Heterogeneous Stacking of 3D MPSoC Architecture: Physical Implementation Analysis and Performance Evaluation

Mohamad Hairol Jabbar[1,2,3], Dominique Houzet[2], Omar Hammami[3]
[1]Department of Computer Engineering, FKEE, UTHM, Johor, Malaysia
[2]GIPSA-Lab, Saint Martin D'Heres, France
[3]ENSTA Paristech, France
E-mail: [1]hairol@uthm.edu.my, [2]dominique.houzet@gipsa-lab.grenoble-inp.fr,
[3]omar.hammami@ensta-paristech.fr

Abstract

3D integration is one of the feasible technologies for producing advanced computing architecture to support ever-increasing demand of higher performance computing especially in mobile devices. The emerging trend of multiprocessor architecture has made Network on Chip (NoC) architecture the best solution for future manycore architecture devices. In this work, we explore the implementation of heterogeneous 3D Multiprocessor System on Chip (MPSoC) stacking architecture and evaluate its performance in terms of timing and power consumption compared with its 2D counterpart. The proposed heterogeneous 3D MPSoC implementation approach is considered to be the best solution for the time being as there are no 3D-aware EDA tools available in the markets that capable of performing 3D optimization as in 2D EDA tools. We also perform physical implementation analysis on the clock tree structure between 2D and 3D architecture and examine the impact of using 2D EDA tools for designing 3D architecture. The implementation is based on industry-specific Tezzaron 3D IC technology and the evaluation is based on the GDSII results from physical design implementations.

Keywords

3D IC, Heterogeneous stacking, MPSoC, NoC, Physical design

1. Introduction

Future technology will have many processing cores to perform highly complex and computational intensive applications. NoC-based Multiprocessor System on Chip (MPSoC) architecture is the solution for this ever-growing demand of higher performance devices due. NoC architecture is the backbone of future multiprocessor communication architecture due to its advantages such as scalability and flexibility as opposed to bus-based architectures and point-to-point links.

In this work, we perform implementation analysis for heterogeneous 3D MPSoC stacking architecture compared with its 2D counterpart to be able to evaluate the performance benefits of 3D technology for MPSoC design. Using layout level netlist, we examine the performance in terms of timing slack and power consumption and provide detailed implementation analysis including clock tree structure and impact of using 2D EDA tools for 3D IC design.

The contributions of this work can be summarized as follows:

1. Analysis on the implementation of heterogeneous 3D MPSoC stacking architecture and compared its timing and power characteristic with 2D MPSoC architecture.
2. Perform detailed physical implementation analysis of 2D vs 3D MPSoC stacking architecture to better understand the implementation issues for 3D MPSoC architecture under the limitations of using 2D EDA tools.

This paper is organized as follows. Section 2 reviews some of the previous works on the heterogeneous 3D stacking to justify the novelty in our work. Section 3 describes the Tezzaron 3D IC technology used in this work followed by the explanation of the baseline 2D MPSoC architecture in section 4. Section 5 presents the heterogeneous 3D MPSoC stacking architectures including the partitioning method. Section 6 presents experimental results for different performance metrics comparing 2D and 3D MPSoC together with detailed physical implementation analysis and finally we conclude the work with directions for future works.

2. Related works

3D heterogeneous architectures have been studied by several researchers but mostly restricted to analysis from software simulation. The most common approach to implement heterogeneous 3D stacking is using memory on logic stacking primarily to achieve higher memory bandwidth due to advantage of huge amount of vertical interconnections. In [1], they have designed and implemented memory on logic architecture for the 64 multicore processors where each data memory for each core is place on another layer on top of its logic layer. The instruction memory is placed on the logic layer in order to have maximum size for data memory for each core. To achieve maximum memory bandwidth, the processor core is designed specifically to consume memory bandwidth at every cycle from the 3D stacked memory by allocating one slot for the memory instruction. However, they do not use NoC architecture for the communication architecture due to the stable, predictable and regular communication pattern in their data-parallel applications.

Instead, they use buffer-based architecture to allow processors communicate between its neighboring blocks. In [2], heterogeneous memory-on-memory architecture is studied by stacking SRAM cache with logic on the 3D DRAM layer with the aim to optimize both performance and energy efficiency. By folding the DRAM bank layers into four layers and then share the same TSVs bus to the logic layers, it reduces the energy from transferring entire row signals. Another work on heterogeneous stacking is done by [3] where they stacked heterogeneous DRAM layers on processor layers. Performance analysis is done using software simulation based on modified CACTI and M5 simulators for full system simulation with multicore processor.

