Literature Review of Evolutionary Hardware Application Based on Nios II and Neural Network

Liu Wei* and Chen Lijunb

Equipment Simulation Training Center, Shijiazhuang Campus of Army Enginnerring University, Shijiazhuang 050003, China

*Corresponding author e-mail: wellslaw@163.com, bclj_dxh@163.com

Abstract. The development and research status of the evolutionary hardware technology are analyzed and summarized. The characteristics of the existing evolutionary circuit design methods are analyzed and compared. The genetic algorithm and other evolutionary control algorithms are analyzed, and the design method of the virtual reconfigurable circuit is studied. An evolutionary circuit design method of "Nios II+VRC array" is proposed, which has a wide range of applications and can be evolved from PC to online evolution.

1. Introduction

Evolutionary hardware technology is a technology that builds compensatory mechanisms in a bionic way.[1-2] The basic idea is to use evolutionary algorithms to reconfigure and reconfigure reconfigurable logic elements in integrated circuit chips to make the system perform certain tasks. The system architecture, connection methods, and local functions will automatically evolve according to the changes in the environment to achieve dynamic adjustment. In modern warfare, weapons and equipment are confronted with a very complex electromagnetic environment. Evolving hardware technology can enable equipment to achieve "self-adaptive, self-diagnosis, and self-repair" in a complex environment, providing a kind of protection for weapons and equipment from a deeper level. A very efficient solution[3-5].

The idea of evolving hardware originated from the basic concept of "self-repairing and self-replication" machines proposed by von Neumann in the 1960s. However, it was limited by the technical conditions at the time and could not be achieved. Until the 1990s, with the emergence of evolutionary computing and large-scale programmable logic devices, this concept was gradually realized[6]. In 1992, Tetsuya Higuchi proposed a method to reconfigure the hardware circuit structure based on genetic algorithms to implement the evolutionary hardware[7]. In 1993, Daniel Mange's research was to apply bionic hardware mechanisms to routine hardware self-healing. Afterwards, there were many researchers abroad engaged in the study of evolutionary hardware[8-12]. Garrison W. Greenwood summarizes the advantages and disadvantages of evolutionary hardware, and implements fault tolerance through adaptive research. Evolutionary hardware is used as the cornerstone for implementing fault-tolerant system functions because it combines Evolutionary algorithms and reconfigurable hardware features[13]. Alicia Morales-Reyes systematically studied the adaptive cell genetic algorithm, and based on the Global Positioning System (GPS), built a fault-tolerant system platform and evaluated the impact of the genetic algorithm on the entire operating system. Patrick Rocke et al. studied the adaptability of the Field Programmable Analog Array (FPAA) and studied its...
application in the SNN (Spiking Neural Network) pole balance controller[14]. Yerbol Sapargaliyev studied the unconstrained evolution of the "QR" circuit for analog computing and proposed a method for combining unconstrained evolution with variable-length chromosomes, and applied it to the design of a circuit that implements the QR-simulated root circuit. Martin Trefzer et al. studied combinatorial sequential logic circuits and compared the differences in hardware resources between unconstrained evolution and constrained evolution. Jiri Jaros et al. studied and designed an evolutionary fault-tolerant scheme for cluster communications. Marcos Paulo Mello Araujo et al. studied the scheme of optimal state allocation for finite state machines based on quantum evolutionary algorithms, and summed up an evolutionary approach to solve the design of finite state machines[15].

China's research on evolutionary hardware started relatively late, and there is a certain gap compared with the level of research abroad, but its development is also very fast. At present, there are many universities and research institutes in China that are engaged in the research of evolutionary hardware. In the past two years, there has been a new development in China's evolutionary hardware. Because of the distributed flexibility and flexibility of spatial wireless small-satellite sensor networks, an adaptive decision function and weight factor have been proposed, and distributed networks have been obtained. The efficient algorithm. In addition, the study of digital sequential logic circuits, the design of third-order decomposition (input decomposition, output decomposition and state decomposition) approach, the entire circuit is divided into a number of small evolutionary sub-circuits to evolve complex sequential logic circuits.

