THE SLAC COMMON-PLATFORM FIRMWARE FOR HIGH-PERFORMANCE SYSTEMS

T. Straumann, R. Claus, M. D'Ewart, J. Frisch, G. Haller, R. Herbst, B. Hong, U. Legat, L. Ma, J. Olsen, B. Reese, L. Ruckman, L. Sapozhnikov, S. Smith, D. Van Winkle, E. Williams, J. Vasquez Stanescu, M. Weaver, C. Xu, A. Young, SLAC, Menlo Park, USA.

THMPL08

Introduction

LCLS-II requires several "High-Performance Systems" (HPS) which must process data for each individual beam-pulse at a rate of up to 1MHz.

This high rate demands that HPS functionality is implemented in FPGA logic rather than traditional software.

In an effort to standardize and share solutions as well as interfaces a "common platform" has been developed at SLAC. This platform consists of hardware, firmware and software components.

Hardware

The common platform builds on top of ATCA technology and employs some COTS components

- ATCA crate, shelf-manager, power-supplies, cooling, etc.
- 10Gb Ethernet switch blade
- Rack-mount linux PC server

as well as custom ATCA carriers and AMCs.

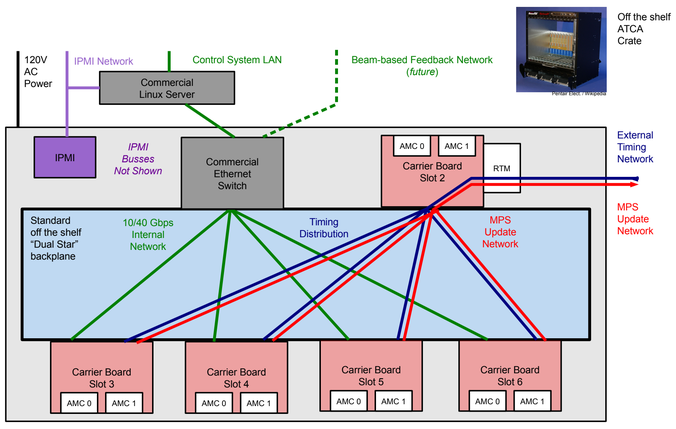


Fig. 1: Block Diagram of Complete ATCA System

The core component for HPS is a custom ATCA AMC carrier board which hosts a FPGA. Application-specific AMCs and/or RTM for I/O can be mounted.

The carrier has 10Gb-Ethernet connectivity to other blades and the linux server.

All software interaction with the carrier firmware is Ethernet based (NAD – "Network Access Device")

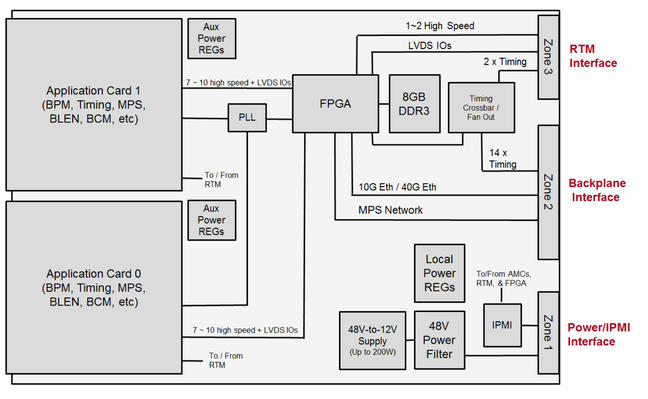


Fig. 2: Block Diagram of the Common AMC Carrier (with Application-specific AMCs)

Typical AMCs host high-speed ADCs, DACs and in some cases special-purpose front-end electronics. RTMs can provide GPIO, triggers, JTAG, debug/lab Ethernet etc.

Firmware

The SLAC common platform comes with a rich set of firmware libraries written in VHDL. At the core is the open-source SURF library which provides many basic building blocks, support for communication protocols and common hardware devices etc.

Other libraries implement basic HPS functionality which is required by many applications, e.g., machine-protection (MPS), beam-synchronous acquisition (BSA), timing, etc.

Specific support for the SLAC AMC Carrier integrates the common HPS support modules, an Ethernet communication protocol stack, hardware support, boot-loader and defines interfaces to the application core which is embedded into the firmware.

