

Forty Years of Logic Synthesis: Memoir

Tsutomu Sasao
Department of Computer Science
Meiji University
Kawasaki, Kanagawa 214-8571, Japan

Abstract—This paper reviews the author’s research activities on logic synthesis for the past 40 years. Topics include, AND-OR minimization, AND-EXOR minimization, LUT cascades, index generation functions, and branching program machines.

I. RESEARCH AT OSAKA UNIVERSITY

A. AND-OR Logic Circuits

In 1972, I graduated from the Department of Electronics Engineering at Osaka University. To do my BS thesis work, I joined the group of Hiroshi Ozaki. However, Kozo Kinoshita who was an Associate Professor of Ozaki’s group took care of me. The title of my BS thesis was *Logic synthesis using fan-in limited NAND gates*. At that time, Saburo Muroga of University of Illinois was working on exact minimum NAND networks using integer programming. I studied his papers as well as Dietmeyer’s book [6].

As for my Ph. D. thesis, I studied *conservative logic circuits*, which is related to reversible logic circuits. After receiving my Ph. D. degree, I became a Research Associate for the group of Hiroaki Terada of the same department. As for the research project, I selected logic design of Programmable Logic Arrays (PLAs).

In 1978, I attended the International Symposium on Multiple-Valued Logic (ISMVL) held at Rosemont, Illinois, to present a paper on PLA design. After the symposium, I visited General Electric (GE) in Syracuse, NY, International Business Machines Corporation (IBM) in Poughkeepsie, NY, and the Signetics in Sunnyvale in CA.

Hiroaki Terada recommended that I visit David Greer of GE, who was working on Associative Logic Arrays [5], which are similar to folded PLAs. As for IBM, I wrote a letter to Se June Hong, who was famous for his PLA minimization program called MINI [4].

Both David Greer and Se June Hong welcomed me, although I could not speak English satisfactorily. They showed me their recent research results.

In IBM, they were using APL language on Tektronix graphic terminal to develop software. This is much more efficient than my environment: FORTRAN language on main frame computers. After showing their system, Se June Hong recommended that I visit Saburo Muroga of University of Illinois, instead of sightseeing in Denver. He kindly made a telephone call to Muroga to make the arrangements. Fortunately, I had an air pass of the United States that allowed me to fly to major cities unlimited times without paying any extra charge.

At the Urbana-Champaign, I met Muroga for the first time. His group was working on the exact minimization of two-level multiple-output logic network to design PLAs.

Next, I visited Signetics¹ in Silicon Valley. Unfortunately, the research of the logic design was not so active there. They were just using the results of Edward J. McCluskey’s research [7].

With this trip, I stayed in the USA for nearly one month, and visited many places working on PLA design, and got very important acquaintance and ideas for my future research. The acquaintances include Saburo Muroga, Se June Hong and Jon T. Butler of Northwestern University.

B. Translation of Muroga’s Book

In 1979, I received an international phone call from Muroga. He asked me to translate his book [9] into Japanese. I agreed to it: It took more than two full years to translate the book, since the English version had more than 600 pages.

Since no personal computer nor no word processor was available, we had to script Japanese sentences by hand. I wrote drafts, and hired students part-time to rewrite them on the manuscript papers provided by the book publishing company². At that time, the production of a book was very time-consuming work: no Internet, no E-mail, no personal computers, no LATEX, and no laser printers.

In May of 1980, ISMVL was held at Evanston, Illinois. After that, we visited Muroga again. In this time, my wife was with me, since we married one month earlier. We stayed at Muroga’s house for four days.

During our stay at Muroga’s house, we selected technical terms for the translation. It was very time-consuming. For each technical term, we spent considerable time. For instance, he disliked my translation “Turing machine” into “Turing KIKAI.” (KIKAI denotes machine in Japanese.) So, we decided to represent it by using Katakana characters which produced word pronounced similar to *machine* in English. In 1981, the Kyoritsu Shuppan published the translation of [9] as a special issue of *BIT Magazine*³. Since the circulation of the book was very good, many Japanese textbooks on logic design published later adopted technical terms used in our translation.

¹Signetics first produced field programmable logic array in 1975.

²Japanese uses more than 6000 different characters, so typewriters were not common at that time.

³The *BIT* was one of the most popular Japanese monthly magazines on computer science. Unfortunately, it was forced to cease its publication.

C. Research at IBM

In 1981, my *IEEE Transactions on Computers* (IEEE TC) paper on complexity of PLAs [10] was published.

During January 1982 to January 1983, I took a one-year sabbatical from Osaka University to stay at the T. J. Watson Research Center of IBM. Kozo Kinoshita recommended that I go there.

