An Evolvable-hardware-based Clock Timing Architecture towards GigaHz Digital Systems

Eiichi Takahashi etakahas@etl.go.jp
Masahiro Murakawa murakawa@etl.go.jp
Kenji Toda toda@etl.go.jp
Tetsuya Higuchi higuchi@etl.go.jp

Computer Science Division, Electrotechnical Laboratory
MBOX 1502, 1–1–4 Umezono, Tsukuba 305-8568 Japan

Abstract

There are increasing demands for high-speed LSIs such as Pentium III(500 MHz) and DEC Alpha (600MHz). However, these fast digital systems have poor yield rates. In early stages of mass production of such LSIs, it is believed that yield rates might be less than 10 %. One of the reasons for poor yield rates is that the timing delays between digital components are not the same as the design specifications. Such discrepancy comes from the variations in the values of parasitic capacitances and resistors along data lines between digital components. Those values differ significantly depending on LSIs. The LSIs that do not satisfy the design specifications are just discarded, leading to poor yields rates. In order to solve this problem, we propose an EHW (Evolvable Hardware)-based clock architecture for high speed digital systems. In stead of just discarding the chips that do not meet the specifications, we genetically adjust the clock timings in the LSI and let it satisfy the specifications. We have developed a LSI, which is used in the high-speed memory tester, to show the advantages of the clock architecture. Simulation results show that the number of LSIs which can operate at 800 MHz increases from 2.9 % to 51.1 % after the clock timing circuits are evolved by genetic algorithms. Therefore, this clock architecture is expected to be one of the basic LSI technologies for GigaHz digital systems.

1 INTRODUCTION

There are increasing demands for high-speed LSIs such as Pentium III(500 MHz) and DEC Alpha (600MHz). However, these fast digital systems have poor yield rates. In early stages of mass production of such LSIs, it is believed that yield rates might be less than 10 %. One of the reasons for poor yield rates is that the timing delays between digital components are not the same as the design specifications[Rabeay 1996]. Such discrepancy comes from the variations in the values of parasitic capacitances and resistors along data lines between digital components(Figure 1). Those values differ significantly depending on LSIs. The difference in the clock timing is usually called "clock skew." The LSIs that do not satisfy the design specifications because of the clock skew are just discarded, leading to poor yields rates.

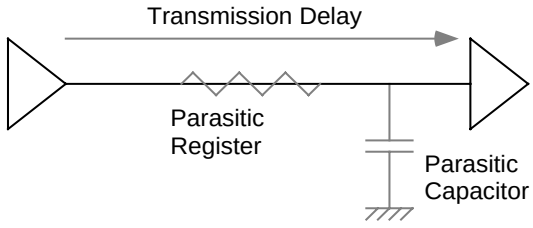

Figure 1: Variations of Transmission Delays

In order to solve this problem, we propose an EHW (Evolvable Hardware)-based clock architecture for high speed digital systems. In stead of just discarding the chips that do not meet the specifications, we genetically adjust the clock timings in the LSI and let it satisfy the specifications.

We have developed a LSI, which is used in the high-speed memory tester, to show the advantages of the clock architecture. Simulation results show that the number of LSIs which can operate at 800 MHz increases from 2.9 % to 51.1 % after the clock timing circuits are evolved by genetic algorithms (Figure 9). Therefore, this clock architecture is expected to be one of the basic LSI technologies for GigaHz digital systems.

This paper is organized as follows. Section 2 describes the basic concepts of EHW. Section 3 introduces the clock timing architecture. In section 4, simulation results are discussed. Section 5 concludes this paper.

2 EVOLVABLE HARDWARE: BASIC CONCEPTS

Evolvable hardware is based on the idea of combining reconfigurable hardware device with genetic algorithms to execute reconfiguration autonomously[Higuchi 1996, Yao 1998, Higuchi 1999].