In recent years, there are three kinds of evolutionary circuit design methods commonly used at China and abroad: an implementation method based on an Application-Specific Integrated Circuit (ASIC). The chip internal structure and configuration bitstream designed by this method are known and practical, but the production cost is high. The process is complex and difficult to implement; based on PC and FPGA implementation, this method is used to achieve slow operation and can not be separated from the PC; based on the Virtual Reconfiguration Circuit (VRC) implementation method, the basic idea is to generate the hardware description language. A virtual configuration bitstream layer implements the mapping relationship between the actual FPGA device and the virtual circuit structure. This method is easier to develop and has good flexibility and versatility.

In the implementation of evolutionary hardware based on VRC, a design scheme of "SOPC (System on Programmable Chip) + VRC" can be adopted. SOPC uses programmable logic technology to integrate the entire system into a single chip. It also has the following advantages: It has the characteristics of a system-on-a-chip, that is, a single chip completes the main logic function of the entire system, and is also a programmable system. On-line programming capabilities on the hardware, you can also modify the application program repeatedly, with good design flexibility. Nios II is a relatively good SOPC product line. In addition, the "SOPC+VRC" design scheme also has the advantages of short evolution time and independent operation, and can be implemented on-chip without the PC[16].

2. Evolutionary Hardware Based on SOPC and Neural Network Model

2.1 Nios II soft processor

At present, the most representative soft-core embedded system processors are the Nios II core of Altera Corporation and the MicroBlaze core of Xilinx. Before implanting the system, the user can use the development software Quartus II and SOPC Builder to build the peripheral system according to the design requirements, so that the embedded system fully meets the user's system design in terms of hardware structure, function characteristics, and resource occupation. Claim. As long as the resources of the FPGA allow, there is no limit to the number of Nios II cores that can be implanted in the same FPGA; in addition, the Nios II implantable FPGA family has almost no limitations. In this respect, Nios II is clearly better than Xilinx's MicroBlaze. In terms of cost, since Nios II is directly launched by Altera Corporation rather than a third-party product, users usually do not have to pay for intellectual property, and the cost of NiosII is only the FPGA logic resources they occupy.
2.2 Neural Networks and VRC

Artificial neural network is a parallel and distributed information processing network structure. The network structure is generally composed of many neurons. Each neuron has a single output. It can be connected to many other neurons. Connection paths, one for each connection path.

2.2.1 Neural network and circuit network mapping

By analyzing the structural characteristics of neural networks and circuit networks through comparison, the three types of peer mapping relationships between them are clearly defined: basic unit mapping, network topology mapping and hierarchical structure mapping.

(1) Basic unit mapping

In a neural network, neurons receive signals transmitted from other cells through synapses on the dendrites and are processed by the cell body. The axons then transfer the results to other cells to complete signal reception, processing, and transmission. Or stored procedure. The basic unit in the circuit network receives the input signal through the connection line. After the logic processing, the processing result is transmitted to other units again, and a similar signal receiving, processing, transferring or storing process is completed. In this way, the following basic unit mapping relationships can be summarized between the neurons in the neural network and the basic units in the circuit network, ie, the logic unit corresponds to the neuron cell body, and nonlinear signal processing is realized; the connection between the units is realized. Dendrites and axons (synapses) correspond to neuronal cells, enabling signal transmission.

(2) Network topology mapping

The neural network is the collective behavior of a large number of neuronal cells. It is not a simple addition of the behaviors of each unit, but functions as an overall network. Circuit networks and neural networks have similar network forms. The similarities between the two are summarized as follows:

a) Both of them are all highly interconnected by a large number of basic units with local processing capabilities;
b) The network functions are determined by the basic unit states of the network and the connection relationships between them;
c) Both of them adopt a distributed network structure;
d) Both of them have adopted parallel work.

