

# A study on the design of D flip-flop with set and reset using GDI scheme

Seungmin Jung\*

Division of Software Convergence,  
Hanshin University, 137  
Hanshinda-gil, Osan-si,  
South Korea

\*E-mail: jasmin@hs.ac.kr

Received for publication  
May 06, 2025.

## Abstract

In this study, we propose a new D flip-flop (DFF) logic gate that includes a set/reset function using gate diffusion input (GDI) technology. Since this cell can be implemented with about 33% less MOS transistors than a conventional CMOS set/reset DFF, it is expected to significantly reduce the chip area occupied by sequential circuits. The cell characteristics were extracted using HSPICE, and a library for Synopsys DC, a synthesis tool, was developed. To verify the applicability of the RTL design flow, a 64-bit counter was designed, logic circuit synthesis was performed, and placement and routing (P&R) layout was performed using openROAD tool. The designed 64-bit counter showed a 22% increase in area in the sequential circuit.

## Keywords

GDI circuit, set/reset D flip-flop, layout area, standard cell library, RTL Design, VLSI

## 1. Introduction

Sequential circuits typically account for approximately 20%–40% of the total circuit elements in a general-purpose digital integrated circuit, although the exact percentage may vary depending on the intended function of the chip and specific design requirements [1, 6–8]. The performance indicators of integrated circuits are defined as PPA, which refers to power consumption, speed, and chip area. Research has been conducted to improve these three performance metrics of semiconductor chips [2–5]. The set and reset functions of a D flip-flop (DFF) are very important in sequential circuits. These functions support synchronization and timing coordination between clock domains. They also contribute to efficient power management by facilitating proper power-on/off sequences and selective activation, thereby reducing power consumption. The DFF cell circuits generally use the

tri-state gate (TSG) method and the pass-transistor gate (PTG) method in the standard CMOS library [1]. The TSG and PTG operate as switches according to the clock signal. Therefore, clock and clock bar signals are required. In this article, we define a DFF cell with set/reset signals as DFFSR. Compared with DFF cells, the DFFSR cell requires additional NAND and OR gates to implement the reset/set function. While a basic DFF uses minimal NAND gates for its operation, adding reset/set functionality makes the circuit more complex, leading to a larger cell area. Therefore, DFFSR occupies more space than a standard DFF. For example, when the PTG circuit structure is applied in the 180 nm standard CMOS process, the area of the DFFSR is known to increase by 1.53 times compared with that of the DFF. In other words, it can be seen that the layout area increases by 53% compared with that of the conventional DFF. This has a significant impact on the increase in the sequential circuit area of the chip and places

restrictions on the logic synthesis of the DFFSR circuit in the RTL design flow. Previous studies have shown results for reducing the area of DFF, but the research on DFFSR is relatively limited [6–8]. One of the studies for reducing the area of DFF cells is the gate diffusion input (GDI) technique [1, 2, 9–19]. In this paper, we propose an efficient DFFSR with a small area by applying the GDI technique.

The paper is organized as follows. Section II introduces the characteristics of GDI. Section III proposes a new structure of DFFSR circuit applying the GDI technique, applies it to a sample RTL circuit to synthesize the logic circuit, and evaluates the performance based on the results reported by Synopsys Design Compiler (DC).

## II. Characteristics of GDI Technique

The basic cell of GDI is shown in Figure 1. GDI uses the gate and source terminals of the CMOS inverter as input signals [9–19]. As shown in Table 1, various logic circuit operations can be performed using the same circuit, depending on the input conditions. Figure 2 shows the GDI circuit implementation of AND, OR, and MUX, as presented in Table 1. It can be seen that similar operations can be implemented with a smaller number of MOS transistors compared with conventional CMOS techniques. Therefore, a small chip area and, in some cases, low power and high-speed operations are possible.

MOS transistors have threshold voltages, so NMOS has problems with VDD transfer, and PMOS has

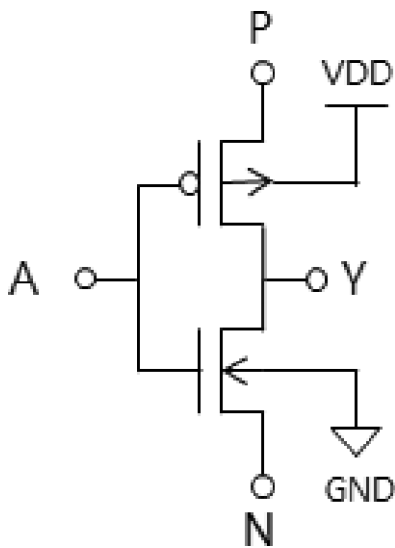


Figure 1: Basic GDI cell and its function. GDI, gate diffusion input.

