CMOS Array Logic Architectures Using Dual-Rail Threshold Logic Circuits

(2線式しきい値論理回路を用いた配列型CMOS論理アーキテクチャに関する研究)

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
ELECTRONIC ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES OF
THE UNIVERSITY OF TOKYO
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

by

Hiroaki Yamaoka

Supervisor: Professor Kunihiro Asada
December 2003

Abstract

Continued advances in integrated circuit (IC) technologies, along with the development of more sophisticated computer-aided design (CAD) tools, enable an increase in the level of integration of silicon chips. By integrating more and more circuits into a single chip, system performance improvements have been achieved. However, the dramatic increase in complexity of contemporary ICs poses an enormous design challenge. Designing a multimillion-transistor circuit and ensuring that it operates correctly when the first silicon returns are daunting tasks that are virtually impossible without the help of well-established design methodologies. Although the recent development of CAD tools offers more complicated IC designs, technology advancements are outpacing the absorption bandwidth of design community.

To address this issue, regular array logic architectures are being explored recently. Although the standard-cell-based design approach, which produces highly area-efficient IC layouts in random structures, is prevailing as the design strategy along with the advent of sophisticated CAD tools, regularity-based circuit implementations of complex functions are becoming an option to designers to reduce design effort and meet the cost and time-to-market goals. The design methodology using regular circuits significantly reduces the iterations of the design process because structured arrays give predictable area, delay, and power consumption early in the design process. However, although the structural regularity of the circuits offers design simplicity, regular circuits generally require larger chip area than random logic implementations realized by the conventional standard-cell-based design approach.

In this thesis, we focused on CMOS array logic architectures, and proposed new ar-

chitectures and their design methodologies to address the drawback of the area-efficiency of regular circuits. Also, we considered low-swing signaling techniques by taking advantage of the regularity of the architectures in order to realize high-speed and low-power operations.

First, we proposed a new array logic architecture based on a programmable logic array (PLA) structure using dual-rail threshold logic circuits. The architecture can realize arbitrary Boolean functions expressed in sum-of-products forms. The threshold logic circuit has a new circuit configuration using a charge sharing scheme and a self-precharge scheme for high-speed and low-power operations. As an application of the proposed array logic architecture, a 32-bit binary comparator was designed and fabricated using a 0.6-μm CMOS technology, and the proposed circuit achieved reductions of cycle time by 20.0%, 45.5%, and 22.6%, and reductions of power-delay product by 32.9%, 34.7%, and 15.4% compared to a domino logic circuit, the conventional single-rail PLA, and the conventional high-speed dual-rail circuit, respectively. The capabilities of high-speed and low-power operations over the conventional array logic architectures can be enhanced as the number of input signals increases and thus the proposed architecture is shown to be more effective in high data bandwidth systems in the future. Also, we investigated the circuit characteristics of single-rail and dual-rail circuits, and showed that dual-rail circuits have advantages in terms of common-mode noise immunity and leakage-current tolerance, especially in advanced CMOS process technologies.

In order to reduce the circuit area of dual-rail array logic architectures, an area-efficient dual-rail array logic architecture, a logic-cell-embedded PLA (LCPLA), was presented. By embedding 2-input logic cells, which realize arbitrary 2-input Boolean functions, in the structure, some classes of logic functions can be implemented efficiently, so that high-speed and low-power operations were also achieved. The logic cells can be designed by connecting some local wires and do not require additional transistors over logic cells of the conventional dual-rail PLA. The advantages over the conventional PLAs and standard-cell-based designs were demonstrated by using benchmark circuits and a developed logic synthesis method, and the LCPLA is shown to be effective to re-

duce the number of product terms. In a structure with a 64-bit input and a 1-bit output including 220 product terms, the LCPLA using a 0.35-μm CMOS technology achieved an area reduction by 36.0% compared to the above-mentioned proposed dual-rail PLA, and the power-delay product was reduced by 74.6% and 46.0% compared to the conventional high-speed single-rail PLA and the above-mentioned proposed dual-rail PLA, respectively. Also, a module generator for LCPLA structures was developed as a design automation tool. The generator performs equation-based transistor sizing to achieve a desired delay goal and allows a wide range of performance to be traded for area. The framework and procedure of the generator are much easier than those of the conventional tools for standard-cell-based designs and thus the generator can be easily applied to complicated IC designs.

As physical design methods to reduce the delay and circuit area of dual-rail array logic architectures, we proposed high-speed and area-efficient design techniques using a divided column scheme and an interdigitated column scheme. As applications of the proposed architecture, a comparator, a priority encoder, and an incrementor for 128-bit data processing were designed, and the proposed schemes achieved a 22.2% delay reduction and a 37.5% area reduction on average over the above-mentioned proposed dual-rail PLA in a 0.13-μm CMOS technology. The high-speed capability of the proposed schemes can be enhanced as the number of input signals increases and thus the proposed architecture is shown to be more effective in high data bandwidth systems in the future. Moreover, a lot of circuit elements can be shared among different macro designs, thus reducing the design complexity significantly.

As an extension of the above-mentioned proposed array logic architectures, new array logic architectures using dual-rail multiple-threshold logic circuits and logic synthesis methods for their architectures were presented. Since the proposed architectures can be optimized in column-circuit level, it is also possible to build the proposed circuit into the above-mentioned proposed architectures and the conventional architectures, and thus further reductions of the number of product terms can be achieved. The advantages of the proposed architectures were demonstrated by using benchmark circuits, and the results

show that the number of product terms of the conventional AND-OR-type PLA structure can be reduced by 22.4% on average by using the proposed architecture. Also, the number of product terms, chip area, and power-delay product of LCPLA can be reduced by 12.7%, 11.5%, and 13.9% on average, respectively, by using a hybrid architecture of the proposed circuit and LCPLA in a 0.35-μm CMOS technology. The architecture of a field-programmable logic array based on the multiple-threshold logic circuits was also proposed and is shown to be effective to implement Boolean functions efficiently. Also, as an application of the multiple-threshold logic circuit, we proposed a new functional memory with the capability of high-speed Hamming-distance-based data search. A 32 $\times$ 128-bit memory macro was designed and fabricated using a 0.35-μm CMOS technology, and high-speed and low-power operations were achieved. Also, we verified a search function of stored data within a Hamming distance of up to 7 from an input data.

Acknowledgments

During my years in graduate school, I have been very fortunate to be surrounded by teachers, family, friends, and colleagues who have continually offered me support and encouragement. I would like to thank, from the bottom of my heart, for their contributions to my professional and personal growth.

I would like to sincerely thank Prof. Kunihiro Asada for his keen insight, guidance, and patience throughout the course of my research. His wisdom and knowledge, both in depth and in breadth; his enthusiasm for teaching and research; the freedom he provides for defining and directing my research; and the time he is willing to spend with me on my research have made the graduate school years a great learning experience and led me to become a full-fledged member of society. I feel very fortunate to have had him as my adviser, and I will forever cherish this experience as an invaluable guidance in my life.

I am deeply grateful to Prof. Makoto Ikeda for discussions on my research, for having patience with my frequent diversions, and for providing a comfortable research environment. He enlightened me the importance of enthusiasm for research and always gave me appropriate guidance when I was in trouble in various situations.

I would like to thank all the members of Asada-Ikeda laboratory for their support, pleasant discussions, and humor.

I would like to acknowledge Prof. Tadashi Shibata, Prof. Takayasu Sakurai, Prof. Masahiro Fujita, and Prof. Minoru Fujishima for many helpful discussions as well as for serving on both my orals and reading committees.

I also wish to thank all the members of VLSI Design and Education Center (VDEC) for their support in prototype chip fabrication. The VLSI chips in this research have

been designed with CAD tools of Synopsys, Inc. and Cadence Design Systems, Inc., and fabricated through the chip fabrication program of VDEC in collaboration with Rohm Corp. and Toppan Printing Corp.

Finally, I would like to express my greatest appreciation to my family for their constant support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Design Productivity Crisis

Continued advances in integrated circuit (IC) technologies, along with the development of more sophisticated computer-aided design (CAD) tools, enable an increase in the level of integration of silicon chips. By integrating more and more circuits into a single chip, system performance improvements have been achieved [1]. However, the dramatic increase in complexity of contemporary ICs poses an enormous design challenge. Designing a multimillion-transistor circuit and ensuring that it operates correctly when the first silicon returns are daunting tasks that are virtually impossible without the help of well-established design methodologies. Although the recent development of CAD tools offers more complicated IC designs, it does not keep up with the increase in complexity. In fact, it has often been suggested that technology advancements are outpacing the absorption bandwidth of design community. This is mentioned in Figure 1.1, which shows how IC complexity in logic transistors is growing faster than the productivity of a design engineer and is creating a *design productivity gap* [1]. One way to address this gap is to increase steadily the size of the design teams working on a project. This trend can be often observed in the design projects of high-performance microprocessors. However, it is

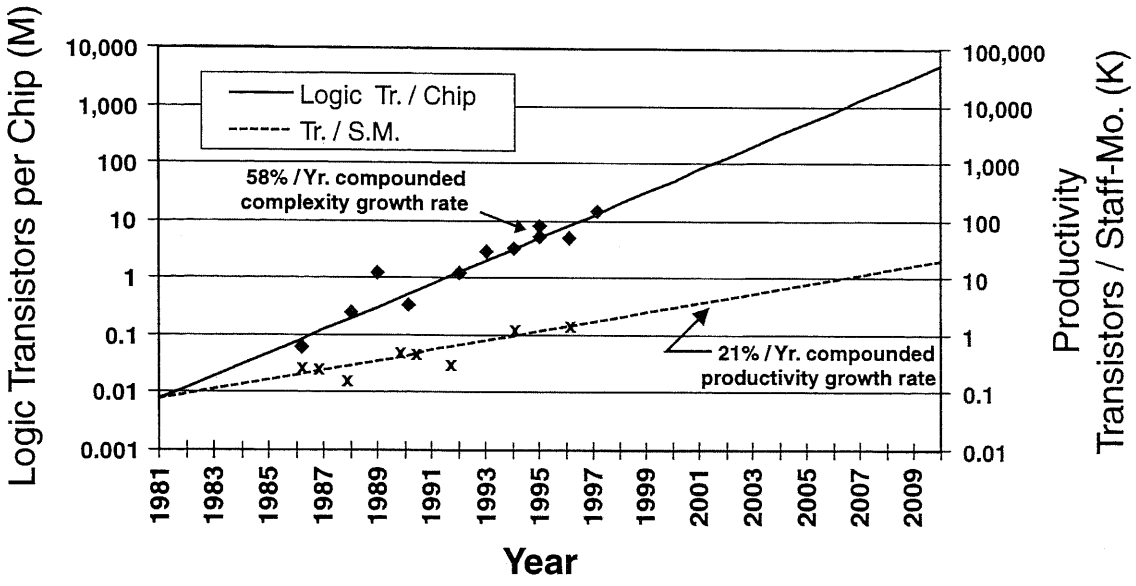


Figure 1.1 Design productivity gap [1].

obvious that this approach can not be sustained in the long term because of the limitation of the number of design engineers in the world.

1.1.2 Design Issues in Deep Sub-Micron ICs

Since the custom-design approach proved to be prohibitively expensive, a wide variety of design approaches have been introduced over the years to shorten and automate the design process. This automation comes at the price of reduced integration density and/or performance.

In recent IC designs, the standard-cell-based design approach, which produces highly area-efficient IC layouts in random structures using logic gates in a library, has become immensely popular and is used for the implementation of virtually all logic elements. The only exceptions are when extreme high performance or low energy consumption is needed, when the structure of a targeted function is very regular such as memory circuits and arithmetic circuits, or when the field-programmability is required. Figure 1.2 shows the semi-custom design flow using the standard-cell-based design approach. The success of the standard-cell-based design approach can be attributed to a number of developments,

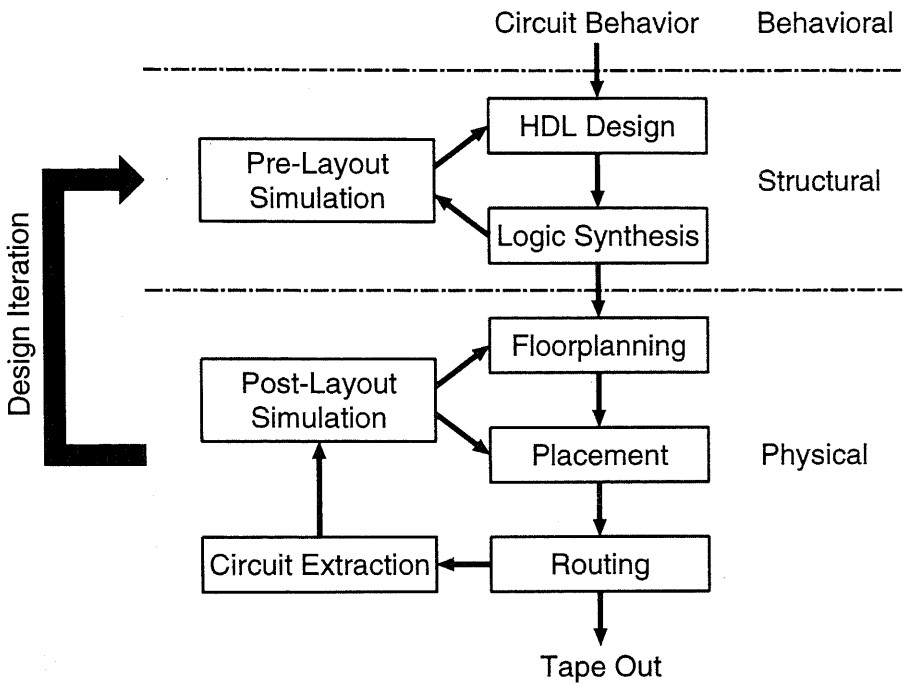


Figure 1.2 Semi-custom design flow.

including the increased quality of automatic placement and routing tools, and the advent of sophisticated logic synthesis tools.

While the design flow has served us for many years, it was found to be severely lacking once technology reached the 0.25-μm CMOS boundary. With design technology proceeding into the deep sub-micron region, layout parasitics, especially from the interconnect, are playing an increasingly important role. Prediction models used by logic and structural synthesis tools have a hard time providing accurate estimates for these parasitics. The chances that the generated design meets timing constraints at the first try are thus very small. The designer is then forced to go through a number of costly iterations of synthesis followed by layout generation until an acceptable artwork that meets the timing constraints is obtained. Each of these iterations may take several days. Just routing a complex chip can take a week on the most advanced computers. The number of needed iterations continues to grow with the scaling of technology. This is called a *timing closure problem* [2] and made it obvious that new solutions and a change in design technology

were required.

To address this problem, regular array logic architectures are being explored recently [3]–[6]. Although the standard-cell-based design approach is prevailing as the design strategy, regularity-based circuit implementations of complex functions are becoming an option to designers to reduce design effort and meet the cost and time-to-market goals. The design methodology using regular circuits significantly reduces the iterations of the design process because structured arrays give predictable area, delay, and power consumption early in the design process. The design, therefore, does not suffer from the timing closure problem. In addition, the issues of signal integrity such as cross-talk noise, which is becoming increasingly serious in deep sub-micron ICs, can be easily predicted and alleviated by taking advantage of the regularity [5]. This enables to shrink device sizes, and a merger of structural and physical design techniques can be achieved in design automation tools.

In the next section, the conventional CMOS array logic architectures are reviewed.

1.2 Conventional CMOS Array Logic Architectures

1.2.1 Programmable Logic Arrays

Figure 1.3 shows a structure of a programmable logic array (PLA). A PLA consists of an AND-plane and an OR-plane, and is capable of implementing arbitrary Boolean functions expressed in sum-of-products forms by connecting switching transistors, which are denoted by dots in the figure, to the input and output wires. Translating a set of two-level logic functions into a physical design, i.e., a task to decide where to place transistors in both the AND-plane and the OR-plane is called *programming*. This task is easily automated, hence the early success of PLAs. Also, PLAs have memory-like array structures and thus are often called *array logic* [7]. The circuit provides the design simplicity by taking advantage of the regularity and also makes the cell design simple because it only requires a small number of circuit components. This offers the process portability as well.

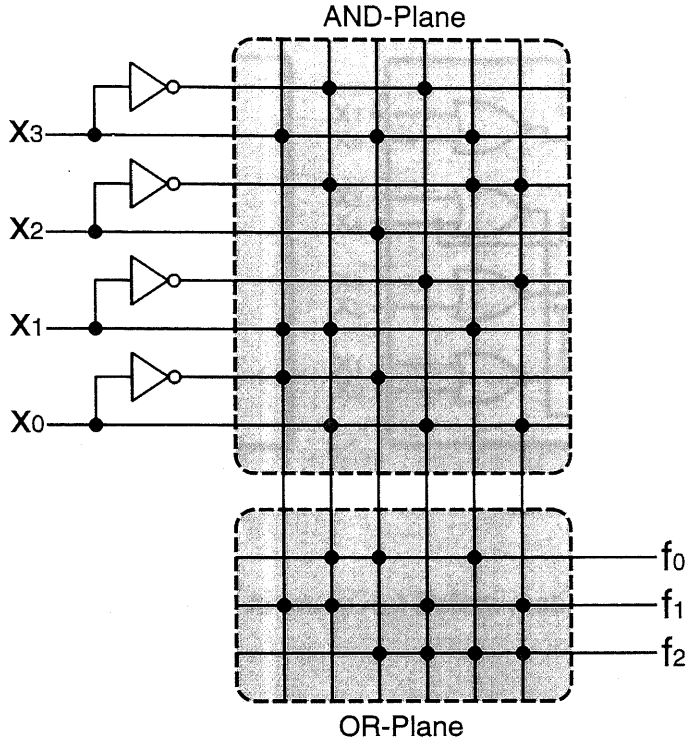


Figure 1.3 Structure of a PLA.

In the past three decades, PLAs have been widely used for combinational and sequential logic circuits because of their simplicity, regularity, and flexibility. These features are being accepted in recent complicated ICs. For example, control logic of the IBM 1-GHz 64-bit PowerPC processors, *guTS* [3] and *Rivina* [4], has been realized in PLAs to reduce the design complexity. The design methodology using PLAs does not require technology mapping, placement, and routing, so that it significantly reduces the iterations of the design process and does not suffer from the timing closure problem. A logic minimization of PLAs can be achieved by ESPRESSO [8], [9], which is a strong two-level logic minimization program. In addition, the issues of signal integrity such as cross-talk noise can be easily predicted and alleviated by taking advantage of the regularity of PLAs [5].

Although the structural regularity of PLAs offers the design simplicity, PLAs generally require larger chip area than random logic implementations realized by standard-cell-based design. To overcome this drawback, a network structure of small PLAs which

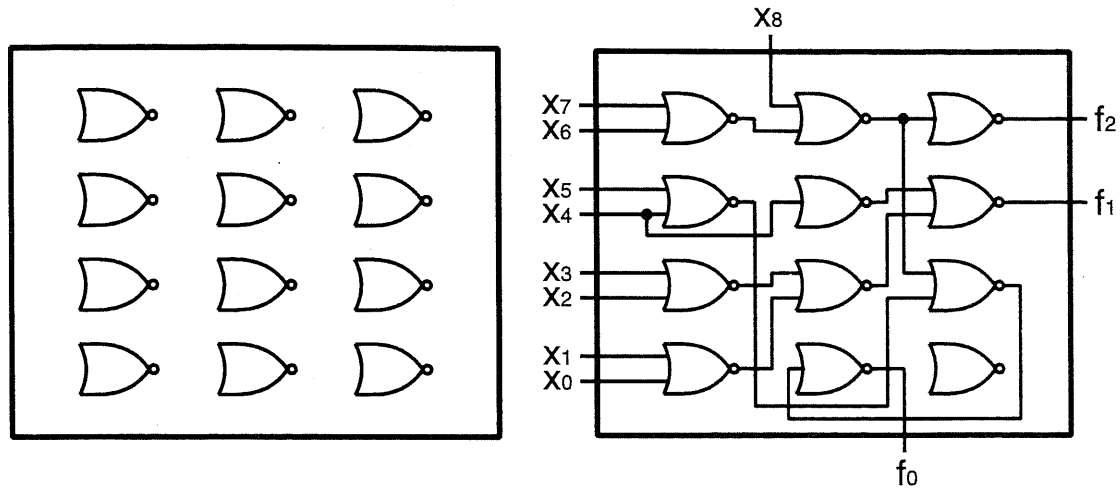


Figure 1.4 Typical gate array structure. (a) Before routing and (b) after routing.

implements Boolean functions efficiently has been proposed [5]. This structure, however, degrades circuit performance due to an increase in logic levels. Besides, although each PLA is surely regular, the global regularity is sacrificed because of placement and routing. Actually, it may be even worse than that in standard-cell-based design which has a row structure, and thus it is difficult to take advantage of the regularity of PLAs.

1.2.2 Gate Arrays

A gate array is an IC chip on which primitive gates are prediffused in matrix form without connections among the gates as shown in Figure 1.4(a) [10]. To transform this prediffused wafer into an actual design, only the desired wires and contact windows are added as shown in Figure 1.4(b), determining the overall function of the chip with only a few metallization steps. These metallization layers can be designed and applied to the prediffused wafer rapidly, thus reducing the turnaround time.

However, the design methodology of gate arrays utilizes random placement, which maps standard cells onto an array of primitive gates, and random routing similar to the standard-cell-based design approach. Thus, although the gate structure is surely regular,

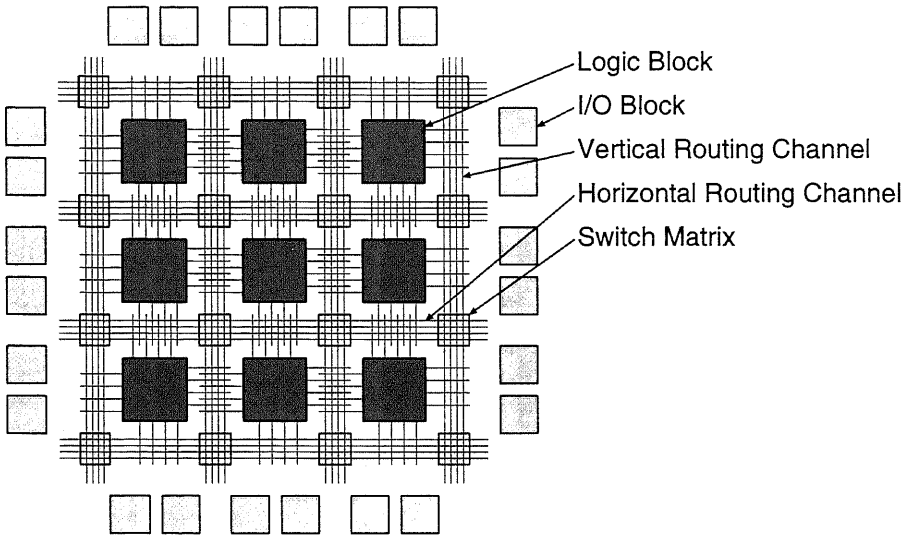


Figure 1.5 Typical FPGA structure.

wiring structures connecting each primitive gate are not regular. As a result, it is difficult to take advantage of the regularity and the design suffers from the timing closure problem with the scaling of technology.

1.2.3 Field-Programmable Logic Arrays

While the prediffused arrays offer a fast road to implementation, it would be even more efficient if dedicated manufacturing steps could be avoided altogether. This leads to the concept of the preprocessed die that can be programmed in the field (i.e., outside the semiconductor foundry) to implement a set of given Boolean functions. Such a programmable, prewired array of cells is called a *field-programmable logic device (FPLD)*. The advantage of this approach is that the manufacturing process is completely separated from the implementation phase and can be amortized over a large number of designs. The implementation itself can be performed at a user site with negligible turnaround time.

There are two major types of FPLDs: *field-programmable gate arrays (FPGAs)* and *complex programmable logic devices (CPLDs)*. Figure 1.5 shows a typical FPGA structure [11], [12], which is composed of four major components: logic blocks, routing

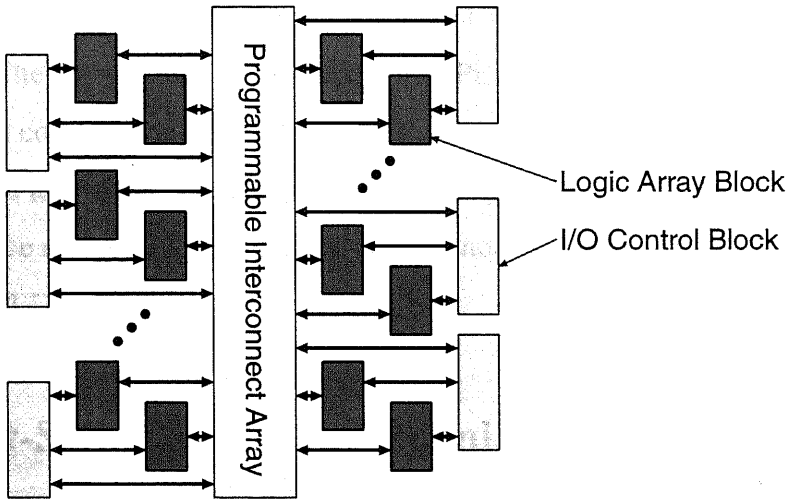


Figure 1.6 Typical CPLD structure.

channels, switch matrix, and I/O blocks. Most logic blocks are based on look-up-tables (LUTs), and some FPGAs have multiplexer-based or gate-based logic blocks. Their success has been propelled by a great deal of algorithmic study and tool development [13] in terms of logic synthesis, cell placement, and signal routing.

The structural regularity in logic blocks and signal wires of FPGAs offers high predictability of performance compared to random logic implementations realized by the standard-cell-based design approach. The major drawback of this technique is a loss in performance and circuit density compared to the more customized approach described in the previous subsections.

1.2.4 Complex Programmable Logic Devices

Similar to FPGAs, CPLDs are widely used as FPLDs. Figure 1.6 shows a typical CPLD structure [11], [12], which is composed of three major components: logic array blocks, a programmable interconnect array, and I/O control blocks. Most CPLDs are based on PLA-style logic array blocks. PLAs are considered as coarse-grained logic cells compared to logic cells in FPGAs because they typically have a large number of inputs and outputs. Although CPLDs provide only medium density, they are generally faster than FPGAs

because PLAs are much larger than LUTs, so the CPLD implementation results in fewer logic levels. The worst-case delay incurred by CPLDs also tends to be more predictable because PLAs communicate directly through a crossbar-like programmable interconnect array. As more and more logic gates are being integrated into a single chip and the drive for performance requires high chip speed and low noise, PLA-based CPLDs show a bright future in the PLD industry.

1.3 Low-Swing Signaling Techniques in CMOS Array Logic Architectures

Low-swing signaling techniques have been widely used in memory circuits, such as SRAMs and DRAMs, with sense amplifiers which detect the small voltage difference between bit-lines and amplifies it to a rail-to-rail signal. Sense amplifiers well suit regular array structures and dual-rail configurations of memory circuits. Power consumption P and propagation delay D of CMOS circuits are approximately given by [14]

$$P = p_t \cdot f_{CLK} \cdot C_L \cdot V_S \cdot V_{DD} + I_{LEAK} \cdot V_{DD} \tag{1.1}$$

$$D = \frac{\gamma \cdot C_L \cdot V_S}{\beta \cdot (V_{DD} - V_{TH})^{\alpha}} \tag{1.2}$$

where p_t is the switching probability, f_{CLK} is the clock frequency, C_L is the load capacitance, V_S is the voltage swing, V_{DD} is the power supply voltage, V_{TH} is the threshold voltage, and I_{LEAK} is the leakage current. γ, β, and α are constant, which depend on process and circuit parameters. The first term in P represents dynamic power consumption due to charging and discharging of the load capacitance, and the second term is due to the leakage current such as subthreshold current and reverse bias current. Thus, reducing the voltage swing V_S is effective for the reductions of both power consumption and propagation delay.

Up to date, many logic circuits that take advantage of the high-speed read-out capability of sense amplifiers have been reported [15]–[25]. It is, however, generally difficult

to generate an optimized sense-amplifier activation signal depending on operating conditions and process variations due to the structural complexity, except for regular circuit structures [16], [24]. As mentioned above, regular circuits have high design predictability and the issues of signal integrity such as cross-talk noise can be predicted and alleviated by taking advantage of the regularity. Thus, in regular circuits, the timing signal can be easily tuned and optimized with timing margins large enough to tolerate operating conditions and process variations.

1.4 Research Objectives and Thesis Organization

In this thesis, we focus on CMOS array logic architectures based on PLA structures, and propose new architectures and their design methodologies to address the drawback of the area-efficiency of regular circuits. Also, we consider low-swing signaling techniques by taking advantage of the regularity of the architectures in order to realize high-speed and low-power operations.

This thesis is organized as follows.

- Chapter 2 describes the concept of CMOS array logic architectures using dual-rail threshold logic circuits. We first address drawbacks of the conventional array logic architectures and the conventional threshold logic circuit, and propose a new array logic architecture and a new threshold logic circuit to overcome the drawbacks.

- Chapter 3 presents an area-efficient array logic architecture using 2-input logic cells. The architecture has 2-input logic cells in the structure and reduces the circuit area by logical methods. We show the effectiveness of the architecture by using a developed logic synthesis method.

- Chapter 4 presents a high-speed and area-efficient array logic architecture using divided and interdigitated column circuits. The architecture is based on the proposed threshold logic circuit described in Chapter 2 and reduces the circuit delay and area

by physical methods. We show the effectiveness of the architecture, especially in arithmetic circuits.

- Chapter 5 presents area-efficient array logic architectures using dual-rail multiple-threshold logic circuits. The architecture is based on multiple-threshold logic and reduces the circuit area by using the proposed logic synthesis methods for their architectures. As an application of the proposed multiple-threshold logic circuit, a new functional memory with the high-speed flexible data search capability is also presented.

- Finally, Chapter 6 gives conclusions of this thesis.

Chapter 2

Concept of CMOS Array Logic Architectures Using Dual-Rail Threshold Logic Circuits

2.1 Introduction

This chapter describes the concept of CMOS array logic architectures using dual-rail threshold logic circuits. First, the conventional array logic architectures and the conventional threshold logic circuit are introduced and their drawbacks are discussed. We then presents a new array logic architecture to overcome their drawbacks. The proposed architecture is based on a PLA structure using newly developed dual-rail threshold logic circuits and realizes arbitrary Boolean functions expressed in sum-of-products forms. The threshold logic circuit has a new circuit configuration using a charge-sharing scheme and a self-precharge scheme for high-speed and low-power operations.

2.2 Previous Work

2.2.1 Single-Rail Design

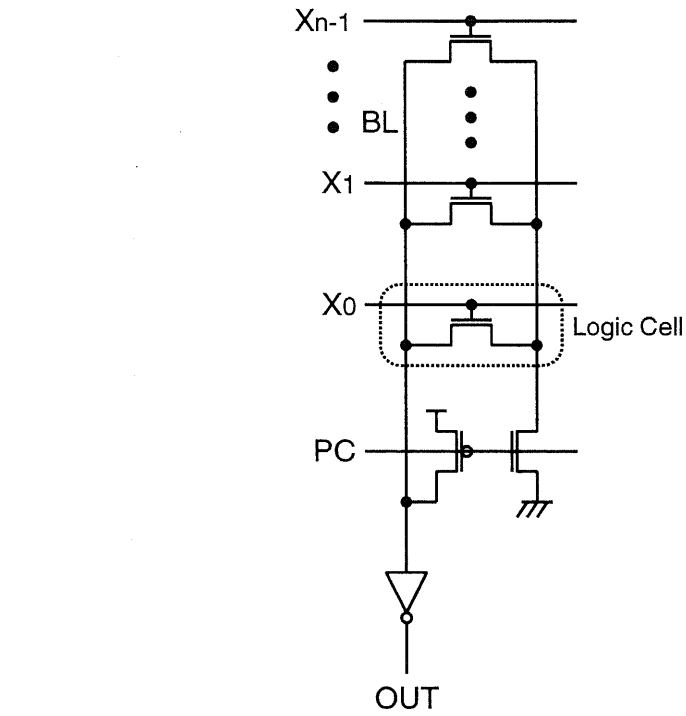
Figure 2.1(a) shows a column circuit of the conventional single-rail PLA [26]. It is composed of a dynamic circuit and has the capability of performing a logical OR of the input signals (x_0–x_{n-1}). A logical AND is also achieved by performing a logical OR of complement input signals with an output inverter gate. Thus, an AND-plane and an OR-plane for a PLA can be realized by arranging the circuits as shown in Figure 2.1(b). However, the circuit has some drawbacks, especially in deep sub-micron technologies.

One of the drawbacks is a voltage drop in a bit-line caused by leakage current. Leakage current of MOS devices is significantly increasing with the advances in process minimization. Thus, leakage current in a bit-line, as shown in Figure 2.2(a), is becoming a serious problem in an evaluation phase. Figure 2.3 shows the relationships between logic cell current and bit-line leakage current in 0.6-μm and 0.13-μm CMOS technologies. As can be seen from this figure, bit-line leakage current becomes serious with the advances in process minimization, and the difference between logic cell current and bit-line leakage current is decreasing. Large bit-line leakage current causes a voltage drop in a bit-line and an inversion of the output signal without activating the input signals.

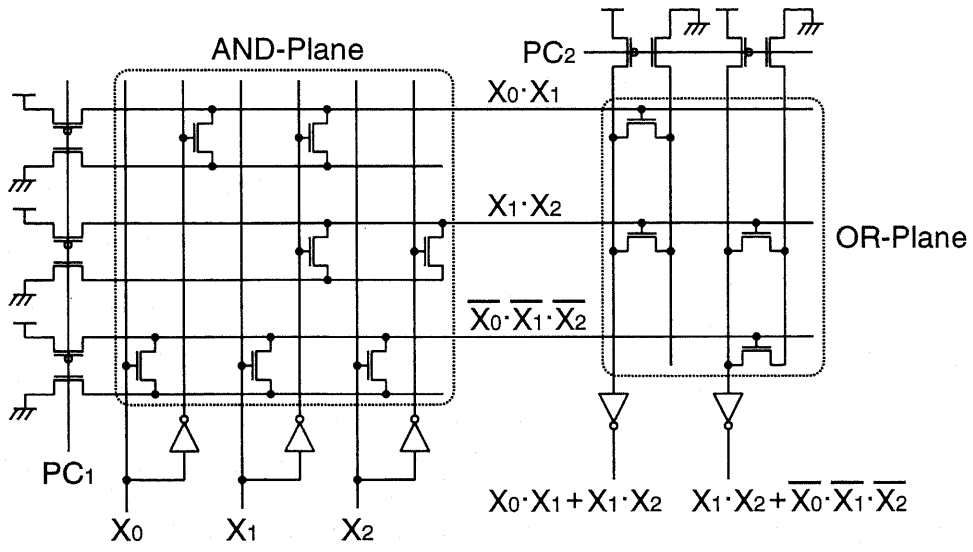
Second, single-rail structures are susceptible to noise such as cross-talk noise and power-supply noise, while dual-rail structures provide the common-mode noise immunity. This kind of noise is becoming serious with the advances in process minimization and the decrease in power supply levels. This also causes an unintended inversion of the output signal.

2.2.2 Dual-Rail Design

Figure 2.4 shows the conventional high-speed dual-rail dynamic circuit [24]. In this circuit, logical OR and NOR of the input signals (x_0–x_{n-1}) can be obtained from the output signals, OUT and $\overline{OUT}$, respectively. A logical AND is also achieved by performing



(a)



(b)

Figure 2.1 (a) Column circuit and (b) overall structure of the conventional single-rail PLA [26].

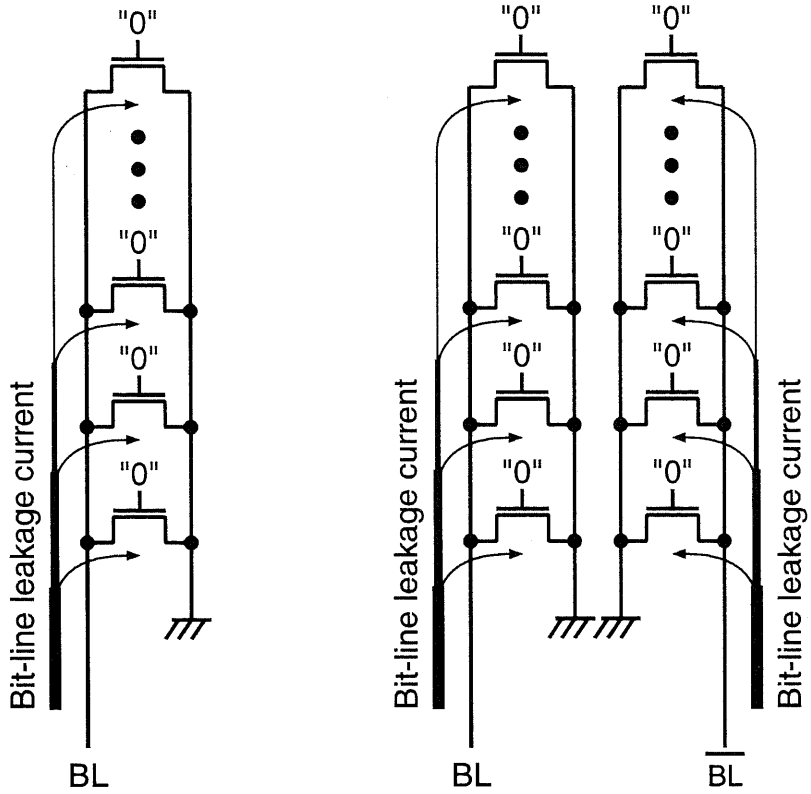
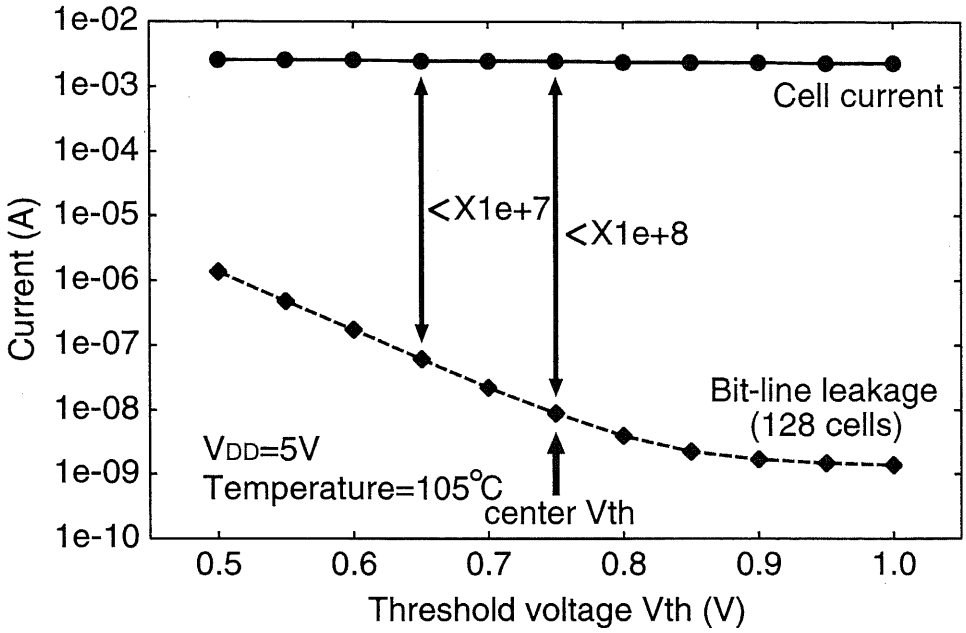


Figure 2.2 Bit-line leakage current in (a) a single-rail circuit and (b) a dual-rail circuit.

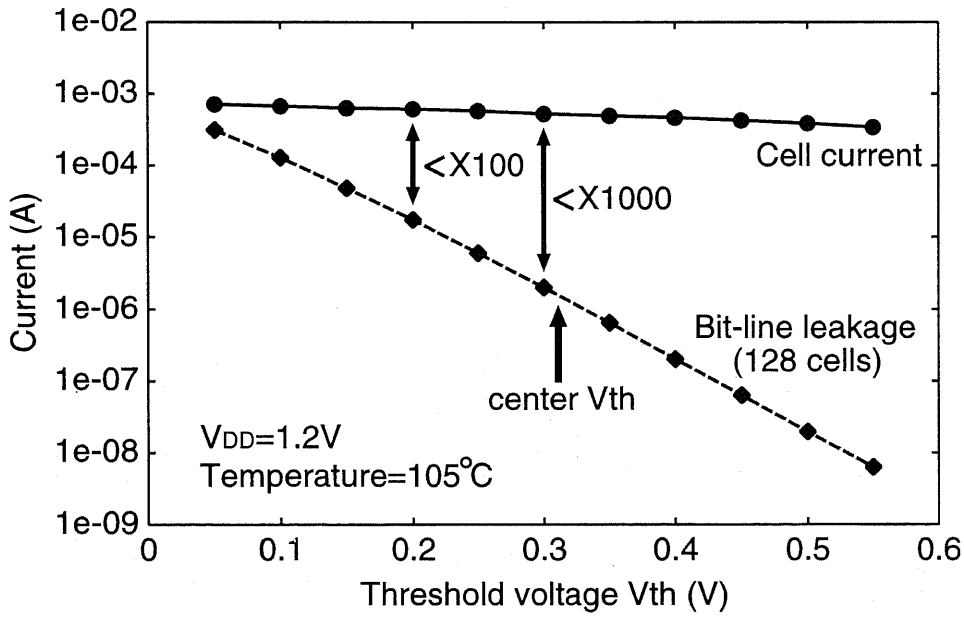
logical NOR of complement input signals. The circuit was proposed to perform a high-speed AND-ing operation for arithmetic circuits with a large number of input signals. The circuit consists of a stack of logic cells, a reference cell, a precharge and equalization circuit, and a sense amplifier. A logic cell has a pair of NMOS devices. One (M4) is used to pull down the bit-line (BL) depending on the input signals. The other (M_{dum}) is used to balance load capacitances of BL and $\overline{BL}$.

