Physical Design of a 3D-Stacked Heterogeneous Multi-Core Processor

Randy Widialaksono, Rangeen Basu Roy Chowdhury, Zhenqian Zhang, Joshua Schabel,
Steve Lipa, Eric Rotenberg, W. Rhett Davis, Paul Franzon
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC, USA
{rhwidial, rbasuro, zzhang18, jcledfo3, slipa, ericro, wdavis, paulf}@ncsu.edu

***Abstract*—With the end of Dennard scaling, three dimensional stacking has emerged as a promising integration technique to improve microprocessor performance. In this paper we present a 3D-SIC physical design methodology for a multi-core processor using commercial off-the-shelf tools. We explain the various flows involved and present the lessons learned during the design process. The logic dies were fabricated with GlobalFoundries 130 nm process and were stacked using the Ziptronix face-to-face (F2F) bonding technology. We also present a comparative analysis which highlights the benefits of 3D integration. Results indicate an order of magnitude decrease in wirelengths for critical inter-core components in the 3D implementation compared to 2D implementations.**

I. Introduction

As performance benefits from technology scaling slows down, computer architects are looking at various architectural techniques to maintain the trend of performance improvement, while meeting the power budget. One promising architecture is the 3D-stacked single-ISA heterogeneous multicore processor (HMP), with provision for fast thread migration [1] between the cores. The large amount of vertical connectivity between the stacked dies enables fast inter-core data and state transfer, improving the overall system performance and energy efficiency. We designed and fabricated a 3D-stacked HMP, a 2D version of which was previously fabricated and validated [2]. The two cores were generated using FabScalar [3] and have different microarchitectures as shown in Table I. The cores in each tier of the stack are connected by high bandwidth inter-tier buses that allow fast thread migration with latency less than 100 cycles.

In this paper we present our physical design methodology of the 3D-stacked HMP. We explain the various flows involved and present the lessons learned during the design process. Each core of the two-core-stack was fabricated on independent dies. We fabricated the two dies with GlobalFoundries 130 nm process, which were then assembled in to a two tier stack using the Ziptronix face-to-face (F2F) bonding technology (Fig. 1) [4]. This is a multi-project chip with four independent designs: the heterogeneous processor, a SIMD core, a DiRAM [5] cache controller, and a prototype inter-tier asynchronous communication bus. Although the methods and learnings presented in this paper apply to the entire chip, our primary focus will remain on the heterogeneous core-stack.

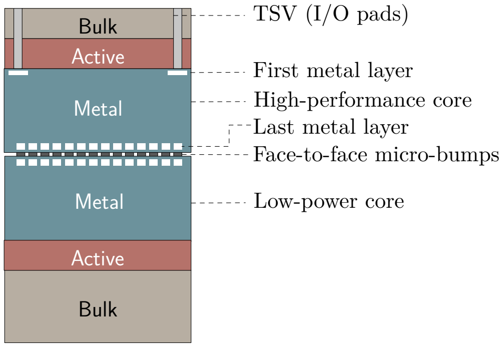


Fig. 1. Cross-section of the face-to-face bonded 3D-IC stack.

The primary advantage of 3D-stacking comes from reduced wirelengths leading to an improvement in routability and signal delays. On the other hand, going 3D also increases design complexity. Therefore, it is crucial to understand the benefits of 3D to justify the extra design effort. To this end, we compared our 3D design with two equivalent 2D designs in terms of wirelengths and delays for some of the key components in our CPU core, which play a crucial role in the fast thread migration process.

A. Thread Migration in HMP

The architectural state of a CPU comprises of register state and memory state. In a traditional HMP design, threads migrate between cores through a very expensive and high latency context save and context restore operation. Our HMP design reduces thread migration latency by using two key features:

TABLE I
CORE TYPES IN THE 3D PROCESSOR STACK.