The structure of reconfigurable hardware devices can be changed any number of times by downloading into the device a software bit string called configuration bits. FPGA (Field Programmable Gate Array) and PLD (Programmable Logic Devices) are typical examples of reconfigurable hardware devices.

A genetic algorithm (GA) is a robust search algorithm loosely based on population genetics[Goldberg 1989]. It effectively seeks solutions from a vast search space at reasonable computation costs.

The basic concept behind the combination of these two elements in EHW is to regard the configuration bits for reconfigurable hardware devices as chromosomes for genetic algorithms (See Figure 2). If a fitness function is properly designed for a task, then the genetic algorithms can autonomously find the best hardware configuration in terms of chromosomes (i.e. configuration bits).

For example, in data compression with EHW, we use a prediction function. Optimal prediction functions vary greatly according to the different kinds of data to be compressed. It is, therefore, not possible to design in advance a prediction hardware function. Instead of specifying a detailed hardware design, we define a fitness function. In the case of data compression, the data compression rate is used as a fitness function. Accordingly, a circuit of prediction function with a higher data compression rate is likely to remain in a population. When a good chromosome is obtained, it is immediately downloaded into the reconfigurable device.

If the prediction performance of a given EHW is reduced due to changes in the nature of the data to be compressed, then the GA process is invoked and the search for a better hardware structure of prediction is initiated. In this way, EHW is capable of both autonomous and dynamic hardware reconfigurations.

In the rest of the paper, we propose a new kind of the EHW.

3 EHW-BASED CLOCK TIMING ARCHITECTURE

The details of the proposed architecture is described in this section. The architecture is a new kind of the EHW.

3.1 BASICS

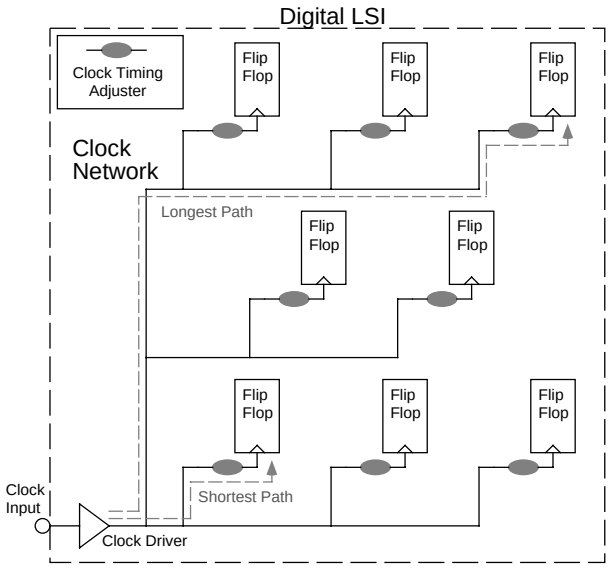

Figure 3: Evolvable Clock Timing Architecture

The clock skew is one of the essential problems which occur in the digital systems. In the traditional way of designing LSIs, many processes include procedures to get rid of the clock skew. To solve the problems, we propose a new EHW-based clock architecture.

Our clock architecture has two key features: one is to introduce delay devices adjustable in the clock timing, the other is to adopt a different evaluation function (also known as a "fitness function" in the GA domain) from traditional one in the clock timing.

The proposed architecture is depicted in Figure 3, and it is quite simple. Clock timing adjusters are simply inserted to the traditional clock architectures.

The clock timing adjusters are described in Section 3.2, and the evaluation function is described in Section 3.3. The implementations of the architecture are discussed in Section 3.4.

3.2 BASIC ELEMENT — CLOCK TIMING ADJUSTER

In Figure 4, the clock timing adjuster is represented, which is used in our clock architecture.

The adjuster has quite simple structure, consisting of a register and a specified delay generator. The bit-width of the register depends on the extent of variation in delay time.

