

HIGH-PERFORMANCE COMPUTE BOARD – A FAULT-TOLERANT MODULE FOR ON-BOARD VISION PROCESSING

Joaquín España Navarro⁽¹⁾, Arne Samuelsson⁽¹⁾, Henrik Gingsjö⁽¹⁾, Julius Barendt⁽¹⁾, Aubrey Dunne⁽²⁾, Léonie Buckley⁽²⁾, Dionysios Reisis⁽³⁾, Angelos Kyriakos⁽³⁾, Elissaios Alexios Papatheofanous⁽³⁾, Charalampos Bezaitis⁽³⁾, Peter Matthijs⁽⁴⁾, Juan Pablo Ramos⁽⁴⁾, David Steenari⁽⁵⁾

⁽¹⁾ Cobham Gaisler AB, 41119 Gothenburg, Sweden

⁽²⁾ Ubotica Technologies Limited, D11KXN4 Dublin, Ireland

⁽³⁾ National and Kapodistrian University of Athens, 15772 Athens, Greece

⁽⁴⁾ QinetiQ Space NV, 9150 Krubek, Belgium

⁽⁵⁾ European Space Agency, ESTEC, 2201 AZ Noordwijk, The Netherlands

ABSTRACT

This technical paper describes the High-Performance Compute Board (HPCB), currently being implemented and tested by a consortium led by Cobham Gaisler in the frame of an ESA project. The first section serves as a brief introduction to the platform, whereas subsequent sections add further detail concerning the architecture, hardware, and software design. Finally, some preliminary test results are presented before summarizing the most relevant aspects of the paper in the conclusions.

1. INTRODUCTION

Cobham Gaisler, in collaboration with Ubotica Technologies, the National and Kapodistrian University of Athens, and QinetiQ Space, is developing a High-Performance Compute Board (HPCB) within the European Space Agency TDE activity “FPGA Accelerated DSP Payload Data Processor Board”.

Compared to existing technology, the HPCB platform will provide more computational resources onboard spacecrafts to process high bit-rate payload data before downlink, thus reducing bandwidth requirements and improving reaction times of space systems.

The target applications include on-board payload processing for optical and radar instruments, as well as visual navigation. The board can be integrated in payload data handling units, mass-memory and/or on-board computer to enable functions such as high-performance on-board image processing, machine vision and standard CCSDS 123.0 image compression. The board design is optimized for the data handling and processing of multiple instruments simultaneously.

The architecture combines up to three Vision Processing Units (VPUs, Intel Movidius Myriad 2), a high-capacity

FPGA (Xilinx Kintex Ultrascale XCKU060 [1], with a potential upgrade to XQRKU060 [2] for the flight version), and a radiation-tolerant microcontroller (Cobham Gaisler GR716 [3]), in order to create a reliable system solution for space applications that matches what is available in the commercial domain in terms of performance and functionality. The use of three VPUs and the multiple cores within each VPU leads to the high performance needed for image/application processing.

The next sections provide further detail into the platform. Firstly, the hardware architecture is presented, followed by a description of the FPGA VHDL design, the microcontroller, and the VPU software design.

2. HARDWARE ARCHITECTURE

The architecture targets a 6U by 160 mm Payload Module implemented according to the OpenVPX standard (VITA 65). The HPCB platform consists of a carrier board, henceforth referred to as the main board, and the mezzanine cards [4]. The main board contains the microcontroller and the FPGA and supports the use of up to three VITA 57.1 FMC mezzanine cards. Each mezzanine contains a VPU and its associated power and memory components. The VPUs can be operated in parallel for maximum performance or redundant configurations. The complete block diagram of the HPCB architecture is depicted in Fig. 1.

The main interfaces of the platform are four SpaceFibre and two SpaceWire links, available on the front panel. The SpaceFibre links nominally operate at 3.125 Gbps (up to 6.25 Gbps) and are routed to the high-speed transceivers of the FPGA, whereas the SpaceWire links run at 100 Mbps and are routed to both FPGA and microcontroller via a cross-point-switch.