Parameter	High-Performance (Top Die)	Low-Power (Bottom Die)
Frontend Width	2	1
Issue Width	3	3
Pipeline Depth	9	9
Issue Queue Size	32	16
Physical Reg. File Size	96	64
Load/Store Queue Size	16/16	16/16
Reorder Buffer Size	64	32
L1 I-Cache	private, 4 KB, 1-way, 8 B block, 1 cycle	
L1 D-Cache	private, 8 KB, 4-way, 16 B block, 2 cycle	

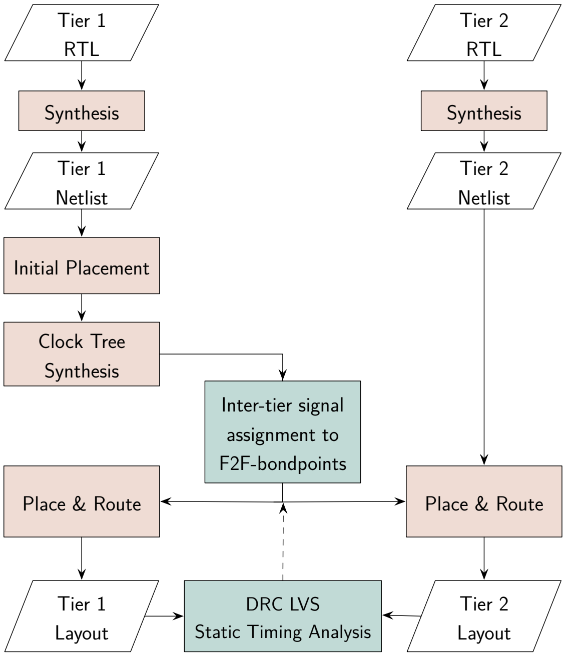


Fig. 2. Overview of H3 3D-IC physical design flow.

- Fast Thread Migration: This uses a pair of register file in each core, called Teleport Register File (TRF), connected through a high bandwidth inter-core bus for instantaneous swap of register state. Each bit of one TRF is connected to a corresponding bit in the second TRF. This creates a large number of inter-core wires that must be routed efficiently.
- Cache Core Decoupling: This allows one CPU core to directly access the Level 1 caches from the other core (remote cache access) thus avoiding cache flushes and cold cache misses. This feature adds additional inter-core datapaths with very strict timing constraints.

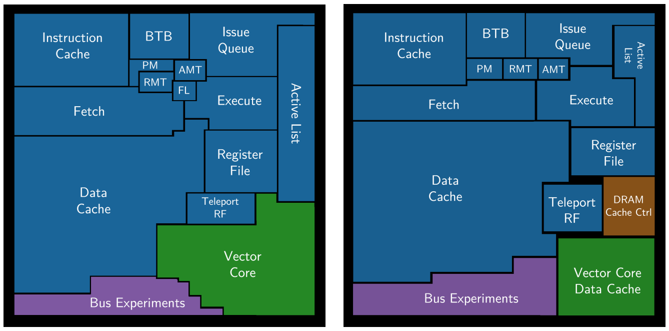
Further details of these two techniques can be found in [1], [6].

II. Design Flow

There are three major challenges in reaching design closure: power delivery, inter-tier signal connectivity, and clock distribution across tiers. In this section, we describe the approaches we take for each of these major design goals.

A. Floorplanning

The full-chip floorplan is shown in Fig. 3. The top die consists of 3 experiments: a high-performance FabScalar core, a vector core, and 3D-IC bus experiments. The bottom die has the low-power FabScalar core, while adding a DRAM cache controller experiment and its I/O pads. Since the floorplan impacts routability, in our initial floorplan, we allocated extra area to partitions with inter-die connections to provide sufficient routing tracks to the F2F vias. The F2F via pitch was determined prior to floorplan. Moreover, due to the size of the top metal shape for the F2F bumps, every F2F via needs an antenna diode. We considered the area overhead of these diode cells during floorplanning.



(a) Top Die

(b) Bottom Die

Fig. 3. Full-chip floorplan (H3 processor partitions in blue).

B. Power Delivery Network

The starting point of power planning was our fabricated 2D prototype where we conducted static IR drop analysis and extensive power measurements during validation of the chip. With power pads only available on one tier, power to the second tier must be delivered through the F2F vias. The 3D power-plan should satisfy the following three requirements - 1) The number of power pads should be sufficient to deliver enough current to both tiers, 2) the power rings must be robust enough to be able to carry the required amount of current from the power I/O pads to the logic on both tiers, and 3) sufficient power must be delivered through the F2F vias to the other tier without significant IR drop. To satisfy these requirements, we first calculated the required number of power I/O pads based on the pad datasheet. To account for the additional tier, we increased the current carrying capacity of the power delivery network compared to the verified 2D prototype. We doubled the width of the power rings, used additional metal layers for the power ring, and doubled the number of vertical power stripes. For efficient power delivery through the F2F vias, we used the following methodology - (1) We placed the exact same power grid structure for both dies to guarantee perfect overlap. (2) The distance between power rings and stripes were

TABLE II
Physical design metrics of the 3D processor stack.

