FPGA-based Implementation of Hardware Technology on Generic Algorithms

ZHONG Wei-Sheng[1,2] ,WANG Yu-Ti[2] ,ZENG Xiao-Shu[1]

1. Mechanical & Electronic College. Nanchang University,Nanchang,330029,China
E-mail: ws0791jx@163.com

2. Environmental Science & Engineering College, Nanchang University,Nanchang,330029,China
E-mail: ws0791jx@163.com

Abstract: The research on Generic Algorithms is usually focused on software implement, which is always restricted in terms of high real-time by computer system because it is a serial calculation. This paper introduces a hardware structure on FPGA-Based Generic Algorithms, which programmed by VHDL language. All modules compilation and simulation were performed by Altera Quartus□ 5.0 and were tested on EP1K100QC208-3. The experimental results show that the design achieves the required functions. At the end of this paper, a new idea of paralleled and pipelined hardware structure based on FPGA is presented.

Key Words: Generic Algorithms; FPGA; VHDL; Parallel; Pipeline

1 INTRODUCTION

Genetic algorithms(hereinafter, "GA") are a class of the overall random search algorithm, evolved out from the biological evolutionism (survival of the fittest, the genetic mechanism of survival of the fittest).GA is simple, universal, robust, parallel processing, as well as a wide range of applications. GA is widely used in areas such as combinatorial optimization, pattern recognition, image processing, artificial intelligence and adaptive control. FPGA, which can program repeatedly, erase and use, has the advantages of multi-function, high integration, small capital input. FPGA can be used to accomplish the circuit design, build, optimization and simulation. This paper presents a new FPGA-based parallel pipeline structure of the hardware design, programmed by VHDL language, which completed the production of hardware and algorithm testing. Therefore, it has great significance for hardware implementation**.**

2 THE GA SYSTEM

2.1 The Structure

GA basic process is shown in Figure 1. The genetic manipulation includes the choice, reversal, crossover and mutation.

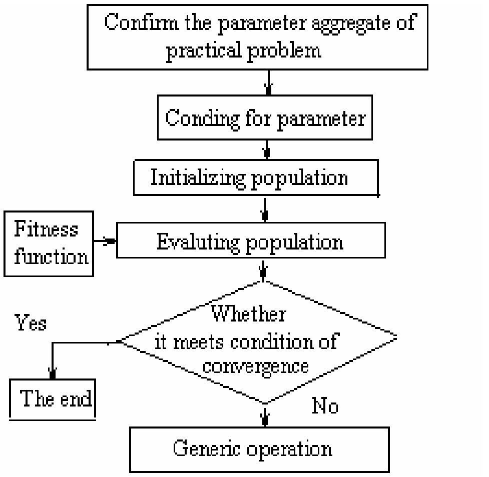

Fig 1.Generic operation

2.2 Module Function

The design of the FPGA GA system block diagram is shown in Figure 2.The main modules of GA system are as follows:

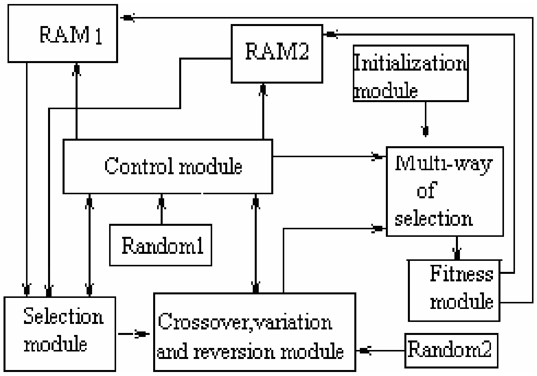

Fig 2 . The FPGA GA system block diagram

.

(1)Control Module: It is a single-state machine, used to control the whole system, so that the various modules can be operated coordinately. First, when the system power-on reset, the module will reset all of the control signals, resume the two chips of memory in the write-state, and make multi-channel selector module connect with the initialization module. (At the same

This work is supported by a grant from the Science and Technique Program of Jiangxi Province(20051B0400900)

978-1-4244-1734-6/08/$25.00 © 2008 IEEE

time, the initialization module produces two individuals at each rising edge of clock, and then they will be sent to the fitness module to calculate their fitness values. After being compared with the fitness values, the larger and the corresponding individual are stored to the memory.)Then the control module starts to count, and convert the counter value to two same memory addresses.

Second, when the number of individuals that the initialization module produces, is twice of the group size, the control module will receive an end-signal from the initialization module. Multi-channel selector is connected with the crossover module mutation module and the reverse module, and disconnected with initialization module. The memorizers are set to read status. The control module gives a start-signal to the selected module, and sends RAM1, RAM2 the addresses from random number generator. (RAM1and RAM2 sends the two individuals and their fitness values to select module.)

