

Fine-Grain Multiple-Valued Reconfigurable VLSI Architecture Based on High Utilization of Hardware Resources

著者	Bai Xu
学位授与機関	Tohoku University
学位授与番号	11301甲第15922号
URL	http://hdl.handle.net/10097/58719

Doctoral Thesis

Fine-Grain Multiple-Valued Reconfigurable VLSI
Architecture Based on High Utilization of
Hardware Resources

(ハードウェアリソースの高稼働率化に基づく細粒度
多値リコンフィギャラブル VLSI アーキテクチャ)

Xu Bai

Intelligent Integrated Systems Laboratory

Department of Computer and Mathematical Sciences

Graduate School of Information Sciences

Tohoku University, Japan

January, 2014

Contents

1	Introduction	6
2	High-performance multiple-valued logic block using current-source-sharing differential-pair circuits	14
2.1	Overview	14
2.2	Review of the multiple-valued fine-grain reconfigurable VLSI	15
2.3	Binary-controlled current-steering technique	22
2.3.1	Review of the MOS current-mode logic	22
2.3.2	Design of the binary-controlled differential-pair circuit	24
2.3.3	Evaluation of the binary-controlled differential-pair circuit	36
2.4	Current-source sharing technique	47

2.4.1	Design of the current-source-sharing differential-pair circuit	47
2.4.2	Evaluation of the current-source-sharing differential-pair circuit	60
2.5	Dual-supply voltage technique for low-power multiple-valued source-coupled logic circuits	63
2.6	Design and evaluation of the multiple-valued cell using current-source-sharing differential-pair circuits	69
2.7	Conclusion	86
3	Area-efficient switch block based on a multiple-valued X-net data transfer scheme	88
3.1	Overview	88
3.2	Multiple-valued fine-grain reconfigurable VLSI based on the binary X-net data transfer scheme	89
3.3	Multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme	98
3.4	Evaluation of the multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme	102
3.5	Conclusion	111

4	High-performance long-distance data transfer using a dynamic tree network	112
4.1	Overview	112
4.2	Long-distance data transfer in the multiple-valued reconfigurable VLSI using only the X-net network	113
4.3	Design of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network	117
4.4	Evaluation of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network	123
4.5	Conclusion	129
5	Conclusion	130
	Bibliography	134
	Acknowledgment	140

Chapter 1

Introduction

A key challenge in the integrated circuit (IC) scaling era is delivering high-performance solutions while minimizing power and cost. Programmable logic devices such as field-programmable gate arrays (FPGAs) are cost-effective from low- to mid-volume applications because functions and interconnections of logic resources can be directly programmed by end users[1]. Figure 1.1 shows the architecture of the conventional FPGA composed of logic blocks, switch blocks and connection blocks. Despite their design cost advantage, it is well understood that FPGAs suffer in terms of area, performance and power consumption relative to full-custom ICs because of the extremely complex switch blocks and connection blocks. The overhead incurred to make FPGAs both general purpose and field-programmable often limits integrations of FPGAs into real-world intelligent systems such as mobile phones,

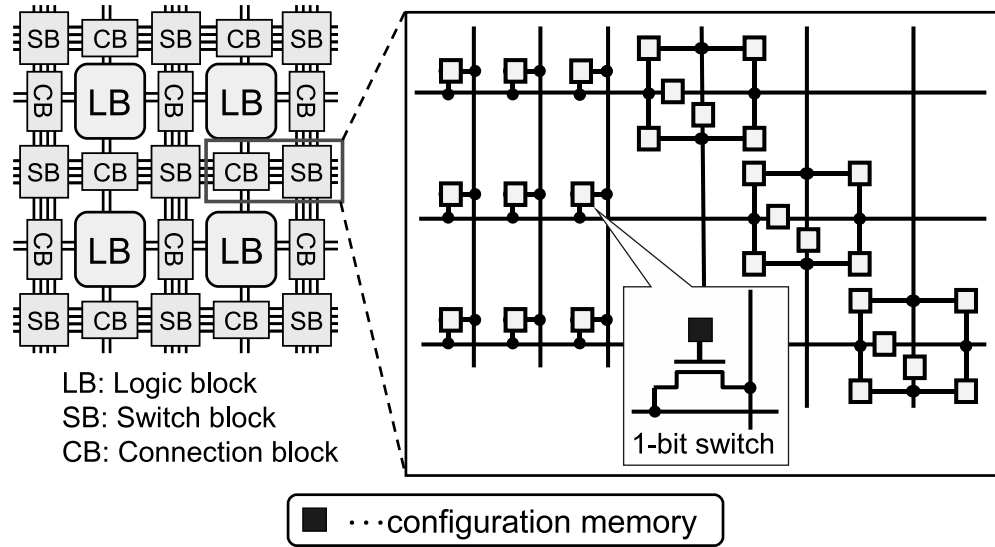


Figure 1.1: Architecture of the conventional Field-Programmable Gate Arrays (FPGA)

digital cameras, televisions, robots and vehicles[2].

A multiple-valued fine-grain reconfigurable VLSI (MVFG-RVLSI) using an eight nearest-neighbor mesh network (8-NNM) shown in Fig. 1.2 has been proposed to solve these problems[3][4]. Fine-grain pipelining and high utilization of a cell make the performance and parallelism high, respectively[5][6]. Also, localized data transfer architecture and multiple-valued signaling are effectively employed for reduction of switch blocks[3]. Moreover, in the multiple-valued reconfigurable VLSI cell, a quaternary-controlled differential-pair circuit is shared as common hardware resource to generate a full-adder sum or implement an arbitrary 2-variable binary function to realize a compact logic block[7].

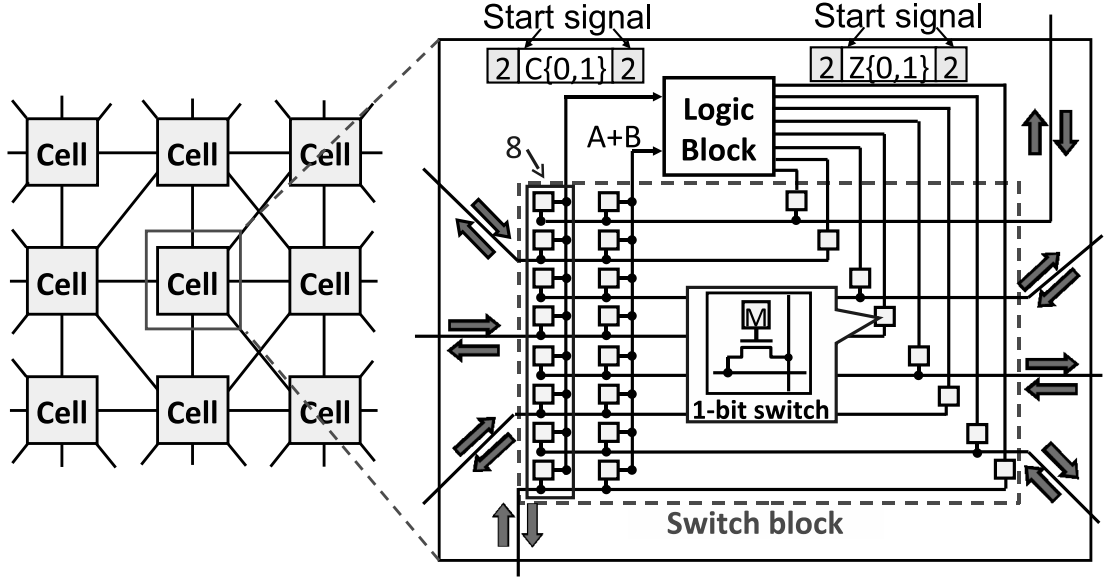


Figure 1.2: Architecture of the multiple-valued fine-grain reconfigurable VLSI using an eight nearest-neighbor mesh network

However, there are still many problems which limit the integration of the MVFG-RVLSI into the real-world intelligent systems. The first problem is that the MVFG-RVLSI cell has relative lower speed and larger power consumption in comparison with an equivalent CMOS reconfigurable VLSI cell of the same architecture[7]. A binary-to-quaternary converter composed of two differential-pair circuits (DPCs) is utilized to generate the quaternary signal for the quaternary-controlled differential-pair circuit, which results in low speed and large power consumption. The quaternary-controlled differential-pair circuit using a fixed reference voltage is not so fast due to the small voltage difference in the gate

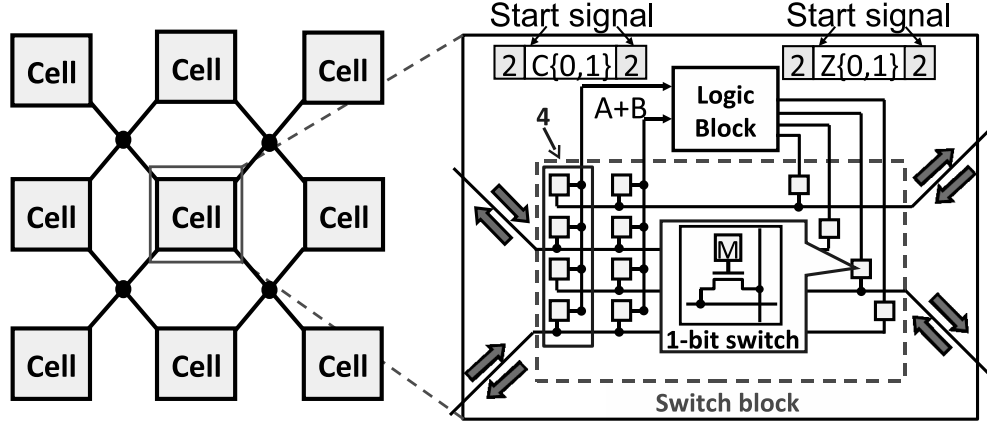


Figure 1.3: Architecture of the multiple-valued reconfigurable VLSI using the X-net

inputs. Also, its noise margin is very small due to the small voltage difference of the dual-rail output. Moreover, to realize the fine-grain bit-serial pipelined operation, many current sources are used to implement current-mode D flip-flops (CMDFFs), which results in large power consumption.

To solve the first problem, Chapter 2 proposes three circuit-level techniques including a binary-controlled current-steering technique, a current-source sharing technique and a dual-supply voltage technique for multiple-valued source-coupled logic circuits.

The binary-controlled current-steering technique is introduced to use a three-level DPC to implement a high-performance arbitrary 2-variable binary function. Also, the voltage difference of the dual-rail output is larger than that of the previous quaternary-controlled differential-pair

circuit, which increases the noise margin. The power consumption and the delay can be greatly reduced without using the binary-to-quaternary converter. HSPICE simulation of the binary-controlled differential-pair circuit is done using a 65nm CMOS design rule. As a result, the delay, the power consumption and the area of the binary-controlled differential-pair circuit are reduced to 33%, 26% and 68%, respectively, in comparison with the previous quaternary-controlled differential-pair circuit with the binary-to-quaternary converter. In comparison with a conventional 2-input look-up table (LUT) [1], the delay and the area of the binary-controlled differential-pair circuit become 83% and 88%, respectively. Also, the binary-controlled differential-pair circuit has lower power consumption when the operating frequency is more than 1.6GHz.

In a proposed current-source-sharing differential-pair circuit, only one current source is shared to implement a logic function and store its result, so that the high utilization of the current source leads to low power consumption. Also, the delay is reduced because the sample stage in the current-mode D-latch is omitted. To demonstrate the advantage of the current-source sharing technique, HSPICE simulation of a current-source-sharing bit-serial adder is done using a 65nm CMOS design rule. The power consumption, the delay and the area of the proposed current-source-sharing bit-serial adder are reduced to 56%, 70% and 83%, re-

spectively, in comparison with the current-mode bit-serial adder. The area and the delay become 88% and 47% of those of the CMOS bit-serial adder, respectively. Also, the proposed current-source-sharing bit-serial adder has lower power consumption when the operating frequency is more than 1.09GHz.

In the dual- V_{DD} multiple-valued source-coupled logic (MVSCL) circuit, a current-voltage (I-V) converter is used to convert a multiple-valued current signal to a multiple-valued voltage signal, and a comparator implemented by the DPC is used to realize a threshold operation. In the I-V converter, V_{DDH} is required for multiple voltage levels V_{DDH} , $V_{DDH} - \Delta V$, \dots , $V_{DDH} - K \times \Delta V$ corresponding to multiple logic values “0”, “1”, \dots , “K”, respectively. In the DPC with a binary dual-rail voltage output, V_{DDL} is used for two voltage levels $V_{DDL} - \Delta V$ and V_{DDL} corresponding to two logic values “0” and “1”, respectively. The speed of the DPC is not decreased by V_{DDL} because it is independent of the supply voltage [9][10]. Moreover, it is different from the conventional dual- V_{DD} CMOS circuit that level shifters are not necessary to be provided to prevent direct-path currents in the dual- V_{DD} MVSCL circuit, because the current flow in the DPC is fixed by a current source.

The second problem is that the MVFG-RVLSI still has complex switch blocks in comparison with full-custom ICs. In the MVFG-RVLSI using

the 8-NNM shown in Fig. 1.2, each cell composed of a switch block and a logic block is connected to eight adjacent cells[8]. The switch block is not so compact because eight nMOS pass transistors and eight configuration memories are provided at each input/output (I/O) of the cell to realize an eight-near neighborhood data transfer.

To solve the second problem, Chapter 3 proposes a multiple-valued X-net data transfer shown in Fig. 1.3 for area-efficient switch blocks. An X-net network is more sufficient than the 8-NNM to realize the eight-near neighborhood data transfer[11]. The X-net network is employed for implementing area-efficient switch blocks without decreasing performance. In the X-net network, one cell is connected to four “X” intersections and each “X” intersection is connected to the other three adjacent cells. Therefore, only four nMOS pass transistors and four configuration memories are provided at each I/O of the cell to realize the eight-near neighborhood data transfer. The high utilization of the nMOS pass transistor and the configuration memories leads to the area-efficient switch block. Moreover, a multiple-valued data transfer scheme is proposed to realize the high utilization of the X-net network, where linear summation of current signals transferred between cells can be realized at each “X” intersection[12][13].

The third problem is that it is necessary to use many cells to real-

ize long-distance data transfer by the nearest-neighbor network, which results in low speed and large power consumption.

In Chapter 4, to solve the third problem, a global dynamic tree network is employed for high-performance bit-parallel long-distance data transfer. In practical applications such as a sum-of-absolute-difference operation (SAD) [12], the local X-net network is frequently used for inter-cell neighborhood data transfer, and the global dynamic tree network is occasionally used for long-distance data transfer between a cell and a data memory. Therefore, the global dynamic tree network is connected not to each cell, but to multiple-cell blocks composed of many cells. To realize highly parallel memory access, a logic-in-memory architecture is introduced, where data transfer between a local memory and the multiple-cell block can be done in each logic-in-memory element (LME). Moreover, to solve speed problems in comparison with a multiple bus and a crossbar network, pipelined switch nodes are provided to improve data transfer throughput.

Chapter 2

High-performance multiple-valued logic block using current-source-sharing differential-pair circuits

2.1 Overview

This chapter presents three circuit-level techniques for a high-speed low-power multiple-valued fine-grain logic block.

A binary-controlled current-steering technique is proposed to use a three-level differential-pair circuit for implementing a high-performance arbitrary two-variable binary function without using a binary-to-quaternary converter.

A current-source-sharing technique is proposed to improve utilization of current sources for low power consumption. One current source can be shared to implement a logic function and store its result for low-power current-mode pipeline.

A dual-supply voltage technique is proposed for low-power multiple-valued source-coupled logic circuits without increasing delay. A high supply voltage is used for multiple voltage levels, and a low supply voltage is used for binary voltage levels. In the differential-pair circuits (DPCs) using the low supply voltage, the delay is not increased because it is independent of the supply voltage[9][10].

As a result, in comparison with the previous multiple-valued cell, the power consumption and the delay of the proposed multiple-valued cell are reduced to 49% and 72%, respectively, without increasing the area and the configuration memory size.

2.2 Review of the multiple-valued fine-grain reconfigurable VLSI

In the multiple-valued fine-grain reconfigurable VLSI (MVFG-RVLSI) using the eight-nearest neighbor mesh network (8-NNM) shown in Fig. 1.2, each cell is composed of a logic block and a switch block, and can be

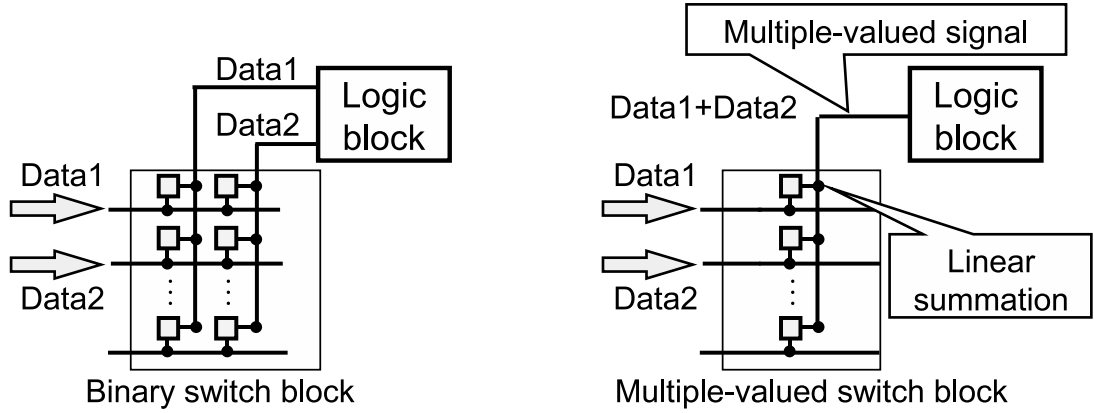


Figure 2.1: Compact multiple-valued switch block

connected to its eight adjacent cells through one-bit switches. Multiple-valued signaling is introduced to implement a compact switch block as shown in Fig. 2.1. In the binary switch block, if there are two binary inputs, 16 one-bit switches are necessary to control data transfer to the logic block. In the multiple-valued switch block, two binary current inputs linearly summed by wiring can be transferred on one line, therefore only eight one-bit switches are used to control data transfer. The complexity of the switch block can be reduced to half by the multiple-valued logic technique.

The behavioral description is given by a control/data flow graph. In the direct allocation of the control/data flow graph shown in Fig. 2.2, each node in the control/data flow graph corresponds to a macro-block in the MVFG-RVLSI and each edge corresponds to a data transfer path

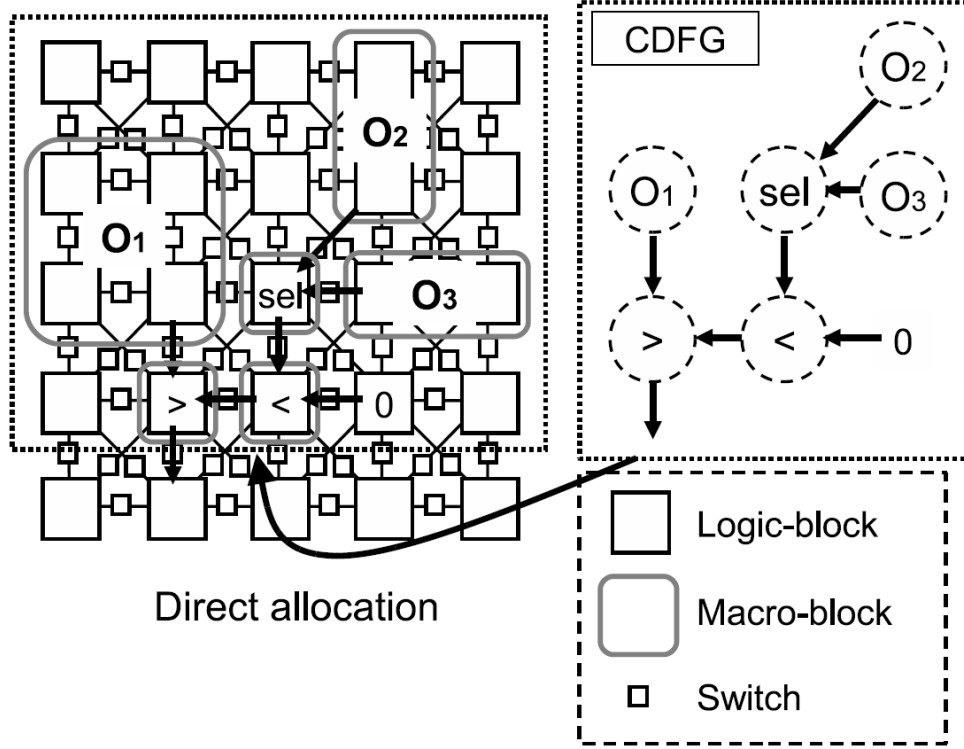


Figure 2.2: Direct allocation of a control/data flow graph

between the macro-blocks, where the macro-block consists of multiple cells. The complexity of logical connections between the macro-blocks becomes almost the same as that of the control/data flow graph. The architecture for the localized data transfer can be effectively employed for reducing the complexity of interconnections and delay due to data transfer between cells[14].

As shown in Fig. 2.3, a multiple-valued cell has been proposed for the MVFG-RVLSI[7]. The cell consists of a multiple-valued switch block,

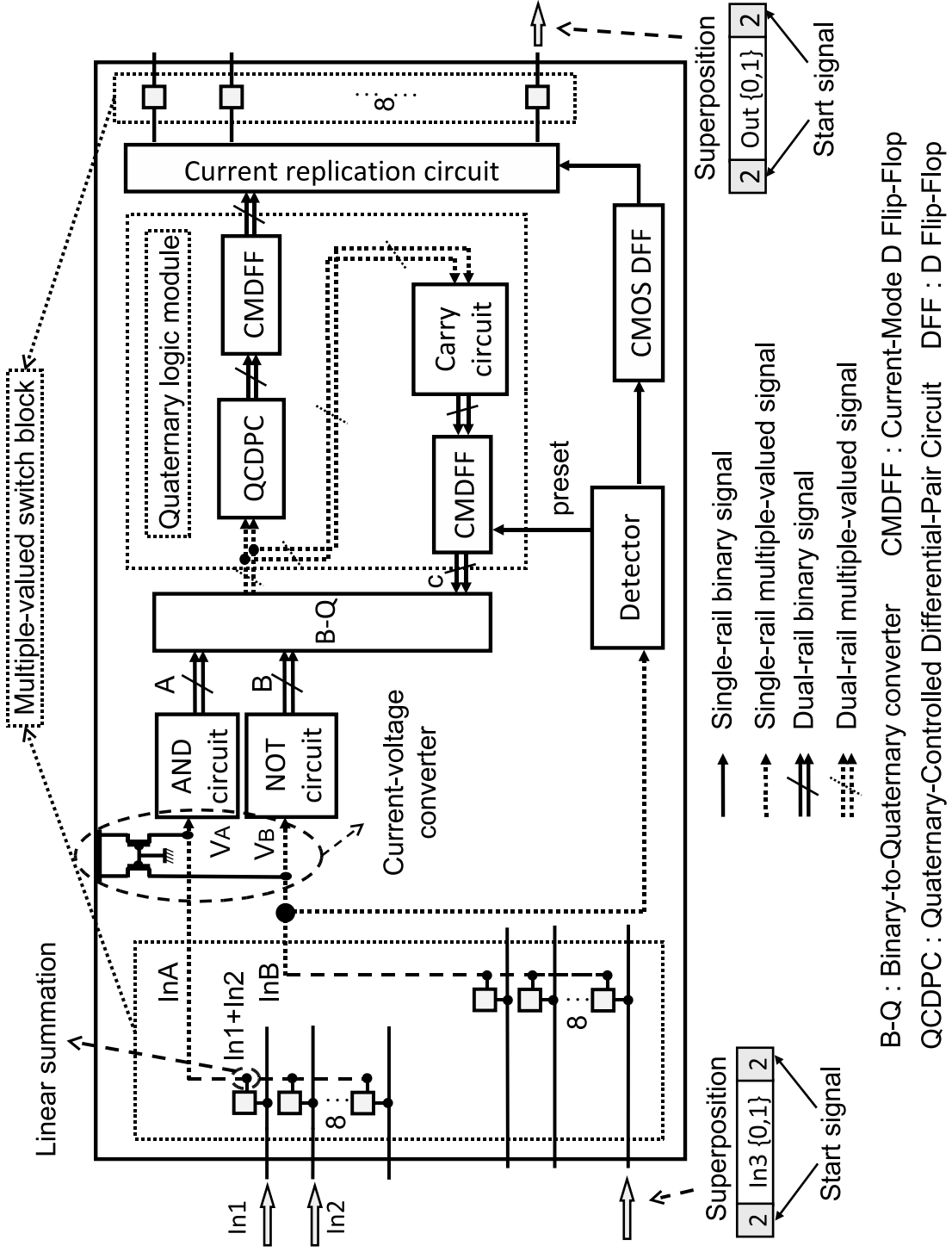
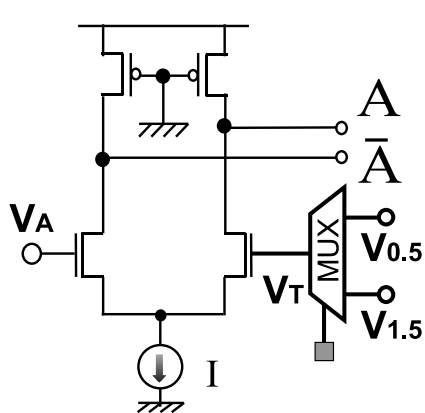
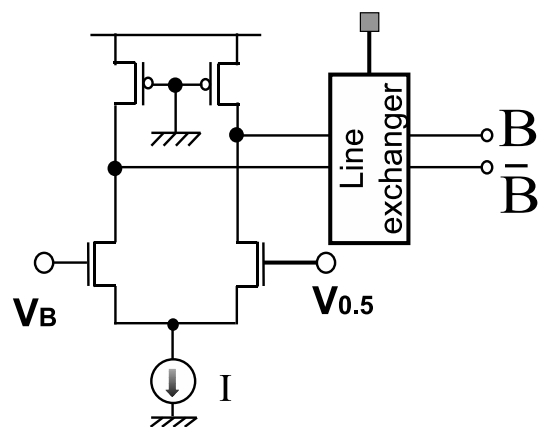


Figure 2.3: Multiple-valued cell using a quaternary-controlled differential-pair circuit

an AND circuit, a NOT circuit, a binary-to-quaternary converter, a quaternary logic module, and a current replication circuit. The inputs and outputs of the cell are represented by single-rail binary current signals. In the multiple-valued switch block, the binary current inputs In1 and In2 are linearly summed by wiring, so that an input InA of a current-voltage converter becomes a three-valued data. In a bit-serial operation, a start signal indicating a head of a one-word data is required to initialize the D flip-flops used for state memory. Superposition of the data and start signal in a single interconnection is introduced to realize compact switch blocks, where the logic value “2” is defined as the start signal to distinguish from data “0” and “1”. Both of the number of interconnections between the cells and the number of switches are reduced to half in comparison with those of a binary representation.



(a) AND circuit



(b) NOT circuit

Figure 2.4: Threshold logic circuits

Table 2.1: Programmable operations of the AND circuit

(a) Dual-rail code		(b) AND-type dual-rail code	
V_A	(x, \bar{x})	V_A	(x, \bar{x})
0	(0, 1)	0	(0, 1)
1	(1, 0)	1	(0, 1)
		2	(1, 0)

Table 2.2: Programmable operations of the NOT circuit

(a) Dual-rail code		(b) NOT-type dual-rail code	
V_B	(y, \bar{y})	V_B	(y, \bar{y})
0	(0, 1)	0	(1, 0)
1	(1, 0)	1	(0, 1)

Both the AND circuit and NOT circuit are constructed by a basic one-level differential-pair circuit shown in Fig. 2.4. In the AND circuit, two operations of Table 2.1 can be programmed. The AND-type dual-rail code is used to generate a partial product in a multiplication and the dual-rail code is used in other cases. In the NOT circuit, two operations of Table 2.2 can be programmed. The NOT-type dual-rail code is used to convert a subtrahend to a 2's complement number in a subtraction and the dual-rail code is used in other cases.

In the quaternary logic module composed of the quaternary-controlled differential-pair circuit, a quaternary-controlled carry circuit and two current-mode D-flip-flops, an arbitrary 2-variable binary function can be realized, and a bit-serial adder can be implemented. However, the binary-to-quaternary converter is required to convert the dual-rail binary voltage signals into a dual-rail quaternary voltage signal, which results in large power consumption and low speed. Also, the quaternary-controlled differential-pair circuit and the quaternary-controlled carry circuit are not so fast due to the small voltage difference in the gate inputs. To overcome the problems, I introduce the binary-controlled differential-pair circuits to implement the high-performance low-power arithmetic logic operations without using the binary-to-quaternary converter.

Also, the current-mode D-flip-flop composed of two current-mode D-latches is used as a register for the bit-serial pipelined operation. To reduce the power consumption of the register, the current-source sharing technique between a series-gating differential-pair circuit and a current-mode D-latch is introduced.

2.3 Binary-controlled current-steering technique

2.3.1 Review of the MOS current-mode logic

In the MVFG-RVLSI, MOS current-mode logic (MCML) is used to perform arithmetic logic operations. The MCML is a differential logic style, and in general consists of three parts which include a load resistor, the pull-down network (PDN) and a current source shown in Fig. 2.5[9].

The load resistor R is pMOS device with fixed gate voltage and is designed to be operated in triode (linear) region in order to model a resistor. The PDN is implemented with standard nMOS differential pairs which are operated in saturation region controlled by dual-rail binary voltage inputs. The current source is an nMOS device with a fixed gate voltage and is designed to be operated in the saturation region to produce relatively constant current.

The MCML does not provide a rail-to-rail output swing. The MCML

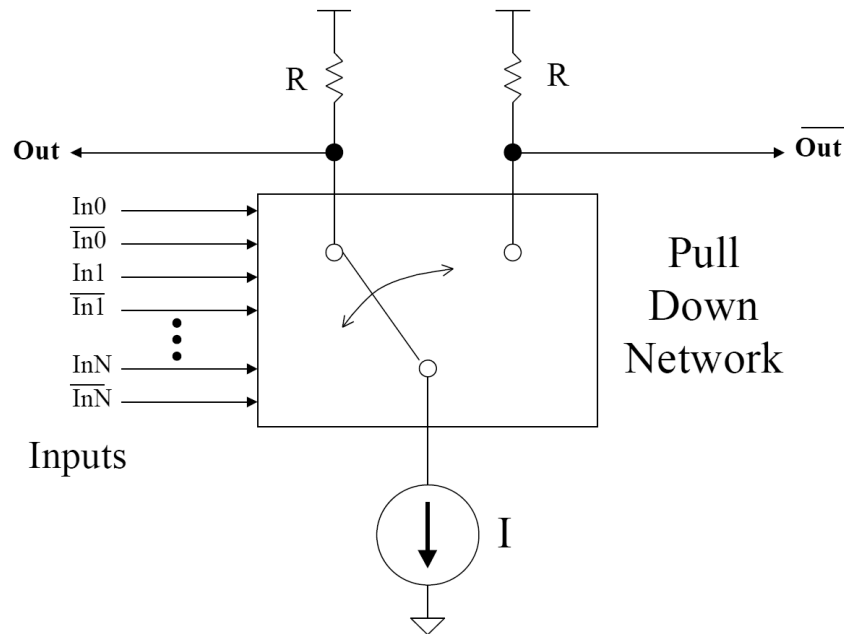


Figure 2.5: General MOS current-mode logic structure

circuits are faster than other logic families, because it uses nMOS transistors only. Due to its differential nature, it is highly immune to common mode noise. It has almost flat power curve over a wide range of frequency as opposed to other logic styles where power consumption increases directly with frequency. Therefore at very high frequencies its power consumption is lower than other logic styles. This makes it a good choice for high-speed and low-power integrated circuit design.

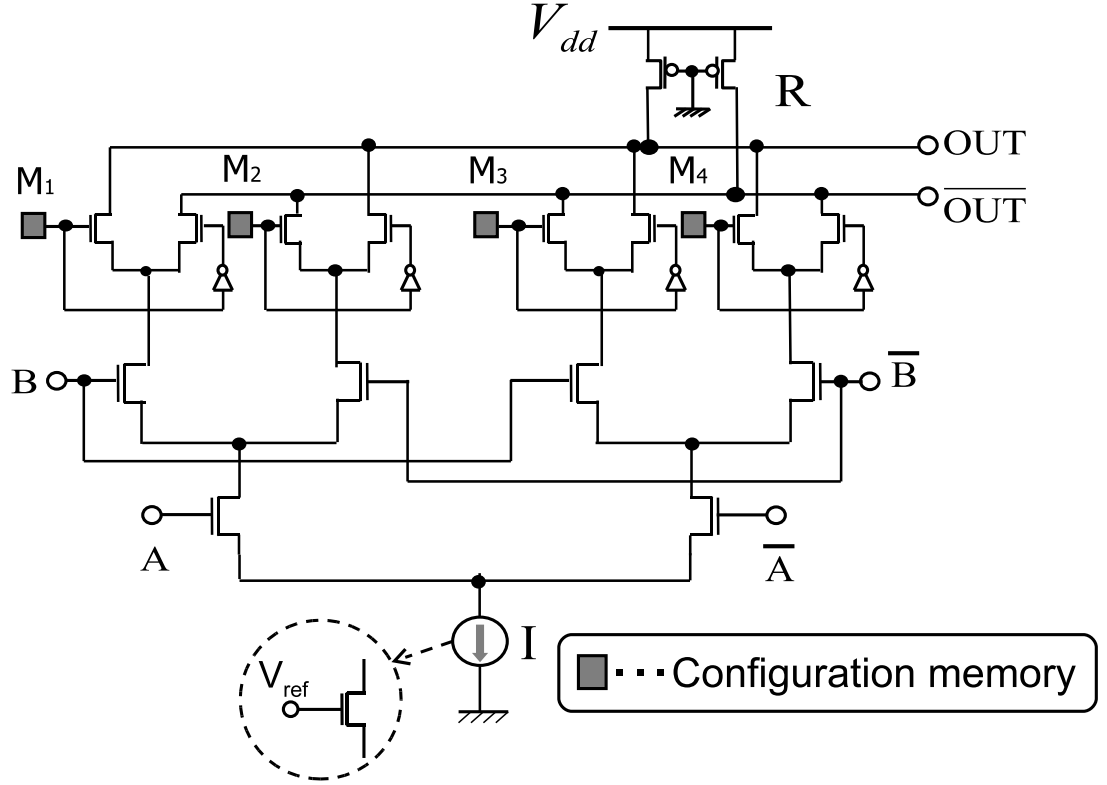


Figure 2.6: Binary-controlled differential-pair circuit

2.3.2 Design of the binary-controlled differential-pair circuit

As shown in Fig. 2.6, the binary-controlled differential-pair circuit is introduced to improve the performance as well as to reduce the power consumption. The dual-rail binary voltage signals generated by the AND circuit and the NOT circuit can be directly connected to the binary-controlled differential-pair circuit without using the binary-to-quaternary converter.

