OULUN YLIOPISTO
UNIVERSITY of OULU

DEGREE PROGRMME IN ELECTRICAL ENGINEERING

MASTER'S THESIS

Implementation of the specification and schematics design for the Ethernet Fronthaul Module

Author	Tomi Parkkonen
Supervisor	Antti Mäntyniemi
Second Examiner	Kari Määttä
(Technical advisor	Markku Järvi)

September 2015

Parkkonen T. (2015) Implementation of the specification and schematics design for the Ethernet Fronthaul Module. University of Oulu, Department of Electrical Engineering, Degree Programme in Electrical Engineering. Master's Thesis, 61 p.

ABSTRACT

This Master's Thesis covers theory and implementation of a device which is designed using a small base station as a reference. The theory chapter consists of the description and theory of a cloud radio access network architecture, a high data rate interface, an active antenna system and a designed device itself. This theory chapter is used to give reasons why the device is designed. The implementation chapter is divided into two chapters, which explains how the implementation specification is done and how the schematics were drawn. The schematics chapter covers the modifications, which are done to the hardware of the original small base station.

Keywords: base station, cloud radio access network, active antenna system, implementation specification, schematics.

Parkkonen T. (2015) Ethernet Fronthaul Moduulin implementointispesifikaation ja piirikaavion suunnittelu. Oulun yliopisto, sähkötekniikan osasto, sähkötekniikan koulutusohjelma. Diplomityö, 61 s.

TIIVISTELMÄ

Tämä Diplomityö käsittelee pienen tukiasemalaitteen pohjalta suunniteltavan laitteen, siihen liittyvän teorian sekä toteutuksen. Laitteeseen liittyvä teoria muodostuu neljästä kappaleesta, jotka käsittelevät cloud radio access network -arkkitehtuuria, nopean data määrän rajapintaa, aktiivi antenni systeemiä sekä itse suunnitellun laitteen teoriaa. Teorialla pyritään pohjustamaan syitä siihen, minkä vuoksi kyseinen laite on haluttu toteuttaa. Laitteen toteutusta käsittelevä kappale on jaettu kahteen osioon, joissa kuvataan implementointispesifikaation toteutus ja piirikaavioiden piirto. Piirikaavio kappaleessa käsitellään muutokset, jotka on tehty pohjana käytettävän tukiaseman laitteistolle.

Avainsanat: tukiasema, cloud radio access network, aktiivi antenni systeemi, implementointispesifikaatio, piirikaavio.

TABLE OF CONTENTS

FOREWORD

This Master's thesis objective was to develop a new kind of prototype device, which acts as a platform for the wireless telecommunication's hardware and software development. During implementing the work, many references were used, for example China Mobile Research Institute, Nokia Flexi Multiradio Active Antenna System [1, 7].

The schematic drawing was made in the co-operation with Markku Järvi and Mikko Keränen. Markku drew half of the designed device's schematics and Mikko drew the power supply schematics.

I would like to thank Markku Järvi for supervising and advising me during the whole work process.

Oulu 25.9.2015

Tomi Parkkonen

LIST OF ABBREVIATIONS AND SYMBOLS

AAS	Active Antenna System
AC	Alternate Current
AE	Active Element
BBU	Baseband Unit
BF	Beam Forming
BTS	Base Transceiver Station
CM	Common Module
CPLD	Complex Programmable Logic Device
CPRI	Common Public Radio Interface
C-RAN	Cloud Radio Access Network
CSI	Channel State Information
DC	Direct Current
DDR3	Double Data Rate 3
DSP	Digital Signal Processor
ECC	Error-Correction Code
EEPROM	Electronically Erasable Programmable Read-Only Memory
EFM	Ethernet Fronthaul Module
FPGA	Field Programmable Gate Array
FZM	Flexi Zone Micro
GPIO	General Purpose Input/Output
GPS	Global Positioning System
HDMI	High Definition Multimedia Interface
HW	Hardware
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
LAN	Local Area Network
LMTS	Local Management Trouble Shooting
LTE	Long-Term Evolution
MBB	Mobile Broadband
MDIO	Media Data Input/Output
MII	Media Independent Interface
MIMO	Multiple-Input Multiple-Output
MUX	Multiplexer
OBSAI	Open Base Station Architecture Initiative
OCXO	Oven-Controlled Crystal Oscillator
O&M	Operation Management
PA-LNA	Power Amplifier-Low Noise Amplifier
PHY	Physical (layer)
PLL	Phase-Locked Loop
PPS	Pulse Per Second
QCFG	Quadrant Configuration

QSFP+	Quad Small Form-Factor Pluggable
RAN	Radio Access Network
RF	Radio Frequency
RJ45	Registered Jack 45
RP	Reference Point
RRH	Remote Radio Head
RTC	Real Time Clock
SDR	Software Defined Radio
SDRAM	Synchronous Dynamic Random Access Memory
SFP+	Enhanced Small Form-Factor Pluggable
SGMII	Serial Gigabit Media Independent Interface
SPD	Speed Select
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SRIO	Serial Rapid Input/Output
SW	Software
TD-LTE 8	Time Division Long-Term Evolution 8
TRX	Transceiver
VSWR	Voltage Standing Wave Ratio
UART	Universal Asynchronous Receiver/Transmitter
UE	User Equipment
USB	Universal Serial Bus

1. INTRODUCTION

Requirements for the wireless telecommunications have increased significantly over the years and are still increasing. The growing number of users forces the base station manufacturers to develop new solutions in order to provide higher speeds and capacity. This adds to the hardware and software requirements, which is not the easiest task to handle. The motivation for this Master's thesis work is to confront the increasing demands of the wireless telecommunications.

The scope of the Master's thesis work is to create the Implementation specification and draw schematics for a device that acts as a platform for the further hardware and software development. Requirements for the device are the following: The device provides a data conversion from Open Base Station Architecture Initiative Reference Point 3-01/Common Public Radio Interface (OBSAI RP3-01/CPRI) to the Ethernet and vice versa and L1 processing for the received/transmitted data. The device also supports level control and beam forming for the Active Antenna System's (AAS's) Active Elements (AEs).

The designed device is based on a small cell outdoor base transceiver station called Flexi Zone Micro (FZM). The reason for using the older design as a reference was the reuse of the FZM's digital hardware and software to save a large amount of work hours. The target of this Master's thesis work was to modify digital circuitry so that it fills the new requirements.

This Master's thesis covers the designed device called Ethernet Fronthaul Module (EFM) in two chapters. The first chapter covers the theory part, which explains the environment where the device is used, why and how it can be used and what it is capable of. The second chapter handles the implementation of the design. It consists of two parts; the first part is the Implementation specification, which explains the device's main components and how these components are connected to each other. The interfaces of the device to the outside world are also discussed in this part. The latter part covers how the schematics for the new parts were drawn and explains the functionality of the components.

2. THEORY

The theory part is divided in to four chapters that cover the Cloud Radio Access Network (C-RAN), the replacement of the 6/9 G OBSAI/CPRI with higher data rate interface, an AAS and the designed device called the EFM.

2.1. Cloud radio access network

The Cloud Radio Access Network is a new cellular network architecture based on the distributed Base Transceiver Station (BTS). The distributed BTS is composed of a Remote Radio Head (RRH) and a Baseband unit (BBU). By separating BTS functions differently between the RRH and the BBU, there are two ways to implement a C-RAN. The BTS functions are divided into three layers: physical layer (layer 1, L1), data link layer (layer 2, L2) and network layer (layer 3, L3), respectively. The first separating method is called "full centralization", where layers: 1, 2, 3 and Operation Management (O&M) are located in the BBU. The second one is "partial centralization", where layer 1 is included into the RRH instead. The separation methods of the BTS functions are presented in Figure 1. [1]

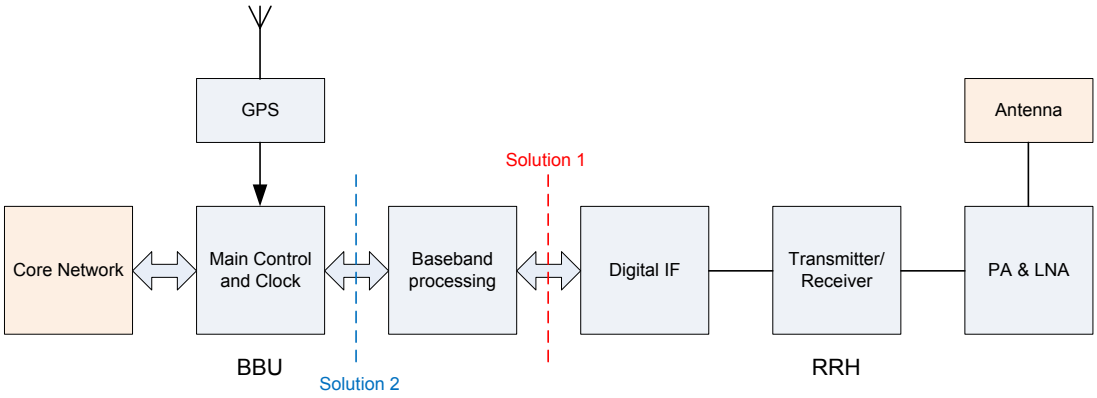

Figure 1. Separation methods of BTS functions.

The C-RAN has two architectures, which are based on the previously discussed splitting methods of the BTS functions. Both architectures have three main parts: the remote radio heads with antennas, the optical transport network that is used to connect the RRHs and the BBU pool with high bandwidth and low-latency. The BBU pool consists of programmable processors and real-time virtualization technology, like a Virtual Base Station. [1]

The "full centralization" architecture can be upgraded easily and its network capacity can be expanded. Other advantages are multi-standard operation support, maximum resource sharing and convenience into multi-cell collaborative signal processing. Carrying the baseband IQ signal between the BBU and the RRHs at high bandwidth is a major disadvantage. While using a Time Division Long-Term Evolution 8 (TD-LTE 8) antenna with bandwidth of 20 MHz, the need for transmission rate is 10 Gbps. This architecture is shown in Figure 2. [1]

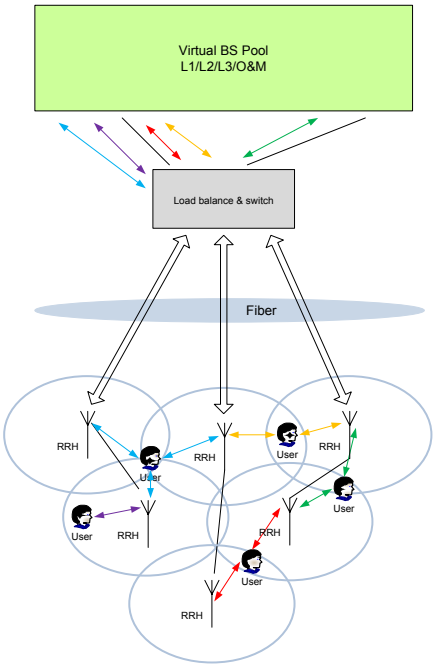

Figure 2. The full centralization C-RAN architecture.

In the "partial centralization" C-RAN architecture, the baseband processing is integrated into the RRH. Because the BBU-RRH connection carries only demodulated data, a much lower transmission bandwidth between the BBU pool and the RRHs is needed. The disadvantage of this architecture is with flexibility in upgrading and convenience into multi-cell collaborative signal processing. Figure 3 presents the partial centralization C-RAN architecture. [1]

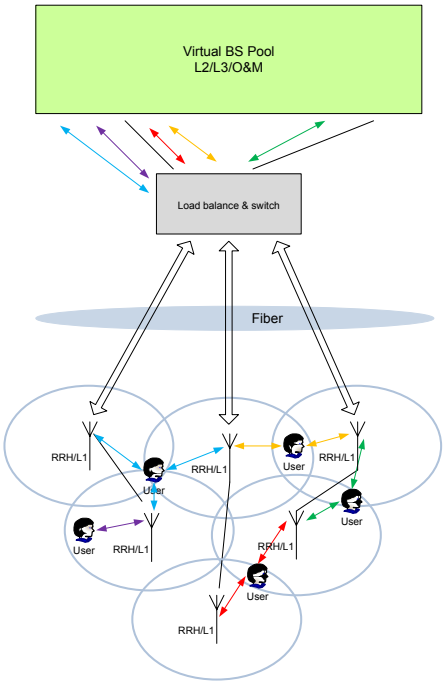

Figure 3. The partial centralization C-RAN architecture.

Both C-RAN architectures enable quick deployment and network upgrade. The only condition for network expansion and capacity improvement is to install the new RRHs and connect them into the BBU pool. Upgrading the BBU pools hardware increases processing capacity if network load grows. An easier way to develop Software Defined Radio (SDR), upgrade RAN and support multi-standard operation is to use the full centralization architecture with a combination of open platform and general purpose processors. [1]

The major difference between the traditional distributed BTS and the C-RAN is that the physical BBU pools virtual BTS can process any particular RRHs radio signals. This means that there are no fixed relationships between the BBUs and the RRHs. Virtualization technology is a great way to maximize a C-RAN system's flexibility. A virtual BTS's can co-operate with the physical BBU pool and it can work together with active User Equipments (UE's) sharing signals, traffic data and Channel State Information (CSI). [1]

2.2. High data rate interface

There are two BTS reference and specification providers in the Radio Access Network (RAN) development, OBSAI and CPRI. The reference architecture of BTS consists of four elements: functional blocks, an external network interface, an external radio interface and internal interfaces between the functional blocks of the BTS. This chapter concentrates on the OBSAI and the CPRI internal interfaces and replacing them with higher data rate interfaces. [2]

The base transceiver station's functional blocks are connected to each other through internal interfaces which are designated as a Reference Point 1 (RP1), Reference Point 2 (RP2), Reference Point 3 (RP3) and Reference Point 4 (RP4). Each of them has their own duty. The RP1 includes clock signals and control data to all blocks. The RP2 transports user data from the Transport Block to the Baseband Block. The RP3 provides transport for the data between the Baseband Block and the RF Block. The RP4 acts as the DC power interface providing a connection between internal modules and DC power sources. Figure 4 presents the BTS reference architecture with reference points. [2]

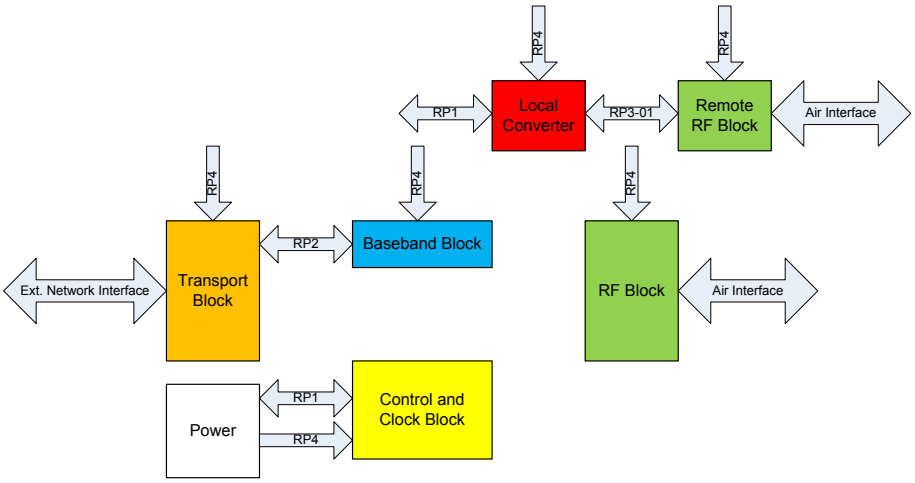

Figure 4. The BTS reference architecture.

The Reference Point 3 has an extended version called the RP3-01 which is designed for data transportation between the BTS and the RRHs. The RP3 and the RP3-01 have equivalent protocol and the same data rates from 768 Mbps to 6144 Mbps. The RP3-01 transfers also RP1 data which includes Ethernet and frame clock bursts and supports data transmission over longer distances than RP3. [3]

When the CPRI specification is used, the basic reference configuration, shown in Figure 5, consists of one BBU and one RRH which are connected with a single CPRI link. As can be seen, no local converter is needed like in OBSAI's case. The basic configuration can be extended into several different topologies like Star and Chain topologies. [4]

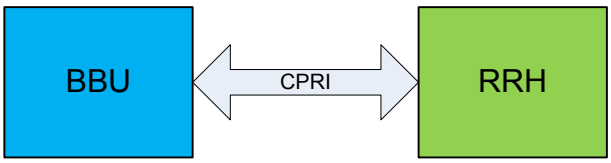

Figure 5. Single CPRI link between the BBU and the RRH.

In larger system configurations, which includes several antennas and carriers it is required that the IQ -data of a certain antenna and a certain antenna carrier is transferred through one CPRI link. In the CPRI specification, the amount of physical links is not restricted, so several CPRI links can be used simultaneously to enhance the system capacity. Figure 6 presents two ways to connect the multiple CPRI links between one BBU and one or several RRHs. The upper one is a basic topology and lower one is called a Star topology. [4]

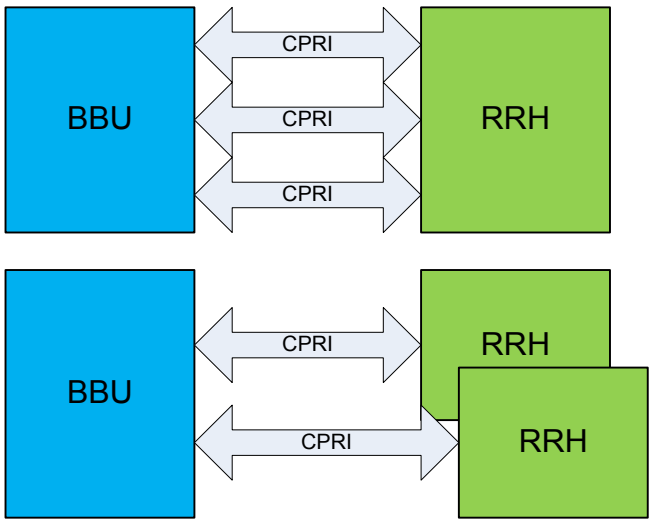

Figure 6. Multiple CPRI links.

There are three basic connection topologies and two topologies, where multiple BBUs are used. The basic ones are called the Chain, the Tree and the Ring – topologies. All five topologies are shown in Figure 7 and 8. [4]

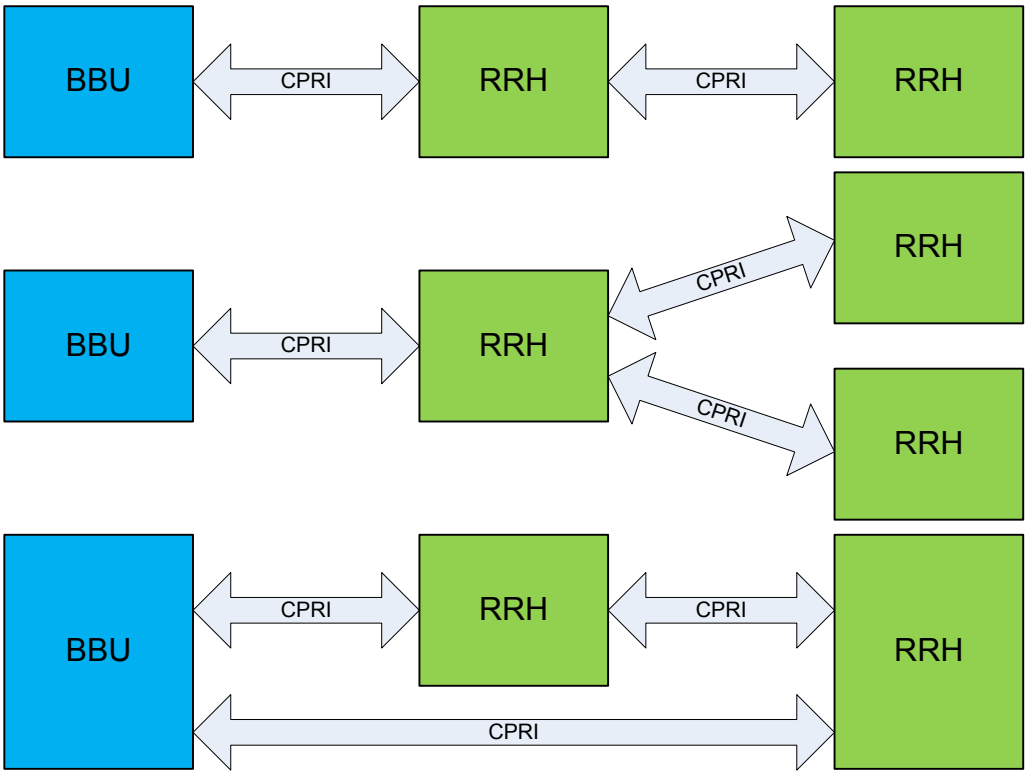

Figure 7. The Chain, Tree and Ring –topologies.

