Low-Power High-Performance Intelligent Camera Framework ROS-FPGA Node

PANADDA SOLOD†1 NATTHA JINDAPETCH†1 DANIEL PINHEIRO LEAL†2
TAKESHI OHKAWA†3

Abstract: In this work, we proposed a ROS-FPGA node for lane detection based on a ROS-compliant FPGA component on PYNQ-Z2 to support the complexity of the Advanced Driver-Assistance System (ADAS). In the preliminary experiment, the investment algorithm of lane detection based on High-Level Synthesis (HLS) was implemented on Programmable Logic (PL), and the traditional lane detection based on OpenCV was implemented on Processing System (PS) to compare the operation time. The PL fabric clock was 125 MHz, and the CPU clock was 650 MHz. The resolution of the input image was fixed to 400x520 pixels. We found that the processing speed of the lane detection algorithm on PL was 24 FPS and about four times faster than the lane detection on PS.

Keywords: ROS-compliant FPGA, lane detection, High-Level Synthesis (HLS).

1. Introduction

Advanced Driver-Assistance System (ADAS) is a popular driver assistance system. Many sub-systems of ADAS, such as Lane Departure Warning System (LDWS) and Lane Keeping Assistance System (LKAS), are parallel operated for situation awareness. Therefore, the requirements of ADAS are high-speed computation and low power consumption.

Robot Operating System (ROS) is a flexible framework for robotics programming that simplifies and increases productivity for multi-platform operations. In [1] and [2], ROS features are used for ADAS communications. [1] was implemented with adaptive AUTOSAR, and [2] was implemented with an autonomous platform using CARLA simulator and tested with Software-In-the-Loop (SIL). Although ROS is suitable to develop the multi-platform operations, the disadvantages of ROS are the low speed features transforming from the application to ROS and the limitation of power consumption. Therefore, ROS-compliant using ZYNQ platform is proposed in [3] to accelerate the robot system function with the same power consumption. [4] proposed the encapsulating hardware description on ROS nodes for the productivity-improving of robot development by mechanism generation from C/C++ in HLS. In addition, ROS2, which is the next version of ROS, is expected to enhance the applicability for real-time systems due to its layered software structure with reliable middleware, i.e. DDS (Data Distribution Service).

In this work, we aim to develop lane detection, which is a part of ADAS based on the ROS-compliant FPGA component by using PYNQ-Z2 at HLS to be a case study for low power with the high performance of the complex operation.

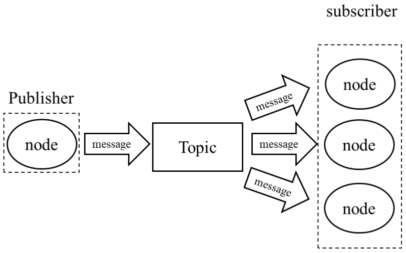

Figure 1. ROS concept

2. Theory

A. Robot Operating System (ROS)

ROS is a flexible operating system for software robot control, which contains the necessary tools and commands for easy development whatever the Application Programming Interface (API) and the library for convenience usage.

The ROS concept is shown in Figure 1. Both publisher and subscriber nodes separately operate and communicate with others through the topic. The whole system of ROS is managed by the ROS master.

B. ROS-compliant FPGA component

ROS-compliant FPGA component is a combination of a robot system and an FPGA technology that is suitable for complex operations which require high efficiency with minimal power consumption. Figure 2 shows the ROS-compliant FPGA component which includes two software parts: a data interface and a process execution. The data is subscribed to input through the topic for formatting to process on FPGA and re-formatting for publishing through the topic to ROS. The concept of the ROS-compliant FPGA component is applicable for ROS2. Our project Forest [5] is a tool proposed to generate ROS2 node, which includes the HLS-FPGA module and ROS2 message interface of the HLS module.

3. Proposal

In this work, we propose a low-power high-performance intelligent camera framework ROS-FPGA node. Lane detection

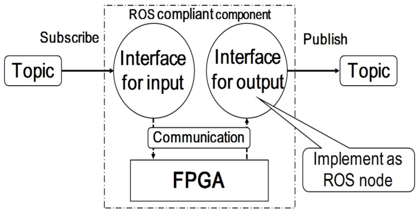

Figure 2. ROS-compliant FPGA component

†1 Department of Electrical Engineering, Prince of Songkla University, Thailand
†2 University of Toronto, Canada
†3 Tokai University, Japan

implementation is chosen as a case study. Following Figure 3, a lane detection, which contains 3 steps of pre-processing, edge detection (Robert edge detection), and line detection (Hough line transform), is created by using HLS and implemented on PL as an accelerator. The raw image is subscribed for formatting from ROS to FPGA to process the lane detection on PL. After the lane detection processing finished, the result of the lane detection on PL, which is the position of the lane (x, y), is reformatted from FPGA to ROS to publish the result to other systems for afterward processing. This combinational structure of ROS and FPGA is expected to integrate and operate the system flexibly. Moreover, the communication among ROS nodes gets more reliable and real-time by using the ROS2 in the proposed framework.