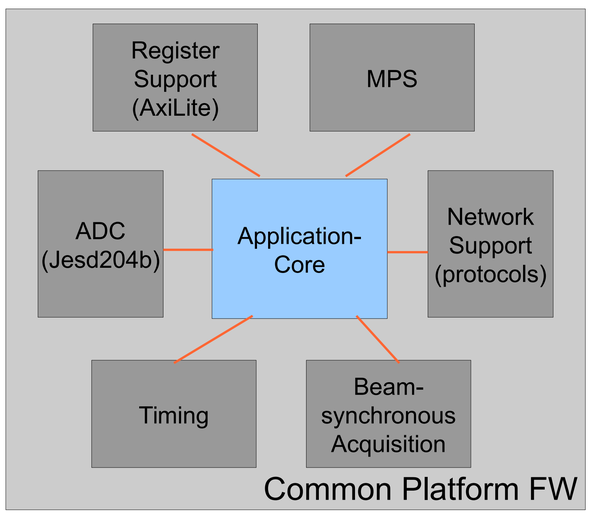


Fig. 3: Firmware Overview. Grey boxes are common-platform components.

Interconnect

While there are various proprietary interfaces inside the common firmware it also builds extensively on AXI-technology

- AXI-Lite
- AXI-Stream
- AXI-4

and many FW modules use AXI interfaces.

The SURF Library

Here are some elements of this open-source library (available on Github)

- Basic RTL: std_logic utils, CRC, 8b10b
- Fifos, dual-port RAM
- Clock-domain crossing
- AXI Support: AXI-Lite registers, AXI-Stream manipulation, AXI-Stream FIFOs, DMA, AXI-Lite crossbar
- Networking: 1GbE, 10GbE MACs, XAUI, IP (no routing or fragmentation), ICMP (limited), ARP, DHCP, UDP, RSSI (reliable UDP – cf. RFC908/1151)
- Hardware Devices: ADCs, EEPROMs, transceivers, …
- Serial: rs232, I2C, JESD204, SPI
- Wrappers for some Xilinx IP

Firmware (cont.)

Timing and BSA

Tight integration of the firmware with the timing system is essential for BSA ("beam-synchronous acquistion) support. E.g., beam-diagnostics (such as BPM) store timestamped readings into deep on-board BSA buffers (in RAM) under control of selection criteria which are also broadcast by the timing system. Globally synchronous readings can thus be obtained by multiple HPS distributed along the machine (downloaded by SW for off-line analysis)

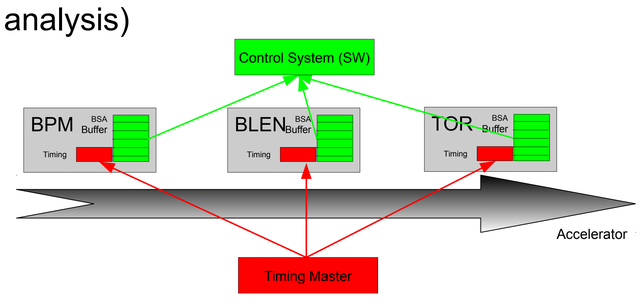
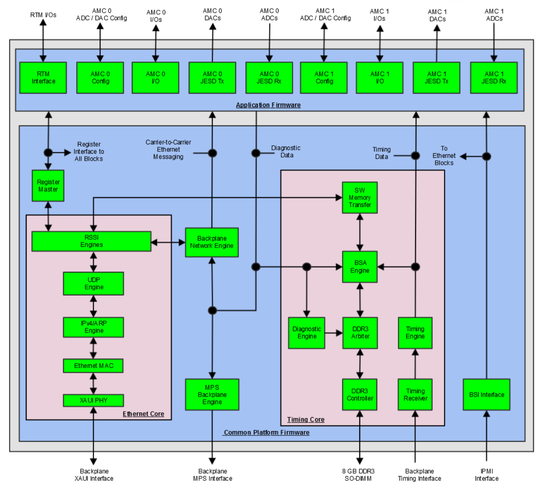


Fig. 4: BSA Overview. Timing System distributes timestamps and BSA selection criteria. HPS store time-stamped readings in real-time based on selection criteria. SW reads out later.

MPS

The machine-protection system is also tightly integrated. Applications post readings on an internal bus, MPS checks against thresholds and communicates with the blade in slot#2 ("concentrator") which escalates to a "master".

Firmware Block Diagram



Build System

The common-platform comes with the ***ruckus*** firmware build system which is heavily based on TCL scripting and integrates thus well with Vivado.

ruckus lets the user compile fpga designs in batch (or GUI) mode and is easy to customize or extend with "hooks" and other methods.

ruckus is open-source (available on Github).

Acknowledgment

This work was supported by U.S. Department of Energy Contract No. DE-AC02-76SF00515.