Only one current source constructed by an nMOS transistor in the

saturation region is necessary to drive the binary-controlled differential-pair circuit, which makes power consumption low. The current I produced by the current source is steered into one of the branches in the binary-controlled differential-pair circuit according to the dual-rail binary voltage inputs. The two values of a dual-rail output are V_{dd} and $V_{dd}-\Delta V$, where ΔV is the output voltage swing and is equals to $I \times R$. R is the equivalent resistance of the pMOS load transistor.

Configuration memories M_1, M_2, M_3 and M_4 are programmed to steer the current I flow through the third-level differential pairs for an arbitrary two-variable binary function shown in Table 2.3. The values of M_1, M_2, M_3 and M_4 and the corresponding function are shown in Table 2.4.

Table 2.3: Arbitrary two-variable binary function

A	B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2.4: Programming of an arbitrary two-variable binary function

Function	M_1	M_2	M_3	M_4
f_0	0	1	1	0
f_1	0	1	1	1
f_2	0	1	0	0
f_3	0	1	0	1
f_4	0	0	1	0
f_5	0	0	1	1
f_6	0	0	0	0
f_7	0	0	0	1
f_8	1	1	1	0
f_9	1	1	1	1
f_{10}	1	1	0	0
f_{11}	1	1	0	1
f_{12}	1	0	1	0
f_{13}	1	0	1	1
f_{14}	1	0	0	0
f_{15}	1	0	0	1

Figure 2.7 shows the input and output waveforms of the binary-controlled differential-pair circuit which is programmed to implement the two-

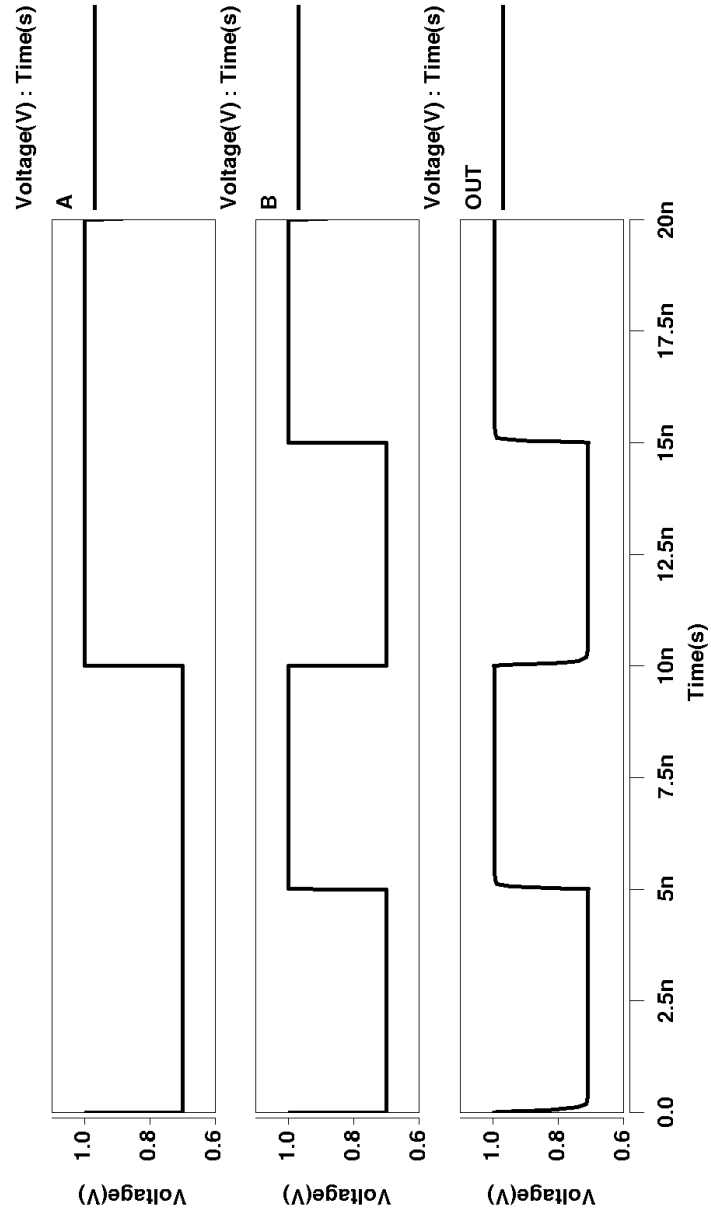


Figure 2.7: Input and output waveforms of the binary-controlled differential-pair circuit (The two-variable binary function f_5 in Table 2.3 is implemented)

variable binary function f_5 in Table 2.3. The configuration memories M_1, M_2, M_3 and M_4 are configured as “0”, “0”, “1” and “1”, respectively. If the input data (A,B) is (0,1) or (1,1), the *OUT* becomes 1.0V corresponding to the logic value “1”. On the other hand, if the input data (A,B) is (0,0) or (1,0), the *OUT* becomes 0.7V corresponding to the logic value “0”.

Noise margins represent “safety margins” that prevent the digital circuit from producing erroneous outputs in the presence of noisy inputs [15][16]. Figure 2.8 shows the previous 2-variable binary function circuit composed of the binary-to-quaternary converter and the quaternary-controlled differential-pair circuit. The quaternary-controlled differential-pair circuit is used to implement the quaternary universal literal realized by linear summation of two half-universal literals $H_1(S)$ and $H_2(S)$ shown in Fig. 2.9. In the quaternary-controlled differential-pair circuit, a complementary quaternary signal $(V_S, \overline{V_S})$ and fixed reference voltages V_{T1} and V_{T2} are applied, which results in the small voltage difference of the dual-rail output as shown in Fig. 2.10. The noise margin of the quaternary-controlled differential-pair circuit is not so large, due to the small output difference.

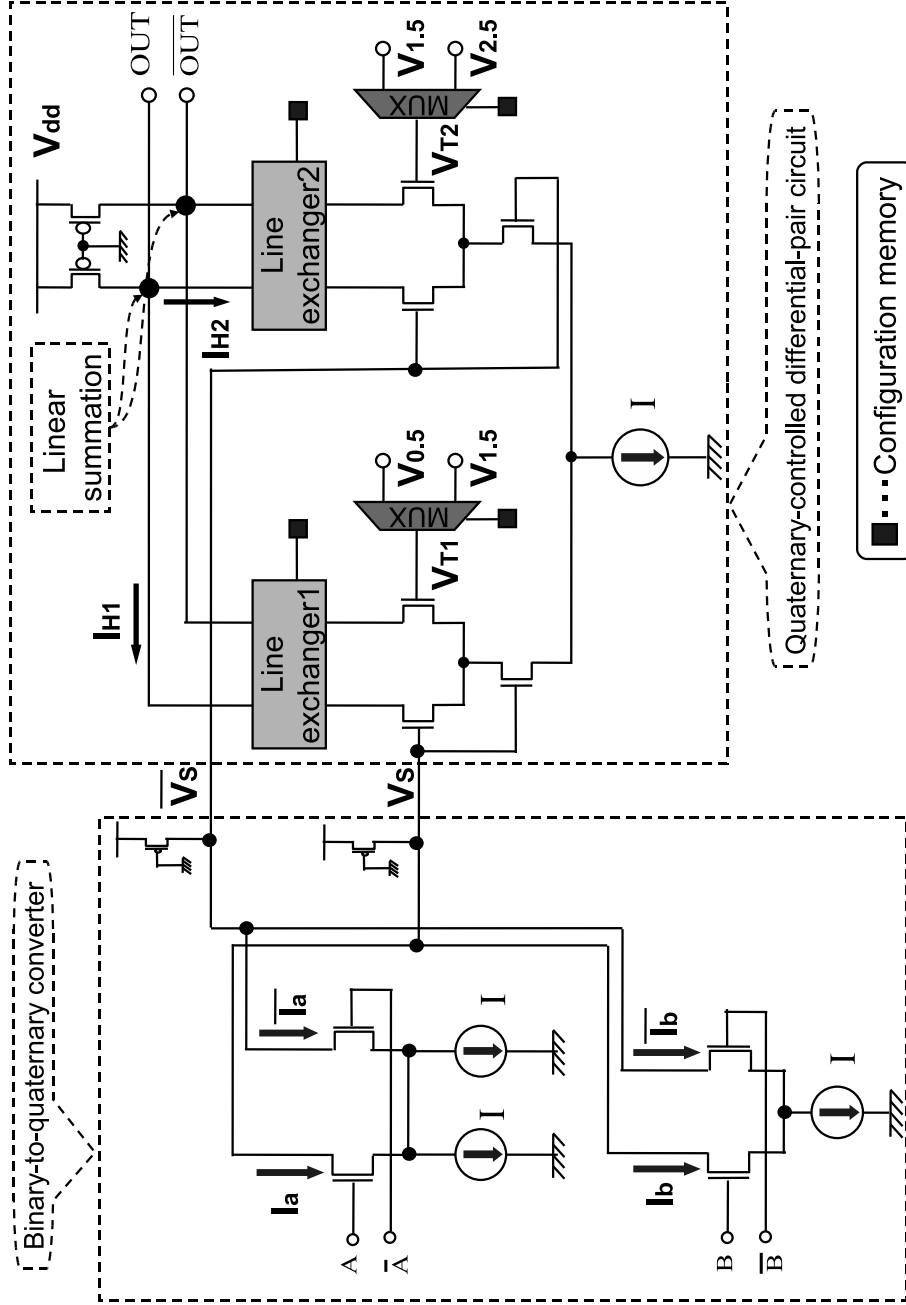


Figure 2.8: Two-variable binary function circuit composed of the binary-to-quaternary converter and the quaternary-controlled differential-pair circuit

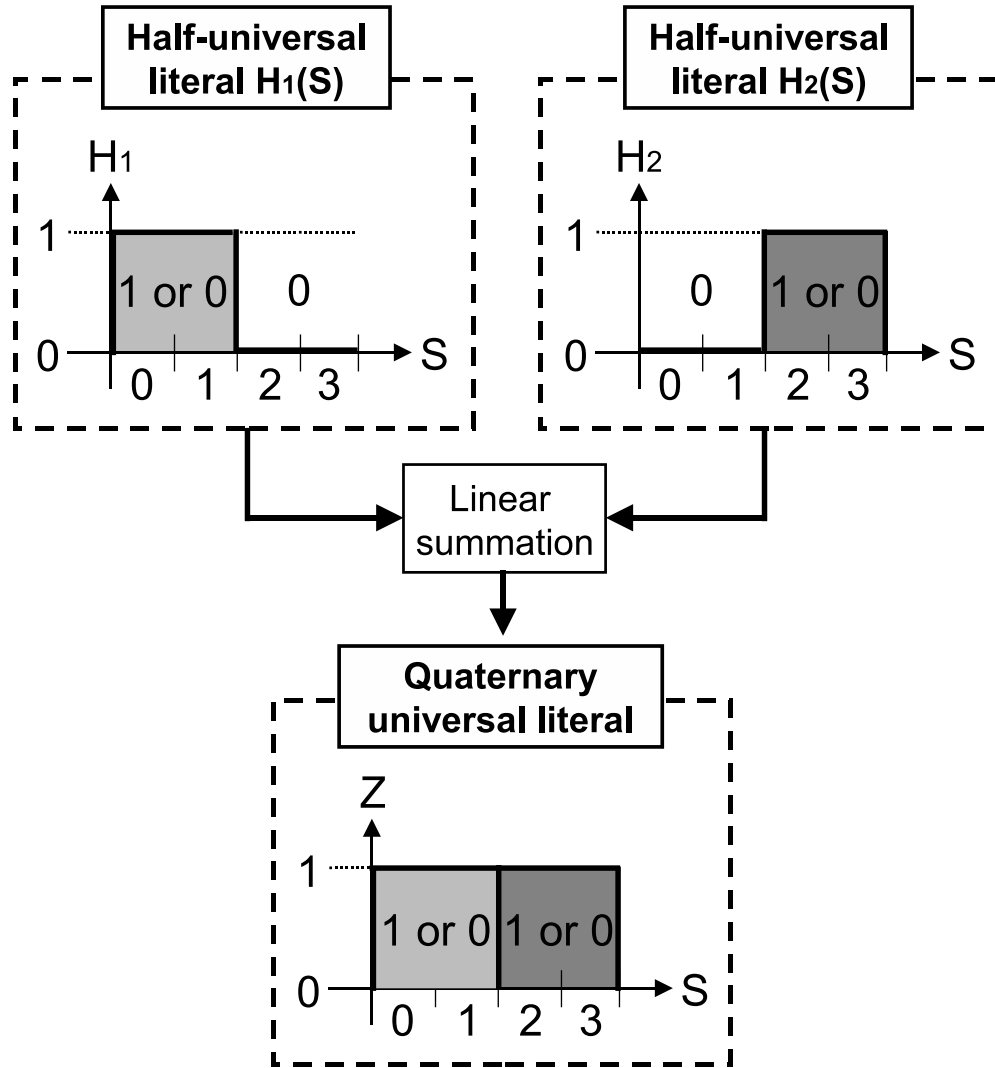


Figure 2.9: Design of a quaternary universal literal using two half-universal literals

In contrast, the dual-rail binary voltage signals generated by the AND circuit and the NOT circuit can be directly connected to the proposed binary-controlled differential-pair circuit. Therefore, the output difference becomes larger than that of the quaternary-controlled differential-pair circuit as shown in Fig. 2.11, which increases the noise margin. The

quaternary-controlled differential-pair circuit and the binary-controlled differential-pair circuit are programmed as inverters to measure noise margins. Noise margins are defined for high and low input levels use the following equations:

$$\text{High noise margin : } NM_H = V_{OH} - V_{IH} \quad (2.1)$$

$$\text{Low noise margin : } NM_L = V_{IL} - V_{OL}$$

where V_{OH} is the minimum allowable output voltage that can be recognized as logic “1”, V_{OL} is the maximum allowable output voltage that can be recognized as logic “0”, V_{IH} is the minimum allowable input voltage that can be recognized as logic “1”, and V_{IL} is the maximum allowable input voltage that can be recognized as logic “0”.

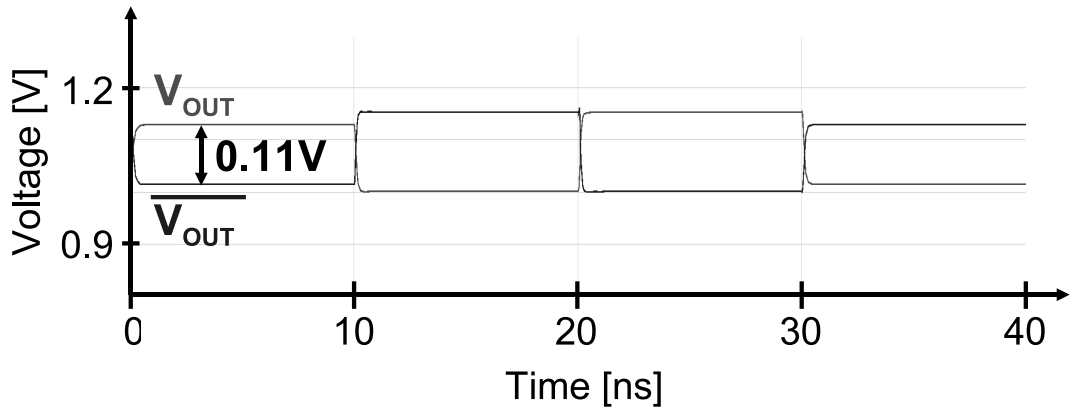


Figure 2.10: Dual-rail output waveform of the quaternary-controlled differential-pair circuit

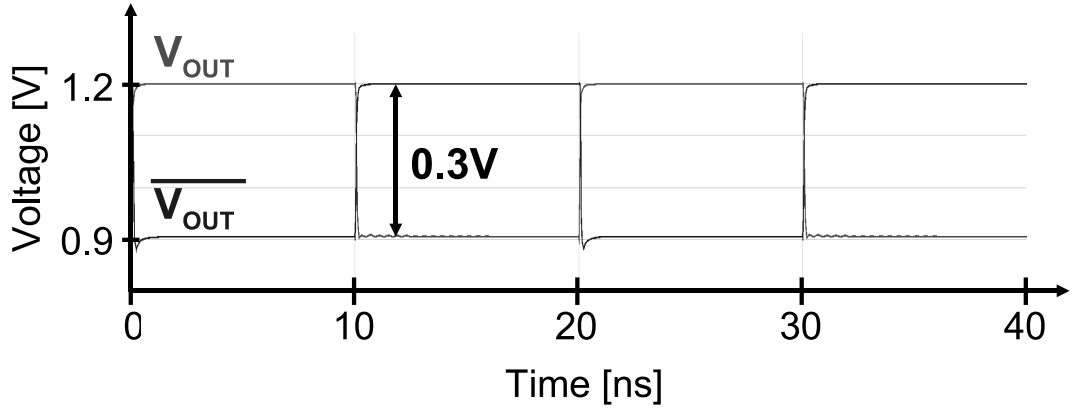


Figure 2.11: Dual-rail output waveform of the binary-controlled differential-pair circuit

Figure 2.12 shows the voltage transfer characteristic of the quaternary-controlled differential-pair circuit. V_{OH} , V_{OL} , V_{IH} and V_{IL} are 0.083V, -0.089V, 0.081V and -0.073V, respectively. Therefore, the NM_H and NM_L become 0.002V and 0.016V, respectively. Figure 2.13 shows the voltage transfer characteristic of the binary-controlled differential-pair circuit. V_{OH} , V_{OL} , V_{IH} and V_{IL} are 0.27V, -0.212V, 0.17V and -0.178V, respectively. Therefore, the NM_H and NM_L become 0.1V and 0.034V, respectively. The NM_H and NM_L in the binary-controlled differential-pair circuit are greatly increased in comparison with the quaternary-controlled differential-pair circuit.

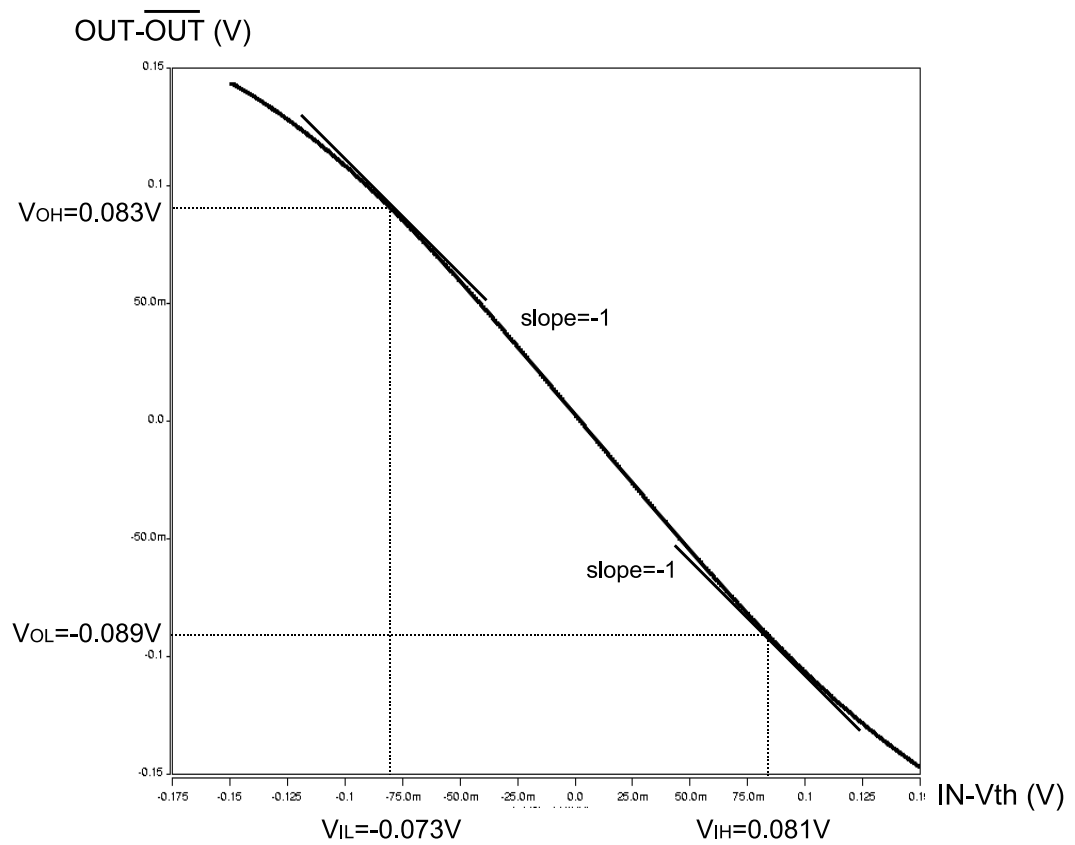


Figure 2.12: Output voltage versus input voltage in the quaternary-controlled differential-pair circuit

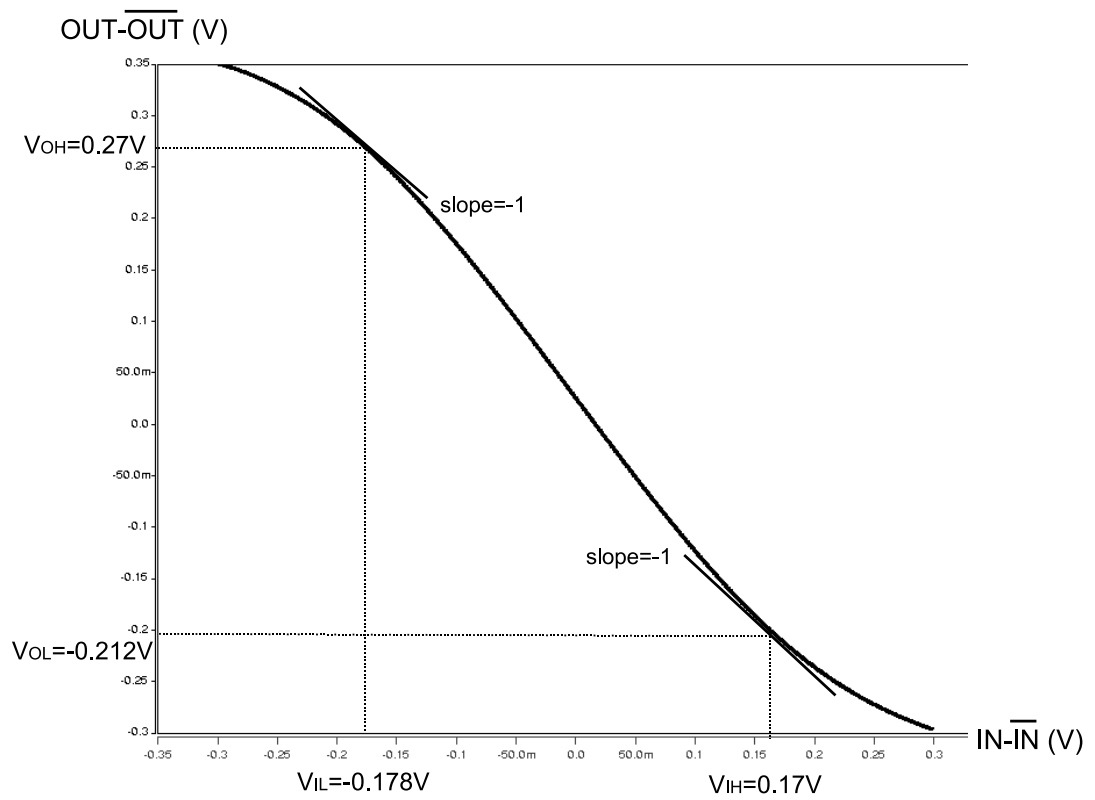


Figure 2.13: Output voltage versus input voltage in the binary-controlled differential-pair circuit

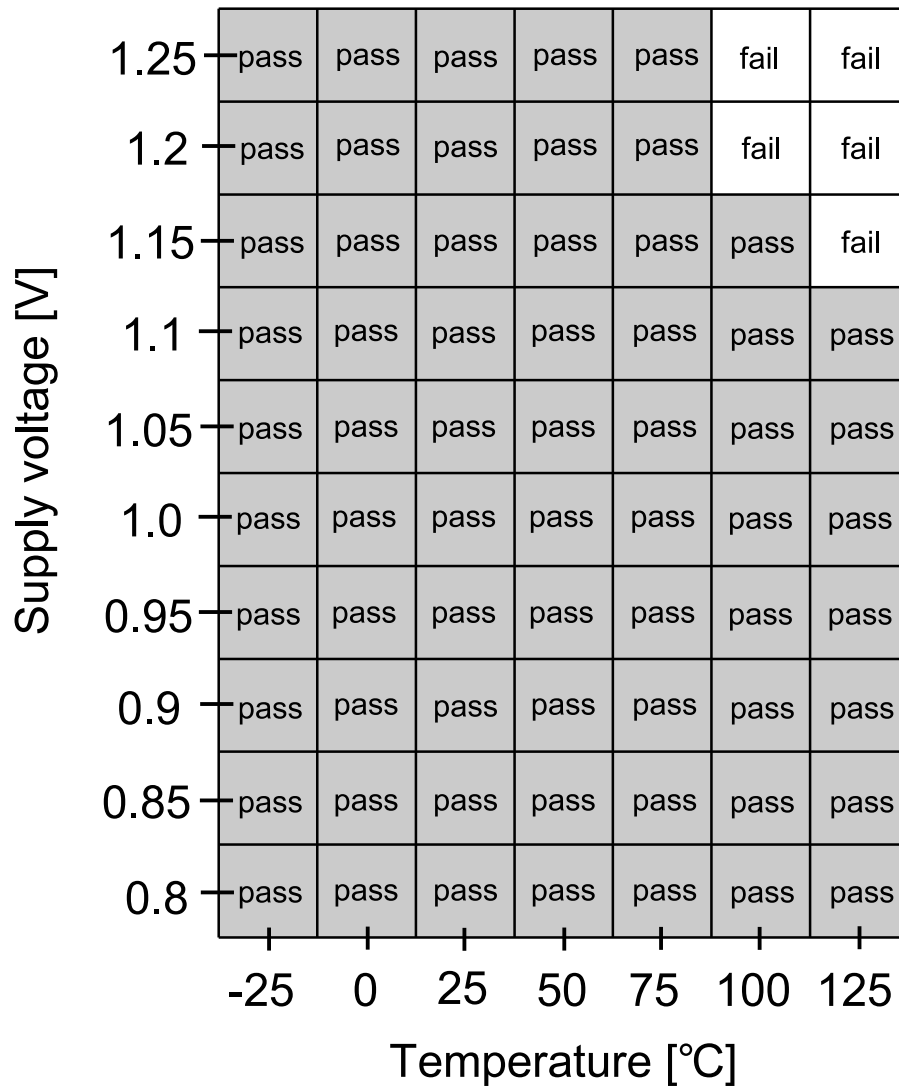


Figure 2.14: Shmoo plot of the binary-controlled differential-pair circuit under threshold-voltage, temperature, and supply voltage variations

Figure 2.14 shows a shmoo plot of the binary-controlled differential-pair circuit under threshold-voltage, temperature, and supply-voltage variations. The temperature and supply voltage vary from -25°C to 125°C and from 0.8V to 1.25V, respectively, and are set up for simulation in the binary-controlled differential-pair circuit uniformly. The range of threshold-voltage variation is $\pm 10\%$ of the threshold voltage, and is set up for simulation in each transistor randomly. As a result, the binary-controlled differential-pair circuit can work correctly from -25°C to 75°C and from 0.8V to 1.25 under $\pm 10\%$ threshold-voltage variation. The lower limit of the supply voltage is 0.6V.

2.3.3 Evaluation of the binary-controlled differential-pair circuit

The binary-controlled differential-pair circuit is fabricated using a 65 nm CMOS process. The supply voltage is 1.2V. Figure 2.15 shows the chip photomicrograph and the layout of the binary-controlled differential-pair circuit. The area is $53.36\mu\text{m}^2$.

Figure 2.16 shows the inputs and outputs waveforms of the binary-controlled differential-pair circuit in the chip. The circuit is programmed to implement the two-variable binary function f_3 in Table 2.3. The values of M_1 , M_2 , M_3 and M_4 are “0”, “1”, “0” and “1”, respectively. If the input data (a,b) is (0,0) or (0,1), the OUT becomes “0”. On the other

hand, if the input data (a,b) is (1,0) or (1,1), the OUT becomes “1”.

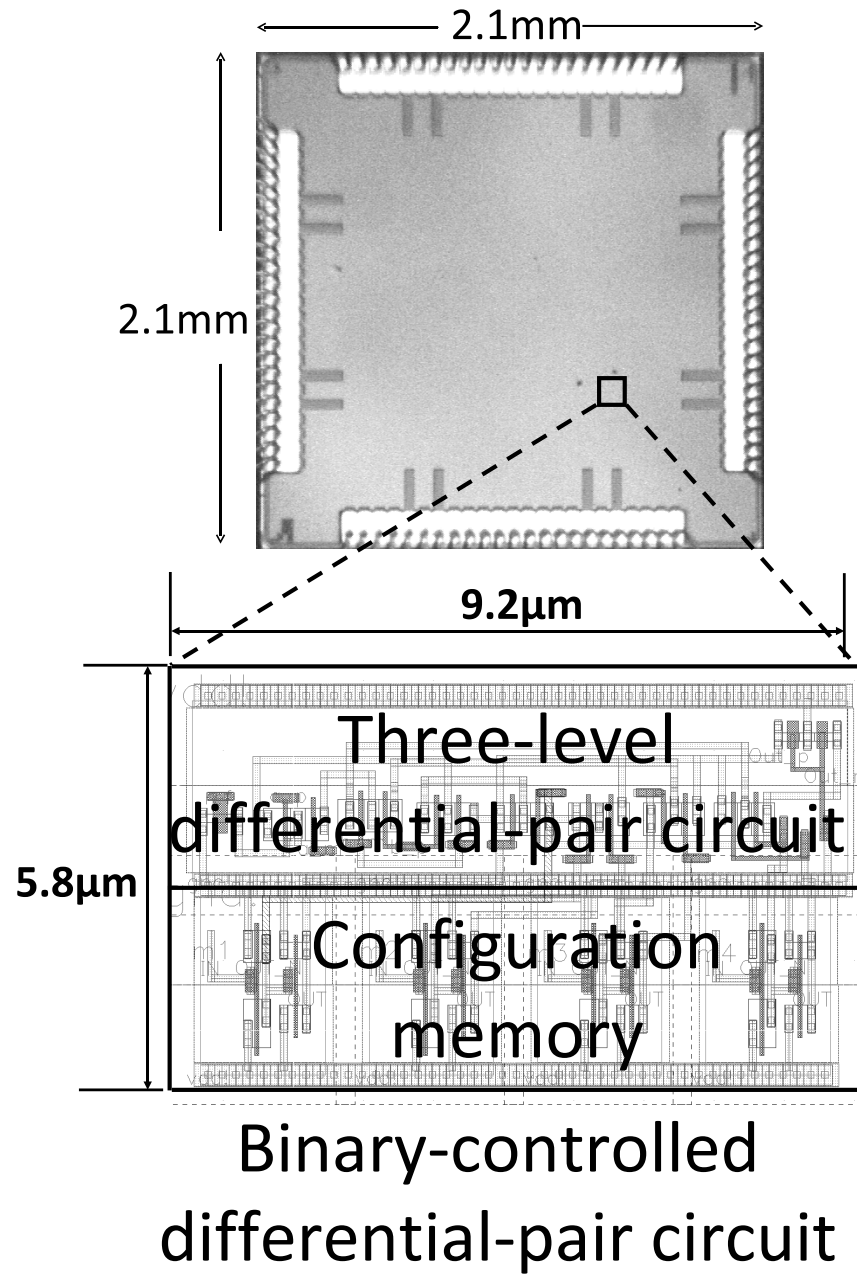


Figure 2.15: Chip photomicrograph and the layout of the binary-controlled differential-pair circuit

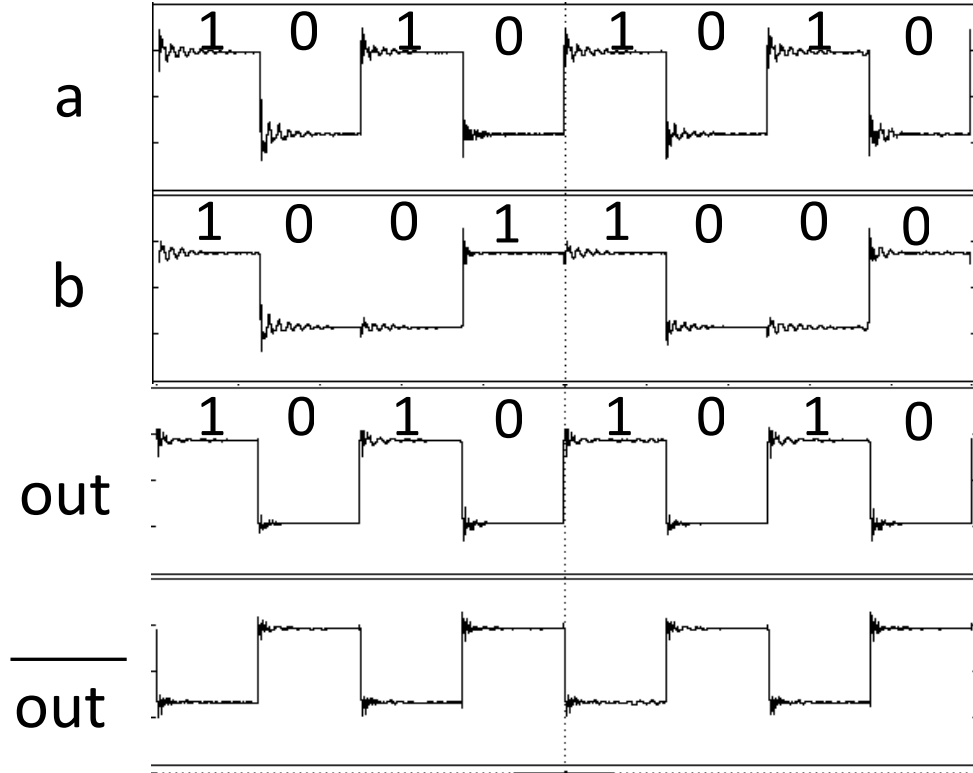


Figure 2.16: Inputs and outputs waveforms of the binary-controlled differential-pair circuit in the chip

The evaluation of the binary-controlled differential-pair circuit is done based on HSPICE simulation using a 65 nm CMOS design rule. The binary-controlled differential-pair circuit is compared with the previous two-variable binary function circuit shown in Fig. 2.8, and with the two-input LUT shown in Fig. 2.17. The LUT is used in the typical commercially available FPGAs as function generators. An n-input LUT can be used to implement an arbitrary n-variable binary function [1][17].

