HARDWARE PLATFORM FOR COEVOLUTIONARY DESIGN

Radek Hrbáček
Master Degree Programme (2), FIT BUT
E-mail: xhrbac01@stud.fit.vutbr.cz

Supervised by: Michaela Šikulová
E-mail: isikulova@fit.vutbr.cz

Abstract: Evolutionary algorithms are well established in the field of optimization problems; their usage as design methods comes to the foreground thanks to the inovation introduced into the design process. Coevolutionary algorithms are natural extension to the evolutionary algorithms in which the fitness of an individual depends on its interactions with other individuals. Applications of these algorithms include electronic circuit design, image filter design, machine learning, symbolic regression and many others. This paper deals with the design of a hardware platform for accelerating coevolutionary design.

Keywords: evolutionary algorithm, coevolution, evolvable hardware, FPGA, MicroBlaze

1 INTRODUCTION

Conventional design approaches have reached the limits in many engineering areas because of growing complexity of new devices. Biologically inspired principles can be encountered increasingly, applied either during the design process or in pursuit of creating new computing platforms (e.g. DNA computing) [2]. An evolutionary algorithm (EA) is a model of biological evolution. Individuals in a population are subject to selective pressure leading to formation and development of functional traits. Advantageous traits increase *fitness* of the individual (i.e., the reproductive capability) and its genome is more likely passed on to next generations. The techniques belonging to EAs (e.g. genetic algorithms, genetic programming and others) differ in their use of genetic operators applied to the *chromosomes*, in the nature of the problem and in implementation details.

Evolutionary design based on EAs is a very computationally intensive design method. Furthermore, the convergence of the fitness value is often not sufficiently guaranteed due to getting stuck in a local extrema. These and other issues can be effectively suppressed by introducing *Coevolutionary Algorithms* (CoEAs) whose principles are the same as for traditional EAs except for fitness computation [3]. Unlike EAs, fitness computation in CoEAs is based on interactions with other individuals and thus the fitness value of one individual can change over time. Populations of either one or more species can be subject to the evolution; in case of two or more populations of the same species, individuals of particular populations are split by a *barrier*. In *compositional* problems the quality of a candidate solution is determined by cooperation of its components. On the contrary, *test-based* problems have commonly two populations – population of candidate solutions and population of tests. The candidate solutions are scored according to the tests and vice versa. There are different approaches to fitness interaction like competitive coevolution, coevolution of fitness predictors or others [3].

2 EVOLVABLE HARDWARE

With the expansion of reconfigurable hardware components, a new field of study called *evolvable hardware* has emerged, where programmable logic devices (mainly FPGAs) are in the spotlight of the research. Other evolvable platforms include programmable analog devices, reconfigurable antennas,

optical systems (lenses and mirrors) and many others (often highly non-conventional) [2]. Evolvable hardware combines reconfigurable hardware with evolutionary algorithms in two different ways. The evolution can be used to design the system before deployment or it can adapt the system during active usage. In case of *adaptable hardware*, the built-in evolution helps the system to respond to changes in the environment or to recover from failures (e.g. due to radiation impact). The evolution can take place either on the actual hardware (intrinsic evolution) or on PC by simulating the reconfigurable device (extrinsic evolution). One of the most popular EAs used in combination with HW is *Cartesian Genetic Programming* (CGP) – a form of Genetic Programming in which the candidate program is represented by cartesian grid of nodes with configurable interconnections and functions [3]. Hardware implementation of CGP is straightforward and the reconfiguration can be based on partial internal reconfiguration [1] or *Virtual Reconfigurable Circuit* (VRC) [2].

3 HARDWARE PLATFORM DESIGN

Evolutionary algorithms are very time-consuming, the most demanding task is commonly the fitness computation. Supposing evolutionary design of digital image filters using CGP, the fitness function is based on filtering an input image and computing a chosen metric (e.g. MDPP, MSE, PSNR). By introducing coevolution one can reduce the set of pixels that the fitness is computed on – only a subset of original image is used, this subset is subject to a separate evolutionary process. Further acceleration can be done by means of task or data parallelism, while the best throughput can be achieved using custom hardware.