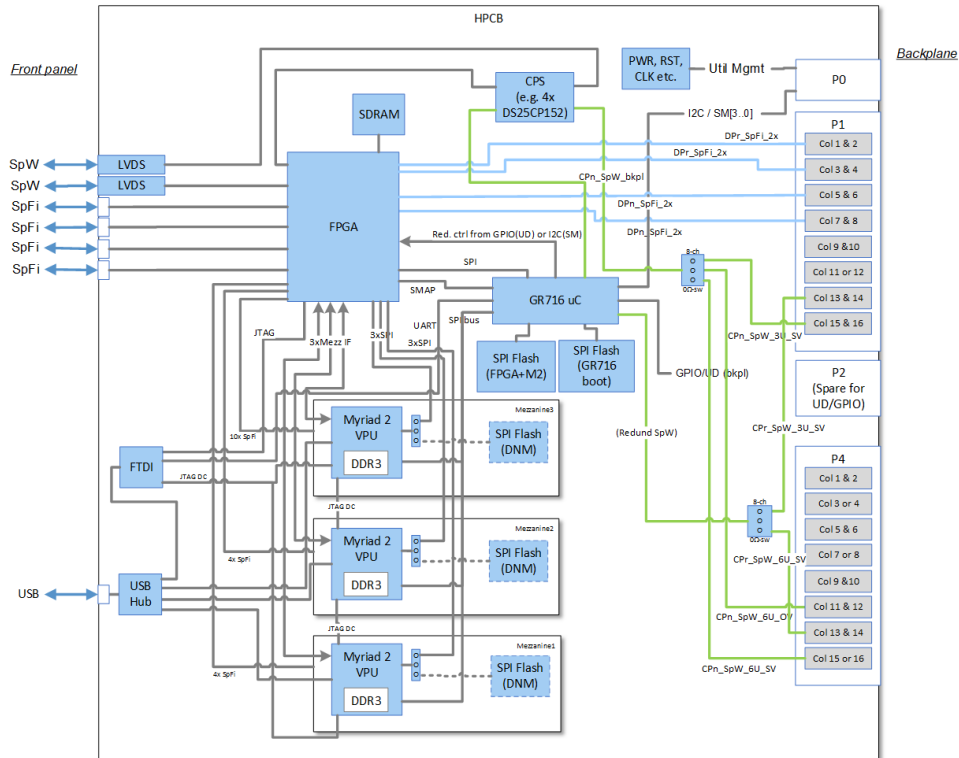


Figure 1. HPCB architecture block diagram

The backplane interfaces are adapted to the OpenVPX standard and include SpaceWire for control and SpaceFibre for data. The board can also be operated stand-alone, equipped with debug links such as USB and JTAG intended for lab usage.

The main board contains a Xilinx XCKU060 FPGA. It acts as a flow manager, delivering data between the Front-End Equipment (via the front-panel links), the VPUs and the OpenVPX System Controller (via the backplane). It is also in charge of booting and monitoring the VPUs, restarting them if necessary.

The FPGA working memory consists of two DDR3 SDRAM memory modules. The FPGA divides the working memory space into several partitions of configurable size, each of which is used for buffering a specific type of application data – raw images, processed results, VPU boot software and VPU configuration.

The HPCB platform includes the Cobham Gaisler GR716 microcontroller, present also on the carrier board. As the rad-tolerant component of the platform, its role is to supervise the overall operation of HPCB by monitoring the status registers of the FPGA and the VPU heartbeats.

The current version of the board includes the first revision of the GR716 microcontroller: GR716A. A pin-compatible second revision of the microcontroller containing a scrubber IP, GR716B, is currently being implemented. Once available, it will replace GR716A on the HPCB platform in order to scrub the FPGA configuration area via the SelectMAP interface.

The microcontroller has access to two SPI Flash memories containing its boot software, the FPGA configuration bitfiles and the VPU boot images. Once booted, the GR716 automatically programs the FPGA via the SelectMAP interface (GR716B only) and sends the VPU boot images to the FPGA SDRAM memory. Alternatively, if the GR716A is used, an additional SPI flash memory with the bitfiles is directly connected to the FPGA.

