Japanese Journal of Applied Physics
Vol. 45, No. 4B, 2006, pp. 3295–3300

A Memory-Based Programmable Logic Device Using Look-Up Table Cascade with Synchronous Static Random Access Memories

Kazuyuki NAKAMURA, Tsutomu SASAO, Munehiro MATSUURA, Katsumasa TANAKA, Kenichi YOSHIZUMI, Hiroki NAKAHARA and Yukihiro IGUCHI[1]

Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
[1]*Meiji University, 1-1-1 Higashimita, Kawasaki 214-8571, Japan*

(Received September 21, 2005; accepted November 21, 2005; published online April 25, 2006)

A large-scale memory-technology-based programmable logic device (PLD) using a look-up table (LUT) cascade is developed in the 0.35-μm standard complementary metal oxide semiconductor (CMOS) logic process. Eight 64 K-bit synchronous SRAMs are connected to form an LUT cascade with a few additional circuits. The features of the LUT cascade include: 1) a flexible cascade connection structure, 2) multi phase pseudo asynchronous operations with synchronous static random access memory (SRAM) cores, and 3) LUT-bypass redundancy. This chip operates at 33 MHz in 8-LUT cascades at 122 mW. Benchmark results show that it achieves a comparable performance to field programmable gate array (FPGAs).
[DOI: 10.1143/JJAP.45.3295]

KEYWORDS: look-up table cascade, programmable logic device, SRAM

1. Introduction

Random access memories (RAMs) and programmable logic arrays (PLAs) are used for programmable logic devices (PLDs) that realize multiple-output combinational logic functions. However, when the number of inputs and/or outputs for the target function is large, these devices require excessive amounts of hardware. Alternatively, field programmable gate arrays (FPGAs) are often used. However, in FPGAs, the area and delay for interconnections among logic cells are much larger than those for logic elements, so the prediction of the performance of the FPGA is difficult without complete physical design. To solve these problems, a look-up table (LUT) cascade architecture that is composed of a serial connection of large-scale memories has been developed.[1,2] It is composed of a serial connection of large-scale memories. It requires a memory size that is only 1/100 to 1/1000 that of the conventional RAM realization. Since LUT cascades are realizable using memory technology, the design, test and production costs of LUT cascades should be quite low. We have developed the first implementation of the LUT cascade.[3] It was a straightforward implementation of LUT cascade connection with asynchronous static RAM (SRAM) cores. Unfortunately, its performance was not acceptable, particularly for power dissipation.

To improve the performance, we developed a second version. To achieve the competitive performance (area, speed, power and cost) for FPGAs, we developed several circuit techniques: 1) flexible cascade connection to increase the memory efficiency and free input/output (I/O) pin assignment, 2) 8/9 multi phase pseudo asynchronous operations with synchronous SRAM cores to achieve high-speed and low-power operations and 3) LUT-bypass redundancy to improve the chip yield.

2. LUT Cascade Architecture

Figure 1 shows a comparison of programmable logic architectures. An FPGA is composed of configurable logic blocks (CLBs) and the programmable interconnections among CLBs. Each CLB includes a 16- or 64-bit memory as an LUT, however, the chip performance is mainly determined by the configuration of interconnections. On the other hand, the LUT cascade uses relatively larger LUTs (1 K–1 Mbit), and the interconnections between LUTs are limited to the adjacent cells in the cascade. The LUT cascade structure is quite different from the structure of FPGAs with smaller LUTs. In the conventional FPGAs, the area for interconnection is fairly large, while in the LUT cascades, the area for interconnection is very small. In an LUT cascade, one programs the LUTs, whereas in an FPGA, one programs both the interconnections and the LUTs. The large area for the interconnections in an FPGA is compensated by the larger LUTs in the cascade.

