A Memory-Based Programmable Logic Device Using a Look-Up Table Cascade with Synchronous SRAMs

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

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

A large-scale memory-technology-based programmable logic device (PLD) using LUT (Look-Up Table) cascade is developed in 0.35um Standard CMOS logic process. Eight 64K-bit synchronous SRAMs are connected to form an LUT cascade with a few additional circuits. The features of the LUT cascade include: 1) flexible cascade connection structure, 2) multi-phase pseudo-asynchronous operations with synchronous SRAM cores, 3) LUT-bypass redundancy. This chip operates at 33MHz in 8-LUT cascades with 122mW. Benchmark results show that it achieves a comparable performance to FPGAs.
KEYWORDS: Look-Up Table Cascade, Programmable Logic Device, SRAM

1. Introduction

RAMs and PLAs (Programmable Logic Arrays) are used for PLDs (Programmable Logic Devices) 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, FPGAs (Field Programmable Gate Arrays) 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, an 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 memory size that is only 1/100 to 1/1000 of the straightforward RAM realization. Since LUT cascades are realizable by using the 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 SRAM cores. Unfortunately, its performance was not acceptable especially for power dissipation.

In order to improve the performance, we developed a second version. To achieve competitive performance (Area, Speed, Power and Cost) to FPGAs, we developed several circuit techniques: 1) flexible cascade connection to increase the memory efficiency and free 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.

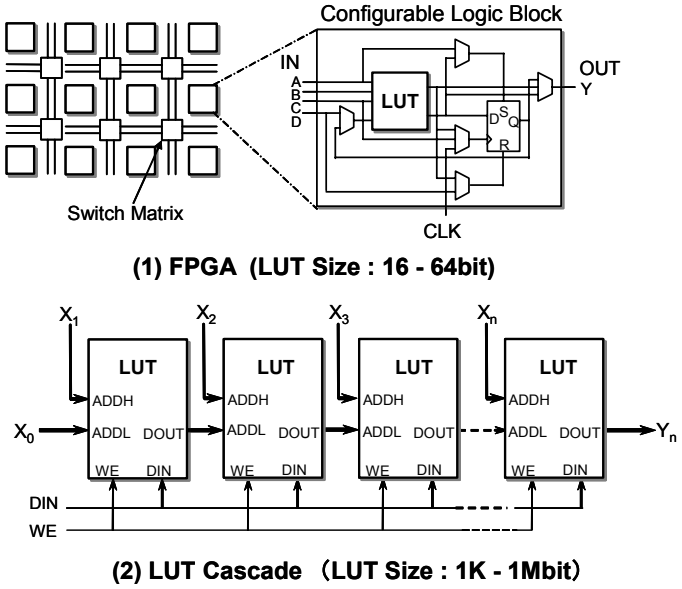


Fig. 1 Comparison of Programmable Logic Architecture

2. LUT Cascade Architecture

Figure 1 compares programmable logic architecture. An FPGA is composed of configurable logic blocks (CLBs) and the programmable interconnections among CLBs. The CLB includes 16-bit or 64-bit memory as a LUT, however, the chip performance is mainly determined by the configuration of interconnections. On the other hand, the LUT cascade uses relatively larger LUTs (1K-bit - 1M-bit), 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 and 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 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 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".

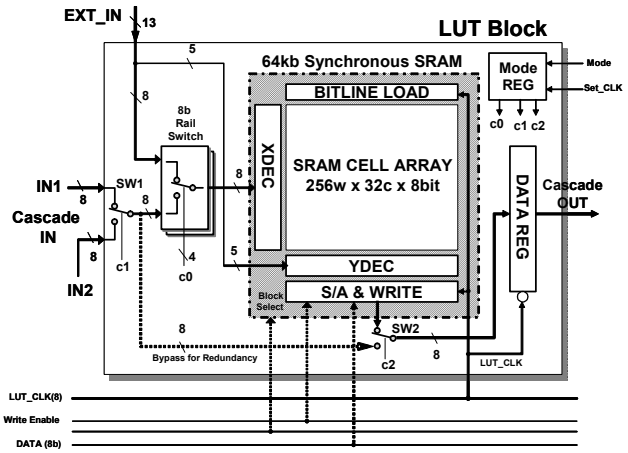


Fig. 2 LUT Block

3. Structure of LUT Block

Figure 2 shows the detail of a single LUT block. Each LUT block consists of a 64kbit synchronous SRAM core with 13b address inputs and 8b data I/O, and a few additional circuits: two 8b switches (SW1, SW2), 8b data register, the mode register and an 8b rail switch. The 8b rails 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 the 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 re-configure the LUT cascade by overwriting the memory contents and we can test this chip easily by memory testing method.

In addition, we made a bypass switch SW2 for redundancy from the cascade input to the output data 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 the chip yield.

