

RTL Design Technique Using Libraries for GDI MOS and CMOS Mixed Mode Synthesis

Seungmin Jung*

Division of Software Convergence, Hanshin University, Osan, Korea
jasmin@hs.ac.kr

Abstract

We propose a novel register transfer level (RTL) design technique that applies both gate diffusion input (GDI) and CMOS process design kit (PDK). We create the GDI-CMOS PDK by merging the GDI synthesis library, which is made by extracting the characteristics of GDI cells, with the CMOS library. We synthesize a 16-bit RISC MCU sample circuit by applying the new RTL design technique, and compare and analyze a performance index, PPA. The proposed design technique can be applied as a new design flow by reducing the disadvantages of GDI and maximizing the advantages of area and power, thereby overcoming the disadvantages of CMOS circuits.

Category: Embedded Systems

Keywords: GDI (gate diffusion input); PDK (process design kit); RTL (register transfer level) design; Synopsys DC (design compiler) library; Logic synthesis; PPA (power, performance, area)

I. INTRODUCTION

Despite the dominance of high-end semiconductor processes like 5 nm to 3 nm, conventional fabrications are still widely used due to their cost-effectiveness, sufficient performance for many applications, and greater reliability in mature designs. Additionally, analog-digital mixed mode circuits and long-lifecycle products benefit from the stability and efficiency offered by larger nodes, which are often better suited for power-sensitive or legacy systems [1-4]. Fingerprint recognition system-on-a-chip (SoC) is an example [4]. It does not need to be smaller in area and it is much more efficient to apply a low-cost process. Power and area reduction is the important parameter which decides the efficiency of VLSI design. In this sense, the gate diffusion input (GDI) technique is expected to enable low power and small cell layout [5-

17]. Fig. 1 shows the GDI base cell, 'AND' and 'OR' state. Different logic functions can be implemented using this single GDI cell as shown in Table 1.

The GDI technique offers significant advantages, such as low power consumption and reduced transistor count, which make it highly suitable for low-power applications. Specially, it allows for compact circuit designs with fewer logic gates, enhancing area efficiency. However, GDI circuits can suffer from signal integrity issues, such as incomplete voltage swings, and pose design complexity challenges compared to traditional CMOS technology, particularly in maintaining reliability across different applications [1].

In previous studies, the GDI circuit was applied in a full-custom design method only to small circuits such as combinational adders or multipliers [5-14]. The application of the GDI technique presents several challenges in

Open Access <http://dx.doi.org/10.5626/JCSE.2024.18.4.196>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 01 December 24; Accepted 10 December 24

*Corresponding Author

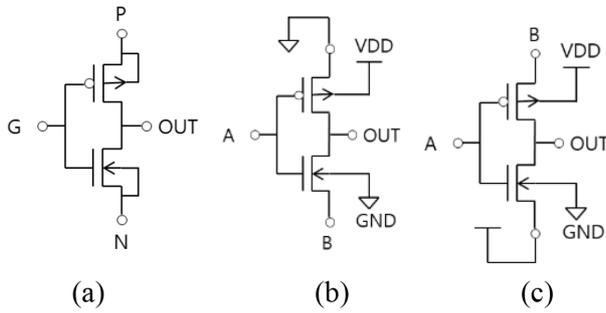


Fig. 1. GDI (a) basic cell and (b) AND, (c) OR state.

Table 1. GDI basic function

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

register transfer level (RTL) design flows. GDI-based circuits require non-standard cell libraries that differ from traditional CMOS, complicating integration into conventional RTL-to-GDSII synthesis tools. Moreover, signal integrity issues such as incomplete voltage swings increase the difficulty of maintaining reliable functionality during synthesis and optimization. The technique also demands custom cell characterization, making it difficult to apply automated design flows, which are optimized for standard CMOS libraries. These reasons make it difficult to adopt the typical cell-based RTL design flow in the GDI library. In this paper, we present a method for applying GDI technology to RTL design flow and analyze the results applied to sample circuits.

In Section II, we describe the problems of conventional GDI circuits design and propose a new hybrid GDI method to improve them. In Sections III and IV, we generate a process design kit (PDK) by extracting the electrical characteristics of 11 GDI cells, design a sample RTL circuit using the proposed algorithm, and compare and analyze its performance with previous CMOS results. Conclusions and future work are discussed in Section V.

