OptoHybrid v3 Firmware

Thomas Lenzi ([thomas.lenzi@cern.ch)](mailto:thomas.lenzi@cern.ch))  
Andrew Peck ([andrew.peck@cern.ch)](mailto:andrew.peck@cern.ch))  
Evaldas Juska ([evaldas.juska@cern.ch](mailto:evaldas.juska@cern.ch))

August 7th, 2017

This document describes how to interact with the OptoHybrid (OH) modules and how to parameterize and use the various functionalities integrated in the firmware.

# Contents

[Contents 1](#_Toc490486035)

[Introduction 3](#_Toc490486036)

[Synchronization 4](#_Toc490486037)

[GBT E-Link Deserialization 5](#_Toc490486038)

[In “16 bit” full-bandwidth mode 5](#_Toc490486039)

[In “10 bit” low-bandwidth mode 5](#_Toc490486040)

[GBT Packets (Full Bandwdith Mode) 6](#_Toc490486041)

[Transmit (FPGA to GBT) 6](#_Toc490486042)

[Receive (GBT to FPGA) 8](#_Toc490486043)

[GBT Packets in Low Bandwidth Mode 9](#_Toc490486044)

[Transmit (FPGA to GBT) 9](#_Toc490486045)

[Receive (GBT to FPGA) 10](#_Toc490486046)

[Trigger 11](#_Toc490486047)

[Registers 12](#_Toc490486048)

[Loopback 13](#_Toc490486049)

[Addressing 13](#_Toc490486050)

[Errors 13](#_Toc490486051)

[Errors to avoid 13](#_Toc490486052)

[Counters 14](#_Toc490486053)

[Addressing 14](#_Toc490486054)

[System Registers 15](#_Toc490486055)

[Addressing 15](#_Toc490486056)

[Status Registers 16](#_Toc490486057)

[Addressing 16](#_Toc490486058)

[ADC 17](#_Toc490486059)

[Addressing 17](#_Toc490486060)

[Errors 17](#_Toc490486061)

[Errors to avoid 17](#_Toc490486062)

[History 18](#_Toc490486063)

# Introduction

# Synchronization

# GBT E-Link Deserialization

With the existence of two different communication modes corresponding yielding 16 bits/bx and 10 bits/bx (duplex) there will be two alternative assignments of which bits will be carried on which e-links.

In “16 bit” full-bandwidth mode:

We specify that one eLink (labelled in firmware as eLink0) be run at 320 MHz and carry the 8 least-significant bits of the packet. The other eLink (labelled in firmware as eLink1) will run at 320 MHz and carry the 8 most-significant bits.

i.e.

|  |  |  |
| --- | --- | --- |
|  | **E-link** | **Payload** |
| RX  (GBT to OH) | eLink1 | TTC[3:0],  data[11:8] |
| eLink0 | data[7:0] |
| TX  (OH to GBT) | eLink1 | data[15:8] |
| eLink0 | data[ 7:0] |

In “10 bit” low-bandwidth mode:

We specify that one eLink (labelled in firmware as eLink0) be run at 80 MHz and carry the 2 least-significant bits of the packet. The other eLink (labelled in firmware as eLink1) will run at 320 MHz and carry the 8 most-significant bits.

i.e.

|  |  |  |
| --- | --- | --- |
|  | **E-link** | **Payload** |
| RX  (GBT to OH) | eLink1 | TTC[3:0],  data[5:2] |
| eLink0 | data[1:0] |
| TX  (OH to GBT) | eLink1 | data[9:2] |
| eLink0 | data[1:0] |

Within each e-link, all bits are transmitted and most-significant bit first.

# GBT Packets (Full Bandwdith Mode)

In “Full Bandwidth Mode” the GBT to FPGA link consists of two 320 Mbps duplex links. In version 3a this was not implemented, and a separate functionality of 1x 320 Mbps duplex and 1x 80Mbps duplex was adopted. This alternative, “Low Bandwidth Mode” is covered in the next section.

### Transmit (FPGA to GBT)

Communication with the optohybrid in the direction of FPGA to GBT is accomplished with a simple 16 bit packet format.

