

Design and Analysis of 3D IC-Based Low Power Stereo Matching Processors

Seung-Ho Ok¹, Kyeong-ryeol Bae¹, Sung Kyu Lim², and Byungin Moon^{1*}

¹School of Electronics Engineering, Kyungpook National University, Daegu 702-701, Korea

²School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

*bihmoon@knu.ac.kr

Abstract—This paper presents comprehensive design and analysis results of 3D IC-based low-power stereo matching processors. Our design efforts range from architecture design and verification to RTL-to-GDSII design and sign-off analysis based on GlobalFoundries 130-*nm* PDK. We conduct comprehensive studies on the area, performance, and power benefits of our 3D IC designs over 2D IC designs. Our 2-tier 3D IC designs attain 43% area, 14% wire length, and 13% power saving over 2D IC designs. We also study a pipeline-based partitioning method shown to be effective at minimizing power consumption and the total number of TSVs while balancing the size of each tier.

Keywords—Stereo matching processor, 3D IC, TSV, low-power design

I. INTRODUCTION

Stereo matching is the process of finding corresponding pixels in a pair of stereo images and extracting depth information by computing the disparity of the corresponding pairs of pixels [1]. The stereo matching processor is widely used in embedded systems such as mobile robots, intelligent vehicles, and unmanned aerial vehicles that demand real-time processing capability with low-power consumption and hardware miniaturization. To meet these requirements, we have used 3D stacking technology based on through-silicon-via (TSVs). Over the last decade, a large variety of stereo matching algorithms have been proposed and categorized in two groups: local and global algorithms [2]. In this paper, we adopted a local algorithm for the implementation of a 3D IC-based stereo matching processor mainly because local algorithms are more straightforward in a pipelined hardware architecture. Thus, they can support real-time processing capabilities more efficiently.

We design our stereo matching processor based on 3D IC technology and compare it with its 2D IC counterpart in terms of wire length and power consumption. For a fair comparison of 2D and 3D ICs, we use the same synthesis output and target clock period during the layout and timing optimization. In this work, we used a macro-level partitioning method that minimizes the total wire length between two tiers of 3D ICs. In addition, we propose a pipeline-based partitioning method that can minimize the total number of TSVs while balancing the size of each tier in a pipelined hardware architecture.

The literature contains several studies related to 3D IC technology. Thorolfsson et al. [3] implemented a 3D IC FFT processor by stacking memory on logic and compared it to a 2D IC. They used the same synthesis output for a fair comparison. Although the aim of this approach is similar to ours, they did not perform a power comparison under the same clock frequency. In addition, they used only one type of partitioning method for the 3D IC design. Neela et al. [4] discussed the 3D IC implementation of a single precision floating-point unit. However, they did not use the same netlist output for the 2D and 3D IC designs. In addition, they only considered the partitioning of logic gates into 3D. Kim et al. [5] demonstrated a many-core processor and

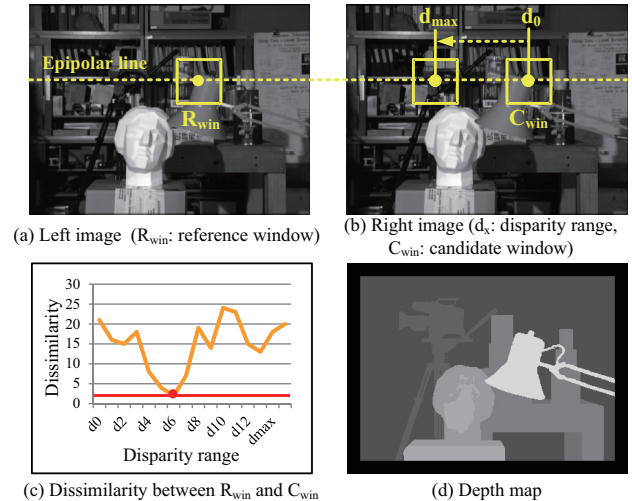


Fig. 1. Illustration of a window-based local stereo matching algorithm.

memory stacking to show the benefits of TSV-based 3D integration, and Tao et al. [6] demonstrated the feasibility of DRAM stacking by implementing SoC for multimedia applications.

To achieve real-time processing capabilities with low power consumption and hardware miniaturization, we implement a stereo matching processor based on 3D IC technology. Because of its high memory bandwidth and high performance requirements for real-time processing, the stereo matching processor can fully exploit the benefits of 3D stacking technology.