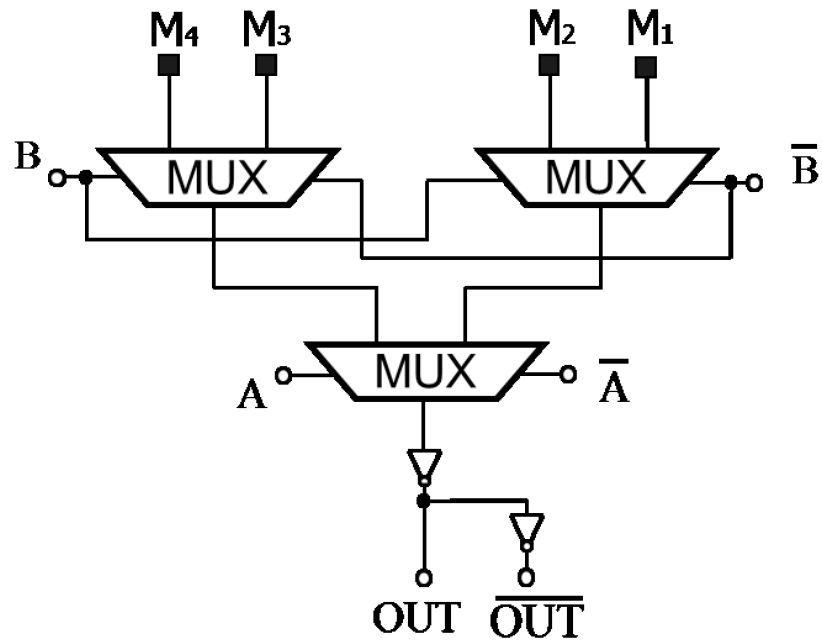


Figure 2.17: Two-input LUT

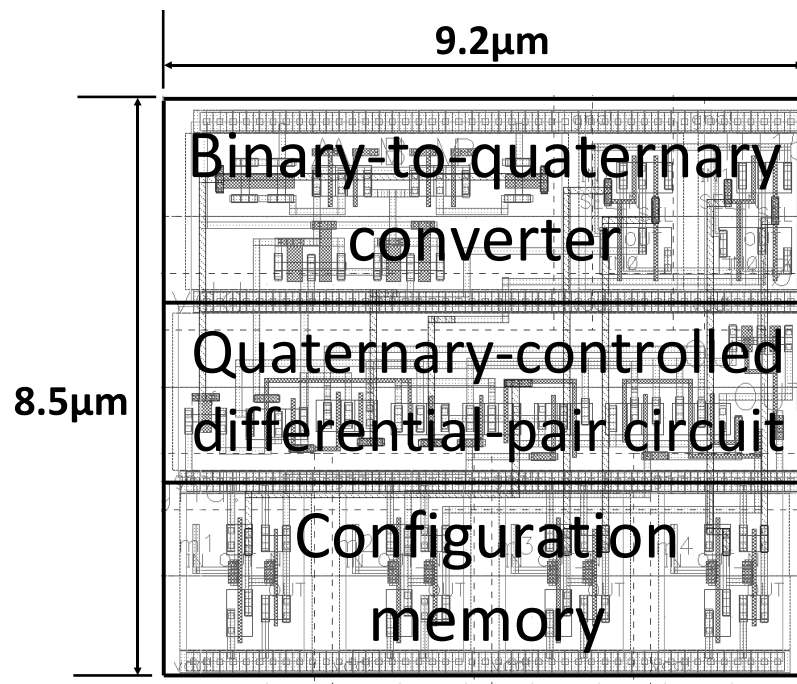


Figure 2.18: Layout of the previous two-variable binary function circuit

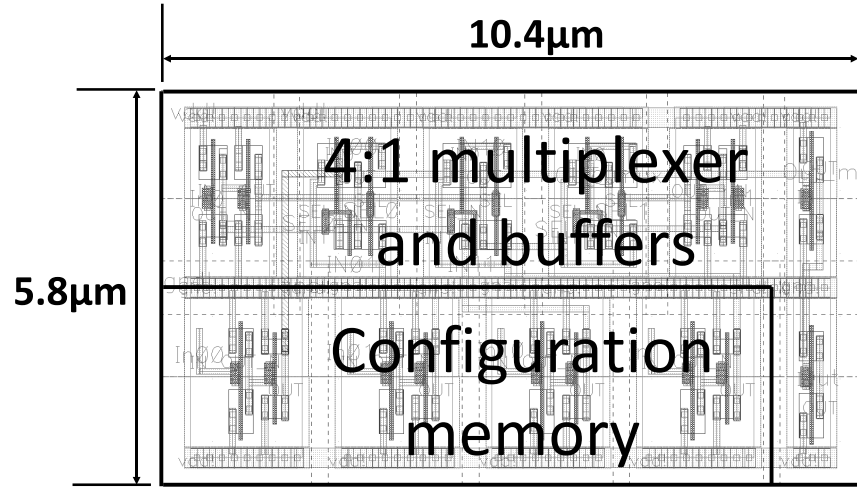


Figure 2.19: Layout of the two-input LUT

Table 2.5: Comparison of the two-variable binary function circuits

	QCDPC with a B-Q converter	Two-input LUT	BCDPC
Supply voltage	1.2V	1.2V	1.0V
Delay	0.15ns	0.06 ns	0.05 ns
Area	$78.2\mu m^2$	$60.32\mu m^2$	$53.36\mu m^2$
Configuration memory count	4	4	4

QCDPC: Quaternary-Controlled Differential-Pair Circuit

B-Q converter: Binary-to-Quaternary converter

BCDPC: Binary-Controlled Differential-Pair Circuit

Figures 2.18 and 2.19 show the layouts of the previous two-variable binary function circuit and the two-input LUT, respectively. The areas of the circuits are $78.2\mu m^2$ and $60.32\mu m^2$, respectively.

Table 2.5 shows the comparison results. The area and the delay of the proposed binary-controlled differential-pair circuit are reduced to 68% and 33%, respectively, in comparison with those of the previous two-variable binary function circuit. The area and the delay become 88% and 83% of those of the 2-input LUT, respectively.

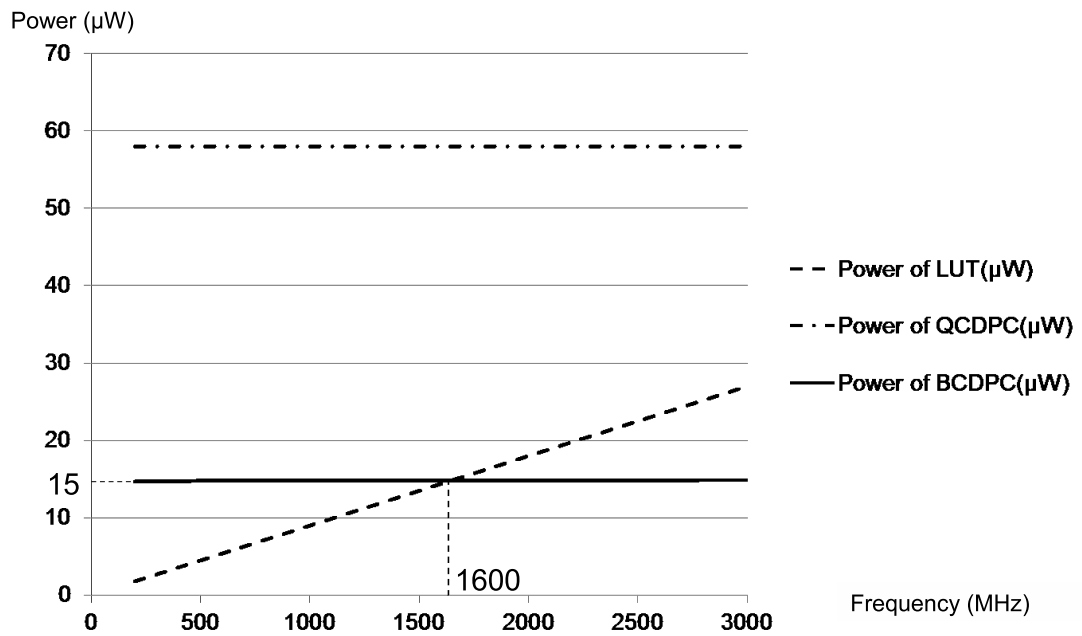


Figure 2.20: Power consumption versus operating frequency in the two-variable binary function circuits

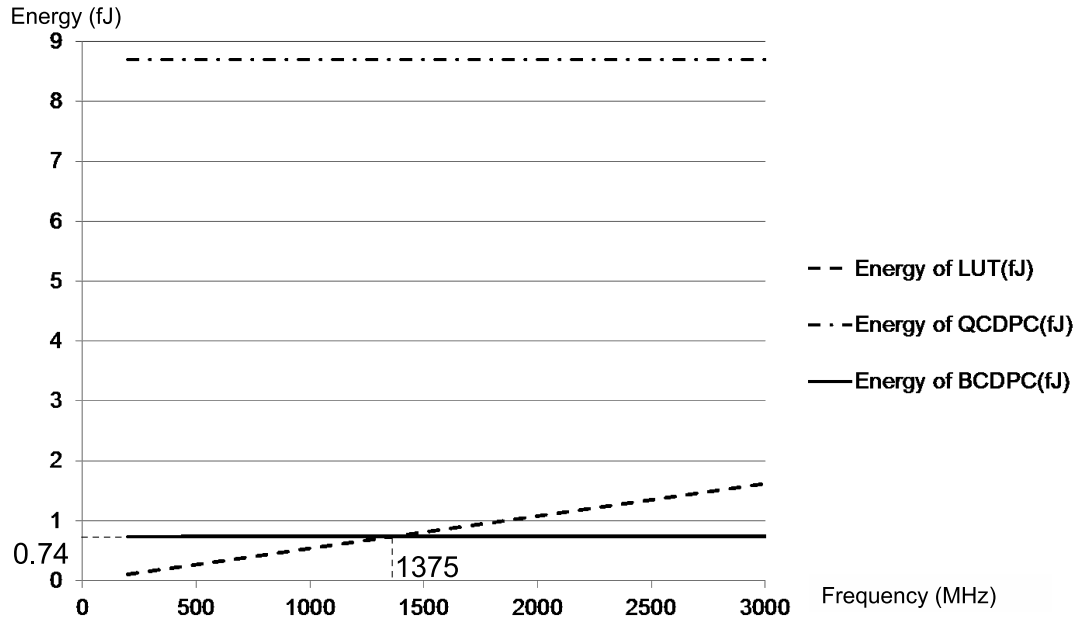


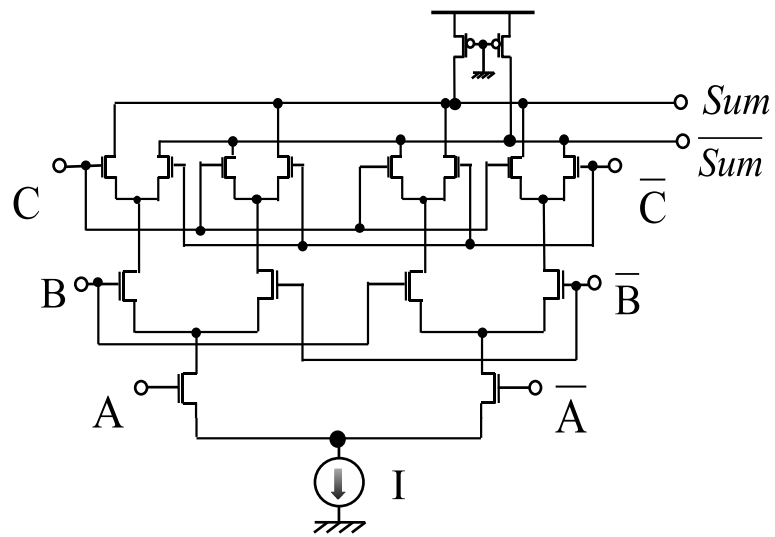
Figure 2.21: Energy consumption versus operating frequency in the two-variable binary function circuits

Figure 2.20 shows the characteristics of power consumption versus operating frequency in the two-variable binary function circuits. The power consumptions of the previous two-variable binary function circuit and the binary-controlled differential-pair circuit are almost constant and equal to $58\mu\text{W}$ and $15\mu\text{W}$, respectively, when the operating frequency increases. The binary-controlled differential-pair circuit has lower power consumption than the two-input LUT when the operating frequency is more than 1.6GHz.

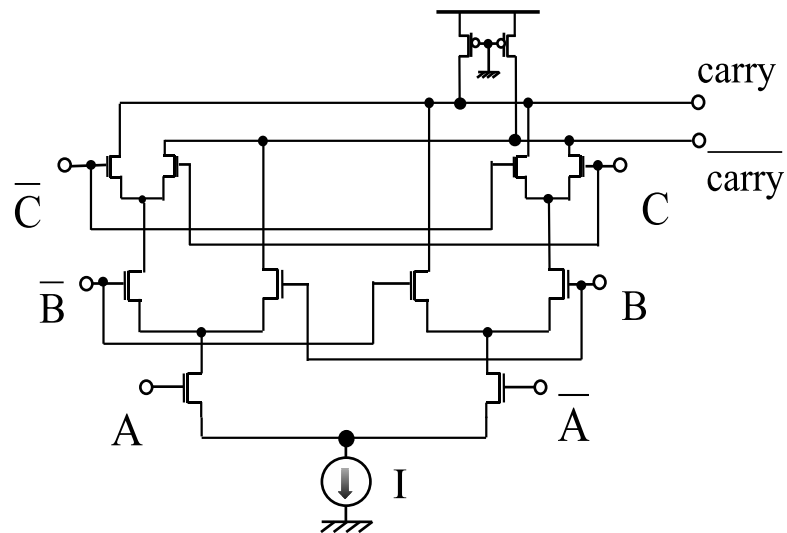
Figure 2.21 shows the characteristics of energy consumption versus operating frequency in the two-variable binary function circuits which are used to implement f_5 in Table 2.3. The energy consumptions of the previous two-variable binary function circuit and the binary-controlled differential-pair circuit are almost constant and equal to 8.7fJ and 0.74fJ, respectively, when the operating frequency increases. The binary-controlled differential-pair circuit has lower energy consumption than the two-input LUT when the operating frequency is more than 1.375GHz.

The power consumption and energy consumption of the binary-controlled differential-pair circuit are dramatically reduced in comparison with the previous two-variable binary function circuit. Also, the binary-controlled differential-pair circuit is suitable for high-frequency operations in comparison with the two-input LUT.

Figure 2.22 shows a current-mode sum circuit and a current-mode carry circuit that construct a full adder in binary current-mode logic[9]. Similar to the binary-controlled differential-pair circuit, either the sum circuit or the carry circuit is constructed by a three-level differential-pair circuit.



Current-mode sum circuit



Current-mode carry circuit

Figure 2.22: Current-mode full-adder circuit

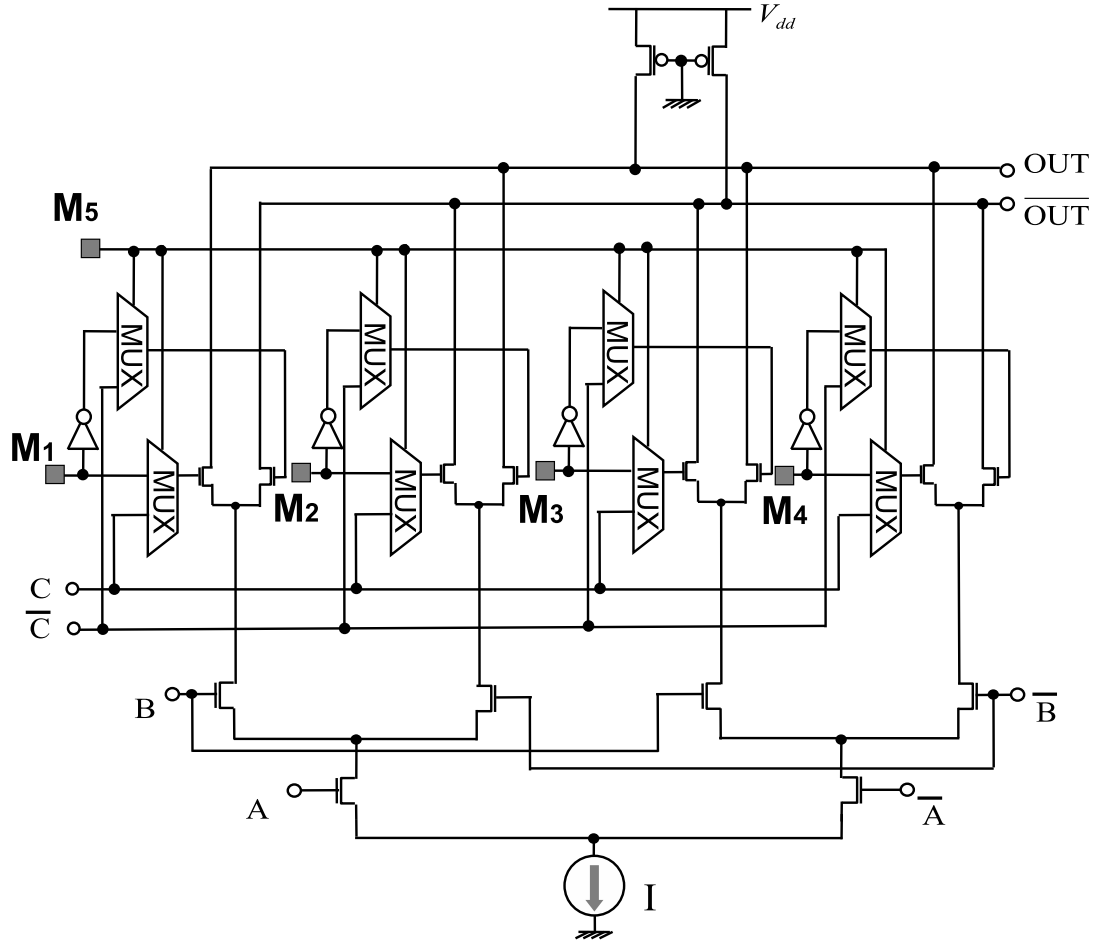


Figure 2.23: Sum-type binary-controlled differential-pair circuit

Therefore, I can share the binary-controlled differential-pair circuit and the current-mode sum (carry) circuit using multiplexers which are controlled by a configuration memory M_5 as shown in Fig. 2.23 (Fig. 2.24). A sum (carry)-type binary-controlled differential-pair circuit can be used to implement an arbitrary two-variable binary function or generate the full-adder sum (carry). An arbitrary two-variable binary function

can be implemented, if multiplexers are used to switch configuration memories M_1, M_2, M_3 and M_4 as the inputs of the third-level differential pairs. The full-adder sum (carry) can be generated, if the multiplexers are used to switch a carry signal (c, \bar{c}) as the input of the third-level differential pairs.

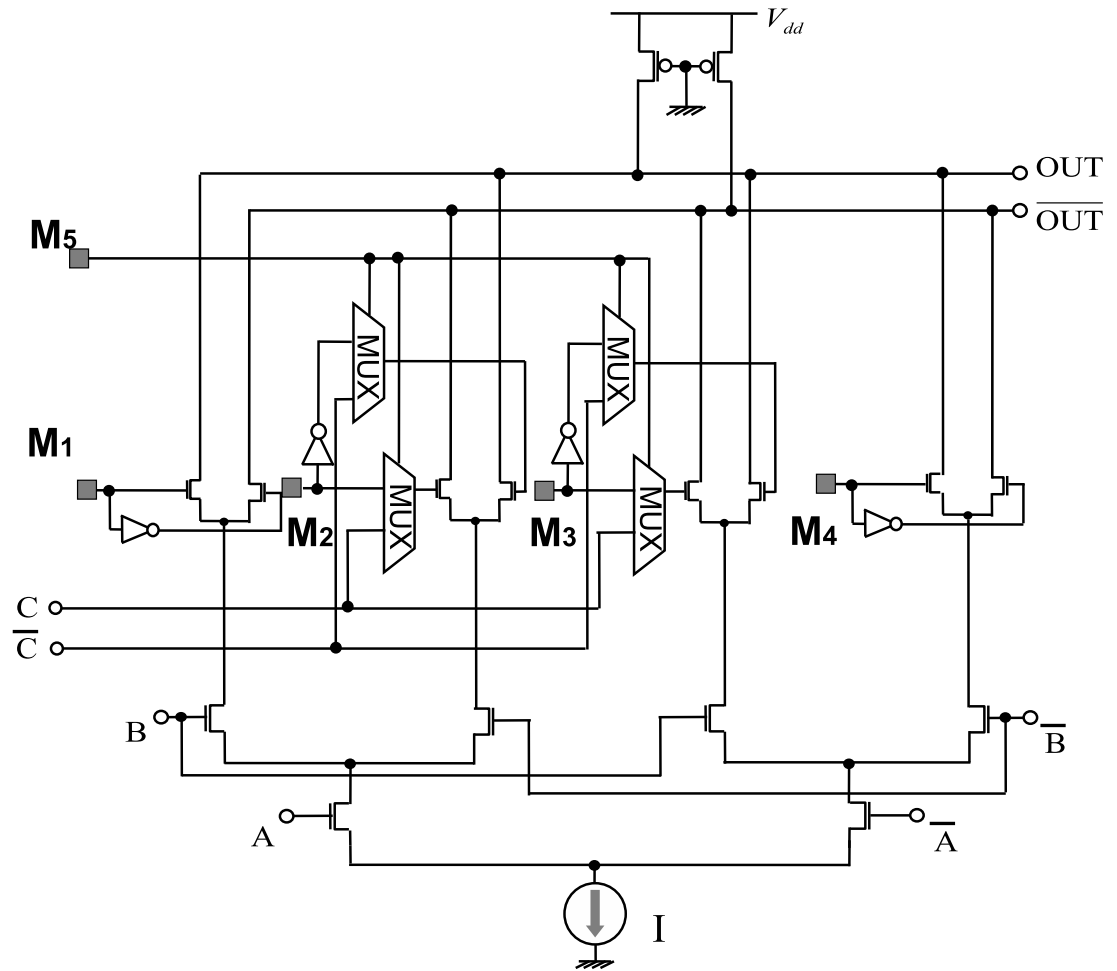


Figure 2.24: Carry-type binary-controlled differential-pair circuit

2.4 Current-source sharing technique

2.4.1 Design of the current-source-sharing differential-pair circuit

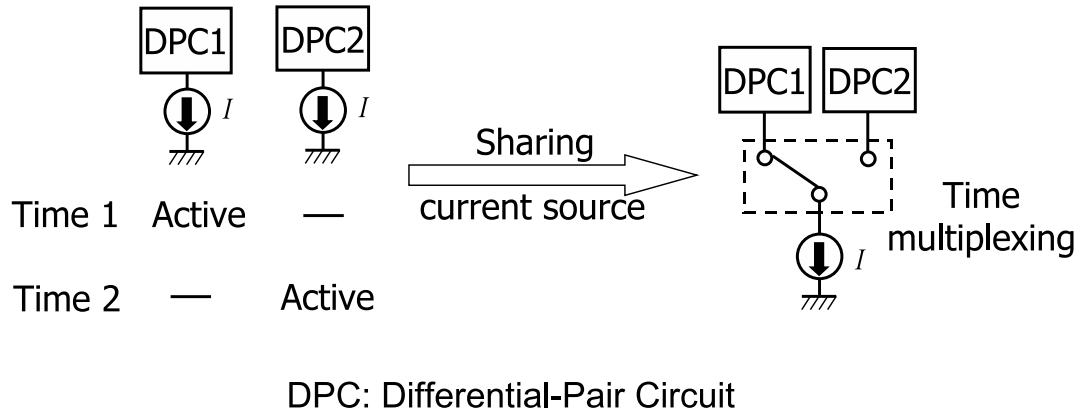


Figure 2.25: Current-source sharing technique in differential-pair circuits

The current-source sharing technique is proposed to improve the utilization of the current sources to implement low-power current-mode logic circuits. If only one of the differential-pair circuits is active at a time, one current source can be shared to drive the differential-pair circuits by time multiplexing, as shown in Fig. 2.25 [8].

Figure 2.26 shows the current-source sharing between a series-gating differential-pair circuit and a current-mode D-latch. The current-mode D-latch consists of a current source, a sample stage and a hold stage[9][18]. A complementary clock signal (Clk, \overline{Clk}) is used to steer the current I produced by the current source. The sample stage implemented by two

nMOS transistors is used to sample the logic function results Z_0 and $\overline{Z_0}$, whereas the hold stage implemented by two cross-coupled nMOS transistors is used to store that data.

In the current-mode D-latch, if Clk is low, the hold stage is inactive and the sample stage is turned “ON” to sample the Z_0 and $\overline{Z_0}$. To implement the same operation, \overline{Clk} can be used to turn “ON” the current source of the series-gating differential-pair circuit to generate the Z_0 and $\overline{Z_0}$. In that case, the current source of the current-mode D-latch is not useful to drive the sample stage to sample the Z_0 and $\overline{Z_0}$. On the other hand, if Clk is high, the hold stage is active and the sample stage is cut off. Therefore, the Z_0 and $\overline{Z_0}$ are not sampled, and the current source of the series-gating differential-pair circuit is not useful to generate the Z_0 and $\overline{Z_0}$. As a result, one current source can be shared to drive a current-source-sharing differential-pair circuit to implement a logic function and store its result. Also, the delay can be reduced because the sample stage in the current-mode D-latch is omitted by the current-source sharing technique.

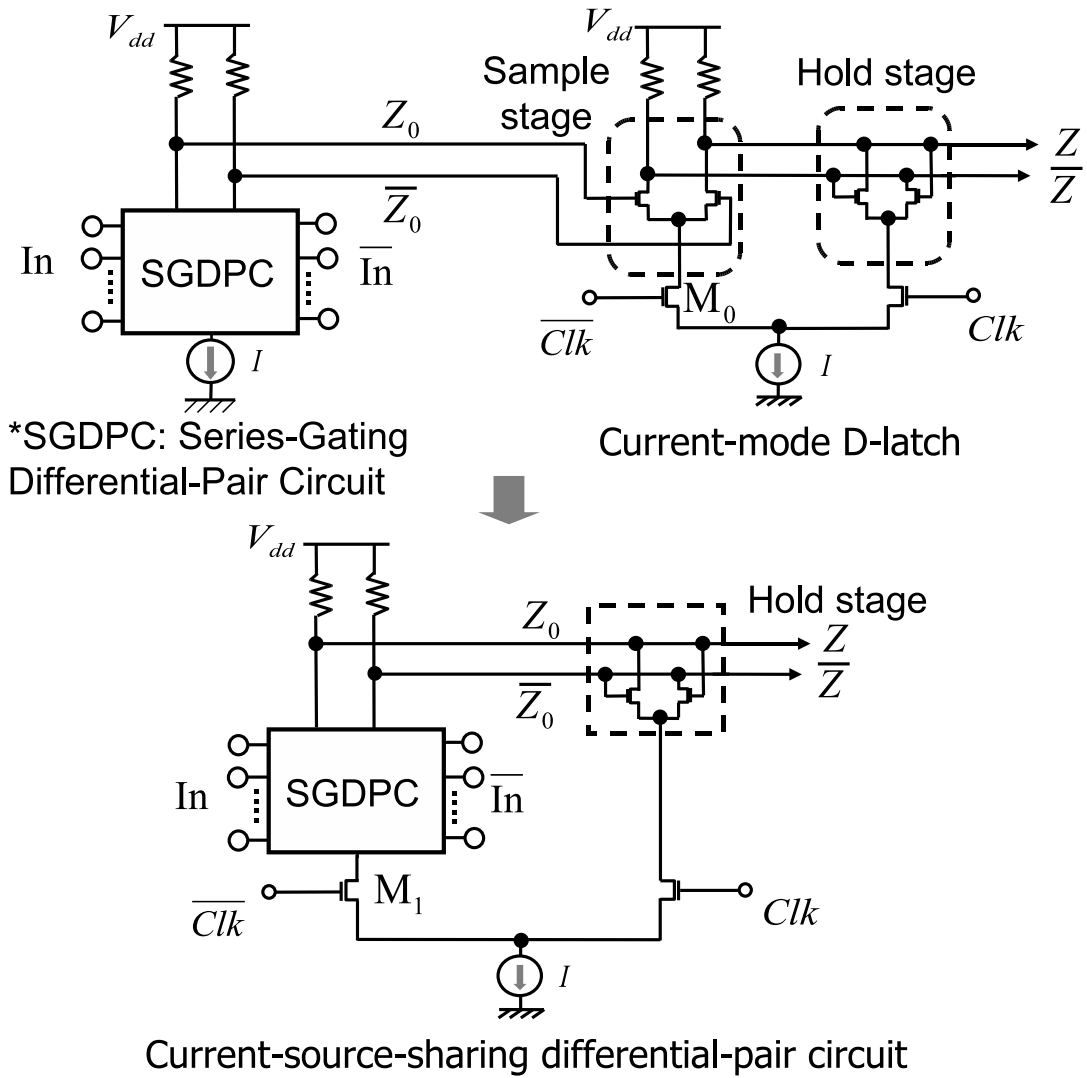


Figure 2.26: Current-source sharing technique between a series-gating differential-pair circuit and a current-mode D-latch

To demonstrate the advantage of the current-source sharing technique, I compare the performance of a current-source-sharing bit-serial adder shown in Fig. 2.27 with those of the current-mode bit-serial adder shown in Fig. 2.28 and the CMOS bit-serial adder shown in Fig. 2.29 based on HSPICE simulation using a 65 nm CMOS design rule.

