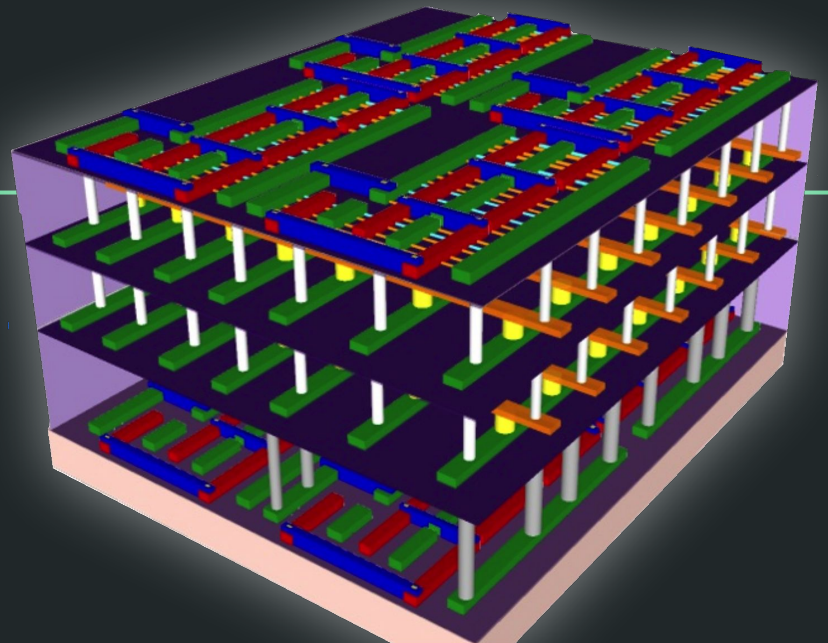
Designing Vertical Processors in Monolithic 3D

Bhargava Gopireddy and Josep Torrellas

ISCA 2019

Presented by Kevin Huang and Ifeanyi Ene



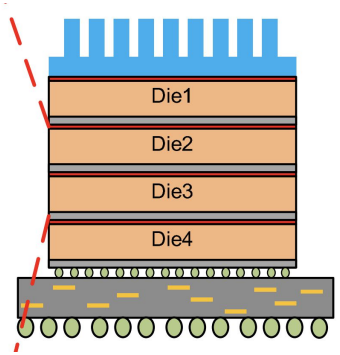
Bhargava Gopireddy
Ph.D @ UIUC
Computer Architect @ Nvidia

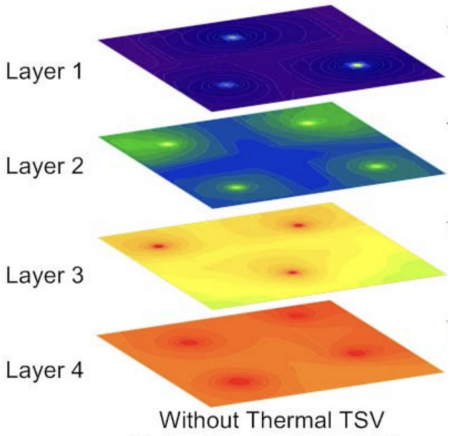
Josep Torrellas
Professor of Computer Science and Willet Faculty Scholar

Review: TSV3D

Through-Silicon Vias

- Vertical electrical connection that passes completely through a silicon wafer [1]
- First 3D chips based on TSV were invented in the 1980s [2]
- Used in many commercial DDR3
- Poor match for vertical processors
  - Inhibits fine-grained hardware partitioning across dies
  - Low conductivity makes cooling layers far from hint sink difficult





1. https://en.wikipedia.org/wiki/Through-silicon_via#cite_note-12
2. Lau, John H. (2010). Reliability of RoHS-Compliant 2D and 3D IC Interconnects.