The main contributions of this paper are as follows. (1) To the best of our knowledge, it presents the first 3D IC-based stereo matching processor. (2) It presents a comprehensive study based on practical implementations of a 3D IC-based stereo matching processor that targets power, area, and wire length reduction and finds that our 2-tier 3D IC designs require 43% less area, 14% shorter wire length, and 13% less power than 2D ICs. (3) We present a pipeline-based partitioning method shown to be effective at minimizing power consumption and the total number of TSVs while balancing the size of each tier.

II. STEREO MATCHING PROCESSOR

A. Matching Algorithm

Fig. 1 illustrates a window-based local stereo matching algorithm. As shown in Fig. 1, this algorithm requires a fixed size of window to find corresponding pixels between a pair of stereo images on the same epipolar line [2]. As a result, it also requires a large number

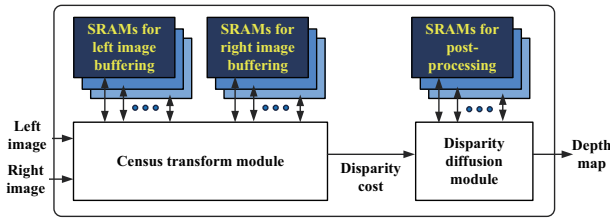


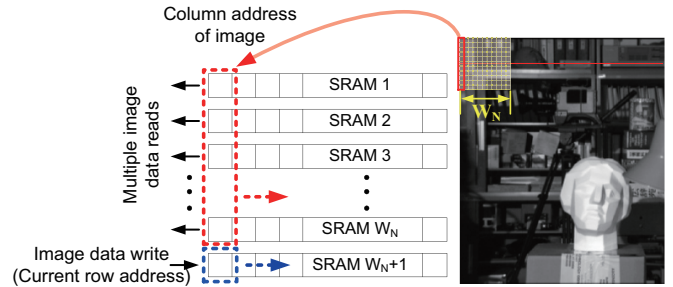
Fig. 2. A block diagram of our stereo matching processor.

of memory macros in the hardware implementation to implement the window-based matching algorithm. Fig. 2 shows a block diagram of the stereo matching processor. The census transform is a non-parametric local transform that presents a characteristic feature of the window as a sequence of bit streams [7]. The algorithm performs the census transform on both left and right images to find corresponding pixels in a pair of stereo images and computes the matching cost of the window using the Hamming distance. After window-based matching, the disparity cost propagates to the disparity diffusion module, improving the accuracy of the final dense depth map. Usually, window-based local stereo matching algorithms have difficulty finding accurate corresponding pixels in a featureless area mainly because of the lack of characteristics in the window. To improve the accuracy of the disparity cost in the featureless area, the disparity diffusion module recovers the disparity cost in the featureless area by diffusing the accurate disparity cost of the neighbor non-featureless area. This diffusion method is based on the assumption that adjacent pixels have the same disparity.

B. Hardware Architecture

The matching algorithm requires sufficient memory for window-based stereo matching to find corresponding pixels from a pair of stereo images. Thus, the primary goal of the stereo matching architecture is to efficiently buffer both left and right images on the memory so that the stereo matching processor can generate a depth map in real time. To fulfill this requirement and deal with the requirement of the wide bandwidth and high-speed data accessing, we used small highly partitioned SRAMs and conducted a window-based operation using a finite number of them. Fig. 3 illustrates how the window with the finite number of SRAMs is generated and propagated horizontally. A pair of stereo images are acquired consecutively from a stereo camera and stored in the SRAMs in rows and processed in columns for window-based matching. As a result, during each cycle, this architecture can concurrently perform multiple reads and a single write operation. Thus, we can handle the requirements of wide bandwidth and high-speed data accessing.

As shown in Fig. 3, the size of memory is primarily determined by the width of the image and the height of the window—the former equals the depth of each SRAM and the latter the number of SRAMs. Therefore, as the width of the image increases, the size of the SRAM increases, and as the size of the window increases, the number of SRAMs also increases. However, a large number of interconnections between the SRAMs and logic cells will cause performance degradation resulting from the high congestion and longer wire length in 2D ICs. Thus, by comparing 2D and 3D ICs, we can show the impact of reduced wire length by vertical stacking on the performance of the stereo matching processor. In this paper, a 8-bit, gray-level 752 x 480 image and a 15 x 15 window is used for the stereo matching and an 11 x 11 window is used for the disparity diffusion. Fig. 4 shows the fully



