2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

The Implementation of a Power Efficient BCNN-based Object Detection Acceleration on a Xilinx FPGA-SoC

Heekyung Kim
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, USA
hkim104@hawk.iit.edu

Ken Choi
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, USA
kchoi@ece.iit.edu

Abstract—**This paper focuses on the power efficient design on the FPGA SoC for the object detection system based on Binary Convolutional Neural Network (BCNN). Especially, for the small IoT devices, such as an intelligent dash-cam, computer vision system installed on an unmanned aerial vehicle, the power consumption could be a significant factor of the performance and scalability. However, the optimized FPGA design has limitations to reduce the overall power consumption amount. We focus on the design of the FPGA Accelerator as well as the effective design of the peripherals including CPU. In our proposed FPGA-SoC design, it supports not only FPGA but also CPU and the peripheral component can be supported by additional virtual memory system for reducing the processing time. Overall customization including customized BCNN, virtual memories for CPU and FPGA part allows our testbed to achieve low power consumption without speed degradation. Our testbed is based on customized YOLOv2 which consists of applied binary and half precision convolution, and pipeline-based architecture with accelerated hardware design on the target device. The target device used in this paper is the Xilinx ZYNQ-SoC based PYNQ Z-1 board. Our proposed system achieves 15.15 frames per second (FPS) and 1.45 watts of power dissipation. Our result shows that our design technique is effective for real-time object detection and low power system.**

Keywords-component; Low Power, Power Efficient, High-performanace, Convolutional Neural Network, Accelerator, FPGA SoC Design, PYNQ Overlay, Real-time Object Detection

I. INTRODUCTION

Convolutional neural networks (CNNs) are significantly effective on detecting targeted objects and suitable for training the system. Typically, general convolution requires intensive computing, thus inevitably, it consumes a huge amount of power during the convolution computation. The effort to reduce the power consumption and to improve high performance is that this type of trained and self-judgeable system will be applied to any autonomous systems including vehicles, IoT devices and portable and wearable devices. Also, it will be the main-line system of the further smart system. Another reason is that ultimately, usable power and available power amount are always limited in electrical power-based systems. In the software aspect, representatively four factors such as the size of reused data, the size of the layers, the size of multiplier-accumulators (MACs), and the size of weights can affect the power dissipation of the CNN model. The more computational complexity is required, the more power consumption it is generated [1]. The batch size, one of the precise factors, may be insignificant effect on power consumption [2], but before applying the above statement regarding the relationship between computational complexity and power consumption to be generalized, the relationship between the number of layers and weights should be considered. For example, we can assume that there are two cases. The first case is the deeper CNN model that has large numbers of layers, but it might have fewer weights during the computation. The second case is the lightened CNN model that has few numbers of layers with a large amount of weights than the first case. As a result, the power consumption of the second case is less than the first case, however, the reduced amount is insignificant. Also, when we consider the difficulty to achieve reducing the number of layers is much harder than reducing the number of weights as an accuracy aspect [3]. For these reasons, reducing power consumption through the software has limitations and does not guarantee the effectiveness of it.

However, the optimized SoC hardware structure can extend the limitation of the software methods. Modified convolution has been suggested in many articles, among them, we focus on the advantage of the binarized convolutional neural network (BCNN) architecture [4,5]. In this paper, we modify the BCNN for our testbed and implement the object detection system based on the PYNQ-Z1 board [6] and Darknet Network & Modified YOLOv2 Model [7]. The overall architecture is shown as in Figure 1. This paper proposes the efficient data transformation

978-1-7281-2980-8/19/$31.00 ©2019 IEEE
DOI 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00060

architecture between CPU and accelerator in FPGA-SoC for high-performance of the object recognition system.