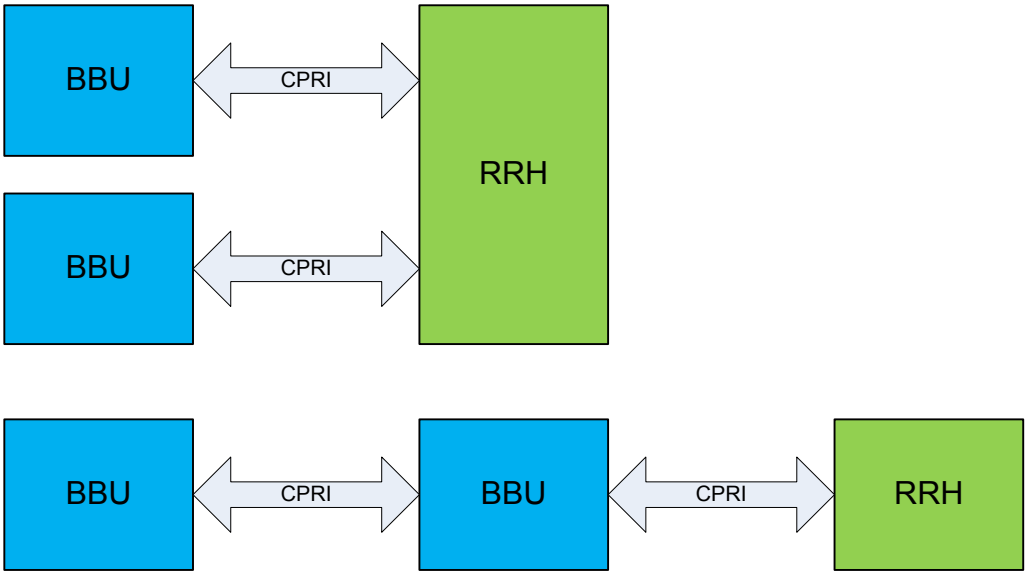

Figure 8. Multiple links with several BBUs and the Chain topology of multiple BBUs.

In the CPRI release 4.0 and higher releases, the BBUs can be used as a networking element, like in Figure 8, but the specification process is not yet complete. [4]

Development of the higher OBSAI/CPRI data rates is necessary because of the increasing demands for higher bandwidth and higher capacity of mobile networks. Currently OBSAI RP3-01 is specified for 6.144 Gbps but CPRI supports 10.1376 Gbps. Still, these speeds are not enough and higher data rates are needed in future solutions. Current OBSAI/CPRI data rates are presented in the Table 1. [3, 4]

Table 1. The OBSAI/CPRI data rates

Base Rate Multiplier	OBSAI (Gbps)	CPRI (Gbps)
1x	0.768	0.614
2x	1.536	1.228
4x	3.072	2.4576
5x	-	3.072
8x	6.144	4.9152
10x	-	6.144
16x	-	9.8304
20x	-	10.1376

One solution for achieving much higher data rates is to use of the Ethernet technology. Ethernet is part of the Local Area Network or (LAN) family. It is covered by the IEEE 802.3 standard. Ethernet is capable of supporting speeds from 1 Gbps to 100 Gbps. Basically, 100 Gbps Ethernet can be constructed from ten parallel 10 Gbps or four parallel 25 Gbps Ethernet paths. [8]

Ethernet paths can be constructed by using the Enhanced Small Form-factor Pluggable (SFP+) transceivers. The SFP+ is a small transceiver which can be used in telecommunications and data communications networks. It can connect switches, routers, fiber cables and also converts optical signal into electrical signal and vice versa. There is a high speed version of the SFP+ called the zSFP+, which can transfer data at 25 Gbps. The zSFP+ can be upgraded in to 40 Gbps. [9, 11]

The Quad Small Form-factor Pluggable (QSFP+) –interface includes four SFP+ transceivers and allows 40 Gbps speed for a data transfer while taking less area on the Printed Circuit Board (PCB). Like in the SFP+ case, there is also a zQSFP+ version of the QSFP+. The zQSFP+ offers four channels which support for data speeds from 25 Gbps up to 40 Gbps. This means that 100 Gbps to 160 Gbps data transfer speeds can be utilized in the future solutions. [9, 11]

2.3. Active antenna system

The active antenna system is an advanced base station platform which is designed to meet the operator requirements for Mobile Broadband (MBB) services. Before the AAS, beam directions were adjusted mechanically. In AAS Beam Forming (BF) is done electrically by phasing the data. The AAS supports Two-Dimensional and Three-Dimensional Multiple-Input Multiple-Output (2D/3D-MIMO) technology to maximize radio resources in both the micro- and the macro-spatial domains. The AAS's most important component is the Three-Dimensional Beam Forming (3D-BF). [5, 6]

The AAS combine the active transceiver array and the passive antenna array into one unit. This reduces cable losses and simplifies site engineering. Two- and three-

dimensional antenna array also enable linear, planar, circular and cylindrical configurations. An antenna element can be weighted adaptively or semi-adaptively. In the AAS, radiation patterns can be controlled freely on the vertical and the horizontal domains. [5, 6]

Electronic beam control allows many spatial processing techniques. Beams can be directed into different horizontal and vertical directions for different operations, frequency bands, network standards and link directions. The AAS follows cells traffic variations and adapts to them providing high efficiency. Forming multiple cells is also possible by separating user equipments to higher and lower data rate cells which increases system capacity. The different AAS beam forming applications are presented in Figure 9. [5, 6]

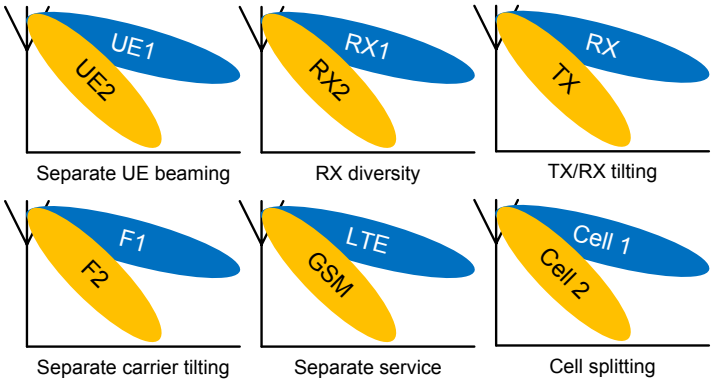

Figure 9. The AAS beaming applications.

As previously mentioned, the 3D-BF is the key component of the AAS technology. The Two-Dimensional Beam Forming (2D-BF) can only cover the horizontal or the vertical domain at a time, so in the horizontal beam forming case, the users who are far away from the main beam will suffer transmission losses in the vertical domain and vice versa. When the 3D-BF is used, both the horizontal and the vertical domains are covered. Beams can be directed on those users who suffer transmission loss in the case of the 2D-BF. This allocates beams for different users. [5, 6]

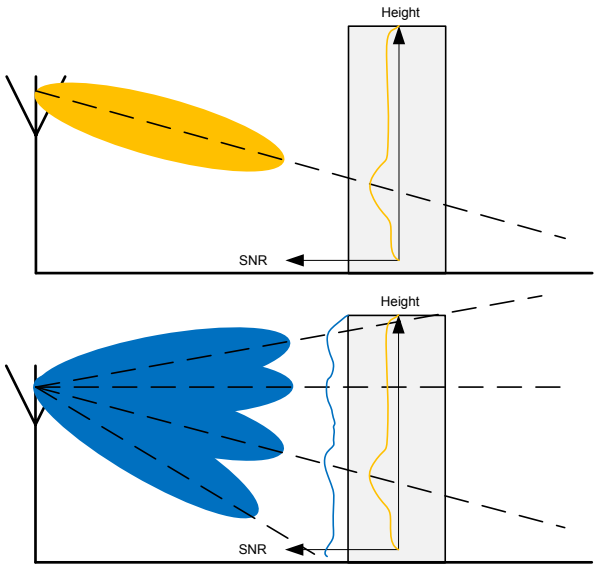

Figure 10. The vertical 2D- and 3D-beam forming.

Figure 11 presents both the 2D- and the 3D-BF view from above. In the 3D-BF, beam can be steered towards the users. This improves user experience and offers much higher quality services. [5, 6]

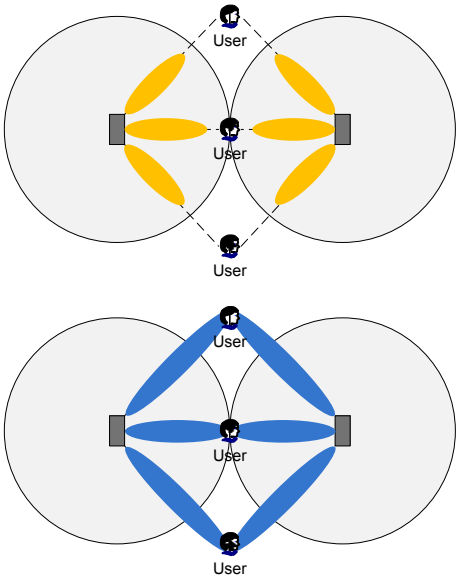

Figure 11. Horizontal- and vertical beam forming.

An AAS can include one or several AEs and one Common Module (CM). Every AE includes digital parts and several Transceiver (TRX) links. The CM controls all the AEs, providing beam forming and module level control. The active elements, common module and baseband unit are connected together using OBSAI/CPRI/Ethernet –fiber cable. Figure 12 in chapter 2.4 presents an AAS hierarchy.

2.4. Ethernet Fronthaul Module

Figure 12 presents how the EFM acts as an interface between the four active antennas (AAs) and the BBU. It provides the OBSAI/CPRI to Ethernet and vice versa conversions for the IQ data, Long-Term Evolution Layer 1 (LTE L1) support and also beam forming and module level control for active antenna modules. It is connected to the active antennas with an OBSAI/CPRI fiber and the connection towards baseband unit is done with an Ethernet fiber.

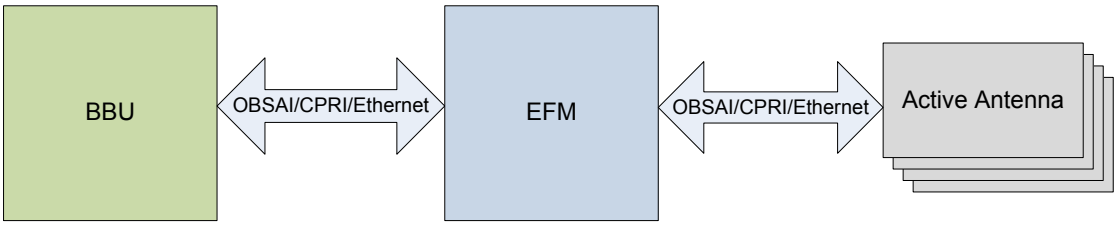

Figure 12. The EFM between the BBU and the AA.

The EFM is based on the FZM all-in one base station which is designed for outdoor and harsh indoor environments. It can act as a stand-alone base station supporting most commonly used frequency bands [6]. The FZM consists of the following parts: digital and small signal radio frequency -circuits, a Power Amplifier-Low Noise Amplifier (PA-LNA) board, a duplexer and Voltage Standing Wave Ratio (VSWR) module, a Global Positioning System (GPS) module and an Alternate Current/Direct Current (AC/DC) power supply module with primary and secondary surges. Figure 13 presents the above-mentioned FZM parts.

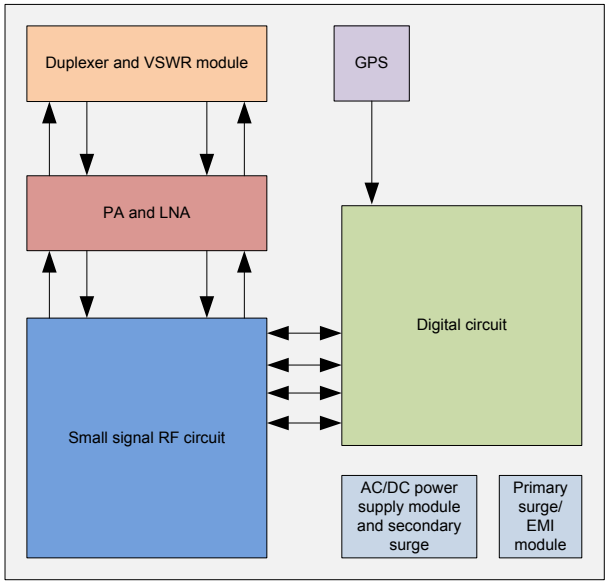

Figure 13. The FZM base station.

The EFM reuses digital circuit, GPS module and software of FZM. Remaining parts are not used. The digital circuit is modified by adding one Field Programmable Gate Array (FPGA) and other needed components; such as Phase-Locked Loop – synthesizers (PLL), Double Data Rate 3 Synchronous Dynamic Random Access

Memory (DDR3 SDRAM) and Flash -memories. The GPS module will stay intact. The EFM uses a 12 V input DC voltage so AC/DC power supply and secondary surge are not needed. Basic block diagram of the EFM is presented in Figure 14.

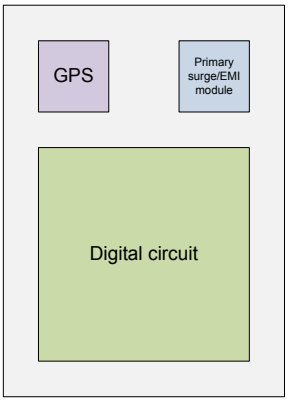

Figure 14. The EFM block diagram.

The main parts of the EFM's digital circuit are two FPGAs, a Digital Signal Processor (DSP), a Complex Programmable Logic Device (CPLD) and a block for clock generation and synchronization. All these parts have different functions on the EFM. The first FPGA handles basic functions like translating voltage levels and supporting different interfaces. Digital signal processing is done with the DSP. The CPLD mainly controls all reset signals and synchronization clock generation block generates clock for the rest of the components on board. The second FPGA is used to do all IQ data conversion and active antenna functions. The LTE L1 processing is done together with DSP. The digital circuits block diagram is presented in Figure 15.

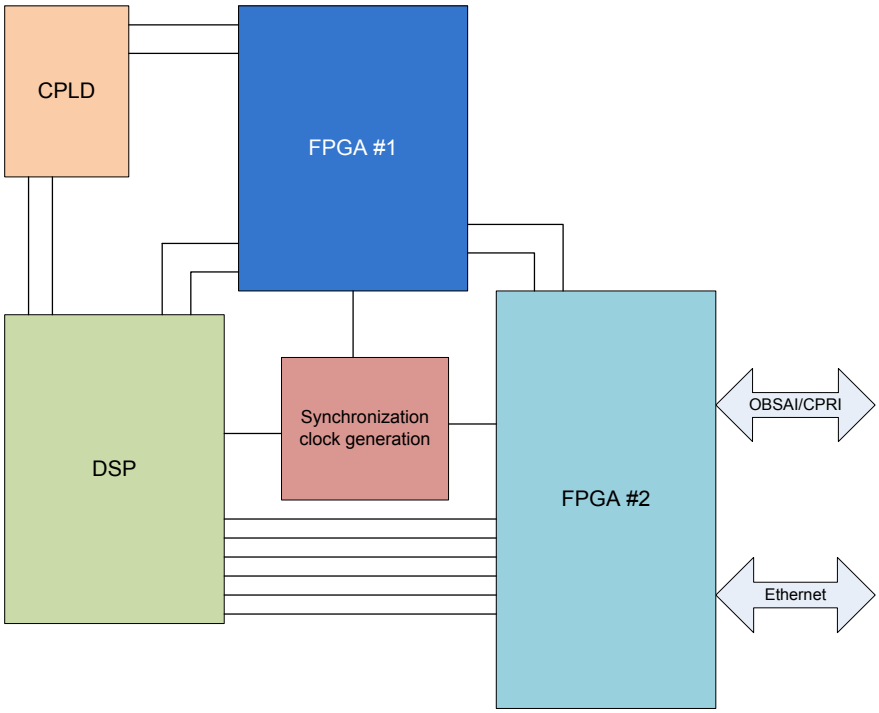

Figure 15. Digital circuit.

The IQ data coming from active antenna's Radio Frequency –parts (RF) is transferred through OBSAI RP3-01/CPRI –link with a data rate of 6-9 Gbps into the EFM's second FPGA (FPGA #2). At first, the received data is converted from serial into parallel data and after that the OBSAI/CPRI protocol is decompressed from it. In the next step, the IQ carrier data is read and saved into memory for compression which is done using the u-law process. The compressed data is then transformed into packets according to the Bearer Independent Protocol (BIP). Packets are prioritized by headers. Preamble and Start Frame Delimiter (SFD) are added and the Cyclic Redundancy Check (CRC) is done. Finally, data is converted back into a serial form for sending it to the baseband unit with a data rate from 1 to 25 Gbps. The conversion process is done backwards when data is moving to the other direction. Figure 16 presents IQ data conversion path.

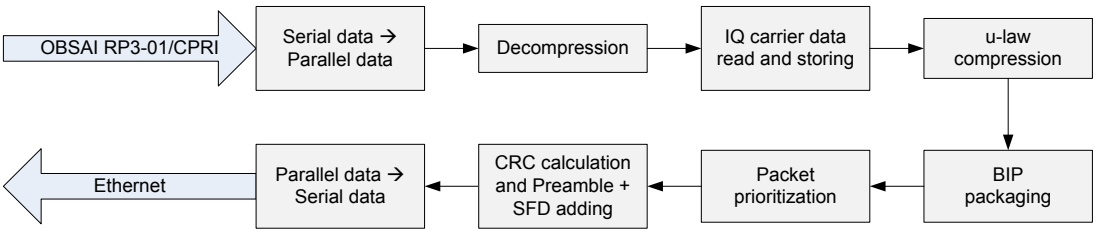

Figure 16. IQ data conversion from OBSAI RP3-01/CPRI to Ethernet.

In the LTE L1 processing, the received data can be processed in either the OBSAI/CPRI or the Ethernet form. Therefore, the IQ data conversion is not necessarily needed. Data is directed into the DSP and after digital signal processing, the data is sent back to the FPGA which then directs it into active antennas or baseband unit. Figure 17 presents the LTE L1 process.

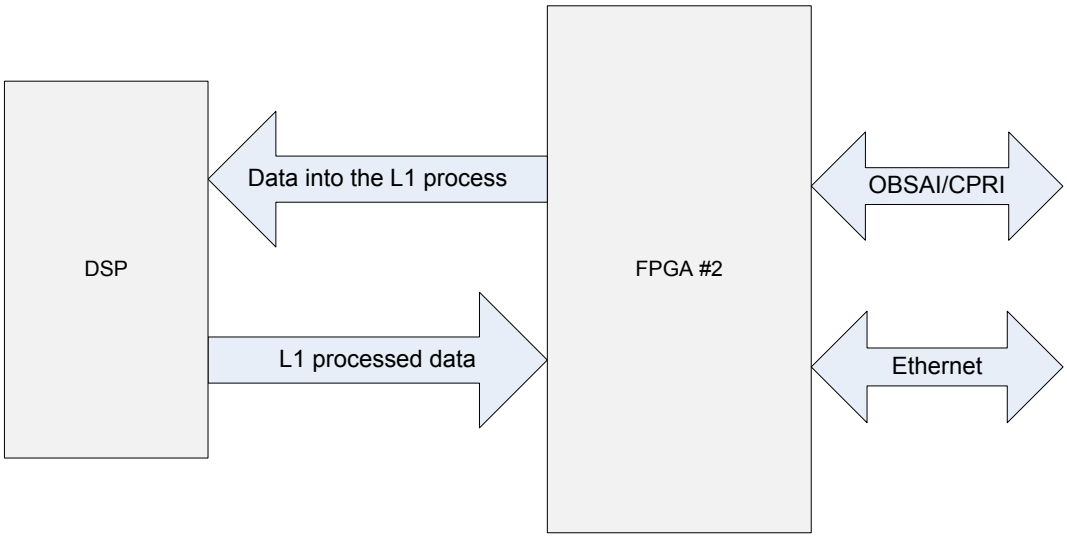

Figure 17. LTE L1 processing.

3. IMPLEMENTATION OF DESIGN

This chapter covers the implementation of the Ethernet fronthaul module. It consists of the following chapters: Hardware (HW) Implementation specification and Schematics. Chapter 3.1 explains what the HW Implementation specification is, what it includes and how it was done. Chapter 3.2 covers drawing the EFM's schematics, based on the Implementation specification covered in chapter 3.1.

3.1. HW Implementation specification

The purpose of the HW Implementation specifications is to define the implementations and the requirements for the designed device; in this case, the EFM. It also gives guidance for the designing of the hardware and software. It includes all the internal blocks: the FPGAs, the DSP and the power-block, the internal and the external interfaces. The FZM's HW Architecture specification was used as a base when writing the EFM's Implementation specification.

3.1.1. General Description

At first, the requirements of the EFM were gathered to specify how the device should work. All the requirements were stored into an excel sheets and then added into the document. A general description chapter was written after all the requirements were listed and it contained an overview, the main components of the device, a block diagram and a list of the external interfaces with pin outs. The overview presents a system environment where the EFM is used. This environment is presented in Figure 12. After the overview, the main components were mentioned in the following order:

- FPGA #1
- FPGA #2
- DSP
- CPLD
- 2 x Oven-Controlled Crystal Oscillator (OCXO)
- Serial Rapid Input/Output (SRIO) -switch
- Ethernet switch (Physical layer, PHY)
- 4 x PLL
- Powers

Drawing a detailed block diagram with the internal and the external interfaces was started by deleting all the components that were not needed from the FZM block diagram. After this, new components, interfaces, nets and busses were placed into the diagram. These new components are listed below and shown in red text and red outline color in Figure 18.

