

Chapter 1

Introduction

The future upgrade of the Large Hadron Collider (LHC) accelerator, the Super Large Hadron Collider (SLHC), will increase the beam luminosity leading to a corresponding growth of the amount of data to be treated by the data acquisition systems. This will thus require high rate data links and high radiation tolerant Application Specific Integrated Circuits (ASICs).

To address these needs, the Gigabit Transceiver (GBT) architecture and transmission protocol was developed to provide the simultaneous transfer of readout data, timing and trigger signals in addition to slow control and monitoring data.

The GBT system can be described in two parts: First part of the system consists of radiation hard GBT ASICs that will act as detectors and will thus be located in the radioactive zone. These ASICs are used to implement bidirectional multipurpose 4.8 Gbit/s optical links for the high-energy physics experiments.

The last part of the system is located in the counting room and consists of the Common Readout Unit (CRU) that will provide an interface between the detector ASIC and an online computer-farm. The CRU consists of Commercial Off-The-Shelf (COTS) components (mainly an FPGA), and will through optical links receive the data from the radiation detector.

This thesis will mainly focus on the CRU software interface on the PC side, and the physical connection between the CRU and the Versatile Link (VLDB) card, where the radiation hard ASIC, the GBTx, is located.

1.1 Field Programmable Gate Array

A Field-Programmable Gate Array (FPGA) is a high density Integrated Circuit (IC) that is designed to be completely programmable by the customer after manufacturing (i.e. when the chip is shipped and "in the field"). The chips are shipped completely "blank", meaning that there are

no pre-programmed logic.¹ An FPGA can either be re-programmed using SRAM technology, or one-time programmed by burning antifuses. The latter method makes it less prone to soft errors when exposed to radiation.

FPGAs are composed of arrays of Configurable Logic Blocks (CLBs) surrounded by programmable routing resources and I/O pads. A CLB consists of a LookUp Table (LUT) together with a clock and simple write logic, and uses the address bus of the LUT as the function input pins and the value at the selected address as the function output. LUTs are considered fast logic, since computing a complex function only requires a single memory lookup. The LUT can be made out of an SRAM memory-block [?].

The FPGA (section 1.1.1) used in this thesis is a re-programmable type. It stores the user hardware setup in an SRAM memory to configure routing and logic functions (for a permanent program storage, the FPGA can be configured to store the hardware setup in the on-board flash memory, which then programs the SRAM when powered on).

1.1.1 Altera's Cyclone V GT

The FPGA used in this thesis is the Cyclone V GT by Altera. GT indicates that the FPGA has transceivers that supports speeds up to 6 Gbit/s [?]. It was chosen for this thesis mainly because of the on-board transceivers that are capable of reaching speeds that surpass the requirements of the GBT-FPGA Multi-Gigabit Transceiver (MGT), i.e 4.8 Gbit/s. Originally, a Terasic Cyclone V GX development board was intended to be used for this thesis. The Terasic board has advantages over the Cyclone V GT board in terms of communication with the outside world such as on-board Usb-to-Uart (more on this in chapter ??). However, it was discovered that the transceivers on the Terasic board were not fast enough for the GBT MGT; maximum supported transceiver speed is only 3.125 Gbit/s [?]. Because of this, the more powerful Cyclone V GT FPGA development board was ordered from the Altera web-pages, replacing the Terasic.

1.2 Hardware Description Language

The most common way to program an FPGA is by Hardware Description Language (HDL), with the major ones being SystemVerilog and VHDL. HDL consists of text-based expressions used to describe digital hardware and use this to further simulate and synthesize the hardware described. A hardware module is simulated by applying information at the inputs and then do a check on the corresponding outputs and verify that they behave as intended. Synthesis of hardware means transforming the HDL code into a netlist of logic and wire connections describing the hardware. This can then be compiled into an FPGA which re-wires the available internal CLBs according to the given instructions. HDLs have become more and more useful as system

¹With the exception of FPGA evaluation boards, which comes shipped with a pre-programmed hardware setup for demo purposes.

complexity have increased [?]. The hardware described in this thesis is done using VHDL, a strongly typed HDL that is known for its capability of describing parallel processes.