(3) Hierarchical structure mapping

Neural networks and circuit networks also have a hierarchical mapping relationship. First, neural
network neurons and circuit network basic units have equivalence. In a neural network, neurons can perform both computational and storage functions, and they can form a unified system through connections. In the circuit network, the basic unit can also accomplish similar functions. The position in the entire network and the position of the neuron in the neural network have certain equivalence. Second, the partial network of the neural network and the functional circuit in the circuit network have equivalence. The neural network contains many local networks with different morphological functions. They are composed of neurons with similar or different properties and realize the local area's features. Circuit elements in a circuit network can also form partial circuits with specific logic functions after they are interconnected. The partial circuits are then interconnected to form a circuit network. The hierarchical status and functional role of local networks are very similar to those of local functional circuits. In summary, the circuit networks, functional circuits, logic units and neural networks, local networks, and neurons have a hierarchical mapping relationship as shown in Fig. 2.

![Mapping Diagram](image_url)

**Fig.2.** The hierarchical mapping between neural networks and circuit networks

Neural network learning algorithms have the disadvantages of slow training speed, easy to fall into local minima, and weak global search ability. However, evolutionary algorithms do not require the continuity of the objective function, and the population operation method is used to search globally and search globally more easily. Excellent solution. Therefore, the use of evolutionary algorithms to learn neural network structure, connection weights, transfer functions, learning rules, etc. has caused more and more attention, and has achieved remarkable results.

From the discussion of the peer-to-peer mapping relationship between neural networks and circuit networks, it can be known that circuit networks and neural networks have similar organizational structures and information processing modes. Therefore, the circuit network can be regarded as a special neural network, and the design process of the circuit network is regarded as a neural network training learning process in which the network structure, connection weights, and transfer functions are all variable. In this way, a large number of mature network structures and description methods in the artificial neural network theory can be used to mathematically represent the circuit network, and the evolutionary algorithm can be used to implement functional self-learning, self-organization and self-adaptation of the circuit network, with the help of network internal redundancy. The self-repairing of partial failure of the circuit is realized, thereby providing an effective way to establish a fault-tolerant and highly reliable circuit system using low-reliability circuit elements[17].

### 2.2.2 Virtual reconfigurable circuit

The virtual reconfigurable circuit is a relatively simple reconfigurable circuit model built by researchers on a common commercial FPGA, and the circuit evolution is studied on this basis. The basic idea is to generate a virtual configuration bitstream layer through the hardware description language, and then generate specific circuits through the FPGA software development system and map it to the actual FPGA device. For gate-level hardware evolution, its reconfigurable basic unit usually selects a set of basic logical function sets for a specific target. Using a virtual reconfigurable method requires the creation of a reconfigurable circuit array inside the FPGA, which is a very regular matrix in the design. Although the configuration bitstream is downloaded to the FPGA, the designer does not know how the circuit array is connected and implemented inside the FPGA, but it is conceivable that the circuit array has to perform the corresponding functions. The array must have been established within the FPGA. The virtual reconfigurable circuit is different from the traditional reconstruction. It
does not involve the encoding and modification of the configuration bit stream file in the reconstruction process, allowing the user to build a reconfigurable circuit that he needs. Thus, it has good flexibility and versatility.

When the FPGA loses the power, its internal configuration is immediately lost. The reason is that the level of BRAM has disappeared, so it does not have any function. For VRC, its internal functions are also maintained or changed by the level, so the effect that can be achieved by reconfiguring the FPGA can also be achieved by reconfiguring the VRC. Of course, there is a difference between the two. This difference is mainly reflected in the configuration of the FPGA with bit stream. The configuration is global. Configured in a virtual reconfigurable manner, the configuration is a very small part of the FPGA. Therefore, the FPGA-based configuration based on virtual reconfigurable implementation actually implements some of the functions of the FPGA with bitstream configuration. This is a basic principle of VRC-based FPGA hardware configuration. There are three main ways of circuit evolution of virtual reconfigurable circuits:

1. Both virtual reconfigurable structures and genetic programs use hardware description language design. The advantage of this method is that the evolution runs fast. When the project actually needs it, the code can be transplanted to the CPLD to achieve mass production. However, the development is more difficult, and the designer's hardware description language capability is very high.