Physical Design Metrics	
Die Dimensions	3.92 mm x 3.92 mm
Core Area per die	9.57 mm^2
Standard Cells (top die)	886,361
Standard Cells (bottom die)	678,854
Memory macros	34
Nets (top die)	482,479
Nets (bottom die)	328,535
Average net length (top die)	64.6 μm
Average net length (bottom die)	66.9 μm
Inter-tier F2F signal nets	6,077
Inter-tier power vias	30,796
Average F2F net length (top die)	86 μm
Average F2F net length (bottom die)	140.3 μm

kept as multiples of the F2F via pitch so that columns of F2F vias completely align with the power stripes. (3) Finally, we connected the power straps to the F2F vias by placing a cell generated by a custom Calibre^{TM} script. This script parses the power strap locations followed by instantiating appropriate vias to the top metal layer. The generated cell was then instantiated at the same coordinates in both dies, thus connecting the power grids of the two tiers through the F2F vias.

C. Inter-tier Signal Connectivity

The initial placement was performed after first removing inter-tier signal ports from the design netlist. This is to allow placement unconstrained by inter-tier I/O, otherwise the place and route tool would incorrectly assume a location for inter-tier I/O ports along the boundary as a design constraint. After the initial placement is obtained, these inter-tier signal ports were added back in using engineering change order (ECO) commands, while incrementally loading a design constraints (SDC) file for load capacitance and timing constraints.

The next step was to synthesize the clock tree, followed by assigning inter-tier nets to F2F vias. First we prioritize F2F vias for the power delivery network, and the remaining available vias were assigned to inter-tier signals using an automated nearest neighbor approach. In case of multiple fan-outs for output inter-tier signals, we assigned them to the nearest available via to the driving cell. While for input signals, we assigned them to an available via nearest to the mean center of its fan-in cells. The via assignment result was used to create a floorplan specification to be imported into Cadence Encounter. The same specification was used for both dies to ensure correct connection. The automated inter-tier signal to via assignment method is based on our previous work in [7], complemented with timing slack information [8].

D. Inter-tier Timing Closure

One of our key design requirement was to avoid performing timing synchronization across cores/tiers. In a wafer-stacking process, inter-tier synchronization is challenging because of the process variation from different wafers. Timing analysis on every permutation of process corners would be necessary to model worst-case scenarios. The architecture was designed to allow the cores to run on independent clocks, except during the thread migration process where the thread swap block in both cores must operate synchronously for a bulk swap of architectural state [9], [10]. This requires clock forwarding from one tier to the other. Hence timing synchronization is still required, albeit the scope is reduced to only an isolated, small percentage of the clock sinks. To achieve better symmetry between the clock trees of the two dies, we used very tight constraints for clock tree synthesis. We also performed extensive post-layout static timing analysis to guarantee that constraints were met for inter-tier paths.

We conducted hold violation fixing as an ECO step after routing the two dies. We performed parasitics extraction, followed by static timing analysis on both dies as a single system using Synopsys PrimeTime^{TM}. We created a wrapper netlist that instantiates both dies. We used worst case combination of PVT corner for the two dies to model inter-die process variation. Based on this analysis, we performed manual hold fixes for each violating paths by using either buffer insertion, gate sizing, or substituting cell types in Cadence Encounter. This process was repeated until all hold paths were satisfied. The iterative process, however, can be prohibitively expensive in terms of time and design effort; the risk can be minimized by constraining the place and route tool with more aggressive hold margin.

E. Physical Verification

In order to validate the final layout of the two dies, each die was independently verified for design rule check (DRC) violations. Layout versus schematic (LVS) checks were also run to confirm the integrity of the die layouts. However, 3D integration requires 3D specific physical verification. We must ensure that the two dies are correctly connected after performing ECO changes, by conducting our custom 3D specific DRC and LVS checks in Calibre.

1) 3D Design Rule Check: There were two mask layer sets related to the 3D process: the TSV layers set for the back-etched I/O pads, and the top metal layer. In our customized I/O pads, the shapes on the TSV layers were manually verified to satisfy required dimensions. For the top metal layer DRC, we ensured that the only shapes that exist are squares that form the F2F via grid with correct size, offset, and pitch.

