Design and Analysis of 3D-MAPS: A Many-Core 3D Processor with Stacked Memory

Michael B. Healy, Krit Athikulwongse, Rohan Goel, Mohammad M. Hossain, Dae Hyun Kim, Young-Joon Lee,
Dean L. Lewis, Tzu-Wei Lin, Chang Liu, Moongon Jung, Brian Ouellette, Mohit Pathak, Hemant Sane,
Guanhao Shen†, Dong Hyuk Woo, Xin Zhao, Gabriel H. Loh†, Hsien-Hsin S. Lee, and Sung Kyu Lim
School of Electrical and Computer Engineering, College of Computing†
Georgia Insitute of Technology
{mbhealy, leehs, limsk}@ece.gatech.edu, loh@cc.gatech.edu†

***Abstract*—We describe the design and analysis of 3D-MAPS, a 64-core 3D-stacked memory-on-processor running at 277 MHz with 63 GB/s memory bandwidth, sent for fabrication using Tezzaron's 3D stacking technology. We also describe the design flow used to implement it using industrial 2D tools and custom add-ons to handle 3D specifics.**

I. Introduction

The potential of 3D IC stacking has been examined by researchers for many years. Only recently has the increasing cost of continuing process technology shrinks and the incredible memory-bandwidth demand of multi- and many-core systems brought 3D technology to the forefront of commercial interest. Many universities and companies are actively investigating and investing in 3D stacking technologies for its promise to deliver this extreme bandwidth.

In this work we demonstrate our methodology for designing and analyzing 3D-MAPS (3D MAssively Parallel processor with Stacked memory), a 64-core 3D-stacked memory-on-processor system that demonstrates nearly an order of magnitude higher memory bandwidth at lower operating frequencies compared to previous efforts [1], [2]. This processor is designed to demonstrate the extreme memory bandwidth available using 3D interconnects above and beyond previous endeavors.

In addition, we address the specific issues that 3D designers will encounter dealing with tools that are not specifically designed to meet their needs. There are several works presented in the literature that describe various 3D architecture design options and physical design algorithms for 3D ICs, but very few in the area of 3D design demonstration and methodology. Thorolfsson *et al.* [3] described the design of an FFT processor with 3D stacked memory. However, they do not discuss cross-talk or power-noise analysis in 3D systems and do not include a thermal analysis. The contributions of this paper are as follows:

- This paper presents 3D-MAPS, arguably the first many-core 3D processor in academia. Our 3D processor contains 64 5-stage pipelined, 2-way in-order VLIW cores. Two dies are stacked in 3D-MAPS, one 64-core die and one SRAM die. Each core owns a dedicated 4KB SRAM tile, which is stacked above the core and connected using face-to-face 3D vias. Our architecture is verified with several multi-core benchmarks. 3D-MAPS demonstrates memory bandwidth of over 63 GB/s based on our verified many-core benchmark simulations.
- This paper describes in detail how to construct the physical layouts of 3D-MAPS processor and how to perform various 3D analysis. Our tool-flow is based on commercial tools from Cadence, Mentor Graphics, and Synopsys and enhanced with various add-ons we developed to handle TSVs and 3D stacking. We provide sign-off 3D timing, power, thermal, IR-drop, signal integrity, and clock waveform analysis results based on DRC/LVS-passed 3D-GDSII layouts.

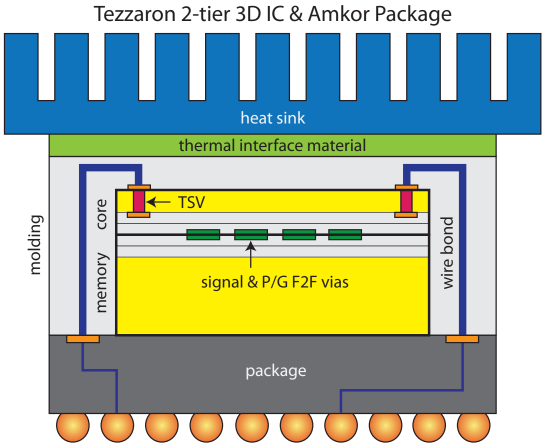

Fig. 1. Side view of the final stacked dies based on Tezzaron's F2F and TSV stacking technology and Amkor's wirebond packaging

3D-MAPS was taped-out in March 2010 using 130nm Global Foundries' technology and Tezzaron's TSV/3D technology. Once the fabrication and package/board design are completed in September 2010, our simulation results will be verified using measurements.