Se June Hong took care of my life in Yorktown Heights. However, the group leader of logic synthesis was Robert K. Brayton. At that time, Brayton was more famous in the area of network analysis than logic synthesis. Se June Hong was just appointed as the group leader of Artificial Intelligence. Brayton's group was working for two-level logic minimizer ESPRESSO by APL language [12]. Also, he was working for multi-level logic generator using weak division [11] with Curtis T. McMullen. In the summer, Gary D. Hachtel, Alberto L. Sangiovanni-Vincentelli, and Mike Lightner gathered for logic synthesis seminar. During one year at IBM, I learned many things: not only on research, but also the culture of United States.

In several years, Brayton moved to the University of California at Berkeley to be the leader of a logic synthesis group: His work on ESPRESSO[12] and MIS[18] are very famous.

When I returned from USA, Japan was in the middle of the fifth-generation computer boom⁴. NEC was selling PC-9800 series personal computers, and people could buy their own computers.

In 1984, I published an IEEE TC paper[14] on decoded PLAs, on the algorithms for detection of essential prime implicants and on output phase optimization. In 1986, I published a handbook on PLA's [16]. Since I was an assistant professor in Osaka University, I could not be a formal advisor of Ph. D. students. However, I essentially advised Keiji Ishikawa, who obtained his Ph. D. degree on optimization of PLAs in 1983. His FORTRAN program is still maintained by me, and it is one of my most important research tools. Also Motoki Higashida developed useful CAD programs.

II. MOVE TO KYUSHU INSTITUTE OF TECHNOLOGY

In April of 1988, I moved to the newly founded campus of Kyushu Institute of Technology (KIT) in Iizuka City, Fukuoka, Japan. To promote the new campus, KIT decided to hold an international symposium in 1992. The symposium consisted of four workshops. Since the progress of logic synthesis and microprocessors were remarkable at that time, we decided to host a workshop on these topics.

We invited Marvin Minsky of MIT as the keynote speaker of the symposium. Robert K. Brayton and Saburo Muroga were main guest speakers for the workshop. Additional invited speakers included Kurt Keutzer, Olivier Coudert, and Fabio Somenzi.

At that time, much research on logic synthesis was done in Japan: Each company was developing its own logic synthesis system. We obtained enough donation for the workshop, since

⁴Now, it is considered as an unsuccessful national project.

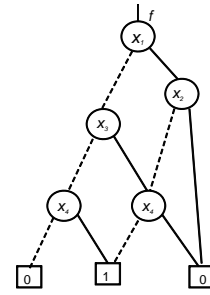


Fig. 2.1. An example of BDD.

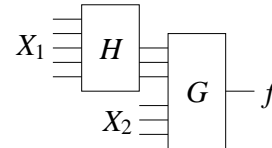


Fig. 2.2. Functional decomposition.

this business was rather good. I selected key papers from the workshop, and published a book with Kluwer [21] in 1993.

For the Kluwer book, I asked Brayton and Coudert to write chapters on two-level logic minimization, and Muroga to write a chapter on transduction method.

Saburo Muroga and Yahiko Kambayashi invented a design method for multi-level logic synthesis called the *Transduction method* in 1970's. Unfortunately, its usefulness was not recognized then. However, in 1986, an efficient implementation of Binary Decision Diagrams (BDD, Fig 2.1) was developed by Randal E. Bryant [17], and the transduction method was implemented on BDDs. After that, the transduction method was used for practical designs in industry. It took 20 years for the theory of the transaction method to be put to practical use.

In this book, I wrote a chapter on functional decompositions using BDDs (Figs. 2.2 and 2.3)⁵, and a chapter on AND-EXOR logic synthesis. Kluwer sold more copies of this book than expected.

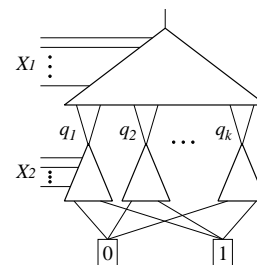


Fig. 2.3. Width of a BDD and column multiplicity.

⁵It shows that the column multiplicity of functional decomposition is equal to the width of the BDD, for the first time.

III. AND-EXOR LOGIC CIRCUIT

A. Philipp W. Besslich

In the fall of 1983, at the library of Osaka University, I encountered a paper of Philipp W. Besslich from the university of Bremen, Germany. In that paper, he detected properties of logic functions using spectrum transformation to simplify logical expressions with many variables. Since I was interested in the benchmark functions used in his paper, I wrote him a letter with a list of questions. After that, we began to exchange our knowledge.