- FPGA #2 (including DDR3 and Flash)
- OCXO #2
- PLL #3 and PLL #4

- SRIO Switch
- Multiplexers
- External interfaces (QSFP+, SFP+, Registered Jack 45 (RJ45), Calibration interface and Minisas)

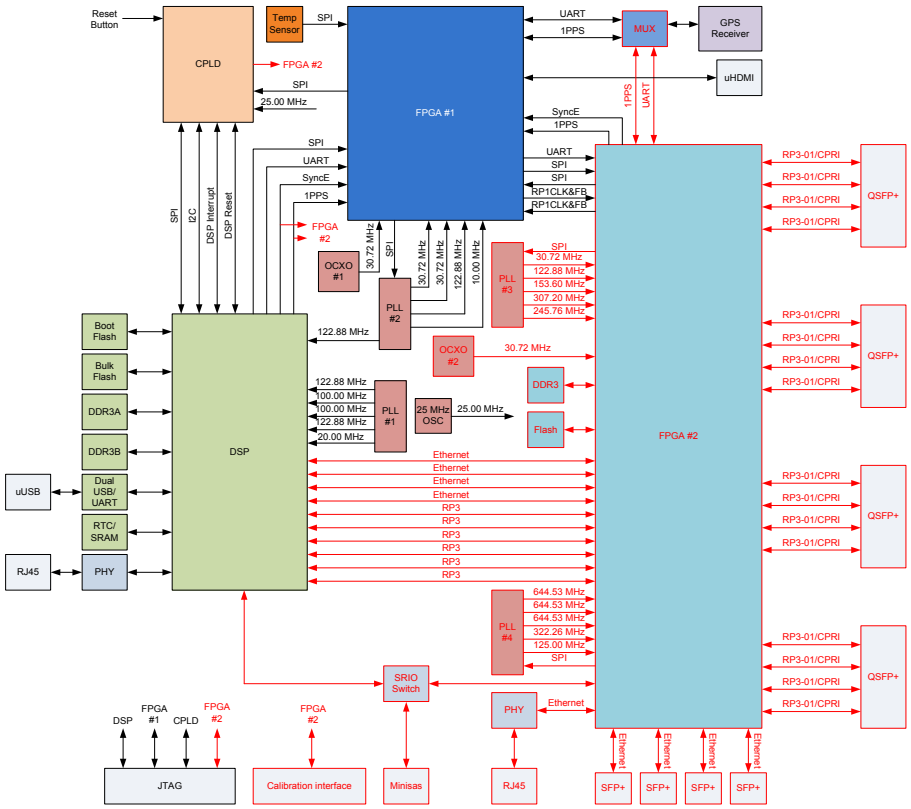

Figure 18. The detailed block diagram.

The external interfaces were listed, including their pin outs, type of signals (electrical or optical) and connection guidelines, after the block diagram. The EFM has the following external interfaces, which can also be seen in Figure 18:

- 4 x QSFP+
- 4 x SFP+
- 1 x µHDMI (micro High Definition Multimedia Interface)
- 1 x Minisas
- 1 x GPS antenna connector
- 1 x µUSB (micro Universal Serial Bus)
- 2 x RJ45
- 1 x JTAG (Joint Test Action Group)
- 1 x Calibration interface

All four QSFP+ are connected to the four active antenna columns, where each column includes four active elements. Total of sixteen active elements enable the three-dimensional beam forming. The QSFP+ interfaces provide optical data transmission between the EFM and the AEs. There are also four SFP+ interfaces to handle Ethernet transmissions between two BBUs and the EFM. Two SFP+ interfaces are connected to one BBU.

The µHDMI includes reference clocks for an external measuring device and an Inter-Internal Circuit (I2C) data bus for reading registers from the FPGA #1. A Minisas interface connects the EFM and a BBU for data transmission. The EFM's SRIO switch transmits and receives data from both the DSP and the FPGA #2. The GPS receiver is connected to the GPS antenna connector. The µUSB and both RJ45 interfaces are used to connect a computer into the EFM board. These interfaces are used for computer based testing. The RJ45 connectors are connected to the DSP and the FPGA #2 through a PHY block. The PHY block consists of 1:1 transformer and an Ethernet transceiver. The µUSB interface is connected to a dual Universal Serial Bus/Universal Asynchronous Receiver/Transmitter (USB/UART) –block, which is connected into the DSP and the FPGA #1. The main components are connected to the JTAG connector. It is used to program and to test the functionality of the components.

Standard and customized pin outs can be used in external interfaces. Table 2 presents standard QSFP+ pin out –table.

Table 2. QSFP+ pin out

Pin	Name	Description	Pin	Name	Description
1	GND	Ground	20	GND	Ground
2	TX2n	Transmitter 2 inverted data	21	RX2n	Receiver 2 inverted data
3	TX2p	Transmitter 2 non-inverted data	22	RX2p	Receiver 2 non-inverted data
4	GND	Ground	23	GND	Ground
5	TX4n	Transmitter 4 inverted data	24	RX4n	Receiver 4 inverted data
6	TX4p	Transmitter 4 non-inverted data	25	RX4p	Receiver 4 non-inverted data
7	GND	Ground	26	GND	Ground
8	ModSelL	Module select on low	27	ModPrsL	Module presence on low
9	LPMode _Reset	Low power mode reset	28	IntL	Interrupt on low
10	VccRX	Receiver voltage	29	VccTX	Transmitter voltage
11	SCL	I2C clock	30	VCC1	
12	SDA	I2C data	31	LPMode	Low power mode
13	GND	Ground	32	GND	Ground
14	RX3p	Receiver 3 non-inverted data	33	TX3p	Transmitter 3 non-inverted data
15	RX3n	Receiver 3 inverted data	34	TX3n	Transmitter 3 inverted data
16	GND	Ground	35	GND	Ground
17	RX1p	Receiver 1 non-inverted data	36	TX1p	Transmitter 1 non-inverted data
18	RX1n	Receiver 1 inverted data	37	TX1n	Transmitter 1 inverted data
19	GND	Ground	38	GND	Ground

3.1.2. Functional blocks

After the general description, the next chapter in the Implementation specification was about the functional blocks. This chapter consists of the operation of the functional components, internal and external interfaces. The Functional blocks -chapter includes all the main components but it also considers a DDR3 and a Flash memories.

3.1.2.1. Digital signal processor

The DSP was covered at first in the Functional blocks -chapter. The DSP has a several memories, two DDR3s working as a main memory, a Boot flash and a Bulk flash memories. The DDR3A is a size of 4 GB and the DDR3B 2 GB. Both DDR3s have an Error-Correction Code (ECC) –memories. The ECC is used to correct corrupted data stored into a DDR3 memory. The 16 MB Boot Flash contains objects required for the DSP's software to boot itself and it is configured via JTAG connector. The 2 GB Bulk Flash is used to support USB operations. The DSP is also connected into a Real Time Clock –Integrated Circuit (RTC IC), which includes 56 bytes of Static Random-Access Memory (SRAM). Basically, the RTC tracks time and calendar data during the normal operation. Figure 19 presents all the memories and the RTC connected to the DSP.

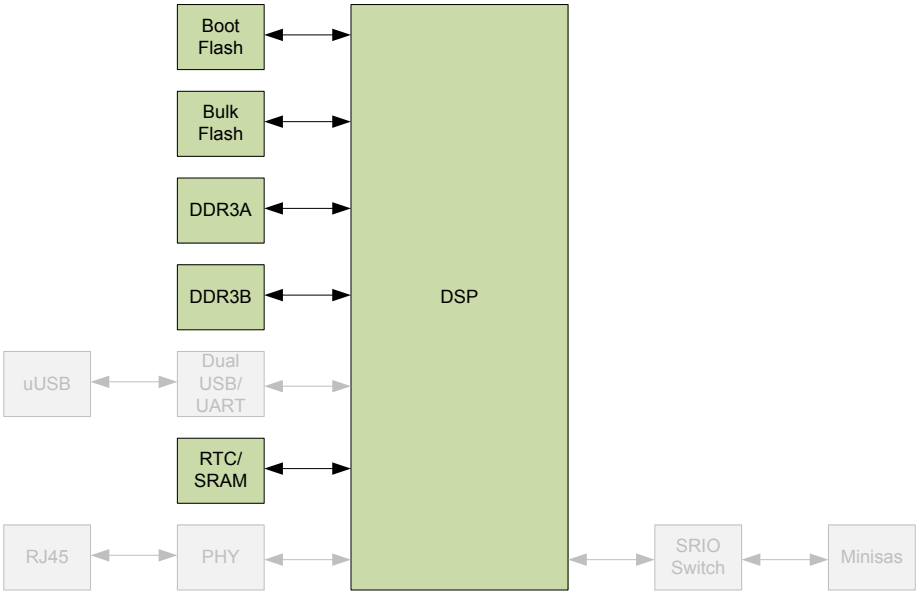

Figure 19. DPS with memories and RTC.

The DSP has three external interfaces towards the outside world: a µUSB connector through the dual USB/UART –interface block, an Ethernet using the RJ45 –connector and a data path to a SRIO switch and another from the switch to the Minisas-connector. As discussed before, the µUSB and the RJ45 are used to connect a computer to the EFM –board for testing. Minisas is used to connect the EFM and the baseband unit together. This connection is used to control and configure the EFM. These external interfaces are presented in Figure 20.

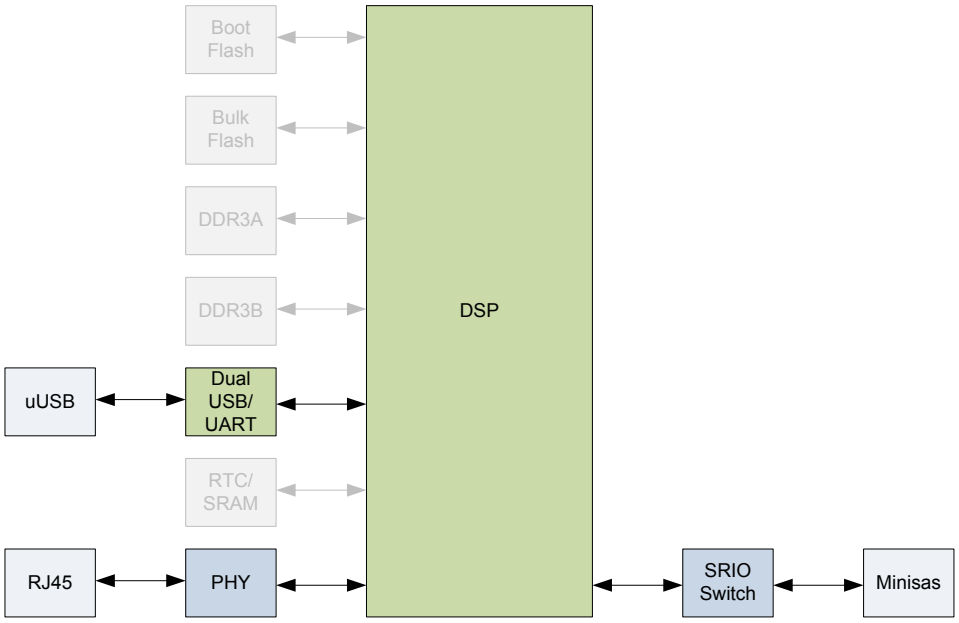

Figure 20. The DSP with the external interfaces.

Both FPGAs and the CPLD are connected with several busses and signals to the DSP ports. The CPLD has four busses: a Serial Peripheral Interface (SPI), an I2C, a DSP Interrupt and a CPLD Resets. The SPI bus is used to transmit and receive configuration data between the DSP and the CPLD. The DSP can access the CPLD's registers for control and status. Through the SPI bus, the CPLD monitors that the DSP's software is not crashed. The CPLD uses I2C bus to support the RTC/SRAMs level transition so the voltages are suitable for both devices. With the DSP Interrupt – signal, the CPLD can give an interrupt for the DSP if necessary. The CPLD Resets bus includes several different reset signals, which are discussed later on the CPLD chapter. These signals are used for resetting the DSP.

The FPGA #1 offers clock synchronization to the DSP by using Synchronous Ethernet (SyncE) clock and Pulse Per Second (1PPS) –signals, controls and configures DSP's registers through SPI bus. The FPGA #1 is connected to the Dual USB/UART block with one UART and to the DSP with another. This enables testing the FPGA #1 as well, which gives flexibility for testing in general. The DSP distributes the SyncE and 1PPS –signals to the FPGA #2. The second FPGA (FPGA #2) is connected to the DSP with six OBSAI RP3 and four 1/10 Gbps Ethernet signals. Data from the RRHs is sent with RP3 and the Ethernet signals to the DSP for L1 processing and then back to the FPGA #2 when the L1 process is done. The complete procedure is explained in Chapter 2.4 and shown in Figures 16 and 17. Connections between the DSP, the CPLD and the FPGAs are shown in Figure 21 below.

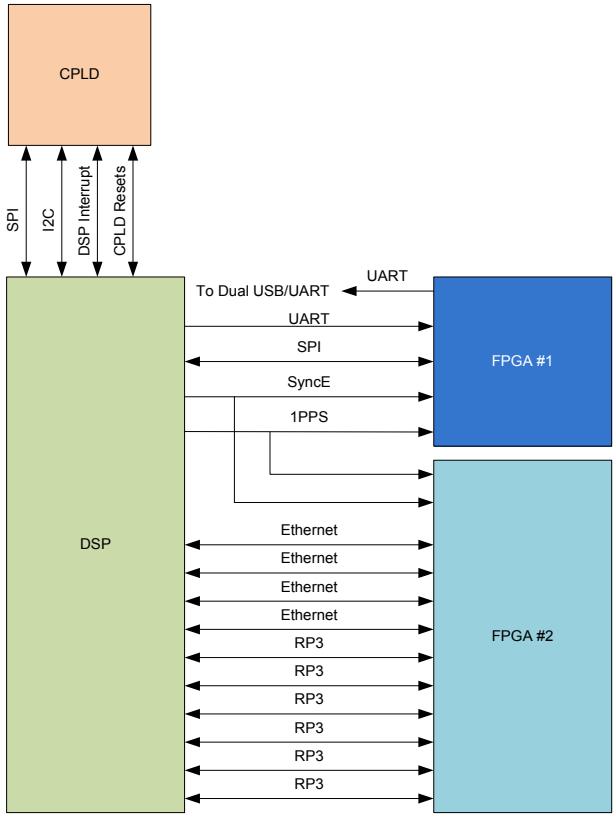

Figure 21. The DSP's connections to other devices.

The DSP requires several clocks in different frequencies which are mainly done with the PLL #1 but some clock signals are coming from other devices, like the FPGAs. Every clock is used for different purpose; one clock is needed for the DSP core, one clock for each DDR3 memories, etc. All clocks connected to the DSP were listed into a table and presented in a figure in the implementation specification. All these DSP clocks are presented in Table 3.

Table 3. DSP clocks

Clock name	Description	Frequency (MHz)
SYSCLK	Core clock	122.88
DDRCLK#1	DDR clock #1	100.00
DDRCLK#2	DDR clock #2	100.00
ARMCLK	ARM clock	122.88
ALTCORECLK	Alternate core clock	122.88
USBCLK	USB clock	20.00
TSREFCLK	FPGA #1 reference clock	122.88
RP1CLK	Frame sync interface clock	30.72
RP1FRAME	Frame burst	3.84

There are two core clocks, System Clock (SYSCLK) and Alternate Core Clock (ALTCORECLK), which the DSP can use. The SYSCLK is the primary core clock of the DSP and it is generated by the PLL #2. The secondary core clock, ALTCORECLK is fed from the PLL #1 to the DSP. The digital signal processor

architecture also includes Advanced RISC Machines (ARM) -processors. These processors need their own clock, also called ARM Clock (ARMCLK), which is done with the PLL #1. The ARMCLK works at 122.88 MHz frequency. Both DDR3 memories have their own 100 MHz clock, the DDR Clock 1 (DDRCLK#1) and the DDR Clock 2 (DDRCLK#2). A 20 MHz USB Clock (USBCLK) is needed to drive the DSP's USB data ports. Three clocks are coming from the FPGAs: the External Reference Clock (TSREFCLK) is used for synchronization between the FPGA #1 and the DSP. Both FPGAs send the OBSAI RP1 Clock (RP1CLK) and the RP1 Frame (RP1FRAME) to the multiplexer, which is connected to the DSP. With the multiplexer, one can choose which FPGA's RP1 signals are sent to the DSP. The RP1CLK is used to synchronize frames and the RP1FRAME is used to drive frame indicators. All DSP clocks are presented in Figure 22.

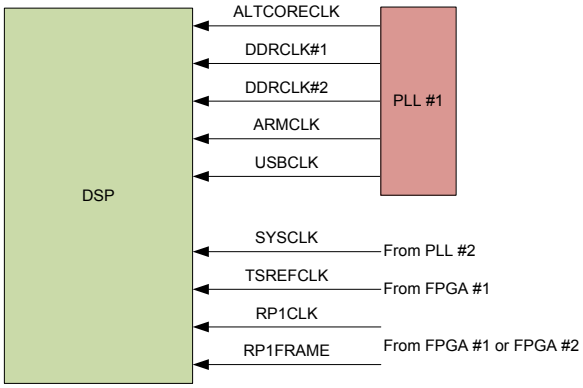

Figure 22. DSP clocking.

3.1.2.2. FPGA #1

After the digital signal processor and other related blocks were covered in the Implementation specification, the rests of the chapters were discussed with in similar manner. From two FPGAs, the FPGA #1 was covered first. The tasks specified for the FPGA #1 are listed below:

- Translating voltage levels for the DSP
- Supports synchronization function that generates system reference
- UART interface support for the GPS, the FPGA #2 and the DSP
- Supports the µHDMI interface
- Supports an external temperature sensor, which is a part of the thermal management system

The FPGA #1 provides voltage level translation for the DSP through SPI control busses. It supports the whole system reference generation by generating a system clock which is derived from the oven-controlled crystal oscillator. Through the UART interface, the FPGA #1 can access to the GPS receiver, the FPGA #2 and the DSP. The µHDMI interface is supported by providing various reference inputs for testing and debugging, like Local Management Trouble Shooting (LMTS). A temperature sensor is also supported with a SPI control bus.

The FPGA #1 has several interfaces towards other components on the board: the OBSAI RP1, the SPI, the UART, the synchronization and the clocking interfaces. Some of these were already discussed in the DSP chapter. There are a total of six OBSAI RP1 interfaces, which consist of frame sync interface clock and frame burst –signals. Two are for the DSP as mentioned before and four for the FPGA #2. The RP1 connections between the FPGAs go both ways, so one pair of a frame clock and a frame burst go from the FPGA #1 to the FPGA #2 and vice versa. As in the DSP case, the RP1 signals have the same purpose between FPGAs.

The FPGA #1's SPI and UART interfaces are used for communication and serial and parallel data translation with other devices. The JTAG is for testing and programming. The FPGA #1 is connected to the following devices through the SPI busses:

- Temperature Sensor
- CPLD
- DSP
- PLL #2
- FPGA #2

The JTAG and UART connections to the devices:

- DSP
- USB/UART
- GPS module
- FPGA #2
- JTAG connector

In Figure 23, all described connections are shown and as well the RP1 clock and the frame burst signals from the FPGA #2 to the DSP.

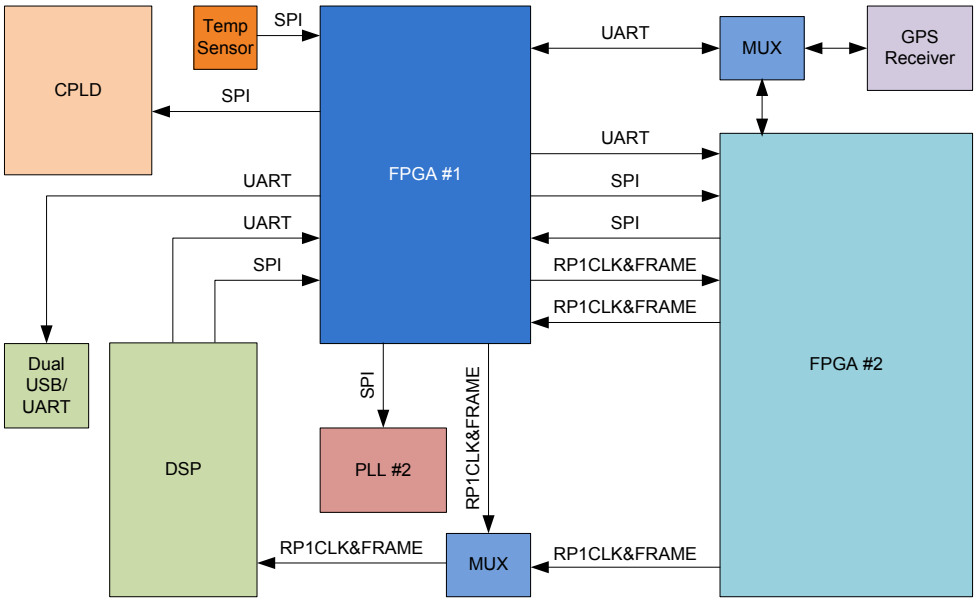

Figure 23. FPGA #1 signals.

Synchronization and clocking is handled with the OCXO #1 and the PLL #2. The OCXO #1 generates 30.72 MHz clock, from which the FPGA #1 derives the 122.88MHz system clock and distributes it to the other components. Four clock signals for the FPGA #1 are generated by the PLL #2. Frequencies of these signals are 10.00 MHz, 122.88 MHz and 30.72 MHz. There are other reference clocks connected to the FPGA #1 as well. Synchronous Ethernets at frequencies of 125 MHz for 1 G Ethernet and 156.25 MHz for 10 G Ethernet are received from the FPGA #2 and the DSP. A 1PPS reference clock signal comes from the GPS module. The 1PPS signal can be received also from the µHDMI connector, when the Ethernet Fronthaul Module is connected to the test equipment. The FPGA #1 can distribute the system 1PPS, received from the GPS, to any other devices on the board and test equipments through µHDMI connector. With the 1PPS, all the devices on board stay synchronized. Figure 24 presents all of these clock signals.

