Bio-Inspired Computing Architectures

Gianluca Tempesti[1] Andy Tyrrell[2]

1. Logic Systems Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland
2. Intelligent Systems Research Group, Department of Electronics, University of York, YO10 5DD, UK.

Motivation

In conventional top-down design of electronic circuits, we begin with an algorithm and a problem specification that are gradually broken down into tractable pieces. We then design the pieces and how they interact, and juxtapose them to obtain a solution. We design almost everything, including computer programs and silicon chips, in this way. This approach, however, places enormous constraints on the nature and interactions of the parts, and severely restricts the ways that we can solve our design problem. By definition, top-down design cannot exploit that feature of living systems typically responsible for their complexity: *emergence*. And the traditional design process *cannot be scaled up to systems with much more complexity than we have at present.* The complexity of living systems is apparent both in their sheer numbers of parts (10^{13} cells in a human being, for example), and in their behaviours: some organisms (e.g. plants, salamanders, hydra) are capable of *regeneration*; organisms are *robust* to damage; organisms are highly *dynamic* (cells are dying and being replaced all the time, proteins are being synthesized and broken down), which gives them enormous adaptive potential; organisms exhibit *developmental homeostasis* (the ability to develop correctly despite assaults from the environment or genetic mutation) and *developmental plasticity* (the ability to make different and appropriate phenotypes in different environments).

All these properties, highly desirable for silicon-based circuits, are crucial for nano/molecular-scale electronics, whose complexity and fragility makes them almost more similar to biological entities than to conventional circuits. Unfortunately, we do not currently know how to design systems with such properties. Individual organisms achieve complexity through *development*; developmental programs are devised through *evolution*. In the project, we shall try to implement analogues of these properties into a new paradigm of software and hardware construction that will respect the known constraints of molecular-scale electronics.

While it is possible that nano/molecular-level electronics will eventually revolutionize the way we think about computing, it is not reasonable to assume that the algorithms designed for silicon-based electronics in the last 50 years will be set aside once nao/molecular-scale devices are introduced. It is therefore of extreme interest to find a way to connect the new technology to the conventional design flow.

More precisely, to tackle the constraints of nano/molecular-scale electronics we should draw inspiration from another complex system based on molecular components: a multi-cellular biological organism. Organisms, in fact, have to meet (albeit on a different scale and in a different context) many of the same constraints that molecular-scale electronic devices will face:

- To develop systems that will be capable of exploiting the staggering amount of components available within nano/molecular-scale devices, we need to consider how to deal with complexity to a much greater degree than we do today.
- Conventional computing architectures rely very strongly on synchronous operation. Achieving this kind of operation on nano/molecular-scale devices does not seem

possible, based on our current knowledge of the structure of such devices. Biological organisms are completely asynchronous systems.

- It has become apparent that, for the foreseeable future at least, the design of fault-free nano/molecular-scale devices will not be feasible. In fact, it is now generally accepted that the density of faults within this kind of devices will be much greater than for silicon-based circuits. Robustness has therefore become a crucial aspect in the modelling of computation based on molecular components.

In general, then, the challenge that we face is the exploration of innovative, bio-inspired approaches that will allow us to manage many unreliable computational units communicating asynchronously and to design complex software applications capable of exploiting these highly-distributed architectures.

Approach

The core of this kind of approach is the application of a bio-inspired approach to the definition of a useful architecture for computation in nano/molecular-scale devices. A possible source of inspiration is the development and structure of multi-cellular organisms. In this context, it is possible to define a hierarchy of increasing complexity ranging from the molecule to the organism, seen as an array of computational cells (processors). In keeping with the biological inspiration, these computational cells should all be identical (in analogy to the *stem cells* used by organisms as a source of undifferentiated cells) and store the same program (the genome of the organism).

Such an approach must be centred around three correlated axes: the development of the architecture of our cells and of the connection network that allows them to communicate, the design of software tools to allow complex algorithms to be mapped on the cellular array, and the definition of a useful and realistic model for the kind of nano/molecular component that could be used to implement the architectures.

Processor architectures

Focusing on the architectural aspects of the approach, drawing inspiration from the structure and operation of multi-cellular organisms for the design of highly-complex computing systems for molecular-scale devices can easily be justified when the features of such organisms are considered. To mention just a few examples:

- The bio-inspired concept of seeing a system as an array of identical cellular processors, each storing a full description of the organism in the form of a genome, implies considerable spatial redundancy but simplifies enormously fault tolerance through reconfiguration.
- Cellular differentiation and developmental plasticity in complex biological organisms can be exploited to design scalable and evolvable developmental hardware that can adapt itself to the algorithm to be executed.
- Examining how cells are created and destroyed in living organisms, both during the cicatrisation and regeneration processes and during growth, can provide invaluable hints on how to implement fault tolerance and developmental homeostasis, key requirements for any circuit based on nano/molecular-scale components.
- In nature, cell communication/signalling mechanisms (intra, inter, synaptic, hormonal, etc.) are able to implement complex communication over long distances with relatively slow asynchronous components and fully meet the known constraints of molecular-scale devices.

A processor architecture designed to implement bio-inspired systems might be centred around three main areas:

- A *developmental subsystem*, which stores the genome program and contains the sub-circuits required to implement cellular differentiation, evolutionary approaches, and self-repair.
- A *computational subsystem*, which contains the dedicated units used to implement the computation required by the algorithm (for example, a DCT for the JPEG algorithm).
- A *communication subsystem*, which will represent a flexible programmable medium able to emulate various biological interactions (e.g., chemicals, diffusion laws, dendritic connections).

An important requirement for the processors will be *flexibility*, that is, the ability to adapt both in functionality and in structure to the requirement of an application. A key challenge in this context is then to design an architecture that can be integrated in a *design environment* allowing the user to exploit its flexibility to quickly realize arrays of cellular processors dedicated to the implementation of a specific algorithm using libraries of pre-defined components.