Table 1: Functions of GDI technique.

| G | Input |     | Output          | Function |
|---|-------|-----|-----------------|----------|
|   | P     | N   |                 |          |
| A | B     | '1' | $A + B$         | OR       |
| A | '0'   | B   | $A \cdot B$     | AND      |
| A | B     | C   | $\bar{A}B + AC$ | MUX      |
| A | '1'   | '0' | $\bar{A}$       | NOT      |

GDI, gate diffusion input.

problems with 0 V transfer. Therefore, the output voltage of the GDI cell can be 0,  $V_{TN}$ ,  $V_{DD}-V_{TP}$ , or  $V_{DD}$ . In other words, an intermediate voltage can appear [1, 2]. Here,  $V_{TN}$  represents the threshold voltage of the NMOS, and  $V_{TP}$  represents the threshold voltage of the PMOS. Output voltage distortion occurs due to the threshold voltage drop of the MOS transistor of the GDI cell. For this reason, the cascade GDI cell structure has the disadvantage that functional errors can occur [2]. The cascade design is defined as a series-connected circuit in which the output of one GDI cell is connected to the input of the next GDI cell. The distortion of the output voltage is a major drawback of GDI technology, which limits the wide application of the GDI technique in chip development. Ultimately, designing a chip composed solely of GDI cells is not feasible. The most effective way to overcome these problems is to ensure that GDI cells are not placed consecutively during the logic circuit synthesis in the RTL design, but are instead mixed with CMOS cells.

Figure 3 shows a hybrid GDI structure that solves the operational errors and enables GDI low-power, small-area chip design using GDI technology [2]. The cascade circuit is a mixture of CMOS and GDI cells in the hybrid GDI technology, as shown in Figure 3. Since the designed chip mixes CMOS and GDI cells, the area reduction may be reduced compared with a chip designed only with GDI cells. In addition, because the placement and routing (P&R) process must apply an EDA environment that includes a logic synthesizer such as openROAD, cadence INNOVUS, and Synopsys Fusion compiler, a new library with characteristics for GDI cells must be provided in addition to the CMOS library.

## III. Design of DFFSR Based on GDI Technique

In this paper, a set/reset DFF cell designed only with traditional CMOS gates is defined as DFFSR\_C, and a set/reset DFF cell with GDI cells

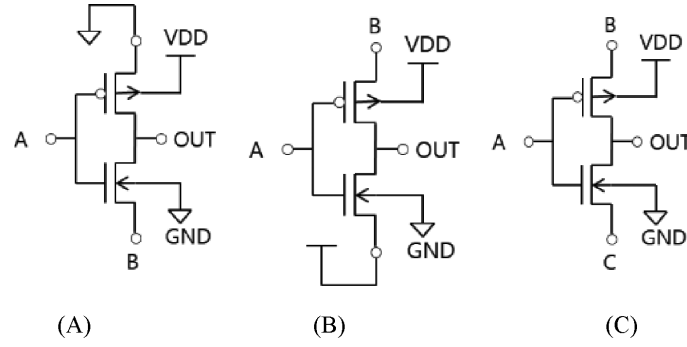


Figure 2: Functional block diagram of fingerprint sensor chip. (A) AND (B) OR (C) MUX.

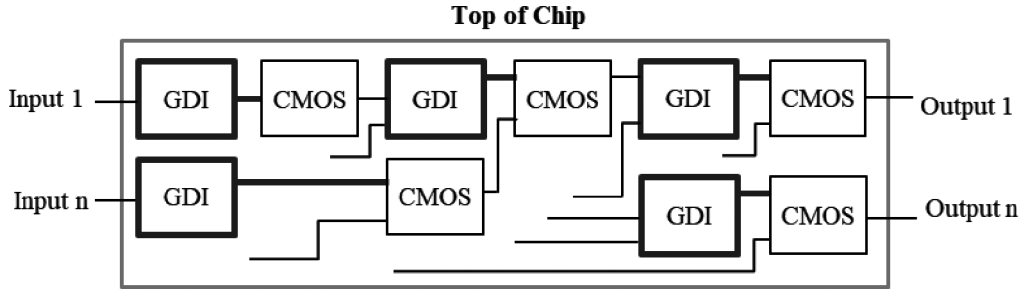


Figure 3: Hybrid GDI technique. GDI, gate diffusion input.