2) 3D Layout versus Schematic Check: Inter-tier signal connectivity was verified by using our custom 3D-LVS script. Labels for I/O ports, which includes inter-tier signals, were placed after the F2F via assignment step during place and route. The 3D-LVS script first extracts the labels and identifies F2F via shapes from the final layout. The LVS script then verifies connectivity by checking for vias to the routing layers, followed by matching the extracted labels and coordinates between the two tiers.

III. Analysis

In this section we will present an analysis of F2F via pitch and a comparative analysis of wirelength, path delay and power consumption between 3D and 2D implementations.

A. Face-to-face Via Pitch Analysis

Designers may have the option to select a F2F via pitch offered by the 3D bonding process technology at potentially different price point and yield. The benefit of using a finer via pitch for inter-tier signals is the reduction of face-to-face via

TABLE III
Face-to-face via experiment parameters

F2F Pitch (µm)	Via Diameter (µm)
1.5	0.75
3	1.5
5	2.5
8	3
10	3

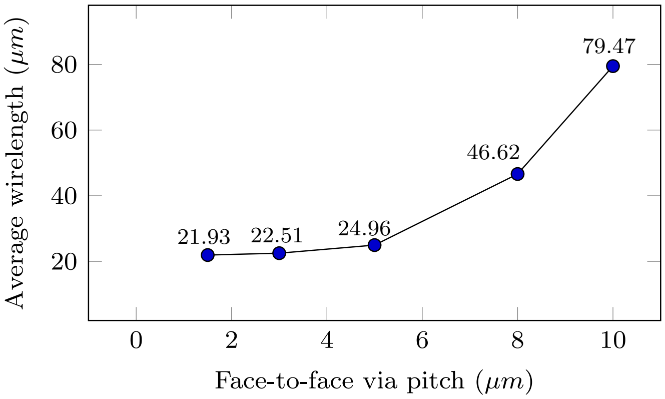


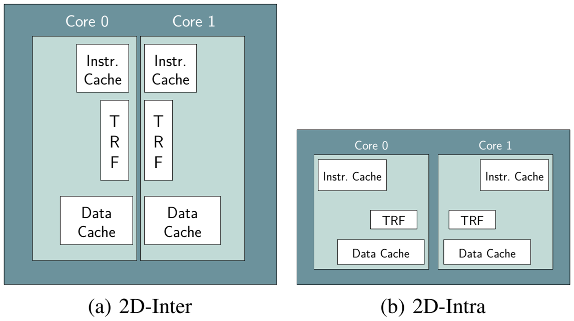
Fig. 4. Impact of face-to-face via pitch on wirelength of *Teleport Register File* inter-tier signals.

contention between inter-tier signals. Contention can be indirectly observed through total/average wirelength measurements. Routing congestion leads to route detours, thus wirelength increases. Fig. 4 presents wirelength measurement results on one tier of the TRF implementation in an isolated experiment using 130 nm technology, across F2F via parameters shown in Table III. The fabricated 3D chip stack uses an 8 µm pitch. We observe that wirelength of the inter-tier signals in the TRF can be halved if a 5 µm pitch is used, but would not decrease much further at finer pitches. In the synthesized TRF netlist, each signal fans out to at least five cells and an antenna diode. These cells could not all be placed near the corresponding F2F via due to timing constraints. This observation shows that logic complexity and floorplanning also impacts the utilization of F2F vias.

B. Comparison with 2D Implementation

A CPU core is highly-sensitive to floorplan changes due to its tight timing constraints. Going 3D relaxes the constraints a little due to a decrease in wirelength. For accurate understanding on the impact of floorplan, we analyze two 2D floorplans illustrated in Fig. 5:

- **2D-Inter:** This floorplan was optimized for inter-core communication, with inter-core structures placed near the edge. The aspect ratio of the partitions were tailored to accommodate the wide inter-core bus interface, namely: 1) *Transport Register File* (TRF) module for register state migration, 2) *Instruction Cache Buffer* module, and 3) *Load Store Unit* module which contains the data cache and multiplexing logic.
- **2D-Intra:** This floorplan consists of a core floorplan optimized for intra-core timing as fabricated in the 3D implementation. The *2D-intra* floorplan consists of two cores mirrored towards each other with inter-core signal pins placed on one side. This floorplan yields a significantly different cell placement compared to the 3D implementation, due to the different I/O pin location constraints. This floorplan models the scenario where the CPU is a hard IP, hence it would not be possible to optimize the floorplan for wide inter-core connections.