A valid packet is three frames long and will consist of the 16 bit frame marker BCBC followed by two 16 bit data frames:

|  |  |
| --- | --- |
| **Frame** | **frame[15:0]** |
| Start | 0xBCBC |
| Data0 | data[31:16] |
| Data1 | data[15: 0] |

For example, this waveform shows the transmission of the 32-bit packet 0x12345678:



Consecutive packets will simply repeat this sequence, e.g:

|  |  |
| --- | --- |
| **Frame** | **frame[15:0]** |
| Start | 0xBCBC |
| Data0 | data[31:16] |
| Data1 | data[15: 0] |
| Start | 0xBCBC |
| Data0 | data[31:16] |
| Data1 | data[15: 0] |

When the transmit FIFO is empty and there is no data to send, the state machine will send an idle frame marker, 0xFCFC:

|  |  |
| --- | --- |
| **Frame** | **frame[15:0]** |
| Start | 0xBCBC |
| Data0 | data[31:16] |
| Data1 | data[15: 0] |
| IDLE | 0xFCFC |
| IDLE | 0xFCFC |
| IDLE | 0xFCFC |
| IDLE | 0xFCFC |
| Start | 0xBCBC |
| Data0 | data[31:16] |
| Data1 | data[15: 0] |

For example, this waveform shows the transmission of the 32-bit packet 0x12345678, followed by the idle transmission of the 0xFCFC frame marker.



The receiving state machine should monitor for 0xBCBC for alignment and record the next two frames as the 32 bit data packet.

If the following frame is 0xBCBC, then the sequence should repeat and record the following two frames, and so on. If the frame is 0xFCFC then the state machine should wait until it sees a start-of-packet marker (0xBCBC).

### Receive (GBT to FPGA)

Communication in the direction of GBT to FPGA is similarly accomplished with a packet format, whose payload includes 32 bits of address and 32 bits of data. Transmit and receive bandwidths are asymmetric, due to the requirement of transmitting synchronous TTC commands on the GBT to FPGA link, so the format on the receive side is somewhat more complicated.

In this direction, the highest 4 bits of the GBT link are reserved for TTC commands:

* Bit15 = L1A
* Bit14 = VFAT Reset
* Bit13 = TTC Resync
* Bit12 = BC0

The remaining 12 bits are used as the GBT data frame used to construct packets.

|  |  |  |
| --- | --- | --- |
| **Frame** | **Highest 4 bits** | **Lowest 12 bits** |
| Start | L1A + reset + Resync + BC0 | 0xABC |
| Header | L1A + reset + Resync + BC0 | request\_valid + wr\_enable + address[31:24] |
| Address 0 | L1A + reset + Resync + BC0 | address[23:12] |
| Address 1 | L1A + reset + Resync + BC0 | address[11: 0] |
| Data 0 | L1A + reset + Resync + BC0 | reserved[3:0] + data[31:24] |
| Data 1 | L1A + reset + Resync + BC0 | data[23:12] |
| Data 2 | L1A + reset + Resync + BC0 | data[11: 0] |

For continuous transmission, this sequence should just be repeated cyclically. i.e. the frame after Data 2 should be “Start” (0xABC). E.g. as follows:

**<START> <HEADER> <ADR0> <ADR1> <DATA0> <DATA1> <DATA2>, <START> <HEADER> <ADR0> ….**

When idle, there are two possible ways of sending data to the OH:

Either cycle through the 7 frame packet format continuously, making sure to set request\_valid to 0 for any null packets. The Optohybrid will process and discard the packet.

Alternatively (preferably) when can send an idle frame marker, for example 0x5FC. The exact choice does not matter as long as it is not 0xABC. This marker can be sent continuously as long as the transmission is idle and the Optohybrid will remain in an idle SYNC state. The state machine will monitor for 0xABC and if received, will continue to process the rest of the frame normally.

# GBT Packets in Low Bandwidth Mode

In “Low Bandwidth Mode” the GBT to FPGA link consists of 1x 320 Mbps duplex and 1x 80Mbps duplex, leaving 10 bits per bunch crossing in each direction.

