CNN Accelerator with Minimal On-Chip Memory Based on Hierarchical Array

HyunWook Son, YongSeok Na, TaeHyun Kim, Ali. A. Al-Hamid, HyungWon Kim
School of Electronics Engineering
Chungbuk National University, Cheongju, Korea
hwson@cbnu.ac.kr, ysna@cbnu.ac.kr, thk4627@cbnu.ac.kr, alihamid@chungbuk.ac.kr, hwkim@cbnu.ac.kr

2021 18th International SoC Design Conference (ISOCC) | 978-1-6654-0174-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISOCC53507.2021.9613997

Abstract—**This paper presents an architecture of CNN accelerator based on a new processing element (PE) array called a diagonal cyclic array. It can significantly reduce the burden of repeated memory accesses for feature data and weight parameters for CNN models. To evaluate the effectiveness of the proposed architecture, we implemented a CNN accelerator for YOLOv4-Tiny consisting of 9 layers. We also present how to optimize the local buffer size with little sacrifice of inference speed. We evaluated the example CNN accelerator using FPGA implementation with 24932 LUTs, 584 DSP blocks and a on-chip memory of only 58KB. It demonstrates an accuracy 58% (mAP0.5) with computation time of 240ms for each input image using a clock speed of 100MHz. This speed is expected to reach 2.4ms using a clock speed of 1GHz, if implemented in a silicon SoC using a sub-micron process.**

Keywords: CNN; Accelerator; SoC; Systolic Array; Memory optimization

I. INTRODUCTION

While GPU-based accelerators often provide advantage of high programmability accommodating any structure of neural network models, they often suffer from the poor utilization of hardware resource such as PE and excessive repeated memory access to the same data for CNN implementation. In contrast, the proposed accelerator architecture offers maximal utilization of PEs near 100% and minimal repeated access to same memory, while it may have limited support for certain CNN structures.

II. PROPOSED ARCHITECTURE OF CNN ACCELERATOR

Fig. 1 illustrates the overall architecture of the proposed CNN accelerator SoC. It consists of a systolic array of PEs called a diagonal cyclic array, an input buffer, a convolution memory (Conv-mem), a post processing block conducting pooling, normalization, and activation function in a pipelined manner, and finally an output memory storing the post processing block's results for each layer.

The CNN accelerator can be connected to external DDR SDRAM through high-speed data bus such as AXI and DRAM controller; see Fig. 1. The accelerator exchanges input and output data between the external memory and the on-chip memory that comprises global input buffer and output buffer. The diagonal cyclic array has a novel hierarchical structure consisting of two levels. Its upper level is composed of N rows and M columns of sub-arrays, while the lower level is composed of N x M sub-arrays. Here N is the number of output channels that can be processed simultaneously by the array, while M is the number of input channels that can be processed simultaneously. Each sub-array has a structure of F x F PEs that can calculate convolution operations for a kernel of size F x F. For the sake of ease of explanation, we take as an example CNN model, YOLOv2-Tiny which employs 3 x 3 convolution kernels in all 9 layers. To implement accelerator of this example, we construct each sub-array using 3 x 3 PEs, which is depicted by Fig. 2.

The sub-array receives input data at the first column from the input buffer every clock cycle, while it receives weight parameters at the row from the weight buffer. The input data is propagated through each row in the horizonal direction, while the weight parameters are propagated through each column in the vertical direction of the sub-array of 3 x 3. The propagation in both directions is conducted every clock cycle. This is how the proposed architecture can achieve near 100 % utilization of PEs.

After a few cycles for feeding 3 input data and 3 weight parameters to the sub-array, the sub-array continuously

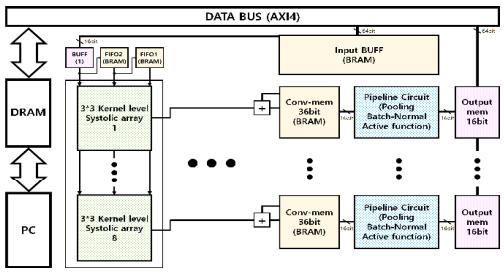

Fig. 1. Proposed CNN Accelerator Architecture

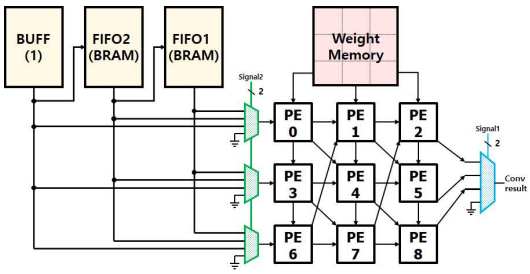

Fig. 2. Procession Systolic Array Architecture

produces 3 x 3 convolutional kernel outputs. Each 3 x 3 kernel is calculated by 3 PEs in each diagonal path. The above example of 3 x 3 sub-array has 3 diagonal paths as illustrated by Fig. 2, which calculates three 3 x 3 kernels consecutively. The consecutive kernels are produced by taking the 3 rows and 3 columns of the input image or input feature map data. The next 3 rows of the input data while keeping the same 3 columns are fed to the sub-array to calculate the next three 3 x 3 kernels.