M3D

Monolithic 3D

- Communicate using Monolithic Interlayer Vias
  - 2 orders smaller than TSVs
  - Ultra-high density
- Fine-grain partitioning of processor structures across layers
- Reduces wire length, energy consumption, and footprint
  - Lower latency
- Different layers have different performance
  - Manufacturing challenges

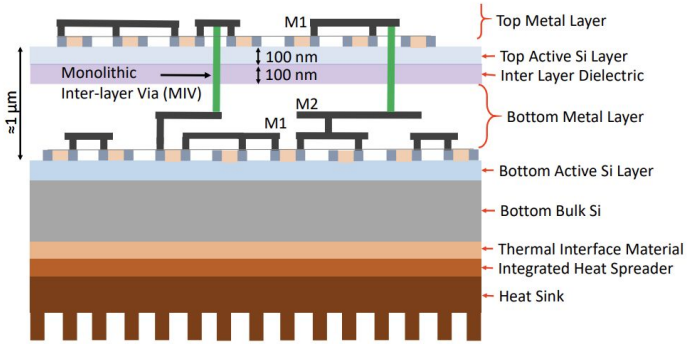
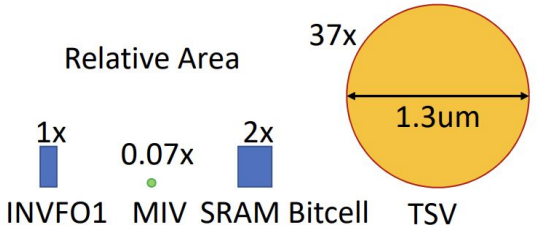


Figure 1: M3D integration of two layers.



Partitioning Granularity and Trade-offs

Transistor Level (N/P) Partitioning

- Places N-type and P-type transistors on two different layers
- Extra overhead for N/P transistor pair via

Gate level (Intra-block) Partitioning ⭐

- Adjacent gates can either be in the same layer or in a different layer
- Reduce footprint of core by up to 50% as well as power consumption

Block Level Partitioning

- Placing individual blocks such as ALUs, RF, IQs, etc in different layers.

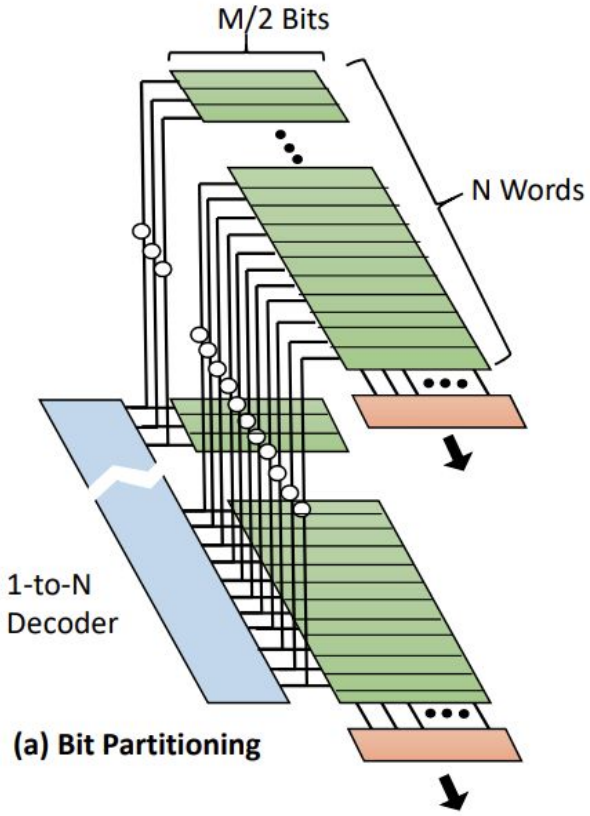
Previous 3D Partitioning Strategies

Bit Partitioning

- Partition bits into two or more layers
  - Spreads half of each word in each layer, placing a driver in each layer
- Best for BTB, DTLB, ITLB, IL1, DL1, and L2 in M3D and nearly all storage structures in TSV.

