_SLIDE port must be synchronous with the recovered clock so RWCLK is used as reference clock for this module.

RX phase aligner

The mission of the *RX Phase Aligner* (PA) is to provide a phase deterministic 40MHz clock (RFCLK) synchronous with RWCLK. The frequency of RFCLK is achieved by dividing the incoming RWCLK by six. This procedure presents a similar problem as when TSCLK was divided to obtain RWCLK. However, in this case there are six possible edges to lock with (now the parallel logic samples in *Single Data Rate* (SDR) mode). The procedure followed for choosing always the same clock edge is explained next. First, the PA waits until GBT RX is aligned and asserts the gbtRx_aligned signal.

Then the PA counts the number of RWCLK cycles until the header_flag signal is asserted by GBT RX (this signal is phase deterministic with respect to TFCLK). The value of the counter will be used as number of phase shifts. Finally, PA shifts the phase of RFCLK which becomes phase deterministic with respect to TFCLK (and to QZ2 of the BE board).

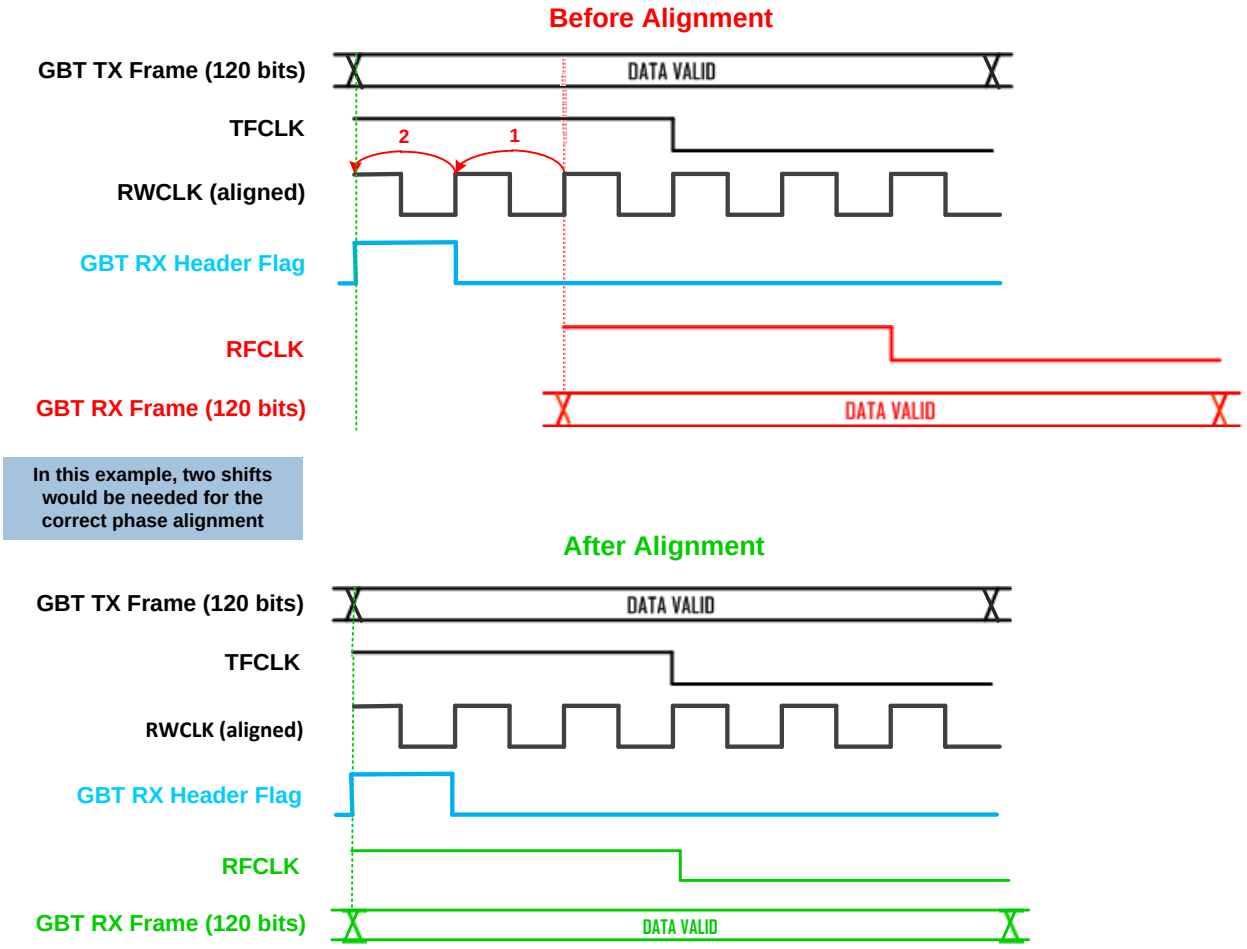


Figure 3-65: RFCLK phase alignment timing diagram.

GBT RX

The configuration of the GBT RX is the same as the configuration adopted for the GBT TX (optimized version and 20 bits interface modification).

Modified files for GBT RX opt:

- Decoding_opt.vhd
- Descrambler_opt.vhd

Modified files for GBT RX 20 bits interface:

- Manual_Frame_Alignment.vhd
- Modulo_20_Counter.vhd
- Right_Shifter_19b.vhd
- Write_RX_DP_RAM.vhd
- Demux_20_to_120bits_opt.vhd
- Pattern_Search_20bit_opt.vhd

The placement of the GBT RX module is between the GTX RX and the UL. Each interface of the GBT RX belongs to a different clock domain so this module is clocked by two reference clocks (RWCLK and RFCLK).

- The part, which interfaces with the GXT RX, presents a 20 bits data path clocked by RWCLK.
- The part that interfaces with the UL presents an 84 bits data path clocked by RFCLK.

The GBT RX module is responsible of providing the number of phase shifts to BSC and the header_flag signal assertion, which is used by PA as reference edge to fit with.

- The **number of phase shifts** is determined by counting the number of bit slips that the GBT RX frame aligner needs to perform until the heather of the GBT frame is aligned correctly.
- The **header_flag** is asserted when the header of the GBT frame is detected. This header is placed at the beginning of the 120 bits frame so it is an indicator of the

rising edge of the TFCLK which is synchronous and deterministic with respect to the 40 MHz crystal oscillator of the BE board.

3.3.7. GTX TX on FE board clocking setup

Due to the requirements of his project (clock recovery and latency determinism), the TX PLL of the GTX TX on the FE board needs a reference clock synchronous to the 40 MHz crystal oscillator of the BE board. It is important to mention that although the *Xilinx Synthesizer Tool* (XST) would allow the utilisation of RWCLK to directly clock the TX PLL on the FE board, this configuration is not recommended by Xilinx. The reason is that this reference clock must be extremely clean [34] (otherwise, bit errors may appear on the data frame).

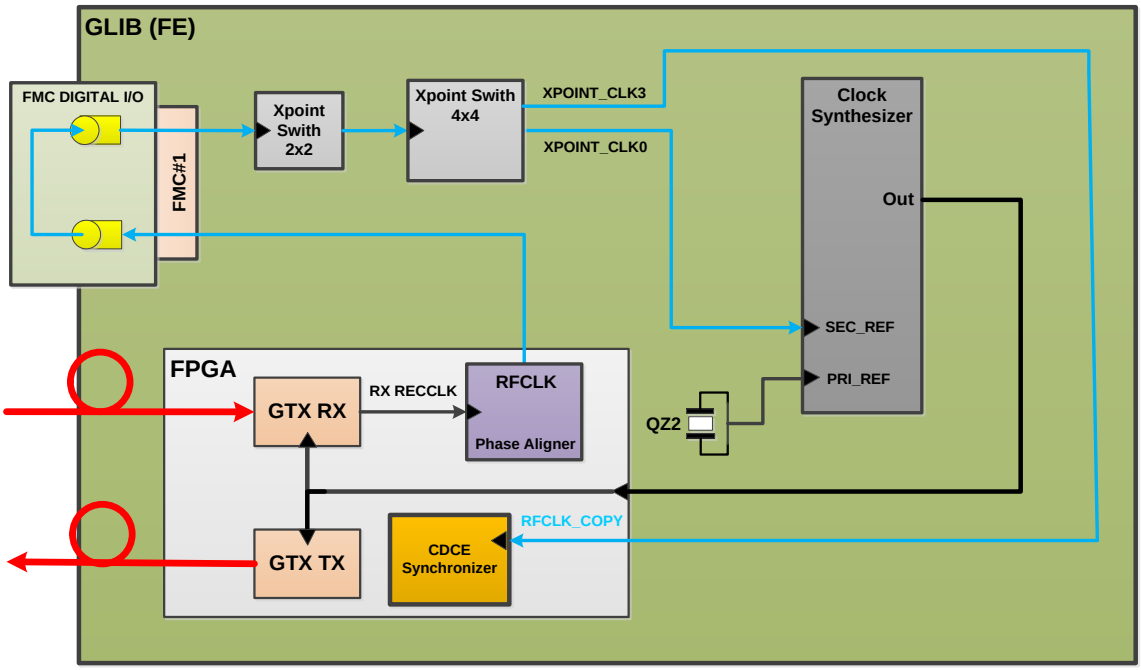


Figure 3-66: RFCLK forwarding block diagram.

The workaround adopted in this project is to feed the secondary reference clock input of the CS with RFCLK and perform a clock swap once RFCLK is stable (RFCLK has been selected for this purpose instead of RWCLK because the CS has been configured for a 40 MHz reference clock). The route followed on the clock forwarding is explained next:

1. RFCLK is forwarded out from the FPGA through an OBUFDS Xilinx primitive (configured with LVDS levels) towards the FMC1_CLK0_C2M pin pair of the FMC#1 socket.
2. The forwarded clock is looped back and guided towards the 2x2 crosspoint switch by the utilisation of an FMC card (FMC DIO 16CH TTL A for this project) which connects the FMC1_CLK0_C2M pin pair with the FMC1_CLK0_M2C pin pair.
3. Finally, RFCLK is guided through the crosspoint switch 2x2 (IC25) and crosspoint switch 4x4 (IC31) towards the secondary reference clock pin of the CS (XPOINT_CLK0) and the user logic of the FPGA (XPOINT_CLK3 → RFCLK_COPY) for clocking the secondary reference clock of the *CDCE synchronizer* module.

At this point, the secondary reference clock port of the CS is feed by RFCLK so that after the clock swap, the GTX TX on the FE board will be able to send data synchronously.

3.3.8. GBT Link communication checking

Pattern generator

The pattern sent by TX is an 84 bits frame that is used as payload of the GBT frame. This patter is provided by the *Pattern Generator* (PG) module (Figure 3-67). For generating the frame, this module utilizes an 84 bits counter that performs an up counting. The reference clock of his module is TFCLK so the payload data rate achieved is 3.36 Gbps (84 bits x 40 MHz).

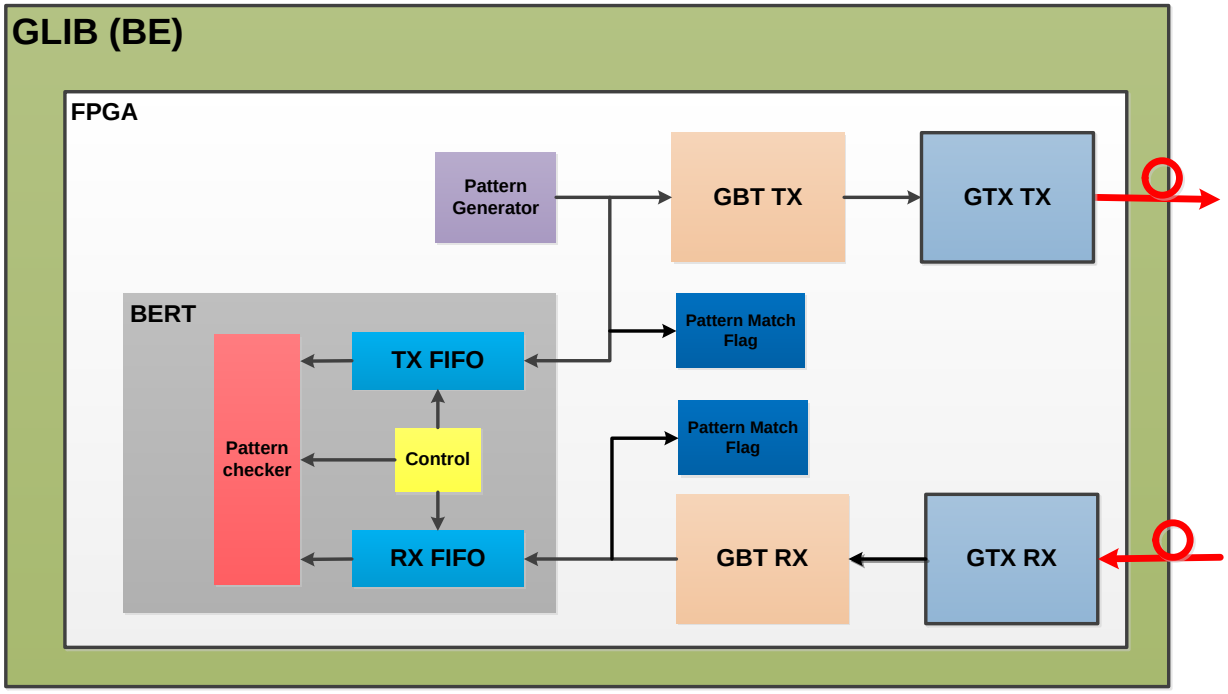


Figure 3-67: GBT Link communication checking block diagram - GLIB (BE).

Bit Error Ratio Test (BERT)

The correct transmission and reception of the data is checked by the *Bit Error Ratio Test* (BERT) (Figure 3-67). This module compares the data that comes directly from the PG with the output data of the GBT RX.

The BERT module is only instantiated on the BE board whereas on the FE board the data output of the GBT RX is connected directly to the data input of the GBT TX. This setup enables the transmission of the data from the BE board to the FE board which sends the same data again to the BE board.

The expected latency of the transmission has to be set manually for the proper functioning of the BERT. The comparison procedure is explained next. Once the sequential reset procedure of the GBT Link has finished, the TX FIFO and the RX FIFO of the BERT are enabled so data from PG is stored on the TX FIFO while data from GBT RX is stored on the RX FIFO. The data provided by GBT RX before the reception of the data from PG is not valid so must be discarded. The invalid data is discarded by reading the RX FIFO as many times as the latency of the communication. After that, the pointers of both FIFOs are placed on the position where the first data sent by PG is stored. Finally, the read procedure of both FIFOs is started at the same time and the internal Pattern Checker of the BERT starts the comparison of the data.

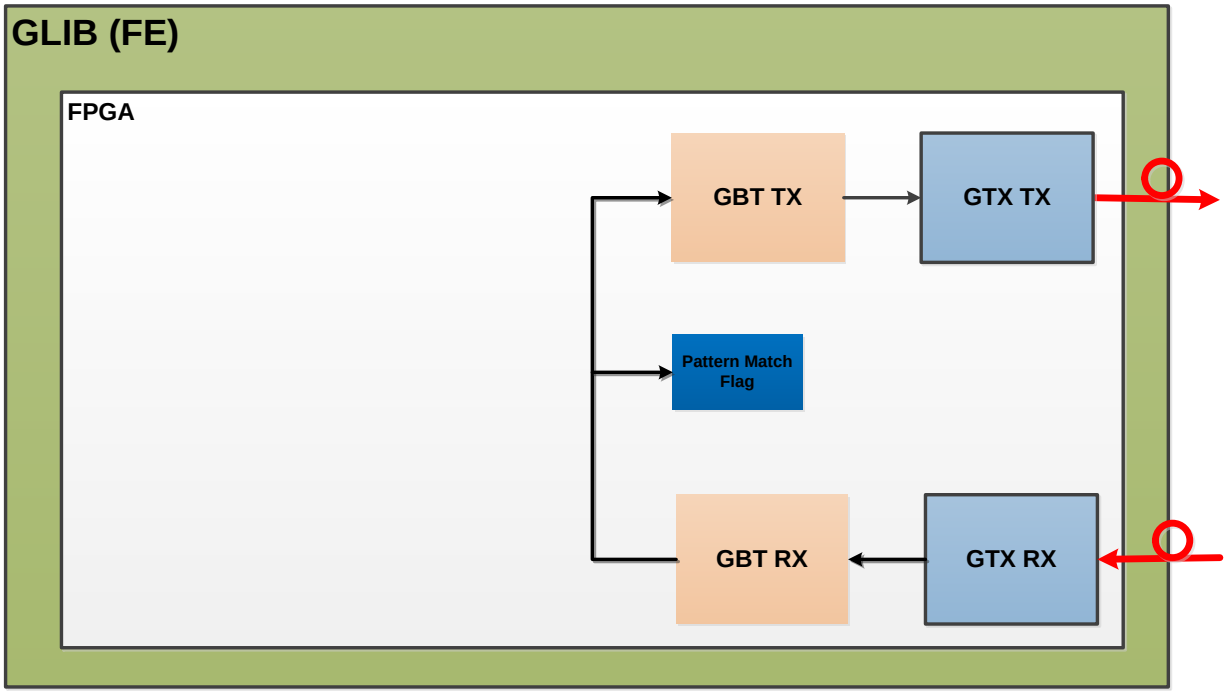


Figure 3-68: GBT Link communication checking block diagram - GLIB (FE).

The BERT module provides the number of words compared and the number of mismatches found. This values and the data send and received are monitored by using ChipScope.

Latency checking setup

The two main objectives of the communication latency checking are the quantification of the latency and the verification of whether this latency is deterministic or not

Although the latency can be checked by comparing the data sent and received using ChipScope, this method is not recommended because ChipScope is an HDL module that could interfere with the actual system.

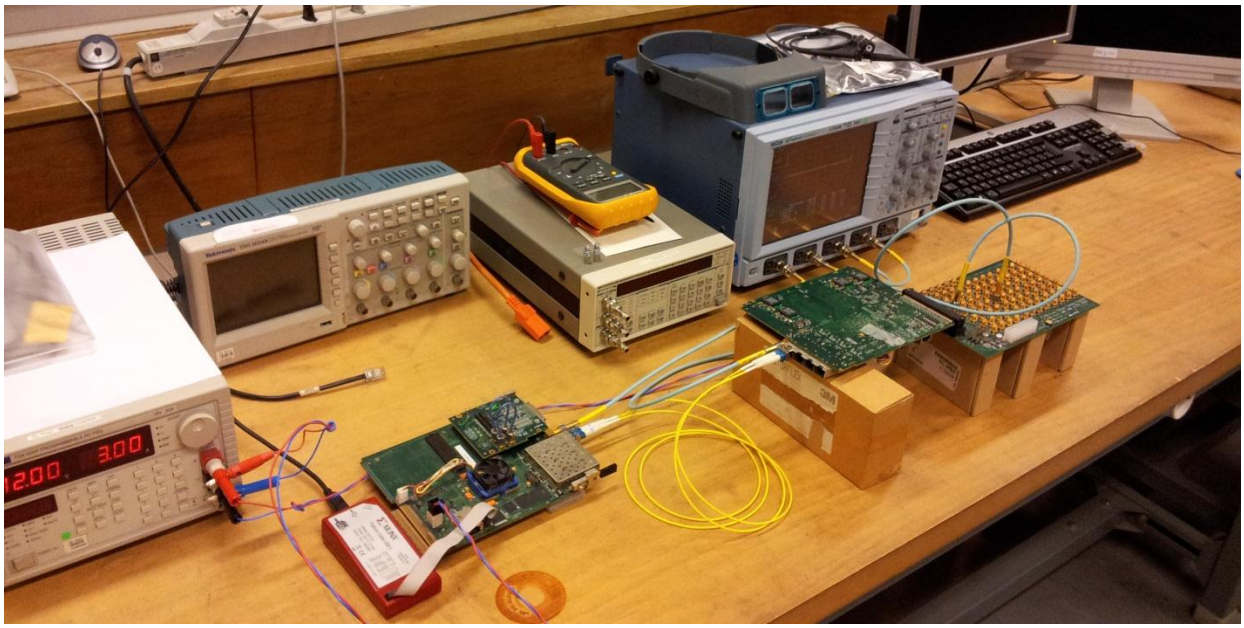


Figure 3-69: Latency checking setup.

The latency checking performed in this project consists in the utilisation of an Oscilloscope (LeCroy WaveRunner LT354M) for the measurement of the time different between three flags. These flags are asserted when a pattern is detected on the data frame (in this case when the last 8 bits of the data frame are equal to 0xFF). The module responsible of the generation of these flags is the *Pattern Match Flag* (PMF) module. It is needed one instantiation of the PMF per flag (Figure 3-67 and Figure 3-68). The first PMF is connected to the PG on the BE board. The second PMF is connected to the data output of the GBT RX on the FE board. The third PMF is connected to the data output of the GBT RX on the BE board. With this distribution it is feasible to measure the latency of each TX-RX link (BE to FE and FE to BE). To be able to perform the measurement with the oscilloscope these flags are forwarded out from

the FPGA. The two flags of the BE board are forwarded towards two SMA connectors on the SMA AMC9000 ULTRA board (Lane 12 TX+ and Lane 13 TX+) which is connected to the AMC port of the GLIB. The 40 MHz clock from the crystal oscillator of the BE board is also forwarded to the on board SMA connector (SMA1) and is used as reference point. On the FE board, the flag is forwarded to the on board SMA connector.

Test results

On the Figure 3-70: Latency checking with the oscilloscope are showed the crystal oscillator reference clock (QZ2) of the BE board (yellow) and the flag of the pattern generator, FE RX and BE RX in red, blue and green respectively.