### Transmit (FPGA to GBT)

Communication with the optohybrid in the direction of FPGA to GBT is accomplished with a simple 10 bit packet format.

A valid packet is three five long and will consist of the a bit frame marker BC followed by three 10 bit data frames:

|  |  |
| --- | --- |
| **Frame** | **frame[9:0]** |
| Start | 2’b00 & 0xBC |
| Data0 | 2’b00 & data[31:24] |
| Data1 | 2’b01 & data[23:16] |
| Data2 | 2’b10 & data[15: 8] |
| Data3 | 2’b11 & data[ 7: 0] |

For example, this waveform shows the transmission of the 32-bit packet 0x12345678:



Note that the highest two bits of the data frames indicate the numbering of the frame in the packet and can be used for error checking.

Consecutive packets will simply repeat this sequence.

When the transmit FIFO is empty and there is no data to send, the state machine will send an idle frame marker, 0xFC. For example, this waveform shows the transmission of the 32-bit packet 0x12345678, followed by the idle transmission of the 0xFC frame marker, followed by the beginning of a new valid packet.



The receiving state machine should monitor for 0xBC for alignment and record the next four frames as the 32 bit data packet.

If the frame following the last data frame is 0xBC, then the sequence should repeat and record the following two frames, and so on. If the frame is 0xFC (or anything else) then the state machine should wait until it sees a start-of-packet marker (0xBC).

### Receive (GBT to FPGA)

Communication in the direction of GBT to FPGA is similarly accomplished with a packet format, whose payload includes 32 bits of address and 32 bits of data. Transmit and receive bandwidths are asymmetric, due to the requirement of transmitting synchronous TTC commands on the GBT to FPGA link, so the format on the receive side is somewhat more complicated.

In this direction, the highest 4 bits of the GBT link are reserved for TTC commands:

* Bit10 = L1A
* Bit9 = VFAT Reset
* Bit8 = TTC Resync
* Bit7 = BC0

The remaining 6 bits are used as the GBT data frame used to construct packets.

|  |  |  |
| --- | --- | --- |
| **Frame** | **Highest 4 bits** | **Lowest 6 bits** |
| Start | L1A + reset + Resync + BC0 | 0x2A |
| Header | L1A + reset + Resync + BC0 | request\_valid + wr\_enable + address[31:28] |
| Address 0 | L1A + reset + Resync + BC0 | address[27:22] |
| Address 1 | L1A + reset + Resync + BC0 | address[21:16] |
| Address 2 | L1A + reset + Resync + BC0 | address[15:10] |
| Address 3 | L1A + reset + Resync + BC0 | address[ 9: 4] |
| Address 4 | L1A + reset + Resync + BC0 | address[3:0] + data[31:30] |
| Data 0 | L1A + reset + Resync + BC0 | data[29:24] |
| Data 1 | L1A + reset + Resync + BC0 | data[23:18] |
| Data 2 | L1A + reset + Resync + BC0 | data[17:12] |
| Data 3 | L1A + reset + Resync + BC0 | data[11: 6] |
| Data 4 | L1A + reset + Resync + BC0 | data[ 5: 0] |

For continuous transmission, this sequence should just be repeated cyclically. i.e. the frame after Data 2 should be “Start” (0xABC). E.g. as follows:

**<START><HEADER><ADR0><ADR1><ADR2><ADR3><ADR4><DATA0><DATA1><DATA2><DATA3><DATA4>, <START>**

When idle, there are two possible ways of sending data to the OH:

Either cycle through the 7 frame packet format continuously, making sure to set request\_valid to 0 for any null packets. The Optohybrid will process and discard the packet.

Alternatively (preferably) when can send an idle frame marker, for example 0x1C. The exact choice does not matter as long as it is not 0x2A. This marker can be sent continuously as long as the transmission is idle and the Optohybrid will remain in an idle SYNC state. The state machine will monitor for 0x2A and if received, will continue to process the rest of the frame normally.