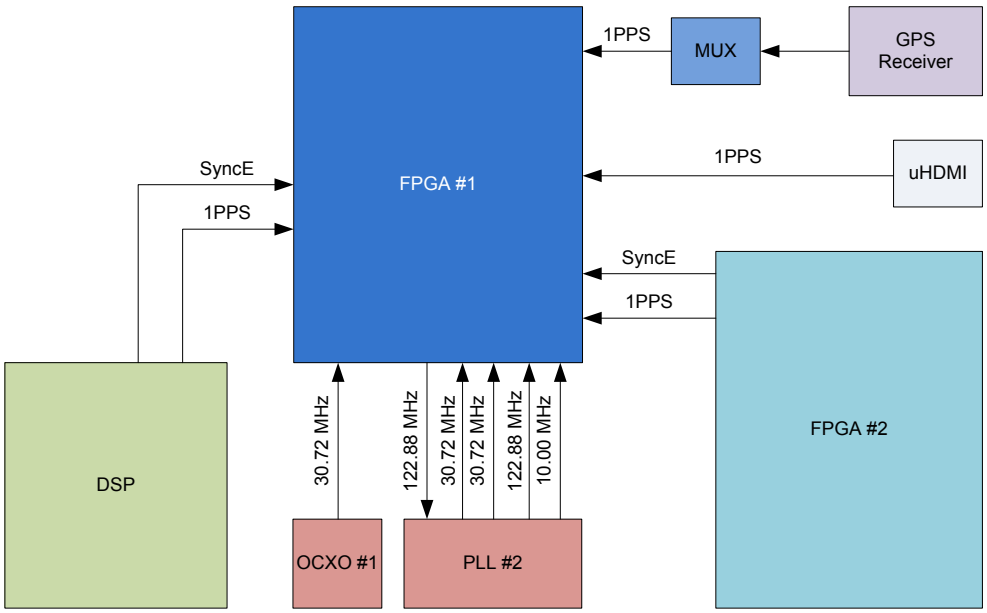

Figure 24. FPGA #1 clocking.

3.1.2.3. FPGA #2

The second FPGA's main tasks were roughly covered in chapter 2.4 but they are listed below:

- Conversion from (1/10/25 G) Ethernet to (6/9 G) OBSAI RP3-01/CPRI and vice versa
- L1 processing with the DSP between the RRH and the BBU
- Providing active element level control and beam forming

The FPGA #2 can do same procedures as the FPGA #1, so the latter can be left out if wanted. The FPGA #2 includes external DDR3 and Flash memories. All the used external interfaces are:

- 4 x (1/10/25 G) Ethernet/(6/9 G) OBSAI RP3-01/CPRI for SFP+

- 16 x (6/9 G) OBSAI RP3-01/CPRI/10G Ethernet for QSFP+
- 4 x 1/10 G Ethernet for DSP
- 6 x 3/6 G OBSAI RP3 for DSP
- 6 x RP1
- Other interfaces (SPI, UART, etc)

The OBSAI/CPRI interfaces are used to make internal connections between the EFM components and external connections between the BBU and the RRHs. The sixteen links of optical OBSAI RP3-01/CPRI are used to connect the EFM into the RRHs. Every QSFP+ takes four links, so only four connectors are needed. Data traffic between the FPGA #2 and the DSP goes through the six 3/6 G OBSAI RP3 -links or the four 1/10 G Ethernet –links. This depends on the form of the data, as explained in chapter 2.4. The EFM can be connected to the BBU through four 1/10/25 G Ethernet links. It is also possible to connect the EFM into the four BBUs by connecting one link per BBU, so there are different configurations available. The RRH data is transmitted to the BBU after data conversion and vice versa. Frame synchronization between the FPGAs and the DSP is done with the RP1 links, as specified before. The OBSAI RP3 and the Ethernet connections to the DSP were already shown in Figure 21 and the OBSAI RP1 in Figure 23, so they are not presented in Figure 25.

The second FPGA has two external interfaces for testing and controlling. One is the Ethernet link for computer testing which goes through an RJ45 –connector to the computer. Another one is the control link which connects the FPGA #2 to the BBU with a Minisas connector. The control link is connected to the same SRIO switch, where the DSP external control link is connected. This is shown in Figure 20. The switch connects the DSP or the FPGA #2 one at a time to the Minisas connector. This link enables communication controlling between the DSP or the FPGA #2 and the BBU.

Other interfaces like the SPI and the UART are used in a similar manner like in the FPGA #1 case. The FPGA #2's SPI interfaces are connected to the first FPGA and two PLLs. The UARTs are connected into the FPGA #1 and the GPS module. The JTAG is used like with the other devices. Calibration interface is used in an AEs beam calibration. Figure 25 presents DDR3 and Flash memories and interfaces that are not shown in previous figures.

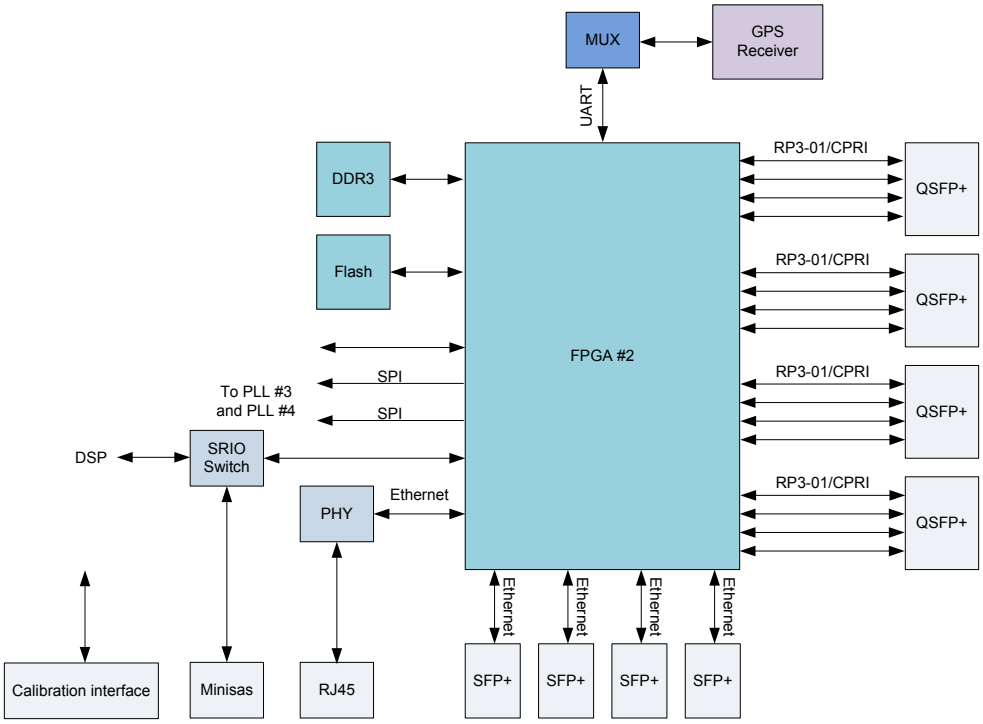

Figure 25. Internal/external interfaces and memories.

The FPGA #2 uses several different clock signals. Main clocks are generated with two PLLs. The PLL #3 distributes eight clock signals at frequencies from 30.72 MHz to 307.20 MHz. The PLL #4 generates three 644.53 MHz, one 125 MHz and one 322.26 MHz clocks for the FPGA #2. The oven-controlled crystal oscillator #2 or OCXO #2 generates a 30.72 MHz clock signal. The FPGA #2 generates 122.88 MHz and 30.72 MHz reference clocks for the PLL #2 and the PLL #3. Three SyncE and two 1PPS reference clocks between both FPGAs and the DSP are covered on pages 24 and 26 and in Figure 24. There is also a 1PPS coming from the GPS. Figure 26 presents the covered second FPGA's clock signals.

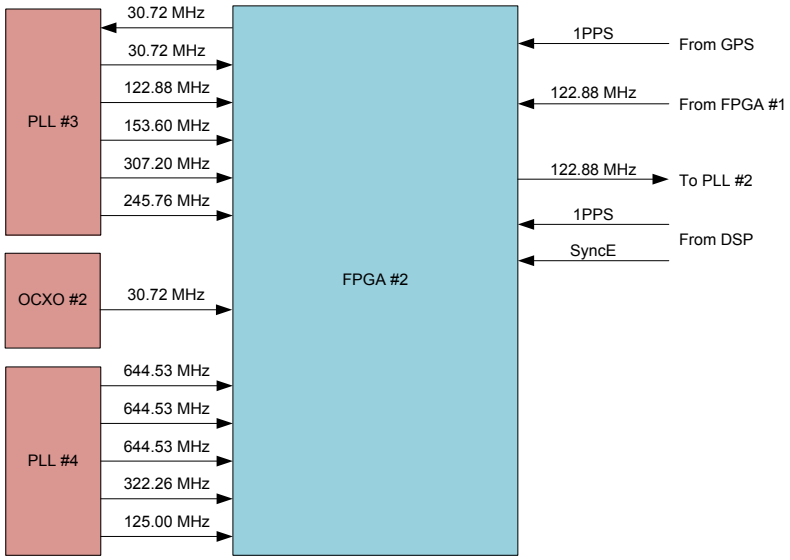

Figure 26. FPGA #2 synchronization and clocking.

3.1.2.4. CPLD

After the FPGAs were covered, the next component was the CPLD, which contains general purpose logic, a user flash memory and a SPI core. It provides the following features:

- Reset control for the EFM components, like the FPGAs, the DSP, the PLLs and the Ethernet PHYs
- Provide reset reasons in a memory mapped register to report the reason for a host reset
- Provide a Watchdog for devices on the EFM board
- Provide support for a slave serial configuration of the FPGA #1. Configuration data will pass through the SPI bus from the DSP.
- Reserve user pins for JTAG to support in-system programming of the DSP flash

The CPLD generates most of the reset and interrupt signals in the EFM. When a reset occurs, the CPLD stores the reason of the reset into its memory register from where the host reset can read it. The Watchdog functionality checks that each device's software is working correctly. The Watchdog acts as a timer, which resets a device if the device's software has not answered in certain period of time. In the slave configuration, the FPGA #1 can be configured with the data that is coming from the DSP. Configuration data is sent into the CPLD through a SPI bus and from there to the FPGAs. There are several pins reserved for JTAG, which can be used to program the DSP flash if necessary. JTAG is also used to program the CPLD itself like other components as well. The CPLD is presented in Figure 27.

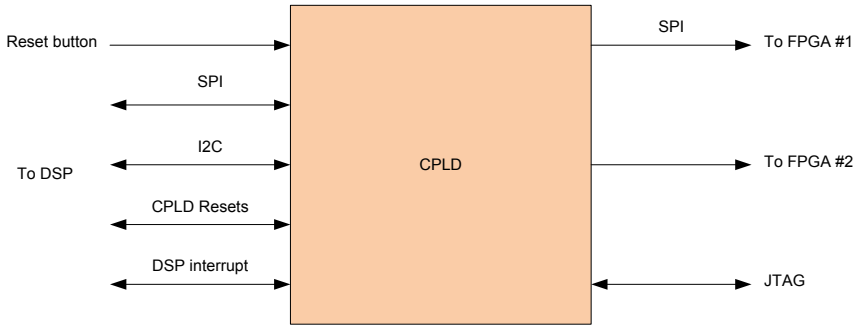

Figure 27. Complex programmable logic device.

As can be seen from Figure 27, there are several interfaces that connect to other devices. The four busses to the DSP are already covered on page 23: the SPI is used for configuration data transmission, to access the CPLD's registers and for DSP software monitoring. The I2C is used for RTC/SRAM voltage level translation. Through the DSP Interrupts connection, the CPLD can give interrupts to the DSP. Finally, the CPLD has several reset signals for the DSP. There is a reset button on the EFM -board connected to the CPLD, which generates reset signal for the entire EFM. All reset signals generated by the CPLD are presented in Table 4.

The configuration data from the DSP is routed through the CPLD to the first FPGA. The second FPGA is also connected with a control bus to the CPLD. The

FPGA #2 configuration can be done with the support of the CPLD. Like the other components, the CPLD can also be programmed and tested using the JTAG interface. For the clock synchronization, the CPLD uses a 25 MHz clock signal, which is generated with a 25 MHz OSC –oscillator, shown in Figure 28.

Table 4. Reset signals generated by the CPLD

Reset type	Effects
Reset button	Resets entire EFM –board.
Power on reset	Resets EFM –board, when power is turned on.
Local reset	When the device has not punched the watchdog in a certain period of time, the local reset is given.
Critical temperature reset	When the temperature of the component, like FPGA, has raised above the critical temperature limit, it is reseted.

3.1.2.5. Synchronization clock generation

Synchronization clock generation was also described in the Implementation specification. All clocks are generated with two oven-controlled oscillators, four phase-locked loops and one 25 MHz oscillator. Both OCXOs generate main 30.72 MHz clocks for the FPGAs. The first phase-locked loop generates five clocks and the second PLL one clock for the DSP. The second PLL generates also four clocks for the FPGA #1. Third and fourth PLLs generate total of ten clocks, five each, for the FPGA #2. The 25 MHz oscillator feeds 25 MHz clock to the CPLD. The Ethernet Fronthaul module's synchronization clock generation components are presented in Figure 28.

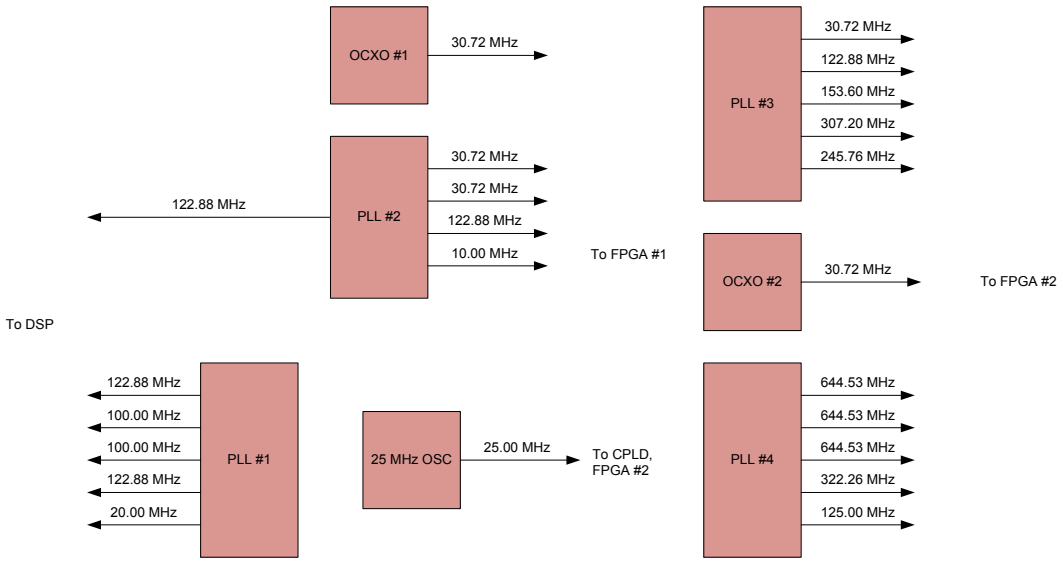

Figure 28. Synchronization clock generation blocks.

3.1.2.6. Power supplies

The last part implemented to the Implementation specification was the EFM's Power block -chapter. The chapter consists of the descriptions of a power manager, an isolated GPS power and DC-DC converters. It also includes output current calculations of the all converters.

The EFM uses 12 V as a primary voltage. The input voltage is connected to the inrush controller, which protects the EFM –board when the power is switched on. After the controller, the 12 V voltage is distributed to the rest of the converters. There are nineteen step-down DC-DC converters, as well as a GPS power and a fan power. All these power supplies are monitored and controlled by a power manager. The power manager monitors the voltages, the currents and the temperatures of the power supplies. It is also used for enabling, resetting and interrupting the power supplies.

The GPS module is powered with an isolated 5 V power supply. The isolated power supply includes a transformer, which makes it more complex than the non-isolated power supplies. The fan power is used to power the EFM's cooling fan. As mentioned before, there are nineteen step-down or buck DC-DC converters to power rest of the devices on the EFM –board. The detailed power block hierarchy presented in Figure 29 covers: voltage conversions and maximum output currents of the power supplies.

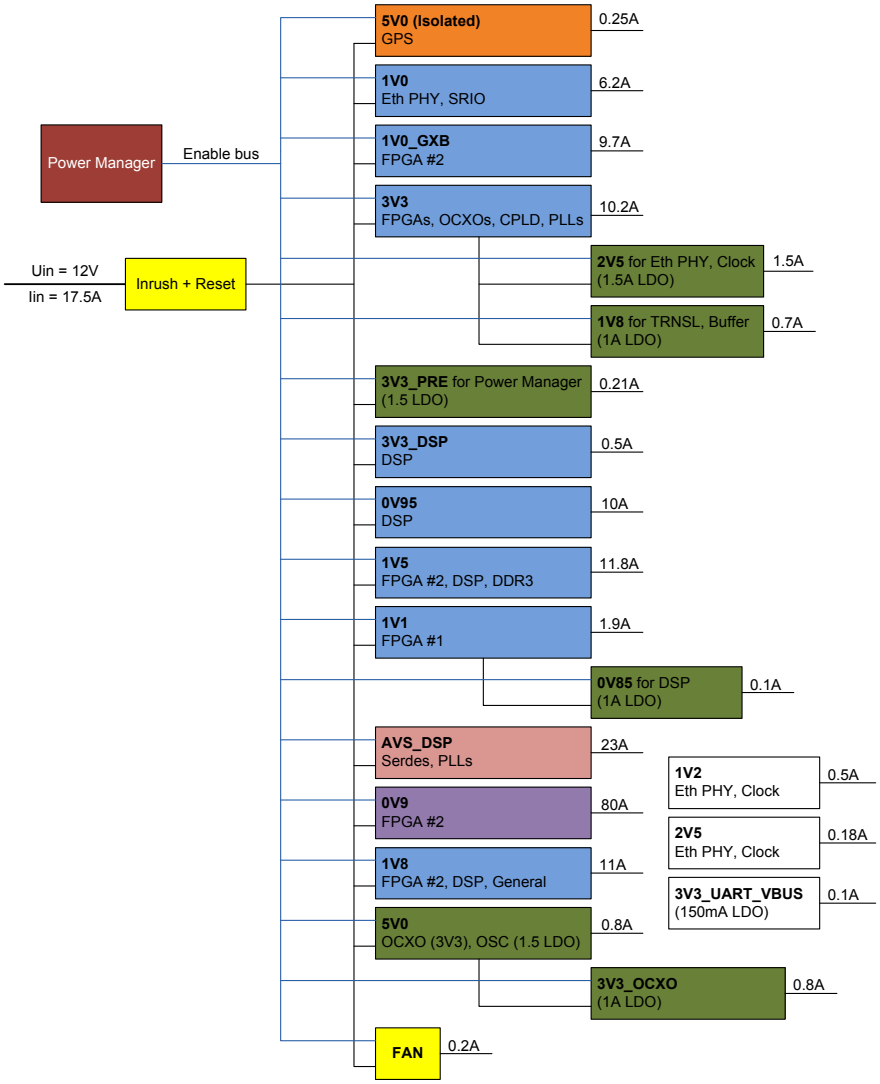

Figure 29. The EFM's power block.

3.2. Schematics

The EFM's schematics were drawn with the Mentor Graphics' DxDesigner software. In the schematic drawing procedure, the HW Implementation specification was used as a reference, especially for connecting all devices together. Also the FZM's schematics were used in the EFM's schematic drawing as a help. Basically, this means that the same schematics were used with some modifications. In this chapter, the main focus is in the schematics of the new components, which were listed on page 19.

3.2.1. Minisas connector and SRIO switch

The Minisas connector is connected to the DSP and the FPGA #2 via SRIO switch. The type of the connector is 1 x 2, which means that there are sixteen pairs of signal lines but in this case only half of them are used. Unused pins are not connected. Three transceiver links out of four are used. One link is for the 10 Gbps Ethernet, which is connected straight to the FPGA #2 and two links for the Serial RapidIO or SRIO. In the schematics, one transceiver link is presented with one transmitter and one receiver differential line. This can be seen from Figure 30, where for example, the signal FPGA_A_SAS_10GE[1:0] is coming from the FPGA #2 to the Minisas connector which means it is a receiver line for the connector. SAS_FPGA_A_10GE[1:0], on the other hand, is a transmitter line of the connector and receiver line for the FPGA #2. There are also control signals connected to the Minisas connector, like I2C, module presence and interrupt. The Minisas connector is presented in Figure 30.

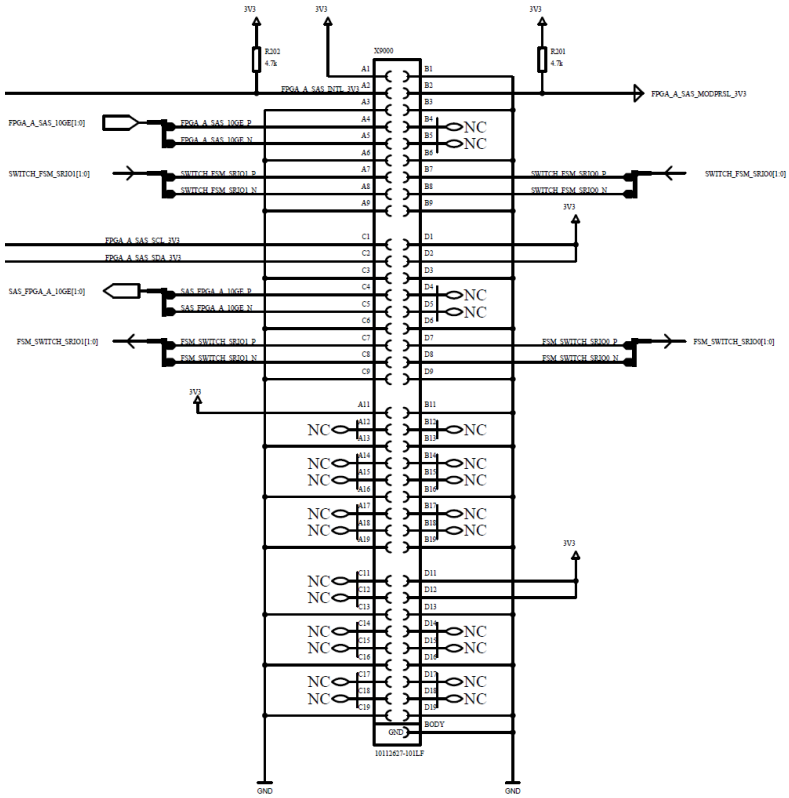

Figure 30. The Minisas connector.

