Accommodating Multi-Tenant FPGAs in the Cloud

Joel Mandebi Mbongue, Christophe Bobda
ECE Department, University of Florida, Gainesville FL, USA
Email: jmandebimbongue@ufl.edu, cbobda@ece.ufl.edu

***Abstract*—This work presents a network-on-chip based architecture enabling multi-tenant access to FPGAs in cloud infrastructures. Prototyping the proposed architecture on Xilinx Virtex Ultrascale+ and Intel Stratix IV demonstrated performance similar to single tenant mode while enabling hardware consolidation.**

I. Introduction

FPGAs are getting an increasing interest from public clouds and cloud research projects. They are particularly attractive because of their ability to serve as energy efficient and customizable hardware accelerators. Commercial clouds have however highlighted the lack of multi-tenancy support, which does not permit hardware consolidation as it is not possible to space-share FPGA resources between multiple tenants. In this paper, we propose an architecture that divides the FPGA into logically isolated regions that we call "*virtual regions*" (VR). The VRs are immersed in a NoC interconnect allowing flexible communication, fast data movement, and low hardware footprint. The proposed architecture enables multi-tenancy as VRs can be allocated to different tenants at runtime.

II. Architecture Overview

Figure 1 illustrates the proposed architecture on FPGA. It leverages the height and width of devices to place columns of routers and VRs. When FPGA devices are wide enough to host multiple columns of routers, logic at the edge of devices that are in general underutilized are exploited for column crossing. In fact, vendor tools tend to place and route designs closer to the center of the chip unless constrained otherwise. The proposed architecture is quite similar to Hoplite with the

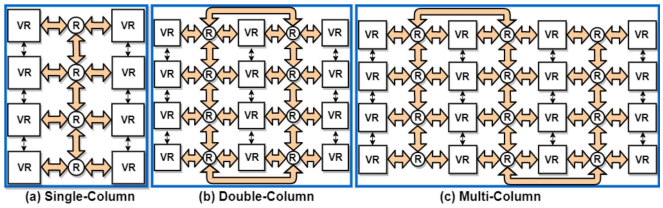

Fig. 1. FPGA Division into Independent Locations

difference that our routers are bidirectional, buffer data, and do not implement deflection [1]. The routers typically use 4 interfaces to limit hardware footprint and routing latency, which is the opposite of architectures such as FLexiTASK that rely on routers with higher number of interfaces for more flexibility [2]. Since routers at the edge of the NoC only utilize 3 interfaces, we implement a second flavor of the router architecture with only 3 ports to lower resource consumption. Each router implements a crossbar matrix along with IO arbiters. VRs essentially include a partially reconfigurable *USER REGION*, a *wrapper* that appends target information on outgoing data packets; and an *Access Monitor* that ensures that data flowing into user accelerators come from the right "*virtual instance*" (VI) running in the cloud (see Figure 2).

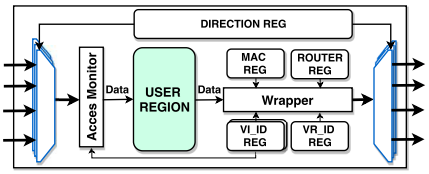

Fig. 2. Virtual Region Components

III. Performance Evaluation

We prototyped the proposed architecture on a Stratix IV (EP4SGX230KF40C2) and a Virtex UltraScale+ (xcvu9p-flgb2104-2-i) FPGAs. Figure 3(a) shows that sharing FPGA between tenants does not incurs significant penalty. In average, IO trips took about $30\mu s$, which is similar to results in [3].

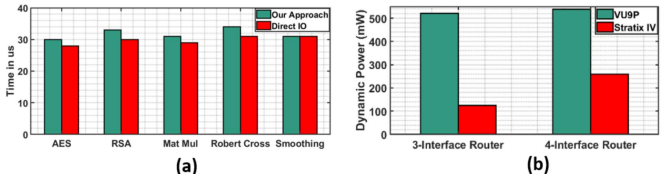

Fig. 3. (a) IO Trip comparison between single tenant + direct IO access and hardware accelerators deployed on the same device (both on VU9P and Stratix IV). (b) Power consumption comparison per device.

Figure 3(b) shows that optimizing the number of interfaces of the routers allows reducing power consumption, which is a consequence of using less resource on FPGA. We also observed a throughput reaching 7Gbps on continuous streams of 400KB and F_{max} topping at 700MHz on both FPGAs.

Acknowledgement

This work was partially supported by the ONR under the Grant CCN 0402-17643-21-0000, and the Air Force Research Lab AFRL/RIGA Cyber Assurance Branch, Rome NY.

References

[1] N. Kapre and J. Gray, "Hoplite: Building austere overlay nocs for fpgas," in *2015 25th International Conference on Field Programmable Logic and Applications (FPL)*, Sep. 2015, pp. 1–8.

[2] J. Mandebi Mbongue, D. Tchuinkou Kwadjo, and C. Bobda, "Flexitask: A flexible fpga overlay for efficient multitasking," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. ACM, 2018, pp. 483–486.

[3] J. M. Mbongue, F. Hategekimana, D. T. Kwadjo, and C. Bobda, "Fpga virtualization in cloud-based infrastructures over virtio," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 242–245.