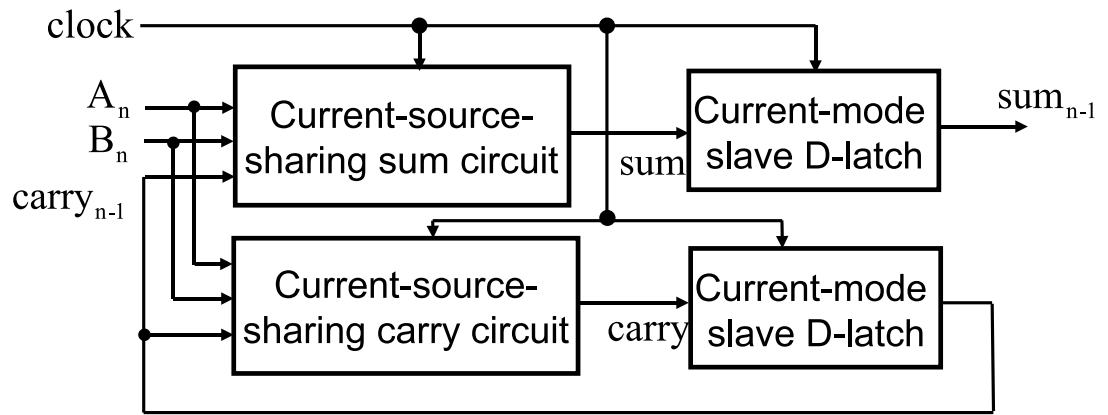


Figure 2.27: Design of the current-source-sharing bit-serial adder

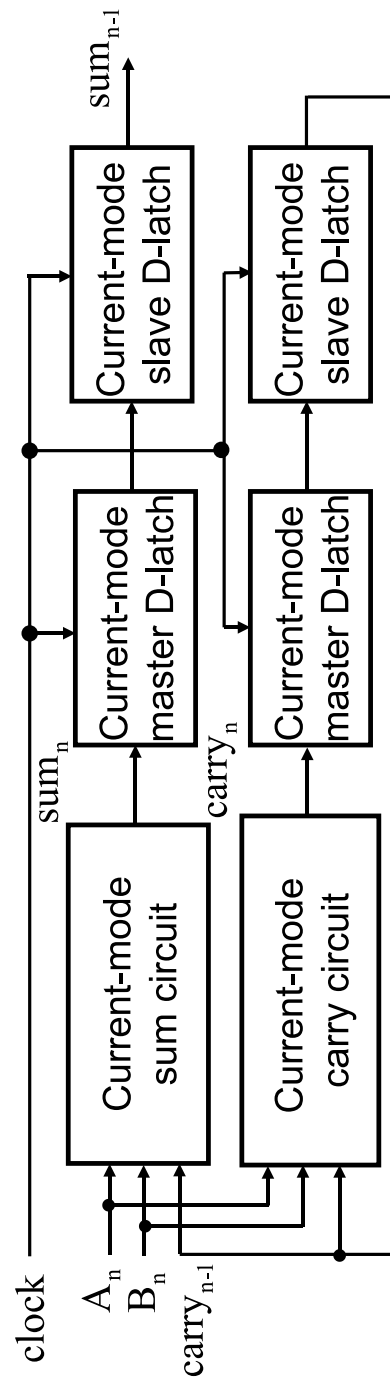


Figure 2.28: Design of the current-mode bit-serial adder

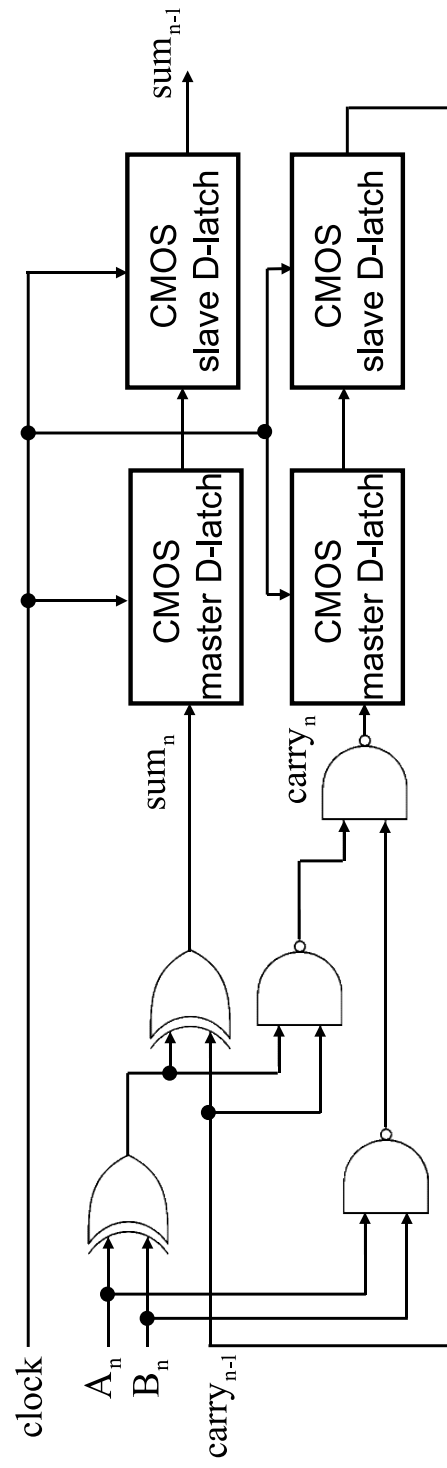


Figure 2.29: Design of the CMOS bit-serial adder

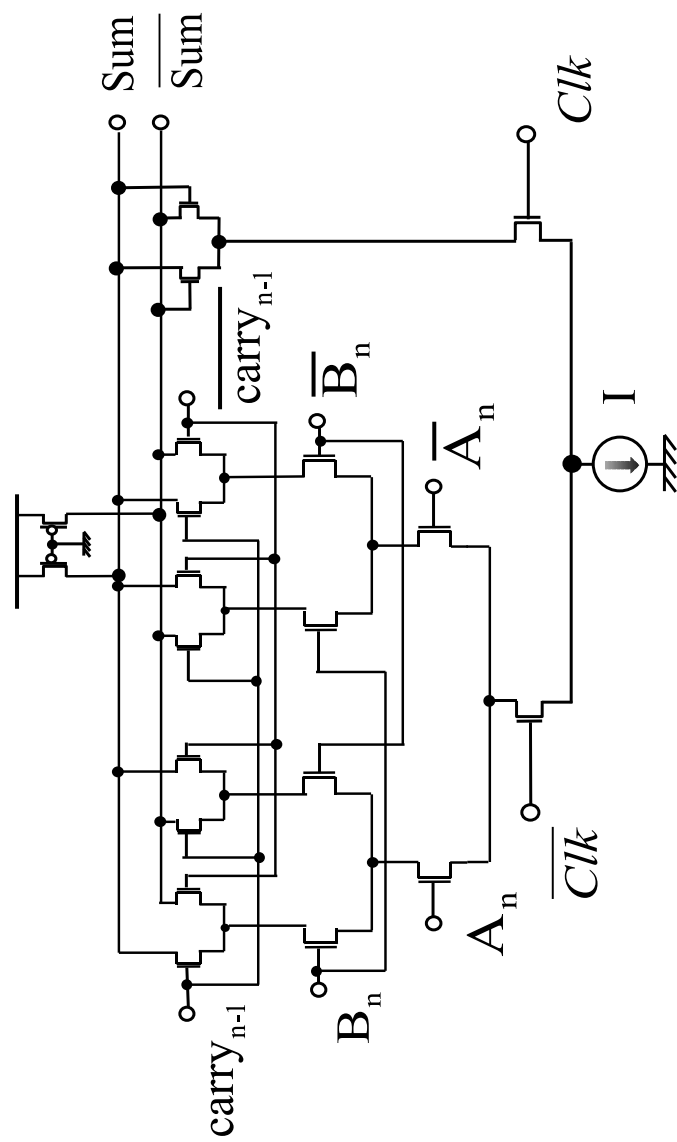


Figure 2.30: Current-source-sharing sum circuit

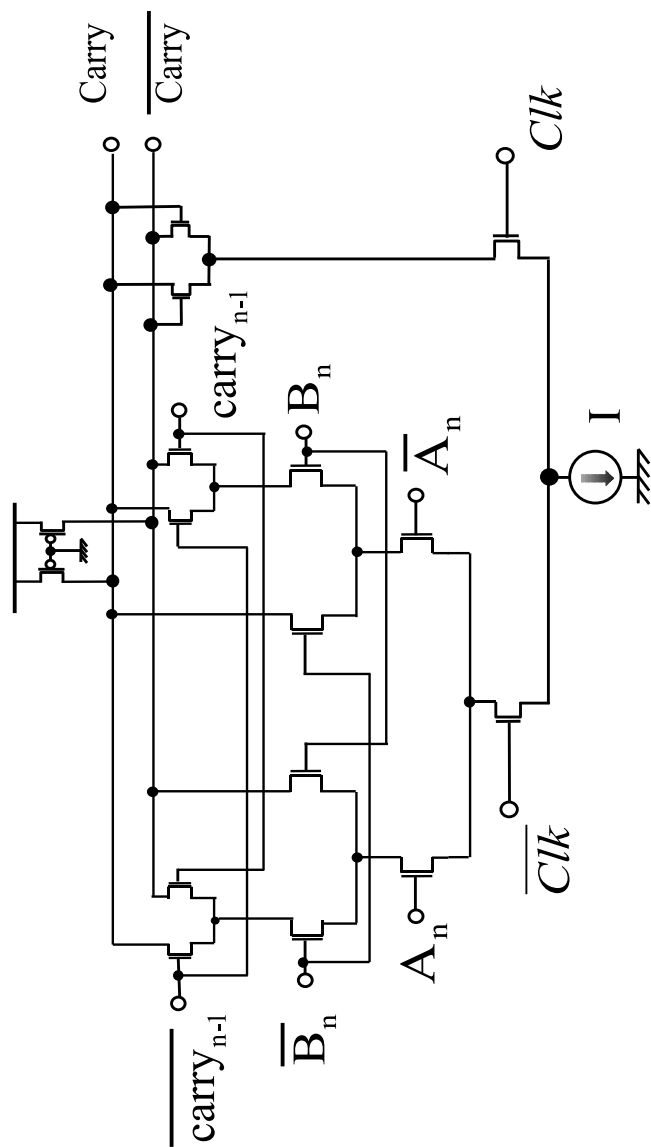


Figure 2.31: Current-source-sharing carry circuit

The current-mode bit-serial adder consists of the current-mode sum circuit, the current-mode carry circuit shown in Fig. 2.22, two current-mode master D-latches and two current-mode slave D-latches. The current-source-sharing bit-serial adder consists of a current-source-sharing sum circuit shown in Fig. 2.30, a current-source-sharing carry circuit shown in Fig. 2.31 and two current-mode slave D-latches.

Figures 2.32, 2.33 and 2.34 show the layouts of the current-source-sharing bit-serial adder, the current-mode bit-serial adder and the CMOS bit-serial adder, respectively. The areas of the bit-serial adders are $66.78\mu m^2$, $80.56\mu m^2$ and $75.4\mu m^2$, respectively. In the current-source-sharing sum (carry) circuit, one current source can be shared to generate the full-adder sum (carry) and store the result. Figure 2.35 shows the input and output waveforms of the current-source-sharing sum and carry circuits. If Clk is low, the current I generated by the current source flows through the sum (carry) circuit. If Clk is high, the current I flows through the hold stage, the sum (carry) result is stored.

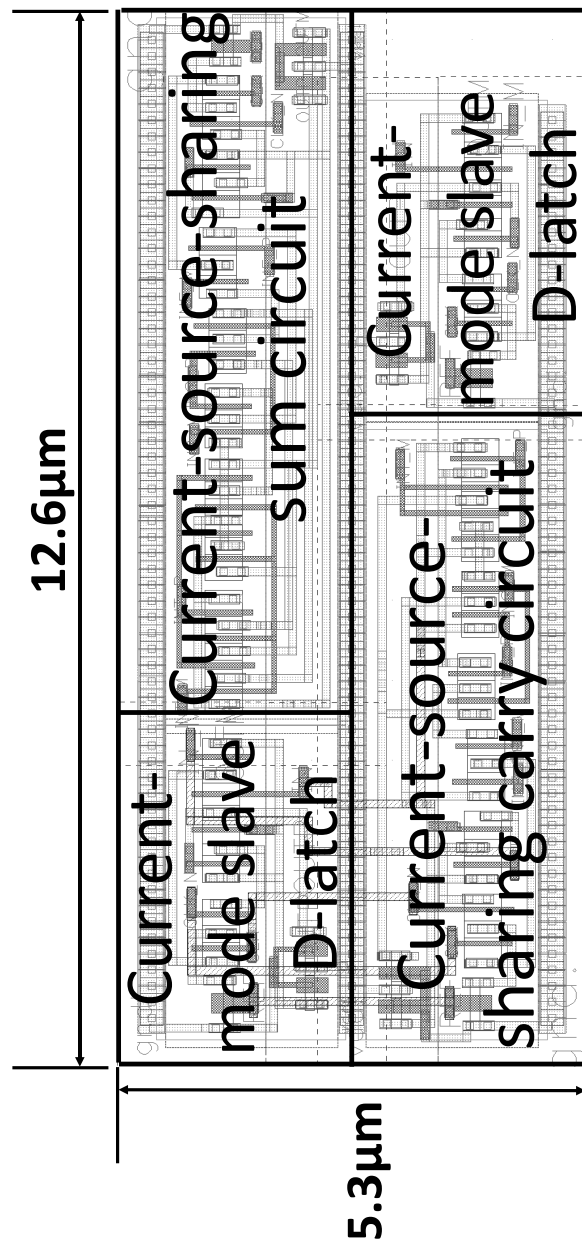


Figure 2.32: Layout of the current-source-sharing bit-serial adder

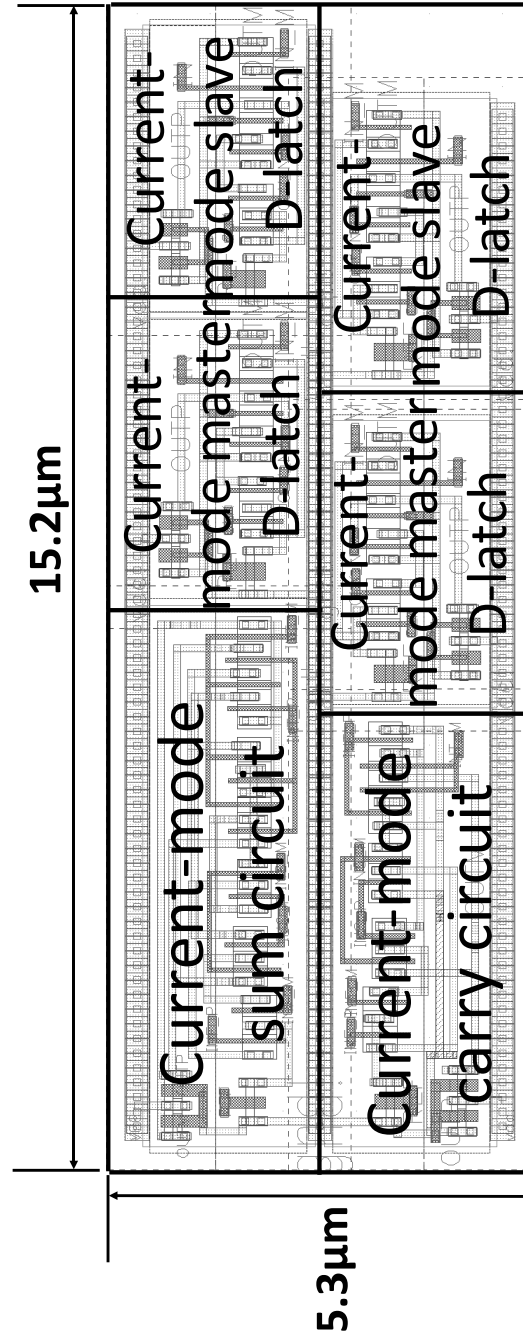


Figure 2.33: Layout of the current-mode bit-serial adder

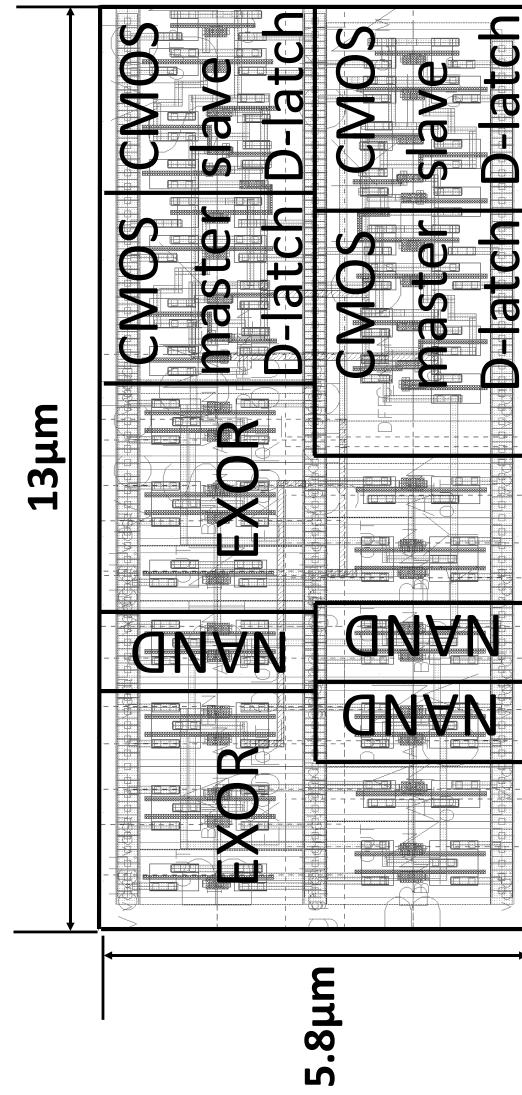


Figure 2.34: Layout of the CMOS bit-serial adder

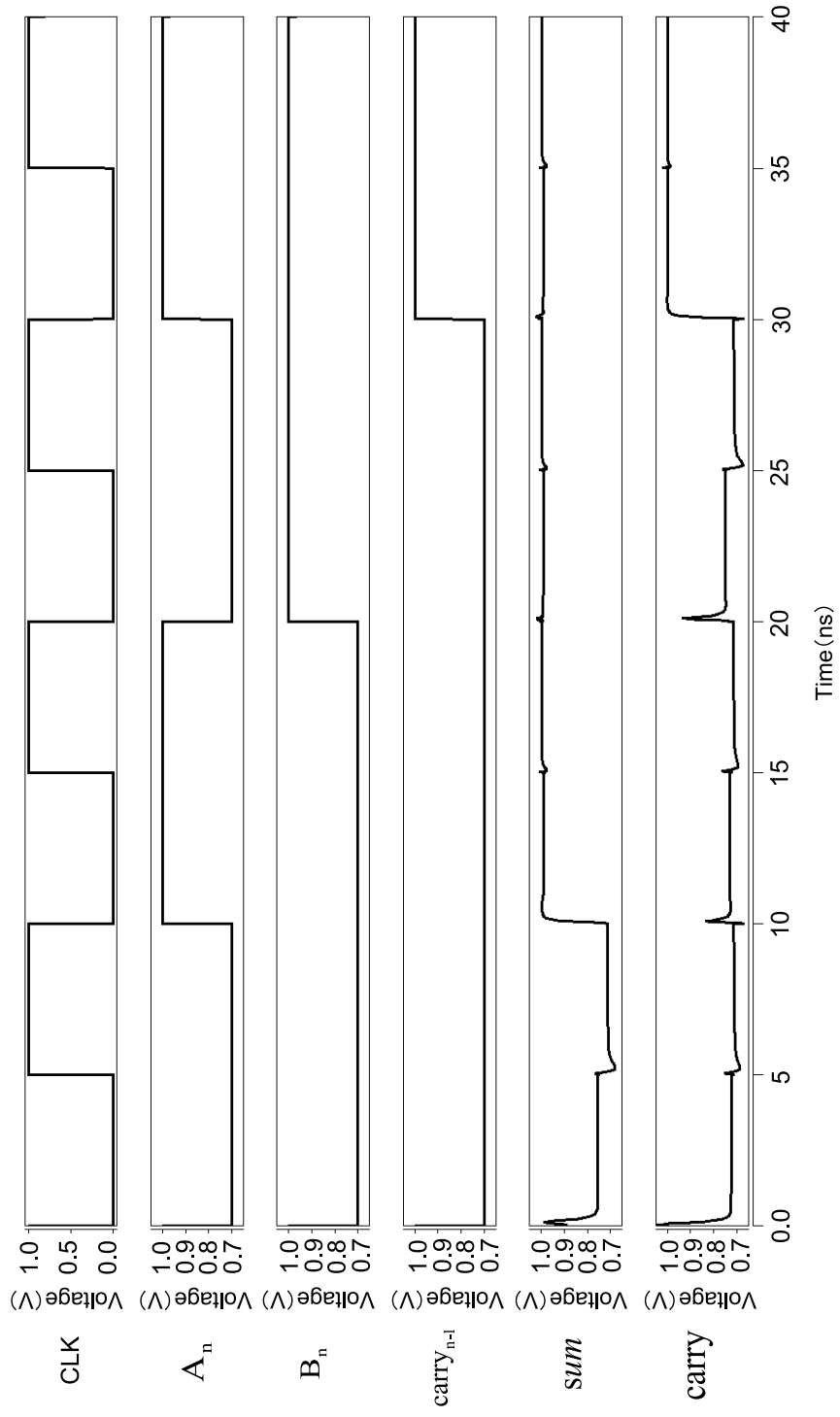


Figure 2.35: Input and output waveforms of the current-source-sharing sum and carry circuits

2.4.2 Evaluation of the current-source-sharing differential-pair circuit

Table 2.6: Comparison of the bit-serial adders (BSAs)

	CMOS BSA	Current-mode BSA	CSSBSA
Supply voltage	1.2V	1.2V	1.2V
Delay	0.15ns	0.1 ns	0.07 ns
Area	$75.4\mu m^2$	$80.56\mu m^2$	$66.78\mu m^2$

Table 2.6 shows the comparison results. The area and the delay of the proposed current-source-sharing bit-serial adder are reduced to 83% and 70%, respectively, in comparison with those of the current-mode bit-serial adder. The area and the delay become 88% and 47% of those of the CMOS bit-serial adder, respectively.

Figure 2.36 shows the characteristics of power consumption versus operating frequency in the bit-serial adders. The power consumptions of the current-mode bit-serial adder and the current-source-sharing bit-serial adder are almost constant and equal to $91\mu W$ and $51\mu W$, respectively, when the operating frequency increases. The current-mode bit-serial adder and the current-source-sharing bit-serial adder have lower power consumptions than the CMOS bit-serial adder when the operating frequencies are more than 1.95GHz and 1.09GHz, respectively.

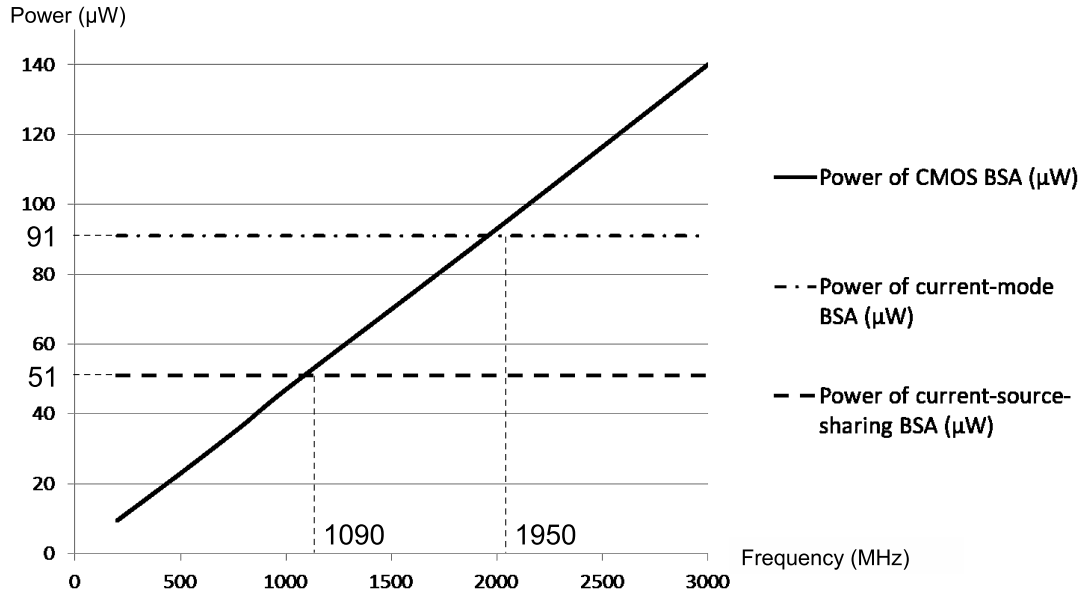


Figure 2.36: Power consumption versus operating frequency in the bit-serial adders (BSAs)

Figure 2.37 shows the characteristics of energy consumption versus operating frequency in the bit-serial adders which are used to implement an 8-bit addition. The energy consumptions of the current-mode bit-serial adder and the current-source-sharing bit-serial adder are almost constant and equal to 72.8fJ and 32.6fJ, respectively, when the operating frequency increases. The current-mode bit-serial adder and the current-source-sharing bit-serial adder have lower energy consumptions than the CMOS bit-serial adder when the operating frequencies are more than 1.3GHz and 0.6GHz, respectively.

The power consumption and the energy consumption of the current-source-sharing bit-serial adder are dramatically reduced in comparison with the current-mode bit-serial adder. Also, the current-source-sharing bit-serial adder is suitable for high-frequency operations in comparison with the CMOS bit-serial adder.

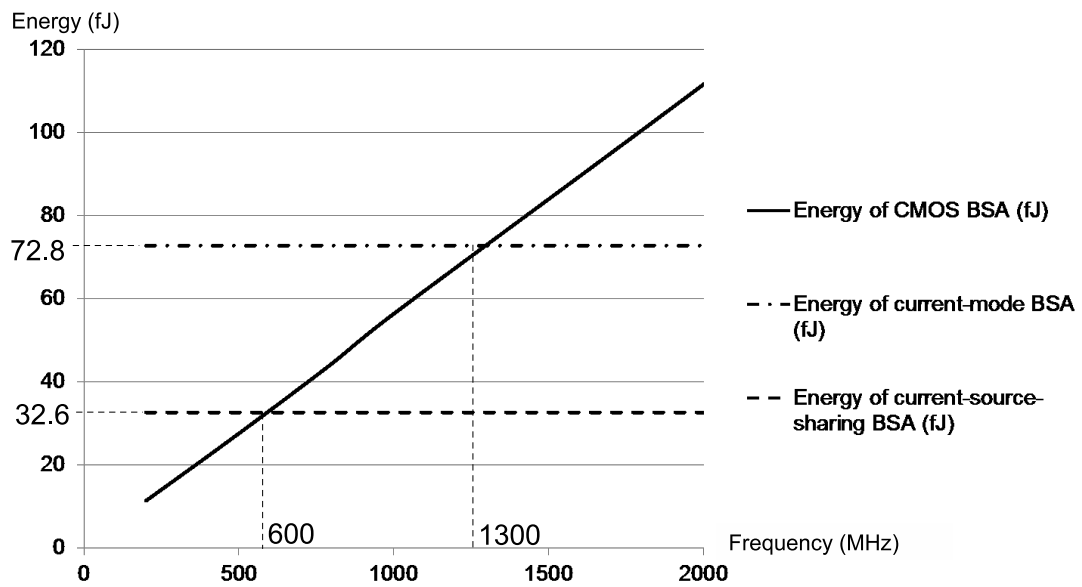


Figure 2.37: Energy consumption versus operating frequency in the bit-serial adders (BSAs)

2.5 Dual-supply voltage technique for low-power multiple-valued source-coupled logic circuits

Figure 2.38 shows the structure of the dual- V_{DD} multiple-valued source-coupled logic circuit which consists of an I-V converter, a comparator and an output generator. A summation I_S of binary current inputs I_1, I_2, \dots, I_k can be realized by wiring without any active devices, so that the resulting arithmetic circuits become simple.

The I-V converter is designed using a pMOS transistor which operates in the linear region. The multiple-valued current signal I_S is converted to a multiple-valued voltage signal V_S . V_{DDH} is required for V_S which is equal to $V_{DDH} - I_S \times R$, where R is the equivalent resistance of the pMOS load transistor.

The comparator implemented by the differential-pair circuit is used to realize a threshold operation. The current I generated by a current source is steered by V_S and a threshold voltage V_{th} . V_{DDL} is used for a dual-rail binary voltage signal (G, \overline{G}) whose value is 0 or $V_{DDL} - I \times R$, where R is the equivalent resistance of the pMOS load transistor.

In the differential-pair circuit, the propagation delay D_{DPC} and power consumption P_{DPC} are given as follows:

$$D_{DPC} = \frac{C \times \Delta V}{I} \quad (2.2)$$

$$P_{DPC} = V_{DD} \times I \quad (2.3)$$

where C is a load capacitance, ΔV is an output voltage swing, V_{DD} is a supply voltage and I is a constant current provided by the current source [9]. V_{DDL} can be used to reduce power consumption without decreasing speed. However, to make the differential-pair circuit work correctly, V_{DDL} should be higher than the threshold voltage of transistors.

The output generator implemented by the differential-pair circuit is used to generate a binary current output Y . When G is “1”, the nMOS transistors MN_1 and MN_2 operate in the saturation region and the cutoff region, respectively, so that the current I flows through MN_1 and Y becomes “1”. Otherwise, when G is “0”, MN_1 and MN_2 operate in the cutoff region and the saturation region, respectively, so that the current I flows through MN_2 and Y becomes “0”. V_{DDL} used in the comparator should be larger than the threshold voltages of MN_1 and MN_2 .

As shown in Fig. 2.39, the power consumption of the dual- V_{DD} MVSCL circuit is determined by the last current generator number k and the differential-pair circuit number m . The power consumption of the last current generators is the same as that of the single- V_{DD} MVSCL circuit, while the power consumption of the DPCs is reduced by the low supply voltage V_{DDL} in comparison with that of the single- V_{DD} MVSCL circuit. Therefore, the reduction ratio of the power consump-

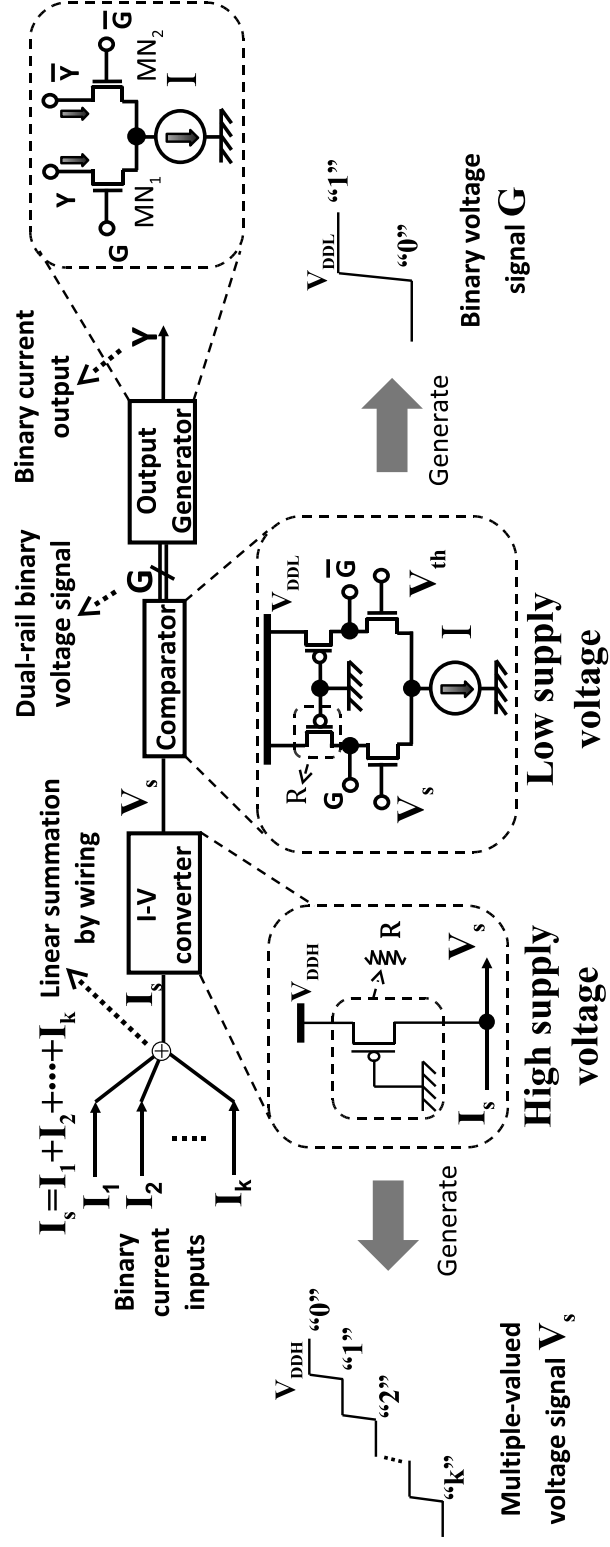


Figure 2.38: Structure of the dual- V_{DD} multiple-valued source-coupled logic circuit

tion is proportional to the differential-pair circuit number m if the low supply voltage V_{DDL} is fixed. It means that the proposed dual-supply voltage technique is more efficient for relatively complex operations.

As shown in Fig. 2.40, a direct-path current flows through a high-supply-voltage CMOS inverter with a low-supply-voltage “logic high” input signal, where the pMOS transistor is not turned off completely. To prevent the direct-path current, a level shifter is used in the conventional dual- V_{DD} CMOS circuit wherever low-supply-voltage gates drive high-supply-voltage gates [19].

As shown in Fig. 2.41, the current output of the output generator is connected to the next I-V converter with V_{DDH} . Therefore, the high-supply-voltage output generator is driven by the low-supply-voltage comparator. In the high-supply-voltage output generator, the current flow is fixed by the current source, so that the level shifter used to prevent the direct-path current is not necessary to be provided.

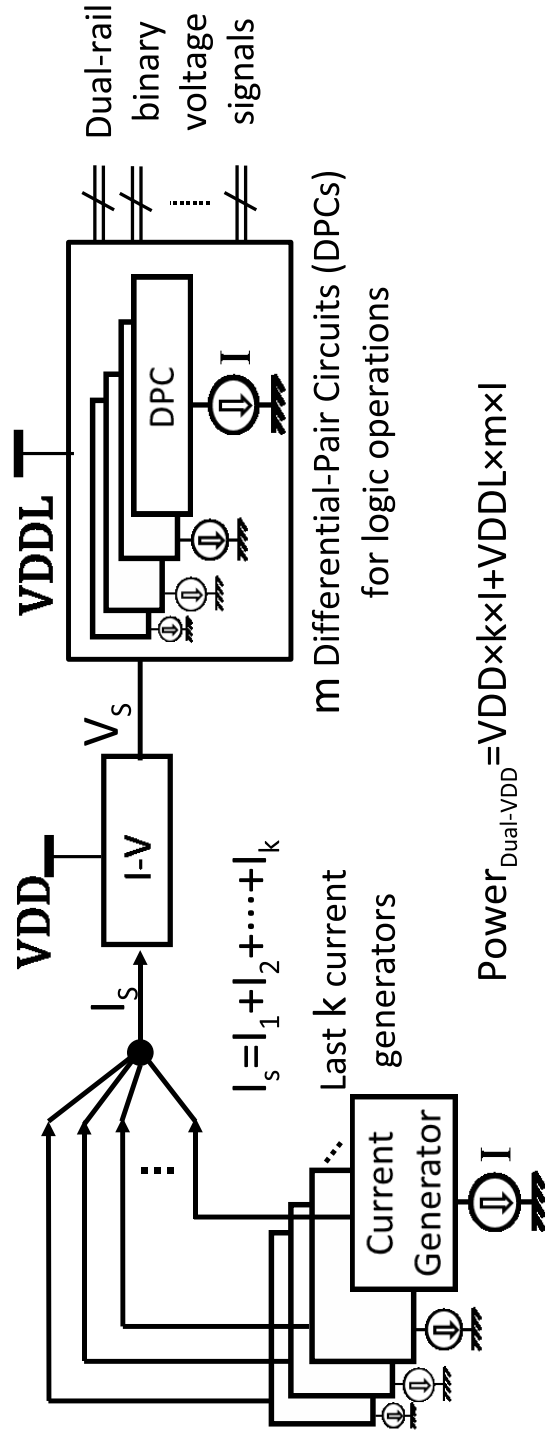


Figure 2.39: Power consumption of the dual- V_{DD} multiple-valued source-coupled logic circuit

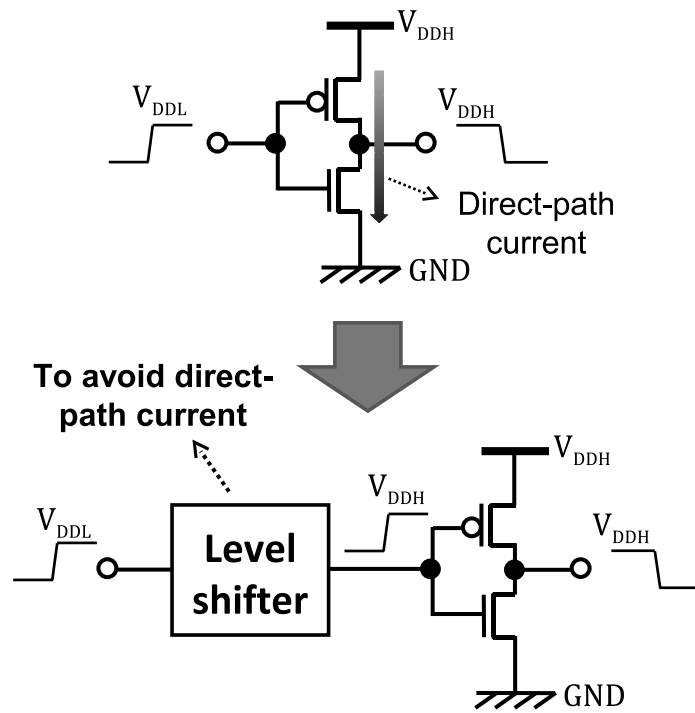


Figure 2.40: Level shifter in the conventional dual- V_{DD} CMOS circuit

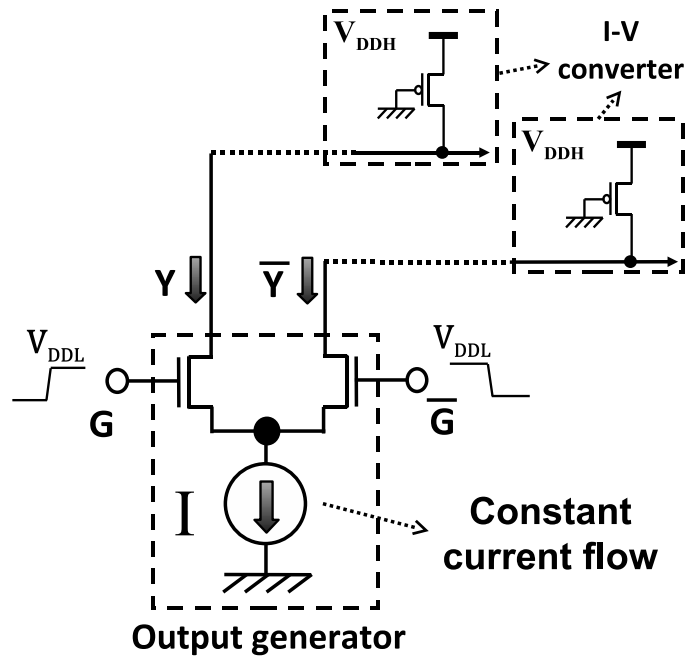


Figure 2.41: Constant current flow in the dual- V_{DD} multiple-valued source-coupled logic circuit

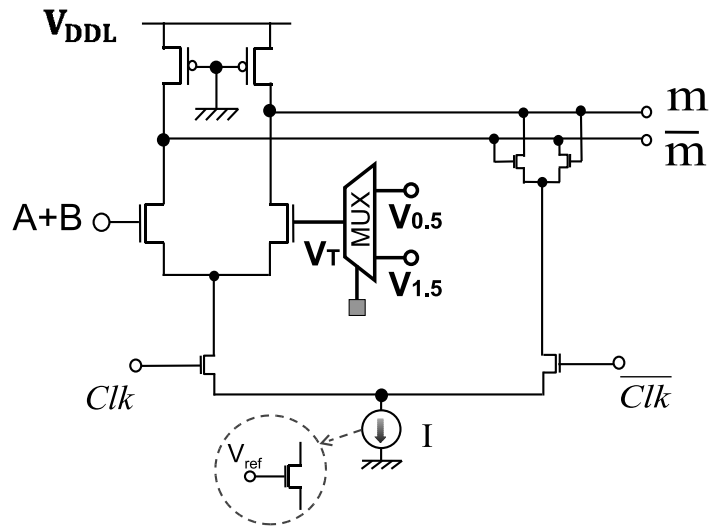
2.6 Design and evaluation of the multiple-valued cell using current-source-sharing differential-pair circuits

Figure 2.42 shows a dual- V_{DD} multiple-valued cell composed of a switch block and a logic block. Binary current inputs A and B are linearly summed by wiring. In a bit-serial operation, a start signal indicating a head of a one-word data is required to initialize D flip-flops used for a state memory. Superposition of a binary current input C and the start signal in a single interconnection is introduced to implement a compact switch block, where the logic value “1” and “0” is defined as C and the logic value “2” is defined as the start signal.

