Technology & Hardware for nEuromorphic coMPuting

- ECSEL Research and Innovation Actions (RIA*) –

Deliverable 4.6
TCAM Block Design for SNN

Work Package	WP 4 – Design and Architecture
Document Date	31--2020
Revision	1.0
Status	FINAL
Dissemination Level	Confidential
Responsible Partner	CEA
Name	Thomas Mesquida
Contact Information	Thomas.Mesquida@cea.fr

* This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826655. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Netherlands, Switzerland".

1. Publishable summary

The Spiking Neural Networks (SNN) developed in TEMPO consist in different types of neurons interconnected in networks in order to compute meaningful tasks. The topology the form can take multiple shapes depending on the task attributed to the network or some of its sub-parts. Neuron layers can have only local connectivity with respect to input feature maps, as it is the case within Convolutional Neural Networks (CNN), complete connectivity for Fully Connected (FC) layers and recursive connections for Recurrent Neural Networks (RNN).

Each of these connectivity types are associated with architecture optimized to support them. CNN rely on event-based kernel fetching within weight memory in order to integrate them into the soma potential of targets neurons depending on the address of the origin spike. FC/RNN rely on weight matrix in which the spike origin address gives which line has to be read and the different columns output their word to stimulated neurons.

However, these implementations are oversized for layers with sparse connectivity. Using weight matrix in order to support highly sparse FC layers will result in only few percent of it actually being used. In order to have the possibility to connect any pair of neurons belonging to 2 successive layers while fully exploiting the designed hardware, we have to change our approach. We would still want the spike origin address to be used has weight search address but only allocate few slots corresponding to its actual targets with non-zero weights. Similar issues can be solved at the routing level within SNN with the following solutions.

The Content Addressable Memory (CAM) implements such mechanism in two parts. Each memory slot in a CAM has an address word and the data associated to this address. When an address word is presented to the CAM, it is compared to the address word of each slots. The slots programmed with this address word match the current input address and the associated data is retrieved. Thus, using CAM instead of weight matrix allows to have heterogeneous connectivity while being a tight fit to the actual needs of the implemented SNN.

Lastly, the Ternary CAM (TCAM) adds support for “don’t care” bits within the stored address words. This means that the address words programmed with “don’t care” bits will match input words no matter the state of its bits in the “don’t care” slots. Thus, TCAM allows to have memory slots that will only match parts of the input address word. This mechanism can be used to implement synapses that will be targeted by multiple different neurons from the previous layer without having to allocate multiple synapse slots within a CAM.