The circuit operates in two phases: phase 1 and phase 2. In phase 1, the *PC* signal and all the input signals, x_0–x_{n-1}, are low. Thus, the bit-lines are precharged high and equalized. When the *PC* signal becomes high and the input signals are activated, the circuit enters phase 2.

Figure 2.5 shows simplified transient relationships between BL and $\overline{BL}$ in phase 2.



(a)



(b)

Figure 2.3 Relationships between logic cell current and bit-line leakage current in (a) 0.6-μm and (b) 0.13-μm CMOS technologies.

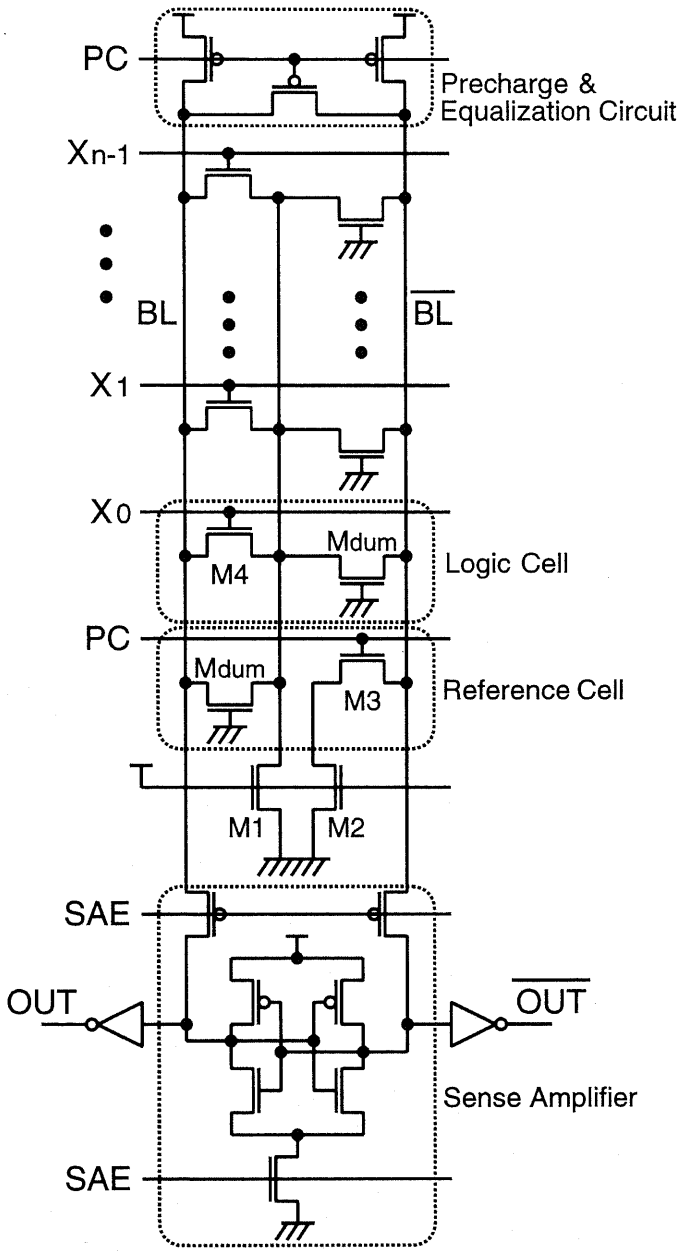


Figure 2.4 Conventional high-speed dual-rail dynamic circuit [24].

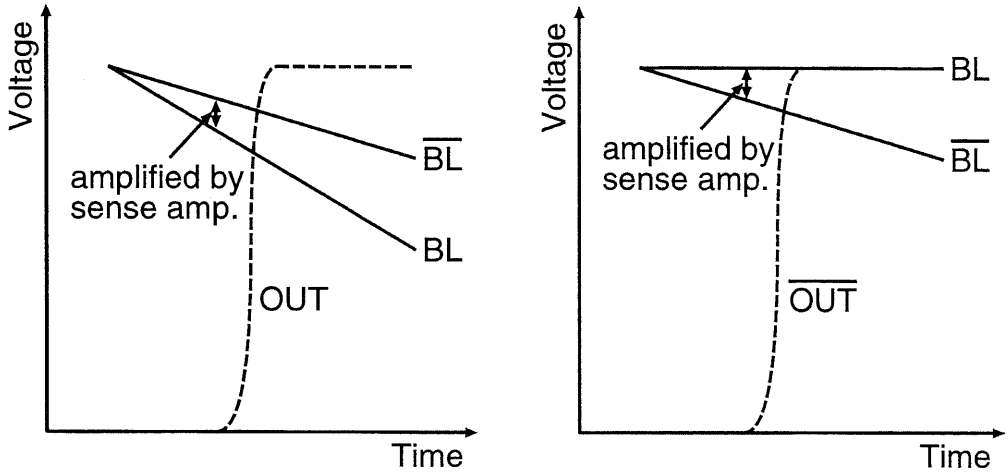


Figure 2.5 Simplified transient relationships between BL and $\overline{BL}$ (a) when at least one of the input signals is high and (b) all the input signals are low.

When at least one of the input signals is high, the voltage potential of BL becomes lower than that of $\overline{BL}$. This is because the device sizes of logic cells and a reference cell are designed so that when only one of the input signals is high, the discharging speed of BL is twice that of $\overline{BL}$. On the other hand, when all the input signals are low, BL stays high and $\overline{BL}$ is discharged by a reference cell. As a result, the voltage potential of $\overline{BL}$ becomes lower than that of BL. The *SAE* signal is activated when the developed voltage difference between the bit-lines becomes larger than the designed sense voltage, which takes the worst case of considerable noise margin and process variations into account. By activating the sense amplifier, one of the output signals, i.e., *OUT* or $\overline{OUT}$ becomes high depending on the developed voltage difference. Thus, this means that the voltage potential of $\overline{BL}$ is used as a *threshold value* for determining the output signal, and the circuit forms a *dual-rail threshold logic circuit* based on a reference voltage of $\overline{BL}$.

In general, dual-rail structures require larger circuit area compared to single-rail structures. On the other hand, dual-rail structures have some advantages over single-rail structures. First, they provide the common-mode noise immunity and the leakage-current-tolerant design because both of the bit-lines have the same characteristics and the fluctu-

ation of bit-line potentials do not affect circuit operations. Leakage current in a dual-rail bit-line shown in Figure 2.2(b) is negligible in terms of circuit operation. These features are becoming increasingly important to realize noise-immune systems in deep sub-micron technologies, while chip area cost is becoming small due to the advances in integration [1]. Moreover, in dual-rail structures, a read-out scheme using sense amplifiers can be easily utilized, and high-speed and low-power operations can be achieved by reducing voltage swings of bit-lines. A sense amplifier, which is commonly used in memory-type circuits, detects the small voltage difference between bit-lines and amplifies it to a rail-to-rail signal. A conventional receiver, such as an inverter shown in Figure 2.1(a), generally needs to accept a rail-to-rail signal. This results in a longer latency for read-out for the conventional circuits than for the circuits using sense amplifiers if a bit-line potential changes slowly. Up to date, many logic circuits that take advantage of the high-speed read-out capability of sense amplifiers have been reported [15]–[25]. However, it is generally difficult to generate an optimized sense-amplifier activation signal depending on operating conditions and process variations due to the structural complexity, except for regular circuit structures [16], [24]. In the array structure of the circuit in Figure 2.4, the activation signal can be easily generated due to the regularity.

The circuit in Figure 2.4, however, has some drawbacks. One of the drawbacks is caused by the difference of the bit-line discharging speed between a lightly loaded bit-line and a heavily loaded bit-line. As mentioned above, the circuit was proposed to perform a high-speed AND-ing operation in the configuration shown in Figure 2.6. A sense-amplifier activation signal is generated depending on a heavily loaded bit-line to ensure the designed sense voltage. However, in this configuration, i.e., the combination of column circuits which have a lightly loaded bit-line and a heavily loaded bit-line, the voltage potential of a lightly loaded bit-line reaches the ground level when the designed sense voltage is generated in a heavily loaded bit-line. This is because of the difference of load capacitances between the bit-lines. Thus, this may introduce a sensing error in the circuit using a lightly loaded bit-line. Moreover, the circuit configuration in Figure 2.6 can not realize arbitrary Boolean functions expressed in sum-of-products forms.

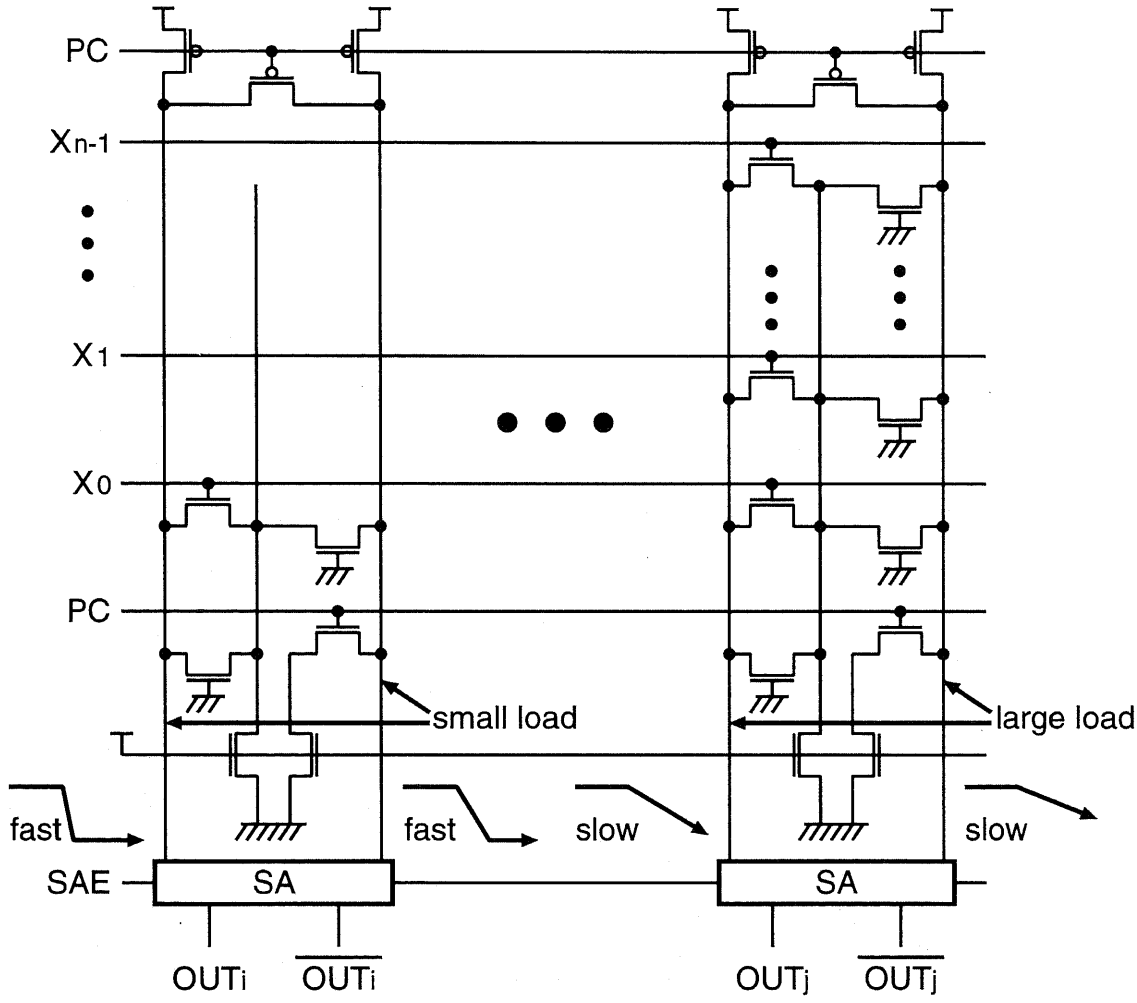


Figure 2.6 Difference of the bit-line discharging speed between a lightly loaded bit-line and a heavily loaded bit-line.

In the next section, we propose a new array logic architecture to overcome these drawbacks.

2.3 Proposed Circuits

2.3.1 Column Circuit

Figure 2.7 shows a column circuit of the proposed PLA. Logical OR and NOR of the input signals (x_0–x_{n-1}) can be obtained from the output signals, OUT and $\overline{OUT}$, respectively. A logical AND is also obtained by performing logical NOR of complement input signals. Thus, an AND-plane and an OR-plane for a PLA can be realized by arranging the column circuits. The circuit is a dual-rail configuration and consists of a stack of logic cells, a reference cell, a virtual ground (VG) controller, a precharge and equalization circuit, and a sense amplifier. A logic cell has a pair of NMOS devices. One (M8) is used to pull down the bit-line (BL) depending on the input signals. The other (M_{dum}) is used to balance load capacitances and leakage current of BL and $\overline{BL}$. The column pitch depends on the width of a logic cell. By using a sense amplifier, the output signals are activated by sensing the differential voltage between the bit-lines. The VG controller is provided to reduce voltage swings of the bit-lines and to expand the voltage difference between the bit-lines.

Figure 2.8 shows a timing diagram of control, input, and output signals, and bit-line potentials in a column circuit. The circuit operates in two phases: phase 1 and phase 2. In phase 1, the PC signal and all the input signals, x_0–x_{n-1}, are low. Thus, the bit-lines are precharged high and equalized. At the same time, VG and $\overline{VG}$ are discharged low. When the PC signal becomes high and the input signals are activated, the circuit enters phase 2, where the differential voltage between the bit-lines is generated depending on the input signals.

Figure 2.9 shows simplified transient relationships between BL and $\overline{BL}$ in phase 2. When at least one of the input signals is high, the voltage potential of BL becomes lower than that of $\overline{BL}$. This is because the device sizes of logic cells and a reference cell are

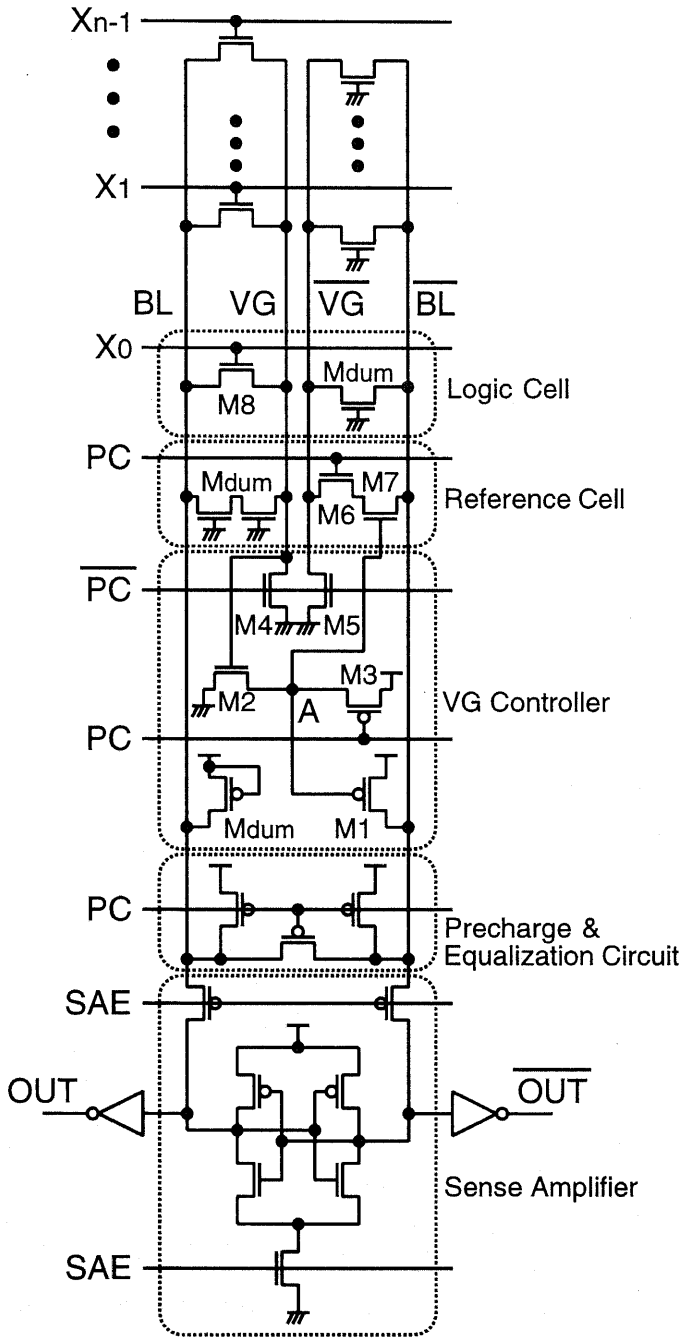


Figure 2.7 Column circuit of the proposed PLA.

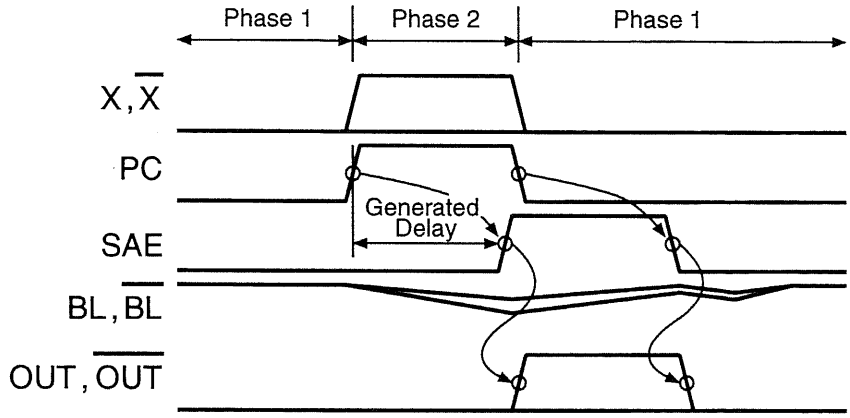


Figure 2.8 Timing diagram of control, input, and output signals, and bit-line potentials in a column circuit.

designed so that when only one of the input signals is high, the discharging speed of BL is twice that of $\overline{BL}$. As a result, the voltage potentials of VG and $\overline{VG}$ rise due to charge sharing with the bit-lines, and the node A is discharged by M2. This causes M1 turned on and M7 turned off. Thus, $\overline{BL}$ starts to be precharged by M1. This self-precharge scheme helps to develop the voltage difference between the bit-lines and avoids reaching ground level of both the bit-lines. Moreover, BL and $\overline{BL}$ are not completely discharged even if all the input signals are high, thus enabling the bit-lines to be precharged faster and resulting in low-power operations. On the other hand, when all the input signals are low, BL stays high and $\overline{BL}$ is discharged by a reference cell. As a result, the voltage potential of $\overline{BL}$ becomes lower than that of BL. The *SAE* signal is activated when the developed voltage difference between the bit-lines becomes larger than the designed sense voltage, which takes the worst case of considerable noise margin and process variations into account. By activating the sense amplifier, one of the output signals, i.e., *OUT* or $\overline{OUT}$ becomes high depending on the developed voltage difference. After the activation of the sense amplifier, the *PC* signal becomes low and the circuit starts to precharge the bit-lines.

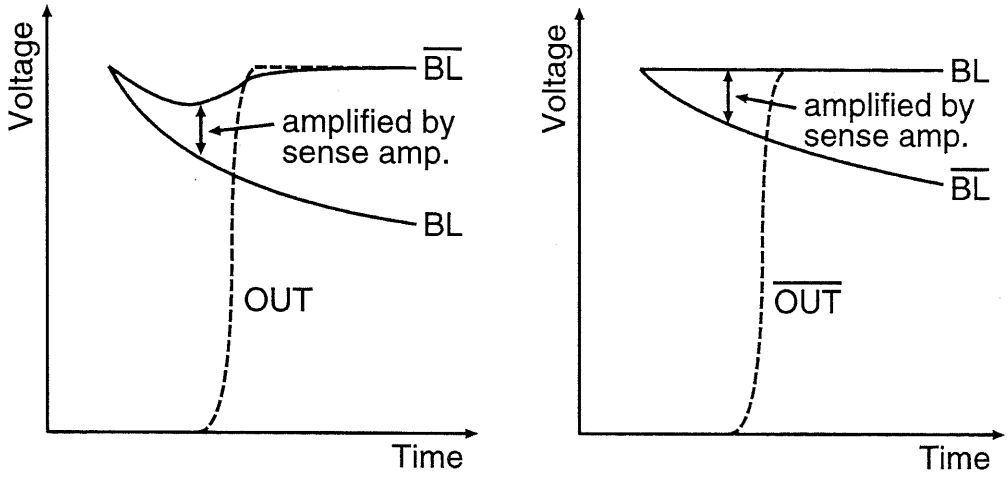


Figure 2.9 Simplified transient relationships between BL and $\overline{BL}$ (a) when at least one of the input signals is high and (b) all the input signals are low.

2.3.2 PLA Configuration

The proposed PLA configuration using column circuits is shown in Figure 2.10. An array of column circuits is used as an AND-plane and an OR-plane. Control signals of the AND-plane are generated from the *CLK* signal with a delay line of a chain of sized inverters. On the other hand, control signals of the OR-plane are generated from a dummy column and a delay line of a chain of sized inverters. The dummy column is designed so that its output signal arrives last in the AND-plane. For this purpose, it has the largest number of logic cells in the AND-plane, and gate terminals of the logic cells are connected to ground. Its output signal, i.e., the *PC* signal of the OR-plane is generated every cycle and follows operating conditions, such as temperature and supply voltage variations, as well as process variations in sync with column circuits in the AND-plane.

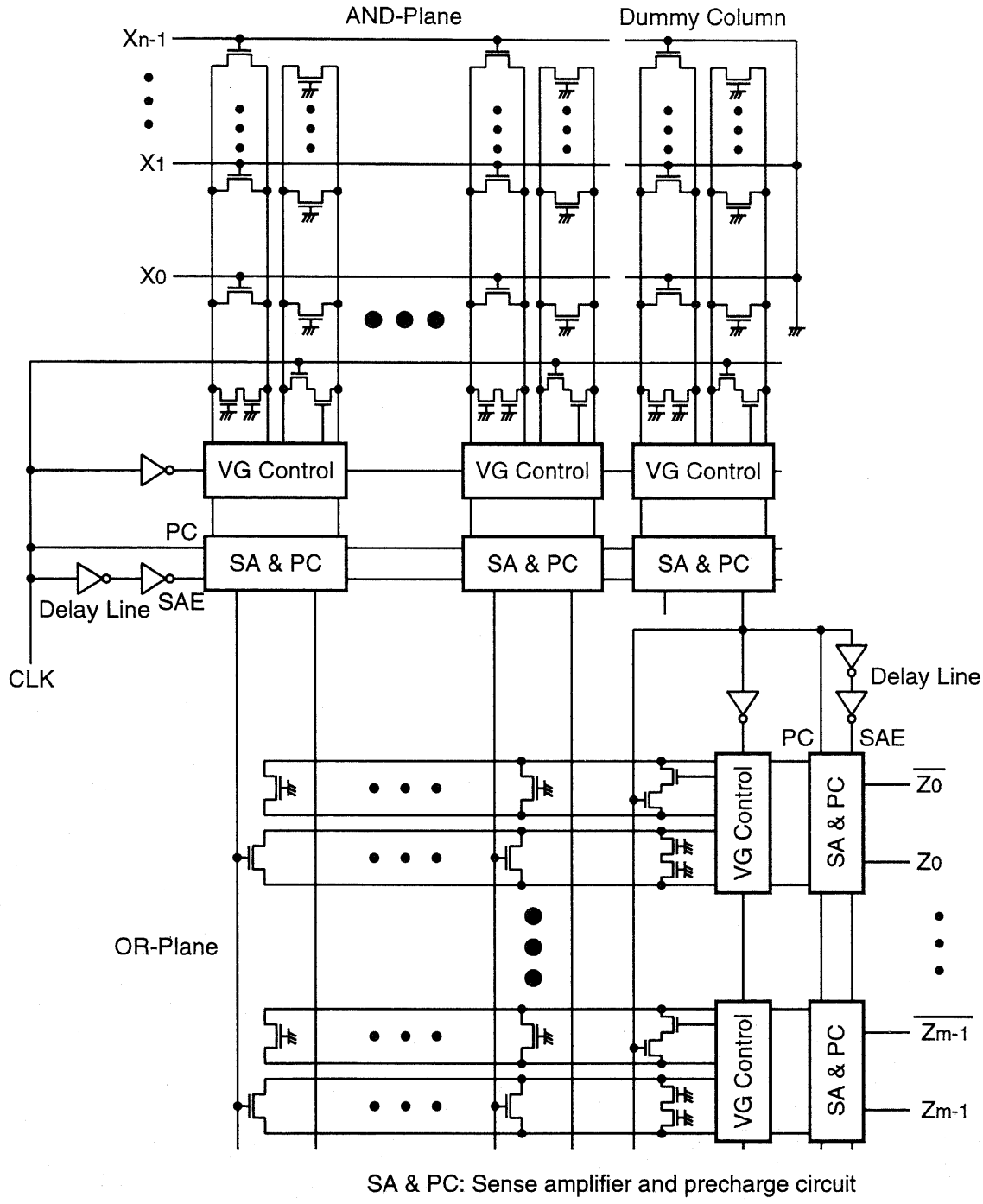


Figure 2.10 Proposed PLA configuration using column circuits.

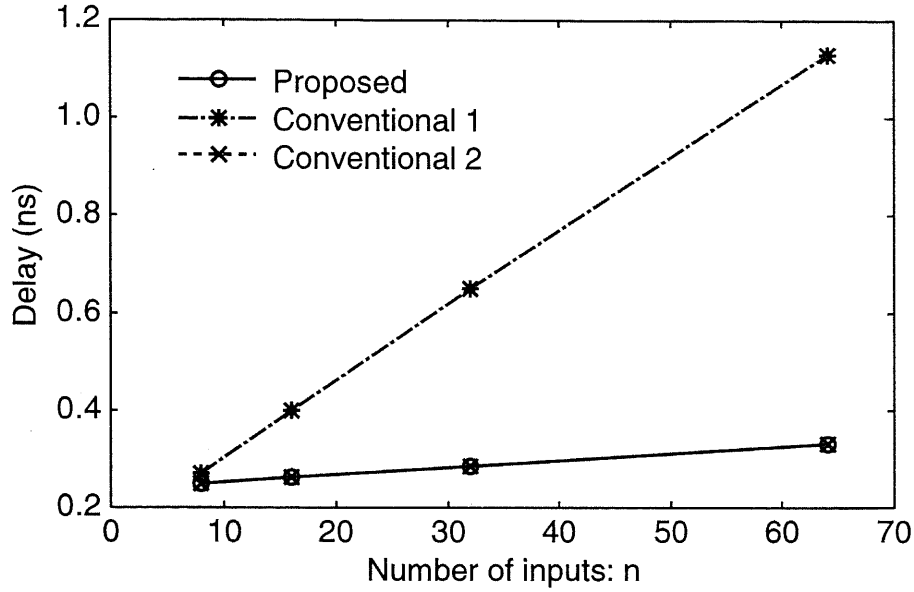


Figure 2.11 Comparison of propagation delay of the conventional and proposed column circuits. The conventional 1 and 2 are the circuits in Figure 2.1(a) and Figure 2.4, respectively.

2.4 Simulation Results

In order to evaluate the performance, power consumption, and area of a column circuit, the proposed and the conventional circuits shown in Figure 2.7, Figure 2.1(a), and Figure 2.4 were designed using a 0.6-μm CMOS technology with a supply voltage of 5 V. The performance and power consumption were obtained from post-layout simulations using HSPICE. Transistors used in logic cells, output inverters, and sense amplifiers have the same sizes among the different circuits, and a sense voltage of 200 mV was used.

Figure 2.11 shows a comparison of propagation delay for various numbers of input signals. The delay of each circuit increases as the number of input signals increases. The characteristics of the proposed circuit are almost the same as those of the conventional circuit in Figure 2.4. The conventional circuit in Figure 2.1(a) with a small number of input signals becomes faster than the proposed circuit as can be predicted from the figure. This is because the differential voltage between the bit-lines is developed quickly due to

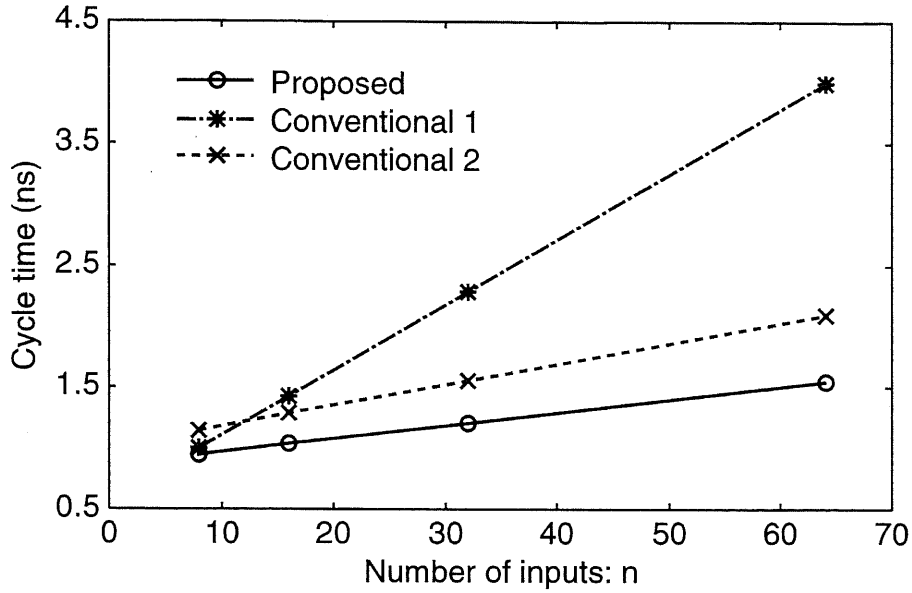


Figure 2.12 Comparison of cycle time of the conventional and proposed column circuits. The conventional 1 and 2 are the circuits in Figure 2.1(a) and Figure 2.4, respectively.

small load capacitances of the bit-lines. Thus, it is difficult to take advantage of the high-speed read-out capability of a sense amplifier in such small circuits. On the other hand, the proposed circuit with a large number of input signals is faster than the conventional circuit in Figure 2.1(a) because the differential voltage between the bit-lines is developed slowly due to large load capacitances of the bit-lines and is efficiently amplified to a rail-to-rail level by a sense amplifier. The proposed circuit with a 64-bit input signal is 3.40 times faster than the conventional circuit in Figure 2.1(a).

Figure 2.12 shows a comparison of cycle time for various numbers of input signals. The cycle time of each circuit increases as the number of input signals increases like the characteristics of propagation delay. The proposed circuit shows the best performance among all the circuits due to reduced voltage swings of the bit-lines using a charge-sharing scheme and a self-precharge scheme, which enable the bit-lines to be precharged faster. The proposed circuit with a 64-bit input signal is 2.58 and 1.35 times faster than the conventional circuits in Figure 2.1(a) and Figure 2.4, respectively.

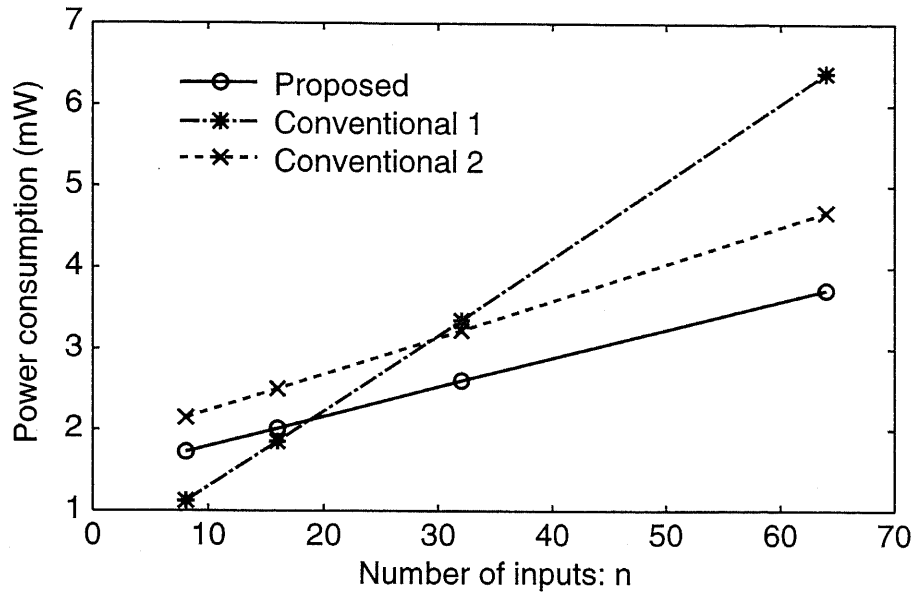


Figure 2.13 Comparison of power consumption of the conventional and proposed column circuits. The conventional 1 and 2 are the circuits in Figure 2.1(a) and Figure 2.4, respectively.

Figure 2.13 shows a comparison of power consumption for various numbers of input signals at the operating frequency of 250 MHz. The power consumption of each circuit increases as the number of input signals increases. The proposed circuit with a small number of input signals consumes larger power than the conventional circuit in Figure 2.1(a). This is because there is additional power due to the use of a sense amplifier. On the other hand, the proposed circuit with a large number of input signals consumes smaller power than the conventional circuit in Figure 2.1(a). This is because the power consumption of a sense amplifier becomes negligible compared to that of a large number of logic cells, and the power consumption of logic cells is efficiently reduced by small voltage swings of the bit-lines. The proposed circuit with a 64-bit input signal achieves power reductions by 41.7% and 20.3% compared to the conventional circuits in Figure 2.1(a) and Figure 2.4, respectively.

In our approach, there are additional transistors over the conventional circuits in Figure 2.1(a) and Figure 2.4 owing to the use of a dual-rail configuration, a VG controller,

Table 2.1 Comparison of physical dimension of a column circuit.

Circuit	Physical dimension (μm × μm)			
(0.6-μm CMOS technology)	8-bit	16-bit	32-bit	64-bit
Single-rail PLA [26]	110 × 7	182 × 7	324 × 7	610 × 7
Dynamic circuit [24]	138 × 19	210 × 19	353 × 19	638 × 19
This work	149 × 19	221 × 19	363 × 19	649 × 19

and a sense amplifier. Thus, the proposed circuit requires a larger chip area than the conventional circuits. Table 2.1 shows a comparison of physical dimension of each circuit. The area of the proposed circuit is larger than those of the conventional circuits. The area penalty is, however, reduced as the number of input signals increases because a VG controller and a sense amplifier consume almost the same area even in larger circuits. Thus, the proposed circuit becomes more effective in high data bandwidth systems.

2.5 Application Design

As an application of the proposed PLA, a 32-bit binary comparator was designed. The circuit receives two 32-bit data and compares the magnitude and equality of the data. A logic function of an equality and magnitude comparator of two 32-bit data, A and B, is expressed as

$$P_i = A_i \cdot B_i + \overline{A_i} \cdot \overline{B_i} = \overline{A_i \oplus B_i} \tag{2.1}$$

$$Z_{A=B} = P_{31} \cdot P_{30} \cdot \cdots \cdot P_2 \cdot P_1 \cdot P_0 \tag{2.2}$$

$$\begin{aligned} Z_{A>B} = {} & A_{31} \cdot \overline{B_{31}} \\ & + P_{31} \cdot A_{30} \cdot \overline{B_{30}} \\ & + P_{31} \cdot P_{30} \cdot A_{29} \cdot \overline{B_{29}} \\ & + \cdots \\ & + P_{31} \cdot P_{30} \cdot \cdots \cdot P_3 \cdot P_2 \cdot A_1 \cdot \overline{B_1} \\ & + P_{31} \cdot P_{30} \cdot \cdots \cdot P_2 \cdot P_1 \cdot A_0 \cdot \overline{B_0} \end{aligned} \tag{2.3}$$

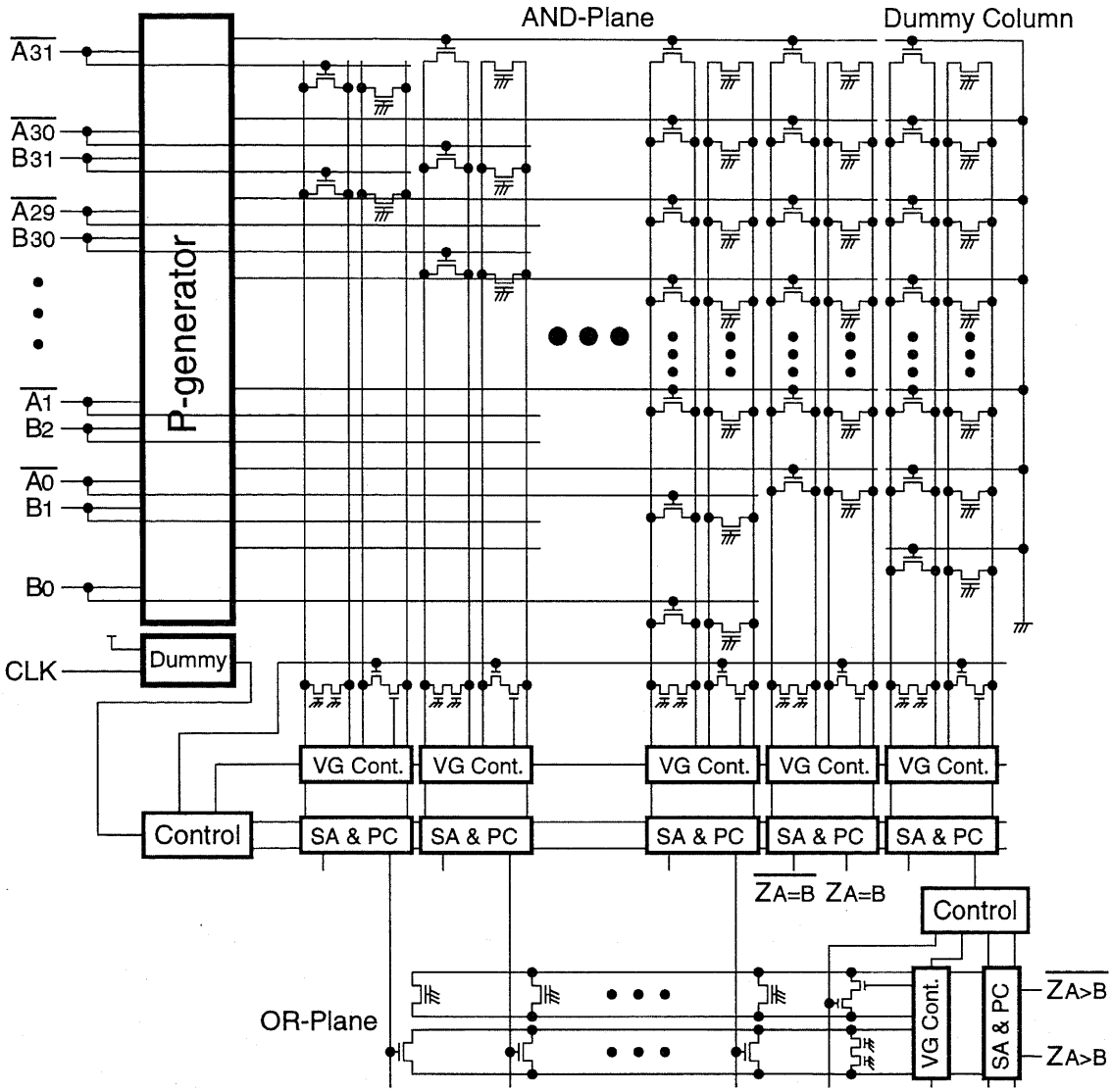


Figure 2.14 Schematic diagram of the proposed 32-bit binary comparator.

Table 2.2 Comparison of a 32-bit binary comparator in different implementation styles.

Circuit (0.6-μm, 5-V V_{DD})	Delay (ns)		Cycle time (ns)	Power (mW) @ 250 MHz	PD product* (pJ)	Area (μm^2)
	$Z_{A=B}$	$Z_{A>B}$				
Domino logic	1.32	1.40	1.50	78	213	164,710
Single-rail PLA [26]	0.69	1.18	2.20	117	219	183,296
Dynamic circuit [24]	0.41	0.89	1.55	130	169	419,634
This work	0.41	0.89	1.20	110	143	430,920

*PD product = Power × Delay ($Z_{A=B} + Z_{A>B}$)

where $Z_{A=B}$ becomes high when $A = B$, while $Z_{A>B}$ becomes high when $A > B$. The schematic diagram of this implementation is shown in Figure 2.14. The logic function $Z_{A=B}$, which is expressed as a product form, is implemented in an AND-plane using the propagate signal P. Similarly, the logic function $Z_{A>B}$, which is expressed as a sum-of-products form, is implemented in an AND-plane and an OR-plane using the propagation signal P. The propagation signal P is generated from 2-input XNOR gates located in the P-generator block and significantly reduces the overall area of the circuit compared to the direct implementation without the P signal. The *CLK* signal is delayed with a dummy block composed of a 2-input XNOR gate in order to be synchronized with the P signal to the AND-plane.

