PROCEEDINGS OF SCIENCE

Firmware development and testing of the ATLAS IBL Back-Of-Crate card

Marius Wensing[*a] **on behalf of the ATLAS Collaboration**

[a] *University of Wuppertal*
Gaußstr. 20, 42119 Wuppertal, Germany
E-Mail: wensing@uni-wuppertal.de

For the new innermost layer of the ATLAS Pixel-Detector at CERN new off-detector hardware needs to be developed. The Back-Of-Crate card (BOC) is driving the optical interface to the detector and distributing the LHC clock to all detector components. A brief overview of the firmware and test results from production and system test will be presented.

Technology and Instrumentation in Particle Physics 2014
2-6 June, 2014
Amsterdam, the Netherlands

[*] Speaker.

PoS(TIPP2014)216

Contents

1. Introduction

ATLAS is one of the four big experiments at the Large Hadron Collider (LHC) at CERN [1]. The innermost part of ATLAS is the pixel detector, which provides measurements of particle position close to the interaction point, thus allowing the reconstruction of primary and secondary vertices. The pixel detector provides around 80 million pixels distributed in 3 layers and 3 disks per side.

Currently (2013/14) it is being upgraded with a new innermost 4th layer, the Insertable B-Layer (IBL) [2]. This upgrade will fulfill the requirements due to the higher than expected luminosity and also enhance the tracking efficiency. The IBL was installed together with a new beam-pipe in 2014 and is equipped with 448 newly developed front-end chips (FE-I4) [3], which will add 12 million pixel cells to the pixel detector. Each front-end chip will be read out at a data rate of 160 Mbps, which cannot be handled by the current readout system. So IBL will require also a redesign of the readout system. This new readout system will be described in the following section.

2. IBL readout system

The IBL readout system, of which a schematic is shown in Figure 1, is based on the readout of the current pixel detector. Each of the 14 IBL staves is connected via an electrical-optical converter (2 Optoboards per stave) to a card pair consisting of the Read-Out-Driver (ROD) and the Back-Of-Crate card (BOC). The ROD is steering the data taking and calibration of the detector, whereas the BOC card provides the interfaces to and from the detector and the higher level readout. Also the BOC card is distributing the LHC clock coming from a Timing- Trigger, Control Interface Module (TIM) to the detector and the ROD. A VME-crate contains 15 ROD/BOC card pairs (14 IBL, 1 Diamond Beam Monitor) as well as a TIM card and a VME-Single Board Computer (SBC).

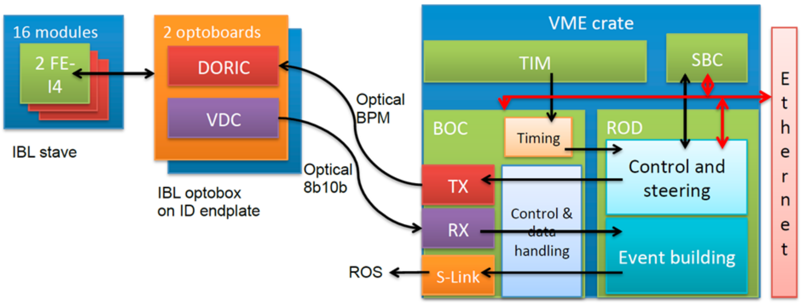


Figure 1: Overview of the IBL readout system

3. IBL BOC card

The new IBL BOC card shown in Figure 2 provides the optical interfaces to the ROD. It is based on the previous Pixel BOC card, but is now equipped with modern FPGAs for signal processing on the card. In total three Xilinx Spartan-6 FPGAs [4] are mounted on the card.