With regards to 3D architecture using NoC, we found limited number of works about heterogeneous stacking based on NoC architecture especially the one implementing physical design. In [4], 3D architecture using combination of heterogeneous IP cores layer and homogeneous mesh NoC layer is studied and performance analysis is done using cycle accurate simulation. The main reason behind their work is that heterogeneous multicore architecture does not have the same IP core and thus the different size between each IP core makes it not suitable to use Mesh NoC where it is normally based on homogeneous multicore architecture with same IP core size. In order to use mesh NoC with the heterogeneous IP core architecture because of regular properties of mesh topology, 3D architecture can be used to realize it by stacking both different layers on top of each other. Another work in [5], they presented a three tiers heterogeneous architecture by using a VesFET-transistor based NoC architecture in the middle layer between core and cache layers in order to reduce the router to router wire links compared with 2D and normal 3D implementation. Their analysis based on HSPICE simulation shows power and latency improvement basically because of router to router distance reduction.

State of the art electronic design usually facilitates globally asynchronous locally synchronous (GALS) architecture to be able to meet design specifications especially for tight power requirements. Power consumption can be reduced up to two times lower for the same architecture using fully synchronous implementation at smaller area overhead using fine-grained clock domain partitioning [6]. Multiprocessor implementation with NoC architecture is nicely fitted with the GALS style where communication architecture can be separated from the computation architecture with different clock speeds hence enabling high performance system with power efficiency [7]. To the best of our knowledge, there is no work investigating the implementation of GALS style 3D multiprocessor architecture to date wherein the main motivation of this study. Deploying GALS architecture in 3D IC technology is also very exciting due to the fact that it gives more design space to be explored with the existence of the vertical architecture in meeting various target implementation requirements.

In this work, we based upon the work in [4] to further investigate the performance of heterogeneous stacking for NoC-based multiprocessor architecture with slight modification to be more realistic implementation considering the router and processor area from the fabricated designs. In particular, a part of the processor component is placed in the same layer with the NoC architecture to cover the empty area due to the smaller NoC area than the processor. Using Tezzaron two-tier technology, we carried out physical design implementation of the heterogeneous 3D stacking MPSoC architecture and compare its performance with the 2D architecture from architectural point of view. This study provides additional architectural exploration for the previously done homogeneous stacking of 3D NoC architectures as well as architectural exploration of the GALS style implementation in 3D architecture. Deep understanding about how performance is affected by different 3D architecture implementations is essential to find the right architectural candidate to fully benefit from the 3D technology.

3. 3D technology

This 3D integration technology is based on Tezzaron [8] that uses TSV for peripheral IOs. The two-tier 3D stacking method is based on wafer-to-wafer bonding, face-to-face method with via-first approach as illustrated in Figure 1 [9]. Inter-die connection is achieved through microbumps structure where it provides high interconnection density up to 40,000 microbumps per mm^2 without interfering to FEOL (front-end-of-line) device or routing layers. Furthermore, as its physical structure is small enough that the delay can be negligible, 3D verification methodology at every stage of physical design flow can be performed to estimate the design performance at early stage of the design and then do modification according to the specifications. We are also able to implement four tiers design by stacking two face-to-face through back-to-back stacking using TSV in order to have higher design complexity.

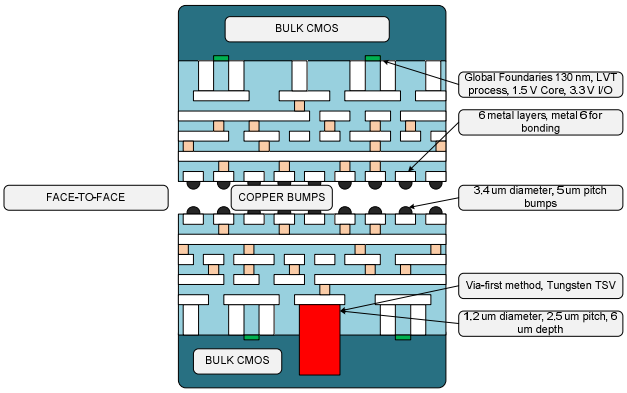

Figure 1: Cross section Tezzaron 3D IC technology

4. Baseline 2D NoC-Based MPSoC Architecture

In this section, we explain the baseline NoC as well as the processor architecture to be used for the 2D and 3D MPSoC implementation analysis.

4.1. Processor architecture

We use an open source processor for our implementation which is readily available without spending much time to develop a new processor. The Openfire processor as shown in Figure 2 is downloaded from Opencores.org. It is a Microblaze clone which is based 32-bit Reduced Instruction Set Computing (RISC) architecture using Harvard architecture that supports Microblaze instruction set architecture (ISA) and compiler tool chain [10]. Comparing with MicroBlaze processor that has hardware multiplier, hardware divider, barrel shifter and floating point unit, Openfire processor has only hardware multiplier and also supports On-chip Processor Bus (OPB) for external interface particularly for accessing instruction and data memory. Although there are other open source synthesizable Microblaze clones available to be used [11], we choose Openfire because it has Fast Simplex Links (FSL) ports (basically a FIFO that support dual clock domains) that we need for simple data and synchronization communication between processors and NoC rather than using more complex in terface such as Open Core Protocol (OCP) and Advanced eXtensible Interface (AXI) which require complex logic for implementation. It supports up to 16 FSL ports as in MicroBlaze allowing us to integrate additional functions such as NoC monitoring service using simple interface to the processor.