2. Use Matlab or Visual Stdio development platform to design application software to run genetic algorithms and FPGA communication to evolve VRC. The advantage of this method is that its design is simple. The disadvantage is that it needs to interact with VRC for a large amount of data during evolution. Communication leads to a slower evolution. What is more important is that this method cannot be run independently from the computer[18].

3. "SOPC+VRC": SOPC runs a genetic algorithm that dynamically reconfigures VRCs that are in the same chip. The required functional circuits are implemented by changing the internal connection structure of the VRC. The advantage of this method is that the evolution time is short and can be run independently. The disadvantage is that it takes up more resources.

3. Evolutionary Hardware based on Genetic Algorithm

The evolutionary hardware is based on the intended circuit or circuit function as a design goal, the internal structure is represented by a binary chromosome, and a series of operations of the genetic algorithm are used to find the specific structure and function corresponding to the expected circuit.

The reconfigurable hardware accepts the control data stream converted by the binary gene string, so that the functions and connections of each unit in the circuit can be determined to form a specific circuit structure. The fitness evaluation of the circuit compares the specific circuit produced with the expected target circuit and determines whether the evolution process should continue. This step is done by calculating the individual's fitness. The evolutionary hardware loop process continuously generates new individuals based on the fitness evaluation of the circuit, namely a new binary gene string, until the circuit structure of the new individual representation satisfies the required fitness[19].

According to individual evaluation methods, circuit evolution can be divided into external evolution and internal evolution. Internally evolved, each individual code generated by the evolution algorithm must be actually downloaded to the programmable device to generate a specific circuit form, and then input the test vector to evaluate the individual performance and calculate the fitness based on the output response of the circuit. According to different carriers, internal evolution can be further divided into the evolution between plates, the evolution of plates, and the evolution of films. The external evolution model requires the establishment of a programmable device simulation model, and software simulation is used on the computer to obtain the fitness information of the coded individual.
4. Feasibility of implementing Evolving Hardware based on Nios II and neural network method

As shown in fig. 3, the evolutionary circuit based on virtual reconfigurability circuit mainly includes two parts: a Nios II soft core and a VRC array. Programmable cells (PE cells) in the VRC consist of configuration registers, multiplexers, and basic logic elements. During the evolution, the evolution algorithm is run through the Nios II soft core, and each evolution result is allocated to the VRC array. The VRC array generates the connection relationship of the PE units according to the evolution result. The adaptive evaluation of the generated VRC array results is performed, and the evaluation results are fed back to the Nios II soft core. The Nios II soft core adjusts the next-generation population in the algorithm according to the evaluation results. This iteration continues until the target circuit is obtained. After the configuration is completed, the VRC array works alone. When the external environment changes and the VRC does not work properly, it establishes communication with the Nios II soft core again. The Nios II soft core operation evolution algorithm reconfigures the VRC array until the requirements are met.

5. Conclusion

The evolutionary system based on the neural network model constructed by the VRC array and the Nios II soft core processor can realize the online evolution without the PC[20]. Only a small FPGA chip can be used to design a reconfigurable target circuit that meets the requirements. That would greatly improves the application range of evolutionary hardware technology.

Acknowledgement

In this paper, the research was sponsored by the National Natural Science Foundation of China (Project No. 61271152).

References

[1] Ran H H, Chen J, Yong S L, et al. Evolvable Hardware Experiment Based on Population Hybrid Genetic Algorithm[C]// International Conference on Mechatronics and Automation Engineering. 2017:472-480.

