Hardware Evolution Platform Reasearch Based on Matrix Coding CGP

Li Chong-cun[1], Xu Li-zhi[2], Song Xue-jun [1]*, Guo Zhen-xing[1], Liu Xiong[3]
[1] College of Physics and Information Engineering , Hebei Advanced Thin Film Laboratory , Hebei Normal University , Shijiazhuang, China
[2] College of Engineering and Physical Sciences, University of New Hampshire, Durham, NH, U.S.
[3] Marvell Technology Group Ltd., Santa Clara, U.S.
e-mail: lccmuzi@163.com, xlze81@gmail.com, sxj226@126.com

***Abstract*—Evolvable hardware is a combination of evolutionary algorithms and reconfigurable hardware, which can change itself structure to adapt to the living environment. Evolvable hardware possessed the characteristics of self-organization, self-adaptation and self-repair. The off-line evolution of digital circuits is similar to a simulation process, which lacks real-time capability and cannot generate actual circuits for every evolutionary digital circuit. The hardware online evolution platform is designed for evolution digital circuit based on Field Programmer Gate Array. Compared with off-line evolution, the platform can monitor the status of the designed circuit in real time, and it is easily to evolve a digital circuit for practical products directly. The multiplier circuit has obtained using the on online evolution platform combining based on matrix coded Cartesian Genetic Programming.**

Keywords- evolutionary hardware; matrix coding CGP; online evolution; hardware evolution platform;

I. INTRODUCTION

The human exploration field is continuous expansion with the development of science and technology. There are higher and higher requirements for reliable operation of the electronic circuits in extremely harsh environments. Evolvable hardware (EHW) [1-3] is a new type of hardware that combines evolutionary algorithms with programmable logic devices.

The evolution hardware takes the evolution algorithm as the search engine and the programmable logic device as the carrier for hardware reconfiguration. It possessed the characteristics of self-organization, self-adaptation and self-repair [4], adjusting internal structure, repairing device faults and realizing the reliable operation of the circuit system [5]. One of EHW key technology, circuit evolution design has been concerned by more and more people. Circuit evolution design uses evolutionary algorithms as search and optimization tools, programmable devices as its implementation carrier, without relying on prior knowledge and external intervention [6], compared with traditional circuit design methods.

The multiplier circuit has obtained using the on online evolution platform combining with matrix coded Cartesian Genetic Programming.

II. MATRIX CODING CGP

Cartesian Genetic Programming (CGP) is a type of graphs and efficient genetic programming method by Julian and Miller [7]. CGP uses real number coding [8] to represent the components of the circuit and the connection relationship in a two-dimensional grid structure. The chromosomal genes include functional genes, connecting genes and output genes, and the coding scheme is closer to the internal structure of the FPGA (Field Programmable Gate Array). The chromosomes are more easily to map circuit structure [5, 9]. CGP is beneficial to design evolution circuit by FPGA.

In Fig.1, there are N_i terminal inputs and O_i terminal outputs in the target circuit. As shown in Fig.1, the CGP matrix coding model consists of the programmable logic cell array of R rows and C columns, the output column and input column.

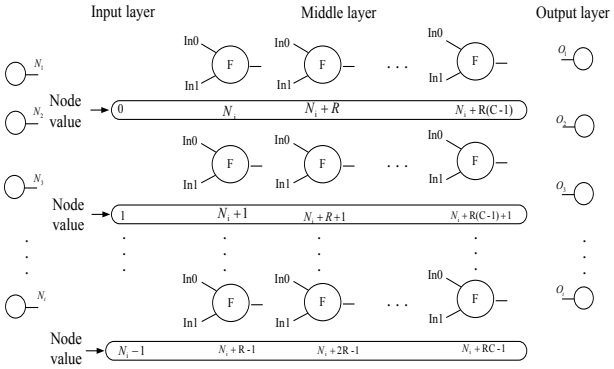

Figure 1. Matrix coding model of CGP

The input column composes input layer of the N_i terminal inputs. The terminal input is responsible for providing circuit input information as a node. The every terminal input dose not occupy gene bit. The output column composes output layer of O_i terminal output. The every terminal output is only an output without logic function, but occupies one gene bit. The middle layer consists of $R \times C$ programmable logic units. The every programmable logic unit is a two-input and one-output logic unit, which can perform different functions.