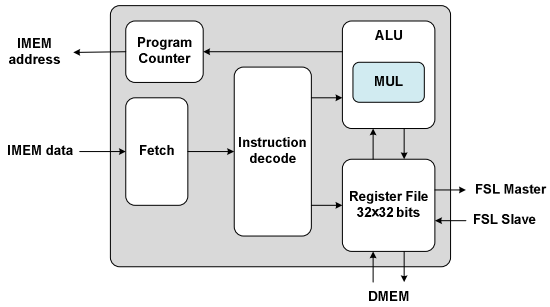

Figure 2: Openfire processor block diagram

The Openfire processor is a simple processor developed initially for configurable processor research [12] but have been used for other purpose [13]. Thus, because of its simplicity, it will not require a large silicon area and thus can be used to develop any small application for testing the NoC in 3D architecture. Additionally, we use only 4KB for instruction and 4KB for data memory in order to limit the die area. These memories are generated using Artisan memory compiler. The processor has 32-words register file implementing using flip-flop registers which consuming most of the processor's logic area.

4.2 NoC architecture

The NoC architecture in this experiment is based on 2D Mesh topology implemented using router and network interface architecture based on our previous paper [14]. The 2D router has four neighboring ports to each side of the router and one local port to the network interface for the processor connection. We extended the 3D architecture implementation in this paper by including processor architecture which allows us to investigate heterogeneous 3D architecture of complete MPSoC design because there exist both memory and logic structure. Figure 3 shows the interconnection structure between processor, network interface unit (NIU) and 2D router for a complete tile block.

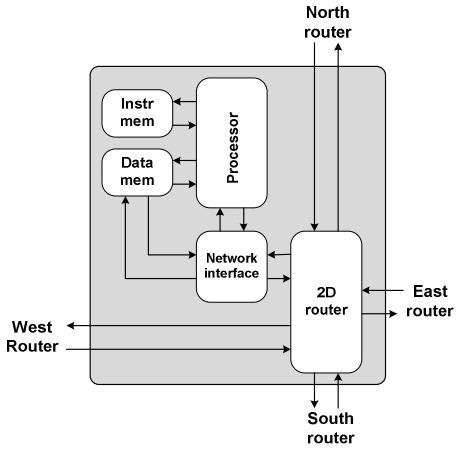

Figure 3: Interconnection structure for a tile block

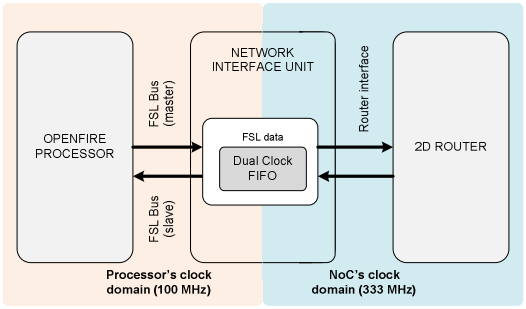

Figure 4: GALS implementation using a dual-clock FIFO

4.3. GALS Implementation

The GALS architecture is appealing from the power perspective where power reduction can be achieved due to the clock gating implementation whereas from performance perspective, it does not directly offers improvement which is depending on the implementation-specific techniques. A number of methods exists for interfacing different clock do mains in the GALS architecture such as plausible clocking, FIFO-based and boundary synchronization as explained in details in [15]. One of the primary concerns of the GALS implementation is the data synchronization between different clock domains. Although FIFO-based GALS style suffers from the additional latency of the FIFO block, careful design and using large FIFO buffers can inherently hide much of the performance penalty [16] at the expense of more area overhead.

The GALS style implementation in this architecture is depicted in Figure 4 which is based on a dual clock FIFO structure for handling clock domain crossing. We use a four-word depth for the FIFO block built-in within a network interface for transferring data from the processor through its FSL master and slave bus operating at 100 MHz to the NoC operating at 333 MHz. For processor to NoC communication, data from FSL bus is first written to the dual clock FIFO before being packetized to be sent to the router for transportation. In contrast, for NoC to processor

communication, the packets arrive from router is first de-packetized before being written to the dual clock FIFO.

4.4. Baseline 2D MPSoC architecture

The 2D NoC-based multiprocessor architecture is shown in Figure 5 as a baseline design for comparison purposes with the heterogeneous 3D MPSoC stacking architecture. The synthesized area using 130 nm technology for each component is shown in Table 1indicating that the tile area is dominated by the memory macros which is about 56% of the total tile area. We have implemented 16 processors with 4KB data memory (dual port) and 4KB instruction memory (single port) for each processor and using 2D Mesh NoC for the inter-processor communication based on the router and network interface explained in [14] which consumes about 24% silicon area using all metal layers available (up to metal 6).