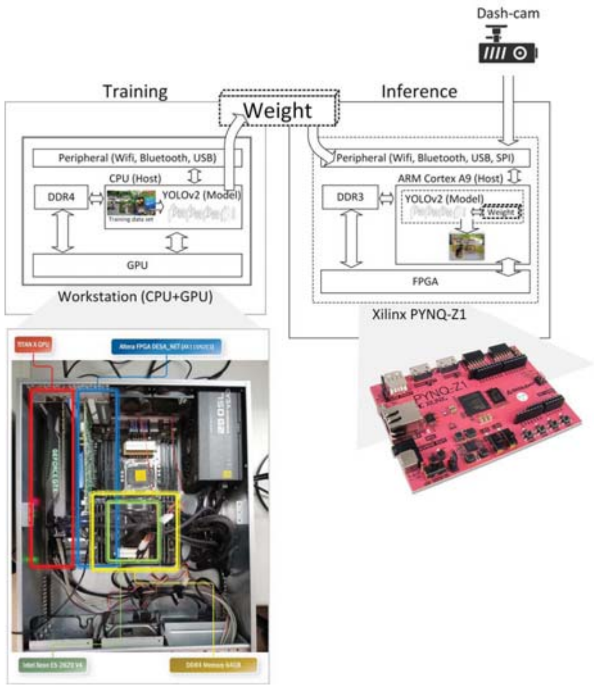


Figure 1. Overall Architecture of Programmable Logic

II. System Architecture

A. Accelerated Binary Convolutional Neural Network

The traditional BCNN [8] needs numerous times of the computing units than a typical CNN model which has the same accuracy. To make BCNN efficient, we used a method to reduce the model size. All BCNN [4] is increasing the resource usage and energy consumption while improving the accuracy with similar processing time. Moreover, it needs additionally pre-processed input data which should be decomposed input data into bitmaps in binarized precision by weight decomposition method. There are two processes required for this pre-decomposition: decomposition and copying bitmaps. Once the input data is decomposed into bitmaps, it is copied into a number of the bitmap. In order to accelerate this process, we implemented the optimized data transferring using AXI interfaces [9], which will be described in the next section.

B. YOLOv2: Object Detection Method for Small SoC

In this study, we implemented the YOLOv2-based accelerator on the PYNQ, Xilinx ZYNQ-SoC board. The YOLOv2 model is utilized to train and test image processing. There are various image detection methods, however, YOLOv2 is one of the state-of-the-art deep neural network models for the object detection system. Based on two criteria, efficiency and accuracy, YOLOv2 performance is approximately 80% accurate and executes at maximum 50 FPS, compared to DPM, R-CNN, Fast R-CNN, Faster R-CNN, SSD, YOLO, R-FCN [10]. It is especially well known for the fast object detection method under microprocessor-based computation circumstance.

C. Virtual Cache Design for CNN Accelerator

Our architecture has two cache memory components. Figure 2 shows that one of them is used for the convolutional binarized neural network. The other one belongs to the host area and supports efficient memory management of data sharing between the ARM cores and accelerator on the FPGA for data coherence.

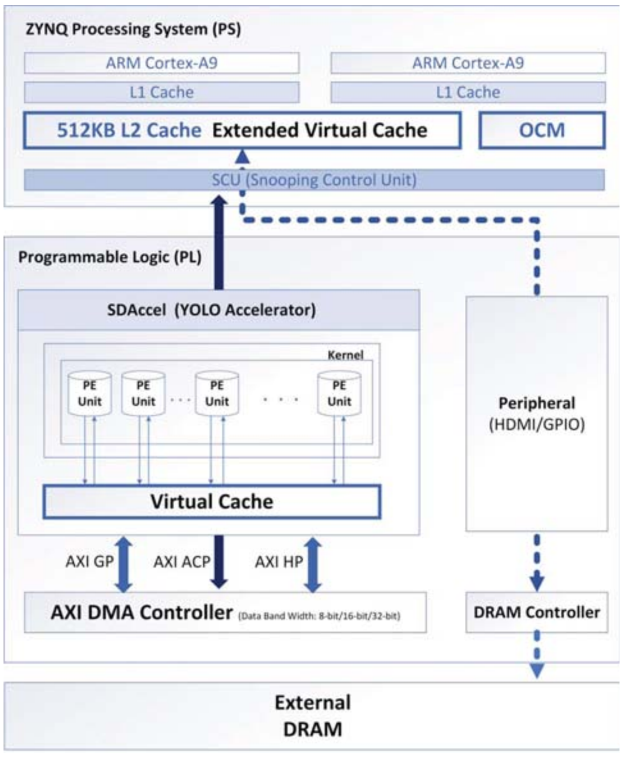
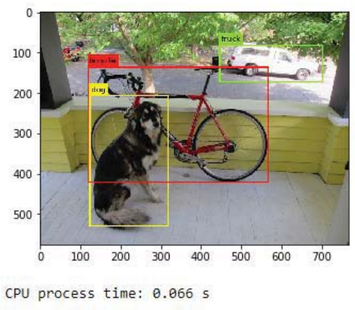


