X2O ATCA IPMC and control solution: Update

D. Acosta, M. Bachtis, D. Campos, A. Greshilov, A. Jelisijevic, E. Juska, J. Konigsberg, A. Madorsky, V. Rekovic, A. Tan

2023-05-09
Hardware overview

- ATCA form factor
- Currently using VU13P-2 A2577 FPGA
  - Dual KU15P modules available
- Platform can be easily adapted to other FPGAs
- Supports 120 optical links up to 25 Gb/s. Boards fully loaded with optics support future extensions of the system and flexibility with SLR routing
  - Backward-compatible with cheaper 10Gbps QSFP modules
- Sufficient signal integrity in both the electrical and optical domains, BER much better than $10^{-12}$
- Optics provide sufficient optical margin with a receiver sensitivity better than -6 dBm to ensure operability at end of life (as laser degrades)
- System management through an on-board Linux system. Use the Xilinx Kria SOM in all the boards
- IPMC fully conforming to IPMI protocol, running on the same Kria SOM. No dedicated IPMC hardware.
- Connectivity required for TCDS2 is available
- Software and firmware framework for board management and subsystem development
The control system based on X2O platform will be applied for upcoming Phase-2 Upgrade of L1 Trigger subsystems:

- EMTF
- OMTF
- GEM
- GMT
- CSC (DAQ system upgrade)
X2O platform update

- X2O pilot production received
- Octopus FPGA module rev 2
  - Halogen-free
  - 1x VU13P
  - New TI clock synthesizer - tested by EP-ESE
  - Improved safety and interlock system with a lattice small FPGA
- Power module with Kria rev 3
  - SD3.0 (clock running at 200 MHz)
  - 10GbE (sustained speed about 5.14 Gbps)
  - DMA-JTAG chain (fast bitstream uploading to FPGA: 20 Mb/sec sustained).
  - IPMC running as an application on Kria
**KRIA for X2O**

- Power module rev 3
- Upgrade ZYNQ module to US+ family
- Selected Xilinx Kria K26 SoM as optimal candidate
  - Low cost: $300
  - 4 GB RAM
  - 4 GTH links 12.5 Gbps
- Faster AXI links to FPGA modules
  - 7.8125 Gbps
  - Max bit rate using CPLL
  - Both QPLLs in quad are needed for TCDS2
  - Rev 2: 3.75 Gbps max
- 10G Ethernet connection
  - In addition to 2x1G
X2O Power Module block schematics

To FPGA and Optical modules:
- High-speed connectors
- Service connectors
- Power connectors

Backplane:
- Gb Ethernet x2
- 10 Gb Ethernet
- IPMI I2C
- TCDS RX
- TCDS TX, DAQ

ZYNQ SoM:
- AXI
- I2C, JTAG
- Power emergency

LVDS fanout:
- Service +3.3V
- Payload +12V

DC-DC converters:
- Control + I2C
- LHC clock
- -48V
X2O Power Module ZYNQ firmware

- JTAG master x2
- Control & status registers
- Power emergency logic
- Watchdog
- CPU
- IPMI I2C slave x2
- Standard I2C master x5
- Chip2Chip master x2
# X2O Power Module firmware modules

<table>
<thead>
<tr>
<th>IP Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMI I2C slave</td>
<td>Customized I2C slave transceiver capable of buffering long transfers. Does not pose any timing requirements to IPMC software.</td>
</tr>
<tr>
<td>Standard I2C master</td>
<td>Standard I2C master modules used to control I2C devices on Power, FPGA, and Optical modules.</td>
</tr>
<tr>
<td>Chip2Chip master</td>
<td>Regular Chip2Chip master, used to extend AXI bus to FPGA modules</td>
</tr>
<tr>
<td>JTAG master</td>
<td>JTAG masters are used for firmware downloading and Xilinx Virtual Cable (XVC) debugging of the FPGA modules</td>
</tr>
<tr>
<td>Power emergency logic</td>
<td>Payload modules can signal emergency conditions (such as overvoltage) using dedicated lines. Emergency logic then shuts down Payload power in 10 ns or less</td>
</tr>
<tr>
<td>Watchdog</td>
<td>The dedicated watchdog hard IP is used to monitor the health of the IPMC software. If IPMC software stops polling the watchdog, the watchdog resets the CPU and shuts down the Payload power.</td>
</tr>
</tbody>
</table>
- Custom I2C receivers (slave) modules
- Operate independently of CPU
- No real-time software handling needed
UF IPMC SW

Historical reference: UF IPMC project started in 2020 for ZYNQ 7000 (32-bit ARM)

UF IPMC² is an Intelligent Platform Management Controller (IPMC) that resides on ATCA boards (X2O) in an embedded ZYNQ³ device (KRIA). UF IPMC is responsible for the communication with Shelf Manager.

