Design document

IBL BOC Prototype Design
Overview

Prepared by
A. Kugel, N. Schroer, Heidelberg University
J. Dopke, T. Flick, T. Heim, M. Neumann, Wuppertal University

February 2011
Version 01

Version management

No	Date	Changes	Author
0.1	2011-02-03	Initial version	A. Kugel
0.2	2011-02-06	First draft	A. Kugel
0.3	2011-02-06	Updates from Jens	A. Kugel
0.4	2011-02-07	DDR3 on BCF changed to DDR2 (availability)	A. Kugel
0.5	2011-02-09	Author list updated	A. Kugel
0.6	2011-02-11	Foto of real SNAP-12	A. Kugel
0.7	2011-02-17	More on SFP options and DCS	A. Kugel

Contents

1. Introduction

The upgrade of the ATLAS pixel detector with the IBL (insertable B-layer) is scheduled for 2013. The on-detector readout is using the new FE-I4 readout ASIC, which needs to be complemented by a novel off-detector readout system. To maintain a maximum level of compatibility with the existing off-detector readout [26] the new system architecture will again be based on a pair of VME cards – the BOC handling the detector and TDAQ interfaces in the rear of the VME crate and the ROD handling control and processing in the front. The overall system and the requirements are described in [1]. The functional

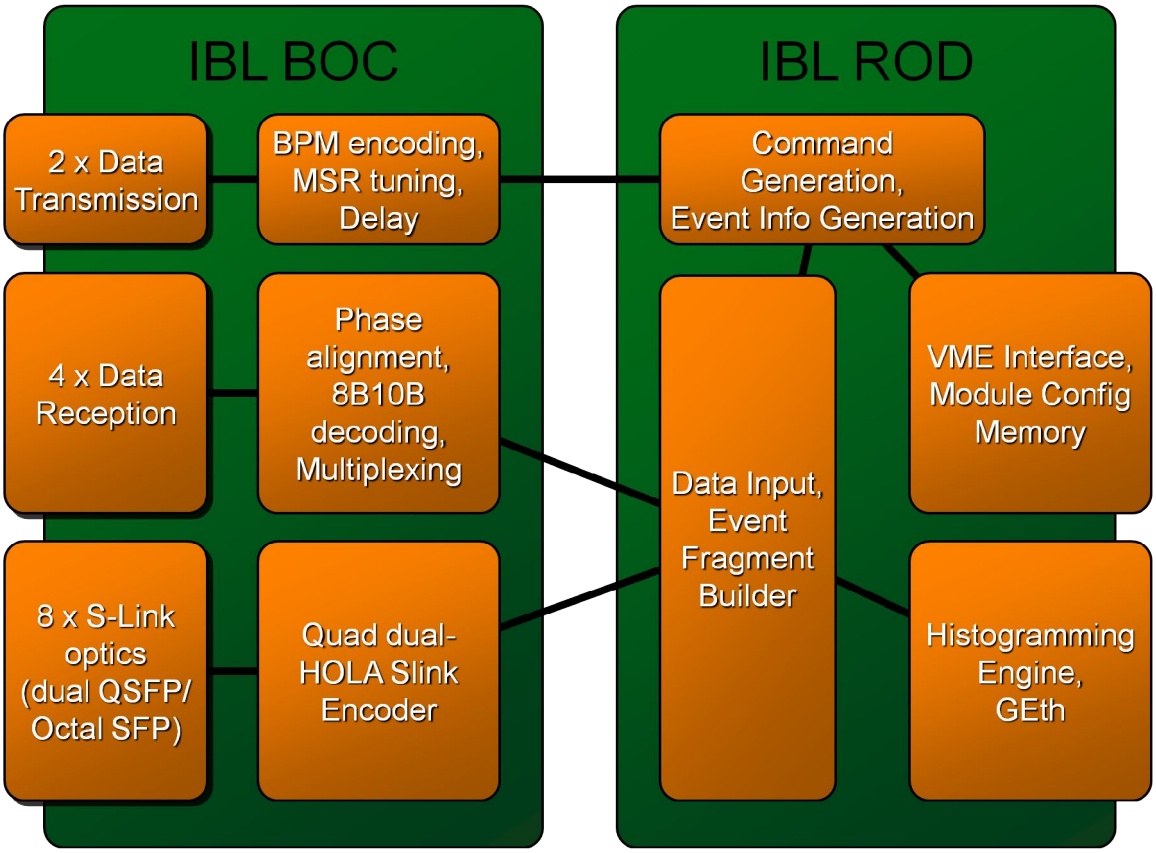

Figure 1: Card pair

decomposition on the card pair is depicted in Figure 1. As far as the BOC is concerned, the three blocks “BPM encoding ...”, “Phase alignment ...” and “dual-HOLA ...” are related to corresponding VHDL firmware modules, while “... Transmission”, “... Reception” and “... Optics” correspond to optical-electrical modules.

1.1. Interface types

The BOC is responsible to handle the control interface to the detector and the data interface from the detector. Both interfaces run via optical links, however of different speed and protocol. Also, data input operates per FE-I4 chip while data output runs per module (2 * FE-I4). Hence the number of output lines is only half the number of input lines. On the existing BOC, custom-made RX and TX modules are used. The TX modules provide individual bi-phase mark (BPM) encoding of the signals together with a tuning feature for the mark-space-ratio (MSR tuning) and fine-tuning of the signal delay. The

optical output power can be adjusted up to around 1mW in order to compensate the sensitivity degradation of the on-detector receiving pin diodes. The same transmission mechanism will be used on the new BOC. However, we plan to perform all encoding and timing adjustments inside an FPGA. Hence the optical transmitters need only to perform the electrical to optical conversion, which can in principle be done with commercial components [2]. However it must be guaranteed that the output power is sufficiently high.

The requirements on the RX path are quite different between the old and new readout, the former running at 40MHz with an unbalanced signal and the latter running at 160MHz with a balanced 8B10B encoded signal. Commercial components [2] are likely to be suitable in this case and will be investigated.

The BOC also handles the interface to the TTC timing system, which provides the 40MHz experiment clock and some other information. The TTC clock is distributed across the private part of the VME backplane as a differential PECL signal to every BOC card. It is the primary reference clock for the BOC and the ROD, the latter getting its copy via the BOC.

The BOC communicates with the ROD across the VME connectors. This communication comprises four different sections (see also Figure 2):

- Board control and configuration via the “setup-bus” interface, a simple extension to the ROD's VME interface.
- Data towards the detector (CMD out)
- Data from the detector (RX decode)
- Data towards the TDAQ subsystem (S-Link)