Below is a small example describing the logic of a D Flip-Flop using VHDL:

```

— Example of a D Flip-Flop triggered on the rising edge of the clock.
library IEEE;
use IEEE.std_logic_1164.all;

entity DFF_high is
  port(
    D : in STD_LOGIC;
    Q : out STD_LOGIC;
    Qbar : out STD_LOGIC;
    RESET : in STD_LOGIC;
    CLK : in STD_LOGIC
  );
end DFF_high;

architecture rtl of DFF_high is
begin

process (CLK, RESET)
begin
  if (RESET = '1') then
    Q <= '0';
    Qbar <= '1';
  elsif (CLK'event and CLK = '1') then — Can also use rising_edge(clk)
    Q <= D;
    Qbar <= not D;
  end if;
end process;

end rtl;

```

1.3 Cyclone V Transceiver Technology

To be able to send serial data in the gigahertz domain, a high-speed transceiver is required. The Cyclone V GT-series FPGAs supports a number of transceiver technologies through the High-Speed Mezzanine Card (HSMC) physical interface that can reach speeds up to 6.144 Gbit/s. This section gives a general description of some of these protocols.

1.3.1 Differential Signals

Common for all protocols described is the fact that the signals are treated differentially.

While a single ended signal involves one conductor between the transmitter and receiver, with the signal swinging from a given voltage to ground; differential signals involves a conduc-

tor pair with two signals that are identical, but with opposite polarity. The pair would ideally have equal path lengths in order to have zero return currents, avoiding problems like *EMI*. In addition, placing the signals as close as possible to one another will give benefits in terms of common noise rejection [?].

When done correctly, differential signals have advantages over single ended signals such as effective isolation from power systems, minimized crosstalk and noise immunity through common-mode noise rejection. It also improves S/N ratio and effectively doubles the signal level at the output ($+v - (-v) = 2v$), which makes it especially useful in low signal applications. The disadvantage comes in an increase in pin count and space required, since differential signals consists of two wires instead of one [?].

1.3.2 Low-Voltage Differential Signaling

Low-Voltage Differential Signaling (LVDS) is said to be the most commonly used differential interface. The interface offers a low power consumption with a voltage swing of 350 mV and good noise immunity. LVDS can deliver data rates up to 3.125 Gbit/s [?].

The Cyclone V GT board has 17 LVDS channels available on the HSMC port A connector. The channels have the ability to transmit and receive data at a rate up to 840 Mbit/s, with support for serialization and deserialization through internal logic. [?]

1.3.3 Current-Mode Logic

For data rates that exceeds 3.125 Gbit/s, Current-Mode Logic (CML) signaling is preferred. This is due to the fact that certain communication standards such as PCIe, SATA and HDMI, shares consistency with CML in signal amplitude and reference to V_{cc} . CML can reach a data rate in excess of 10 Gbit/s, but has a higher power consumption than LVDS, with a voltage swing of approximately 800 mV [?].

The Cyclone V GT board has 4 Pseudo-CML (PCML) channels available on both port A and B HSMC connectors. The channels have the ability to transmit and receive data at a rate up to 5.0 Gbit/s, just over the 4.8 Gbit/s range required by the GBT MGT. [?]

Chapter 2

The Gigabit Transceiver

History,
mgt,
Short
about
the elec-
tronic
compo-
nents,
gbt-sca,
gbtx

2.1 Encoding modes

The "GBT-Frame" mode, which is Reed-Solomon Based; the "8b10b" mode; and the "Wide-Bus" mode which is without encoding.

2.2 GBT-FPGA Core

The following sections describe the different components that makes up the GBT-FPGA Core. The information was obtained by reading the GBT-FPGA User Guide [?].

2.2.1 GBT Bank

The GBT Bank is defined as the top module of the GBT-FPGA Core. It integrates up to four GBT Links and contains the ports required to operate the GBT Links.

2.2.2 GBT Link

The GBT Link is the actual channel of the link. It is composed of three components: GBT Tx, GBT Rx, and the MGT. The following subsections gives a brief description of these components.

GBT Tx

The GBT Tx component is responsible for scrambling and encoding data before transmitting it through the MGT.

GBT Rx

The GBT Rx component is responsible for receiving, decoding and de-scrambling the data through the MGT.

Multi-Gigabit Transceiver

The MGT is responsible for the transmitting, receiving, serialization and de-serialization of the GBT data. It is divided into a transmitter and a receiver part.

The transmitter contains a PISO with two input clocks; one for parallel data and one for serial data. It shifts in 40 bit words from the GBT Tx with a reference clock of 120 MHz, serializes the data and sends it out with the help of a dedicated Tx Phase-Locked Loop (PLL) that generates a serial clock of 2400MHz.

The receiver contains a Clock & Data Recovery (CDR) block, a SIPO, a RXRECCLK Phase Aligner block and a Barrel-shifter.