V_{DDH} is provided in the I-V converters, where three-valued current signals I_P and I_Q are converted to three-valued voltage signals V_P and V_Q . To express such three-valued voltage signals, a high-supply voltage 1.2V is provided for voltage levels 1.2V, 0.9V and 0.6V corresponding to logic values “0”, “1” and “2”, respectively.

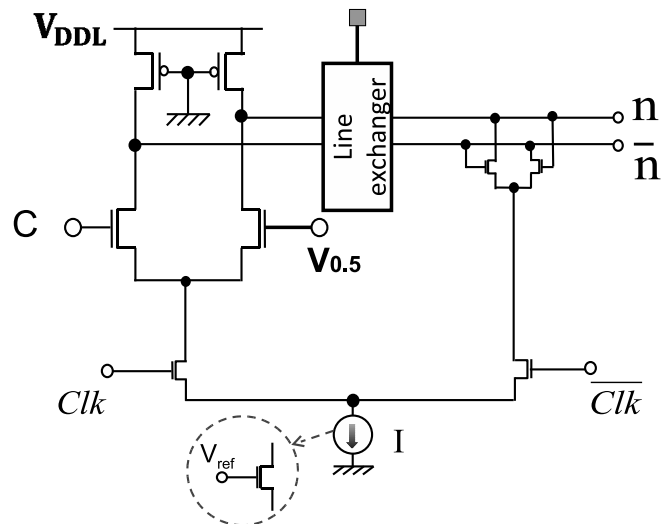
V_{DDL} is provided in the other parts including a current-source-sharing AND circuit, a current-source-sharing NOT circuit, a start signal detector and a current-source-sharing binary logic module which are constructed by the differential-pair circuits. In the start signal detector im-

plemented by a one-level differential-pair circuit, the threshold is set “1.5” to make the output “1” for the input logic value “2”.



(a) Current-source-sharing AND circuit

■ ... Configuration memory



(b) Current-source-sharing NOT circuit

Figure 2.43: Current-source-sharing threshold logic circuits

Figure 2.43 shows the current-source-sharing AND circuit and the current-source-sharing NOT circuit constructed by a two-level differential-pair circuit. A low-supply voltage 0.9V is provided for voltage levels 0.9V and 0.6V corresponding to logic values “1”, “0”, respectively. One current source is shared to drive the current-source-sharing AND (NOT) circuit, where the programmable operations shown in Table 2.1 (Table 2.2) is performed if CLK is high, and the operation result is stored if CLK is low. An AND operation is selected to generate a partial product in a multiplication, and a NOT operation is selected to converter the input to a 2’s complement number in a subtraction.

Figure 2.44 shows the input and output waveforms of the current-source-sharing AND circuit which is programed to realize an AND operation. The voltage levels 1.2V, 0.9V, and 0.6V of V_P correspond to (A=0,B=0), (A=0,B=1) or (A=1,B=0), and (A=1,B=1), respectively. To realize an AND operation of A and B, a threshold “1.5” is selected to realize a threshold operation in the current-source-sharing AND circuit.

Figure 2.45 shows the input and output waveforms of the current-source-sharing NOT circuit which is programed to realize a NOT operation. The voltage levels 1.2V and 0.9V of V_Q correspond to C=0 and C=1, respectively. To realize a NOT operation of C, the exchange pattern shown in Fig 2.46 is selected in the line exchanger.

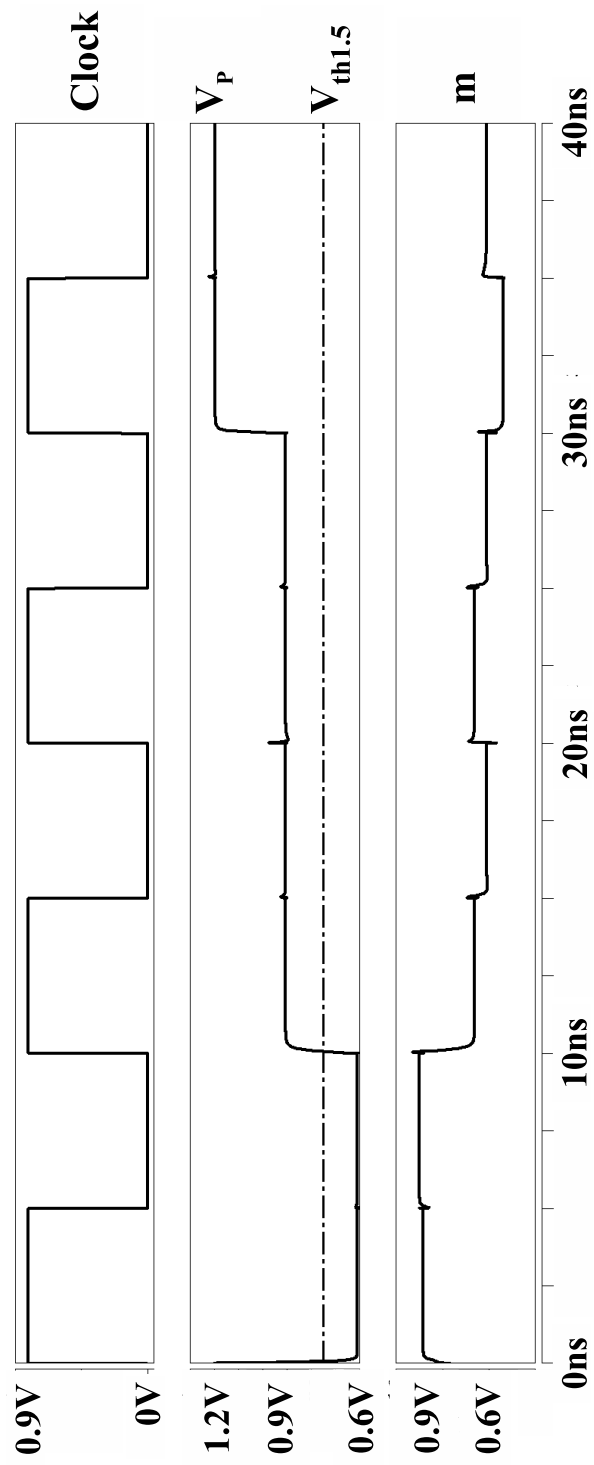


Figure 2.44: Input and output waveforms of the current-source-sharing AND circuit

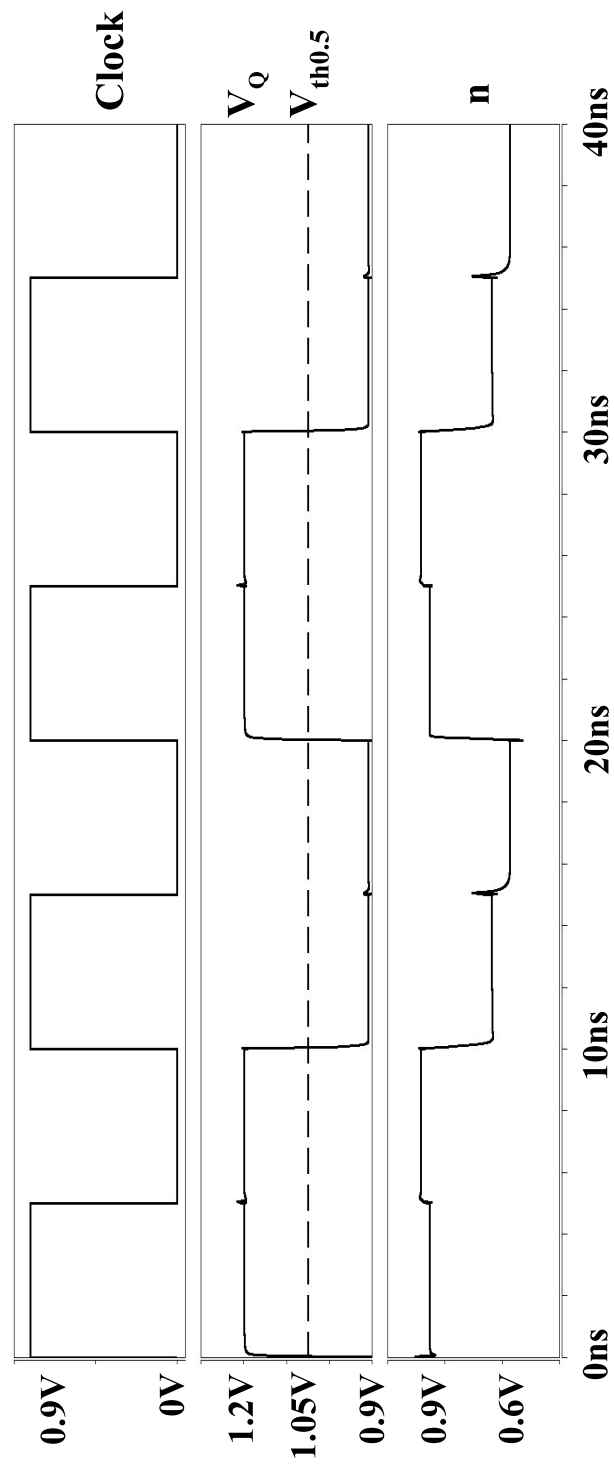


Figure 2.45: Input and output waveforms of the current-source-sharing NOT circuit

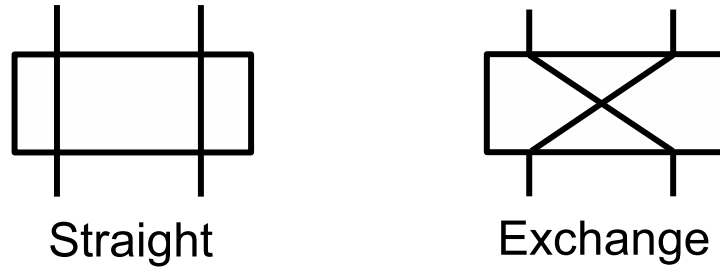
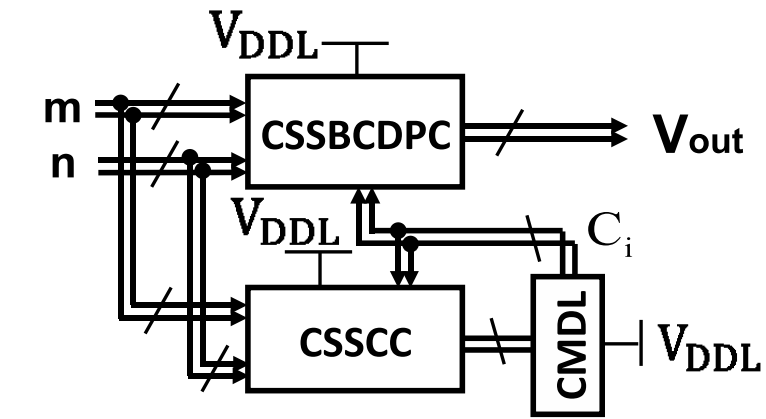


Figure 2.46: Switching patterns of the line exchanger



CSSBCDPC : **C**urrent-**S**ource-**S**haring **B**inary-
Controlled **D**ifferential-**P**air **C**ircuit
CSSCC : **C**urrent-**S**ource-**S**haring **C**arry **C**ircuit
CMDL : **C**urrent-**M**ode **D**-**L**atch

Figure 2.47: Current-source-sharing binary logic module

As shown in Fig. 2.47, the current-source-sharing binary logic module is composed of a current-source-sharing binary-controlled differential-pair circuit, a current-source-sharing carry circuit and a current-mode D-latch. The current-source-sharing binary logic module can perform an arbitrary two-variable binary function shown in Table 2.3 or implement

a bit-serial adder [8].

In the current-source-sharing binary-controlled differential-pair circuit shown in Fig. 2.48, the current I produced by the current source is steered into one of the branches according to the dual-rail binary input voltages. In the first-level differential pair, when Clk is low, the current I flows through the nMOS transistor MN_1 . The current-source-sharing binary-controlled differential-pair circuit is programmed to realize an arbitrary two-variable binary function shown in Table 2.3 or generate the full-adder sum. When Clk is high, the current flows through the nMOS transistor MN_2 and the operation result is stored by two cross-coupled nMOS transistors. The binary voltages (m, \overline{m}) and (n, \overline{n}) generated by the current-source-sharing threshold logic circuits are used as the inputs of the second-level and third-level differential pairs, respectively. Multiplexers controlled by a configuration memory M_5 are used to select the inputs of the fourth-level differential pairs. An arbitrary two-variable binary function can be realized, if the configuration memories M_1, M_2, M_3 and M_4 are selected to connect with the forth-level differential pairs. The full-adder sum can be generated, if the carry signal $(C_i, \overline{C_i})$ is selected as the input of the forth-level differential pairs.

Figure 2.49 shows the input and output waveforms of the current-source-sharing binary-controlled differential-pair circuit which is pro-

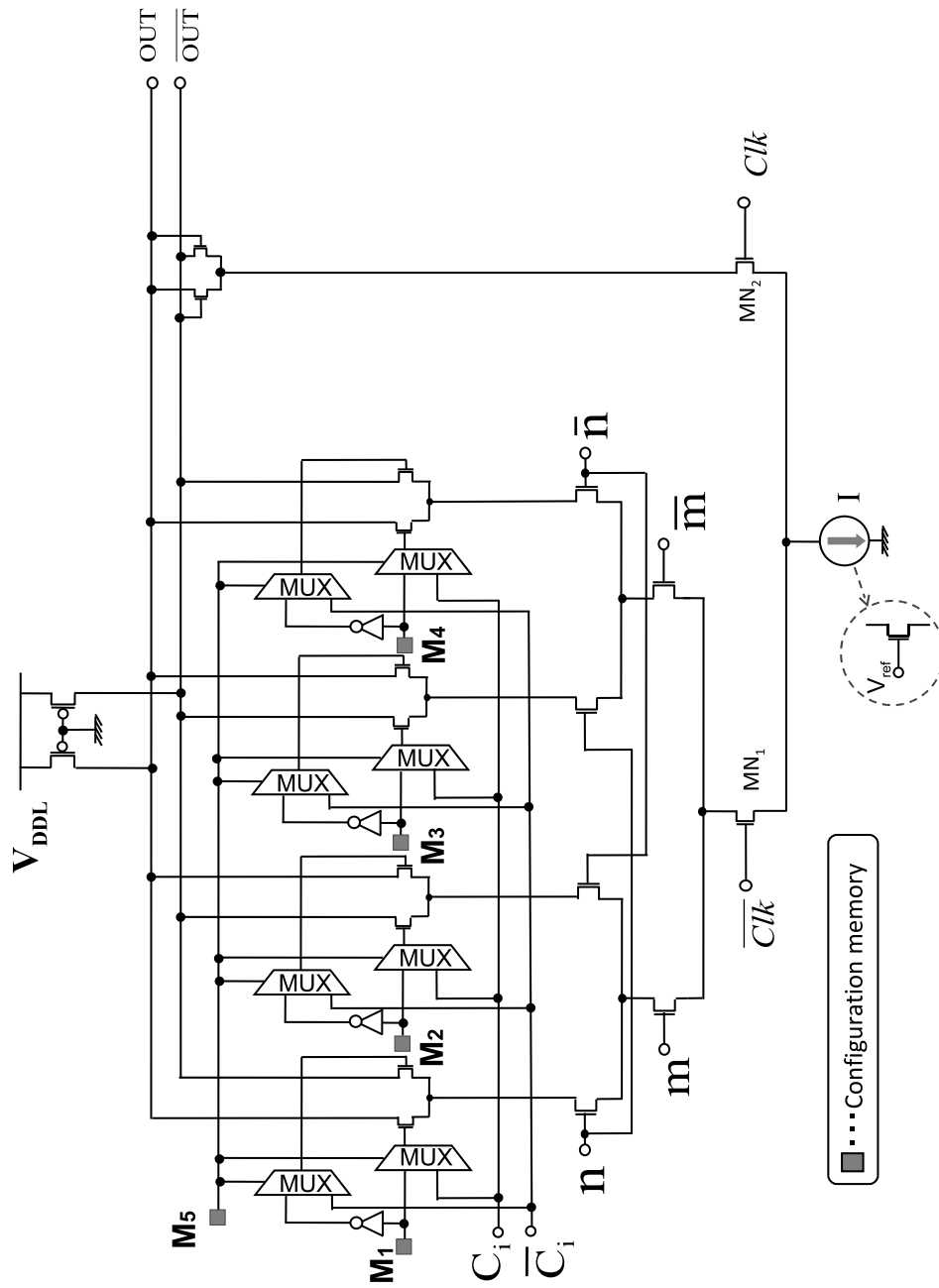


Figure 2.48: Current-source-sharing binary-controlled differential-pair circuit

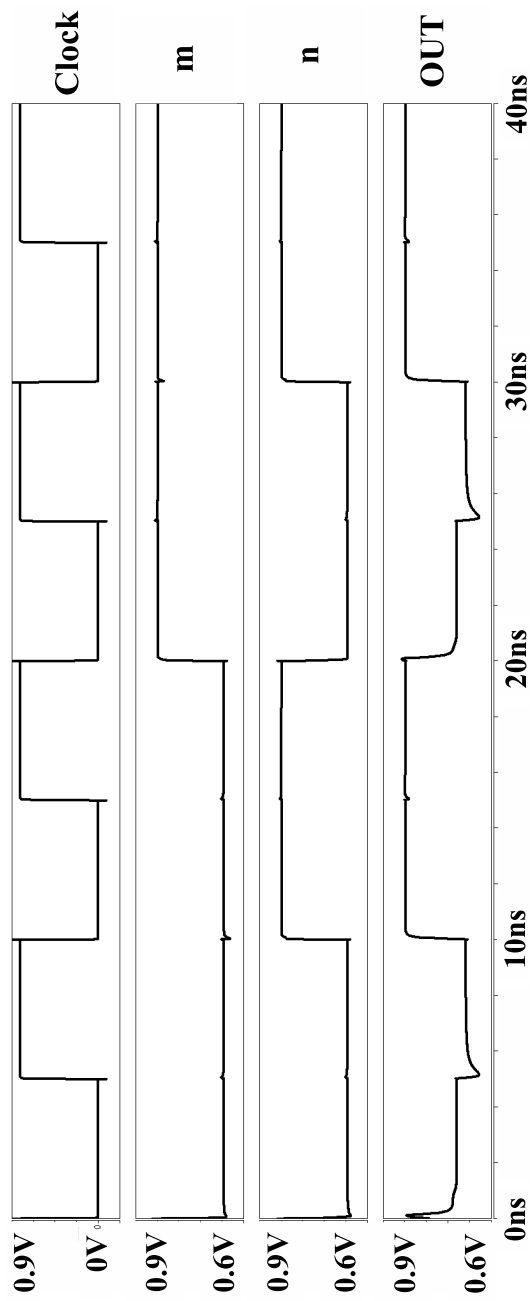


Figure 2.49: Input and output waveforms of the current-source-sharing binary-controlled differential-pair circuit (The two-variable binary function f_5 in Table 2.3 is realized.)

grammed to realize the two-variable binary function f_5 in Table 2.3. The configuration memories M_1 , M_2 , M_3 and M_4 are configured as “0”, “0”, “1” and “1”, respectively. If the input data (m,n) is (0,1) or (1,1), OUT becomes $0.9V$ corresponding to the logic value “1”. On the other hand, if the input data (m,n) is (0,0) or (1,0), OUT becomes $0.6V$ corresponding to the logic value “0”.

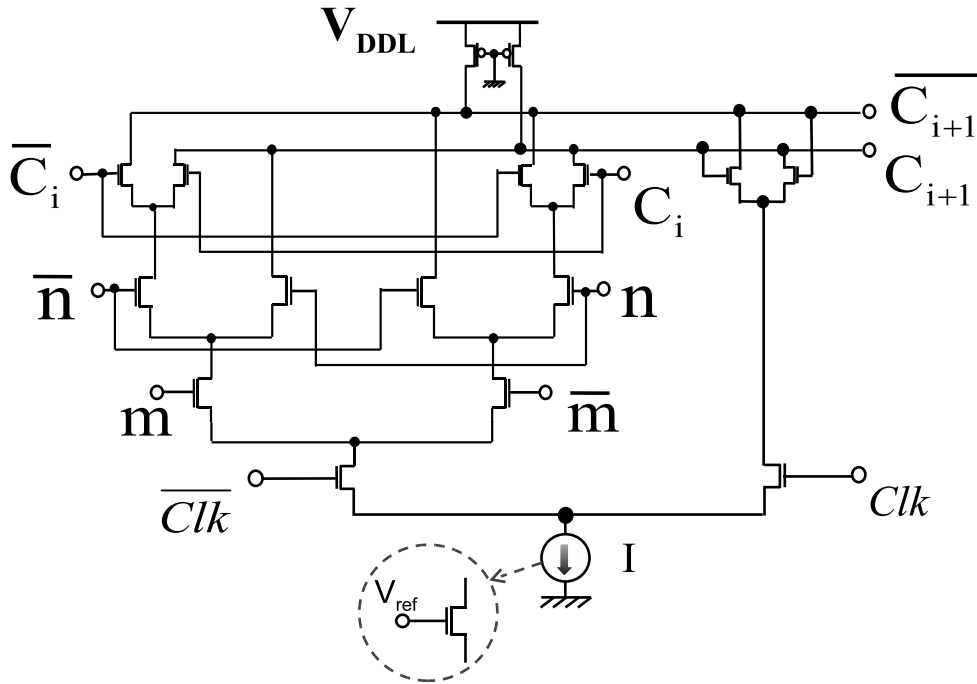


Figure 2.50: Current-source-sharing carry circuit

Figure 2.50 shows the current-source-sharing carry circuit. When Clk is low, the current I flows through the left path and the full-adder carry is generated. When Clk is high, the current I flows through the right path and the full-adder carry is stored by two cross-coupled nMOS

transistors.

The evaluation of the proposed multiple-valued cell is done based on HSPICE simulation using a 65 nm CMOS design rule. The high-supply voltage V_{DDH} , the low-supply voltage V_{DDL} and the unit current I are 1.2V, 0.9V and $10\mu\text{A}$, respectively.

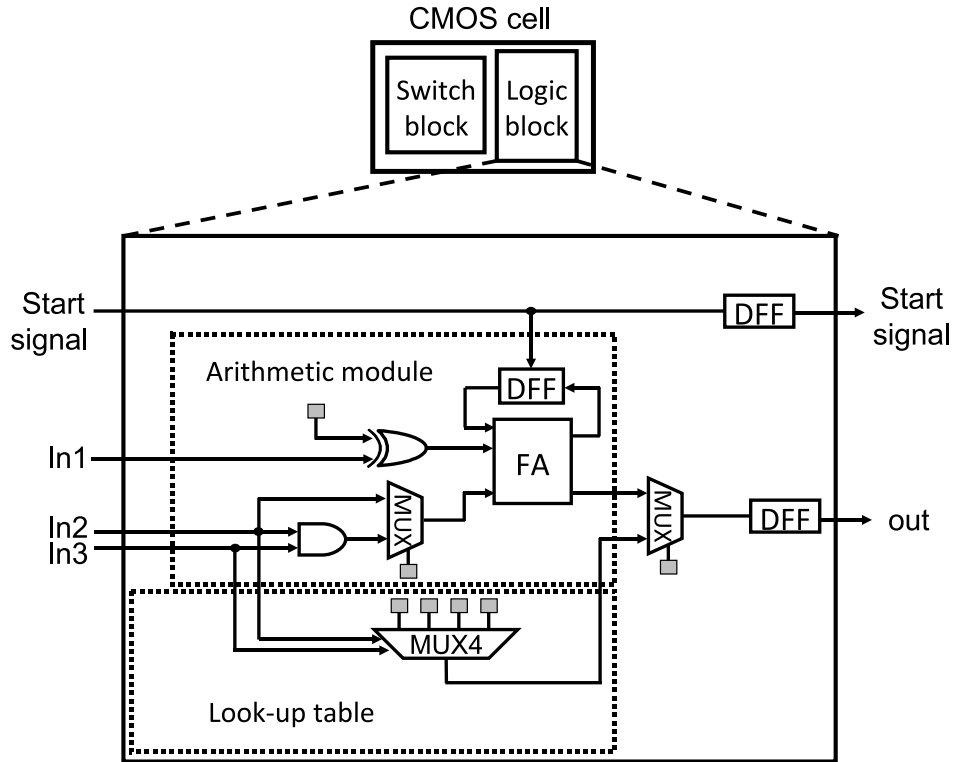


Figure 2.51: Equivalent CMOS cell

The proposed multiple-valued cell is compared with the previous multiple-valued cell, and with the equivalent CMOS cell shown in Fig. 2.51. The equivalent CMOS cell is designed using the library provided by VDEC. Figures 2.52, 2.53 and 2.54 show the layouts of the pro-

posed multiple-valued cell, the previous multiple-valued cell and the equivalent CMOS cell, respectively. The areas of the cells are $576\mu m^2$, $576\mu m^2$ and $706\mu m^2$, respectively.

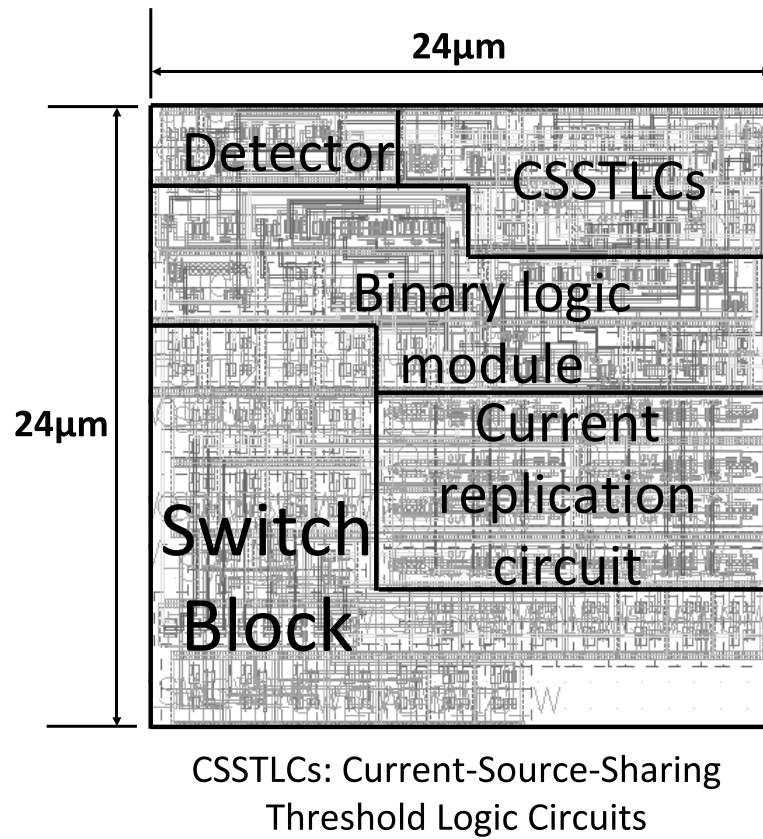


Figure 2.52: Layout of the multiple-valued cell using the binary-controlled differential-pair circuit

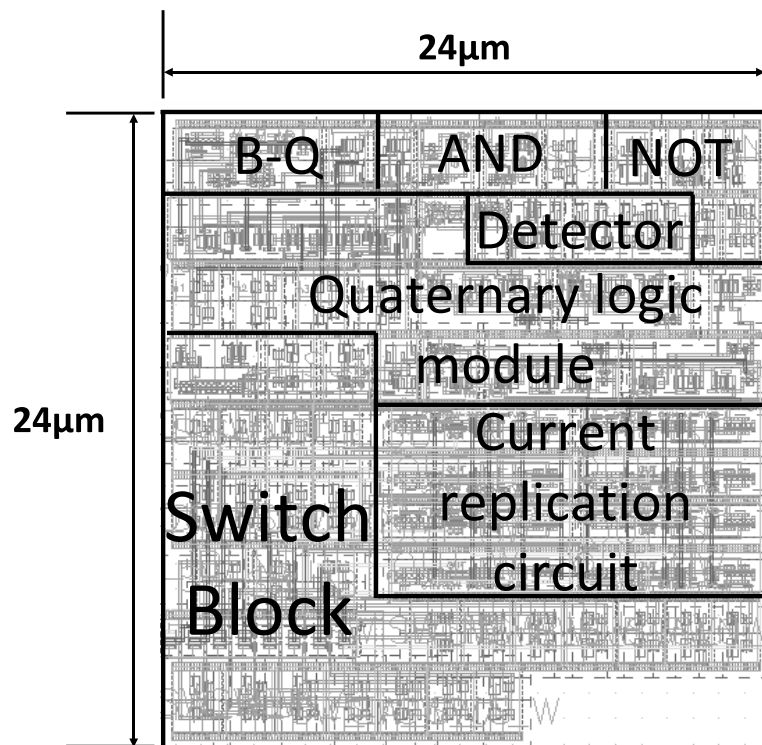


Figure 2.53: Layout of the multiple-valued cell using the quaternary-controlled differential-pair circuit

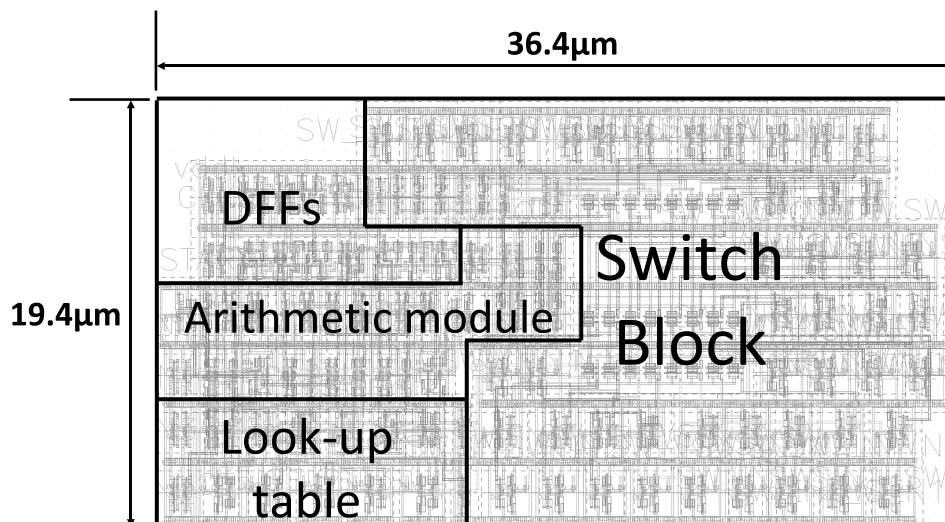


Figure 2.54: Layout of the equivalent CMOS cell

Table 2.7: Comparison results of the cells

	CMOS cell	Previous MV cell	Proposed MV cell
Supply voltage	1.2V	1.2V	$V_{DDH}=1.2V$ $V_{DDL} = 0.9V$
Delay	0.4ns	0.55 ns	0.4 ns
Configuration memory count	55	31	31
Area	$706\mu m^2$	$576\mu m^2$	$576\mu m^2$

* Two-variable binary function

** addition, subtraction, multiplication

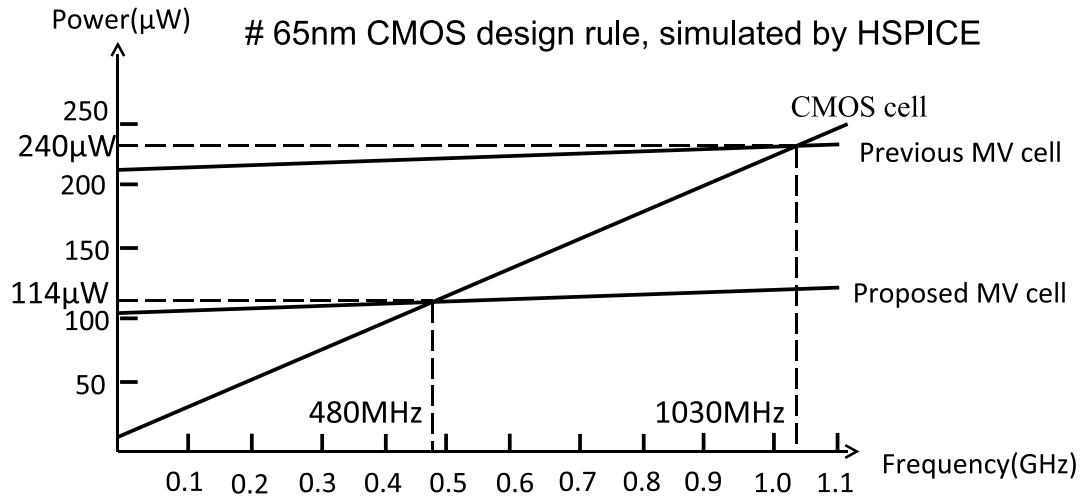


Figure 2.55: Power consumption versus operating frequency in the proposed multiple-valued (MV) cell, the previous MV cell and the equivalent CMOS cell

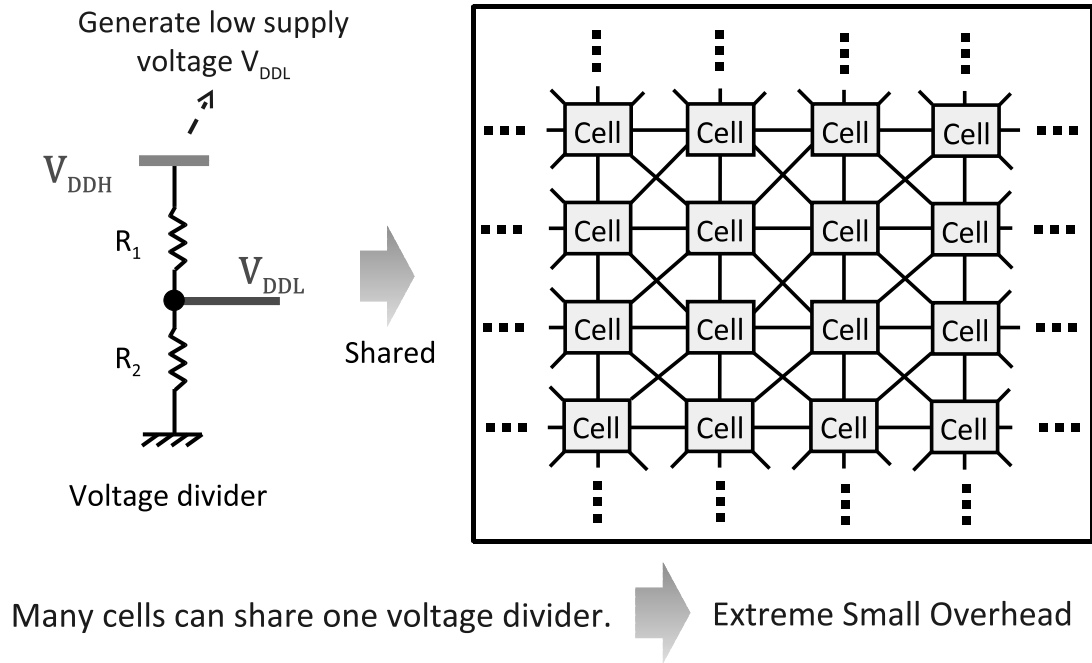


Figure 2.56: Low supply voltage V_{DDL} generator

Table 2.7 shows the comparison results. The delay of the proposed multiple-valued cell is reduced to 72%, in comparison with that of the previous multiple-valued cell. The configuration memory size and the area of the proposed multiple-valued cell are reduced to 56% and 82%, respectively, in comparison with those of the equivalent CMOS cell.