is defined as DFFSR\_G. Figure 4 shows typical PTG CMOS DFF cell with master/slave stage and set/reset signal. A typical DFFSR\_C uses a minimum of 42 MOS transistors in total. The master stage uses 16 MOS transistors, and the slave stage uses 20 MOS transistors. Table 2 shows its operational truth table. RN represents the negative reset signal, SN represents the negative set signal, and D and CK signals represent data and clock, respectively. DFFSR\_C consists of four PTGs and two OR and NAND cells each in addition to four inverter cells. The RN and SN signals are asynchronous input signals that can occur independently of CK.

Figure 5 shows a novel PTG D flip-flop cell (DFFSR\_G) with master/slave stage and set/reset signal based on the GDI technique. The proposed DFFSR\_G circuit uses two GDI MUX circuits and two GDI OR circuits, as shown in the blue dotted boxes. The remaining NOT series cells use CMOS cells. The PTG circuit applied in DFFSR\_C is replaced with GDI MUX circuits, and the CMOS OR cell is replaced with GDI OR circuits. Conventional DFFSR\_C requires four CLK and CLKB signals each, but DFFSR\_G requires only two CLK input signals; CLKB is not required in DFFSR\_G. Therefore, an added CLKB circuit is not

required. In this case, the number of transistors can be dramatically reduced overall. DFFSR\_G can overcome the drawback by avoiding the cascaded GDI circuit structure mentioned in Section II. This circuit demonstrates a hybrid GDI structure.

The proposed DFFSR\_G uses just 28 MOS transistors. The GDI OR gate reduces the area of the CMOS OR and PTG cells while simultaneously avoiding the cascaded GDI structure. Because GDI gate outputs may not fully swing, static current may occur in subsequent CMOS stages. The static current may occur in the inverter and NAND cell connected to the next stage of the GDI cell, as shown in the DFFSR\_G circuit of Figure 5.

There are various ways to reduce the static current, but in this study, the width and length of the MOS transistors of the inverter and NAND cells connected to the GDI cell were optimized while maintaining the circuit structure. In this case, the current is reduced, however the driving strength of the CMOS cell is reduced, which may increase the delay time and affect the operating speed. When applied to DFFSR\_G, it is necessary to adjust the power consumption and the operating speed.

The characteristics of the DFFSR\_G cell are extracted and compared with the DFFSR\_C.

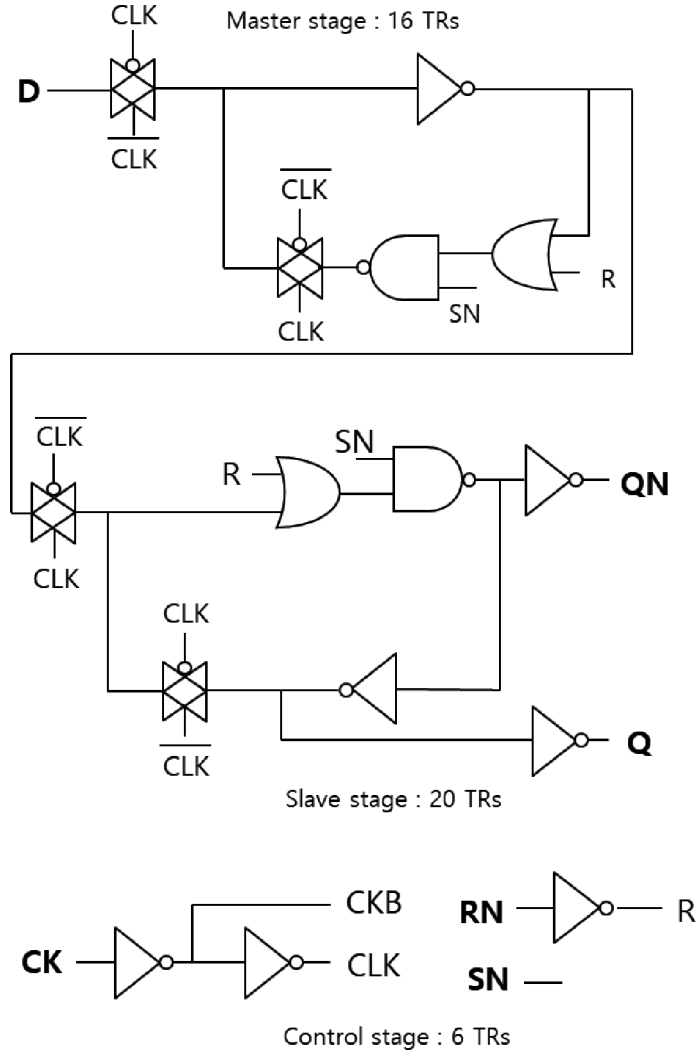


Figure 4: Conventional CMOS set/reset D flip-flop (DFFSR\_C).