In Fig.1, all the nodes number from the input nodes to the middle nodes. The N_i terminal inputs number from 0 to $N_i - 1$. The $R \times C$ programmable logic units number from N_i to $(R \times C + N_i - 1)$.

As shown in Fig.2, a programmable logic unit corresponding to a gene node. The gene node consists of three genes, a functional gene F and two connection genes In_0 and In_1.

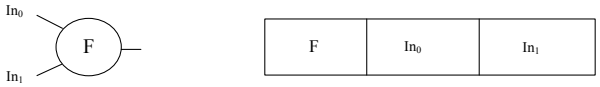

Figure 2. Programmable node and corresponding gene node

The gen F represents the function of the programmable node. The connection genes In_0, In_1 are inputs of the programmable node，equal the node number which provide signal to In_0 and In_1. The value of the connection genes is related to the matrix connectivity L [10]. Set L taking the maximum value, $L_{\max} = C$. The value of the connected gene in k column can be the number of all previous nodes before k column. The output gene value can connect to any previous node or input. Each programmable logic unit corresponds to a logic gate. There are six kinds of functional basic logic gates in Table 1.

TABLE I. FUNCTIONAL GENE CODING TABLE

Coded value	0	1	2	3	4	5
Logical name	AND	OR	XOR	NAND	NOR	XNOR
Logical symbol						

The chromosome is composes of middle layer and output layer. Suppose the number of output gens is m. The chromosome length can be calculated by the equation:

$$Lg=3\times RC+m \tag{1}$$

The experiments is accomplished for reconstruction a two-bit multiplier circuit online, set the size of the matrix coding model to be 4 terminal inputs, 4 terminal outputs and middle layer programmable logic units. The size of middle layer is 3 rows 5 columns of programmable logic cell arrays.

According to the chromosome equation, the chromosome length $Lg = 49$ and the connectivity $L = 5$. The randomly generated chromosome matrix coding model is shown in Fig.3.

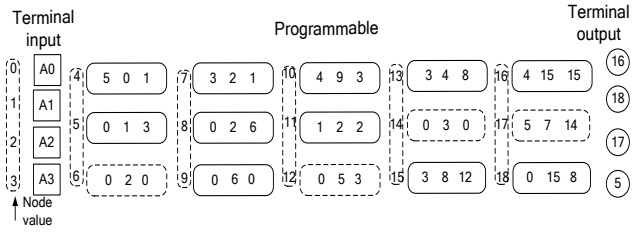

Figure 3. Matrix coding model of chromosomes

The four terminal output genes are 16, 18, 17, 5 in figure 3. It means that the four terminal output values are selected from the output of the numbered 16, 18, 17, and 5 that the programmable logic cells. Taking node of number 18 as an example, the functional gene of the node is 0, and the two connected genes are 15 and 8. The function of node 18 is logical AND, and the two inputs are respectively from the output of node 15 and 8. Using this method can decode the generated chromosome through evolution, the chromosome mapping circuit shown in Fig.4:

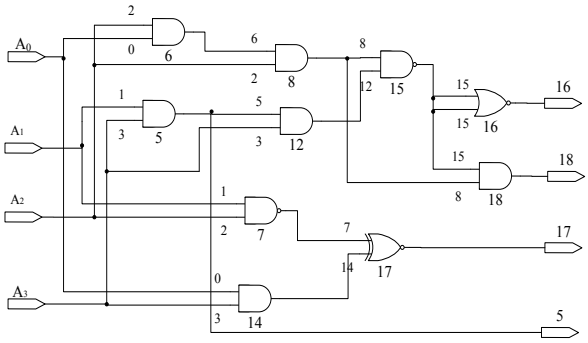

Figure 4. Mapping circuit of chromosome

III. HARDWARE EVOLUTION PLATFORM DESING

Evolutionary circuits can be divided into internal evolutionary and external evolutionary types [11]. Internal evolution is also called online evolution. The online evolution of the circuit design is the combination of PC and the hardware platform. Each new chromosome is downloaded to the hardware platform. The circuit is generated in the module, and then tested and evaluated online.

Online evolution requires high-powered, high-speed and high reliable reconfigurable hardware. External evolution is also called offline evolution. Offline evolution uses PC to establish evolution model simulates test and evaluation by software. All evolution is completed only on PC, which lacks real-time capability and cannot generate actual circuits for every evolutionary digital circuit.