Figure 2.55 shows the characteristic of power consumption versus operating frequency in the cells. The power consumption of the previous multiple-valued cell is reduced to 49% in comparison with that of the proposed multiple-valued cell. The previous multiple-valued cell and the proposed multiple-valued cell have lower power consumption than

the equivalent CMOS cell when the operating frequencies are more than 1030MHz and 480MHz, respectively.

Figure 2.56 shows a low supply voltage V_{DDL} generator. A voltage divider composed of two resistors R_1 and R_2 is used to convert V_{DDH} to V_{DDL} . The V_{DDL} generator is shared by many cells, which leads to extremely small overhead.

Table 2.8: PVT (Process, Supply Voltage, Temperature) corners

PVT corner	Process (PMOS/NMOS)	Supply voltage	Temperature
Slow	Slow/Slow	1.08 V	85°C
Typical	Typical/Typical	1.2 V	25°C
Fast	Fast/Fast	1.32 V	-40°C

Simulation of the proposed multiple-valued cell is done at PVT (Process, Supply Voltage, Temperature) corners shown in Table 2.8. Figure 2.57 shows the input and output waveforms of the proposed multiple-valued cell which is programmed to implement the two-variable binary function f_6 in Table 2.3. If the input data is (0,1) or (1,0), OUT becomes “1”. On the other hand, if the input data is (0,0) or (1,1), OUT becomes “0”.

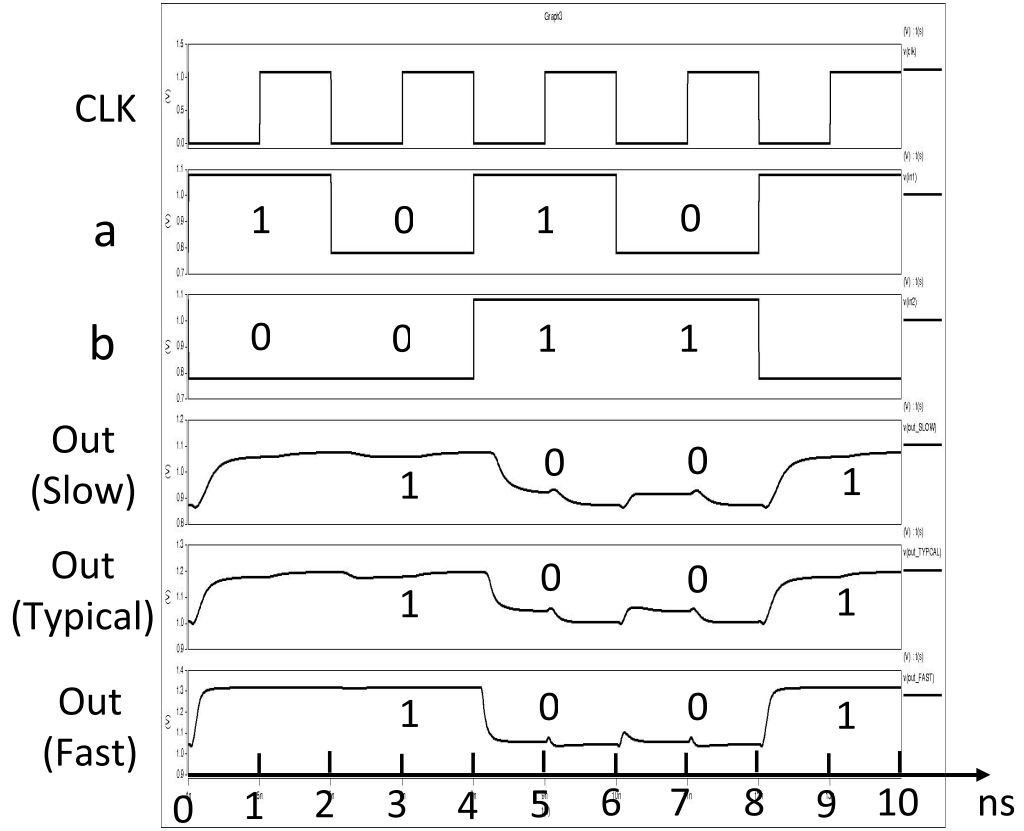


Figure 2.57: Input and output waveforms of the proposed multiple-valued fine-grain cell at PVT corners

2.7 Conclusion

Chapter 2 proposes an area-efficient high-speed low-power multiple-valued cell for the fine-grain reconfigurable VLSI architecture. A multiple-valued switch block and threshold logic circuits are utilized to achieve compactness. A binary-controlled differential-pair circuit is proposed to implement a high-speed low-power arbitrary two-variable binary func-

tion. Also, its increased noise margins are useful to prevent it from producing erroneous outputs in the presence of noisy inputs.

A current-source sharing technique between a series-gating differential-pair circuit and a current-mode D-latch is proposed to improve the utilization of the current source for low power consumption. It is also useful to omit the sample stage in the current-mode D-latch to improve speed.

A dual-supply-voltage multiple-valued source-coupled logic circuit is proposed for low power consumption. A high-supply voltage is provided for multiple-valued signaling, and a low-supply voltage is used to achieve low-power operations in differential-pair circuits without decreasing speed.

It is demonstrated that the power consumption and the delay of the proposed multiple-valued cell are reduced to 49% and 72%, respectively, in comparison with those of the previous multiple-valued cell. The configuration memory size and the area are reduced to 56% and 82%, respectively, in comparison with those of the equivalent CMOS cell. Also, the proposed multiple-valued cell has lower power consumption than the equivalent CMOS cell when the operating frequency is more than 480MHz.

Chapter 3

Area-efficient switch block based on a multiple-valued X-net data transfer scheme

3.1 Overview

This chapter presents a multiple-valued X-net data transfer scheme and its application to the multiple-valued fine-grain reconfigurable VLSI (MVFG-RVLSI). The X-net network inspired by MasPar Computer Corporation [20] is employed for high utilization of one-bit switches for area-efficient switch blocks. In the conventional binary X-net data transfer scheme, only one binary data can be transferred at each “X” intersection called one-to-one 1-bit data transfer, which causes low utilization of the “X” intersection. To solve the problem, a multiple-valued X-net

data transfer scheme is introduced to implement one-to-one two-bit data transfer, two-to-one data transfer and summation for high utilization of the “X” intersection. The multiple-valued X-net data transfer scheme is applied to the MVFG-RVLSI. As a result, in comparison with the previous MVFG-RVLSI using the eight nearest-neighborhood network (8-NNM), the area and the area ratio of the switch blocks of the proposed MVFG-RVLSI based on the multiple-valued X-net data transfer scheme are reduced to 73% and 63%, respectively, without increasing the delay and the power consumption.

3.2 Multiple-valued fine-grain reconfigurable VLSI based on the binary X-net data transfer scheme

Major single instruction multiple data (SIMD) machines appeared contain a neighborhood interconnection network allowing regular data communications. In [20] and [21], a processor array SIMD architecture, called Massively Parallel Computer (MasPar), is presented. The X-net network inspired by the Massively Parallel Computer (MasPar) gathers all the cells in a two-D grid, allowing each cell to communicate with its eight neighbors using a binary data transfer scheme[11].

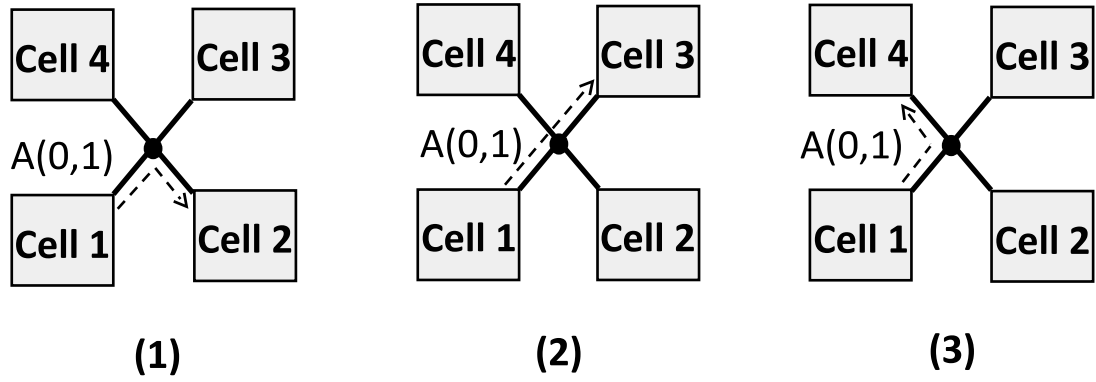


Figure 3.1: One-to-one data transfer at each “X” intersection in the binary X-net data transfer scheme

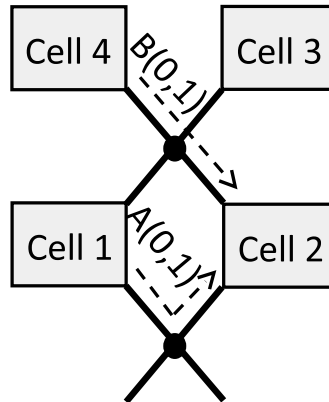


Figure 3.2: Two-to-one data transfer in a binary data transfer scheme

To transfer a data from $cell_i$ to its right adjacent $cell_{i+1}$, the $cell_i$ transmits out its northeast corner and the $cell_{i+1}$ reads from its northwest corner (one-to-one data transfer). Figure 3.1 shows three kinds of one-to-one data transfer modes at each “X” intersection. One binary data A can be transferred from the $cell\ 1$ to the adjacent $cell\ 2$, $cell\ 3$ and $cell\ 4$.

However, if two binary data A and B are transferred from the *cell* 1 and *cell* 4 to the common adjacent *cell* 2 simultaneously (two-to-one data transfer), two “X” intersections are required in the binary data transfer scheme as shown in Fig. 3.2, which results in low utilization of the “X” intersection.

The cell of the MVFG-RVLSI using the X-net network is fabricated using a 65 nm CMOS process. The supply voltage is 1.2V. Figure 3.3 shows the chip photomicrograph and the layout of the cell of the MVFG-RVLSI using the X-net network. The area of the cell is $422 \mu m^2$, in a 65nm CMOS technology. As a result, the cell area of the MVFG-RVLSI using the X-net network is reduced to 73% in comparison with that of the MVFG-RVLSI using the eight nearest-neighbor mesh network (8-NNM). The logic block area and the switch block area of the cell in the MVFG-RVLSI using the X-net network are reduced to 84% and 50%, respectively, in comparison with those of the MVFG-RVLSI using the 8-NNM. Figure 3.4 shows the inputs and outputs waveforms of the cell of the MVFG-RVLSI using the X-net network in the chip. The cell is programmed to implement a NOT gate with the DFF.

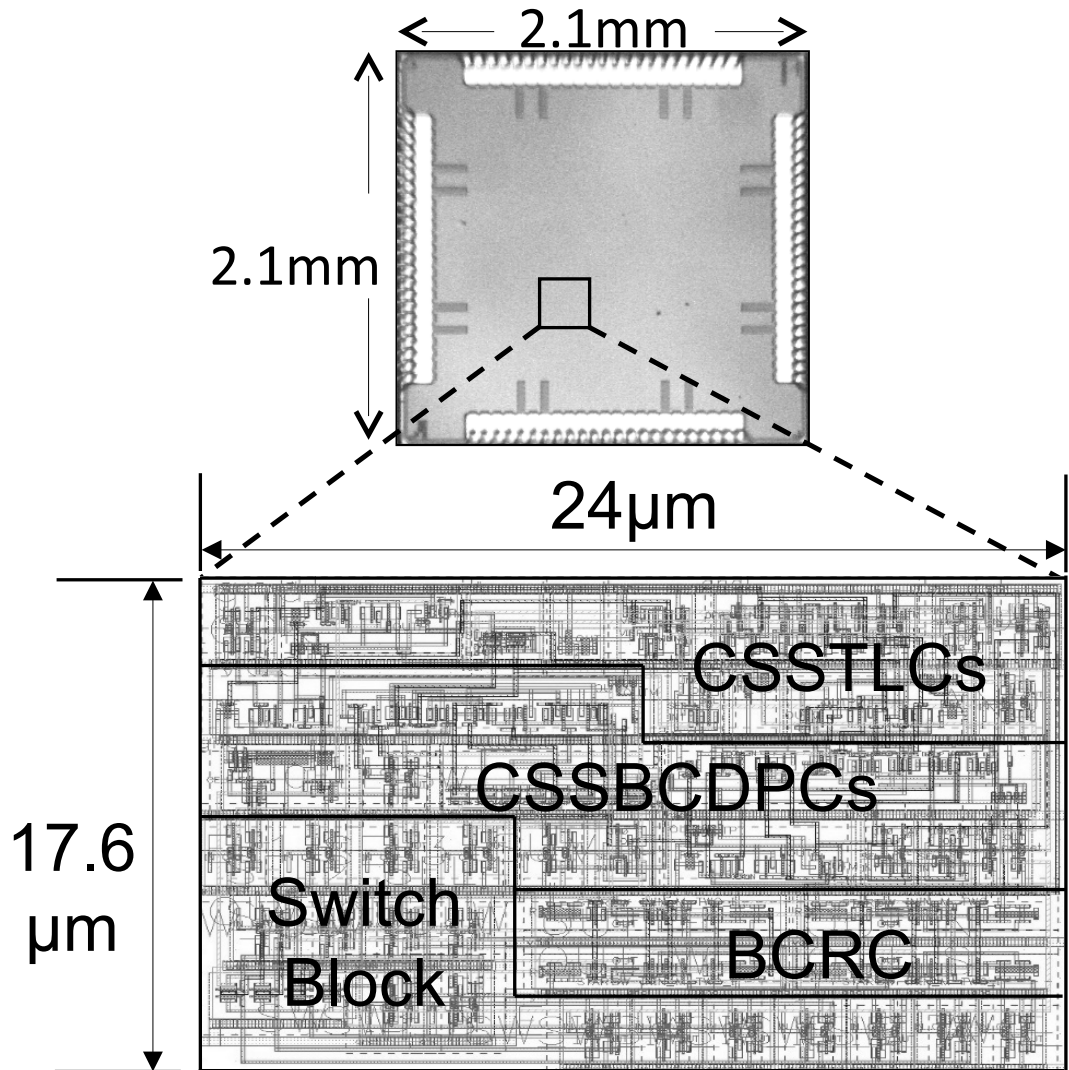


Figure 3.3: Chip photomicrograph and the layout of the cell in the MVFG-RVLSI using the X-net network

Let us map a 2×2 -bit multiplier onto the MVFG-RVLSIs. Figure 3.5 shows the scheduling and allocation of the 2×2 -bit multiplier. Figures

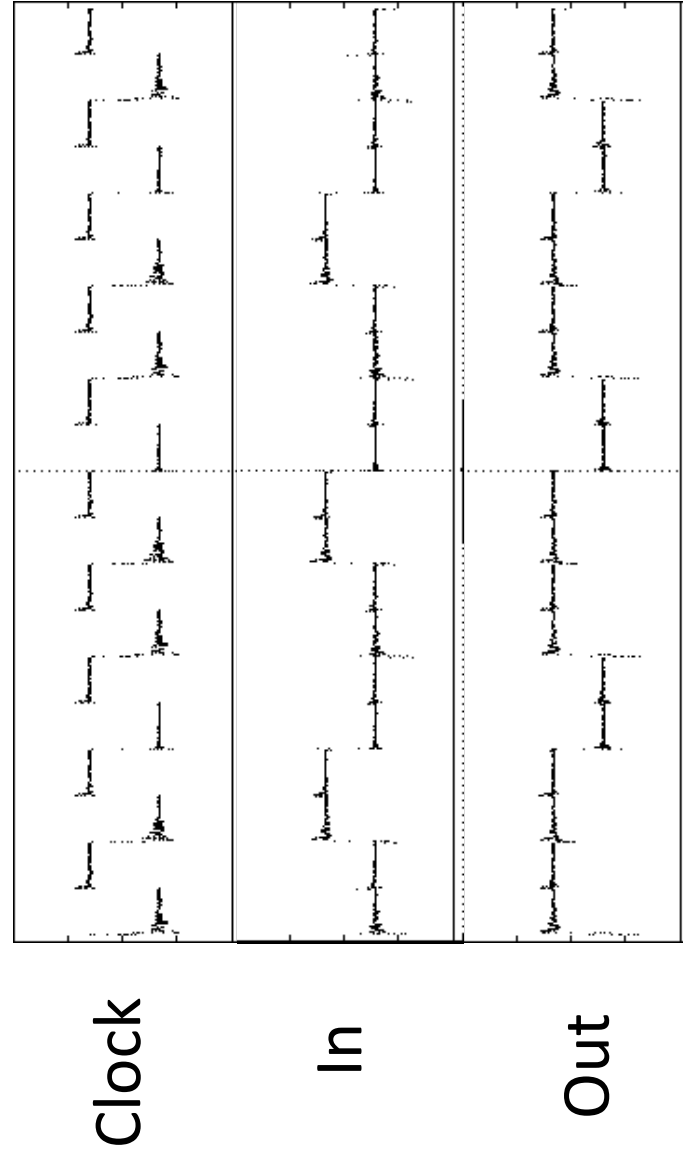


Figure 3.4: Inputs and outputs waveforms of the cell in the MVFG-RVLSI using the X-net network in the chip

3.6 and 3.7 show the allocation results for the MVFG-RVLSI using the 8-NNM and the MVFG-RVLSI based on the binary X-net data transfer scheme (BX-DTS), respectively. Table 3.1 shows the comparison results. The configuration memory count and the area of the MVFG-RVLSI based on the BX-DTS is reduced by 21% and 6%, respectively, in comparison with those of the MVFG-RVLSI using the 8-NNM. However, the computation time and the power consumption are increased by 25% and 27%, respectively.

The reason why the computation time and the power consumption are increased is that only one binary data can be transferred among cells in the BX-DTS. Simultaneously, both the cells in the X-net network and the 8-NNM can be programmed to implement both an AND circuit and a full adder which are basic components of the 2×2 -bit multiplier. In the MVFG-RVLSI using the 8-NNM where the start signal can be superposed with the data signal, we can map both the AND circuit and the full adder in one cell as shown in Fig. 3.6. In the MVFG-RVLSI based on the BX-DTS where the start signal and the data signal are transferred separately, we need to map the AND circuit and the full adder onto two cells as shown in Fig. 3.7, which increases the power consumption and computation time. To solve these problems, we propose a MVFG-RVLSI based on a multiple-valued X-net data transfer scheme (MVX-DTS).

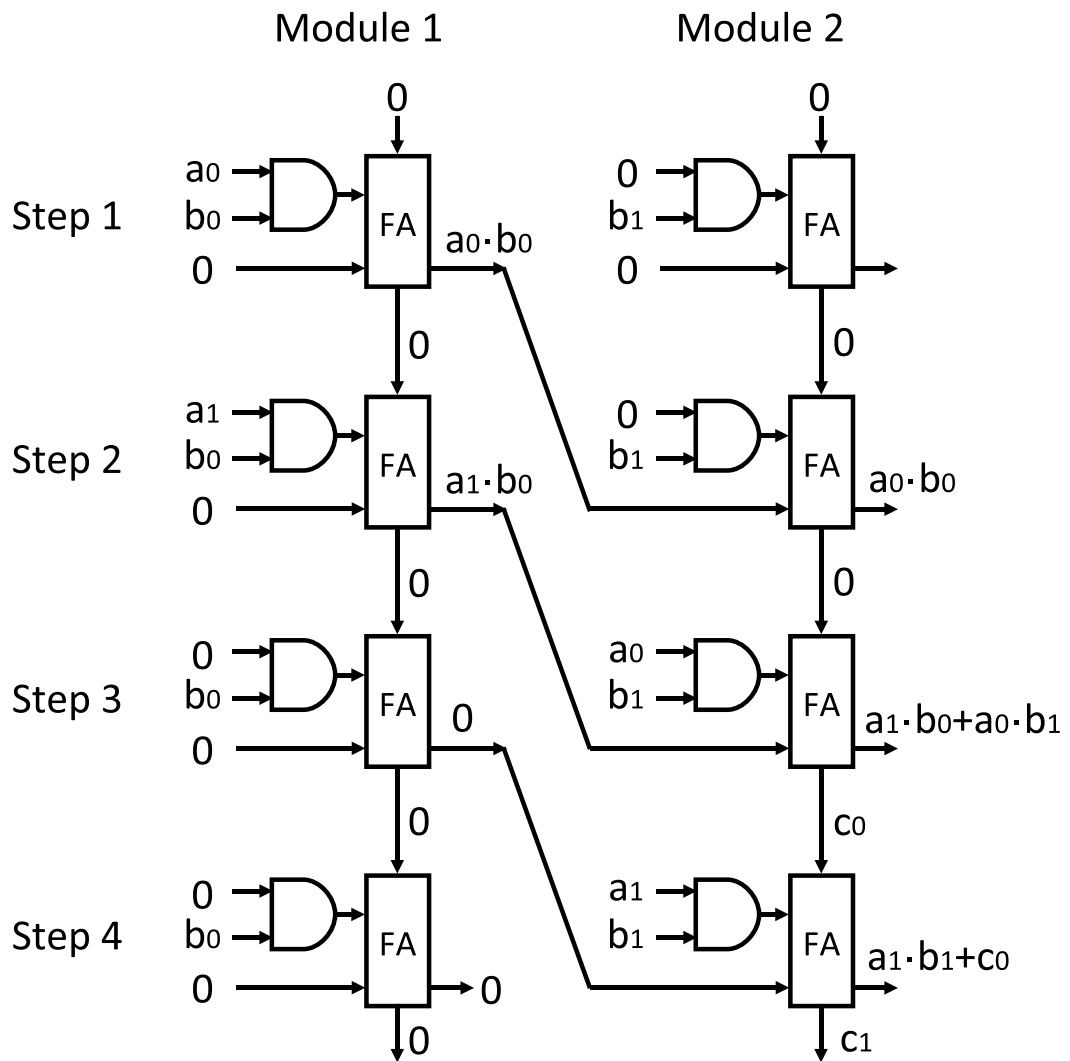


Figure 3.5: Scheduling and allocation of the 2×2 -bit multiplier

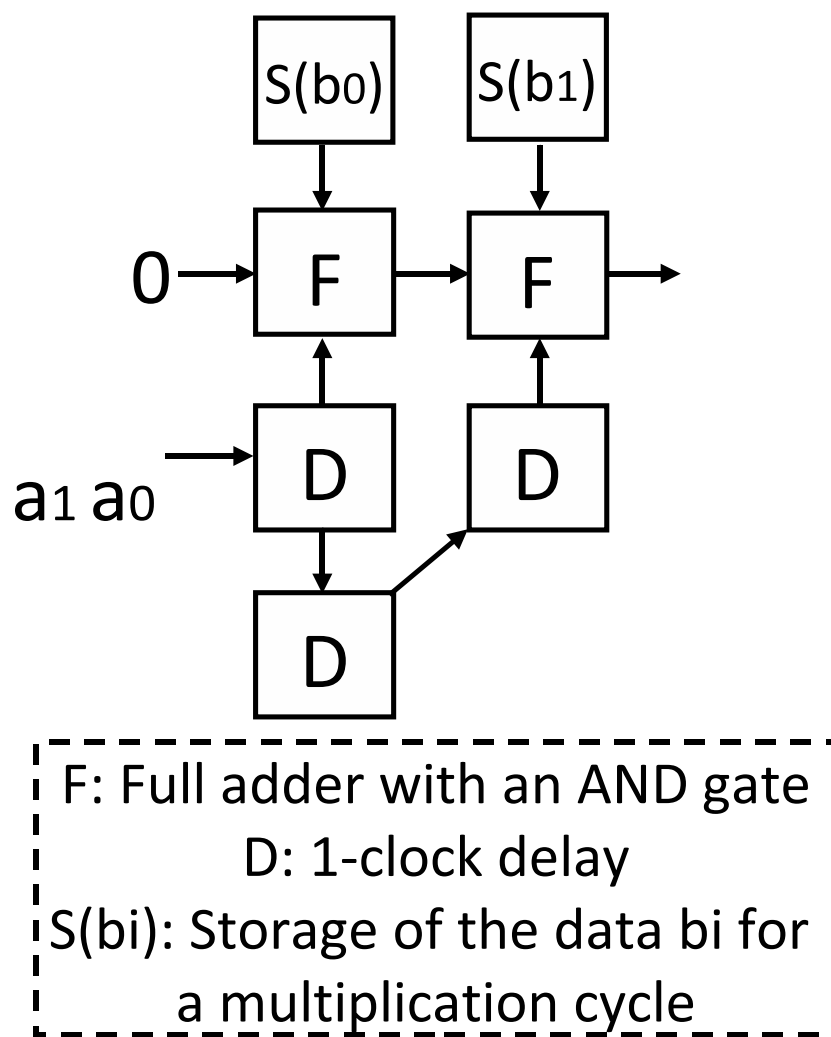
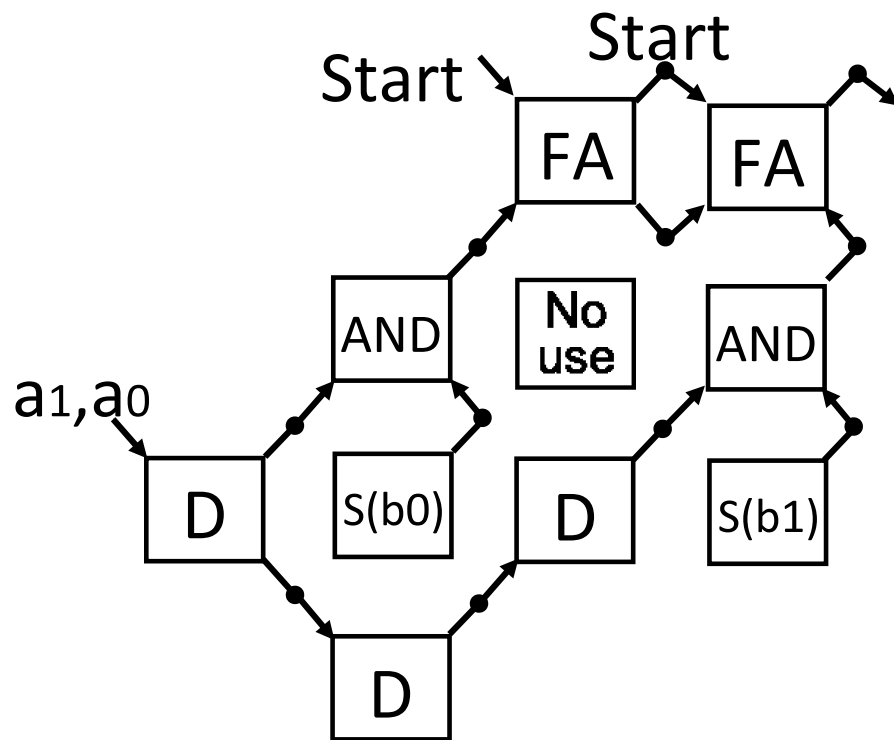


Figure 3.6: Allocation result of the 2×2 -bit multiplier for the multiple-valued reconfigurable VLSI using the eight nearest-neighbor mesh network



F: Full adder with an AND gate
 D: 1-clock delay
 S(bi): Storage of the data bi for
 a multiplication cycle

Figure 3.7: Allocation result of the 2×2 -bit multiplier for the multiple-valued reconfigurable VLSI based on the binary X-net data transfer scheme

Table 3.1: Comparison results of multiple-valued reconfigurable VLSIs in the 2×2 -bit multiplication

	MVFG-RVLSI using 8-NNM	MVFG-RVLSI based on BX-DTS
Supply voltage	1.2V	1.2V
Computation time	1.6 ns	2.0 ns
Cell count	7	9
Configuration memory count	217	171
Area	$4032 \mu m^2$	$3798 \mu m^2$
Power consumption @800MHz	$906 \mu W$	$1150 \mu W$

3.3 Multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme

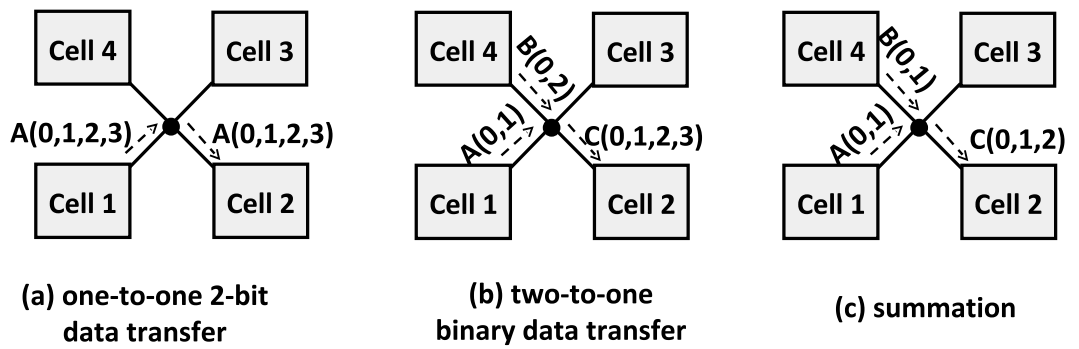


Figure 3.8: Three data transfer patterns at each “X” intersection in the multiple-valued X-net data transfer scheme

In the MVX-DTS, multiple-valued current signals are transferred among cells. Two binary data A and B from two adjacent cells can be transferred to one common adjacent cell at each “X” intersection (two-to-one data transfer) as shown in Fig. 3.8(b). A and B should be (0, 1) and (0, 2), respectively, and C becomes a quaternary data (0, 1, 2, 3) which expresses two-bit information. On the other hand, summation of A and B can be realized at each “X” intersection as shown in Fig. 3.8(c). A and B should be (0, 1) and (0, 1), respectively, and C becomes a ternary data (0, 1, 2). All the one-to-one two-bit data transfer, the two-to-one binary data transfer and the summation can be realized at each “X” intersection in the MVX-DTS as shown in Fig. 3.8, which leads to high utilization of the “X” intersection.

In the MVFG-RVLSI based on the MVX-DTS, there are two methods to realize the linear summation of the binary input currents A and B. One is that A and B are linearly summed at the “X” intersection, if A and B are transferred from a common “X” intersection. The other is that A and B are linearly summed in the switch block, if A and B are transferred from two different “X” intersections. In a bit-serial operation, a start signal indicating a head of a one-word data is required to initialize D flip-flops used for a state memory. Superposition of the binary input current C and the start signal in a single interconnection is introduced to

Table 3.2: Comparison results of multiple-valued reconfigurable VLSIs in the 2×2 -bit multiplication

	MVFG-RVLSI using the 8-NNM	MVFG-RVLSI based on the MVX-DTS
Supply voltage	1.2V	1.2V
Computation time	1.6 ns	1.6 ns
Cell count	7	7
Configuration memory count	217	133
Area	$4032 \mu m^2$	$2954 \mu m^2$
Power consumption @800MHz	$904 \mu W$	$904 \mu W$

implement compact switch blocks, where the logic value “1” and “0” is defined as C and the logic value “2” is defined as the start signal.