In the basic LUT cascade structure as shown in Fig. 1(2), the data outputs (DOUT) of an LUT are directly connected to the lower address inputs (ADDL) of the adjacent LUT. The higher address inputs (ADDH) of each LUT and the lower address inputs (ADDL) of the first stage are used as the inputs of logic functions. The output of logic functions are obtained from the data output of the final LUT. The wires between adjacent cells are called “rails”.

3. Structure of LUT Block

Figure 2 shows a schematic of a single LUT block. Each LUT block consists of a 64 kbit synchronous SRAM core with 13b address inputs and 8b data I/O, and a few additional circuits: two 8b switches (SW1 and SW2), an 8b data register, a mode register and an 8b rail switch. The 8b rail switch selects either the signals from the previous LUT block or external inputs (EXT_IN). An LUT cascade can be implemented by a simple series connection of LUT blocks.

Each LUT block also has connections to the control signals used for programming and testing. An LUT block is simply selected by the block select signals (BS) and all address inputs of the LUT can be directly controlled by the external address inputs. In this mode, this chip works like a conventional memory: read and write operations can be performed through the common data bus lines. By taking advantage of this memory-compatible mode, we can reconfigure the LUT cascade by overwriting the memory contents and we can test this chip easily by the memory testing method.

In addition, we made a bypass switch (SW2) for redundancy from the cascade input to the output data

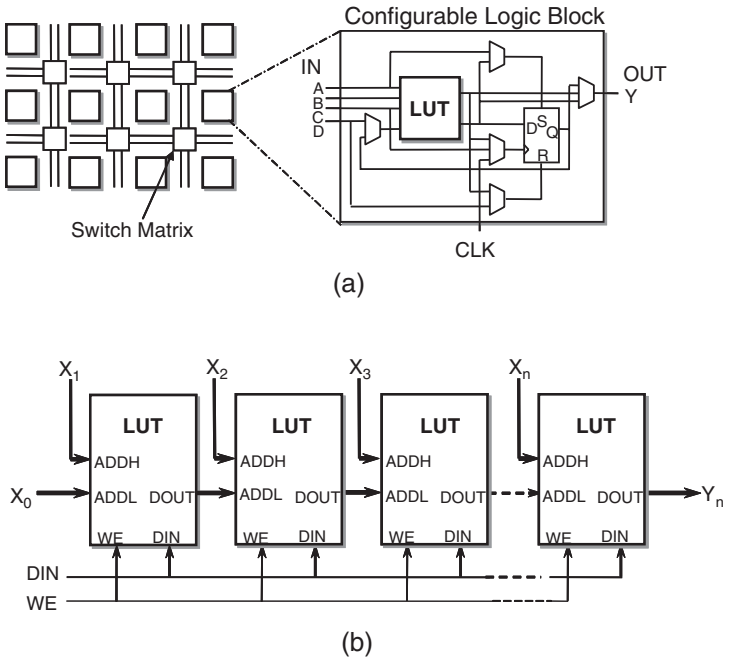

Fig. 1. Comparison of programmable logic architectures: (a) FPGA (LUT size: 16–64 bit) and (b) LUT cascade (LUT size: 1 K–1 Mbit).