2.6 Simulation and Measurement Results

The proposed comparator was designed and fabricated using a 0.6-μm, 3-metal CMOS technology with a supply voltage of 5 V. A simulated comparison between different implementation styles of the comparator and simulated waveforms of the proposed circuit are shown in Table 2.2 and Figure 2.15, respectively. The performances were obtained from post-layout simulations using HSPICE. In the domino logic circuit, carry look-ahead gates [26] are utilized for high-speed operations. The conventional high-speed dynamic circuit was arranged in the same manner as the proposed PLA. The output signals $Z_{A=B}$

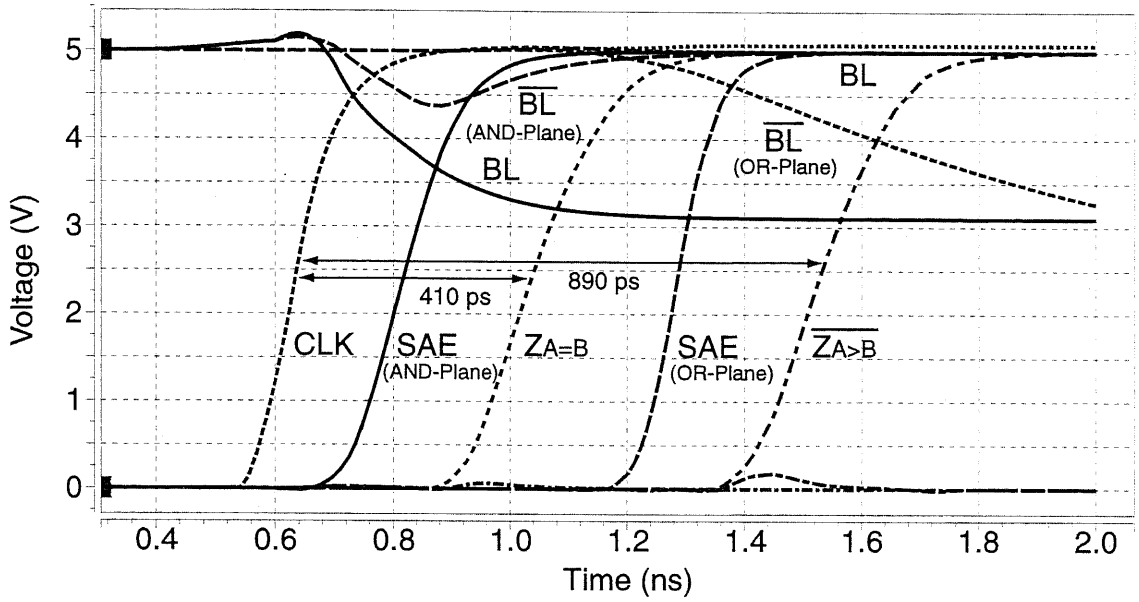


Figure 2.15 Simulated waveforms of the proposed 32-bit binary comparator.

and $Z_{A>B}$ are, respectively, 3.21 and 1.57 times faster than those of the domino logic circuit, and 1.68 and 1.32 times faster than those of the conventional single-rail PLA. Also, the proposed circuit achieved reductions of cycle time by 20.0%, 45.5%, and 22.6%, and reductions of power-delay product by 32.9%, 34.7%, and 15.4% compared to the domino logic circuit, the conventional single-rail PLA, and the conventional high-speed dynamic circuit, respectively. The area penalty is 2.62, 2.35, and 1.03 times compared to the domino logic circuit, the conventional single-rail PLA, and the conventional high-speed dynamic circuit, respectively.

A designed chip microphotograph is shown in Figure 2.16. The operations of the circuit were successfully verified by a functional test using a logic tester at frequencies of up to 100 MHz, which is the limitation of the test equipment, at room temperature with a supply voltage of 5 V. Figure 2.17 shows measured waveforms along the critical path using an electron-beam tester at room temperature. The timebase resolution of the test equipment is 10 ps. The measured delays from the *CLK* signal to $Z_{A=B}$ and $Z_{A>B}$ are 490 ps and 1,030 ps, respectively. The features of the test chip are summarized in Table 2.3.

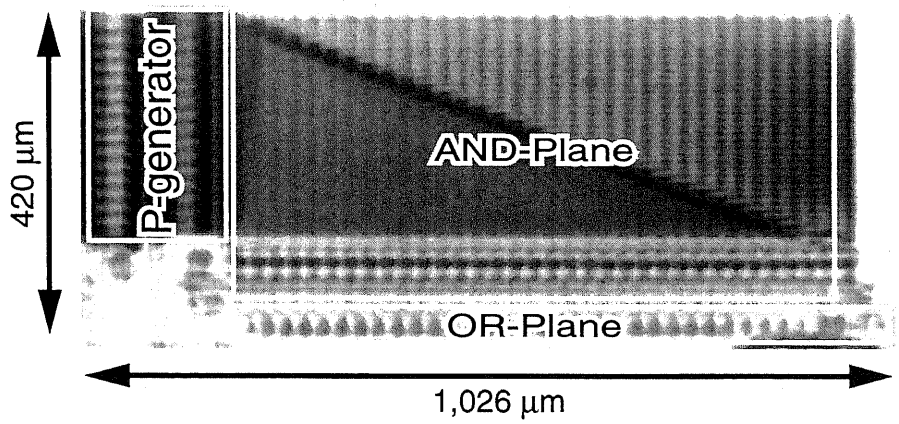


Figure 2.16 Chip microphotograph of the proposed 32-bit binary comparator.

Table 2.3 Features of the test chip.

Process technology	0.6-μm CMOS, 3-metal
Supply voltage	5 V
Transistor count	2.4 k
Physical dimension	420 × 1,026 μm^2
Simulated cycle time	1.20 ns (830 MHz)
Simulated power consumption	353 mW (@ 830 MHz)
Simulated delay	$Z_{A=B}$: 410 ps, $Z_{A>B}$: 890 ps
Measured delay	$Z_{A=B}$: 490 ps, $Z_{A>B}$: 1,030 ps

2.7 Summary

In this chapter, the concept of CMOS array logic architectures using dual-rail threshold logic circuits was described. We first addressed some drawbacks of the conventional array logic architectures and the conventional threshold logic circuit, and proposed a new array logic architecture to overcome these drawbacks. The proposed architecture is based on a PLA using newly developed dual-rail threshold logic circuits and realizes arbitrary Boolean functions expressed in sum-of-products forms. The threshold logic circuit has a new circuit configuration using a charge-sharing scheme and a self-precharge scheme for high-speed and low-power operations. As an application of the proposed array logic

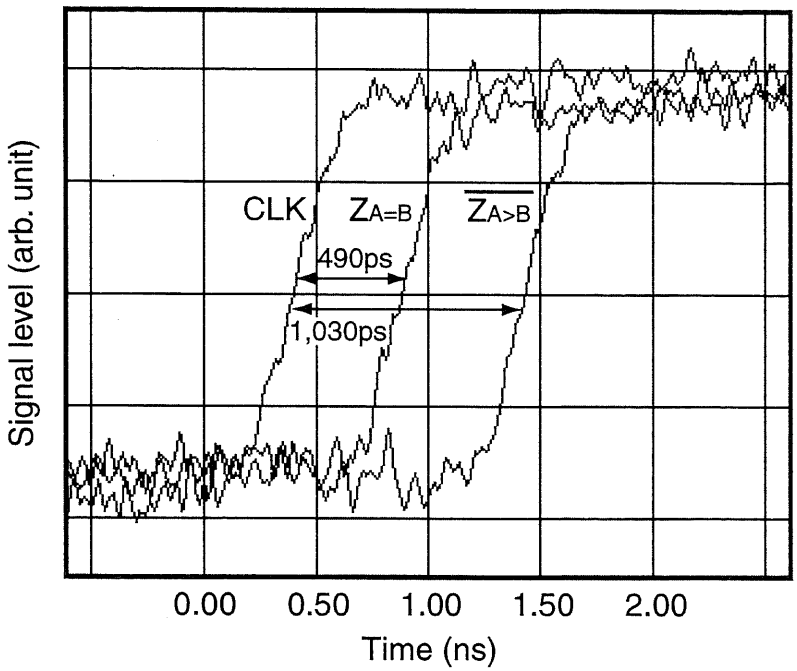


Figure 2.17 Measured waveforms by an electron-beam tester.

architecture, a 32-bit binary comparator was designed and fabricated using a 0.6-μm, 3-metal CMOS technology. The output signals $Z_{A=B}$ and $Z_{A>B}$ are, respectively, 3.21 and 1.57 times faster than those of the domino logic circuit, and 1.68 and 1.32 times faster than those of the conventional single-rail PLA. Also, the proposed circuit achieved reductions of cycle time by 20.0%, 45.5%, and 22.6%, and reductions of power-delay product by 32.9%, 34.7%, and 15.4% compared to the domino logic circuit, the conventional single-rail PLA, and the conventional high-speed dynamic circuit, respectively. The operations of the circuit were successfully verified by a functional test using a logic tester and an electron-beam tester at frequencies of up to 100 MHz at room temperature with a supply voltage of 5 V.

Chapter 3

Logical Compaction by 2-Input Logic Cells

3.1 Introduction

In this chapter, we propose an area-efficient dual-rail array logic architecture, a *logic cell-embedded PLA (LCPLA)*, which is based on the dual-rail PLA described in Chapter 2 and has 2-input logic cells in the structure. In general, although the structural regularity of PLAs offers the design simplicity, it requires a large chip area due to its low area-efficiency compared to random logic implementations. To overcome this drawback, a network structure of small PLAs which implements Boolean functions efficiently has been proposed [5]. This structure, however, degrades the circuit performance due to an increase in logic levels. Besides, although each PLA is surely regular, the global regularity is sacrificed because of placement and routing. Actually, it may be even worse than that in standard-cell-based designs that have a row structure, and thus it is difficult to take advantage of the regularity of PLAs. In our approach, 2-input logic cells which realize arbitrary 2-input Boolean functions can be embedded in a dual-rail PLA. The logic cells can be designed by connecting some local wires and do not require additional transistors over logic cells of the proposed PLA in Chapter 2. By using the logic cells, some classes

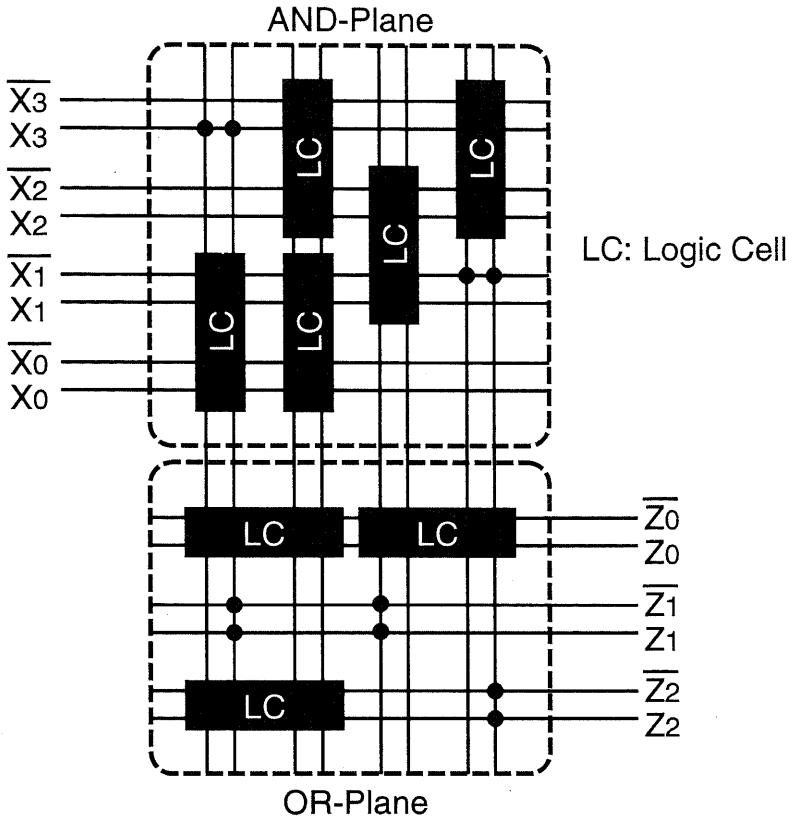


Figure 3.1 LCPLA structure.

of logic functions can be implemented efficiently, so that high-speed and low-power operations are also achieved due to the reduction of circuit elements.

3.2 Proposed Architecture

Figure 3.1 shows a structure of LCPLA. The structure has a dual-rail configuration. The x_i and $\overline{x}_i$ signals are the primary input and its negation, respectively. The z_j and $\overline{z}_j$ signals are the outputs of the PLA. An LCPLA contains an AND-plane and an OR-plane, and is capable of implementing arbitrary Boolean functions expressed in sum-of-products forms by connecting switching transistors, which are denoted by dots in the figure, to the input and output wires. In addition, an LCPLA utilizes 2-input logic cells (LCs) in both planes. The 2-input logic cells realize arbitrary 2-input Boolean functions of two adjacent signals

and are embedded in the PLA to reduce chip area. These logic cells can be easily designed from the original logic cell by connecting some local wires. Logical AND and OR of the outputs of the logic cells can be obtained from the outputs of AND-plane and OR-plane, respectively. Thus, an LCPLA can realize an LC-AND-LC-OR structure, i.e., 4-level logic implementation.

3.3 Proposed Circuits

3.3.1 Column Circuit

Figure 3.2 shows a column circuit of LCPLA. The circuit has a dual-rail configuration and consists of a stack of the basic logic cells shown in Figure 3.3, a reference cell, a virtual ground (VG) controller, a precharge and equalization circuit, and a sense amplifier. Logical OR and NOR of the outputs of the basic logic cells can be obtained from the output signals, OUT and $\overline{OUT}$, respectively. A logical AND is also obtained by performing a logical NOR of complement input signals. Thus, an AND-plane and an OR-plane for a PLA can be realized by arranging the column circuits. By using a sense amplifier, the output signals are activated by sensing the differential voltage between the bit-lines, BL and $\overline{\text{BL}}$. VG is provided to reduce the voltage swings of the bit-lines.

There are 11 types of basic logic cells, which can be embedded in the column circuit, as shown in Figure 3.3. Each cell has a pair of NMOS devices, except for type 0. One is used to pull down the bit-line (BL) depending on the input signals (x_0–x_{n-1}). The other is to balance the load capacitances and leakage current of BL and $\overline{\text{BL}}$. This symmetrical structure results in a robust operation of the circuit, especially with respect to common-mode noise. The basic logic cells of type 3–10 are composed of pass-transistors and are realized by connecting some local wires. By using the basic logic cells, the logic function of a column circuit is expressed as

$$OUT = \sum_{i=0}^{n-1} y_i \tag{3.1}$$

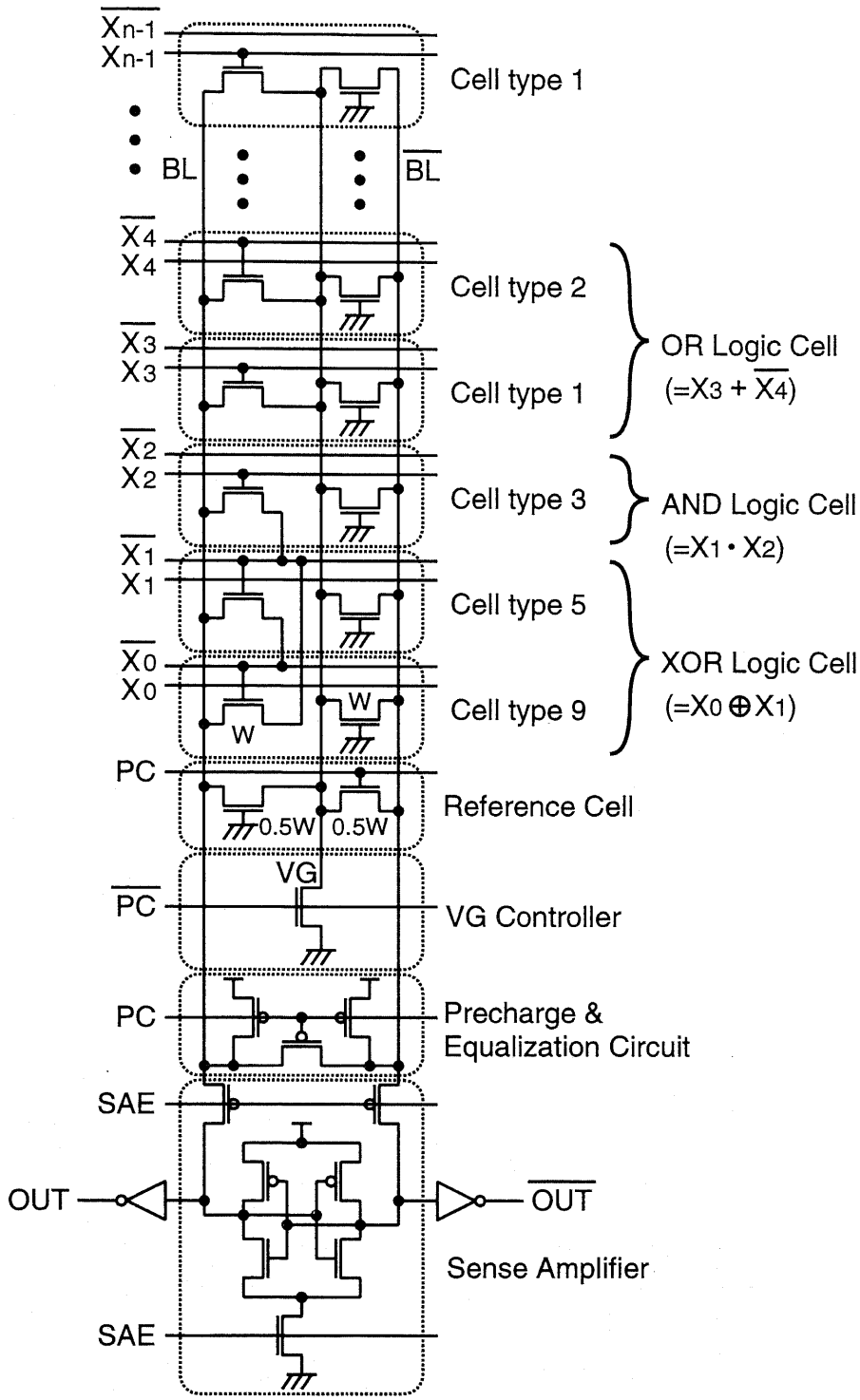


Figure 3.2 Column circuit of LCPLA.

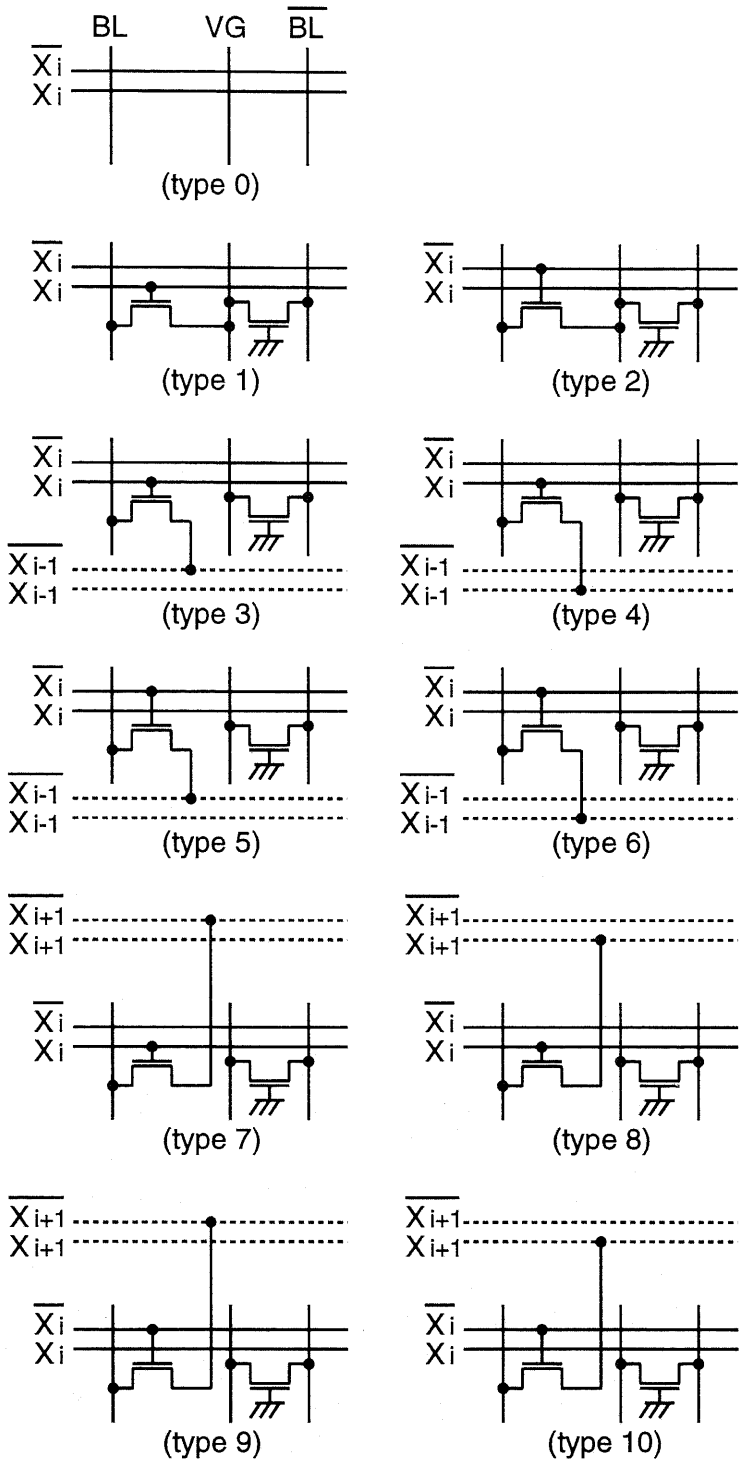


Figure 3.3 Basic logic cells.

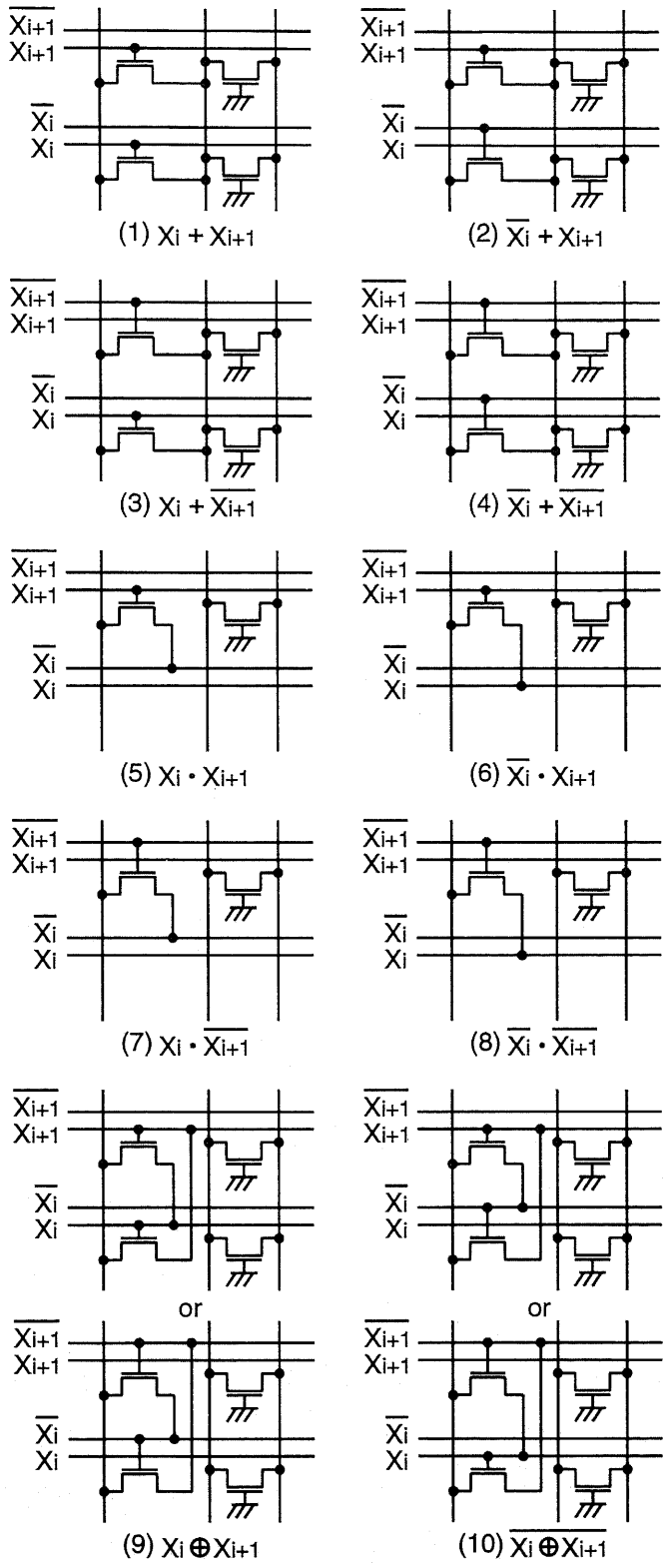


Figure 3.4 2-input logic cells (LCs) using the basic logic cells in Figure 3.3.

$$y_i = \begin{cases} 0 & \cdots \text{ type 0} \\ x_i & \cdots \text{ type 1} \\ \overline{x}_i & \cdots \text{ type 2} \\ x_i \cdot x_{i-1} & \cdots \text{ type 3 } (i \neq 0) \\ x_i \cdot \overline{x}_{i-1} & \cdots \text{ type 4 } (i \neq 0) \\ \overline{x}_i \cdot x_{i-1} & \cdots \text{ type 5 } (i \neq 0) \\ \overline{x}_i \cdot \overline{x}_{i-1} & \cdots \text{ type 6 } (i \neq 0) \\ x_i \cdot x_{i+1} & \cdots \text{ type 7 } (i \neq n-1) \\ x_i \cdot \overline{x}_{i+1} & \cdots \text{ type 8 } (i \neq n-1) \\ \overline{x}_i \cdot x_{i+1} & \cdots \text{ type 9 } (i \neq n-1) \\ \overline{x}_i \cdot \overline{x}_{i+1} & \cdots \text{ type 10 } (i \neq n-1) \end{cases} \tag{3.2}$$

where y_i is a logic function realized by a basic logic cell and corresponds to the type of the basic logic cell.

2-input logic cells using the basic logic cells in Figure 3.3 are shown in Figure 3.4. By combining the basic logic cells, arbitrary 2-input logic functions of two adjacent signals can be realized. The 2-input logic cells of (1)–(4), which realize a logical OR, are formed by the basic logic cells of type 1–2. On the other hand, the 2-input logic cells of (5)–(8), which realize a logical AND, are formed by the basic logic cells of type 3–6. The 2-input logic cells of (9)–(10), which realize logical XOR and XNOR, have two configurations, respectively, and are formed by the basic logic cells of type 3–10. These cells do not degrade the circuit area because pass-transistors used in the structure give no additional area. Thus, the resulting circuits cannot be worse in terms of area over the proposed PLA in Chapter 2.

Figure 3.5 shows a timing diagram of control signals, input and output signals, and bit-line potentials in a column circuit. The circuit operates in two phases: phase 1 and phase 2. In phase 1, the *PC* signal and all the primary inputs, x_0–x_{n-1} and $\overline{x}_0$–$\overline{x}_{n-1}$, are low. Thus, the bit-lines are precharged high and equalized. At the same time, the VG node is discharged low. When the *PC* signal becomes high and the primary inputs are activated, the circuit enters phase 2.

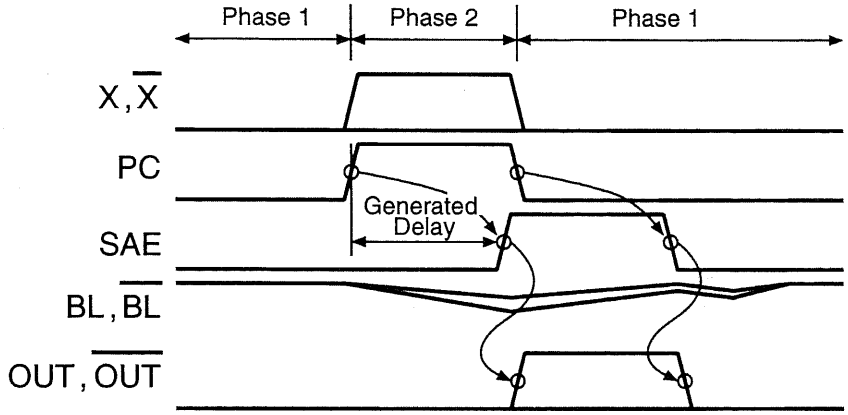


Figure 3.5 Timing diagram of control signals, input and output signals, and bit-line potentials in a column circuit.

In phase 2, $\overline{BL}$ is pulled down by charge sharing with VG through a reference cell. When at least one of the basic logic cells pulls BL down, the voltage potential of BL becomes lower than that of $\overline{BL}$. Otherwise, BL stays high. This is because the device size of the basic logic cells is W, while that of the reference cell is $0.5W$ as shown in Figure 3.2, where W is the channel width of a transistor. This half-size device is provided to avoid the meta-stable condition, which may be caused when there is no pull-down path on BL. The *SAE* signal is activated when the developed voltage difference between the bit-lines becomes larger than the designed sense voltage, which takes the worst case of considerable noise margin and process variations into account. By activating the sense amplifier, one of the output signals, i.e., *OUT* or $\overline{OUT}$ becomes high depending on the developed voltage difference. After the activation of the sense amplifier, the *PC* signal becomes low and the circuit starts to precharge the bit-lines.

Also, in phase 2, short-circuit current between input wires may be caused depending on the combinations of the basic logic cells. For example, when the state of ($\overline{x}_0$, $\overline{x}_1$, x_2) = (low, high, high) arises in the circuit of Figure 3.2, $\overline{x}_0$ pulls BL down via the pass-transistor controlled by $\overline{x}_1$. At the same time, $\overline{x}_1$ tries to drive BL into high via the pass-transistor controlled by x_2. Thus, this makes a current path from $\overline{x}_1$ to $\overline{x}_0$. However, short-circuit current between the input wires is caused only when the voltage potential of

BL becomes lower than $V_{DD} - V_t$ because an NMOS transistor is used as a pull-up device in this case, where V_{DD} is the power supply voltage and V_t is the threshold voltage of the transistor. This also means that the basic logic cells never drive BL into V_{DD}. Moreover, the threshold voltage (V_t) of the pull-up device is not constant with respect to the voltage difference between the substrate and the source of the transistor, and is enhanced because the source of the pull-up device is precharged high. This is known as the *body effect* [26] and helps to reduce the short-circuit current. Also, when avoiding the short-circuit current is required, which is often the case with systems where power consumption is the primary concern, it is realized by introducing a restriction in how to combine the basic logic cells, i.e., by avoiding two or more 2-input logic cells using pass-transistors in a column circuit. We evaluated and confirmed operations of LCPLAs, which take all possible cases of the combinations of the basic logic cells into account, in various logic functions in the following sections.

3.3.2 PLA Configuration

The overall configuration of LCPLA is shown in Figure 3.6. An array of column circuits is used as an AND-plane and an OR-plane. Control signals of the AND-plane are generated from the *CLK* signal with a delay line of a chain of sized inverters. On the other hand, control signals of the OR-plane are generated from a dummy column and a delay line of a chain of sized inverters. The dummy column is designed so that its output signal arrives last in the AND-plane. For this purpose, it has the largest number of basic logic cells in the AND-plane, and gate terminals of the basic logic cells are connected to ground. Its output signal, i.e., the *PC* signal of the OR-plane is generated every cycle and follows operating conditions, such as temperature and supply voltage variations, as well as process variations in sync with column circuits in the AND-plane.

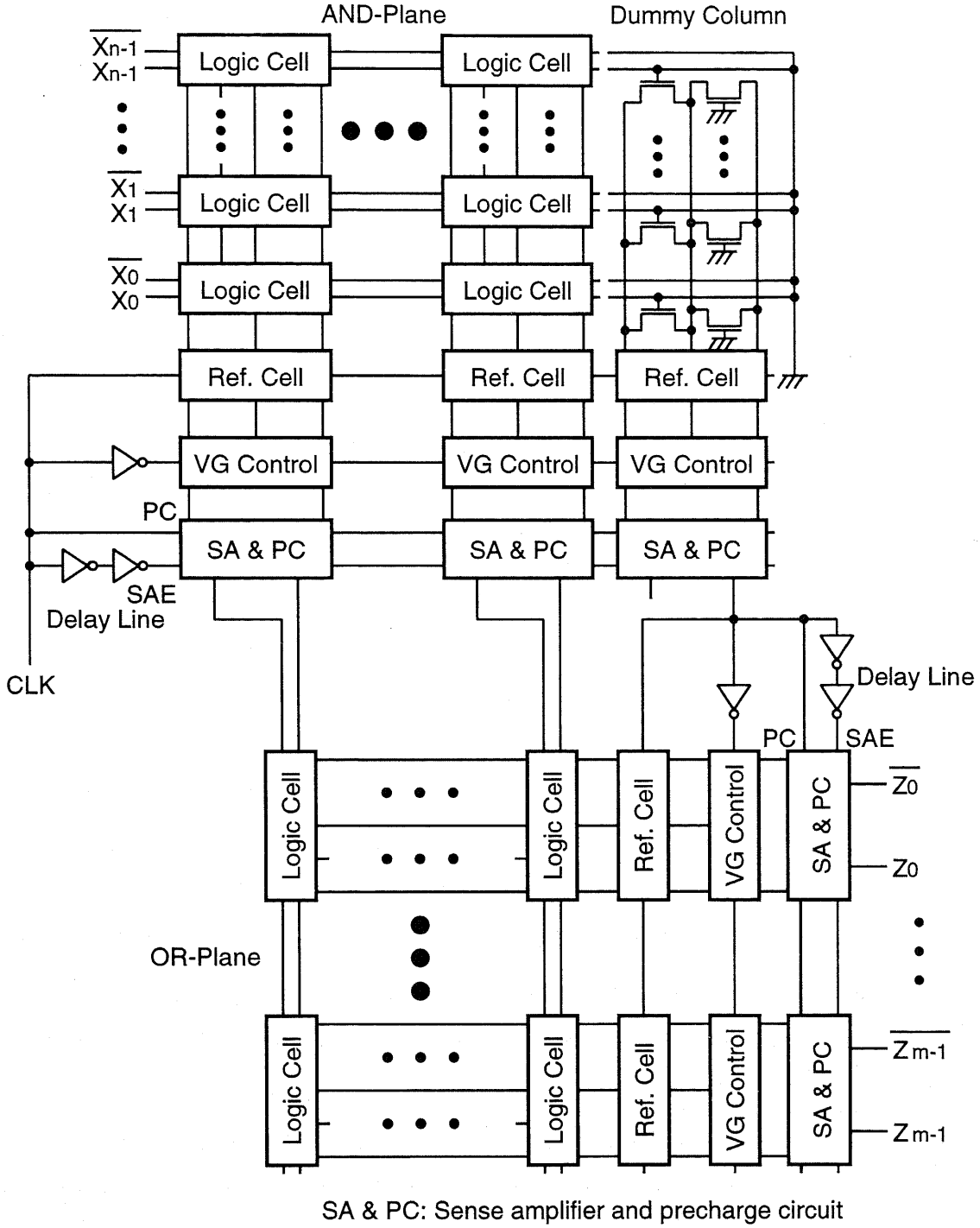


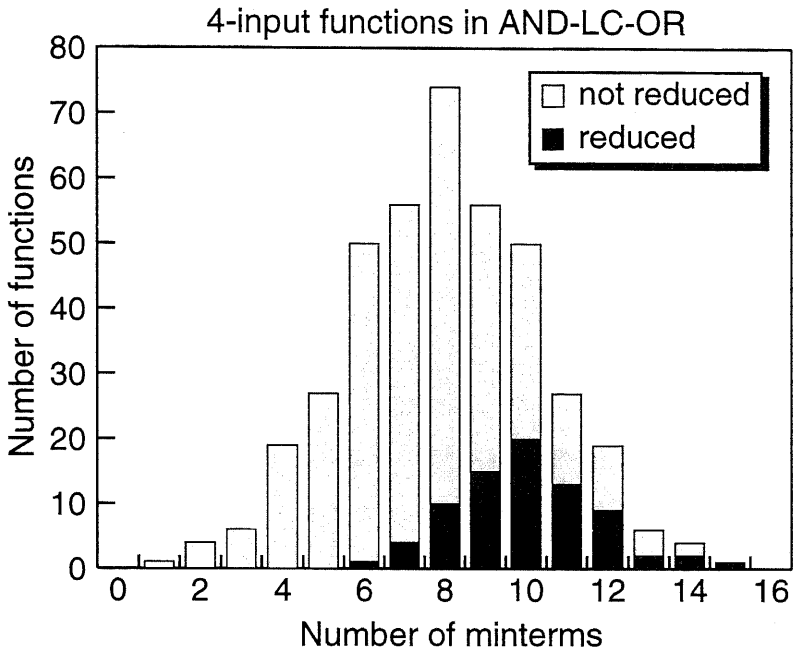
Figure 3.6 Overall configuration of LCPLA.

3.4 Results of Logic Synthesis

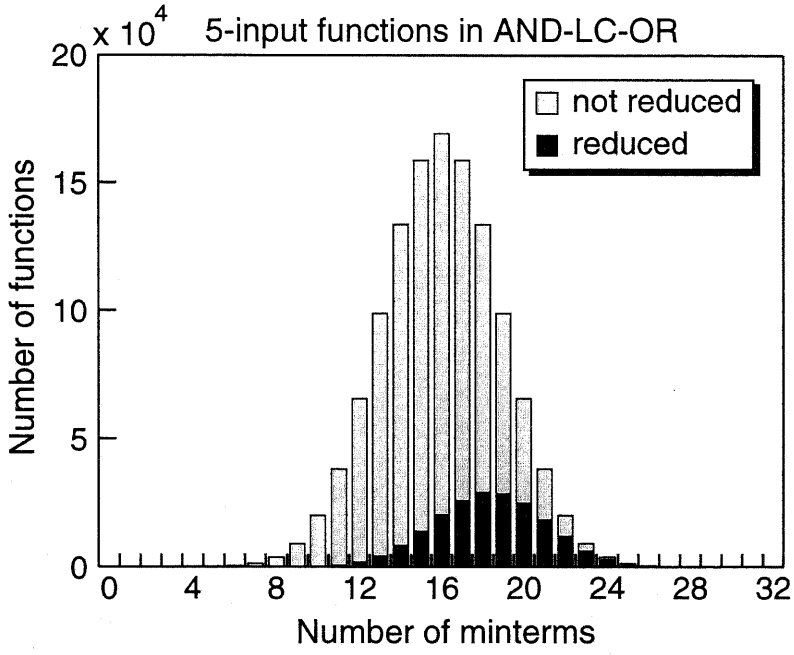
In order to implement a given logic function into an LCPLA, we proposed a method of logic synthesis for LC-AND-LC-OR logic structures [27]. By implementing the method as a part of ESPRESSO-MV [28], the area advantage over the conventional structure was demonstrated.

Figure 3.7 and Figure 3.8 show the number of 4- and 5-input logic functions which have a smaller number of product terms than that of the functions produced by ESPRESSO-MV. In this figure, AND-LC-OR and LC-AND-LC-OR correspond to PLAs which have 2-input logic cells in OR-plane and both planes, respectively. Since it is not necessary to deal with all the functions that are expressed by the negation and permutation of some variables in a Boolean function, the Boolean functions in the *NP-equivalence class* [29] were examined. In the design of a PLA, minimizing the number of product terms (i.e., the number of column circuits in an AND-plane) results in reducing the overall area. Also, an LCPLA can be implemented in the same area as the proposed PLA in Chapter 2 even if the number of product terms can not be reduced by using 2-input logic cells. The results show that LCPLA is effective to reduce the number of product terms, especially in the LC-AND-LC-OR structure.

Table 3.1 shows the results of logic synthesis on the PLA benchmark circuits [8], [9]. In this table, LC-AND-OR, AND-LC-OR, and LC-AND-LC-OR correspond to PLAs which have 2-input logic cells in AND-plane, OR-plane, and both planes, respectively. The results show that the LC-AND-LC-OR structure can realize the Boolean functions in the least number of product terms among the four types of PLAs. In particular, the results indicate that the LCPLA is suitable for arithmetic circuits, such as add6, which are difficult to be implemented in the conventional PLA [30]. By using LC-AND-LC-OR structures, the number of product terms was reduced by 55.9% on average compared to that in AND-OR structures.

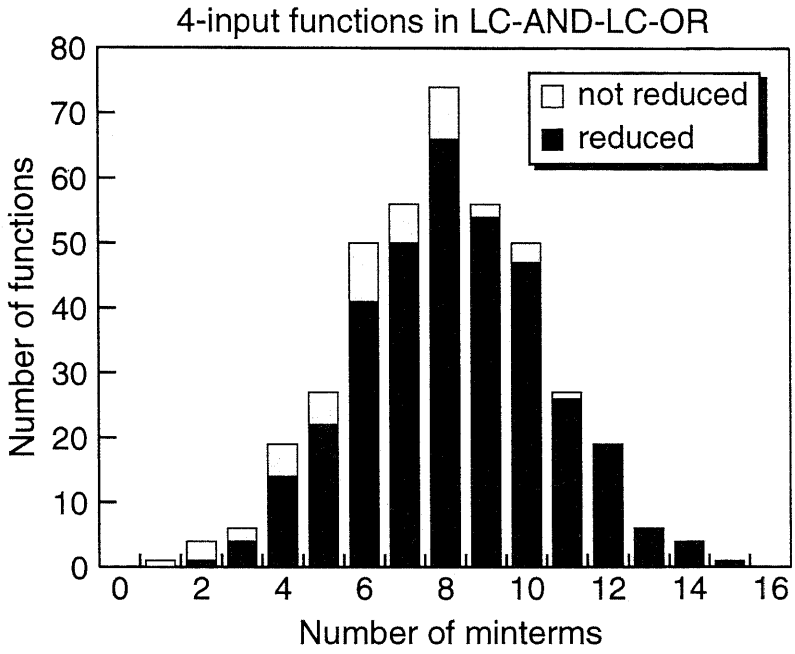


(a)

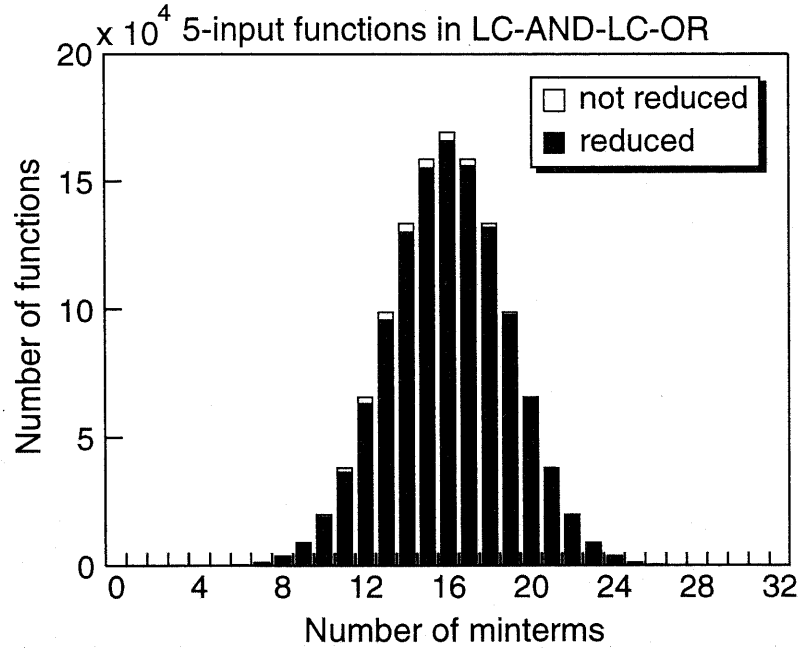


(b)

Figure 3.7 The number of logic functions which have a smaller number of product terms than that of the functions produced by ESPRESSO-MV in AND-LC-OR structures. (a) 4-input functions and (b) 5-input functions.