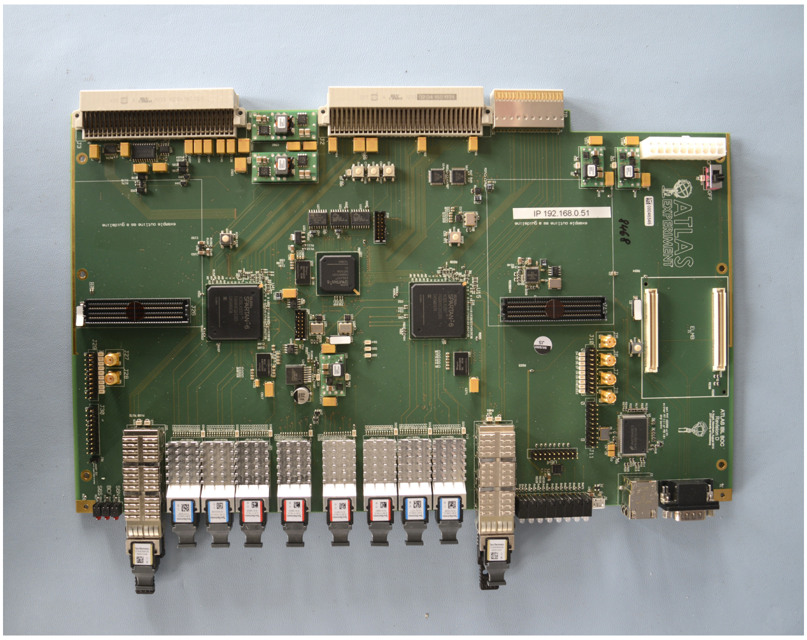


Figure 2: Photo of the BOC card equipped with commercial optics

Figure 3 shows the block diagram of the all components on the card. The BOC Control FPGA (BCF) is handling all the slow control interfaces coming from the ROD as well as the Gigabit Ethernet connection. The two independent BOC Main FPGAs (BMF) are responsible for the signal processing and monitoring of incoming data. On the right hand side the optical components are shown. The detector interface is using commercial SNAP12 plugins [5] for transmitting and receiving data. Each card can handle up to 48 optical channels per direction. In normal IBL operation only 2 out of 4 TX-plugins are used, but it is foreseen to use the new IBL BOC card also for upgrading the current pixel detector readout, which needs the additional TX-plugins. The connection to the higher level readout uses QSFP transceiver modules. Four of these links are dedicated to the

ATLAS Fast Tracker (FTK) [6], the other four links are going to the ATLAS ReadOut Subsystem (ROS) [7].