The delay generator include small delay devices with unit-time delay, with twice, four times, eight times, and so forth, and these delay devices are serially connected. Each bit of the chromosomes register corresponds with one of the small delay devices, and the

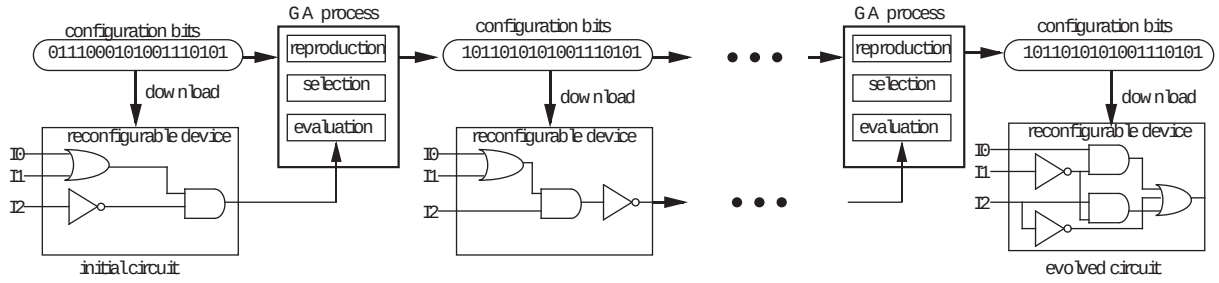

Figure 2: Basic Concept of Evolvable Hardware

value of the bit controls whether the delay device is turned on or not. Taking an example of the figure, if the unit delay value is 10ps, the adjuster generates 60ps (10ps $\times$ 4 + 10ps $\times$ 2) timing delay as a whole.

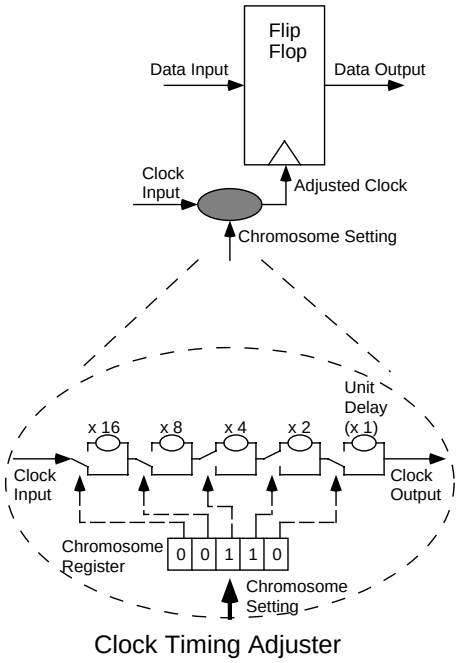

Figure 4: Clock Timing Adjustable Flip Flops

In addition, to generate "negative delays" not only "positive delays," advanced clock can be provided to the systems.

3.3 EVALUATION FUNCTION

The other of the key features is an evaluation function, which is also known as a "fitness function" in the GA domain. The normal goal of other clock timing adjustment methods is to correct clock skew. In the recent studies on high speed digital LSIs, several electronic methods to zero clock skew are proposed[Geannopoupos 1998, Shibayama 1998].

However, the goal with our proposed architecture is more precise, in that it seeks to correct the system to conform to the designer's specifications. The adjustment proceeds as follows:

1. Loading a tentative chromosome set into a chip
2. Conducting ordinary LSI chip test, supplying a series of test data and system clock at the speed of the specifications
3. After the chip test, stopping the system clock and extracting all FF values
4. Comparing the extracted FF values with expected values generated by logical simulation
5. Calculating the evaluation function with the comparison
6. Generating new generation of chromosomes using the evaluation function

Note that the clock timing is not directly measured at all.

Figure 5 shows how to calculate the evaluation function. In the figure, two FFs out of eight fail to work, and its evaluation value is 80%. Note that chips with evaluation values that are not 100% are taken as defective ones.