(a)



(b)

Figure 3.8 The number of logic functions which have a smaller number of product terms than that of the functions produced by ESPRESSO-MV in LC-AND-LC-OR structures. (a) 4-input functions and (b) 5-input functions.

Table 3.1 Results of logic synthesis ("Time" is CPU time on Pentium III (600 MHz) processor for synthesis).

	AND-OR		LC-AND-OR		AND-LC-OR		LC-AND-LC-OR	
Circuit	# product terms	Time (sec)	# product terms	Time (sec)	# product terms	Time (sec)	# product terms	Time (sec)
Z5xp1	65	0.1	53	0.1	62	0.6	52	2.0
add6	355	0.5	37	0.1	325	93.2	37	1.5
addm4	200	0.4	109	0.3	193	4.9	99	13.4
adr4	75	0.1	17	0.0	69	0.7	17	0.1
dist	123	0.1	75	0.1	120	2.3	70	4.8
f51m	77	0.1	51	0.1	69	0.4	48	2.0
l8err	52	0.1	39	0.0	49	0.6	38	1.0
m181	42	0.1	30	0.1	40	0.2	28	14.6
mlp4	128	0.2	97	0.1	124	1.3	92	25.6
rd73	127	0.0	37	0.0	113	5.0	34	1.4
root	57	0.1	42	0.0	52	0.9	40	1.4
sqr6	49	0.0	42	0.0	49	0.1	40	0.3
Total	1,350	1.8	629	0.9	1,265	110.2	595	68.1
Ratio	1.00	1.00	0.466	0.500	0.937	61.2	0.441	37.8

3.5 Simulation Results

Table 3.2 shows a comparison of standard-cell-based designs and LCPLA on the PLA benchmark circuits [8], [9]. Each circuit was designed using a 0.35-μm, 3-metal-layer CMOS technology with a supply voltage of 3.3 V. Logic synthesis of the standard-cell-based designs was performed using Synopsys Design Compiler [31] with delay priority and 395 standard cells. The standard cells are based on the Rohm 0.35-μm CMOS technology and were provided by VLSI Design and Education Center (VDEC) [32]. The synthesized circuits were placed and routed using Synopsys Apollo [31]. The delay and power characteristics of the standard-cell-based designs were obtained by the timing and power analysis of Synopsys Design Compiler with wire load information, which is given

Table 3.2 Comparison of standard-cell-based designs and LCPLA.

	Area (μm^2)		Delay (ns)		Power (mW) @ 200 MHz		PD product (pJ)	
Circuit	Standard cell	This work	Standard cell	This work	Standard cell	This work	Standard cell	This work
Z5xp1	14,042	83,785	1.22	0.69	7.8	23.0	9.52	15.87
add6	17,110	68,507	1.28	0.68	10.2	16.9	13.06	11.49
addm4	56,608	143,503	1.53	0.74	30.4	45.2	46.51	33.45
adr4	5,561	32,727	0.94	0.65	3.3	8.0	3.10	5.20
dist	47,306	90,381	1.64	0.71	24.6	30.5	40.34	21.66
f51m	14,042	75,429	1.12	0.69	8.2	21.6	9.18	14.90
l8err	15,812	63,081	1.36	0.67	8.6	17.2	11.70	11.52
m181	10,700	65,832	0.83	0.67	5.9	14.2	4.90	9.51
mlp4	47,306	129,760	1.32	0.73	24.0	39.6	31.68	28.91
rd73	18,088	44,463	1.15	0.67	10.7	15.2	12.31	10.18
root	19,418	57,747	1.28	0.68	10.9	17.7	13.95	12.04
sqr6	15,340	71,124	1.07	0.67	8.6	19.3	9.20	12.93
Total	281,333	926,339	14.74	8.25	153.2	268.4	205.45	187.66
Ratio	1.00	3.29	1.00	0.560	1.00	1.75	1.00	0.913

by placement and routing, while the area characteristics were obtained by physical layouts after placement and routing. On the other hand, the delay and power characteristics of LCPLA were obtained from post-layout simulations using HSPICE, where LC-AND-LC-OR structures were used. Also, the area characteristics represent physical layouts of LC-AND-LC-OR structures, and a sense voltage of 200 mV was used. The results show that the LCPLA requires on average 3.29 times the area of the standard-cell-based designs. This is mainly because of dual-rail structures and the difference of area-efficiency between regular and random structures. The delay of the LCPLA is on average 0.560 times that of the standard-cell-based designs. In addition to the high-speed capability of sense amplifiers in the proposed structure, the LCPLA implements logic functions in 2-level logic forms with a two-level logic network and thus results in having superior delay

Table 3.3 Comparison of the conventional PLAs and LCPLA in an example circuit.

Circuit (0.35-μm, 3.3-V V_{DD})	Configuration	Area (μm^2)	Delay (ns)	Power (mW) @ 80 MHz	PD product (normalized)
PLA using domino circuit [26]	single-rail	489,440	3.56	95	1.00
Blair *et al.* [33]	single-rail	489,582	3.08	405	3.69
Dhong *et al.* [34]	single-rail	492,048	2.80	176	1.46
Wang *et al.* [35]	single-rail	498,078	2.56	92	0.696
Proposed PLA in Chapter 2	dual-rail	939,798	1.52	73	0.328
LCPLA (LC-AND-LC-OR)	dual-rail	601,624	1.39	43	0.177

characteristics. In the standard-cell-based designs, the increase in delay is incurred in traversing the different logic levels of the design. Moreover, the maximum delay of the standard-cell-based designs is 1.64 ns (dist), while the minimum delay is 0.83 ns (m181). Thus, the maximum difference of delay between each circuit is 0.81 ns. On the other hand, the maximum delay of the LCPLA is 0.74 ns (addm4), while the minimum delay is 0.65 ns (adr4). Thus, the difference of delay between each circuit is limited only within 0.09 ns. Also, the worst case delay of LCPLA can be easily obtained from the delay of the column circuit having a maximally loaded bit-line. This feature of LCPLA makes the circuit delay predictable early in the design process, and thus it is possible to eliminate the ambiguity of the design and to alleviate the timing closure problem. The power consumption of the LCPLA is on average 1.75 times that of the standard-cell-based designs. This is mainly because input signal wires and bit-lines are charged and discharged every cycle, and the power consumption of sense amplifiers in LCPLA is relatively large compared to the other components. However, the increase in power consumption becomes small as the number of input signals increases because a sense amplifier consumes the same power even in larger circuits. The power-delay (PD) product of the LCPLA is on average 0.913 times that of the standard-cell-based designs.

Table 3.3 shows a comparison of the conventional PLAs and LCPLA. In order to compare the conventional PLAs with LCPLA, we designed an example circuit using a 0.35-

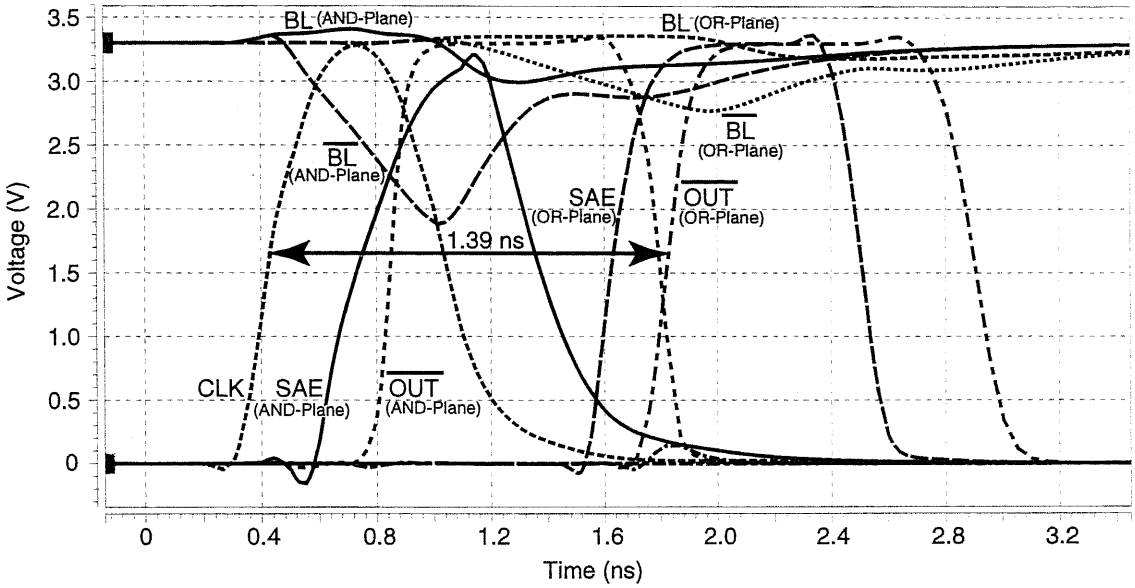


Figure 3.9 Simulated waveforms of the LCPLA.

μm, 3-metal-layer CMOS technology with a supply voltage of 3.3 V. The example circuit has a 64-bit input and a 1-bit output, and its logic function was generated randomly. The generated logic function was optimized using ESPRESSO [8], [9] before implementation. The number of product terms and the number of basic logic cells in the optimized logic function are 220 and 3,368, respectively. This circuit scale is relatively large over the PLA benchmark circuits [8], [9], and the circuit becomes a good criterion as a large PLA, which suits the recent microprocessors [3], [4]. Also, since area, delay, and power consumption of a PLA mainly depend on the numbers of product terms and basic logic cells if the numbers of inputs and outputs are fixed, we can see how much the reductions of the numbers of product terms and basic logic cells affect on these circuit characteristics by using the example circuit. The performances were obtained from post-layout simulations using HSPICE. Simulated waveforms of the LCPLA are shown in Figure 3.9. By using an LC-AND-LC-OR structure, the number of product terms was reduced from 220 to 136 (38.2% reduction) and the number of basic logic cells was reduced from 3,368 to 2,042 (39.4% reduction). As a result, the proposed circuit achieved an area reduction by 36.0% compared to the proposed PLA in Chapter 2. Also, the circuit delay and power

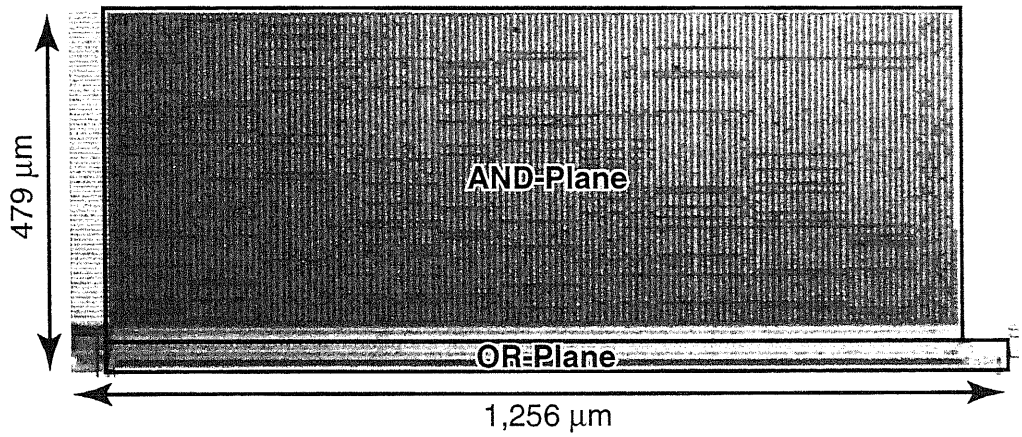


Figure 3.10 Chip microphotograph of the LCPLA.

consumption of the LCPLA are the smallest among all of the PLAs in Table 3.3, and the power-delay (PD) product was reduced by 74.6% and 46.0% compared to the conventional high-speed and low-power single-rail PLA [35] and the proposed PLA in Chapter 2, respectively.

3.6 Measurement Results

The LCPLA in Table 3.3 was fabricated using a 0.35-μm, 3-metal-layer CMOS technology. A chip microphotograph is shown in Figure 3.10, which includes 136 and 1 column circuits in the AND-plane and the OR-plane, respectively. The physical dimension is 479 $\times$ 1,256 μm^2. The operations of the LCPLA were successfully verified with a functional test using a logic tester at frequencies of up to 100 MHz, which is the limitation of the test equipment, at room temperature with a supply voltage of 3.3 V.

In order to measure the propagation delay of the test chip, a delay measurement using an electron-beam tester was also performed at the same conditions as the functional test. Figure 3.11 shows measured waveforms along the critical path. The timebase resolution of the test equipment is 10 ps. The measured delay from the *CLK* signal to the output signal of the PLA was 1.46 ns, while the power consumption is 241 mW at a simulated frequency of 500 MHz. The features of the test chip are summarized in Table 3.4.

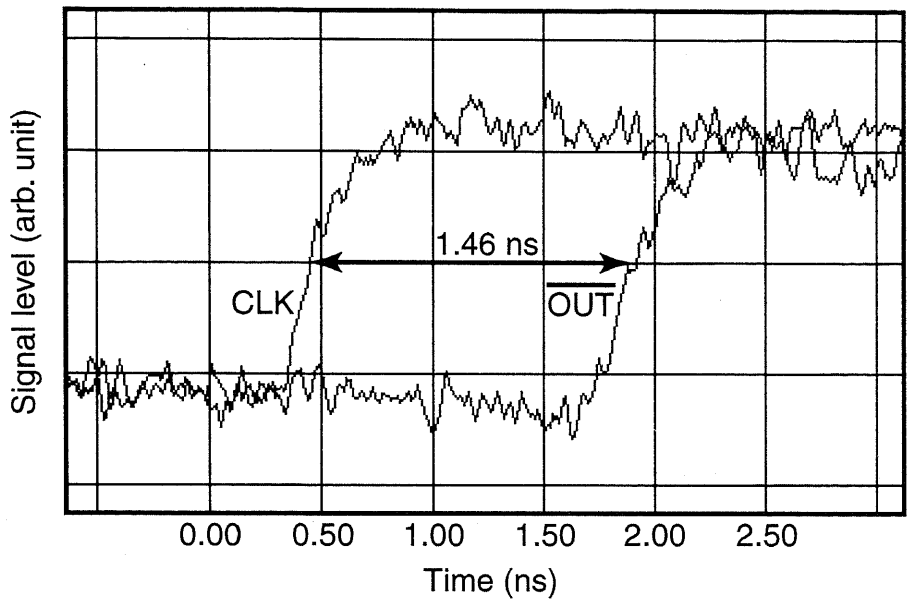


Figure 3.11 Measured waveforms by an electron-beam tester.

Table 3.4 Features of the test chip.

Process technology	0.35-μm CMOS, 3-metal
Supply voltage	3.3 V
Transistor count	7.1 k
# product terms	136
# basic logic cells (AND-plane)	1,906
# basic logic cells (OR-plane)	136
Physical dimension	479 $\times$ 1,256 μm^2
Simulated cycle time	1.98 ns (500 MHz)
Simulated power consumption	241 mW (@ 500 MHz)
Measured delay time	1.46 ns

3.7 Methodology of Automatic Module Generation

In order to automatically generate LCPLA layouts from circuit specifications, we developed a module generator as a design automation tool of LCPLA with a structural improvement for the generator. In the circuit in Figure 3.6, a sense amplifier activation pulse is generated from a chain of sized inverters. However, designing a chain of sized inverters according to a desired timing margin makes the design complex. To reduce the design complexity, we propose a delay generator, which is composed of dynamic circuits with dummy cells, as shown in Figure 3.12. Dummy cells are designed in the minimum size. The delay from the *PC* signal to the *SAE* signal depends on the number of dummy cells in a delay generator. The relationship between the generated delay and the number of dummy cells is shown in Figure 3.13. A desired timing signal can be easily generated by changing the number of dummy cells.

Figure 3.14 shows the generator organization. The generator takes a personality matrix, a target delay, and physical layouts of basic components except for logic cells. A personality matrix represents a desired circuit configuration to be implemented. Finally, the generator produces an LCPLA layout in a GDS-II form [36] and a timing-annotated HDL behavior model. In the following subsections, the input data format and the generator modules are described in detail.

3.7.1 Personality Matrix

A personality matrix represents a circuit configuration of an LCPLA structure using the basic logic cells shown in Figure 3.3. We defined symbols and corresponding basic logic cells in a personality matrix as shown in Table 3.5. By using this representation, a circuit configuration using an LCPLA structure, for example, can be expressed as shown in Figure 3.15. The matrix on the left and right represents circuit configurations in an AND-plane and an OR-plane, respectively. A personality matrix optimized for an LCPLA structure can be obtained from the developed method of logic synthesis [27].

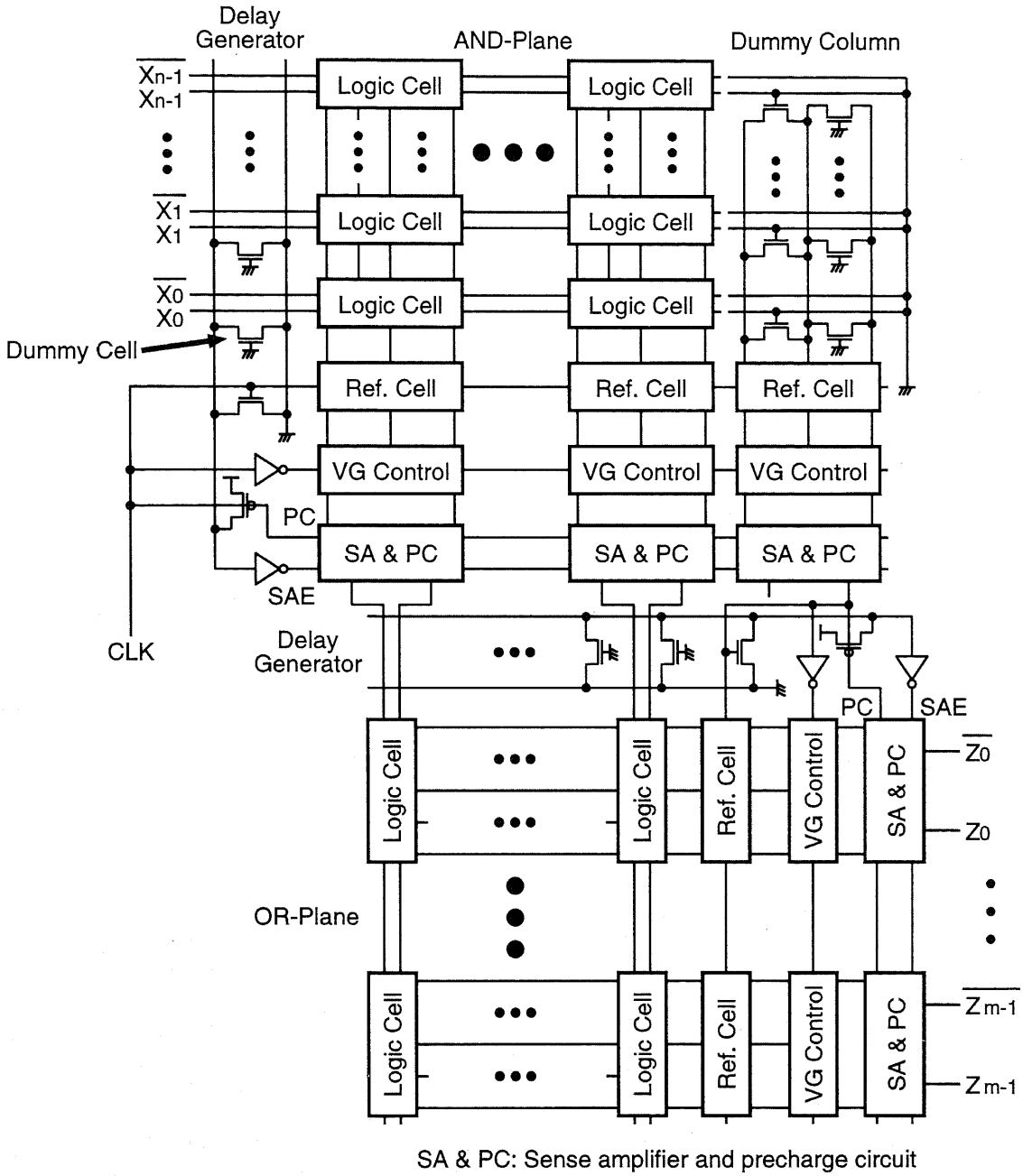


Figure 3.12 LCPLA configuration with delay generators.

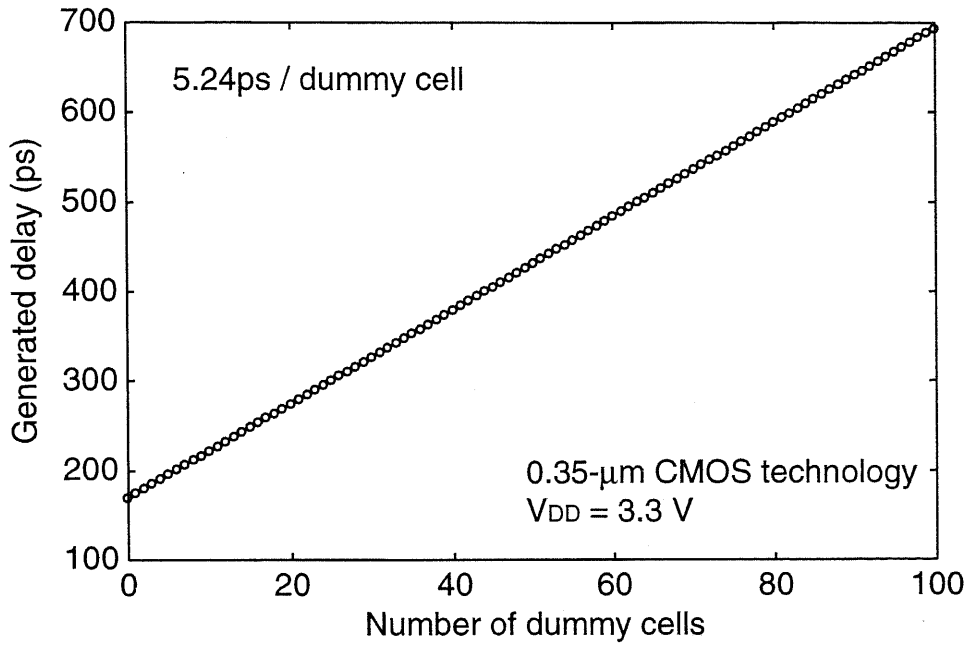


Figure 3.13 Control range of a delay generator.

3.7.2 Layout Generation

The logic cell generation module takes a target delay and generates physical layouts of the basic logic cells shown in Figure 3.3. Generated layouts are stored in a library with the other circuit components given by users. The module performs equation-based transistor sizing to achieve a desired delay goal. Widening the sizes of basic logic cells results in reducing the circuit delay. The sizes of basic logic cells give that those of a reference cell and driver circuits based on the maximally loaded input wire and bit-line.

The basic cell placement module takes a personality matrix and basic cell layouts from a library, and generates an LCPLA layout. The placement is performed based on a circuit configuration represented by a personality matrix, design rules, and the size information of basic cells. The number of dummy cells in a delay generator is determined so that sense amplifiers are activated when the developed voltage difference between the maximally loaded bit-lines becomes larger than the designed sense voltage. As can be seen from Figure 3.13, a delay generator has the linearity in delay with respect to the number of

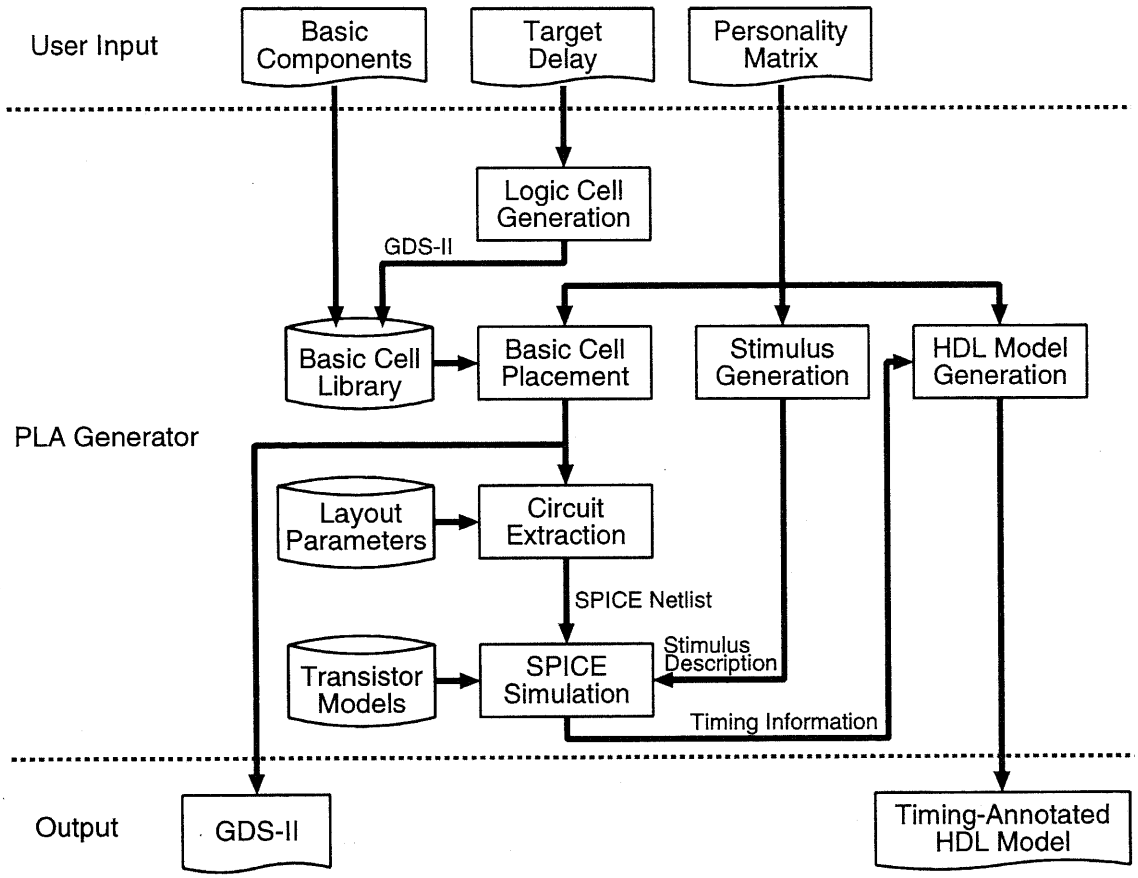


Figure 3.14 Generator organization.

Table 3.5 Symbols and corresponding basic logic cells in a personality matrix.

Symbol	Basic logic cell
-	type 0
0	type 1
1	type 2
2	type 3
3	type 4
4	type 5
5	type 6
6	type 7
7	type 8
8	type 9
9	type 10

x_0 x_1 x_2 x_3	z_0 z_1 z_2
- 0 3 2	0 - 6
8 - 0 1	- 2 8
- 4 8 -	2 - -

Figure 3.15 Example of a personality matrix (4-bit inputs, 3-bit outputs, and 3 product terms).

dummy cells. By using this feature, a delay generator can be easily characterized.

3.7.3 Behavior Model Generation

The HDL model generation module takes a personality matrix and timing information given by SPICE simulations. Based on these information, the module generates a timing-annotated HDL behavior model which includes data setup and hold times in addition to a circuit delay. The data setup time depends on the drive capability of input drivers, and the data hold time is defined when the developed voltage difference between the maximally loaded bit-lines becomes larger than the designed sense voltage. As shown in

```
module pla(clk, x, z);
  input clk;
  input[3:0] x;
  output[2:0] z;
  reg[2:0] y;
  parameter delay = 0.89;
  always @(posedge clk) begin
    y[0] <= x[1] | ~x[1] & x[2] | x[2] & x[3];
    y[1] <= ~x[0] & x[1] | x[2] | ~x[3];
    y[2] <= x[0] & ~x[1] | ~x[2] & x[3];
  end
  specify
    $setup (x, posedge clk, 0.10);
    $hold  (posedge clk, x, 0.18);
  endspecify
  assign #delay z[0] = y[0] | y[1] & y[2];
  assign #delay z[1] = y[0] & y[1];
  assign #delay z[2] = y[0] & y[1] | ~y[1] & y[2];
endmodule
```

Figure 3.16 Example of an HDL behavior model (4-bit inputs, 3-bit outputs, and 3 product terms).

Table 3.6 Circuit delay and CPU time of LCPLA generation in example circuits.

Circuit (0.35-μm, 3.3-V V_{DD})	# inputs	# product terms	# logic cells	Delay (ns)	CPU time (sec)
example 1	16	87	621	1.01	33.1
example 2	32	144	1,459	1.25	117.8
example 3	64	220	3,368	1.39	441.2

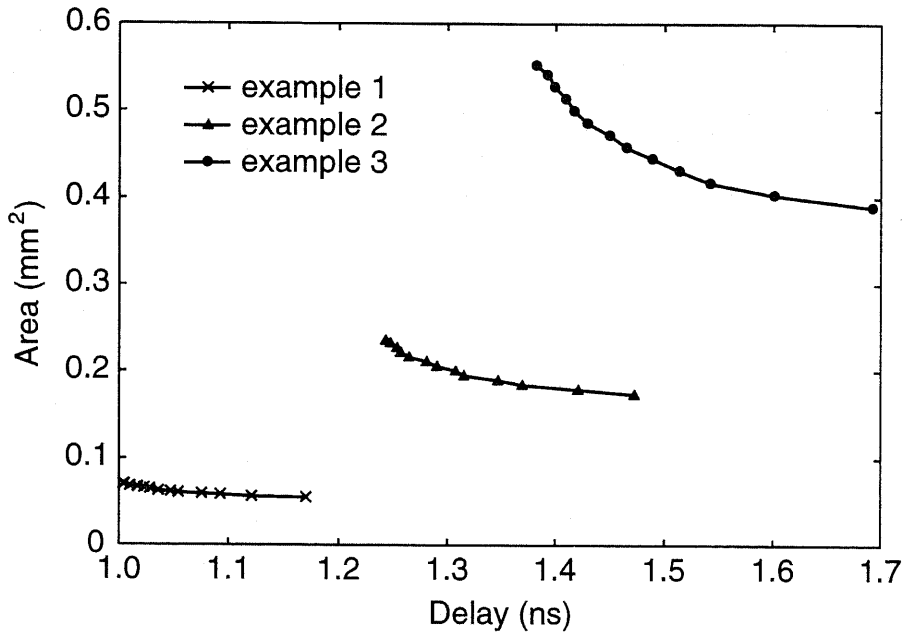


Figure 3.17 Relationship between the circuit delay and area.

Figure 3.5 and Figure 3.12, the output signals of an LCPLA are generated by activating sense amplifiers and thus there is no skew between the output signals. This indicates that a circuit delay described in an HDL behavior model means the propagation delay of all the output signals. Thus, the delay behavior of an LCPLA can be easily represented by this model. This is another advantage of LCPLA structure. Figure 3.16 shows an example of an HDL behavior model which represents a circuit configuration given by the personality matrix in Figure 3.15.

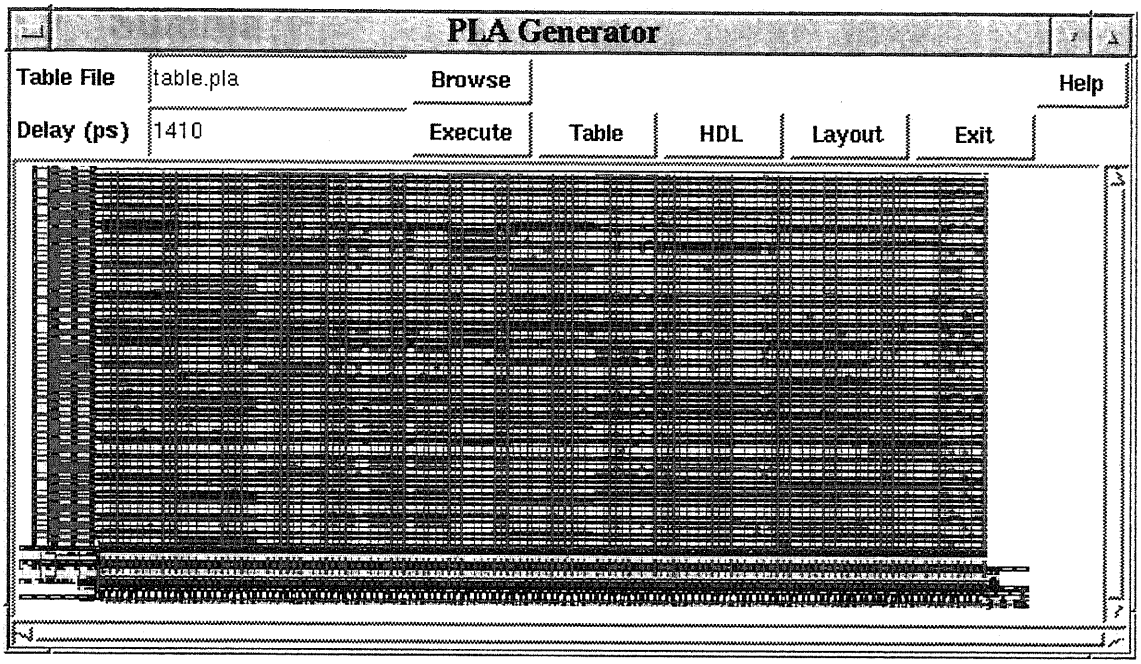


Figure 3.18 Overview of the LCPLA generator and a generated layout of the example 3.

3.8 Generation Results

The module organization shown in Figure 3.14 was integrated and applied to various logic circuits. Table 3.6 shows the circuit delay and CPU time of LCPLA generation in three example circuits. The example circuits have appropriate circuit sizes in practical applications and were generated using a 0.35-μm, 3-metal-layer CMOS technology with a supply voltage of 3.3 V. PLA generations were performed to minimize the circuit delay on a Sun UltraSPARC-III 900 MHz processor. The results show that the entire layout and the behavior model of each circuit can be obtained within 8 minutes. Figure 3.17 shows the relationship between the circuit delay and area which were obtained by traversing transistor size spaces in the example circuits. The results show that for larger PLAs, the generator allows a wide range of performance to be traded for area. Figure 3.18 shows the overview of the LCPLA generator and a generated layout of example 3.

3.9 Summary

In this chapter, an area-efficient dual-rail array logic architecture, a logic-cell-embedded PLA (LCPLA), was presented. By embedding 2-input logic cells, which realize arbitrary 2-input Boolean functions, in the structure, some classes of logic functions can be implemented efficiently, so that high-speed and low-power operations were also achieved. The logic cells can be designed by connecting some local wires and do not require additional transistors over logic cells of the proposed PLA in Chapter 2. The advantages over the conventional PLAs and standard-cell-based designs were demonstrated by using benchmark circuits and a developed logic synthesis method, and the LCPLA is shown to be effective to reduce the number of product terms. In a structure with a 64-bit input and a 1-bit output including 220 product terms, the LCPLA using a 0.35-μm CMOS technology achieved an area reduction by 36.0% compared to the proposed PLA in Chapter 2, and the power-delay product was reduced by 74.6% and 46.0% compared to the conventional high-speed single-rail PLA and the proposed PLA in Chapter 2, respectively. A test chip of this configuration was fabricated using a 0.35-μm, 3-metal-layer CMOS technology, and was verified with a functional test using a logic tester and an electron-beam tester at frequencies of up to 100 MHz with a supply voltage of 3.3 V. Also, a module generator for LCPLA structures was developed as a design automation tool. The generator performs equation-based transistor sizing to achieve a desired delay goal and allows a wide range of performance to be traded for area. The framework and procedure of the generator are much easier than those of the conventional tools for standard-cell-based designs and thus the generator can be easily applied to complicated IC designs.

Chapter 4

Delay Reduction and Physical Compaction by Divided and Interdigitated Column Circuits

4.1 Introduction

In the column circuits discussed in the previous chapters, fast OR-ing and AND-ing operations can be performed by using the high-speed sensing capability of the circuit. However, the circuit delay increases in proportion to the number of input signals and thus becomes large due to a large bit-line capacitance in high data bandwidth processing such as 128-bit systems. In high data bandwidth systems such as 128-bit graphic engines [37], [38], a high-speed operation is an important factor to realize real-time image processing and so forth. Moreover, as can be seen in an arithmetic circuit shown in Chapter 2, an unused chip area increases as the number of input signals increases.

In this chapter, we propose circuit design techniques for a high-speed and area-efficient dual-rail array logic architecture. The proposed circuit includes three schemes: 1) a *divided column scheme (DCS)*, 2) a *programmable sense-amplifier activation scheme (PSAS)*, and 3) an *interdigitated column scheme (ICS)*. In the DCS, a column circuit of

a PLA is divided and each circuit operates in parallel. This enhances the performance of the PLA, and this scheme becomes more effective as input data bandwidth increases. The PSAS is used to generate an activation pulse for sense amplifiers in the PLA. In this scheme, the proposed delay generators enable to minimize a timing margin depending on process variations and operating conditions. The ICS is used to enhance the area-efficiency of the PLA, where a method of physical compaction is employed. This scheme is effective for circuits which have the regularity in logic function such as arithmetic circuits. We applied these schemes to some arithmetic circuits and evaluated the effectiveness of the proposed schemes.

4.2 Proposed Circuit Design Techniques

4.2.1 Divided Column Scheme

As discussed in Chapter 2, the circuit delay of the column circuit in Figure 4.1 increases in proportion to the number of input signals and thus becomes large due to a large bit-line capacitance in high data bandwidth processing. To reduce the circuit delay, we propose a divided column scheme (DCS). Figure 4.2 shows a divided column circuit using the DCS. In the circuit, the bit-line (BL) is divided into two wires (BL_1 and BL_2) with logic cells ($\overline{BL}$ is also divided in the same manner), and each bit-line has a sense amplifier and a control block. Also, the inverter gates, which are used as receiver circuits of a sense amplifier in Figure 4.1, are replaced with 2-input NAND and NOR gates to realize OR-ing and NOR-ing functions of the input signals. Dividing bit-lines reduces bit-line capacitances and thus results in a reduction of bit-line delay. The circuit delay of the column circuit in Figure 4.1 can be expressed as

$$n \cdot t_{BL} + t_{SA} + t_{INV} \tag{4.1}$$

where t_{BL} is the bit-line delay of a logic cell, and t_{SA} is the delay of a sense amplifier and also includes the bit-line delay of control circuits. t_{INV} is the delay of an inverter gate, and

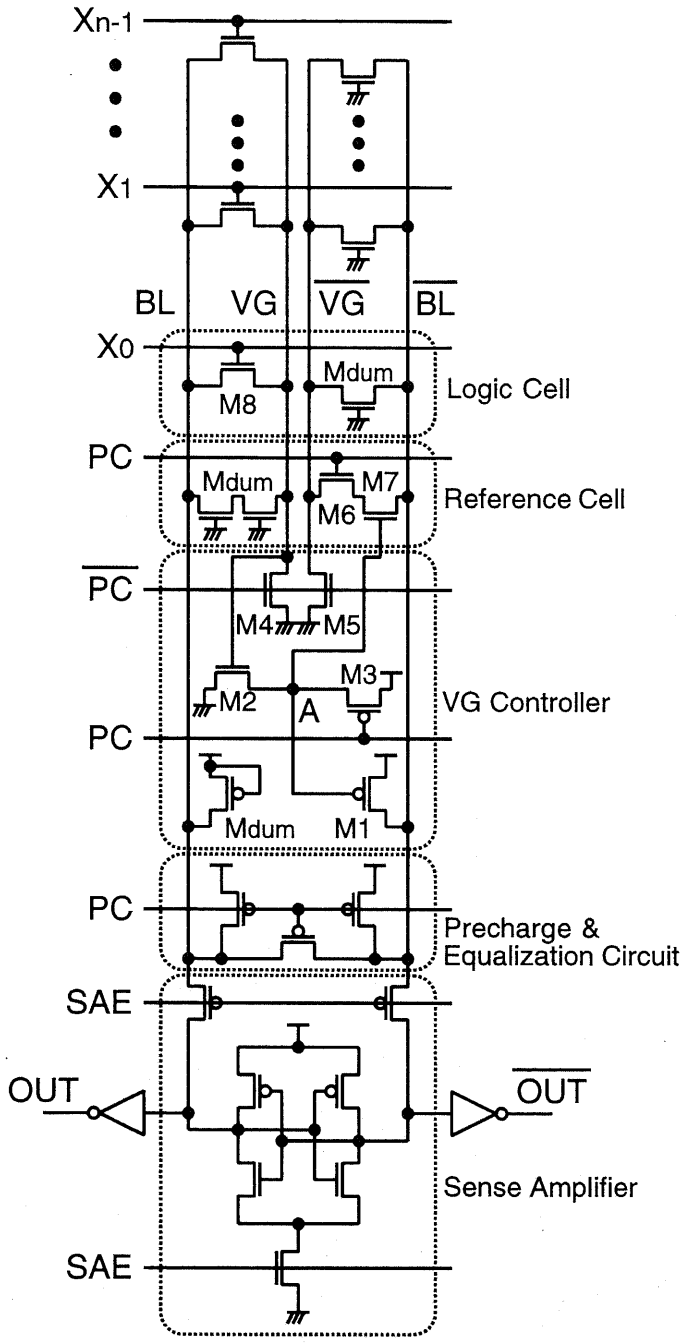


Figure 4.1 Column circuit proposed in Chapter 2.