The simulations were performed by HSPICE and Spectre with 180nm standard CMOS process parameters at 1.8 V operating voltage. The MOS length for PMOS is 180 nm, the width is 900 nm, and for NMOS it is 600 nm, which is the same as in the rest of the CMOS circuit, except for the GDI cell, which is adjusted for optimization. The operating frequency is 100 MHz for power calculation. Figure 6 shows the operation simulation results of DFFSR\_G. When the SN signal is logic 0 and the RN signal is logic 1, the output Q performs a “set” operation to 1, and when SN and RN have opposite conditions, Q performs a “reset” operation to 0. Spectre simulation results show that the proposed DFFSR\_G has the same functionality as DFFSR\_C. Table 3 summarizes of the performance comparison. This result means that DFFSR\_G has an area

reduction effect of about 33% compared with DFFSR\_C, which uses 42 MOS, and is expected to contribute to reducing the sequential circuit area. However, it is disadvantageous compared with DFFSR\_C in terms of operation speed. The proposed DFFSR\_G cell can be applied to the design of small chip area, low-power chips suitable for mobile devices. The direction of future research is to design to achieve improvements at or above the level of CMOS through analysis of operating speed.

#### IV. RTL Implementation of DFFSR\_G

As a sample design to verify the applicability of the RTL design flow and the performance of the DFFSR\_G cell, a 64-bit counter with set/reset operations was designed, and layout was performed

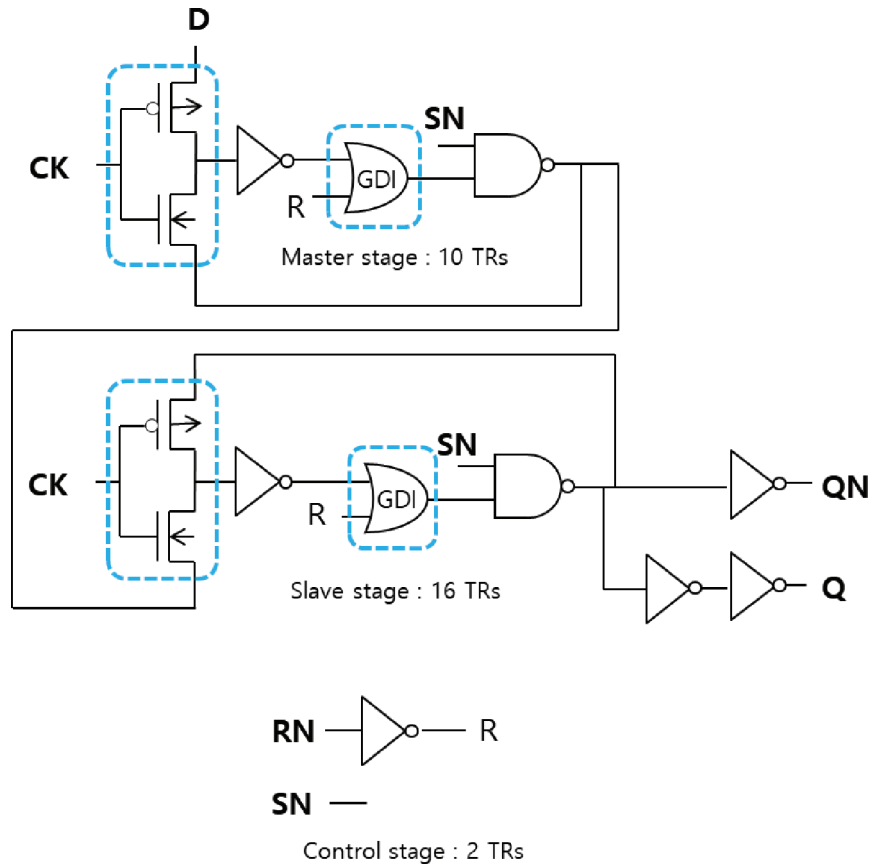


Figure 5: Proposed GDI set/reset D flip-flop (DFFSR\_G). CK, DFFSR\_G, GDI, gate diffusion input.