The third major element of the platform is the Intel Movidius Myriad 2 VPU, present on the FMC cards together with its power and memory components. Up to three Myriad 2 VPUs can operate in parallel, thus enabling redundant configurations or improving the processing capabilities of the system. Each VPU has a dedicated 4Gb DDR3 acting as its working memory.

The interfaces between the GR716 and the FPGA consist of a SPI link, used to access the internal registers of the FPGA, a general-purpose I²C link and the

SelectMAP interface. Additional GPIOs are used to forward important information from the FMC cards to the GR716, such as the Myriad 2 heartbeats.

The main interfaces between the FPGA and each VPU consist of a SPI link for command and control; Camera Interface (CIF) for sending images and configuration data; Liquid Cristal Display (LCD) for receiving the processed data and I²C for current monitoring. Additional GPIOs are available, as required by the VITA 57.1 standard for FMC cards.

3. FPGA VHDL DESIGN

The demo application developed in this activity demonstrates the use of the interfaces available in the HPCB platform. The primary role of the FPGA is to control the external interfaces, act as a flow manager distributing data among the external subsystems and control the operation of the VPUs.

The block diagram of the VHDL design can be seen in Fig. 2. For the sake of simplicity, only one Myriad 2 interface is depicted.

The design is based on the IP cores available in the ESA IP catalogue, namely LEON2FT package, SpaceWire codec with RMAP functionality, SpaceFibre codec and ShyLoC compressor. Additional IP cores have been developed in the frame of this activity to address the missing functionality, such as the central engine and the CIF/LCD interface.

Most IPs are connected to a central AMBA bus. The AHB controller IP acts as the arbiter of the bus, receiving bus requests from the AHB masters and granting access accordingly. A separate APB bus for peripherals is connected to the central bus via an AHB to APB bridge, also referred to as the APB controller. This IP behaves as an AHB slave in the central AMBA bus and as the only master in the APB bus. When the APB registers are accessed from any AHB master, the APB controller forwards it to the corresponding APB slave so that the AMBA transaction can complete.

The design contains several IPs that provide visibility over the entire AMBA address space to external units. The SpaceWire IP used in the design implements RMAP targets and can be connected to Front-End Equipment via the SpaceWire front-panel links. The SPI slave controller (SPI2AHB, from the Gaisler IP core library, GRLIB) handles the SPI link with the GR716 microcontroller. Finally, a JTAG controller (AHBJTAG, from GRLIB) can be used for debugging purposes.

Application data is primary transmitted over SpaceWire and SpaceFibre. It shall be encapsulated in CCSDS packets as per the ECSS-E-ST-50-53C standard. For SpaceWire, non-RMAP packets are verified and, if the format is correct, forwarded to the Protocol / Data Engine (the HPCB central controller IP, explained later in this section), otherwise the packets are discarded.