II. 3D Stacking Technology

The 3D-MAPS processor will be fabricated using a six-metal $130nm$ process provided by Global Foundries that is modified to include through-silicon vias (TSVs) according to the specification of Tezzaron Semiconductor. The TSVs are manufactured in a via-first process. Trenches are etched into the silicon and filled with Tungsten. Then devices and metal layers are patterned. Next, wafers are flipped and bonded. Finally, one wafer is thinned until the trenched TSVs are revealed from the backside. This produces a two-layer face-to-face bonded stack that uses TSVs for IO. Because the wafers are bonded before thinning, there is never a need to handle a thinned wafer. Figure 1 shows a diagram of the completed die stack. With metal layers, the thinned die is $12\mu m$ thick and the thick die is $765\mu m$.

The Tezzaron process produces very small TSVs that are approximately $1.2\mu m$ wide with $2.5\mu m$ minimum pitch and $6\mu m$ height. The face-to-face (F2F) connection, which is used for the main die-to-die communication, uses $3.4\mu m$ Metal 6 pads with $5\mu m$ pitch. The TSVs have a parasitic resistance of around $600m\Omega$ and a parasitic capacitance of about $15fF$. The F2F connection has negligible

This material is based upon work supported by the US Department of Defence.

resistance and capacitance, about the same as a local via. The 3D-MAPS die footprint is $5 \times 5mm$.[1] Therefore, the maximum face-to-face connection count is one million. The 130nm Global Foundries standard cell library provided to us includes only peripheral-style IO. For that reason, we include functional TSVs only underneath the IO-cell pads.

III. 3D-MAPS Architecture

A. Core Architecture

The goal of our 3D-MAPS architecture is to demonstrate the rich bandwidth made possible by the high-density die-to-die vias when running data-parallel applications. Given that our design is area- and power-constrained, we also want to make cores and inter-core communication highly power-efficient [4] by eliminating unnecessary large, complex structures during the architectural planning stage.

Under this design philosophy, for the single core we first defined a custom two-way VLIW architecture to eliminate area- and wire-dominated components such as complex decoder, dynamic instruction scheduler, reorder buffer, data disambiguation mechanism, etc. Instead, we offload these functionalities to the software. In our two-way instruction format, one slot is dedicated to a memory instruction to consume memory bandwidth every cycle from the 3D-stacked memory while the other slot is tailored for an ALU instruction. When the memory instruction is absent, our ISA allows certain commonly used ALU instructions to be executed in the memory pipeline. Our ISA supports auto-increment to further increase memory bandwidth utilization by improving the memory-to-ALU instruction ratio.

To cope with control hazards without any impact on our limited area, we employed delay slots for change-of-flow instructions. Nonetheless, they are made completely transparent to the programmers as our assembler's optimizer will reschedule and ensure the correctness of the final binary. With assistance from the system software, the implementation can be relieved from those power- and area-consuming units such as branch predictor, branch target buffer, and pipeline squashing mechanism for mis-speculation while maintaining similar execution efficiency of being speculative.

The design of inter-core communication in a many-core processor can lead to numerous implications for power, performance, and routing area. To minimize power consumed by the interconnect, we employed a point-to-point 2D mesh communication paradigm controlled by explicit communication and synchronization instructions. In particular, we found a 2D mesh network addresses some issues of two other alternatives: 2D torus and folded-torus network topology. First, a simple 2D mesh eliminates the long wires that connect two cores on the boundaries of the same row or column in a 2D torus. Second, it halves the wire routing space required over the core-to-core boundary of a folded torus. Although such an explicit communication model could reduce programmers' productivity, it can provide higher performance, yet reduce dynamic power at the same time. In particular, we argue that a network-on-chip router would be overkill, since most of the data-parallel applications targeted for our processor demonstrate stable, predictable, and regular communication patterns among cores when properly partitioned. Each of our 3D-MAPS cores features four buffers for sending or receiving data from its north, south, east, or west neighbor. Synchronization among cores is achieved by a global barrier, whose implementation was laid out as an H-tree on the core layer.

