RISC-V FPGA Platform toward ROS-based Robotics Application

Jaewon Lee
Georgia Institute of Technology

Hanning Chen
Georgia Institute of Technology

Jeffrey Young
Georgia Institute of Technology

Hyesoon Kim
Georgia Institute of Technology

***Abstract*—RISC-V is free and open standard instruction set architecture following reduced instruction set computer principle. Because of its openness and scalability, RISC-V has been adapted not only for embedded CPUs such as mobile and IoT market, but also for heavy-workload CPUs such as the data center or super computing field. On top of it, Robotics is also a good application of RISC-V because security and reliability become crucial issues of robotics system. These problems could be solved by enthusiastic open source community members as they have shown on open source operating system. However, running RISC-V on local FPGA becomes harder than before because now RISC-V foundation are focusing on cloud-based FPGA environment. We have experienced that recently released OS and toolchains for RISC-V are not working well on the previous CPU image for local FPGA. In this paper we design the local FPGA platform for RISC-V processor and run the robotics application on mainstream Robot Operating System on top of the RISC-V processor. This platform allow us to explore the architecture space of RISC-V CPU for robotics application, and get the insight of the RISC-V CPU architecture for optimal performance and the secure system.**

***Index Terms*—RISC-V, FPGA, ROS**

I. RISC-V Open Architecture

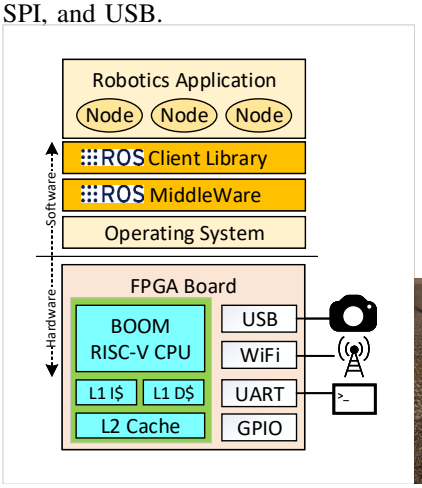
RISC-V open instruction set architecture (ISA) is open to public and free to use. This has attracted many industry SoC vendors and researchers because SoC vendors desire to reduce the SoC cost by using free CPU while the researchers have experienced the difficulty from researching with proprietary black-boxed CPU. RISC-V provided helpful ecosystem for the developers who want to implement whole software/hardware stack on their local FPGA. They can easily generate RTLs with Chipyard platform, and select the operating system, compiler, development and debugging tools for their own RISC-V CPU. However, RISC-V Foundation start providing the AWS cloud-based FPGA smulation environment [1], many tools for local FPGA is not updated actively. so the developers working on local FPGA need to put more effort to run their system.

II. Robot Operating System (ROS)

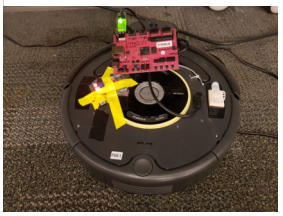
ROS largely consists of RCL (ROS Client Library) and RMW (ROS MiddleWare). RCL provides the interface between user application and ROS and supports the applications programmed by various language such as C, C++, or Python. The tasks of RCL are interfacing with user application while RMW is more likely to handle the ROS system operation. RMW schedules the tasks created by applications, and manages the communication messages from internal nodes and external nodes depending on the priority given by users.

III. System Overview

Figure 1 depicts the overview of our platform. We implements RISC-V BOOM(Berkeley Out-of-Order Machine) on an FPGA board running ROS and applications. Internally, RISC-V Core is connected with other peripherals via system bus, so it can communicate with outer world by using UART, I2C, SPI, and USB.



(a) HW/SW Stack



(b) FPGA on Robot Vacuum Cleaner

Fig. 1: System Overview

The FPGA board which implements RISC-V platform is connected with a robot vacuum cleaner via USB. It could process the input image from camera on the robot and navigate with the result processed.

References

[1] S. Karandikar, H. Mao, D. Kim, D. Biancolin, A. Amid, D. Lee, N. Pemberton, E. Amaro, C. Schmidt, A. Chopra, Q. Huang, K. Kovacs, B. Nikolić, R. H. Katz, J. Bachrach, and K. Asanović, "Firesim: Fpga-accelerated cycle-exact scale-out system simulation in the public cloud," *IEEE Micro*, vol. 39, no. 3, pp. 56–65, 2019.