II. DISADVANTAGES OF GDI TECHNIQUE

The possible output voltages can be 0, V_{TN}, V_{DD}-V_{Tp}, and V_{DD} depending on the input stimulus due to the structure of the GDI basic cell. If the circuit is designed with only GDI cells, the output characteristics need to be analyzed. In particular, since the bulk bias

Table 2. HSPICE simulation result of series GDI logic (1.8 V, 180n standard CMOS process)

A	B	GDI operation		CMOS operation		Function error	
		OR	AND	OR	AND	OR	AND
L	WL	0	0	0	0	O	O
L	WH	1	0	1	0	O	O
WL	L	1	0	0	0	x	O
WL	WL	1	0	0	0	x	O
WL	WH	1	0	1	0	O	O
WL	H	1	0	1	0	O	O
WH	L	1	0	1	0	O	O
WH	WL	1	0	1	0	O	O
WH	WH	1	0	1	1	O	x
WH	H	1	0	1	1	O	x
H	WL	1	0	1	0	O	O
H	WH	1	1	1	1	O	O

V_{BS} is not 0, the output voltage characteristics become complicated. If the intermediate levels of V_{DD}-V_{Tp} and V_{Tn} are input to the next-stage GDI cell, the characteristics need to be analyzed. In a GDI cell with two inputs, there are 16 possible combinations of four voltage levels. In this paper, the OR and AND cell operations were measured with HSPICE for 12 input stimuli including the intermediate voltage. Here, V_{Tn} is designated as WL (weak low) and V_{DD}-V_{Tp} is designated as WH (weak high) in the truth table. Table 2 shows the simulation results of the GDI OR and AND cells. Operational errors occurred in the combination of WL and L in the OR cell and in the input combination of WH and H in the AND cell. 'x' indicates a functional failure of the GDI cascade circuit compared to CMOS. Failure of the GDI cell to achieve the full voltage swing can result in operational failure, especially when the cells are arranged in series. In this configuration, the incomplete voltage level can propagate through the series-connected cells, causing the subsequent stages to interpret ambiguous logic levels, resulting in incorrect operation. This voltage degradation makes GDI cells unsuitable for complex multi-stage designs where reliable signal transmission is essential. Series GDI designs require innovative solutions.

III. PROPOSED GDI TECHNIQUE FOR RTL FLOW

A. GDI-CMOS PDK Generation

In this study, the characteristics of the primitive 11

Table 3. Characteristic results of 11 GDI Cells (unit: %)

Cell	Area	Power	Timing delay		
			x1	x3	x10
OR2	67	77	-22	-107	-297
OR3	63	77	37	6	-68
OR4	60	82	57	42	3
AND2	67	78	-37	-102	-166
AND3	63	78	-24	-89	-161
AND4	60	78	-7	-70	-148
MUX	83	80	-30	-106	-230
XOR2	67	60	59	48	20
XOR3	60	82	10	-17	-80
DFF	50	-4.7	20	6	-12
DFFRS	22	1	15	4	-2
Average improvement	60	63	7	-35	-104

GDI cells were extracted to generate the PDK for GDI. The PDK includes the layout area, delay time, and power consumption of each cell, and is applied as a library for RTL synthesis. The simulation was performed with HSPICE using 180n standard CMOS process parameters at 1.8 V operating voltage. The rise and fall times of the input signals are 100 ps, and the CL (load capacitance) of the output pins are applied to x1, x3, and x10 of the unit fan-out, respectively. The length of the MOS is 180n, and the width is the same as the CMOS inverter. The simulation period is 20 ns. Table 3 compares the MOS transistor count, power consumption, and operating speed of the GDI and CMOS basic cells.