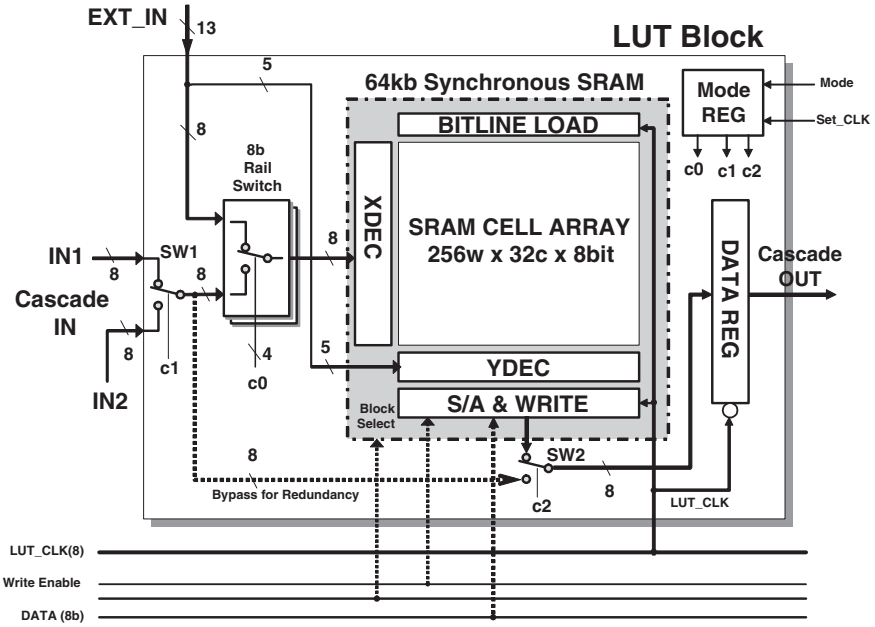

Fig. 2. LUT block.

registers. This just makes the input data skip to the next LUT block without accessing the memory array. Therefore, a faulty block can be bypassed to improve chip yield.

4. Structure of LUT Cascade

Figure 3 shows the block diagram of our implementation. It consists of eight LUT blocks. The LUT cascade LSI is simply realized by a cascade connection of LUT blocks. Each LUT block has a 64 K-bit memory, so this chip contains 512 K-bit memory cells. To increase memory efficiency and free I/O pin assignment, we developed a flexible cascade connection structure. In Fig. 2, the switch (SW1) selects one of two inputs (IN1 or IN2) to the cascade. This is connected to two adjacent LUT blocks, horizontally and vertically. As a result, the eight LUTs form a single loop, dual 4-LUT loops, or quadruple 2-LUT loops. With each loop structure, any LUT can be the first stage of the cascade, and this increases I/O pin assignment flexibility. As shown in Fig. 4, all the output Y terminals can be assigned to the upper positions by taking advantage of the vertical connections.

5. Pseudo Asynchronous Interleaved Operation

Since the memory in an LUT cascade operates as a data

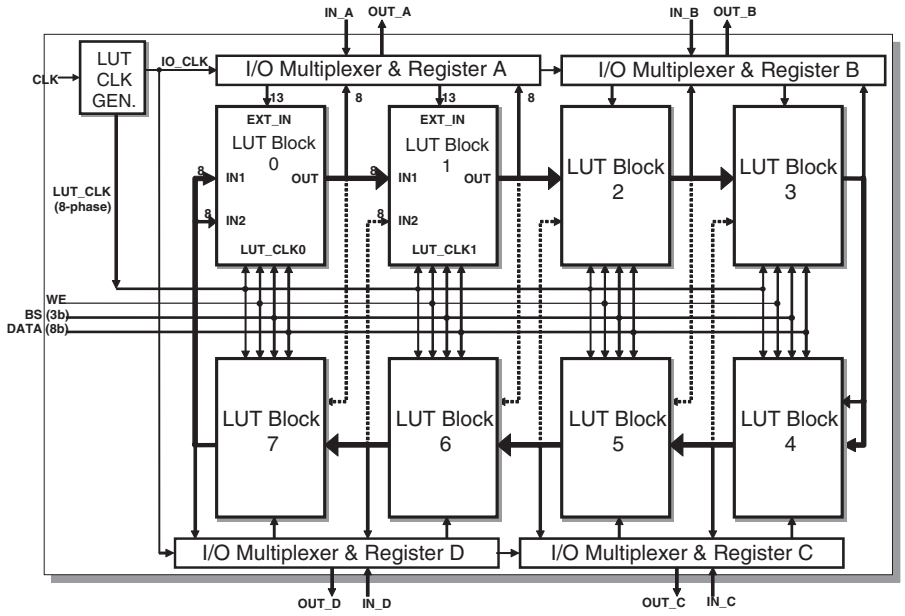

Fig. 3. Block diagram of LUT cascade LSI.