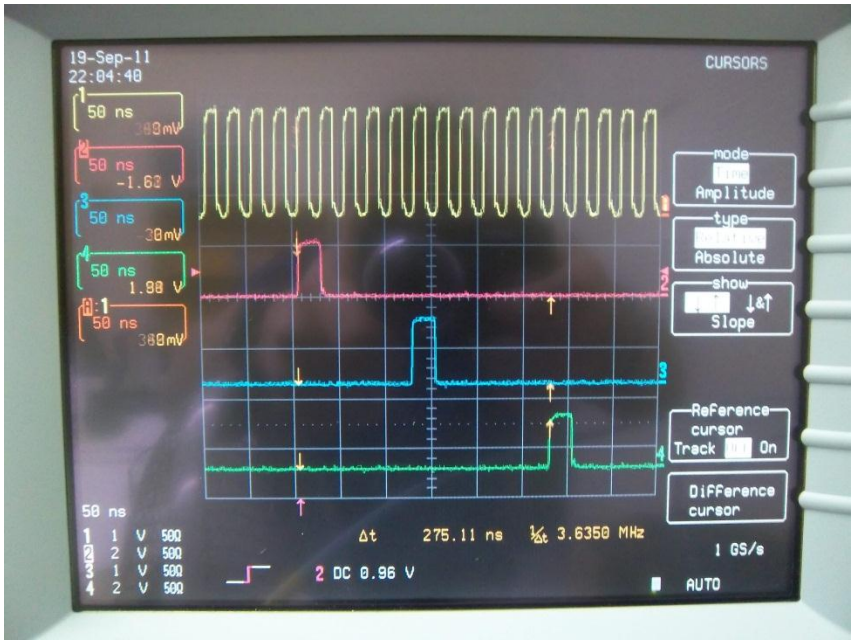


Figure 3-70: Latency checking with the oscilloscope.

These flag signals have been checked with the oscilloscope after series of resets and even power ups following a manual procedure. In all cases the result of the test in terms of integer numbers of BE crystal oscillator clock cycles were identical, measuring five cycles of latency between the PG and the FE RX flags while the latency between the FE RX and the BE RX flags was six cycles. These values are very close to the results presented in [28]. The difference is due to the latency measured in this test is not only of the GBT Link. It also includes the latency added by: the internal path from the flag generator to the IOB of the FPGA, the path through the IOB, the coaxial cable that

connects the board with the oscilloscope and depending on the case, either the path which connects the pin of the FPGA with the on board SMA connector (crystal oscillator and FE RX flags) or the path which connects the FPGA pin with the SMA connector on the AMC SMA ULTRA9000 board (PG and BE RX flags). Therefore, the actual latency of the GBT Link would be even shorter than the latency shown by the oscilloscope. The quantification of the latencies added by the above mentioned elements have not been calculated because the values measured by the oscilloscope were completely suitable for the purposes of the GLIB. However, it is important to mention that the latency of the link may change due to the length of the optical fibre cables (in this case both cables were 1 m length).

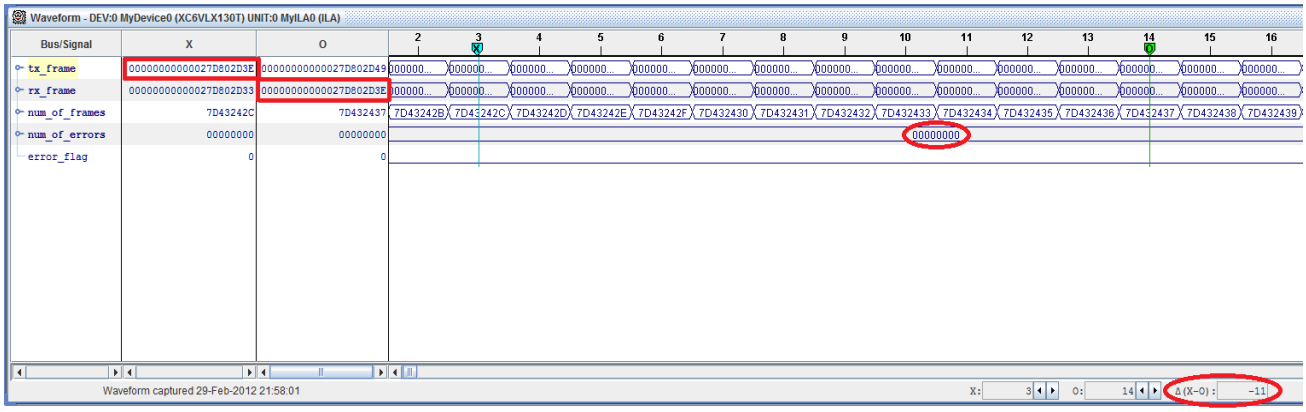


Figure 3-71: BERT signals taped with ChipScope.

On the other hand, the integrity of the data sent through the GBT Link has been confirmed by checking the values provided by the BERT module using ChipScope. In all cases, no errors were found in the data.

Table 3-8: GBT Link latency.

	Standard configuration	**GLIB configuration**
GBT Link latency (TX and RX) [28] (TFCLK cycles)	21 cycles (±1 cycle)	5 cycles
Latency determinism	One clock cycle uncertainty	Deterministic

Due to the results of these test performed to the GBT Link, we can say that the communication between the BE-FE boards has been a conspicuously success.

3.4. Conclusion of the BE/FE link simulation

Since deterministic and low latency in communications is not a usual requirement, the commercial high speed transceivers do not provide these features. However, since both features are necessary for the GBT communication protocol because of the special requirements of the HEP experiments, our team was focused on overcoming these difficulties. We have managed to meet our requirements by configuring the GTX transceiver of the Virtex-6 in a special way that bypasses all stages that introduce uncertainty in the latency. With the project presented in this chapter, we have verified experimentally that our goals have been reached.

Chapter 4 - Conclusions and future plans

4.1. Conclusions

The new GBT link to be used on the SLHC requires a new generation of electronic devices in order to comply the new specifications. Together with the development of the GBT chipset which covers the requirements of the GBT link on the FE side, it is necessary to provide a hardware platform for the GBT link on the BE side. For that reason, we have designed and built the **Gigabit Link Interface Board (GLIB)**, an FPGA-based platform that is under development since September 2010. The GLIB can be used both for the evaluation of optical links in the laboratory as well as for triggering and/or data acquisition systems in beam or irradiation tests of detector modules in high energy physics experiments. The board has been conceived in the form of a double wide *Advanced Mezzanine Card* (AMC) that operates either stand-alone or residing inside a micro *Telecommunications Computing Architecture* (μTCA) crate. The FPGA of the GLIB features *Multi-Gigabit Transceivers* (MGT) operating at rates of up to 6.5 Gbps. This performance matches comfortably the specifications of the GBT and Versatile Link projects with its targeted data rate of 4.8 Gbps. Other major components of the GLIB are the cage for *Small Form Factor Pluggable Plus* (SFP+) optical modules, the AMC connector, the FMC sockets, the SRAM devices, the FLASH, the clock synthesizer, the socket for the *Module Management Controller* (MMC) mezzanine card and the 1000Base-T interface. The GLIB I/O capability can be further enhanced with two *FPGA Mezzanine Cards* (FMCs) connected to the FMC sockets mentioned above. This gives users the flexibility to adapt the GLIB interface to their system, by for instance adding connectivity to the TTC network at the BE, or connecting to e-links at the FE. It is also worth to mention the versatility of the clocking scheme on GLIB, which is an essential part of the system. Regarding the GLIB firmware, the system is implemented around a simple *IP-based control protocol* (IPbus) designed for controlling xTCA-based hardware over Gigabit Ethernet. All necessary interfaces to the external hardware are instantiated, including I^2C communication with the on-board temperature sensors and the serial memory, SPI communication with the clock synthesizer, parallel memory interface with the two SRAMs operating at 160 MHz as well as FLASH memory interface with multiboot and fallback features. Additionally, the firmware instantiates a number of GBT *Hardware Description Language* (HDL) modules that together with the associated MGTs are used for interfacing with GBT protocol based systems. It is

important to mention that the firmware architecture is designed in such way that allows the user to add his logic without the need of editing anything outside the user logic block. Concerning the software, at this point of the development, the communication between the GLIB board and the computer of the user is based on PyChips, a Hardware Access Library (HAL) written in Python, which provides functions to enable transactions with the hardware attached to the IPbus, which is included in the GLIB firmware. The functions provided by PyChips are called by scripts written in Python as well and executed on the Command Prompt of a Windows operating system (7, Vista, XP etc.). Apart from the standard IPbus software framework distribution, we are developing a stand-alone JAVA-based Graphical Users Interface (GUI). Within the frame of the GLIB project, some auxiliary boards (mainly FMCs) are currently under development. One of them is the **Timing, Trigger and Control (TTC) FMC** developed based on a commercial Clock & Data Recovery (CDR) integrated circuit for interfacing the GLIB with the TTC system used currently in the LHC experiments. Another board under development is the **VTRx FMC**, a mezzanine card that interfaces the various flavours of VTRx with the GLIB system. A third system which is currently under development is an **AMC to PCI Express x4 adapter**, that provides a high bandwidth interconnection between a stand-alone GLIB and a PC. The first 2 GLIB prototypes have been built in early 2011. Already from the first prototypes, most of the board's features were functioning without problems. The few issues that have been identified, they have been understood and solved in the second version of the board that has been assembled in summer 2011. This version has been tested extensively and its bench-top performance has been found satisfactory. Currently, 6 more pieces of the GLIB, in this case v2, has been produced so with the two GLIB v1 a total of 8 GLIB prototypes have been released. Three of these new boards have been delivered to some selected beta users. Together with the hardware, we also deliver the associated firmware and software so from the very first moment the user can be focused on the development of the actual user logic because all features of the GLIB are already available. Moreover, with this hardware/firmware/software mixed scheme, the system becomes more reliable due to the extensive tests that have been performed to the delivered versions, the support to the user is more effective because the critical parts of the system do not need to be modified and the system can be easily updated when needed.

4.2. Future plans

First of all, it is important to mention that the development and testing of the GLIB continues beyond my stay at CERN as Technical Student. For the coming months, the GLIB team has several courses of action to deal with before going into production. The output synchronization problem of the CS is expected to be solved by detecting the error utilising the internal resources of the FPGA. The PCIe firmware and the in-crate operation of the GLIB are also planned for these months. The production version of the board is expected to be available in July 2012. Within the frame of this project, some additional auxiliary boards (mainly FMCs) are currently under development. The second prototype of the Optical CDR FMC is already available as well as the first prototype of the Versatile Link FMC. The GLIB team is also exploring new communication interfaces (e.g. 10Gbe). For the latest information concerning the GLIB project, please visit the project's web site [35].

4.3. The GLIB team

- P. Vichoudis[(a)]
- M. Barros Marin[(a,b)]
- S. Baron[(a)]
- V. Bobillier[(a)]
- S. Haas[(a)]
- M. Hansen[(a)]
- M. Joos[(a)]
- L. Lobato Pardavila[(a,c)]
- P. Petit[(a)]
- F. Vasey[(a)]

[(a)] CERN 1211 Geneva 23, Switzerland

[(b)] Escuela Técnica Superior de Ingenieros Industriales
Universidad Nacional de Educación a Distancia (UNED)
28040 Madrid, Spain

[(c)] Escuela Superior de Ingeniería Informática
University of Vigo
32004 Ourense, Spain

Bibliography

B. Schmidt, "A glance at LHC Detector Systems," CERN, 2011.

B. Petersen, "Trigger and Data Acquisition," CERN, 2011.

C. Foudas et al., "The CMS Tracker Readout Front End Driver," CERN, 2004.

C. Soos et al., "FPGA-based Bit-Error-Ratio Tester for SEU-hardened Optical Links," CERN, 2009.

CERN, "CMS Tracker Optical Control Link Specification," CERN, 2004.

D. Günther, "LHC Detectors," ETH Institute for Particle Physics, 2007.

F. Vasey, "Fast Readout Links," CERN, 2011.

Faster Technology, "FM-S14 User Manual," 2011.

Fa-Xin Yu et al., "Overview of Radiation Hardening Techniques for IC Design," School of Aeronautics and Astronautics, Zhejiang University, 2010.

FIBER OPTICS. [Online]. Available: http://www.arcelect.com/fibercable.htm. [Accessed 20 01 2012].

G. Rogers and T. Pauly, "Monitoring of the ATLAS Level-1 Trigger," CERN, 2005.

H. Spieler, "Front-End Electronics and Signal Processing," Physics Division (Lawrence Berkeley National Laboratory), 2002

J. Troska et al., "Optical Readout and Control Systems for the CMS Tracker," CERN, 2003.

N. Garelli, "Upgrade of Trigger and Data Acquisition Systems for the LHC Experiments," CERN, 2011.

NASA, "SPACE RADIATION EFFECTS ON ELECTRONIC COMPONENTS IN LOW-EARTH ORBIT," NASA, 1996.

P. Aspell, "Conception et mise au point de l´´electronique," CERN, 2001.

P. Mutti, "Fast Data Acquisition, Signal Processing and Its Integration Within Instrument Control," ill Neutrons for Science, 2011.

P. P. M. Jansweijer and H.Z. Peek, "Measuring propagation delay over a 1.25 Gbps bidirectional data link," NIKHEF National Institute for Subatomic Physics, 2010.

P. Van Esch, "Overview of detector front end electronics," EIROforum 2011, 2011.

R. Giordano and A. Aloisio, "Fixed-Latency, Multi-Gigabit Serial Links With Xilinx FPGAs," IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 58, NO. 1, 2011.

S. Baron, "Phase and latency reproducibility in COTS - Phase and latency solutions for FPGAs and Integrated PLLs," CERN, 2009.

STx Slilicon Turnkey Express, "SMA-AMC Ultra 9000 Datasheet".

Timbercon. [Online]. Available: http://www.timbercon.com/copper-cable-vs-fiber-optics/. [Accessed 20 01 2012].

W. Snoeys, "TOTEM CSC electronics," CERN, 2006.

Xilinx, "ChipScope Integrated Bit Error Ratio Test (IBERT) for Virtex-6 GTX - DS732,", 2009.

–, "Platform Cable USB II - DS593 (v1.2.1)," 2011.

–, "Virtex-6 FPGA Clocking Resources User Guide - UG362 (v2.0)," Xilinx, 2011.

–, "Virtex-6 FPGA Memory Resources User Guide - UG363 (v1.6)," 2011.

–, "Virtex-6 FPGA SelectIO Resources User Guide - UG361 (v1.3)," , 2010.

References

[1] CERN, "CERN public web page," [Online]. Available: http://www.cern.ch. [Accessed 08 01 2012].

[2] CERN, "CERN faq, LHC the guide," 2009.

[3] Wikipedia. [Online]. Available: http://www.wikipedia.org. [Accessed 01 2012].

[4] LHC public web page. [Online]. Available: http://public.web.cern.ch/public/en/LHC/LHC-en.html. [Accessed 11 01 2012].

[5] ATLAS public web page. [Online]. Available: http://atlas.ch/. [Accessed 10 01 2012].

[6] CMS public web page. [Online]. Available: http://cms.web.cern.ch/. [Accessed 09 01 2012].

[7] ALICE public web page. [Online]. Available: http://aliceinfo.cern.ch/Public/Welcome.html. [Accessed 09 01 2012].

[8] LHCb public web page. [Online]. Available: http://lhcb-public.web.cern.ch/lhcb-public/. [Accessed 11 01 2012].

[9] TOTEM public web page. [Online]. Available: http://totem-experiment.web.cern.ch/totem-experiment/. [Accessed 12 01 2012].

[10] LHCf public web page. [Online]. Available: http://public.web.cern.ch/public/en/lhc/LHCf-en.html. [Accessed 11 01 2012].

[11] WLCG public web page. [Online]. Available: http://lcg.web.cern.ch/lcg/public/default.htm. [Accessed 20 02 2012].

[12] SLHC public web page. [Online]. Available: http://slhcpp.web.cern.ch/SLHCPP/. [Accessed 18 01 2012].

[13] P. Moreira, A. Marchioro and K. Kloukinas, "The GBT, a Proposed Architecture for Multi-Gb/s Data Transmission in High Energy Physics," CERN, 2007.

[14] P. Vichoudis, M. Barros Marin, S. Baron, V. Bobillier, S. Haas, M. Hansen, M. Joos, L. Lobato Pardavila and F. Vasey, "First results with the Gigabit Link Interface Board (GLIB)," CERN, 2012.

[15] Code Archive for Cern Trigger UpgradeS (CACTUS), "PyChips," [Online]. Available: https://cactus.hepforge.org/index.php. [Accessed 16 02 2012].

[16] Python public web page. [Online]. Available: http://python.org/. [Accessed 17 02 2012].

[17] FOX ELECTRONICS, "FXO-HC73 SERIES Datasheet," FOX ELECTRONICS, 2008.

[18] Texas Instruments, "SN65LVDS1 Datasheet," 2008.

[19] Texas Instruments, "SN65LVDT122 Datasheet," 2004.

[20] Texas Instruments, "SN65LVDT125 Datasheet," 2004.

[21] Texas Instruments, "CDCE62005 Datasheet," 2011.

[22] Xilinx, "Virtex-6 Family Overview - DS150 (v2.4)," 2012.

[23] Xilinx, "Xilinx," [Online]. Available: http://www.xilinx.com/. [Accessed 02 2012].

[24] Xilinx, "Virtex-6 FPGA GTX Transceivers User Guide - UG366 (v2.6)," Xilinx, 2011.

[25] Apache SVN. [Online]. Available: http://subversion.apache.org/. [Accessed 11 02 2012].

[26] GBT Starter Kit web page. [Online]. Available: https://espace.cern.ch/GBT-Project/GBT-FPGA/default.aspx. [Accessed 16 01 2012].

[27] Altera public web page. [Online]. Available: http://www.altera.com/. [Accessed 09 01 2012].

[28] S. MUSCHTER, C. BOHM, S. BARON, J.P. CACHEMICHE and C. SOOS,

"Optimizing latency in Xilinx FPGA Implementations of the GBT," Stockholm University - CERN, 2010.

[29] MERGE OPTICS, "TRX10GDP0310 Datasheet".

[30] FMC DIO 16CH TTL A web page. [Online]. Available: http://www.ohwr.org/projects/fmc-dio-16chttla/wiki. [Accessed 23 01 2012].

[31] Open Hardware Repository (OHWR). [Online]. Available: http://www.ohwr.org/. [Accessed 26 02 2012].

[32] STx Slilicon Turnkey Express, "SMA-AMC Ultra 9000 Datasheet".

[33] TTi - TSX-P Series. [Online]. Available: http://www.tti-test.com/products-tti/psu/tsxp-series.htm. [Accessed 14 02 2012].

[34] Xilinx, "AR#38506," 2011.

Curriculum Vitae

Name: MANOEL BARROS MARIN
Nationality: Spanish
Address: Apartado de Correos Nº 42 (36800) Redondela, Pontevedra (Spain)
Telephone: (+34)625411130 / (+34)986408004 / (+41)762629823
Email: m.barros@ieee.org / manoel.barros.marin@gmail.com

EDUCATION

- 2007- 2012: B.Eng. in Industrial Electronics (Ingeniería técnica industrial, electrónica industrial) from the Universidad Nacional de Educación a Distancia (UNED), Spain.
- 2006-2007: "University Entrance" course ("Curso de acceso para mayores de 25 años"), 1 year.

RELEVANT WORK EXPERIENCE

- 2011-2012: Technical Student at CERN.

AREAS OF EXPERTISE AND SKILLS

- Electronic System Design (from concept, through to schematic and layout).
- Hardware Description Languages (VHDL/Verilog) for Field Programmable Gate Arrays (FPGAs).
- Electronic Design Automation (EDA) tools.
- Programming for microcontrollers (Assembly and Embedded C).
- Computer Programming Languages (Java, Python, C/C++).
- LabVIEW graphical development platform (G).
- Implementation and characterization of serial optical/electrical links.
- Computer software and hardware (operating systems, productivity suites, image editors, web design applications, PC hardware setup and configuration).

AWARDS AND DISTINCTIONS

- 2010: Prize Winner of "The Best Student of the Industrial Engineering Faculty" (academic season 2009-2010)".
- 2009: Nominated for Prize Winner of "The Best Student of the Industrial Engineering Faculty (academic season 2008-2009)".
- 2009: Grant from the "Foundation Manuel Ventura Figueroa" for the academic season 2008-2009.
- 2008: Grant from the "Foundation Manuel Ventura Figueroa" for the academic season 2007-2008.

LANGUAGES

- Spanish (mother tongue)
- Galician (mother tongue)
- English (fluent)
- Portuguese (fluent)

RELATED COURSES AND SEMINARS

- 2012: Cambridge First Certificate in English (FCE).
- 2012: "INTEL ATOM communicating with XILINX FPGA through PCIe (Intel, Microsoft, Xilinx) à Hands On Labs" held by Silica-Xilinx-Microsoft-Intel at CERN.
- 2011: "2nd EIROforum School of instrumentation 2011" at EPN Science Campus.

- 2011: Course "Expert VHDL for FPGA design" of 40 hours held by Doulos at CERN.
- 2011: "Practical introduction to PCI Express with FPGAs" seminar held by Michal Husejko and John Evans (CERN) at CERN.
- 2011: "Xilinx FPGA Serial IO transceiver seminar" held by Silica-Xilinx at CERN.
- 2011: "LX9 Speedway Seminar" held by Silica-Xilinx at CERN.
- 2011: "Reconfigurable FPGAs in radioactive environments" seminar held by Massimo Violante and Luca Sterpone (Politecnico di Torino) at CERN.
- 2011: "Integrated Circuits Packaging PH-ESE" seminar held by David Porret at CERN.
- 2011: "Soft-error-rate prediction for programmable circuits methodology tools and studied cases" seminar held by Raoul Velazco (TIMA laboratory) at CERN.
- 2011: "Design Challenges and Future Directions of Low-power High-Speed IOs" seminar held by Thomas Toifl (IBM) at CERN.
- 2011: "Overview of Electronic System Level Design (Catapult C)" seminar held by Spyridon Georgakakis (Technological Educational Institute of Piraeus) and John Evans (CERN) at CERN.
- 2010: Course "Design and implementation of signal processing systems through FPGA" of 30 hours at the Vigo University.
- 2010: Course "Embedded systems in FPGA" of 30 hours at the Vigo University.
- 2010: Course "Process Control and Matlab/Simulink" of 35 hours at the UNED University.
- 2010: Course "Curso práctico de administrador y desarrollador de bases de datos" (SQL database) of 35 hours at the Vigo University.
- 2010: "IV conference about technologies and solutions for the Industrial Automation" (JAI 2010) at the Vigo University.
- 2009: Course "Specialization in FPGA" of 35 hours at the University of UNED.
- 2008: "I Jornadas de control automático y robótica con herramientas Open Source" at the Santiago de Compostela University.
- 2006: Course "Interpretation of schematics" of 60 hours held by "Fundación laboral de la contrucción" and "Consellería de traballo, direción xeral de formación e colocación. (Xunta de Galicia)".
- 2006: Course "Solar thermal energy" of 40 hours held by "Fundación laboral de la contrucción" and "Consellería de traballo, direción xeral de formación e colocación. (Xunta de Galicia)".
- 2002-2003: "Electronics for Vehicles" course ("Electricidad/Electrónica de Vehículos"), 675 Hours.