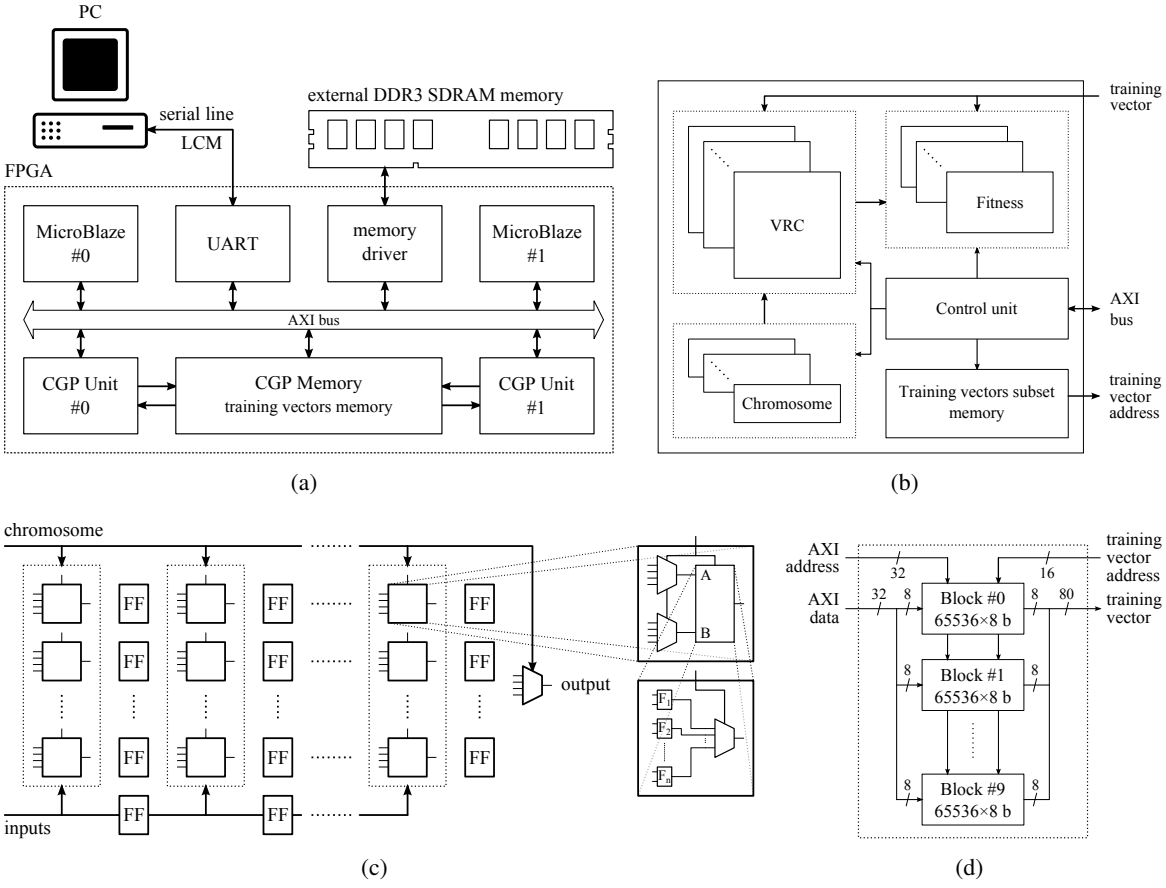

Figure 1: (a) Hardware platform scheme, detailed architecture of (b) CGP Unit, (c) Virtual Reconfigurable Circuit and (d) CGP Memory.

By taking the nature of the problem into account, the following requirements on the hardware platform arise – two evolutionary processes need to be running in parallel with the ability to communicate with each other and the fitness calculation should be as fast as possible. Currently, two suitable target platforms are available – conventional FPGAs and a combination of a processor and programmable logic (e.g. Xilinx Zynq All Programmable SoC [1]). Due to the focus on fitness calculation acceleration, a standard FPGA was chosen as the more flexible option. Figure 1 shows the proposed hardware platform architecture. The system consists of two MicroBlaze soft processors supplemented by two independent acceleration units (CGP Units) and training vectors memory (CGP Memory), all components are interconnected by the AXI bus and additional memory channels. Communication with a service application running on a PC is performed through serial port (UART) and LCM communication library. CGP Unit includes a set of VRCs and fitness units and each slice is able to process one training vector per clock cycle thanks to full pipelining. The remaining steps of the evolutionary process (individuals manipulation, communication) take place on the MicroBlaze processor. The CGP Memory component is designed to achieve very high throughput, one write port (connected to the AXI bus) and two read ports enable to supply both CGP Units with data.

The evolutionary design is going as follows. Initially, original and noisy images are transfered to the DDR3 memory, training vectors (each vector consist of chosen, e.g. 3×3 or 5×5, pixel neighbourhood from the noisy image and one pixel from the original image) are put together and copied to the CGP Memory. After that, the design process is initiated – populations of candidate filters and training vectors subsets are created, chromosomes are transfered to the CGP Units and the fitness computation is run. During the computation, it is possible to transfer next chromosomes on the background (all chromosome registers are shadowed). Thanks to this overlap, the hardware utilization grows. Finally, when the evolution is completed, the best individual's chromosome is sent to the PC.

4 CONCLUSION

This paper presents an FPGA based platform allowing very efficient coevolutionary design. The platform utilizes MicroBlaze soft processors in combination with acceleration units. The design was verified on Xilinx Virtex 6 FPGA, working frequency 100 MHz has been achieved with up to 2×5 parallel VRCs operating simultaneously. This massive parallelism brings great throughput and thus a significant speedup of coevolutionary algorithms. Further work will exploit the possibilities of this platform to design innovative non-linear image filters, for example for impulse noise removal.

ACKNOWLEDGEMENT

This work was partially supported by the BUT FIT grant FIT-S-11-1 and the research plan MSM 0021630528.

REFERENCES

[1] Dobai, R. and Sekanina, L. Towards Evolvable Systems Based on the Xilinx Zynq Platform. *2013 IEEE International Conference on Evolvable Systems - From Biology to Hardware*, ICES 2013.

[2] Sekanina, L., Vašíček, Z., Růžička, R. et al. *Evoluční hardware: Od automatického generování patentovatelných invencí k sebemodifikujícím se strojům*. [b.m.]: Academia, 2009. 328 p. Edice Gerstner. ISBN 978-80-200-1729-1.

[3] Šikulová, M. and Sekanina, L. Acceleration of Evolutionary Image Filter Design Using Coevolution in Cartesian GP. In *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*. [b.m.]: Springer Verlag, 2012. P. 163–172. LNCS 7491. ISBN 978-3-642-32936-4.