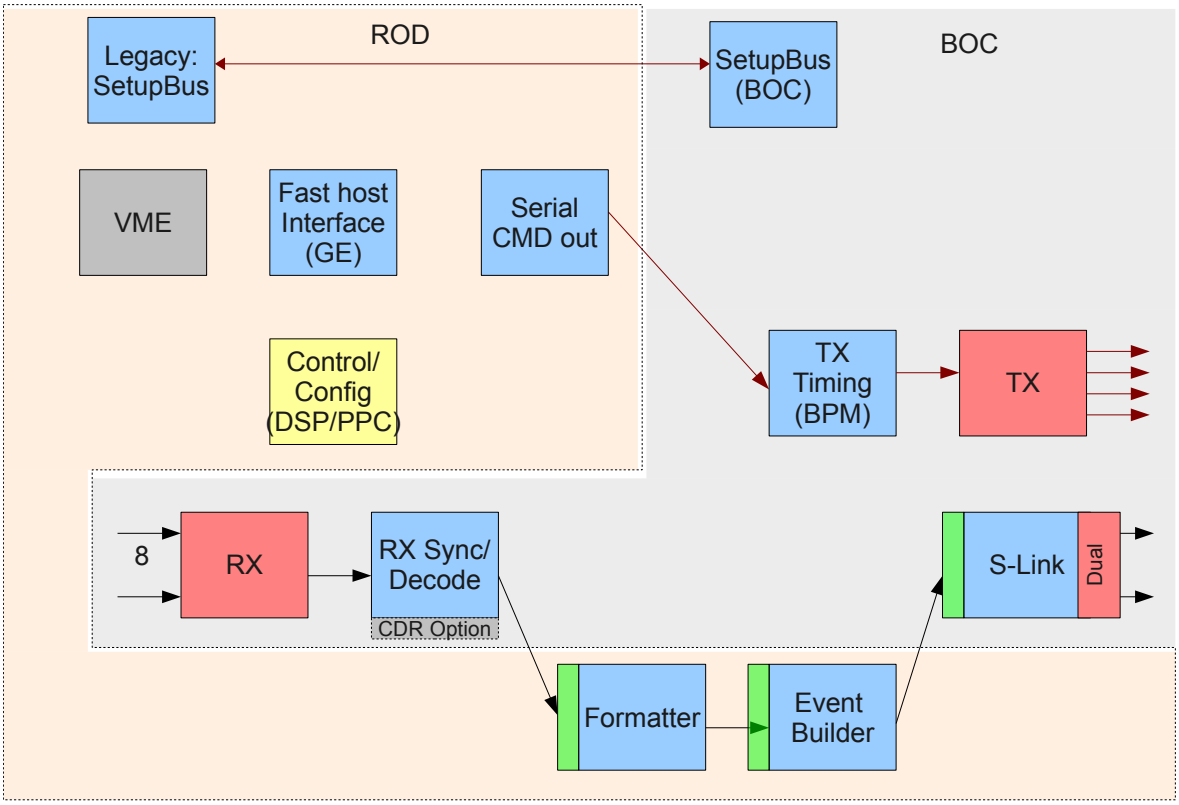

Figure 2: Functional blocks

The connectivity and the electrical properties of the signals involved on these 4 communication paths must be implemented in a way that interoperability between a new and an old card on either side can be established. Of course a special firmware on the new card (ROD or BOC) will be needed in this case.

Finally, the data processed by the ROD must be sent to the TDAQ subsystem via the standard ATLAS read-out links (ROLs). These ROLs use a 2.0 Gbit/s bi-directional optical interface, driven by the CERN HOLA protocol [3], which only provides a uni-directional user data interface operating at 160MB/s plus flow-control according to the S-Link specification[1] [4]. The HOLA S-Link on the BOC will be implemented directly[2] in the FPGAs and will support twin-channel operation for FTK compatibility. The use of quad opto-transceivers (QSFPs [5]) will be evaluated, as this could simplify routing of the fibers in the cabinet. One of the Spartan-6 devices will connect to 4 individual SFP transceiver and the other to a QSFP transceiver.

[1] The S-Link spec states a nominal bandwidth of 160MB/s, however user data-rates of up to 200MB/s can be achieved over the 2Gbit/s link.

[2] The S-Link mezzanine implementation has been encouraged during the development phase of the experiment. With stable operation there is no need to continue this style and an embedded implementation of the HOLA S-Link can be used on the new readout system. The corresponding end – the ROBINs within the TDAQ system – already use an embedded implementation, therefore there is no risk in choosing this approach.

2. Hardware implementation

The BOC hardware implementation takes advantage from recent FPGA technology and from the advancements provided by the FE-I4 chip compared to FE-I3. The majority (if not all) of the timing adjustment logic used on the existing BOC can be either integrated into the FPGAs or is no longer required. Likewise the multitude of level-shifter circuitry can be omitted. Serially terminated FPGA I/O buffers (SSTL3-I) enable to run the BOC-ROD communication at higher speeds[3] but can still be used for LVTTL signalling[4] when operating an old card together with a new one.

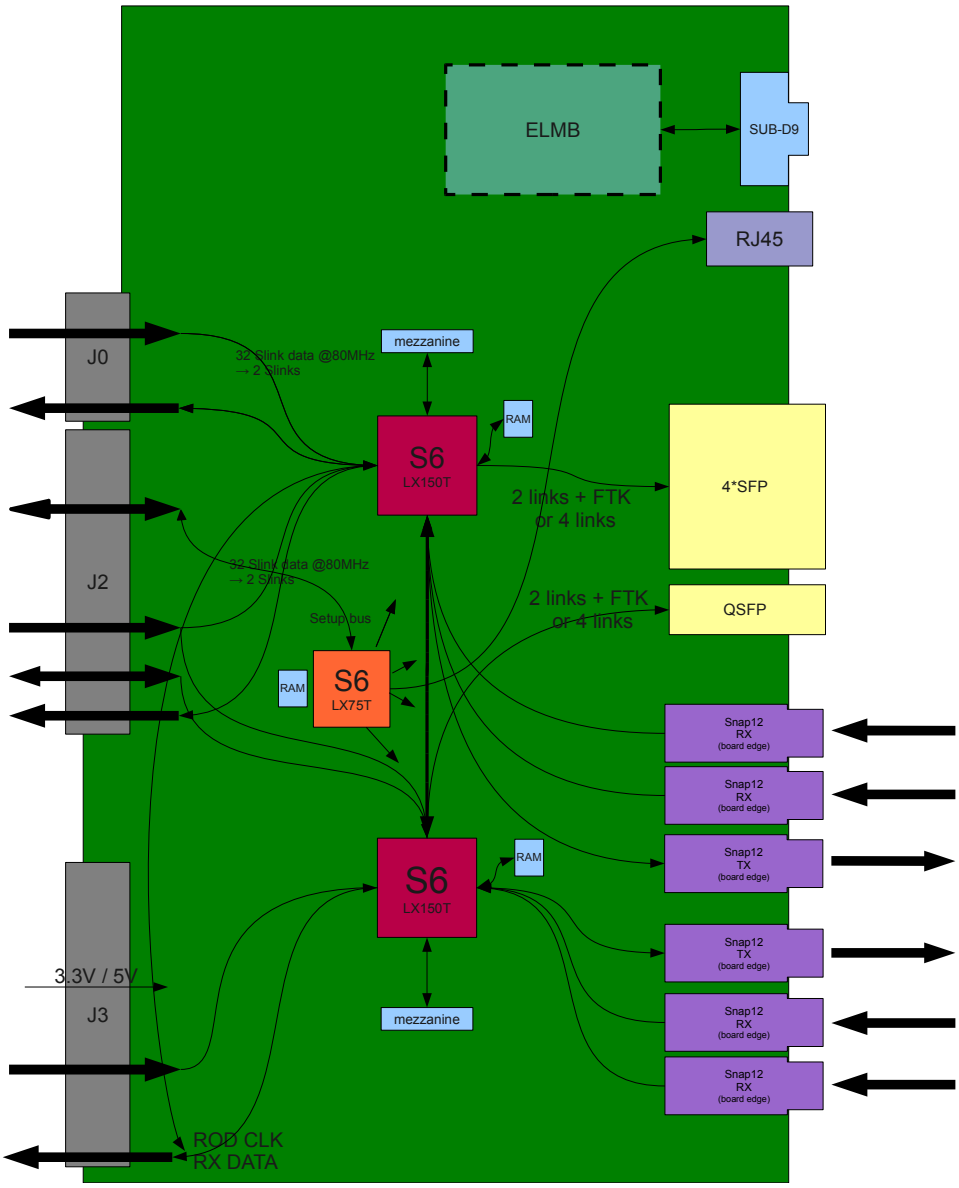

Figure 3: BOC sketch

The major components on the new BOC are shown in Figure 3. The two Xilinx Spartan-6LX150T FPGA implement two functionally identical slices. Each slice attaches to one

[3] The signal layout will consider 80Mbit/s, but higher speeds will be investigated as well.

[4] With different firmware the I/O buffers can be configured for different I/O standards, which just require a compatible supply voltage (here: 3.3V).

TX interface, two RX interfaces, two dual-HOLA S-Links and communicates with the ROD. In addition, each FPGA has a single DDR-2 memory device attached and a low pin-count FMC connector. The FMC connectors can be used to carry mezzanines with test circuitry during prototype or volume production factory tests. In case neither the SNAP-12 transceivers nor the old-style TX plug-ins provide adequate functionality, the FMC mezzanines can carry custom optical modules. The DDR-2 memory can be used to store data from local operational monitoring.

Auxiliary functions, primarily the setup-bus and the FPGA configuration logic, are integrated in an additional board-controller FPGA (BCF) of type Spartan-6LX75T.

2.1. Board-controller FPGA

The primary board-control functions of the BOC are related to the setup-bus and power-up configuration (see following chapters). Only little additional logic is required for the mandatory functions and all could well be implemented in a CPLD. However, with the availability of complex functions at relatively low cost provided by the Spartan-6 FPGA family a different approach is taken on the new BOC: a smaller FPGA (XC6SLX75T-3FGG484C[5]) accompanies the two main FPGAs and acts as intelligent board-controller (Figure 4).