	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	28%	22%	40%	14%	15%	37%
TSV3D	25%	19%	31%	4%	-3%	4%

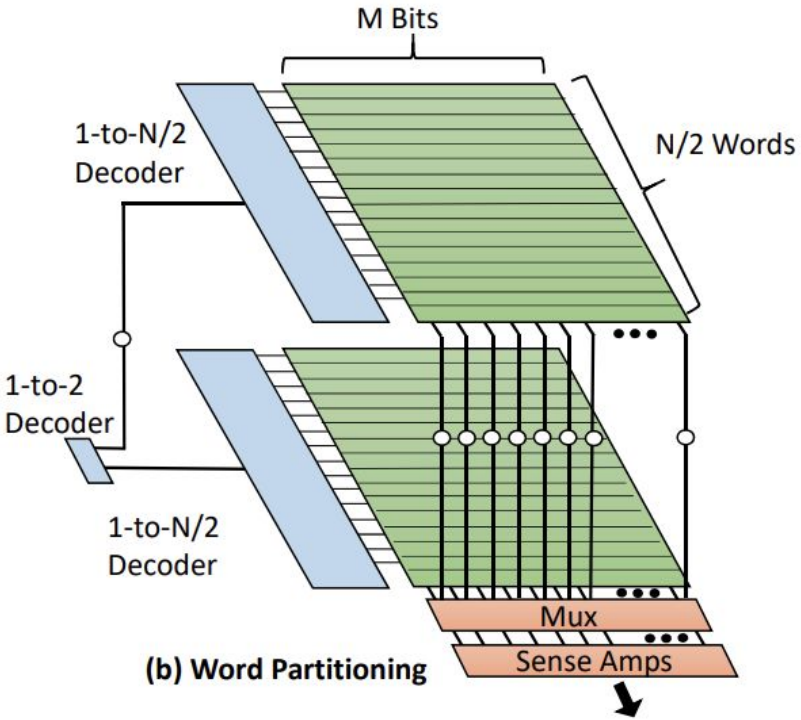
Table 3: Percentage reduction in access latency, access energy, and area footprint through bit partitioning.



3D Partitioning Strategies

Word Partitioning

- Spreads half the words in each layer, and places a driver in each layer
- Number of vias needed is equal to array width
- BP generally preferred over WP because BP reduces wordline access latency



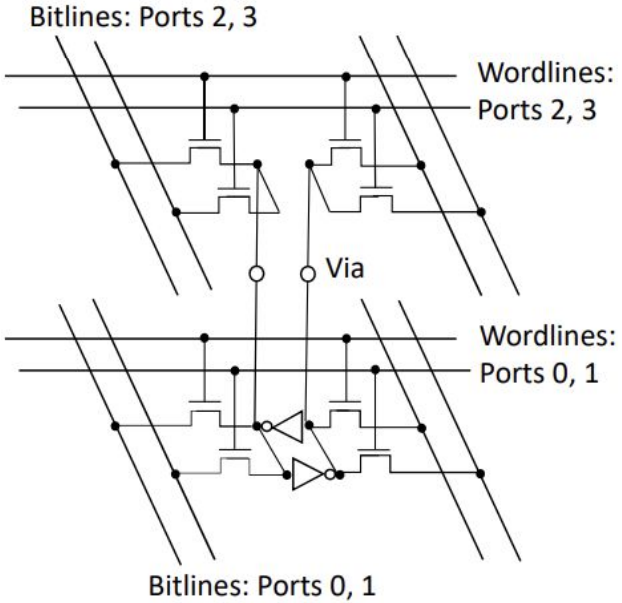
	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	27%	35%	43%	14%	36%	57%
TSV	24%	32%	39%	-6%	9%	19%

Table 4: Percentage reduction in access latency, access energy, and area footprint through word partitioning.

3D Partitioning Strategies

Port Partitioning

- Places half its ports in one layer and rest of ports in the other layer
- For SRAM specifically, PP requires two vias per SRAM bit cell
- Reduces both wordline and bitline length by nearly half, hence reducing latency, energy, and area.



(c) Port Partitioning

	Register File (RF)			Branch Pred. Table (BPT)		
	Laten.	Ener.	Footpr.	Laten.	Ener.	Footpr.
M3D	41%	38%	56%	-	-	-
TSV	-361%	-84%	-498%	-	-	-

Table 5: Percentage reduction in access latency, access energy, and area footprint through port partitioning.

