CPU with integrated SRAM Architecture Comparison ASML

4CM70, Integrated System Design - Q2 (2025)

Group 04

Full Name	Student Id
Philip Offermans	1853244
Tycho Brouwer	1753320
Dean Vermee	2348470
Floris Widdershoven	1735322
Romke van Heezik	1333372

Eindhoven, January 9, 2026

Abstract

Contents

1 | Introduction (1 page)

A CPU (Central Processing Unit) is a key component of a computer system. Its role is to interpret and execute instructions obtained from memory, coordinating all operations within a computer. Modern CPUs typically consist of multiple (logic) cores, where each core is capable of independently executing instructions. CPUs also contain different levels of (SRAM) cache memory, which are small but extremely fast memory blocks located close to the cores. These caches reduce the time it takes for the CPU to access frequently used data and instructions, significantly improving overall performance. A schematic representation of a CPU is shown in Figure 1.1.

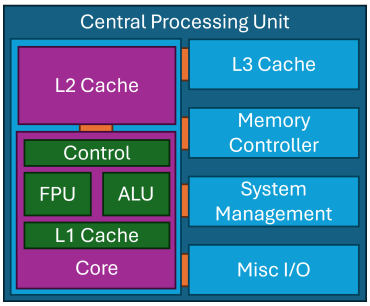

Figure 1.1: CPU Diagram Image

1.1 | Motivation

With the development of artificial intelligence, the demand for faster chips has never been greater. This is driven by the insight that larger datasets and increased computational power enable more capable machine-learning models (Kaplan et al., 2020). In modern semiconductor manufacturing, companies such as ASML and their customers are facing significant challenges in further scaling down transistor size while maintaining development cost, power efficiency, and production yields (Heyman, 2024; Zhang et al., 2024). Historically, these advancements have been described by Moore's Law (Moore, 1998).

One of the issues is the asymmetric scaling between different components. For example, static random-access memory (SRAM) has not scaled at the same rate as logic transistors (Sadaf et al., 2025). This poses a bottleneck on further development, as SRAM occupies a larger and larger area of the total die area. To address these challenges, the semiconductor industry is exploring alternative approaches to integrating a central processing unit (CPU) with random-access memory (RAM). One approach chip design companies started developing is the use of three-dimensional integration and multi-chiplet architectures, which enable vertical stacking of logic and memory layers (Jeong et al., 2022; Zhang et al., 2024). This brings memory closer to the computation, which is key in increasing performance (Sadaf et al., 2025; Wong & Salahuddin, 2015). An example is AMD's 3D V-Cache technology, which vertically stacks additional cache memory on the compute die. These innovations effectively increase on-chip cache capacity and performance without the need for further planar scaling of SRAM cells (Agarwal et al., 2022; Wuu et al., 2022). However, they also introduce new engineering challenges, including thermal management, greater manufacturing complexity, and potential impacts on production yield and cost.

In this research, we examine several chip architectures and compare them in terms of fabrication feasibility, computational performance, cost, energy consumption, and thermal characteristics. Our primary focus is the relationship between CPUs and RAM, and how different integration approaches influence overall system behavior. To structure this comparison, we employ a Design Structure Matrix (DSM) to analyze the interactions, dependencies, and trade-offs between these architectures (Eppinger & Browning, 2012).

1.2 | Scope and Objectives

The project compares multiple (3D) CPU architectures to identify which of them are promising for future development when further logic down-scaling is no longer possible (i.e. to continue Moore's Law) by modeling and evaluating interdependencies using the DSM approach.

2 | Case Description (1 page)

2.1 | System Function and Characteristics

Terminology, main function(s) of the case system. Explain using a picture the working of the system, and the relevant system characteristics.

2.2 | Subsystems and System Performance Parameters

Based on a standard CPU architecture, its function has been divided into five subsystems.

2.2.1 | Core-Cache Interconnect System

2.2.2 | CPU Cores

2.2.3 | Power Distribution System

2.2.4 | SRAM Cache

2.2.5 | Thermal Dissipation System

Which aspects of system performance or behavior are considered in your study. Outline which question about the case needs to be answered. List every parameter variable (Area, Temperature, Conductivity etc.)?

3 | DSM modeling method (3 pages)

This section presents the Design Structure Matrix (DSM)-based modeling methods used to address the research question of how the architectural design of CPU-Cache integration influences performance, power, and thermal behavior by decomposing the CPU architecture into parameters across key components. The DSM provides a structured representation of the interdependencies between the design parameters. Modelling of these couplings and their weights enables a structured analysis of the considered architectures.