(a) Multiple reads and a single write operation of SRAMs (W_N : window size)

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{max}	
SRAM 1	W	R	R	R	W	R	R	R	W	R	R	
SRAM 2	R	W	R	R	R	W	R	R	R	W	R	W
SRAM 3	R	R	W	R	R	R	W	R	R	R	R	R
SRAM W_N	R	R	R	W	R	R	R	W	R	R	R	W
SRAM W_N+1	R	R	R	R	W	R	R	R	W	R	R	R

W : Write
R : Read

(b) Modular operation of SRAMs (R_x : row address of image)

Fig. 3. SRAM operation for our window-based matching algorithm.

TABLE I. FEATURES OF OUR STEREO MATCHING PROCESSOR.

Max. frequency	312MHz
Image size (pixel)	752 x 480
Disparity range (pixel)	64
Max. frame rate	108 frames/s @ 312MHz
Max. bandwidth	12.8GB/s @ 312MHz

pipelined hardware architecture of the stereo matching processor, and Table I summarizes the features of the stereo matching processor.

III. DESIGN AND ANALYSIS FLOW

A. Design Flow

Fig. 5 shows the overall design flow of 2D and 3D ICs. From the given register-transfer level (RTL) description of the stereo matching processor written in Verilog-HDL, we use a conventional design flow for the 2D IC design. We use Synopsys Design Compiler to generate a top-level synthesized netlist. For a fair comparison between 2D and 3D ICs, we use the same synthesized netlist for 2D and 3D ICs. For the 2D IC layout, we perform floorplanning, placement, clock tree synthesis, routing, and timing optimization using Cadence Encounter.

For the primary step of the partitioning of 3D ICs, we divided the functional modules and memory macros in the top-level netlist, assigned them to the tiers, and determined the number of TSVs that interconnect each tier. Thus, it is in this step that the overall characteristic and performance of 3D ICs are determined. For the two types of 3D IC designs, we partitioned the top-level synthesized netlist in both macro- and pipeline-level styles using *group* and *ungroup* commands in Synopsys Design Compiler. Then, we extracted the partitioned netlist for each tier and inserted the TSVs into the netlist. We manually placed the TSVs prior to gate placement and did the layout separately for each tier in the same way as we performed the conventional 2D IC layout.

B. Macro-Level Partitioning

We divided the top-level netlist into logic cells and memory macros, shown in Fig. 6 (a). As the gates are placed vertically over the memory macros, this partitioning method minimizes wire lengths

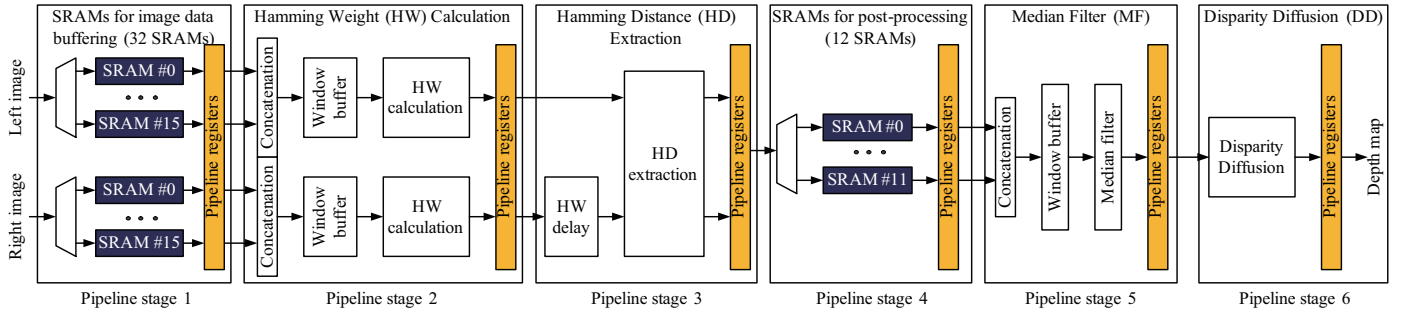


Fig. 4. Pipeline architecture of our stereo matching processor.

