Hardware architecture of Self-Organizing Maps

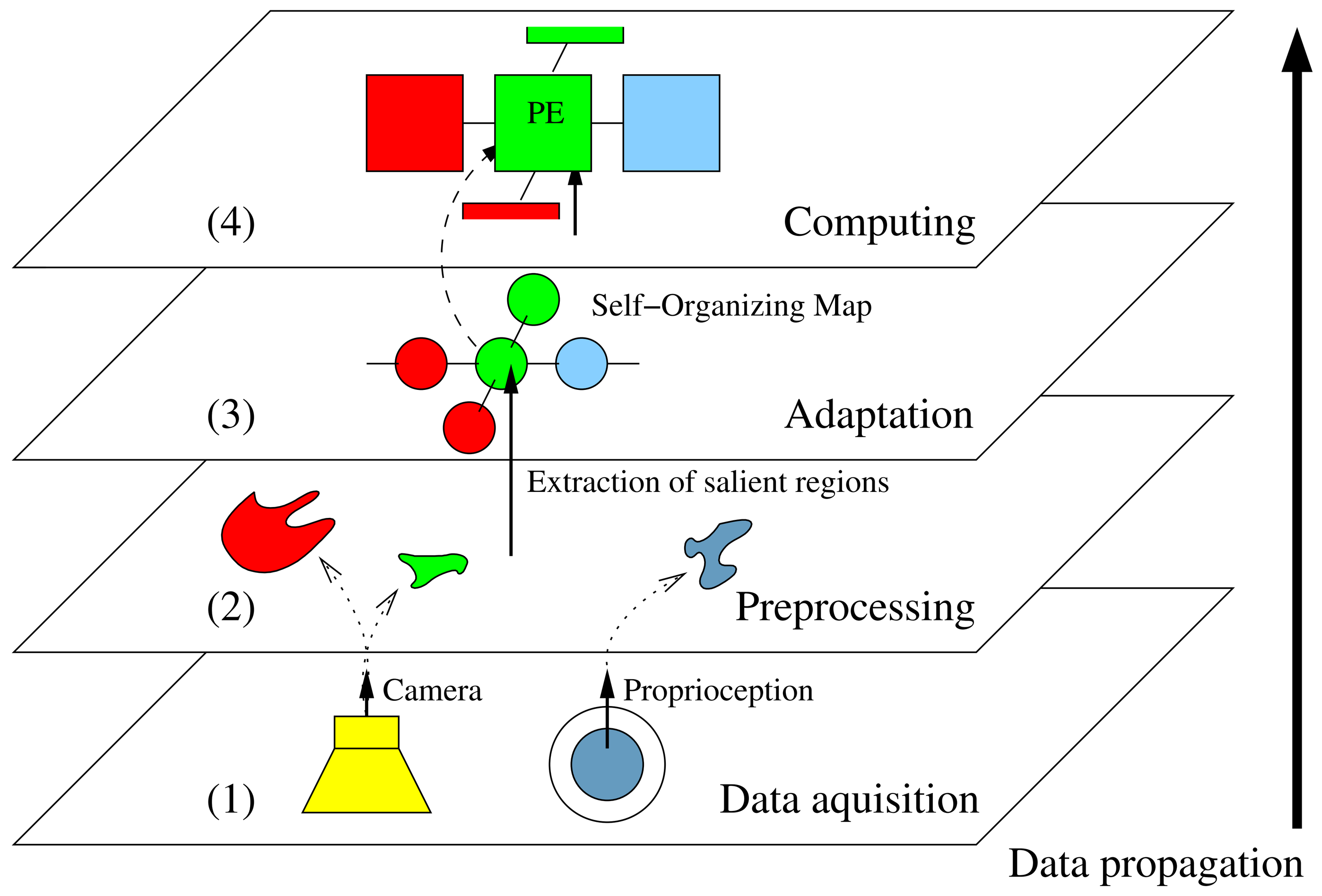
Laurent Rodriguez, Laurent Fiack and Benoît Miramond