3.1 | System Decomposition & Parameter Selection

(tycho)
List the DSM elements (parameters), the modules they belong to (core, cache, interconnect, power, thermal), and the system boundary (what and why things are included)

3.2 | Dependency Quantification & Weighting

The weighting scheme we have used and how the values have been assigned

3.3 | DSM Analysis Techniques

(tycho)
What analysis methods will we use to analyse the DSM

3.4 | Assumptions & Methodological Limitations

Small text about the assumptions (weighting scheme is the major one, I think)

4 | Results (3 pages)

5 | Conclusion (0.5 page)

Acknowledgements

References

Agarwal, R., Cheng, P., Shah, P., Wilkerson, B., Swaminathan, R., Wuu, J., & Mandalapu, C. (2022). 3d packaging for heterogeneous integration. *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*, 1103–1107. https://doi.org/10.1109/ECTC51906.2022.00178

Eppinger, S. D., & Browning, T. R. (2012, May). *Design structure matrix methods and applications*. The MIT Press. https://doi.org/10.7551/mitpress/8896.001.0001

Heyman, K. (2024). *Sram scaling issues, and what comes next*. Retrieved November 25, 2025, from https://semiengineering.com/sram-scaling-issues-and-what-comes-next/

Jeong, J., Geum, D.-M., & Kim, S. (2022). Heterogeneous and monolithic 3d integration technology for mixed-signal ics [3013]. *Electronics*, *11*(19), 3013. https://doi.org/10.3390/electronics11193013

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. https://arxiv.org/abs/2001.08361

Moore, G. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE*, *86*(1), 82–85. https://doi.org/10.1109/JPROC.1998.658762

Sadaf, M. U. K., Chen, Z., Subbulakshmi Radhakrishnan, S., Sun, Y., Ding, L., Graves, A. R., Yang, Y., Redwing, J. M., & Das, S. (2025). Enabling static random-access memory cell scaling with monolithic 3d integration of 2d field-effect transistors [4879]. *Nature communications*, *16*(1), 4879. https://doi.org/10.1038/s41467-025-59993-8

Wong, H.-S. P., & Salahuddin, S. (2015). Memory leads the way to better computing [191]. *Nature nanotechnology*, *10*(3), 191–4. https://doi.org/10.1038/nnano.2015.29

Wuu, J., Agarwal, R., Ciraula, M., Dietz, C., Johnson, B., Johnson, D., Schreiber, R., Swaminathan, R., Walker, W., & Naffziger, S. (2022). 3d v-cache: The implementation of a hybrid-bonded 64mb stacked cache for a 7nm x86-64 cpu. *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, *65*, 428–429. https://doi.org/10.1109/ISSCC42614.2022.9731565

Zhang, Q., Zhang, Y., Luo, Y., & Yin, H. (2024). New structure transistors for advanced technology node cmos ics. *National Science Review*, *11*(3), nwae008. https://doi.org/10.1093/nsr/nwae008

A | Model Relations & Parameter Weights

Table A.1: Cache Models & Weights

Model	Parameter	Weight	Foundation
CacheTransistorModel			
Requiring	capacity	1	
	associativity	0.1	
	block-size	0	
Returning	number-of-transistors		
AreaModel			
Requiring	transistors	1	
	process-node-factor	-1	
Returning	die-area		
CacheWireLengthModel			
Requiring	die-area	0.5	
	number-of-banks	-0.5	
	process-node-factor	-1	
Returning	internal-wire-length		
CacheAccessTimeModel			
Requiring	capacity	0.3	
	associativity	0.5	
	internal-wire-length	0.5	
	voltage	-1	
	temperature	0.3	
	process-node-factor	-1	
Returning	access-time		
FrequencyModel			
Requiring	voltage	1	
	temperature	-0.5	
	process-node-factor	-1	
Returning	frequency		
CacheHitRateModel			
Requiring	capacity	0.7	
	associativity	0.3	
	block-size	0.15	
Returning	hit-rate		
CacheAMATModel			
Requiring	access-time	1	
	hit-rate	-1	
	miss-penalty	1	
Returning	latency		
BandwidthModel			
Requiring	bus-width	1	
	clock-frequency	1	
Returning	bandwidth		
CurrentModel			
Requiring	transistors	1	
	process-node-factor	-0.5	
	temperature	2	
	voltage	1	
	clock-frequency	1	
Returning	dynamic-current		

Table A.2: Cache Models & Weights (continued)

Model	Parameter	Weight	Foundation
CurrentModel			
Requiring	transistors	1	
	process-node-factor	-0.5	
	temperature	2	
	voltage	1	
	clock-frequency	1	
Returning	leakage-current		
PowerModel			
Requiring	voltage	1	
	dynamic-current	1	
Returning	dynamic-power		
PowerModel			
Requiring	voltage	1	
	leakage-current	1	
Returning	leakage-power		
TotalPowerModel			
Requiring	dynamic-power	1	
	leakage-power	1	
Returning	power-consumption		
PowerThermalModel			
Requiring	power-consumption	1	
Returning	thermal-load		
PowerConsumptionRelation			
Requiring	cache-cooling-capability	1	
	cache-power-capability	-0.5	
Returning	max-power-consumption		
TemperatureModel			
Requiring	thermal-load	1	
	thermal-resistance	-0.5	
	ambient-temperature	2	
Returning	temperature		