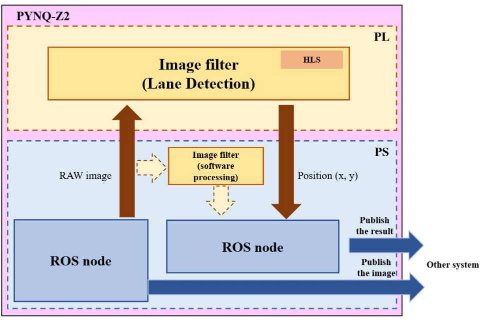

Figure 3. Framework ROS-FPGA node of lane detection

4. Experiment

In this experiment, we compared the processing time of lane detection on different parts on PYNQ-Z2 (PL and PS) at PL fabric clock 125 MHz and CPU clock 650 MHz and measured the resource usage and power consumption of the whole lane detection on PYNQ-Z2 by ROS system is included later. In Figure 4, the raw image is read from memory on PS to be an input of both lane detections on PL and PS. The output of the lane detection is written back to the memory of PS. The lane detection, which is implemented on PL is developed by using C/C++ on HLS, while the lane detection on PS is developed by using conventional OpenCV.

From Table 1, we found that the processing time of the lane detection on PL spent 0.041 seconds each frame. Meanwhile, lane detection on PS spent 0.153 seconds each frame.

The resource usage and the power consumption of the lane detection platform on Pynq-Z2 are illustrated in Table 2. The resource is illustrated as a number of resources and the percentage of resource usage from the available resources. There are BRAM, DSP48E, Flip-Flops (FF), and Look-Up Table (LUT). The power consumption is illustrated as Watt of the dynamic power consumption and the static power consumption.

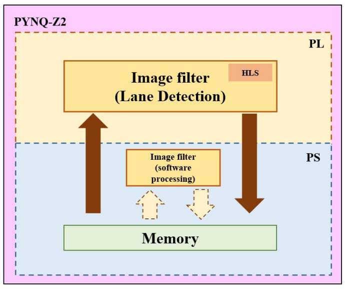

Figure 4. 1st experiment setup

Table 1. Processing time and performance

The processing part of Lane Detection	Processing time (sec.)	Performance (FPS)
Processing system (PS)	0.153	6.5
Programmable Logic (PL)	0.041	24

5. Conclusion

This work has presented the framework of ROS-FPGA node on PYNQ-Z2 with lane detection as a case study to be guild line for a complex system. The operating speed without communication to ROS, the lane detection on PL was faster than the lane detection on PS about 4 times.

The next step of our work is ROS communication considering. The input of raw image and the position output will be communication to ROS2 through the interface instead of read/write from the memory on PS.

Table 2. Resource usage and power consumption of lane detection on Pynq-Z2

Resource usage on Pynq-Z2				Power consumption (Watt)	
BRAM	DSP48E	FF	LUT	Dynamic	Static
24 (17%)	15 (7%)	4,043 (4%)	4,472 (8%)	1.372 (91%)	0.139 (9%)

Reference

[1] D. Stojanović, M. Krunić, N. Četić, and N. Lukić, ‘Source code generators for ADAS feature deployment in context of ROS and adaptive AUTOSAR applications’, in 2019 27th Telecommunications Forum (TELFOR), Nov. 2019, pp. 1–4.

[2] S. Stević, M. Krunić, M. Dragojević, and N. Kaprocki, ‘Development and Validation of ADAS Perception Application in ROS Environment Integrated with CARLA Simulator’, in 2019 27th Telecommunications Forum (TELFOR), Nov. 2019, pp. 1–4.

[3] K. Yamashina, T. Ohkawa, K. Ootsu, and T. Yokota, ‘Proposal of ROS-compliant FPGA Component for Low-Power Robotic Systems’, arXiv:1508.07123 [cs], Aug. 2015, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1508.07123.

[4] T. Ohkawa, Y. Sugata, H. Watanabe, N. Ogura, K. Ootsu, and T. Yokota, ‘High Level Synthesis of ROS Protocol Interpretation and Communication Circuit for FPGA’, in 2019 IEEE/ACM 2nd Intl. Workshop on Robotics Software Engineering (RoSE), May 2019, pp. 33–36.

[5] D. Pinheiro Leal, M. Sugaya, H. Amano, T. Ohkawa, “FPGA Acceleration of ROS2-Based Reinforcement Learning Agents”, 8th International Workshop on Computer Systems and Architectures (CSA'20) held in CANDAR’20, Nov. 24-27, 2020. (to appear)