Simulation and Analysis of Genetic Algorithm Based on FPGA

B.Umamaheswari[1] and N.Rajeswaran[2]

[1]P.G Student/VLSI Design, SNS College of Technology, Coimbatore, Tamilnadu.
[2]Dept.of ECE, SNS College of Technology, Coimbatore, Tamilnadu.

Abstract

In this paper, we propose a technique that utilizes the genetic algorithm for various VLSI circuits. In GA, we proposed the method of automatic test pattern generation (ATPG) is used to generate test vectors. Experiment results showed that the proposed algorithm reduce the complexity of the circuits and also the execution time. The design is realized using VHDL and then fabricated on FPGA.

Keywords: Genetic algorithm (GA), Automatic Test Pattern Generation (ATPG), FPGA.

Introduction

An evolutionary algorithm is a subfield of artificial intelligence. An evolutionary algorithm refers to the evolutionary computational models using randomness and genetic inspired operations. Evolutionary algorithm involves selection, recombination, random variation and competition of the individuals in the population adequately represented potential solutions.

A genetic or evolutionary algorithm applies the principles of evolution found in nature to the problem of finding an optimal solution to a solver problem. In genetic algorithm the problem is encoded in a series of bit strings are manipulated by the algorithm. In a evolutionary algorithm the decision variables and problem functions are used directly.

A drawback of any evolutionary algorithm is that a solution is “better” only in comparisons with others. This algorithm has no concept of an optimal solution or any way to test whether a solution is optimal. This also means that an evolutionary algorithm never knows for certain when to stop or number of iterations or candidate solutions, that you wish to allow it to explore.

Genetic algorithm

Genetic algorithm (GA) is an adaptive heuristic search algorithm based on the mechanism of natural selection and evaluation.GA is a artificial intelligence procedure and robust search method. This technique is efficient for finding combinatorial optimization problem. The objective of GA is to find optimal solution to a problem. Genetic algorithm belongs to the class of evolutionary algorithm which generates solution to optimization problems using techniques inspired by natural evolution such as inheritance, mutation, selection and crossover [4].

Proposed method

Hardware Architecture of GA

In this paper the evolvable hardware is used. This evolvable hardware can be implemented by combining hardware architecture of GA with evolvable computing logic [10]. This paper describes the implementation of evolvable hardware with the state machine hardware. The hardware architecture of genetic algorithm model based on FPGA consists of two units. They are processing unit and control unit. Figure 2 shows the hardware architecture of genetic algorithm [8].

Processing unit

The function of the processing unit includes initial population generation, fitness evaluation and genetic operation. There are five hardware modules in the processing unit. They are generation modules, selection modules, crossover modules, mutation modules and random number generation module (RNG). RNG generates random number for other modules.

Control unit

The control unit acts as a control state machine. The state machine of the control unit can be used to decide the operating sequence of initial population generation, population storage, fitness evaluation, selection, crossover and mutation. It can automatically send control signal to the processing unit.

Operation

The control signal can assure a correct executing in circles of these modules in the processing unit, depending on the operating rule about the sequence of these operations. The control unit receives the current state signals and generates the next state. These two units work co-ordinately to perform the calculation of GA.

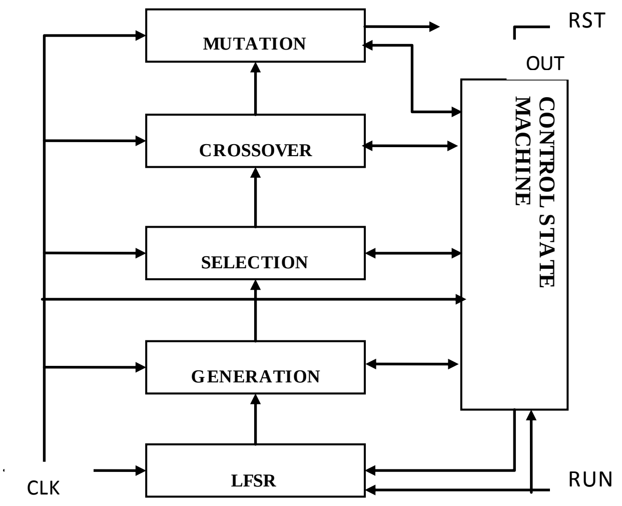


Fig.2 Hardware Architecture of GA

Control state machine

The modules of processing unit are controlled by the control state machine of the control unit and can work on two states. They are active state and sleeping state. The figure 3 shows the binary decision diagram of the control state machine. The state machine consists of four states. They are idle, birth, GA, store.