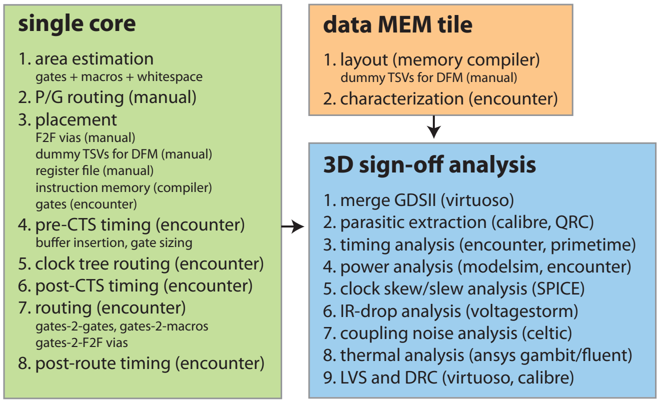
single core
1. area estimation
gates + macros + whitespace
2. P/G routing (manual)
3. placement
F2F vias (manual)
dummy TSVs for DFM (manual)
register file (manual)
instruction memory (compiler)
gates (encounter)
4. pre-CTS timing (encounter)
buffer insertion, gate sizing
5. clock tree routing (encounter)
6. post-CTS timing (encounter)
7. routing (encounter)
gates-2-gates, gates-2-macros
gates-2-F2F vias
8. post-route timing (encounter)
data MEM tile
1. layout (memory compiler)
dummy TSVs for DFM (manual)
2. characterization (encounter)
3D sign-off analysis
1. merge GDSII (virtuoso)
2. parasitic extraction (calibre, QRC)
3. timing analysis (encounter, primetime)
4. power analysis (modelsim, encounter)
5. clock skew/slew analysis (SPICE)
6. IR-drop analysis (voltagestorm)
7. coupling noise analysis (celtic)
8. thermal analysis (ansys gambit/fluent)
9. LVS and DRC (virtuoso, calibre)

Fig. 2. Our flow for the design and analysis of single-core and single-memory tile stack.

B. Architecture Verification

In the verification process, we developed a multi-level framework to rigorously verify each stage of the design. Our baseline reference models are simply the outputs generated by an x86 machine running our benchmarks written in a high-level language. We then rewrote the benchmark using pseudo-assembly language at register-transfer level in C, *e.g.*, declaring an array of variables to emulate the register file of our architecture, and test the benchmark on an x86 machine. Our pseudo-assembly codes were then ported using our 3D-MAPS ISA, assembled and optimized by our assembler, and simulated on our architectural simulator. The simulation output was verified at cycle-level in lock-step with that of the pseudo-assembly reference model. Up to this point, we have an architectural simulator that conforms to correct functional behavior and predicts the performance of the benchmarks for 3D-MAPS. Finally, we verify the RTL design against our architectural simulator using pipeline traces.

C. Off-chip Interface

The primary design goal for the off-chip interface was to minimize the pin count. Therefore, the interface was modeled after the IEEE 1149.1 test access port with two key deviations. First, we us a custom *test control state machine* (TCSM), which has complete control of the chip, managing functional test, memory initialization, and program execution. Second, we have four pairs of *test data in* and *test data out* (TDI and TDO) pins, instead of the standard one pair. Internally, the sixty-four cores are grouped into four groups of sixteen cores each. The chains in each bank connect serially to one pair of I/O pins.

IV. Physical Design Methodology

Figure 2 shows the overall physical design flow used to produce single-core plus single-memory tile layouts in 3D-MAPS. The physical design flow begins with an RTL description of the processor core written in VHDL. Our top-level module contains a single core (bottom-die), four data memory banks (top), one instruction memory bank (bottom), and a custom-designed register file (bottom). We then use Synopsys Design Compiler to compile VHDL into structural Verilog for each die. The compiled Verilog is then input into Cadence Encounter to perform the automated physical design steps. We use Cadence Encounter to perform gate placement, sizing and buffering optimization, signal routing, clock routing, and power and ground network generation. We also use many of the tools integrated into Encounter to perform early analysis on the design to ensure reasonableness before sign-off analysis is undertaken. However, Cadence Encounter and its integrated point-tools do not understand F2F vias,

[1] This is the space assigned to us as part of the 2009 DARPA/Tezzaron multi-project wafer run.

