# Elements of Arithmetic in Spiking Neural Nets: extended abstract

Andrew Carnell and Daniel Richardson,
Computer Science, Bath University, Bath BA2 7AY, UK
email: csparc@bath.ac.uk

June 11, 2004

## 1 Extended Abstract

We are concerned to show how *explicitly* to construct (rather than train) recurrent spiking neural networks which compute given arithmetic functions.

There are simple spiking neural networks which support periodic oscillations. For example, a single neuron with an excitatory time delayed link to itself will spike periodically if the synaptic efficacy is sufficiently high. Two such periodic subsystems with the same parameters may oscillate with the same periods but be out of phase. If we regard the neurons as deterministic, phase differences between two equiperiodic subsystems will persist. These phase differences can then be used to represent numerical quantities. It is then possible to represent integer variables, as arrays of matched pairs of oscillators, $(A_1, B_1), \ldots, (A_n, B_n)$, together with a switch $(A_0, B_0)$, to record sign. For each $i$, $A_i$ and $B_i$ have the same parameters, and the same periods; however $A_1, \ldots, A_n$ all have different periods. The array will represent zero if every pair $(A_i, B_i)$, for $i = 1...n$ is in phase. We consider $A_1, \ldots, A_n$ as negative, and $B_1, \ldots, B_n$ as positive.

We can also represent the increment and decrement operations, $X := X + 1$, $X := X - 1$. To increment the variable, we give a standard excitatory impulse to all the positive oscillators $B_1, \ldots, B_n$. The impulse advances the phase of all the positive oscillators. It is synchronised with the spiking of the positive oscillators so that the amount of phase advance does not depend on when the increment is done. The number $n$ is represented by giving $n$ increments starting from zero. Decrements are done by the same operation applied to the negative oscillators.

We give examples, showing increment of a matched pair of oscillators, using leaky integrate and fire neurons and the CSIM language.

It is possible, in general, to represent the $if$ construction, the *while* construction, and some other programming language constructions, *Seq*, *Par*, and *Alt*, as in Occam. In this way it is possible to represent addition, subtraction and multiplication of integers in a spiking neural net. We show how to construct a spiking neural net to carry out any computation determined by any given program in a certain simple programming language.

We do not claim that biological systems do arithmetic, or do it in this way. We do show however that computations based on phase differences of oscillators can have unlimited complexity.