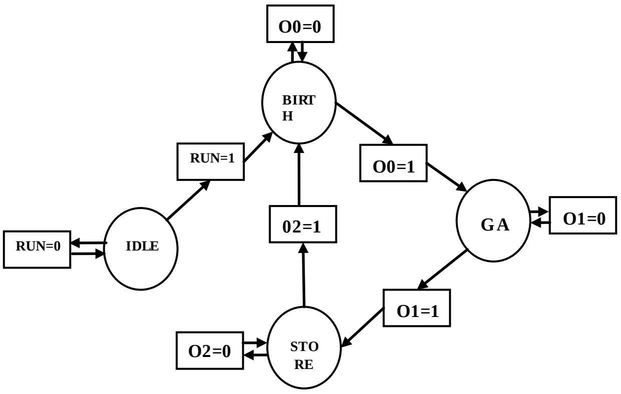


Fig.3 control state machine

Linear feedback shift register

Random number is of great importance to the operation of genetic algorithm. Normally random numbers are made by using linear feedback shift register (LFSR) based random number generators [3]. LFSR is a shift register whose input bit is a linear function of its previous state [6]. The two main parts of the LFSR is the shift register and feedback function.

RESULTS

This experiment is carried on the VLSI circuits. These circuits are simulated and implemented in FPGA. The proposed algorithm is simulated using VHDL language figure 4, 5 and 6 shows the experimental results for LFSR, control state machine and genetic algorithm.