The SRIO links are connected to the SRIO switch and from there to the FPGA #2 or the DSP. The SRIO switch is used for routing and distributing data between the DSP, the FPGA #2 and the BBU. It consists of a transceiver and a control parts. The transceiver part has sixteen differential ports and all transmission data is moves through these ports. The transceiver link from the DSP is connected to the transceiver port 0 and the link from the FPGA #2 is connected to the port 4. One of these links is routed to the Minisas connector via transceiver port 9 or 10. The receiver lines are connected to the left side and transmitter lines to the right side of the SRIO transceiver. Rests of the pins are left unconnected because they are not needed in this case. The SRIO transceiver part is shown in Figure 31.

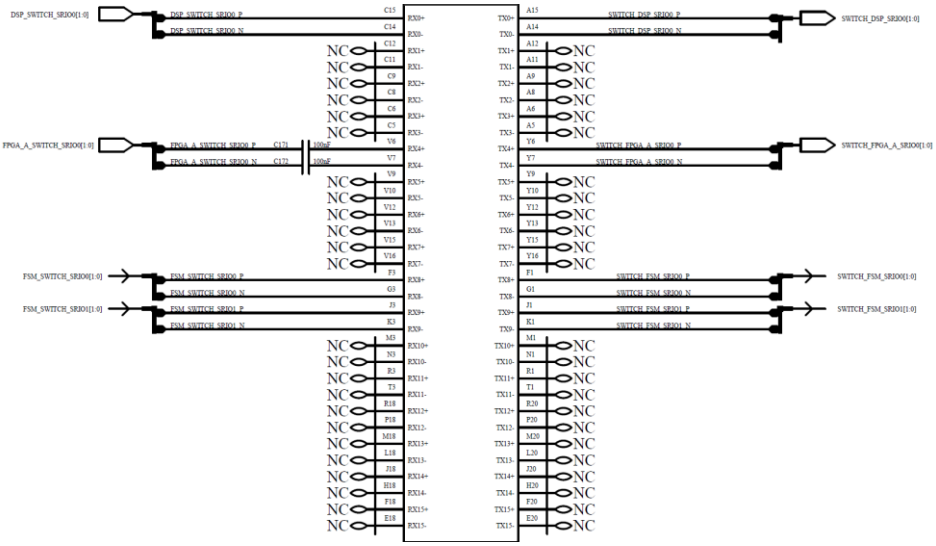

Figure 31. The SRIO transceiver part.

The switch can be programmed and controlled with the control part. The control section is shown in Figure 32. The control part receives a reference clock from the clock block, which acts as a system clock for the switch. The reference clock is connected into the REF_CLK+/- pins. The SRIO switch receives a reset and an interrupt -signals from the FPGA #2. The reset signal is a zero active and it is used when the switch is needed to be reset. The reset signal is connected to the _RST pin. The interrupt signal is used when an error occurs in the device. The interrupt signal is connected to the interrupt pin. There are three pins for the transceiver port speed configuration, called Speed Select pins (SPD). The speed of the transceiver ports can be configured by setting pins at a high or a low state. Different port speeds are listed in Table 5.

Table 5. The SRIO switch port speed selection

Value on the pins (SPD2, SPD1, SPD 0)	Port speed (Gbaud)
000	1.25
001	2.5
01X	5.0
101	3.125
11X	6.25

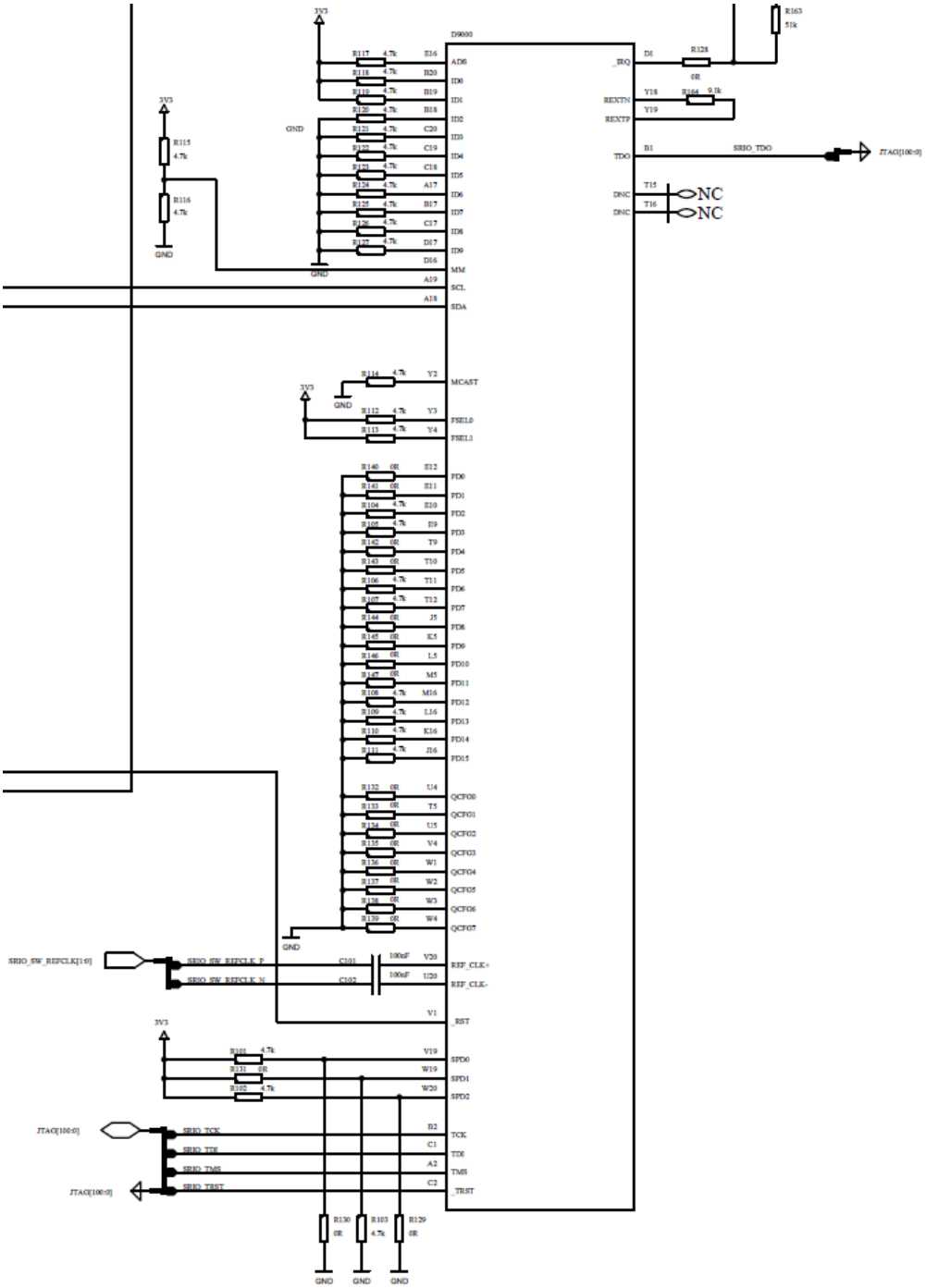

Figure 32. The SRIO control part.

As can be seen from Figure 32, the pins SPD0 and SPD2 are tied at 3.3 V and the pin SPD1 to the ground through 4.7 kΩ resistors. This sets the port speed to the 3.125 Gbaud. Baud rate refers to the number of signal or symbol changes that occur per second. If one symbol includes only one bit, the bit rate and the baud rate are the same. But if one symbol includes two bits, the overall bit rate is twice larger than baud rate. For example, if the baud rate was 4800 then the bit rate would be 9600 [13]. In this case, the baud rate is same as the bit rate so the port speed is 3.125 Gbps.

Port Disable pins (PD0-PD15) are used to activate transceiver ports. There is one pin per transceiver port. When port disable pin n is left floating, then transceiver port n is active. If pin is connected to the ground, the port is not active. In this case, the pins 0, 1, 4, 5 and 8-11 are connected to the ground via 0 ohm resistors. In manufacturing, these 0 ohm resistors are left uninstalled to the board which leaves pin unconnected. Other port disable pins are connected to the ground via 4.7 kΩ resistors. The 0 ohm resistor pads are left to the board, so the configuration of the active transceiver ports can be changed.

The SRIO switch's transceiver ports are divided into four quadrants. Every quadrant has two configuration pins. The widths of the transceiver ports can be configured with these pins. There are three widths; 1x, 2x and 4x. Choosing the widths is done in the same way as the transceiver ports activation. When the Quadrant Configuration (QCFG) pins are connected to the ground, the width is 4x and when pins are floating, the width is 1x. In this case, all port widths are set to 1x because the signals connected to the SRIO switch consists of one differential pair. The SRIO switch port width configuration is presented in Table 6.

Table 6. The SRIO switch quadrant pin setting

Quadrant pin setting	Port width
00	4x
01	2x, 2x
10	2x, 1x, 1x
11	1x, 1x, 1x, 1x

The JTAG connection can be used to program and check the status of the switch. The programming can also be done with an I2C bus.

3.2.2. Ethernet

The RJ45-connector and Ethernet PHY's magnetic transformer are presented in Figure 33. The RJ45 has four TRX signal pairs and transceiver signal pairs are connected to the Ethernet transformer, which turn ratio is 1:1. The transformer isolates the Ethernet transceiver from the RJ45-connector. The transformer's Media Center Taps (MCT) are terminated via 75 ohm to the ground and 1 M ohm resistors and 1 μF capacitors. The Chip Center Taps (TCT) are also connected to the ground via 100 nF capacitors. The Ethernet transceiver Media Data Input/Output (MDIO) pins are connected to the transformer's chip side pins and RJ45 –connector to the media side pins.

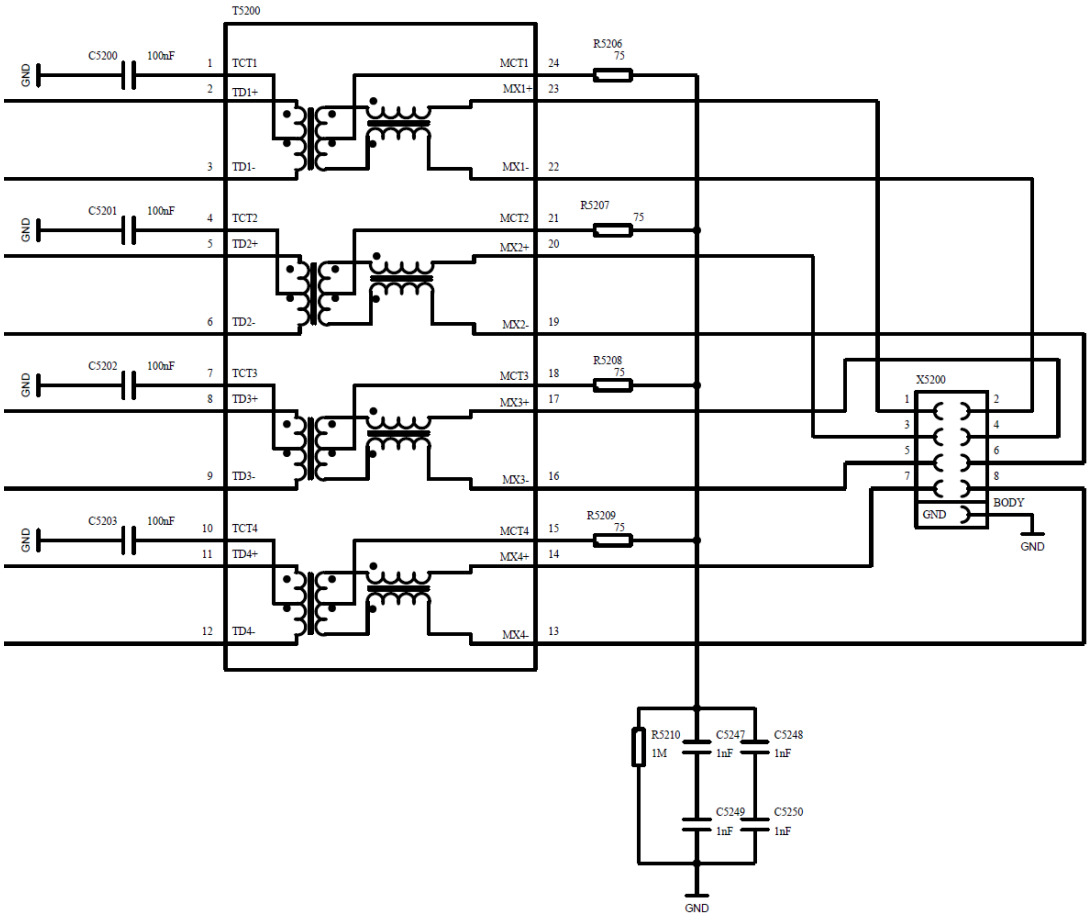

Figure 33. The Ethernet transformer and the RJ45-connector.

The Ethernet PHY or transceiver is located between the second FPGA and the RJ45-connector. It is connected into the FPGA #2 via Serial Gigabit Media Independent Interface (SGMII). The SGMII is a digital interface that provides an 8-bit wide datapath between the 1000 Mbit/s PHY and a Media Access Control (MAC) -sublayer. It also supports the 4-bit Media Independent Interface (MII), which is defined in the Institution of the Electrical and Electronics Engineers (IEEE) 802.3z specification [14]. The Ethernet PHY converts the 2-bit SGMII data coming from the FPGA #2 to the 8-bit MDIO -form. After conversion, the data is transmitted to the magnetic transformer through the MDIO interface. The data received from the magnetic transformer is finally converted back to the 2-bit form and sent to the FPGA #2.

There are also another two wire MDIO (MDC and MDIO) coming from the FPGA #2. It is a serial interface to connect a management entity and a managed PHY for the purposes of controlling the PHY and gathering status from the PHY [15]. The Ethernet PHY receives an external differential clock called BH_CLK25[1:0] from the clock block. A 125 MHz recovery clock is generated and sent from the 125 MHz pin to the first FPGA. The Ethernet PHY receives a reset-signal from the CPLD or the second FPGA. When the hardware reset is asserted, the 25 MHz clock is connected to the 125 MHz clock pin, which informs the FPGA #1 that the PHY is reset. The Ethernet PHY's three Light Emitting Diode (LED) pins are connected to the CPLD, which drives the LEDs to give visual information of how the PHY is working. The LEDs informs if the Ethernet transceiver is in an active state and what

the speed of the data transmission is. The Ethernet PHY/transceiver is presented in Figure 34.

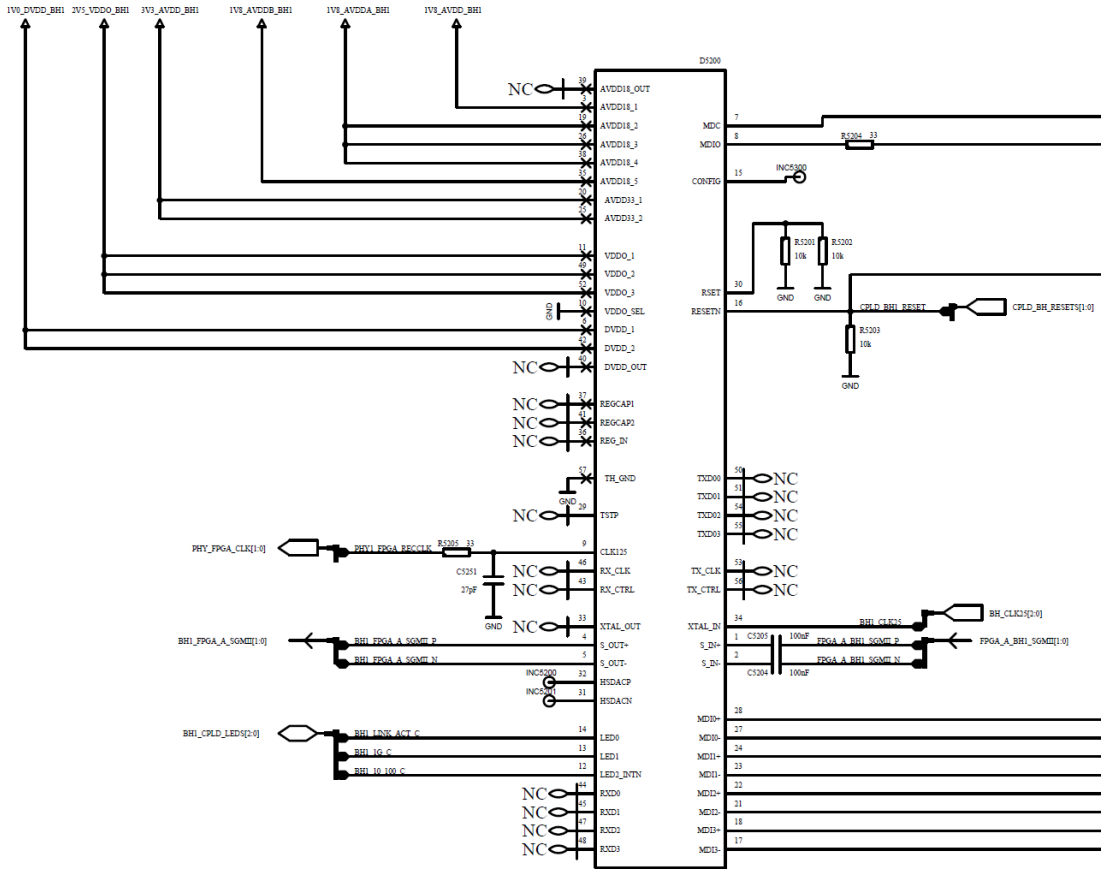

Figure 34. The Ethernet transceiver.

3.2.3. SFP+ and QSFP+

It is possible to connect four 1/10 Gbps Ethernet or OBSAI/CPRI links between FPGA #2 and one or several BBUs with SFP+ -connectors. The SFP+ connector uses a 3.3 V supply voltage, which is connected to pins called VccR and VccT. Respectively, pins VeeR and VeeT are connected to the ground. There are also eight control signals, which are tied to 3.3 V with 4.7 kΩ and 10 kΩ pull-up resistors. These control signals are listed below:

- TX Fault (Transmitter Fault)
- TX Disable (Transmitter Disable)
- SCL (Serial Interface Clock)
- SDA (Serial Interface Data)
- MOD ABS (Module Absent)
- RX LOS (Receiver Los)
- RS0 (Rate Select 0)
- RS1 (Rate Select 1)

The TX Fault -signal indicates if there is some kind of a laser fault in the transmission link. TX Disable is used to shut down the transmitter's optical link. I2C SCL and SDA are used for SFP+ module configuration. Module Absent informs if the SFP+ module connected to the board. RX LOS informs that the received signals power is too low. With the RS -signals, the transmission bandwidth can be adjusted from reduced to full bandwidth. Transceiver link is connected to the differential Transmitter Data (TD)+/- and Receiver Data (RD)+/- pins. Pins connected to the ground, below in Figure 35, are the cage where the SFP+ transceiver module is inserted. [16]

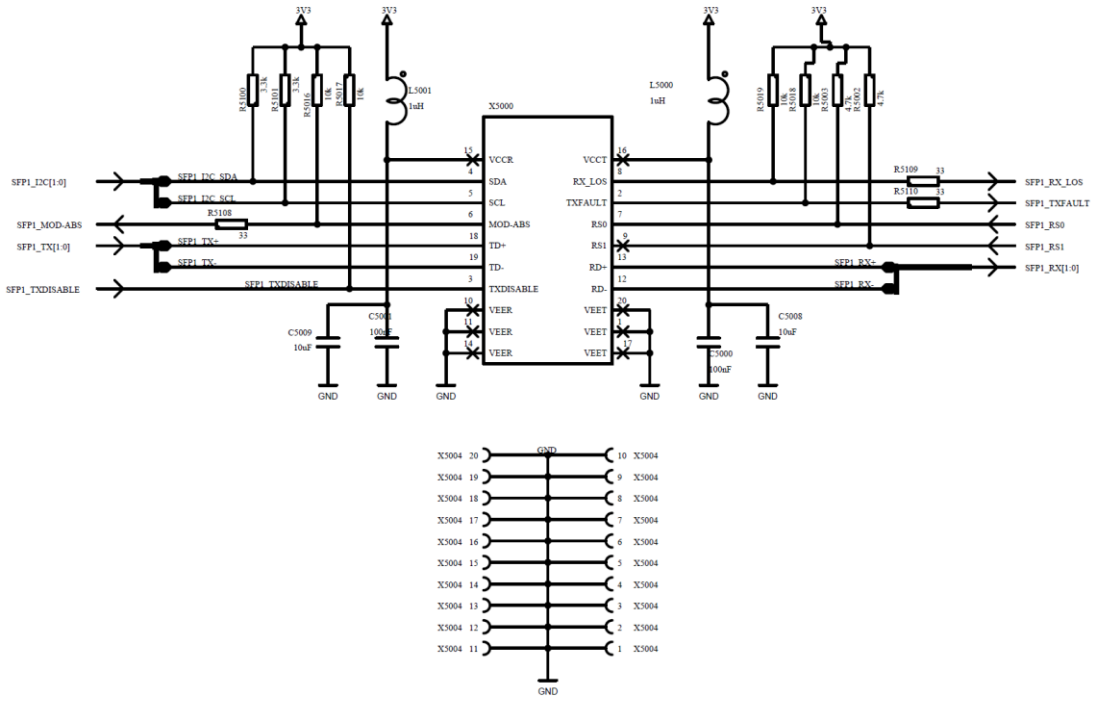

Figure 35. The SFP+ module and the cage.