# Trigger

# Registers

A summary table of the wishbone slaves in the Optohybrid v3 is given here. Detailed descriptions of individual registers follow on subsequent pages.

|  |  |  |  |
| --- | --- | --- | --- |
| Slave | Name | Addresss | Function |
| Wishbone Slaves | | | |
| 0 | Loopback | 0x40000000 | Loopback register receives data and responds with the same data, for testing communication and error rates. Data in the loopback register has no effects in the FPGA. |
| 1 | Counters | 0x410000YY | Counters register allows readback of a number of counters |
| 2 | System | 0x420000YY | System register is responsible for controlling various settings in the Optohybrid |
| 3 | Status | 0x430000YY | Status register is responsible for readback of various status |
| 4 | ADC | 0x440000YY | Controls the System Monitor of the Virtex-6 FPGA |

In an address such as 0x420000YY, the lowest 8 bits denoted by YY are used to select between individual end-points in the wishbone bus. Splitting the request is handled by the individual wishbone slaves.

## Loopback

This module is designed for testing GBT communication. It will echo as a response whatever data is written to it as a request.

### Addressing

Module ID 0

Address 0x40000000

|  |  |  |
| --- | --- | --- |
| Y register | Mode | Function |
| Loopback Registers | | |
| 0 | Read/write | Echoes back the last written value as a wishbone response |

### Errors

None

### Errors to avoid

* None

## Counters

This module holds all the counters of the OptoHybrid. Writing to a given register will reset its value.

All counters are 32 bit

### Addressing

Module ID 1

Address 0x410000YY

|  |  |  |
| --- | --- | --- |
| Y register | Mode | Function |
| Control | | |
| 0 | Read | *Snap Enable*  Write (anything) to this register to take a snapshot of the counters, freezing a shadow copy for synchronous readout. This MUST be flagged before reading out a register or it will not be updated.  The register will readback a counter of the number of snapshots taken |
| 1 | Read | *Reset All*  Write anything to this register to reset all of the counters  The register will readback a counter of the number of reset\_alls done. This register is reset by writing to the register. A non-writing request will readback the contents. |
| Wishbone |  |  |
| 2 | Read | *Wishbone master strobes* |
| 3 | Read | *Wishbone master acknowledgments* |
| 4—8 | Read | *Wishbone slaves strobe*  Order: Loopback, counters, system, status, adc |
| 9—13 | Read | *Wishbone slaves acknowledgments*  Order: Loopback, counters, system, status, adc |
| TTC |  |  |
| 14 | Read | *L1A Counter* |
| 15 | Read | *Resync Counter* |
| 16 | Read | *BC0 Counter* |
| 17 | Read | *BC0 Sync Error Counter* |
| MMCM | | |
| 18 | Read | *GBT Frame Clock MMCM Locked* |
| 19 | Read | *Fabric Clock MMCM Locked* |
| 20 | Read | *Both MMCMs Locked* |
| GBT | | |
| 21 | Read | *GBT Link Errors* |
| SEM |  |  |
| 22 | Read | *SEM Corrections* |
| S-Bits |  |  |
| 23 | Read | *SBit Overflows* |
| 24—47 | Read | *Active VFAT Flags count the number of the number of bx with at least one S-bit in a VFAT* |
| Latency Timers | | |
| 48—71 | Read | *Latency timer: Number of BX between the SBits and the L1A, for each VFAT* |

## System Registers

System registers are writable registers used to control programmable settings of the Optohybrid