Table 2: Truth table of DFFSR operation.

| RN | SN | D | CK | Q[n+1] | QN[n+1] |
|----|----|---|----|--------|---------|
| 0  | 0  | X | X  | 0      | 1       |
| 1  | 0  | X | X  | 1      | 0       |
| 0  | 0  | X | X  | 1      | 0       |
| 1  | 1  | 0 | ↑  | 0      | 1       |
| 1  | 1  | 1 | ↑  | 1      | 0       |
| 1  | 1  | X | ↓  | Q[n]   | QN[n]   |

through P&R. The 64-bit counter was developed using the Verilog-HDL language, and the logic circuit was synthesized using Synopsys DC. The open EDA tool “openROAD” was used for P&R. Figure 7 shows the RTL code of the 64-bit counter.

The timing and power values of the remaining GDI cells, including DFFSR\_G, were extracted using HSPICE. The 180nm CMOS process parameters were used for logic circuit synthesis and P&R, and

implemented using the Design Kit (D/K). The characteristic values of DFFSR\_G cells were added to the Synopsys DC library. As shown in Figure 8, the DFFSR\_G library file was generated by modifying the values of each timing field, area, and power field based on the characterization results, using the conventional library file. The library file of the text format was converted to a binary “db” file using the “write\_lib” command of DC and applied.

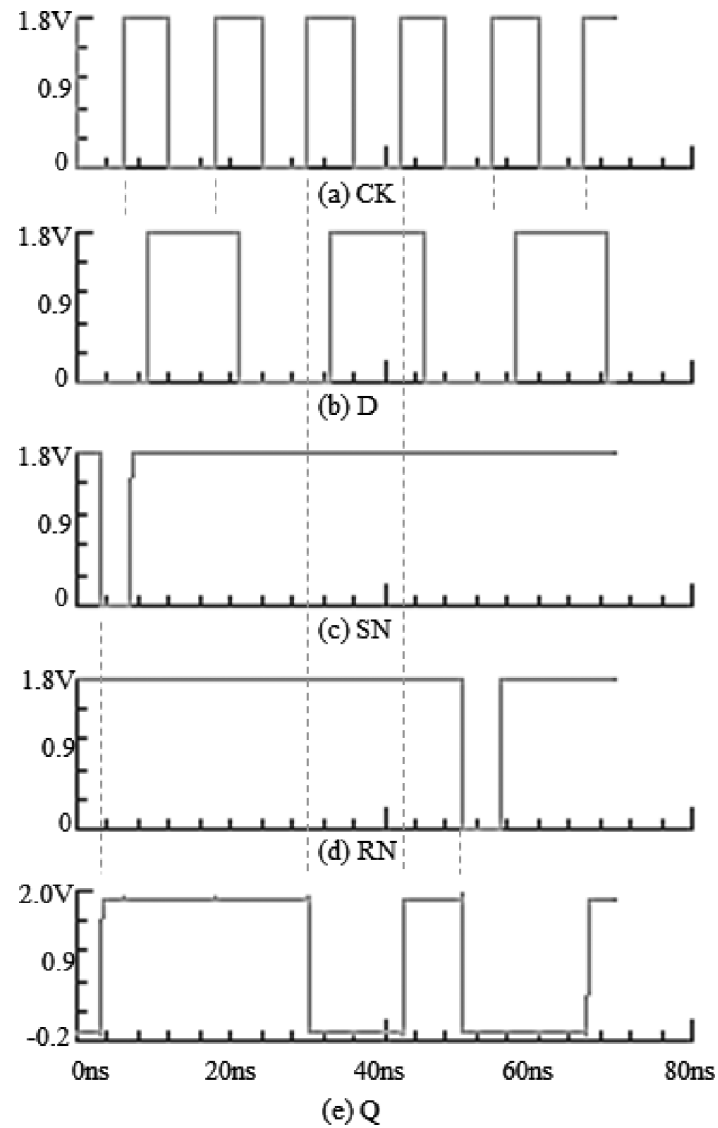


Figure 6: Functional simulation result of DFFSR\_G.

