Digital Circuit Design using Evolvable Hardware

A.Sumathi,
Department of Electronics and Communication,
Adhiyamaan college of Engineering, Hosur
Email: sumi_saravana@yahoo.co.in

Abstract

This paper describes the application of intrinsic evolvable hardware to combinational circuit design and synthesis, as an alternative to conventional approaches. This novel reconfigurable architecture is inspired by Cartesian Genetic Programming and dedicated for implementing high performance digital image filters on a custom Xilinx Virtex FPGA xcvlOOO, together with a flexible local interconnection hierarchy. As a highly parallel architecture, it scales linearly with the filter complexity. It is reconfigured by an external genetic reconfiguration processing unit with a hardware GA implementation embedded. Due to pipelining, parallelization and no function call overhead. it yields a significant speedup of one to two orders of magnitude over a software implementation, which is especially useful for the real-time applications. The experimental results conclude that in terms of computational effort, filtered image signal and implementation cost, the intrinsic evolvable hardware solution outperforms traditional approaches.

1. Introduction

Evolvable hardware (EHW) applies techniques derived from Evolutionary Computation (Ee), i.e. Genetic Algorithms (GAs), Evolutionary Strategy (ES) and Genetic Programming (GP), to hardware design and synthesis. Early evolvable hardware experiments used simulations in fitness evaluation and only the elite chromosome was downloaded to the hardware, which was labeled extrinsic evolvable hardware. With the advent of Programmable Logic Devices (PLDs), such as Field Programmable Gate Arrays (FPGAs) and Reconfigurable Processing Units (RPUs) , it is possible for the implementation solutions to be fast enough to evaluate a real hardware circuit within an evolutionary computation framework; thus it is reconfigured for each member of the population. This is called intrinsic evolvable hardware.

Intrinsic evolvable hardware research is often restr by the limitations of the hardware resources. Most evol hardware experiments have been carried out extrinsic since the genotype abstraction does not impose limitation in its implementation. However, intr evolvable hardware performs more accurately in the world environment, especially unpredictable situations, as space exploration. Intrinsic evolvable hardware is more tolerant of inaccuracy within a particular simul environment.

Evolution has produced circuits comparable to t designed by human experts. Automatic design of r efficient circuits was examined in [I], which guaranteed functional correctness of the solution. In order to use hardware resource efficiently, Layzell [2] designed a new intrinsic evolvable hardware test platform in term tackling the present FPGA drawbacks. The unconstra nature of evolutionary techniques and their robustness in presence of faults was argued theoretically experimentally in [3]. Additionally, genetic programm proved to be able to design circuits concerning the reus substructure [4]. The filtering properties of the evolved arrays have been given much emphasis and examinatio [5].

Digital filters are evolved in this work, not only bec of their potential useful in research and industry, but because they provide a promising application area developing evolvable hardware. In the field of digital im processing particularly, a broad and disparate range applications using evolutionary computation may be fo in the literature, including the use of genetic algorithm the segmentation of medical resonance imaging scans [6 genetic program that performs edge detection on dimensional signals [7], the evolution of genetic program detect edges in petrography images [8], and the evolutio spatial masks to detect edges within gray scale images [9 is worth mentioning that Sekanina [10][1 I][12] achieved evolutionary design of digital image filters v virtual reconfigurable circuits in which Cartesian Gen Programming (CGP) was extended to a functional le Instead of CLBs and 1-bit interconnection wires, Configurable Functional Blocks (CFBs) and 8-bit data-paths are utilized, making a considerable improvement in performance. Digital image processing operations, such as image smoothing, edge detection, and image compression, have been carried out in an extrinsic EHW environment, which exhibits the potential of EHW in digital image operator design.

This paper presents an intrinsic evolvable hardware architecture, dedicated for implementing high performance digital image filters on a custom Xilinx Virtex FPGA xcv1000, together with a flexible local interconnect hierarchy. The novelty lies in the hardware design of the processing elements and the external genetic reconfiguration unit that is presented as the strategy for the intrinsic evolvable hardware execution. This highly parallel architecture provides reconfigurability and realization of a given circuit specification and scales linearly with the filter complexity.