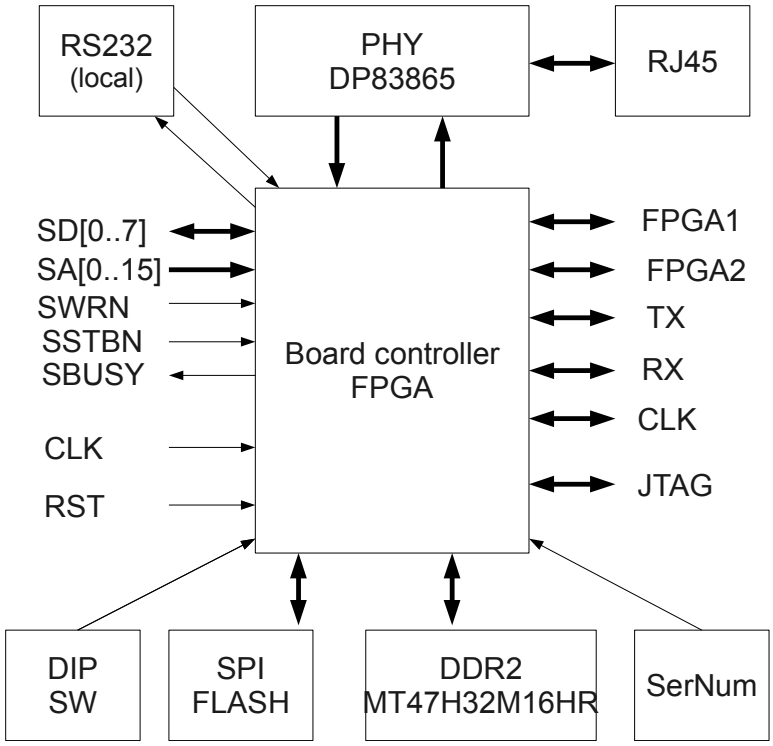

Figure 4: Board controller FPGA

The setup-bus logic is implemented in the usual way (VHDL code for the FPGA fabric) and provides the functionality known from the existing BOC:

- Control- and status information are provided via a few dedicated lines between the BCF and the main FPGAs.
- Control and status lines of the optical modules are connected to the BCF.

[5] Even smaller devices (LX45T, LX25T) are available in the same package

- SRAM, 1kB (FPGA internal)

The logical interface towards the DAQ software will be as similar to the existing card as possible, given the constraints of the hardware differences.

The BCF loads it's power-up configuration from an attached serial FLASH memory. Subsequently, the controller firmware downloads the configuration bitstreams to the main FPGAs in order to complete power-up configuration of the BOC.

Besides these legacy functions a Microblaze soft-processor running a standard Linux[6] kernel will be implemented in the controller FPGA, which enables a number of advanced features like:

- remote access via an ethernet port to the BOC during tests and stand-alone operation
- emulation of the crate-controller and setup-bus in stand-alone mode
- firmware upgrades via Ethernet
- software-controlled board configuration
- extended operational monitoring.

Only little additional hardware is required to implement such a minimal Linux system: next to the FPGA a single DDR2 memory chip [15], an Ethernet PHY [16] plus RJ45 connector, a serial port driver and a serial FLASH storing the OS. A silicon serial-number device [13] provides board-identification and is used to derive the Ethernet MAC address from.

2.2. Clock and timing

On the existing BOC card a complex timing system had to be used with the following basic functions:

- System clock generation for BOC and ROD derived from TTC clock (via backplane).
- Individual adjustment of delay and mark-space-ration on the CMD path. Implemented via programmable TX-modules.
- Individual adjustment of sampling phase and delay on the RX path. Implemented via programmable delay lines (PHOS4).

[6] Microblaze Linux is available at ZITI. Expected FPGA resource utilisation is in the order of 10%.

A further complication came from the fact, that the different components were operating at different voltage levels – 5V-PECL, 3V-PECL, LVDS and LVCMOS – requiring a large number of level-shifting circuitry. An overview of the old clock system is shown in Figure 5.

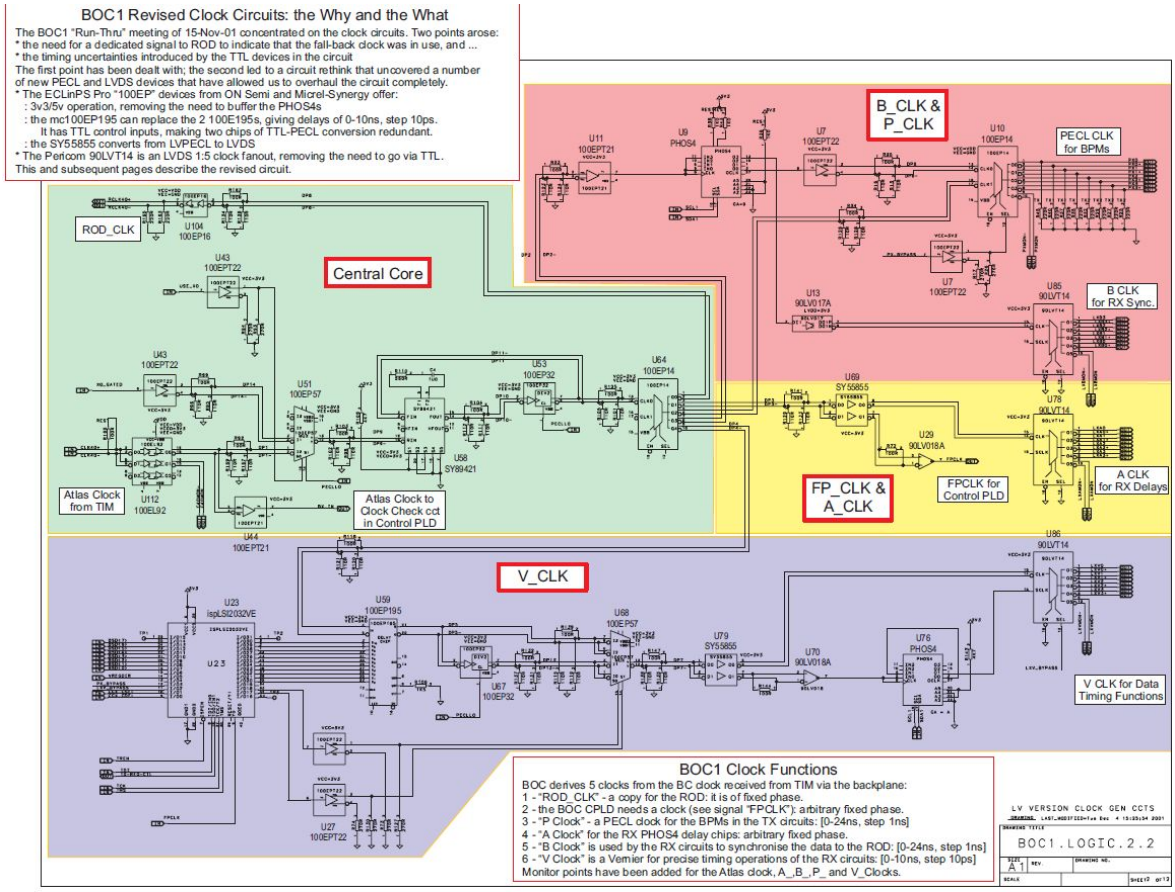

Figure 5: Old BOC clock systm

With the simplified optical interfaces and the advanced timing features of the modern Spartan-6 FPGAs the clock system of the new BOC can be reduced to something like the "central core", see Figure 6. Level translation is required when receiving (5V-PECL into 3V-PECL, MC100LVEL92 [10]) and driving (3V-PECL to 5V-PECL, MC100EP16 [9]) the backplane clock signals. A programmable delay (2 cascaded MC100EP195B [11]) is required in order to synchronise the phases of the BOC and ROD clocks with the ATLAS TTC clock. A programmable delay with a range of 20ns together with the FPGA option to invert the clock provides sufficient margin to adjust the phases. A jitter-cleaner TI CDCE62002 [12] after the delay line provides two clean clock outputs, which are fed to the local BOC resources and towards the ROD with low skew. The jitter-cleaner also provides fall-back to a local oscillator in case the TTC clock is not available. Input clock selection can be done automatically or manually, controlled by the BCF.

The individual delay adjustment of the CMD output signals is discussed in chapter 4.1.

In addition, there are a number of individual clocks required, not related to the ATLAS timing system:

- 25 MHz crystal for Ethernet PHY

- 100/125MHz selectable MGT reference clock. This enables to operate the transceivers at 1.25GBit/s, 2.0Gbit/s (HOLA default), 2.5GBit/s and 3.125GBit/s
- 100MHz FPGA design reference clock

Low-jitter clock oscillators [14] and buffers will be used for this purpose.