Figure 2. Overall System Architecture

III. Implementation

Instead of the regular ZYNQ-7020 board, we utilize PYNQ which consists of the ZYNQ-7020 chip, ARM Cortex-A9 processor, 512 MB DDR3 RAM, etc [6]. With the software framework, running Python on the Linux kernel to control APIs and drivers. In the application level, the user is capable of creating .ipynb format script to execute the Python code to show the process and the result on the same page. The processing result through the Jupyter notebook is shown in Figure 3. The core chip is based on the FPGA SoC platform which is composed of Programmable Logic (PL) and Programmable System (PS). Most of the processes are executed on the host part, ARM Cortex A9 side, and as a host, PS part will be controlling the data flow and hosting the entire process unit, as an accelerator, PL will support the computational operation.



```
CPU process time: 0.066 s
```

Figure 3. The Sample of the Jupyter notebook Framework

The basic software developing environment is based on Jupyter Notebook which is web-based software programming platform. It is also supporting Python, C/C++ programming languages, and other open source libraries such as OpenCV.

A. Accelerating Binary Convolutional Neural Network

As we already pointed out in the previous section, the typical BCNN has some disadvantages for the object detection process such as latency, inefficient hardware resource utilization, and power consumption. In the reference training environment [11], they used CIFAR-10 dataset, however, we used VOC data, and obtained the new weight training result. Using the result of [11], we were able to determine the optimal bit length for the precise binarized weight and tested the sample image by using the YOLOv2. The significant feature of the PYNQ is the AXI interfaces as shown in Figure 2. There are three types of AXIs, which are AXI-HP, AXI-GP, AXI-ACP, respectively [9]. AXI-HP supports the PL bus masters with high bandwidth data paths, thus it is suitable for data transformation between DRAM and OCM (On-chip Memory) [12]. AXI-GP is used for general use, but it can be directly connected from the master interconnect to the slave interconnect. AXI-ACP provides low-latency access for PL master. Also, it is contained AXI interface which can access to the L1 and L2, and OCM with maintaining data coherence. Those three AXI interfaces were assigned by the type properties as shown in Figure 2.

B. Experimentation on FPGA

To execute the sample code on this device without any performance degradation, it is required to take the BRAM or LUT for the memory structure. In the typical memory design, the memory area consists of the BRAMs. However, we extended the utilization amount of the LUT and LUTRAM for improving speed. In the paper [5, 12], by data flow, they modified, removed or edited the layers. The given environment setup of the Linux memory management system does not support the accelerator, consequently it needs the zero-copying process for initializing in the user space before accessing to the contiguous physical memory. We applied this method to our hardware design. It also can be applied to the accelerator memory for data coherence. By building customized IP blocks, we modified the pipelining of the data access to be suitable for the CNN model data flow.

IV. Result

As a result, we were able to obtain the latency results for YOLOv2, which was 66ms which is equivalent to 15.15 FPS and the power consumption amount was 1.45 watts. Compared with the [4], our implementation power efficiency by 10.44 times than CPU's, improved by 31.65% than ZCU102 implementation, and 48 times than GPU implementation result as shown in Table I.