Table 3: Summary of DFFSR performance comparison.

|             | Number of<br>MOS TR. | clk_q_tplh<br>(sec) | clk_q_tphl<br>(sec) | average (sec) | power_avg<br>(uW/MHz) |
|-------------|----------------------|---------------------|---------------------|---------------|-----------------------|
| DFFSR_C     | 42                   | 3.60E-10            | 3.17E-10            | 3.39E-10      | 0.211                 |
| DFFSR_G     | 28                   | 5.97E-10            | 6.04E-10            | 6.00E-10      | 0.205                 |
| Improvement | 33%                  |                     |                     | -77%          | 3%                    |

The 64-bit counter layout was placed and routed using the openROAD tool. The process is the “gf180” platform of the 180n process 6-metal. Figure 9 show the design constraint specification

files for openROAD, where the core utilization was applied to 65%, and the operating frequency was 100 MHz with 10 ns period. The SDC file shown in Figure 9 is compatible with Synopsys ICC I, II, or

```

module counter64(
    input wire clk,          // Clock input
    input wire rst,          // Asynchronous reset input
    input wire set,          // Asynchronous set input
    input wire [63:0] set_value, // Value to set the counter to
    output reg [63:0] count // 64-bit counter output
);

always @(posedge clk or posedge rst or posedge set) begin
    if (rst) begin
        count <= 64'd0; // Reset counter to 0
    end else if (set) begin
        count <= set_value; // Set counter to set_value
    end else begin
        count <= count + 64'd1; // Increment counter
    end
end
endmodule

```

Figure 7: Verilog-HDL based RTL code for synthesis and P&R verification using DFFSR\_G. P&R, placement and routing.

```

export DESIGN_NICKNAME = counter64
export DESIGN_NAME = counter64

export PLATFORM = gf180
export VERILOG_FILES
= ./designs/src/${DESIGN_NICKNAME}/counter64.v W

export SDC_FILE
= ./designs/${PLATFORM}/${DESIGN_NICKNAME}/constraint.sdc
export CORE_UTILIZATION = 65
export PLACE_DENSITY_LB_ADDON = 0.1
export export CORE_ASPECT_RATIO = 1

```

Figure 8: Configuration condition for openROAD.

Fusion Compiler. Figure 10 shows the 64-bit counter layout applying the DFFSR\_G, and the layout was performed by the openROAD tool. Table 4 shows the circuit synthesis and P&R results. The number of synthesized logic circuit cells is 196, and the chip size was measured as 8,618.4  $\mu\text{m}^2$ . The P&R performance results showed an area reduction effect of about 22% in the sequential circuit. Since it includes the area according to routing and placement, it seems to have decreased compared with the 33%

cell area reduction. It is expected that the area effect will show up to 33%, depending on the sample chip. In addition, the 64-bit counter is a sample for comparing only the results of the sequential circuit area. It is somewhat limited in verifying the final result as a result of applying a circuit with a relatively small number of cell instances. Therefore, if DFFSR\_G is applied to implement a large-scale integrated circuit, it is expected to be effective in designing a chip with similar power consumption and a small area.

```
current_design counter64
set clk_name clk
set clk_port_name clk
set clk_period 10.0
set clk_io_pct 0.2

set clk_port [get_ports $clk_port_name]
create_clock -name $clk_name -period $clk_period $clk_port
set non_clock_inputs [search -inline -all -not -exact [all_inputs]
$clk_port]
set_max_area 0

set_input_delay [expr $clk_period * $clk_io_pct] -clock $clk_name
$non_clock_inputs
set_output_delay [expr $clk_period * $clk_io_pct] -clock
$clk_name [all_outputs]
set_false_path -from [get_ports {rst_ni}]
```

Figure 9: SDC condition for openROAD.