TSVs are too big for PP!
Two vias per SRAM bit cell

3D Partitioning Strategies Summarized

Structure [Words; Bits per Word] × Banks	Best Partition		Latency Reduc.(%)		Energy Reduc.(%)		Footprint Reduc.(%)	
	M3D	TSV.	M3D	TSV.	M3D	TSV.	M3D	TSV.
RF [160; 64]	PP	BP	41	25	38	19	56	31
IQ [84; 16]	PP	BP	26	17	35	5	50	32
SQ [56; 48]	PP	BP	14	-3	21	-18	44	0
LQ [72; 48]	PP	BP	15	2	36	8	48	10
RAT [32; 8]	PP	WP	20	10	32	5	45	-11
BPT [4096; 8]	WP	BP	14	4	36	-3	57	4
BTB [4096; 32]	BP	BP	15	-6	20	-10	37	-20
DTLB [192; 64] ×8	BP	BP	26	18	28	20	35	22
ITLB [192; 64] ×4	BP	BP	20	7	28	11	36	11
IL1 [256; 256] ×4	BP	BP	30	14	36	23	41	25
DL1 [128; 256] ×8	BP	BP	41	31	40	33	44	34
L2 [512; 512] ×8	BP	BP	32	24	47	42	53	46

Table 6: Best partition method for each structure, and percentage reduction in latency, energy and area footprint.

Partitioning A Core in M3D - Logic Stages

Logic Stages

- We can fold each core into about half of its original area, and two cores can share global wires, reducing global footprint. Reduces delays for same # cores.

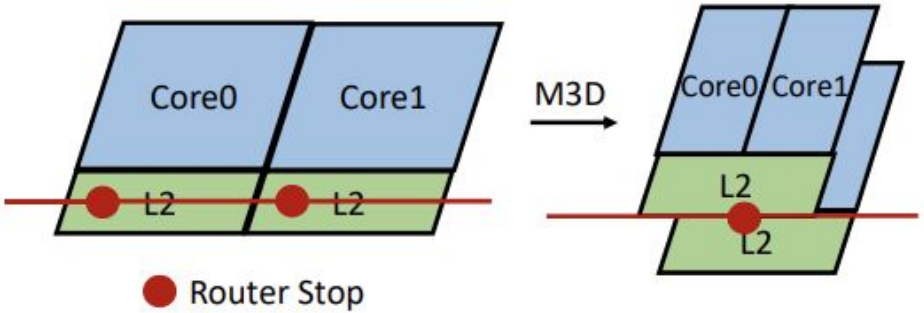


Figure 4: Two cores sharing the L2s and the router stop.

Heterogeneous Layer Partitioning

Motivation

- The top layer in M3D is processed at a lower temperature, resulting in slower transistors compared to the bottom layer
- To compensate, designers adopt heterogeneous partitioning strategies

Design Adaptations

- Critical logic (e.g., key signal paths) is kept in the bottom layer to maintain performance
- In storage structures, fewer ports are allocated to the top layer, and transistor sizes are increased to offset slower speeds
- Asymmetry in partitioning (e.g., assigning 2/3 of an array to the bottom layer) helps balance performance with area and energy considerations

Hetero-Layer Partitioning (contd.)

Structure	Partitioning Technique	
Logic Stage	Critical paths in bottom layer; non-critical paths in top	
Storage Structure	Port Partitioning	Asymmetric partitioning of ports, and larger access transistors in top layer
	Bit or Word Partitioning	Asymmetric partitioning of array, and larger bit cells in top layer
Mixed Stage	Combination of the previous two techniques	

Table 7: Partitioning techniques for a hetero-layer M3D core.

Hetero-Layer Partitioning (contd.)

Logic Stages

- In an integer execution unit, the critical carry propagate and sum paths are assigned to the bottom layer, while non-critical blocks (with ample slack) are placed in the top layer
- For the decode stage, simple decoders remain in the bottom layer, while the complex decoder and µcode ROM (which are less performance-critical) are moved to the top