(a) 2D-Inter (b) 2D-Intra

Fig. 5. 2D floorplans of the heterogeneous multi-core processor.

We compare the wirelengths and delays of various components in these two 2D floorplans to those in our 3D floorplan (Fig. 3).

1) Wirelength Analysis: Fig. 6 shows total wirelength of inter-tier signals spanning across both cores. The average wirelength of inter-tier signals in the TRF of *2D-intra* is almost double that of *2D-inter* since the TRF module was placed 540 µm from the partition edge. We observe that the instruction cache path did not improve as much as the data cache path when going from *2D-intra* to *2D-inter* to 3D. This is due to timing-driven placement that places the cells closer to internal circuits rather than the partition edge. This observation demonstrates the competing interest between intra-core and inter-core timing constraints in the 2D designs. Although the data cache in both *2D-inter* and *2D-intra* is placed near the edge, the aspect ratio difference affects the cell distance and available routing resources. Overall, going 3D reduced the average wirelength of the system by 10% and 22% compared to *2D-inter* and *2D-intra* respectively. This wirelength reduction translates to less parasitics hence decreasing the core's power consumption in 3D.

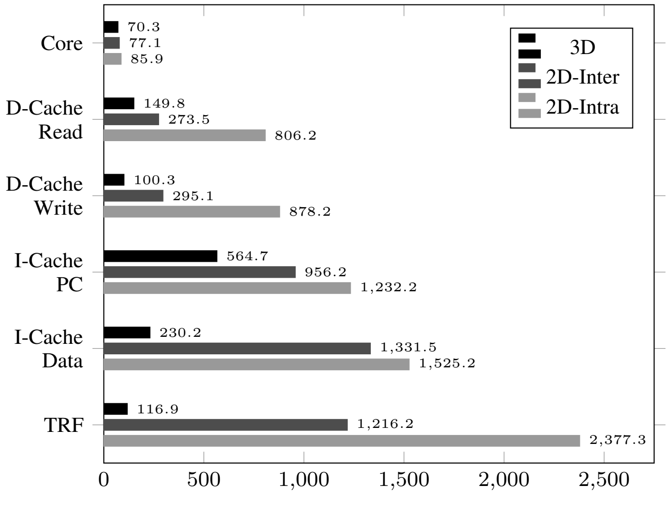


Fig. 6. Average wirelength (μm) of various CPU components related to inter-core connectivity.

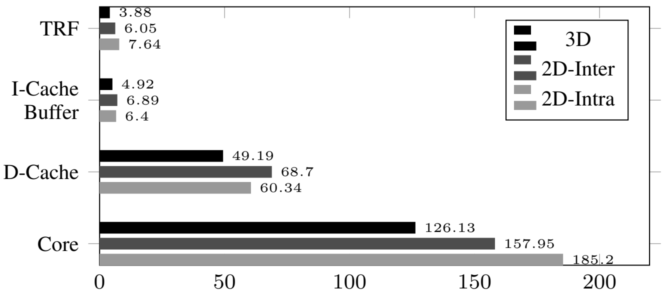


Fig. 7. Power consumption (mW) of various CPU components related to inter-core connectivity.

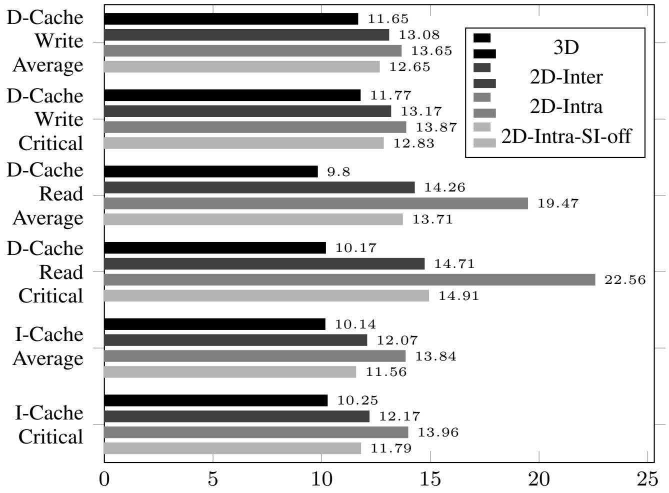


Fig. 8. Path delays (ns) of inter-core cache datapaths.