The online hardware evolution platform is design, the module is shown in Fig.5. The platform contains three modules mainly, that is Nios II algorithm module, virtual reconstruction module and communication control module.

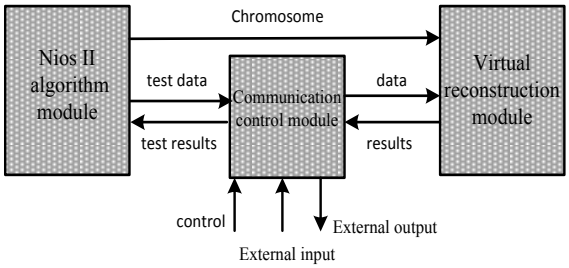

Figure 5. Hardware platform

A. Nios II Algorithm Module

The Nios II algorithm module is the Central Processing Unit of the whole platform. It runs evolutionary algorithm to generate chromosome and evaluate the circuit that maps the chromosome.

Nios II soft core processor is the core of the Nios II algorithm module. In the Nios II IDE development environment, the C language code of the evolutionary algorithm is downloaded into the Nios II soft core processor.

Each new chromosome generated by the Nios II algorithm module is send to the VRC module, and then the test data will be sent to test the chromosome mapping circuit. And the evolutionary algorithm will run according to the test results returned.

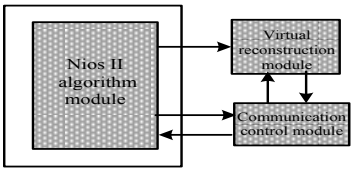

Figure 6. NiosII Algorithm Module

B. Virtual Reconstruction Module

Virtual Reconstruction Module uses a higher level representation to specify the target circuit architecture and functions. VRM is the second reconfiguration layer developed at the top of FPGA [12, 13]. Because the configuration bit-stream of FPGA is not public, the chromosome is unable to configuration the FPGA directly. VRM can realize reconfigure FPGA indirectly. The chromosome from Nios II algorithm module can configuration FPGA through VRM. The virtual reconstruction module is consist of PE (Programmable Element) block and output block.

1) PE block: The block implement chromosome mapping circuit. The Programmable Element block in the VRM is consisted of several PE units, in Fig.7. The functions and connections of PE units with each other can configuration by changing the configuration bit-stream of the PE units.

In effect, the chromosome mapping circuit changes the configuration bit-stream of the VRM, while the original configuration bit-stream of FPGA is fixed throughout the evolution. VRM implements the mapping of chromosomal from genotypes to phenotypes as shown in Fig.7. There is a one-to-one corresponding between each gene node of the chromosome and PE unit in PE block.

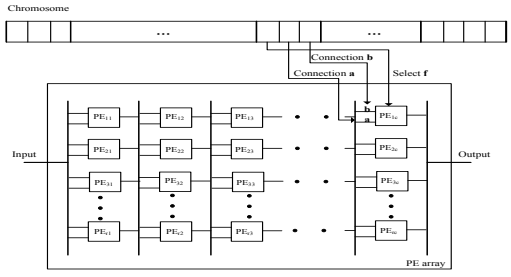

Figure 7. Genotype-phenotype Mapping

The PE unit is included two parts, configurable connection unit and configurable function unit in Figure 8.

After the Virtual Reconstruction Module receives a chromosome from Nios II algorithm module, gene node configures the function unit and connection unit of the PE automatically, according to the chromosome, thereby realizing the chromosome to circuit mapping.

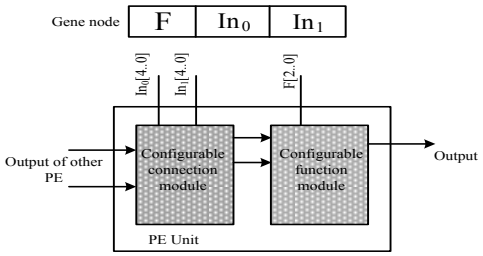

Figure 8. PE unit

The two connected genes of the gene node are expressed by a configurable module to determine the input source of the PE unit. The functional gene of the gene node, determine the function of the PE unit by a configurable function unit. The configurable connection unit can implement one of the six basic logic functions in Table 1. Taking the gene node number 18 in Fig.3 as an example, the gene value of the node is 0,15,8. The real chromosome code generated by the algorithm module is rotated by the machine. The real code of gene node is sent to PE unit and conversion to the binary code 000, 01111, 01000. The function of the PE unit is logical 'and' corresponding function gene '0'. The two inputs of the PE unit are come from the PE unit number 15 and number 8.