About one year later, he sent me a letter that he wanted to visit Japan to work with me by a fellowship sponsored by the German Academic Exchange Service (DAAD) and the Japan Society for the Promotion of Science (JSPS). Soon, he successfully obtained the grant.

When he visited Osaka University in 1986, he brought a Pascal program that simplified AND-EXOR expressions. The algorithm first obtained the spectrum of the logic function to detect features of the function, and then simplified the AND-EXOR expression. At that time, I was not interested in AND-EXOR expressions, and also was ignorant about the spectral transformation of logic functions. However, after some discussions with him, I noticed that a simplification method for AND-OR expressions (MINI) can also be applied to AND-EXOR expressions. In a few days, I developed a much faster and more effective program than Besslich's.

In addition to the logic minimizer, Besslich was interested in the benchmark functions for AND-EXOR circuits. He proposed symmetric functions $SB(n, k)$ represented by EXORing the products of k variables out of n variables. For example, when $n = 4$ and $k = 2$, $SB(4, 2)$ consists of $\binom{4}{2} = 6$ logical products:

$$SB(4, 2) = x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4.$$

With my new AND-EXOR minimizer, I obtained the numbers of products needed to represent $SB(n, k)$ functions by AND-EXOR expressions, for different values of n and k . Also, I minimized AND-EXOR expressions for all the representative classes of four variable functions. We found that AND-EXOR expressions (ESOPs) require many fewer products than AND-OR expressions (SOPs). We published this result as an IEEE TC paper [19], that is one of the most cited papers among our publications.

Although Besslich stayed Japan only one month, the outcome was remarkable. As for the $SB(n, k)$ function, many years later, I realized that the function was considered by Yasuo Komamiya in 1959⁶.

⁶Yasuo Komamiya attended the Japanese multiple-valued logic workshop held in Shikanoshima Island in July 1990, where Norio Koda presented our work on the simplification of AND-EXOR expressions. After that, Komamiya sent me his book and papers. In fact, the book and paper contain the theory on $SB(n, k)$ functions. Unfortunately, I realized it after Komamiya passed away. Since Komamiya's work on adder design was not well known, Radomir S. Stankovic and I wrote an IEEE TCAD paper to promote Komamiya's work [25].

After Besslich returned to Germany, I developed an iterative AND-EXOR minimizer called EXMIN. This can treat multiple-valued input logic functions, and can simplify AND-EXOR PLAs with input decoders. With this program, I designed various benchmark functions, and showed that AND-EXOR networks can be much simpler than corresponding AND-OR networks for arithmetic circuits and telecommunication circuits. Later, the results were published as a paper of *IEEE Transactions on Computer-Aided Design and Integrated Circuits Systems* (IEEE TCAD) [20].

After I moved to Kyushu Institute of Technology (KIT: Iizuka), Besslich visited Japan again. During his stay in Iizuka, his health condition was not so good. After he returned to Germany, he visited hospital to find that he was had pancreatic cancer. He sent me a letter saying that he could live only a few months more, and he was doing only high priority tasks: to take care of his students. At that time, I asked him to be the European area program chair for ISMVL-1992 to be held in Sendai, Japan. Unfortunately, he could not attend ISMVL-1992, but his wife sent me a letter informing me that he passed away on June 2, 1992.

B. Reed-Muller Workshop

As for the simplification of AND-EXOR expressions, Norio Koda of Tokuyama Technical College spent a sabbatical year in our group. We published several papers on the upper and lower bounds on the number of products in AND-EXOR expressions (ESOPs). After that, he became a Ph. D. student of mine, and in 1996, he received his Ph.D. degree.

The impact of 1990 IEEE TC paper with Besslich [19] was rather high. In 1993, Wolfgang Rosenstiel from Germany informed me that he was organizing a Reed-Muller workshop in Hamburg. So, I attended the workshop with Norio Koda. The attendance at the Reed-Muller 1993 workshop was small, but many key persons on decision diagrams and EXOR logic gathered.

At the end of the workshop, we decided that the second workshop would be held in 1995 in Makuhari, Chiba, Japan. The organizers were myself and Masahiro Fujita⁷ of Fujitsu America Laboratory. We planned to publish a monograph [23] in 1996, in addition to the workshop proceedings. So, we invited authors whose papers can be chapters of the book. The workshop site was provided by Fujitsu, and I obtained several grants to support the travel expenses for overseas attendees. We had many participants, and the workshop was quite successful.

The 9th workshop was held in 2009 in Naha, Okinawa, Japan. In this time, we published a monograph [34] from Morgan & Claypool. In addition to the proceedings. The 11th workshop will be held in 2013 in Toyama, Japan.