PUBLICATIONS AND PAPERS

- P. Vichoudis, M. Barros Marin, S. Baron, V. Bobillier, S. Haas, M. Hansen, M. Joos, L. Lobato Pardavila, F. Vasey "First results with the Gigabit Link Interface Board (GLIB)", published in IOP Journal of Instrumentation, December 2011, JINST 6 C12060

MEMBERSHIPS

- Institute of Electrical and Electronics Engineers (IEEE).

INTERESTS

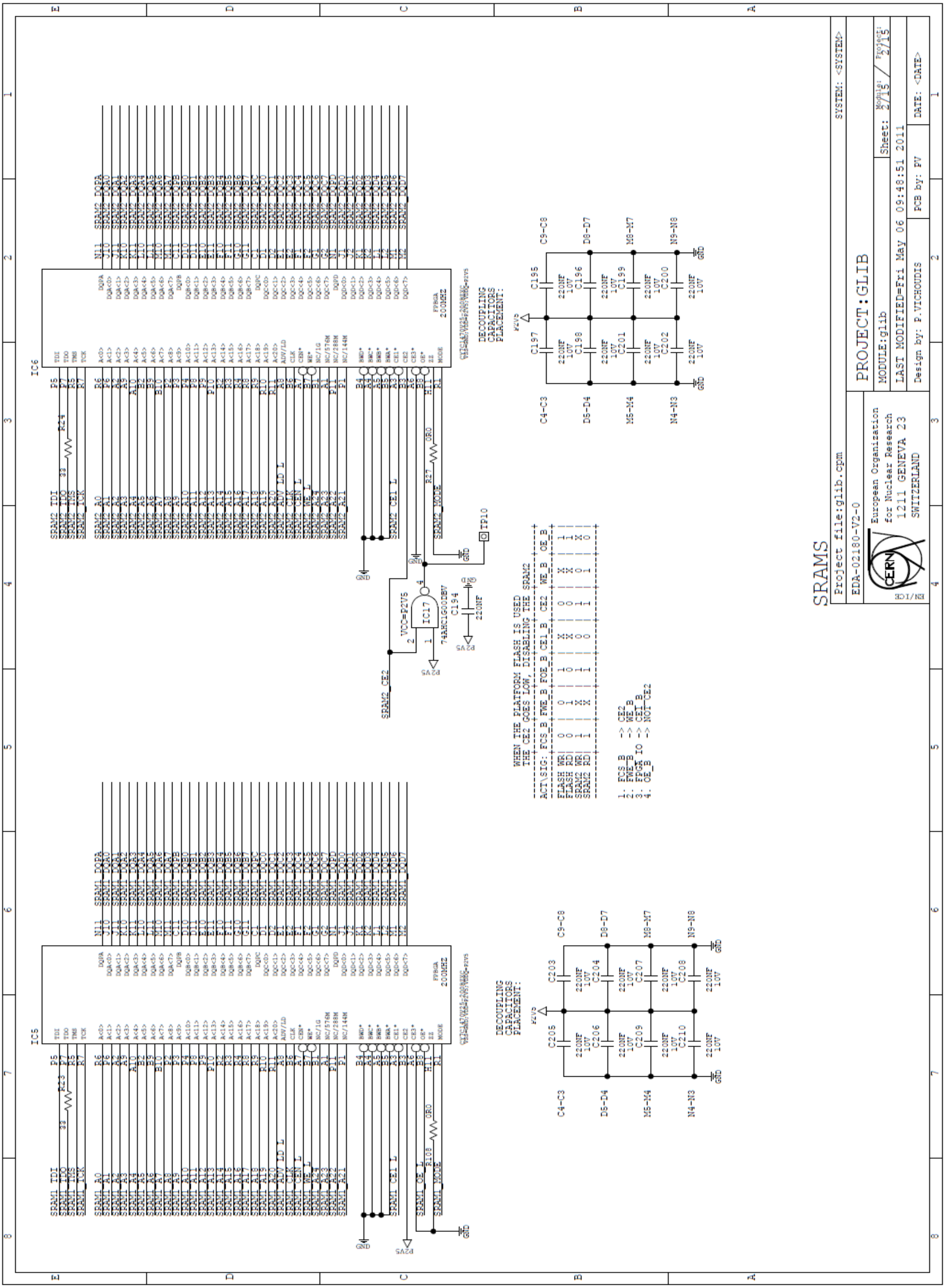
- Sports: Biking, snowboarding, kite-surfing, fitness
- Literature

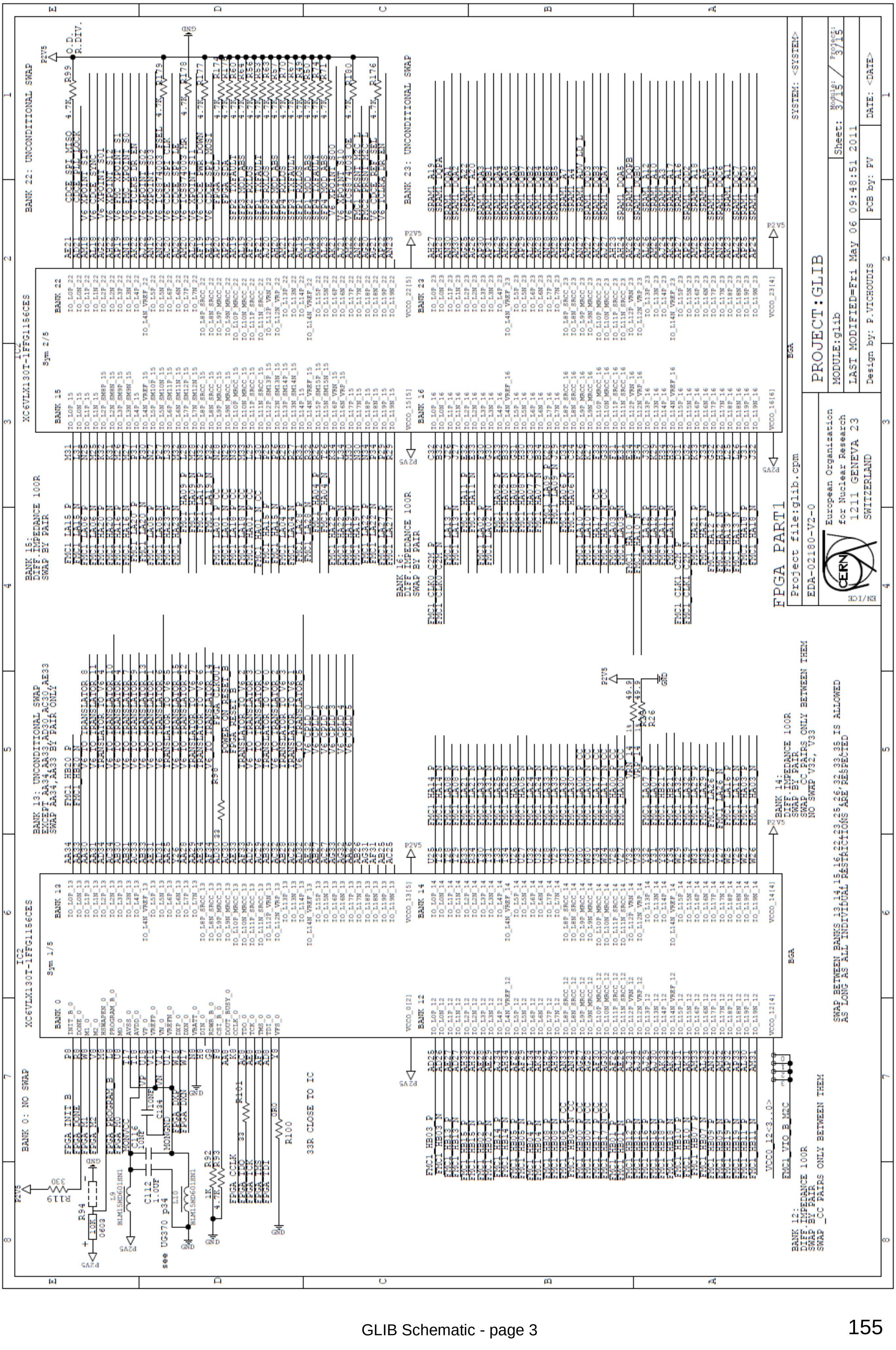
OTHER INFORMATION

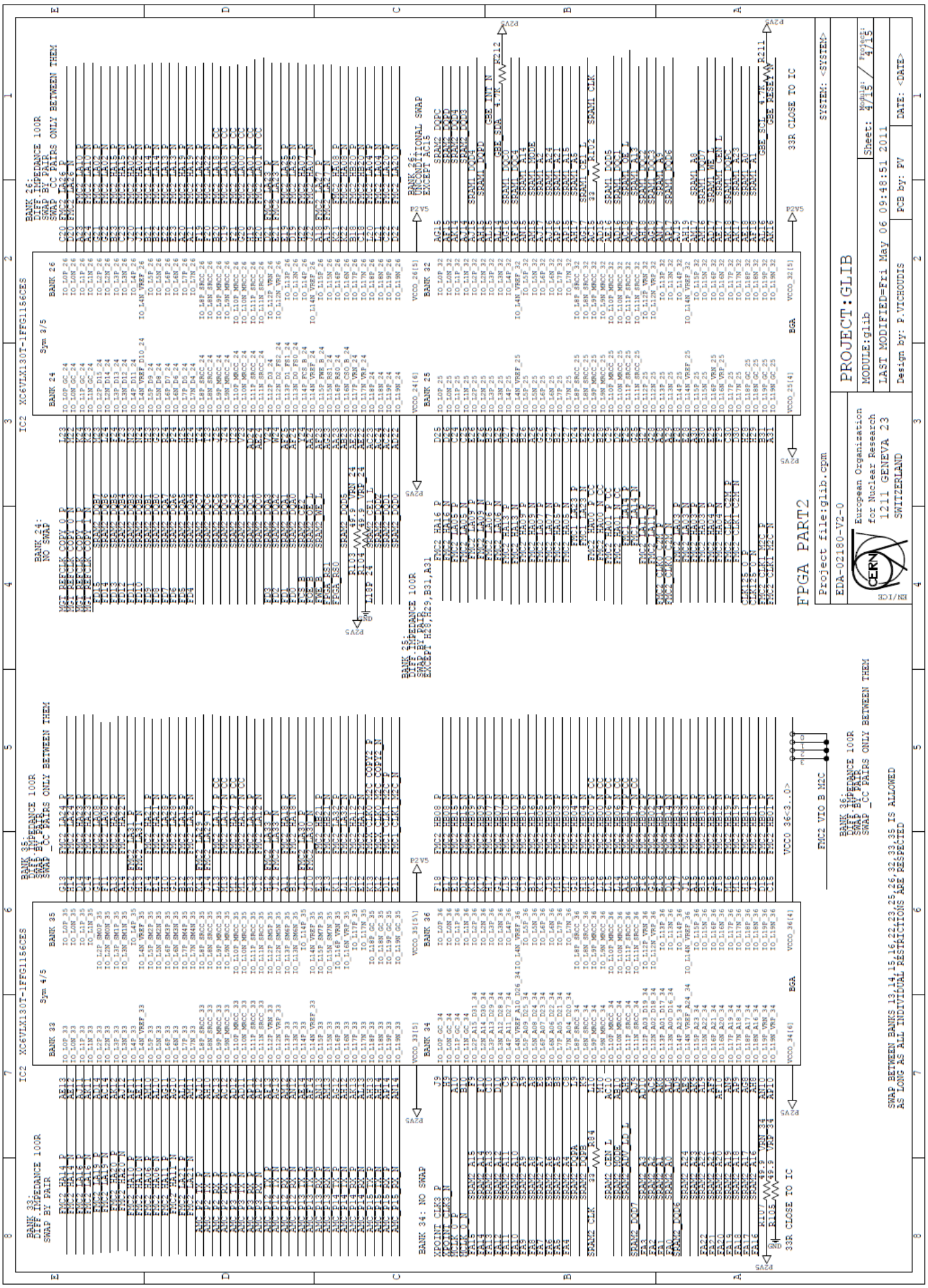
- Volunteer at the foundation "Proxecto Home Galicia" from March 2006 until January 2011.

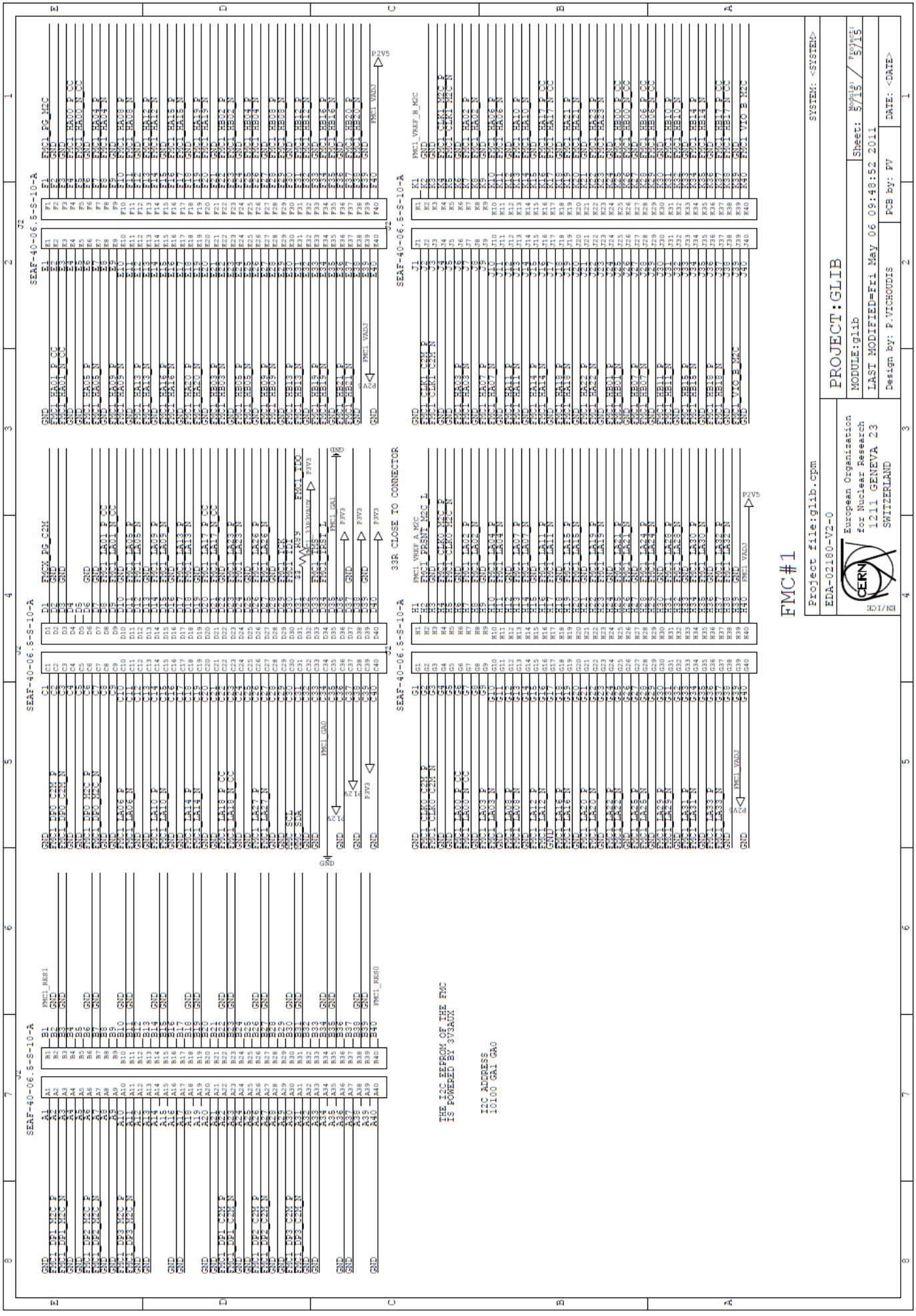
Appendix A - GLIB Schematic

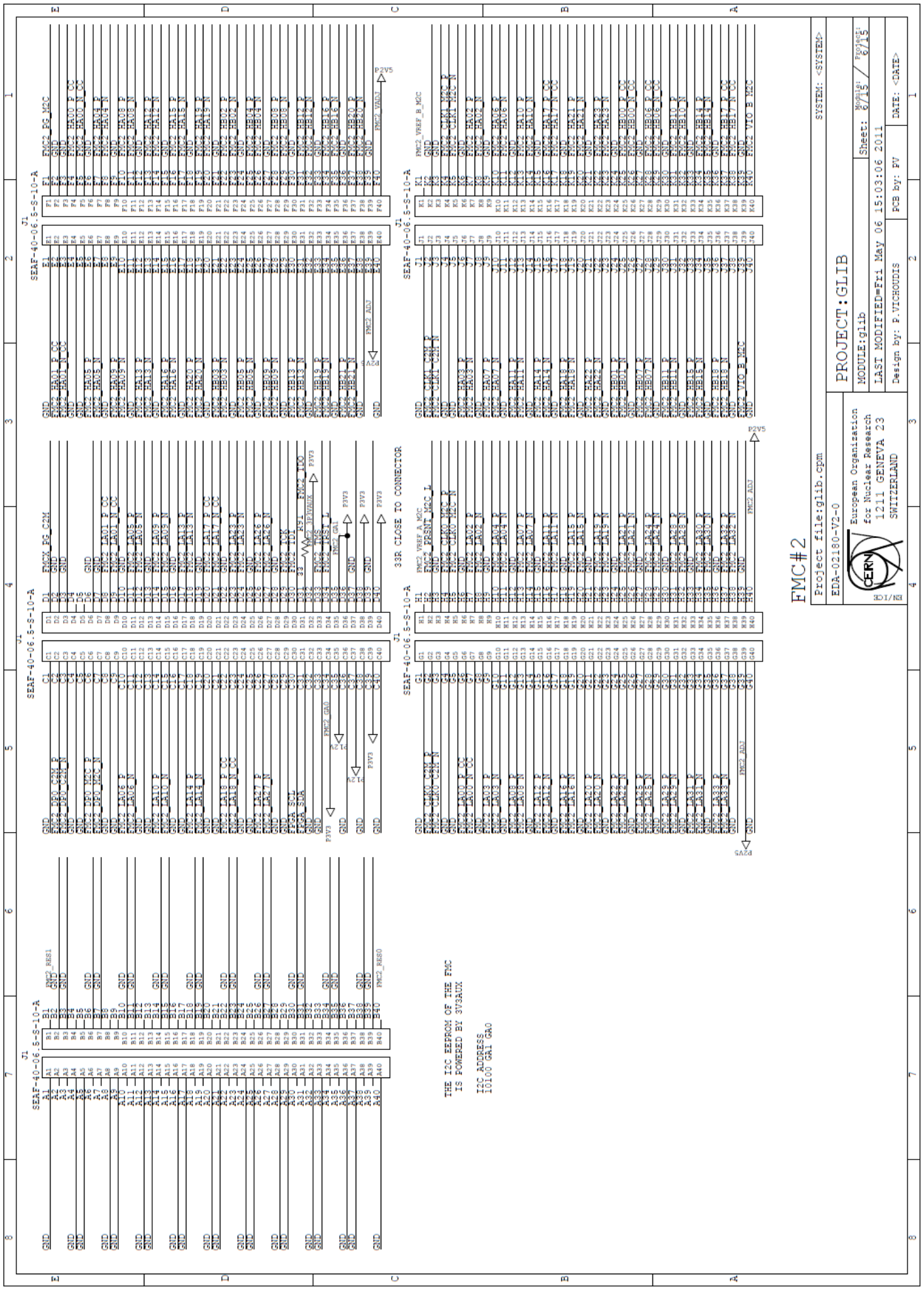
Project file:glib.cpm		SYSTEM: <SYSTEM>
EDA-02180-V2-0	PROJECT:GLIB	
EN/ICE CERN European Organization for Nuclear Research 1211 GENEVA 23 SWITZERLAND	MODULE:glib	Sheet: Module: 1/15 / Project: 1/15
	LAST MODIFIED=Tue Apr 26 17:33:29 2011	
	Design by: P.VICHOUDIS / PCB by: PV	DATE: <DATE>