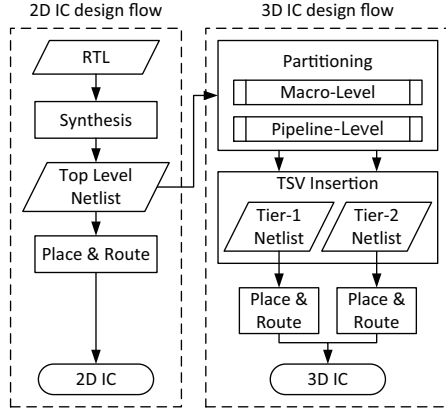


Fig. 5. Overall RTL-to-GDSII design flow for 2D and 3D IC designs.

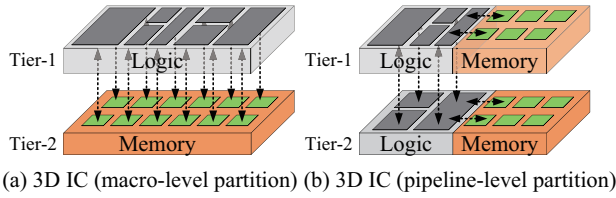


Fig. 6. Two types of partitioning methods. (a) conventional macro-level partitioning method, (b) proposed pipeline-level partitioning method.

between the logic and memory macros, thus maximizing the benefits of 3D ICs. However, the total number of TSVs and the die size are proportional to the number the macros. Therefore, as the number of macros increases, the total number of TSVs and the die size also increase. In this case, if the total area of the macros is not proportional to the total area of the logic cells, maintaining a balance of the die sizes of the tiers is difficult. Moreover, the increased number of TSVs will cause silicon area overhead and increase routing congestion.

C. Pipeline-Level Partitioning

The main idea of this partitioning method is to minimize the total number of TSVs and to balance the die sizes of the tiers with minimal effort in a pipelined hardware architecture. In this partitioning method, with the basic concept of pipelining (memory is dedicated to its own pipeline stage, shown in Fig. 4), we simply split the pipeline stages into two groups. In this case, the total number of TSVs is determined by the number of signals between the two groups. However, the balance of the die sizes of the tiers is not guaranteed by simply dividing the pipeline stages into two groups. We balanced them by

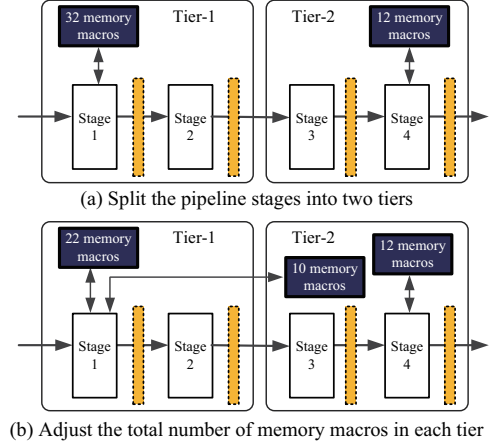


Fig. 7. An example of the pipeline-level partitioning method.

adjusting the total number of memory macros. In this case, the total number of TSVs increased as the number of adjusted memory macros were increased. Fig. 7 illustrates the two steps of the pipeline-level partitioning method. In this example, we assume that the silicon areas of the stages and memory macros are identical.

D. Timing and Power Analysis Flow

We conducted a static timing analysis (STA) using Synopsys PrimeTime with the layout netlist and a RC parasitic file (.spef) that contains resistance and capacitance values for all the nets. Then, if the timing was met, we performed a power analysis using PrimeTime. Although existing commercial tools can perform STA and power analysis for 2D IC, they cannot do so for 3D ICs. Fig. 8 shows the timing and power analysis flow for 3D ICs. For the 3D STA, we created a top module netlist that combined the netlist of each tier of 3D ICs and a top-level parasitic file that contained the RC parasitic of TSVs. Then, with the layout netlists and RC parasitic files, we performed 3D STA using PrimeTime. After that, if the timing was not met, we extracted the boundary constraints of each tier using PrimeTime and then, using these boundary constraints, we iterated timing optimization during the layout. If the timing was met, we conducted a power analysis using PrimeTime with the layout netlists and the RC parasitic files.

