A Soft-IP for Performance Measuring of the Zynq Ultrascale+ CPU/FPGA interface

Farnam Khalili*† and Roberto Giorgi*

* *Department of Information Engineering and Mathematics, University of Siena, Italy*
† *Department of Information Engineering, University of Florence, Italy*

ABSTRACT

FPGA MPSoC platforms are widely known for their advantages of reconfigurability, high-performance, and power efficiency especially in domain-specific architectures over general processors. In these platforms, one of the key performance metrics is the data throughput between the processing system (PS) and the programmable logic (PL) parts, particularly when it comes to offload a computationally intensive workload to the PL. This paper presents a soft-IP and its needed software to evaluate the performance of the data transfer by AXI-Lite interface between PS and PL of the Xilinx Zynq Ultrascale+. For that goal, we use an AXI-Lite register based interface between the PS and the PL and a CRC16 engine implemented on the PL in order to validate the correctness of the received data packets and find the maximum achievable throughput from the PS, while the software is running on Linux Ubuntu.

KEYWORDS: FPGA, Zynq Ultrascale, Performance Evaluation.

1 Introduction

FPGAs are digital chips, through which designers can configure any hardware functions into the integrated circuits. They have emerged with their flexibility and reconfigurability, which allow the users to modify and optimize the hardware with no manufacturing cost. FPGAs and CPUs are nowadays available on the same die with little extra cost, thus opening the possibilities of co-designing system software and soft-IPs for the PL, in order to accelerate parts of the software into the PL [Xil]. Some examples of applications that follow these acceleration paradigm includes Convolutional Neural Networks (CNNs) [GS+17], DNA sequencing [CCF+16], and real-time video processing [GN13]. For those applications, in which an FPGA MPSoC is deployed, studying the PS-PL interface and throughput of the transferred data can be beneficial. In this study, we use the AXIOM board [GPK19,GKP19,Gio17], which

[1]E-mail: {khalili,giorgi}@dii.unisi.it

is based on a Xilinx Zynq Ultrascale+ System on a Chip (SoC) [Xil], which allows the users to benefit from the 4-core ARM Cortex-A53 sub-system along with a full FPGA fabric. We build a set of hardware and software modules to evaluate the performance of the PS AXI Master ports (i.e., HPM0_FPD[1], HPM1_FPD, and HPM0_LPD[2]). In this respect, we designed the following components:

1. A soft-IP, which is connected to the PS AXI Master ports.
2. A Linux device driver for the proposed soft-IP.
3. A software Application Programming Interface (API) to interact with this soft-IP (and the related test programs).

2 Proposed Soft-IP

The context of the proposed soft-IP is depicted in Figure 1. The AXI Lite Register Module is an IP core proposed in [KPG18], through which software can transfer: i) a set of data packets, ii) a pre-calculated CRC16, and iii) the total number of bytes to be transferred. The PL sends: i) the computed time for one data transfer, ii) the results of CRC16 Engine module, and iii) the acknowledgement to verify that all the packets are received correctly.

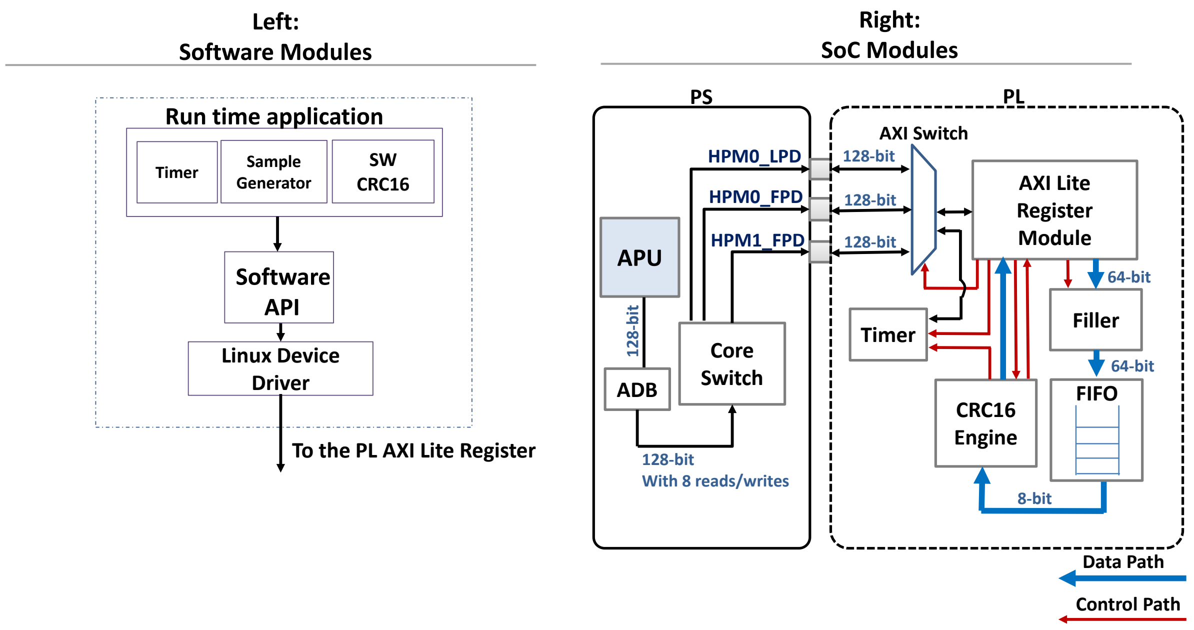

Figure 1: Block diagram of the implemented hardware architecture to evaluate the performance of the PS AXI Master ports. The left side presents the included software modules as well as the provided software API. The right side shows the utilized infrastructure on the SoC FPGA Zynq Ultrascale+. APU: Application Processing Unit. ADB: AMBA Domain Bridge. AXI Switch is controllable by the user space software to select the port under evaluation.