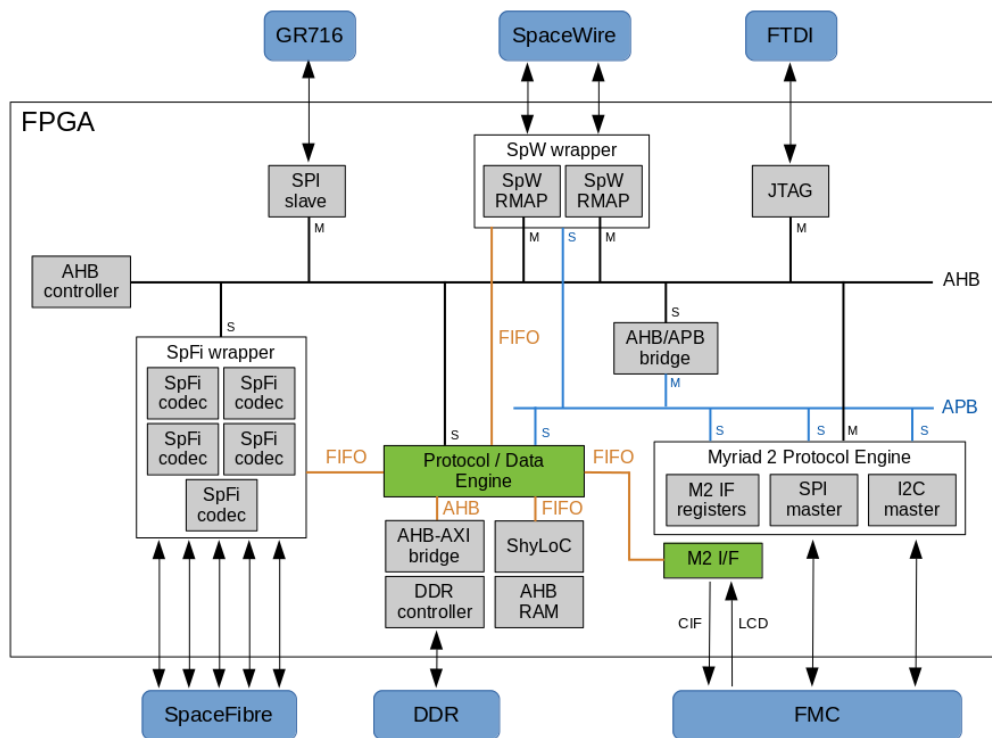


Figure 2. VHDL design block diagram

There are 5 SpaceFibre IPs instantiated in the design: 4 of them connected to the front-panel links to interface Front-End Equipment, and another one connected to the backplane, interfacing the System Controller. Each IP implements 2 virtual channels. The virtual channel 0 is used for control: the format of the packets shall be RMAP and the FPGA acts as a packet router between the System Controller (backplane) and the Front-End Equipment (front-panel), based on the destination address of the packet. On the other hand, the virtual channel 1 is used for application data and shall be encapsulated in CCSDS packets. Depending on the source generating the packets, the application data may contain input raw images to process (input packets received from the front-panel links), VPU configuration data (input packets received via the backplane) and the VPU processed data (output packets sent over the backplane). The different SpaceWire and SpaceFibre data paths are depicted in Fig. 3.

The IP controlling the overall functionality of the platform is the Protocol / Data Engine, henceforth referred to as PDE, which has been implemented specifically for this activity. Its two main goals are to control the VPUs and to distribute data between the Front-End Equipment, the System Controller and the Myriad 2 VPUs. The PDE instantiates an independent AMBA bus, also known as the memory bus, to manage the content of the FPGA working memory (DDR3 SDRAM, 4 GiB in the demo application). This bus has a single master, the PDE, and a single slave, the memory controller, therefore no arbiter is required and both IPs are connected directly.

The SDRAM content is solely handled by an FSM in the PDE. When idle, the PDE will wait for requests to access the SDRAM from the different IPs of the design. Examples of this are the raw images received over SpaceWire or SpaceFibre, the processed results returned from the Myriad and the VPU configuration data received from the System Controller. The PDE assigns priorities based on the criticality of the data.

Additionally, the PDE implements a separate FSM per Myriad to control its operation. Firstly, once the FPGA has received the VPU boot images from the GR716 via SPI, the PDE proceeds to complete the primary and secondary boot of the Myriads via SPI and CIF, respectively. This 2-step process is further explained later in this document. Once this process finalizes, the PDE will wait until new configuration data is received from the System Controller, which in turn is sent to the Myriad via CIF. Once the VPU is ready to process data, the FPGA will forward raw images to the VPU via CIF and retrieve the results via the LCD interface. These results are later encapsulated in CCSDS packets and transferred to the System Controller via SpaceFibre.

The PDE supports 4 modes of operation. An IDLE mode, where the Myriads do not perform any calculation. In SINGLE mode, each Myriad processes a different image, thus maximizing the throughput of the system. Lastly, DMR and TMR are redundant modes in which the FPGA sends the same frame to 2 or 3 Myriads, respectively. The results are then compared: in DMR, the results from both Myriads shall match, whereas in TMR voting takes place. The VPUs yielding anomalous results are automatically reset and reprogrammed by the FPGA.

