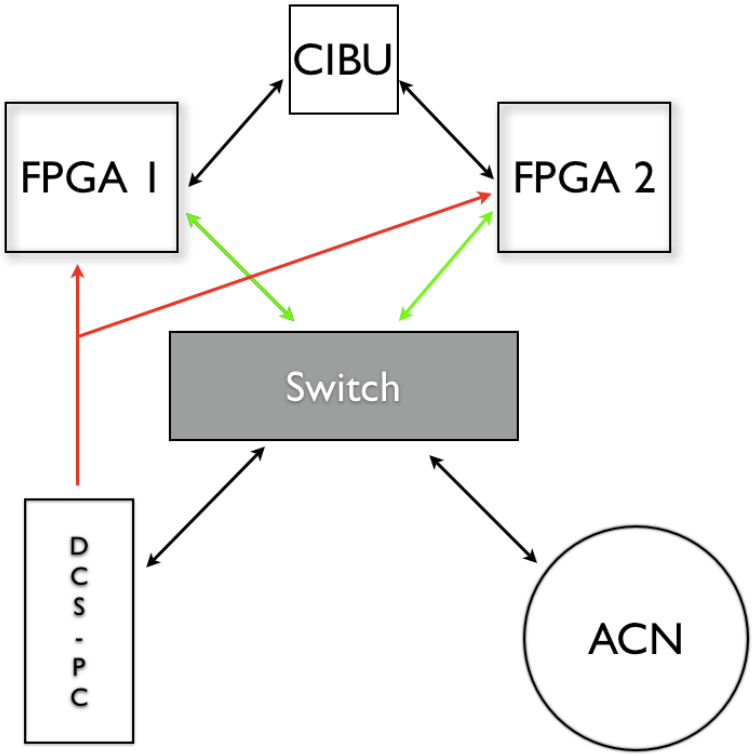
Documentation Interface PC - FPGA (BCM Only)

Boštjan Maček, Michael Niegl
Version 1.2 (Draft) 180408
For reference see also
https://twiki.cern.ch/twiki/bin/view/Atlas/BcmConfigParams



Red = JTAG
Green = Ethernet 100 MBit (necessary!)
Black = Ethernet any speed, TTL signals

Both FPGAs are connected to the ATLAS Controlled Network (ACN) via a 100 MBit Ethernet connection. The FPGAs both have static IP addresses. The DCS PC running PVSS is also connected to the ACN. The PC is needed for the following FPGA tasks:

- Programming & Debugging via JTAG & ChipScope
- Receiving and storing the buffer dump data after a beam abort
- Receiving and making available status messages & statistics continuously to the shifter
- Setting control parameters as well as trigger actions

Both FPGAs are constantly connected to the DCS PC via the Xilinx JTAG-USB Programmers. The PC has 3 USB ports so when integrating a third FPGA for the BLM a USB Hub is needed for flexibility purposes. ChipScope allows the handling of several FPGAs from one PC, they just need to be specifically choosen when establishing a connection. JTAG programming initiated by the PC always overrides the boot image on the Compact Flash card.

The procedure for the buffer dumps is the following:
After receiving a post mortem signal from the CIBU or creating a beam abort signal itself, both FPGAs go into read-out state, ie they both stop capturing data and freeze the two big buffers. They both send a special packet (read-out-ready) to the DCS PC indicating that they are ready to be read out. The PC upon receiving these messages starts the read-out on FPGA 1. For security reasons a handshaking mechanism is implemented, meaning every single packet form the FPGA has to be acknowledged by the PC. In case of errors the last packet is resent. After the transfer of all the data from FPGA 1 is finished and the special read-out over packet is sent the PC starts the transfer of the data from FPGA 2. Both FPGAs immediately return to capture state as soon as the transfer of all their data is complete.

The packets from the FPGA are fully RFC compliant UDP packets. Detailed description:

Start-packet per buffer dump

```
003005ecb5ea000a35019b2c08004500
00e400004000fffafbb0c0a80007808d
3d32ffff0ff000cf0000f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0
f0f0f0f0f0f0f0f0f0f0e00001000000
00
```

Sample packet with DDR-data

```
003005ecb5ea000a35019b2c08004500
00e400004000fffafbb0c0a80007808d
3d32ffff0ff000cf0000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000e00002000000
00
```

003005ecb5ea000a35019b2c08004500
00e400004000fffafbb0c0a80007808d
3d32ffff0ff000cf0000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
00

Sample packet with DDR2-data

003005ecb5ea000a35019b2c08004500
00e400004000fffafbb0c0a80007808d
3d32ffff0ff000cf0000101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
1010101010101010101010101010101010
00

Read-Out Ready Packet

003005ecb5ea000a35019b2c08004500
00e400004000fffafbb0c0a80007808d
3d32ffff0ff000cf0000010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
0101010101010101010101010101010101
00