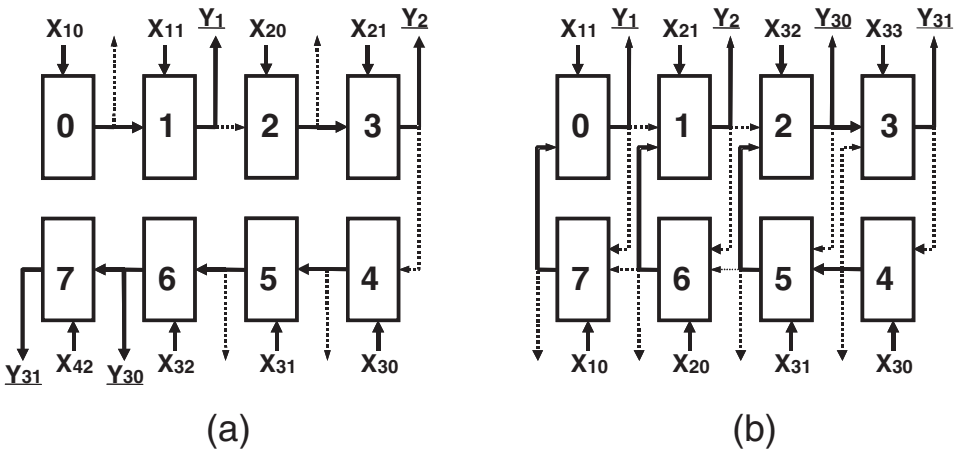

Fig. 4. Mapping examples of $4 + 2 + 2$ LUT-cascades: (a) conventional and (b) new (flexible cascade connection).

path, consecutive asynchronous memory access operation is required. We employed asynchronous SRAM for an LUT block in a previous version, however, it causes DC current flow in the memory cell and sense amplifier.[3)] In the LUT cascade operation, all memory blocks should operate simultaneously, so the total power dissipation in an LUT cascade LSI with asynchronous SRAMs will be too large. To solve the power and consecutive access problems, we developed a pseudo asynchronous interleaved operation with synchronous SRAMs using a multi phase clock. Figure 5 shows the developed 8/9 multi phase clock generator. First, the phase locked loop (PLL) generates a clock that is 9 times ($9\times$) the frequency of the original clock. Then, nine-phase non-overlap clock signals (p0–p8) are generated from the complement of the $9\times$ clock and IO_CLK. Here, eight signals (p0–p7) are selected for control clock signals for LUT blocks. The synchronous SRAM in each LUT block is operated in the "high" period of the LUT clock. The output data of the LUT block are latched into registers at the falling edge of each LUT clock (see Fig. 2). This 8/9 phase operation makes the data setup and hold timing margins (tm1 and tm2) 1/18 of the I/O clock cycle time among the I/O registers and the first and last LUT blocks as shown in Fig. 5.

6. Measurement Result

Figure 6 shows a photomicrograph of the LUT cascade LSI developed by the 0.35 μm standard CMOS logic process. The memory cell size is $5.2 \times 7.5\,\mu m^2$ (6Tr SRAM). This chip includes 512 K-bit cells and a PLL control clock generator. The chip size is $9.8 \times 9.8\,mm^2$ and its core size is $5.1 \times 7.1\,mm^2$. The ratio of the memory cell area to the core area (memory cell efficiency) is 52%. We verified that the conversion of a simple memory into the LUT cascade requires few additional circuits. It almost looks like a conventional large-scale memory. In an LUT block, 99.5% of the area is devoted to the SRAM circuit, while only 0.5% is devoted to switches and registers.