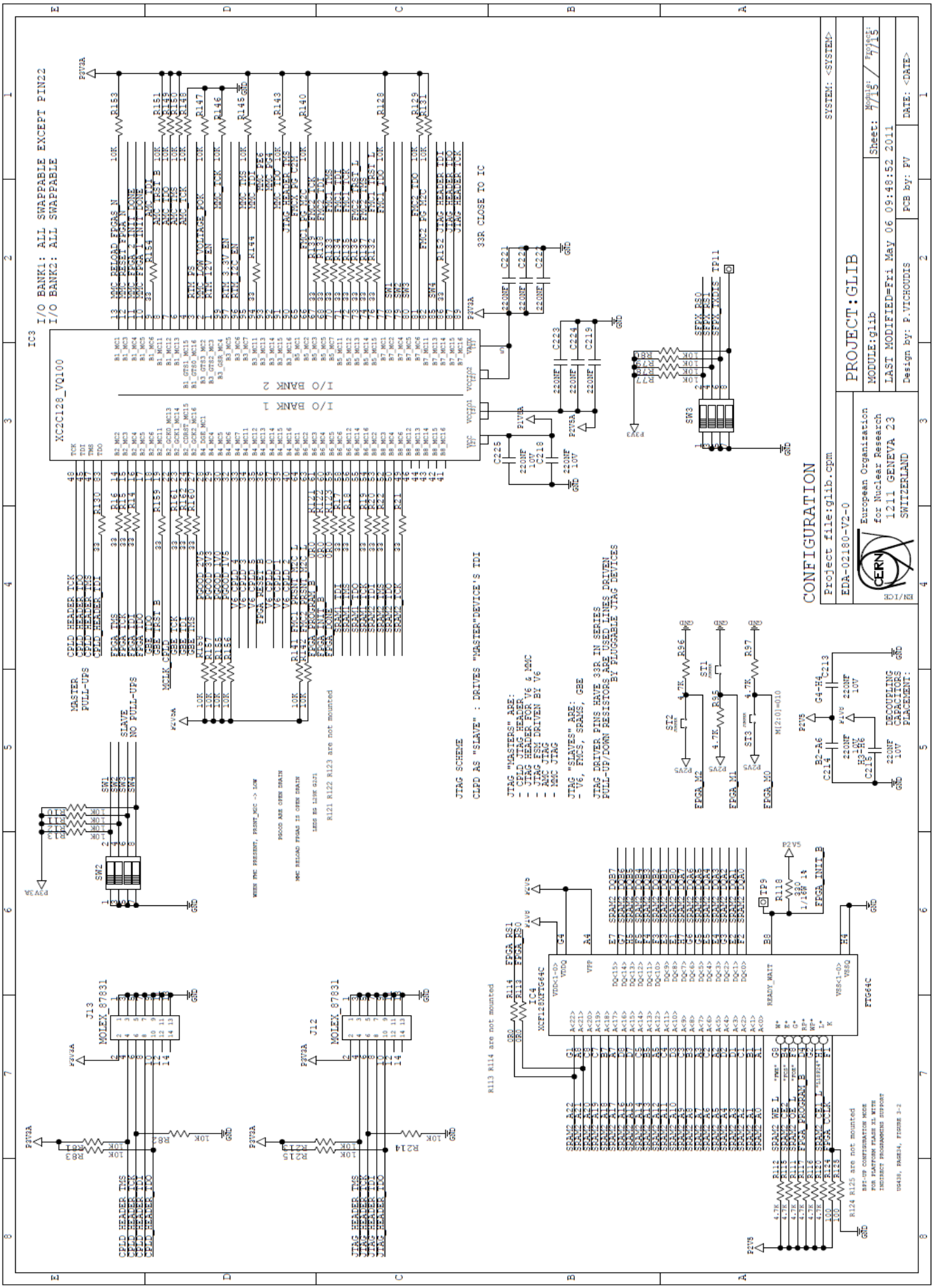

SRAMS
Project file:glib.cpm
EDA-02180-V2-0
European Organization
for Nuclear Research
1211 GENEVA 23
SWITZERLAND
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:51 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Sheet: 2/15
DECOUPLING CAPACITORS PLACEMENT:
WHEN THE PLATFORM FLASH IS USED THE CE2 GOES LOW, DISABLING THE SRAM2

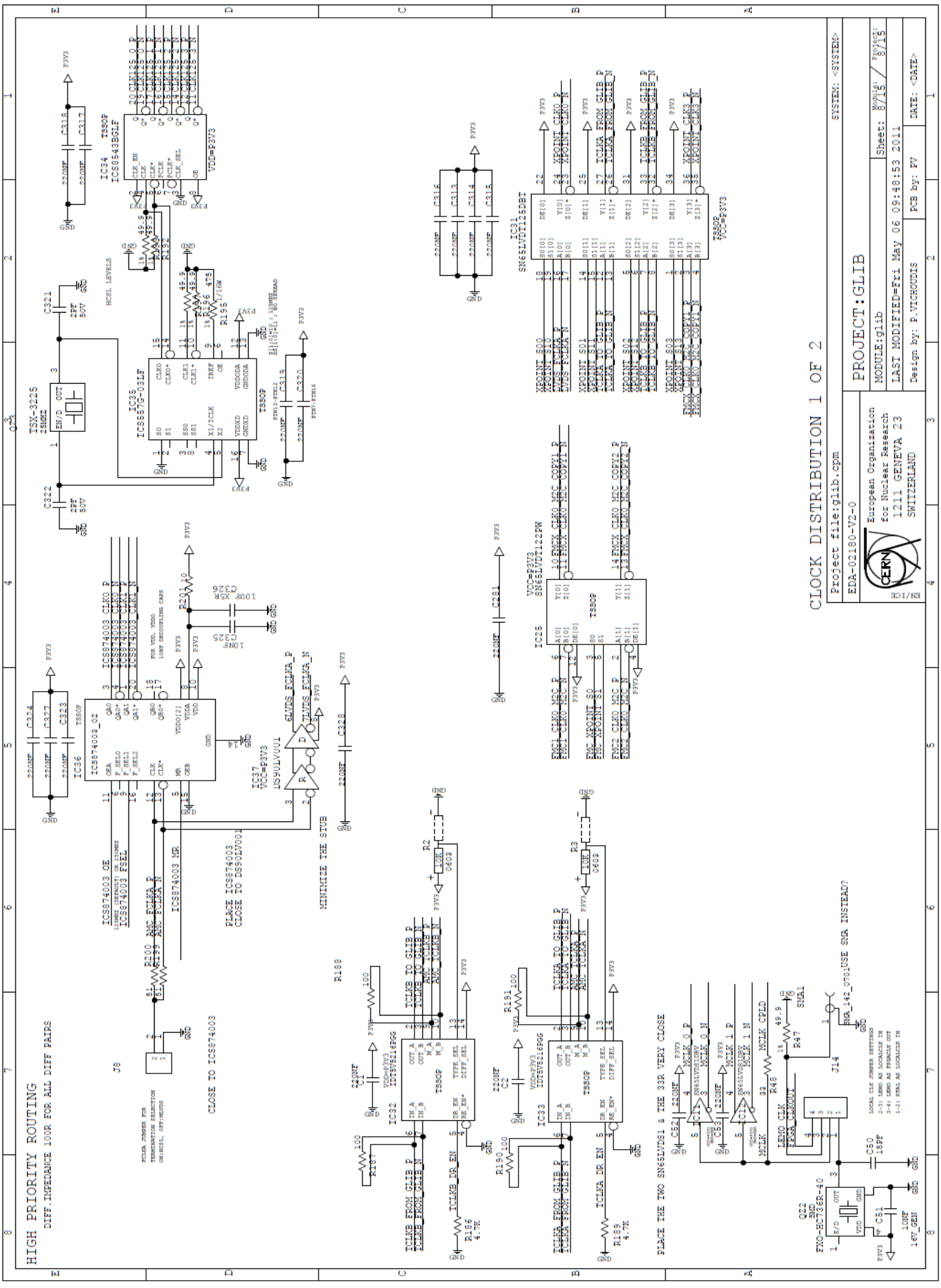

FPGA PART1
PROJECT:GLIB
European Organization
for Nuclear Research
1211 GENEVA 23
SWITZERLAND

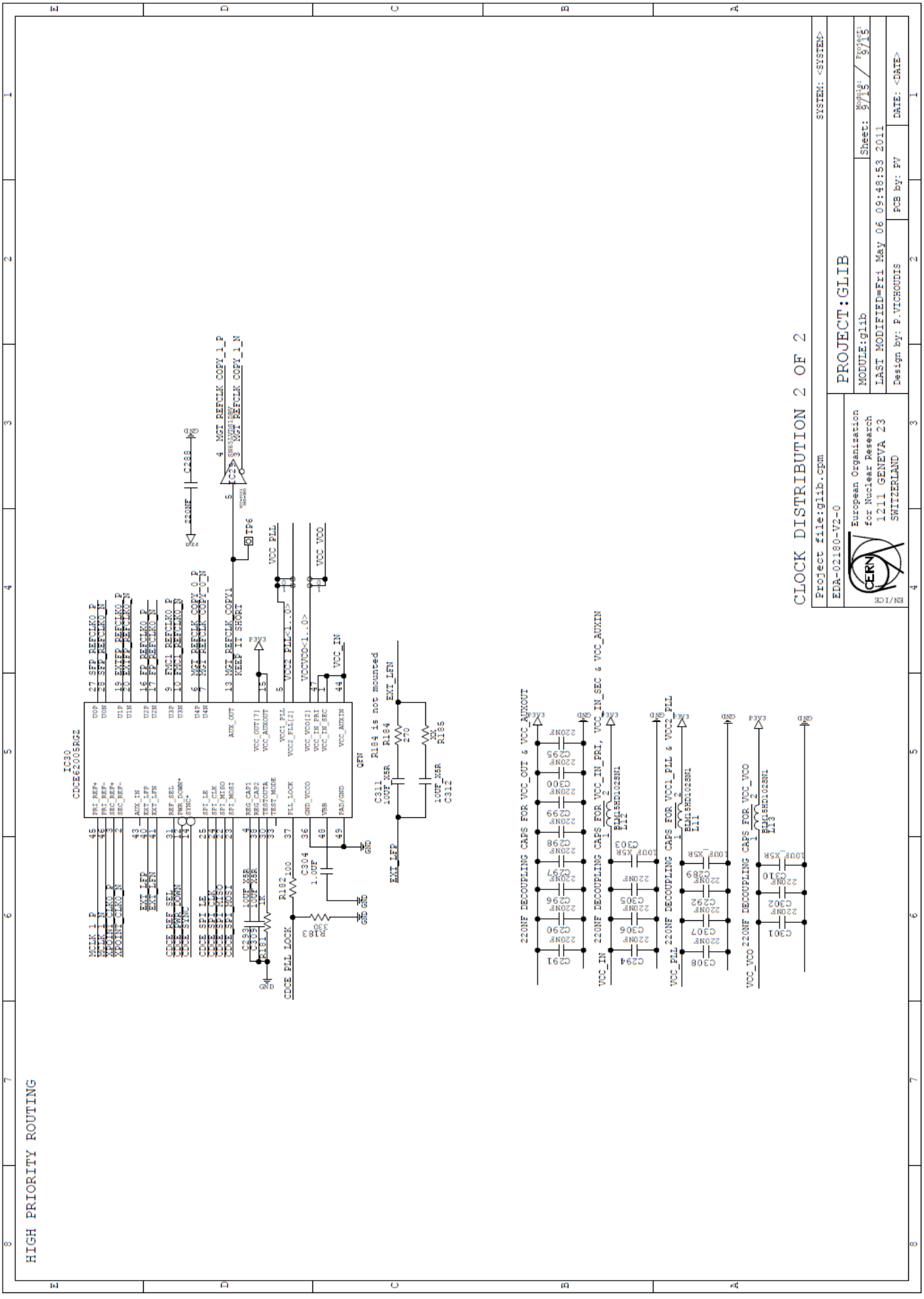

PROJECT:GLIB
FPGA PART2
Project file:glib.cpm
EDA-02180-V2-0
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:51 2011
Design by: P.VICHOUDIS
PCB by: PV
SYSTEM: <SYSTEM>
Sheet: Module / Project
4/15
DATE: <DATE>
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
IC2 XC6VLX130T-1FF1156CES
BANK 24: NO SWAP
BANK 25
BANK 26
BANK 32
BANK 33
BANK 34: NO SWAP
BANK 35
BANK 36
BANK 16: DIFF IMPEDANCE 100R SWAP BY PAIR
BANK 33: DIFF IMPEDANCE 100R SWAP BY PAIR
BANK 35: DIFF IMPEDANCE 100R SWAP _CC PAIRS ONLY BETWEEN THEM
BANK 36: DIFF IMPEDANCE 100R SWAP _CC PAIRS ONLY BETWEEN THEM
SWAP BETWEEN BANKS 13,14,15,16,22,23,25,26,32,33,35 IS ALLOWED
AS LONG AS ALL INDIVIDUAL RESTRICTIONS ARE RESPECTED
33R CLOSE TO IC
SRAM1 CEN_L
SRAM1 WE_L
SRAM1 OE_L
SRAM2 CEN_L
SRAM2 WE_L
SRAM2 OE_L
GBE INT_N
GBE SCL 4.7K
GBE RESET_N
FMC2 VIO B M2C

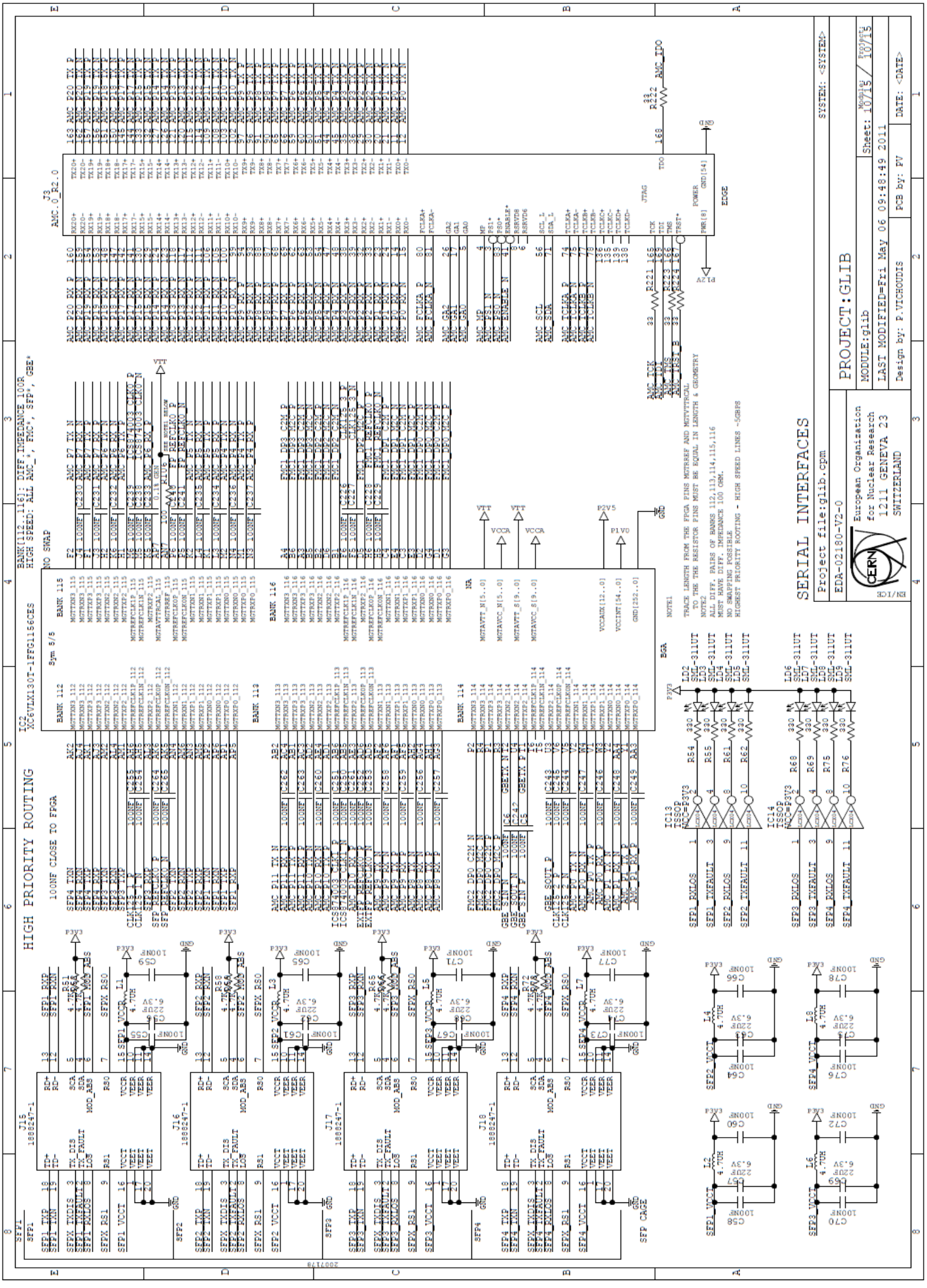

FMC#1
PROJECT:GLIB
MODULE:glib
Sheet: 5/15 / 5/15
LAST MODIFIED=Fri May 06 09:40:52 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
SEAF-40-06.5-S-10-A
33R CLOSE TO CONNECTOR
THE I2C EEPROM OF THE FMC
IS POWERED BY 3V3AUX
I2C ADDRESS
10100 GA1 GA0

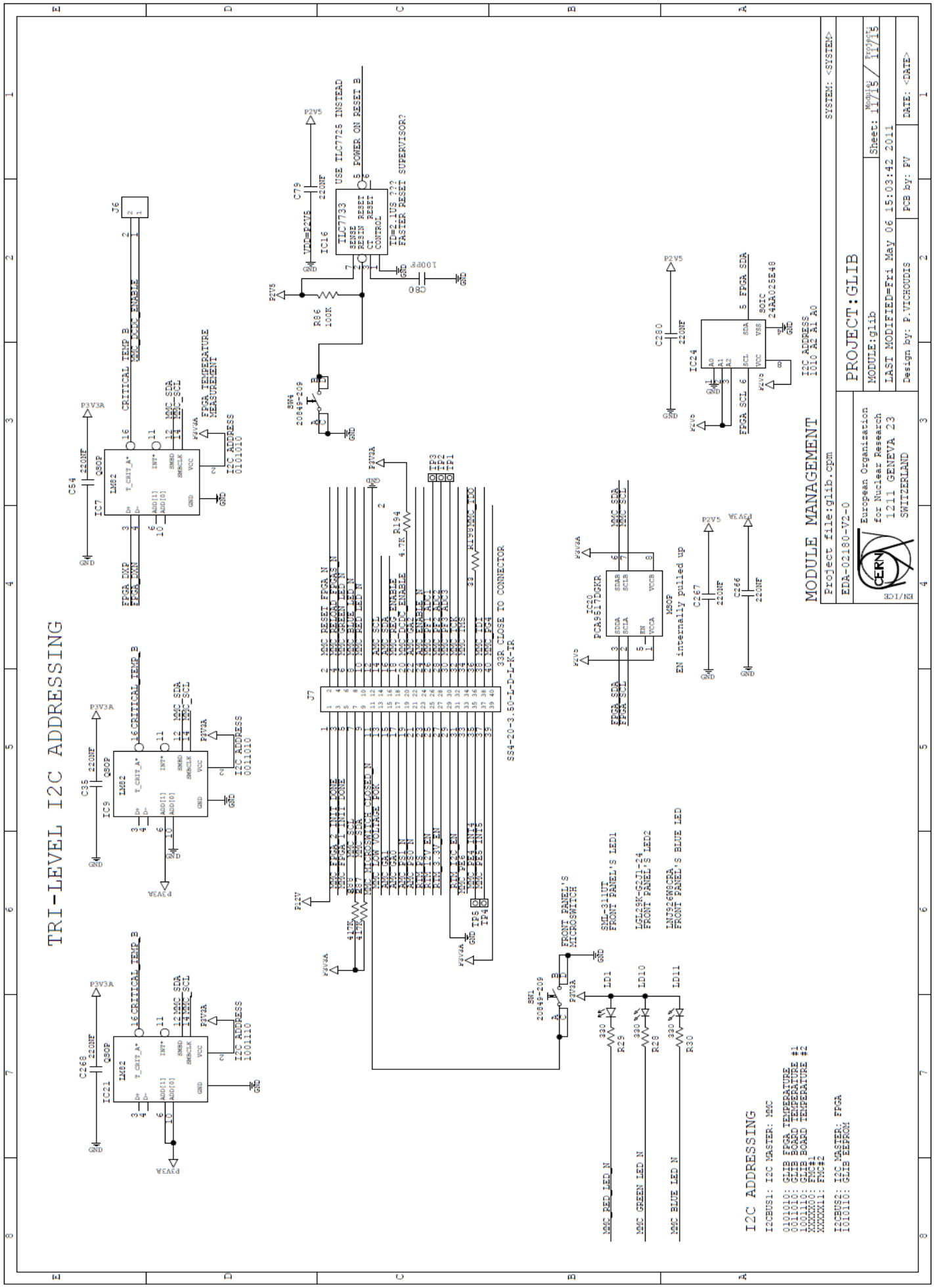

FMC#2
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 15:03:06 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
Sheet: 6/15 / 6/15
SEAF-40-06.5-S-10-A
33R CLOSE TO CONNECTOR
THE I2C EEPROM OF THE FMC IS POWERED BY 3V3AUX
I2C ADDRESS 10100 GA1 GA0