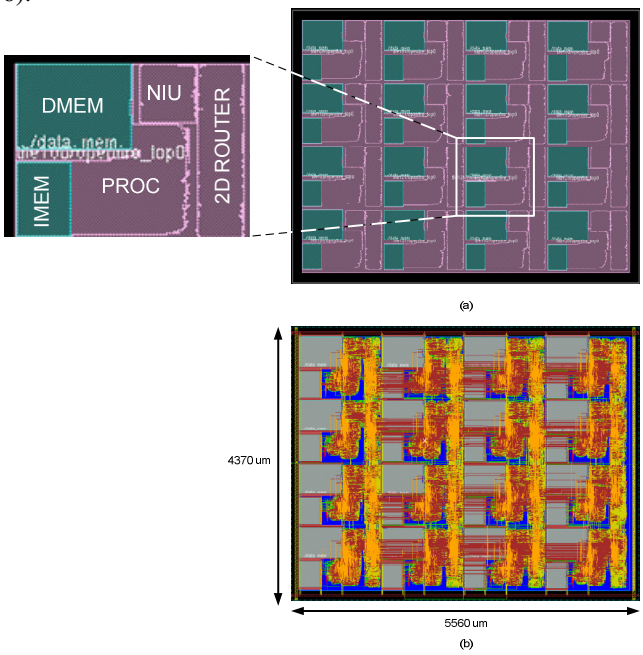

Figure 5: Baseline 2D MPSoC architecture (a) amoeba view (b) routed layout

5. Heterogeneous Stacking of 3D NoC-based MPSoC architecture

In this section, we will discuss the architecture and its physical implementation of heterogeneous 3D MPSoC architecture.

5.1. MPSoC Partitioning Technique

For the heterogeneous stacking, we divided the 2D design into a tile of processor and another tile for NoC architecture as shown in Figure 6. The floorplan and routed layout is shown Figure 7and Figure 8 for bottom and top tier respectively. The processor with its data memory is placed in the bottom tier while the NoC with the instruction memory is placed in the top tier. The vertical connection is made of signals from network interface in the NoC to the processor and to the data memory and also from the processor to the instruction memory. Therefore, first we set the location of the microbumps in the bottom tier around processors and data memory, then we floorplan the top tier for the NoC architecture by placing the network interface under the microbumps locations created from the bottom tier to be as close as possible. Stacking method proposed in [4] is not realistic because real routers have relatively small area compared with the processor or any other IP cores as fabricated in [17] and [18] which will creates large empty silicon area and therefore we decide to modify the floorplan by moving the instruction memory block to the top tier to be placed with the NoC architecture.

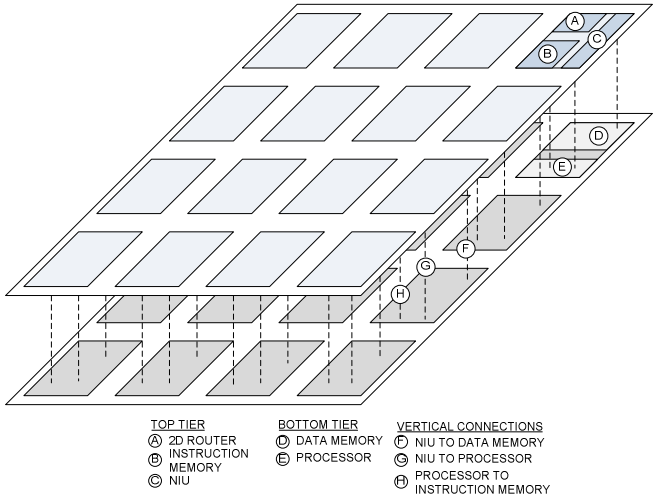

Figure 6: Partitioning for heterogeneous 3D MPSOC architecture

Table 1: Synthesize area for each block in a tile

Components	Area (um^2)	Percentage (%)
Openfire CPU	161.04	18
Instruction memory (4 KB)	156.44	17
Data memory (4KB)	352.55	39
NIU	63.10	7
2D router	151.07	17
Total area per tile	884.19	100

One of the novel features in this study is that we employ GALS in the 3D architecture wherein the NoC and the processor operate in different clock domains since the processor is quite slow compared with the speed of NoC. To the best of our knowledge, this work is the first to conduct physical design implementation analysis of 3D GALS for multiprocessor architecture. The GALS clocking style avoids global clock tree structure which essentially reduces power consumption since clock tree has prominent portion of the total power consumption of a system. A part from that, this implementation style also enables Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS) [19] methods for balancing power consumption and performance at real time and also allows efficient thermal management specifically for 3D architecture having higher temperature effect. Based on the

GALS architecture, each tier can be run at different frequencies where the NoC at the top layer is clocked at 3 ns while the processor at the bottom layer is clocked at 10 ns period. This type of floorplan provides easier thermal management technique by placing the hot layer clocked at higher frequency close to the heat sink enabling fast thermal transfer [20]. From the testing point of view, this floorplan also allows easier method for 3D architecture pre-bond testing of the NoC as well as processor architecture since they are located in separate layer.