2) Power Consumption: The average power measurement results shown in Fig. 7 were obtained from Cadence Encounter on the parasitics extracted layout. In this experiment we assume that both CPU cores are active. The 3D design consistently consumes less power than the 2D designs, especially for modules which facilitate inter-core state transfer. The wirelength reduction in the 3D design translates to 20% and 31% savings in core power consumption compared to *2D-inter* and *2D-intra* respectively.

3) Cache Datapath Delay: Fig. 8 shows path delay comparison between the 3D and 2D floorplans. The instruction cache path consists of two buses, instruction bus and program counter bus. Instruction is obtained from reading the instruction cache, which was implemented as a synchronous compiled memory macro, hence provides a timing path endpoint. The program counter path has more logic compared to the instruction path, thus it has higher critical path delay and average path delay. In Fig. 8, the values associated with instruction cache are for the more critical program counter path.

The data cache path consists of four buses: read address bus, read data bus, write address bus, and write data bus. The data cache memory array was implemented with standard cell flip-flops, and its address decoder was synthesized into standard cell gates. In the cache-core decoupling implementation, a remote data cache read access crosses the inter-core interface twice within a single cycle. The place and route tool had to insert buffers in these long timing paths in order to meet timing constraints. In contrast to a read access, a write cache access only needs to cross the inter-core interface once. From Fig. 8 we observe that the path delay for remote data cache read access is longer than a write access. The critical path delay for a data-cache read access in *2D-intra* is $22.56\,\mathrm{ns}$, which violates the target clock period of $15\,\mathrm{ns}$. This path however meets the timing constraint when signal integrity analysis was turned off, indicating large crosstalk effects on the very wide inter-core bus. This timing path was met with a large slack margin in 3D, facilitated by the cross-tier interface.

IV. Conclusion

In this paper we discussed our methodology for physical design of a logic-on-logic 3D-stacked design using current off-the-shelf 2D electronic design automation tools. We also discussed the physical design benefits of a 3D implementation over 2D implementations. The 3D processor design demonstrated 30% power improvement compared to a 2D implementation. Timing closure in a 3D stacked design remains challenging with current tools and is open for future work.

Acknowledgement

The 3D-IC heterogeneous multi-core processor project is supported by a grant from Intel Corporation.

References

[1] E. Rotenberg, B. H. Dwiel, E. Forbes, Z. Zhang, R. Widialaksono, R. B. R. Chowdhury, N. Tshibangu, S. Lipa, W. R. Davis, and P. D. Franzon, "Rationale for a 3d heterogeneous multi-core processor," in *Computer Design (ICCD), 2013 IEEE 31st International Conference on*, pp. 154–168, 2013. ID: 1.

[2] E. Forbes, Z. Zhang, R. Widialaksono, B. Dwiel, R. B. R. Chowdhury, V. Srinivasan, S. Lipa, E. Rotenberg, W. R. Davis, and P. D. Franzon, "Under 100-cycle thread migration latency in a single-isa heterogeneous multi-core processor," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, pp. 1–1, Aug 2015.

[3] N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg, "FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores Within a Canonical Superscalar Template," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA-38, pp. 11–22, June 2011.

[4] P. Enquist, "Scalable direct bond technology and applications driving adoption," in *3D Systems Integration Conference (3DIC), 2011 IEEE International*, pp. 1–5, Jan 2012.

[5] D. Chapman, "Diram architecture overview," *Tezzaron Semiconductors*, 2014.

[6] V. Srinivasan, "Phase ii implementation and verification of the h3 processor," Master's thesis, North Carolina State University, 2015.

[7] R. Widialaksono, W. Zhao, W. R. Davis, and P. Franzon, "Leveraging 3d-ic for on-chip timing uncertainty measurements," in *3D Systems Integration Conference (3DIC), 2014 International*, pp. 1–4, Dec 2014.

[8] R. Widialaksono, *Three-Dimensional Integration of Heterogeneous Multi-Core Processors*. PhD thesis, North Carolina State University, Raleigh, June 2016.

[9] Z. Zhang and P. Franzon, "Tsv-based, modular and collision detectable face-to-back shared bus design," in *3D Systems Integration Conference (3DIC), 2013 IEEE International*, pp. 1–5, Oct 2013.

[10] Z. Zhang, *Design of On-chip Bus of Heterogeneous 3DIC Micro-processors*. PhD thesis, North Carolina State University, Raleigh, June 2016.