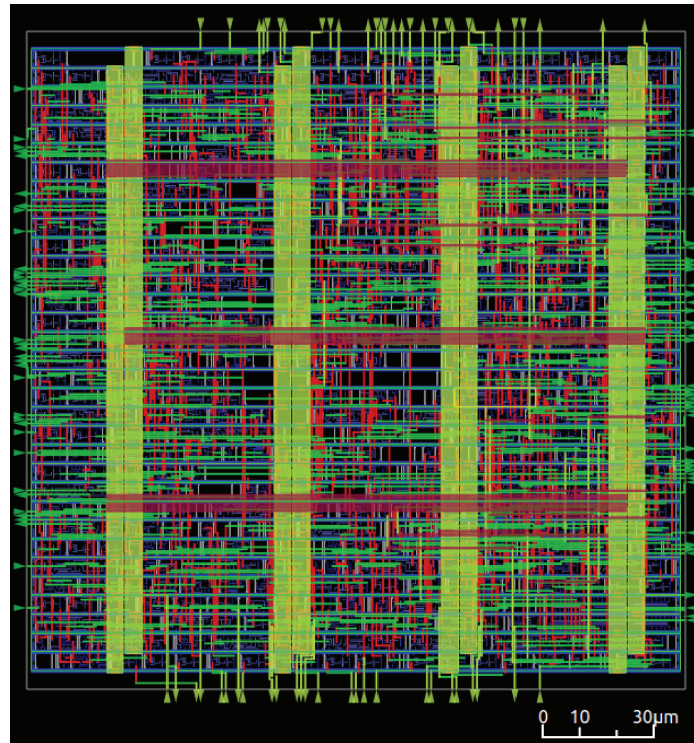


Figure 10: Layout result of 64 bit counter using DFFSR\_G.

## V. Conclusion

We proposed the DFFSR\_G cell as a novel DFF with set/reset function using the GDI technique. DFFSR\_G is the cell that solves the problems of cascaded

GDI and applies the advantages of GDI. The layout area of this cell is approximately 33% smaller than that of a CMOS cell with the same operation, so it is expected to significantly reduce the area occupied by the sequential circuit in the full chip. As a sample

**Table 4: Summary of 64 bit counter P&R.**

|                                           | DFFSR_C | DFFSR_G |
|-------------------------------------------|---------|---------|
| Number of I/O ports                       | 131     |         |
| Number of nets                            | 390     |         |
| Number of cells                           | 196     |         |
| Combinational area ( $\mu\text{m}^2$ )    | 4301.0  | 4301.0  |
| Noncombinational area ( $\mu\text{m}^2$ ) | 5535.1  | 4317.4  |
| Total cell area ( $\mu\text{m}^2$ )       | 9836.2  | 8618.4  |

P&R, placement and routing.

design to verify the applicability of the RTL design flow and the performance of the DFFSR\_G cell, a 64-bit counter with set/reset operations was designed, and layout was performed through P&R. The number of synthesized logic circuit cells is 196, and the chip size was measured as  $8,618.4 \mu\text{m}^2$ . The P&R performance results show an area reduction effect of about 22% based on the sequential circuit. If DFFSR\_G is applied to implement a large-scale integrated circuit, it is expected to be effective in designing a chip with similar power consumption and a small area. Future work will focus on improving operating speed.

## Acknowledgments

This work was supported by Hanshin University Research Grant.