Although, EXOR logic circuits are mathematically interesting, they are not so popular in industry. Thus, we have fewer papers than in previous Reed-Muller workshops. Instead, we have more papers on quantum logic circuits and reversible logic circuits.

⁷Currently, University of Tokyo

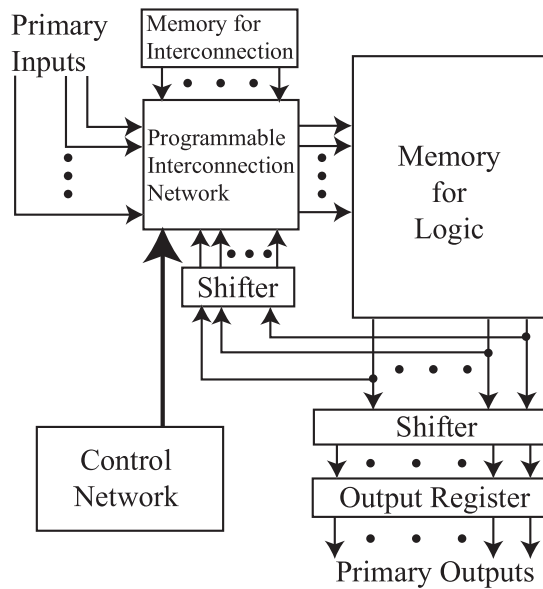


Fig. 4.1. LUT Cascade Emulator.

In 1998, Debatosh Debnath from Bangladesh obtained his Ph. D. degree on the research of *AND-OR-EXOR three-level logic circuits*.

IV. LUT CASCADE

In 2000, I was appointed as the director of the *Center for Microelectronic System (CMS)* of KIT. This position required me to attend many meetings of both KIT and Kitakyushu city. These meetings consumed most of my time, and I had to reduce my research time. The only achievements in this period were publications of English textbooks. In 1999, I published a book from Kluwer [24], which is an English translation of [22]. Also, in 2002, we published an edited book [26], which was sponsored by the *International Workshop on Logic Synthesis (IWLS)*.

In 2001, I applied for the research grant called *The Takeda Techno-Entrepreneurship Award*. At that time, we were working on a new Programmable Logic Device (PLD), so I wrote a proposal on it. Fig. 4.1 shows the architecture of the programmable device, which included the memory for logic and the programmable interconnection network. This architecture is used to emulate the look-up table (LUT) cascade as shown in Fig. 4.2. The feature of this grant was that they reviewed proposals interactively using the Internet. During the review process, they requested various supporting data and documents, and I spent considerable time on them. However, this also gave me a chance to consider the problem deeply and to explore related papers. Finally, I obtained the best award. In addition to considerable amount of award money, this idea became the seed to write a proposal to the MEXT (Ministry of Education, Culture, Sports, Science and Technology of Japan) Cluster Project (the first stage) that started in 2002.

An **LUT cascade** (Fig. 4.3) is a circuit obtained by connecting LUTs (memories) in series. The main feature of it

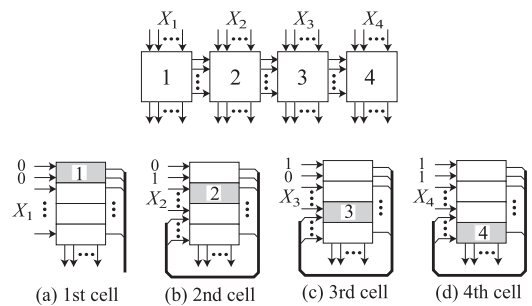


Fig. 4.2. Operation of an LUT cascade emulator.

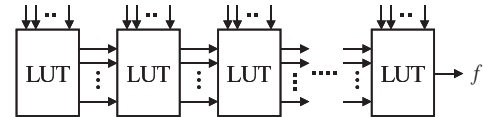


Fig. 4.3. LUT cascade.

is that the circuit can be directly derived from the BDD of the function. This circuit is derived by repeatedly applying functional decompositions as shown in Fig. 2.2. To implement an efficient circuit, the function must have a BDD with small width.

The LUT cascade emulator simulates various LUT cascades with different number of cells and different number of inputs for each cells. Thus, it is more versatile, but is slower than the LUT cascade. However, it is 5 to 40 times faster than microprocessors running at the same clock speed. The LUT cascade emulator can be also implemented by an ordinary personal computer (PC). The logic simulator based on an LUT cascade emulator is 3.5 to 10.6 times faster than the leveled compiled code (LCC) simulator on the same PC [30].