The evaluation function provides several benefits. One of the benefits is related to measuring clock timing, which is quite costly. Since our architecture doesn't have to measure clock timing, calculating the evaluation values is much lighter than the clock timing measurement.

Other benefit is an automatic delay adjusting behavior, depicted in Figure 6. In the figure and both in "Chip A" and "Chip B", an uppermost signal represents one of traditional clocking systems such that stick to the clock timing specifications, while a lowest signal represents another of clocking systems adopting our approach such that stick to the logical correctness. The middle signal, labeled "Before Adjusting", has several "up edges", which shows variations

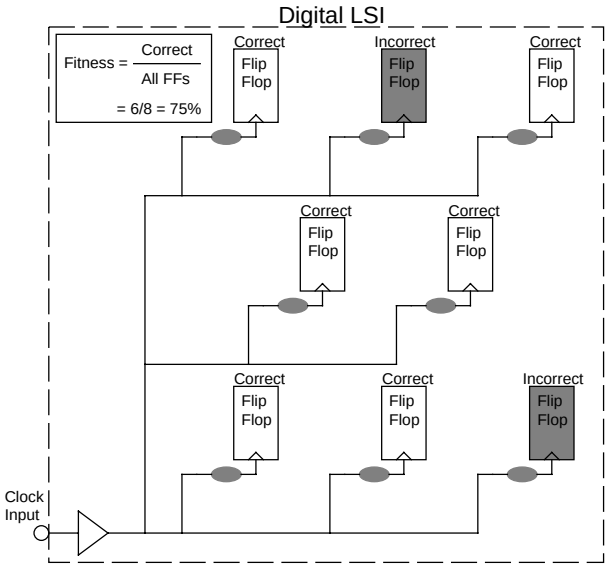

Figure 5: Evaluation Function

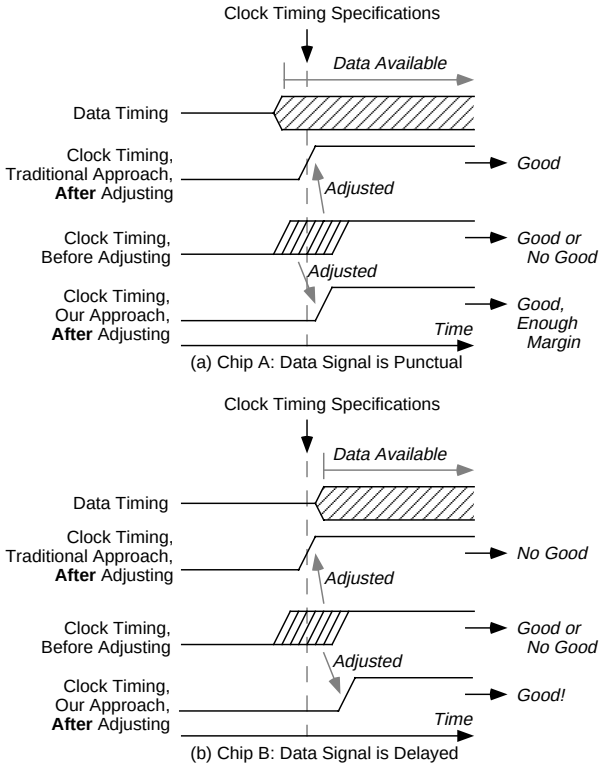

Figure 6: Automatic Delay Adjusting Behavior

of clock timing, i.e. the clock skew problem. In the traditional clocking systems, almost all of clocks are always on time for the specifications. In contrast, in the improved clocking system adopting our approach, all clock timings are adjusted to maintain the logical correctness of the workings of FFs, though they would not satisfy the timing specifications.

In normal cases like "(a) Chip A" where the data signal meets the timing specifications, both clocking systems operate properly. However, in cases like "(b) Chip B" where the data signal is delayed from the specifications, which come not to be uncommon situations in high speed ("GigaHz") digital systems, the traditional clocking systems doesn't operate properly while the systems using our approach still operate properly. The reason is that in the systems using our approach the clock timing adjusters could make the clock be automatically delayed corresponding to the signal delay.

