Genetic Algorithm Accelerator GAA-II

Shin'ichi Wakabayashi[†] Tetsushi Koide[‡] Naoyoshi Toshine[†] Masataka Yamane[†] Hajime Ueno[†]

[†]Faculty of Engineering, Hiroshima University
4-1, Kagamiyama 1 chome, Higashi-Hiroshima, 739-8527 JAPAN
Phone: +81-824-24-7676, Fax: +81-824-22-7195
e-mail: {wakaba,toshi}@ecs.hiroshima-u.ac.jp

[‡]VLSI Design and Education Center, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656 JAPAN
Phone: +81-3-5841-8908, Fax: +81-3-5841-8907
e-mail: koide@vdec.u-tokyo.ac.jp

Abstract— We have developed a new GA hardware called GAA-I (Genetic Algorithm Accelerator-I), in which the crossover operator to be applied to each individual was dynamically selected during the algorithm execution. GAA-I has some restrictions due to the limited chip size. In this paper, we extend the GAA-I and propose a new GA hardware, GAA-II, so that large, complex optimization problems can be solved. Furthermore, GAA-II has capability of parallel processing with other GAA-II chips. The GAA-II chip has been fabricated as a CMOS standard cell chip with 0.6 μm technology.

I. Introduction

Genetic algorithms (*GAs*) [1] are known as one of robust heuristic algorithms for complex optimization problems in various fields of engineering. GA provides robust capability of exploring in the solution space of a given problem. There are two notorious problems on GAs to realize their performance. One is the parameter tuning. The other problem of GAs is the computation time. To solve the former problem, we have proposed an adaptive method which selects adaptively crossover operators based on a new measure of superiority of an individual called the *elite degree*[2]. To reduce the execution time of GA, hardware implementation of GA has been proposed[3].

We have developed an adaptive GA hardware called GAA-I (Genetic Algorithm Accelerator-I), in which the crossover operator to be applied to each individual was dynamically selected during the algorithm execution[5, 6]. The GAA-I was implemented as an LSI chip, and its effectiveness was verified by simulation and experiments with the evaluation board. However, the GAA-I has some restrictions due to the limited chip size. For example, the number of parameters to which the user can set values is relatively few. In this paper, we extend the GAA-I and propose a new GA hardware, GAA-II, so that large, complex optimization problems can be solved. GAA-II also has capability of parallel processing with other GAA-II chips.

II. The GAA-II

A. Specifications of the GAA-II

Specifications of the GAA-I & II chips are summarized in Table I. In the GAA-II, the ranges of some parameter values are extended so that large, complex optimization problems can be solved.

B. Overall Design

Figure 1 shows the overall design of the GAA-II. The GAA-II consists of three modules as follows.

(1) GAA-II chip.

The GAA-II chip is the main module of the whole system, and performs all operations except the fitness calculation. The GAA-II chip itself consists of several submodules (units).

TABLE I
Specifications of the GAA-I & II.

	GAA-I	GAA-II
Population size	64, 128	32, 64, 128, 256
#bits of an individual	64 bits	64, 128, 256, 512 1024, 2048 bits
Fitness	16 bits	16, 32bits
Crossover	2-point, uniform, adaptive	multipoint(1-5), uniform, adaptive
Selection	roulette selection elitist strategy	roulette selection elitist strategy

(2) System Memory (SM).

The 64 Kword static RAM is attached to the GAA-II chip, where 1 word = 32 bits. This memory is used to store the information of each chromosome. Information of each chromosome consists of (a) binary coded representation of an individual (64 words in max), (b) fitness value (1 word), and (c) family tree information (0.5 word). The family tree information of a chromosome is used to calculate the elite degree. The system memory is also used as the data table for calculating the elite degree. Initial values of the system memory are set by the external CPU.

(3) Fitness Evaluation Module (FEM).

Fitness of each individual will be evaluated by this module. The FEM may be implemented on FPGAs. Each time a pair of chromosomes are created by the Crossover and Mutation Unit in the GAA-II chip, data of chromosomes are passed to the FEM through the FEM Interface. Since the FEM receives two chromosome data, parallel processing of fitness evaluation may be achieved if the FEM is designed to calculate the values of fitness functions for two data in parallel.

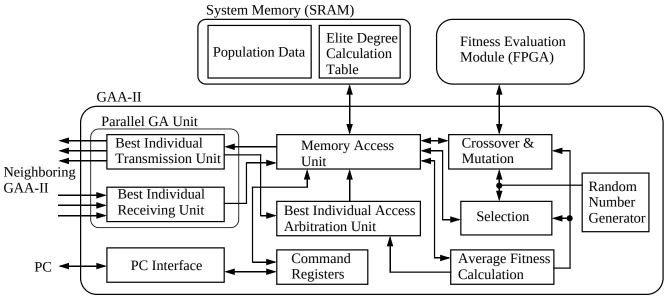
System Memory (SRAM)
Population Data
Elite Degree Calculation Table
Fitness Evaluation Module (FPGA)
GAA-II
Parallel GA Unit
Best Individual Transmission Unit
Best Individual Receiving Unit
Neighboring GAA-II
Memory Access Unit
Crossover & Mutation
Random Number Generator
Best Individual Access Arbitration Unit
Selection
PC
PC Interface
Command Registers
Average Fitness Calculation

Fig. 1. Overall design of the GAA-II.