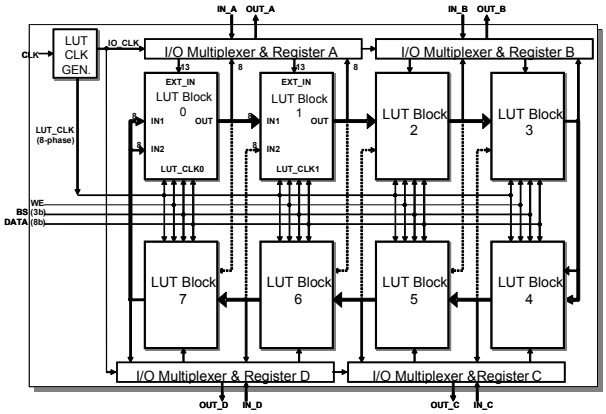


Fig.3 Block Diagram of LUT Cascade LSI

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 64K-bit memory, so this chip contains 512K-bit memory cells. In order 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,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 outputs Y terminals can be assigned in the upper positions by taking advantage of the vertical connections.

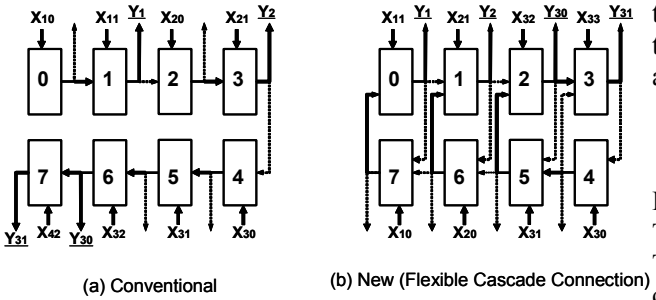


Fig.4 Mapping Examples of 4+2+2 LUT-Cascades

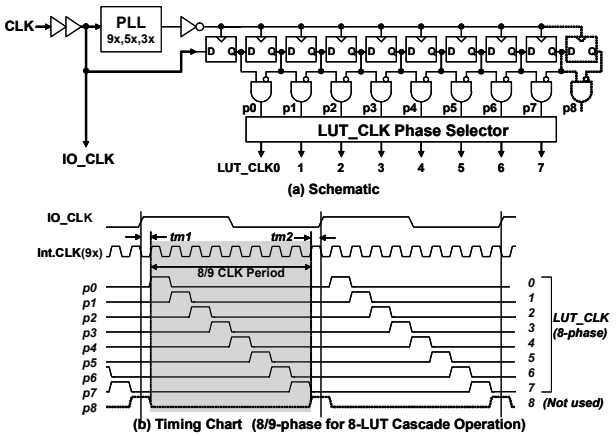


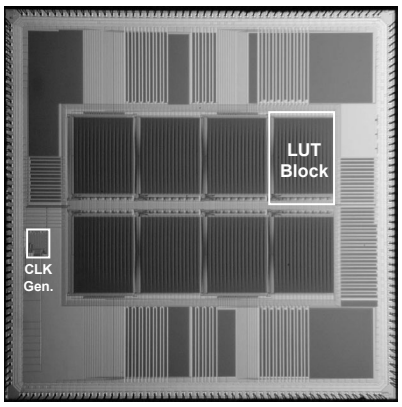
Fig.5 Multi-phase LUT-CLK Generator

5. Pseudo-asynchronous Interleaved Operation

Since the memory in an LUT cascade operates as a data-path, consecutive asynchronous memory access operation is required. We employed asynchronous SRAM for an LUT block in 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. In order 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 PLL generates a clock that is 9 times (9x) the frequency of the original clock. Then 9-phase non-overlap clock signals (p0-p8) are generated from the complement of the 9x clock and IO_CLK. Here, 8 signals (p0-p7) are selected for control clock signals for LUT blocks. 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,tm2) 1/18 of the I/O clock cycle time among the I/O registers and the first LUT block and the last LUT block as shown in Fig.5.

6. Measurement Result

Figure 6 shows a photomicrograph of the LUT cascade LSI developed by 0.35um standard CMOS logic process. The memory cell size is 5.2um x 7.5um (6Tr SRAM). This chip includes 512K-bit cells and a PLL control clock generator. The chip size is 9.8mm x 9.8mm and its core size is 5.1mm x 7.1mm. The ratio of the memory cell area to the core area (memory cell efficiency) is 52%. We verified that 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.



(Core Size : 5.1mm x 7.1mm, Chip Size : 9.8mm x 9.8mm, 208pins)

Fig.6 Chip Photomicrograph

Figure 7(a) shows the simulated internal delay time distribution in the critical path. The latency of an internal LUT is 3.3ns. Figure 7(b) shows the pseudo asynchronous operation in three operation modes. In the dual 4-LUT cascade mode, 4 operations in 5 phases are performed. The operating frequency of 33MHz in the single 8-LUT cascade mode with 122mW is experimentally confirmed. Table 1 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].