Built 64-bit ARM Embedded Linux system on KRIA module within X2O boards:
- Petalinux kernel version 2022.2
- CentOS 8

Main points:
- Based on the IPMI & PICMG cores of the coreIPM open-source project⁴.
- Provides full basic functionality⁵ for sensor monitoring.
- Provides easy customization of USER sensors
  - UCLA team has successfully implemented and tested custom Octopus board sensors (rev.2).
- Supports the use of various sensor readout interfaces, including I2C

Current status:
- (done) UF IPMC sw version for VU13P with QSFP-DD (rev.2)
- (in progress) UF IPMC sw version for VU13P with QSFP 30-cage module (rev.3).

NOTE: No separate IPMC hardware module is needed.

Normal CPU usage of UF IPMC process is 2-3%

ATCA shelf manager recognizing some of our temperature sensors (rev.2)
➢ TEMPLATES are provided
➢ UF IPMC Manual is provided
➢ Separate USER code space (for easy update)

Sensor enable

Upper non-recoverable threshold exceeded (Power OFF)

Upper critical threshold exceeded (Power OFF)

Upper non-critical threshold exceeded (Fans speed - UP)

Back to normal (Fans speed - DOWN)

Sensor disable
reaction on power emergency

- quick payload shutdown required in case of power emergency, specifically overvoltage.

- dedicated power emergency signal is used for shutting down the payload power via firmware logic
  - no cpu participation.
  - reaction time as fast as a few nanoseconds.

- ipmc eventually lets the shelf manager know that a failure took place, but there is no software timing limitation.

- shutting down the payload power as a result of the sensor readout via i2c (or another interface) is also possible.
  - reaction time much longer than power emergency logic in firmware.
  - used for thermal shutdown.
UF IPMC implements all functionality necessary for normal operation of the device in the ATCA chassis. It was tested in three different ways:

1. In the CMS-standard ATCA chassis
2. In the COMTEL ATCA chassis
3. On the Polaris Compliance test stand in CERN

All errors detected by Polaris Compliance test stand are expected due to either implementation features, or non-critical functionality not currently implemented in the UF IPMC.

No unexplained errors have been detected. None of the detected errors preclude UF IPMC from normal operation in ATCA chassis.

Thanks a lot to CERN EP-ESE team for the assistance with compliance tests!
Control solution

High-level design (proposed by phase-2 online SW group* (K. Lannon, T. Williams, A. Akpınar, D. Gastler, R. Knowlton, A. Mitra, D. Monk, J. Sweet):

1. HERD
   - On-board application providing run control & monitoring interface (1 per board)
   - Builds on SWATCH library from phase-1 L1T

2. SHEP (short for shepherd)
   - Off-board application — each instance supervises a single subsystem

* [https://indico.cern.ch/event/1099319/contributions/4625725/attachments/2359823/4027959/L1TPhase2OnlineSoftware_20211206.pdf](https://indico.cern.ch/event/1099319/contributions/4625725/attachments/2359823/4027959/L1TPhase2OnlineSoftware_20211206.pdf) (see from slide 13)
X2O SWATCH

- X2O/ Phase-2/SWATCH:
  - Installed and tested SHEP.
  - Installed and tested initial version of HERD plugin:
    - (done) MGT configurator (tested for VU13P).
    - (done) DMA-proxy driver for bitstream uploading to FPGA via DMA-JTAG chain (tested for KU15P, VU13P).
    - (done) FPGA programming from HOST (tested for KU15P, VU13P).
    - (done) Added utilities: I2C tool, devmem tool, semaphores, configs parsers.
    - (done) Added clock(sync & async) synthesizer configuration (tested for KU15P, VU13P).
  - (in progress) X2O HERD plugin for Octopus (VU13P module).
FPGA MGT builder is a set of software tools with the following functionality:

- Automatic generation of firmware structure with support for arbitrary MGT configurations. This includes:
  - Different bit rates and encodings in RX and TX parts of the same MGT
  - Using CPLL and QPLL as needed for each MGT, programmable separately for RX and TX parts (within the constraints of the particular FPGA architecture)
  - Automatic routing and assignment of available reference clocks for each MGT
  - Grouping MGTs into interfaces with programmable names and indexes, which makes using them in the top-level design much easier
  - Automatic generation of all constraints related to MGTs
    - Reference clock location, timing, and grouping
    - MGT location
    - User clock timing
  - The software is designed to make porting a project to different board design or FPGA an easy task.
  - Targeting lowest-latency serial links, by disabling RX and TX buffers