TSVs, and 3D stacking, i.e., multiple die definitions. Thus, we have developed several add-ons that directly manipulate LEF/DEF and other intermediate files to manage F2F vias, TSV, and 3D stacking. The instruction and data memory bank, tile, and die designs are done with a memory compiler provided by Artisan.

A. 3D Power Ground Network Generation

The power and ground distribution networks are mainly generated using the stripe and ring generation commands in Cadence Encounter. The goal is to have the rings on both the core layer and memory layer line up. By lining up these rings we can connect them using the vast array of face-to-face (F2F) connections. This allows the creation of a very low resistance connection for the power and ground distribution to the memory layer. Decoupling capacitors (decaps) are inserted into the design using Cadence Encounter. This is done prior to placement to provide an even distribution of capacitance. In the memory layer, we insert a large number of decaps on the power rings in the blank space around the memory banks. This allows the memory layer to provide large amounts of on-demand current to the cores.

B. F2F Via and TSV Placement

Communication between the core and memory dies occurs through the face-to-face (F2F) vias as shown in Figure 1. Any net that connects to a F2F via, and thus circuitry on the other die, is called a 3D net. The individual design for each wafer therefore must contain pins for all nets that cross the F2F boundary. The memory layer contains only the data memory banks and their connections. Accordingly, we first fix the location of the memory banks, then we manually place pins in both dies directly above the pins on the memory banks. The TSVs are used only for off-chip IO and power/ground connections in our current version of 3D-MAPS. The foundry-supplied IO cell library is peripheral-style only. Therefore, the only electrically active TSVs are placed inside the IO cells underneath the bond-pad.

One unique requirement that the Tezzaron TSV process imposes is on mandatory minimum TSV pitch of $250\mu m$ throughout the entire wafer. Thus, there needs to be at least one TSV inside every $250\mu m$ window. This requirement is to maintain the planarity of the wafer during chemical and mechanical polishing (CMP). Because the 3D-MAPS cores are $560 \times 560\mu m$ and we do not use TSVs inside the core region, we must place a 3×3 array of dummy TSVs inside each core to meet this maximum pitch requirement. We manually inserted these dummy TSVs before placement.

C. 3D Placement and Routing

Cadence Encounter is used to perform placement and routing at both the many-core level and the single-core level. The 3D connection information is propagated to the placer through the use of fixed pins on Metal 6 representing the F2F connections. These pins constrain the placement to correctly optimize for the full 3D system.

Cadence Encounter is also used to perform sizing and buffering optimizations, and NanoRoute is used to perform routing. The 3D connection information is propagated to the optimizer and router through back-annotation of capacitance and arrival time requirements on the fixed pins. These constraints force the optimization engine and the router to correctly account for both sides of the 3D nets.

D. 3D Clock Routing

We perform clock routing using the clock tree synthesis functions of Cadence Encounter. The clock network is contained mainly within the core layer. Each memory bank in the memory layer has a clock pin that is propagated to the core layer using a fixed F2F connection. This pin is annotated with the capacitance of the routing inside the memory layer as well as the gate capacitance of the clock pin on the memory bank itself. This minimizes the clock skew for both the single core and memory tile stack. At the many-core level, each core has a single input clock pin.

V. 3D Sign-Off Analysis

The existing Cadence, Synopsys, and Mentor Graphics tools are designed for 2D ICs and do not handle 3D designs and TSVs. The following sections describe our strategy to extend these tools to analyze and verify 3D-MAPS.

A. 3D Timing and Signal Integrity Analysis

Our 3D timing analysis is based on Synopsys PrimeTime. First, we prepare the Verilog netlist files of both dies and the SPEF files containing extracted parasitic values for all the nets of the dies. Then, we create a top-level Verilog netlist that instantiates the design of each die and connects the 3D nets using F2F connections. We also create an SPEF file that has a parasitic model of the F2F connections. After that, we run PrimeTime with all the Verilog files and the SPEF files to get the timing analysis results.

