HPC PCIe FPGA Clusters with Raspberry Pis

Andrew E Wilson

***Abstract*—A scalable, affordable option for HPC FPGA development. This platform allows for custom PCIe hardware and software development, FPGA acceleration integration, and building a PoE Cluster sandbox.**

I. Introduction

Field Programmable Gate Arrays (FPGAs) are often used in data centers for various applications. These devices can provide custom hardware systems using a vast quantity of reprogrammable resources. FPGAs contain lookup tables (LUTs), flip-flops (FFs), high-speed I/O, digital signal processing (DSP) units, and block RAM to meet the needs of the designer. The hardware provided for FPGA data center cards is often accessible through cloud services but does not allow the users to make low-level changes to the PCIe setup. It is costly to acquire such hardware (around $5000) [1]. This makes the barrier of entry for FPGA PCIe data center development almost impossible for any student or hobbyist.

II. Platform

This project introduces an affordable commercial-off-the-shelf (COTS) option for FPGA PCIe development (as low as $200). This includes a Raspberry Pi CM4, a Nitefury II FPGA (or other variation), and the Compute Blade baseboard. The CM4 device is an inexpensive device capable of providing a single PCIe lane (Gen 2, 8 Gbps). This can be combined with an inexpensive PCIe Xilinx FPGA (the Nitefury II) [2]. This FPGA is capable of 4 lanes of Gen 2 PCIe and contains 200k LUTs. The PoE Compute Blade baseboard provides a simple affordable setup for these two devices [3]. Figure 1 shows the components connected together in a complete setup.

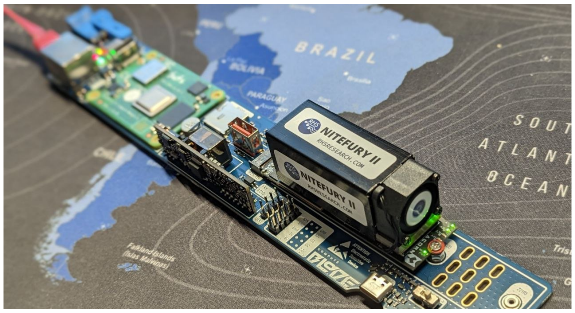

Fig. 1. Compute Blade with NiteFury II.

III. PCIe Development

The FPGA allows for the deployment of custom PCIe hardware. This allows for custom Vendor and Device IDs, as well as control of the quantity and size of the PCIe address bars. Xilinx does provide a complete PCIe solution (XDMA) with a software driver for the Raspberry Pi [4]. The designer can also implement the primitive PCIe block and configure how the hardware interacts with the data (Figure 2). There is another option for interaction with the raw transceiver data where the designer can implement their own PCIe PHY (ex. litepcie [5]).

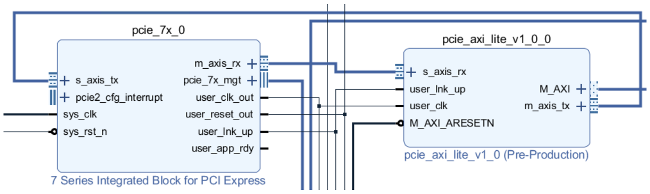

Fig. 2. Xilinx PCIe Primitive Example.

IV. FPGA Data Center Cluster

FPGAs have often been used for hardware acceleration of various applications such as video, deep neural networks, database querying, data compression, and much more. Xilinx offers a GitHub repository for many examples [6]. This platform can easily be scaled to provide a cluster for complex high-performance computing (HPC) development and testing. The PoE platform allows for an Ethernet to provide power, control, and data.

V. Project Status

The goal of this project is to provide various example templates to help others use this COTS platform. The GitHub repository currently contains examples of various PCIe implementations [7]. This includes examples for XDMA (streaming and memory mapped), a simple PCIe to AXI interface (with python driver), and the open-source litepcie. The current examples can reach a bandwidth of 330 MBps (or 2.46 Gbps) using XDMA drivers. The software driver for the litepcie example is currently halting the raspberry pi and will need further debugging. Further development will include additional hardware acceleration examples for several applications.

References

[1] Xilinx. Alveo u200 data center accelerator card. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/alveo/u200.html

[2] D. Reynolds. Rhs research. [Online]. Available: https://rhsresearch.com/

[3] U. Lab. Compute blade kickstarter. [Online]. Available: https://www.kickstarter.com/projects/uptimelab/compute-blade

[4] L. Barker. Saturn. [Online]. Available: https://github.com/laurencebarker/Saturn

[5] Enjoy-Digital. LitePCIe. [Online]. Available: https://github.com/enjoy-digital/litepcie

[6] Xilinx. Vitis libraries. [Online]. Available: https://github.com/Xilinx/Vitis_Libraries

[7] A. E. Wilson. FPGA RPi Blade. [Online]. Available: https://github.com/AEW2015/FPGA_RPi_Blade