Enabling RISC-V in Large Scale FPGA Platforms

Mazure D., Kropotov A., Cano F., F, Fradj, B., Fell A., Cervero T., Perdomo E., Davis J.
Barcelona Supercomputing Center (BSC-CNS)
Barcelona, Spain
{daniel.jimenez2, alexander.kropotov, francelly.canoladino, alexander.fell,
teresa.cervero, elias.perdomo, john.davis}@bsc.es

I. Introduction

Big FPGA-based infrastructures naturally aim to support plenty of different designs, which normally can be divided into two major blocks: FPGA components for interfacing with outer modules (e.g. PCIe, Ethernet, etc) and the implemented accelerator design for executing a targeted application. Generally, adapting any design to a specific FPGA infrastructure comprise non-negligible development time. In this scenario, several solutions emerged offering FPGA shells working out of the box, as it is the case of Xilinx Vitis Platforms and Amazon F1. These provide to the end-user a black-box approach adapted to the targeted FPGA particularities, demanding from him just certain level of compliance's design with a given set of interfaces. In the end, these solutions offer a quick bring-up for the intended designs, since all the board hardware connectivity complexity is hidden.

II. MEEP FPGA Shell (MEEP Shell)

The MareNostrum Exascale Experimental Platform (MEEP) [1] consists of a cluster with 96 FPGAs aiming to support a wide variety of accelerators that could exploit the massive number of resources. As part of the infrastructure includes a MEEP FPGA Shell, focused on self-hosted RISC-V architectures, which solves the infrastructure challenges by giving to the end-user a flexible and open platform to work with. Its novelty, with respect of the industrial proposals, is its flexibility by giving total freedom to add or remove components and functionalities at will. It currently gives support PCIe, HBM, 10Gb Ethernet, 100GbE, and Aurora FPGA links through QSFP+ out of the box. It is open source and extensible, meaning that any custom IP can be added to the Shell when it is not necessarily a specific part of the accelerator.

III. Continuous Integration and TCL for timing closure

The MEEP Shell has been designed considering not only the use of a ready-to-use U55C/U280 platform but also as a merging space for other FPGA based utilities. That is the case with the FPGA-oriented CI/CD and a TCL scripting approach for timing closure. Combining these utilities in the same project creates a fantastic approach for FPGA and non-FPGA users to benefit from one or all of these features to have a complete setup that fulfills the requirements of extensive, challenging projects.

The MEEP project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 946002. The JU receives support from the European Union's Horizon 2020 research and innovation program and Spain, Croatia, and Turkey.

In the RISC-V context, the MEEP FPGA Shell provides all I/O connectivity that a CPU requires from the system. At the same time, it can be extended or modified based on the deployment scenario. Difficulties such as the lack of a bootable storage medium like an SD card in Xilinx Alveo boards can be easily overcome for RISC-V Linux implementations, while Xilinx boards like Genesys 2 and others benefit from the SD card to boot Linux with open-source projects like OpenPiton or CV6. The MEEP FPGA Shell offers in exchange a PCIe-based solution, in which the boot process is handled by a PCIe sequence where the Linux kernel, the file system, and the device tree are all transferred to the HBM memory from the host, and then the RISC-V processor is released from its power-on reset. The boot process continues normally: the RISC-V CPU starts fetching instructions from a bootROM, and eventually jumps to the main memory (HBM) to find the bootloader (OpenSBI, Uboot, BBL, etc.).

The CI/CD mentioned above can be easily extended to be used in the RISC-V context. Any CI/CD is normally configured through a set of YAML files that define the entire pipeline. The MEEP FPGA Shell is constructed with placeholders for jobs that are specific to a given end-user design. This means that the end-user only needs to link the YAML file to the general flow to extend the MEEP FPGA Shell base functionality.

IV. Conclusion

While benefiting from these features, we have been able to bring up different RISC-V-based FPGA systems, highlighting OpenPiton implementations with different CPUs, such as Lagarto or CV6. Further than that, other RISC-V projects are now supported by the MEEP Shell, namely *PI, DVINO, Sarganta, BlackParrot, and RSD. The MEEP Shell creates an abstraction layer between the Alveo boards and the RTL that defines the accelerator. The MEEP Shell is compatible with any RISC-V based design. Moreover, the MEEP FPGA Shell is capable of storing a library with all supported targets, so that the end-user can build any of the supported projects instantaneously. All these designs trigger the basic CI/CD jobs, followed by those jobs that are specifically defined by the accelerator. At the same time, every design running under the MEEP Shell environment benefits from the scripted TCL, which is a feature typically underestimated, capable of iterating over different directives and making decisions based on the estimated timing closure analysis at different stages of the FPGA implementation flow.

References

[1] A. Fell *et al.*, "The MareNostrum Experimental Exascale Platform (MEEP)," *Supercomput. Front. Innov*, vol. 8, no. 1, pp. 62–81, 2021.