2) Outputl Module: The output module is the output port of the entire circuit. The terminal output of the circuit shown in Fig.9.

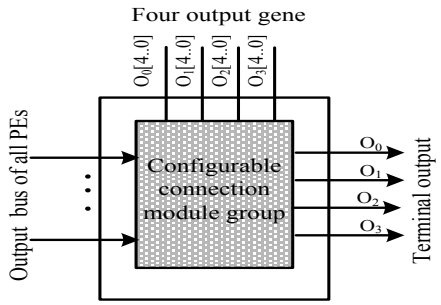

Figure 9. Output Module

The output module is composed of four multipath data selectors, The output module is configured by output genes of the chromosome. The output result of a PE unit is selected as the output result of the target circuit. The size of the output module is customized according to the number of output of the target circuit.

Taking the chromosome code in Fig.3 as an example, the model has 4 terminal outputs, the output module is 4 configurable connection modules. The gene values of the terminal output O_0, O_1, O_2 and O_3 were 16, 18, 17, and 5

respectively. After the chromosome is conversion to 10000, 10010, 10001 and 00101, that is output value of the O_0, O_1, O_2 and O_3 is derived from the output of the PE unit with node numbers 16, 18, 17 and 5.

C. Communication Control Module

The communication control module can implement the test data transmission between Nios II algorithm module and VRM. The module can implement the communication between VRM and external signals. When a correct circuit is evolved, the target circuit can use as a product independent.

The communication control module is the hub of data communication between VRM module, Nios II algorithm module, external input and external output, As in Fig.10.

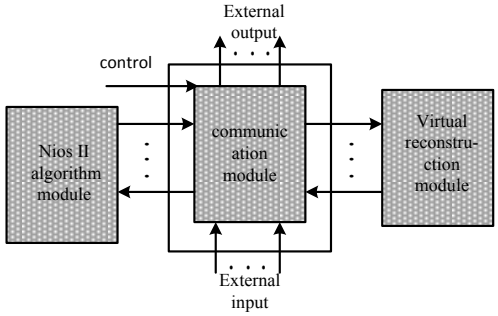

Figure 10. Schematic of communication control module

When VRM implements the mapping of chromosomal genotypes to phenotypes, the control port is set to '1'.

The external input and the external output are in a high-impedance state. The Nios II algorithm module test data output port is connected to the input port of the VRM by the communication control module. The test result receiving port of the Nios II algorithm module connected to the circuit output port of the VRM by the communication control module. After completing the above communication, Nios II algorithm module will test the evolution circuit of VRM and evolve the chromosome mapping circuit according to the online evaluating.

When the circuit meets requirement, the control port is set to '0'.

The test data transmission port and the test result receiving port of the Nios II algorithm module are both high-impedance state. The external input port is connected to the input port of the VRM, and the external output port is connected to the output port of the VRM, the target circuit can use as a product independent.

If the circuit does not meet the requirements, Nios II algorithm module continue to evolve the chromosome circuit until satisfactory results.

IV. EXPERIMENTS AND ANALYSES

A two-multiplier circuit is evolved using the on online hardware evolution platform based on matrix coded Cartesian Genetic Programming. The chromosomes are encoded with the matrix of 3 rows 5 columns. The population evolves according to the $(\mu+\lambda)$ evolution strategy [14]. The experiment takes μ=1, λ = 4 the chromosome is mutated according to a mutation rate of 0.06. The experiments were divided into 10 groups. Each group repeated experiments of 100 times, recording the evolution algebra obtained by each evolution. Calculating the average evolution algebra of each group, shown in Fig.11.

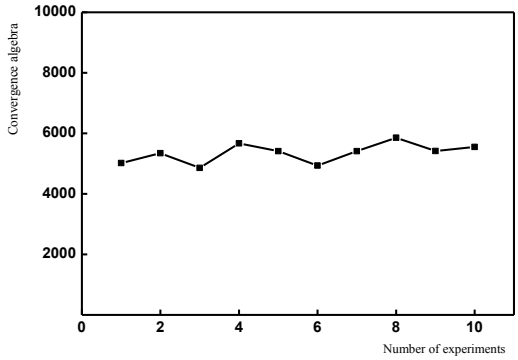

Figure 11. Convergence algebra diagram

In Fig.11, the average convergence algebra of circuit evolution is about 5300 generations. There is certain contingency in the evolution process of the circuit. In course of the experiment, some experiments can obtain the target circuit after thousands or even hundreds of generations. Some experiments have not been able to obtain the target circuit after ten thousands of generations. The convergence range is relatively large of normal evolutionary algebra. The convergence algebraic relationship diagram has certain fluctuations. The general trend is to converge around 5300 generations.