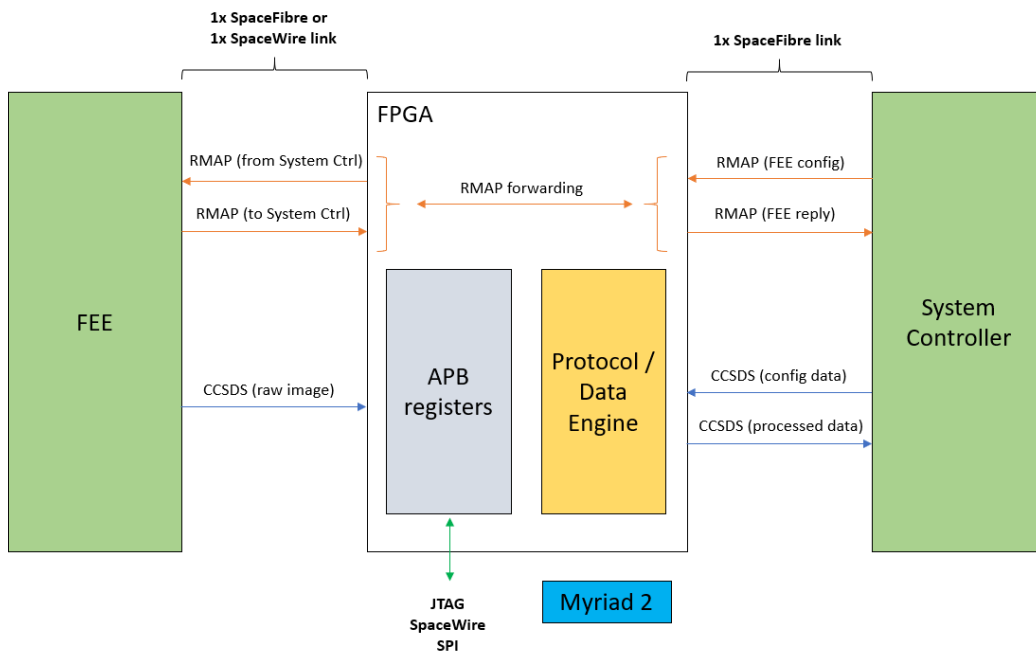


Figure 3. FPGA control and data paths

The IP controlling the CIF and LCD interfaces has also been developed during this activity. Its main purpose is to receive the binary data to be transmitted to the Myriad from the PDE and encapsulate it in CIF frames. Likewise, it depacketizes the LCD frames returned by the Myriad and forwards the binary data to the PDE. The IP features clock-domain-crossing mechanisms, so that the CIF and LCD codecs operate in CIF/LCD clock domains, respectively, while the interface towards the PDE employs the AMBA clock.

Additional functionality best suited for hardware will be implemented in the final stage of the FPGA design. Two examples are data reduction and temporal binning. For the former, the ShyLoC compressor from the ESA IP catalogue will be used in order to apply CCSDS 123.0 compression to the processed results. Temporal binning will be included as part of the PDE IP.

4. MICROCONTROLLER SOFTWARE

The software that will run on the GR716 (the supervisor) is located in a SPI Flash memory directly connected to the microcontroller. The supervisor will be loaded into GR716 work memory and run by the built-in GR716 bootloader. After booting the system, the supervisor software will transfer the boot images for the Myriad devices from the connected SPI flash to the FPGA working memory (DDR3 SDRAM) via SPI. When the transfer is complete the supervisor will issue a command to the FPGA to boot the VPUs. The supervisor will then poll the FPGA until it reports that the VPUs are booted correctly and will subsequently enter idle mode, only to wake up in set intervals to check the status of the system as well as the VPU heartbeats. The FPGA receives the heartbeats via the FMC connector and forwards them to the microcontroller via GPIO pins.