It is also shown that the delay time increases sharply compared to CMOS as CL increases. Therefore, the load of the output pin in the GDI circuit is expected to be synthesized less than the fan-out x3 as much as possible. The basic sequential cells are the D-flip-flop (DFF) gate from the standard cell library and the D-flip-flop with set-reset function (DFFSR) gate. These circuits consist of two GDI multiplexers (MUXs) and CMOS inverters [2]. In GDI technique, sequential cells have the advantage of reducing the chip area by more than 30% compared to CMOS. In addition, the output stage always has CMOS level voltage, unlike combinational GDI circuits. This provides an advantageous aspect for GDI netlist synthesis. The GDI-CMOS PDK was created with the characteristics of 11 GDI cells, each identical to the 11 CMOS ones. The GDI-CMOS library also includes NOT gates such as NAND, NOR, and inverter applied in CMOS. In the GDI-CMOS library, the cell name convention is set to three groups: '*_CMOS,' '*_GDI,' and '*_GDI_CMOS.' The '*_GDI_CMOS' means sequential circuit DFF and DFFSR. Grouping cell names can give various constraints

such as synthesis of only CMOS cells, synthesis of only GDI cells, and hybrid mode during Synopsys design compiler (DC) synthesis.

B. GDI Technique for RTL Design

In logic circuit synthesis, the continuous connection of GDI cells causes circuit operation errors. A solution to improve this problem is to place CMOS cells at the output of GDI cells. This approach can complement the weaknesses of the GDI library by applying the robust driving function and signal integrity of CMOS cells. By inserting CMOS cells between GDI cells, the design

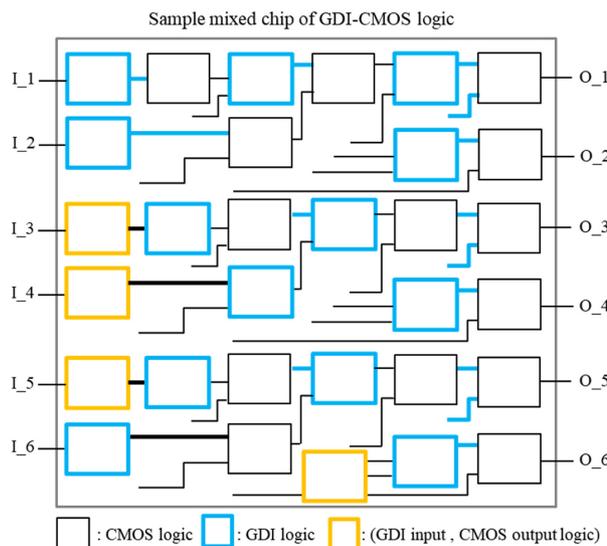


Fig. 2. Proposed GDI, CMOS mixed technique.

achieves improved reliability while sacrificing less power and area advantages of GDI. This technique ensures proper signal propagation, reducing the risk of functional errors in complex digital circuits. Fig. 2 shows the structure of the proposed GDI-CMOS chip. I_1 to 6 represent input pins, and O_1 to 6 represent output pins. The blue box represents GDI logic, and the orange box represents a flip-flop cell such as DFF or DFFSR. These circuits are designed as a composite structure where the input stage is GDI circuit and the output stage represents CMOS level output. The rules of circuit configuration are as follows.

- 1) Since all inputs outside the chip are CMOS level, input pins in the mixed-mode chip top use GDI cells and output pins use CMOS cells.
- 2) The current cell must be a CMOS cell if at least one of its input pins is connected to the output of the GDI cell.
- 3) The inputs of DFF and DFFSR cells are GDI level and the outputs are CMOS level, so the input terminals connect CMOS cells and the output terminals connect GDI cells.

Implementing this approach requires several preparatory steps during the design phase. The first is customizing the GDI and CMOS mixed cell libraries. Both GDI and CMOS cells must be available in the standard cell library and have accurate characterization data including timing, power, and area metrics. This allows the synthesis tool to properly utilize both cell types during the design process. The previous section described the creation process of the GDI-CMOS PDK. The second is setting design constraints. The synthesis tool, such as Synopsys DC, must be configured with appropriate constraints that force the GDI cell output terminals to be placed with CMOS cells.