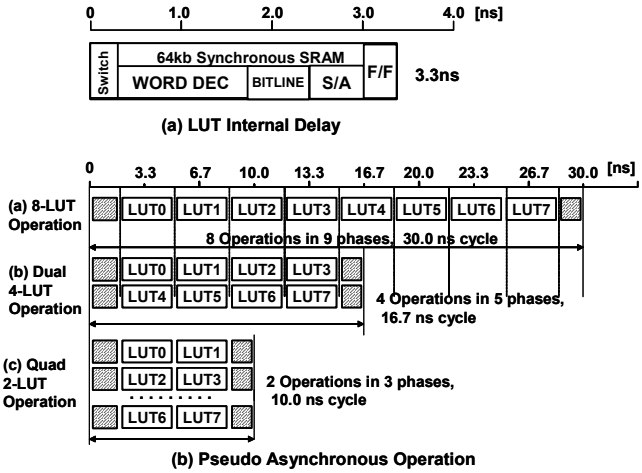


Fig.7 Internal LUT Operation

Table 1 Operation Modes and Chip Characteristics

Operation Mode	Multi-Phase Operation	Max. CLK	Power (@Max. CLK)
8-LUT	8 / 9	33MHz	122mW
dual 4-LUT	4 / 5	61MHz	221mW
Quad 2-LUT	2 / 3	100MHz	401mW

7. Development System for LUT Cascades

The commonly used FPGA architecture is an "island-style" structure, where array of logic blocks are surrounded by routing channel. Recent FPGA use clusters. A cluster is group of basic logic elements that are fully connected by a mux-based cross bar switches [4]. 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 LUT cascade is mush simpler than that of FPGAs.

8. Performance Comparison with FPGA

To compare the performance (area, delay, power) of LUT cascades with FPGAs, we mapped simple benchmark functions to the LUT cascade and a commercial FPGA (Xilinx XCV50: 0.22um, 5-Layer metal, 2.5V, 384 CLBs) [9] [10]. We used commercial logic synthesis and layout tools for the design of FPGA. On the other hand, for the design of the LUT cascade, we used our newly developed logic synthesis tool that converts BDDs (Binary Decision Diagrams) into LUT cascades [11]. Table 2 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 20MHz operation to compare with the 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 process technology, the LUT cascade achieves comparable performance to the FPGAs.

Table 2 Experimental Result of Function Mapping

Target Function [1]	No. of inputs	No. of Outputs	LUT Cascade (0.35um CMOS, 3.3V)				FPGA (0.22um CMOS, 2.5V) [2]			
			No. of LUTs	Area [mm²]	Delay [ns]	Power [mW@20MHz]	No of CLBs	Area [mm²]	Delay [ns]	Power [mW@20MHz]
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**

By taking the difference of 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. Conclusion

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 the 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]. LUT cascades are one method to separate these complex

problems. The LUT cascade LSI is a new and promising reconfigurable logic device for future sub-100nm LSIs.

Acknowledgement

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 work was supported by fund from the Japanese Ministry of MEXT via Kitakyushu knowledge-based cluster project

References

[1] Y.Iguchi, T.Sasao, and M.Matsuura, "Realization of multiple-output functions by reconfigurable cascades", International Conference on Computer Design (ICCD2001), Sep.2001, pp388-393.

[2] T.Sasao and M.Matsuura, "A method to decompose multiple-output circuits by using binary decision diagrams", Proc. of 41st Design Automation Conference, Jun. 2004, pp.428-433.

[3] K.Nakamura et,al., " Programmable Logic Device with an 8-stage cascade of 64K-bit Asynchronous SRAMs ", Digest of Cool Chips VIII, May. 2005.

[4] V. Betz and J. Rose, "How much logic should go in an FPGA logic block?", in IEEE Design & Test Magazine, Vol. 15, No. 1, Jan-March 1998, pp. 10-15.

[5] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deep-submicron FPGA performance and density," in IEEE Trans. on VLSI, Vol 12, No. 3, March 2004, pp. 288-298.

[6] E. M. Sentovich et. al., SIS: A System for Sequential Circuit Analysis, Tech. Report No. UCB/ERL M92/41. University of California, Berkeley, 1992.

[7] J. Cong and Y. Ding, "FlowMap: An optimal technology map- ping algorithm for delay optimization in lookup-table based FPGA designs," IEEE Trans. CAD, 13(1), pp.1-12, June, 1994.

[8] A. Yan, R. Cheng, S.J.E. Wilton, ``On the sensitivity of FPGA architectural conclusions to the experimental assumptions, tools, and techniques", in the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, Feb. 2002, pp. 147-156.

[9] MCNC-benchmark functions: http: //www.cbl.ncsu.edu /www/

[10] http://www.xilinx.com/

[11] A. Mishchenko and T. Sasao, "Encoding of Boolean functions and its application to LUT cascade synthesis, " International Workshop on Logic and Synthesis (IWLS2002), New Orleans, Louisiana, June 4-7, 2002, pp.115-120.F

[12] R. K. Brayton, “The future of logic synthesis and verification,” in S. Hassoun and T. Sasao (eds.), Logic Synthesis and Verification, Kluwer, Nov. 2001.