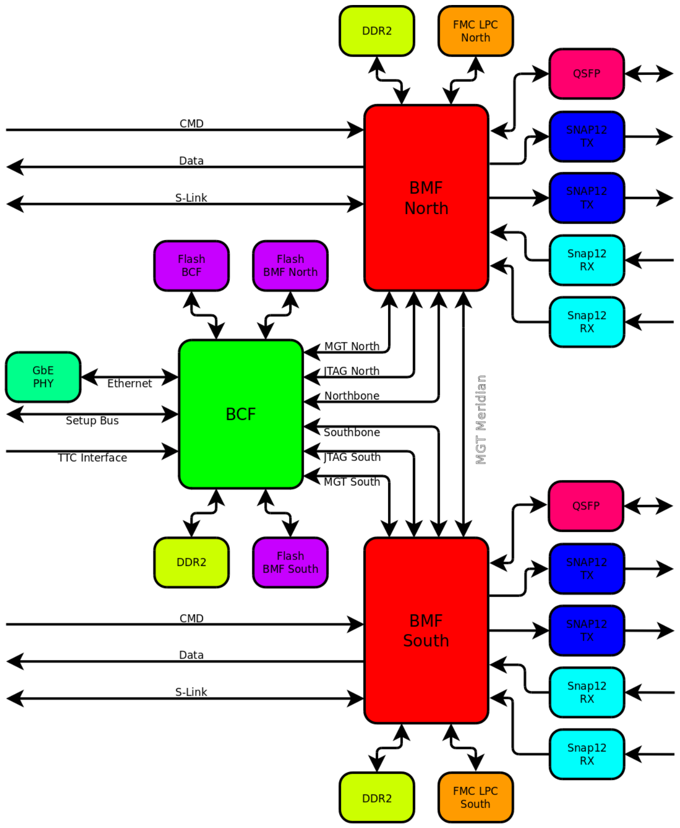


Figure 3: Overview of the BOC card components

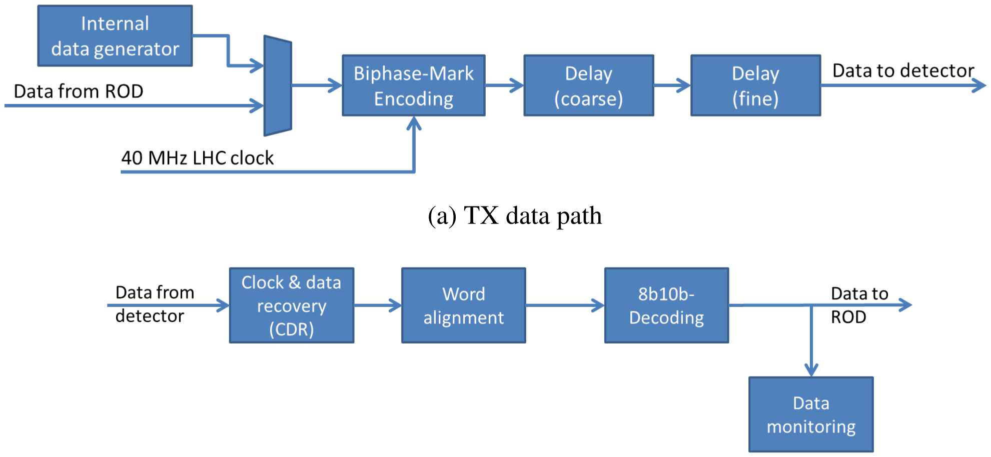
4. Firmware development and testing

4.1 General overview

This section will provide an overview of the firmware. A detailed view can be found in [8]. In general the firmware is divided into two blocks, the BCF firmware and the BMF firmware. The control part of the firmware resides inside the BCF firmware whereas the signal processing belongs to the BMF firmware.

Central part of the BCF firmware is a Wishbone interconnect, which is steering the central control path on the card. It is connected to the backplane for ROD access of registers, as well as to a Microblaze processor, which is responsible for the Ethernet communication. Connected to this interconnect are registers on the BCF mainly steering the programming of the card and the clocking circuit, as well as some general purpose memory and the two BMF register spaces.

The firmware of the BMFs is identical for both FPGAs and consists of two main parts. The TX path, which is shown in Figure 4a is responsible for mixing the serial data coming from the ROD together with the 40 MHz LHC clock. A Biphase-Mark-Encoding (BPM) is used for distributing the clock and the data to the modules. Additionally, a delay for adjusting the timing of the detector



(a) TX data path

(b) RX data path

Figure 4: Block schematic of the signal processing path

with respect to the bunchcrossing is applied to the signal. It can be adjusted on a coarse basis with 6.25 ns per setting. The fine delay adjustment is possible in steps of 100 ps. A detailed explanation of the fine delay block is given in section 4.2. For debugging purpose each of the transmitters can generate a serial data stream internally for testing the card without a ROD connected to it.

The other important part is the RX path shown in Figure 4b. First the incoming serial data is being phase aligned to the sampling phase to ensure correct decoding of the data. The word alignment module then searches for word boundaries in the 8b10b encoding. The encoding therefore defines comma words, which are unique in the datastream. The aligned data is fed into the 8b10b-decoding block and forwarded to the ROD. Data monitoring functions are available to measure the signal quality. This includes monitoring of decoding and disparity errors, as well as frame errors. Also data regarding the module health status is being monitored and forwarded to the detector control system (DCS).

4.2 Fine Delay implementation

The fine delay is important for adjusting the timing of the detector. It should give about 100 ps of delay and the duty-cycle of the signal should not exceed $(50 \pm 2)\%$, to ensure proper operation of the Optoboard. Different approaches were taken into account for the implementation of this fine delay module:

- External delay chips
- Propagation delay in the FPGA
- IODELAY2 primitives in the FPGA

External delay chips would require a lot of space on the PCB, so the using internal FPGA resources was tested first. Especially the IODELAY2 primitive has interesting features, as it was designed for timing adjustment of signals, e.g. for timing of memory interfaces. Three configuration modes are available: Fixed Output Delay, Fixed Input Delay and Variable Input Delay.

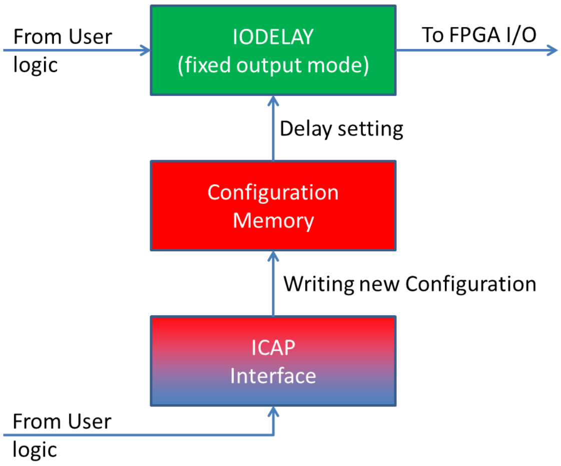


Figure 5: Partial Reconfiguration approach for setting the fine delay

The aim is now to get a variable output delay, but this mode is not directly supported by the primitive. So a first approach using the variable input delay of an unused I/O pin and routing this signal through the FPGA to the output (see Figure 6) was tested. In [8] this approach showed a good behaviour in terms of delay but a very bad behaviour in terms of duty-cycle which exceeded the limits. So the new approach of changing the IODELAY2 by using partial reconfiguration was evaluated.