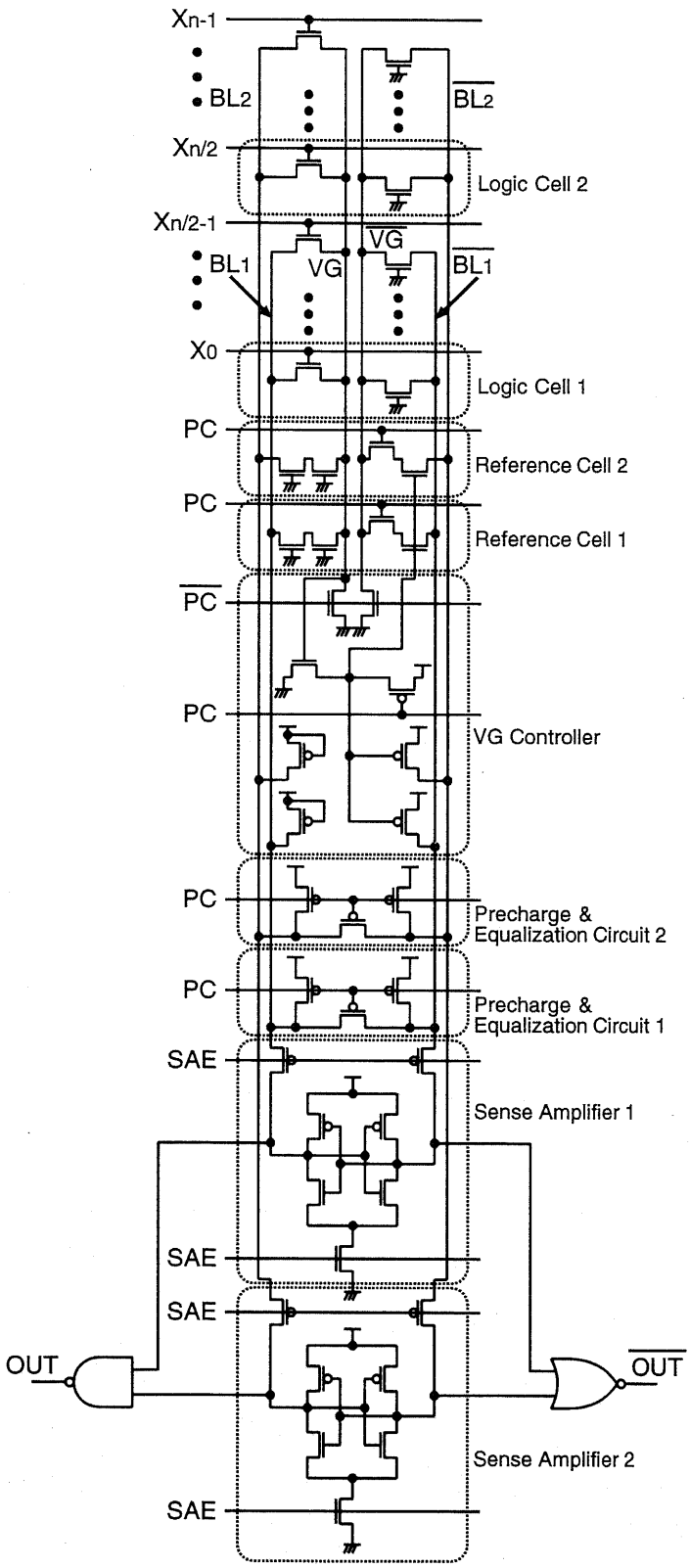


Figure 4.2 Divided column circuit using the DCS.

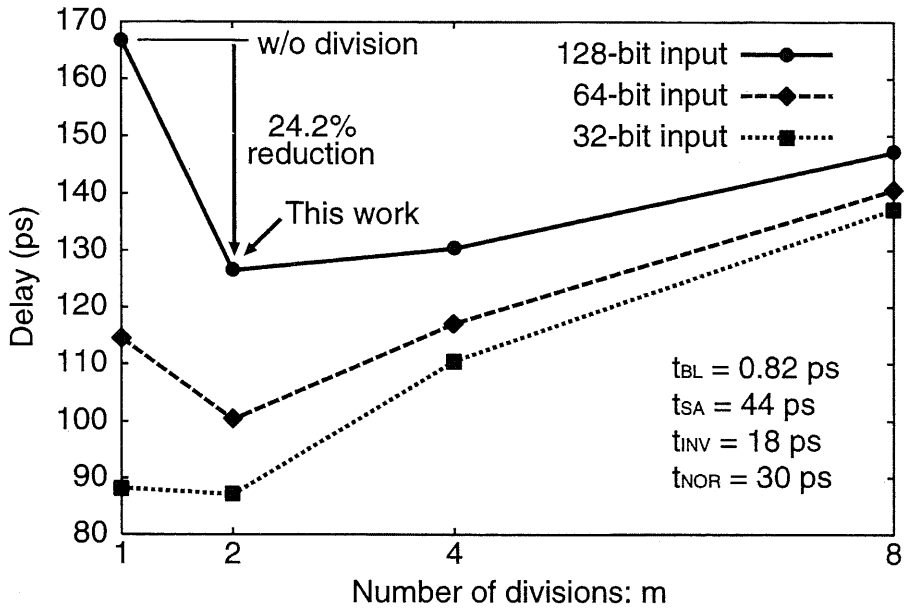


Figure 4.3 Relationship between the delay of a column circuit and the number of divisions (0.13-μm CMOS technology, 1.2-V power supply).

n is the number of input signals. On the other hand, the circuit delay of a column circuit divided into m bit-lines can be expressed as

$$\frac{n}{m} \cdot t_{BL} + t_{SA} + \log_2(m) \cdot t_{NOR} \qquad (m = 2^i, i \geq 1, t_{NOR} \geq t_{NAND}) \tag{4.2}$$

where t_{NOR} and t_{NAND} are the delays of a NOR gate and a NAND gate, respectively. Also, the number of divisions (m) is assumed to be 2^i to form NOR and NAND gates in tree-structured circuits. Thus, the delay of overall NOR gates has an increase of $O(\log_2(m))$. Although the bit-lines in control blocks in Figure 4.2 become longer than those in Figure 4.1 due to the duplication of a control block, the precharge and equalization circuits and the precharge circuits in VG controllers, which correspond to M1 in Figure 4.1, can be designed in smaller sizes due to the reduction of bit-line capacitances in logic cells. This reduces the bit-line capacitances in each control block, and the capacitances become smaller than that in Figure 4.1. The relationship between the delay of a column circuit and the number of divisions is shown in Figure 4.3. The performances were obtained

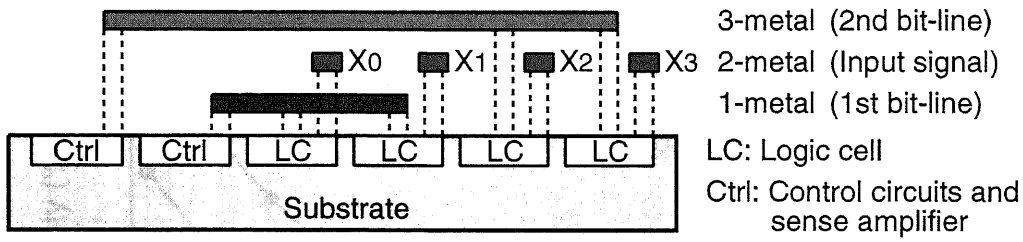


Figure 4.4 Physical layer structure in the DCS.

from post-layout simulations using a 0.13-μm CMOS technology with a supply voltage of 1.2 V. As can be seen from this figure, the high-speed capability of the DCS can be enhanced as the number of input signals increases. The circuit delay can be optimized by the number of divisions, and the DCS becomes most effective at the number of divisions of 2 in a 128-bit input column circuit. We chose the number of divisions of 2 in terms of performance and circuit area. As a result, the circuit delay can be reduced by 24.2% compared to the column circuit in Figure 4.1.

Figure 4.4 shows a physical layer structure in the DCS. The area penalty of the DCS can be reduced by using upper layer wires as bit-lines. Although the second bit-lines are longer than the first bit-lines due to the difference of the placement of logic cells, the capacitance to ground, which occupies a large part of wiring capacitance, can be reduced by using upper layer wires, while the bit-lines of the conventional PLAs are generally designed using lower layer wires. The column pitch depends on the width of a logic cell in a column circuit. Figure 4.5 shows the relationship between the area and power penalties of the DCS and the number of input signals, where a column circuit having the number of divisions of 2 was compared with the column circuit in Figure 4.1, and the input signals of the divided column circuit are formed in the same manner as the configuration shown in Figure 4.2. The penalties due to an additional sense amplifier and a control block decrease as the number of input signals increases. This is because a sense amplifier and a control block consume almost the same area and power even in larger circuits. The area and power penalties in a 128-bit input column circuit are 3.6% and 16.5%, respectively.

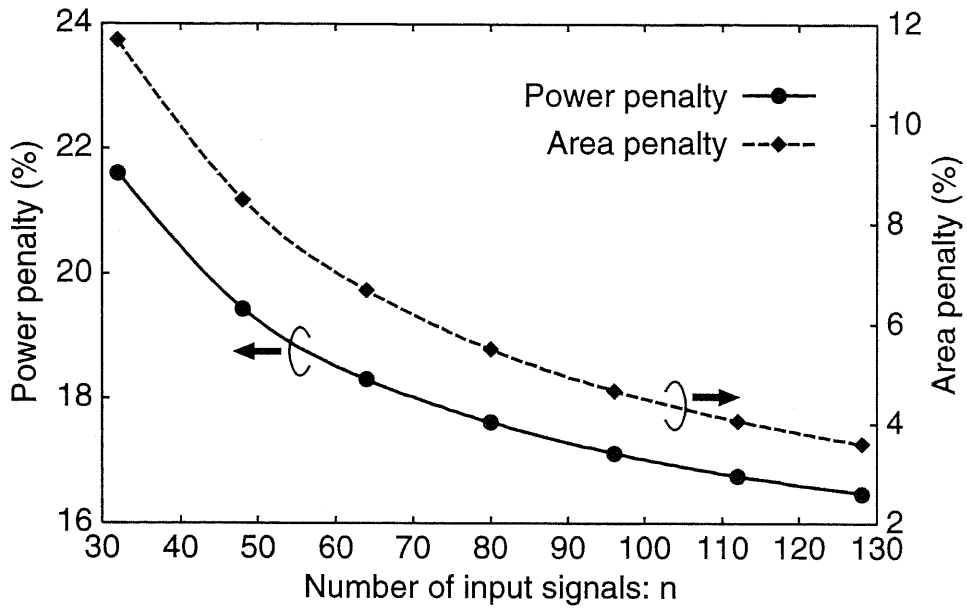


Figure 4.5 Relationship between the area and power penalties of the DCS and the number of input signals (0.13-μm CMOS technology, 1.2-V power supply).

4.2.2 Programmable Sense-Amplifier Activation Scheme

The proposed PLA configuration using column circuits is shown in Figure 4.6. Arbitrary Boolean functions can be implemented in the PLA by using the OR-ing and AND-ing capability of a column circuit. An array of column circuits is used as an AND-plane and an OR-plane. The *PC* signals for an AND-plane and an OR-plane are generated from the *CLK* signal and a dummy column circuit, respectively. The dummy column circuit is designed so that its output signal arrives last in an AND-plane. In the dual-rail PLA proposed in Chapter 2, a sense-amplifier activation pulse is generated from a chain of sized inverters. It is, however, difficult to make an adequate timing pulse depending on process variations and operating conditions. Moreover, designing a chain of sized inverters according to a desired timing margin makes the design complex. To reduce the design complexity and make an adequate timing pulse, we propose a programmable sense-amplifier activation scheme (PSAS). In this scheme, delay generators, which are composed of dynamic circuits with dummy cells and programmable cells, are used as

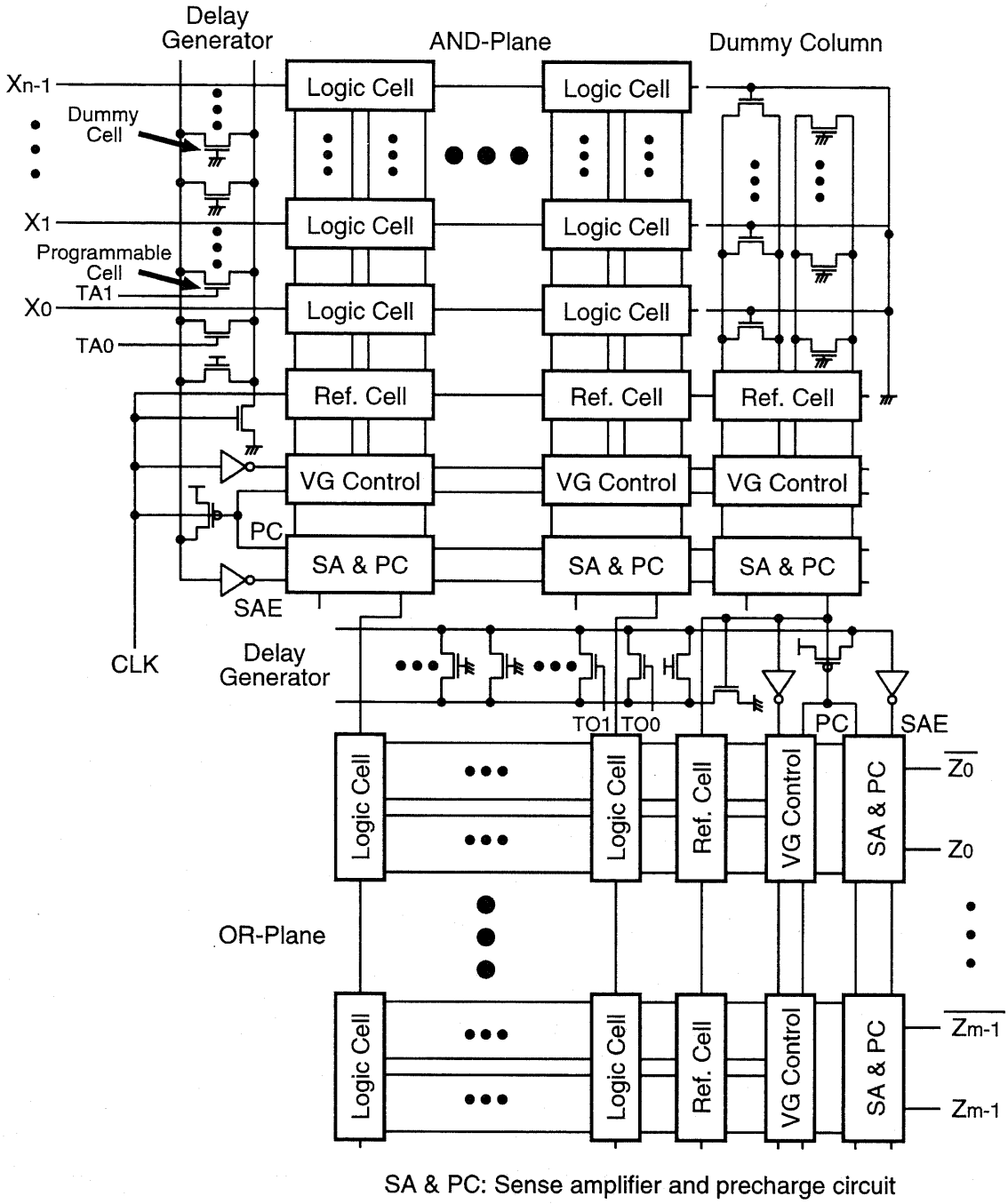


Figure 4.6 PLA configuration using the PSAS.

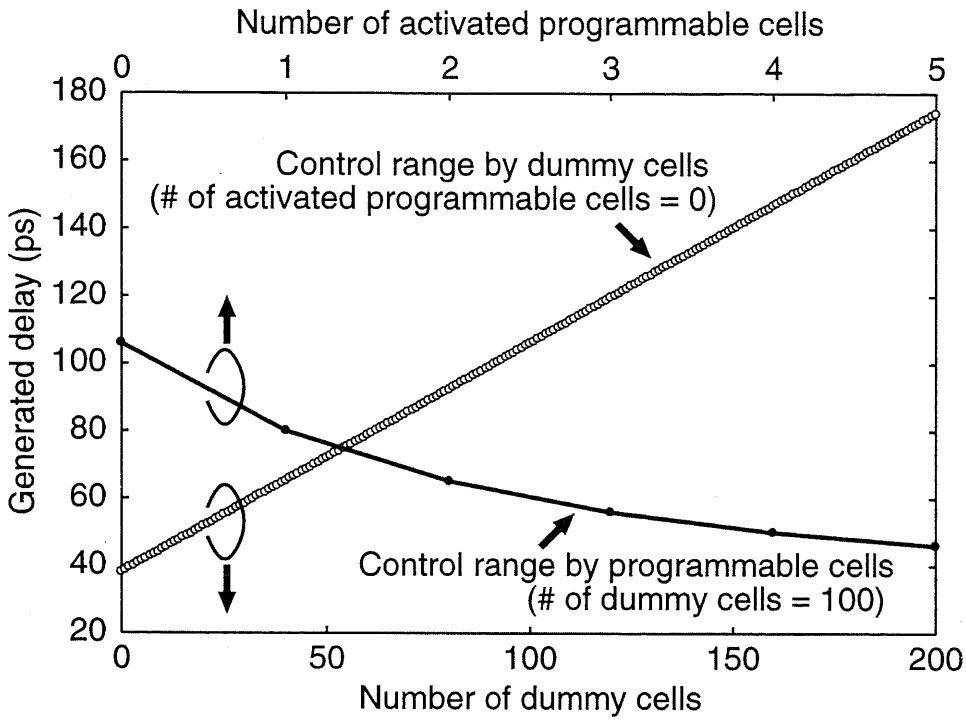


Figure 4.7 Control range of a delay generator (0.13-μm CMOS technology, 1.2-V power supply).

shown in Figure 4.6. Dummy cells and programmable cells are designed in the minimum size. The delay from the *PC* signal to the *SAE* signal depends on the number of dummy cells in a delay generator. The relationship between generated delay and the number of dummy cells is shown in Figure 4.7. The performances were obtained from post-layout simulations using a 0.13-μm CMOS technology with a supply voltage of 1.2 V. After the decision of a desired timing margin taking process variations and operating conditions into account, the timing signal can be generated by changing the number of dummy cells, and the delay increases in proportion to the number of dummy cells. Also, the delay can be controlled by programmable cells after circuit implementation. The number of programmable cells to be activated is decided by functional tests of the chip, and this enables to minimize the timing margin depending on process variations and operating conditions. The relationship between generated delay and the number of activated programmable cells

is also shown in Figure 4.7. The number of programmable cells designed in a delay generator also depends on process variations and operating conditions, and programmable cells can be easily controlled by memory circuits such as shift registers with a small number of control signals.

4.2.3 Interdigitated Column Scheme

In widely used logic functions, the circuit configuration shown in Figure 4.8, i.e., the combination of column circuits which have a small number of logic cells and a large number of logic cells is often structured, where it is assumed that the DCS is used in column circuits which have a large number of logic cells. The logic functions realized by the circuit in Figure 4.8 are expressed as

$$\overline{OUT}_0 = \overline{x}_0 \tag{4.3}$$

$$\overline{OUT}_1 = \overline{x}_0 \cdot \overline{x}_1 \tag{4.4}$$

$$\overline{OUT}_2 = \overline{x}_0 \cdot \overline{x}_1 \cdot \overline{x}_2 \tag{4.5}$$

$$\cdots$$

$$\overline{OUT}_{n-2} = \overline{x}_0 \cdot \overline{x}_1 \cdot \cdots \cdot \overline{x}_{n-3} \cdot \overline{x}_{n-2} \tag{4.6}$$

$$\overline{OUT}_{n-1} = \overline{x}_0 \cdot \overline{x}_1 \cdot \cdots \cdot \overline{x}_{n-3} \cdot \overline{x}_{n-2} \cdot \overline{x}_{n-1}. \tag{4.7}$$

This configuration is known as a structure that is often seen in arithmetic circuits [26]. This structure, however, degrades the area efficiency because the circuit area above the column circuits which have a small number of logic cells is not used. To enhance the area efficiency, we propose an interdigitated column scheme (ICS). The ICS interdigitates two column circuits into a column circuit in order to uniform the height of column circuits as shown in Figure 4.9. This scheme is not effective for all functions but for such circuits shown in Figure 4.8. Figure 4.10 shows a physical layer structure in the ICS. Additional wires for input signals, which run the circuit diagonally, and interdigitated bit-lines are realized by upper layer wires. This makes no area penalty in additional wires and bit-lines, and the higher area efficiency can be achieved. Moreover, dividing input signal

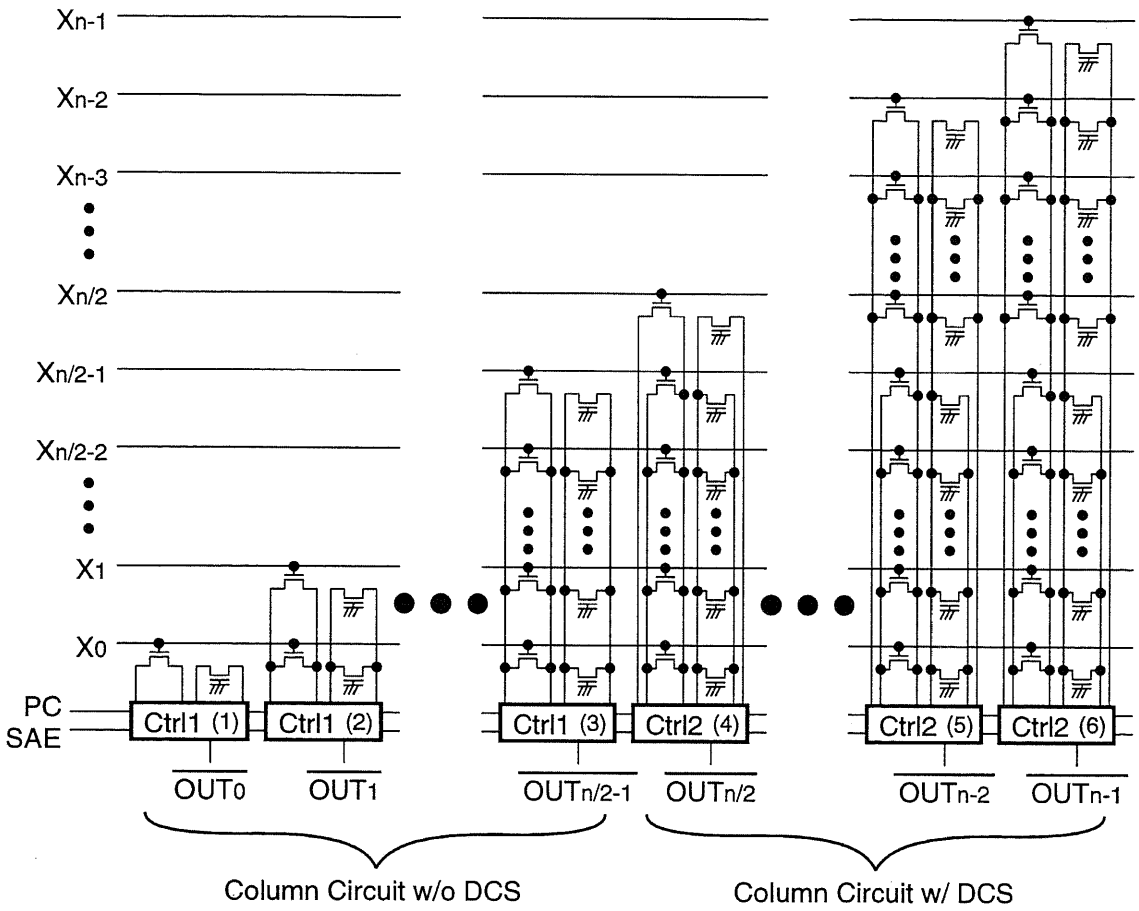


Figure 4.8 Conventional column placement.

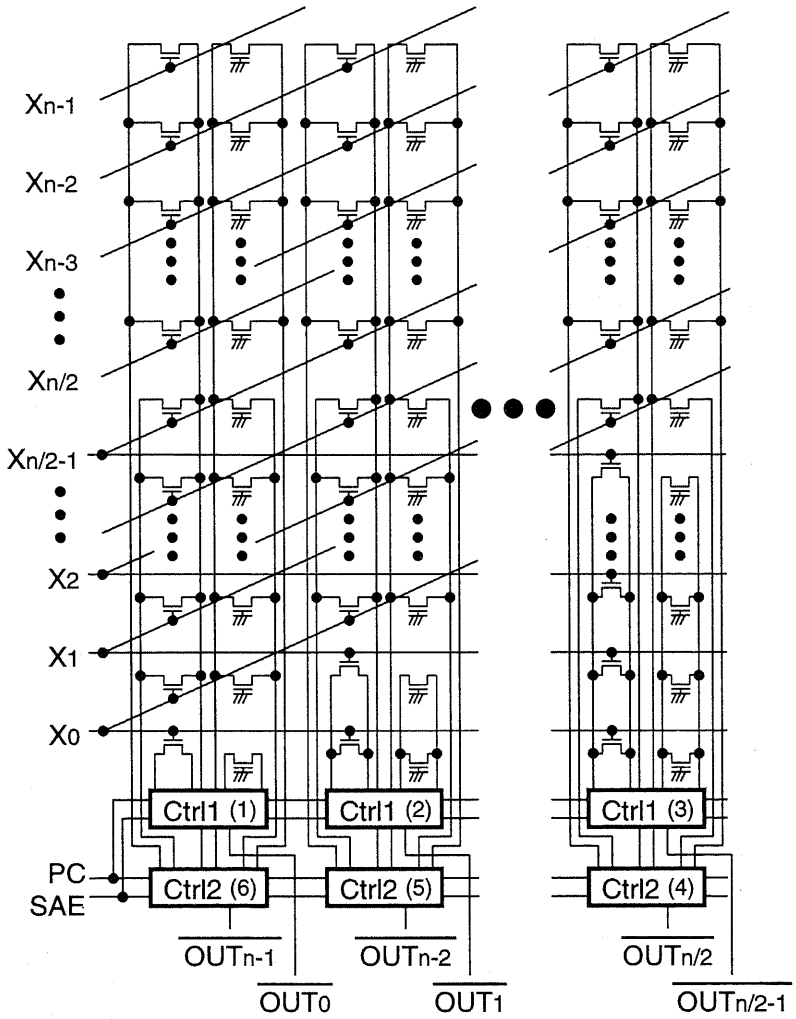


Figure 4.9 Column placement using the ICS.

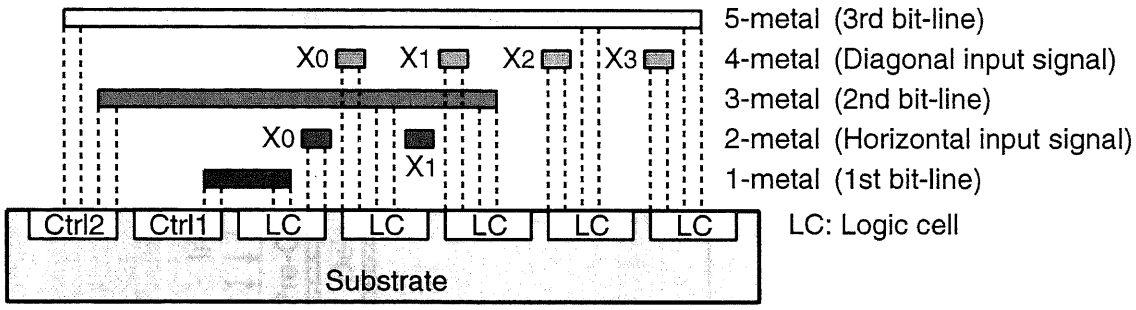


Figure 4.10 Physical layer structure in the ICS.

wires into horizontal and diagonal wires reduces wire resistances compared to the circuit configuration in Figure 4.8, thus resulting in enhancing the circuit performance. By using the ICS, the circuit area of the configuration in Figure 4.8 with a 128-bit input signal was reduced by 48.2% in a 0.13-μm CMOS technology.

4.3 Application Design

4.3.1 Common Array Structure

As applications of the proposed PLA, a comparator, a priority encoder, and an incrementor for 128-bit data processing were designed. Figure 4.11 shows the block diagram of the common array structure used in the design. All the circuits have a structure in Figure 4.11. A pre-condition circuit and a post-condition circuit are composed of static CMOS gates such as AND, NOR, and XOR gates in bit-slice structures, and are used depending on the application. These circuits help to implement a given logic function efficiently while maintaining the regularity of the circuit. The regularity of the condition circuits well suits the structure of PLA and thus the increase in design complexity is small. An AND-plane using the DCS and the ICS is used as the common logic array, i.e., it is shared among the applications. This results in reducing the design complexity significantly. An OR-plane using the DCS is used as an output logic array depending on the application.

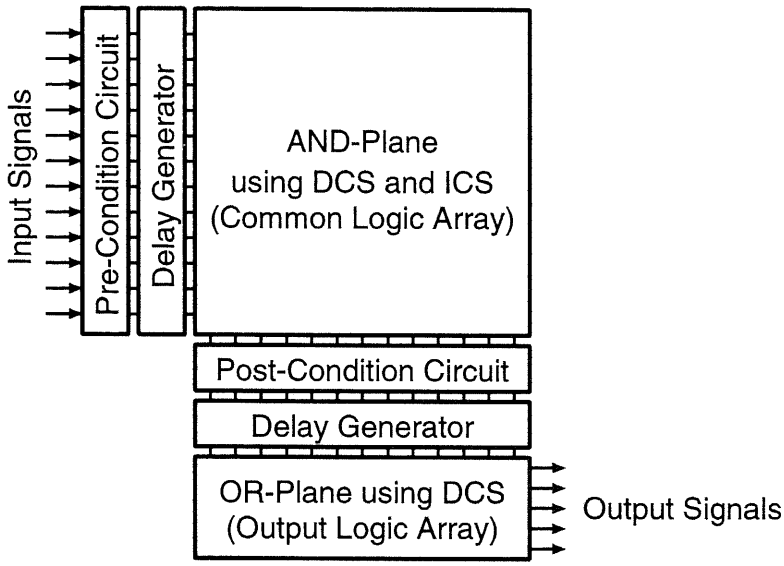


Figure 4.11 Common array structure.

4.3.2 Application Circuits

Comparator

A comparator receives two 128-bit data and compares the magnitude and equality of the data. A logic function of an equality and magnitude comparator of two 128-bit data, A and B, is expressed as

$$P_i = \overline{A_i \oplus B_i} \tag{4.8}$$

$$G_i = A_i \cdot \overline{B_i} \tag{4.9}$$

$$Z_{A=B} = P_{127} \cdot P_{126} \cdot \cdots \cdot P_2 \cdot P_1 \cdot P_0 \tag{4.10}$$

$$\begin{aligned} Z_{A>B} = {} & G_{127} \\ & + P_{127} \cdot G_{126} \\ & + P_{127} \cdot P_{126} \cdot G_{125} \\ & + \cdots \\ & + P_{127} \cdot P_{126} \cdot \cdots \cdot P_3 \cdot P_2 \cdot G_1 \\ & + P_{127} \cdot P_{126} \cdot \cdots \cdot P_2 \cdot P_1 \cdot G_0 \end{aligned} \tag{4.11}$$

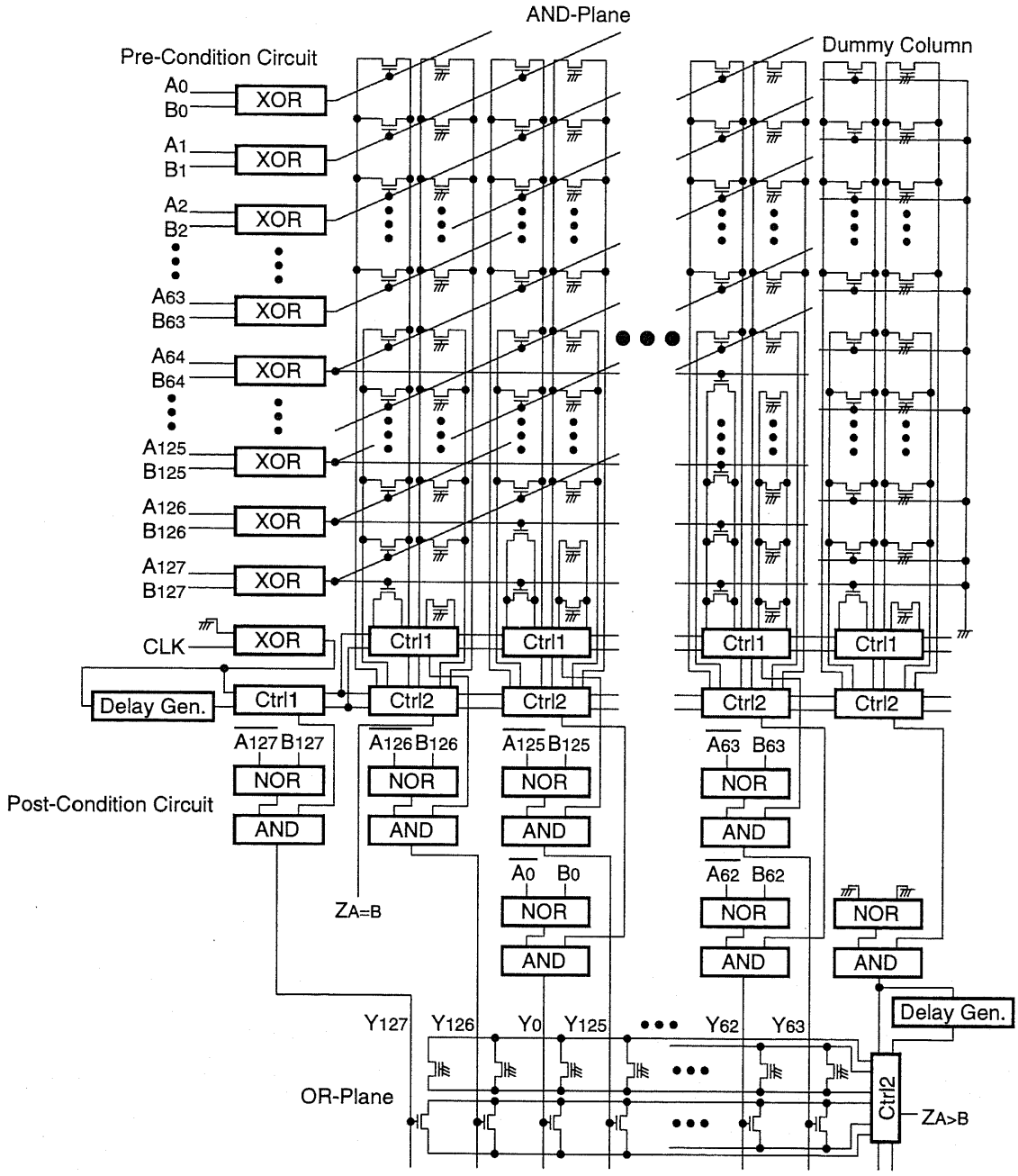


Figure 4.12 Schematic diagram of the proposed 128-bit comparator.

where $Z_{A=B}$ becomes high when $A = B$, while $Z_{A>B}$ becomes high when $A > B$. The schematic diagram of this implementation is shown in Figure 4.12. The logic function $Z_{A=B}$, which is expressed as a product form, is implemented in an AND-plane using the DCS with the propagate signal P. Similarly, the logic function $Z_{A>B}$, which is expressed as a sum-of-products form, is implemented in an AND-plane using the DCS and the ICS, and an OR-plane using the DCS with the propagate and generate signals, P and G. The P and G signals are generated from static 2-input XOR and NOR gates, which are used as a pre-condition circuit and a post-condition circuit, and significantly reduce the overall area of the circuit compared to the direct implementation without the P and G signals. Also, static 2-input AND gates are used as a post-condition circuit to obtain the intermediate result (Y).

Priority Encoder

A priority encoder receives a 128-bit data and detects the highest priority nonzero bit. An intermediate logic function of a priority encoder of a 128-bit data, A, is expressed as

$$\begin{aligned} Y_i &= \left(\overline{A_0 + A_1 + A_2 + \cdots + A_{i-1} + A_i}\right) \\ &\quad \cdot (A_0 + A_1 + A_2 + \cdots + A_i + A_{i+1}) \\ \left(Y_{127}\right. &= \left.\left(\overline{A_0 + A_1 + A_2 + \cdots + A_{126} + A_{127}}\right)\right) \end{aligned} \tag{4.12}$$

where Y corresponds to the position of the highest priority nonzero bit, while Y_{127} becomes high when the input data bits are all zeros. Finally, Y is encoded into an 8-bit output (Z). The schematic diagram of this implementation is shown in Figure 4.13, where the same AND-plane as that of the comparator is used and shared with the other applications. Since a column circuit is formed in a dual-rail structure and generates both true and complement results, the NOR and OR results in Eq. (4.12) are available from the i-th column and the $(i + 1)$-th column, respectively. Static 2-input AND gates are used as a post-condition circuit to obtain the intermediate result (Y).

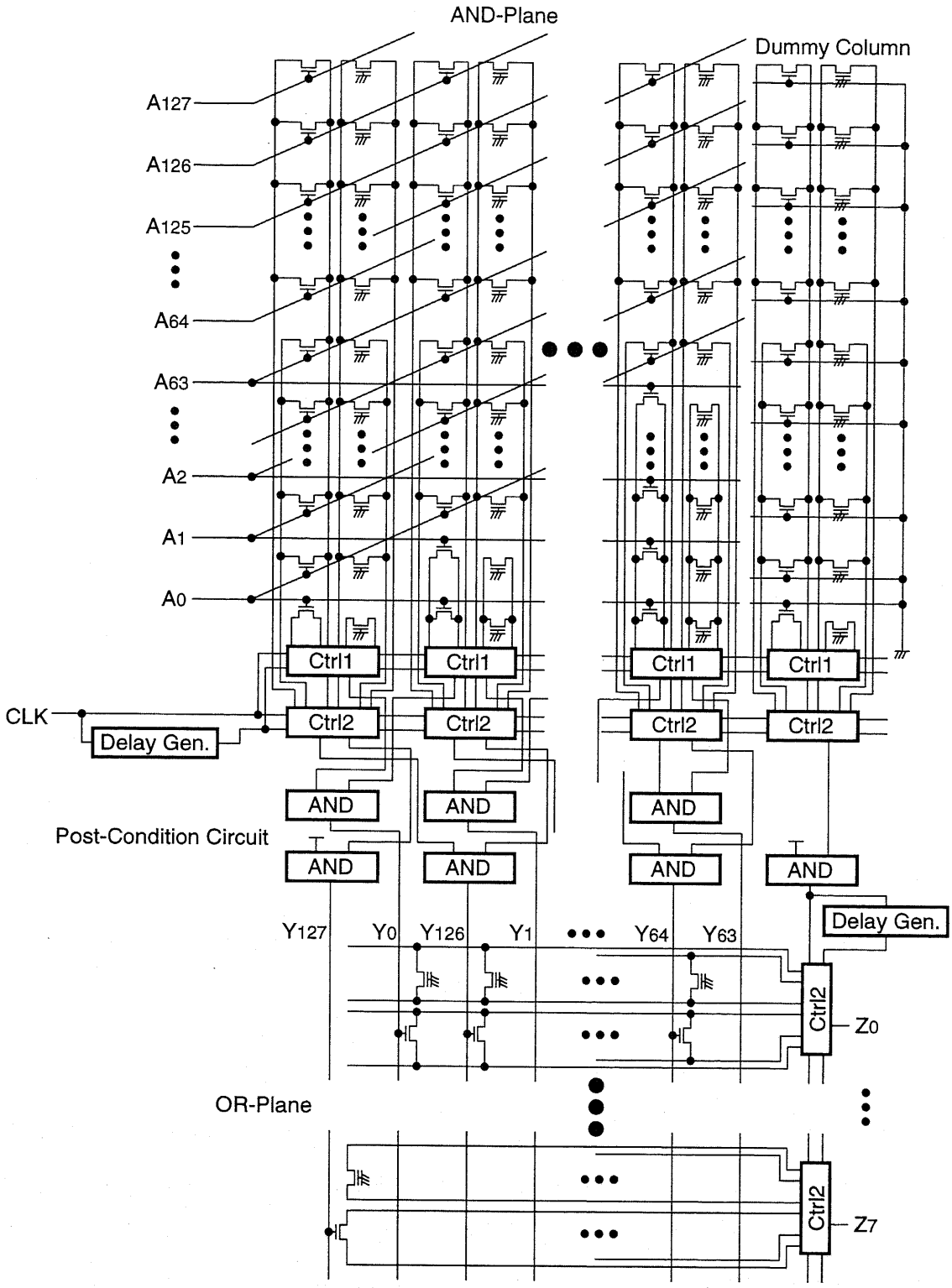


Figure 4.13 Schematic diagram of the proposed 128-bit priority encoder.

Incrementor

An incrementor receives a 128-bit data and increments the data by one. A logic function of an incrementor of a 128-bit data, A, is expressed as

$$Y_i = A_i \oplus (A_{i+1} \cdot A_{i+2} \cdot A_{i+3} \cdot \cdots \cdot A_{126} \cdot A_{127}). \tag{4.13}$$

The schematic diagram of this implementation is shown in Figure 4.14, where the same AND-plane as that of the comparator is used and shared with the other applications. Static 2-input XOR gates are used as a post-condition circuit to obtain the final result (Y).