Since the goal of the MEXT knowledge Cluster Project was products for business, we worked very hard to find useful applications of LUT cascades and their emulators. However, to find 'killer' applications was not easy, and I spent many frustrating months. Finally, I found that an LUT cascade can be used in a numerical computation network to simplify the circuit. Thus, I talked to Jon T. Butler and Mark Riedel who were staying in Iizuka, and quickly wrote a paper for a SASIMI workshop [28].

The design the circuit required both the knowledge of logic functions and numerical functions, such as trigonometric functions and logarithm functions. Since everything was formulated by mathematics, an automatic design tools were relatively easy to develop. I thought that the network can be easily implemented by an FPGA, and also the experiment can be done in a short time. So I asked Shinobu Nagayama to develop an automatic design tool. At that time, he finished his Ph. D. degree shorter than the standard course, and had six months of spare time before moving to his new position. In this project, Nagayama successfully wrote many papers [32], [33], and later we were awarded a best paper award from the Information Processing Society of Japan [37].

TABLE 5.1
REGISTERED VECTOR TABLE

Vector						Index
x_1	x_2	x_3	x_4	x_5	x_6	f
0	0	0	0	1	0	1
0	1	0	0	1	0	2
0	0	1	0	1	0	3
0	0	1	1	1	0	4
0	0	0	0	0	1	5

In the first stage of cluster project (2001-2005), I had to supervise other projects at KIT, to evaluate other projects of three universities, in addition to do my own research project. This was too much work, and I could not concentrate on my own research work. Especially, for the first year, in addition to the LUT cascade project, I was also involved in the development of a voice recognition system. This was a difficult time.

During this time, Hassan Babu from Bangladesh, and Shinobu Nagayama obtained their Ph. D. degrees in 2000 and 2004, respectively. In 2006, Yukihiro Iguchi of Meiji University spent a year in Iizuka, and we worked with various applications for LUT cascades: logic simulator, radix converter [31], and digital filter. With Kazuyuki Nakamura of CMS, we developed a LUT cascade chip⁸[29].

V. INDEX GENERATION FUNCTION

In the MEXT Cluster Project, the final goal was to develop a commercial product by doing joint research with industries. At the end of the first stage of the Cluster Project, Masayuki Chiba of YAMAHA Corporation showed me a concept of **index generator** as the key device in the network appliance. I realized that this circuit can be easily implemented by an LUT cascade. From this idea, YAMAHA developed an LSI prototype using an LUT cascade emulator.

An **index generation function** is an integer valued function:

$$\{0, 1\}^n \rightarrow \{0, 1, \dots, k\}.$$

For example, in the case of $n = 6$ and $k = 5$, the function maps $k = 5$ different two-valued vectors into $k = 5$ distinct integers as shown in Table 5.1.

An index generation function can be directly implemented by a Content Addressable Memory (CAM), but CAMs dissipate high power and are expensive. An index generation function can be also implemented by an LUT cascade. However, when the value of k is large, the sizes of the LUTs are too large. Thus, I invented a better realization than LUT cascades for this function. The **Index Generation Unit (IGU)** shown in Fig. 5.1 can realize index generation functions quite efficiently. The design problem of an IGU can be formulated as minimization problem of the variables for an incompletely specified function. This method is quite efficient, and we can easily implement a practical pattern matching network by using an FPGA and memories. An IGU can easily implement

⁸99.5% of the chip area was devoted to the SRAM, while only 0.5 % of chip area was devoted to switches and registers. Thus, it looks like a conventional LSI memory.

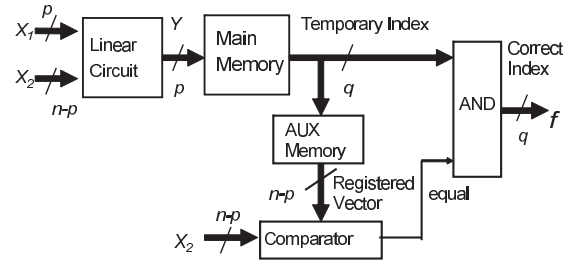


Fig. 5.1. Index Generation Unit (IGU).

a network for $k > 10^6$. Furthermore, if we use a linear transformation, we can drastically reduce the size of the memory. With this idea, we published many papers. Especially, these were interesting mathematical problems and this became a fruitful research endeavor.

Also, in the second stage of the Cluster Project that started in 2007, real products for business were expected. Thus, we concentrated on pattern matching circuits for communication applications. We developed circuits and algorithms for that.

In the second stage of the cluster project, the government hired personnel who performed administration of other research groups. Thus, I could concentrate to my own research project. This was quite helpful, since I had a hard time in the first stage of the cluster project (2001 to 2005).