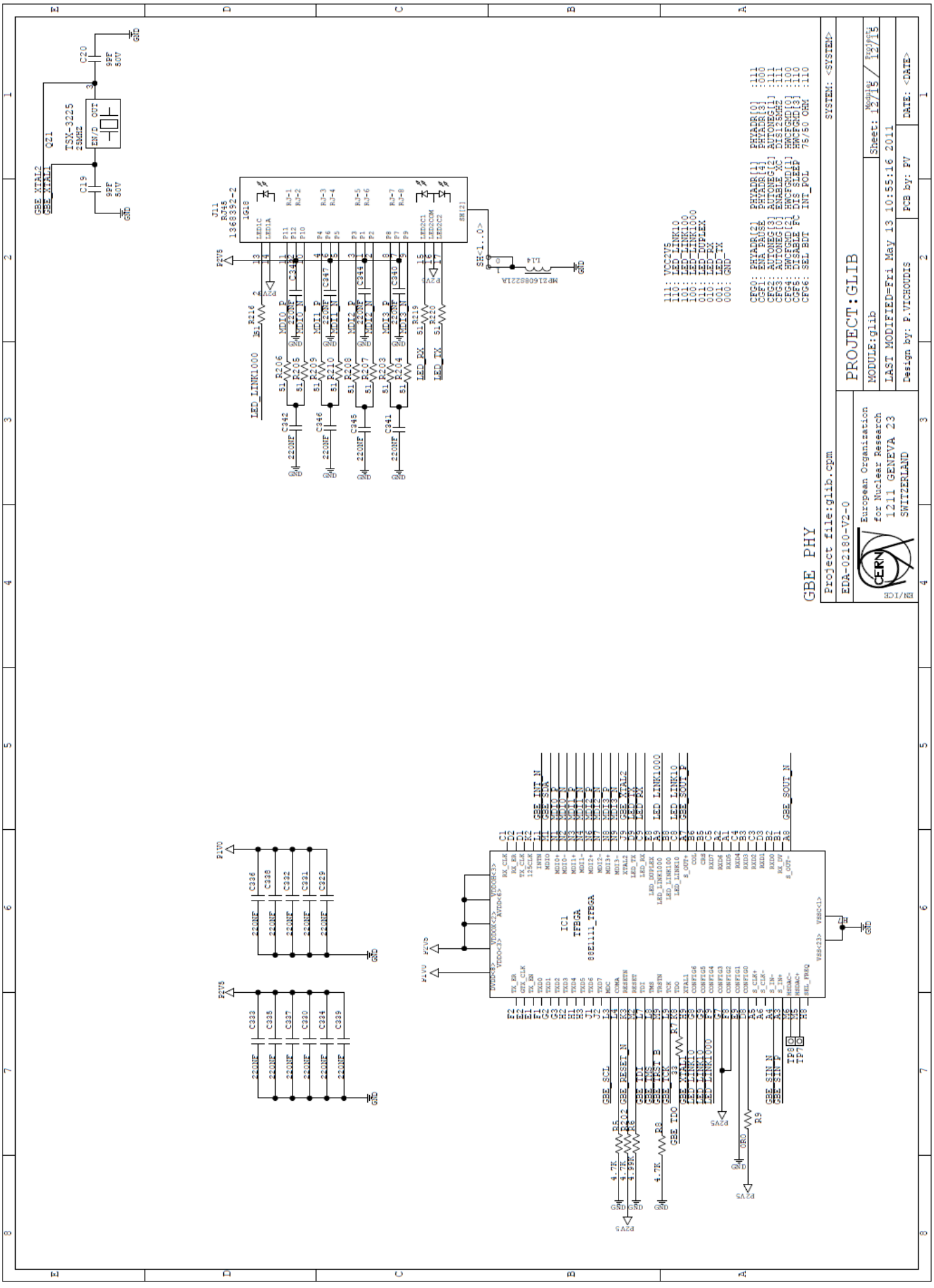

PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:52 2011
Design by: P.VICHOUDIS
PCB by: FV
DATE: <DATE>
SYSTEM: <SYSTEM>
Sheet: 7/15 / Project: 7/15
CONFIGURATION
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
IC3
XC2C128_VQ100
I/O BANK 1
I/O BANK 2
I/O BANK1: ALL SWAPPABLE EXCEPT PIN22
I/O BANK2: ALL SWAPPABLE
33R CLOSE TO IC
JTAG SCHEME
CLPD AS "SLAVE" : DRIVES "MASTER"DEVICE'S TDI
JTAG "MASTERS" ARE:
- CPLD JTAG HEADER
- JTAG HEADER FOR V6 & M6C
- JTAG FSM DRIVEN BY V6
- AMC JTAG
- M6C JTAG
JTAG "SLAVES" ARE:
- V6, FMCS, SRAMS, GBE
JTAG DRIVER PINS HAVE 33R IN SERIES
PULL-UP/DOWN RESISTORS ARE USED LINES DRIVEN
BY PLDGRABLE JTAG DEVICES
MASTER
FULL-UPS
SLAVE
NO FULL-UPS
DECOUPLING CAPACITORS PLACEMENT:
IC4
XCF128XFTG64C
FTG64C
J13
J12
MOLEX_87831
SW1
SW2
SW3
SW4
ST1
ST2
ST3
TP9
TP11
R118
FPGA_INIT_B
FPGA_M2
FPGA_M1
FPGA_M0
M[2:0]=010
CPLD HEADER TMS
CPLD HEADER TCK
CPLD HEADER TDI
CPLD HEADER TDO
JTAG HEADER TMS
JTAG HEADER TCK
JTAG HEADER TDI
JTAG HEADER TDO
SRAM2_WE_L
SRAM2_CE2
SRAM2_OE_L
FPGA_PROGRAM_B
FPGA_CCLK
SRAM2_CE1_L
SRAM2_DQ7
SRAM2_DQ6
SRAM2_DQ5
SRAM2_DQ4
SRAM2_DQ3
SRAM2_DQ2
SRAM2_DQ1
SRAM2_DQ0
SRAM2_A22
SRAM2_A21
SRAM2_A20
SRAM2_A19
SRAM2_A18
SRAM2_A17
SRAM2_A16
SRAM2_A15
SRAM2_A14
SRAM2_A13
SRAM2_A12
SRAM2_A11
SRAM2_A10
SRAM2_A9
SRAM2_A8
SRAM2_A7
SRAM2_A6
SRAM2_A5
SRAM2_A4
SRAM2_A3
SRAM2_A2
SRAM2_A1
SRAM2_A0
FPGA_RS1
FPGA_RS0
SFPX_RS0
SFPX_RS1
SFPX_MODIS
R113 R114 are not mounted
R124 R125 are not mounted
R121 R122 R123 are not mounted
P3V3A
P2V5A
P1V8A
P1V2A
GND

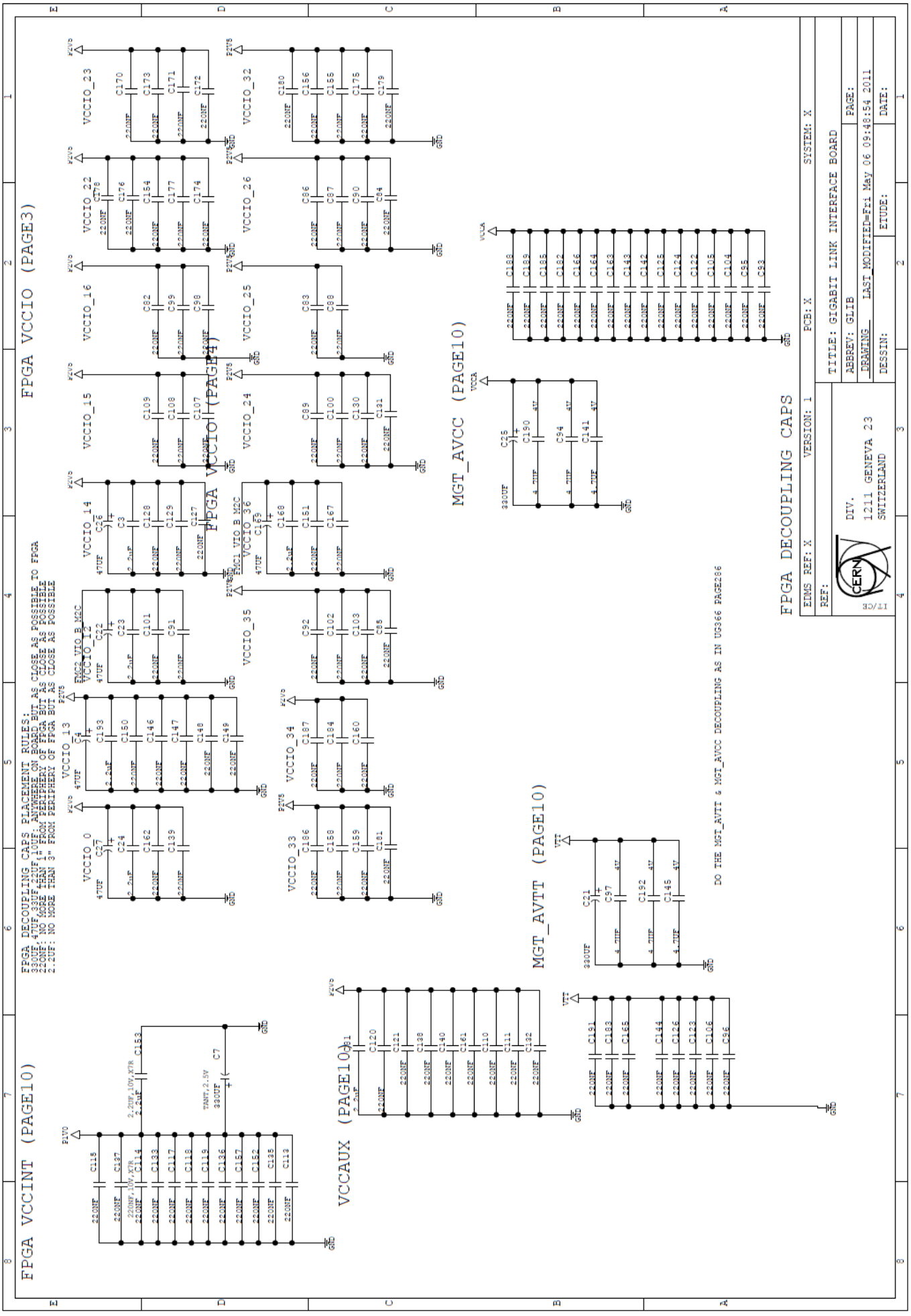

CLOCK DISTRIBUTION 1 OF 2
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:53 2011
Design by: P.VICHOUDIS
PCB by: PV
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
HIGH PRIORITY ROUTING
DIFF.IMPEDANCE 100R FOR ALL DIFF PAIRS
PLACE THE TWO SN65LVDS1 & THE 33R VERY CLOSE
MINIMIZE THE STUB
CLOSE TO ICS874003
PLACE ICS874003 CLOSE TO DS90LV001
IC31 SN65LVDT125DBT
IC25 SN65LVDT122PW
IC32 IC33 IDT5V5216PGG
IC34 ICS8543BGLF
IC35 ICS85573-031F
IC36 ICS874003_02
IC37 US9ULVUU1
J8
J14
Q22 EXO-HC736R-40

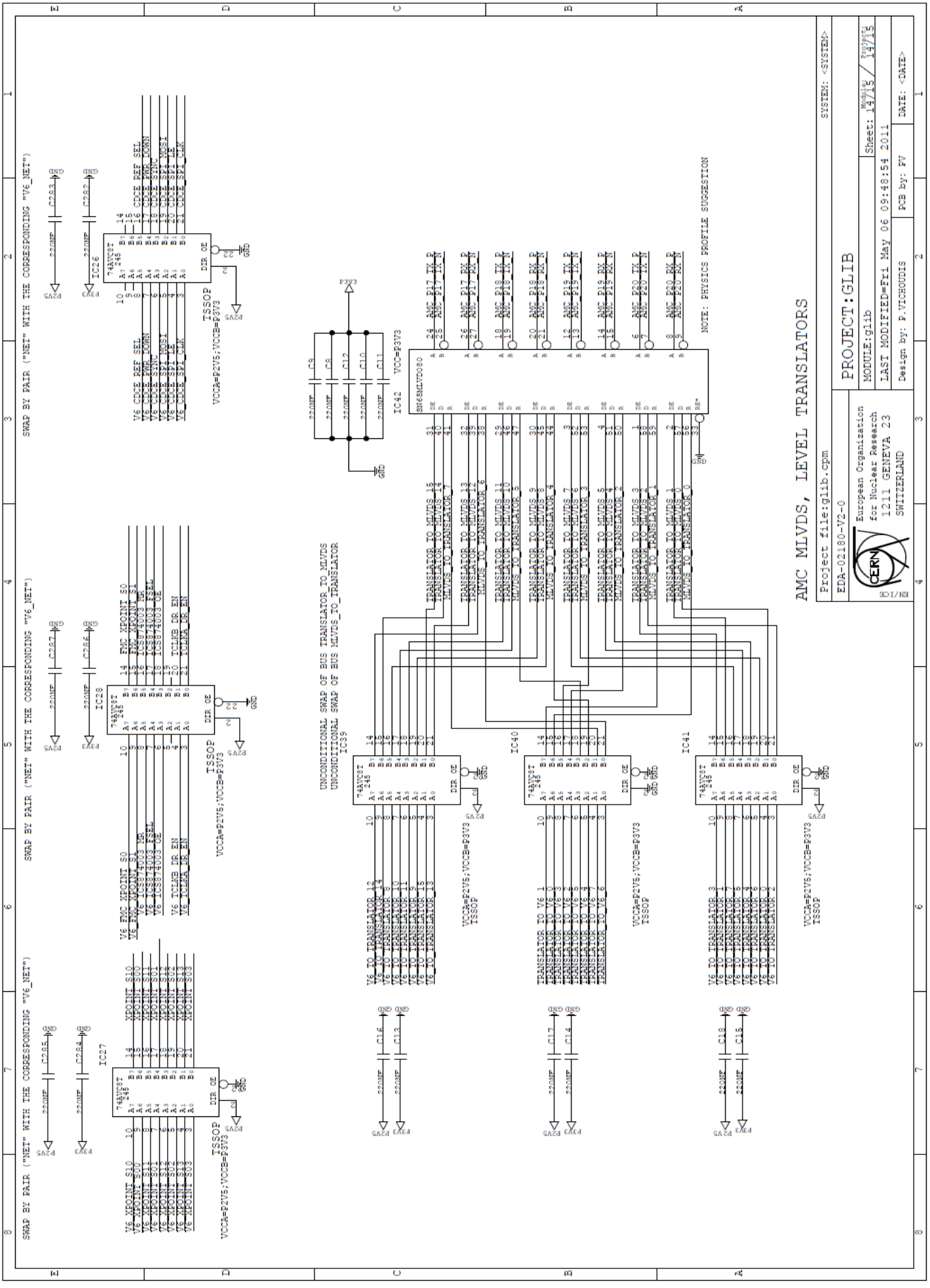

HIGH PRIORITY ROUTING
CLOCK DISTRIBUTION 2 OF 2
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:53 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
IC30
CDCE62005RGZ
27 SFP REFCLK0_P
28 SFP REFCLK0_N
19 EXTFP REFCLK0_P
20 EXTFP REFCLK0_N
16 FP REFCLK0_P
17 FP REFCLK0_N
9 EMC1 REFCLK0_P
10 EMC1 REFCLK0_N
6 MGT REFCLK COPY_0_P
7 MGT REFCLK COPY_0_N
4 MGT REFCLK COPY_1_P
3 MGT REFCLK COPY_1_N
13 MGT REFCLK COPY1
KEEP IT SHORT
MCLK_1_P
MCLK_1_N
XPOINT_CLKO_P
XPOINT_CLKO_N
EXT_LFP
EXT_LFN
CDCE_REF_SEL
CDCE_PWR_DOWN
CDCE_SYNC
CDCE_SPI_LE
CDCE_SPI_CLK
CDCE_SPI_MISO
CDCE_SPI_MOSI
CDCE_PLL_LOCK
R182 100
C304 1.0UF
R183 0R
C303 100UF_X5R
C301 100UF_X5R
R181 1K
C311 10UF_X5R
R184 270
R185 XX
C312 10UF_X5R
R184 is not mounted
C238 220NF
TP6
VCC_PLL
VCC_VCO
VCC_IN
220NF DECOUPLING CAPS FOR VCC_OUT & VCC_AUXOUT
220NF DECOUPLING CAPS FOR VCC_IN_PRI, VCC_IN_SEC & VCC_AUXIN
220NF DECOUPLING CAPS FOR VCC1_PLL & VCC2_PLL
220NF DECOUPLING CAPS FOR VCC_VCO
BLM15HD102SN1
L112
L111
L113
C291
C290
C296
C297
C298
C299
C300
C294
C305
C306
C308
C307
C292
C289
C310
C302
C301
P3V3
GND

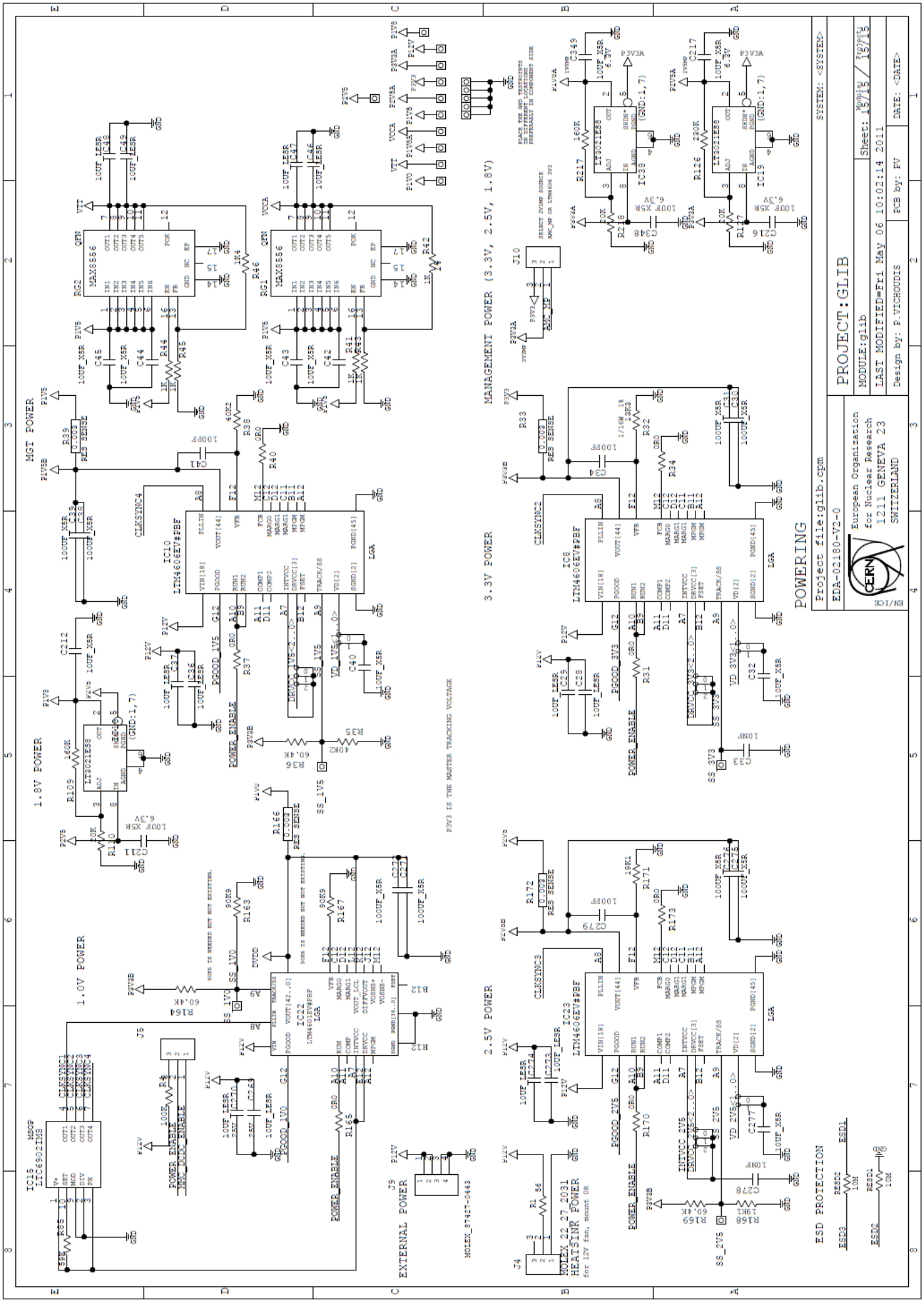

HIGH PRIORITY ROUTING
SERIAL INTERFACES
PROJECT:GLIB
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:49 2011
Design by: P.VICHOUDIS
PCB by: PV
SYSTEM: <SYSTEM>
DATE: <DATE>
Sheet: 10/15 / Pages: 10/15
IC2
XC6VLX130T-1FFG1156CES
BANK(112,116): DIFF_IMPEDANCE 100R
HIGH SPEED: ALL AMC_*, FMC*, SFP*, GBE*
NO SWAP
100NF CLOSE TO FPGA
TRACE LENGTH FROM THE FPGA PINS MGTRREF AND MGTVTTRCAL TO THE THE RESISTOR PINS MUST BE EQUAL IN LENGTH & GEOMETRY
ALL DIFF. PAIRS OF BANKS 112,113,114,115,116 MUST HAVE DIFF. IMPEDANCE 100 OHM.
NO SWAPPING POSSIBLE
HIGHEST PRIORITY ROUTING - HIGH SPEED LINES -5GBPS
J3
AMC_0_R2_0
EDGE
IC13
IC14
TSSOP
SFP1
SFP2
SFP3
SFP4
SFP CAGE
J15
J16
J17
J18
1888247-1
SFP1_RXLOS
SFP1_TXFAULT
SFP2_RXLOS
SFP2_TXFAULT
SFP3_RXLOS
SFP3_TXFAULT
SFP4_RXLOS
SFP4_TXFAULT
SFP1_VCCR_L1
SFP2_VCCR_L3
SFP3_VCCR_L5
SFP4_VCCR_L7
SFP1_VCCT_L2
SFP2_VCCT_L4
SFP3_VCCT_L6
SFP4_VCCT_L8
SFP1_MOD_ABS
SFP2_MOD_ABS
SFP3_MOD_ABS
SFP4_MOD_ABS
SFP1_RXP
SFP1_RXN
SFP1_TXP
SFP1_TXN
SFP2_RXP
SFP2_RXN
SFP2_TXP
SFP2_TXN
SFP3_RXP
SFP3_RXN
SFP3_TXP
SFP3_TXN
SFP4_RXP
SFP4_RXN
SFP4_TXP
SFP4_TXN
SFP_REFCLKO_P
SFP_REFCLKO_N
SFP_REFCLKI_P
SFP_REFCLKI_N
SFPX_RS0
SFPX_RS1
SFPX_TXDIS
SFPX_TXFAULT
SFPX_RXLOS
SFPX_MOD_ABS
SFPX_SCL
SFPX_SDA
SFPX_VCCR
SFPX_VCCT
SFPX_VEER
SFPX_VEET
SFPX_RD+
SFPX_RD-
SFPX_TD+
SFPX_TD-
AMC_TCK
AMC_TDI
AMC_TDO
AMC_TMS
AMC_TEST_B
AMC_SCL
AMC_SDA
AMC_FCLKA_P
AMC_FCLKA_N
AMC_FCLKB_P
AMC_FCLKB_N
AMC_GA2
AMC_GA1
AMC_GA0
AMC_MP
AMC_PS1_N
AMC_PS0_N
AMC_ENABLE_N
FMC1_DP3_C2M_P
FMC1_DP3_C2M_N
FMC1_DP2_C2M_P
FMC1_DP2_C2M_N
FMC1_DP1_C2M_P
FMC1_DP1_C2M_N
FMC1_DP0_C2M_P
FMC1_DP0_C2M_N
FMC1_PRSNT_M2C_B
FMC1_REFCLK0_P
FMC1_REFCLK0_N
GBE_SOUT_P
GBE_SOUT_N
GBE_SIN_P
GBE_SIN_N
P3V3
P2V5
P1V0
VTT
VCCA
GND