4.4 Simulation Results

The application circuits were designed using a 0.13-μm, 5-metal-layer CMOS technology with a supply voltage of 1.2 V. Figure 4.15 and Table 4.1 show the designed macro layouts and a comparison between different implementation styles, respectively. The number of dummy cells in a delay generator for an AND-plane is 170, and sense amplifiers in the PLA are activated when the developed voltage difference between bit-lines becomes larger than the designed sense voltage of 120 mV. Also, five programmable cells were designed and can reduce the sense-amplifier activation delay by up to 60 ps from the designed delay depending on process variations and operating conditions. The random logic circuits were designed using standard cells, and are based on the above-mentioned logic functions and a constraint that minimizes the circuit delay. The proposed PLA in Chapter 2 were designed with pre- and post-condition circuits shown in Figure 4.11. The performances were obtained from post-layout simulations using HSPICE. The results show that the proposed circuit achieves delay reductions by 38.1% and 22.2% on average over the random logic circuit and the proposed PLA in Chapter 2, respectively. Figure 4.16 shows simulated waveforms of the proposed 128-bit comparator.

The power consumption of the proposed circuit is larger than that of the proposed PLA in Chapter 2. This is because of duplicated sense amplifiers in the DCS. However, as shown in Figure 4.5, the increase in power consumption becomes small as the number

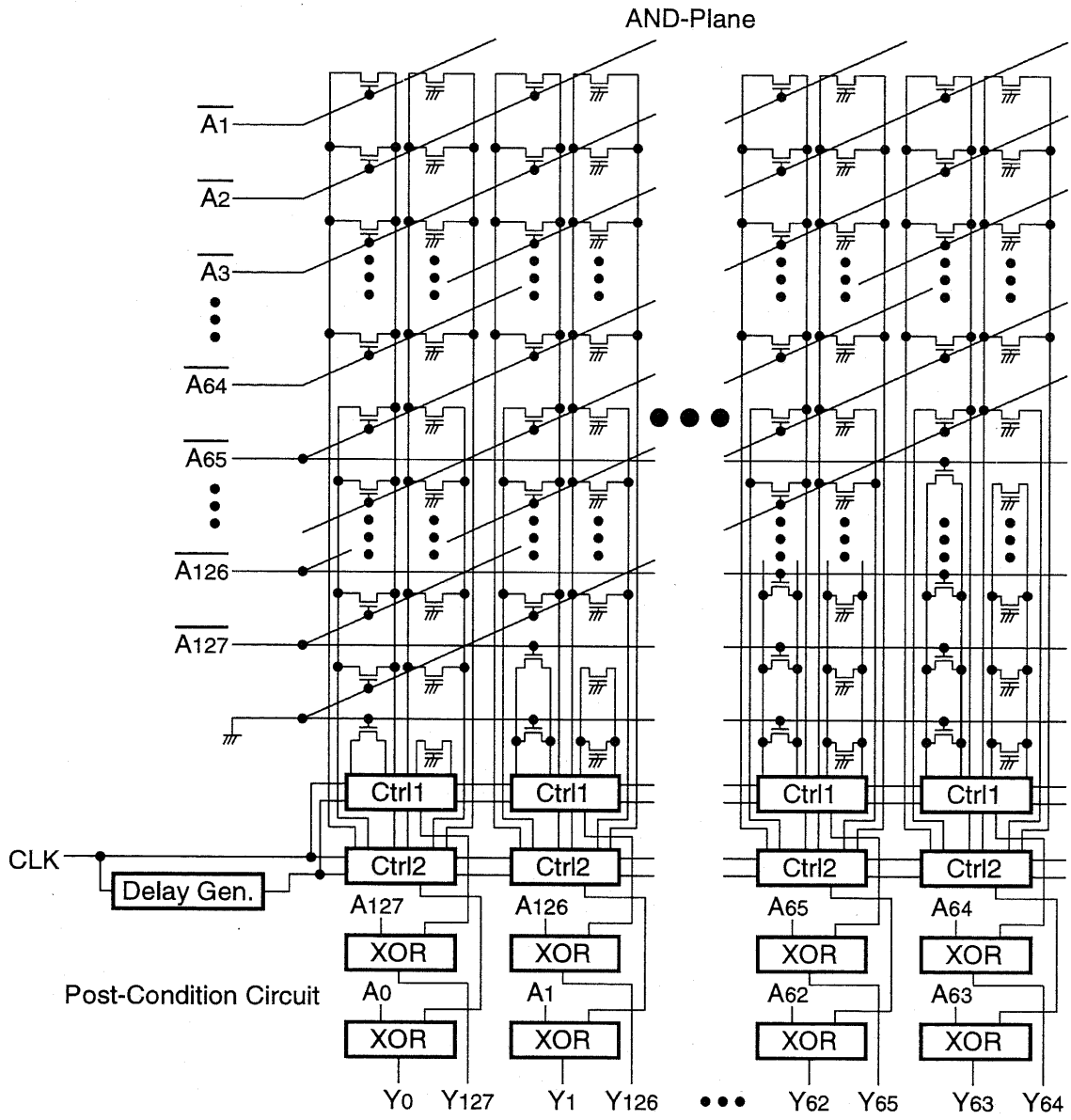


Figure 4.14 Schematic diagram of the proposed 128-bit incrementor.

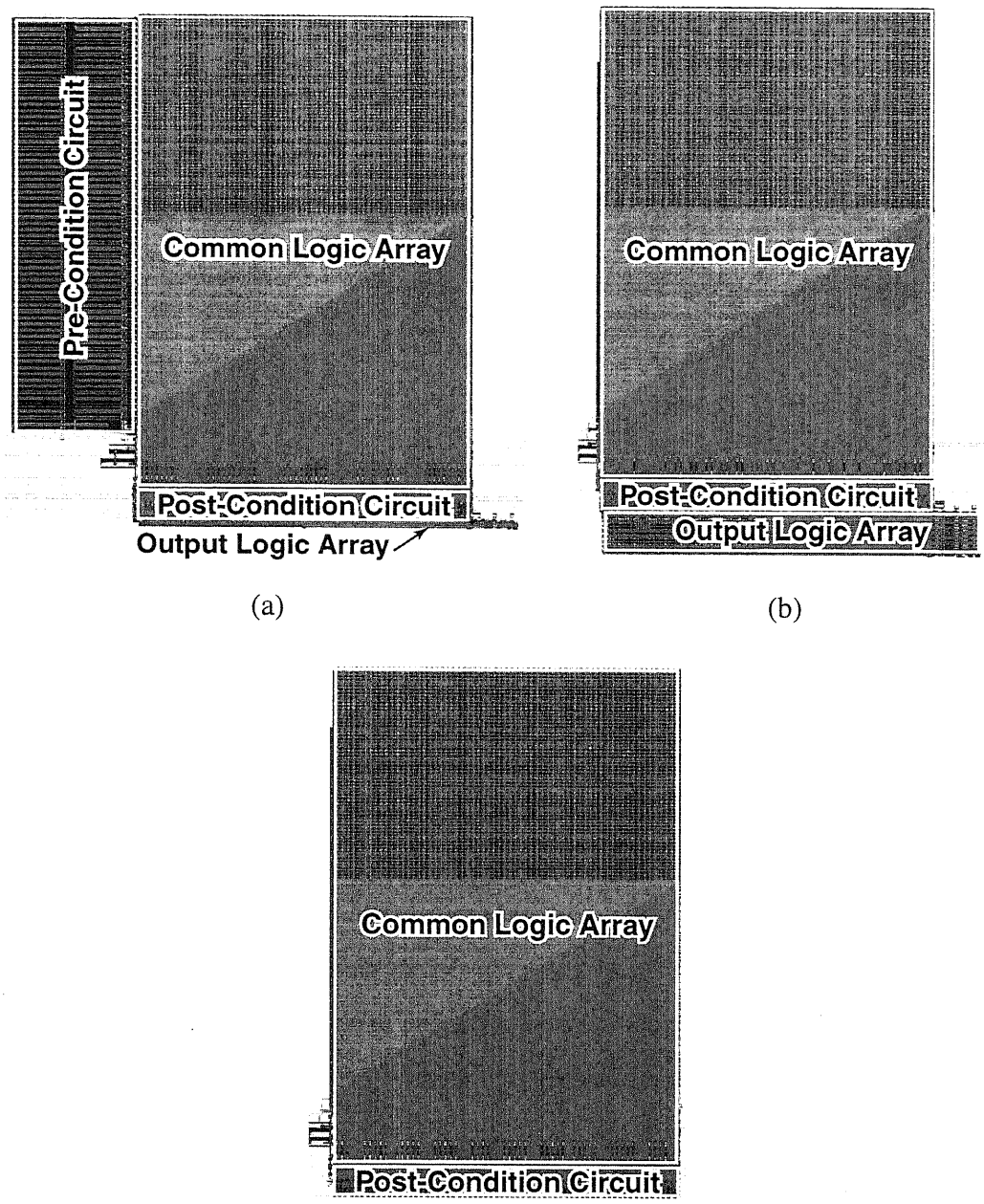


Figure 4.15 Designed macro layouts. (a) Comparator, (b) priority encoder, and (c) incrementor.

Table 4.1 Comparison between different implementation styles.

Circuit (0.13-μm, 1.2-V V_{DD}, 128-bit)	Delay (ps)	Power (mW) @ 1.25 GHz	PD product (normalized)	Area (μm^2)
Comparator				
Random logic	$Z_{A=B}$: 521, $Z_{A>B}$: 551	53.8	1.00	15,750
Proposed PLA in Chapter 2	$Z_{A=B}$: 295, $Z_{A>B}$: 635	30.0	0.484	101,797
This work	$Z_{A=B}$: 234, $Z_{A>B}$: 503	35.3	0.451	71,815
Priority encoder				
Random logic	617	29.9	1.00	14,396
Proposed PLA in Chapter 2	550	18.5	0.552	97,054
This work	432	22.3	0.522	59,328
Incrementor				
Random logic	420	24.7	1.00	13,110
Proposed PLA in Chapter 2	235	17.7	0.401	84,104
This work	174	22.1	0.371	46,992

of input signals increases because a sense amplifier consumes almost the same power even in larger circuits. On the other hand, the power consumption is smaller than that of the random logic circuit. This is mainly because of reduced voltage swings of bit-lines and a small number of short-circuit current paths. In the PLA, short-circuit current is not consumed in logic cells which occupy a large part of the circuit. Also, the proposed circuits have advantages in terms of operating energy. The power-delay (PD) product can be reduced by 55.2% and 6.5% on average over the random logic circuit and the proposed PLA in Chapter 2, respectively.

In our approach, a dual-rail structure and an array structure are utilized, thus requiring larger circuit area compared to random logic implementations. On the other hand, it is possible to improve the common-mode noise immunity by using a dual-rail structure. Moreover, an array structure reduces the iteration of the design process because structured arrays give predictable area, delay, and power consumption early in the design process. These features are becoming important to realize noise-immune systems and

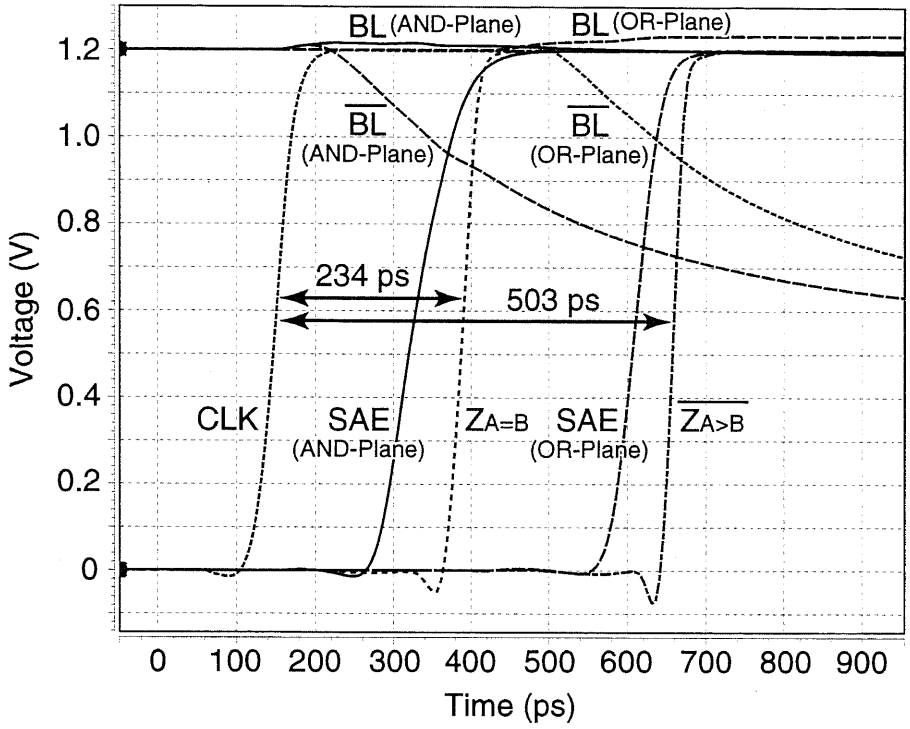


Figure 4.16 Simulated waveforms of the designed 128-bit comparator.

to meet design cost and time-to-market goals in deep sub-micron IC designs, while chip area cost is becoming small due to the advances in integration [1]. By using the ICS, the proposed circuit achieves an area reduction by 37.5% on average over the proposed PLA in Chapter 2.

4.5 Summary

In this chapter, high-speed and area-efficient design techniques for dual-rail array logic architectures were presented. As applications of the proposed architectures, a comparator, a priority encoder, and an incrementor for 128-bit data processing were designed. The proposed circuit design schemes achieved a 22.2% delay reduction and a 37.5% area reduction on average over the proposed PLA in Chapter 2 in a 0.13-μm CMOS technology. The high-speed capability of the proposed schemes can be enhanced in high data

bandwidth systems. Moreover, a lot of circuit elements can be shared among different macro designs, thus reducing the design complexity significantly. Also, by using the proposed schemes, an activation pulse for sense amplifiers in the circuit can be optimized depending on process variations and operating conditions.

Chapter 5

Extension to Multiple-Threshold Logic

5.1 Introduction

The threshold logic circuits in the previous chapters are used to detect the input signals activated to high. When at least one of the input signals is high, the output signal of the circuit becomes high and thus a logical OR operation for a PLA is realized as well as a logical AND operation by using the complement input signals. This is considered that the threshold logic circuit has a threshold value of 0.5 with respect to the number of input signals activated to high. Although this function is based on a threshold voltage generated from a reference cell, it is further possible to generate multiple threshold voltages, such as 1.5, 2.5, 3.5, and so forth, by using multiple reference cells.

In this chapter, we propose new array logic architectures using dual-rail threshold logic circuits and methods of logic synthesis for their architectures. The threshold logic circuits realize multiple-threshold logic functions which have a weight of 1 in each input signal and are formed in a two-level network. The design using multiple-threshold logic functions is known as a methodology that typically offers the capability of realizing complex Boolean functions using a smaller number of logic gates and/or fewer logic stages [39], [40]. The proposed architectures are designed with multiple-threshold logic functions to reduce the overall area of the circuit. Also, since the proposed architectures

can be optimized in column-circuit level, it is possible to build the proposed circuit into LCPLA proposed in Chapter 3 and the conventional area-efficient design techniques [5], [41]. Thus, further reductions of the number of product terms can be achieved. Moreover, as an application of the threshold logic circuit, a new functional memory with the high-speed flexible data search capability is also presented.

5.2 Proposed Architecture

Figure 5.1 shows the concept of the proposed array logic architecture. The architecture is composed of a two-level network of threshold elements denoted by circles in the figure with a threshold value T, and thus the threshold elements are formed in a PLA structure. The x and z signals are the binary inputs and outputs of the architecture, respectively. Each threshold element can have a different threshold value and is designed so that the overall area of the architecture can be minimized. The conventional PLA, which is composed of an AND-plane and an OR-plane, can be considered as a PLA that has a threshold value of 0.5 in the network of Figure 5.1.

5.3 Proposed Circuits

5.3.1 Column Circuit

Figure 5.2 shows the proposed threshold logic circuit, which is used as a threshold element in Figure 5.1. The circuit has a dual-rail configuration and consists of a stack of logic cells, a reference cell, a virtual ground (VG) controller, a precharge and equalization circuit, and a sense amplifier. A threshold logic function and its negation of the input signals (x_0–x_{n-1}) can be obtained from the output signals, OUT and $\overline{OUT}$, respectively, and the function depends on the pattern of embedded logic cells. By using a sense amplifier, the output signals are activated by sensing the differential voltage between the bit-lines, BL and $\overline{\mathrm{BL}}$. The VG controller is used to expand the voltage difference between the bit-lines.

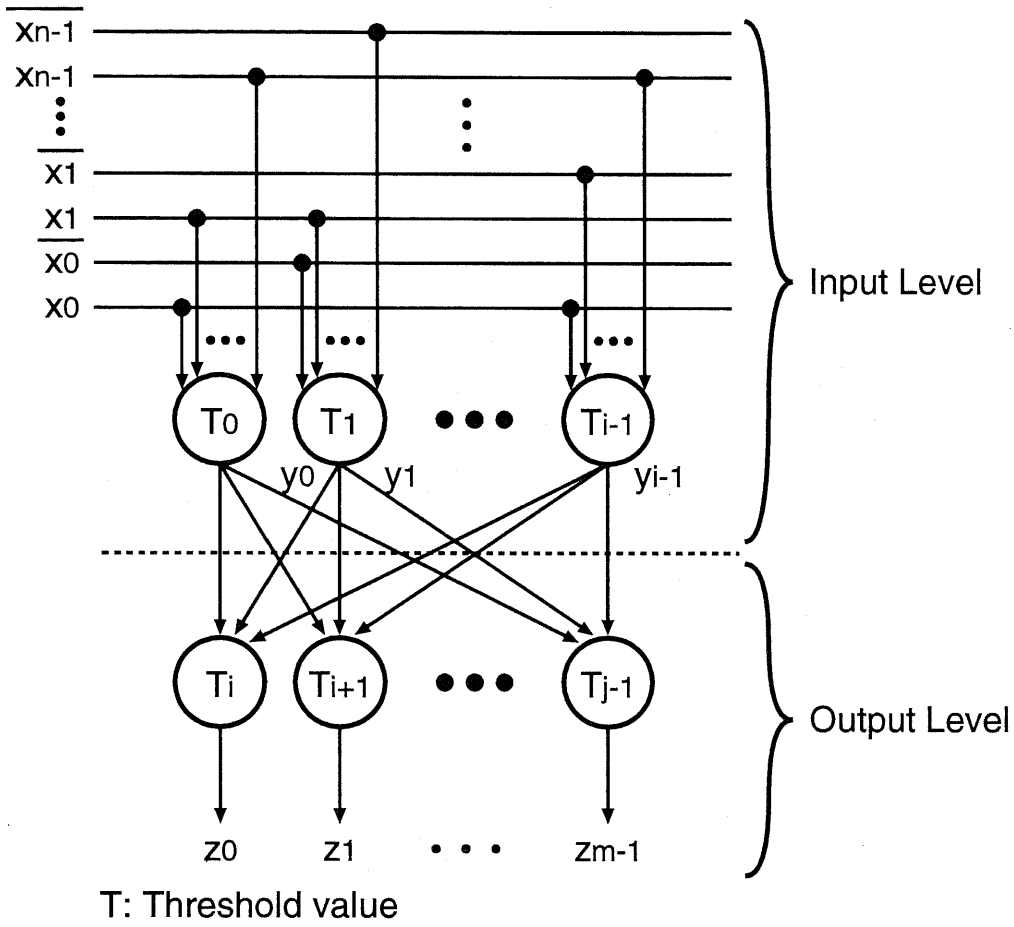


Figure 5.1 Concept of the proposed array logic architecture.

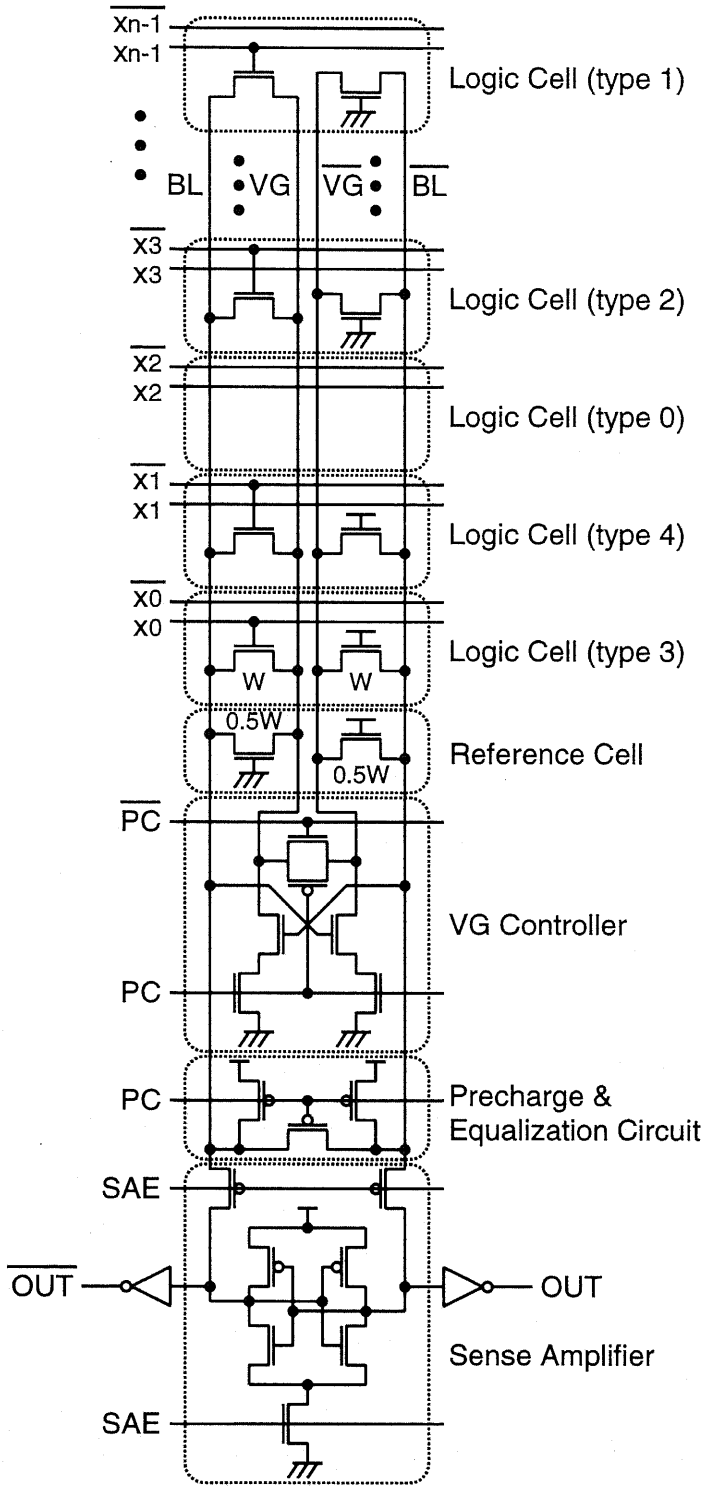


Figure 5.2 Proposed threshold logic circuit.

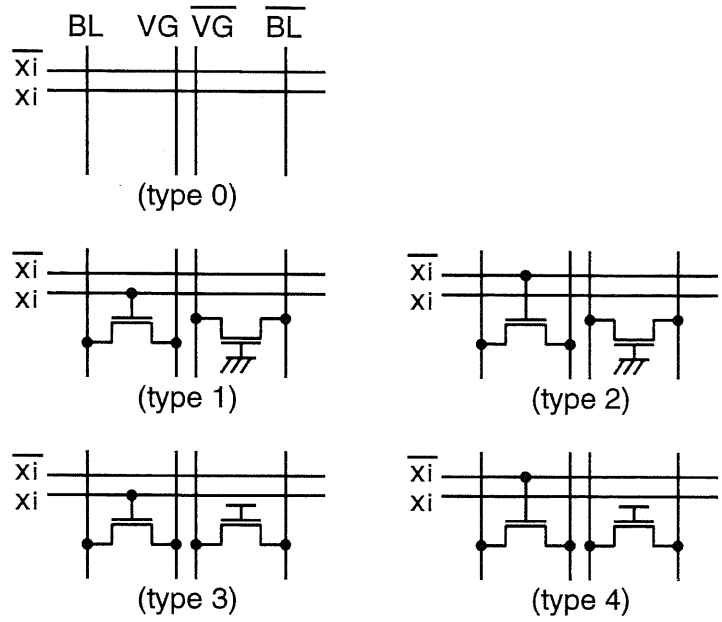


Figure 5.3 Logic cells.

In the circuit, five types of logic cells shown in Figure 5.3 can be embedded to realize arbitrary threshold logic functions which have a weight of 1 in each input signal. Each logic cell has a pair of NMOS devices, except for type 0. One is used to pull down the bit-line (BL) depending on the input signal (x_i) or its negation ($\overline{x_i}$). The other is used to pull down the reference bit-line ($\overline{\text{BL}}$) or to balance the load capacitances and leakage current of BL and $\overline{\text{BL}}$. This symmetrical structure of the circuit results in a robust operation, especially with respect to common-mode noise. By using the logic cells, the output signal *OUT* can be expressed as

$$OUT = \begin{cases} 1 & if \sum_{i=0}^{n-1}(x_i p_i + \overline{x}_i q_i) \leq T \\ 0 & otherwise \end{cases} \tag{5.1}$$

$$p_i = \begin{cases} 1 & if\ x_i \text{ is connected to a logic cell of type 1 or type 3} \\ 0 & otherwise \end{cases} \tag{5.2}$$

$$q_i = \begin{cases} 1 & if\ \overline{x}_i \text{ is connected to a logic cell of type 2 or type 4} \\ 0 & otherwise \end{cases} \tag{5.3}$$

$$T = N_3 + N_4 + 0.5 \tag{5.4}$$

where the terms in the condition of Eq.(5.1) represent real numbers, and T is a threshold value. N_3 and N_4 are the numbers of logic cells of type 3 and type 4 in a threshold logic circuit, respectively.

On the other hand, the logic function of a threshold logic circuit is expressed as a *unate function* [29], where the number of literals in a product term depends on T. For example, when four input signals, x_0, $\overline{x}_1$, x_2, $\overline{x}_3$, are connected to logic cells, possible Boolean functions which can be obtained from OUT are as follows.

$$\begin{aligned} f_0(x_0, x_1, x_2, x_3) &= \overline{x}_0 x_1 \overline{x}_2 x_3 \\ &= t^4_{0.5}(x_0, \overline{x}_1, x_2, \overline{x}_3) \qquad (T = 0.5) \end{aligned} \tag{5.5}$$

$$\begin{aligned} f_1(x_0, x_1, x_2, x_3) &= \overline{x}_0 x_1 \overline{x}_2 + \overline{x}_0 x_1 x_3 + \overline{x}_0 \overline{x}_2 x_3 + x_1 \overline{x}_2 x_3 \\ &= t^4_{1.5}(x_0, \overline{x}_1, x_2, \overline{x}_3) \qquad (T = 1.5) \end{aligned} \tag{5.6}$$

$$\begin{aligned} f_2(x_0, x_1, x_2, x_3) &= \overline{x}_0 x_1 + \overline{x}_0 \overline{x}_2 + \overline{x}_0 x_3 + x_1 \overline{x}_2 + x_1 x_3 + \overline{x}_2 x_3 \\ &= t^4_{2.5}(x_0, \overline{x}_1, x_2, \overline{x}_3) \qquad (T = 2.5) \end{aligned} \tag{5.7}$$

$$\begin{aligned} f_3(x_0, x_1, x_2, x_3) &= \overline{x}_0 + x_1 + \overline{x}_2 + x_3 \\ &= t^4_{3.5}(x_0, \overline{x}_1, x_2, \overline{x}_3) \qquad (T = 3.5) \end{aligned} \tag{5.8}$$

where $t^i_j(\boldsymbol{x})$ is the definition of a threshold logic function. $\boldsymbol{x}$ is a set of input signals connected to logic cells, i is the number of elements in $\boldsymbol{x}$, and j is a threshold value T. Thus, there are k logic functions when connecting k input signals to logic cells, and the number of product terms (${}_iC_{i-j+0.5}$) of each function can be reduced to 1 by using a threshold logic circuit. Also, the functions have the following characteristics.

$$f_2 = f_1^d = \overline{f}_1(\overline{x}_0, \overline{x}_1, \overline{x}_2, \overline{x}_3) \tag{5.9}$$

$$f_3 = f_0^d = \overline{f}_0(\overline{x}_0, \overline{x}_1, \overline{x}_2, \overline{x}_3). \tag{5.10}$$

In general, this can be expressed as

$$f_{n-i-1} = f_i^d = \overline{f}_i(\overline{x}_0, \overline{x}_1, \cdots, \overline{x}_{n-1}) \quad (i = 0, 1, \cdots, n-1). \tag{5.11}$$

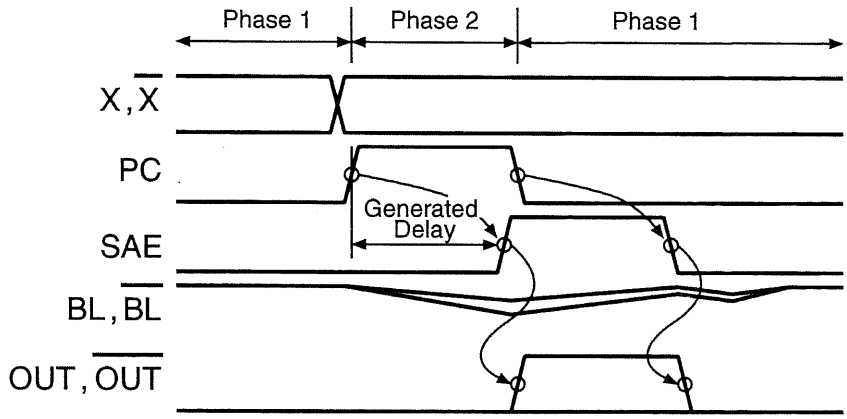


Figure 5.4 Timing diagram of control, input, and output signals, and bit-line potentials in a threshold logic circuit.

Note that the negations of the input signals ($\overline{x}_0$–$\overline{x}_{n-1}$) and the output signal ($\overline{OUT}$) are available from the circuit in Figure 5.2. This means that f_2 and f_3 can be obtained from f_1 and f_0, respectively. Thus, the maximum number of logic cells of type 3 and type 4 can be limited to

$$\max(N_3 + N_4) = \begin{cases} \frac{n}{2} - 1 & if\ n \text{ is an even number} \\ \frac{n-1}{2} & if\ n \text{ is an odd number.} \end{cases} \tag{5.12}$$

This helps to reduce the voltage swing of $\overline{BL}$ and thus results in low-power operations.

Figure 5.4 shows a timing diagram of control, input, and output signals, and bit-line potentials in a threshold logic circuit. The circuit operates in two phases: phase 1 and phase 2. In phase 1, the *PC* signal is low. Thus, the bit-lines are precharged high and equalized. When the *PC* signal becomes high, the circuit enters phase 2.

In phase 2, BL and $\overline{BL}$ are pulled down by logic cells and a reference cell depending on the input signals. The pull-down speed depends on the number of logic-cell current paths in BL and $\overline{BL}$. The device size of a logic cell is W, where W is the channel width of a transistor. On the other hand, that of a reference cell is $0.5W$ as shown in Figure 5.2. This half-size device is provided to avoid the meta-stable condition, which may be caused when the number of logic-cell current paths in BL is equal to that in $\overline{BL}$, and corresponds

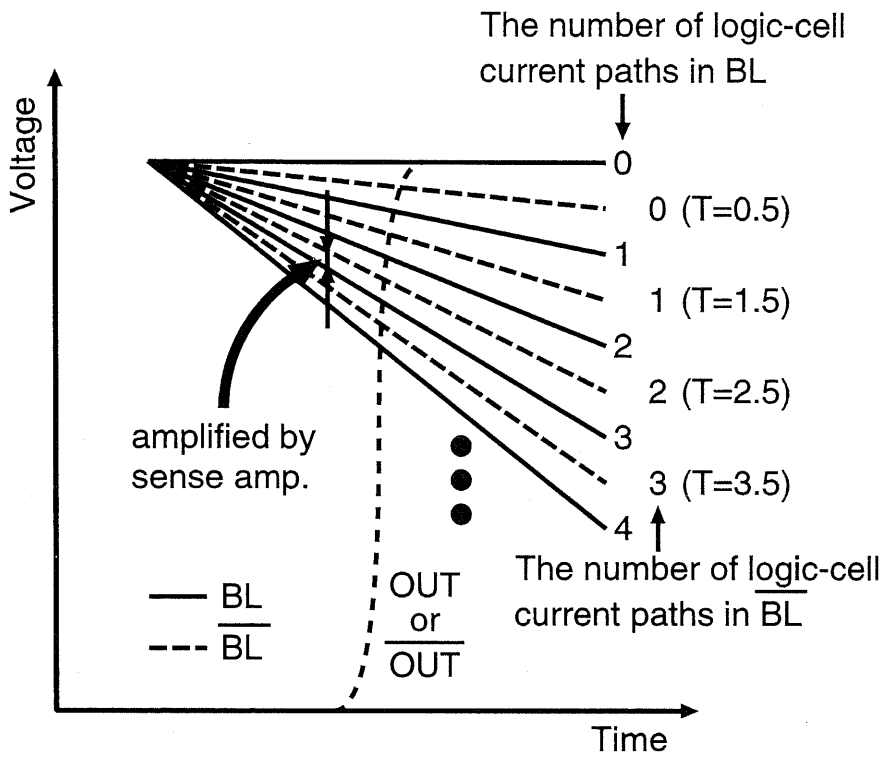


Figure 5.5 Simplified transient relationships between the bit-lines (BL and $\overline{\text{BL}}$).

to a threshold value of 0.5. Figure 5.5 shows simplified transient relationships between the bit-lines when changing the number of logic-cell current paths in BL and $\overline{\text{BL}}$. As can be seen from this figure, the voltage potential of $\overline{\text{BL}}$ is used as a threshold value T, and the output signal OUT becomes high when the voltage potential of BL is higher than that of $\overline{\text{BL}}$. Note that the bit-line which has a higher voltage potential than the other bit-line never reaches ground level because of the VG controller. This helps to expand the voltage difference between the bit-lines. The *SAE* signal is activated when the developed voltage difference between the bit-lines becomes larger than the designed sense voltage, which takes the worst case of considerable noise margin and process variations into account. By activating the sense amplifier, one of the output signals, OUT or $\overline{OUT}$, becomes high depending on the developed voltage difference.

5.3.2 PLA Configuration

The PLA configuration using threshold logic circuits is shown in Figure 5.6, where the configuration shown in Chapter 4 is used. An array of threshold logic circuits is used as an input-level plane and an output-level plane. The *PC* signals for the input-level plane and the output-level plane are generated from the *CLK* signal and a dummy column circuit, respectively. The dummy column circuit is designed so that its output signal arrives last in the input-level plane. The *SAE* signal which activates sense amplifiers is generated from a delay generator. The delay generator is composed of dynamic circuits with dummy cells and programmable cells as shown in Figure 5.6. Dummy cells and programmable cells are designed in the minimum size. The delay from the *PC* signal to the *SAE* signal depends on the number of dummy cells in a delay generator. The relationship between generated delay and the number of dummy cells is shown in Figure 5.7. The performances were obtained from post-layout simulations using a 0.35-μm CMOS technology with a supply voltage of 3.3 V. After the decision of a desired timing margin taking process variations and operating conditions into account, the timing signal can be generated by changing the number of dummy cells, and the delay increases in proportion to the number of dummy cells. Also, the delay can be controlled by programmable cells after circuit implementation. The number of programmable cells to be activated is decided by functional tests of the chip, and this enables to minimize the timing margin depending on process variations and operating conditions. The relationship between generated delay and the number of activated programmable cells is also shown in Figure 5.7. The number of programmable cells designed in a delay generator also depends on process variations and operating conditions, and programmable cells can be easily controlled by memory circuits such as shift registers with a small number of control signals.

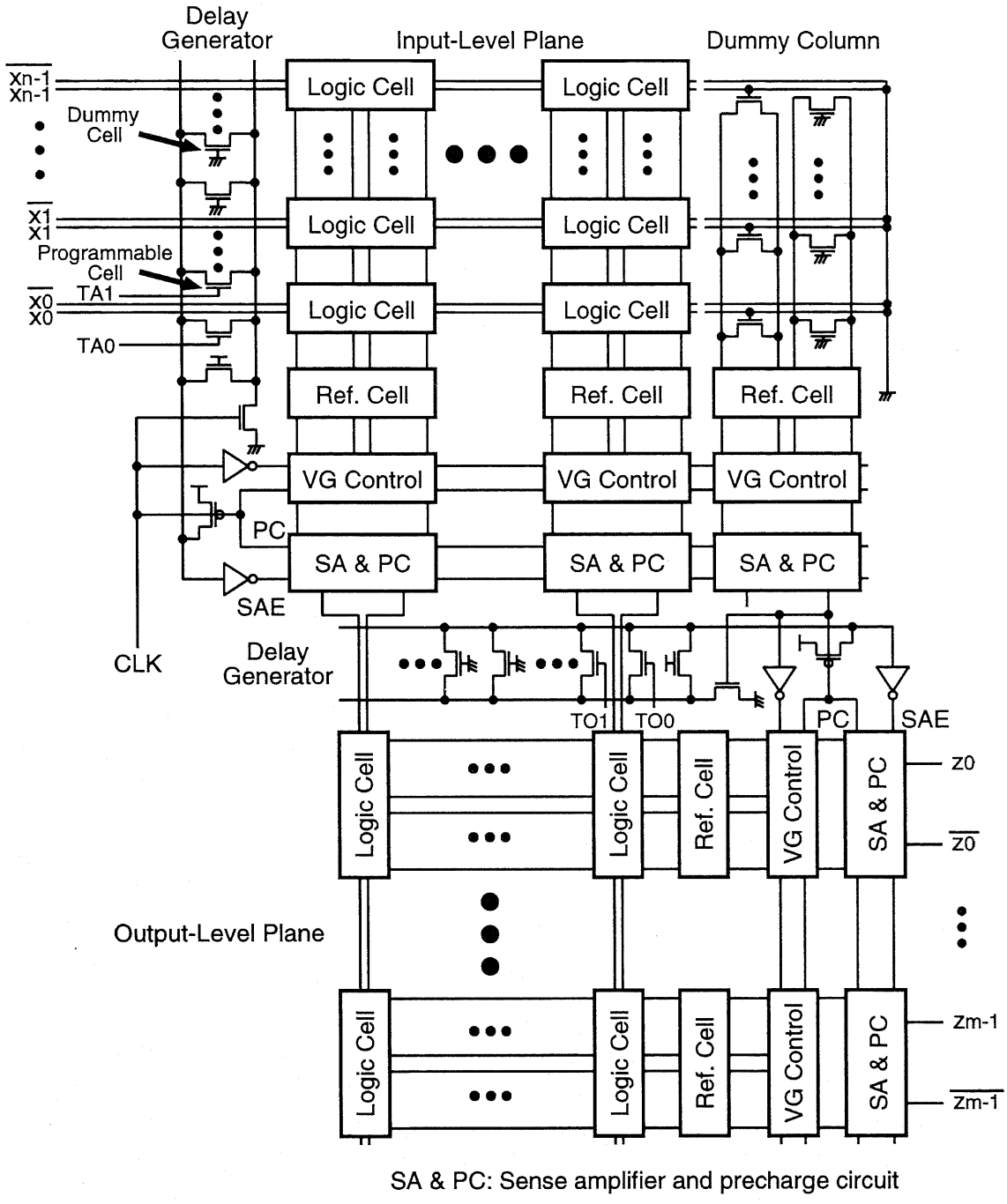


Figure 5.6 PLA configuration using threshold logic circuits.

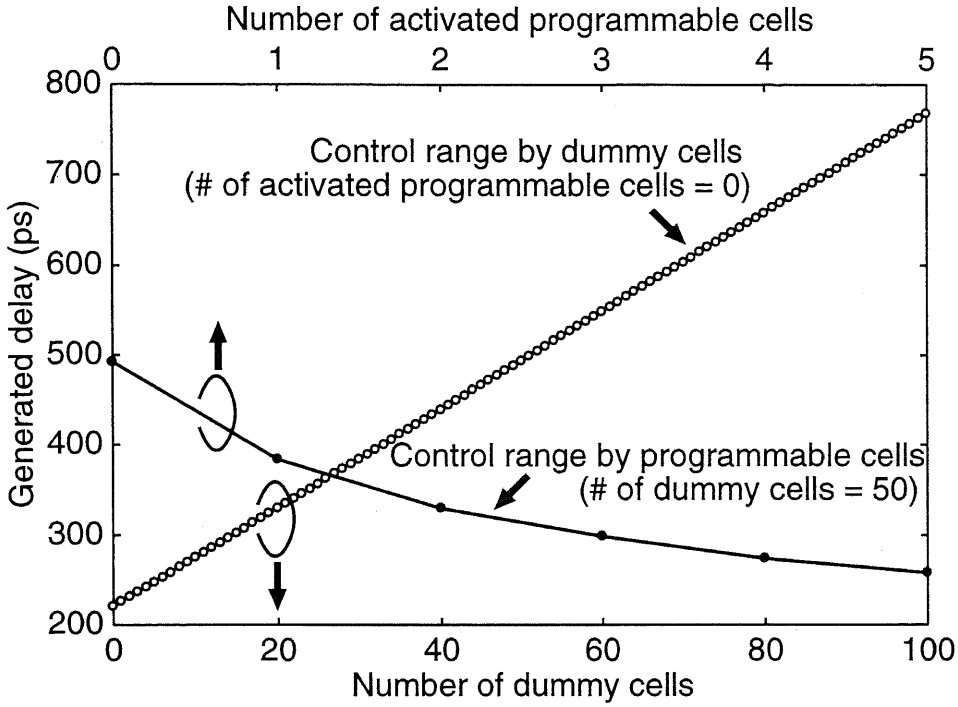


Figure 5.7 Control range of a delay generator (0.35-μm CMOS technology, 3.3-V power supply).