Third, the select module sends an end-signal to control module once a choice is made, while sending the result, the two individual to crossover module, mutation module and reversal module. At the same time, the control module gives start signals to crossover module, mutation module and reversal module.

Finally, when the crossover module, mutation module and reversal module finish, the control module receives an end-signal, then set the two state memorizers write enabled, and save the values that is the counter value plus 1. Then fitness module sends other individual and its fitness values to the two memorizers respectively and the control module chooses the selection module again. When operated repeated in this way until the result meets requirements, the whole system is stopped. Genetic algorithm thus concluded;

⑵ Random Modules: a total of two, are mainly responsible for generating corresponding random number;

⑶ Initialization modules: are responsible for generating initialization groups;

⑷ Multi-choice modules: are mainly responsible for sending the new individuals generated from initialization and cross-variation respectively into the fitness module.

⑸ Fitness modules: are responsible for calculating the fitness values of two new individuals, and selecting the larger value of the individuals, sending the individual and the fitness value respectively into RAM1 and RAM2 at the same time;

⑹ Choice modules: are mainly responsible for selecting two individuals with larger fitness values. When it receives the first pair of individuals and their fitness values, it will choose the individual with larger fitness value, which will be preserved temporarily, in accordance with competition mechanism. So does the second pair received. Finally, send two individuals to the crossover, variation and reversion mutations, and then select the individual “max” with largest fitness value of every generation.

⑺ Crossover, variation and reversion modules: When the two individuals sent by choice module are received, compare the probability of crossover, variation, reversion probability with inherent probability of random modules in sequence in order to determine whether to do the crossover , variation, and reversion operation: if the probability is less than inherent probability, then do the corresponding operation, generate and output a pair of new individuals finally; Otherwise, output the original directly.

⑻ Memory: RAM1 is used to store individuals, and RAM2 is used to store the corresponding individual fitness values, both of which use the same address. Dual-address dual-port RAM is provided by LPM module of ACEX1K in QuartusII7.1 in this paper, which has an input port and two output ports. RAM1 (or RAM2) is divided into upper and lower storage areas to store respectively the individuals of father and offspring (or fitness value), and to visit through the control module.

2.3 FPGA-based hardware and algorithm testing

The modules of the system are all programmed by VHDL. KHF-4 CPLD / FPGA development system for experimental developed by Zhejiang Tianhuang Industrial Co., Ltd is adopted for hardware implementation and algorithm testing. Its objective chip is EP1K100QC208-3 of Altera Corporation FPGA ACEX1K series which is a SRAM-based device, with look-up table (LUT) and embedded array block (EAB) to provide a high-density structure, which can be used to provide 100,000 doors. Each embedded array block is 16-bit wide, and can provide double ports, 49,125 RAM bits. And it has 147 pins, which drives 2.5 V devices, also can be driven by these voltage; bidirectional I / O pins’ execution speed will be up to 40 MHz.

The testing function used in this paper is

$$F(x)=\Sigma x_i^2;\ i\in[1,2];0\leqslant X_i\leqslant 25 \quad (1)$$

By the clock: 40MHz.

Genetic parameters are as follows: Population size: 32; crossover probability 0.875; variation probability: 0.0703125; reversal probability: 0.1; individual strings Length: 16; genetic number: 64; fitness value-string length: 17. System Simulation map is shown in Figure 3.

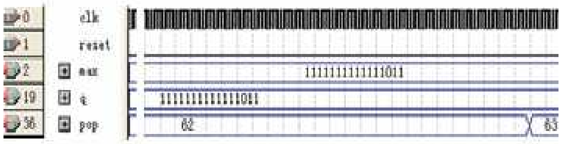

Fig.3. Final result diagram

The Max is the individual with the largest fitness value, and pop is the corresponding to the genetic number. Figure 3 shows that due to the randomness and local convergence of genetic-population operation, the individual ultimately converges to the 1111 1111 1111 1011, and did not achieve the desired convergence value 1111 1111 1111 1111.

3 A NEW DESIGN IDEA

The above GA design has not realized the assembly line parallelization of GA, when specially meets the large scale, the speed of realization will be still restricted, and easy to present itself with the partially converge and precocity. Therefore, we put forward a new design idea of paralleled and pipelined hardware structure based on FPGA. Its system overall construction diagram and sub-population evolution diagram is shown in Figure 4, Figure 5 respectively.