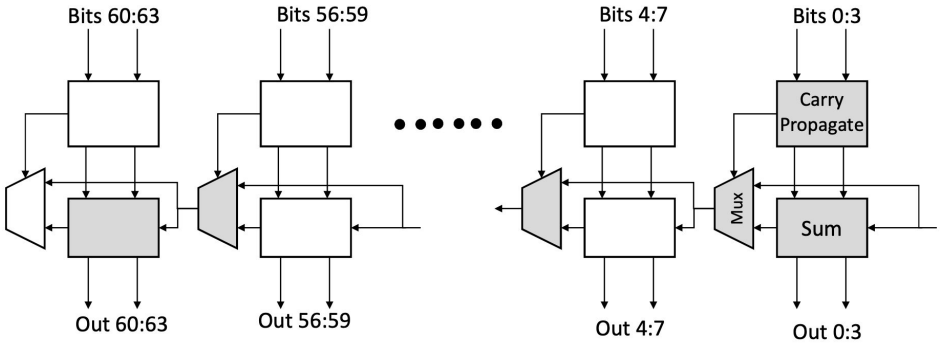


Figure 5: ALU with shaded critical-path blocks.

Hetero-Layer Partitioning (contd.)

Storage Stages

- In a register file, an optimized split (e.g., 10 ports in the bottom layer vs. 8 in the top with double-width transistors) achieves up to 47% area reduction compared to 2D designs
- Similar adaptations are applied in the issue queue, store queue, and branch prediction table to maintain high performance

	RF (%)	IQ (%)	SQ (%)	LQ (%)	RAT (%)	BPT (%)	BTB (%)	DTLB (%)	ITLB (%)	IL1 (%)	DL1 (%)	L2 (%)
Latency	40	24	13	13	20	13	13	23	18	27	37	29
Energy	32	30	17	30	24	30	16	25	25	33	36	42
Area	47	47	43	47	44	40	26	25	28	30	31	42

Table 8: Percentage reduction in access latency, access energy, and area footprint with the best hetero-layer partitioning compared to a 2D layout.

Architectures Enabled by M3D

Exploiting Wire Delay Reduction

- Faster clock frequencies by shortening interconnects
- Option to increase issue width or add extra ports while keeping the same frequency
- Lower voltage operation to reduce power consumption and allow more cores in the same power budget

Heterogeneous M3D Designs

- Use different transistor technologies in the two layers (e.g., high-performance in the bottom, low-power in the top)
- Achieve energy savings while balancing performance

Novel Architectures

- Integrate specialized accelerators or additional computing engines on the top layer
- Enable tight coupling between general-purpose cores and memory (such as non-volatile memory)
- Support entirely new computing paradigms by merging diverse processing elements into one chip

Evaluation Methodology

Simulation Environment

- Architectural simulator (Multi2Sim) is used to model a 4-core out-of-order processor with detailed parameters
- CACTI and McPAT tools provide power, timing, and area estimations for logic and memory structures

Parameter	Value
Cores	4 out-of-order cores, V_{dd}=0.8V
Core width	Dispatch/Issue/Commit: 4/6/4
Int/FP RF; ROB	160/160 registers; 192 entries
Issue queue	84 entries
Ld/St queue	72/56 entries
Branch pred.	Tournament, with 4K entries in selector, in local predictor, and in global predictor; 32-entry RAS
BTB	4K-entry, 4-way
FUs & latencies:	
4 ALU	1 cycle
2 Int Mult/Div	2/4 cycles
2 LSU	1 cycle
2 FPU	Add/Mult/Div: 2/4/8 cycles; Add/Mult issue every cycle; Div issues every 8 cycles
Private I-cache	32KB, 4-way, 32B line, Round-trip (RT): 3 cycles
Private D-cache	32KB, 8-way, WB, 32B line, RT: 4 cycles
Private L2	256KB, 8-way, WB, 64B line, RT: 10 cycles
Shared L3	Per core: 2MB, 16-way, WB, 64B line, RT: 32cycles
DRAM latency	RT after L3: 50ns
Network	Ring with MESI directory-based protocol

Table 9: Parameters of the simulated architecture.

Evaluation Methodology (contd.)