If errors are detected or if a VPU heartbeat is missing, the microcontroller will identify the faulty VPU and reset it before entering idle mode again.

Once the second revision of the microcontroller is available, it will replace the first revision in the HPCB platform. The second revision includes a scrubber IP that allows both to program the FPGA and scrub its configuration area. The golden copy of the FPGA bitfiles and mask are located in a SPI flash connected to the microcontroller. Both programming and scrubbing are performed via the Xilinx SelectMAP interface.

In anticipation of the scrubbing functionality, the carrier board includes a SPI flash directly connected to the FPGA to allow self-booting via the serial interface.

5. VPU SOFTWARE

The role of the Myriad VPUs on HPCB is to perform Image Signal Processing (ISP), Computer Vision (CV) and AI inference on input data frames received from the FEE via the FPGA, and return the processed results back to the FPGA, where they are stored and made available to the System Controller. Myriad 2 is particularly suited to this application for several reasons: (1) it is designed from the ground up for high throughput image processing at low power; (2) it contains dedicated hardware blocks for implementing common ISP and CV tasks, which are extremely power efficient; and (3) it contains 12 SHAVE VLIW vector engines for SIMD manipulation of data stored in a directly interfaced, highly multi-ported, random access CMX memory [5]. Layered on top of these characteristics are two optimised firmware modules that interact directly with the HW blocks and SHAVE engines: SIPP and Fathom. SIPP is a framework that manages the just-in-time row-wise processing of image data through heterogenous hardware-software pipelines in a streaming fashion, thus maximising pipelining across the filters, and is capable of achieving full streaming rates of up to 500Mpix/s. Fathom manages AI inference on the SHAVES, controlling weights and data DMAs to and from CMX, and ensuring partitioning across the SHAVES.

These features provide excellent building blocks for a high-performance vision processing platform for space application. Nevertheless, further enhancements have been built into the HPCB firmware to manage the above resources and modules in a manner that is cognisant of the target domain while being application agnostic and decoupled from firmware development. The HPCB deployment uses the CVAI Toolkit from Ubotica, which enables ISP and CV pipelines to be developed in a drag-and-drop configuration tool GUI. Via the CVAI Toolkit the user has the ability to design and compile image processing pipelines into a Pipeline Configuration Descriptor (PCD) file, which can then be deployed on HPCB, where they are interpreted in Myriad firmware via the CVAI runtime engine that interprets these files and dynamically builds the pipelines. The toolkit also facilitates the deployment of AI models in consort with the Intel OpenVINO toolkit [6]. OpenVINO enables the optimisation and targeting of a vast range of AI models to Myriad, integrating with the most common NN development frameworks (Keras, TensorFlow, PyTorch, MXNet). The HPCB architecture facilitates the System Controller to load the NN blobs and PCDs onto Myriad via System Controller commands to the FPGA. At runtime, the System Controller dynamically specifies the pipeline and/or blob to activate, along with the input and output frame parameters (dimensions, bit depths). All subsequent frames that are transferred from FPGA to Myriad are processed through the specified

pipeline/blob and returned to the FPGA. This processing flow is outlined in Fig. 4.

HPCB firmware on Myriad 2 implements the Finite State Machine (FSM) shown in Fig. 5. A two-stage boot process enables a primary boot over SPI from the FPGA using a relatively small boot image, followed by secondary boot of the full firmware image over the significantly higher bandwidth CIF interface. Once the secondary boot is complete, a Built In Self Test (BIST) assess the main memories and caches of the system, with results available to the FPGA via SPI queries. The main operating stage of the FSM implements a server paradigm wherein configurations and processing requests are received from the FPGA, and the Myriad processes the requests according to the current configuration. Prior to any processing requests, the FPGA transfers the necessary pipeline PCD and NN blob files, which are orchestrated on Myriad's on-device DDR via a custom memory manager. Multiple PCDs and blobs can be resident on device concurrently, with the pipeline and network to run configured by the System Controller. This provides the flexibility to compile multiple distinct pipelines into a single PCD, with per-frame switching between pipelines and networks possible, for example, for the processing of frames from two independent sensors.

