Common Firmware Components

for the

CMS Level-1 Trigger

Version 2013.03.21

This document enumerates the firmware components that could be shared amongst the upgraded Level-1 electronic subsystems.

The scope, hardware platform constraints, and software requirements are described for each component.

# IPBus Suite

The IPBus Suite is an ensemble of firmware (VHDL) and software (C++ and Erlang) that implements the IPBus protocol on a Xillinx FPGA.

Software is already available (i.e. uHAL and ControlHub).

**Scope:**

Worldwide.

**Platform Constraints:**

The IPBus firmware is independent of the board type (i.e. MP7, CTP6/7, etc.).

The IPBus firmware depends on some Xilinx Intellectual Property (IP) cores. However, if needed, then the firmware could be adapted to other vendors.

The following FPGA families are currently supported: Virtex-5, Virtex-6, Virtex-7, Spartan-6, Kintex-7.

**Software requirements:**

The IPBus protocol can not be used if the board does not have an IP address assigned.

# Module Management Controller (MMC)

This is and ensemble of electronics, firmware, and software. The MMC provides the capability to control and monitor the board (i.e. temperature, status, power management, etc.). That is, the MMC implements the capabilities exposed by the Detector Control System (DCS) to the end-user.

Additionally, the MMC could be used for bootstrapping purposes (e.g. IP address assignment, FPGA firmware load and /or upload, etc.).

**Scope:**

Not clear.

**Platform constraints:**

The capabilities offered by the MMC, and the soft control interface on top of the IPMI protocol is not standardized. In addition, the soft control protocol depends on the IPMC controller implementation on the MicroTCA Carrier Hub (MCH).

Therefore, currently there are as many platforms (and interfaces) as combinations of MMCs and MCH.

**Software Requirements:**

Notice that if the MMC/MCH interfaces are not homogenized, then the DCS will have to accommodate the differences (or not implement them).

It is hardly possible to imagine a DCS without a common interface to control and monitor the boards. Therefore, the exposed IPMI interface to the MMC should be common for all CMS.

# Optical Receivers and Transmitters

**Scope:**

Level-1 Trigger

**Platform Constraints:**

**Software requirements:**

# Firmware Uploading and Loading

This encompasses an ensemble of electronics, firmware, and software.

This ensemble provides two capabilities: uploading of firmware images from a control PC to a non-volatile memory in the electronics devic, and the capability to load the previously uploaded images from the non-volatile medium to the FPGA (either on demand or after power cycling).

**Scope:**

Not clear.

**Platform Constraints:**

The firmware to upload and load firmware depends on the electronic interfaces. These interfaces are different for each board type.

**Software Requirements:**

At the very least, it should be possible to agree on the firmware upload capability from a high-level point of view (e.g. memory plus a register to trigger the image loading).

# Pattern Generation and Spy Memories

Memories for pattern generation and memories to spy on input and output channels are meant to assure the correct input-output function of the hardware. In other words, these memories allow the implementation of interconnection and pattern test.

A complete set of memories should allow the storage of input patterns for each device, and the retrieval of input and output buffers. This will allow the validation of the algorithm behavior and its latency.

**Scope**:

Level-1 Trigger.

**Platform constrains:**

The electronic and firmware implementation of the memories will depend on the board type.

**Software Requirements:**

The software will be simplified if the spy memories of the input and output channels, and the pattern generation ones have a common interface across the Level-1:

* *Spy trigger signal register*. The register tells the firmware which TTC signal should trigger the readout.
* *Arm Spy Memory*. After setting the register, the buffer will be filled starting from the time the *Spy trigger signal* is received from the TTC system.
* *Buffer.*
* *Buffer Size.*

# Reset

All the FPGA should have a reset register accessible from IPBus and IPMI.

**Scope**:

Level-1 Trigger.

**Platform constrains:**

The reset mechanism will entirely depend on the board type.

**Software Requirements:**

The software will be simplified if all the firmware offers the same address to reset the board through IPMI and IPBus.

# Firmware version, Configuration key, and Configuration Status

It should be trivial to reserve three words on the FPGA memory to contain the firmware version, the configuration status (e.g. halted, configured, etc.), and the configuration key.

The following capabilities can be implemented if these registers are available in the hardware:

* *Firmware version register* (read-only). This will allow to crosscheck the firmware version o the configuration data against the firmware version on the FPGA.
* *Configuration Key* *register* (read-write). In the absence of single-event upsets, this register tells the software not to reload the quasi-static data (e.g. Look-up Tables).
* *Configuration State register* (read-write). After a software failure, the state of the Finite State Machine (FSM) can be recovered from this register without requiring any reconfiguration of the system.