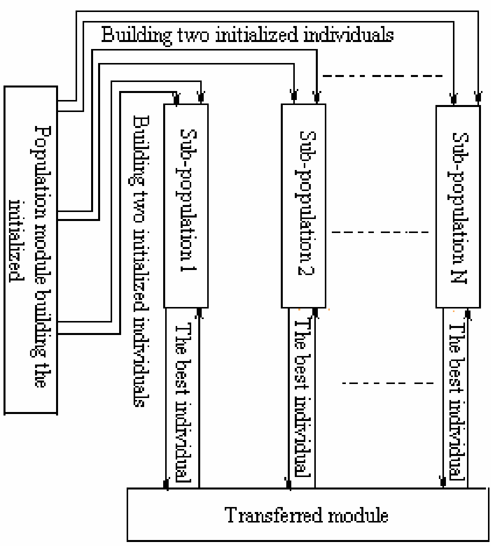

Fig 4.The whole frame figure of paralleled and pipelined hardware system

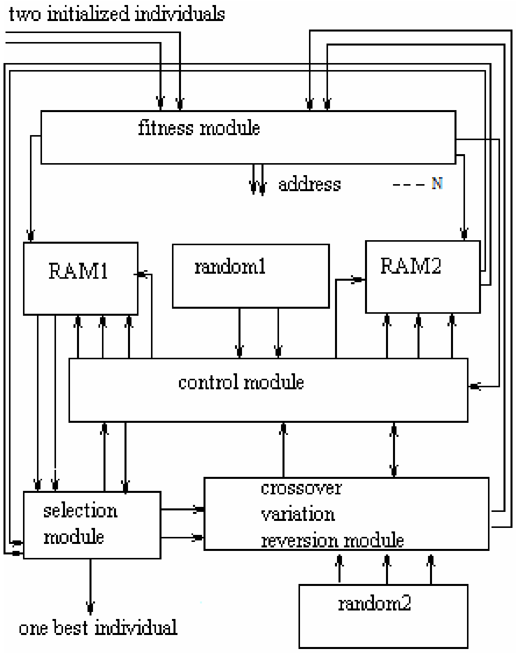

Fig 5.The evolution of sub-population frame

The method of GA parallelization has three ways: Main-Slave parallelization, coarse granule model and fine granule model.

In Figure 4, the initialization module produces a 2N individual at the rise of each clock, and at the same time transmits two individuals to each sub-population. Hereafter each individual carries on circulation heredity respectively. When some sub-population's heredity algebra achieves the predetermined value or the difference between the individual average sufficiency value of some generation and his/her previous generation's is smaller than some predetermined value, this sub-population suspends its heredity operation. When all sub-populations suspended, each sub-population simultaneously will deliver their best individual to the migration module.

After receiving the best individual, the migration module delivers them simultaneously to the N sub-population, used to replace their worst individual respectively. Hereafter, the migration module will simultaneously transmit a start signal to N population, so as to carry on the next heredity operation. Circulate repeatedly until the number of migration arrive the predetermined value.

In Figure 5, it is more or less the same with the preceding GA design l. What is different is that the sufficiency module will take on other two duties, besides the omitted multi-channel selectors.

[1] Seek the mean value of suitable value of each generation, and compare the difference between the absolute suitable value of the current and the preceding generation with the predetermined value. If it is smaller than predetermined value, transmit the command to the controller to suspend the heredity operation.

[2] Locate N individual and their address of each generation with the bad sufficiency value, in order to exchange with the best individual which comes from the migration module.

4 CONCLUSION

The GA natural parallelism and FPGA parallelism essentially constitute their natural matching, and using the hardware (relative to the software), can greatly enhance the effectiveness as well as the stability of the system. This article introduces GA structure function based on the FPGA, and by using the VHDL programming language, carries on the translation, the simulation in the QUARTUS7.1 integration environment, and passes the algorithm test on the CPLD/FPGA experiment development system. Finally we propose a new idea of the parallelization assembly line hardware based on FPGA, which provide the possibility for the real-time application of the GA.

REFERENCES

[1] G.L. Chen, X.F. Wang, Genetic Algorithm and Application. Posts &Telecom Press, Beijing, China, 2001.

[2] H. Yang, A.G. Li, Newest Operative Technology Guide of FPGA/CPLD. Tsinghua University Press, Beijing, China, 2005.

[3] Z.D. Wang. The research of implementation of generic algorithms Based on FPGA. Instruments & Meters User, 2006.

[4] L. Fang, H.C. Zhang, L.L. Zhu. The research of implementation of hardware on FPGA-based parallel genetic algorithm. Transactions of Nanjing University of Aeronautics & Astronautics, 2004.