Design Configurations Evaluated

- Baseline 2D design versus several M3D designs (iso-layer, hetero-layer, and aggressive hetero-layer variants)
- Both single-core and multicore evaluations are conducted to assess performance, energy savings, and thermal efficiency

Name	Configuration
Single Core	
Base	Baseline 2D, f=3.3GHz
M3D-Iso	Iso-layer M3D, f=3.83GHz
M3D-HetNaive	Hetero-layer M3D without modifications, f=3.5GHz
M3D-Het	Hetero-layer M3D with our modifications, f=3.79GHz
M3D-HetAgg	Aggressive M3D-Het, f=4.34GHz
TSV3D	Conventional TSV3D, f=3.3GHz
MultiCore	
M3D-Het	M3D-Het + Shared L2s, 4 cores, f=3.79GHz
M3D-Het-W	M3D-Het + Shared L2s, Issue=8, 4 cores, f=3.3GHz
M3D-Het-2X	M3D-Het + Shared L2s, 8 cores, f=3.3GHz, V_{dd}=0.75V
TSV3D	Conventional TSV3D + Shared L2s, 4 cores, f=3.3GHz

Table 11: Core configurations evaluated.

Results – Single Core

Performance Gains

- Iso-layer M3D designs achieve up to 28% faster execution than 2D baselines due to increased frequency and shorter critical paths
- Aggressive hetero-layer designs can push performance improvements further, with reductions in key delays (e.g., load-to-use and branch misprediction paths)

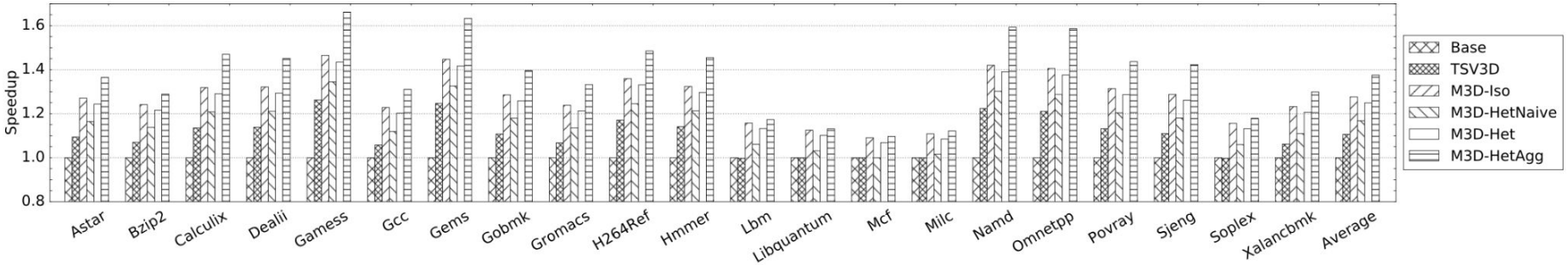


Figure 6: Speed-up of different M3D designs over *Base* (2D).

Results – Single Core (contd.)

Energy and Area Efficiency

- Reported energy consumption is reduced by 39–41%, while area footprint is substantially lowered, enhancing overall efficiency

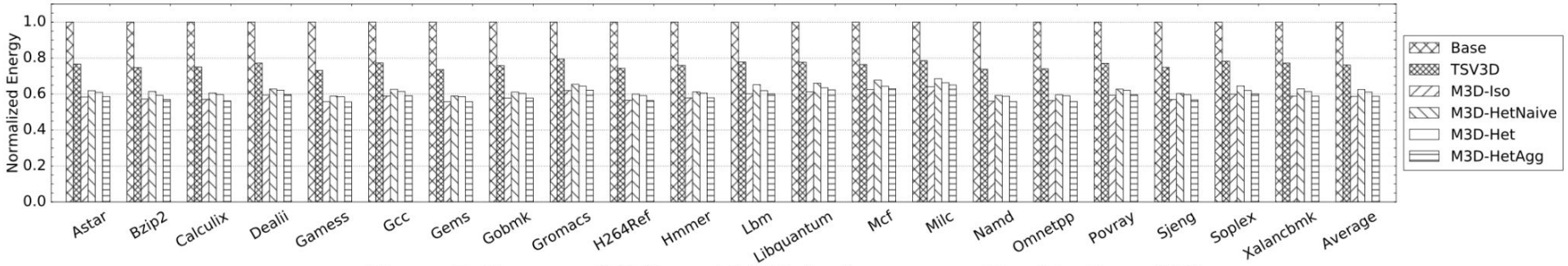


Figure 7: Energy of different M3D designs normalized to *Base* (2D).

Results – Single Core (contd.)