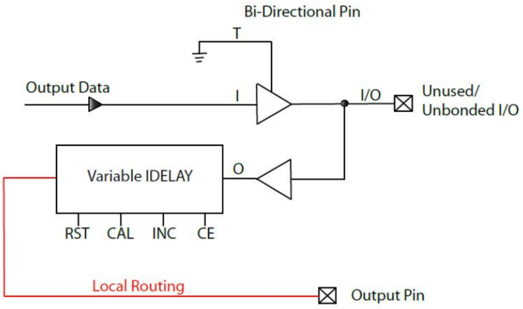
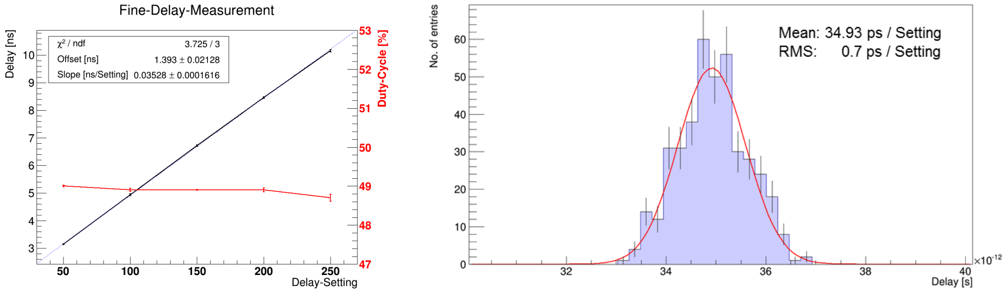


Figure 6: IODELAY2 workaround for a variable output delay by Xilinx (AR34276, [9])

Figure 5 shows the method how the delay is being updated during operation. During operation of the card the internal configuration memory can be accessed through an Internal Configuration Access Port (ICAP). This allows writing of configuration bits. By writing the configuration bits which affect the delay setting in the IODELAY2-primitive the delay can be changed. As the mapping of the configuration bits is part of the Intellectual Property the meaning of each configuration bit is unknown. This open question leads to the software part of this approach. After synthesis and place & route, a netlist containing the FPGA configuration is written. From this netlist the bitfile is being generated. Xilinx provides a tool called "FPGA-Editor" which can change the configuration inside the netlist. By changing only the delay value of the output delay a differential bitfile with only changed configuration frames can be generated. This bitfile will be loaded through the ICAP interface into the FPGA configuration memory. The output delay will then immediatly change.

During the production of the IBL cards each channel for each board has been measured against the 40 MHz board reference clock using an oscilloscope. Figure 7 shows the result of the measurements. In the left plot a single channel measurement is shown. The delay behaves linear regarding the setting of the output delay and the duty cycle is only minimal affected. Over all cards and channels the slope of the delay is (34.9 ± 0.7) ps per setting.



(a) Single channel measurement

(b) Histogram of all measurements

Figure 7: Measurement results for the fine delay block

4.3 FE-I4 emulator

To test the full readout chain front-end chip generated data is needed. This can be achieved by connecting a front-end chip electrically or optically to the BOC card. It was decided to implement a minimalistic emulator, to be able to test the full readout chain if no front-end is availble. This can also be used during ATLAS milestone runs when the detector is not operational. The emulator provides some basic features like:

- Reading/writing global registers
- Switch between run-mode and conf-mode
- Manual/automatic hit generation on trigger

Using the emulator is not different from using a real front-end chip during data taking. After the reception of a trigger command the front-end emulator will send out a configurable number of hits. Figure 8 shows the occupancy histogram of a single emulated front-end, which was triggered 10000 times with 200 hits per trigger. The distribution of the hits is good over the full chip. Only some pixels have a really high occupancy, which can be traced down to the random-number generator used for generating the random hits. By using the manual hit injection mode the user can also specify a certain hit pattern, which is useful for calibration scans.