5.4 Design Methodology

5.4.1 Logic Synthesis for a PLA Using Threshold Logic Circuits

In order to confirm the effectiveness of the proposed threshold logic circuits, we propose a method of logic synthesis for the proposed architecture. To find threshold logic functions in sum-of-products forms, we consider threshold logic circuits which have arbitrary threshold values and a threshold value of 0.5 in an input-level plane and an output-level plane, respectively. Thus, the output-level plane can be viewed as an OR-plane in the network.

The flow and the procedure of logic synthesis are shown in Figure 5.8 and Figure 5.9, respectively, where we define a threshold logic function which can be realized by the circuit in Figure 5.2 as a *threshold term*. The procedure mainly consists of three steps:

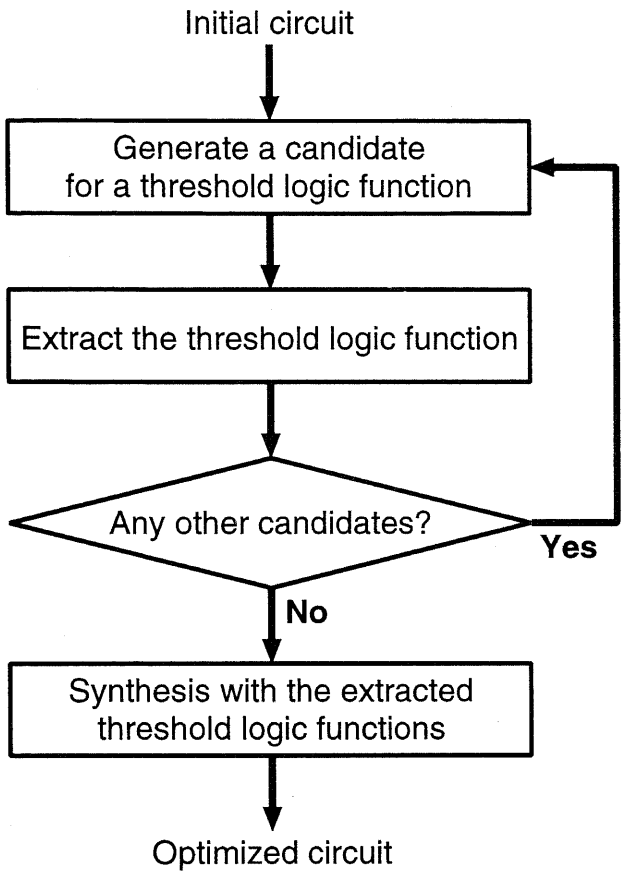


Figure 5.8 Flow of logic synthesis for a PLA using threshold logic circuits.

1) generation of candidates for threshold terms, 2) extraction of threshold terms from a given logic function, and 3) synthesis with the extracted threshold terms. In step 1, all possible threshold terms are generated, but product terms, i.e., threshold logic functions which have a threshold value of 0.5 are not included in the candidates. This is because product terms do not contribute to reductions of circuit area. In step 2, tautology checking of the function

$$h = f + \overline{g} \tag{5.13}$$

is performed to detect whether or not a given logic function contains threshold terms. If the condition $h = 1$ is satisfied for all combinations of the input signals, a threshold term g can be extracted from a given logic function f. In step 3, we generate a *satisfiability don't*

Given: a sum of product terms f

Procedure Logic synthesis for a PLA using threshold terms

$C = \{\}$

for all possible threshold terms g in f ($T \geq 1.5$)

$h = f + \overline{g}$

if $h = 1$ is satisfied for all combinations of the input signals **then**

$C = C \cup g$

end if

end for

$h = 0$

for each threshold term g_i in C

Create a new variable p_i to represent g_i.

$h = h + (g_i \oplus p_i)$

$f = f + p_i$

end for

Simplify f with h as a *don't care* set and obtain f_{TH}.

Replace p_i in f_{TH} with g_i.

return f_{TH}

end Procedure

Figure 5.9 Procedure of logic synthesis for a PLA using threshold logic circuits.

care set [29] by using the extracted threshold terms, and the set is used to minimize the number of terms, where the number of terms is the total number of product and threshold terms. For example, suppose a Boolean function such as

$$f = x_0\overline{x}_1 + x_0x_3 + \overline{x}_1x_3 + \overline{x}_0x_1\overline{x}_2 \tag{5.14}$$

and extracted threshold terms

$$t^3_{1.5}(\overline{x}_0, x_1, \overline{x}_3),\ \ t^4_{1.5}(\overline{x}_0, x_1, x_2, \overline{x}_3),\ \ t^4_{1.5}(\overline{x}_0, x_1, \overline{x}_2, \overline{x}_3). \tag{5.15}$$

By using these threshold terms, we create new variables, p_0, p_1, p_2, to represent each threshold terms and generate the function

$$\begin{aligned} h = &(t^3_{1.5}(\overline{x}_0, x_1, \overline{x}_3) \oplus p_0) + (t^4_{1.5}(\overline{x}_0, x_1, x_2, \overline{x}_3) \oplus p_1) \\ &+ (t^4_{1.5}(\overline{x}_0, x_1, \overline{x}_2, \overline{x}_3) \oplus p_2). \end{aligned} \tag{5.16}$$

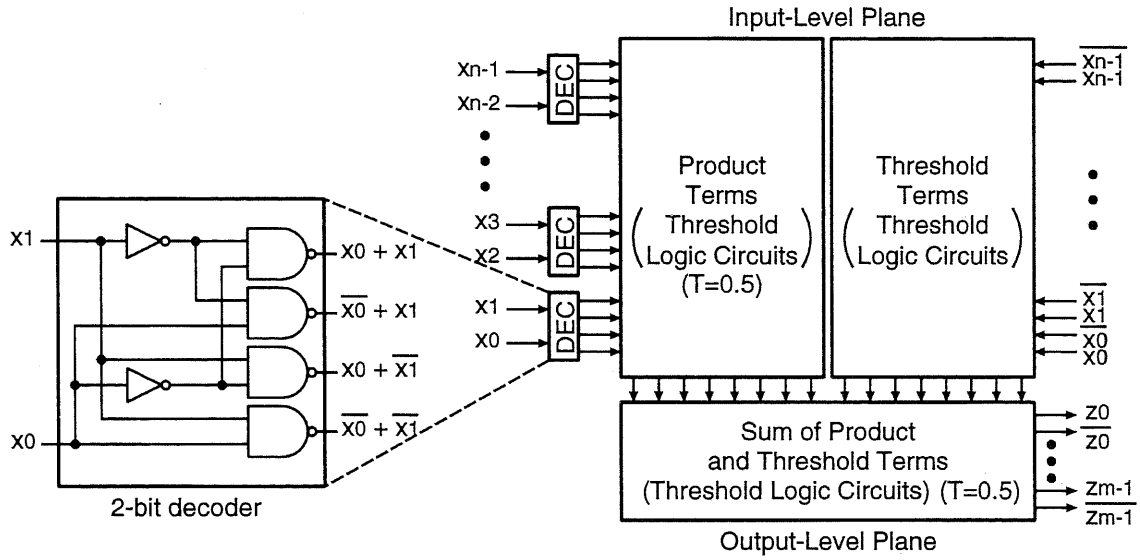
Finally, we solve a covering problem of f with h as a don't care set and obtain an optimized expression

$$f_{TH} = p_0 + \overline{x}_0x_1\overline{x}_2 = t^3_{1.5}(\overline{x}_0, x_1, \overline{x}_3) + \overline{x}_0x_1\overline{x}_2. \tag{5.17}$$

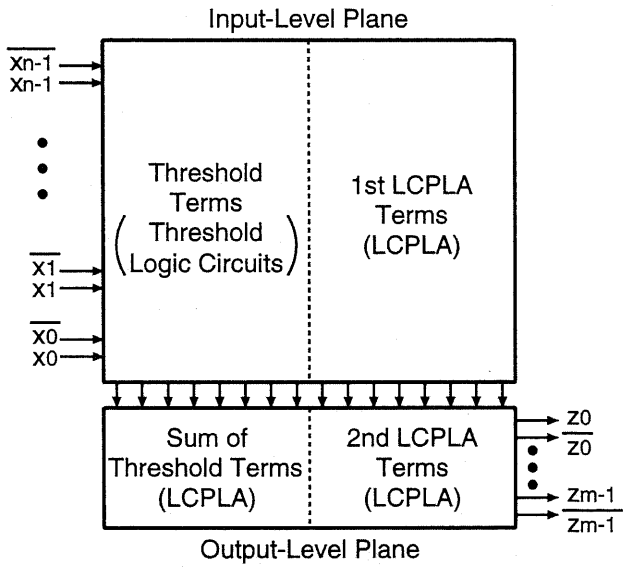
Also, in multiple-output functions, the procedure is performed by considering all outputs at the same time.

5.4.2 Logic Synthesis for Hybrid Architectures

As can be seen in the previous section, the proposed architecture can be designed in column-circuit level, so that it is also possible to build the proposed circuit into LCPLA proposed in Chapter 3 and the conventional area-efficient design techniques. Figure 5.10 shows the proposed hybrid architectures with two design styles: a PLA with 2-bit decoders [41] and LCPLA. The LCPLA has 2-input logic cells in an AND-plane and an OR-plane, and can be considered as an extension of a PLA with 2-input decoders. The logic cells can be embedded in a dual-rail structure without degradation of circuit performance, while 2-bit decoders degrades circuit performance due to an additional gate delay



(a)



(b)

Figure 5.10 Proposed hybrid architectures with (a) a PLA with 2-bit decoders [41] and (b) LCPLA.

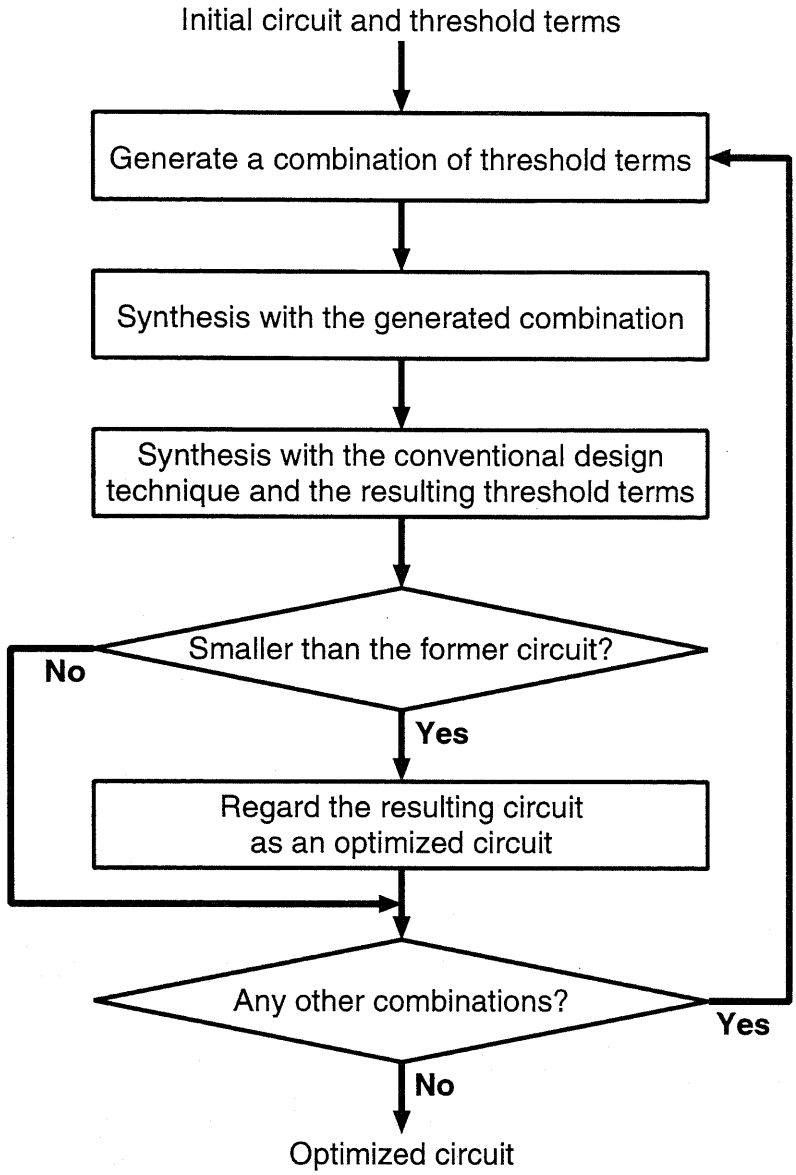


Figure 5.11 Flow of logic synthesis for a hybrid architecture.

Given: a sum of product terms f (original function) and \
a set of threshold terms G (given by f_{TH} in Figure 5.9)
Procedure Logic synthesis for a hybrid architecture
$f_{INI} = f$
$f_{OPT} = f$
for all possible combinations C of threshold terms in G
$h = 0$
$f = f_{INI}$
for each threshold term g_i in C
Create a new variable p_i to represent g_i.
$h = h + (g_i \oplus p_i)$
$f = f + p_i$
end for
Simplify f with h as a *don't care* set and obtain f_{TH}.
$f = f_{TH}$
Remove p_i from f.
Replace p_i in f_{TH} with g_i.
$h = 0$
for each threshold term g_i in f_{TH}
$h = h + g_i$
end for
Simplify f with the conventional design technique
and h as a *don't care* set, and obtain f_{PR}.
if the total number of product terms in f_{PR} and \
threshold terms in f_{TH} is smaller than that in f_{OPT} **then**
$f_{OPT} = f_{PR} + h$
end if
end for
return f_{OPT}
end Procedure

Figure 5.12 Procedure of logic synthesis for a hybrid architecture.

of the decoders. The flow and the procedure of logic synthesis for a hybrid architecture are shown in Figure 5.11 and Figure 5.12, respectively. Threshold terms given by the output of the method in Figure 5.9 are used as input of this procedure. All possible combinations of given threshold terms are explored with the conventional design technique to minimize the number of terms in a given logic function f. Also, in multiple-output functions, the procedure is performed by considering all outputs at the same time.

5.5 Results of Logic Synthesis

We first investigated the complexity of threshold logic functions in the conventional PLAs. Table 5.1 shows a comparison of the number of product terms of threshold logic functions in various PLA styles, where the functions having the number of input signals up to 8 are shown. In this table, AND-OR is the conventional PLA which has an AND-plane and an OR-plane, and AND-OR with 2-bit decoders [41] and LCPLA are also shown. We used a two-level logic minimizer, ESPRESSO [8], [9], to optimize AND-OR structures and to generate PLAs with 2-bit decoders. As can be seen from this table, the complexity of threshold logic functions of $t^i_{i/2}$ or $t^i_{i/2+0.5}$ is highest when the number of input signals is fixed, and threshold logic circuits become more effective as the number of input signals increases.

Table 5.2 shows the results of logic synthesis on some PLA benchmark circuits [8], [9]. In addition to the structures of AND-OR, AND-OR with 2-bit decoders, LCPLA, and the proposed architecture, the results in hybrid architectures with 2-bit decoders and LCPLA are also shown. Each structure is based on the circuit optimized by ESPRESSO, and the structure of AND-OR with 2-input decoders was generated by ESPRESSO. Also, tautology checking and optimization in the procedures of Figure 5.9 and Figure 5.12 were performed by ESPRESSO. Note that the proposed architecture and a hybrid architecture with LCPLA can be implemented in the same area as LCPLA even if the number of product terms can not be reduced by logic synthesis. Also, LCPLA is preferable to an AND-OR structure with 2-bit decoders because 2-input logic cells in LCPLA can be

Table 5.1 Comparison of the number of product terms of threshold logic functions in various PLA styles.

Function	AND-OR	AND-OR + 2-bit decoders	LCPLA	This work
$t^2_{0.5}$	1	1	1	1
$t^2_{1.5}$	2	1	1	1
$t^3_{0.5}$	1	1	1	1
$t^3_{1.5}$	3	2	2	1
$t^3_{2.5}$	3	2	1	1
$t^4_{0.5}$	1	1	1	1
$t^4_{1.5}$	4	2	2	1
$t^4_{2.5}$	6	3	2	1
$t^4_{3.5}$	4	2	1	1
$t^5_{0.5}$	1	1	1	1
$t^5_{1.5}$	5	3	3	1
$t^5_{2.5}$	10	5	4	1
$t^5_{3.5}$	10	5	4	1
$t^5_{4.5}$	5	3	1	1
$t^6_{0.5}$	1	1	1	1
$t^6_{1.5}$	6	3	3	1
$t^6_{2.5}$	15	6	5	1
$t^6_{3.5}$	20	7	6	1
$t^6_{4.5}$	15	6	4	1
$t^6_{5.5}$	6	3	1	1
$t^7_{0.5}$	1	1	1	1
$t^7_{1.5}$	7	4	4	1
$t^7_{2.5}$	21	9	8	1
$t^7_{3.5}$	35	13	10	1
$t^7_{4.5}$	35	13	8	1
$t^7_{5.5}$	21	9	6	1
$t^7_{6.5}$	7	4	1	1
$t^8_{0.5}$	1	1	1	1
$t^8_{1.5}$	8	4	4	1
$t^8_{2.5}$	28	10	9	1
$t^8_{3.5}$	56	16	12	1
$t^8_{4.5}$	70	19	13	1
$t^8_{5.5}$	56	16	10	1
$t^8_{6.5}$	28	10	6	1
$t^8_{7.5}$	8	4	1	1

Table 5.2 Results of logic synthesis.

	AND-OR	AND-OR + 2-bit decoders	LCPLA	This work	
Circuit	# product terms	# product terms	# product terms	# product terms	# threshold terms
adr4	75	17	17	61	7
max128	83	71	70	45	9
mlp4	128	97	92	123	11
rd53	31	12	12	26	1
rd73	127	37	34	85	2
root	57	42	40	52	3
z4	59	16	16	45	7
Z9sym	86	26	26	1	33
Total	646	318	307	438 + 63	
Ratio	1.00	0.492	0.475	0.776	

	This work + 2-bit decoders		This work + LCPLA	
Circuit	# product terms	# threshold terms	# product terms	# threshold terms
adr4	17	0	17	0
max128	40	9	40	9
mlp4	87	1	87	1
rd53	9	1	9	1
rd73	20	2	20	2
root	42	0	40	0
z4	16	0	16	0
Z9sym	26	0	26	0
Total	257 + 13		255 + 13	
Ratio	0.418		0.415	

embedded without degradation of circuit performance, while 2-bit decoders degrades circuit performance due to an additional gate delay of the decoders. The results show that the number of product terms of the conventional AND-OR structure can be reduced by 22.4%, 58.2%, and 58.5% on average by using the proposed architecture and hybrid architectures with 2-bit decoders and LCPLA, respectively. Also, the number of product terms of LCPLA can be reduced by 12.7% on average by using a hybrid architecture with LCPLA.

5.6 Simulation Results

Table 5.3 shows a comparison of area, delay, and power consumption between different implementation styles. Each circuit was designed using a 0.35-μm, 3-metal-layer CMOS technology with a supply voltage of 3.3 V. Logic synthesis of the standard-cell-based designs was performed using Synopsys Design Compiler [31] with delay priority and 395 standard cells. The standard cells are based on the Rohm 0.35-μm CMOS technology and were provided by VLSI Design and Education Center (VDEC) [32]. The synthesized circuits were placed and routed using Synopsys Apollo [31]. The delay and power characteristics of the standard-cell-based designs were obtained by the timing and power analysis of Synopsys Design Compiler with wire load information, which is given by placement and routing, while the area characteristics were obtained by physical layouts after placement and routing. On the other hand, the delay and power characteristics of the PLAs were obtained from post-layout simulations using HSPICE. Also, the area characteristics represent physical layouts, and a sense voltage of 200 mV was used. The results show that the hybrid architecture with LCPLA requires on average 2.85 times the area of the standard-cell-based designs. This is mainly because of dual-rail structures and the difference of area-efficiency between regular and random structures. The delay of the hybrid architecture is on average 0.591 times that of the standard-cell-based designs. In addition to the high-speed capability of sense amplifiers, the hybrid architecture implements logic functions in a two-level logic network and thus results in having superior

Table 5.3 Comparison of area, delay, and power consumption between different implementation styles.

	Area (μm^2)			Delay (ns)		
Circuit	Standard cell	LCPLA	This work + LCPLA	Standard cell	LCPLA	This work + LCPLA
adr4	5,561	32,727	32,727	0.94	0.65	0.65
max128	36,132	163,953	122,513	1.12	0.70	0.72
mlp4	47,306	129,760	124,821	1.32	0.73	0.77
rd53	4,737	21,406	19,699	0.90	0.64	0.69
rd73	18,088	44,463	33,131	1.15	0.67	0.71
root	19,418	57,747	57,747	1.28	0.68	0.68
z4	3,886	28,891	28,891	0.75	0.65	0.65
Z9sym	24,669	36,635	36,635	1.89	0.66	0.66
Total	159,797	515,582	456,164	9.35	5.38	5.53
Ratio	1.00	3.23	2.85	1.00	0.575	0.591

	Power (mW) @ 200 MHz			PD product (pJ)		
Circuit	Standard cell	LCPLA	This work + LCPLA	Standard cell	LCPLA	This work + LCPLA
adr4	3.3	8.0	8.0	3.10	5.20	5.20
max128	18.6	38.5	24.9	20.83	26.95	17.93
mlp4	24.0	39.6	36.0	31.68	28.91	27.72
rd53	3.0	5.4	4.7	2.70	3.46	3.24
rd73	10.7	15.2	9.7	12.31	10.18	6.89
root	10.9	17.7	17.7	13.95	12.04	12.04
z4	2.5	7.3	7.3	1.88	4.75	4.75
Z9sym	14.2	10.6	10.6	26.84	7.00	7.00
Total	87.2	142.3	118.9	113.29	98.49	84.77
Ratio	1.00	1.63	1.36	1.00	0.869	0.748

delay characteristics. In the standard-cell-based designs, the increase in delay is incurred in traversing the different logic levels of the circuits. Moreover, the maximum delay of the standard-cell-based designs is 1.89 ns (Z9sym), while the minimum delay is 0.75 ns (z4). Thus, the maximum difference of delay between each circuit is 1.14 ns. On the other hand, the maximum delay of the hybrid architecture is 0.77 ns (mlp4), while the minimum delay is 0.65 ns (adr4 and z4). Thus, the difference of delay between each circuit is limited only within 0.12 ns. Also, the worst case delay of the hybrid architecture can be easily obtained from the delay of a column circuit having a maximally loaded bit-line. This feature makes the circuit delay predictable early in the design process, and thus it is possible to eliminate the ambiguity of the design and to alleviate the timing closure problem. The power consumption of the hybrid architecture is on average 1.36 times that of the standard-cell-based designs. This is mainly because dual-rail bit-lines are charged and discharged every cycle, and the power consumption of sense amplifiers in the PLA is relatively large compared to the other components. However, the increase in power consumption becomes small as the number of input signals increases because a sense amplifier consumes the same power even in larger circuits. The power-delay (PD) product of the hybrid architecture is on average 0.748 times that of the standard-cell-based designs.

5.7 Field-Programmable Design

As an extension of the proposed architecture, we propose a field-programmable logic array (FPLA) based on threshold logic circuits. Figure 5.13 shows the proposed field-programmable threshold logic circuit. The circuit is an extension of the threshold logic circuit in Figure 5.2 and additional transistors with SRAM cells are connected in series with logic and reference cells to configure circuit functions. When M1 is turned on, BL is pulled down by field-programmable logic cells depending on the input signals. Also, when M4 is turned on, $\overline{\text{BL}}$ is pulled down by field-programmable logic cells depending on the condition of M3. Thus, when M1 and M4 are turned off, the logic cell is considered as a logic cell of type 0 in Figure 5.3. M1, M3, and M4 are configured by a 2-bit SRAM

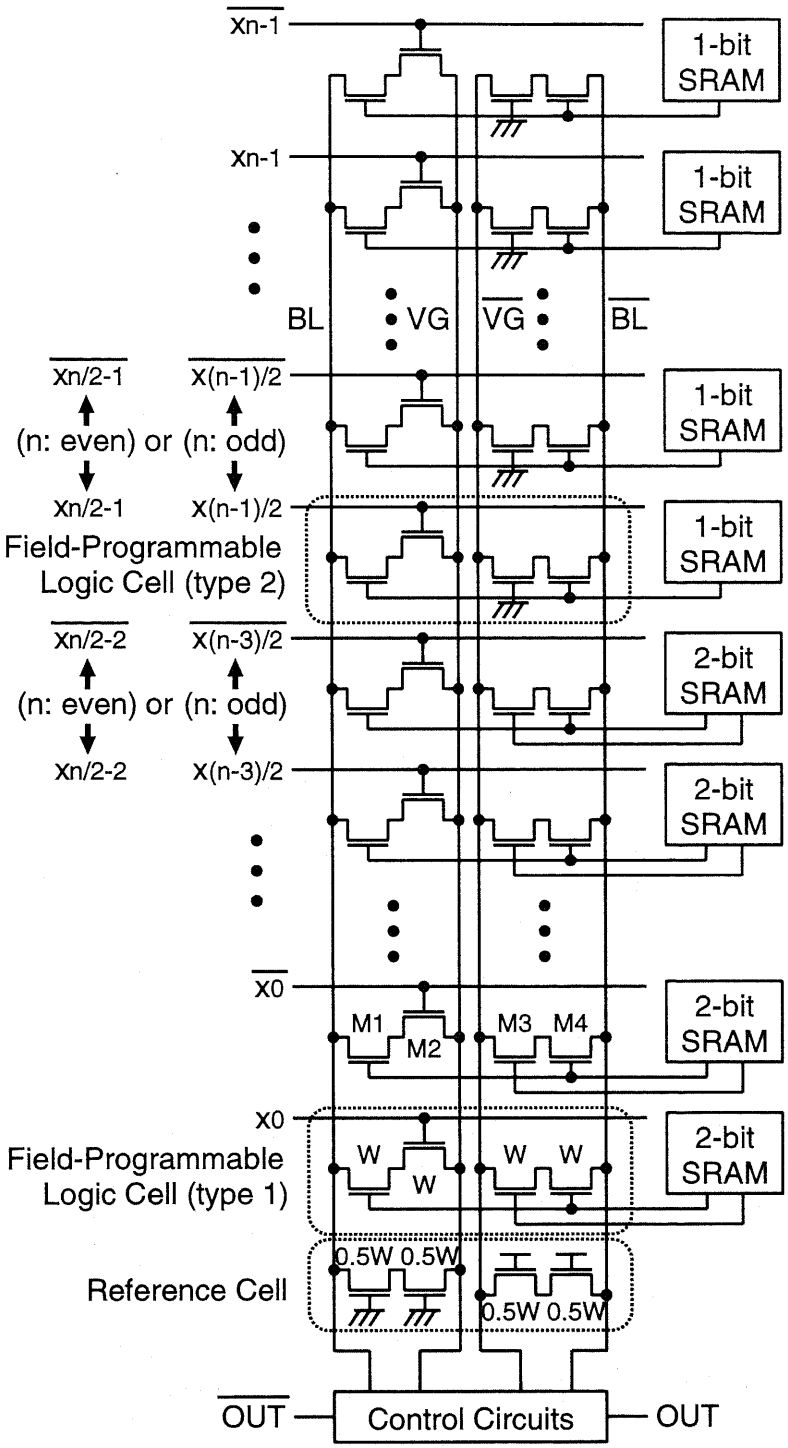


Figure 5.13 Proposed field-programmable threshold logic circuits for threshold terms (control circuits in the figure are shown in Figure 5.2).

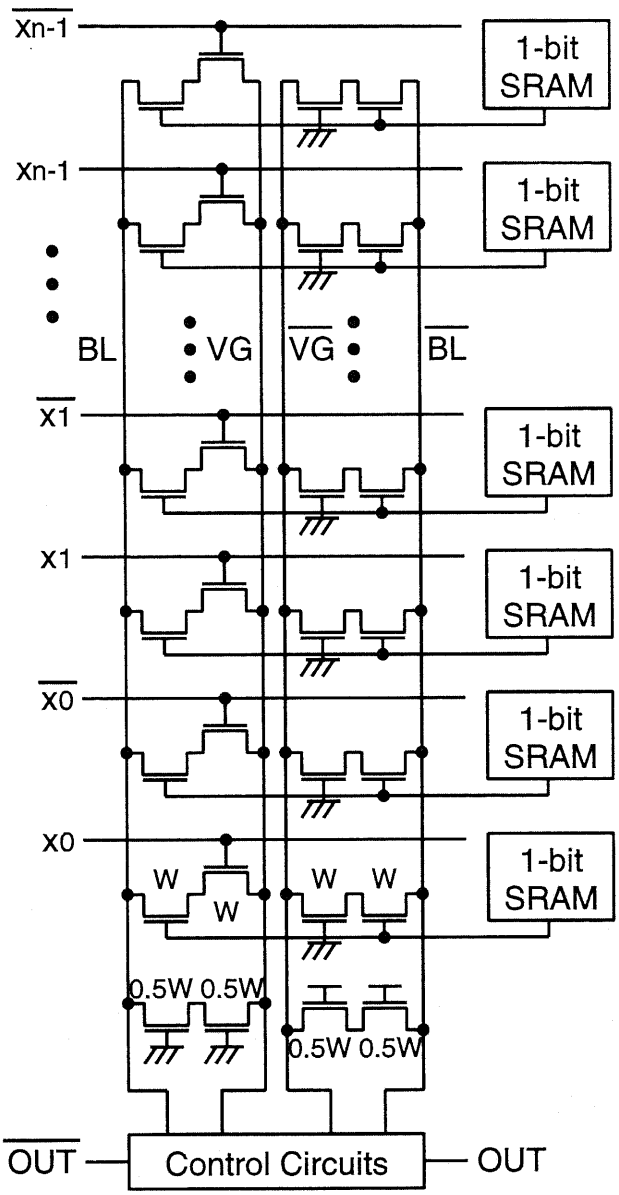


Figure 5.14 Proposed field-programmable threshold logic circuits for product terms (control circuits in the figure are shown in Figure 5.2).

cell depending on a given logic function. As discussed above, the maximum number of logic cells of type 3 and type 4 in Figure 5.3 is given by Eq.(5.12). Thus, 1-bit SRAM cells are used instead of 2-bit SRAM cells in field-programmable logic cells of type 2.

In the proposed FPLA, the hybrid architecture with 2-bit decoders shown in Figure 5.10(a) can be utilized. As can be seen from Table 5.2, this architecture is effective to implement Boolean functions efficiently. On the other hand, LCPLA requires various configurations for embedded logic cells, so that a large capacity of memory is needed and the architecture shown in Figure 5.10(b) is not suitable for field-programmable design in terms of circuit density. In the architecture of Figure 5.10(a), the circuit of Figure 5.13 is used as a part of the input-level plane to realize threshold terms. Figure 5.14 shows the proposed field-programmable threshold logic circuit for product terms. The circuit has a threshold value of 0.5 and is used as a part of the input-level plane and whole of the output-level plane in the architecture of Figure 5.10(a).

5.8 Extension to the Design of Functional Memory

As an application of the threshold logic circuit proposed in the previous sections, we propose a new functional memory with the high-speed flexible data search capability. A data search function is a key operation for many applications such as image processing [42], network communication [43], [44], cache memory [45], and so on. The system performance of these applications can be significantly enhanced by using special-purpose hardwares, where parallel data comparisons are performed to achieve fast data search operations. For this purpose, content-addressable memories (CAMs) are widely used to achieve the functions of the exact-match [46]–[51] or nearest-match data search [52]–[55] between stored data and an input data. However, these search functions are not always enough in applications that need an ambiguous data search to cope with the presence of noise and variability in the environment [56]–[58]. In addition to the conventional function of the exact-match data search, the proposed functional memory is capable of retrieving the address of stored data within a given Hamming distance from an input data

in a single clock cycle. This function is called the *threshold-match* search hereafter. If the exact-match search CAM is used to achieve this function, iterative search operations using additional circuits, such as a mask register to generate all possible search keys and a register to accumulate intermediate results, are required, so that it takes a lot of clock cycles to complete the operation. In the proposed scheme, the search operation can be performed in a single clock cycle as well as the exact-match operation, thus resulting in a high throughput rate. Moreover, the proposed circuit does not require current-mode operations, while the conventional high-speed CAM [55] operates in a current mode and suffers from static power consumption. Thus, the issue of static power consumption is not included in the proposed scheme. In addition to the threshold-match search, it is also possible to retrieve the address of stored data with a given Hamming distance from an input data by performing the threshold-match search twice. This function is called the *distance-match* search hereafter. These functions are not only effective for applications that cope with the ambiguity against noise and variability [56]–[58] but more adequate to realize human impression-like functions compared to the exact-match data search.

5.8.1 Proposed Architecture

The block diagram of the proposed functional memory is shown in Figure 5.15. It mainly consists of a memory cell array with threshold logic circuits, a function selector, a priority decision circuit, and peripheral circuits for memory read and write operations. The mask-pattern register provides *don't care* conditions, which are used in an additional search operation, to the memory cell array. Binary data are stored in the memory cell array through the input-pattern and mask-pattern registers, and can be manipulated by four types of operations: search, masked search, read, and write. The search function is performed by comparing an input pattern with all the stored data using a distance parameter *REF*. There are two possible results of comparison in each word depending on a search condition: match and mismatch. The results are provided to the priority decision circuit through the function selector, which selects a search function, i.e., the threshold-match or

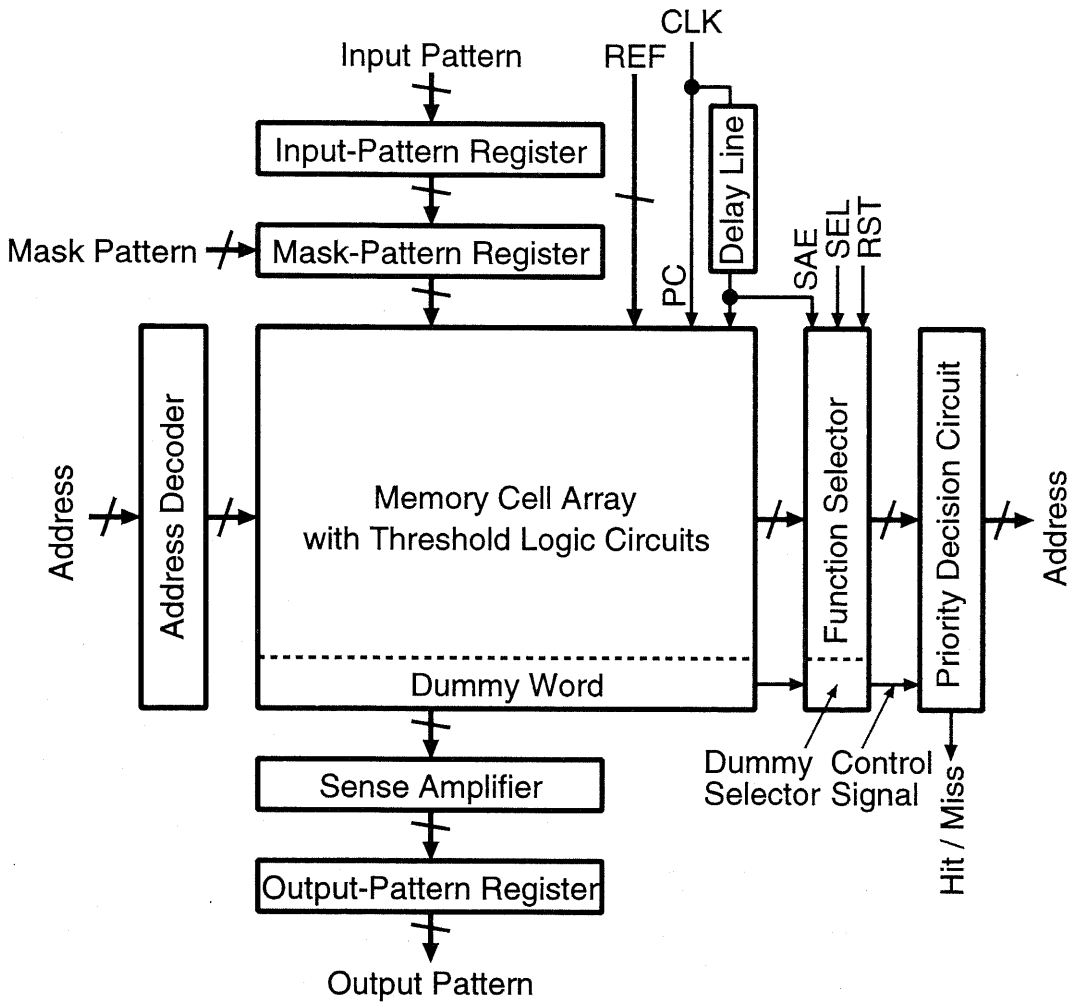


Figure 5.15 Block diagram of the proposed functional memory.

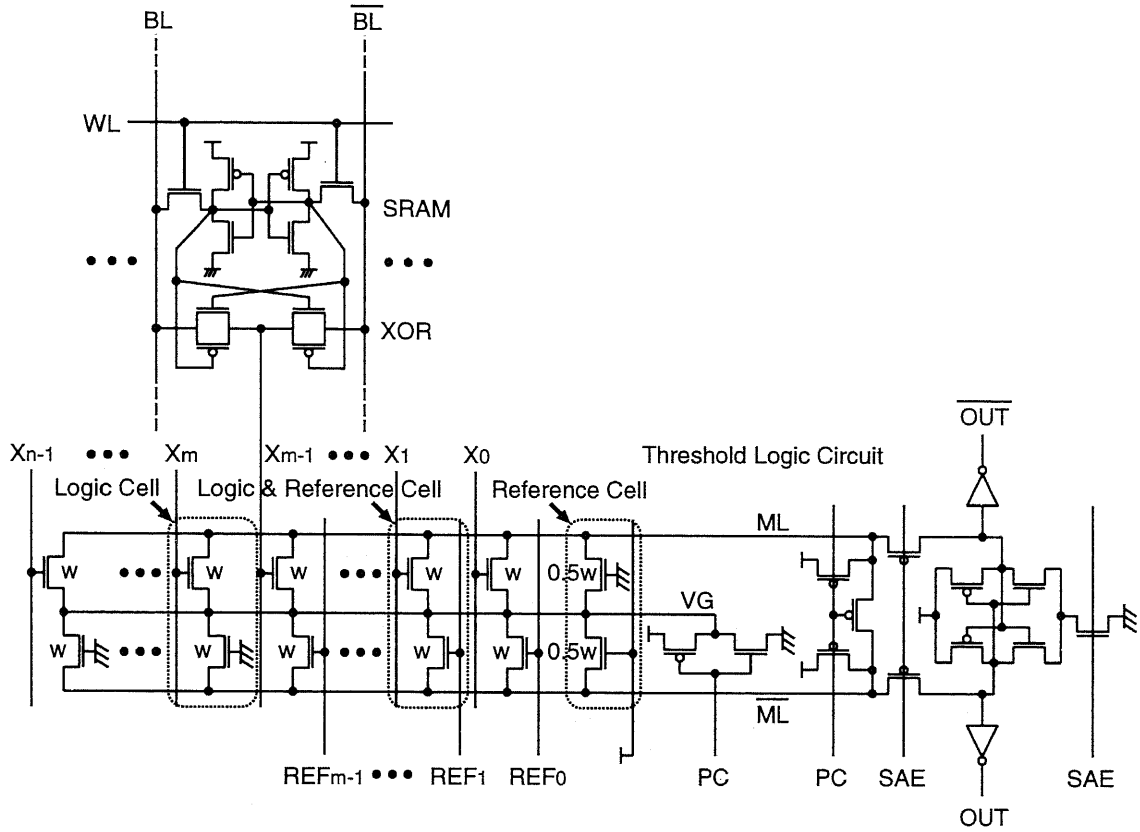
distance-match search. Finally, the address with the highest priority can be obtained from the priority decision circuit.

5.8.2 Proposed Circuits

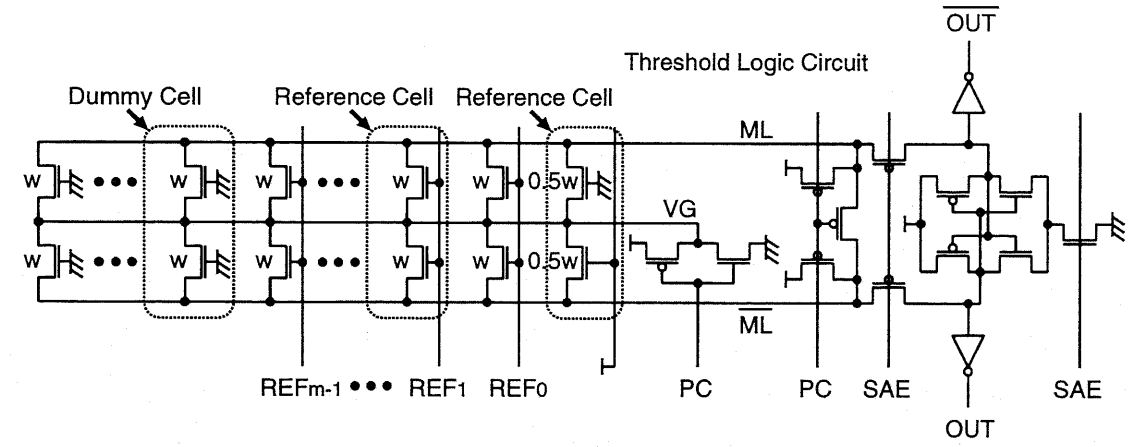
Figure 5.16(a) shows a word circuit of the proposed memory. The circuit consists of three major elements: six-transistor SRAM cells, data comparison circuits (2-input XOR gates), and a threshold logic circuit. Moreover, the threshold logic circuit consists of a stack of logic cells, reference cells, a precharge and equalization circuit, and a sense amplifier. Also, VG is precharged high in a precharge phase. This allows the input signals (x_0–x_{n-1}) to arrive before the *PC* signal becomes high, and is effective to avoid introducing errors caused by signal skew between the input and reference signals. The XOR gates compare stored data with an input pattern on the bit-lines (BL and $\overline{\text{BL}}$). The threshold logic circuit is controlled by the output signals of the XOR gates, multiple reference signals (REF_0–REF_{m-1}), and control signals, *PC* and *SAE*, which are generated from the *CLK* signal with a delay line of a chain of sized inverters. The multiple reference signals generate multiple threshold voltages in the match-line $\overline{\text{ML}}$. The purpose of dummy devices, of which the gate terminals are connected to ground, is to balance load capacitances and leakage current of the dual-rail match-line (ML and $\overline{\text{ML}}$). The memory operations and corresponding logic values on the bit-lines are summarized in Table 5.4.