IV. EXPERIMENTAL RESULTS

We implemented our 2D and 3D IC designs using Global-Foundries 130-nm PDK and 44 single-port SRAM macros generated

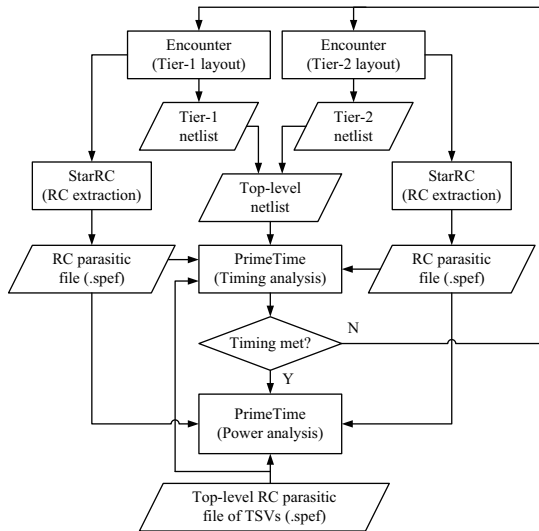


Fig. 8. Sign-off timing and power analysis flow for 3D ICs.

TABLE II. SUMMARY OF SYNTHESIS RESULTS.

Technology	GlobalFoundries 1.5V 130-nm
Total cell area (μm^2)	2,977,542
Target clock period (ns)	3.2
Slack (ns)	0.0
# of memory macros	44
Total memory capacity	44 x 752 bytes = 31.3KB

with an ARM memory compiler. We chose this commercial technology setting because of a recent successful 3D IC development published in [5]. For a fair comparison between 2D and 3D ICs, we use the same synthesis output and target clock period during the layout and timing optimization for the entire design. Tables II and III summarize the synthesis results and memory macros, respectively. We bonded Tiers 1 and 2 in a face-to-back style and connected them using a $2.2\text{-}\mu\text{m}$ via-first TSVs for 3D integration, shown in Fig. 9. In addition, since the capacitance and resistance of TSVs are not negligible in the timing and power analysis, we used 10fF for the TSV capacitance and $50\text{m}\Omega$ for the resistance of TSV during the timing and power analysis. The clock tree synthesis for 3D IC design is difficult because no commercial EDA tools can fully handle clock trees for 3D ICs. Thus, we treated each tier as if it had its own clock tree network and then performed clock tree synthesis separately. Then we directly connected the clock source of Tier-1 to Tier-2 through a TSV.

A. Partitioning Style Comparisons

For the 3D IC designs, we used both the conventional macro-level and the proposed pipeline-level partitioning methods. The 3D IC with the macro-level partitioning method (3D-MP), which consists of a logic tier (Tier-1) and a memory macro tier (Tier-2), minimizes wire lengths between the logic and memory macros, shown in Fig. 6 (a). For the 3D-MP design, we placed a total of 425 signal TSVs uniformly on Tier-1 according to the location of the I/Os of each memory macro.

For the 3D IC design with the pipeline-level partitioning method (3D-PP), we split pipeline stages to minimize the total number of TSVs. Then, we balanced the total cell area of each tier by adjusting the total number of memory macros of each tier. We assigned the

TABLE III. SUMMARY OF MEMORY MACROS.

Type of memory	Single-port SRAM
Memory macro dimension (μm)	162.4 x 248.9
# control pins	3
# address pins	10
# input data pins	8
# output data pins	8
Total # pins	29
Capacity (bits)	752 x 8 = 6,016

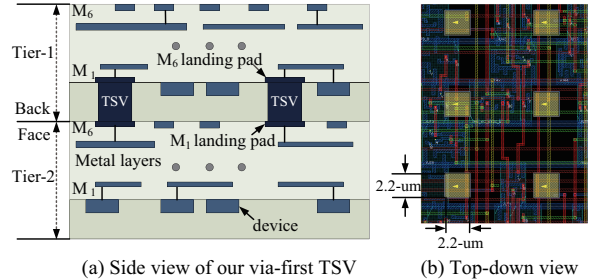


Fig. 9. TSV and bonding technologies used in this paper.

first three pipeline stages (1, 2, and 3) of Fig. 4 to Tier-1, and the remaining pipeline stages (4, 5, and 6) to Tier-2. In this case, before adjusting the total number of memory macros of each tier, we assigned 67.5% and 32.5% out of the total cell area to Tier-1 and Tier-2, respectively. Thus, with the same footprint size, Tier-1 will suffer more from routing congestion. From the cell area report, shown in Table IV, we learned that each SRAM macro occupies 1.4% of the total cell area. To balance the total cell area of each tier, we moved 12 SRAM macros from Tier-1 to Tier-2. For the 3D-PP design, we placed a total of 221 signal TSVs in the center area of Tier-1. Since the goal of this study is not to find an optimal location of TSVs for 3D ICs, we placed TSVs manually in the center area chip for the 3D-PP design mainly because most TSVs come from logic cells on Tier-1 and connect to memory macros on Tier-2. Table V shows the type and number of TSVs of 3D ICs.