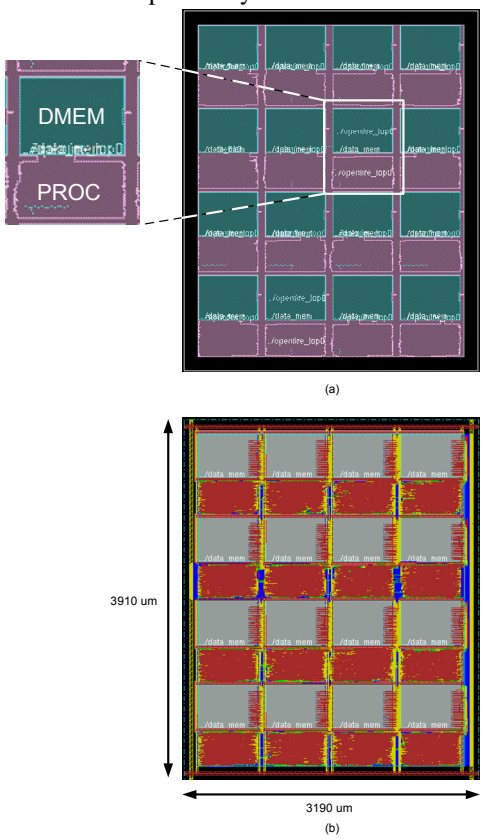

Figure 7: Bottom tier of heterogeneous 3D MPSoC architecture (a) amoeba view (b) routed layout

6. Experimental Results

It can be seen from Table 2 that there is almost 50% reduction of core area for heterogeneous 3D stacking compared with the 2D architecture due to the partitioning of NoC architecture and instruction memory into another layer. The number of gates however is slightly increased over 2D architecture mainly because of separate optimization flow of both tiers during place and route step. Out of 188 vertical connections per tile (NIU to/from processor and data memeory), 70 connections are for the processor FSL connections whereas the rest of vertical connections are for the data and instruction memory connections. We can also see a slight increase of total wirelength in heterogeneous 3D stacking compared with the 2D architecture due to separate 2D optimization process during place and route step. As shown in Table II, the speed of the NoC is improved in 3D architecture.

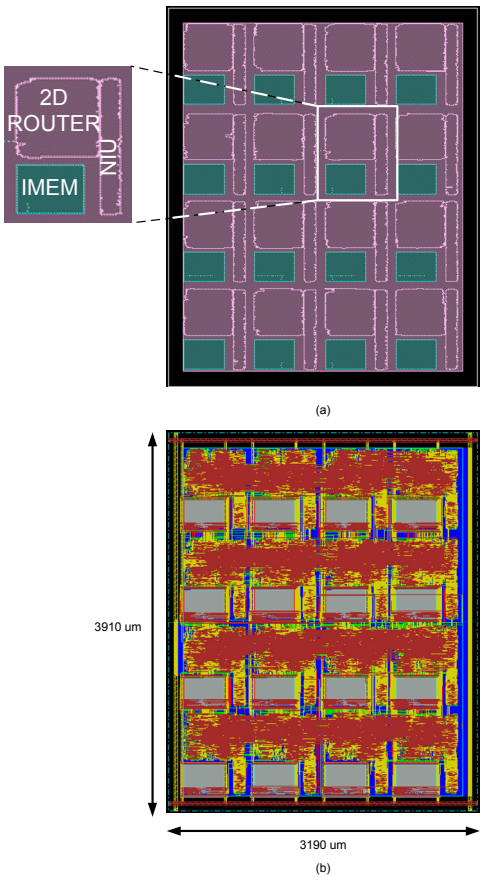

Figure 8: Top tier of heterogeneous 3D MPSoC architecture (a) amoeba view (b) routed layout

