Spiking row-by-row FPGA Multi-kernel and Multi-layer Convolution Processor.

Ricardo Tapiador-Morales, Antonio Rios-Navarro, Juan P. Dominguez-Morales,
D. Gutierrez-Galan, Alejandro Linares-Barranco
Robotics and Tech of Computers Lab. University of Seville. Seville, Spain 41012
Email: ricardo@atc.us.es http://www.rtc.us.es/

***Abstract*—Spiking convolutional neural networks have become a novel approach for machine vision tasks, due to the latency to process an input stimulus from a scene, and the low power consumption of these kind of solutions. Event-based systems only perform sum operations instead of sum of products of frame-based systems. In this work an upgrade of a neuromorphic event-based convolution accelerator for SCNN, which is able to perform multiple layers with different kernel sizes, is presented. The system has a latency per layer from 1.44 μs to 9.98μs for kernel sizes from 1x1 to 7x7.**

***Index Terms*—Spiking Convolutional Neural Networks, FPGA, Computer vision, Neuromorphic engineering.**

I. Introduction

Neuromorphic engineering develops VLSI systems taking inspiration from the structure and functioning of human brain, where information is encoded in spikes that are processed by layers of neurons. From this new approach, several sensors have been developed, such as DVS [1], where the particular behaviour of cells of the mammal's retina are mimicked on mixed-signal chips. Each pixel of these sensors behaves as a neuron that produces an spike (event) if changes of luminosity reach a threshold. The main advantage of this kind of sensor compared with frame based vision sensors is that not all pixels of the image are processed. Hence, only those that detect changes in the scene produce an event. This flow of events can be used to solve machine vision tasks, such as detecting [2], tracking objects or being processed by Spiking Convolutional Neural Networks (SCNN) for more general scenarios.

In a previous work, a 128x128 convolution processor implemented in FPGA was presented. The processor is based on the LIF neuron model and it implements the refractory period and leakage properties. This processor is able to perform 64 convolutions with kernel sizes from 1x1 to 7x7, reading data row by row instead of pixel by pixel, reducing memory access, thus the latency. However this processor presented an important limitation, it can only implement one layer with a kernel size at a time, so if less than 64 convolutions are performed, some convolution units are not working, wasting computational resources.

In this work, we present a fully configurable version of the FPGA convolution processor that divides the convolution units in layers; therefore, each layer can perform convolutions with different kernel sizes and sending their output events to the next layer automatically. This new characteristic reduces the amount of wasted computational resources, making it easier to perform an SCNN.

The paper is structured as follows: Section II explains how the engine works with the new features added. Section III presents the conclusions.

II. Spiking Convolution Architecture

A. System overview

The architecture presented in this paper is an updated version of the one presented on [3].

The design has a 32-bit bus to be configured with a host microcontroller and two Address-Event Representation (**AER**) bus [4] to receive/send events. AER bus follows a four step handshake protocol to connect with neuromorphic devices.

The processor stores the kernel weights, the membrane potential, leakage and refractory period timestamps of all neurons in available block RAM (**BRAM**) memory from the FPGA. In order to apply leakage and refractory period, the design has two global counters, which counts are compared with timestamps stored in BRAM. However, those counters can overflow and then, leakage and refractory period would be applied wrongly. With the aim to avoid this situation, two mechanisms based on distributed RAM (**LUTRAM**) were developed. Those mechanisms store in LUTRAM memory a flag bit that indicates that an overflow has occurred applying leakage and refractory period correctly.

Convolution operation is performed in the convolution engine (**CE**) module, it consists of a state machine that reads rows of data from memories performing the spiking convolution. During convolution timestamps are compared with the global counters in order to apply leakage and refractory period. This module generates a spike if the membrane potential of a neuron reaches a configurable threshold, sending the (x,y) position of the spike as an event with the convolution ID.

The system presented in this paper groups convolution engines into layers with their respective kernel sizes.

The following subsections describe the new features of the architecture and the FPGA results.

B. Configuration module

The microcontroller configures a layer mask (**ML**), this mask indicates for each convolution engine which layer they belong to. Through this mask, convolution engines are able to select their parameters and send output events to the next

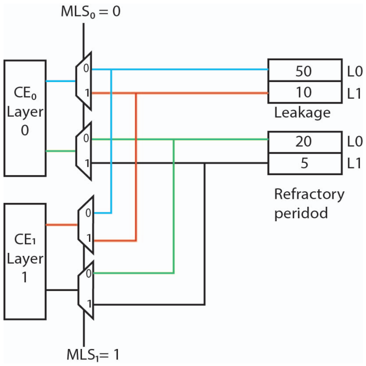

Fig. 1: Parameter selection using Layer Mask for two CE of different layers.

layer. The fact of having multiple layers implies that memory registers to store the parameters and the multiplexers size increase in function of the number of layers. Fig.1 represents an example of parameter selection for 2 CE of different layers. In this figure, there are two register banks that store the leakage and the refractory period values for both layers. Through ML they select the channel to read the corresponding layer data.