[1]High-performance master port in full power domain (with more features).
[2]High-performance master port in low power domain (with less features).

2.1 Software API

In software, the sample data is generated at run time, and the CRC16 of the whole generated data is calculated by a small software application before starting the PL operation. The software API is provided to interact with the Linux device driver and provide access to the PL sub-modules, which includes the AXI switch (in order to select desired PS AXI master port that is under evaluation), the PL Timer, the CRC16 engine and the FILLER. The functions of the API are briefly described in Table 1.

Table 1: Designed software API to interact with the proposed soft-IP

API	Description
set_crc_config(ps_crc_value, num_bytes)	It is used to transfer the total number of bytes and the calculated CRC16 in order to configure the CRC16 engine, and the FILLER.
set_port(port_num)	It is provided to select the AXI master port that is under evaluation (i.e., HPM0_FPD, HPM1_FPD, and HPM0_LPD).
stream_start_req(*data_set)	The prepared data is sent to the PL by calling this function, in a while loop to stress the PL register connected to one of the PS Master ports and to acquire the maximum achievable data throughput between PS and PL.
set_pl_timer()	Once the streaming of data begins, for calculating the transfer time, the timer is started by this function.
catch_pl_timer()	At the end, when the interrupt is received, all the metrics regarding the data transfer time can be acquired by calling this function. The corresponding data throughput can be achieved by dividing the number of bytes that sent over transmission time.
pl_reset()	This function resets all the PL sub-modules.
get_pl_crc()	Reads the result of the CRC16 engine.
irq_pending()	Reads the pending interrupts.
irq_clear()	To clear the interrupts.
irq_mask()	To enable the appropriate interrupt.

2.2 PL modules

In the PL side, one register is dedicated for the control commands like the START command to initiate the PL tasks. The FILLER module is written in Xilinx HLS, and gets the number of bytes and data from AXI Lite Register Module and fills up the FIFO through the AXI stream interface with the upcoming data. Then, the CRC16 engine dequeues the FIFO until the desired number of bytes are processed. The CRC16 engine is a VHDL module, which implements combinational logic having a latency of less than one clock cycle. It calculates the CRC16 value by using a given generator polynomial[1] for each 2048 of data

At the end, the computed CRC value is compared with the received CRC value from the PS, and if they match, the CRC16 engine enables the interrupt and the corresponding bit flags for the PS. The resource utilization of the PL module is reported in Table 2.

[1]The polynomial that we used is P(x)= $x^{15} + x^{13} + x^{12} + x^{11} + x^9 + x^7 + x^5 + x^3 + x^2 + 1$ to obtain the highest error detection capabilities by achieving hamming distance for the longest possible data word sizes [KC04].

Table 2: Resource utilization of PL modules on XCZU9EG

Resource	Utilization	%
LUT	3930	1.43
LUT RAM	203	0.14
FF	4987	0.91
BRAM	1.5	0.16
IO	27	13.24
BUFG	2	0.50

3 Conclusion

In this study, we illustrated a soft-IP intended to give to the software users of an CPU+FPGA MPSoC the possibility to precisely measure performance metrics such as the actual bandwidth of the high-speed master ports for transferring data from the CPU to the programmable logic of the Zynq Ultrascale+ platform. A CRC16 Engine with a delay less than one clock cycle was implemented in VHDL to verify the correctness of incoming data packets from the PS. We developed the Linux device driver for our PL design as well as a software API, through which the application can interact with the PL at run time. By our observation, we found that there are important parameters, which have significant impact on the data throughput. These parameters are AXI frequency, AXI data-width, and FIFO depth.

References

[CCF+16] Yu-Ting Chen, Jason Cong, Zhenman Fang, Jie Lei, and Peng Wei. When spark meets fpgas: A case study for next-generation {DNA} sequencing acceleration. In *8th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 16)*, 2016.

[Gio17] R. Giorgi. AXIOM: A 64-bit reconfigurable hardware/software platform for scalable embedded computing. In *IEEE 6th Mediterranean Conf. on Embedded Computing (MECO)*, pages 113–116, June 2017.

[GKP19] Roberto Giorgi, Farnam Khalili, and Marco Procaccini. Energy efficiency exploration on the zynq ultrascale+. In *2018 30th International Conference on Microelectronics (ICM)*, pages 48–54. IEEE, 2019.

[GN13] Mariangela Genovese and Ettore Napoli. Fpga-based architecture for real time segmentation and denoising of hd video. *Journal of Real-Time Image Processing*, 8(4):389–401, 2013.

[GPK19] Roberto Giorgi, Marco Procaccini, and Farnam Khalili. Axiom: A scalable, efficient and reconfigurable embedded platform. *Design, Automation and Test in Europe, the european event for electronic system design and test (DATE)*, 2019.

[GS+17] Kaiyuan Guo, Sui, et al. Angel-eye: A complete design flow for mapping cnn onto embedded fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):35–47, 2017.

[KC04] Philip Koopman and Tridib Chakravarty. Cyclic redundancy code (crc) polynomial selection for embedded networks. In *International Conference on Dependable Systems and Networks, 2004*, pages 145–154. IEEE, 2004.

[KPG18] Farnam Khalili, Marco Procaccini, and Roberto Giorgi. Reconfigurable logic interface architecture for cpu-fpga accelerators. *HiPEAC ACACES*, 2018.

[Xil] Xilinx Inc. Xilinx UltraScale Architecture.