C. Parallel GA

One of the problems of GAs is premature convergence to a local optimum. To solve this problem, parallel GAs have been investigated, in which a whole population is divided into some number of subpopulations, and individuals are exchanged between subpopulations [4]. Exchanging individuals among subpopulations is called *migration*. GAA-II supports the parallel execution of adaptive genetic algorithms by asynchronously exchanging individual data among three neighboring GAA-II chips. GAA-II chips may be connected with the Cube-Connected-Cycles topology, in which $P = n2^n (n \geq 2)$ GAA-II chips will be connected to realize a large size of parallel GA hardware. The migration process is executed in parallel with execution of GA operators such as crossover and mutation.

III. VLSI Implementation

The GAA-II chip has been designed with the Verilog HDL, and synthesized with the Synopsis Design Compiler. The layout design has been done with the Avant! Apollo. The chip has been fabricated as a 9.0mm^2 standard cell chip with 0.6μm CMOS technology with three metal layers. As the results of synthesis, the numbers of cells, nets, and signal pins of the chip were 37292, 37540, and 159, respectively. The post-layout simulation showed that the chip will be able to run with a 50 MHz clock. The GAA-II chip was realized as a 208 pin QFP. Figure 2 shows the chip image of the GAA-II chip.

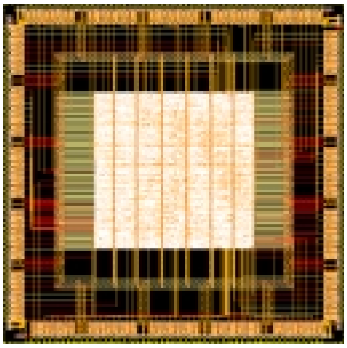

Fig. 2. Chip image of the GAA-II chip.

IV. Evaluation

To evaluate the GAA-II chip, we performed several simulation experiments. First, we compared the GAA-II with the GAA-I by solving several benchmark test functions (DeJong's test functions). Tables II and III show the average values of obtained results in 10 runs.

From Table II, the GAA-II (16-bit fitness) obtained a solution as good as ones by the GAA-I in a shorter execution time. But, the GAA-II (32-bit fitness) obtained optimal solutions although its execution time increases. Therefore the GAA-II achieves higher performance as GA hardware than the GAA-I.

Next, we evaluate parallel GA performance of the GAA-II. In this experiment, we use a parallel GA model, in which four GAA-II chips are fully connected. From Table III, parallel GA obtained better solutions than single GA in a smaller number of generations. Furthermore, parallel GA with migration obtained them in the smallest number of generations, because of keeping diversity in the population.

TABLE II

GAA-I vs. GAA-II ($f1$: optimal solution 0).

	GAA-I	GAA-II 16-bit fitness	GAA-II 32-bit fitness
solution	0.000462	0.000378	0
generation	94	91	463
clock	478828	549569	2351039
execution time [ms]	31.9	10.9	47.0

TABLE III

Evaluation of parallel GA ($f5$: optimal solution 0.998004).

	GAA-I	GAA-II × 4 no migration	GAA-II × 4 migration
solution	1.419420	1.059203	1.018403
generation	1160	670	263
clock	8438317	3977307	1561553
execution time[ms]	562.5	79.5	31.2

V. Conclusion

In this paper, we extend the GAA-I and propose a new adaptive GA hardware, GAA-II, so that large, complex optimization problems can be solved. GAA-II also has capability of parallel processing with other GAA-II chips. Simulation results showed that GAA-II is more efficient and effective than GAA-I. Future research is to introduce higher parallelism into the chip.

Acknowledgment

The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm Corporation and Toppan Printing Corporation. This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 10680356) from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co. (1989).

[2] K. Hatta, K. Matsuda, S. Wakabayashi, and T. Koide: "On-the-fly crossover adaptation of genetic algorithms," *Proc. IEE/IEEE Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.197–202 (1997).

[3] S. D. Scott, A. Samal and S. Seth: "HGA: A hardware-based genetic algorithm," *Proc. ACM/SIGDA 3rd International Symposium on FPGA*, pp.53–59 (1995).

[4] J. Stender (eds.): *Parallel Genetic Algorithms: Theory and Applications*, IOS Press (1993).

[5] S. Wakabayashi, T. Koide, K. Hatta, Y. Nakayama, M. Goto, N. Toshine: "GAA : A VLSI genetic algorithm accelerator with on-the-fly adaptation of crossover operators", *Proc. IEEE International Symposium on Circuits and Systems, Vol.2*, pp.268–271 (1998).

[6] S. Wakabayashi, T. Koide, N.Toshine, M. Goto, Y. Nakayama, K. Hatta: "An LSI implementation of an adaptive genetic algorithm with on-the-fly crossover operator selection," *Proc. Asia and South Pacific Design Automation Conference*, pp.37–40 (1999).