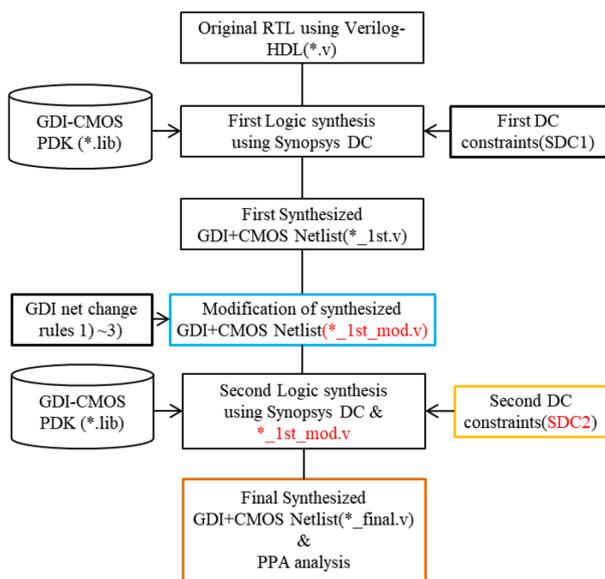


Fig. 3. Proposed GDI-CMOS RTL design flow.

These constraints must include design rules that specify that the GDI output must be connected to the CMOS input. However, commercial logic synthesizers such as Synopsys DC and Cadence Genus do not provide these constraints. Therefore, this study applied the design flow as shown in Fig. 3.

Logic synthesis is performed in two stages. In the first stage, synthesis is performed by applying GDI-CMOS PDK and constraints 'SCD1.' Fig. 4 shows the SCD1 script and Fig. 5 shows an example of the netlist synthesis result for a simple RTL. As shown in Fig. 5, some of the cells synthesized as GDI are changed to CMOS cells according to the modified netlist rules 1) to 3) suggested in the previous section. In the example of Fig. 6, the instances U13 and U15 cells are changed from the existing GDI to CMOS cells. This is due to the rule of prohibiting the placement of consecutive GDI cells. In the second stage, the synthesis is performed by applying the modified netlist and constraints 'SDC2' constraints.

```

read_verilog -r ./mixed_synth/mixed_synth_rtl.v
current_design mixed_synth

link
uniquify

compile
write -format verilog -hierarchy -
output ./mixed_synth/mixed_synth_gdi_1st.v

create_clock -name clk -period 2.0 [get_ports clk]
set_max_area 0

check_design
compile

write -format verilog -hierarchy -
output ./mixed_synth/mixed_synth_gdi_1st.v
    
```

Fig. 4. Constraints 'SDC1' scripts example.

```

module mixed_synth ( a, b, c, d, out, clk );
input a, b, c, d, clk;
output out;
wire y2, N12, y, N25, n7, n8, n9, n10;
DFFX1_GDI_CMOS y1_reg ( .D(n7), .CK(clk), .QN(N25) );
DFFX1_GDI_CMOS y2_reg ( .D(1'b0), .CK(clk), .Q(y2) );
DFFX1_GDI_CMOS y_reg ( .D(N12), .CK(clk), .Q(y) );
NOR4BBX1_CMOS U12 ( .AN(c), .BN(d), .C(a), .D(b), .Y(n7) );
OR2X1_GDI U13 ( .A(n8), .B(N25), .Y(N12) );
AND2X1_GDI U14 ( .A(y2), .B(n10), .Y(n8) );
OR2X1_GDI U15 ( .A(n9), .B(c), .Y(n10) );
AND2X1_GDI U16 ( .A(a), .B(d), .Y(n9) );
OR3X1_GDI U17 ( .A(d), .B(a), .C(y), .Y(out) );
endmodule
    
```

Fig. 5. First synthesized netlist (*_1st.v) example.

```

input a, b, c, d, clk;
output out;
wire y2, N12, y, N25, n7, n8, n9, n10;
DFFX1_GDI_CMOS y1_reg (.D(n7), .CK(clk), .QN(N25));
DFFX1_GDI_CMOS y2_reg (.D(1'b0), .CK(clk), .Q(y2));
DFFX1_GDI_CMOS y_reg (.D(N12), .CK(clk), .Q(y));
NOR4BBX1_CMOS U12 (.AN(c), .BN(d), .C(a), .D(b), .Y(n7));
OR2X1_CMOS U13 (.A(n8), .B(N25), .Y(N12));
AND2X1_GDI U14 (.A(y2), .B(n10), .Y(n8));
OR2X1_CMOS U15 (.A(n9), .B(c), .Y(n10));
AND2X1_GDI U16 (.A(a), .B(d), .Y(n9));
OR3X1_GDI U17 (.A(d), .B(a), .C(y), .Y(out));
endmodule
    
```