C. *Cycle output*

Convolutions can be configured in groups with different kernel sizes and the output events contain information about the used convolution engine. It is possible to perform multiple convolutions in one chip.

The convolution id with the layer mask allows the output multiplexer to route the output event to the next layer or the output AER bus. Following the previous example of two CE from different layers, Fig. 2 illustrated how data is routed between layers to the output. This new characteristic is a novelty in this kind of processors since other processors need to reconfigure the system between convolution layers to load the weights of the following layer, whereas other designs remove this reconfiguration step duplicating the hardware with the kernels loaded, but this solution increases the power consumption.

Resource	Utilization	Available	Utilization %
LUT	257503	277400	92,83
LUTRAM	50851	108200	47.00
FF	179925	554800	32,43
BRAM	713.5	755	94,37
IO	43	362	11,88

TABLE I: FPGA resources.

D. *FPGA implementation*

The design was described in System Verilog and synthesized for a Zynq-7100 MMP platform using Vivado 2016.4.

The design presented works at 90 MHz with a latency of 1.44-9.98 μs and an input throughput of 0.69-0.10 Mevps (megaevents per second) for kernel sizes from 1x1 to 7x7 respectively. FPGA resources consumed for 64 convolution engine with two convolution layers are shown in Table I. Fig. 3 shows an example of filters output from the system.

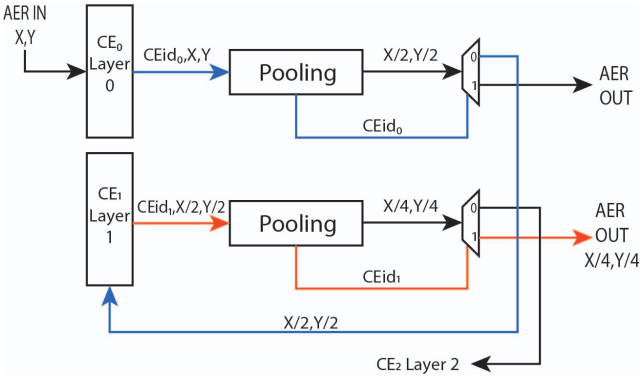

Fig. 2: Event routing for two convolution layer with pooling enabled.

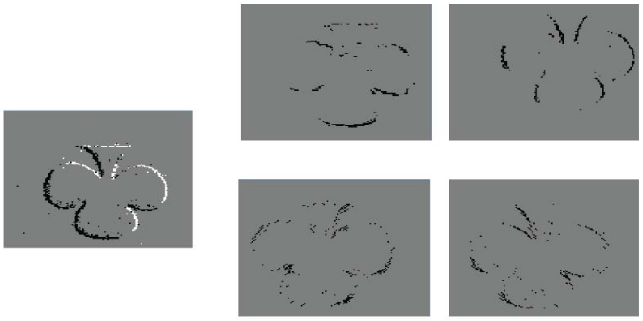

Fig. 3: Example of filters output for one layer of Gabor filters and a second layer of Sobel kernels.

III. Conclusions

This article has presented a neuromorphic multi-kernel and multi-layer convolution processor for FPGA. The architecture has added new features from previous works to be capable to implement multiple layers in the same chip to deploy an spiking convolutional neural network in a future.

References

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 x 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, feb 2008.

[2] A. Linares-Barranco, H. Liu, A. Rios-Navarro, F. Gomez-Rodriguez, D. P. Moeys, and T. Delbrück, “Approaching retinal ganglion cell modeling and FPGA implementation for robotics,” *Entropy*, vol. 20, no. 6, p. 475, 2018. [Online]. Available: https://doi.org/10.3390/e20060475

[3] R. Tapiador-Morales, A. Linares-Barranco, A. Jimenez-Fernandez, and G. Jimenez-Moreno, “Neuromorphic lif row-by-row multiconvolution processor for fpga,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 1, pp. 159–169, Feb 2019.

[4] R. Berner, T. Delbruck, A. Civit-Balcells, and A. Linares-Barranco, “A 5 meps $100 usb2.0 address-event monitor-sequencer interface,” in *2007 IEEE International Symposium on Circuits and Systems*. IEEE, 2007, pp. 2451–2454.