In the applications of LUT cascade, Qin Hui from China worked on a cryptography network, and Hiroki Nakahara worked on a logic simulator. They obtained their Ph. D. degrees in March and September of 2007, respectively. Also, in 2011, I published a monograph [38] that summarizes LUT cascades and index generation functions.

VI. BRANCHING PROGRAM MACHINE

The branching program machine [13] (Fig. 6.1) is an idea that I encountered when I was in Osaka University. Unlike an ordinary microprocessor, it is a very simple processor having only two instructions: conditional jump and output instructions. Its instruction program can be directly generated from the binary decision diagram [8], and can be easily implemented by an FPGA.

For many years, I tried to enhance the idea. I noticed that with multiple-valued decision diagrams and multiprocessors, we can implement a very power-efficient and high-performance controller⁹ [35], [36]. This idea took nearly 30 years from the initial concept to a practical circuit.

In the MEXT Cluster Project, I hired Hiroki Nakahara who is very good at developing FPGA systems. Indeed, he quickly implemented various decision diagrams, and published many papers: He developed various branching program machines by FPGAs, virus detection networks, and packet classification networks. He also received paper awards.

⁹The parallel branching program machine consisting of 128 branching program machines and a programmable interconnection on an FPGA, is more than 20 times faster than the Intel Core2Duo micro-processor, and dissipates a quarter of the power.

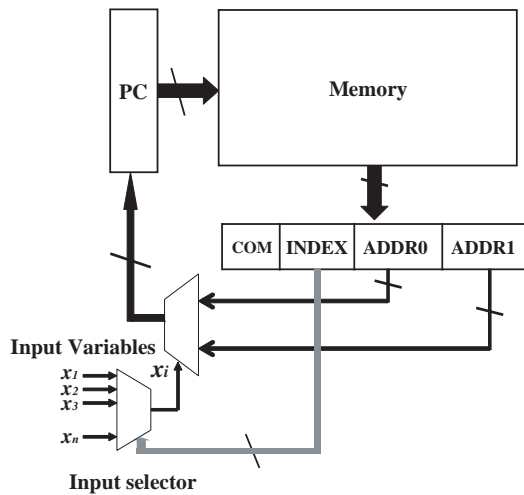


Fig. 6.1. Branching Program Machine.

TABLE 7.1
COMPARISON OF PROGRAMMABLE LOGIC DEVICES

Device	Interconnections	Memory
PLA	Fixed	Logic Circuit
FPGA	Static Programmable	Logic Circuit
LUT Cascade	Fixed	Logic Circuit
LUT Cascade Emulator	Dynamic Programmable	Logic Circuit
Branching Program Machine	Fixed	Program

VII. SUMMARY

A. Programmable Logic Device

For these 40 years, we have worked with various Programmable Logic Devices (PLDs). Table 7.1 summarizes these PLDs. A PLA corresponds to an SOP or an ESOP, while an LUT cascade corresponds to a BDD. The LUT cascade is considered as a special case of an FPGA, where interconnections are fixed and the structure is restricted to cascade. Architectures of LUT cascade emulator and branching program machine are similar. The difference is that the LUT cascade emulator use memory to store logic circuits, while the branching program machine use memory to store instructions. Also, the program for the branching program machine corresponds to a BDD. As for the speed, the PLA is the fastest, while branching program machine is the slowest to implement logic functions.

B. Rise and Fall of Logic Synthesis

1971 was the year when Intel announced the 4004 microprocessor. After this year, the boom of microelectronics occurred. Both memories and microprocessors advanced according to the Moore's law for 40 years. In such a promising year, I luckily started research on logic synthesis. Related industries also enjoyed a boom. For many years, research on logic synthesis appeared in the spotlight after a difficult time in 1960s [2], [6].

During 1980's, Japanese industries had many researchers: Nippon Telegraph and Telephone Company (NTT), NEC, Fujitsu, Hitachi, Sony, Panasonic and Sharp. In the USA, IBM, Bell Laboratories, and GE had many researches on logic synthesis.

In 1984, Xilinx was founded and they started FPGA business. In 1986, Synopsys was founded¹⁰, and they started logic synthesis business. After 1986, CAD researchers moved to Synopsys, Cadence, and Intel.

In the USA, the international workshop on logic and synthesis (IWLS) had been held regularly since 1987. Unfortunately, the number of attendees has declined in recent years.

In the era of *Switching Circuit Theory* [1], [3], logic elements were quite expensive, and the *Programmable Cellular Logic* was just a dream. However, now, such devices are routinely used as FPGAs. Also, *Spectral Transform* [3], [15] was just an academic research, but it may be useful for logic synthesis [40].