ETIS / CNRS UMR 8051 - ENSEA - Université de Cergy-Pontoise

6 avenue du Ponceau F-95000 Cergy-Pontoise Cedex, France

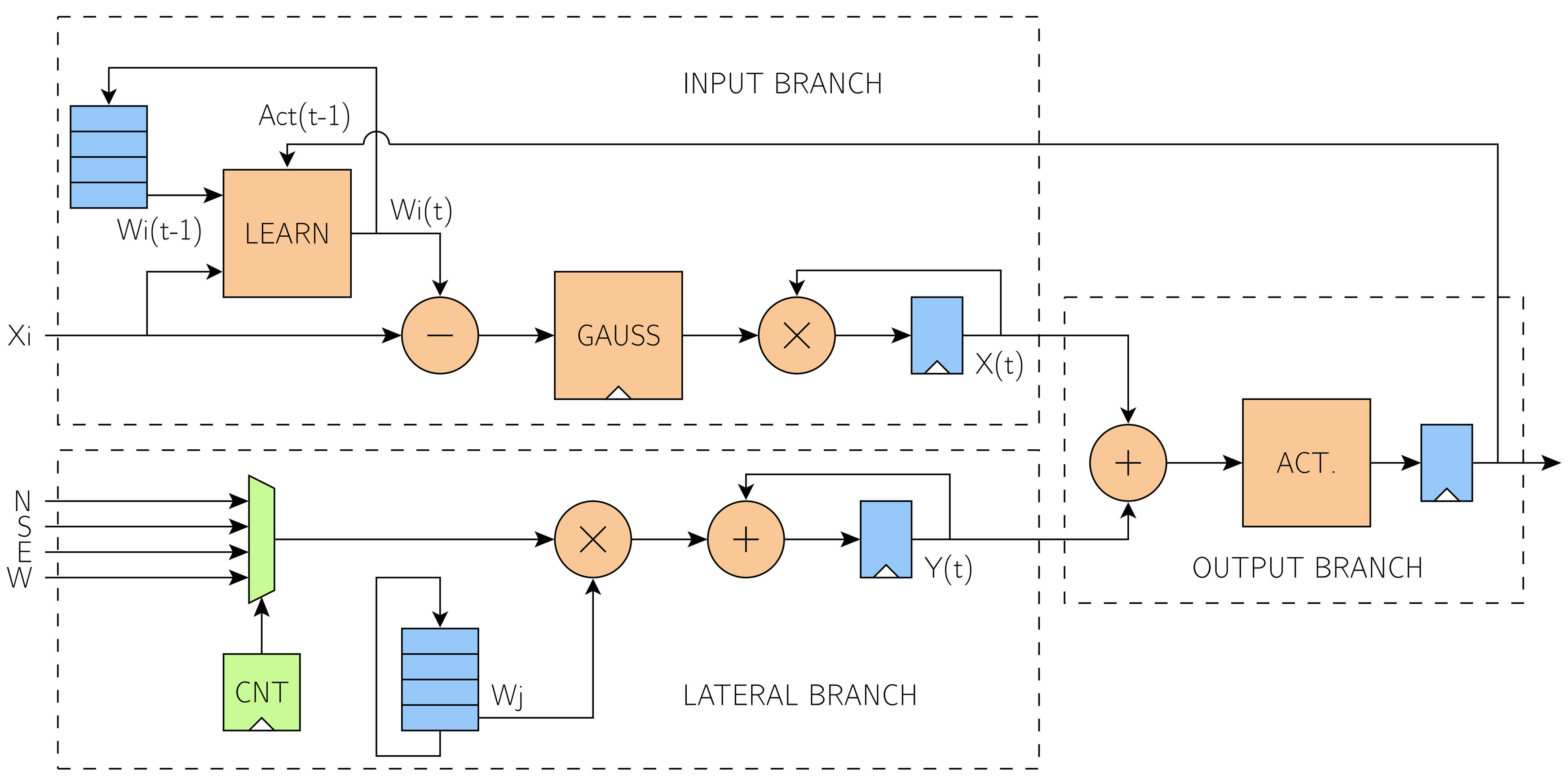
Introduction and State of the art

We define a hardware controller in which a grid of processing elements (PEs) will support a set of neuro-cognitive processes in order to drive a robot in different tasks. We propose an original artificial neural network named DMAD-SOM inspired by Neural Fields equations that have shown self-organizing behaviors and can be suitable for this purpose. It can be used to take allocation decisions locally, taking in account the state of the whole system through the emergent behavior of the network.