Fig. 6. Modification of synthesized netlist (*_1st_mod.v).

```

read_db_cmos_gdi.db

read_verilog -r ./mixed_synth/mixed_synth_gdi_1st_mod.v
current_design mixed_synth

link
uniquify

create_clock -name clk -period 2.0 [get_ports clk]
set_max_area 0
set_dont_use [get_cells cmos_gdi/*_GDI]
set_dont_touch [get_cells *]

check_design
compile

write -format verilog -hierarchy -
output ./mixed_synth/mixed_synth_gdi_final.v
    
```

Fig. 7. Constraints 'SDC2' scripts example.

SDC2 is set to extract the final PPA that applies the GDI-CMOS PDK without changing the modified netlist during the synthesis process. Fig. 7 shows the SDC2 script. SDC2 includes constraints to ensure that a first-order revision of the synthesized netlist is maintained, and thus compiles only once, unlike SDC1.

IV. Implementation of 16-Bit RISC MCU

In this paper, we implemented the 16-bit reduced instruction set computing (RISC) microcontroller (MCU) to evaluate the performance of a chip that applied the RTL design flow using the proposed GDI-CMOS PDK. In general, application processors are representative IP of system semiconductors and are used as samples suitable for evaluating and comparing the performance of new foundry processes. We compared and analyzed the results with those of a general CMOS-only RTL design flow. The 16-bit RISC processor is implemented using the

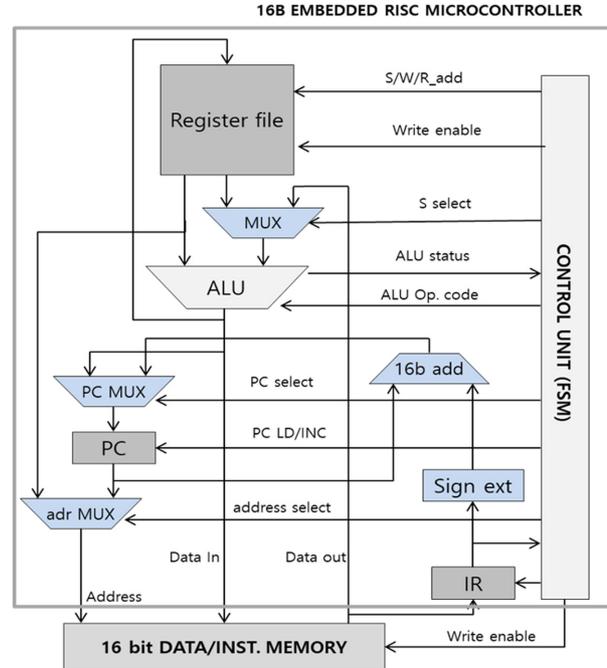


Fig. 8. Functional block diagram of 16-bit RISC MCU.

Verilog hardware description language (HDL), as shown in Fig. 8. The processor has a four-stage pipeline. The applied embedded MCU is based on Harvard architecture with separate data memory and instruction memory to avoid bottlenecks and enable pipelines. Table 4 shows the comparison of the synthesis results of the normal 1.8 V 180nm CMOS library and the GDI-CMOS library for the sample circuit, a 16-bit RISC MCU, using Synopsys DC reporting PPA. The GDI-CMOS library was applied by applying the new RTL design technique proposed in this study, as shown in Fig. 3.

The synthesized circuit scale showed an instance count of 1670 to 1680, and above all, it showed excellent results in the total chip area. The chip area applied with the GDI-CMOS library was 22,434 μm^2 , showing an area reduction effect of approximately 48% or more. Since it is Synopsys DC result, the routing area is not included. In terms of power consumption, there is not much difference, but it also showed excellent results in dynamic switching power and leakage power. In terms of operating speed, the synthesis results with a design constraint of 9 ns period did not show TNS (total negative slack) in either case, and the CMOS circuit is expected to be slightly superior to the GDI-CMOS library. This is because the CMOS has about 0.48 ns of margin in the value of CPS (critical path slack), so the operating period can be further reduced while the maximum operating frequency can be increased. This trend is expected to continue even if the cell types of the GDI-CMOS library are increased and a larger sample RTL is applied. Ultimately, it is expected that a much superior chip can be developed in terms of