The QSFP+ interfaces are used to connect RRHs to the EFM. As mentioned before, the QSFP+ consists of four SFP+ interfaces, which means that there are four transceiver links. The transceiver links are connected to the second FPGA's transceiver banks; this will be covered later in the document. 3.3 V supply voltage is used in the same way as in the SFP+ case. There are also five control signals in the QSFP+, like module presence and I2C These signals are coming from the FPGA #2's IO bank. The QSFP+ connector's pin out is presented in Table 2, where the pins or signals are explained more precisely. The QSFP+ schematics also include the connector cage, which pins are connected to the ground as shown in Figure 36.

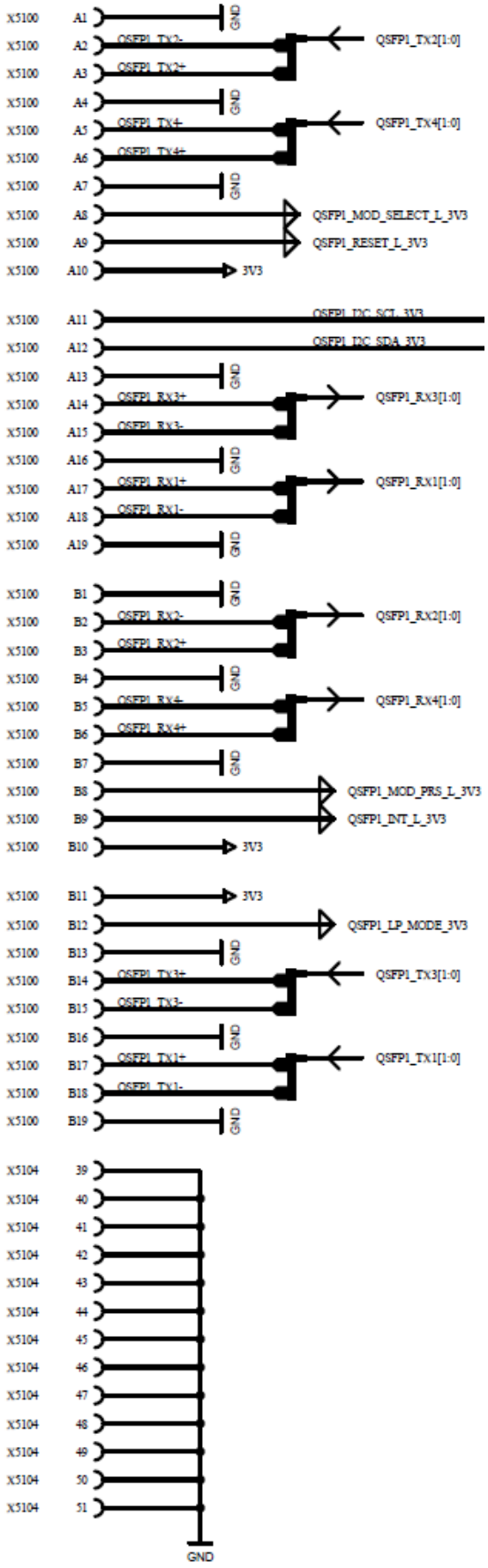

Figure 36. The QSFP+ -connector.

3.2.4. Calibration interface

The Calibration interface is a 120 pin connector, which is connected to the FPGA #2. Calibration signals are presented in Table 6 below and the interface in Figure 37.

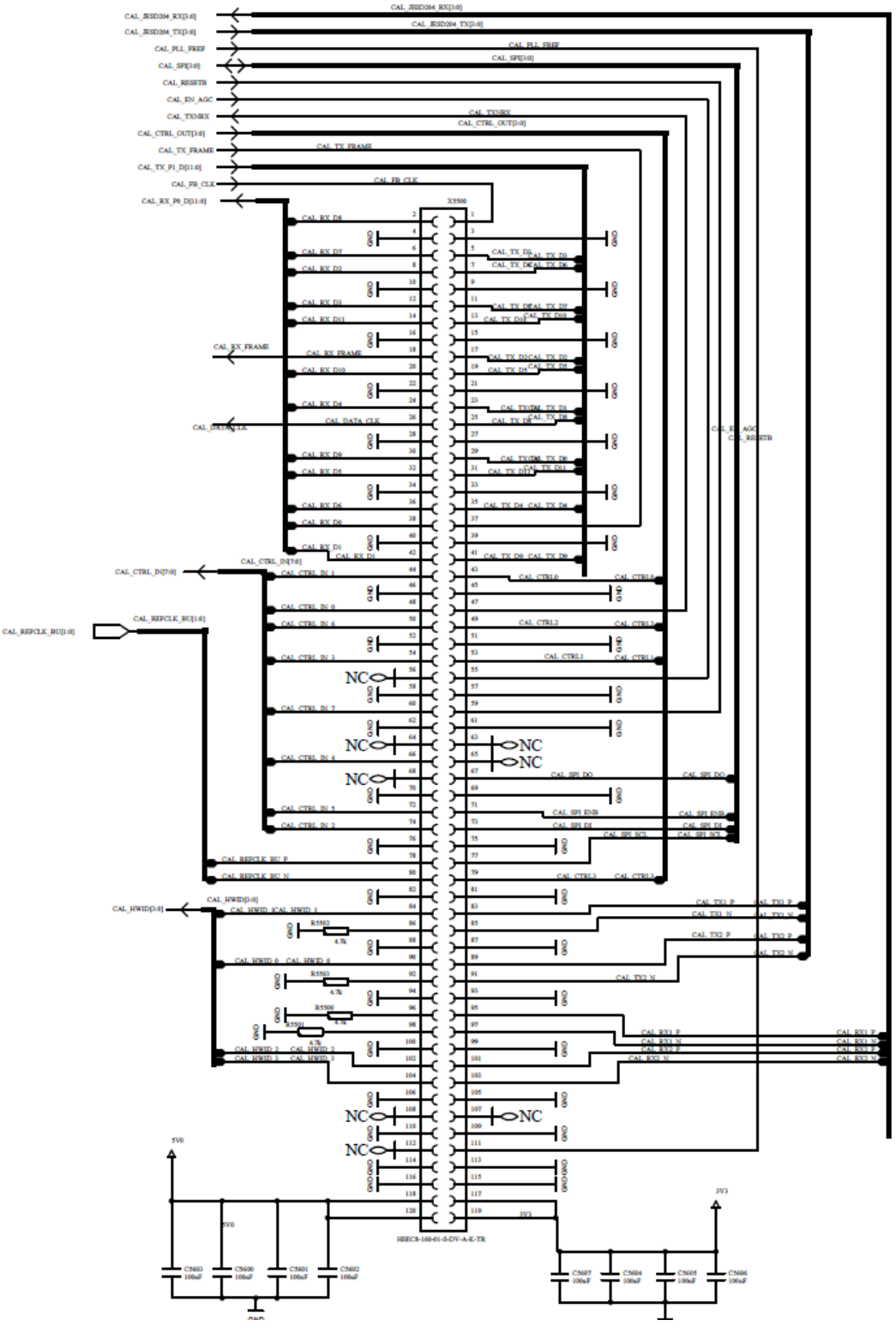

Figure 37. The Calibration interface.

Table 7. Calibration signals

Signal	Description
CAL_RX_P0_D[11:0]	Received data bus (EFM → CR)
CAL_TX_P1_D[11:0]	Transmitted data bus (CR → EFM)
CAL_CTRL_IN[7:0]	Received control bus
CAL_CTRL_OUT[3:0]	Transmitted control bus
CAL_SPI[3:0]	SPI bus for monitoring
CAL_DATA_CLK	Received data clock
CAL_FB_CLK	Transmitted data clock
CAL_RX_FRAME	Received frame indication
CAL_TX_FRAME	Transmitted frame indication
CAL_TXNRX	TX/RX burst control
CAL_ENABLE	Burst start & stop
CAL_EN_AGC	Enable RX automatic gain control
CAL_RESETB	Hardware reset
PLL_FREF	Reference frequency for the Calibration radio
CAL_JESD204_RX[3:0]	Received JESD204 data bus
CAL_JESD204_TX[3:0]	Transmitted JESD204 data bus

The calibration interface is connected to the Calibration Radio (CR). The calibration radio is used to create calibration RF and IQ signals for the active antenna beam width tuning and adjustment.

3.2.5. OCXO #2, PLL #3 and PLL#4

Two new PLLs and one OCXO generate clocks for the FPGA #2. The OCXO #2 also includes several other components, like a temperature sensor, Digital-Analog (DA) -converter, an operational amplifier and a clock divider. The OCXO #2 is controlled by the FPGA #2 with the data of the temperature sensor, which monitors the oven temperature. The temperature sensor informs the FPGA #2 if the OCXO #2 oven temperature is too high or too low.

The DA converter receives digital control data from the FPGA #2 and converts it into an analog voltage at the voltage output. The DAC's voltage output is connected to the operational amplifiers non-inverted input. The amplifier acts as an output buffer for the DAC for driving heavy resistive loads, which in this case is the OCXO #2. The amplifiers output voltage is connected to the OCXO #2's control voltage pin via 1 kΩ -resistor. The OCXO's output frequency can be controlled with the control voltage. For example, when the control voltage is increased, the output frequency increases as well and vice versa. The OCXO #2 receives an output reference voltage from the DAC, which is compared to the voltage level of a 30.72 MHz clock. The generated 30.72 MHz clock is in the single end form so it is fed to the clock divider which converts it into the differential form. After this, the differential 30.72 MHz clock is sent to the FPGA #2. The OCXO #2 also generates an oven alarm signal, which is in the low state if the oven is not working correctly. Figure 38 shows above-mentioned devices.

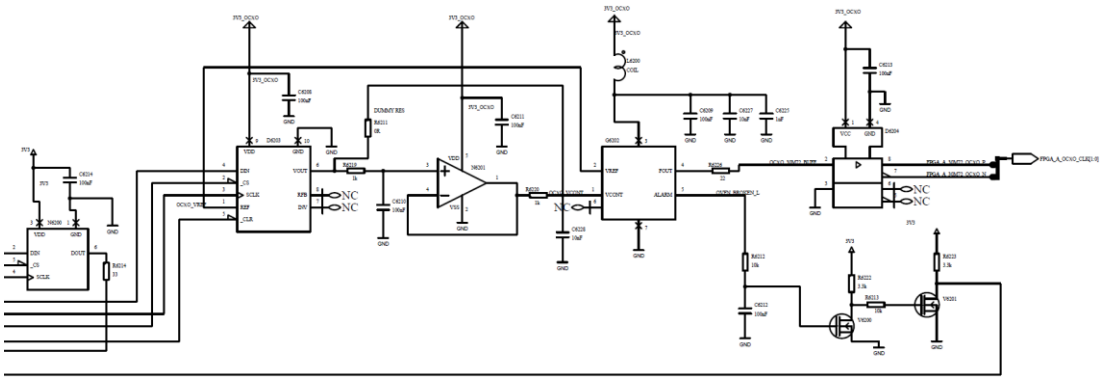

Figure 38. From left to right: the Temperature sensor, the DAC, the buffer, the OCXO #2 and the clock divider.

The PLL #3 generates several differential clock signals from the selected input reference clock. The generated clock frequencies are already discussed in the previous chapter, on page 29. The PLL #3 has two 30.72 MHz reference clocks, primary from the PLL #2 and secondary from the FPGA #2. The reference clock can be selected with an FPGA #2 control bus signal called PLL3_REF_SEL. Eight differential clocks are generated from the chosen reference clock to the output ports Y0-Y7. From there, the differential clock signals are sent to the FPGA #2. The control bus also includes PLL3_RESET_N, PLL3_SYNCN and PLL3_STATUS -signals.

When the FPGA #2 sets the PLL3_RESET_N-signal low, the PLL #3's outputs are disabled to avoid any false frequencies when the EFM is powered up. During configuring via the SPI bus, the reset cannot be used, so the PLL3_SYNCN-signal is used instead to disable the device's outputs. The PLL3_STATUS -signal is used to send PLL #3's current status to the FPGA #2. There is a SPI between the FPGA #2 and the PLL #3 which is used by the FPGA #2 to monitor and configure the PLL #3 when necessary. PLL #3 uses 1.8 V supply voltage. The PLL #3 is presented in Figure 39.

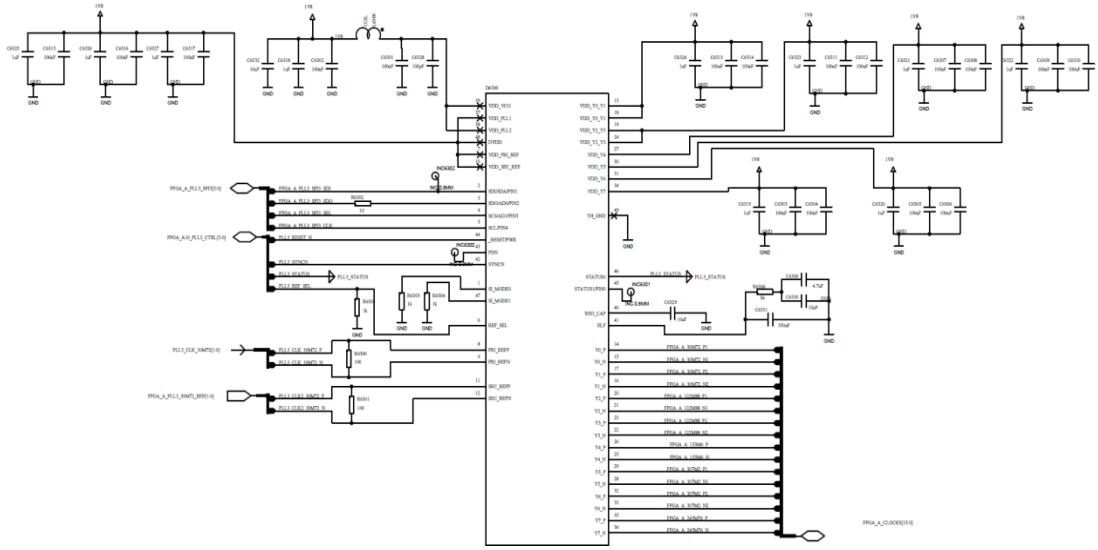

Figure 39. PLL #3.

The PLL #4 is a different device than the PLL #3 but basically it works in the same way: There are two input reference clock signals to choose from. The frequencies of the input reference clocks are 30.72 MHz (like in the PLL #3 case) and 122.88 MHz. One of these clocks is chosen to be the source for the generated output clocks. There are ten differential output clock ports, which are connected to 0 ohm resistors. By removing these resistors, the output clocks can be disconnected from the FPGA #2.

The FPGA #2 can control the PLL4 with the RST_PLL4, the PLL4_STATUS and the SPI signals. The SPI bus is connected to the PLL #4 via 0 ohm resistors and it is used to monitor and configure the PLL #4. The PLL #4 configuration can also be done with external Electronically Erasable Programmable Read-Only Memory (EEPROM). This can be done by removing SPI bus 0 ohm resistors and connecting EEPROM into the EEPROM SPI interface pins. The pins can be found at the lower right corner of the PLL #4.

The RST_PLL4-signal is used to reset the PLL4's internal circuitry to the default values. The FPGA #2 receives the PLL4's current state via the PLL4_STATUS-signal. The PLL #4 also has master clock pins, where an external high speed oscillator can be connected. The single-ended clock generated by the oscillator is used as a reference for the PLL #4's inner blocks. There are also JTAG pins, which can be used to test the PLL #4. The PLL #4 is presented in Figure 40.

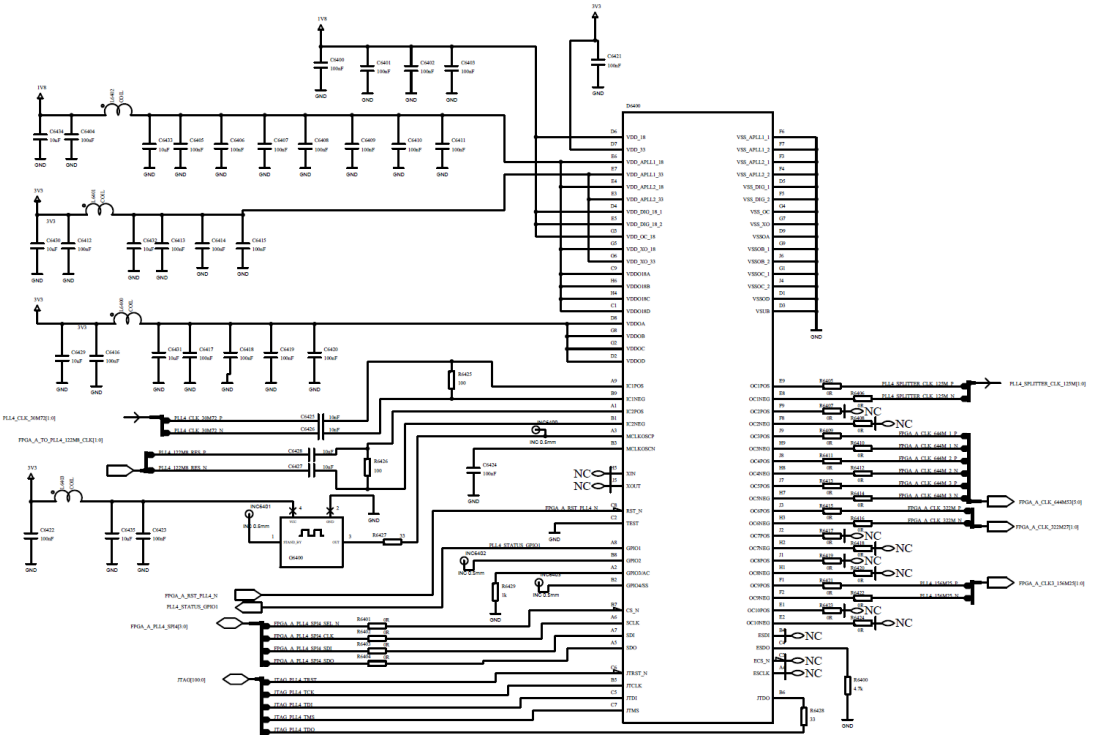

Figure 40. PLL #4

3.2.6. FPGA #2

Large devices, which include several hundreds of pins, are divided into symbol blocks. This helps drawing of the schematics and makes schematic sheets much more readable. The FPGA #2 is divided into the seven symbol blocks in DxDesigner. These blocks are listed below. There are also the second FPGA's DDR3 and Flash – memories that are covered in their own sheets after the FPGA #2 blocks.

- Configuration symbol
- Input/Output bank symbols 1-3
- Transceiver bank symbols 1-2
- Reference symbol

3.2.6.1. FPGA #2 configuration

The FPGA #2's configuration symbol block is presented in Figure 41. The configuration pins are located on the right side of the block and on the left are the pins of the IO-banks 2A and 4F. The JTAG connector is connected to the FPGA #2's JTAG pins, which are used for configuration and testing. Below the JTAG pins are the Mode Select (MSEL) pins 0-2, which are used to select the configuration scheme by connecting them to a 1.8 V voltage or ground with pull-up or pull down resistors. A standard Active Serial (AS) configuration is used by connecting pins 0 and 1 to the 1.8 V and pin 2 to the ground. The FPGA #2's internal pull-ups can be determined with low active Input/Output Pullup (nIO_PULLUP) pin. The configuration IO-pins have 25 kΩ internal pull-up resistors. The nIO_PULLUP-pin is connected to the 1.8 V voltage with 1 kΩ resistor.

The low active Status (nSTATUS) pin is connected to the CPLD and it is used to inform if any errors happened during the FPGA #2 configuration. If an error has occurred, the CPLD resets the FPGA #2. There is also a Configure Done (CONF_DONE) pin, which informs the CPLD when the FPGA #2 has received the configuration data. When the data is received, the FPGA #2 sets the pin high; otherwise, it is kept low. The CPLD configuration control signal is connected to the FPGA #2's low active Configure (nCONFIG) pin. When this pin is pulled low, the FPGA #2 loses its configuration data and enters a reset state. After the CPLD releases the pin to a high state, the reconfiguration is initiated. The enabling of the FPGA #2 is done by setting the low active Chip Enable (nCE) pin low state. Accordingly, setting nCE pin high state disables the FPGA #2

The low active Chip Select Output (nCSO) pins 0-2 are used to enable the Flash memory in the AS configuration scheme. Because only one Flash memory is used, only one nCSO pin is needed. The configuration data for the Flash device in the AS configuration is transmitted through Active Serial Data (AS_DATA0 – AS_DATA3) pins. There are two modes in the AS configuration, the one bit (1x) and the four bit (4x) -mode. The 1x mode means that only the AS_DATA0 pin is used for the configuration data transmission. In the 4x mode, all four data pins from 0 to 3 are used in configuration. In the AS configuration, the Data Clock (DCLK) pin is used as an output from the FPGA #2 that provides timing for the AS configuration.

On the left side of the symbol block, there is a 100 MHz clock connected to the FPGA #2's Clock User (CLKUSR) pin. The FPGA #2's transceivers are calibrated

via this pin and the 100 MHz clock is received from the synchronization clock – block. The pins in the lower right corner of the symbol are used in another configuration scheme called Passive Serial configuration (PS). This scheme is not used though so the pins are left unconnected.