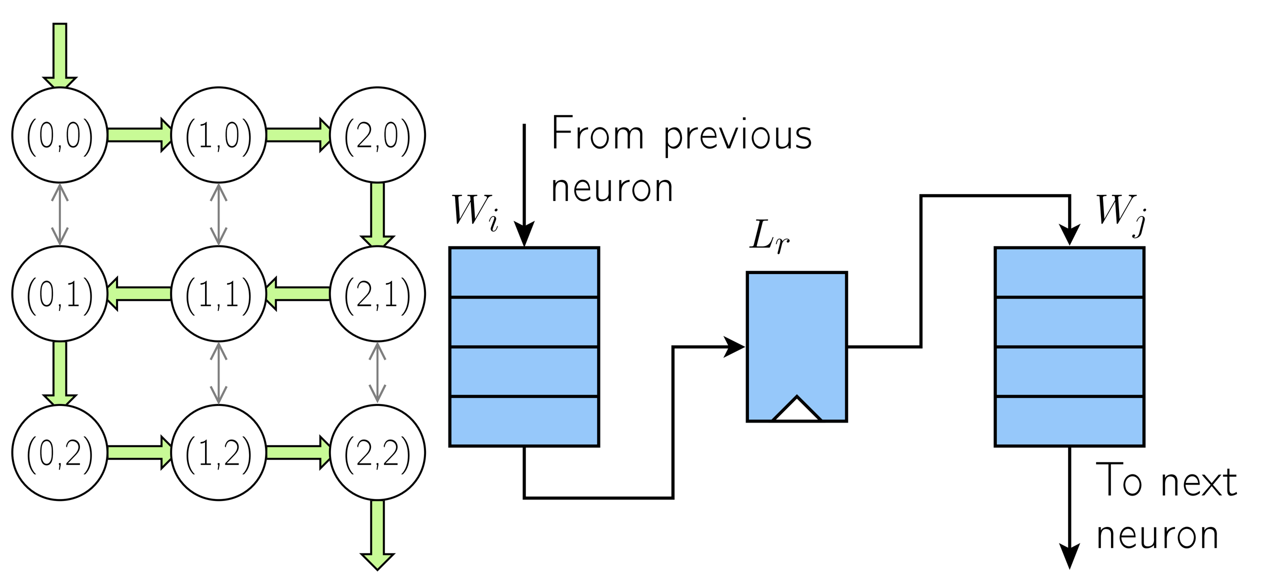
Hardware Implementation

The neuron is composed of the input, the lateral and the output branches.

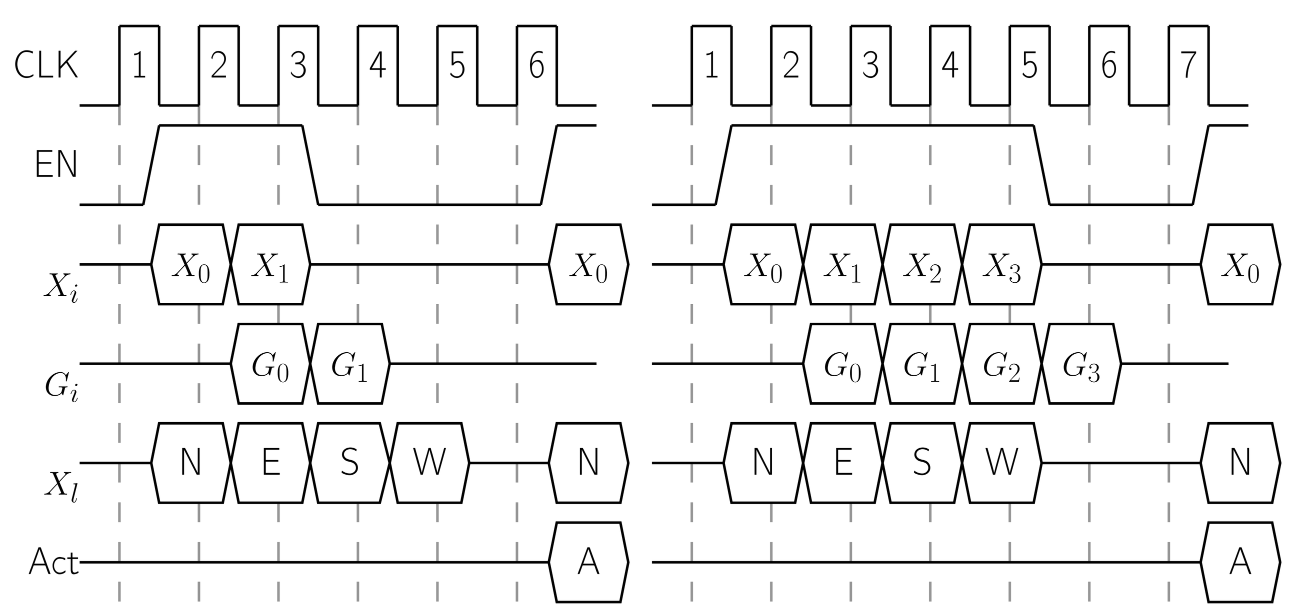


The **input branch** computes the formula described in eq. 2. The weights are stored in a shift register with a loopback responsible for the learning. Finally, the product is computed by an iterative multiplier.