Figure 7(a) shows the simulated internal delay time distribution in the critical path. The latency of an internal

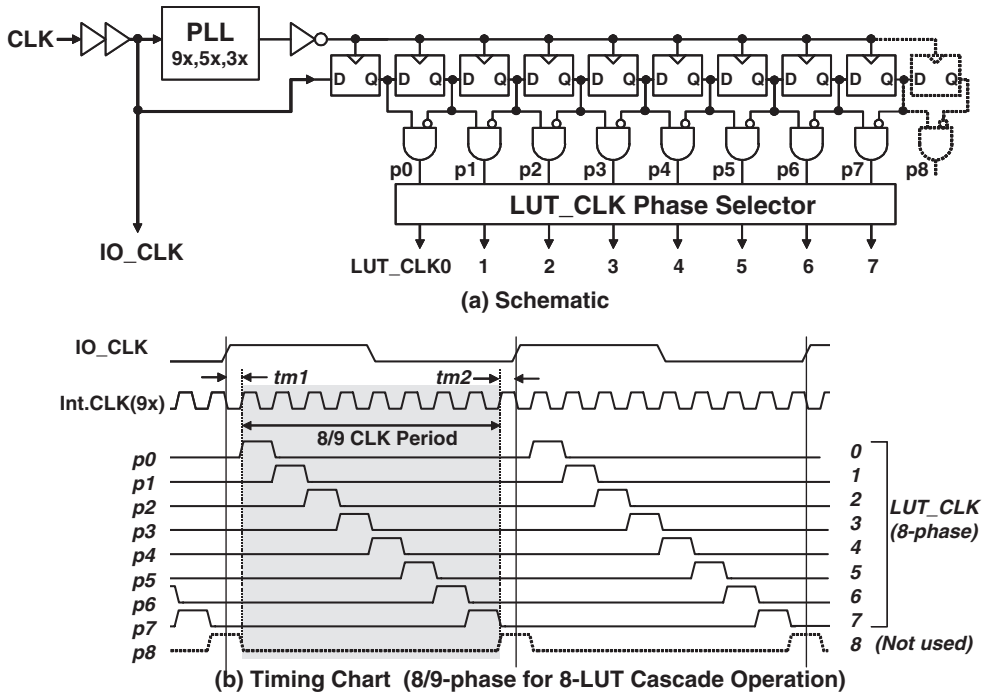

Fig. 5. Multi phase LUT-CLK generator: (a) schematic and (b) timing chart (8/9 phase for 8-LUT cascade operation).

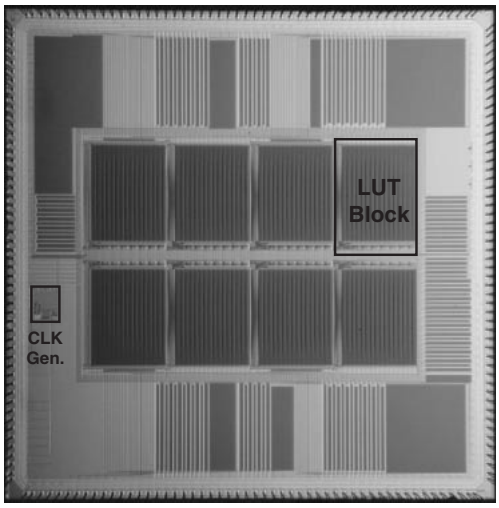

Fig. 6. Chip photomicrograph.

Table I. Operation modes and chip characteristics.

Operation mode	Multi-phase operation	Max. CLK (MHz)	Power (@Max. CLK) (mW)
8-LUT	8/9	33	122
Dual 4-LUT	4/5	61	221
Quad 2-LUT	2/3	100	401

LUT is 3.3-ns. Figure 7(b) shows the pseudo asynchronous operation in three operation modes. In the dual 4-LUT cascade mode, four operations in five phases are performed. The operating frequency of 33-MHz in the single 8-LUT cascade mode with 122 mW is experimentally confirmed. Table I summarizes the maximum operating frequency and power dissipation. Note that a design with asynchronous SRAMs dissipates about 10 times more power than this design.[3)]