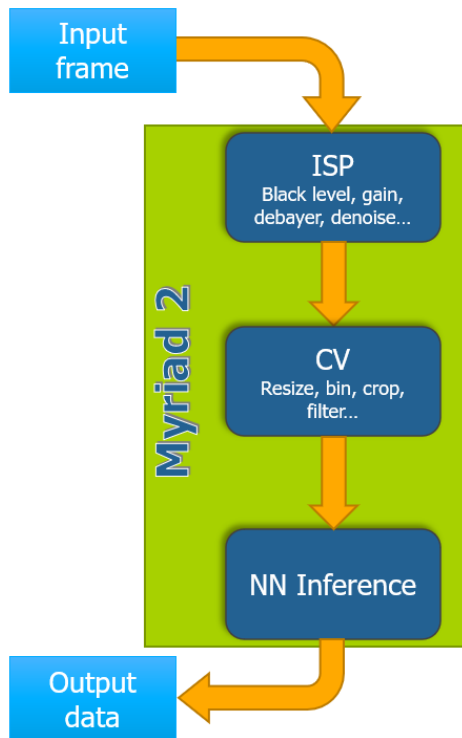


Figure 4. Frame processing flow

HPCB Myriad 2 FSM

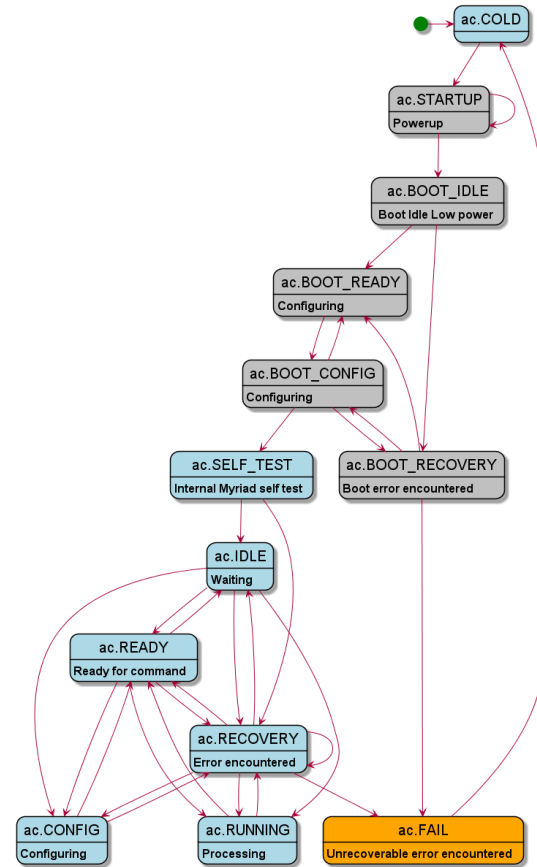


Figure 5. Myriad 2 firmware FSM

A sequential frame-level processing paradigm is used between the FPGA and the Myriads since: (1) such a scheme reduces the buffering required on Myriad, thereby better ensuring that the in-package DDR is sufficient to enable storage of frames, PCDs and NN blobs; (2) a sequential processing scheme enables a more straightforward, and therefore more reliable, architecture for the FSM; and (3) it enables a more straightforward, and therefore reliable, SW control interface between FPGA and Myriad 2. Nevertheless, three levels of data parallelism are achieved on HPCB when processing using the Myriad 2s. Firstly, frame-level parallelisation is achieved via the use of multiple Mezzanines on the main carrier board. Secondly, the Myriad processes a single frame in a pipelined fashion, and where hardware acceleration blocks are included in a processing pipeline, these blocks execute in a pipelined-parallel fashion. The SIPP architecture ensures that HW blocks are fed row data in lock-step with their processing. SW blocks execute in parallel across all assigned SHAVES, with image rows partitioned across SHAVES. Thirdly, the individual HW and SW blocks within a pipeline parallel-process blocks of contiguous pixels. For HW blocks this is directly implemented in the HW. For SW blocks, the SHAVES

process pixel data with their vector instruction set (where possible).