B. Overall Layout Comparisons

Figs. 10 and 12 show the design quality comparisons and the layout snapshots of 2D and 3D ICs, respectively. Table VI summarizes the results of the comparison of the layout. First of all, we observed that the chip footprint of the 3D ICs was 43% smaller than that of the 2D IC. In addition, the total wire lengths of the 3D-MP and the 3D-PP were 14% and 4%, respectively, shorter than those of the 2D IC by taking advantage of vertical stacking and the smaller footprint area. We also observed that the total number of buffers of the 3D ICs was nearly 20% lower than that of the 2D IC mainly because of shortened wire lengths. As a result, 3D ICs consumed less power than the 2D IC. In fact, the 3D-MPs and 3D-PPs consumed 13% and 7% less power, respectively, than the 2D IC.

However, in the case of the 3D-PP, although the total wire length of the clock tree was 7% shorter than that of the 2D IC, their total number of clock tree buffers was 4% higher than that of the 2D IC. One explanation for this finding is that we performed clock tree synthesis separately and then directly connected the clock source of Tier-2 to that of Tier-1 through a TSV. In this case, existing commercial EDA tools perform the clock tree synthesis without any awareness of the other clock tree. Thus, the clock tree of the 3D IC could not be optimized well. However, the 3D-MP does not have a

TABLE IV. TOTAL AREA FOR EACH PIPELINE STAGE.

Pipeline stage number	Pipeline stage name	Cell area (μm^2)	Percentage (%)
1	32 SRAM macros	1,294,065	43.5
2	Hamming weight calculation	255,783	8.6
3	Hamming distance extraction	459,072	15.4
4	12 SRAM macros	485,274	16.3
5	Median filter	241,725	8.1
6	Disparity diffusion	241,623	8.1
Total		2,977,542	100.0

TABLE V. TSV USAGE IN OUR 3D DESIGNS.

3D IC design type	Type of signal	# TSVs
3D-MP	SRAM control signals	29
	SRAM address signals	20
	SRAM data signals	376
	Total # TSVs	425
3D-PP	Logic signals between pipeline stages	56
	SRAM control signals	41
	SRAM address signals	20
	SRAM data signals	104
	Total # TSVs	221

large clock tree network on Tier-2, so it suffers less from the clock tree optimization problem.

When the 3D-MP is compared with the 3D-PP, the 3D-PP uses a smaller number of TSVs, but the 3D-MP outperforms the 3D-PP, particularly in terms of total wire length and total number of clock tree buffers. This finding could result from several factors: The location of the TSVs of the 3D-PP may not be an optimal location, and the 3D-PP may require more buffers than the 3D-MP because its clock tree network is less optimized. From these observations, we learned that an optimal location of TSVs and the 3D clock tree synthesis play important roles in 3D IC design, which fully exploits the benefits of vertical stacking.

C. Detailed Power Analysis

Table VII and Fig. 11 show the detailed power comparisons between 2D and 3D ICs. We observed that memory macros consumed over 35% of total power in all of the designs because of the memory-intensive nature of the stereo matching processor. In addition, even though the total wire length and the total number of clock tree buffers occupy around 4% of the total wire length and 5% of total buffers, the clock network consumed over 30% of the power because of the high switching activity of the clock tree network and the larger number of registers for pipelining. As shown in Fig. 11, most of the power savings are achieved in the combinational power, indicating that the reduced number of buffers in the interconnect plays an important role in reducing the power consumption of 3D IC designs.

D. Discussion

This study shows that we can achieve significant power reduction with our 3D IC-based stereo matching processors. Our studies indicate that the major sources of power reduction are a smaller number of buffers and shortened wires that stem from vertical stacking and a reduced footprint area. We also learned that the proposed pipeline-level partitioning method minimizes the total number of TSVs while balancing the footprint area of each tier. However, we also observed that the reduction in the total number of signal TSVs does not always lead to an optimal design with regard to total wire length and power reduction. To fully exploit the benefits of vertical stacking, designers must determine optimal locations for TSVs in physical layouts.