### Addressing

Module ID 2

Address 0x420000YY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y register | Mode | Bits | Function | |
| VFAT | | | |
| 1 | Read / write | 4:0 | *SBit magic select for HDMI* | |
| 2 | Write only | 0 | *VFAT reset*  Write any data to this register and it will reset the VFATs (for 1 bx.. is this sufficient??). This register automatically returns to a 0 state | |
|  |  |  |  | |
| 2 | Read / write | 23:0 | *VFAT trigger data mask*  Allows to mask individual VFATs for trigger data | |
| 3 | Read / write | 29:0 | *TDC SBits selects* | |
| 4 | Read / write | 1:0 | *TDC SBits mode*  0 : single VFAT2  1 : Eta row (OR of 3 VFAT2s)  2 : Sector (OR of 4 VFAT2s)  3 : no output | |
| 5 | Read / write | 0 | FMM ignore start/stop  The FMM will ignore the value of force stop and will either immediately send trigger data or wait for the first BC0, depending on the configuration of “fmm\_dont\_wait” | |
| 6 | Read / write | 0 | FMM force stop  This will force the FMM state machine to idle the trigger outputs, unless ignore start/stop is asserted. | |
| 7 | Read / write | 0 | FMM dont\_wait  The FMM state machine which controls the trigger will not wait for receipt of a BC0 before sending trigger data. | |
| 8 | Read / write | 11:0 | TTC BXN Offset  This 12 bit configuration will set the “zero” value of the local BC0 counter which will be assumed after resync. Adjust this to account for different latencies to chambers. When this register is correctly set the BC0 error counters in the status register should be zero.  Note that since the OH does not have a startup configuration this must be programmed by the CTP-7 after every hard reset. | |
| 9 | Read / write | 3:0 | *Trigger Deadtime*  Adds a deadtime, 0-15 bunch crossings, on each S-bit which will prevent retriggering on that same S-bit during the time window specified by the deadtime. This is meant to supress afterpulsing in the trigger data path. | |

## Status Registers

List of status registers

### Addressing

Module ID 3

Address 0x430000YY

|  |  |  |  |
| --- | --- | --- | --- |
| Y register | Mode | Bit(s) | Function |
| Version | | | |
| 0 | Read only | 31:0 | *OH firmware date*  *Hex encoded: YYYYMMDD* |
| 1 | Read only | 31:0 | *Firmware version Hex Encoded*  Major 8 bits  Minor 8 bits  Version 8 bits  HW 8 bits = Hardware ID A = v3A, etc. |
| Clocking | | | |
| 2 | Read only | 0  1  2 | *Both MMCMs Locked*  *GBT SerDes Frame Clock MMCM Locked*  *Fabric clock MMCM Locked* |
| Soft Error Mitigation | | | |
| 3 | Read only | 0 | *Critical error induced by SEU detected* |
| GBT |  |  |  |
| 4 | Read only | 0  1  2 | *GBT TX Ready*  *GBT RX Valid*  *GBT RX Ready* |
| Trigger |  |  |  |
| 5 | Read only | 31:0 | *Averaged cluster rate in Hertz* |
| TTC |  |  |  |
| 6 | Read only | 31:0 | *Local BC0 Counts* |
| 7 | Read only | 31:0 | *Received BC0 Counts from GBT* |
| 8 | Read only | 31:0 | *TTC Orbit Counter* |
| 9 | Read only | 11:0 | *TTC BXN Count* |

## ADC

This module is directly connected to the xADC of the Virtex-6 FPGA. Refer to the following user guide for a full list of register: <http://www.xilinx.com/support/documentation/user_guides/ug370.pdf>

### Addressing

Module ID 4

Address 0x440000YY

|  |  |  |
| --- | --- | --- |
| Y register | Mode | Function |
| ADC registers | | |
| 0 - 79 | Read/write | Returns the conversion value of a given channel  0 = Temperature  1 = VCCINT  2 = VCCAUX  3 = VP/VN  16-31 = VAUX[0:15] (not connected)  32 = Temperature max  33 = VCCINT max  34 = VCCAUX max  36 = Temperature min  37 = VCCINT min  38 = VCCAUX max  64-66 = Control registers[0:2]  67-71 = Test registers[0:4]  72-79 = Sequencer registers[0:7] |

### 

### Errors

The module returns an error if the parameters are not in spec.

### Errors to avoid

* The register ID must be in the range 0 to 79.

# History

|  |  |  |
| --- | --- | --- |
| Date | Author | Modification |
| 8/7/2017 | AP | Initial conversion to OHv3 from Thomas’ v2 document |
| 8/10/2017 | AP | Description of GBT link formats |
| 8/14/2017 | AP | Updates register definitions according to latest firmware |