[2] Krishnaveni K, Ranjith C, Rani S P J V. Evolvable Hardware Architecture Using Genetic Algorithm for Distributed Arithmetic FIR Filter[M]// Artificial Intelligence and Evolutionary
Computations in Engineering Systems. 2017.

[3] Tao Y, Zhang L, Zhang Y. A projection-based decomposition for the scalability of evolvable hardware[M]. Springer-Verlag, 2016.

[4] Zhang, Junbin, Jinyan, et al. Optimal design of RTCs in digital circuit fault self-repair based on global signal optimization[J]. Chinese Journal of Aeronautics, 2016, 29(6):1774-1787.

[5] Alzahrani A, Demara R F. Fast Online Diagnosis and Recovery of Reconfigurable Logic Fabrics using Design Disjunction[J]. IEEE Transactions on Computers, 2016, 65(10):3055-3069.

[6] Salvador R, Otero A, Mora J, et al. Self-reconfigurable Evolvable Hardware System for Adaptive Image Processing[J]. IEEE Transactions on Computers, 2013, 62(8):1481-1493.

[7] Oreifej R S, Demara R F. Intrinsic Evolvable Hardware Platform for Digital Circuit Design and Repair Using Genetic Algorithms[J]. Applied Soft Computing Journal, 2012, 12(8):2470-2480.

[8] Kaufmann P, Glette K, Gruber T, et al. Classification of Electromyographic Signals: Comparing Evolvable Hardware to Conventional Classifiers[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(1):46-63.

[9] Walker J A. Overcoming variability through transistor reconfiguration: Evolvable hardware on the panda architecture[J]. Natural Computing, 2015, 49:153-173.

[10] Zhang Y, Smith S L, Tyrrell A M. Digital Circuit Design using Intrinsic Evolvable Hardware[C]// Evolvable Hardware, 2004. Proceedings. 2004 NASA/DoD Conference on. IEEE, 2012:55-62.

[11] Sekanina L, Vasicek Z. Approximate circuit design by means of evolvable hardware[C]// IEEE International Conference on Evolvable Systems. IEEE, 2013:21-28.

[12] Zhao W, Ren A, Tang S, et al. Evolvable Hardware System Based on FPGA[J]. Energy Procedia, 2011, 13(Complete):4451-4454.

[13] Ren A, Zhao W, Tang S, et al. Implement of evolvable hardware based improved genetic algorithm[C]// Seventh International Conference on Natural Computation. IEEE, 2011:2112-2115.

[14] Kavvadias N, Giannakopoulou V, Nikolaidis S. Development of a customized processor architecture for accelerating genetic algorithms ☆[J]. Microprocessors & Microsystems, 2007, 31(5):347-359.

[15] Huang H C. Intelligent Motion Control for Four-Wheeled Holonomic Mobile Robots Using FPGA-Based Artificial Immune System Algorithm[J]. Advances in Mechanical Engineering, 2013,(2013-11-5, 2013, 2013(5):589510-589510.

[16] Jin W, Chang H P, Chong H L. Implementing Multi-VRC Cores to Evolve Combinational Logic Circuits in Parallel[C]// International Conference on Evolvable Systems: From Biology To Hardware. Springer-Verlag, 2007:23-34.

[17] Glette K, Kaufmann P, Assad C, et al. Investigating evolvable hardware classification for the BioSleeve electromyographic interface[C]// IEEE International Conference on Evolvable Systems. IEEE, 2013:73-80.

[18] Li C, Lou J, Yu J, et al. The Design and Application of Evolvable Hardware IP Core[M]// Unifying Electrical Engineering and Electronics Engineering. Springer New York, 2014:1811-1818.

[19] Bu H, Chen L, Lai J. A LUT-based VRC model for random logic function evolution[C]// IEEE, International Conference on Asic. IEEE, 2009:1-5.

[20] Lukáš Sekanina. Evolutionary functional recovery in virtual reconfigurable circuits[J]. Acm Journal on Emerging Technologies in Computing Systems, 2007, 3(2):8.