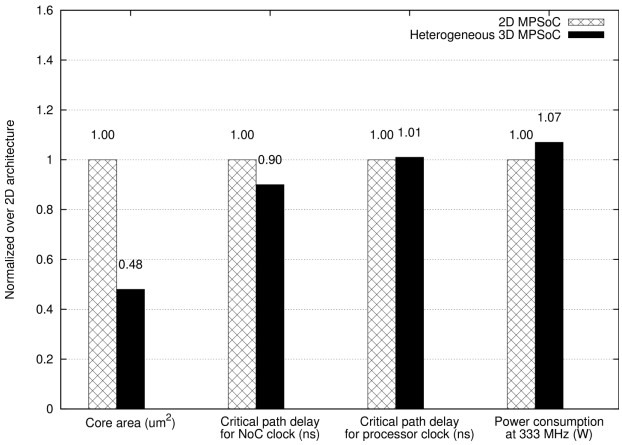

Figure 9: Performance comparison for 2D and heterogeneous 3D MPSoC architecture

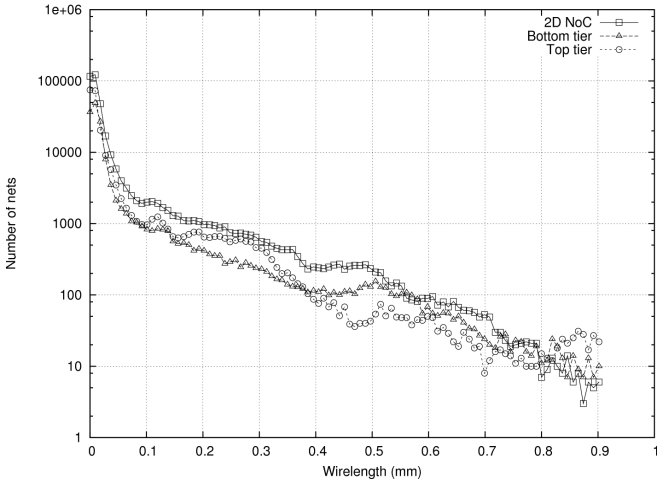

Figure 10: Horizontal wirelength distribution for 2D MPSoC and heterogeneous 3D MPSoC (bottom and top tier)

The performance comparisons between 2D and 3D design are shown in Table 2 and Figure 9 where it clearly shows that heterogeneous 3D MPSoC stacking improves slightly in the NoC speed. Performance in the NoC speed is partially increased because of the area reduction which contributes to wirelength reduction for the critical path (from input to register path). In terms of power consumption, the marginally increased of 3D architecture power consumption over 2D architecture is due to the increased of logic gates in 3D architecture as well as its total wirelength as a result of separate place and route run for each tier.

Figure 10 shows the horizontal wirelength distribution of 2D MPSoC, bottom tier and top tier of heterogeneous 3D stacking where below 0.8 mm length, it can be seen that the number of wires for the heterogeneous 3D stacking is decreased but have more wires for wirelength between 0.8 mm and 0.9 mm. As we run separate place and route for each tier, therefore the tool will optimize each tier accordingly without considering the complete 3D architecture which could be the reason of this trend.

6.1 2D vs 3D Clock Tree Analysis

Clock tree synthesis for 3D architecture has been studied especially for synthesizing clock tree in many tiers targeting low skew as well as low power consumption. In [21], several clock tree topologies have been analyzed based on the fabricated three-tier 3D chip using MIT Lincoln Lab technology. Measured data from the fabricated chip suggesting that the H-tree structure gives the lowest skew but highest power consumption compared with the other clock tree structures. Several clock tree schemes have also been proposed considering various objectives such as timing yield, fault tolerant, TSVs blockage problem, testability and process variation between dies and within a die [22] [23] [24] [25] [26].

Table 2: Performance comparison for 2D and Heterogeneous 3D MPSoC stacking

Parameters	2D architecture	3D heterogeneous stacking
Core area (mm2)	21.4	10.4
Number of gates (million)	2.70	2.73
Number of total microbumps	-	3011
Nimber of microbumps per tile	-	188
Microbumps for IMEM per tile	-	42
Microbumps for DMEM per tile	-	76
Microbumps for FSL per tile	-	70
Total mesure de longueur (m)	21.1	21.4
Critical path delay for NoC clock (ns)	3.51	3.19
Critical path delay for processor clock (ns)	9.92	10.09
Power Consumption @ 333 MHz (W)	1.38	1.48

Several physical design implementations of 3D architecture has been reported previously conducting performance analysis based on layout-level netlist. However, there is no details discussion regarding the implications of the generated clock tree structure using 2D CTS tools to the overall 3D clock tree structure. Even though there are some works used 2D tool to generate the clock tree [27] [28], nonetheless they did not measure the impact of the method to the 3D timing performance which is the aim of this particular discussion. In this section, comparison of clock tree structure between the baseline 2D architecture and heterogeneous 3D stacking is carried out to have better insight as well as to highlight issues related to the 3D clock tree structure.

One of the benefits of deploying GALS architecture is that we are able to control the rising value of clock skew in the fully synchronous implementation especially for advanced technologies where very dense clock tree structure is created due to the higher registers density. The higher level of clock tree structure increases the clock skew value as well as more sensitive to the on-chip variation (OCV) [25]. In GALS architecture, as the clock skew constraints is limited only to its block boundaries thereby open up design spaces for performance enhancement as well as less hardware requirement since the complexity of the clock distribution is reduced.