ACKNOWLEDGMENTS

In Osaka University, Hiroshi Ozaki and Kozo Kinoshita gave me the basic knowledge of logic design. Saburo Muroga trained my writing ability through the translation of his book [9]. When writing a paper, we had to carefully select words, and construct a sentence by spending enough time. When writing an English sentence, write it so that any student can understand. For Japanese, it seem to be too verbose. This is what I learned from Muroga.

In IBM, Se June Hong and Robert K. Brayton took care of me. Logic design and switching theory are now *classic*, however, I am still enjoying lectures and writing papers: They are my bread and butter.

In Kyushu Institute of Technology, I had less odd-jobs than the professors in the universities in Tokyo or Kansai area. Thus, I could concentrate into my favorite research. I could publish six English books. This is partly due to the help of Jon T. Butler who visited Iizuka almost every year, and improved presentation of papers, including this paper. The MEXT Cluster Projects gave me generous grants for ten years. In addition, JSPS Research Grants supported me for almost every year for 30 years.

Many companies (Fujitsu, Hitachi, Mitsubishi, NEC, OKI, Renesas and Sharp) donated to my research work, which were useful to invite foreign researchers. From the overseas, many researchers (Besslich, Brand, Butler, Mishchenko, Perkowski, Reidel, Stankovic) visited me. Kyushu Institute of Technology has the Alumni Association called *Meisen Kai*. It also supported travel expenses for overseas visitors and myself.

The Ph.D. students worked very hard. Especially, Shinobu Nagayama and Hiroki Nakahara were outstanding post doctoral researchers. In the Lab., Munehiro Matsuura and Yukihiko Iguchi of Meiji University helped me in various aspects.