Moreover, adjusting under various condition, including variations of power voltage and temperature, makes chips robuster because they obtain sufficient timing margins.

3.4 IMPLEMENTATION

There are a few discussions in the implementation, such as how to check data latched by FFs, and fill up the chromosome registers.

To apply our method to some chips, data latched by FFs in the chips have to be observed. This doesn't mean continuous observation of all FFs, but by using some sequences of data, i.e. test patterns, correctness of the workings of FFs has to be determined. Recently, a lot of LSIs commonly have mechanisms, such as scan path methods, to enable FFs inside the chips to be observed. Our method can utilize the mechanism sufficiently.

On the other hand, when LSIs have a lot of the clock timing adjusters, they contain quite many chromosome registers. Therefore, there must be some efficient methods to set up the registers. One of such solutions is depicted in Figure 7. All chromosome registers are connected serially, and are made to be a long shift register. Using this method, setting up the registers can be done by a simple shift-in operation. Although shift-in time can be reduced by dividing the register into N shift registers, the shift-in operation is not time-consuming: when one million bits are shifted-in with 1MHz clock it takes just one second.

4 SIMULATION STUDY

The simulation experiments conducted to adjust clock timing for a circuit are described next.

For evaluation, the memory test pattern generator as shown in Figure 8 was selected, and converted to the simulation model to be evaluated. It is simple but quite important logic block for memory testing devices. It is used to generate access patterns for test-

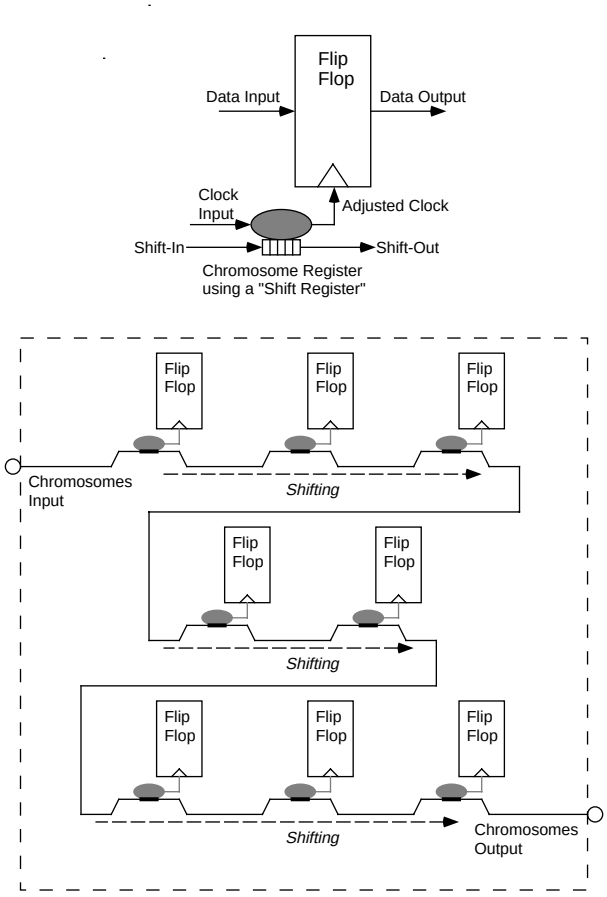

Figure 7: How to Fill up Chromosome Registers

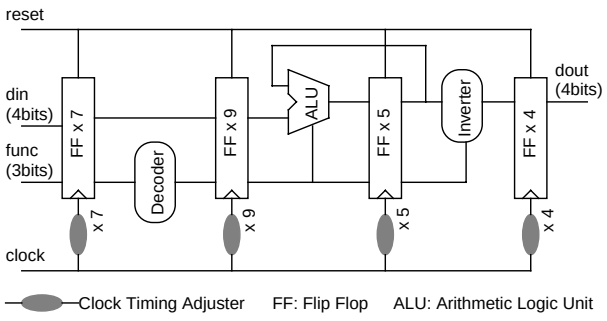

Figure 8: Evolvable Memory Test Pattern Generator

ing memory in the same speed as accessing real memory chips. For example, in testing memory modules for the Direct Rambus systems, the testing hardwares should supply pretty long test patterns in the speed of 800MHz[Rambus 1997].