Figure 5.17 shows a timing diagram of control, input, and output signals, and match-line potentials in a threshold logic circuit. The *PC* signal precharges and equalizes the match-lines. The output signals of the XOR gates (x_0–x_{n-1}) must arrive before the *PC* signal becomes high. The *SAE* signal activates a sense amplifier and isolates it from the match-lines when the developed voltage difference between the match-lines becomes larger than the designed sense voltage, which takes the worst case of considerable noise margin and process variations into account. The output signals (*OUT* and $\overline{OUT}$) are obtained by sensing the differential voltage of the match-lines.

The pull-down speed of $\overline{\text{ML}}$ depends on the number of the *REF* signals activated to



(a)



(b)

Figure 5.16 Word circuit with the proposed threshold logic circuit. (a) Memory word and (b) dummy word.

Table 5.4 Memory operations and corresponding logic values on the bit-lines.

Logic value	BL	$\overline{\text{BL}}$	Operations
1	1	0	write, search
0	0	1	write, search
masked	0	0	masked search
High-Z	High-Z	High-Z	read

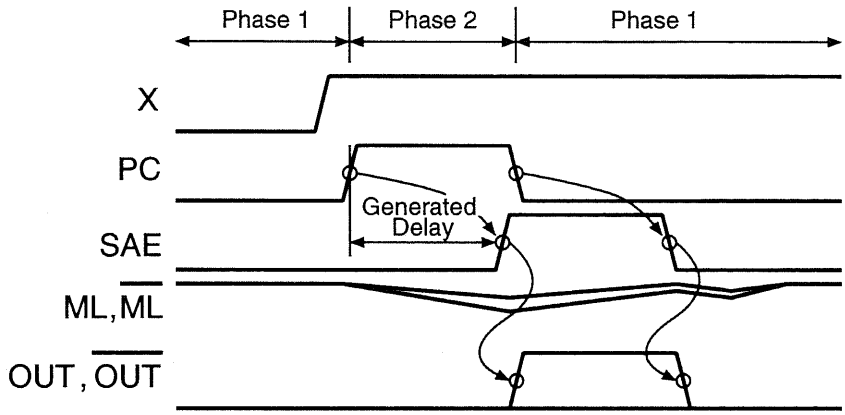


Figure 5.17 Timing diagram of control, input, and output signals, and match-line potentials in a threshold logic circuit.

high, corresponding to a given Hamming distance. On the other hand, that of ML depends on the number of the output signals of the XOR gates activated to high. The device size of a logic cell is W, where W is the channel width of a transistor. On the other hand, that of a reference cell is $0.5W$ as shown in Figure 5.16. This half-size device is provided to avoid the meta-stable condition, which may be caused when the distance between stored data and an input pattern is equal to that given by the *REF* signals. Figure 5.18 shows transient relationships between the match-lines. Stored data within a given distance can be detected in a single clock cycle by comparing the voltage potentials of ML and $\overline{\text{ML}}$.

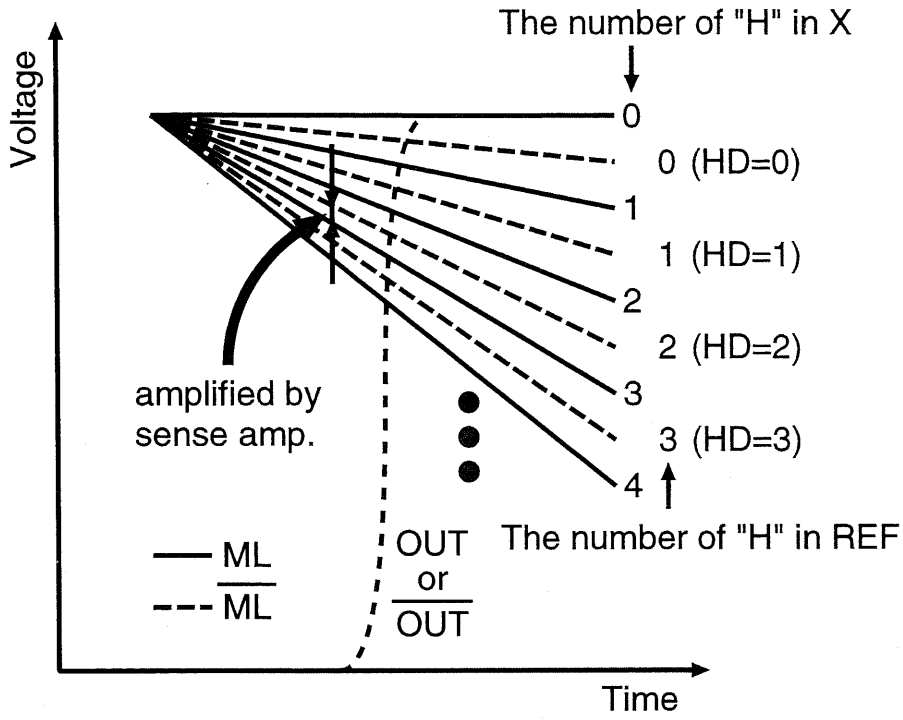


Figure 5.18 Simplified transient relationships between the match-lines (ML and $\overline{\text{ML}}$).

The output signal OUT can be expressed as

$$OUT = \begin{cases} 1 & if \sum_{i=0}^{n-1} x_i \leq HD \\ 0 & otherwise \end{cases} \tag{5.18}$$

where HD is a Hamming distance given by the REF signals.

In addition, a virtual ground (VG) is used in the pull-down path of the match-lines. The VG node is forced to high during the precharge phase of the match-lines, so that states of SRAM bit-lines do not affect the precharge operation. Thus, word circuits can precharge the match-lines without discharging the bit-lines. Moreover, voltage swings of the match-lines can be reduced by controlling the PC signal at an appropriate time. This results in reducing a precharging time and power consumption.

Figure 5.16(b) shows a dummy word circuit, which is placed at the end of a memory cell array as shown in Figure 5.15. The purpose of this circuit is to provide a control signal

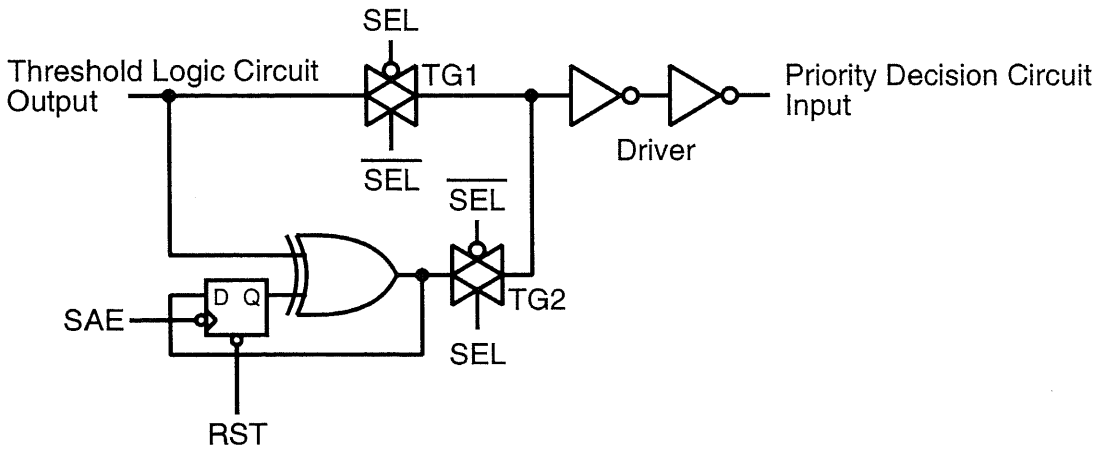
for the priority decision circuit. The circuit does not include SRAM cells, and the devices on both match-lines are connected to the *REF* signals to generate the minimum voltage difference of the match-lines in all of the words. Since the circuit has the same structure as a memory word circuit, a control signal in sync with the other words can be generated across operating conditions and process variations.

The output signals of each word are transferred to the function selector, which selects a search function, i.e., the threshold-match or distance-match search. Figure 5.19(a) shows the structure of the function selector. If *SEL* is low, the output signal of a threshold logic circuit is provided to the priority decision circuit through a pass gate (TG1) and a driver circuit. Thus, the threshold-match search is performed in a single clock cycle by using the capability of a threshold logic circuit. On the other hand, if *SEL* is high, the distance-match search is done by comparing the output signal of a threshold logic circuit with the former output, which is stored in a D flip-flop, by using a 2-input XOR gate, so that it takes two clock cycles. For example, suppose that a Hamming distance of 3 is given by the *REF* signals in the first search operation. If a Hamming distance of 4 is given in the next search operation, stored data with a Hamming distance of 4 can be detected by comparing two search results.

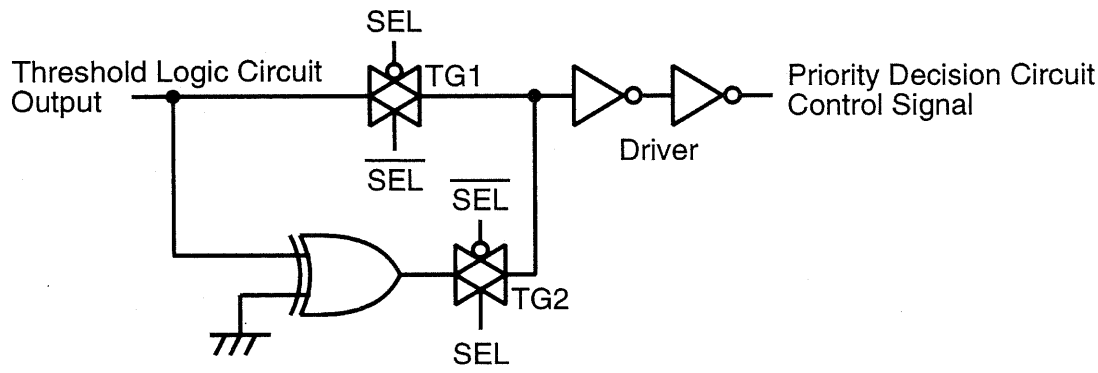
Figure 5.19(b) shows the function selector for a dummy word, which is placed at the end of the function selector's block as shown in Figure 5.15. The circuit operates in sync with the other selectors in the same manner as a dummy word and provides a control signal for the priority decision circuit every clock cycle.

5.8.3 Simulation Results

A 32 × 128-bit memory macro was designed using a 0.35-μm, 3-metal CMOS technology. Figure 5.20 shows simulated waveforms of a threshold-match search operation with a supply voltage of 3.3 V. The delay from the *CLK* signal to the output of the priority decision circuit is 2.34 ns. The power consumption is 399 mW at an operating frequency of 256 MHz. The features of the memory macro are summarized in Table 5.5. The distance-



(a)



(b)

Figure 5.19 Function selector for (a) a memory word and (b) a dummy word.

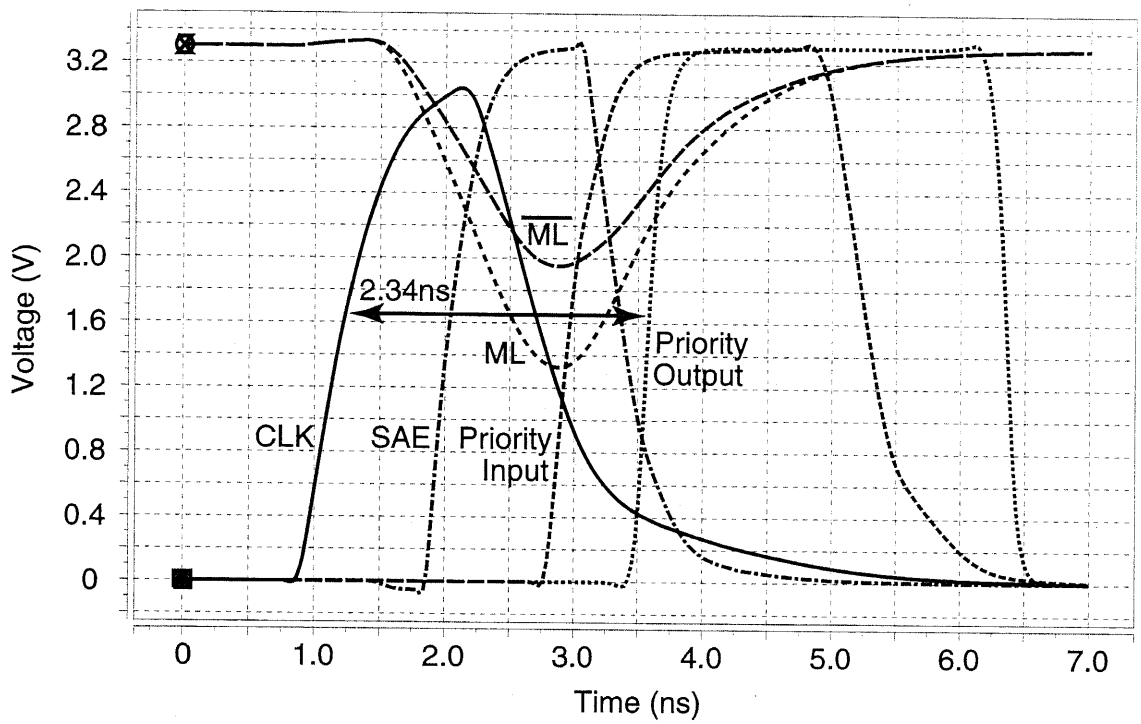


Figure 5.20 Simulated waveforms of a threshold-match search operation.

match search takes two clock cycles, so that a search cycle time is added in delay.

In order to evaluate the robustness of a threshold logic circuit against process variations, a Monte Carlo simulation for the deviation of threshold voltage (V_{th}) of transistors was performed. The simulation results are shown in Figure 5.21. 3σ is assumed to be from 0 V to 200 mV, where σ is the standard deviation of V_{th}. This V_{th} deviation was applied to the NMOS devices of logic cells, a reference cell, and a sense amplifier in a threshold logic circuit, which mainly affect circuit operations. The results show that a threshold logic circuit achieves robust operations up to the number of references of 7 (i.e., Hamming distance of 7) under considerable deviation of 80 mV. This indicates that the *REF* signals of up to a distance of 7 (i.e., $m=7$ in Figure 5.16) can be provided to a threshold logic circuit.

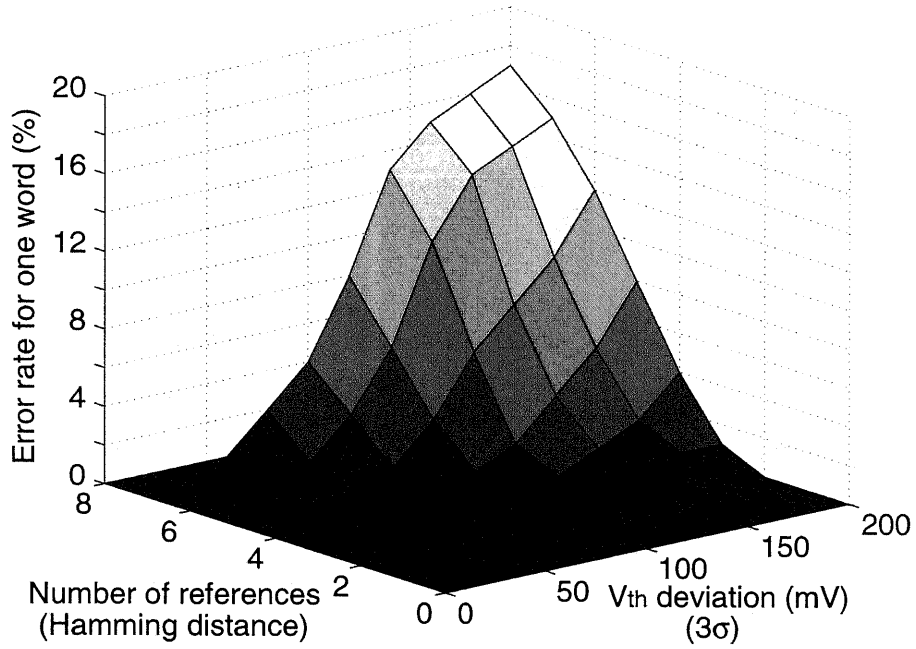


Figure 5.21 Results of Monte Carlo simulation of a threshold logic circuit.

5.8.4 Measurement Results

The designed memory macro was fabricated using a 0.35-μm, 3-metal CMOS technology. A chip microphotograph is shown in Figure 5.22. The operations of the macro within a Hamming distance of 7 were successfully verified by a functional test using a logic tester at frequencies of up to 100 MHz, which is the limitation of the test equipment, at room temperature with a supply voltage of 3.3 V.

Figure 5.23 shows measured waveforms of a threshold-match search operation using an electron-beam tester at the same conditions as the functional test. The timebase resolution of the test equipment is 10 ps. The measurement was performed with a given Hamming distance of 7 by the *REF* signals. The delay from the *CLK* signal to the output of the priority decision circuit was 2.25 ns. This performance means that the proposed memory is faster than the conventional high-speed nearest-match CAM [55], which takes at least deca-ns for search, when the threshold-match search is emulated using the nearest-match search within the minimum distance of 7 bits. The measured characteristics of the macro are summarized in Table 5.5.

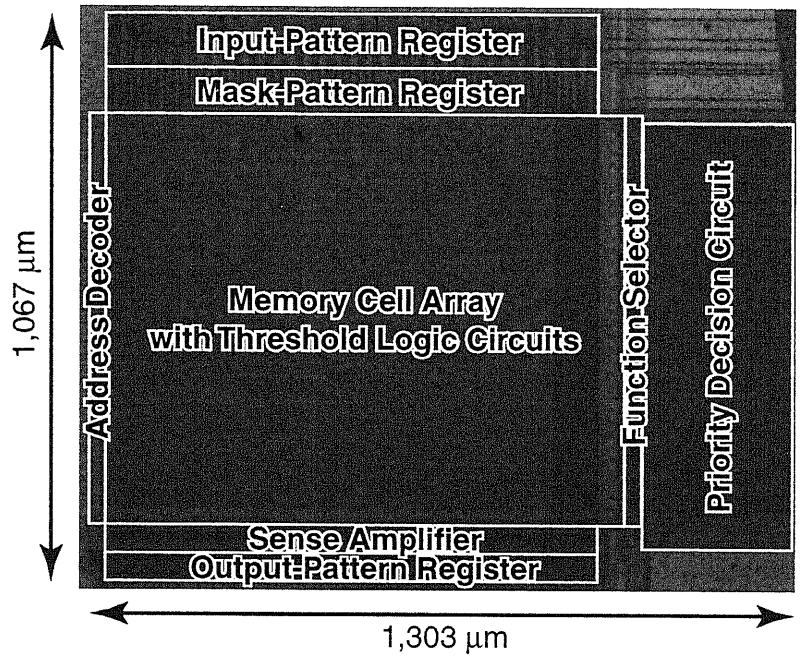


Figure 5.22 Chip microphotograph of the memory macro.

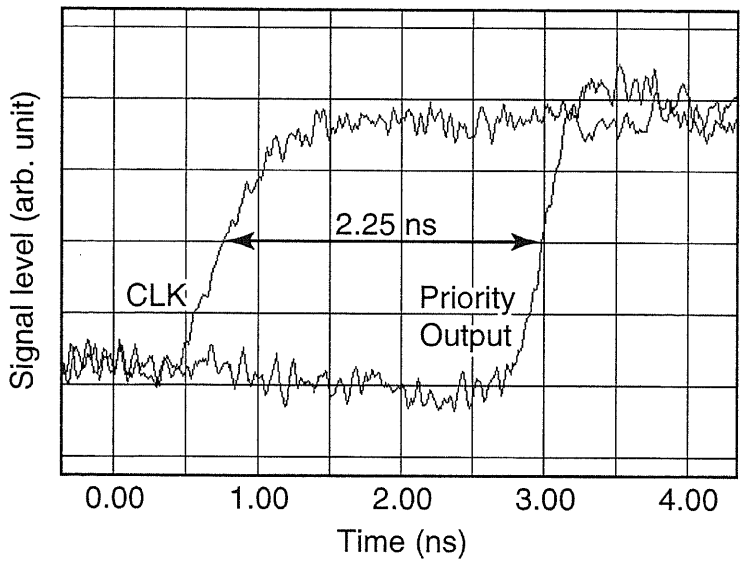


Figure 5.23 Measured waveforms by an electron-beam tester.

Table 5.5 Features of the memory macro.

Process technology	0.35-μm CMOS, 3-metal
Supply voltage	3.3 V
Organization	32 × 128-bit
Cell size	22.50 × 7.00 μm^2
Macro size	1,067 × 1,303 μm^2
Transistor count	73.7 k
Simulated search cycle time	3.90 ns (256 MHz)
Simulated power consumption	399 mW (@ 256 MHz search operation)
Simulated delay	2.34 ns (threshold-match search)
	2.65 ns + 1 cycle (distance-match search)
Measured delay	2.25 ns (threshold-match search)
	2.50 ns + 1 cycle (distance-match search)

5.9 Summary

In this chapter, new array logic architectures using dual-rail threshold logic circuits and logic synthesis methods for their architectures were presented. The threshold logic circuits realize multiple-threshold logic functions and are formed in a two-level network. Since the proposed architectures can be optimized in column-circuit level, it is also possible to build the proposed circuit into the proposed design techniques in the previous chapters and the conventional area-efficient design techniques, and thus further reductions of the number of product terms can be achieved. The advantages of the proposed architectures were demonstrated by using benchmark circuits, and the results show that the number of product terms of the conventional AND-OR structure can be reduced by 22.4% on average by using the proposed architecture. Also, the number of product terms, chip area, and power-delay product of LCPLA can be reduced by 12.7%, 11.5%, and 13.9% on average, respectively, by using a hybrid architecture of the proposed circuit and LCPLA. The architecture of a field-programmable logic array based on the threshold logic circuits was also developed and is shown to be effective to implement Boolean functions efficiently.

Also, as an application of the threshold logic circuit, we proposed a new functional memory with the capability of high-speed Hamming-distance-based data search. By using the threshold logic circuit, a single cycle search operation without the issue of static power consumption can be achieved. A 32×128-bit memory macro was designed and fabricated using a 0.35-μm CMOS technology, and we verified a search function of stored data within a Hamming distance of up to 7 from an input data.

Chapter 6

Conclusions

This thesis focused on CMOS array logic architectures using dual-rail threshold logic circuits. We addressed the problems of designing the circuits and proposed new architectures.

In Chapter 2, we proposed a new array logic architecture based on a PLA structure using dual-rail threshold logic circuits. The architecture can realize arbitrary Boolean functions expressed in sum-of-products forms without introducing sensing errors of sense amplifiers which can be seen in the conventional high-speed dual-rail circuit. The threshold logic circuit has a new circuit configuration using a charge sharing scheme and a self-precharge scheme for high-speed and low-power operations. As an application of the proposed array logic architecture, a 32-bit binary comparator was designed and fabricated using a 0.6-μm CMOS technology, and the proposed circuit achieved reductions of cycle time by 20.0%, 45.5%, and 22.6%, and reductions of power-delay product by 32.9%, 34.7%, and 15.4% compared to a domino logic circuit, the conventional single-rail PLA, and the conventional high-speed dual-rail circuit, respectively. The capabilities of high-speed and low-power operations over the conventional array logic architectures can be enhanced as the number of input signals increases and thus the proposed architecture is shown to be more effective in high data bandwidth systems in the future. Also, we investigated the circuit characteristics of single-rail and dual-rail circuits, and showed that dual-rail circuits have advantages in terms of common-mode noise immunity and

leakage-current tolerance, especially in advanced CMOS process technologies.

In Chapter 3, an area-efficient dual-rail array logic architecture, a logic-cell-embedded PLA (LCPLA), was presented. By embedding 2-input logic cells, which realize arbitrary 2-input Boolean functions, in the structure, some classes of logic functions can be implemented efficiently, so that high-speed and low-power operations were also achieved. The logic cells can be designed by connecting some local wires and do not require additional transistors over logic cells of the conventional dual-rail PLA. The advantages over the conventional PLAs and standard-cell-based designs were demonstrated by using benchmark circuits and a developed logic synthesis method, and the LCPLA is shown to be effective to reduce the number of product terms. In a structure with a 64-bit input and a 1-bit output including 220 product terms, the LCPLA using a 0.35-μm CMOS technology achieved an area reduction by 36.0% compared to the proposed PLA in Chapter 2, and the power-delay product was reduced by 74.6% and 46.0% compared to the conventional high-speed single-rail PLA and the proposed PLA in Chapter 2, respectively. Also, a module generator for LCPLA structures was developed as a design automation tool. The generator performs equation-based transistor sizing to achieve a desired delay goal and allows a wide range of performance to be traded for area. The framework and procedure of the generator are much easier than those of the conventional tools for standard-cell-based designs and thus the generator can be easily applied to complicated IC designs in the future.

In Chapter 4, we proposed a high-speed and area-efficient dual-rail array logic architecture using a divided column scheme and an interdigitated column scheme. The architecture is based on the proposed threshold logic circuit described in Chapter 2 and the circuit delay and area can be reduced by physical methods. As applications of the proposed architecture, a comparator, a priority encoder, and an incrementor for 128-bit data processing were designed, and the proposed schemes achieved a 22.2% delay reduction and a 37.5% area reduction on average over the proposed PLA in Chapter 2 in a 0.13-μm CMOS technology. The high-speed capability of the proposed schemes can be enhanced as the number of input signals increases and thus the proposed architecture is shown to

be more effective in high data bandwidth systems in the future. Moreover, a lot of circuit elements can be shared among different macro designs, thus reducing the design complexity significantly. Also, by using the proposed schemes, an activation pulse for sense amplifiers in the circuit can be optimized depending on process variations and operating conditions. This feature will become more important in the future because device fluctuations are expected to have a significant impact on circuit operations in advanced CMOS process technologies.

In Chapter 5, new array logic architectures using dual-rail multiple-threshold logic circuits and logic synthesis methods for their architectures were presented. Since the proposed architectures can be optimized in column-circuit level, it is also possible to build the proposed circuit into the proposed design techniques in the previous chapters and the conventional area-efficient design techniques, and thus further reductions of the number of product terms can be achieved. The advantages of the proposed architectures were demonstrated by using benchmark circuits, and the results show that the number of product terms of the conventional AND-OR-type PLA structure can be reduced by 22.4% on average by using the proposed architecture. Also, the number of product terms, chip area, and power-delay product of LCPLA can be reduced by 12.7%, 11.5%, and 13.9% on average, respectively, by using a hybrid architecture of the proposed circuit and LCPLA in a 0.35-μm CMOS technology. The architecture of a field-programmable logic array based on the multiple-threshold logic circuits was also developed and is shown to be effective to implement Boolean functions efficiently. Also, as an application of the multiple-threshold logic circuit, we proposed a new functional memory with the capability of high-speed Hamming-distance-based data search. A 32 $\times$ 128-bit memory macro was designed and fabricated using a 0.35-μm CMOS technology, and high-speed and low-power operations were achieved. Also, we verified a search function of stored data within a Hamming distance of up to 7 from an input data.

As can be seen from the above results, the proposed logical and physical design techniques are effective in terms of area, speed, power consumption, and noise immunity for the design of CMOS array logic architectures, and can contribute to the development of

CMOS logic design. Along with these advantages and the high predictability of area, speed, power consumption, and signal integrity, the proposed architectures are strong candidates as preferable design methodologies in future IC designs.

Bibliography

[1] *International Technology Roadmap for Semiconductors (ITRS).* SIA, Sematech, Inc., 1999.

[2] R. Bryant *et al.*, "Limitations and challenges of computer-aided design technology for CMOS VLSI," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 341–365, Mar. 2001.

[3] S. Posluszny *et al.*, "Design methodology for a 1.0 GHz microprocessor," *Proceedings of IEEE International Conference on Computer Design (ICCD)*, 1998, pp. 17–23.

[4] S. Posluszny *et al.*, ""Timing closure by design," a high frequency microprocessor design methodology," *Proceedings of IEEE/ACM Design Automation Conference (DAC)*, 2000, pp. 712–717.

[5] S. Khatri, R. Brayton, and A. Sangiovanni-Vincentelli, "Cross-talk immune VLSI design using a network of PLAs embedded in a regular layout fabric," *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2000, pp. 412–418.

[6] L. Pileggi *et al.*, "Exploring regular fabrics to optimize the performance cost trade-off," *Proceedings of IEEE/ACM Design Automation Conference (DAC)*, 2003, pp. 782–787.

[7] H. Fleisher and L. Maissel, "An introduction to array logic," *IBM Journal of Research and Development*, vol. 19, pp. 98–109, Mar. 1975.

[8] R. Brayton, G. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, Boston, MA, 1984.

[9] UCB Design Technology Warehouse, http://www-cad.eecs.berkeley.edu/Software/software.html.

[10] W. Chen, *The VLSI Handbook*. CRC Press, 2000.

[11] Y. Chang, D. Wong, and C. Wong, "Programmable logic devices," *Encyclopedia of Electrical and Electronics Engineering*, 1999.

[12] S. Brown and J. Rose, "Architecture of FPGAs and CPLDs: a tutorial," *IEEE Design and Test of Computers*, vol. 13, no. 2, pp. 42–57, 1996.

[13] J. Cong and Y. Ding, "Combinational logic synthesis for LUT based field programmable gate arrays," *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, no. 2, pp. 145–204, Apr. 1996.

[14] T. Kuroda and T. Sakurai, "Overview of low-power ULSI circuit techniques," *IEICE Transactions on Electronics*, vol. E78-C, no. 4, pp. 334–344, Apr. 1995.

[15] M. Matsui *et al.*, "200 MHz video compression macrocells using low-swing differential logic," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1994, pp. 76–77.

[16] M. Matsui *et al.*, "A 200 MHz 13 mm^2 2-D DCT macrocell using sense-amplifying pipeline flip-flop scheme," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1482–1490, Dec. 1994.

[17] H. Huang, K. Cheng, Y. Chu, and C. Wu, "True-single-phase all-N-logic differential logic (TADL) for very high-speed complex VLSI," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 1996, vol. 4, pp. 296–299.

[18] H. Huang and Y. Chu, "Unbalanced current latch sense amplifier for low-power high-speed PLD's," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 1996, vol. 4, pp. 193–196.

[19] M. Khellah and M. Elmasry, "Use of charge sharing to reduce energy consumption in wide fan-in gates," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 1998, vol. 2, pp. 9–12.

[20] M. Khellah and M. Elmasry, "Low-power design of high-capacitive CMOS circuits using a new charge sharing scheme," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1999, pp. 286–287.

[21] B. Nikolić, V. Stojanović, V. Oklobdžija, W. Jia, J. Chiu, and M. Leung, "Sense amplifier-based flip-flop," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1999, pp. 282–283.

[22] B. Kong, J. Im, Y. Kim, S. Jang, and Y. Jun, "Asynchronous sense differential logic," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1999, pp. 284–285.

[23] J. King, S. Lee, and H. Park, "A sense amplifier-based CMOS flip-flop with an enhanced output transition time for high-performance microprocessors," *Proceedings of International Conference on VLSI and CAD (ICVC)*, 1999, pp. 384–387.

[24] O. Takahashi, N. Aoki, J. Silberman, and S. Dhong, "A 1-GHz logic circuit family with sense amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, pp. 616–622, May 1999.

[25] G. Kristovski and Y. Pogrebnoy, "New sense amplifier for small-swing CMOS logic circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 6, pp. 573–576, June 2000.

[26] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective, 2nd edition*. Addison-Wesley, Boston, MA, 1993.

[27] H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic synthesis for PLA with 2-input logic elements," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002, vol. 3, pp. 373–376.

[28] R. Rudell and A. Sangiovanni-Vincentelli, "Multiple-valued minimization for PLA optimization," *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, no. 5, pp. 727–750, Sept. 1987.

[29] T. Sasao, *Switching Theory for Logic Synthesis*. Kluwer Academic Publishers, Boston, MA, 1999.

[30] M. Schmookler, "Design of large ALUs using multiple PLA macros," *IBM Journal of Research and Development*, vol. 24, pp. 2–14, Jan. 1980.

[31] Synopsys, Inc., http://www.synopsys.com.

[32] VLSI Design and Education Center (VDEC), http://www.vdec.u-tokyo.ac.jp/English/index.html.

[33] G. Blair, "PLA design for single-clock CMOS," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1211–1213, Aug. 1992.

[34] Y. Dhong and C. Tsang, "High speed CMOS POS PLA using predischarged OR array and charge sharing AND array," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, pp. 557–564, Aug. 1992.

[35] C. Wang, C. Wu, R. Hwang, and C. Kao, "A low-power and high-speed dynamic PLA circuit configuration for single-clock CMOS," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, pp. 857–861, July 1999.

[36] *Design Data Translators Reference*. Cadence Design Systems, Inc., http://www.cadence.com.

[37] M. Suzuoki *et al.*, "A microprocessor with a 128-bit CPU, ten floating-point MAC's, four floating-point dividers, and an MPEG-2 decoder," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 11, pp. 1608–1618, Nov. 1999.

[38] N. Ide *et al.*, "2.44-GFLOPS 300-MHz floating-point vector-processing unit for high-performance 3-D graphics computing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1025–1033, July 2000.

[39] P. Lewis and C. Coates, *Threshold Logic*. John Wiley & Sons, NY, 1967.

[40] S. Muroga, *Threshold Logic and Its Applications*. John Wiley & Sons, NY, 1971.

[41] T. Sasao, "Input variable assignment and output phase optimization of PLA's," *IEEE Transactions on Computers*, vol. C-33, no. 10, pp. 879–894, Oct. 1984.

[42] T. Ogura and M. Nakanishi, "CAM-based highly-parallel image processing hardware," *IEICE Transactions on Electronics*, vol. E80-C, no. 7, pp. 868–874, July 1997.

[43] C. Dou and M. Shieh, "A CAM-based VLSI architecture for shared buffer ATM switch with fuzzy controlled buffer management," *Proceedings of IEEE International Conference on Computer Design (ICCD)*, 1996, pp. 149–152.

[44] C. Dou, S. Jiang, and K. Leu, "A novel CAM/RAM based buffer manager for next generation IP routers," *Proceedings of International Workshop on Electronic Design, Test and Applications (DELTA)*, 2002, pp. 111–115.

[45] J. Mulderz, N. Quach, and M. Flynn, "An area model for on-chip memories and its application," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 2, pp. 98–106, Feb. 1991.

[46] T. Ogura, J. Yamada, S. Yamada, and M. Tanno, "A 20-kbit associative memory LSI for artificial intelligence machines," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 4, pp. 1014–1020, Aug. 1989.

[47] T. Miwa, H. Yamada, Y. Hirota, T. Satoh, and H. Hara, "A 1-Mb 2-Tr/b nonvolatile CAM based on flash memory technologies," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1601–1609, Nov. 1996.

[48] C. Zukowski and S. Wang, "Use of selective precharge for low-power content-addressable memories," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 1997, vol. 3, pp. 1788–1791.

[49] F. Shafai, K. Schultz, G. Gibson, A. Bluschke, and D. Somppi, "Fully parallel 30-MHz, 2.5-Mb CAM," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1690–1696, Nov. 1998.

[50] T. Ikenaga and T. Ogura, "A fully parallel 1-Mb CAM LSI for real-time pixel-parallel image processing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 4, pp. 536–544, Apr. 2000.

[51] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 6, pp. 956–968, June 2001.

[52] T. Yamashita, T. Shibata, and T. Ohmi, "Neuron MOS winner-take-all circuit and its application to associative memory," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1993, pp. 236–237.

[53] M. Nagata, T. Yoneda, D. Nomasaki, M. Sano, and A. Iwata, "A minimum-distance search circuit using dual-line PWM signal processing and charge-packet," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 1997, pp. 42–43.

[54] M. Ikeda and K. Asada, "Time-domain minimum-distance detector and its application to low-power coding scheme on chip-interface," *Proceedings of European Solid-State Circuits Conference (ESSCIRC)*, 1998, pp. 464–467.

[55] H. Mattausch, T. Gyohten, Y. Soda, and T. Koide, "An architecture for compact associative memories with deca-ns nearest-match capability up to large distances," *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, 2001, pp. 170–171.

[56] M. Schneider and S. Chang, "A robust content based digital signature for image authentication," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 1996, vol. 3, pp. 227–230.

[57] S. Antani and L. Agnihotri, "Gujarati character recognition," *Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, 1999, pp. 418–421.

[58] M. Negin, T. Chmielewski, M. Salganicoff, U. Seelen, P. Venetainer, and G. Zhang, "An iris biometric system for public and personal use," *IEEE Computer*, vol. 33, no. 2, pp. 70–75, Feb. 2000.

List of Publications

Technical Journals

1. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed PLA using dynamic array logic circuits with latch sense amplifiers," *IEICE Transactions on Electronics*, vol. E84-C, no. 9, pp. 1240–1246, Sept. 2001.

2. H. Yamaoka, H. Yoshida, M. Ikeda, and K. Asada, "A logic-cell-embedded PLA (LCPLA): an area-efficient dual-rail array logic architecture," to be published in *IEICE Transactions on Electronics*, vol. E87-C, no. 2, pp. 238–245, Feb. 2004.

3. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed and area-efficient dual-rail PLA using divided and interdigitated column circuits," to be published in *IEICE Transactions on Electronics*, 2004.

Proceedings of International Conferences

1. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed PLA using array logic circuits with latch sense amplifiers and a charge sharing scheme," *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2001, pp. 3–4.

2. H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic synthesis for AND-XOR-OR sense-amplifying PLA," *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC) and International Conference on VLSI Design*, 2002, pp. 166–171.

3. H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic synthesis for PLA with 2-input logic elements," *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002, vol. 3, pp. 373–376.

4. H. Yamaoka, H. Yoshida, M. Ikeda, and K. Asada, "A dual-rail PLA with 2-input logic cells," *Proceedings of European Solid-State Circuits Conference (ESSCIRC)*, 2002, pp. 203–206.

5. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed functional memory with a capability of Hamming-distance-based data search by dynamic threshold logic circuits," *Proceedings of European Solid-State Circuits Conference (ESSCIRC)*, 2002, pp. 667–670.

6. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed logic circuit family with interdigitated array structure for deep sub-micron IC design," *Proceedings of European Solid-State Circuits Conference (ESSCIRC)*, 2003, pp. 189–192.

7. H. Yamaoka, M. Ikeda, and K. Asada, "PLA-based logic design using dual-rail threshold logic circuits for deep sub-micron ICs," submitted to *IEEE/ACM Design Automation Conference (DAC)*, 2004.

Proceedings of Domestic Conferences and Technical Society Meetings

1. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed PLA with latch sense amplifiers," *Proceedings of IEICE General Conference*, C-12-16, p. 111, Mar. 2000 (in Japanese).

2. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed PLA with latch sense amplifiers," *Proceedings of IEICE System LSI Biwako Workshop*, pp. 223–226, Nov. 2000 (poster in Japanese).

3. H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic synthesis for XOR-based dual-rail PLA," *Proceedings of IPSJ DA Symposium*, pp. 31–36, July 2001 (in Japanese).

4. H. Yamaoka and K. Asada, "A threshold logic-based high-speed Hamming distance detector and its evaluation," *IEICE Technical Report*, SDM2001-134, pp. 37–42, Aug. 2001 (in Japanese).

5. H. Yoshida, H. Yamaoka, M. Ikeda, and K. Asada, "Logic synthesis for PLA with 2-input logic elements," *IEICE Technical Report*, CPSY2001-72, pp. 67–72, Nov. 2001 (in Japanese).

6. H. Yamaoka, M. Ikeda, and K. Asada, "A dual-rail PLA with 2-input logic cells," *IEICE Technical Report*, SDM2002-156, pp. 19–24, Aug. 2002 (in Japanese).

7. U. Ekinciel, H. Yamaoka, M. Ikeda, and K. Asada, "A module generator for a dual-rail PLA with 2-input logic cells," *Proceedings of IEICE Society Conference*, C-3-7, Sept. 2002.

8. U. Ekinciel, H. Yamaoka, M. Ikeda, and K. Asada, "Performance driven design methodology for a dual-rail PLA with 2-input logic cells," *IEICE Technical Report*, VLD2002-152, pp. 37–42, Mar. 2003.

9. H. Yamaoka, M. Ikeda, and K. Asada, "A high-speed functional memory with a capability of Hamming-distance-selective data search using threshold logic circuits," *IEICE Technical Report*, VLD2002-151, pp. 31–36, Mar. 2003 (in Japanese).

Awards

1. H. Yamaoka, H. Yoshida, U. Ekinciel, and K. Asada, "A module generator for a dual-rail PLA with 2-input logic cells," *Nikkei-BP LSI IP Design Award (Challenge Award)*, May 2002.

2. H. Yamaoka, H. Yoshida, U. Ekinciel, and K. Asada, "A module generator for a dual-rail PLA with 2-input logic cells," *Nikkei-BP LSI IP Design Award (IP Award)*, June 2003.