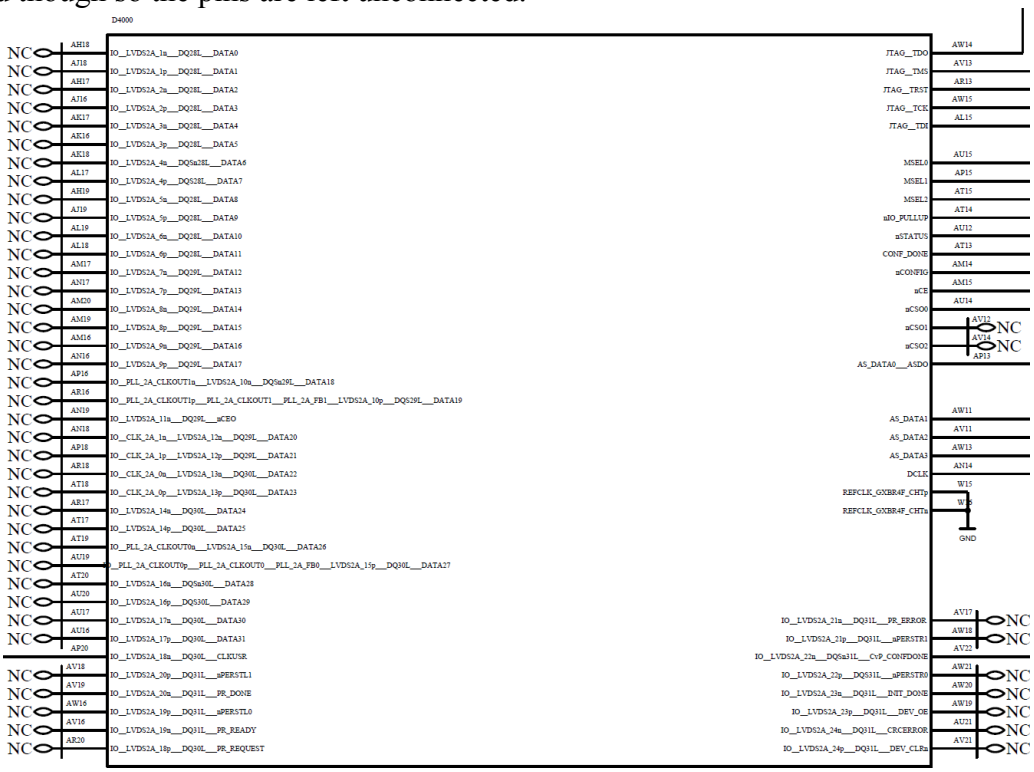

Figure 41. The FPGA #2 configuration symbol block.

The FPGA #2 can be configured with several schemes but in this case only two configuration methods are enabled. In the AS configuration, the configuration data is stored in the Flash memory. Programming can be done using the JTAG interface, by using the FPGA #2 as a bridge between JTAG interface and the Flash memory. After this, the Flash device configures the FPGA #2 when the FPGA #2 and the Flash memory are powered up in the next time. The configuration data moves between the data pins 0, 1, 2 and 3 of the Flash and the FPGA #2 while the DCLK provides timing for the data. The Flash device is enabled via active low Chip Select (NCS) pin. The Flash device's block symbol is shown in Figure 42.

Another used configuration method is the JTAG-based configuration, where the FPGA #2 is configured via JTAG signals Test Clock (TCK), Test Data In (TDI), Test Data Out (TDO), Test Mode Select (TMS) and Test Reset (TRST). MSEL pin settings do not matter in this configuration method but when using JTAG-based configuration, the nCE pin must be connected to the ground. Signals are connected to the FPGA #2 pins, which use the same names as the signals. The FPGA #2 configuration symbol block also includes IO banks 2A and 4F, which are not used in this case.

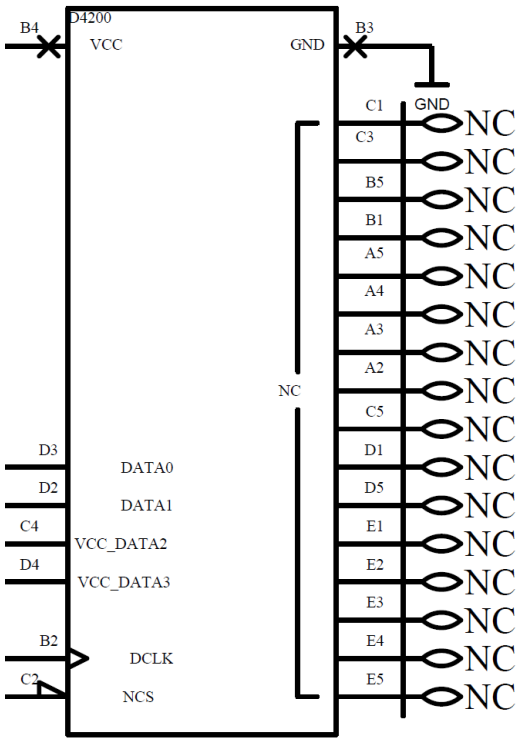

Figure 42. The Flash memory.

3.2.6.2. FPGA #2 input/output

The FPGA #2 has total of 14 General Purpose Input/Output (GPIO) banks, which are divided across three symbol blocks. The GPIO banks consist of Low Voltage Differential Signaling (LVDS) I/O pins, which support differential and single-ended I/O 1.8 V voltage standards. These banks are not the same size and some of them have less I/O-pins than the others. Each LVDS pin can be used as a transmitter or receiver. Every bank includes eight pins for the synchronization clocks generated by the PLLs and the OCXO #2. The layout design was also considered when connecting signals to the FPGA #2's I/O pins to avoid unnecessary signal path crossings.

The first I/O symbol consists of banks 2I, 2J, 2K, 2L and two-thirds of the 3H. All pins of the 2I and one-third of the 2J pins are used for the QSFP+ control signals, presented in Table 2. The SRIO switch and the Minisas control signals are also connected to the 2J bank. The 2K bank is reserved for the control signals of the SFP+ connectors. The control signals voltages from the SFP+ are level shifted from 3.3 V voltage to 1.8 V voltage because the FPGA #2's I/O-pins use the lower voltage. Two kinds of level shifters used in the EFM: unidirectional and bidirectional. The unidirectional level shifters are used to translate almost all of the control signals except the I2C and the SPI signals. The I2C and the SPI signals are two-way, so the bidirectional level shifters are required. The second FPGA's DDR3 memory's 15 bit address, 6 bit control and the 16 bit data bus are connected to the banks 2L and 2K. All DDR3 signals are explained in the DDR3 memory chapter in this Master's thesis. Figure 43 presents the FPGA #2's first I/O symbol block.

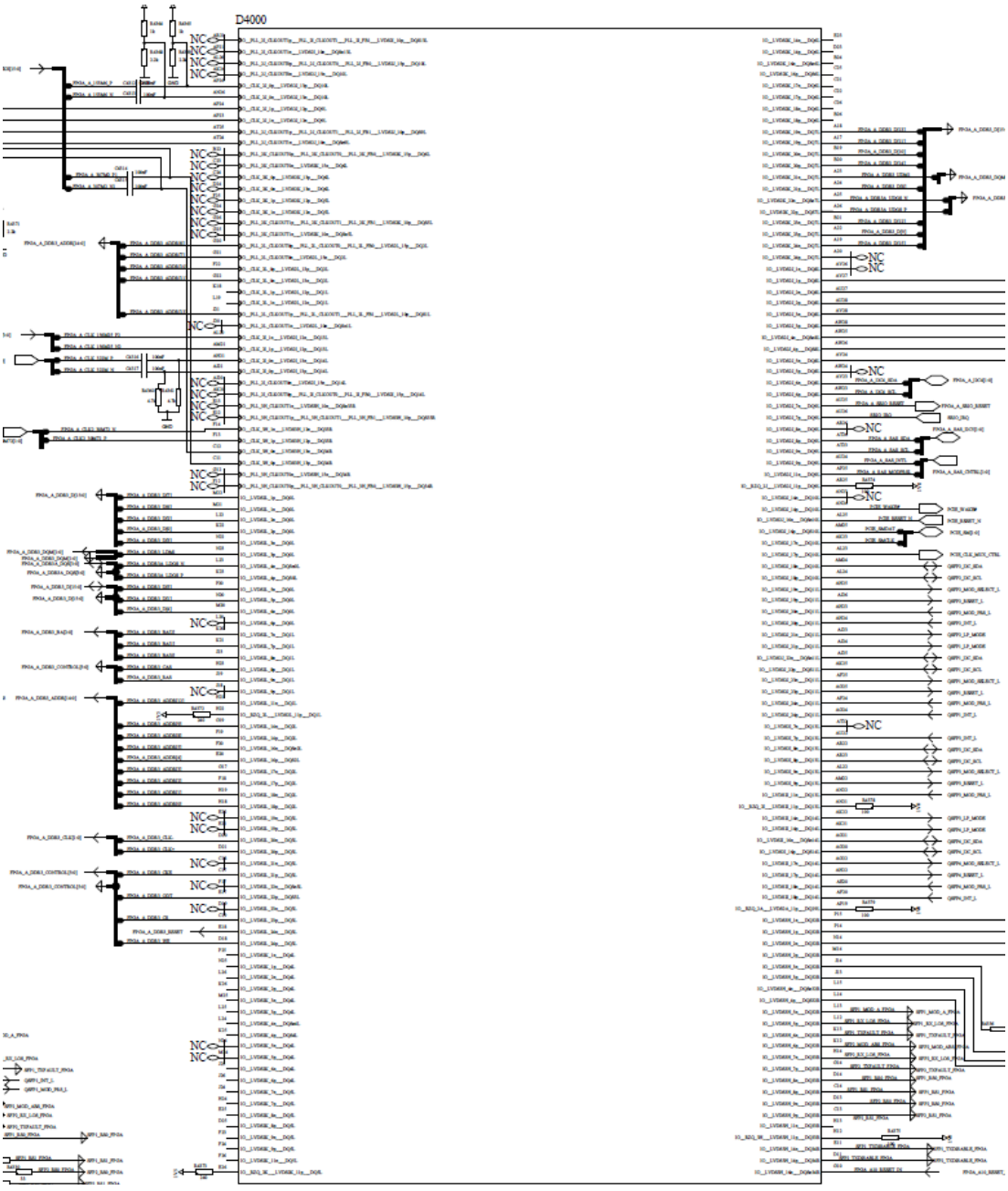

Figure 43. The FPGA #2's I/O symbol block 1.

The second I/O symbol block consists of banks 3D, 3E, 3F and 3G and the rest of the 3H bank. The RP1 signals between the FPGAs are connected to the 3D and the 3E bank clock pins. The calibration radio connector is connected to the banks 3G and 3E. The PLL SPI and control signals are connected to the FPGA #2's bank 3F. The PLL #3 SPI and control signals are 1.8 V, so they are connected straight to the FPGA #2 but the PLL #4 SPI signals are 3.3 V and require level shifting. The OCXO #2's control data signals are connected to the 3F bank's pins as well. The 3D bank is used to connect the SPI signals between the FPGAs and because the FPGA #1's I/O pins use 2.5 V voltage level translation is needed. Remaining pins of the bank 3D are reserved for the GPS module's UART and 1PPS signals. The second I/O block is presented in Figure 44.

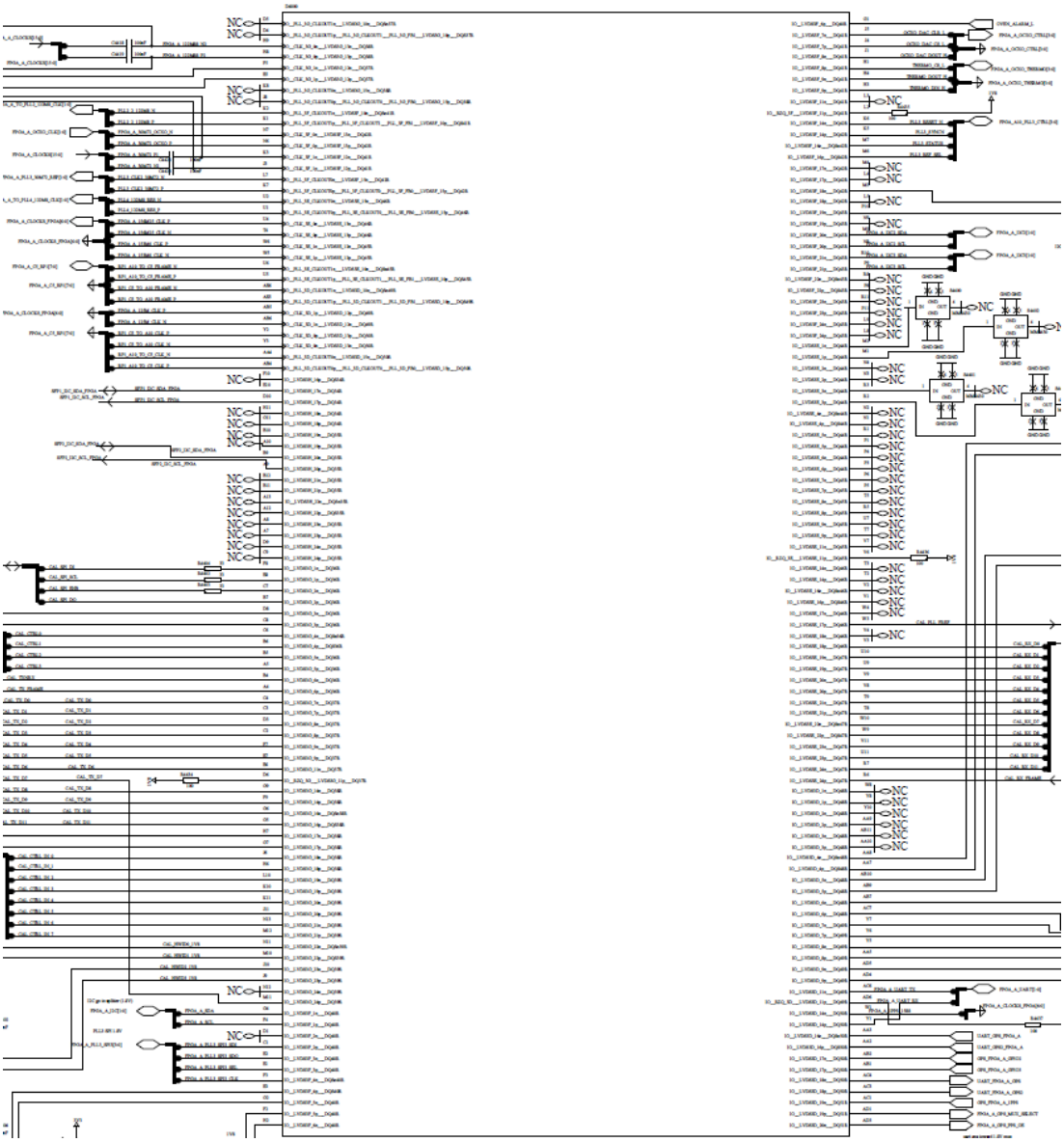

Figure 44. The FPGA #2's I/O symbol block 2.

The I/O banks 3A, 3B and 3C are in the FPGA #2's third I/O symbol block. Most of the pins of these three banks are not used. Mainly the synchronization clock pins and few pins from bank 3B are in use. The 125 MHz, the 30.72 MHz and couple of reference clocks and the RP1 signals between the DSP and the FPGA #2 are the signals connected to the FPGA #2's clock pins. A reset signal for the FPGA #2's Ethernet transceiver, multiplexer selection control and some DSP signals are connected to the 3B bank. Figure 45 presents part of the third I/O symbol block.

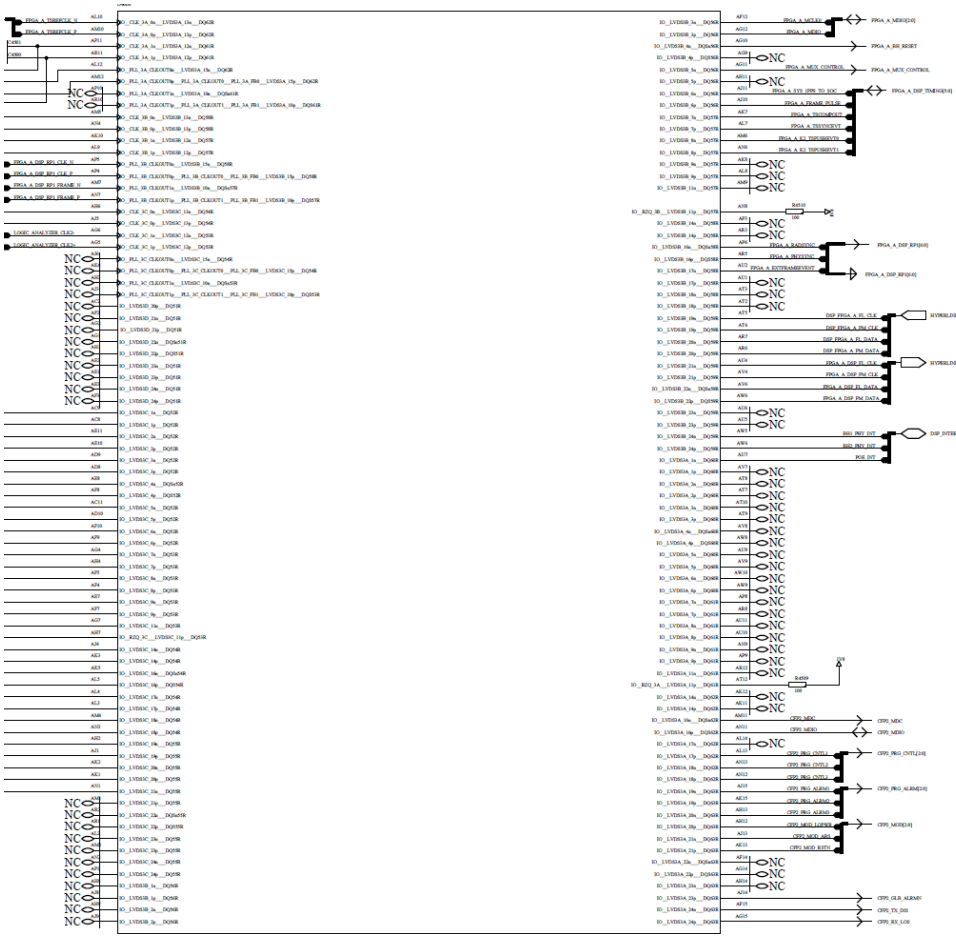

Figure 45. The FPGA #2's I/O symbol block 3.

3.2.6.3. FPGA #2 transceivers

The FPGA #2 includes 48 transceiver channels, which are divided in eight banks. Each transceiver bank consists of six differential transceiver channels and two reference clock channels. These transceiver channels are used for supporting data rates from 1 Gbps to 28.3 Gbps. The high speed differential reference clock channels are used for core clock generation. The transceiver banks are divided into two symbol blocks in the DxDesigner and both include four transceiver banks. The transceiver banks are named the same way as the GPIO banks. Figure 46 presents the FPGA #2 transceiver symbol block 1, which includes some connected optional transceiver links. These optional links are not use in the EFM.

The 1J bank connects the Ethernet links from the DSP to the FPGA #2. One 644.53 MHz clock and one 125 MHz clock are connected into the reference clock pins. The 1J bank is located at the upper left corner of the symbol while the bank 1I, which connects RP3 links coming from the DSP is located at the upper right corner. The 30.72 MHz reference clock is connected to the 1I bank's reference clock pair while the other pair is tied to ground. The QSFP+ #4-connector's transceiver links, two calibration radio transceiver links and one 122.88 MHz reference clock are connected to the 1H bank. The 1H bank is located at the lower left corner of the transceiver symbol block. The 1G bank is located at the lower right corner and it

receives 156.25 MHz and 644.53 MHz reference clocks. The transceiver links connected to the 1G bank are optional but can be taken into use if desired.

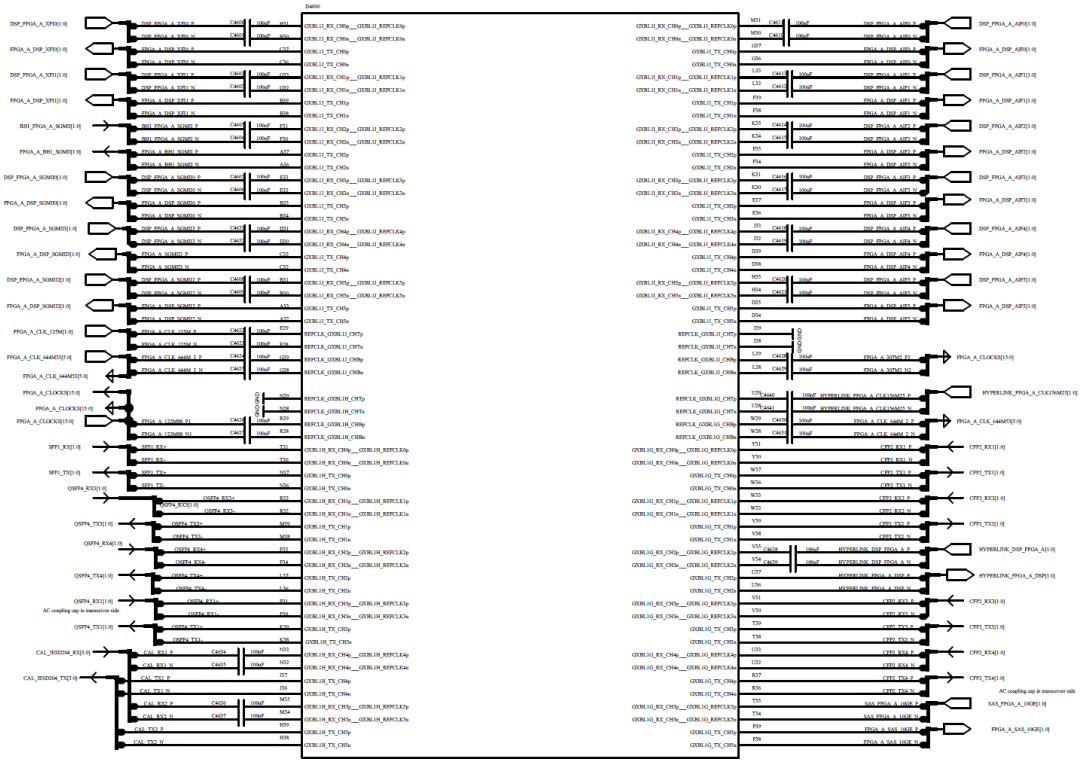

Figure 46. The FPGA #2's transceiver bank 1.