For the above example, we implemented an CNN accelerator with an upper level of the diagonal cyclic array of N = 8 output channels and M = 4 input channels. It generates 8 output convolutional kernels simultaneously, which are temporarily stored in the Conv-Mem. The contents of Conv-Mem are read out and passed to eight parallel post-processing blocks, whose outputs are finally stored in the on-chip output buffer. The diagonal cyclic array starts when the input buffer and weigh buffer are filled by DRAM controller. Then all sub-arrays take input data and weight parameters corresponding to 4 input channels and 8 output channels simultaneously.

The input buffer consists of one register and two FIFOs as illustrated by Fig. 3. Fig. 3 shows the steps of input data propagation from the register to FIFO2 and then to FIFO1. In each step, the red arrows indicate which data entry of FIFO1 feeds which PE.

In order to minimize the required size of the on-chip memories, we partition the input image or input feature map into an array of slices. Each slice is defined by a square of size S x S. In the next section, we analyze the impact of slice size on the on-chip memory size and the speed of CNN accelerator.

III. Evaluation of the Proposed Architecture

We implemented the example CNN accelerator described above using Verilog HDL and FPGA. The accelerator includes a diagonal cyclic array of 32 sub-arrays (8 x 4). Each sub-array is constructed as an array of 3 x 3 PEs. In order to determine the optimal slice size, we conducted analysis of required on-chip memory size (Kbytes) and computation speed in terms of frames per second (FPS). Fig. 4 compares the slice size from 26 x 26 to 416 x 416. Note that the CNN model used in this example implementation is YOLOv2-Tiny whose input image size is 416 x 416 pixels. While the on-chip memory (Bram for FPGA) rapidly grows along with the slice

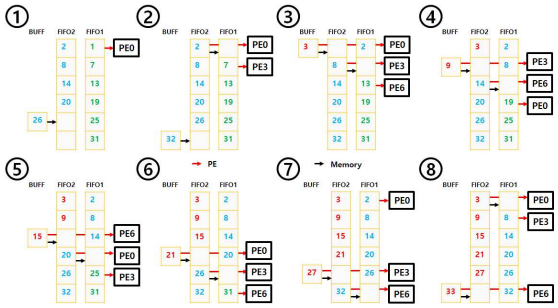

Fig. 3. Propagate Data flow of Memory structure with PSA data flow

Slice Size	Bram Size(KB)	FPS
26-26	27.488	5.97963778
52-26	54.944	6.05458633
104-26	109.856	6.10581537
208-26	219.68	6.13006339
416-26	439.328	6.15106133
416-52	871.968	6.16276415
416-104	1737.248	6.16863227
416-208	3467.808	6.17003595
416-416	6928.928	6.17073803

Fig. 4. Memory size and Latency per each slice size

size, the speed (FPS) exhibits little increase showing virtually no improvement. This analysis concludes that smallest slice of 26 x 26 is the best choice for minimal memory size at little sacrifice in the speed.

Therefore, we chose a slice of 26 x 26 in the final implementation of the accelerator in the FPGA (Xilinx based Next Video FPGA). The implementation consumed 24932 LUTs, 584 DSP blocks, 13 Block Rams for a total of only 58 KBs. This implementation requires an extremely small on-chip memory, which is 20 times smaller than the previous work reported in [3].

IV. Conclusions

This paper presented a compact CNN accelerator architecture that employs a novel hierarchical PE array called diagonal cyclic array. It minimizes the memory accesses for the same data, while maximizing the utilization of PEs in the array. FPGA implementation and analysis of optimal slice memory demonstrated that the proposed architecture can reduce the on-chip memory size by 20 times compared with a previous work [3]. Therefore, the proposed architecture can be an effective solution to constructing compact low-power CNN accelerator chips for mobile devices.

V. Acknowkedgnent

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2020-0-01462) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)" and also supported by the AURI(Korea Association of University, Research institute and Industry) grant funded by the Korea Government(MSS : Ministry of SMEs and Startups). (No. S2929950, HRD program for 2020)

VI. Reference

[1] C. -N. Liu, Y. -A. Lai, C. -H. Kuo and S. -A. Zhan, "Design of 2D Systolic Array Accelerator for Quantized Convolutional Neural Networks," 2021 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), 2021, pp. 1-4, doi: 10.1109/VLSI-DAT52063.2021.9427336.

[2] A. Samajdar, Y. Zhu, P. Whatmough, M. Mattinaa, T. Krishnag, "SCALE-Sim: Systolic CNN Accelerator Simulator," arXiv:1811.02883v2 [cs.DC] 2 Feb 2019.

[3] J. Zhang et al., "A Low-Latency FPGA Implementation for Real-Time Object Detection," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.940157