Users of HPCB can thus design and build vision processing pipelines, and architect, train and compile NN models, prior to deployment across HPCB's Myriads without any requirement of embedded code development or firmware compilation. Furthermore, new models and pipelines can be uploaded and applied during flight at runtime without interruption to the HPCB system operation. The ability to perform tightly coupled and task-flexible image pre-processing followed by NN inference, all in a single flow on Myriad and without data going off-device, provides a performant and flexible vision processing solution.

6. PRELIMINARY TEST RESULTS

The design is currently being finalized and validated. This section contains some preliminary figures about the performance of the system.

The VHDL design operates with an AMBA clock of 50 MHz. This clock is generated by the DDR3 memory controller. The SpaceWire links run at 100 Mbps, whereas the SpaceFibre links support both 3.125 Gbps (default) and 2.5 Gbps. The CIF and LCD interfaces currently operate at 50 MHz with a bit depth configurable by the user of either 8 or 16 bits per pixel.

The utilization figures of the VHDL design are shown in Fig. 6, excluding the hardware accelerators and the voting functionality. In the light of the results, it is deemed possible to triplicate the design in the FPGA in order to improve the reliability of the platform. This might have an impact on the timing performance, so the maximum achievable frequency in a TMR system should be assessed.

Resource	Utilization	Available	Utilization %
LUT	76026	331680	22.92
LUTRAM	2952	146880	2.01
FF	64298	663360	9.69
BRAM	77	1080	7.13
DSP	25	2760	0.91
IO	438	624	70.19
GT	5	32	15.63
BUFG	33	624	5.29
MMCM	5	12	41.67
PLL	3	24	12.50

Figure 6. FPGA resource utilization

While benchmarking the platform the maximum throughput of the platform will be evaluated. This includes an assessment of the maximum AMBA clock and CIF/LCD frequencies, as well as an increase of the SpaceWire and SpaceFibre bitrates to 200 Mbps and 6.25 Gbps, respectively. Fault-injection in the VPUs will be performed to verify the functionality in

redundant modes where multiple Myriads compute the same raw data.

7. CONCLUSIONS

This technical paper has presented the High-Performance Compute Board. The HPCB platform allows to simultaneously interface multiple instruments operating at high bitrates. AI techniques from the commercial domain have been introduced in space by using COTS parts, which are combined with system-level mitigation techniques to create a reliable platform with very high processing capabilities.

The demo application developed during this activity demonstrates the use of the platform and its multiple interfaces. The design is easily scalable depending on the target mission. The main components of the board – the microcontroller, the FPGA and the VPUs – can be reprogrammed to be tailored to the needs of the activity. New pipelines and neural networks can be easily deployed without further code development in the VPU firmware.

The design is currently being finalized and the verification campaign ongoing. The full validation is expected to conclude by the end of Q3 2021.

REFERENCES

- [1] Kintex Ultrascale family of FPGAs, Xilinx: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale.html> [Accessed 12/6/21]
- [2] Rad-tolerant Kintex Ultrascale family, Xilinx: <https://www.xilinx.com/products/silicon-devices/fpga/rt-kintex-ultrascale.html> [Accessed 12/6/21]
- [3] Cobham Gaisler GR716 microcontroller: <https://www.gaisler.com/index.php/products/components/gr716> [Accessed 12/6/21]
- [4] HPCB carrier and mezzanine cards datasheets: <https://www.gaisler.com/index.php/products/boards/gr-vpx-xcku060> [Accessed 12/6/21]
- [5] B. Barry et al., "Always-on Vision Processing Unit for Mobile Applications," in IEEE Micro, vol. 35, no. 2, pp. 56-66, Mar.-Apr. 2015
- [6] OpenVINO Toolkit, Intel Corporation, <https://docs.openvino toolkit.org> [Accessed 10/6/21]