TRI-LEVEL I2C ADDRESSING
MODULE MANAGEMENT
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 15:03:42 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organisation for Nuclear Research
1211 GENEVA 23
SWITZERLAND
I2C ADDRESSING
I2CBUS1: I2C MASTER: MMC
0101010: GLIB FPGA TEMPERATURE
0011010: GLIB BOARD TEMPERATURE #1
1001110: GLIB BOARD TEMPERATURE #2
XXXXX000: FMC#1
XXXXX111: FMC#2
I2CBUS2: I2C MASTER: FPGA
1010110: GLIB EEPROM

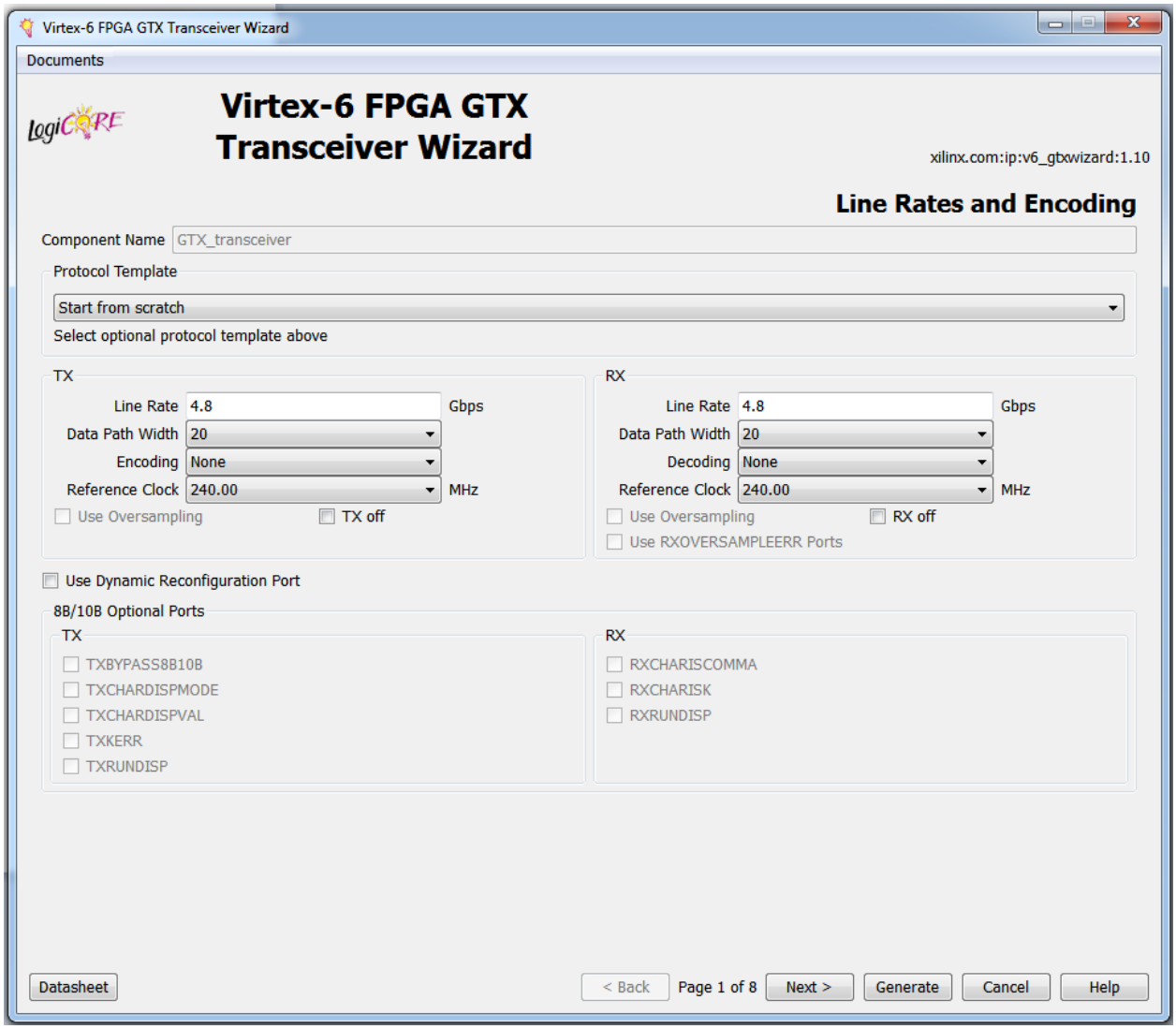

GBE PHY
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 13 10:55:16 2011
Design by: P.VICHOUDIS
PCB by: FV
SYSTEM: <SYSTEM>
DATE: <DATE>
Sheet: 12/15 / 15
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
IC1
IFBGA
88E1111_IFBGA
J11
RJ45
1368392-2
Q21
TSX-3225
25MHz
C20
C19
9PF
50V
R216
R206
R205
R209
R208
R207
R203
R204
R219
R220
51
220NF
LED_LINK1000
LED_DUPLEX
LED_RX
LED_TX
LED_LINK100
LED_LINK10
GBE_INT_N
GBE_SCK
GBE_SOUT_P
GBE_SOUT_N
GBE_XTAL2
GBE_SCL
GBE_RESET_N
GBE_TDI
GBE_TMS
GBE_TRST_B
GBE_TCK
GBE_TDO
GBE_XTAL1
GBE_SIN_N
GBE_SIN_P
R5
R502
R6
R8
R9
4.7K
4.99K
33
GND
P2V5
P1V0
P3V3


FPGA VCCINT (PAGE10)
FPGA VCCIO (PAGE3)
FPGA VCCIO (PAGE4)
MGT_AVCC (PAGE10)
MGT_AVTT (PAGE10)
VCCAUX (PAGE10)
FPGA DECOUPLING CAPS
TITLE: GIGABIT LINK INTERFACE BOARD
ABBREV: GLIB
DRAWING
DESSIN:
DATE:
PAGE:
ETUDE:
LAST_MODIFIED=Fri May 06 09:48:54 2011
DIV.
1211 GENEVA 23
SWITZERLAND
EDMS REF: X
VERSION: 1
PCB: X
SYSTEM: X
REF:
DO THE MGT_AVTT & MGT_AVCC DECOUPLING AS IN UG366 PAGE266

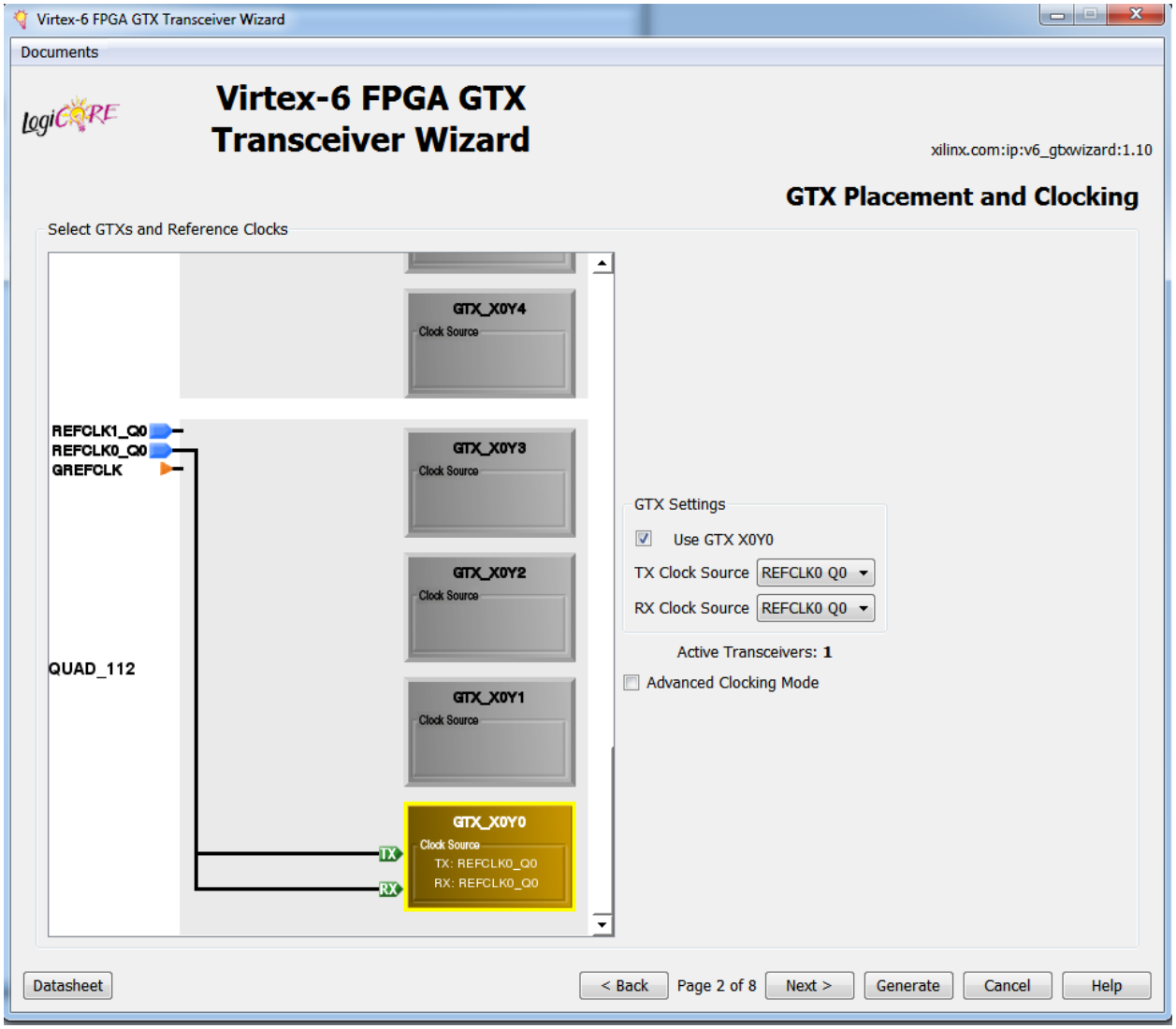

AMC MLVDS, LEVEL TRANSLATORS
PROJECT:GLIB
MODULE:glib
LAST MODIFIED=Fri May 06 09:48:54 2011
Design by: F.VICHOUDIS
PCB by: FV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organization for Nuclear Research
1211 GENEVA 23
SWITZERLAND
NOTE: PHYSICS PROFILE SUGGESTION
UNCONDITIONAL SWAP OF BUS TRANSLATOR_TO_MLVDS
UNCONDITIONAL SWAP OF BUS MLVDS_TO_TRANSLATOR
SWAP BY PAIR ("NET" WITH THE CORRESPONDING "V6_NET")
IC39
IC40
IC41
IC42
TSSOP
VCCA=F2V5;VCCB=F3V3


PROJECT:GLIB
MODULE:glib
Sheet: 15/15 / 15/15
LAST MODIFIED=Fri May 06 10:02:14 2011
Design by: P.VICHOUDIS
PCB by: PV
DATE: <DATE>
SYSTEM: <SYSTEM>
Project file:glib.cpm
EDA-02180-V2-0
European Organisation for Nuclear Research
1211 GENEVA 23
SWITZERLAND
POWERING
MGT POWER
1.8V POWER
1.0V POWER
3.3V POWER
2.5V POWER
MANAGEMENT POWER (3.3V, 2.5V, 1.8V)
EXTERNAL POWER
MOLEX 22-27-2031
HEATSINK POWER
ESD PROTECTION
P3V3 IS THE MASTER TRACKING VOLTAGE