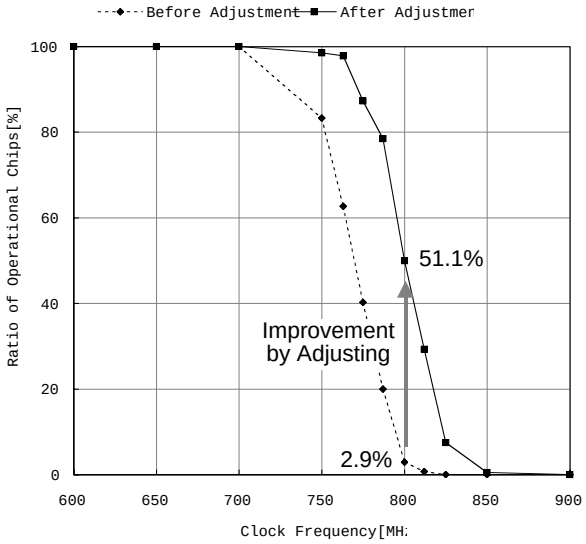

Figure 9: Simulation Result(Distribution Function)

In the simulation, the number of samples was 1,000; the population size was 50; and the number of generations was 20. Device parameters were assumed to vary according to a normal distribution. The results are depicted in Figure 9. The X-axis represents the clock speed, with the percentage of chips, which could operate at a given speed, is plotted along the Y-axis. The most significant result from these simulations was the finding that 50% of the chips can be adjusted to operate at improved speeds of 800MHz, whereas only 2.9% could operate at such speeds before adjustment.

The same results in another view are depicted in Figure 10. The density function is plotted along the Y-axis, and represents the distribution of the highest clock speed where each chip can operate properly. The curve which is similar to the normal distribution is shifted to the right and is distorted by adjusting clock timing, and the graph evidently represents the improvement in clock speed.

These show that the proposed architecture not only improved the yield, but also provides higher standard chips.

Now, real chips for the memory test pattern generator with an EHW-based clock timing adjusters are being made. Their manufacturing is in the final phase. The die photograph is shown in Figure 11. Using gate-array structure, a regular lattice-like pattern is seen. Their performance is summarized in Table 1. The high

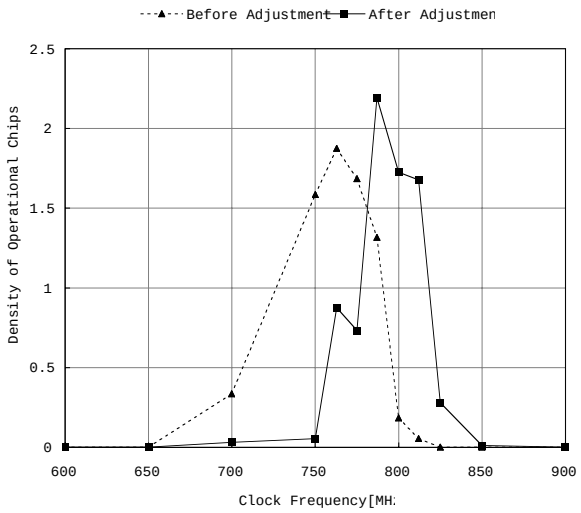

Figure 10: Simulation Result(Density Function)