The clock tree synthesis for both architectures is done automatically by the CTS Engine in SoC Encounter where the clock specification file is generated based on the supplied timing constraints. A microbump per clock signal has been placed at the center of the top tier to enable balance

distribution between both tiers from the clock source that coming from the top tier. As shown in the figures, CTS Engine synthesized the clock tree with H-tree topology at the first three or four levels. Table III presents the clock tree synthesis structure between 2D and 3D design where it is clearly shown that the clock tree structure of 3D design (combine both bottom and top tiers clock tree structure) for processor clock and NoC clock have less number of clock tree level compared with the 2D design. For the number of sinks and number of buffers, the difference between 2D and 3D design is not very significant for both processor and NoC clock which is indicating that 3D design does simplify the clock tree structure through reducing the number of clock tree level for the same number of sinks and buffers. Another point is that generating clock tree synthesis in 3D design using 2D physical design tool does not have differ substantially whether the clock tree structure is exist only in a single tier of the 3D design or exist in both tiers.

Reduction of the number of clock tree level could potentially improve power consumption where clock network has substantial portion of total power consumption in a chip especially in advanced technology [29]. However, as shown in Table 3, the clock skew of processor clock in 3D architecture is larger than in 2D design whereas NoC clock the opposite trend. The possible reason for the large skew of processor clock in 3D architecture is because the processor clock tree for both tiers has been generated and optimized separately during place and route step which although the optimization process is able to reduce the number of clock tree level, however the tool does not able to minimize the clock skew because it does not see the complete 3D architecture during the optimization process.

Table 3: Clock tree structure for 2D MPSoC and heterogeneous 3D MPSoC architecture

Parameters	2D		3D (bottom tier)		3D (top tier)	
	Processor clock	NoC clock	Processor clock	NoC clock	Processor clock	NoC clock
Level	17	10	7	-	6	8
Number of buffers	944	1580	879	-	72	1599
Number of sinks	40928	72832	38640	-	2288	72832
Skew (ns)	0.40	0.43	Processor clock skew = 0.76 NoC clock skew = 0.07			

6.2 Implications of 3D IC design using 2D EDA tools

One of the primary limitation of using 2D EDA tools for designing and implementing 3D IC architecture is the lack of design exploration support. To be able to gain as much performance as possible from the 3D technology, the need for design exploration is utmost important to evaluate different implementation trade-offs for a specific target hardware or application before proceeding with complete design implementation flow. Specific to the heterogeneous 3D stacking at block-level partitioning, as long as the critical paths reside inside the block architecture thereby using 2D EDA tools seem to be sufficient enough to be able to design as well as doing optimization due to the fact that the tools does not require to see the complete 3D architecture.

7. Conclusion

In this chapter, we have discussed the physical design implementation of heterogeneous 3D stacking of NoC-based MPSoC architecture. We explored other feasible 3D architecture implementation of MPSoC architecture to analyze its performance as well as to have more understanding with regards to the architectural design trade-offs for MPSoC implementation using 3D technology under the limitation of using 2D EDA tools. The GALS style implementation provides benefits due to separate clock domains between communication and computation architecture which could be the main interest for employing it in 3D architecture. One of the important points in designing 3D architecture for heterogeneous 3D stacking architecture with block-level partitioning is that 2D EDA tools can be used as in a normal flow 2D design by carefully partitioning the design to have 2D critical paths located within a tier and thus does not need 3D-specific optimization process.

References

[1] M. B. Healy, K. Athikulwongse, R. Goel, M. Hossain, D. H. Kim, Y.-J. Lee, D. L. Lewis, T.-W. Lin, C. Liu, M. Jung, B. Ouellette, M. Pathak, H. Sane, G. Shen, D. H. Woo, X. Zhao, G. H. Loh, H.-H. S. Lee, and S. K. Lim, "Design and analysis of 3D-MAPS: A many-core 3D processor with stacked memory," in *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, 2010, pp. 1–4.

[2] D. H. Woo, N. H. Seong, and H.-H. S. Lee, "Heterogeneous die stacking of SRAM row cache and 3-D DRAM: An empirical design evaluation," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, 2011, pp. 1–4.

[3] H. Sun, J. Liu, R. S. Anigundi, N. Zheng, J.-Q. Lu, K. Rose, and T. Zhang, "Design of 3D DRAM and Its Application in 3D Integrated Multi-Core Computing Systems," *Design & Test of Computers, IEEE*, vol. 26, no. 5, pp. 36–47, 2009.

[4] V. De Paulo and C. Ababei, "3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans," *International Journal of Reconfigurable Computing*, vol. 2010, pp. 1–12, 2010.

[5] V. S. Nandakumar and M. Marek-Sadowska, "A Low Energy Network-on-Chip Fabric for 3-D Multi-Core Architectures," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 2, no. 2, pp. 266–277, 2012.