## References

- [1] Seungmin Jung, "Implementation of Novel GDI D-Flip-Flop for RTL Design," *Journal of Computing Science and Engineering*, Vol. 17, No. 4, Dec 2023, pp. 161-168. <https://doi.org/10.5626/JCSE.2023.17.4.161>
- [2] Seungmin Jung, "A Study on the VLSI Implementation of Fingerprint Thinning Processors Using Hybrid GDI Technique," *Journal of Computing Science and Engineering*, Vol. 17, No. 1, March 2023, pp. 20-29. <https://doi.org/10.5626/JCSE.2023.17.1.20>
- [3] Malik Summar Asghar, Saad Arslan, Ali A. Al-Hamid and HyungWon Kim, "A Compact and Low-Power SoC Design for Spiking Neural Network Based on Current Multiplier Charge Injector Synapse," *Sensors MDPI*, Vol. 23, No. 14, Jul 2023. <https://doi.org/10.1109/tcsii.2020.3027879>
- [4] Hyeopgoo Yeo, "Touch fingerprint sensor based on sensor cell isolation technique with pseudo direct signaling," *International Journal on Smart Sensing and Intelligent Systems*, vol.12, no.1, Apr. 2019. <https://doi.org/10.21307/ijssis-2019-001>
- [5] Seungmin Jung, "Image Processor and RISC MCU Embedded Single Chip Fingerprint Sensor," *Journal of Sensor and Actuator Networks*, vol. 9, no.4, Nov. 2020. <https://doi.org/10.3390/jsan9040051>
- [6] S. Savadipalayam Venkatachalam, S. Sabapathy Arumugam and S. Sivasubramaniyam, "Design of Low Power Flip Flop Based on Modified GDI Primitive Cells and Its Implementation in Sequential Circuits," *International Journal of Advances in Computer and Electronics Engineering* Vol.2 No.5, pp. 22 – 32, May 2017. <https://doi.org/10.1109/ICCSP.2016.7754397>
- [7] Jin-Fa Lin, Zheng-Jie Hong, Chang-Ming Tsai, Bo-Cheng Wu and Shao-Wei Yu, "Novel Low-Complexity and Low-Power Flip-Flop Design," *Electronics MDPI*, Vol. 9, No. 5, May 2020. <https://doi.org/10.3390/electronics9050783>
- [8] Arkadiy Morgenshtein, Alexander Fish and Israel A. Wagner, "An efficient implementation of D-flip-flop using the GDI technique," *IEEE International Symposium on Circuits and Systems (ISCAS) 2004, Proceedings of the 2004 International Symposium 2004*. <https://doi.org/10.1109/ISCAS.2004.1329361>
- [9] N. Kandasamy, Chethana Sanjeevaiah, Nagarjuna Telagam and Ramya Merisala, "Hybrid 4:16 Decoder Using Variable Bias GDI Technique," *Lecture Notes in Electrical Engineering* book (LNEE), Vol. 711, Feb. 2021.
- [10] Mehedi Hasan, Hasan U. Zaman, Mainul Hossain, Parag Biswas, Sharnali Islam, "Gate Diffusion Input technique based full swing and scalable 1-bit hybrid Full Adder for high performance applications," *Engineering Science and Technology, an International Journal*, Vol. 23, No. 6, pp 1364-1373, Dec 2020. <https://doi.org/10.1016/j.jestch.2020.05.008>
- [11] Gautam Nayan, Rajesh Kumar Prasad, Praveen Kumar Y G, Dr. M Z Kurian, "A Review On Modified Gate Diffusion Input Logic: An Approach For Area And Power Efficient Digital System Design," *Proceedings of Second International Conference on Emerging Trends In Science & Technologies For Engineering Systems (ICETSE-2019)*, Jul 2020.
- [12] E. Abiri, Abdolreza DarabiSanaz Salem, "Design of multiple-valued logic gates using gate-diffusion input for image processing applications," *Computers & Electrical Engineering* Vol. 69, pp142-157, Jul. 2018. <https://doi.org/10.1016/j.compeleceng.2018.05.019>

- [13] G.R.Mahendra Babu, S.Bhavani, "Primitive Cells using Gate Diffusion Input Technique: a Low Power Approach," International Journal of Recent Technology and Engineering, Vol. 8, No 1S5, June 2019.
- [14] Arkadiy Morgenshtein, Alexander Fish, and Israel A. Wagner, "Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 10, No. 5, Oct. 2002. <https://doi.org/10.1109/TVLSI.2002.801578>
- [15] Nehru KandasamyN. Mohan Kumar, Nagarjuna Telagam, Girish Shankar Mishra, "Analysis of Self Checking and Self Resetting Logic in CLA and CSA Circuits Using Gate Diffusion Input Technique," International Journal of Advances in Computer and Electronics Engineering, pp. 22 – 32, Vol.2 No.5, Nov 2019. <https://doi.org/10.1109/ICSSIT46314.2019.8987817>
- [16] M. Hasan, U.K. Saha, A. Sorwar, M.A.Z. Dipto, M.S. Hossain, H.U. Zaman, "A novel hybrid full adder based on gate diffusion input technique, transmission gate and static CMOS logic," IEEE 2019 10th International Conference on Computing, Communication and Networking Technologies, 2019. <https://doi.org/10.1109/ICCCNT45670.2019.8944888>
- [17] A. Gard, G. Joshi, "Gate diffusion input based 4-bit vedic multiplier design," The Institution of Engineering and Technology, Vol. 12, No. 6, 2018. <https://doi.org/10.1049/iet-cds.2017.0454>
- [18] A. Morgenshtein, I. Shwartz, A. Fish, "Full swing gate diffusion input (GDI) logic: case study for low power CLA adder design," INTEGRATION, the VLSI journal, Vol. 47, No. 1, 2014. <https://doi.org/10.1016/j.vlsi.2013.04.002>
- [19] M. Shoba, R. Nakkeeran, "Energy and area efficient hierarchy multiplier based on vedic mathematics and GDI logic," Engineering Science and Technology, an International Journal, pp 321-331, Vol. 20, No. 1, Feb. 2017. <https://doi.org/10.1016/j.jestch.2016.06.007>