3D signal integrity analysis must also contain a 3D component because nets may have enough coupling capacitance to be considered a problem only when all dies are considered simultaneously. For signal integrity analysis, we use Cadence CeltIC. Again, we input an SPEF file that contains the information for both dies and the parasitics from the F2F connections. Then with the merged Verilog netlist, CeltIC finds all the paths with noise violations.

B. 3D Power Noise Analysis

We perform 3D power noise analysis using Cadence VoltageStorm. The stand-alone VoltageStorm takes in a DEF file, technology files, and power dissipation files to generate both peak and average power noise values. Performing this analysis for a 3D design is a challenge.

For our design, we perform true 3D power noise analysis with VoltageStorm. To accomplish this, we compile a technology file that contains all of the 3D layers. This technology file contains multiple copies of each metal layer, one for each layer in the 3D stack. Then, 3D DEF files are constructed from the design of each layer. A separate LEF file must also be constructed that contains instances specific to each layer. Finally, VoltageStorm produces true 3D power noise values.

C. 3D Thermal Analysis

3D designs have the potential to suffer from significant thermal problems due to the higher thermal resistance between active silicon layers and the heatsink. We use ANSYS Gambit and Fluent for our thermal analysis. Gambit is a meshing and model generation software that sets up thermal analysis problems. Fluent is the simulation engine that calculates the thermal distribution of the chip.

Gambit is used to model the 3D chip-stack and includes both core and memory layers, as well as a model of the rest of the package and a $5mm$ tall heatsink. The stack is first divided into numerous thermal layers such as gates, poly, Metals 1-6, via, dielectric, *etc*. To determine the material properties for each mesh volume, the GDSII file is parsed to determine the correct ratio between the various materials of each layer at each particular grid point.

This ratio is used to calculate the weighted average of the material properties and to determine the effective thermal conductivity of that grid point. Power sources are then inserted into some mesh volumes. Finally, Fluent is used to calculate the steady-state thermal map.

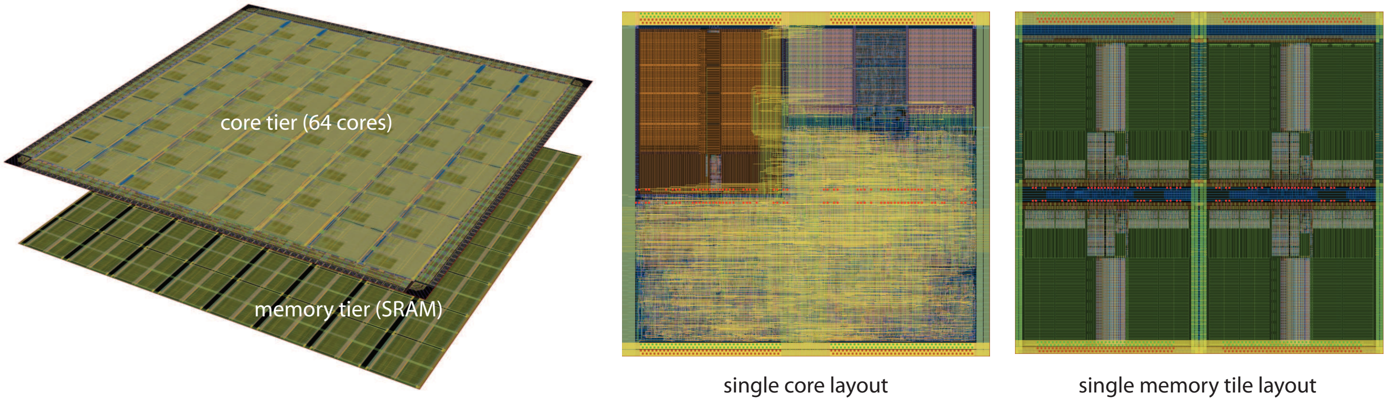

Fig. 3. Various layout views of the 3D-MAPS processor.

TABLE I
ARCHITECTURAL PERFORMANCE METRICS.