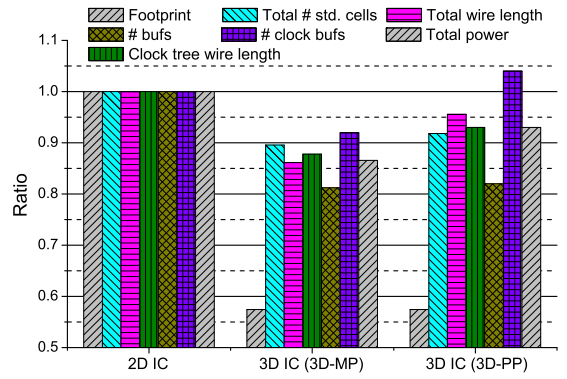


Fig. 10. Design quality comparisons between 2D IC and 3D ICs.

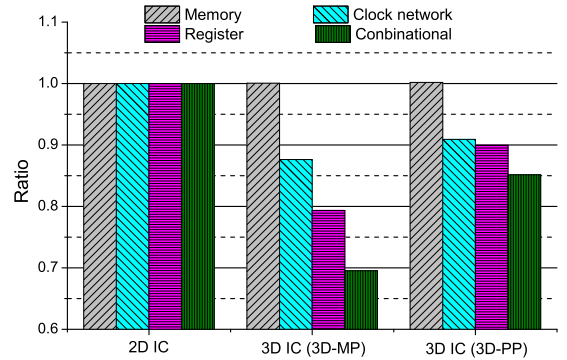


Fig. 11. Detailed power comparisons between 2D IC and 3D ICs.

V. CONCLUSION

We presented comprehensive analysis results of 3D IC-based low-power stereo matching processors. We used two types of partitioning methods to fully exploit the benefits of the vertical stacking technology, namely, macro- and pipeline-level partitioning. We observed that our 3D IC-based stereo matching processors enabled low power consumption mainly due to a reduction in buffer usage and shortened wire lengths resulting from a reduced footprint area. We achieved a total power reduction of 13% and 7% with our macro- and pipeline-level partitioning methods, respectively. Our 3D IC study was only done in 130-nm technology. However, more advanced technology nodes provide more wire layers for routing. Thus, there are less routing congestion problems in the advanced technology nodes. However, overall performance of 3D IC designs is more influenced by the constraints of the design and the proportion of the memory power consumption than the total number of routing layers. Our future work includes the comparison of benefits of 3D stacking between different types of technology nodes.

ACKNOWLEDGMENT

This investigation was financially supported by Semiconductor Industry Collaborative Project between Kyungpook National University and Samsung Electronics Co. Ltd. This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) support program (NIPA-2013-H0401-13-1005) supervised by the NIPA (National IT Industry Promotion Agency).

TABLE VI. OVERALL LAYOUT COMPARISONS BETWEEN 3D-MP (MACRO-LEVEL PARTITIONING), 3D-PP (PIPELINE-LEVEL PARTITIONING), AND 2D.

	3D IC (3D-MP)			3D IC (3D-PP)			2D IC
	Tier-1	Tier-2	Total	Tier-1	Tier-2	Total	
Target clock period (n_s)	3.2			3.2			3.2
Footprint (μm)	2350 x 1350	2350 x 1350	2350 x 1350	2350 x 1350	2350 x 1350	2350 x 1350	2350 x 2350
Total wire length (μm)	4,507,365	220,485	4,727,850	2,662,651	2,582,875	5,245,526	5,488,514
Clock net wire length (μm)	184,683	18,293	202,977	119,704	95,636	215,340	231,232
Longest path delay (n_s)	3.19			3.12			3.05
# standard cells	87,245	205	87,450	50,113	39,528	89,641	97,630
# buffers	15,242	161	15,403	8,575	7,051	15,626	18,968
# clock buffers	740	61	801	508	399	907	871
Total power (mW)	871.10			931.40			1006.20

TABLE VII. DETAILED POWER COMPARISONS BETWEEN 3D-MP (MACRO-LEVEL PARTITIONING), 3D-PP (PIPELINE-LEVEL PARTITIONING), AND 2D.