7. Development System for LUT Cascades

The commonly used FPGA architecture is an "island-style" structure, where the array of logic blocks is surrounded by routing channels. Recent FPGAs use clusters. A cluster is group of basic logic elements that are fully connected by mux-based cross-bar switches.[4)] A recent study shows that LUT sizes of 4 or 6, and cluster size of between 3 and 10 provides the best area-delay product for an FPGA.[5)] In fact, the architecture's of FPGAs are becoming more and more complex. To design such FPGAs, we need various CAD tools: logic optimization,[6)] technology mapping,[7)] logic clustering, and placement and routing. Also, the design results heavily depend on these tools.[8)] However, the LUT cascade architecture is very simple and requires virtually no placement nor layout. In addition, cascade is directly generated from the BDDs. Thus, the design system of the LUT cascade is much simpler than that of FPGAs.

8. Performance Comparison with FPGA

To compare the performance (area, delay and power) of LUT cascades with FPGAs, we mapped simple benchmark functions to the LUT cascade and a commercial FPGA (Xilinx XCV50: 0.22 μm, five-layer metal, 2.5 V, 384 CLBs).[9,10)] We used commercial logic synthesis and layout tools for the design of the FPGA. On the other hand, for the design of the LUT cascade, we used our newly developed logic synthesis tool that converts binary decision diagrams (BDDs) into LUT cascades.[11)] Table II summarizes the experimental results. In LUT cascades, the area, the latency and the power can simply be estimated by the number of used LUTs. The power dissipations for LUT cascades are normalized for 20-MHz operation to compare with the

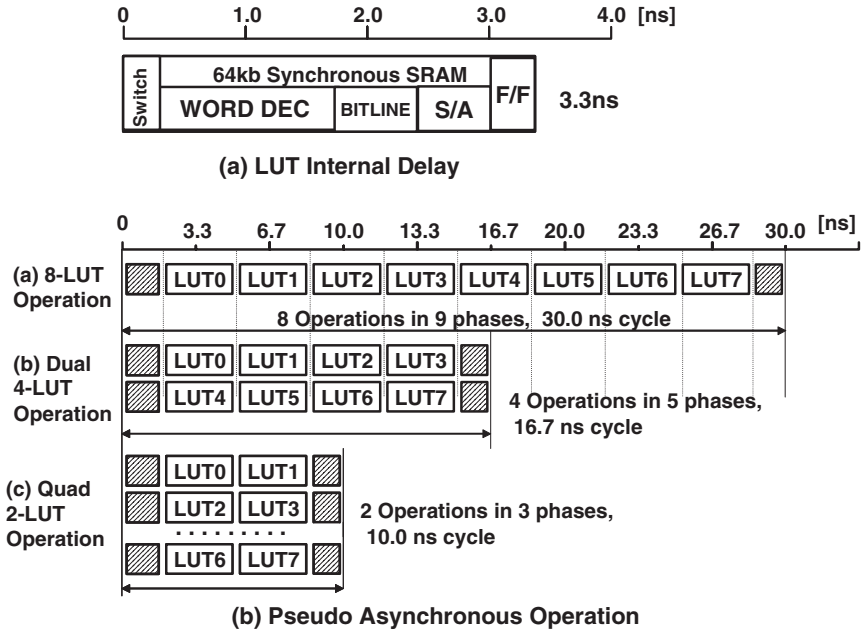

Fig. 7. Internal LUT operation: (a) LUT internal delay and (b) pseudo asynchronous operation.

Table II. Experimental result of function mapping.