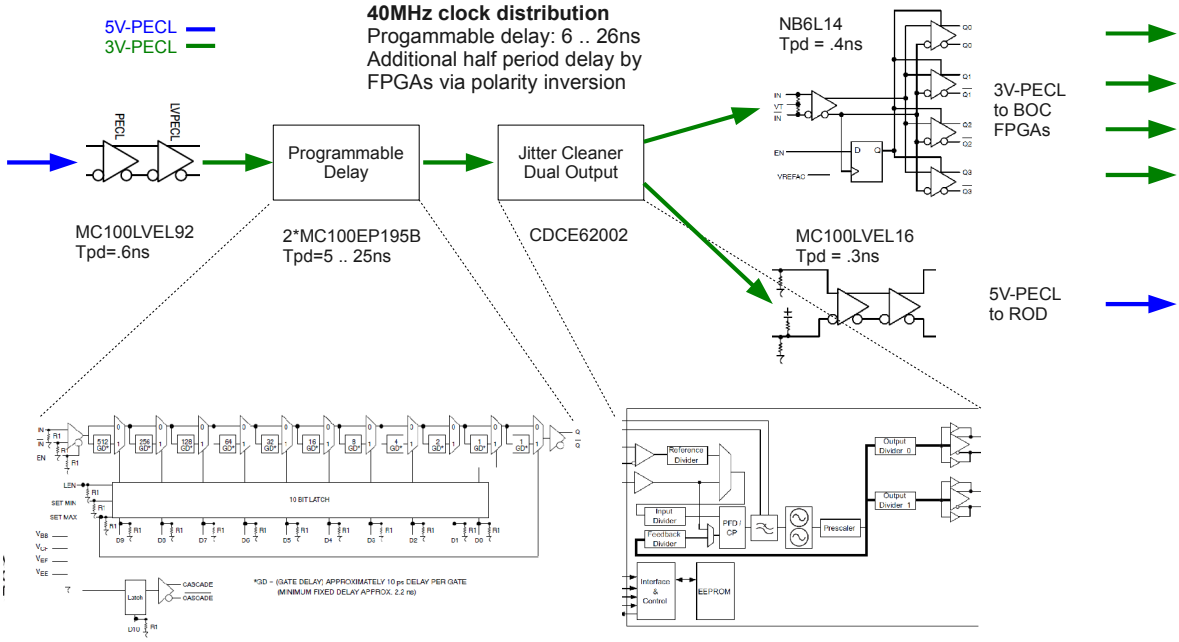

Figure 6: New BOC clock system

2.3. ROD-BOC communication

As stated above, ROD-BOC communication is comprised of four different groups[7]: CMD, RX, S-Link and setup-bus. The first three groups are made of high-speed signals, while the setup-bus is relatively slow[8].

For backward-compatibility all signals existing on the VME connectors J2 and J3 must be connected appropriately anyway. Therefore, the locations of clock and setup-bus related lines are fixed. However all “high-speed” signals can be used in a flexible way, as we don't use bus buffers but connect the FPGAs directly to the VME connectors. The J0 connector will be used as well, which doesn't exist on the old cards.

The total number of backplane signals available for the four signal groups is 250. This doesn't satisfy the connectivity requirements if all signals run at the native data-rate of 40Mbit/s. Therefore some signals must be multiplexed onto fewer lines running at a higher data-rate (80Mbit/s). As a result, the four groups require a total of 229 lines out of 250. This already includes the extension of the set-bus to 16 bit addresses. The following signal allocation will be used:

[7] There are some static signals as well, like “BOC-OK”, which have reserved locations and are not considered in the calculation

[8] Currently Wuppertal is looking at improving the speed of the setup-bus in order to reduce the VME busy period.

Group	Channels	Lines/channel	Composition	Speed [MHz]
CMD	2	8	Individual lines	40
RX	8	12	8 data, 2 addr, 2 ctl	80
S-Link	4	23	16 data, 4 ctl, 2 status, 1 spare	80

The signals of these 3 groups will use SSTL3-I I/O-standard supported by the Spartan-6 FPGAs. The SSTL3-I I/O buffers are shown in Figure 8. All terminating resistors are realised via the Spartan-6 on-chip termination features, which simplifies board layout and allows to re-configure the FPGAs pins for different I/O-standards like LVTTL or LVCMOS. The reference voltage of 1.5V is derived from a 3.0V precision voltage reference (e.g. Burr Brown REF3330) on both ROD and BOC.

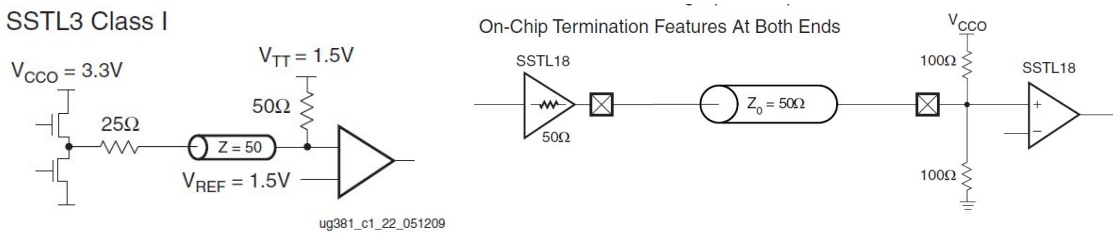

Figure 8: SSTL3 buffers (principle: left, Spartan-6 on chip termination example: right)

Trace routing of all SSTL3 signals must be done with matching lengths on a per-channel basis. Due to the relatively low speed a match of +/- 1ns is sufficient. The fine-tuning will be done with the help of calibration of the Spartan-6 input delays (IDELAY). For this, all SSTL3 output channels will contain logic to provide a known data pattern which is used to calibrate the corresponding inputs. IDELAY calibration is required after power-up only.

The setup-bus with it's lower speed requirements will use 3.3V LVTTL/LVCMOS signalling levels.

2.4. Power-up configuration

The FPGAs are SRAM based devices and require download of a configuration bitstream after power-up. A variety of options is available to accomplish this, as described in the Xilinx Spartan-6 configuration user guide [17]. The lowest-level approach for FPGA configuration is to use a JTAG programmer attached to a JTAG connector. Depending on the wiring of the on-board JTAG signals, either a single device can be accessed or several devices arranged in a JTAG "chain". On the BOC, we use a two JTAG levels. The JTAG configuration lines of the BCF are connected to a dedicated JTAG connector. A second JTAG connector is attached to user I/O pins of the BCF, likewise the JTAG ports of the two main FPGAs. Thus, the BCF firmware can create a second JTAG chain, consisting of one or two FPGAs attached to the second connector. This is the typical setup for lab tests. The Xilinx JTAG programmer software will also detect and program the SPI FLASH attached to the BCF SPI configuration port (see below).

Using a JTAG programmer is not suitable for normal operation. Instead, a two-step start-up sequence is used here: firstly, the BCF loads it's configuration in "Master Serial Peripheral Interface" mode from a 64Mbit SPI FLASH[9]. Subsequently the BCF firmware

[9] Xilinx programmer supports specific devices only, here Numyonix M25P64-VMF6P

reads the configuration data for the two main FPGAs from a second[10] SPI FLASH and downloads it via the *Slave Serial* configuration ports. Depending on the configuration the BCF will finally load software from a third[11] SPI FLASH.

The BCF firmware makes all configuration ports and the SPI FLASH memories available for access via the setup-bus and potentially for access from the embedded Microblaze processor, enabling firmware updates via VME or Ethernet. A sketch of the configuration logic is shown in Figure 9.