Design Type	Power Group	Internal Power (mW)	Switching Power (mW)	Leakage Power (mW)	Total Power (mW)	Percentage (%)
3D IC (3D-MP)	Memory	353.80	1.46	0.08	355.34	40.79
	Clock network	209.70	68.20	0.00	277.90	31.90
	Register	31.80	12.40	0.00	44.20	5.07
	Combinational	89.80	103.80	0.00	193.60	22.23
	Total	685.10	185.86	0.08	871.04	100.00
3D IC (3D-PP)	Memory	353.70	2.02	0.08	355.80	38.20
	Clock network	219.50	68.80	0.00	288.30	30.96
	Register	36.40	13.70	0.00	50.10	5.38
	Combinational	113.80	123.30	0.00	237.10	25.46
	Total	723.40	207.82	0.08	931.30	100.00
2D IC	Memory	353.70	1.33	0.08	355.11	35.29
	Clock network	246.10	71.00	0.00	317.10	31.51
	Register	41.00	14.70	0.00	55.70	5.54
	Combinational	135.80	142.60	0.00	278.40	27.67
	Total	776.60	229.63	0.08	1006.31	100.00

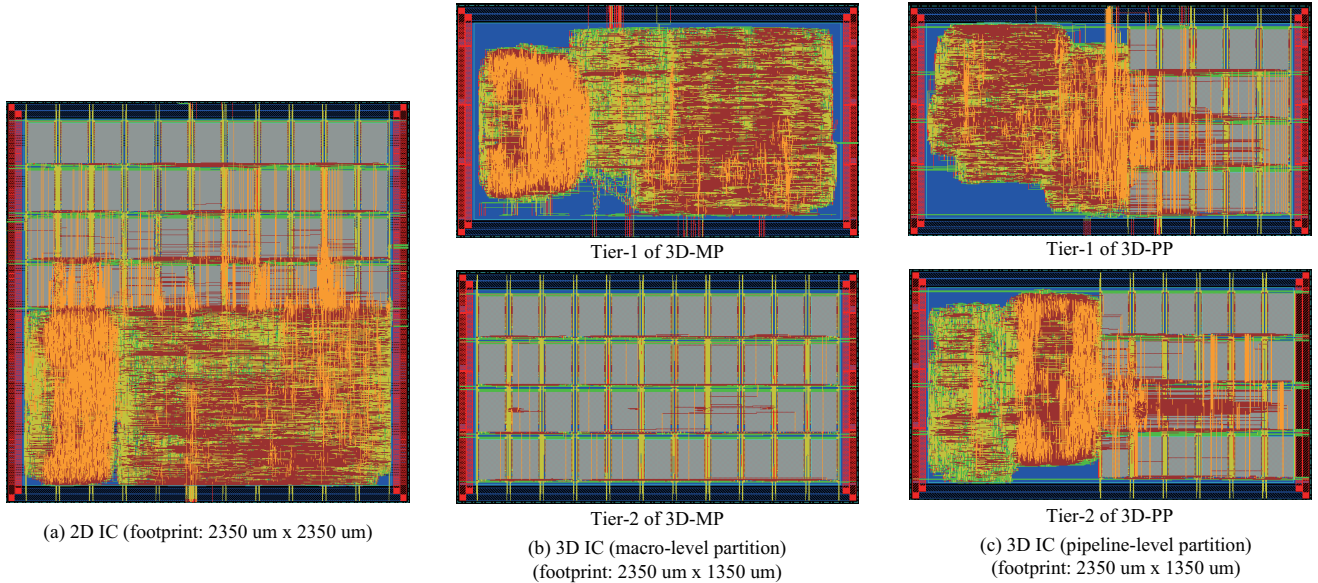


Fig. 12. Layout snapshots: (a) 2D IC, (b) 3D-MP (macro-level partitioning), (c) 3D-PP (pipeline-level partitioning).

REFERENCES

- [1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.
- [2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.
- [3] T. Thorolfsson, K. Gonsalves, and P. Franzone, "Design automation for a 3dic fft processor for synthetic aperture radar: A case study," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, July, pp. 51–56.
- [4] G. Neela and J. Draper, "Challenges in 3dic implementation of a design using current cad tools," in *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, 2012, pp. 478–481.
- [5] D. H. Kim, *et al.*, "3D-MAPS: 3D Massively parallel processor with stacked memory," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, pp. 188–190.
- [6] T. Zhang, *et al.*, "A 3d soc design for h.264 application with on-chip dram stacking," in *3D Systems Integration Conference (3DIC), 2010 IEEE International*, Nov., pp. 1–6.
- [7] H. Saito, *et al.*, "A chip-stacked memory for on-chip sram-rich socs and processors," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 15–22, 2010.
- [8] R. Zabih and J. W. Li, "Non-parametric local transforms for computing visual correspondence," in *Proc. of the European Conference on Computer Vision*. Springer-Verlag, 1994, pp. 151–158.