Target function[1)]	No. of inputs	No. of outputs	LUT cascade (0.35 µm CMOS, 3.3 V)				FPGA (0.22 µm CMOS, 2.5 V)[2)]			
			No. of LUTs	Area (mm^2)	Delay (ns)	Power (mW@20 MHz)	No of CLBs	Area (mm^2)	Delay (ns)	Power (mW@20 MHz)
C432	36	7	6	27.2	23.1	71.6	98	7.7	35.6	56.7
b3	32	20	4	18.1	16.5	53.3	125	9.8	17.3	63.6
chkn	29	7	4	18.1	16.5	53.3	121	9.5	29.1	66.1
ibm	48	17	4	18.1	16.5	53.3	95	7.4	17.2	62.3
in5	24	14	4	18.1	16.5	53.3	118	9.2	18.2	52.0
in7	26	10	4	18.1	16.5	53.3	73	5.7	16.8	46.9
rckl	32	7	4	18.1	16.5	53.3	94	7.3	18.0	53.3
shift	19	16	4	18.1	16.5	53.3	76	5.9	14.8	52.2
vg2	25	8	3	13.6	13.2	44.1	69	5.4	16.2	46.8
x1dn	27	6	3	13.6	13.2	44.1	66	5.2	16.1	45.0
x6dn	39	5	4	18.1	16.5	53.3	119	9.3	19.7	67.5
x9dn	27	7	3	13.6	13.2	44.1	69	5.4	17.2	46.9
Ave.	30.3	10.3	3.9	**17.7**	**16.2**	**52.5**	93.6	**7.3**	**19.7**	**54.9**

results of FPGA. The areas for FPGA include not only areas for logic blocks but also the areas for interconnections, and the delay times for FPGA are results of physical layout. In spite of the disadvantage of the process technology, the LUT cascade achieves a comparable performance to the FPGAs.

By taking the difference in process technology into account, we can conclude that, by using the same process technology as the FPGA for the LUT cascade, we can achieve a comparable layout area with less delay time and less power dissipation.

9. Conclusions

The second generation of an LUT cascade LSI with a flexible cascade architecture, pseudo asynchronous operation and LUT-bypass redundancy scheme has been developed. We experimentally confirmed its competitive performance to FPGAs. With advanced high-density memory technologies, such as Gbit DRAM technologies, we can improve the area efficiency by a factor of 100 or higher.

In the future, LSI design will be much more time-consuming than before, since both logical and physical designs must be considered at the same time.[12)] The use of LUT cascades is one way of separating these complex problems. The LUT cascade LSI is a new and promising reconfigurable logic device for future sub-100 nm LSIs.

Acknowledgements

The chip has been fabricated through VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation. This study was supported by a fund from the Ministry of Education, Culture, Sports, Science and Technology of Japan via the Kitakyushu knowledge-based cluster project.

1) Y. Iguchi, T. Sasao and M. Matsuura: Int. Conf. Computer Design (ICCD2001), Sep. 2001, p. 388.
2) T. Sasao and M. Matsuura: Proc. 41st Design Automation Conf., June 2004, p. 428.
3) K. Nakamura, T. Sasao, M. Matsuura, K. Tanaka, K. Yoshizumi, H. Qin and Y. Iguchi: Dig. Cool Chips VIII, May, 2005.
4) V. Betz and J. Rose: IEEE Des. Test Mag. **15** (1998) 10.
5) E. Ahmed and J. Rose: IEEE Trans. VLSI Syst. **12** (2004) 288.
6) E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton and A. Sangiovanni-Vincentelli: Tech. Rep. No. UCB/ERL M92/41, University of California, Berkeley, 1992.
7) J. Cong and Y. Ding: IEEE Trans. Comput.-Aided Des. **13** (1994) 1.
8) A. Yan, R. Cheng and S. J. E. Wilton: ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays, Feb. 2002, p. 147.
9) MCNC-benchmark functions: http://www.cbl.ncsu.edu/www/
10) http://www.xilinx.com/
11) A. Mishchenko and T. Sasao: Int. Workshop on Logic and Synthesis (IWLS2002), June 2002, p. 115.
12) R. K. Brayton: in *Logic Synthesis and Verification*, ed. S. Hassoun and T. Sasao (Kluwer, Dordrecht, 2001).