Figure 3.9 shows the allocation result of the 2×2 -bit multiplier for the MVFG-RVLSI based on the MVX-DTS. Table 3.2 shows the comparison results. The configuration memory count and the area of the MVFG-RVLSI based on the MVX-DTS are reduced to 61% and 73%, respectively, in comparison with those of the MVFG-RVLSI using the 8-NNM, while the computation time, the power consumption and the cell count are kept same.

3.4 Evaluation of the multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme

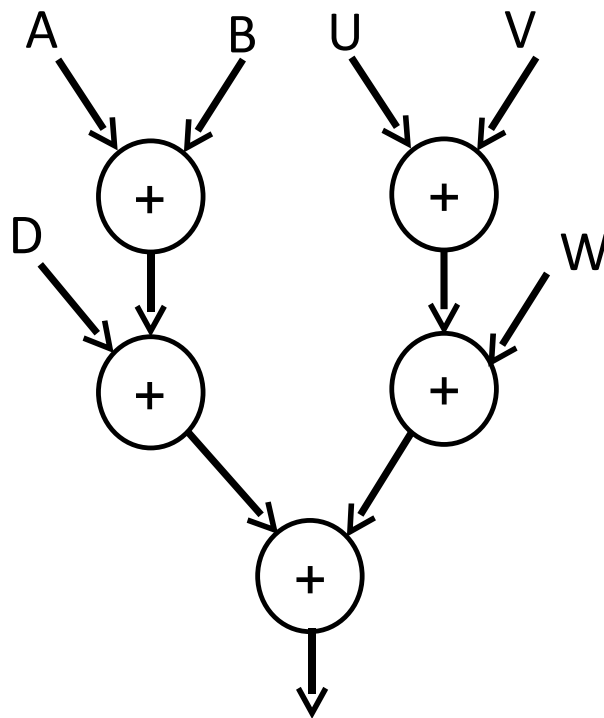


Figure 3.10: Data flow graph for the 6-input addition

Let us consider a 6-input addition, which is one of the fundamental arithmetic operations. Figure 3.10 shows its data follow graph (DFG). Figures 3.11 and 3.12 show the allocation results for the MVFG-RVLSI using the 8-NNM and the MVFG-RVLSI based on the MVX-DTS, respectively. Table 3.3 shows the comparison result. The configuration

memory count and the area of the MVFG-RVLSI the MVX-DTS are reduced to 61% and 73%, respectively, in comparison with those of the MVFG-RVLSI using the 8-NNM. The computation time, the power consumption and the cell count of the MVFG-RVLSI the MVX-DTS are same as those of the MVFG-RVLSI using the 8-NNM.

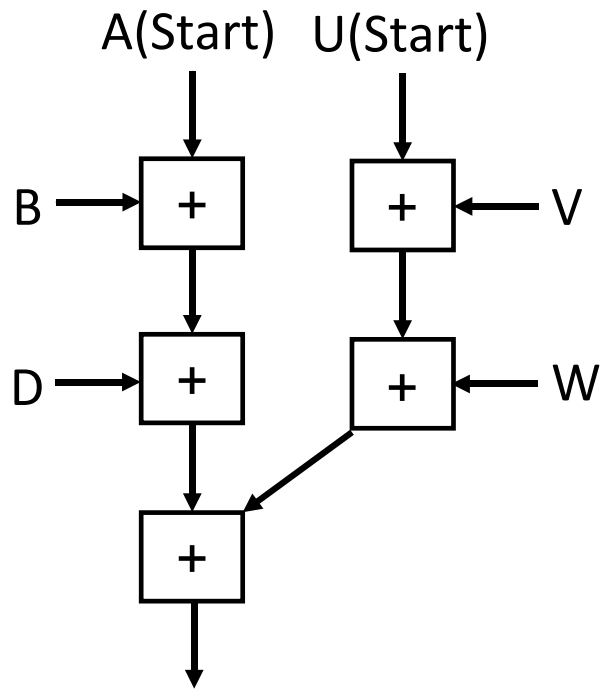


Figure 3.11: Allocation of the 6-input addition onto the multiple-valued fine-grain reconfigurable VLSI using the eight nearest-neighbor mesh network

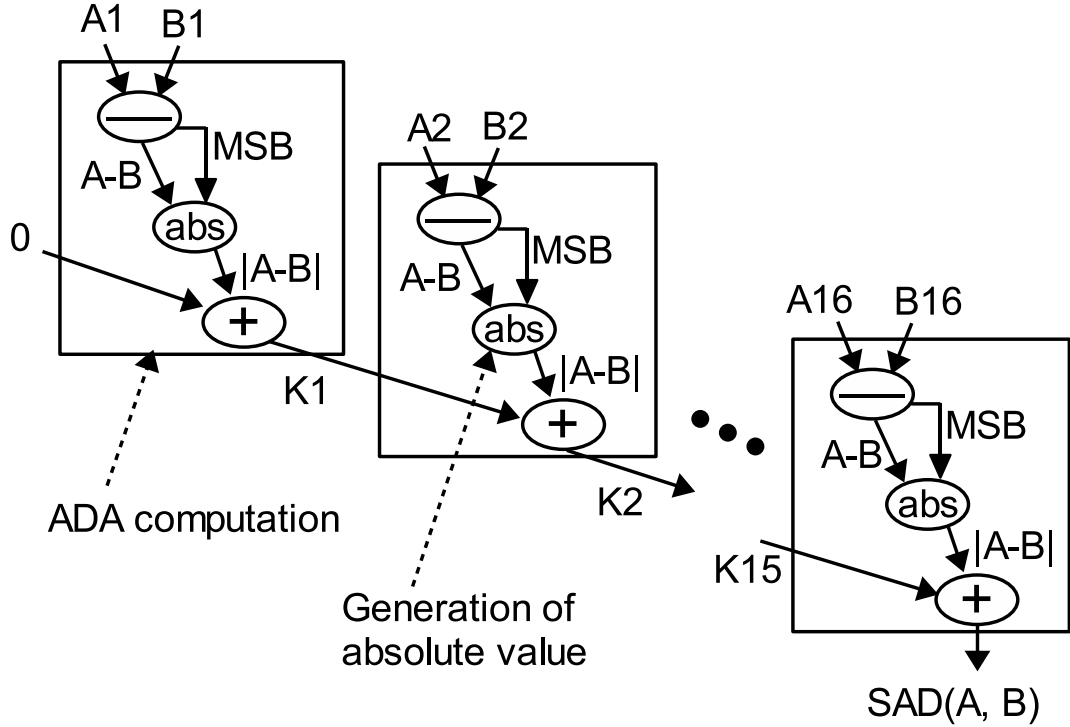


Figure 3.13: Control/data flow graph for a sum-of-absolute-differences operation

Let us consider a sum-of-absolute-differences (SAD) operation which is widely used as a similarity measure in template matching. The sum-of-absolute-differences operation is expressed as

$$SAD = |A1 - B1| + |A2 - B2| + \dots + |A16 - B16| \quad (3.1)$$

where the CDFG is shown in Fig. 3.13. The sum-of-absolute-differences operation is performed by iteration of an absolute difference operation and addition (ADA). Figures 3.14, 3.15 and 3.16 show the allocation results of the 8-bit ADA for the MVFG-RVLSI using the 8-NNM, the MVFG-RVLSI based on the BX-DTS and the MVFG-RVLSI based on

the MVX-DTS, respectively. Table 3.4 shows the comparison results. The configuration memory count and the area of the MVFG-RVLSI based on the BX-DTS are reduced to 73% and 87%, respectively, in comparison with the MVFG-RVLSI using the 8-NNM. The computation time and the cell count are increased by 20% and 19%, respectively. The configuration memory count and the area of the MVFG-RVLSI based on the MVX-DTS are reduced to 61% and 73%, respectively, in comparison with the MVFG-RVLSI using the 8-NNM, while the computation time, the power consumption and the cell count are kept same.

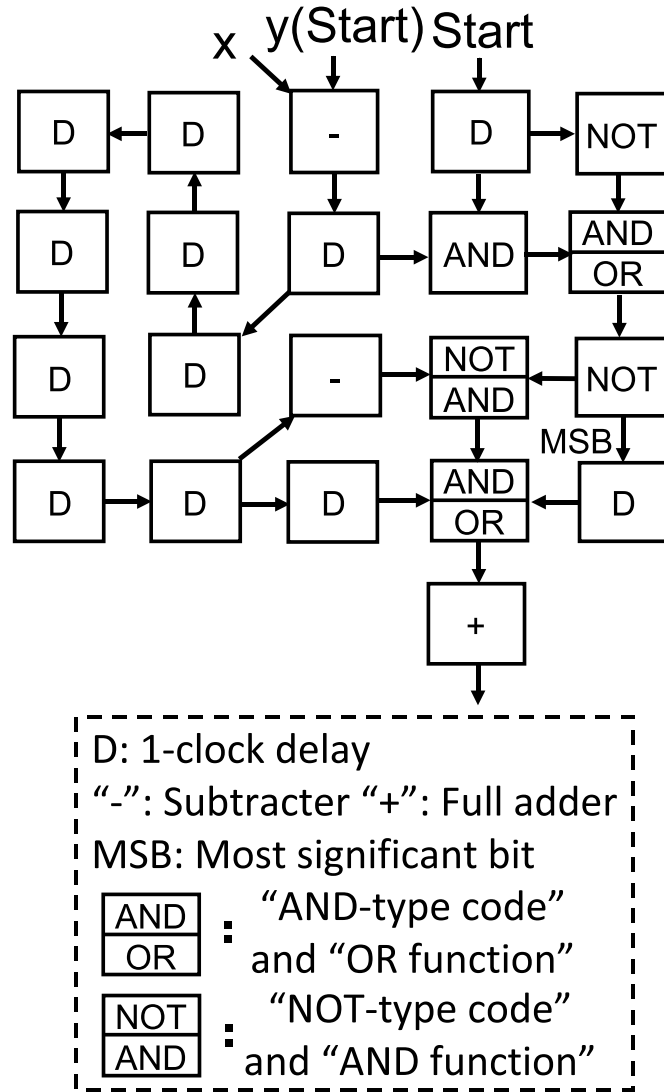
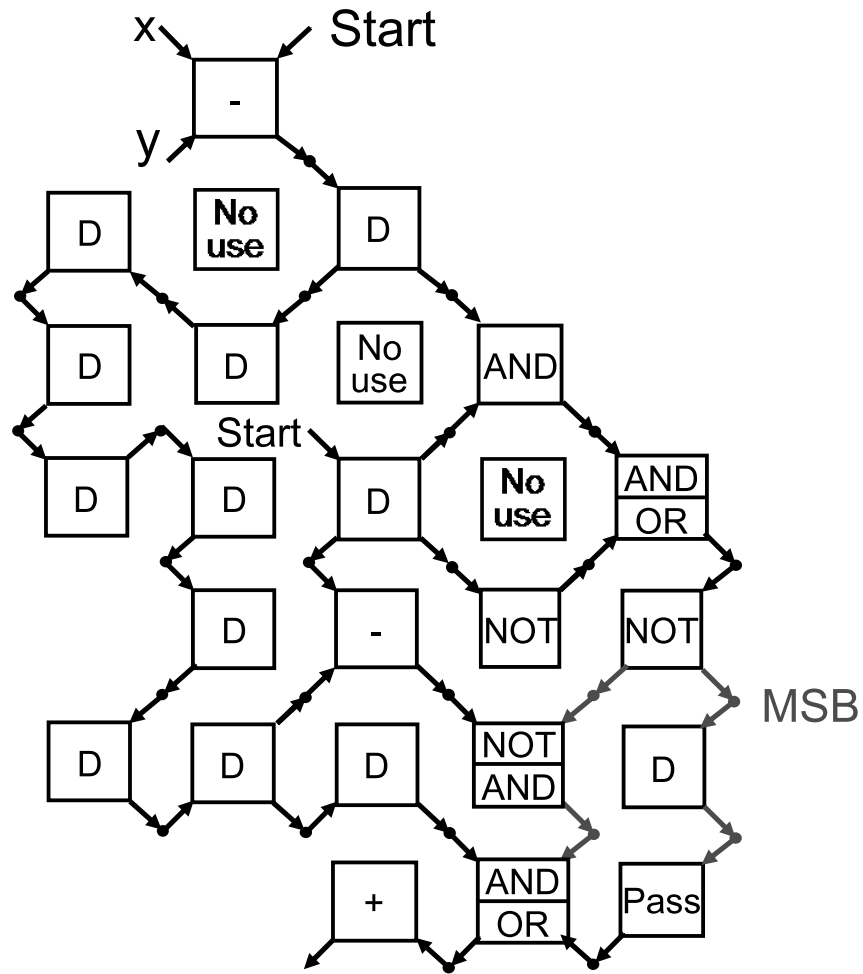


Figure 3.14: Allocation result of the absolute difference operation and addition for the multiple-valued fine-grain reconfigurable VLSI using the eight nearest-neighbor mesh network



MSB: Most significant bit

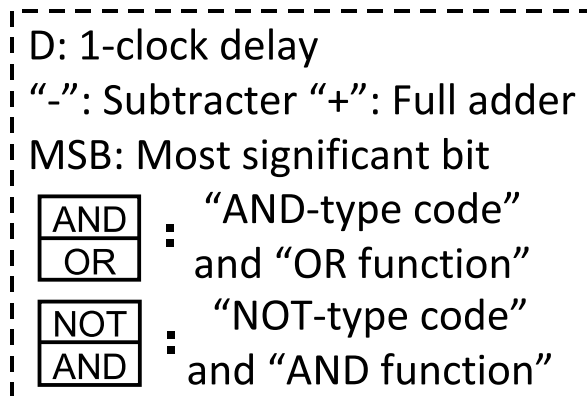


Figure 3.15: Allocation result of the absolute difference operation and addition for the multiple-valued fine-grain reconfigurable VLSI based on the binary X-net data transfer scheme

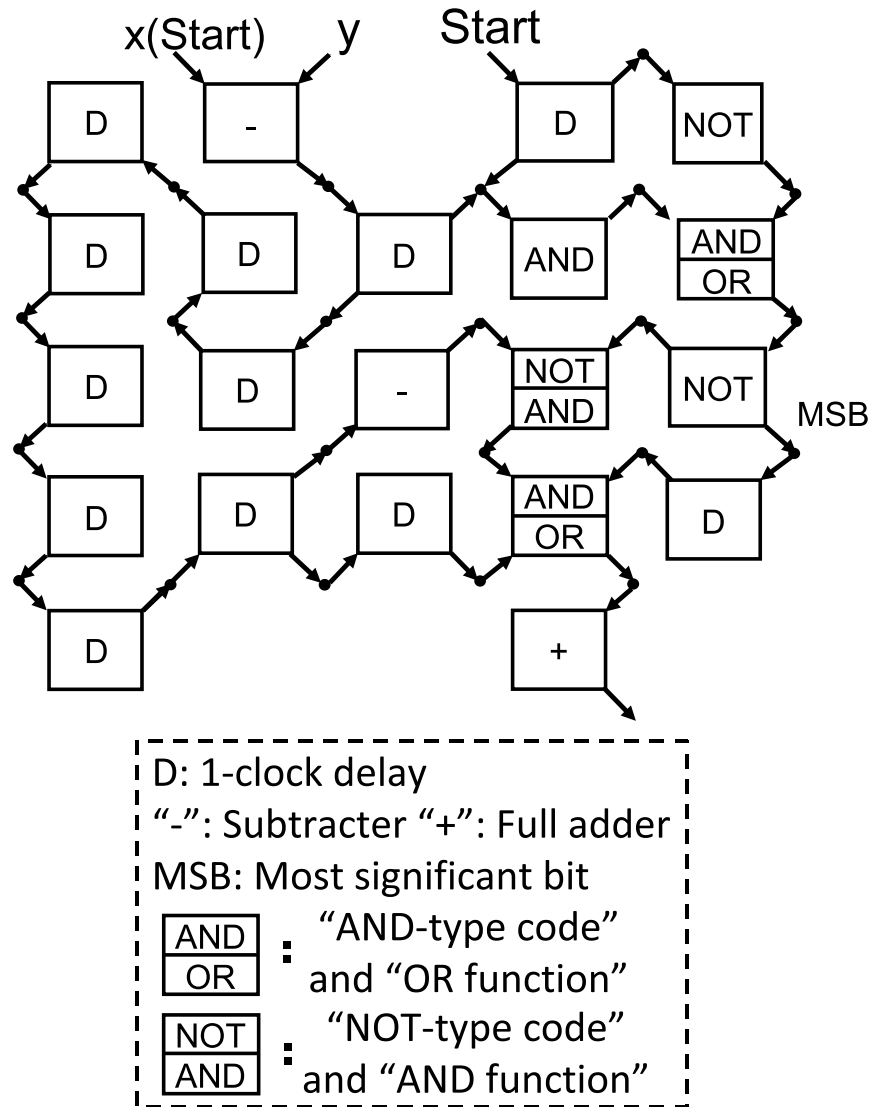


Figure 3.16: Allocation result of the absolute difference operation and addition for the multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme

Table 3.4: Comparison results of the multiple-valued fine-grain reconfigurable VLSIs (MVFG-RVLSIs) in the sum-of-absolute-differences operation

	MVFG-RVLSI using MVFG-RVLSI based on MVFG-RVLSI based on			
	8 nearest-neighbor	binary X-net data	multiple-valued X-net	
	mesh network	transfer scheme	data transfer scheme	
Supply voltage	1.2V	1.2V	1.2V	1.2V
Computation time	8 ns	10 ns	8 ns	8 ns
Cell count	21	25	21	21
Configuration memory count	651	475	399	399
Area	12096 μm^2	10550 μm^2	8862 μm^2	8862 μm^2
Power consumption @ 800MHz	2640 μW	2640 μW	2640 μW	2640 μW

3.5 Conclusion

Chapter 3 proposes a multiple-valued X-net data transfer scheme to realize an area-efficient multiple-valued fine-grain reconfigurable VLSI without increasing the delay and power consumption. The X-net network is effectively employed for reducing the complexity of the interconnections and switch blocks in the multiple-valued fine-grain reconfigurable VLSI. The multiple-valued X-net data transfer scheme is proposed to improve the utilization of the X-net network, which leads to high speed, low power consumption and small area in comparison with a conventional binary X-net data transfer scheme. It is demonstrated that the configuration memory count and the area of the multiple-valued fine-grain reconfigurable VLSI based on the multiple-valued X-net data transfer scheme are reduced to 61% and 73%, respectively, in comparison with those of the multiple-valued fine-grain reconfigurable VLSI using the eight nearest-neighbor mesh network.

Chapter 4

High-performance long-distance data transfer using a dynamic tree network

4.1 Overview

In the chapter 3, the X-net network has been proposed to realize simple interconnections and compact switch blocks for eight-near neighborhood data transfer in the multiple-valued fine-grain reconfigurable VLSI (MVFG-RVLSI). However, not only localized data transfer but also long-distance data transfer between cells, and between a cell and a data memory is necessary for practical applications. In the MVFG-RVLSI using only the X-net network, many cells are required for the long-distance data transfer, which causes low utilization of the cells. This chapter presents a global dynamic tree network for high-performance

long-distance data transfer in the MVFG-RVLSI. Moreover, a logic-in-memory architecture is employed for solving data transfer bottleneck between a block data memory and a cell. To evaluate the MVFG-RVLSI, a fast Fourier transform (FFT) operation is mapped onto a previous MVFG-RVLSI using only the X-net network and the MVFG-RVLSI using a global tree local X-net network (GTLX). As a result, the computation time, the power consumption and the transistor count of the MVFG-RVLSI using the GTLX are reduced by 25%, 36% and 56%, respectively, in comparison with those of the MVFG-RVLSI using only the X-net network.

4.2 Long-distance data transfer in the multiple-valued reconfigurable VLSI using only the X-net network

Figure 4.1 shows the long-distance data transfer between the cells A and B in the MVFG-RVLSI using only the X-net network. In the Cell 1, two one-bit switches S1 and S2 are turned ON to pass data, which results in low speed and low utilization of the cell. The Cell 2 is programmed as a D flip-flop (DFF) to amplify a voltage data signal and improve throughput, which causes low speed, large power consumption and low utilization of the cell.

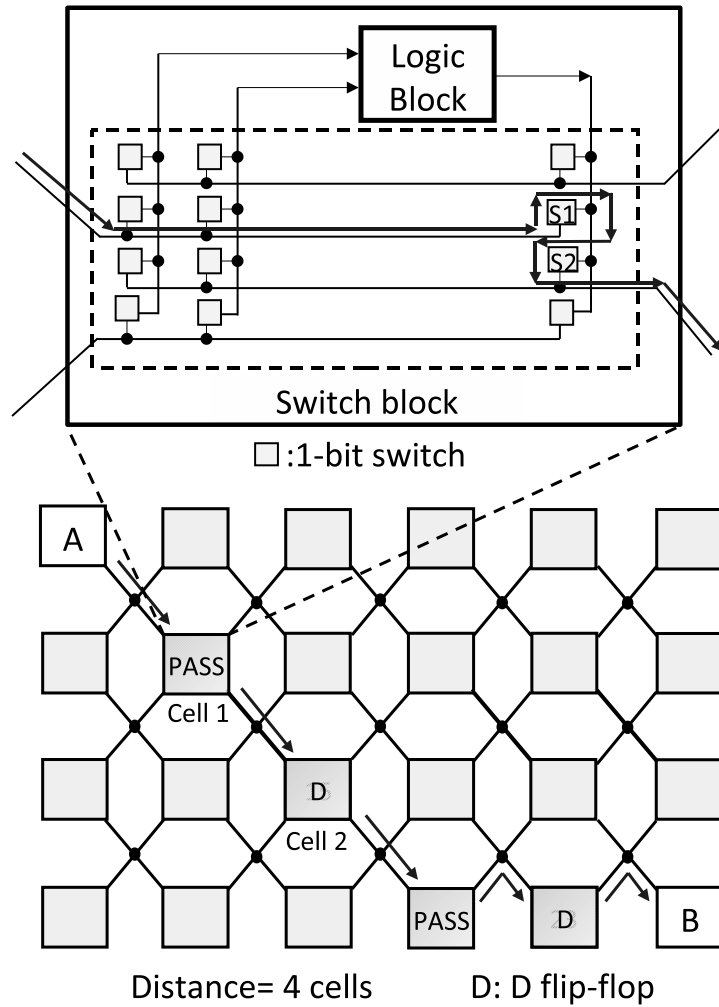


Figure 4.1: Long-distance data transfer in the multiple-valued fine-grain reconfigurable VLSI using only the X-net network

Figure 4.2 shows the data memory location in the MVFG-RVLSI using only the X-net network. Many small-sized data memories are located around the boundary of a cell array composed of a large number of cells. Each cell is connected to eight adjacent cells by the X-net network.

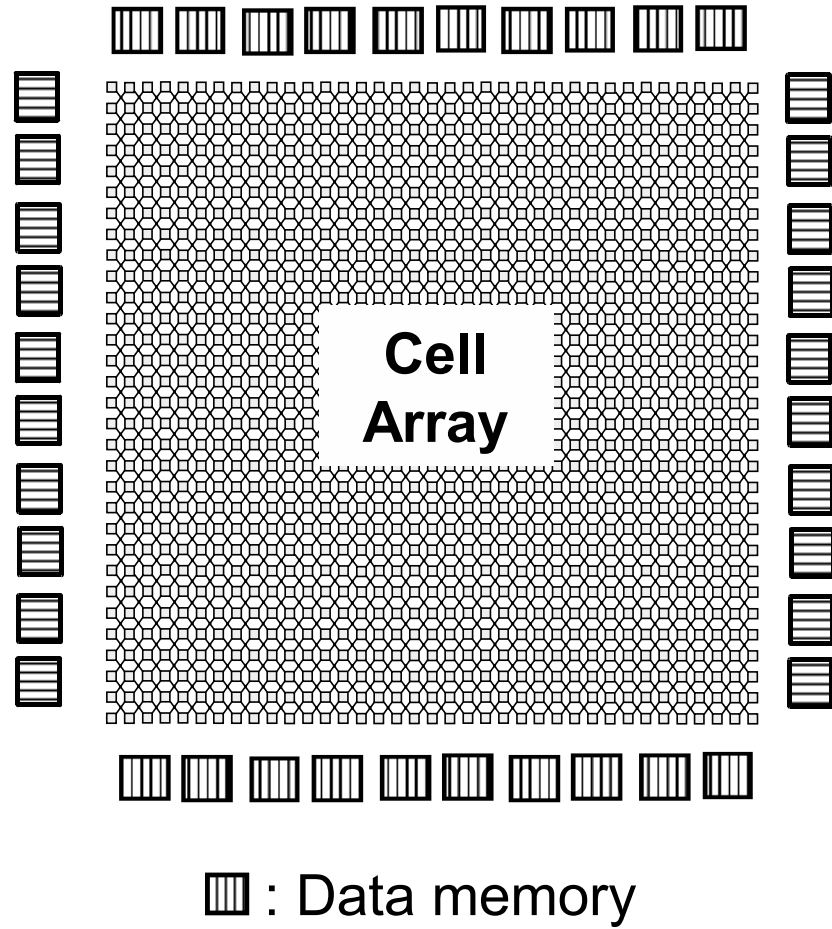


Figure 4.2: Data memory location in the multiple-valued fine-grain reconfigurable VLSI

As shown in Fig. 4.3, each data memory is connected to several edge cells by registers. So that, to realize data access between the data memory and a non-edged cell C, many cells are used for data relay, which results in low speed, large power consumption and low utilization of the cells.

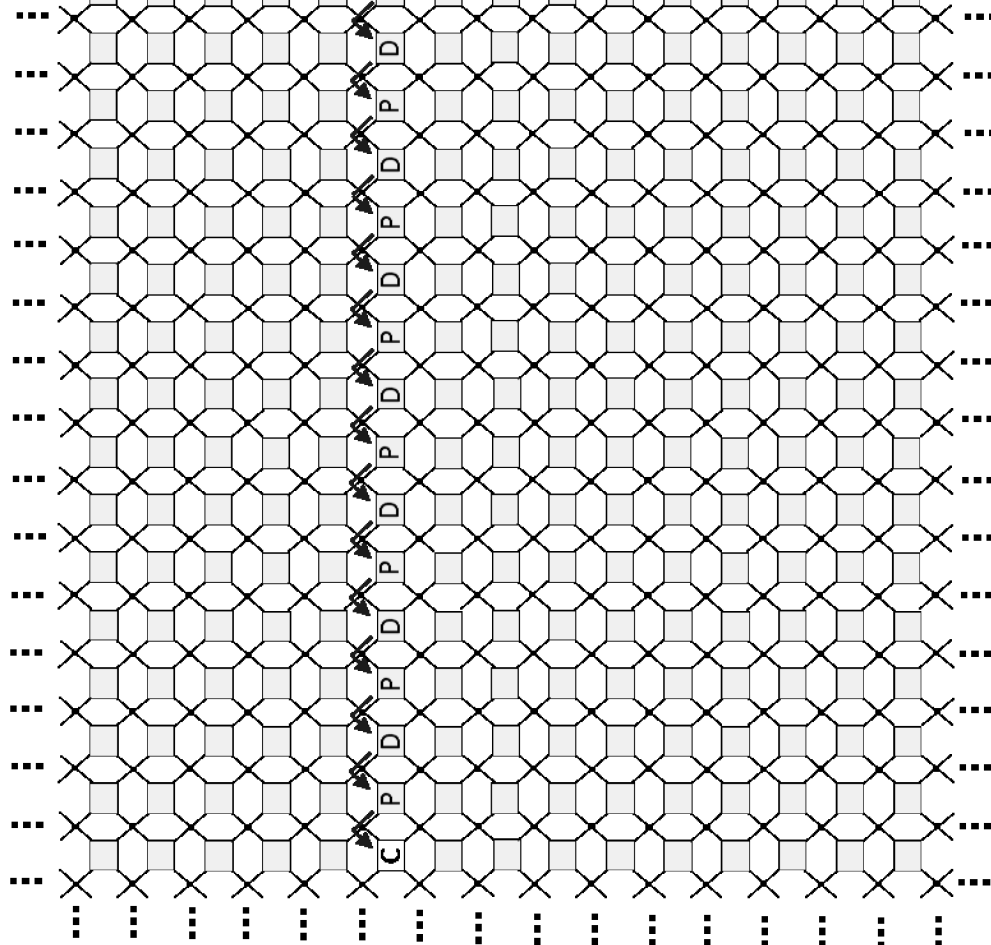


Figure 4.3: Data access between a data memory and a cell in the multiple-valued reconfigurable VLSI using only the X-net network

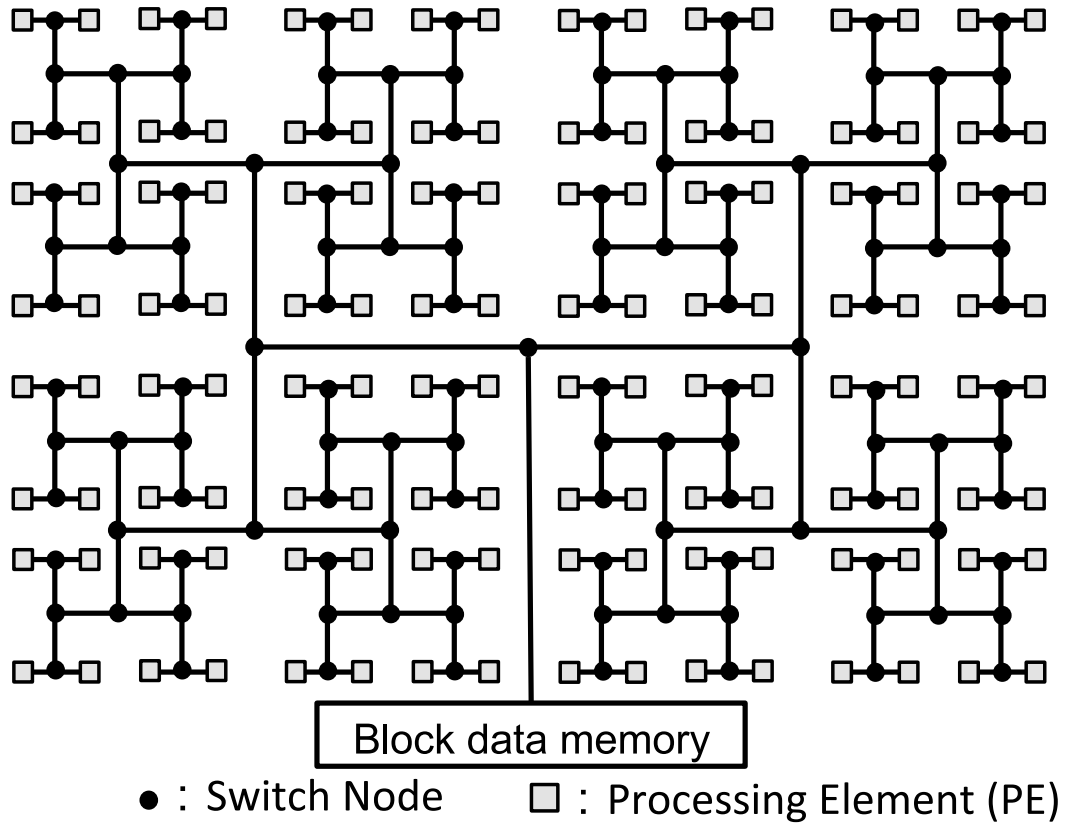


Figure 4.4: Conventional architecture using the dynamic tree network

4.3 Design of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network

The dynamic tree network as one kind of the multistage networks is employed for high-performance long-distance data transfer in the MVFG-RVLSI. As shown in Fig. 4.4, all processing elements (PEs) are config-

ured as “leaves” of the tree network. Data can be transferred between the PEs through one or more switch nodes. Each switch node has three I/O ports; one connected to a parent switch node and the other two connected to child nodes (or PEs, at the bottom level)[22]. One port of the switch node at the top level is connected to a block data memory to access data between the block data memory and the PE array. The tree network can be utilized to realize both the long-distance inter-PE data transfer and the data access between the block data memory and the PE array, which leads to high utilization of the dynamic tree network.

However, very long interconnection and many switch nodes are required for the data access between the block data memory and the PE array, which results in low speed and large power consumption. Moreover, only one data can be accessed and other many data cannot be accessed in parallel, which causes low utilization of the PEs. The data transfer bottleneck can be greatly reduced by employing a logic-in-memory architecture shown in Fig. 4.5, because it can make the interconnection length and the switch node count between a local memory (LM) and the PEs very short and small, respectively[23]. The size of the LM is same as that of the data memory in the previous MVFG-RVLSI shown in Fig. 4.2. In the LME, the LM and eight PEs communicate with each other by a three-level sub tree. The switch node at the third level has four I/O

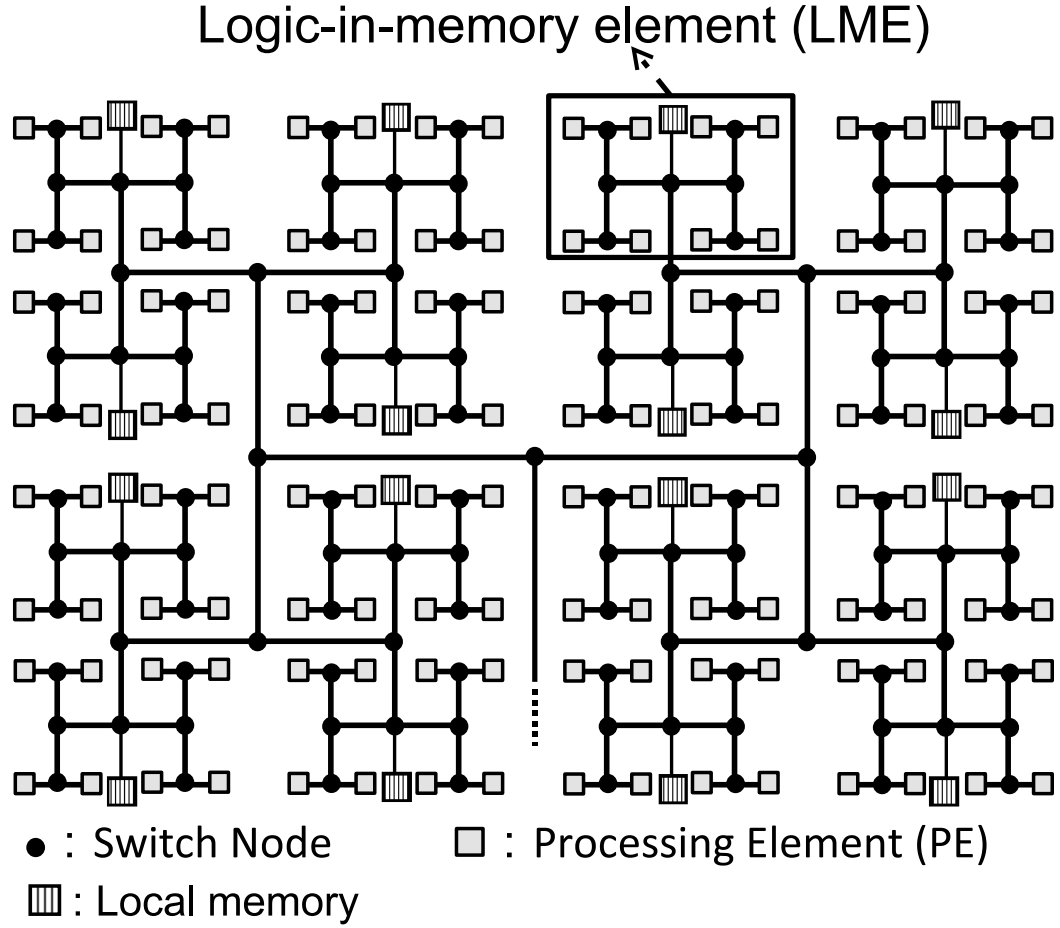


Figure 4.5: Logic-in-memory architecture using the dynamic tree network

ports; one connected to the LM and the rest connected to other switch nodes.