Read-Out Over Packet

Red = IP & UDP Header
Blue = Datatype indicator, 0 = raw data, e = proc data
Green = continuous packet number
Yellow = Zero padding
Rest = data

The data on the PC side is stored on the local hard disk as well as automatically copied to CASTOR. They should be split into one file each for raw and processed data as well as per FPGA, therefore 1 buffer dump results in 4 separate files. The filename nomenclature is the following:

bufdump_FPGA<1/2>_<proc/raw>_<date>_<time>.dat

The status messages to the DCS are to be sent with a frequency of 1 Hz. They contain general FPGA status information as well as data statistics. The packet structure is defined in <EDMS no.>

In the direction PC -> FPGA the following protocol is to be used:
Embedded in a regular UDP packet the data field contains again a header followed by the actual data. The IP address needs to be the address of the FPGA (list to follow) and the port is 0xffff = 65535. After the end of the UDP header, ie beginning with the first byte after the UDP checksum follow:

Header:

2 bytes packet data-length (excluding length, packet number & datatype fields)
2.5 bytes packet number
1.5 bytes datatype

Data:

data (length dependent on datatype, theoretical maximum is 65535 bytes)

Datatype can have the following values:

0x008: general commands
0x009: input delays
0x00A: control interfaces to other ATLAS systems
0x00B: extended event ID parameters
0x00C: general commands
0x00D: Acknowledge packet

The data looks like:

<u>Datatype 0x008:</u>

length: 1 byte
logic: positive (1 = true, 0 = false)
Bit 0: FPGA Reset
Bit 1: Start buffer dump (shouldn't be available in run mode for manual setting)
Bit 2: Abort buffer dump
Bit 3: Start S-Link transmission
Bit 4: Stop S-Link transmission
Bit 5: Pause S-Link transmission
Bit 6: Fill Buffers with internal pattern generator
Bit 7: Start ATLAS run for datataking

Datatype 0x009:

length: 9 bytes
logic: positive (1 = true, 0 = false)
Byte 0: 6 bit integer delay value ch 1
Byte 1: 6 bit integer delay value ch 2
Byte 2: 6 bit integer delay value ch 3
Byte 3: 6 bit integer delay value ch 4
Byte 4: 6 bit integer delay value ch 5
Byte 5: 6 bit integer delay value ch 6
Byte 6: 6 bit integer delay value ch 7
Byte 7: 6 bit integer delay value ch 8
Byte 8: Mask for delay values; mapping: Bit 0 = Byte 0, ...

Datatype 0x00A:

length: 8 byte
logic: positive (1 = true, 0 = false)
Byte 0: Number of bunches for S-Link read-out per L1A
Byte 1: ECR count load value (8 bit integer)
Byte 2: L1A count load value (24 bit integer)
Byte 3: L1A count load value (24 bit integer)
Byte 4: L1A count load value (24 bit integer)
Byte 5:
- Bit 0: Set CTP 9
- Bit 1: Don't Care
- Bit 2: Don't Care
- Bit 3: Don't Care
- Bit 4: Set Injection Permit
- Bit 5: Set Beam Permit
- Bit 6: Set DSS Warning
- Bit 7: Set DSS Abort

Byte 6: Set CTP 8 downto 1
Byte 7: Mask for commands, mapping:
- Bit 0: Number of Bunches
- Bit 1: ECR count
- Bit 2: L1A count
- Bit 3: DSS Abort
- Bit 4: DSS Warning
- Bit 5: Beam Permit
- Bit 6: Injection Permit
- Bit 7: CTP (all 9 bits)

Datatype 0x00B:

length: 9 byte
logic: positive (1 = true, 0 = false)
Byte 0: Run number
Byte 1: Run number
Byte 2: Run number
Byte 3: Run number
Byte 4: Detector event type
Byte 5: Detector event type
Byte 6: Detector event type
Byte 7: Detector event type

Byte 8: Mask, mapping:
- Bit 0: Run number
- Bit 1: Event type
- Bit 2-7: unused

Datatype 0x00C:

length: 6 byte
logic: positive (1 = true, 0 = false)
Byte 0: Source ID
Byte 1: Source ID
Byte 2: Source ID
Byte 3: Algo Select
Byte 4: RIO Input Mask
Byte 5: Mask, mapping:
- Bit 0: Source ID
- Bit 1: Algo Select
- Bit 2: Ethernet Packet Acknowledge
- Bit 3: Statistics Counter Reset
- Bit 4: Get Status Packet
- Bit 5: RIO Input Mask
- Bit 6-7: unused

Datatype 0x00D:

length: 4 byte
logic: positive (1 = true, 0 = false)
Byte 0: Acknowledge:
- 0x00: No packet received
- 0x0f: Last packet error
- 0xff: Last packet ok
- others: don't care/reserved

Byte 1: All Channels LV status
- Bit 0: Ch1
- Bit 1: Ch2
- ...

Byte 2: All Channels HV status
- Bit 0: Ch1
- Bit 1: Ch2
- ...

Byte 3: Reserved