Table 4. RTL design results of 16-bit RISC MCU (180n PDK)

		PDK library for Synopsys DC	
		CMOS	GDI-CMOS
Performance (speed) (@clock period: 9 ns)	Critical path length (ns)	8.2	8.76
	Critical path slack (ns)	0.63	0.15
	Total negative slack (ns)	0	0
Power consumption (@VDD=1.8 V)	Cell internal power (mW)	2.3097	2.1939
	Net switching power (uW)	509.6	263.7
	Total dynamic power (mW)	2.8193	2.4577
	Cell leakage power (nW)	284.3	175.4
Chip area	Combinational area (μm^2)	27,815.2	16,744.0
	Non-combinational area (μm^2)	15,381.3	7,690.6
	Total area (μm^2)	43,196.6	24,434.8
	Leaf cell (instance) count	1,680	1,670
RTL design technique		One-step synthesis	Two-step synthesis

PPA by applying the GDI-CMOS library and a new RTL design technique.

V. CONCLUSION

In this paper, we propose a new RTL design technique that utilizes both GDI PDK and CMOS PDK. We develop a PDK library by extracting the electrical characteristics of 11 sample GDI cells, and generate a GDI-CMOS PDK by integrating it with a CMOS library. We synthesize a 16-bit RISC MCU sample circuit by applying the new RTL design technique, and compare and analyze the performance index PPA. The proposed design technique can be applied as a new design technique that overcomes the shortcomings of CMOS circuits by reducing the shortcomings of GDI and maximizing the advantages of area and power. We compare and analyze the characteristics of hybrid GDI-based combinational and sequential basic gates and CMOS gates in a 1.8 V, 180 n CMOS process. We also propose a circuit synthesis algorithm. This algorithm replaces the entire circuit composed of only CMOS with a hybrid GDI circuit containing more than 50% of GDI cells. The 16-bit RISC MCU was synthesized with Synopsys DC using the GDI-CMOS 180n library developed as a sample circuit based on Verilog-HDL. The synthesized netlist size had a maximum instance count of 1670 and a total cell area of 24434 $\mu\text{m} \times \mu\text{m}$ excluding the routing area. The chip area applied with the GDI-CMOS library showed an area reduction effect of about 48% or more compared to the CMOS-only synthesis result. There is no significant difference in terms of power consumption, but it also shows excellent results in dynamic switching power and leakage power. In terms of operation speed, CMOS

has a margin of about 0.48ns in the critical path margin (CPS) value, which seems to increase the maximum operation frequency and further shorten the operation period. The verification results show that the proposed hybrid GDI technique can be applied to SoC designs with small cell area and low power consumption.

CONFLICT OF INTEREST

The author has declared that no competing interests exist.

ACKNOWLEDGEMENTS

This work was supported by Hanshin University Research Grant.

REFERENCES

1. S. 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, pp. 20-29, 2023. <https://doi.org/10.5626/JCSE.2023.17.1.20>
2. S. Jung, "Implementation of novel GDI D-Flip-Flop for RTL design," *Journal of Computing Science and Engineering*, vol. 17, no. 4, pp. 161-168, 2023. <https://doi.org/10.5626/JCSE.2023.17.4.161>
3. H. 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, pp. 1-9, 2019. <https://doi.org/10.21307/ijssis-2019-001>
4. S. Jung, "Image processor and RISC MCU embedded