The **lateral branch** computes the formula described in eq. 3. The synchronization of the weights and the lateral input selection is done by a simple 2-bit counter. The $y(t)$ term is then computed by a classical multiplier-accumulator. Finally the **output branch** is composed of an adder, an activation block and a register. The activation block is a simple saturation.



A **synchronization path** is created to configure some internal values.



DMAD-SOM Model description

This new model can be situated somewhere between the KSOM and the NF and is able to take local clustering decisions in a complete distributed way. It's behavior emerges from local decisions taken relatively to the activities of the neurons and their neighbors. It is then named "Distributed Multiplicative Activity Dependent Self Organizing Map" (DMAD-SOM).

$$P(t) = X(t) + Y(t) \tag{1}$$

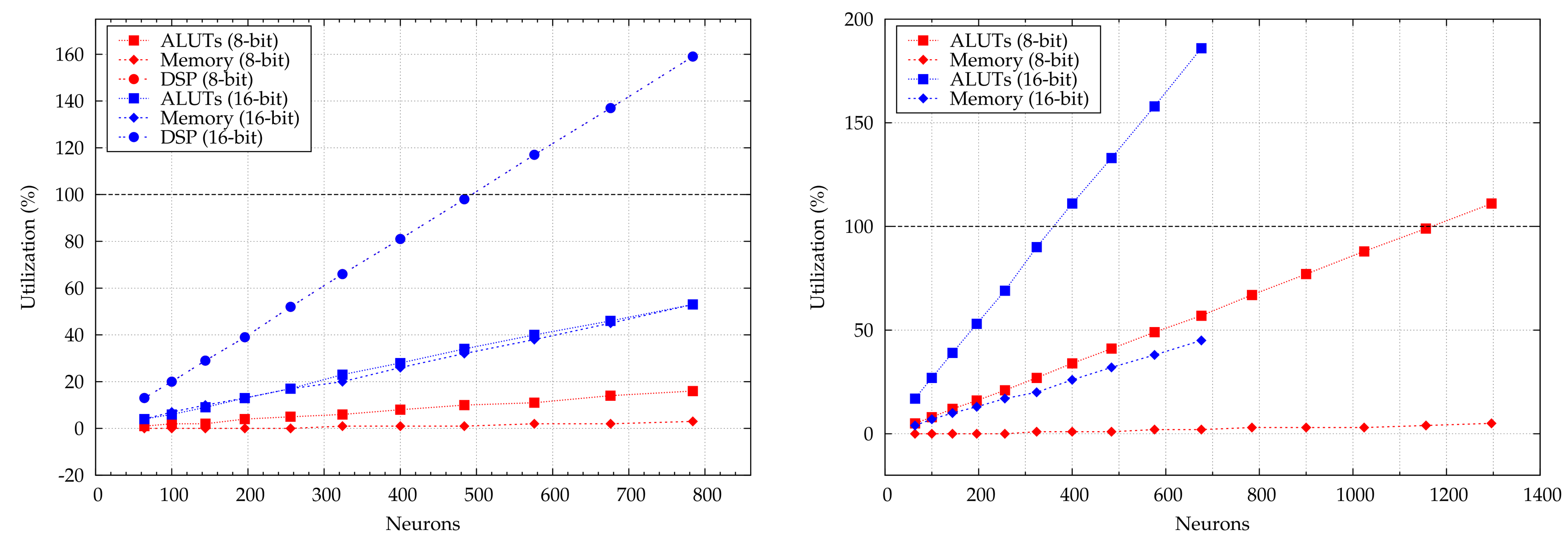
$$\Delta X(t) = -X(t) + \prod_i e^{-(w_i(t)-x_i(t))^2} \tag{2}$$