In terms of time and resources available, the phenotype abstraction determined the choice of intrinsic or extrinsic implementation. Section 2 of this paper describes a high level phenotype abstraction that makes intrinsic evolvable hardware possible. In section 3, we employ a simple GA and configure the hardware using an integer string genotype. For the EHW hardware implementation, there are two distinct phases, the evolutionary design phase and the execution phase, which are described in sections 4 and 5 respectively. The experimental results in section 6 reveal that this evolutionary design of digital image filter outperforms conventional approaches. An interesting conclusion is then drawn in section 7.

2. Phenotype Abstraction

Evolutionary computation is sometimes able to improve or even replace human design of combinational circuits. However, this is usually achieved either through a computational effort that involves the sampling of a large number of individuals and the evolution of a great number of generations, or by devising new evolutionary techniques. For the latter, a sensible combination of the reconfigurable primitives and the evolutionary operators is essential to the success of intrinsic EHW design. The work in this paper employs intrinsic EHW by devising an array of compact processing elements and an external genetic reconfiguration unit. It outperforms human design in terms of computational effort and implementation cost.

We approached the problem using Cartesian Genetic Programming, which proved to be successful, not only in gate level evolution of three-bit adders, but also in functional level design of digital image filters [10]. In terms of image filter evolution that employs at least nine 8-bit inputs (for a 3x3 neighborhood) and one 8-bit output, it is almost impossible to achieve the 72 inputs and 8 outputs required in a conventional CGP architecture. We adopted the extended CGP architecture [10] and adapted it for our own purpose, as shown in figure 1. The inputs are labeled I0 through I8 and relate to the nine pixels of a conventional image processing 3x3 neighbourhood mask used to process grayscale images of 256x256 pixels (8 bits/pixel). The Processing Elements (PEs) are indexed from the top left (labeled nine), row-wise, and then column-wise finishing with the output PE (labeled 33). The two inputs of every PE can be connected to one of the outputs from the previous *l* columns (where *l* is the level-back parameter, equal in this experiment to 2). Every PE executes a certain function from table 1, (depending on the function code configuration, *cfg3*) which is applied to its two inputs, *X* and *Y*.

Table 1. Function codes (*cfg3* index in brackets)

Code	Function	Code	Function
F0: 0000	X >> 1	F8: 1000 (3)	(X+Y+1) >> 1
F1: 0001	X >> 2	F9: 1001	X & 0x0F
F2: 0010	~ X	F10: 1010	X & 0xF0
F3: 0011	X & Y	F11: 1011	X | 0x0F
F4: 0100	X | Y	F12: 1100	X | 0x F0
F5: 0101 (1)	X ^ Y	F13: 1101 (4)	(X&0x0F) | (Y&0xF0)
F6: 0110	X + Y	F14: 1110	(X&0x0F) ^ (Y&0xF0)
F7: 0111 (2)	(X+Y) >> 1	F15: 1111	(X&0x0F) & (Y&0xF0)

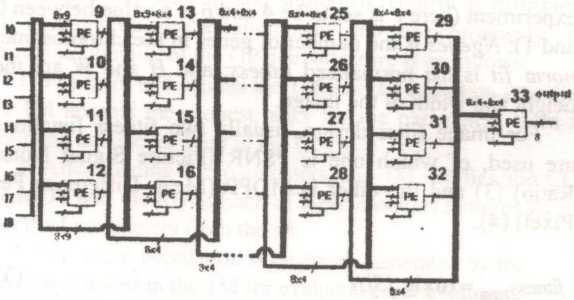

Figure 1. The Extended Cartesian Genetic Programming Reconfigurable Architecture

3. Genotype and Genetic Algorithms

The logical configuration of the circuit is defined by a set (or *chromosome*) of 25 integer triplets (or *genes*), one for each of the 25 PEs in the reconfigurable architecture shown in figure 1. The first two integers of each triplet represent the source of inputs to the PE (*cfg1* & *cfg2*) as labeled in figure 1. The third integer of the triplet (*cfg3*) indexes the function from Table 1 to be applied by the PE, as previously described. Figure 2 gives an example of a chromosome, describing the 4x6 array of the extended CGP architecture.