- single chip fingerprint sensor,” *Journal of Sensor and Actuator Networks*, vol. 9, no. 4, article no. 51, 2020. <https://doi.org/10.3390/jsan9040051>
5. N. Kandasamy, C. Sanjeevaiah, N. Telagam, and R. Merisala, “Hybrid 4:16 decoder using variable bias GDI technique,” in *Advances in Electrical and Computer Technologies*. Singapore: Springer, 2020, pp. 637-647. https://doi.org/10.1007/978-981-15-9019-1_55
 6. G. Nayan, R. K. Prasad, Y. G. Praveen Kumar, and D. M. Kurian, “A review on modified gate diffusion input logic: an approach for area and power efficient digital system design,” in *Proceedings of the Second International Conference on Emerging Trends in Science & Technologies For Engineering Systems (ICETSE)*, Chickballapur, Karnataka, India, 2019, pp. 1-17. <https://doi.org/10.2139/ssrn.3507293>
 7. M. Hasan, H. U. Zaman, M. Hossain, P. Biswas, and S. 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, 2020. <https://doi.org/10.1016/j.jestch.2020.05.008>
 8. M. Hasan, U. K. Saha, A. Sorwar, M. A. Z. Dipto, M. S. Hossain, and H. U. Zaman, “A novel hybrid full adder based on gate diffusion input technique, transmission gate and static CMOS logic,” in *Proceedings of 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, pp. 1-6. <https://doi.org/10.1109/ICCCNT45670.2019.8944888>
 9. N. Kandasamy, N. M. Kumar, N. Telagam, F. Ahmad, and G. Mishra, “Analysis of self checking and self resetting logic in CLA and CSA circuits using gate diffusion input technique,” in *Proceedings of 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2019, pp. 1-6. <https://doi.org/10.1109/ICSSIT46314.2019.8987817>
 10. G. M. Babu and S. Bhavani, “Primitive cells using gate diffusion input technique: a low power approach,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 1(S5), pp. 257-260, 2019.
 11. E. Abiri, A. Darabi, and S. Salem, “Design of multiple-valued logic gates using gate-diffusion input for image processing applications,” *Computers & Electrical Engineering*, vol. 69, pp. 142-157, 2018. <https://doi.org/10.1016/j.compeleceng.2018.05.019>
 12. M. Shoba and R. Nakkeeran, “Energy and area efficient hierarchy multiplier architecture based on Vedic mathematics and GDI logic,” *Engineering Science and Technology; an International Journal*, vol. 20, no. 1, pp. 321-331, 2017. <https://doi.org/10.1016/j.jestch.2016.06.007>
 13. A. Garg and G. Joshi, “Gate diffusion input based 4-bit Vedic multiplier design,” *IET Circuits, Devices & Systems*, vol. 12, no. 6, pp. 764-770, 2018. <https://doi.org/10.1049/iet-cds.2017.0454>
 14. A. Morgenshtein, V. Yuzhaninov, A. Kovshilovsky, and A. Fish, “Full-swing gate diffusion input logic: case-study of low-power CLA adder design,” *Integration*, vol. 47, no. 1, pp. 62-70, 2014. <https://doi.org/10.1016/j.vlsi.2013.04.002>
 15. S. S. Venkatachalam, S. S. Arumugam, and S. Sivasubramaniam, “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, 2017.
 16. A. Morgenshtein, A. Fish, and I. A. Wagner, “An efficient implementation of D-Flip-Flop using the GDI technique,” in *Proceedings of 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, Vancouver, Canada, 2004, pp. 673-676. <https://doi.org/10.1109/ISCAS.2004.1329361>
 17. A. Morgenshtein, A. Fish, and A. I. 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, pp. 566-581, 2002. <https://doi.org/10.1109/TVLSI.2002.801578>



Seungmin Jung <https://orcid.org/0000-0002-2324-1734>

Seungmin Jung received the B.S., M.S., and Ph.D. degrees in Department of Electronic Engineering from Yonsei University, Seoul, Korea, in 1990, 1992, and 2006, respectively. From 1992 to 1997, he was Senior Engineer of the System LSI Division at Samsung Electronics Co. Ltd., where he worked on the design of standard cell library, synchronous and asynchronous compiled memory circuits and also worked on the design of full-custom and semi-custom VLSI. From 1998 to 2006, he joined the faculty of Yong-In Arts & Science University, Yongin, Korea, where he was assistant professor in the Information and Communication Department. In 2006, he joined the faculty of Hanshin University, Osan, Korea, where he is currently professor in Division of Software Convergence. His research interests include biometric CMOS sensors, SoC (System On Chip) design and mixed-mode VLSI circuit design. In particular, his recent research interests have focused on designing hardware accelerators for all or part of algorithms, including computer vision and AI for object recognition.