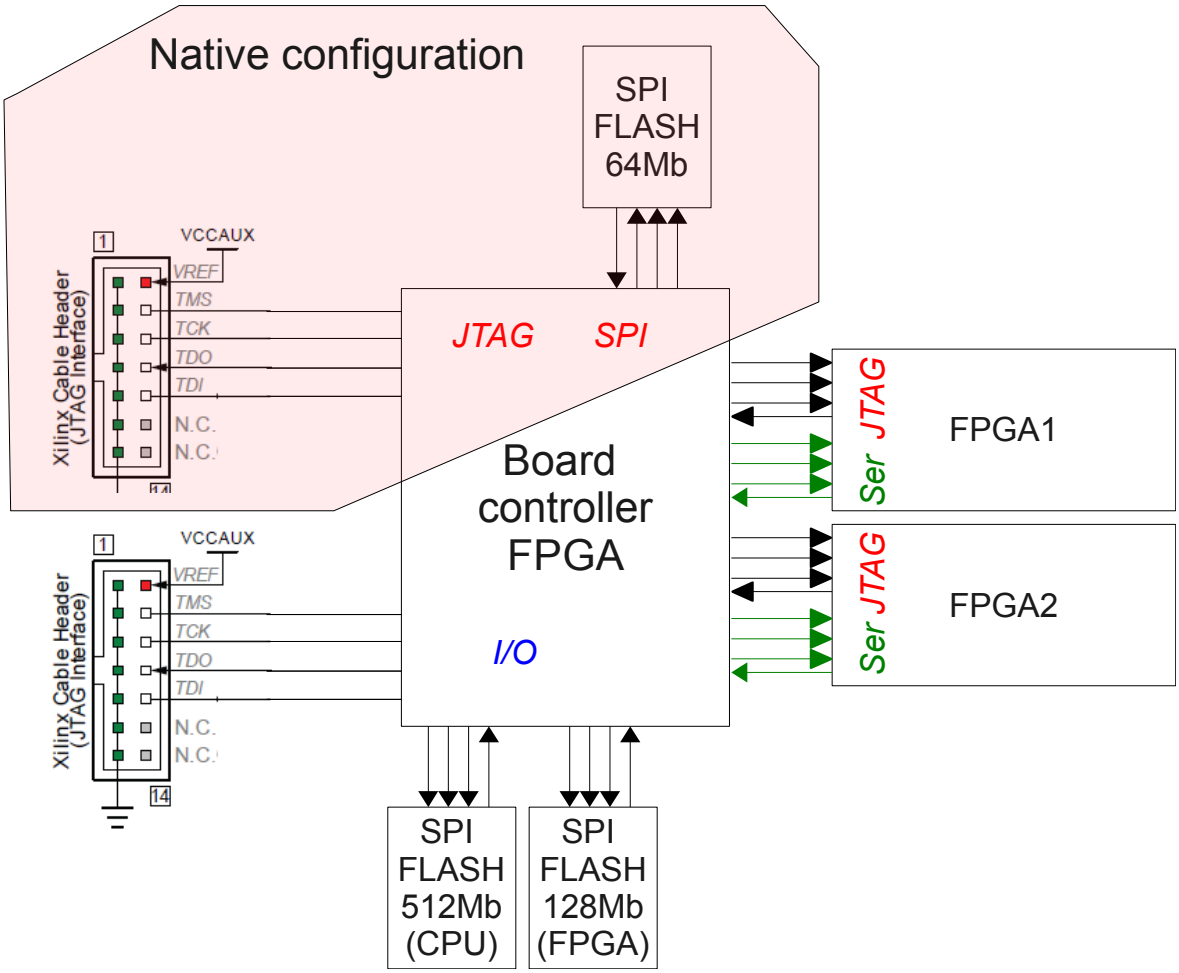

Figure 9: Configuration interfaces

[10] Bitstream sizes are 19,624,736 bit for XC6SLX75T and 33,761,696 bit for XC6SLX150T. Any standard, fast 128Mbit SPI FLASH (e.g. S25FL128P0XMFI001) is sufficient for the 2 main FPGAs.

[11] A basic Linux OS requires around 6MB. A FLASH size of 128Mbit would be sufficient, however 512Mbit will provide more flexibility, e.g. SST25VF512A

3. S-Link

Within ATLAS, the HOLA S-Link implementation is the standard readout link between detectors and the TDAQ system. At present, all detectors use the mezzanine version of the link source card (LSC), while the TDAQ ROBIN cards [6] use an embedded version of the link destination card (LDC), thus fixing HOLA as de-facto ROL implementation

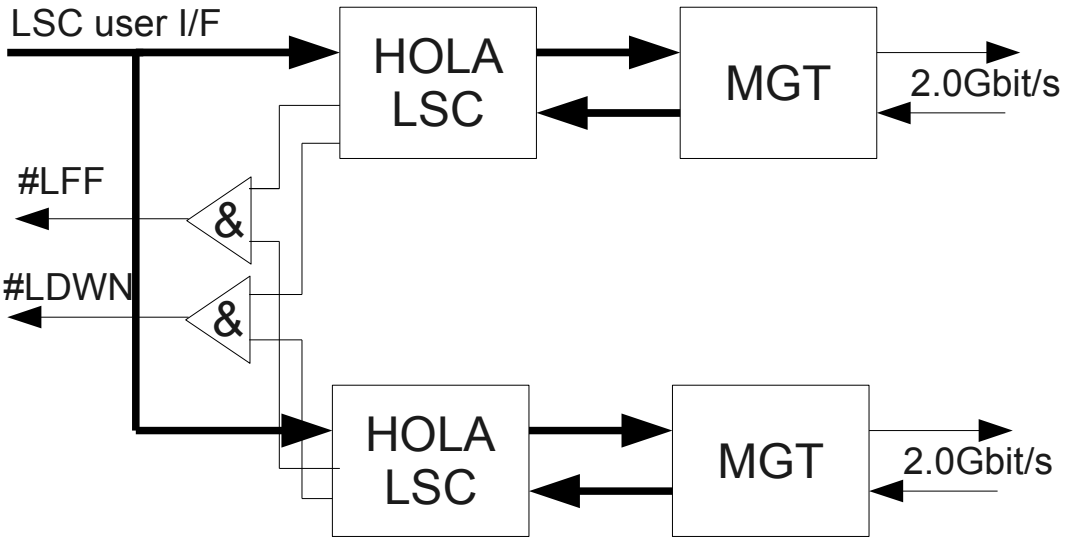

Figure 10: Dual-HOLA

for ATLAS. The BOC will take advantage from this fact by using an embedded S-Link implementation as well, with the protocol core and the high-speed serialisers/deserialisers inside the FPGA. To support the requirements of the FTK [7], the S-Link user data have to be copied to two ROLs, one feeding the TDAQ system and the other feeding the FTK trigger. This is done inside the main FPGAs in a way compatible to the dual-HOLA implementation [ibidem]. The LSC user interface is duplicated onto two HOLA LSC cores. The status signals link-full-flag (#LFF) and link-down (#LDWN) are combined, see Figure 10. Status generation can be selectively enabled via a programmable register.

Concerning the HOLA S-Link firmware, CERN maintains the VHDL source code implementing HOLA source and destination modules interfacing to an external TLK2501 serialiser/deserialiser device. This code can be used as is on the BOC, provided the on-chip gigabit transceivers (MGT[12]) can be configured to a TLK2501 compatibility mode. An initial proof-of-concept design targeted at Virtex-5 FPGA technology has been completed with reasonable success. Migration to Spartan-6 is work in progress:

- Simulation of LSC to LDC communication via FPGA gigabit transceivers is completed. The levels of S-Link user-data (at the LSC) and link-data (at the LDC) and the internal levels of LCS-TX and LDC-RX data corresponding to TLK2501 signals can be observerd - Figure 11 shows the propagation of data through these four levels.

[12] Xilinx nomenclature for high-speed serialisers/deserialisers varies from family to family. This documents uses the terminology “multi-gigabit-transceiver” - MGT which describes the functionality best.