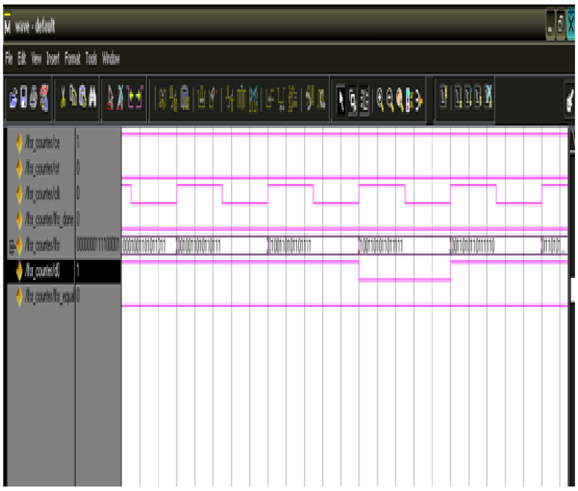


Fig.4 Simulated waveform for LFSR

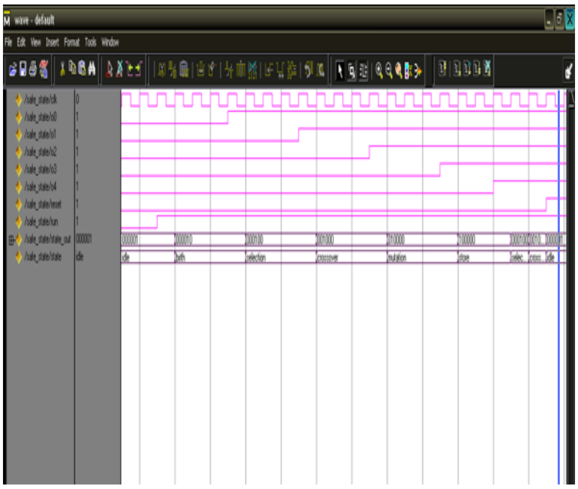
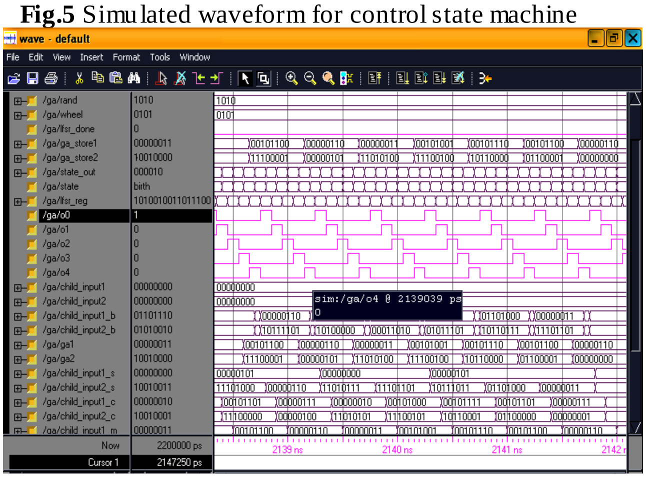


Fig.5 Simulated waveform for control state machine



/ga/rand	1010
/ga/wheel	0101
/ga/lfsr_done	0
/ga/ga_store1	00000011
/ga/ga_store2	10010000
/ga/state_out	000010
/ga/state	birth
/ga/lfsr_reg	1010010011011100
/ga/o0	1
/ga/o1	0
/ga/o2	0
/ga/o3	0
/ga/o4	0
/ga/child_input1	00000000
/ga/child_input2	00000000
/ga/child_input1_b	01101110
/ga/child_input2_b	01010010
/ga/ga1	00000011
/ga/ga2	10010000
/ga/child_input1_s	00000000
/ga/child_input2_s	10010011
/ga/child_input1_c	00000010
/ga/child_input2_c	10010001
/ga/child_input1_m	00000011
Now	2200000 ps
Cursor 1	2147250 ps

Fig.6 Simulated waveform of Genetic Algorithm

Conclusion

Genetic algorithm is promising methods for solving difficult technological problems. This project proposed a genetic algorithm for automatic test pattern generation in VLSI circuits. This method is used to improve the performance of the system. Here the control state machine and random data generation is used. So the execution time is reduced. GA provides efficient techniques for optimization and machine learning applications.

References

[1] Chatchawit Aporntrwan, Prabhas changstitvatana, 2001,"A Hardware implementation of the compact genetic algorithm".

[2] Iouliia Skliarova, Antonio B. Ferrari, 2001," FPGA- based implementation of genetic algorithm for the travelling salesman problem and its industrial application"

[3] Javad Frounchi, Mohammad Hossein Zarifi, Sanaz Asgari Far, Hamed Taghipour, 2003," design and analysis of random number generator for implementation of genetic algorithm using FPGA"

[4] Manoj Kumar, Mohammad Husain, Naveen Upreti & Deeti Gupta, 2010," Genetic Algorithm: review and application"

[5] Masaya Yoshikawa, Hironori Yamauchi, and Hidekazu Terai, 2008,"Hybrid architecture of Genetic algorithm and simulated annealing"

[6] Rachana singh, Arvind Rajawat, 2010, "Implementation and analysis of immune genetic algorithm for generating test pattern in VLSI circuits".

[7] Shruthi narayanan , Carla purdy, 2005" Hardware implementation of genetic algorithm modules for intelligent systems"

[8] Tatsubiro Tachibana, Yoshibiro Murata, Naoki shibata, kriichi Yasumoto and Minoru Ito, 2000, "General architecture for hardware implementation of genetic algorithm".

[9] Tatsubiro Tachibana, Yoshibiro Murata, Naoki shibata, kriichi Yasumoto and Minoru Ito,"A hardware implementation method of multi-objective genetic algorithms"

[10] Tu Lei and Zhu ming cheng, wang jing_xia, 2002,"The Hardware Implementation of a Genetic Algorithm Model with FPGA".