Table A.3: Core Models & Weights

Model	Parameter	Weight	Foundation
CurrentModel			
Requiring	transistors	1	
	process-node-factor	-0.5	
	temperature	2	
	voltage	1	
	clock-frequency	1	
Returning	current		
PowerModel			
Requiring	voltage	1	
	current	1	
Returning	power-consumption		
AreaModel			
Requiring	transistors	1	
	process-node-factor	-1	
Returning	die-area		
FrequencyModel			
Requiring	voltage	1	
	temperature	-0.5	
	process-node-factor	-1	
Returning	frequency		
PowerThermalModel			
Requiring	power-consumption	1	
Returning	thermal-load		
PowerConsumptionRelation			
Requiring	cooling-capability	1	
	power-capability	1	
Returning	max-power-consumption		
TemperatureModel			
Requiring	thermal-load	1	
	thermal-resistance	1	
	ambient-temperature	1	
Returning	temperature		
PerformanceModel			
Requiring	clock-frequency	0.4	
	voltage	0.2	
	process-node-factor	-0.2	
	transistors	0.1	
	cache-latency	-0.3	
	interconnect-latency	-0.2	
Returning	computational-performance		
	required-cache-bandwidth		

Table A.4: Interconnect Models & Weights

Model	Parameter	Weight	Foundation
BandwidthModel			
Requiring	bus-width	1	
	frequency	1	
Returning	bandwidth		
InterconnectAreaModel			
Requiring	length	1	
	bus-width	1	
	process-node-factor	-1	
Returning	die-area		
InterconnectLatencyModel			
Requiring	length	1	
	bandwidth	-1	
	frequency	-1	
Returning	latency		
InterconnectLengthConnector			
Requiring	length-2D	1	
	length-3D	1	
Returning	length		
InterconnectLengthModel2D			
Requiring	total-cpu-die-area	0.5	
	process-node-factor	-1	
Returning	length-3D		
InterconnectLengthModel3D			
Requiring	total-cpu-die-area	0.25	
	process-node-factor	-1	
Returning	length-3D		

Table A.5: Power Distribution System Models & Weights

Model	Parameter	Weight	Foundation
TSVGeometryModel			
Requiring	cache-die-area	1	
	cache-tsv-density	1	
Returning	cache-tsv-count		
TSVGeometryModel			
Requiring	core-die-area	1	
	core-tsv-density	1	
Returning	core-tsv-count		
PowerModel			
Requiring	cache-voltage	1	
	cache-current-capability	1	
Returning	cache-power-capability		
PowerModel			
Requiring	core-voltage	1	
	core-current-capability	1	
Returning	core-power-capability		
TotalPowerModel			
Requiring	core-power-capability	1	
	cache-power-capability	1	
Returning	power-capability		
TSVElectricalModel			
Requiring	cache-tsv-count	1	
	cache-tsv-diameter	2	
	cache-tsv-length	0.5	
	cache-max-temp	0.5	
	tsv-thermal-conductivity	-1	
	tsv-resistivity	-0.5	
	tim-resistance	-0.5	
	contact-resistance	-0.5	
Returning	cache-current-capability		
TSVElectricalModel			
Requiring	core-tsv-count	1	
	core-tsv-diameter	2	
	core-tsv-length	0.5	
	core-max-temp	0.5	
	tsv-thermal-conductivity	-1	
	tsv-resistivity	-0.5	
	tim-resistance	-0.5	
	contact-resistance	-0.5	
Returning	core-current-capability		

Table A.6: Thermal Dissipation System Models & Weights

Model	Parameter	Weight	Foundation
ThermalCoolingCapabilityModel			
Requiring	coolant-flowrate	1	
	coolant-inlet-temperature	-1	
	coolant-outlet-temperature	1	
	coolant-specific-hear	1	
Returning	cooling-capability		
ThermalCapabilityModel			
Requiring	cooling-capability	1	
	core-cooler-resistance	-1	
	cache-cooler-resistance	-1	
Returning	core-cooling-capability		
	cache-cooling-capability		
ThermalResistanceModel			
Requiring	core-die-area	-1	
	core-tsv-density	-1	
	core-tsv-diameter	-1	
	tsv-conductivity	-1	
	tim-resistance	1	
Returning	core-thermal-resistance		
ThermalResistanceModel			
Requiring	cache-die-area	-1	
	cache-tsv-density	-1	
	cache-tsv-diameter	-1	
	tsv-conductivity	-1	
	tim-resistance	1	
Returning	cache-thermal-resistance		
ThermalTimResistanceModel			
Requiring	total-die-area	-1	
	tim-thickness	1	
	tim-conductivity	-1	
	contact-resistance	1	
Returning	tim-resistance		

Table A.7: System Models & Weights

Model	Parameter	Weight	Foundation
TotalPowerModel			
Requiring	core-power-consumption	1	
	cache-power-consumption	1	
Returning	power-consumption		
EfficiencyModel			
Requiring	power-consumption	-1	
	core-computational-performance	1	
Returning	efficiency		
TotalDieAreaModel			
Requiring	core-die-area	1	
	cache-die-area	1	
	interconnect-die-area	1	
Returning	die-area		