The second transceiver symbol block of the FPGA #2 is presented in Figure 47. It consists of banks 1F, 1E, 1D and 1C connecting the SFP+ connectors and rest of the QSFP+ connectors, the SRIO switch and the Minisas connector to the FPGA #2. The QSFP+ #1 connector's four transceiver links and 156.25 MHz reference clock are connected into the bank 1F with two optional TRX links. The 1F bank can be found on the upper left corner of the symbol. The 1E bank is used to connect the QSFP+ #2 TRX links. It also has two optional links and an optional reference clock. This 1E bank is located at the upper right corner of the symbol. The QSFP #3's transceiver links are connected to the 1D bank, which is located in the lower left corner. The 1D bank also includes the SRIO switch data links and the Ethernet links for the Minisas connector. The 245.76 MHz reference clock is connected to the 1D bank. The last transceiver bank is the 1C at the lower right corner of the symbol. It connects the SFP+ transceiver links to the FPGA #2. The 1C bank is includes two optional TRX links, one 644.53 MHz clock and optional reference clocks.

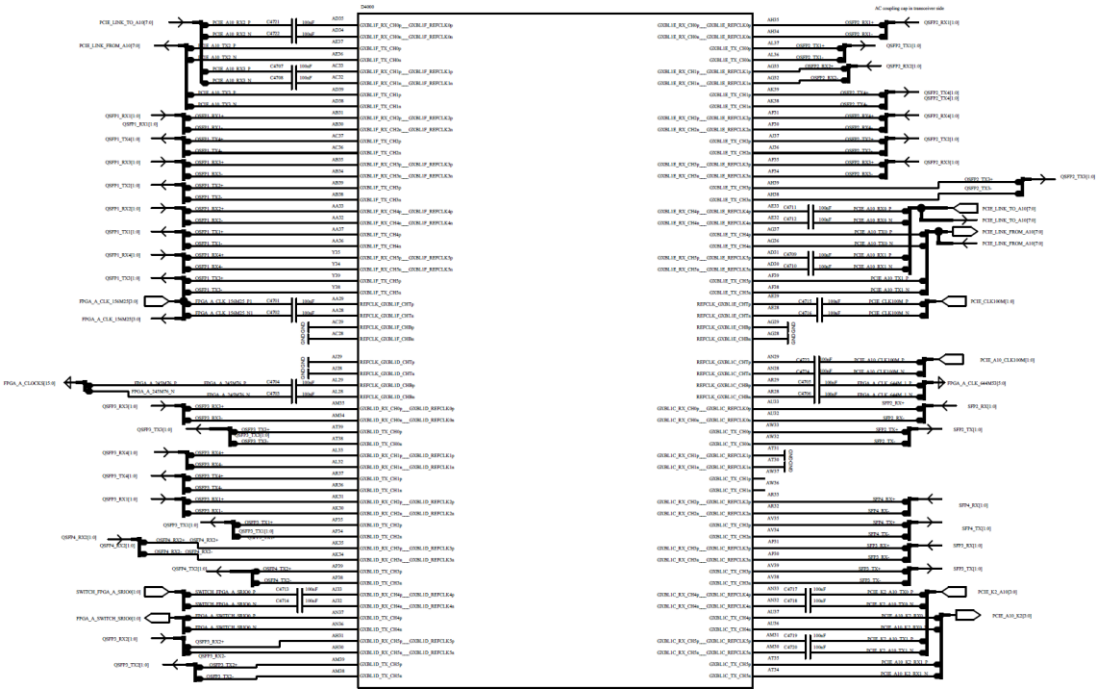

Figure 47. The FPGA #2's transceiver bank 2.

3.2.6.4. FPGA #2 reference

The FPGA #2 has one more block which is used to: set the reference voltage for the IO and transceiver pins, monitor external voltages connected to the FPGA #2, monitor FPGA #2's internal temperature and battery back-up. The transceivers are divided into four sections, Bottom-Left (BL), Bottom-Right (BR), Top-Left (TL) and Top-Right (TR) and the FPGA #2 has Resistor Reference (RREF) pins for each section. These pins are called RREF_BL, RREF_TR, RREF_BR and RREF_TL. These are connected to the ground with two 1 kΩ resistors, when the FPGA #2's transceivers pins are used.

There are also two differential Voltage Sensor Input (VSIG0 and VSIG1) pins. These pins are used to monitor FPGA #2's external analog voltages but this feature is not used in this case. The FPGA #2 has an internal temperature sensing diode, which can be monitored with an external temperature sensor. The sensor can be connected to Temperature Diode (TEMPDIODE)p/n pins. The FPGA #2's internal temperature can be monitored with these pins. The second FPGA has a pin for the Battery back-up called VCCBAT for battery back-up. It is used to power a design security volatile key. The VCCBAT pin is connected to the 1.8 V voltage, which will leave the design security volatile key unused.

The Voltage Reference (VREF) pins on the right side of the reference block symbol provide an input for a reference voltage for each I/O bank. Mainly, these VREF pins are not used, so they are connected to ground. The IO bank, where the DDR3 memory is connected, uses a 0.75 V reference voltage, called DDR3_VREF. There is also optional 25 MHz clock generated by the 25 MHz oscillator. The FPGA #2 reference symbol block is presented in Figure 48.

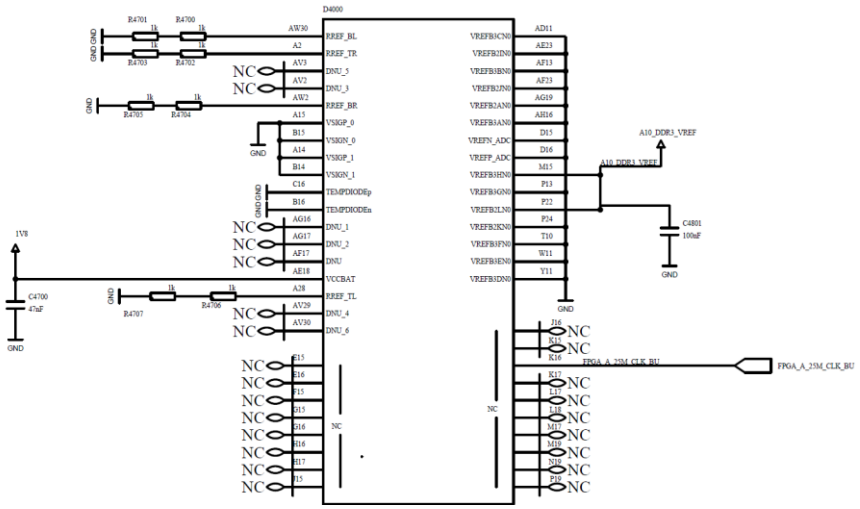

Figure 48. The FPGA #2's reference symbol block.

3.2.6.5. FPGA #2 DDR3 SDRAM

The DDR3 memory device is connected to the FPGA #2 with the following signals:

- DDR3_ADDR[13:0] (DDR3 Address)
  - DDR3 memory's 14-bit address bus is used to point the location where the data is written or read.
- DDR3_DQ[15:0] (DDR3 Data)
  - Data between the FPGA #2 and the DDR3 device is moved with a 16-bit data bus.
- DDR3_BA[2:0] (DDR 3 Bank Address)
  - The DDR3 memory's internal data bank can be chosen with this signal.
- DDR3_CLK[1:0] (DDR3 Clock)
  - Differential system clock is used to synchronize the DDR3 memory.
- DDR3_DQS[3:0] (DDR3 Data Strobe)
  - Data strobe, which is used to capture data. In READ mode, the DQS acts as an additional data output; in WRITE mode, the DQS acts as a clock to capture data.
- DDR3_DM[1:0] (DDR3 Data Mask)
  - DM is used to mask the data which is written to the DDR3 memory.
- DDR3_CONTROL[5:0] (DDR3 Control)
  - DDR3_CKE (DDR3 Clock Enable) enables clock on the DDR3 memory.
  - DDR3_ODT (DDR3 On-Die Termination) enables internal termination resistance to the DDR3 memory.
  - DDR3_RAS (DDR3 Row Address Strobe) actuates row address oriented internal circuitry.
  - DDR3_CAS (DDR3 Column Address Strobe) actuates column oriented internal circuitry.

    - DDR3_WE (DDR3 Write Enable) enables writing into the DDR3 memory
    - DDR3_CS (DDR3 Chip Select) selects the chip what is going to be used.
- DDR3_RESET (DDR3 Reset)
    - Resets the DDR3 memory device.

The DDR3 memory symbol is presented in Figure 49. The size of the memory is 128 Mbit times 16, equaling 256 Mbytes. This is enough for the FPGA #2, but if more memory is needed, an even larger FPGA is needed as well because there are not enough pins for more memory in the current FPGA #2. The DRR3 address bus connected to the left side of the memory device on Figure 49. Below the address bus are the rest of the busses except for the data bus, which is located on the right side of the memory.

The memory device also includes the Voltage Reference for Control, Command and Address (VREF_CA), Voltage Reference for Data (VREF_DQ) and External reference for output calibration (ZQ) pins. The VREF_CA pin provides a reference voltage for control, command and address inputs. Similarly, VREF_DQ provides a reference voltage for the data inputs. The FPGA #2's DDR3_VREF supply voltage is connected to these reference voltage pins. The external reference pin, ZQ is used for output drive calibration. The DDR3 SDRAM use programmable impedance output buffer, which is calibrated by connecting ZQ pin to ground with a 240 ohm resistor. The resistor defines the output driver impedance Ron.

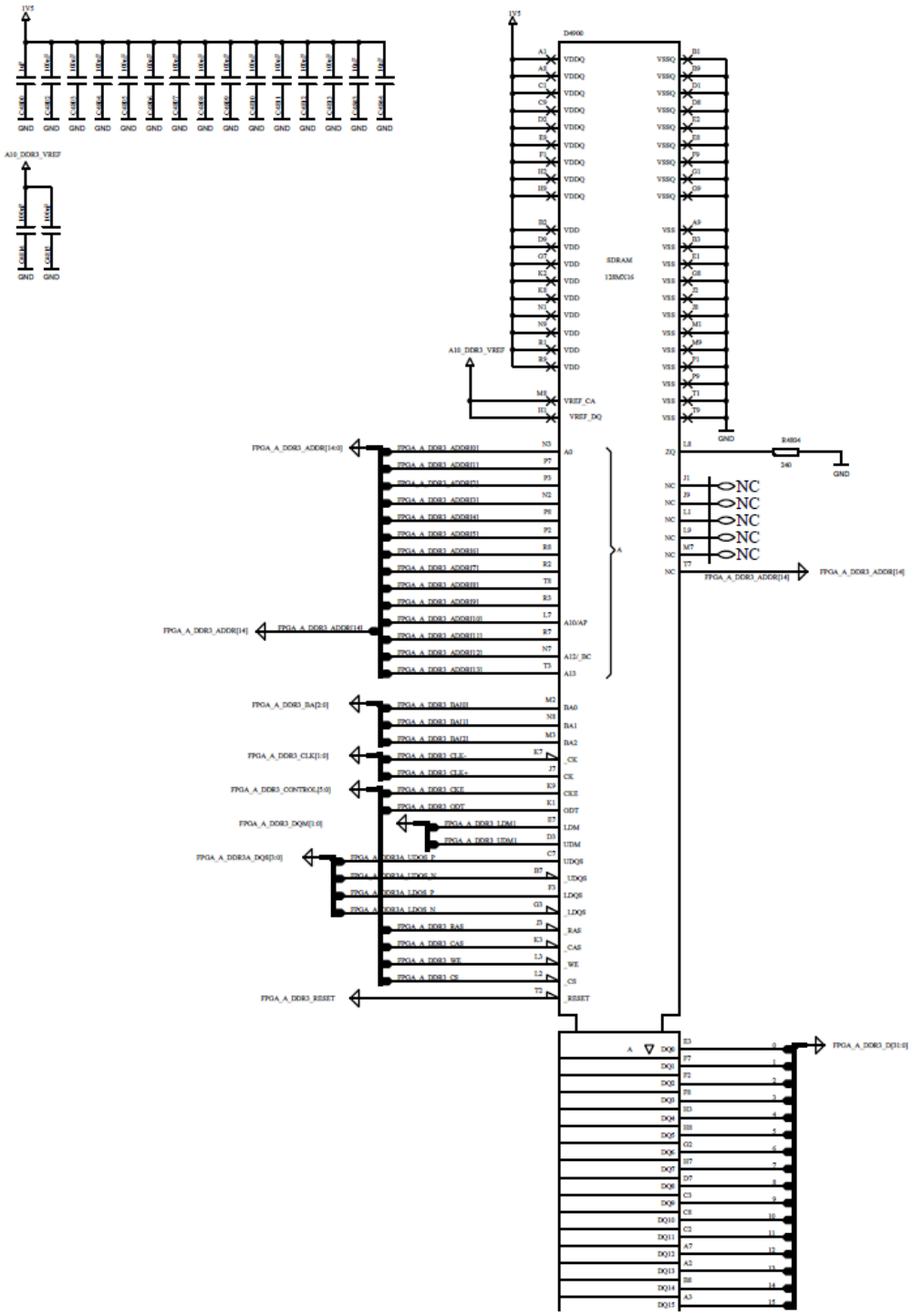

Figure 49. The DDR3 memory device.

4. CONCLUSIONS

The target of this Master's thesis was to implement a device for hardware and software development. The device itself was based on small base station called the Flexi Zone Micro. The main purpose for using older design as a reference was the reuse of the hardware and software. This saved work hours. The Flexi Zone Micro was chosen because it had a suitable digital circuitry and software for this situation.

The architecture specification of the Flexi Zone Micro was used as a reference document for the Implementation specification. After all the basic specifications, like interfaces, were defined; the search for the suitable components began. The components were chosen, so that they could manage only the functions that were pointed to them and nothing more. This kept the manufacturing costs as low as possible. The components can be updated to different ones in other Ethernet Fronthaul Module variants.

Even during the design process, some already chosen components were changed. The DDR3 SDRAM's quantities were changed few times: The DSP's DDR3A memory was increased from 2 GB to 4GB and DDR3B from 1 GB to 2GB. The second FPGA's DDR3 memory was decreased from 512 MB to 256 MB. Some of the connectors were also changed. These changes affected the drawing of the schematics. For example, signal and bus quantities were altered.

The Ethernet Fronthaul Module designed in this Master's Thesis work will be used as a development platform for the wireless telecommunication design solutions in the future. In the Cloud RAN architecture, the Ethernet Fronthaul Module would be embedded into the Remote Radio Head, which enables L1 baseband processing in the RRH -board. This would lead to the partial centralization C-RAN architecture. It is possible to move layers: L2 and L3 and operation management to the RRH –board in the future. This would mean that the BTS functions could be done without the BBU pool.

The Ethernet Fronthaul Module can be used in an indoor BTS solution's development as well. For example, the master EFM could be connected to the several slave EFMs which are connected to small indoor base transceiver stations called Pico cells. In this way, a wireless network can be distributed to every floor of the large apartment building or large shopping centre where every floor has several Pico cells and a one slave EFM. The master EFM could be located at the first floor or at the top floor of the building.

The indoor BTS setup presented above and the AAS require external interfaces to connect the EFM and the Pico cells or the AEs together. The current maximum data rate in the SFP+ and QSFP+ interfaces used in the EFM is 10 Gbps. In the future, these external interfaces are replaced with zSFP+ and zQSFP+ external interfaces which can support 25 Gbps data rates.

Currently, only a few tests are done to the EFM because the software development of the FPGA #2 is still ongoing, so data conversions, L1 processing and AAS features are not working. These made tests include: visual check for the EFM board, programming and configuring the power manager, the CPLD, the FPGA #1, the DSP and the FPGA #2. It should be noted that the FPGA #2 was programmed with incomplete software. Each device is programmed and configured with a specified programming adapter and computer software. The adapters are connected between a test computer and the EFM board. The adapter, programming software and the programmable device itself are coming from the same manufacturer.

After configuring the power manager, the functional tests for the power supplies were done. During the functional tests, it turned out that the DSP took too much current from the 0.85V power supply which crashed during power up. This problem was solved by removing few voltage filter capacitors from the DSP. Other solution was to change the 1.8V power supply to the higher performance power supply. Other power supplies passed the functional tests.

The CPLD's functionality testing included the FPGA #1 and the DSP flash programming, a reset testing and the watchdog testing. Programming the FPGA #1 and the DSP flash were successful meaning that the CPLD passes through the programming data as it should. Reset control was tested by pressing the factory reset button and connecting the test computer to the EFM board through the µUSB connector. Reset commands are written to the CPLD via the DSP. The Watchdog provided reset for the EFM during the DSP's Ethernet link testing.

The EFM synchronization clocks' waveforms and frequencies were checked with an oscilloscope. The clocks generated by the both OCXOs, PLLs #1 and #2 and 122.88 MHz system clock generated by the FPGA #1 were measured. The FPGA #1's 122.88 MHz system clock was measured to ensure that the FPGA #1 is working correctly. The PLL #3 and PLL #4 were not tested because both of them are driven by the FPGA #2. Until the FPGA #2's software is ready, the PLL #3 and the PLL#4 are in the reset state. All of the clocks had square waveforms and frequencies were also correct except the one clock which had a frequency of the 30.72 MHz instead of the 153.6 MHz. This clock is generated by the PLL #2 and it is currently under investigation.

The DSP was tested by forming the Ethernet connection between the test computer and the DSP through the DSP's RJ45 connector. An Electronic Identity (EID) and boot image were downloaded to the DSP's boot flash. The FPGA #2 testing is mainly been the AS and the JTAG configuring. The FPGA #2 and the EFM testing will begin when the FPGA #2's software is ready.

5. SUMMARY

This Master's Thesis covers the design procedure of the device called Ethernet Fronthaul Module. The EFM is designed for the wireless telecommunication hardware and software development. The Master's Thesis itself is divided into the two chapters: the theory and the implementation of design.

The theory chapter covers: The C-RAN, the high data rate interfaces, the AAS and the EFM. The cloud radio access network's architecture presents the direction for the new wireless telecommunication standard development. The basic operation and different setups of the Active Antenna System are presented. The high speed data interfaces and replacing them with higher data rate interfaces are covered. Finally, the theory of the designed Ethernet Fronthaul Module is explained in the theory chapter.

The implementation of design chapter covers: the HW Implementation specification -chapter and the schematics chapter. The HW Implementation specification chapter presents the used components in the EFM. The internal interfaces between the components of the EFM and the external interfaces from the EFM to the RRH and the BBU are covered in the HW Implementation specification. The schematics chapter presents the schematic symbol blocks and covers the operation of the new components.

6. REFERENCES

[1] China Mobile Research Institute (2011) C-RAN The Road Towards Green RAN. URL: http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper_v2_5_EN.pdf

[2] Open Base Station Architecture Initiative (2006) BTS System Reference Document Version 2.0. URL: http://www.obsai.com/specs/OBSAI_System_Spec_V2.0.pdf

[3] Open Base Station Architecture Initiative (2010) Reference Point 3 Specification. URL: http://www.obsai.com/specs/RP3_Specification_V4.2.pdf

[4] Common Public Radio Interface (2013) CPRI Specification V6.0. URL: http://www.cpri.info/downloads/CPRI_v_6_0_2013-08-30.pdf

[5] Huawei (2012) Active Antenna System: Utilizing the Full Potential of Radio Sources in the Spatial Domain. URL: http://www.huawei.com/en/static/AAS-129092-1-197969.pdf

[6] Nokia (2014) Flexi Multiradio Active Antenna System. URL: http://networks.nokia.com/portfolio/products/mobile-broadband/single-ran-advanced/flexi-multiradio-active-antenna-system

[7] Nokia (2014) Flexi Zone: The most versatile small cell solution today. URL: http://networks.nokia.com/portfolio/solutions/heterogeneous-networks/flexi-zone

[8] What is 25 Gigabit Ethernet and why would you want it? (2014) URL: http://www.plexxi.com/2014/07/whats-25-gigabit-ethernet-want/#sthash.RAIfeitn.dpbs

[9] Definition – What does Small Form-Factor Pluggable Transceiver (SFP) mean? URL: http://www.techopedia.com/definition/24949/small-form-factor-pluggable-transceiver-sfp

[10] Ethernet (IEEE 802.3) URL: http://www.cisco.com/c/en/us/td/docs/net_mgmt/active_network_abstraction/3-7/reference/guide/ANARefGuide37/ethrnt.html

[11] Pluggable Input/Output Solutions (2012) URL: http://www.element14.com/community/docs/DOC-64353/l/sfp-board-mount-connector-by-te-connectivity

[12] RapidIO Interconnect Specification (2014) URL: http://www.rapidio.org/wp-content/uploads/2014/10/RapidIO-3.1-Specification.pdf

[13] What's The Difference Between Bit Rate And Baud Rate? (2012) URL: http://electronicdesign.com/communications/what-s-difference-between-bit-rate-and-baud-rate

[14] Serial-GMII Specification (2001) URL: http://www.angelfire.com/electronic2/sharads/protocols/MII_Protocols/sgmii.pdf

[15] IEEE P802.3ae MDC/MDIO (2001) URL: http://www.ieee802.org/3/efm/public/sep01/turner_1_0901.pdf

[16] Small Form-factor Pluggable (SFP) Transceiver Multisource Agreement (MSA) (2000) URL: http://www.schelto.com/SFP/SFP%20MSA.pdf