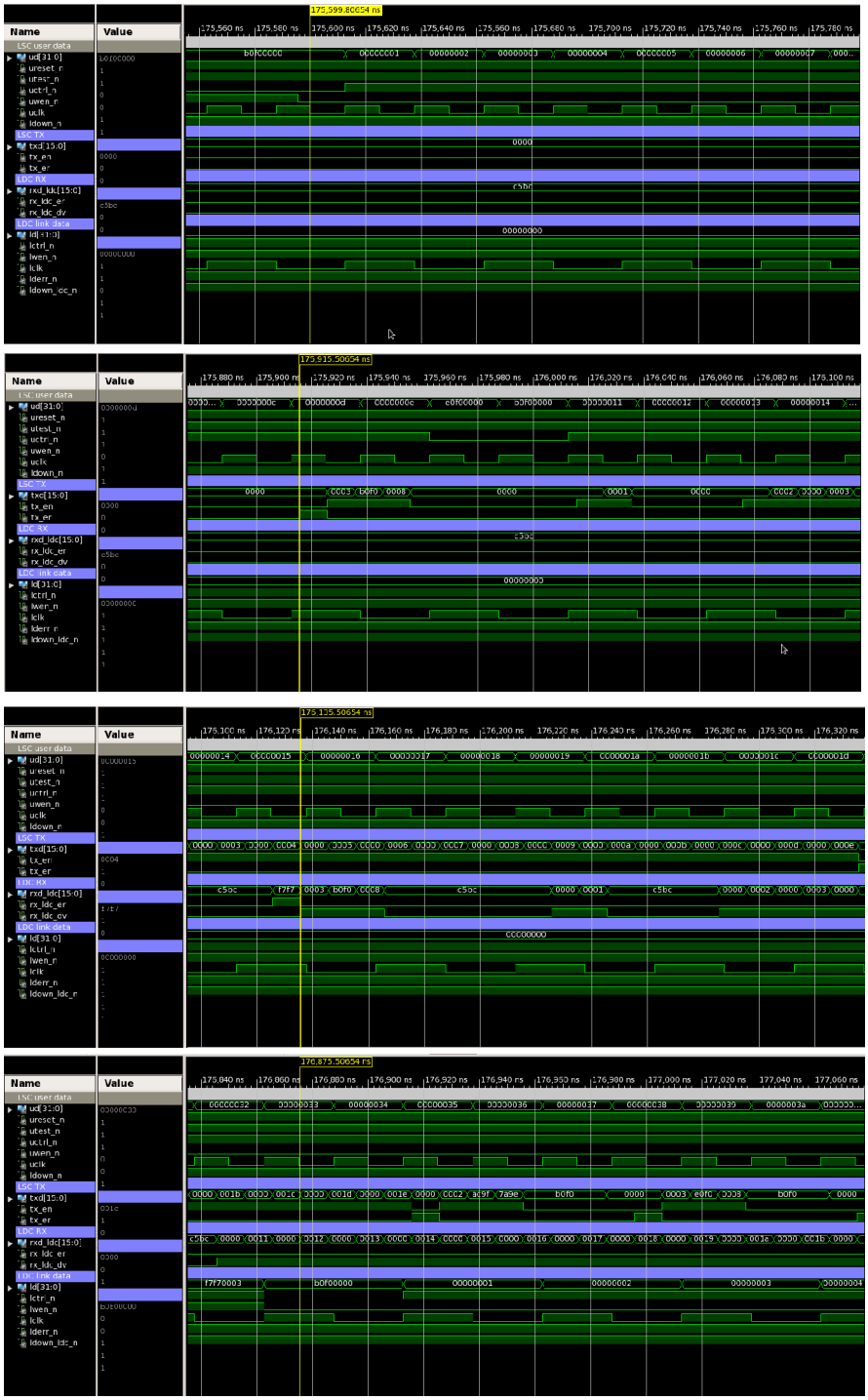

Figure 11: S-Link simulation

- Architecture migration of an LSC-LDC pair into a Spartan-6 FPGA is completed, resource utilisation is very low:
  - Number of occupied Slices: 287 out of 23,038 1%
  - Number of bonded IOBs: 95 out of 540 17%
  - Number of RAMB8BWERs: 1 out of 536 1%
  - Number of BUFG/BUFGMUXs: 7 out of 16 43%

- Number of DCM/DCM_CLKGENs: 2 out of 12 16%
- Number of GTPA1_DUALs: 1 out of 4 25%

The relatively high utilisation of BUFGs is an artefact from having both LSC and LDC implemented independently. The two twin-LSCs of the target implementation will consume less[13].

- Verification of the design and functional tests connecting a Spartan-6 LSC to a real ROBIN are under preparation.

The BOC exposes the user-data interface of the S-Link source (LSC) to the ROD. In order to reduce the number of signals lines the databits are transmitted in double-data-rate (DDR) style at 80Mbit/s using both edges of the 40MHz system clock (see also section 2.3.). The mapping is shown in Table 1.

Line	Signal(s)	Group
1	UD00 / UD01	Data
2	UD02 / UD03	Data
3	UD04 / UD05	Data
4	UD06 / UD07	Data
5	UD08 / UD09	Data
6	UD10 / UD11	Data
7	UD12 / UD13	Data
8	UD14 / UD15	Data
9	UD16 / UD17	Data
10	UD18 / UD19	Data
11	UD20 / UD21	Data
12	UD22 / UD23	Data
13	UD24 / UD25	Data
14	UD26 / UD27	Data
15	UD28 / UD29	Data
16	UD30 / UD31	Data
17	#UCTRL	Control
18	#UWE	Control
19	#URESET	Control
20	#UTEST	Control
21	#LDOWN	Status
22	#LFF	Status
23	UCLK	Unused, spare

Table 1: S-Link signals

[13] A sample quad gigabit transceiver designs with 2 GTPA1_DUALs consumes 3 BUFGs and 1 DCM.

The link return lines (LRL) are not used (as on the existing BOC). The data-width indication (UDW) is statically fixed to 32bit. Transmission of the user clock (UCLK) is not necessary, as the communication is system synchronous. This way each S-Link channel consists of 23 lines including 1 spare (unused UCLK). The four S-Link channels occupy rows A, B and C of connector J0, row D plus some pins of rows A, B and C of connector J2.

Typically, every ROL has a dedicated optical transceiver according to the SFP specification [8]. Following this approach requires eight SFP transceivers to be installed on the BOC, which is not a problem and can be done using one 4x1-ganged SFP cage per FPGA (Figure 12, left). An alternative solution which saves some PCB real-estate is to use QSFP transceivers, which contain 4 optical transmitters and 4 optical receivers in one device. A break-out fiber assembly attaches to the QSFP and converts into 4 individual LC-style bi-directional fibers (Figure 12, right). From the point-of-view of overall mechanics this could be an advantage.

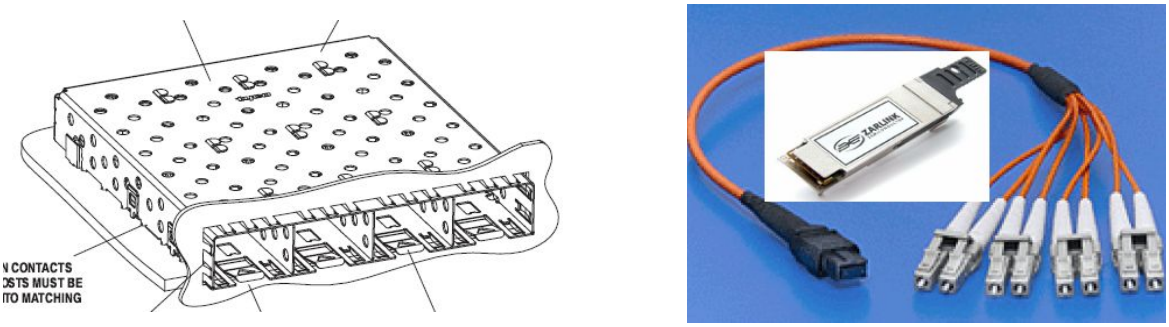

Figure 12: SFP options

4. Detector interfaces

Custom made electrical-optical interfaces (Figure 14) are used on the existing pixel readout system [23][24]. A 12 channel TX-module drives the optical fibers with an adjustable output power of up to 1.2mW optical (0.8dBm). In addition, the module provides bi-phase-mark-encoding, timing control circuitry for signal delay and mark-space-ratio adjustment on a per channel basis.

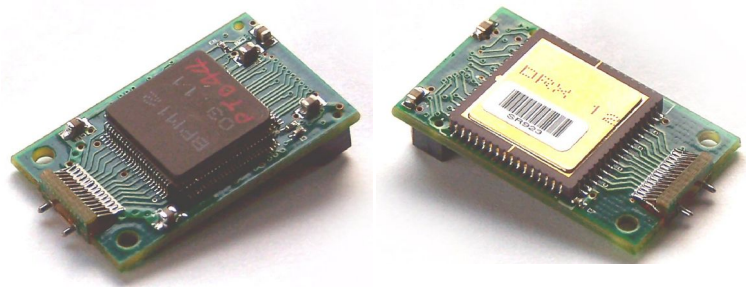

Figure 14: Current TX (left) and RX (right) modules

A 12 channel RX-module receives data from the detector, with individual DACs setting the receive threshold current. PHOS4 delay lines on the BOC enable de-skewing of the input data.

The chip technology used by the TX-modules is difficult to obtain today. The RX-modules are not suitable for the increased bandwidth of the new FE-I4 readout ASICs. In addition, all the discrete timing logic on the BOC and on the modules shall be integrated into the BOC FPGAs. Therefore, new electrical-optical interface modules have to be

found or developed for the new off-detector readout system. For the prototype BOC we will use commercial SNAP-12 transmitters and receivers which interface directly to the FPGAs. Those perform all encoding, decoding and timing adjustments. If this is a suitable solution for the final system has to be evaluated.

Two connectors with 5V compatible drivers are included on the BOC to support TX-modules of the current style.

4.1. CMD output

The command output path (CMD) comprises three main functions:

- Bi-phase-mark-encoding: data and clock are combined into a DC-balanced stream, as shown in Figure 15. This is a trivial operation and consumes virtually no resources in the FPGA.