The logic-in-memory architecture using the dynamic tree network is applied to the MVFG-RVLSI using only the X-net network, where the ultra-fine grain cell is composed of 10 differential-pair circuits and a CMOS DFF[24]. If the dynamic tree network is connected to each cell, the cost becomes extremely large. Also, in practical applications, most

of the cells require neighborhood data transfer for bit-serial operations. For example, let us consider the SAD widely used as a similarity measure in template matching. The SAD is expressed as

$$SAD = |A1 - B1| + |A2 - B2| + \cdots + |A16 - B16| \quad (4.1)$$

where the control/data flow graph is shown in Fig. 3.13. The SAD is performed by iteration of an absolute difference operation and addition (ADA). Figure 3.16 shows the allocation result of the eight-bit ADA for the MVFG-RVLSI based on the multiple-valued X-net data transfer scheme[12]. Only three cells are necessary to be connected to the dynamic tree network to receive or send data. Therefore, the dynamic tree network should be connected to multiple-cell blocks composed of many cells, but not to each cell.

Figure 4.6 shows the MVFG-RVLSI using the GTLX. The dynamic tree network is utilized to realize bit-parallel global data transfer (eight bits as an example), and the X-net network is utilized to realize bit-serial localized data transfer between the cells for logic operations. Two kinds of the switch nodes are utilized to control the global data transfer. A non-pipelined switch node is composed of three one-bit switches, and a pipelined switch node employed for high throughput and correct data transfer is composed of the eight DFFs and six one-bit switches.

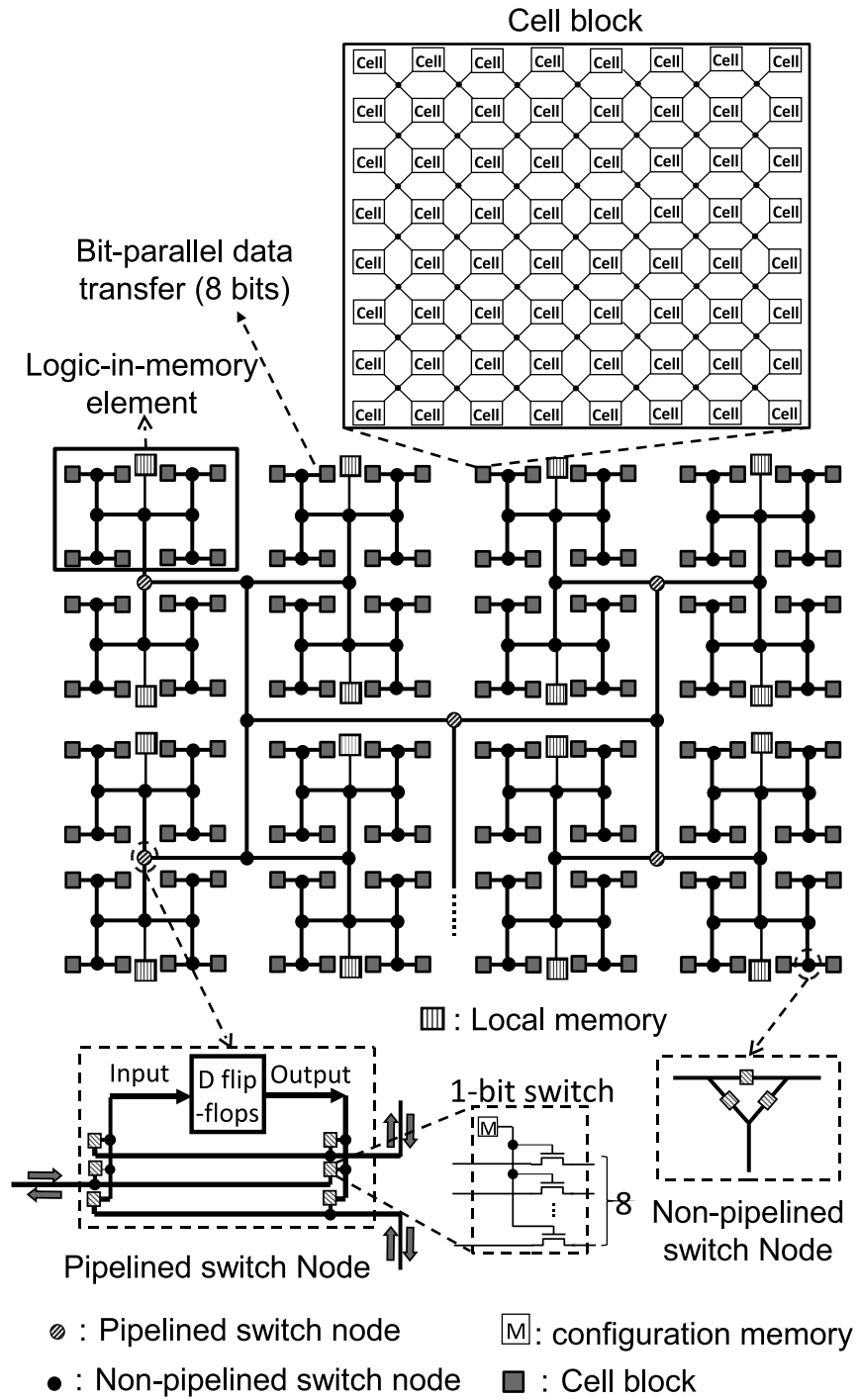


Figure 4.6: Architecture of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network

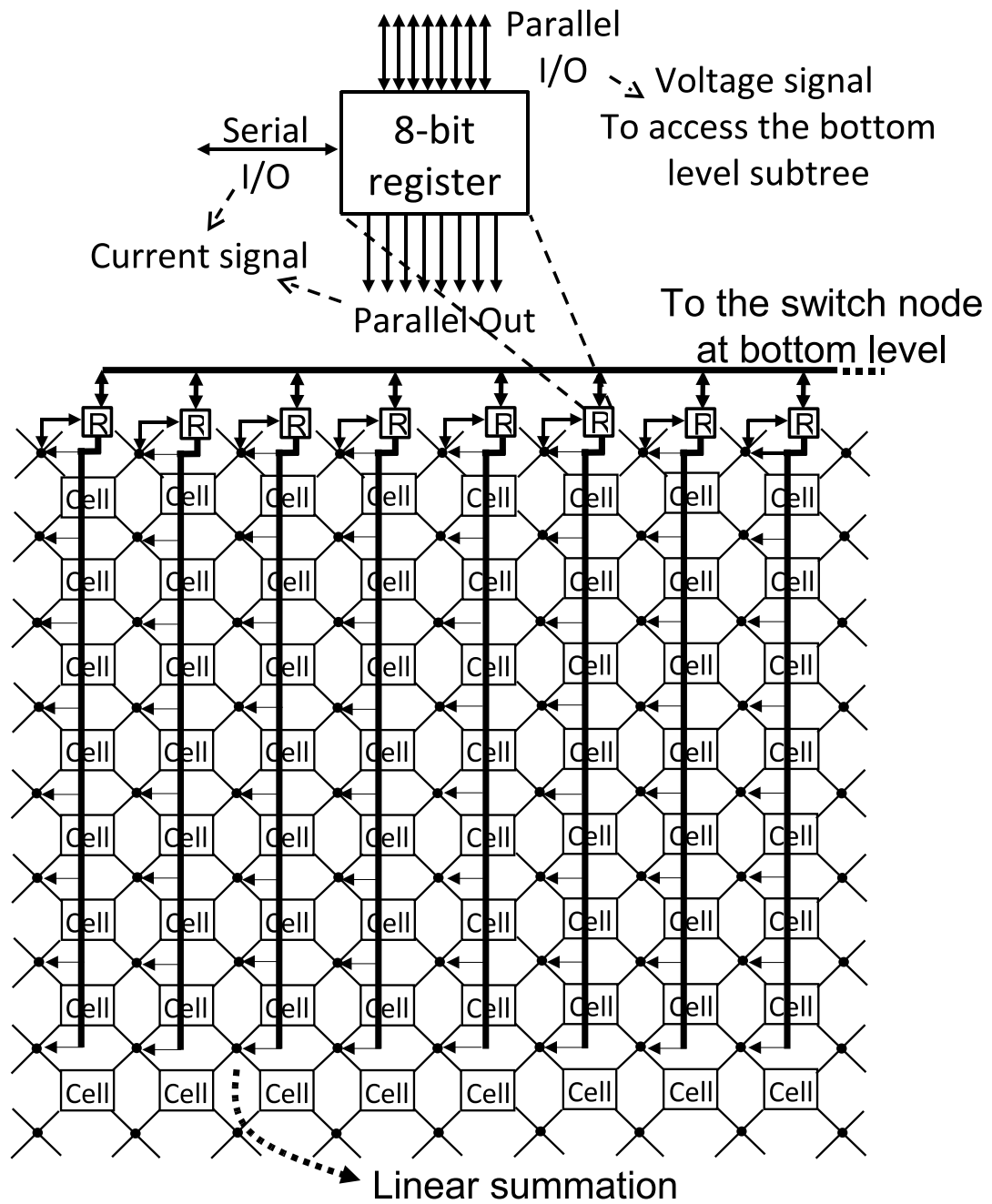


Figure 4.7: Interconnections between the dynamic tree network and the X-net network

Figure 4.7 shows the interconnections between the dynamic tree network and the X-net network by eight eight-bit registers. Each register has a parallel voltage I/O port, a serial current I/O port and a parallel current output port. The parallel voltage I/O port is utilized to access the tree network for the global bit-parallel data transfer. The serial current I/O port is utilized to access a fixed “X” intersection for bit-serial operations. The parallel current output port is utilized to access eight fixed vertical “X” intersections for serial-parallel operations such as a serial-parallel multiplication shown in Fig. 3.9. At each “X” intersection, the current signals from the current port and an adjacent cell can be linearly summed by wiring, which leads to high utilization of the X-net network.

4.4 Evaluation of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network

The evaluation of the MVFG-RVLSI using the GTLX is done based on HSPICE simulation using a 65 nm CMOS design rule. An eight-bit data is transferred from one cell to 100 cells away in the MVFG-RVLSI using only the X-net network, and the MVFG-RVLSI using the GTLX, respectively. Table 4.1 shows the comparison results. In comparison

Table 4.1: Comparison of the long-distance data transfer in the multiple-valued fine-grain reconfigurable VLSIs

	MVFG-RVLSI using only the X-net	MVFG-RVLSI using the global tree local X-net
Supply voltage	1.2 V	1.2 V
Computation time	29 ns	11 ns
Transistor count	32800	3778
Configuration memory count	1900	172
Power consumption at 1 GHz	6100 μ W	498 μ W
Power-delay product	177 pJ	5.5 pJ
Word length= 8 bit, Distance= 100 cells		

with the MVFG-RVLSI using only the X-net network, the computation time, the transistor count, the configuration memory count, the power consumption and the power-delay product of the MVFG-RVLSI using the GTLX are reduced by 62%, 88%, 91%, 92% and 97%, respectively.

Let us consider the FFT operation which is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. The most common FFT algorithm is the Cooley-Tukey FFT algorithm composed of many butterfly operations. Figure 4.8 shows the control/data flow graph of the butterfly operation.

Figure 4.9 shows the allocation of the butterfly operation onto the MVFG-RVLSI using only the X-net network. The inputs B_r , W_r , B_i

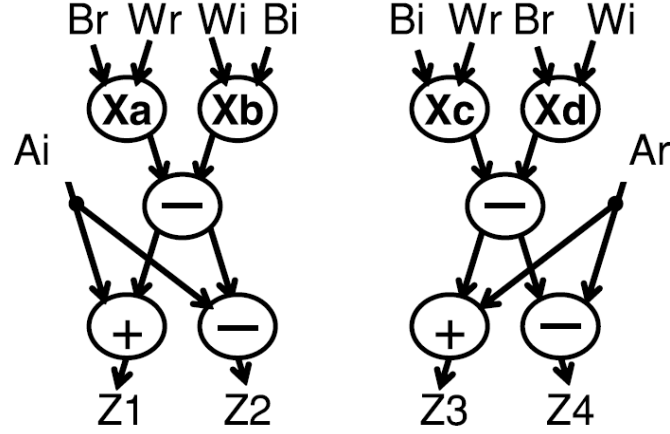


Figure 4.8: Control/data flow graph of the butterfly operation

and W_i are transferred from the fixed eight-bit registers to the edge cells, which is the optimal allocation onto the MVFG-RVLSI using only the X-net network. However, 16 cells are used to implement two serial-in parallel-out registers for the serial-parallel multiplication, and 24 cells are used to access the input A_i and the outputs O_{1r} , O_{2r} between the fixed eight-bit registers and the non-edge cells, which causes low speed, large power consumption and low utilization of the cells.

Figure 4.10 shows the allocation of the butterfly operation onto the MVFG-RVLSI using the GTLX. All of the inputs B_r , W_r , B_i , W_i , A_i and the outputs O_{1r} , O_{2r} are transferred between the eight-bit registers provided at the bottom level sub tree and the adjacent cells. Moreover, the parallel current signals from the eight-bit registers and the current

signals from the adjacent cells are linearly summed by wiring, which leads to high utilization of the X-net network.

Table 4.2 shows the comparison results. The computation time, the transistor count, the configuration memory count, the power consumption and the power-delay product of the MVFG-RVLSI using the GTLX are reduced by 25%, 56%, 59%, 36% and 52%, respectively, in comparison with those of the MVFG-RVLSI using only the X-net network. The performance is greatly improved in the MVFG-RVLSI using the GTLX, even though in comparison with the optimal allocation in the MVFG-RVLSI using only the X-net network.

Table 4.2: Comparison of the butterfly operation modules in the multiple-valued fine-grain reconfigurable VLSIs

	MVFG-RVLSI using only the X-net	MVFG-RVLSI using the global tree local X-net
Supply voltage	1.2 V	1.2 V
Computation time	28 ns	21 ns
Transistor count	39366	17320
Configuration memory count	2194	895
Power consumption at 1 GHz	9435 μ W	6024 μ W
Power-delay product	264 pJ	127 pJ

Word length= 8 bit

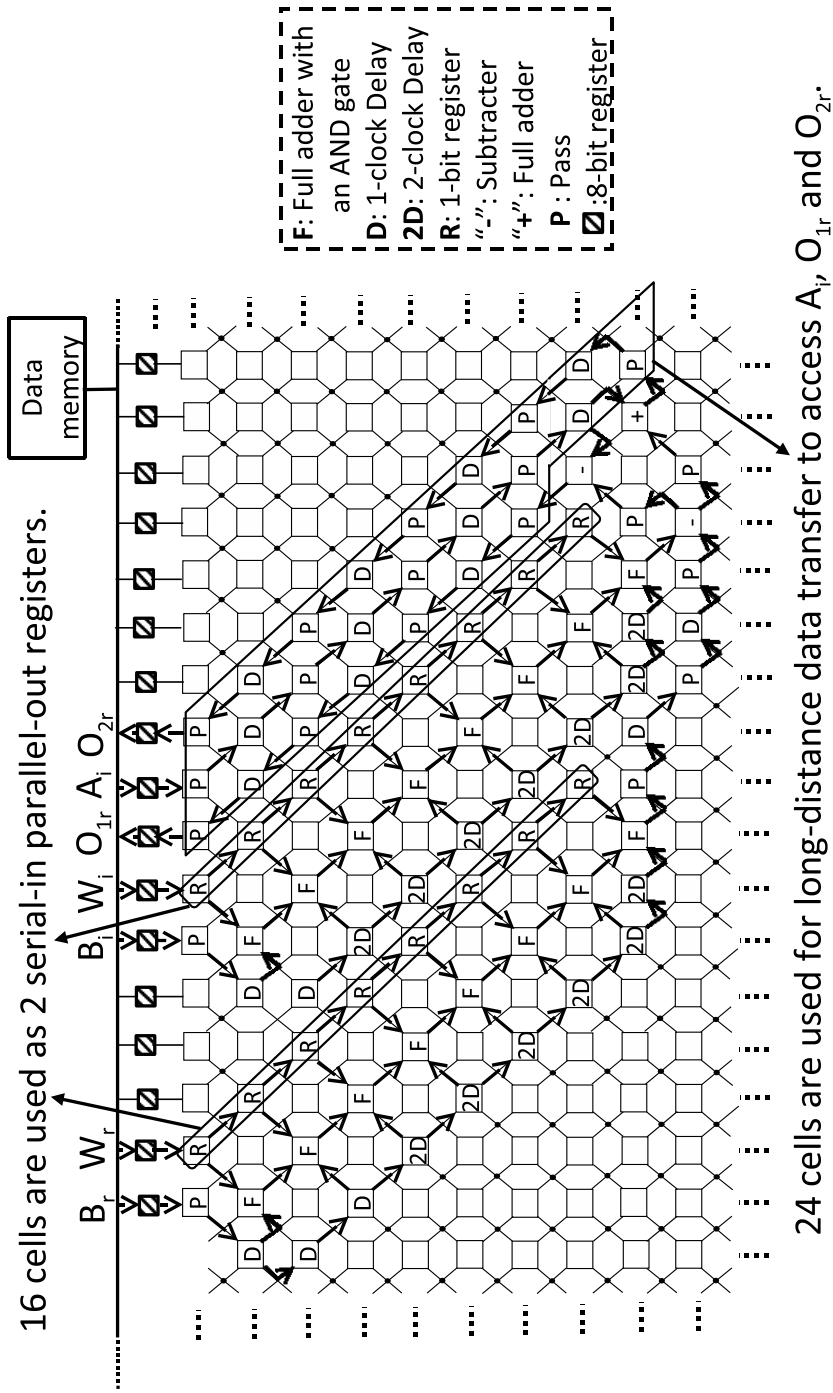


Figure 4.9: Allocation of the butterfly operation onto the multiple-valued fine-grain reconfigurable VLSI using only the X-net network

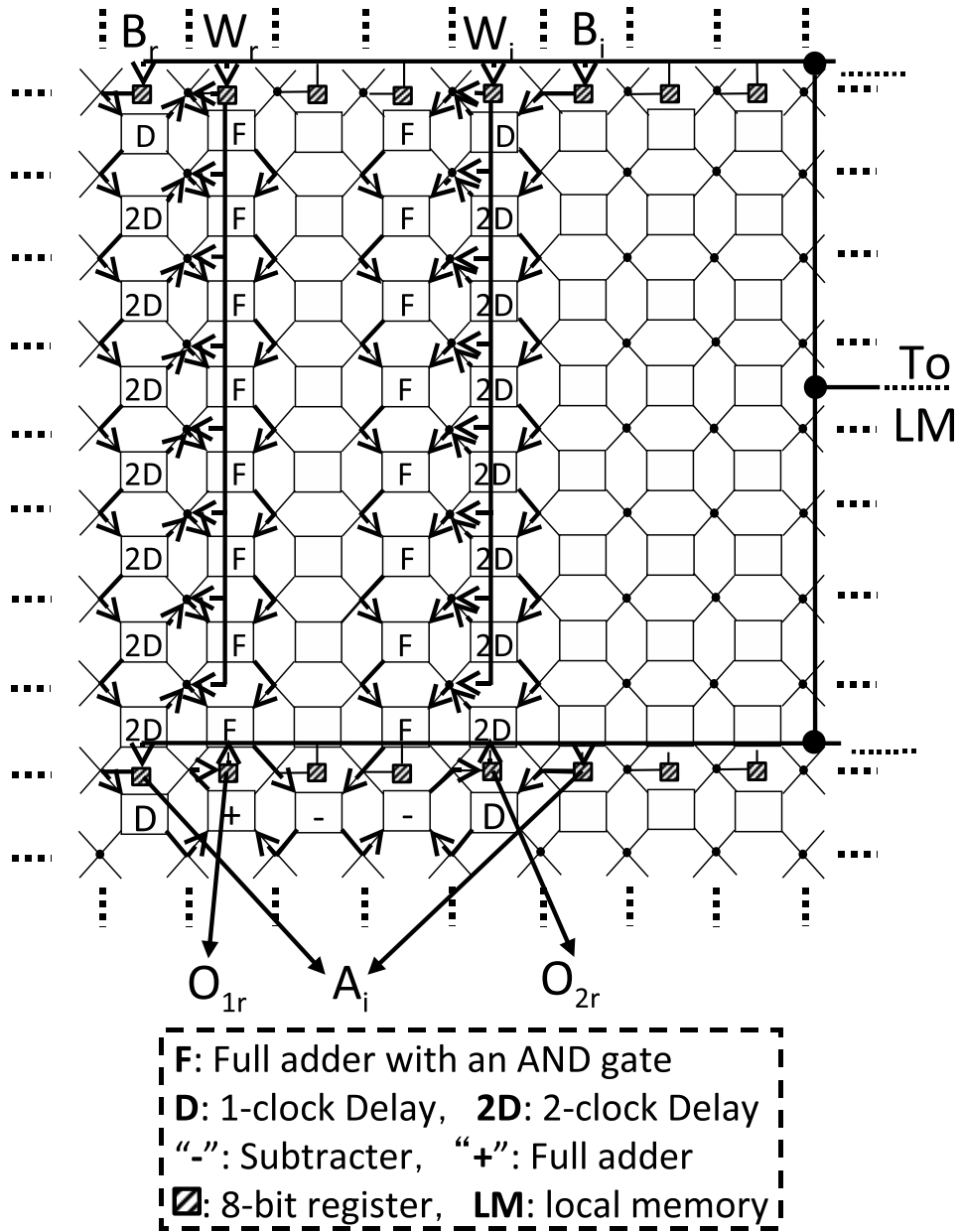


Figure 4.10: Allocation of the butterfly operation onto the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network

4.5 Conclusion

Chapter 4 presents a global tree local X-net network for high-performance data transfer in the multiple-valued fine-grain reconfigurable VLSI. A pipelined dynamic tree network is employed for high-throughput bit-parallel global data transfer, and an X-net network is employed for simple bit-serial localized data transfer for logic operations.

The logic-in-memory architecture is utilized to solve the bottleneck problem between a data memory at the top level sub tree and each cell at the bottom level sub tree. A register with a serial current I/O port, a parallel voltage I/O port, and a parallel current output port is introduced to realize flexible interconnections between the dynamic tree network and the X-net network. Moreover, linear summation of the current signals from the register and an adjacent cell can be realized at each “X” intersection, which leads to high utilization of the X-net network.

It is demonstrated that the computation time, the transistor count, the configuration memory count, the power consumption and the power-delay product of the multiple-valued fine-grain reconfigurable VLSI using the global tree local X-net network are reduced by 25%, 56%, 59%, 36% and 52%, respectively, in comparison with those of the multiple-valued fine-grain reconfigurable VLSI using only the X-net network.

Chapter 5

Conclusion

In this research, three circuit-level and two architecture-level techniques are proposed for high-performance multiple-valued fine-grain reconfigurable VLSI. In Chapter 2, a binary-controlled current-steering technique using a three-level differential-pair circuit is proposed for high-speed and low-power arbitrary two-variable binary operation. A current-source sharing technique is proposed to realize high utilization of current sources for low power consumption. A dual-supply voltage technique is proposed for low-power multiple-valued source-coupled logic circuits. In Chapter 3, an X-net network is employed for realizing high utilization of one-bit switches for area-efficient switch blocks. Moreover, a multiple-valued data transfer scheme is proposed to realize high utilization of the “X” intersections for simple interconnections. In Chapter 4, a global dynamic tree network for long-distance data transfer is employed

for realizing high utilization of cells for high speed and low power consumption.

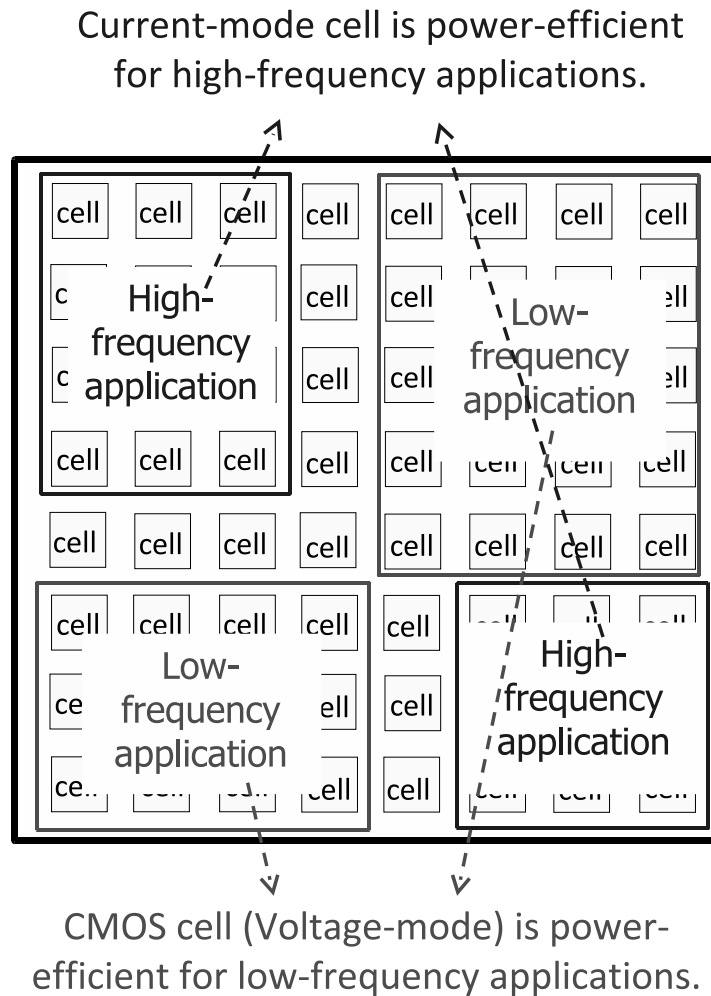


Figure 5.1: Voltage-mode/current-mode hybrid fine-grain reconfigurable VLSI architecture

For the future work, it is important to develop a voltage-mode/current-mode hybrid fine-grain reconfigurable VLSI to minimize power con-

sumption [25]. Simultaneously, high-frequency and low-frequency operations can be realized by many hybrid cells on one reconfigurable VLSI chip as shown in Fig.5.1. In each hybrid cell, the voltage and current mode can be selected for low-power operations at low and high frequency, respectively, according to speed requirement.

Bibliography

- [1] C. Bobda, *Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications*. Springer, 2007.
- [2] I. Kuon and J. Rose, “Measuring the Gap Between FPGAs and ASICs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, Feb. 2007.
- [3] H. M. Munirul and M. Kameyama, “Architecture of a Fine-Grain Field-Programmable VLSI Based on Multiple-Valued Source-Coupled Logic,” *IEICE Transactions on Electronics*, vol. E87-C, no. 11, pp. 1869–1875, 2004.
- [4] A. Ishikawa, N. Okada, and M. Kameyama, “Low-Power Multiple-Valued Reconfigurable VLSI Based on Superposition of Bit-Serial Data and Current-Source Control Signals,” in *Proceedings of the 40th IEEE International Symposium on Multiple-Valued Logic*, Barcelona, Spain, May 2010, pp. 179–184.

- [5] M. Hariyama, W. Chong, and M. Kameyama, “Field-Programmable VLSI Based on a Bit-Serial Fine-Grain Architecture,” *IEICE Transactions on Electronics*, vol. E87-C, no. 11, pp. 1897 – 1902, 2004.
- [6] V. Stamatis and D. Soudris, *Fine- and Coarse-Grain Reconfigurable Computing*. Springer, 2007.
- [7] N. Okada and M. Kameyama, “Fine-Grain Multiple-Valued Reconfigurable VLSI Using Series-Gating Differential-Pair Circuits and Its Evaluation,” *IEICE Transactions on Electronics*, vol. E91-C, no. 9, pp. 1437–1443, Nov. 2008.
- [8] X. Bai and M. Kameyama, “Current-Source-Sharing Differential-Pair Circuits for a Low-Power Fine-Grain Reconfigurable VLSI Architecture,” in *Proceedings of the 42nd IEEE International Symposium on Multiple-Valued Logic*, Victoria, Canada, May 2012, pp. 208–213.
- [9] Musicer.J.M and Rabaey.J, “MOS current mode logic for low power, low noise CORDIC computation in mixed-signal environments ,” in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2000, pp. 102–107.

- [10] Hassan.H, Anis.M, and Elmasry.M, “An Efficient Delay Model for MOS Current-Mode Logic Automated Design and Optimization,” *IEEE transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 2041–2052, Aug. 2010.
- [11] X. Wang and L. Bandi, “X-Network: An Area-Efficient and High-Performance On-Chip Wormhole-Switching Network ,” in *Proceedings of 12th IEEE International Conference on High Performance Computing and Communications*, 2000, pp. 362–368.
- [12] X. Bai and M. Kameyama, “ A Bit-Serial Reconfigurable VLSI Based on a Multiple-Valued X-Net Data Transfer Scheme,” *IEICE Transactions on Information and Systems*, vol. E96-D, no. 7, pp. 1449 – 1456, 2013.
- [13] ———, “An Area-Efficient Multiple-Valued Reconfigurable VLSI Architecture Using an X-Net,” in *Proceedings of the 43rd IEEE International Symposium on Multiple-Valued Logic*, May 2013, pp. 272–277.
- [14] N. Ohsawa, O. Sakamoto, M. Hariyama, and M. Kameyama, “ Program-Counter-Less Bit-Serial Field-Programmable VLSI Processor with Mesh-Connected Cellular Array Structure ,” in *Pro-*

- ceedings of IEEE Computer Society Annual Symposium on VLSI 2004*, Feb. 2004, pp. 258–259.
- [15] J. S. Yuan and L. Yang, “Teaching digital noise and noise margin issues in engineering education,” *IEEE Transactions on Education*, vol. 48, no. 1, pp. 162–168, Feb. 2005.
- [16] S. Bruma, “Impact of on-chip process variations on MCML performance,” in *Proceedings of IEEE International System-on-Chip (SoC) Conference.*, Sep. 2003, pp. 135–140.
- [17] M. Gokhale and P. S. Graham, *Reconfigurable Computing: accelerating computation with field-programmable gate array*. Springer, 2005.
- [18] Hassan.H, Anis.M, and Elmasry.M, “MOS Current Mode Circuits: Analysis, Design, and Variability,” *IEEE transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 8, pp. 885–898, Aug. 2005.
- [19] A. U. Diril, Y. S. Dhillon, A. Chatterjee, and A. D. Singh, “Level-Shifter Free Design of Low Power Dual Supply Voltage CMOS Circuits Using Dual Threshold Voltages,” *IEEE Transactions on*

Very Large Scale Integration (VLSI) Systems, vol. 13, no. 9, pp. 1103 – 1107, 2005.

- [20] J. Nickolls, “The design of the maspar MP-1: a cost effective massively parallel computer ,” in *Proceedings of 35th IEEE Computer Society International Conference*, 1990, pp. 102–107.
- [21] J. N. Kalamatianos and E. Manolakos, “Parallel computation of higher order moments on the maspar-1 machine,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 1832–1835.
- [22] T. L. Casavant, P. Tvrđik, and F. Plasil, *Parallel Computers: Theory and Practice*. IEEE Press, 1995.
- [23] H. Kimura, T. Hanyu, M. Kameyama, Y. Fujimori, T. Nakamura, and H. Takasu, “Complementary Ferroelectric-Capacitor Logic for Low-Power Logic-in-Memory VLSI,” *IEEE Journal of Solid-State Circuits*, vol. SC-39, no. 6, pp. 919–926, 2004.
- [24] X. Bai and M. Kameyama, “A Multiple-Valued Reconfigurable VLSI Architecture Using Binary-Controlled Differential-Pair Circuits,” *IEICE Transactions on Electronics*, vol. E96-C, no. 8, pp. 1083 – 1093, 2013.

- [25] —, “Design and Evaluation of a Voltage-Mode/Current-Mode Hybrid Logic Circuit for a Low-Power Fine-Grain Reconfigurable VLSI,” in *Proceedings of the International SoC Design Conference*, Busan, Korea, 2013, pp. 384–387.

Acknowledgment

This dissertation is the summary of my doctoral research work in the Intelligent Integrated Systems Laboratory, Graduate School of Information Sciences, Tohoku University. The work has been ambitious and highly challenging. However, without the help, support and encouragement mentioned below, I would have never been able to complete this work.

First I would like to express my sincere appreciation to Professor Mitchitaka Kameyama, Graduate School of Information Sciences for his inspiring guidance throughout this research. Without his continuous encouragement and wise comments, this effort would not have been possible.

I would like to thank Professor Koji Nakajima and Professor Takahiro Hanyu, Research Institute of Electrical Communication for their impressive comments and suggestions.

I would like to thank Associate Professor Masanori Hariyama, Graduate School of Information Sciences for his impressive comments and encouragement throughout the whole research.

I would like to thank Assistant Professor Lukac Martin and Project Assistant Professor Hasitha Muthumala Waidyasooriya, Graduate School of Information Sciences for his impressive comments and encouragement.

I would like to thank Technical Official Akio Sasaki and all the members of the Intelligent Integrated Systems Laboratory for providing an excellent and inspiring working atmosphere. The large experience and knowledge gathered here, which all serves as a stable basis for further scientific research.

Finally, I want to thank my parents, without whom I would never have been able to achieve my aim. I also want to thank my wife for her support and understanding.

Xu Bai

January, 2014.