$$\Delta Y(t) = -Y(t) + \sum_j (W_j(t)U_j(t)) \tag{3}$$

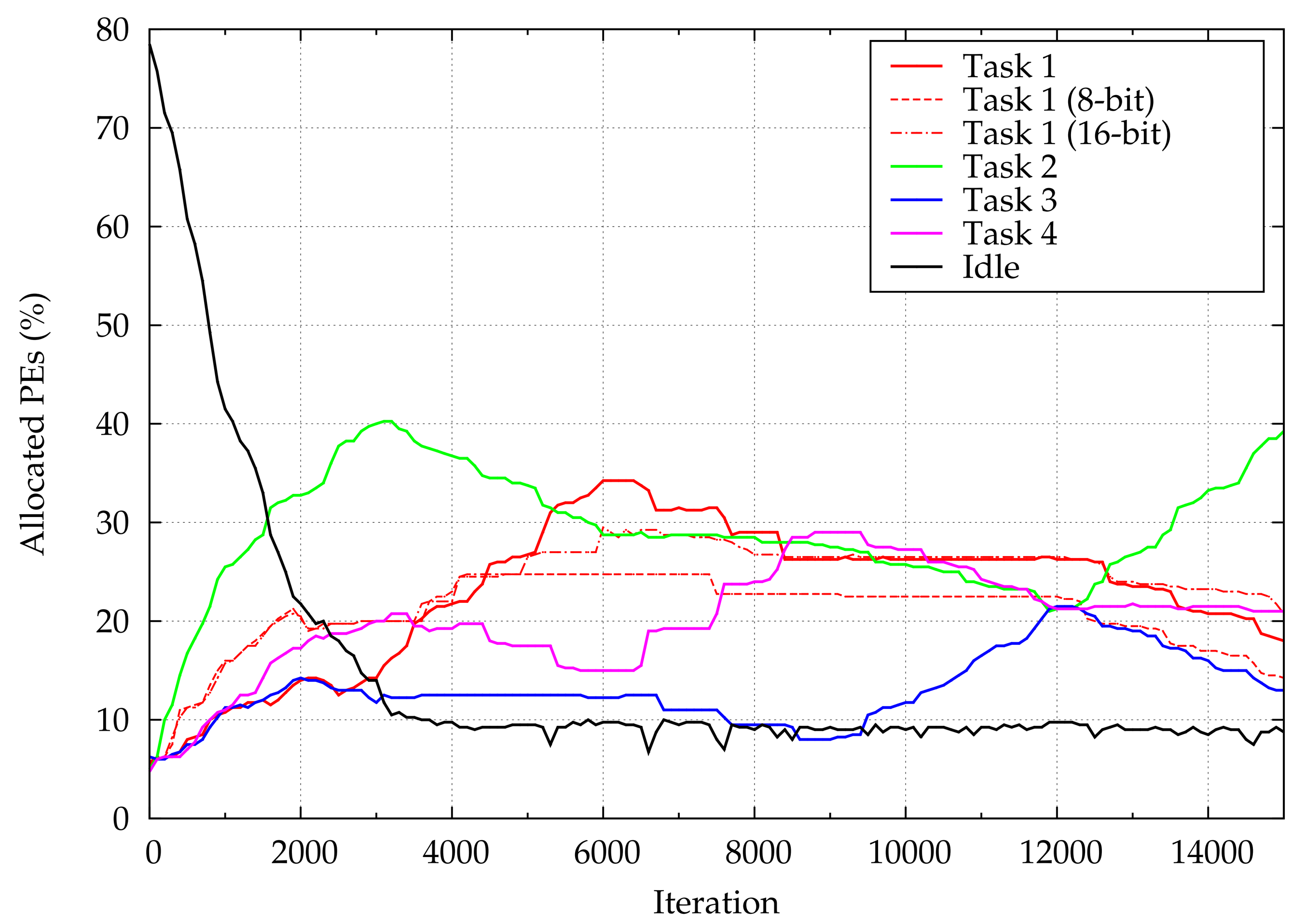
$$\Delta w_i(t) = l \times U(t)(x_i - w_i) \tag{4}$$

Results

We explore the impact of some parameters on the resources consumption. This exploration was done on a brand new Altera Stratix V 5SGSMD8N3F45 which is composed of $512k$ ALUTs, $50M$ memory bits and 1963 DSPs.



In this experiment the whole map learns the relative *richness* of each input in order to adapt the size of the processing clusters associated to 4 tasks. The number of PE allocated to Task 2 grows higher than the others from iteration 0 to 3000, reducing the number of non-allocated (Idle) PE. The size of each cluster then varies according to time. But, due to the memory effect of the DMAD-SOM, the cluster C_2 slowly goes down to the real stimuli value (0.2 at iteration 12000). The RTL results are correlated with the simulation ones with a difference resulting from the fixed-point encoding precision (dashed curves).



This behaviour can be easily obtained in a centralized way. The main contribution is to solve the problem in a bio-inspired and totally distributed way.

References

[1] Laurent Rodriguez, Benoît Miramond, Imen Kalboussi, and Bertrand Granado. Embodied computing : self adaptation in bio-inspired reconfigurable architectures. In *proc. of 19th Reconfigurable Architectures Workshop*, page 6, june 2012.

[2] ANR SATURN team. The SATURN project - Self-Adaptive Technologies for Upgraded Reconfigurable Neural computing, french research agency (anr) http://projet-saturn.ensea.fr.