V. CONCLUSION

The online hardware evolution platform is designed based on FPGA using matrix coding CGP. The construction and principle of the online hardware evolution platform are researched and explained in detail. The multiplier circuit is evolved using the on online hardware evolution platform combining based on matrix coded Cartesian Genetic Programming. The multiplier experiments have shown that the platform is reliable. The online evolution platform has many advantages compared with offline evolution circuit. The platform can monitor the state of the circuit in real time. The target circuit evolved by the online evolution platform can be used as the product directly. The platform brings the designed circuit closer to the actual use of the product. The online evolution hardware platform is played a positive role in circuit evolution. The platform is provided a reliable online evolution platform for the research of circuit evolution design. Comply with the development trend of circuit evolution design.

ACKNOWLEDGEMENT

* Corresponding author, Song Xue-jun .

This research was supported by the Project of Joint Research Program "Chunhui Jihua" of the Ministry Education (Z20177023), Hebei Natural Science Foundation

and Key Basic Research Project (F2018205178) and Hebei Normal University Graduate Innovation Funding Project (CXZZSS2018069).

REFERENCES

[1] X. Yao and T. Higuchi, "Promises and challenges of Evolvable hardware," Berlin, Heidelberg, 1997, pp. 55-78.

[2] P. C. Haddow and A. M. Tyrrell, "Challenges of evolvable hardware: past, present and the path to a promising future," *Genetic Programming and Evolvable Machines,* vol. 12, pp. 183-215, September 01 2011.

[3] L. Sekanina, "Evolvable Hardware," in *Evolvable Components: From Theory to Hardware Implementations*, L. Sekanina, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 42-66.

[4] C. Xinyou and L. Dongliang, "A Survey on Evolvable Hardware Research," *Science & Technology Information,* pp. 446-447+424, 2010.

[5] W. Hongyu, "The Research of On-chip Circuits Evolutionary Design based on FPGA," master, Hebei Normal University, 2017.

[6] B. Lei and Y. Chenglong, "Digital Circuit Evolutionary Design Based on Expanding Multi-chromosome Cartesian Genetic Programming," *Ship Board Electronic Countermeasure,* vol. 36, pp. 109-113, 2013.

[7] J. F. Miller and P. Thomson, "Cartesian Genetic Programming," in *Genetic Programming*, Berlin, Heidelberg, 2000, pp. 121-132.

[8] J. F. Miller, "GECCO 2013 tutorial: Cartesian genetic programming," in *Conference Companion on Genetic and Evolutionary Computation*, 2013, pp. 715-740.

[9] L. Dan-yang, C. Jin-yan, Z. Sai, and M. Ya-feng, "Evolutionary Design of Fault-Tolerant Digital Circuit Based on Cartesian Genetic Programming," *Donghua University(English Edition),* vol. 33, pp. 231-234, 2016.

[10] Gajda, Z., Sekanina, and L., "Gate-level optimization of polymorphic circuits using Cartesian Genetic Programming," *Evolutionary Computation, 2009. CEC '09. IEEE Congress on,* 2009.

[11] Z. Yaolei, "Research on the approach to design digital circuit based on on-line evolution," master, Nanjing University of Aeronautics and Astronautics 2007.

[12] O. Elnokity, l. I. Mahmoud, M. K. Refai, and H. M. farahat, "Hardware implementation of virtual recongurable circuit for fault tolerant evolvable hardware system on FPGA," *Engineering and Technology Research,* vol. 15, 2015.

[13] J. Wang and C.-H. Lee, "Virtual reconfigurable architecture for evolving combinational logic circuits," *Central South University,* vol. 21, pp. 1862-1870, May 01 2014.

[14] H. Liang, W. Luo, and X. Wang, "A three-step decomposition method for the evolutionary design of sequential logic circuits," *Genetic Programming and Evolvable Machines,* vol. 10, pp. 231-262, 2009.