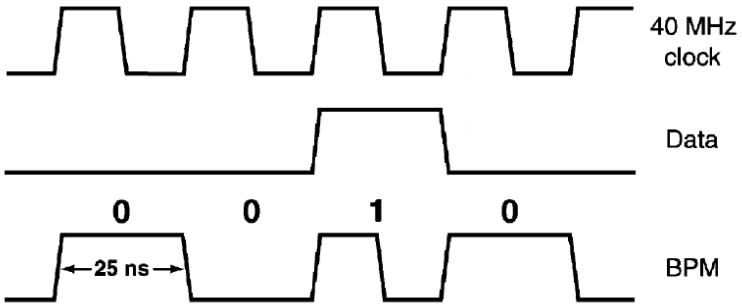

Figure 15: Bi-phase-mark encoding

- The mark-space-ration of the encoded stream has to be adjusted in order to help the circuitry on the receiving side to properly recover the encoded clock [27]. The delays are realised in the FPGA with a chain of delay elements[14] with programmable length. The fixed delay in front of the AND-gate shortens the ON-state to less than 50%. The delay in front of the OR-gate then extends the ON-state to the desired length, see Figure 16.

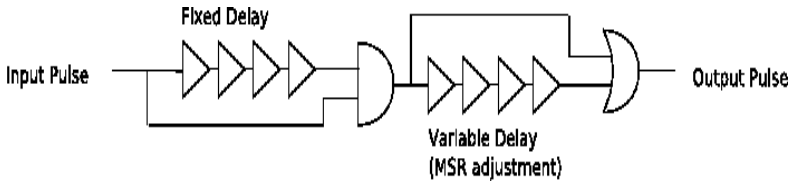

Figure 16: Mark-space-ratio adjustment

- Finally a programmable delay is applied to the the mark-space-ratio adjusted signal, using the same delay mechanism in the FPGA.

FPGA delay elements have different timing properties according to the FPGA family. In Spartan-6, the typical delay of is 300ps – 85 delay elements are sufficient to cover a full LHC clock period of 25ns[15]. Figure 17 shows the functional blocks associated with CMD output.

[14] Delay elements are made from MUX6 or MUX7 elements of the FPGA fabric, depending on the family.
[15] 85 MUXes corresponds to approximately 0.1% of FPGA resources.

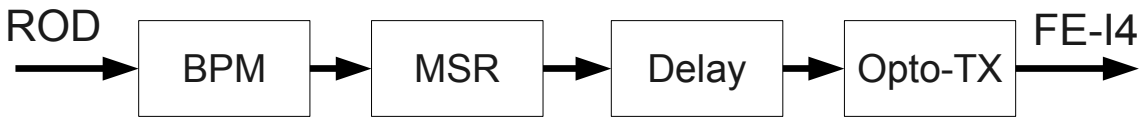

Figure 17: CMD output path

A commercial SNAP-12 transmitter (Figure 18) is used to drive the optical fibers. SNAP-12 is specified for gigabit transmission and operation at 40MHz has to be verified still[16]. Also, the output power is lower than originally specified for pixel readout: only -2dBm maximum optical power (0.6mW).

Figure 18: SNAP-12 TX (front), RX (rear)

However there is a chance that the integrated control circuitry (Figure 19) can be used to increase output power.

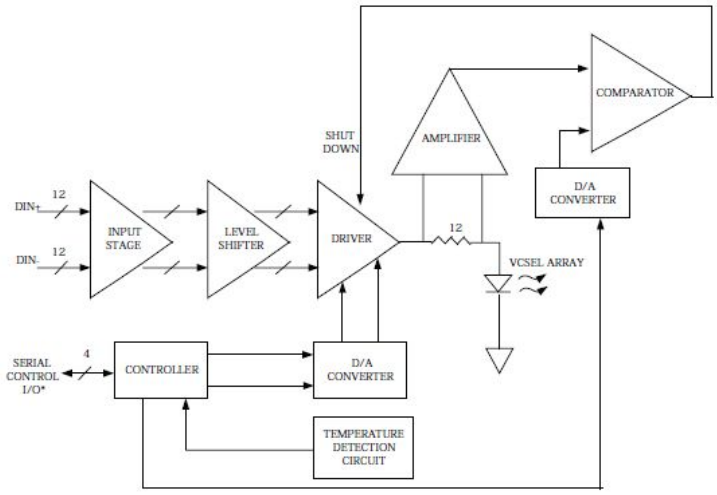

Figure 19: SNAP-12 transmitter

4.2. Front-end data input

The new FE-I4 ASIC sends data with DC-balanced 8B10B encoding at a line-rate of 160Mbit/s. A SNAP-12 receiver is used to receive optical data and convert into electrical levels. As with the transceivers, these devices are made for gigabit data-rates and operation at 160Mbit/s has to be verified. Also, the sensitivity might not be appropriate[17].

[16] Private communication with the Reflex Photonics FAE: “The Snap 12s will work fine down to a few ten's of kBps as long as it is a balanced pattern ...”

[17] SNAP-12 sensitivity is -18dBm, saturation -1dBm

Assuming the SNAP-12 receiver is able to provide a good signal, the FPGA will first need to do a phase alignment using multi-phase sampling [25]. As BOC and detector run from the same clock (the LHC clock) no elastic buffers for clock-data-recovery are required. The aligned data stream is then processed by an 8B10B-decoder, which provides 8-bit wide data words at 16MHz to a small FIFO, implemented with distributed memory in the FPGA.

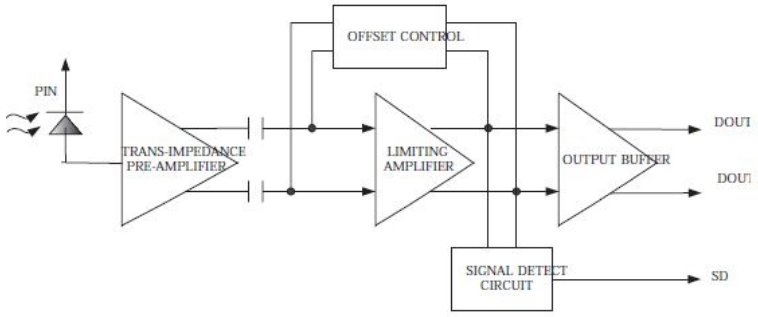

Figure 20: SNAP-12 receiver

An arbiter/multiplexer stage samples the FIFOs from four input channels and encodes data, channel address and a valid signal at 80MHz onto one of the RX-channels towards the ROD. Figure 21 shows the functional blocks associated with RX input.

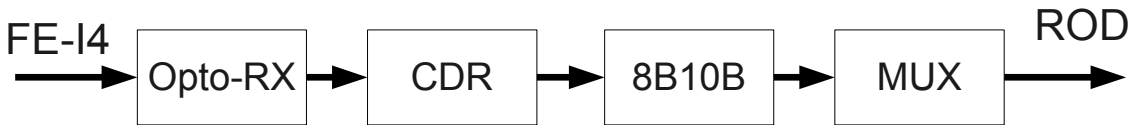

Figure 21: RX input

No processing is done at this stage and all non-idle FE-I4 data words are forwarded to the ROD unmodified.

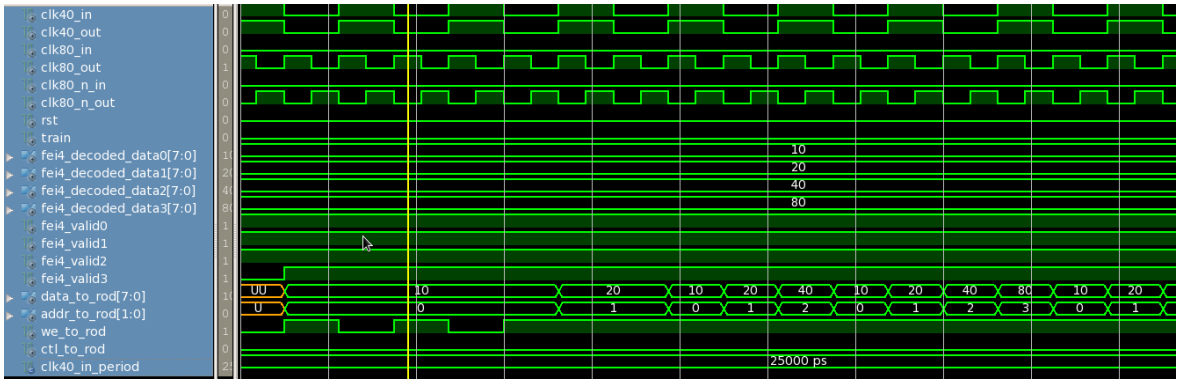

Figure 22: RX channel Boc2Rod

Figure 22 shows the transmission of input data[18] into four FIFOs, readout and encoding with channel address (0..3). The implementation consumes 450 slices, less than 1% of the FPGA resources.