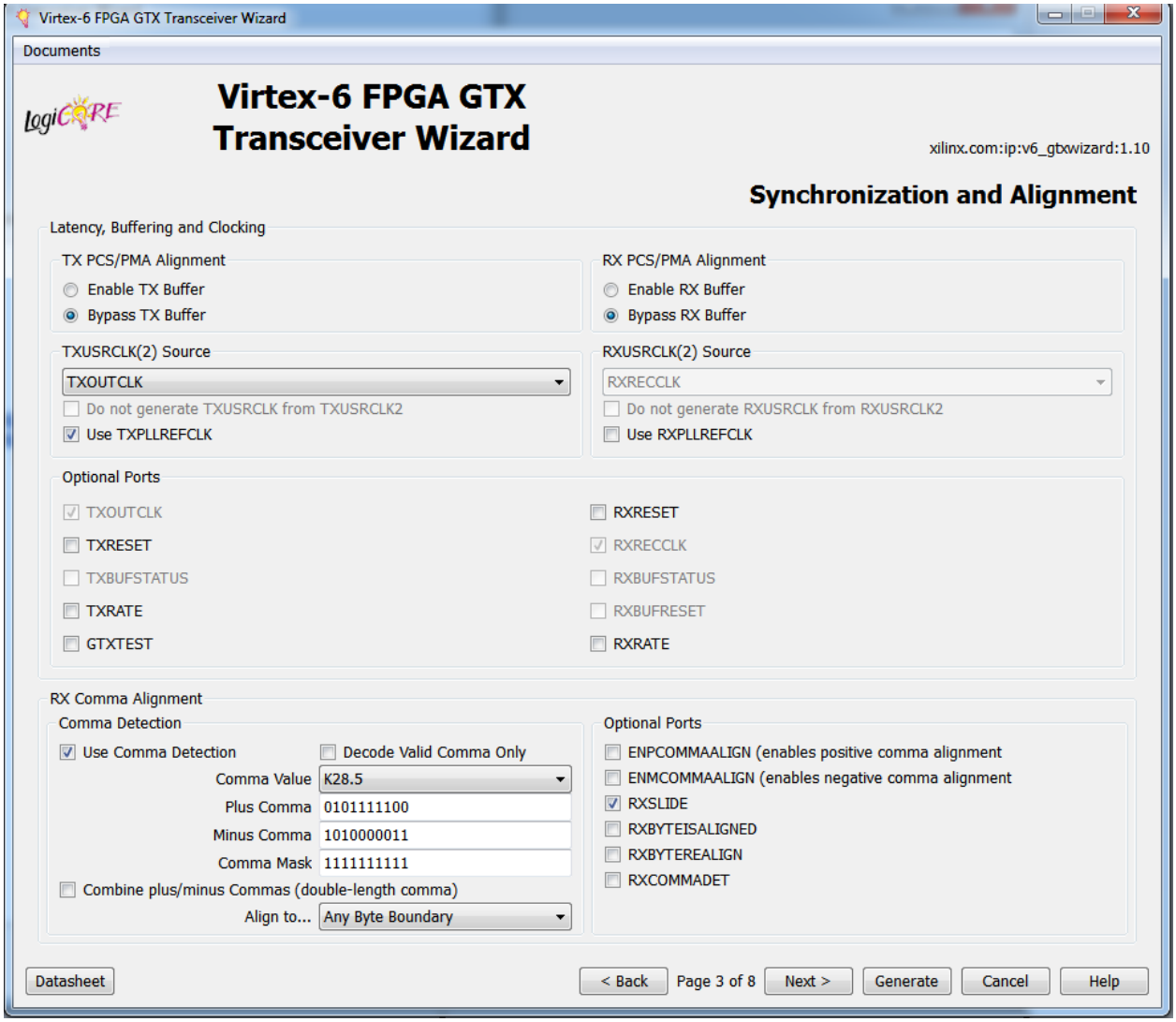
Appendix B - GTX Configuration Wizard



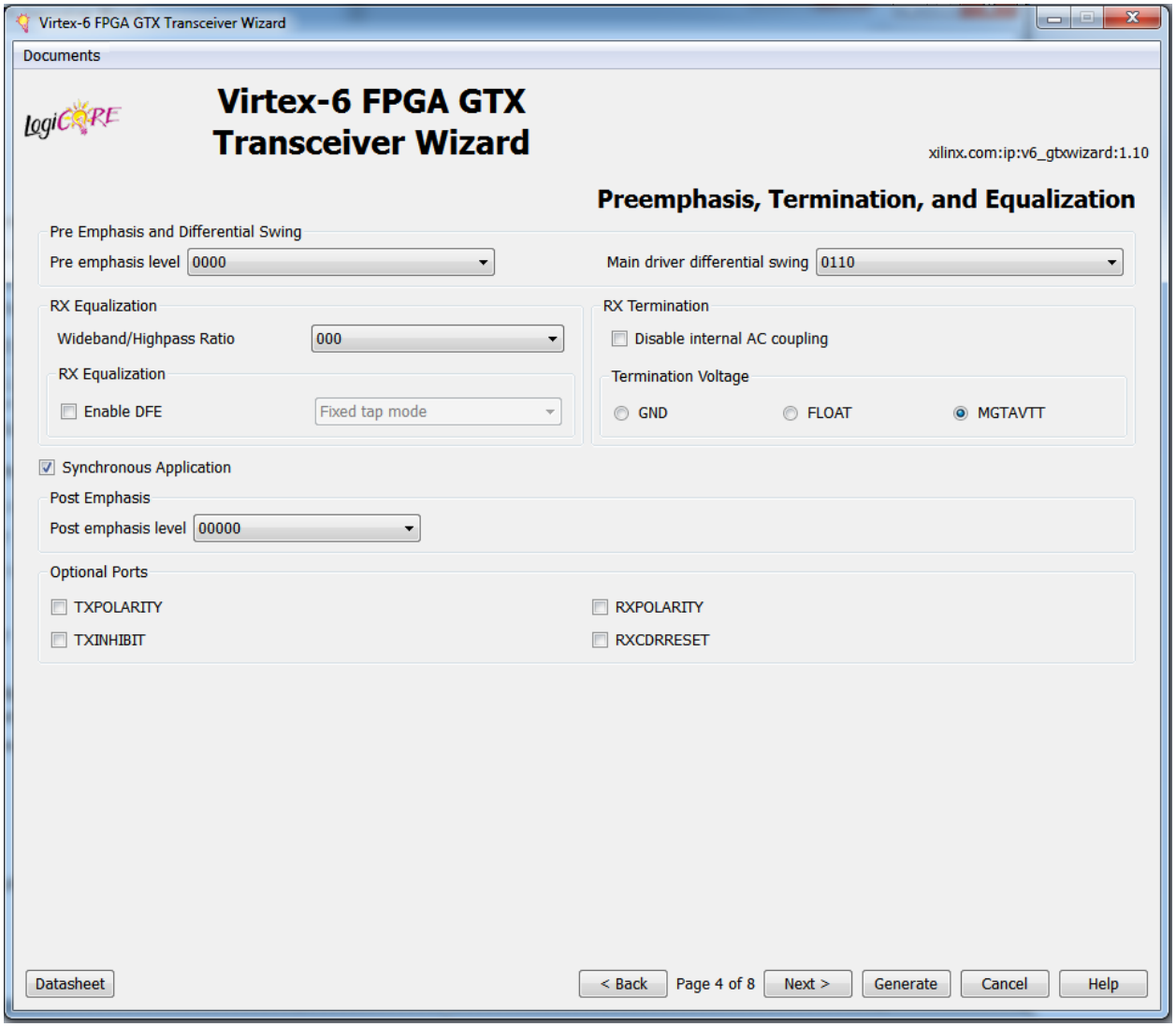
GTX transceiver wizard - page 1



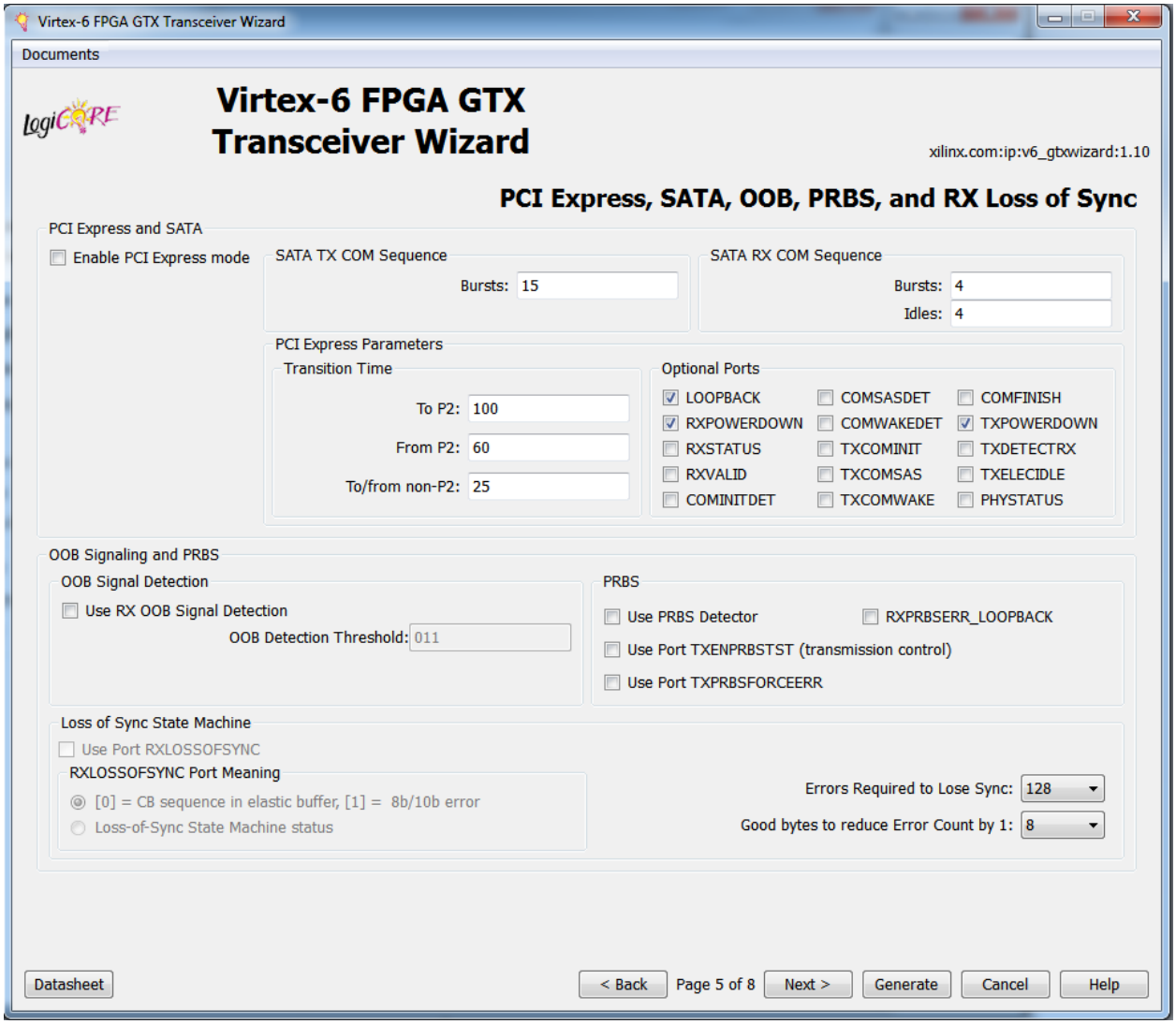
GTX transceiver wizard - page 2



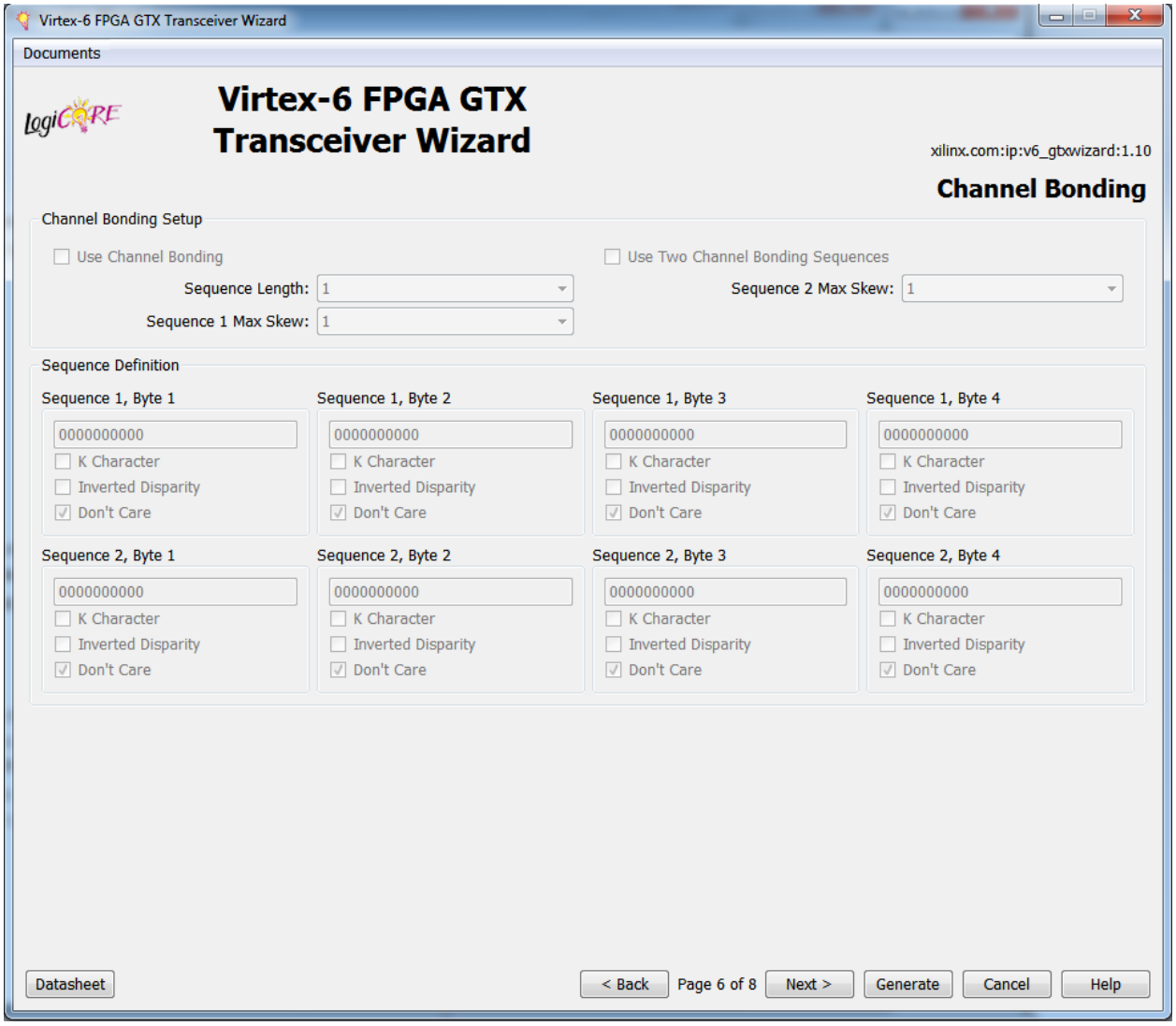
GTX transceiver wizard - page 3



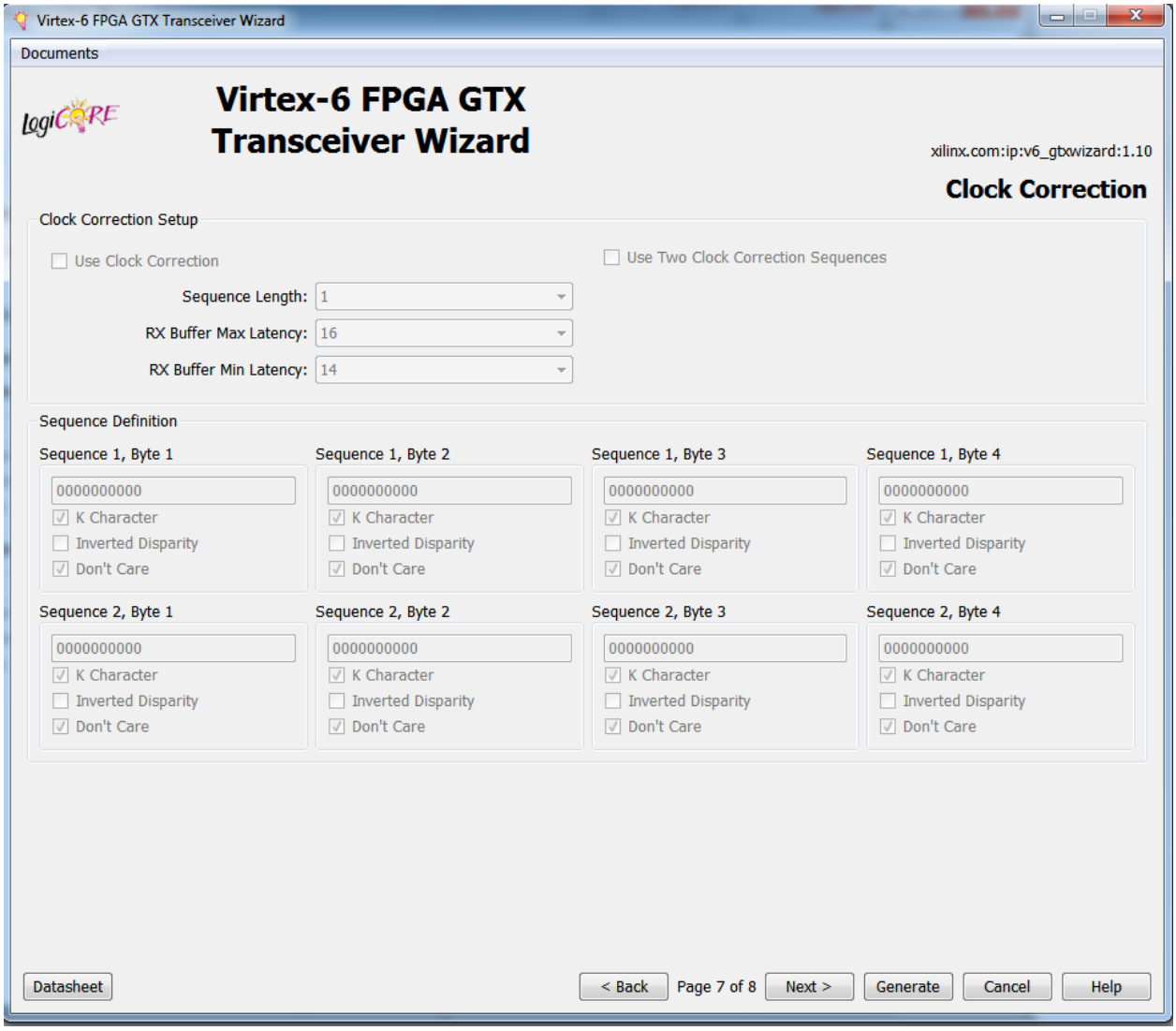
GTX transceiver wizard - page 4



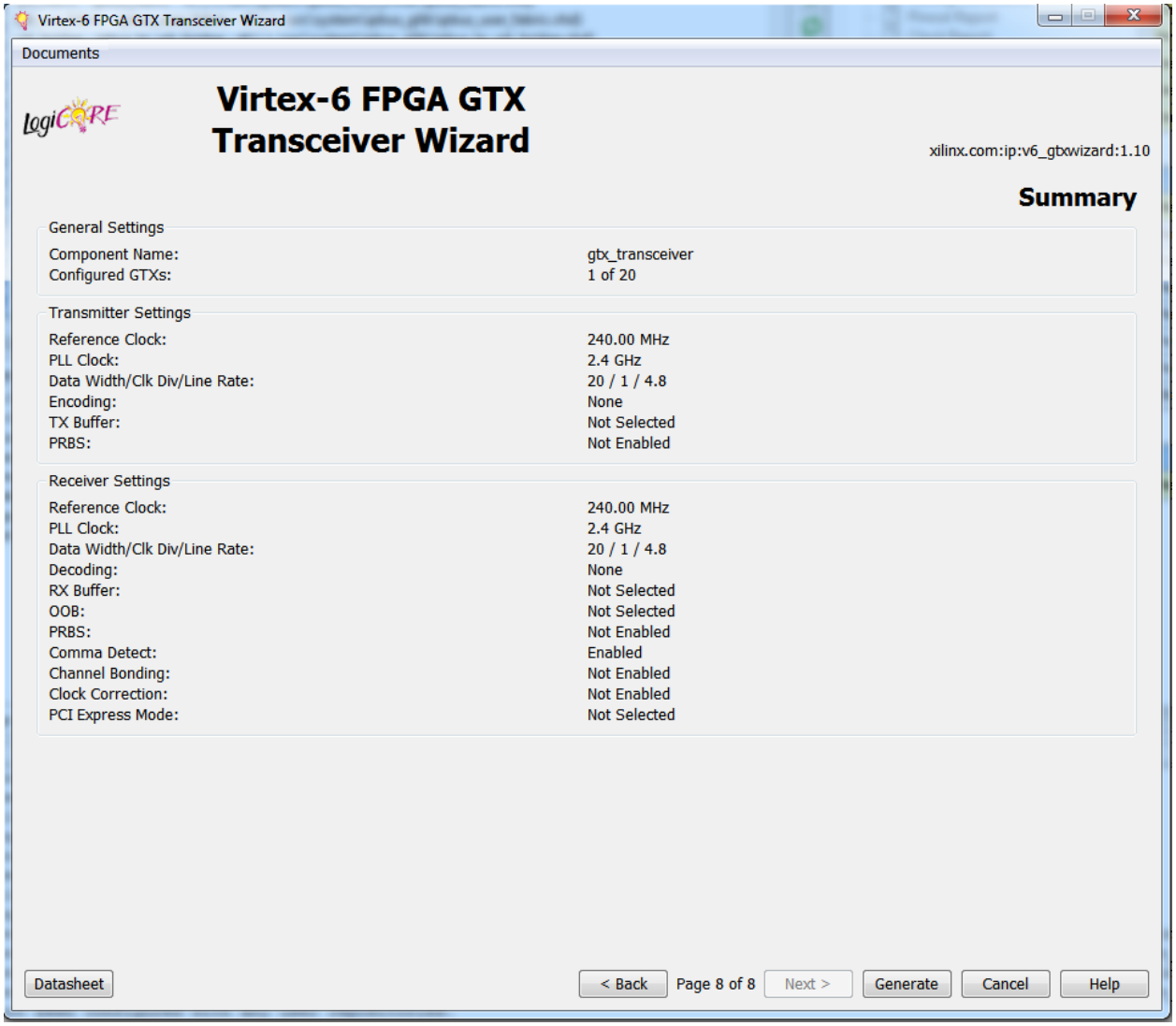
GTX transceiver wizard - page 5



GTX transceiver wizard - page 6



GTX transceiver wizard - page 7



GTX transceiver wizard - page 8

Appendix C - HDL Codes

glib_be.vhd

```
--------------------------------------------------------------------------------------------------------
------------------------------------ Module Information ------------------------------------------------
--------------------------------------------------------------------------------------------------------
--                                                                                                    --
-- Company:                CERN (PH-ESE-BE)                                                           --
-- Engineer:               Manoel Barros Marin (manoel.barros.marin@cern.ch) (m.barros@ieee.org)       --
--                                                                                                    --
-- Create Date:            10/12/2011                                                                 --
-- Project Name:           BE/FE Link simulation                                                      --
-- Module Name:            glib_be                                                                    --
--                                                                                                    --
-- Languaje:               VHDL'93                                                                    --
--                                                                                                    --
-- Target Devices:         GLIB(Virtex 6)                                                             --
-- Tool versions:          ISE 13.2                                                                   --
--                                                                                                    --
-- Revision:               1.0                                                                        --
--                                                                                                    --
-- Additional Comments:                                                                               --
--                                                                                                    --
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
-- IEEE VHDL standard library:
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
-- Xilinx devices library:
library unisim;
use unisim.vcomponents.all;
--------------------------------------------------------------------------------------------------------
------------------------------------ Module Body -------------------------------------------------------
--------------------------------------------------------------------------------------------------------
entity glib_be is
   port (
      -- Reset:
      POWER_ON_RESET_B                        : in  std_logic;
      -- Clock:
      MCLK_0_P                                : in  std_logic;
      MCLK_0_N                                : in  std_logic;
      MGT_REFCLK_COPY_0_P                     : in  std_logic;
      MGT_REFCLK_COPY_0_N                     : in  std_logic;
      SFP_REFCLK0_P                           : in  std_logic;
      SFP_REFCLK0_N                           : in  std_logic;
      -- Clock Synthesizer:
      CDCE_PWR_DOWN                           : out std_logic;
      CDCE_SYNC                               : out std_logic;
      CDCE_REF_SEL                            : out std_logic;
      CDCE_SPI_CLK                            : out std_logic;
      CDCE_SPI_LE                             : out std_logic;
      CDCE_SPI_MOSI                           : out std_logic;
      CDCE_SPI_MISO                           : in  std_logic;
      CDCE_PLL_LOCK                           : in  std_logic;
      -- GTX transceiver:
      SFP1_TXN                                : out std_logic;
      SFP1_TXP                                : out std_logic;
      SFP1_RXN                                : in  std_logic;
      SFP1_RXP                                : in  std_logic;
      -- FMC#2 LED:
      FMC2_LA18_P_CC                          : out std_logic;
      -- SMA outputs:
      SMA_ON_GLIB                             : out std_logic;
      AMC_P12_TX_P                            : out std_logic;
      AMC_P13_TX_P                            : out std_logic;
```

```
64        -- Dummy:
65        DUMMY_PORT                                : out std_logic
66        --============================================================================--
67     );
68  end glib_be;
69  architecture structural of glib_be is
70     --============================== Declarations ==============================--
71     attribute S : string;   -- This attribute is for avoiding signal trimming.
72     -- General Reset:
73     signal general_reset                         : std_logic;
74     -- Clocks:
75     signal mclk0                                 : std_logic;
76     signal gtxrefclk                             : std_logic;
77     signal tfclk                                 : std_logic;
78     signal twclk                                 : std_logic;
79     signal rwclk                                 : std_logic;
80     signal rfclk                                 : std_logic;
81     -- CDCE Synchronizer:
82     signal cs_sync_done                          : std_logic;
83     -- GBT Link Reset Control:
84     signal gtxTx_reset                           : std_logic;
85     signal gtxRx_reset                           : std_logic;
86     signal gbtTx_reset                           : std_logic;
87     signal gbtRx_reset                           : std_logic;
88     -- Pattern Generator:
89     signal tx_frame                              : std_logic_vector(83 downto 0);
90     -- GBT TX:
91     signal txWord_from_gbtTx                     : std_logic_vector(19 downto 0);
92     -- GTX TX:
93     signal rxWord_from_gtxRx                     : std_logic_vector(19 downto 0);
94     signal gtx_aligned                           : std_logic;
95     -- GBT RX:
96     signal gbtRx_aligned                         : std_logic;
97     signal bit_slip_nbr                          : std_logic_vector(4 downto 0);
98     signal header_flag                           : std_logic;
99     signal rx_frame                              : std_logic_vector(83 downto 0);
100    -- Pattern Checker:
101    signal num_of_frames                         : std_logic_vector(31 downto 0);
102    attribute S of num_of_frames                 : signal is "true";
103    signal num_of_errors                         : std_logic_vector(31 downto 0);
104    attribute S of num_of_errors                 : signal is "true";
105    signal error_flag                            : std_logic;
106    attribute S of error_flag                    : signal is "true";
107    -- Pattern Match Flags:
108    signal tx_frame_flag                         : std_logic;
109    signal rx_frame_flag                         : std_logic;
110    -- RX Phase Aligner:
111    signal done_from_phaseAligner                : std_logic;
112    -- Output buffers:
113    signal to_obuf                               : std_logic_vector(2 downto 0);
114    signal from_obuf                             : std_logic_vector(2 downto 0);
115    --============================================================================--
116 --============================================================================--
117 -----      --=================================================--
118 begin    --================== Architecture Body ==================--
119 -----      --=================================================--
120 --============================================================================--
121    --=========================  Port Assignments  =========================--
122    -- Clock Synthesizer:
123    CDCE_SPI_CLK                                 <= '0';
124    CDCE_SPI_LE                                  <= '1';
125    CDCE_SPI_MOSI                                <= '0';
```

glib_be.vhd - page 2

```
126     -- Output buffers (24mA, LVCMOS25, Fast Slew Rate):
127     SMA_ON_GLIB     <= from_obuf(0);  to_obuf(0) <= mclk0;
128     AMC_P12_TX_P    <= from_obuf(1);  to_obuf(1) <= tx_frame_flag;
129     AMC_P13_TX_P    <= from_obuf(2);  to_obuf(2) <= rx_frame_flag;
130     -- Dummy signals:
131     DUMMY_PORT      <= '1' when  num_of_frames = x"FFFFFFFF" or
132                                  num_of_errors = x"FFFFFFFF" or
133                                  error_flag    = '1'
134                        else '0';
135     --=================================================================--
136     --====================== Component Instantiations ======================--
137     -- Clock Buffers:
138     mclk0_ibufgds: IBUFGDS
139        generic map (
140           IBUF_LOW_PWR                          => FALSE,
141           IOSTANDARD                            => "LVDS_25")
142        port map (
143           O                                     => mclk0,
144           I                                     => MCLK_0_P,
145           IB                                    => MCLK_0_N
146        );
147     mgt_refclk_copy_0_ibufgds: IBUFGDS
148        generic map (
149           IBUF_LOW_PWR                          => FALSE,
150           IOSTANDARD                            => "LVDS_25")
151        port map (
152           O                                     => tfclk,
153           I                                     => MGT_REFCLK_COPY_0_P,
154           IB                                    => MGT_REFCLK_COPY_0_N
155        );
156     sfp_refclk0_ibufds_gtxe1: ibufds_gtxe1
157        port map (
158           O                                     => gtxrefclk,
159           ODIV2                                 => open,
160           CEB                                   => '0',
161           I                                     => SFP_REFCLK0_P,
162           IB                                    => SFP_REFCLK0_N
163        );
164     -- General Reset:
165     generalReset: entity work.reset_ctrl
166        port map (
167           CLK_I                                 => mclk0,
168           EXT_RESET1_B_I                        => POWER_ON_RESET_B,
169           EXT_RESET2_B_I                        => '1',
170           RST_O                                 => general_reset
171        );
```