[6] K.-S. Chong, K.-L. Chang, B.-H. Gwee, and J. S. Chang, "Synchronous-Logic and Globally-Asynchronous-Locally-Synchronous (GALS) Acoustic Digital Signal Processors," *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 3, pp. 769–780, 2012.

[7] L. A. Plana, S. B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, and S. Yang, "A GALS Infrastructure for a Massively Parallel Multiprocessor," *Design & Test of Computers, IEEE*, vol. 24, no. 5, pp. 454–463, 2007.

[8] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1214–1224, 2006.

[9] R. S. Patti, "Homogeneous 3D Integration," in *Three Dimensional System Integration: IC Stacking Process and Design*, A. Papanikolaou, D. Soudris, and R. Radojcic, Eds. Springer US, 2011, pp. 51–71.

[10] A. R. Marschner, S. Craven, and P. Athanas, "A Sandbox For Exploring The Openfire Processor," in *ERSA'07*, 2007, pp. 248–251.

[11] T. Kranenburg, "Design of a Portable and Customizable Microprocessor for Rapid System Prototyping," Master Thesis, Delft University, 2009.

[12] S. Craven, C. Patterson, and P. Athanas, "Configurable Soft Processor Arrays Using the OpenFire Processor," in *MAPLD 05*, 2005.

[13] A. R. Marschner, "An FPGA-based Target Acquisition System," Master Thesis, Virginia Polytechnic Institute and State University, 2007.

[14] M. H. Jabbar, D. Houzet, and O. Hammami, "3D multiprocessor with 3D NoC architecture based on Tezzaron technology," in *3D Systems Integration Conference (3DIC), 2011 IEEE International*, 2012, pp. 1–5.

[15] M. Krstic, E. Grass, F. K. Gurkaynak, and P. Vivet, "Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook," *Design & Test of Computers, IEEE*, vol. 24, no. 5, pp. 430–441, 2007.

[16] Z. Yu and B. M. Baas, "High Performance, Energy Efficiency, and Scalability With GALS Chip Multiprocessors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, no. 1, pp. 66–79, 2009.

[17] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 29–41, 2008.

[18] M. Grange, A. Y. Weldezion, D. Pamunuwa, R. Weerasekera, Z. Lu, A. Jantsch, and D. Shippen, "Physical mapping and performance study of a multi-clock 3-Dimensional Network-on-Chip mesh," in *3D System Integration, 2009. 3DIC 2009. IEEE International Conference on*, 2009, pp. 1–7.

[19] U. Y. Ogras, R. Marculescu, D. Marculescu, and E. G. Jung, "Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, no. 3, pp. 330–341, 2009.

[20] J. L. Ayala, A. Sridhar, and D. Cuesta, "Thermal modeling and analysis of 3D multi-processor chips," *Integration, the VLSI Journal*, vol. 43, no. 4, pp. 327–341, Sep. 2010.

[21] V. F. Pavlidis, I. Savidis, and E. Friedman, "Clock Distribution Networks in 3-D Integrated Systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 12, pp. 2256–2266, 2011.

[22] X. Zhao, S. Mukhopadhyay, and S. K. Lim, "Variation-tolerant and low-power clock network design for 3D ICs," in *Electronic Components and Technology Conference (ECTC), 2011 IEEE 61st*, 2011, pp. 2007–2014.

[23] C.-L. Lung, Y.-S. Su, S.-H. Huang, Y. Shi, and S.-C. Chang, "Fault-tolerant 3D clock network," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, 2011, pp. 645–651.

[24] X. Zhao and S. K. Lim, "Through-silicon-via-induced obstacle-aware clock tree synthesis for 3D ICs," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, 2012, pp. 347–352.

[25] J.-S. Yang, J. Pak, X. Zhao, S. K. Lim, and D. Z. Pan, "Robust Clock Tree Synthesis with timing yield optimization for 3D-ICs," in *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, 2011, pp. 621–626.

[26] X. Zhao, D. L. Lewis, H.-H. S. Lee, and S. K. Lim, "Low-Power Clock Tree Design for Pre-Bond Testing of 3-D Stacked ICs," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 5, pp. 732–745, 2011.

[27] M. B. Healy, "Performance and Temperature Aware Floorplanning Optimization for 2D and 3D Microarchitectures," Master Thesis, Georgia Institute of Technology, 2006.

[28] T. Thorolfsson, "Three-Dimensional Integration of Synthetic Aperture Radar Processors," PhD Thesis, North Carolina State University, 2011.

[29] P. Salihundam, S. Jain, T. Jacob, S. Kumar, V. Erraguntla, Y. Hoskote, S. Vangal, G. Ruhl, and N. Borkar, "A 2 Tb/s 6 x 4 Mesh Network for a Single-Chip Cloud Computer With DVFS in 45 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 4, pp. 757–766, 2011.