Benchmark	Memory Bandwidth (GB/s)	IPC per core	BIPS
string_search	8.9	0.65	11.52
matrix_multiply	13.8	0.32	5.67
median	63.8	1.62	28.72
aes_encrypt	49.5	0.97	17.20
motion estimation	24.1	1.20	21.27
histogram	30.3	0.90	15.96
edge detection	15.6	0.95	16.84
k-means	40.6	0.94	16.66

VI. Simulation and Layout Results

Table I shows the results from our many-core architectural simulations of the 3D-MAPS processor. Using our optimized multi-core benchmark suite aimed for our sponsor's applications, the table reports their respective memory bandwidth in gigabytes per second (GB/s), performance in both instructions per cycle per core (IPC), and billions of instructions per second (BIPS). Depending on each application's behavior, 3D-MAPS achieves memory bandwidth up to 63.8 GB/s, which is higher than that of a modern Intel Core i7 processor and comparable to the memory bandwidth of a high-end GPGPU running at four times the frequency with much larger area.

Table II shows the summary of 3D-MAPS layout. Figure 3 shows various layout views of the 3D-MAPS processor. The core footprint is $560 \times 560\mu m$. The layout of one tile of SRAM memory is also shown. A single tile contains 4 banks of 1KB data memory. Thus, the total SRAM data memory capacity of 3D-MAPS processor is $4KB \times 64 = 256KB$. The full many-core layout of the core layer has dimension of $5 \times 5mm$. Each core is arrayed in an 8×8 grid and core-to-core communication occurs using short wires. The core-to-core pitch is $570\mu m$.

Figure 3 shows the F2F connections used for the 3D communication (red) and power and ground network distribution (orange and green). There are 668 power and ground F2F connections per core, and $42,752$ power and ground F2F connections over the entire die. Each core also uses 116 F2F connections for signals and clock, for a total of $7,424$ F2F connections over the entire die. There are $1,784$ TSVs used for IO and 576 dummy TSVs. Timing optimization inserted 970 buffers.

The supply voltage for 3D-MAPS is 1.5V. The total IR-drop inside a single core is about $13mV$ inside one core. The total IR-drop inside a single memory tile is about $10mV$. These values include true 3D-aware IR-drop analysis using sign-off-level Cadence VoltageStorm.

3D timing analysis reports that the maximum frequency is $277MHz$. The timing critical path runs through the double-pumped, four-ported register file. The longest delay for a 3D net is for the address bus, which has a sink in each of the four memory banks and thus has large wirelength. The maximum crosstalk noise value on the worst net is $674mV$, which is very close to the noise limit. The next highest noise value is much lower at $592mV$. The maximum temperature from simulation is $47°C$.

TABLE II
PHYSICAL DESIGN SUMMARY.

Process technology	Global Foundries 130nm
Die size	$5 \times 5mm$
Core footprint	$560 \times 560\mu m$
Core-to-core pitch	$570\mu m$
PG 3D connections/core	668
Total PG 3D connectiions	$42,752$
Data 3D connections/core	116
Total data 3D connections	$7,424$
TSVs/IO pad	204
Total IO TSVs	$47,940$
Dummy TSVs	$6,540$
Total maximum IR-drop	$78mV$
Maximum operating frequency	$277MHz$

VII. Conclusions

We have presented the design, layout, and analysis of 3D-MAPS, a 64-core memory on processor 3D stacked system. It was built from the ground up to demonstrate extreme memory bandwidth using 3D connections. The layout and analysis was performed using commercial tools with several custom add-ons to enable full 3D awareness. 3D-MAPS simulates correctly at 277 MHz and verified architectural simulations show that it achieves memory bandwidth above 63 GB/s on selected benchmarks.

References

[1] K. N. et al. An inductive-coupling link for 3D integration of a 90nm CMOS processor and a 65nm CMOS SRAM. In *IEEE Int. Solid-State Circuits Conf.*, pages 480–481,481a, feb. 2009.

[2] U. K. et al. 8Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology. *IEEE Journal of Solid-State Circuits*, 45(1):111–119, jan. 2010.

[3] T. Thorolfsson, K. Gonsalves, and P. Franzon. Design automation for a 3dic fft processor for synthetic aperture radar: A case study. In *Proc. ACM Design Automation Conf.*, pages 51–56, 2009.

[4] D. H. Woo and H.-H. S. Lee. Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era. *IEEE Computer*, 41(12):24–31, 2008.