4.4 Production test of the card

This section will briefly describe the way each card is being tested after reception from the manufacturer. The first step is a full optical inspection of the card, which ensures, that all components are mounted correctly and no soldering problems, e.g. bridges or flux remainings, are there. When the optical inspection is OK, a short electrical test is being performed. The aim is to

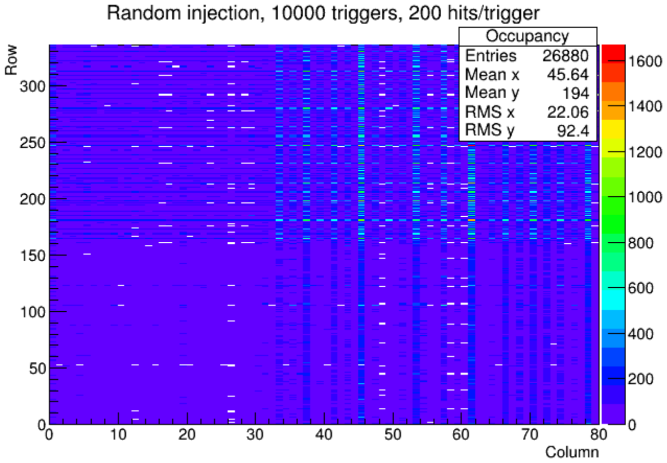


Figure 8: Occupancy histogram for 10000 triggers with 200 hits/trigger

identify shorts in the supply voltages and check the power consumption of each board in unprogrammed state. Afterwards the BCF flash memory will be programmed with the firmware image. The firmware image includes the FPGA configuration as well as the Microblaze software which is needed for Ethernet communication. If the boot process of the BCF has finished and the ethernet communication is ok, the BMF firmware image will be programmed into the BMF flash memories. If all FPGAs have been booted, the current consumption with and without optical plugins will be measured.

When the first electrical tests are finished and the communication to the card has been established a number of automatic tests will be performed. These test will check the basic communication functionality as well as the signal integrity on the optical lines. Using an oscilloscope with an optical probe some signal parameters are being verified:

- frequency
- duty-cycle
- rise/fall time
- signal power
- eye-diagram
- coarse and fine delay calibration

The production test finishes with tests of the VME backplane. The BOC to ROD communication is tested with signal generators and signal checking circuits inside the FPGAs of BOC and ROD. Adding a ROS-receiver to the setup will also check the optical connection to the higher-level readout over the QSFP plugins. If all the tests are passed the card can be released to CERN for being installed in the counting rooms.

5. Conclusion

As a conclusion the firmware development for the ATLAS IBL BOC card is nearly finished. All tests which have been performed during the production of the cards show a very good performance. The implementation of the fine delay using IODELAY2 primitives and reconfiguring them with the method of partial reconfiguration has been used successfully and will be used during operation of the card. The emulator inside the firmware will also make the development of firmware and software in the readout system much easier.

Acknowledgements

We like to thank the German Federal Ministry of Education and Research (BMBF) for funding the IBL BOC project as well as the INFN for their support.

References

[1] The ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, 2008 JINST **3** S08003

[2] The ATLAS Collaboration, *ATLAS Insertable B-Layer Technical Design Report*, CERN-LHCC-2010-013

[3] M. Garcia-Sciveres, et al., *The FE-I4 pixel readout integrated circuit*, Nucl. Instr. Meth **A 636** (2011), p. 155-159

[4] Xilinx Inc., *Spartan-6 FPGA Familiy*, http://www.xilinx.com/products/silicon-devices/fpga/spartan-6/index.htm

[5] *SNAP12 Multi-Source-Agreement*

[6] The ATLAS Collaboration, *Fast TracKer (FTK) Technical Design Report*, CERN-LHCC-2013-007

[7] J. Vermeulen, et al., *ATLAS DataFlow: the Read-Out Subsystem, Results from Trigger and Data-Acquisition System Testbed Studies and from Modeling*, Real Time Conference, 2005. 14th IEEE-NPSS

[8] M. Wensing, et al., *Testing and firmware development for the ATLAS IBL BOC prototype*, 2012 JINST **7** C12027

[9] Xilinx Inc., *AR # 34276*, http://www.xilinx.com/support/answers/34276.html