- Software framework:
  - Reads configuration settings and programs all DRP registers and port settings in each MGT and COMMON modules
  - Customizable reset procedures, with main functionality provided in the example design
  - Written in portable C++, can be adapted for nearly any system, including embedded processors
  - Does not need rework if the MGT configuration is changed

- Configuration sources:
  - All source configuration files are kept in Excel XLSX format
    - Makes working with them much easier
  - Data format is optimized for direct copying from Xilinx manuals and example source code, with minimal manual rework
  - A Python script is provided for exporting configuration files into plain text
References

1. X20 firmware project repository: https://github.com/madorskya/apex
2. UF IPMC github repository: https://gitlab.cern.ch/x2o/UF_IPMC
4. coreIPM open-source project: http://www.coreipm.com/
6. X2O HERD plugin: https://gitlab.cern.ch/cms-cactus/phase2/software/plugins/x2o
7. FPGA MGT Builder repository: https://github.com/madorskya/mgt_builder
Backup
UF IPMC Polaris Compliance test results

UF IPMC tasks:

- Mandatory:
  - 58 (passed)
  - 17 (failed) – not critical

- Optional:
  - 18 (HPM.x) - not implemented

- Debug:
  - 30 (skipped) - not important

ELMA IPMC tasks:

- Mandatory:
  - 42 (passed)
  - 31 (failed)

- Optional:
  - 18 (HPM.x) - not implemented

- Debug:
  - 32 (skipped)
All required basic functionality for correct IPMC operation is implemented and tested. The following test have passed:

- IPMC state transition commands (M1, M2, M3, M4, M5, M6)
- Monitoring “Criteria Met” conditions within insertion/extraction procedures
- FRU info commands (mandatory header, can be fully completed depending on the Hardware Platform used)
- SDR commands (implemented with Human-Readable .toml format files as Full Sensor Record Type 01h that could be changed at any time without recompilation of project)
- FRU Hot Swap sensor
- IPMB-0 state sensor
- FRU Handle Switch sensor
- Dummy custom sensors (USERS can easily implement their own custom sensors)
- Sensor monitoring functionality
- Power Management commands
- Event Generation functionality:
  - Hot Swap Event messages within IPMC state transitions
  - Hot Swap Event messages within Abnormal Operation Stage
  - Hot Swap Event messages within IPMB-0 state monitoring
- Power faults handling functionality
- Fans Speed Up reaction to the exceeded thresholds within Temperature sensor implementation (USER defined functionality)
- LED commands (commands are present, but not used. In the current implementation on X2O platform doesn’t need to have this functionality. If needed USERs can add it as required)
- Reset functionality (commands are present, but not used. Currently not required in X2O platform implementation).
<table>
<thead>
<tr>
<th>Task count</th>
<th>Description</th>
<th>Reason for error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Get SEL command</td>
<td>Not implemented: Not required because UF IPMC is using regular log files</td>
</tr>
<tr>
<td>1</td>
<td>Max FRU Device ID in the Get PICMG Properties response is 0</td>
<td>It’s sufficient to use FRU Device ID=0</td>
</tr>
<tr>
<td>2</td>
<td>Stopping at intermediate IPMC states</td>
<td>Not allowed in UF IPMC</td>
</tr>
<tr>
<td>3</td>
<td>Read FRU Data failed – 8 bytes instead of 15</td>
<td>Expected. Only mandatory header (8 bytes) is implemented in UF IPMC</td>
</tr>
<tr>
<td>2</td>
<td>Multirecord Info Area is not present in FRU Information</td>
<td>Not implemented</td>
</tr>
<tr>
<td>1</td>
<td>Product Info Area is not present in FRU Information</td>
<td>Not implemented</td>
</tr>
<tr>
<td>6</td>
<td>Temperature Event Messages fail: USER defined fields</td>
<td>UF IPMC does not allow changing Temperature sensor records via Shelf Manager commands. They should only be modified by updating corresponding SDR configuration files.</td>
</tr>
<tr>
<td>1</td>
<td>Watchdog Timer Commands Support failed</td>
<td>Not implemented  (Fixed by using ZYNQ built-in watchdog system timer)</td>
</tr>
</tbody>
</table>