glib_be.vhd - page 3

```
172     -- CDCE Synchronizer:
173     cdceSynchronizer: entity work.cdce_synchronizer
174        port map (
175           RESET_I                                 => general_reset,
176           IPBUS_CTRL_I                            => '0',  -- '0' -> User Control
177           IPBUS_SEL_I                             => '1',
178           IPBUS_PWRDOWN_I                         => '1',
179           IPBUS_SYNC_I                            => '1',
180           USER_SEL_I                              => '1',  -- '1' -> Pri Sel
181           USER_SYNC_I                             => '1',  -- '1' -> No Manual Sync
182           USER_PWRDOWN_I                          => '1',  -- '1' -> No Manual PWR_DOWN
183           PRI_CLK_I                               => mclk0,
184           SEC_CLK_I                               => '0',
185           PWRDOWN_O                               => CDCE_PWR_DOWN,
186           SYNC_O                                  => CDCE_SYNC ,
187           REF_SEL_O                               => CDCE_REF_SEL,
188           SYNC_CLK_O                              => open,
189           SYNC_CMD_O                              => open,
190           SYNC_BUSY_O                             => open,
191           SYNC_DONE_O                             => cs_sync_done
192     );
193     -- GBT Link Reset Control:
194     gbtLinkResetControl: entity work.glibLink_rst_ctrl
195        generic map (
196           BE_or_FE                                => "BE")
197        port map (
198           CLK_I                                   => mclk0,
199           RESET_I                                 => general_reset,
200           GTX_TXRESET_O                           => gtxTx_reset,
201           GTX_RXRESET_O                           => gtxRx_reset,
202           GBT_TXRESET_O                           => gbtTx_reset,
203           GBT_RXRESET_O                           => gbtRx_reset,
204           CLOCK_SWAP_O                            => open,
205           BUSY_O                                  => open,
206           DONE_O                                  => open
207        );
208     -- Pattern Generator:
209     patternGenerator: entity work.pattern_generator
210        port map (
211           RESET_I                                 => not cs_sync_done,
212           CLK_I                                   => tfclk,
213           FRAME_O                                 => tx_frame
214        );
215     -- TX Frame Pattern Match Flag:
216     txFramePatternMatchFlag: entity work.patternMatchFlag
217        port map (
218           RESET_I                                 => general_reset,
219           CLK_I                                   => tfclk,
220           MATCHDATA_I                             => tx_frame,
221           MATCHFLAG_O                             => tx_frame_flag
222        );
223     -- GBT Link:
224     gbtTx: entity work.GBT_tx
225        port map (
226           TX_RESET_I                              => gbtTx_reset,
227           TX_WORD_CLK_I                           => twclk,
228           TX_FRAME_CLK_I                          => tfclk,
229           TX_DATA_I                               => tx_frame,
230           TX_FRAME_O                              => open,
231           TX_WORD_O                               => txWord_from_gbtTx,
232           TX_HEADER_O                             => open
233        );
```

```
234     gtx: entity work.gtxWrapper
235        PORT MAP(
236           GTX_TX_REFCLK_I                         => gtxrefclk,
237           GTX_RX_REFCLK_I                         => gtxrefclk,
238           GTX_TX_WORDCLK_O                        => twclk,
239           GTX_RX_WORDCLK_O                        => rwclk,
240           GTX_TX_RESET_I                          => gtxTx_reset,
241           GTX_RX_RESET_I                          => gtxRx_reset,
242           GTX_TX_SYNC_RESET_I                     => '0',
243           GTX_RX_SYNC_RESET_I                     => '0',
244           GTX_RESETDONE_O                         => open,
245           GTX_PHASEALINGDONE_O                    => gtx_aligned,
246           GTX_RX_SLIDE_NBR_I                      => bit_slip_nbr,
247           GTX_RX_SLIDE_RUN_I                      => gbtRx_aligned,
248           GTX_LOOPBACK_I                          => "000",
249           GTX_TX_POWERDOWN_I                      => "00",
250           GTX_RX_POWERDOWN_I                      => "00",
251           GTX_TX_DATA_I                           => txWord_from_gbtTx,
252           GTX_RX_DATA_O                           => rxWord_from_gtxRx,
253           GTX_TXP_O                               => SFP1_TXP,
254           GTX_TXN_O                               => SFP1_TXN,
255           GTX_RXP_I                               => SFP1_RXP,
256           GTX_RXN_I                               => SFP1_RXN
257        );
258     phaseAligner: entity work.phase_aligner
259        PORT MAP (
260           RESET_I                                 => not gbtRx_aligned,
261           CLK240_I                                => rwclk,
262           CLK40_O                                 => rfclk,
263           CLK80_O                                 => open,
264           CLK120A_O                               => open,
265           CLK120B_O                               => open,
266           CLK160A_O                               => open,
267           CLK160B_O                               => open,
268           CLK240_O                                => open,
269           SYNC_I                                  => header_flag,
270           DONE_O                                  => done_from_phaseAligner
271        );
272     gbtRx: entity work.GBT_rx
273        PORT MAP (
274           RX_RESET_I                              => gbtRx_reset,
275           RX_WORD_CLK_I                           => rwclk,
276           RX_FRAME_CLK_I                          => rfclk,
277           RX_WORD_I                               => rxWord_from_gtxRx,
278           RX_DATA_O                               => rx_frame,
279           RX_DATA_DV_O                            => open,
280           RX_GTX_ALIGNED_I                        => gtx_aligned,
281           RX_BIT_SLIP_CMD_O                       => open,
282           RX_BIT_SLIP_NBR_O                       => bit_slip_nbr,
283           RX_ALIGNED_O                            => gbtRx_aligned,
284           RX_WRITE_ADDRESS_O                      => open,
285           RX_WORD_O                               => open,
286           RX_FRAME_O                              => open,
287           RX_FRAME_DV_O                           => open,
288           RX_HEADER_FLAG_O                        => header_flag,
289           RX_SHIFTEDWORD_O                        => open
290        );
```

```
291      -- RX Frame Pattern Match Flag:
292      rxFramePatternMatchFlag: entity work.patternMatchFlag
293         port map (
294            RESET_I                                 => not gbtRx_aligned,
295            CLK_I                                   => rfclk,
296            MATCHDATA_I                             => rx_frame,
297            MATCHFLAG_O                             => rx_frame_flag
298         );
299      BitErrorRatioTest: entity work.bert
300         PORT MAP (
301            TX_CLK_I                                => tfclk,
302            RX_CLK_I                                => rfclk,
303            TX_I                                    => x"000000000" & tx_frame,
304            RX_I                                    => x"000000000" & rx_frame,
305            IF_CLK_I                                => tfclk,
306            RESET_I                                 => general_reset,
307            ENABLE_I                                => done_from_phaseAligner,
308            LOAD_I                                  => '1',
309            LATENCY_I                               => "01011", -- 11d
310            NUMBER_OF_WORDS_O(31 downto 0)          => num_of_frames,
311            NUMBER_OF_WORDS_O(63 downto 32)         => open,
312            NUMBER_OF_WORD_ERRORS_O (31 downto 0)   => num_of_errors,
313            NUMBER_OF_WORD_ERRORS_O (63 downto 32)  => open,
314            WORD_ERROR_O                            => error_flag
315         );
316      -- Output buffers:
317      smaOut_generate: for i in 0 to 2 generate
318         smaOut_obuf: OBUF
319            generic map (
320               DRIVE                                => 24,
321               IOSTANDARD                           => "LVCMOS25",
322               SLEW                                 => "FAST")
323            port map (
324               O                                    => from_obuf(i),
325               I                                    => to_obuf(i)
326            );
327      end generate;
328      --================================================================--
329   end structural;
330   --=================================================================================================--
331   --=================================================================================================--
```

glib_be.vhd - page 6

glib_fe.vhd

```
1  ----------------------------------------------------------------------------------------------------
2  ------------------------------------ Module Information ------------------------------------------
3  ----------------------------------------------------------------------------------------------------
4  --                                                                                                --
5  -- Company:                CERN (PH-ESE-BE)                                                       --
6  -- Engineer:               Manoel Barros Marin (manoel.barros.marin@cern.ch) (m.barros@ieee.org)   --
7  --                                                                                                --
8  -- Create Date:            10/12/2011                                                             --
9  -- Project Name:           BE/FE Link simulation                                                  --
10 -- Module Name:            glib_be                                                                --
11 --                                                                                                --
12 -- Languaje:               VHDL'93                                                                --
13 --                                                                                                --
14 -- Target Devices:         GLIB(Virtex 6)                                                         --
15 -- Tool versions:          ISE 13.2                                                               --
16 --                                                                                                --
17 -- Revision:               1.0                                                                    --
18 --                                                                                                --
19 -- Additional Comments:                                                                           --
20 --                                                                                                --
21 ----------------------------------------------------------------------------------------------------
22 ----------------------------------------------------------------------------------------------------
23 -- IEEE VHDL standard library:
24 library ieee;
25 use ieee.std_logic_1164.all;
26 use ieee.numeric_std.all;
27 -- Xilinx devices library:
28 library unisim;
29 use unisim.vcomponents.all;
30 ----------------------------------------------------------------------------------------------------
31 ------------------------------------- Module Body ------------------------------------------------
32 ----------------------------------------------------------------------------------------------------
33 entity glib_be is
34    port (
35       -- Reset:
36       POWER_ON_RESET_B                        : in  std_logic;
37       -- Clock:
38       MCLK_0_P                                : in  std_logic;
39       MCLK_0_N                                : in  std_logic;
40       MGT_REFCLK_COPY_0_P                     : in  std_logic;
41       MGT_REFCLK_COPY_0_N                     : in  std_logic;
42       SFP_REFCLK0_P                           : in  std_logic;
43       SFP_REFCLK0_N                           : in  std_logic;
44       -- Clock Synthesizer:
45       CDCE_PWR_DOWN                           : out std_logic;
46       CDCE_SYNC                               : out std_logic;
47       CDCE_REF_SEL                            : out std_logic;
48       CDCE_SPI_CLK                            : out std_logic;
49       CDCE_SPI_LE                             : out std_logic;
50       CDCE_SPI_MOSI                           : out std_logic;
51       CDCE_SPI_MISO                           : in  std_logic;
52       CDCE_PLL_LOCK                           : in  std_logic;
53       -- GTX transceiver:
54       SFP1_TXN                                : out std_logic;
55       SFP1_TXP                                : out std_logic;
56       SFP1_RXN                                : in  std_logic;
57       SFP1_RXP                                : in  std_logic;
58       -- FMC#2 LED:
59       FMC2_LA18_P_CC                          : out std_logic;
60       -- SMA outputs:
61       SMA_ON_GLIB                             : out std_logic;
62       AMC_P12_TX_P                            : out std_logic;
63       AMC_P13_TX_P                            : out std_logic;
```

```
64        -- Dummy:
65        DUMMY_PORT                                : out std_logic
66        --============================================================================--
67     );
68  end glib_be;
69  architecture structural of glib_be is
70     --============================= Declarations ==============================--
71     attribute S : string;   -- This attribute is for avoiding signal trimming.
72     -- General Reset:
73     signal general_reset                         : std_logic;
74     -- Clocks:
75     signal mclk0                                 : std_logic;
76     signal gtxrefclk                             : std_logic;
77     signal tfclk                                 : std_logic;
78     signal twclk                                 : std_logic;
79     signal rwclk                                 : std_logic;
80     signal rfclk                                 : std_logic;
81     -- CDCE Synchronizer:
82     signal cs_sync_done                          : std_logic;
83     -- GBT Link Reset Control:
84     signal gtxTx_reset                           : std_logic;
85     signal gtxRx_reset                           : std_logic;
86     signal gbtTx_reset                           : std_logic;
87     signal gbtRx_reset                           : std_logic;
88     -- Pattern Generator:
89     signal tx_frame                              : std_logic_vector(83 downto 0);
90     -- GBT TX:
91     signal txWord_from_gbtTx                     : std_logic_vector(19 downto 0);
92     -- GTX TX:
93     signal rxWord_from_gtxRx                     : std_logic_vector(19 downto 0);
94     signal gtx_aligned                           : std_logic;
95     -- GBT RX:
96     signal gbtRx_aligned                         : std_logic;
97     signal bit_slip_nbr                          : std_logic_vector(4 downto 0);
98     signal header_flag                           : std_logic;
99     signal rx_frame                              : std_logic_vector(83 downto 0);
100    -- Pattern Checker:
101    signal num_of_frames                         : std_logic_vector(31 downto 0);
102    attribute S of num_of_frames                 : signal is "true";
103    signal num_of_errors                         : std_logic_vector(31 downto 0);
104    attribute S of num_of_errors                 : signal is "true";
105    signal error_flag                            : std_logic;
106    attribute S of error_flag                    : signal is "true";
107    -- Pattern Match Flags:
108    signal tx_frame_flag                         : std_logic;
109    signal rx_frame_flag                         : std_logic;
110    -- RX Phase Aligner:
111    signal done_from_phaseAligner                : std_logic;
112    -- Output buffers:
113    signal to_obuf                               : std_logic_vector(2 downto 0);
114    signal from_obuf                             : std_logic_vector(2 downto 0);
115    --============================================================================--
116 --============================================================================--
117 -----      --=========================================================--
118 begin    --================== Architecture Body ==================--
119 -----      --=========================================================--
120 --============================================================================--
121    --=========================  Port Assignments  =========================--
122    -- Clock Synthesizer:
123    CDCE_SPI_CLK                                 <= '0';
124    CDCE_SPI_LE                                  <= '1';
125    CDCE_SPI_MOSI                                <= '0';
```

```
126   -- Output buffers (24mA, LVCMOS25, Fast Slew Rate):
127   SMA_ON_GLIB     <= from_obuf(0);  to_obuf(0) <= mclk0;
128   AMC_P12_TX_P    <= from_obuf(1);  to_obuf(1) <= tx_frame_flag;
129   AMC_P13_TX_P    <= from_obuf(2);  to_obuf(2) <= rx_frame_flag;
130   -- Dummy signals:
131   DUMMY_PORT      <= '1' when  num_of_frames = x"FFFFFFFF" or
132                                num_of_errors = x"FFFFFFFF" or
133                                error_flag    = '1'
134                     else '0';
135   --================================================================--
136   --====================== Component Instantiations ======================--
137   -- Clock Buffers:
138   mclk0_ibufgds: IBUFGDS
139      generic map (
140         IBUF_LOW_PWR                          => FALSE,
141         IOSTANDARD                            => "LVDS_25")
142      port map (
143         O                                     => mclk0,
144         I                                     => MCLK_0_P,
145         IB                                    => MCLK_0_N
146      );
147   mgt_refclk_copy_0_ibufgds: IBUFGDS
148      generic map (
149         IBUF_LOW_PWR                          => FALSE,
150         IOSTANDARD                            => "LVDS_25")
151      port map (
152         O                                     => tfclk,
153         I                                     => MGT_REFCLK_COPY_0_P,
154         IB                                    => MGT_REFCLK_COPY_0_N
155      );
156   sfp_refclk0_ibufds_gtxe1: ibufds_gtxe1
157      port map (
158         O                                     => gtxrefclk,
159         ODIV2                                 => open,
160         CEB                                   => '0',
161         I                                     => SFP_REFCLK0_P,
162         IB                                    => SFP_REFCLK0_N
163      );
164   -- General Reset:
165   generalReset: entity work.reset_ctrl
166      port map (
167         CLK_I                                 => mclk0,
168         EXT_RESET1_B_I                        => POWER_ON_RESET_B,
169         EXT_RESET2_B_I                        => '1',
170         RST_O                                 => general_reset
171      );
```

glib_fe.vhd - page 3

```
172    -- CDCE Synchronizer:
173    cdceSynchronizer: entity work.cdce_synchronizer
174       port map (
175          RESET_I                                 => general_reset,
176          IPBUS_CTRL_I                            => '0',  -- '0' -> User Control
177          IPBUS_SEL_I                             => '1',
178          IPBUS_PWRDOWN_I                         => '1',
179          IPBUS_SYNC_I                            => '1',
180          USER_SEL_I                              => '1',  -- '1' -> Pri Sel
181          USER_SYNC_I                             => '1',  -- '1' -> No Manual Sync
182          USER_PWRDOWN_I                          => '1',  -- '1' -> No Manual PWR_DOWN
183          PRI_CLK_I                               => mclk0,
184          SEC_CLK_I                               => '0',
185          PWRDOWN_O                               => CDCE_PWR_DOWN,
186          SYNC_O                                  => CDCE_SYNC ,
187          REF_SEL_O                               => CDCE_REF_SEL,
188          SYNC_CLK_O                              => open,
189          SYNC_CMD_O                              => open,
190          SYNC_BUSY_O                             => open,
191          SYNC_DONE_O                             => cs_sync_done
192    );
193    -- GBT Link Reset Control:
194    gbtLinkResetControl: entity work.glibLink_rst_ctrl
195       generic map (
196          BE_or_FE                                => "BE")
197       port map (
198          CLK_I                                   => mclk0,
199          RESET_I                                 => general_reset,
200          GTX_TXRESET_O                           => gtxTx_reset,
201          GTX_RXRESET_O                           => gtxRx_reset,
202          GBT_TXRESET_O                           => gbtTx_reset,
203          GBT_RXRESET_O                           => gbtRx_reset,
204          CLOCK_SWAP_O                            => open,
205          BUSY_O                                  => open,
206          DONE_O                                  => open
207       );
208    -- Pattern Generator:
209    patternGenerator: entity work.pattern_generator
210       port map (
211          RESET_I                                 => not cs_sync_done,
212          CLK_I                                   => tfclk,
213          FRAME_O                                 => tx_frame
214       );
215    -- TX Frame Pattern Match Flag:
216    txFramePatternMatchFlag: entity work.patternMatchFlag
217       port map (
218          RESET_I                                 => general_reset,
219          CLK_I                                   => tfclk,
220          MATCHDATA_I                             => tx_frame,
221          MATCHFLAG_O                             => tx_frame_flag
222       );
223    -- GBT Link:
224    gbtTx: entity work.GBT_tx
225       port map (
226          TX_RESET_I                              => gbtTx_reset,
227          TX_WORD_CLK_I                           => twclk,
228          TX_FRAME_CLK_I                          => tfclk,
229          TX_DATA_I                               => tx_frame,
230          TX_FRAME_O                              => open,
231          TX_WORD_O                               => txWord_from_gbtTx,
232          TX_HEADER_O                             => open
233       );
```

```
234     gtx: entity work.gtxWrapper
235        PORT MAP(
236           GTX_TX_REFCLK_I                         => gtxrefclk,
237           GTX_RX_REFCLK_I                         => gtxrefclk,
238           GTX_TX_WORDCLK_O                        => twclk,
239           GTX_RX_WORDCLK_O                        => rwclk,
240           GTX_TX_RESET_I                          => gtxTx_reset,
241           GTX_RX_RESET_I                          => gtxRx_reset,
242           GTX_TX_SYNC_RESET_I                     => '0',
243           GTX_RX_SYNC_RESET_I                     => '0',
244           GTX_RESETDONE_O                         => open,
245           GTX_PHASEALINGDONE_O                    => gtx_aligned,
246           GTX_RX_SLIDE_NBR_I                      => bit_slip_nbr,
247           GTX_RX_SLIDE_RUN_I                      => gbtRx_aligned,
248           GTX_LOOPBACK_I                          => "000",
249           GTX_TX_POWERDOWN_I                      => "00",
250           GTX_RX_POWERDOWN_I                      => "00",
251           GTX_TX_DATA_I                           => txWord_from_gbtTx,
252           GTX_RX_DATA_O                           => rxWord_from_gtxRx,
253           GTX_TXP_O                               => SFP1_TXP,
254           GTX_TXN_O                               => SFP1_TXN,
255           GTX_RXP_I                               => SFP1_RXP,
256           GTX_RXN_I                               => SFP1_RXN
257        );
258     phaseAligner: entity work.phase_aligner
259        PORT MAP (
260           RESET_I                                 => not gbtRx_aligned,
261           CLK240_I                                => rwclk,
262           CLK40_O                                 => rfclk,
263           CLK80_O                                 => open,
264           CLK120A_O                               => open,
265           CLK120B_O                               => open,
266           CLK160A_O                               => open,
267           CLK160B_O                               => open,
268           CLK240_O                                => open,
269           SYNC_I                                  => header_flag,
270           DONE_O                                  => done_from_phaseAligner
271        );
272     gbtRx: entity work.GBT_rx
273        PORT MAP (
274           RX_RESET_I                              => gbtRx_reset,
275           RX_WORD_CLK_I                           => rwclk,
276           RX_FRAME_CLK_I                          => rfclk,
277           RX_WORD_I                               => rxWord_from_gtxRx,
278           RX_DATA_O                               => rx_frame,
279           RX_DATA_DV_O                            => open,
280           RX_GTX_ALIGNED_I                        => gtx_aligned,
281           RX_BIT_SLIP_CMD_O                       => open,
282           RX_BIT_SLIP_NBR_O                       => bit_slip_nbr,
283           RX_ALIGNED_O                            => gbtRx_aligned,
284           RX_WRITE_ADDRESS_O                      => open,
285           RX_WORD_O                               => open,
286           RX_FRAME_O                              => open,
287           RX_FRAME_DV_O                           => open,
288           RX_HEADER_FLAG_O                        => header_flag,
289           RX_SHIFTEDWORD_O                        => open
290        );
```

```
291     -- RX Frame Pattern Match Flag:
292     rxFramePatternMatchFlag: entity work.patternMatchFlag
293        port map (
294           RESET_I                              => not gbtRx_aligned,
295           CLK_I                                => rfclk,
296           MATCHDATA_I                          => rx_frame,
297           MATCHFLAG_O                          => rx_frame_flag
298        );
299     BitErrorRatioTest: entity work.bert
300        PORT MAP (
301           TX_CLK_I                             => tfclk,
302           RX_CLK_I                             => rfclk,
303           TX_I                                 => x"000000000" & tx_frame,
304           RX_I                                 => x"000000000" & rx_frame,
305           IF_CLK_I                             => tfclk,
306           RESET_I                              => general_reset,
307           ENABLE_I                             => done_from_phaseAligner,
308           LOAD_I                               => '1',
309           LATENCY_I                            => "01011", -- 11d
310           NUMBER_OF_WORDS_O(31 downto 0)       => num_of_frames,
311           NUMBER_OF_WORDS_O(63 downto 32)      => open,
312           NUMBER_OF_WORD_ERRORS_O (31 downto 0)  => num_of_errors,
313           NUMBER_OF_WORD_ERRORS_O (63 downto 32) => open,
314           WORD_ERROR_O                         => error_flag
315        );
316     -- Output buffers:
317     smaOut_generate: for i in 0 to 2 generate
318        smaOut_obuf: OBUF
319           generic map (
320              DRIVE                             => 24,
321              IOSTANDARD                        => "LVCMOS25",
322              SLEW                              => "FAST")
323           port map (
324              O                                 => from_obuf(i),
325              I                                 => to_obuf(i)
326           );
327     end generate;
328     --================================================================--
329  end structural;
330  --=================================================================================================--
331  --=================================================================================================--
```

glib_fe.vhd - page 6

Appendix D - Content of the Compact Disc

- **pfc_itie_uned_manoel_barros_marin.pdf -** File that includes a printable version of this thesis (PFC).
- **BE_FE_link_simulation_ISE_projects\ -** This folder contains the ISE projects and the HDL files of the BE/FE link simulation.
  - **glib_be_project\ -** This folder contains the ISE project and the HDL files utilised on the BE simulation.
  - **glib_fe_project\ -** This folder contains the ISE project and the HDL files utilised on the FE simulation.
- **GLIB documents\ -** This folder contains files and documents related to the development of the GLIB.
  - **Assembly\ -** This folder contains the schematics, documents and files related to the assembly of the GLIB
  - **Board manufacturing\ -** This folder contains the schematics, documents and files related to the manufacturing of the PCB of the GLIB.
  - **Component datasheets\ -** This folder contains the datasheets of the integrated circuits of the GLIB.
  - **Mechanical parts\ -** This folder contains the schematics and files related to the mechanical parts of the GLIB.
  - **Schematics\ -** This folder contains the schematics of the component interconnection of the GLIB in A3 format.