Thermal Benefits

- Improved vertical thermal conduction ensures minimal temperature variation across layers, contributing to robust operation under high frequencies

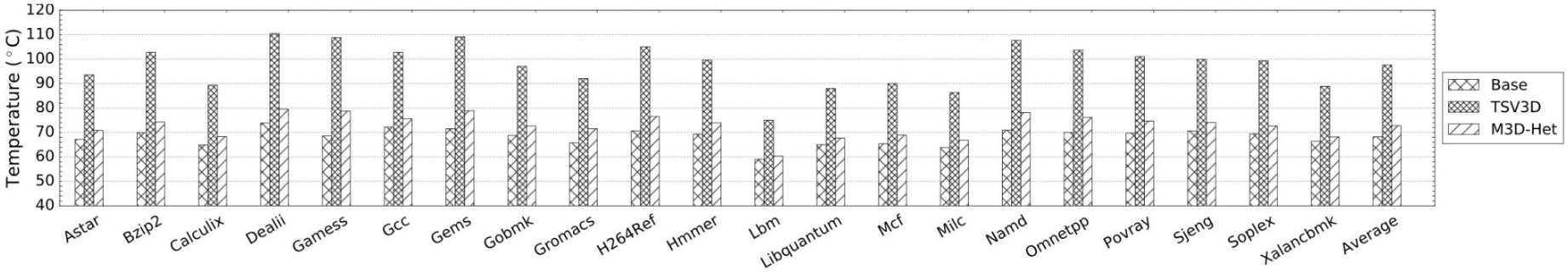


Figure 8: Peak temperature in centigrade degrees for different designs.

Results – Multicore

Scalability

- Multicore architectures based on M3D can incorporate twice as many cores under a similar power budget compared to 2D designs

Performance Metrics

- When cores share L2 caches, multicore designs achieve up to 92% faster performance with 39% less energy consumption

Design Variants

- Variations such as M3D-Het-W (wide core) and M3D-Het-2X (increased core count with voltage scaling) demonstrate trade-offs between frequency, power, and throughput

Results – Multicore (contd.)

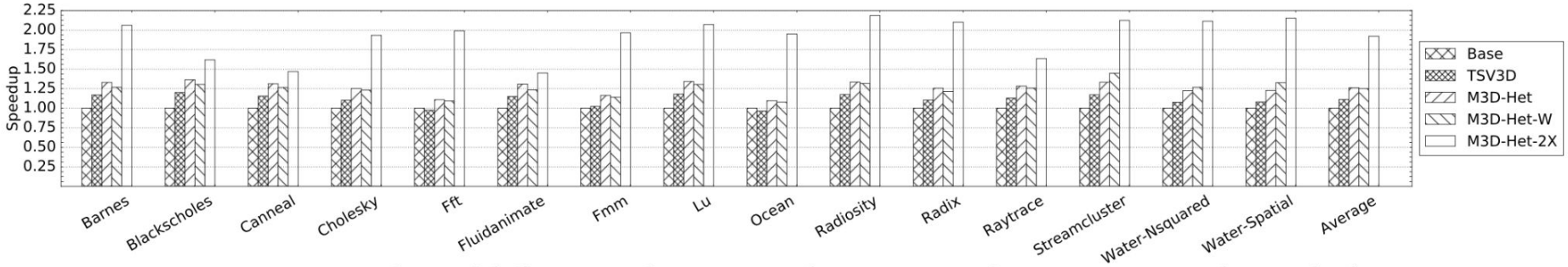


Figure 9: Speed-up of different multicore M3D designs over a four-core *Base* multicore (2D).