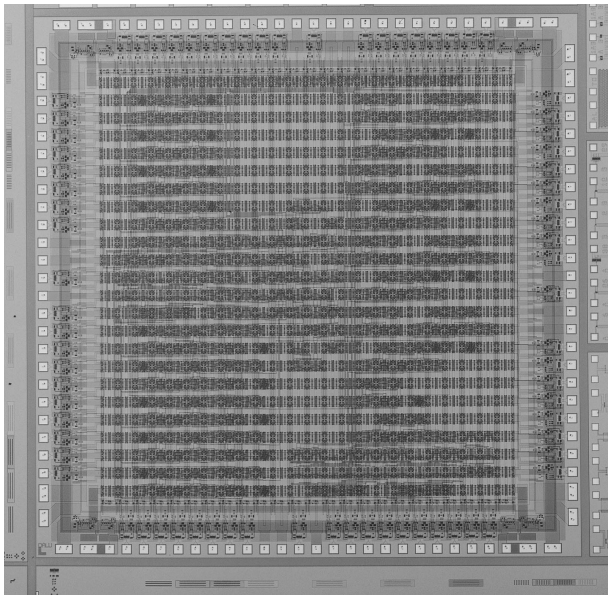

Figure 11: Die Photograph of the Experimental Chip

Table 1: Performance Summary of the Experimental Chips

Process technology	Bipolar, 3-metal layers
Number of gates	2,000 gates
Die size	6.5 mm × 6.5 mm
I/O pins	80 pins
Power dissipation	6.3W
Clock frequency	500MHz
Power supply voltage	VEE=−4.5V±0.3V VTT=−2.0V±5%
Package	128pin QFP with heat sink

speed bipolar technology is adopted, which is using for the chips in communication devices.

5 CONCLUSION

An EHW-based clock timing architecture was proposed, and the results of simulation study were revealed.

The proposed architecture has great potential to overcome current obstacles in a number of practical applications, and we have already started to investigate some of these. This method of clock design is a revolutionary step forward in the development of such digital systems.

Acknowledgments

The authors thank N. Otsu and K. Ohmaki for support, the imprecise system research group, and the real-time system research group for discussion, and T. Mano, N. Narumi and K. Yasuda in the NTT Electronics Technology Corporation for the technical support.

References

[Rabeay 1996] J.Rabaey, "Digital Integrated Circuits," Prentice Hall, 1996.

[Higuchi 1996] T. Higuchi, M. Iwata, and W. Liu (editors), Proc. of First International Conference on Evolvable Systems, Lecture Notes on Computer Science, 1259, Springer Verlag, 1996.

[Yao 1998] X. Yao and T. Higuchi, "Promises and Challenges of Evolvable Hardware," IEEE Transactions on Systems, Man, and Cybernetics, Part C, volume 28, number 4, November 1998.

[Higuchi 1999] T. Higuchi, N. Kajihara, "Evolvable Hardware Chips for Industrial Applications," Communications of ACM, ACM, April 1999 (to appear).

[Goldberg 1989] Goldberg, David, " Genetic Algorithms in Search, Optimization, and Machine Learning," Reading Mass.: Addison-Wesley, 1989.

[Geannopoupos 1998] Geannopoulos, Dai, "An Adaptive Digital Deskewing Circuit for Clock Distribution Networks," ISSCC Digest of Technical Papers, pp.400-401, Feb. 1998.

[Shibayama 1998] Shibayama, Mizuno, Abiko, Ono, Masuoka, Matsumoto, Tamura, Yamada, Nishizawa, Kawamoto, Inoue, Nakazawa, Sakai, Yamashina, "Device-Deviation Tolerant over-1GHz Clock Distribution Scheme with Skew-Immune Race-Free Impulse Latch Circuits," ISSCC Digest of Technical Papers, pp.402-403, Feb. 1998.

[Rambus 1997] Rambus Inc., "Direct Rambus Technology Overview," `http://www.rambus.com/docs/drtechov.pdf`, (PDF Document on the WWW), Oct. 1997.