TABLE I. Comparison with Embedded Platforms with Respect to the YOLOv2 Detection (Batch size 1)

Platform	CPU [12]	GPU [12]	ZCU102 [12]	PYNQ	PYNQ (Proposed)
Device	Quad-core ARM Cortex-A57	256-core Pascal GPU	Quad-core Cortex-A53	Dual-core Cortex-A9	Dual-core Cortex-A9
Clock Freq.	1.9 GHz	1.3 GHz	0.3 GHz	0.65 GHz	0.65 GHz
Memory	32GB eMMC Flash	8GB LPDDR4	32.1MB BRAM	512MB BRAM	512MB BRAM
Model	Mixed-precision YOLOv2				Accelerated BCNN-based YOLOv2
mAP(%)	85.2			69.6	79.6
Time[ms] (FPS) [1/sec]	4210.0 (0.23)	715.9 (1.39)	28.0 (35.71)	6250.7 (0.16)	66 (15.15)
Power	4.0	7.0	4.5	1.91	1.45
Efficiency [FPS/W]	0.057	0.211	7.93	0.083	10.44

The decreased power dissipation amount by proposed methods was 20% compared with the original design's power dissipation result. The accuracy rate is 79.6%, the speed is 15.15 FPS, power consumption is 1.45 watts. This result is significantly improved by compared to the result when we implemented the [13] structure on the PYNQ without any accelerating method. The processing time of the PYNQ without applying accelerating method is 0.16 FPS, 1.91 watts. Compared to it, our result represented 24% reduced power consumption by 24% and 125times power efficient.

V. Conclusion

This paper aimed to improve the energy efficiency, and high accuracy and fastest CNN-based object detection model for small devices, such as UAV, vehicles, IoT devices, and more. YOLOv2 is well known for the fastest and efficient object detection method. For the microprocessor-based board, YOLOv2 is still heavy network for real-time implementation. However, we were able to achieve power efficient and real-time object detection by implementing BCNN-based YOLOv2 with optimized FPGA accelerator design. In this paper, we suggested two factors that must be considered in the hardware design. First, the FPGA structure described in Section II Part B, there are significant hardware components such as LUT, LUTRAM, and BRAM. These elements affect directly on increasing the dynamic power consumption. Second, based on our assumption, proposed design techniques can reduce power consumption without any speed loss and improve system performance with keeping low power dissipation. In future work, we are planning to apply to other various network or model to verify the effectiveness of our suggested design method.

ACKNOWLEDGMENT

The work was financially supported by Industrial Core Technology Development Program of MOTIE/KEIT (10083639, Development of Camera-based Real-time Artificial Intelligence System for Detecting Driving Environment & Recognizing Objects on Road Simultaneously).

REFERENCES

[1] T. Yang, et al., "Designing Energy-Efficient Convolutional Neural Networks Using Energy-Aware Pruning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6071-6079.

[2] Al Culureiello. et al., "An Analysis of Deep Neural Network Models for Practical Applications," arXiv preprint arXiv:1605.07678 [cs.CV], 2017

[3] F.N. Iandola, et al., "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size", arXiv preprint arXiv:1602.07360 [cs.CV], 2016.

[4] Y. Zhou, et al., "Deep learning binary neural network on an FPGA," *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 281-284.

[5] M. Shimoda, S. Sato and H. Nakahara, "All binarized convolutional neural network and its implementation on an FPGA," *2017 International Conference on Field Programmable Technology (ICFPT)*, 2017, pp. 291-294.

[6] Digilent, Inc., "PYNQ-Z1 Board Reference Manual," 2017, https://reference.digilentinc.com/_media/reference/programmable-logic/pynq-z1/pynq-rm.pdf.

[7] J. Redmon et al., "You only look once: Unified real-time object detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[8] H. Nakahara, et al., "An object detector based on multiscale sliding window search using a fully pipelined binarized CNN on an FPGA," *2017 International Conference on Field Programmable Technology (ICFPT)*, 2017, pp. 168-175.

[9] Xilinx Revision, "PS/PL Interfaces," 2018, https://pynq.readthedocs.io/en/v2.4/overlay_design_methodology/pspl_interface.html.

[10] Z. Yang, J. Li and H. Li, "Real-time Pedestrian and Vehicle Detection for Autonomous Driving," *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018, pp. 179-184.

[11] I. Hubara, et al., "Binarized Neural Networks" Advances in Neural Information Processing Systems," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 4114-4122.

[12] T. Xiao, et al., "Unified Virtual Memory Support for Deep CNN Accelerator on SoC FPGA," *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, 2015, pp. 64-76.

[13] H. Nakahara, M. Shimoda and S. Sato, "A Demonstration of FPGA-Based You Only Look Once Version2 (YOLOv2)," *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, 2018, pp. 457-4571.