[18] Channel data encode channel number as 0x10, 0x20, 0x40, 0x80.

5. Laser interlock mechanism

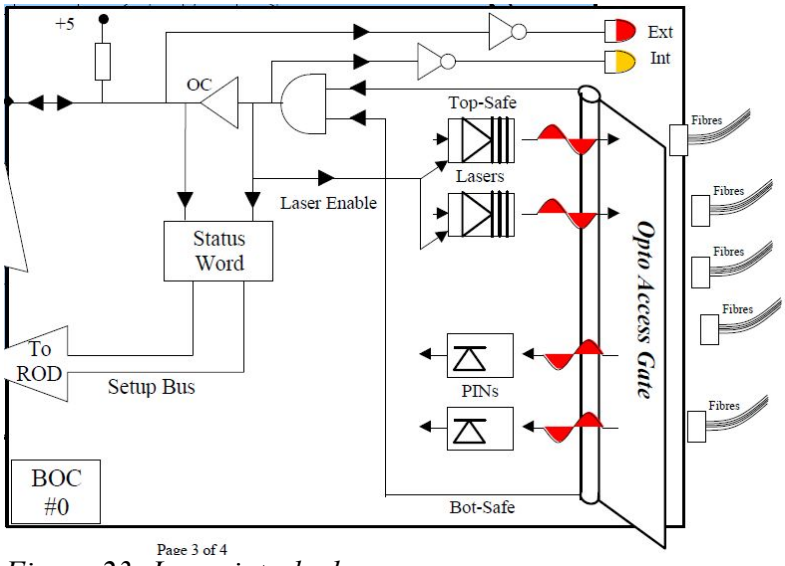

Figure 23: Laser interlock

An interlock mechanism (Figure 23, [21]) is used by the present pixel readout system, which disables the lasers once physical access to the optical interfaces is performed. An equivalent circuitry will be implemented on the new BOC.

6. DCS

DCS [18] is the DAQ system comprising the control of the sub-detectors and of the common infrastructure of the experiment and the communication with the services of CERN (cooling, ventilation, electricity distribution, safety etc.) and the LHC accelerator. The interface to the detectors is done via a CAN-bus infrastructure and locally deployed mezzanine boards, the ELMBs (embedded local monitoring boards) [19].

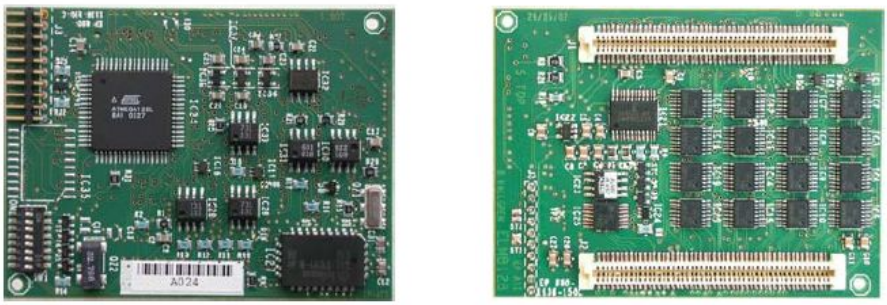

Figure 24: ELMB with ADC/MUX

ELMBs come in two flavors, analog (with ADC/DACs, see Figure 24) and digital (without converters). The analog ELMB provides 32 differential (64 single ended) channels with a programmable full-scale range from 25mV to 5V. The analog channels will be used to monitor supply voltages, currents and local temperatures (via NTC resistors). A digital connection between ELMB and PCF will be provided as well, for future use.

The ELMB connects via two FCI “BergStak” connectors to the BOC (61083-101400LF on BOC, 61082-101400LF on ELMB).

7. References

[1] IBL readout document (restricted access), ATU-SYS-CD-0001, https://espace.cern.ch/atlas-ibl/OffDecWG/Shared%20Documents/Readout%20System/EDMS%20Docs/ATU-SYS-CD-0001.doc

[2] SNAP-12 parallel optics, e.g. Reflex Photonics SN-T12-C00501/SN-R12-C00501 (http://reflexphotonics.com/PDFs/SN-970-004-00_Rev_3-6_InterBoard_Board-Edge_Data_Sheet.pdf), Avago HFBR-772BWZ/HFBR-782BZ (http://www.avagotech.com/docs/AV02-1158EN), Zarlink ZL60101MLDL/ZL60102MLDL (http://www.tycoelectronics.com/catalog/bin/TE.Connect?C=28066&M=PPROP&P=&BML=&LG=1&PG=1&IDS=519863,519864,519848,519847,520057&N=1)

[3] HOLA S-Link documentation: http://hsi.web.cern.ch/HSI/s-link/devices/hola

[4] S-Link specification: http://hsi.web.cern.ch/HSI/s%2Dlink/spec/spec/s-link.pdf

[5] QSFP, Zarlink ZL60505MKDB, http://www.tycoelectronics.com/catalog/bin/TE.Connect?C=28066&M=PPROP&P=&BML=&LG=1&PG=1&IDS=519863,519864,519848,519847,520057&N=3

[6] R. Cranfield The ATLAS ROBIN, J. Instrum. 3 (2008) T01002, http://dx.doi.org/10.1088/1748-0221/3/01/T01002

[7] FTK Technical Proposal, http://hep.uchicago.edu/~shochet/FTK_NSF/FTK_TP.pdf

[8] SFP multi-source agreement, ftp://ftp.seagate.com/sff/INF-8074.PDF

[9] MC10EP16, 3V-PECL to 5V-PECL driver, http://www.onsemi.com/pub/Collateral/MC10EP16-D.PDF

[10] MC100LVEL92, 5V-PECL to 3V-PECL receiver, http://www.onsemi.com/pub/Collateral/MC100LVEL92-D.PDF

[11] MC100EP195B, Programmable delay line, http://www.onsemi.com/pub/Collateral/MC100EP195B-D.PDF

[12] TI CDCE62002 jitter-cleaner, http://focus.ti.com/docs/prod/folders/print/cdce62002.html

[13] Maxim DS2401, silicon serial number, http://datasheets.maxim-ic.com/en/ds/DS2401.pdf

[14] SiLabs SI532 dual-frequency oscillator, http://www.silabs.com/pages/DownloadDoc.aspx?FILEURL=Support%20Documents/TechnicalDocs/si532.pdf&src=ProductMatrix

[15] DDR2 RAM 32Mbit*16, e.g. Micron MT47H32M16HR-25E:G, http://www.micron.com/get-document/?documentId=455&file=512MbDDR2.pdf

[16] Gigabit Ethernet PHY, e.g. National http://www.national.com/profile/snip.cgi/openDS=DP83865

[17] Spartan-6 configuration user guide, http://www.xilinx.com/support/documentation/user_guides/ug380.pdf

[18] ATLAS DCS system, https://twiki.cern.ch/twiki/bin/viewauth/Atlas/DetectorControlSystemMainPage

[19] DSC ELMB documentation, http://elmb.web.cern.ch/ELMB/ELMBhome.html

[20] FCI BergStak series, http://portal.fciconnect.com

[21] ATLAS Pixel BOC interlock mechanism, https://www.hep.phy.cam.ac.uk/~goodrick/AtBOC/BOC_Interlock.pdf

[22] G. Aad et al., ATLAS pixel detector electronics and sensors, http://iopscience.iop.org/1748-0221/3/07/P07007/pdf/1748-0221_3_07_P07007.pdf

[23] M. L. Chu et al., The off-detector opto-electronics for the optical links of the ATLAS Semiconductor Tracker and Pixel detector, http://dx.doi.org/10.1016/j.nima.2004.04.228

[24] T. Flick et al., Optical readout in a multi-module system test for the ATLAS pixel detector, http://dx.doi.org/10.1016/j.nima.2006.04.083

[25] Xilinx CDR application note xapp224, http://www.xilinx.com/support/documentation/application_notes/xapp224.pdf

[26] T. Flick, Studies on the Optical Readout for the ATLAS Pixel Detector, PhD thesis, Wuppertal, 2006, http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:hbz:468-20060600

[27] J. Dopke, Studies for automated tuning of the quality of the Back-of-Crate card for the ATLAS Pixel detector, Diploma thesis, Wuppertal, 2007, http://www.atlas.uni-wuppertal.de/~dopke/diplom.pdf