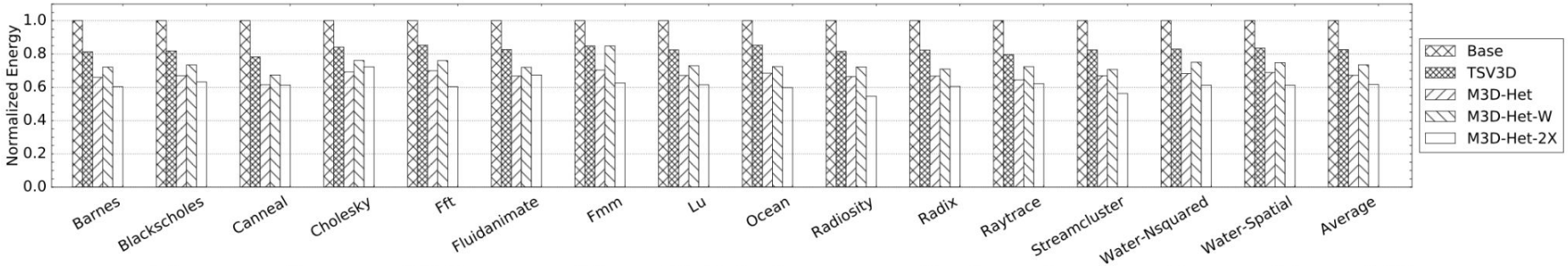


Figure 10: Energy of different multicore M3D designs normalized to a four-core *Base* multicore (2D).

What did the Paper get Right?

What did the Paper get Wrong?

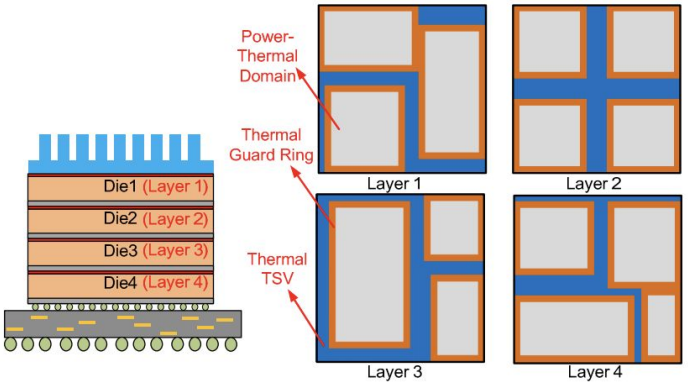


Fig. 8. Power-thermal domain setup in a 4-layer TSV 3D-IC with back-to-face bonding technique.

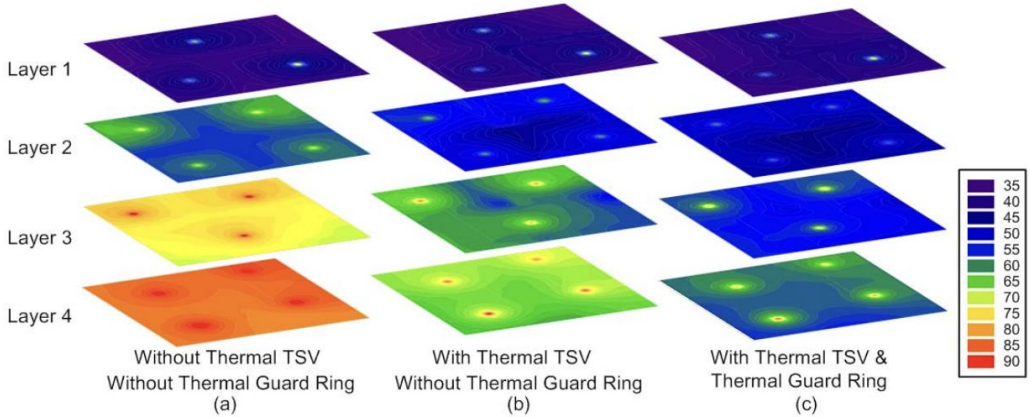


Fig. 9. Temperature distribution planes of three TSV 3D-IC structures (a) without the thermal guard rings and thermal TSVs (b) without thermal guard rings but with thermal TSVs (c) with both thermal guard rings and thermal TSVs.

Conclusion

- Partitioned the processor for M3D into two layers (logic and storage), considering the top layer's lower-performance transistors.
- Placed critical logic paths in the bottom layer.
- Used asymmetric partitioning for storage: the top layer has fewer ports with larger access transistors or a shorter bitcell subarray with larger bitcells.
- Under conservative M3D assumptions, the M3D core ran applications 25% faster and used 39% less energy than a 2D core.
- An aggressive M3D design achieved 38% faster performance and 41% lower energy consumption compared to a 2D core.
- With a similar power budget, an M3D multicore could double the number of cores of a 2D multicore, running applications 92% faster while consuming 39% less energy.
- The M3D core was also thermally efficient.