¹⁰The original company name was *Optimal Solutions*

REFERENCES

- [1] M. A. Harrison, *Introduction to Switching and Automata Theory*, McGraw-Hill, 1965.
- [2] T. D. Friedman and S. C. Yang, "Methods used in an automatic logic design generator (ALERT)," *IEEE TC*, Vol. 18, No.7, pp. 593-614, July 1969.
- [3] A. Mukhopadhyay (ed.), *Recent Developments in Switching Theory*, Academic Press, New York, 1971.
- [4] S. J. Hong, R. G. Cain, and D. L. Ostapko, "MINI: A heuristic approach for logic minimization," *IBM Journal of Research and Development*, Vol. 18, pp. 443-458, 1974.
- [5] D. L. Greer, "An associative logic matrix," *IEEE Journal of Solid-State Circuits*, Vol. SC-11, No. 5, pp. 679-691, Oct. 1976.
- [6] D. L. Dietmeyer, *Logical Design of Digital Systems* (second edition), Allyn and Bacon Inc., Boston 1978.
- [7] P. Bricaud and J. Campbell, "Multiple output PLA minimization: EMIN," Proc., 1978 Wescon Professional Program, Los Angeles, California, Sept. 12-14, 1978.
- [8] S. B. Akers, "Binary decision diagrams," *IEEE TC*, Vol. 27, No. 6, pp. 509-516, 1978.
- [9] S. Muroga, *Logic design and Switching Theory*, Wiley-Interscience Publication, 1979.
- [10] T. Sasao, "Multiple-valued decomposition of generalized Boolean functions and the complexity of programmable logic arrays," *IEEE TC*, Vol. C-30, No. 9, pp. 635-643, Sept. 1981.
- [11] R. K. Brayton, and C. McMullen, "The decomposition and factorization of Boolean expressions." *International Symposium on Circuit and Systems (ISCAS)*, pp. 49-54, Rome, April 1982..
- [12] R. K. Brayton, G. D. Hachtel, C. McMullen, and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Press, 1984.
- [13] P. J. A. Zsombor-Murray, L. J. Vroomen, R. D. Hudson, Le-Ngoc Tho, and P. H. Holck, "Binary-decision-based programmable controllers, Part I-III" *IEEE Micro*, Vol. 3. No. 4, pp. 67-83 (Part I), July-Aug. 1983, Vol. 3. No. 5, pp. 16-26 (Part II), Oct. 1983, Vol. 3. No. 6, pp. 24-39 (Part III), Nov-Dec. 1983.
- [14] T. Sasao, "Input variable assignment and output phase optimization of PLA's," *IEEE TC*, vol. 33, pp. 879-894, 1984.
- [15] S. L. Hurst, D. M. Miller, J. C. Muzio, *Spectral Techniques in Digital Logic*, Academic Press, London, 1985.
- [16] T. Sasao, *How to Make and How to Use Programmable Logic Arrays* (in Japanese), *Nikkan Kogyo Shinbunsha*, 1984.
- [17] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE TC*, vol. C-35, no. 8, pp. 677-691, Aug. 1986.
- [18] R. K. Brayton, R. L. Rudell, A. L. Sangiovanni-vincentelli, A. R. Wang, "MIS: A Multiple-Level Logic Optimization System," *IEEE TCAD*, Vol. 6, No. 6, pp. 1062-1081, 1987.
- [19] T. Sasao and P. Besslich, "On the complexity of MOD-2 Sum PLA's," *IEEE TC*, Vol. 39, No. 2, pp. 262-266, Feb. 1990.
- [20] T. Sasao, "EXMIN2 : A simplification algorithm for exclusive-OR-Sum-of-products expressions for multiple-valued input two-valued output functions," *IEEE TCAD*, vol. 12, No. 5, May 1993, pp. 621-632.
- [21] T. Sasao (ed.), *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993.
- [22] T. Sasao, *Logic Design: Switching Circuit Theory* (in Japanese), Kindai Kagaku Sha (1995-01).
- [23] T. Sasao and M. Fujita (eds.), *Representations of Discrete Functions*, Kluwer Academic Publisher, May 1996.
- [24] T. Sasao, *Switching Theory for Logic Synthesis*, Kluwer Academic Publishers, 1999.
- [25] R. S. Stankovic and T. Sasao, "A discussion on the history of research in arithmetic and Reed-Muller expressions," *IEEE TCAD*, Vol. 20, No. 9, Sept. 2001, pp. 1177-1179.
- [26] S. Hassoun and T. Sasao (eds.), *Logic Synthesis and Verification*, Kluwer Academic Publishers, Oct. 2001.
- [27] T. Sasao, "Survey of research projects conducted by Sasao's group," <http://www.lsi-cad.com/sasao/Papers/pub2004.html>
- [28] T. Sasao, J. T. Butler, and M. D. Riedel, "Application of LUT cascades to numerical function generators," *The 12th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI-2004)*, Oct. 18-19, 2004, Kanazawa, Japan, pp.422-429.
- [29] K. Nakamura, T. Sasao, M. Matsuura, K. Tanaka, K. Yoshizumi, H. Nakahara and Y. Iguchi, "A memory-based programmable logic device using look-up table cascade with synchronous static random access memories," *Japanese Journal of Applied Physics*, Vol. 45, No. 4B, 2006, April, 2006, pp. 3295-3300.
- [30] H. Nakahara and T. Sasao, "A PC-based logic simulator using a look-up table cascade emulator," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E89-A, No.12, Dec. 2006, pp.3471-3481.
- [31] Y. Iguchi, T. Sasao, and M. Matsuura, "Design methods of radix converters using arithmetic decompositions," *IEICE Trans. on Information and Systems*, Vol.E90-D, No.6, June 2007, pp.905-914.
- [32] T. Sasao, S. Nagayama and J. T. Butler, "Numerical function generators using LUT cascades," *IEEE TC*, Vol.56, No.6, June 2007, pp.826-838.
- [33] S. Nagayama and T. Sasao, "Complexities of graph-based representations for elementary functions," *IEEE TC*, Vol. C-58, No. 1, Jan. 2009, pp.106-119.
- [34] T. Sasao and J. T. Butler, (eds.) *Progress in Applications of Boolean Functions*, Morgan & Claypool Publishers, Jan 2010. pp.1-153.
- [35] T. Sasao, H. Nakahara, M. Matsuura, Y. Kawamura, and J. T. Butler, "A quaternary decision diagram machine: Optimization of its code," *IEICE Trans. on Information and Systems*, Vol. E93-D No. 8 pp. 2026-2035, Aug. 2010.
- [36] H. Nakahara, T. Sasao, M. Matsuura, and Y. Kawamura, "A parallel branching program machine for sequential circuits: Implementation and evaluation," *IEICE Trans. on Information and Systems*, Vol. E93-D No. 8 pp. 2048-2058, Aug. 2010.
- [37] S. Nagayama, T. Sasao and J. T. Butler, "Programmable architectures and design methods for two-variable numeric function generators," *IPSIJ Trans. on System LSI Design Methodology*, Vol. 3, pp.118-129, Feb. 2010
- [38] T. Sasao, *Memory-Based Logic Synthesis*, Springer, March 2011.
- [39] T. Sasao, "Survey of research projects conducted by Sasao's group (FY2004-FY2011)," <http://www.lsi-cad.com/sasao/Papers/pub2012.html>
- [40] T. Sasao, "An application of autocorrelation functions to find linear decompositions for incompletely specified index generation functions," *International Symposium on Multiple-Valued Logic*, (ISMVL-2013), May 2013 (to be published).