

# CPU with integrated SRAM Architecture Comparison ASML - Project Plan

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

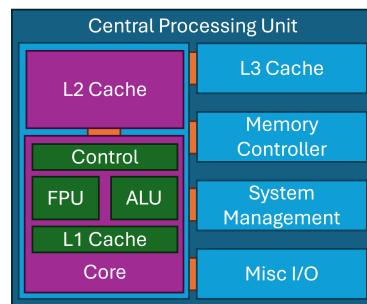
Eindhoven, January 6, 2026

## **Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Scope and Objectives . . . . .	1
<b>2</b>	<b>Literature Review</b>	<b>2</b>
2.1	CPU Architecture Evaluation Methods . . . . .	2
2.2	Related DSM Models . . . . .	2
<b>3</b>	<b>Methodology</b>	<b>3</b>
3.1	Key Components . . . . .	4
3.2	Dependencies . . . . .	5
3.3	DSM Analysis . . . . .	6
<b>4</b>	<b>Execution Plan</b>	<b>7</b>
4.1	Information sources . . . . .	7
4.2	Time limitation . . . . .	7
4.3	Schedule and task division . . . . .	7
	<b>References</b>	<b>8</b>
<b>A</b>	<b>Appendix Methodology</b>	<b>9</b>

## 1 | Introduction

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 | Literature Review

### 2.1 | CPU Architecture Evaluation Methods

In the literature, two dominant methods stand out for evaluating different CPU architectures: simulation and Physical Power, Performance, and Area analysis (PPA-analysis).

#### 2.1.1 | Simulation

In a simulation-based evaluation, a CPU or one of its subsystems is modeled, and computational tools are used to estimate performance, power, and other design metrics. Reed et al. (2005) compared a conventional 2D on-chip data cache with a 3D-stacked equivalent. In their approach, the 2D cache layout was “folded” into two layers so that distant regions became vertically aligned. Their evaluation focused on the area, power, and performance of the cache subsystem. However, their study did not include thermal or power-delivery analysis, both of which are essential when assessing CPU architectures.

A more recent study by Zhu et al. (2021), in contrast, simulated full processor designs and incorporated thermal modeling as well as power-integrity analysis. Their simulations evaluated a broader collection of metrics, including performance, power, area, temperature, voltage-drop integrity, interconnect behavior, and memory-access characteristics.

Together, these studies demonstrate how simulation can quantify key aspects of CPU architectures, making architectural comparisons possible. They also highlight the architectural characteristics of CPUs that matter when evaluating different designs.

#### 2.1.2 | Physical PPA-analysis

Physical PPA analysis evaluates CPU architectures using an actual physical implementation rather than relying solely on simulations. In this approach, the CPU is physically implemented, after which Electronic Design Automation (EDA) tools are used to measure the power consumption and performance. For example, Kim et al. (2024) implemented a CPU in three different 3D integration configurations (monolithic 3D, hybrid wafer bonding, and microbump stacking) and compared each variant’s PPA against a traditional 2D layout. This illustrates how physical PPA-analysis enables a direct comparison between different CPU architectures.

### 2.2 | Related DSM Models

Chip integration efforts have been subject to a decomposition analysis. For example, integrating driver chips, LED’s and sensors into a single semiconductor (de Borst et al., 2016). This integration introduced similar effects to heat conduction. We also need to analyze each parameter, how it relates to other parameters based on physics, and how we can optimize them. Therefore, a multi-objective optimization approach can be helpful. This has been utilized by (Xu et al., 2025) to determine an optimized design for an integrated micro pin-fin channel cooling that incorporates Through Silicon Vias (TSVs). Although no DSM is used in this process, it appears to be useful for our use case, as our goal is to find an optimal chip architecture with respect to multiple objectives or parameters.

The mentioned papers try to find the optimal structure (Xu et al., 2025) and the importance of each parameter (de Borst et al., 2016) as well as the underlying physical relations governing the system.

#### 2.2.1 | Problem solving methods

The paper by (de Borst et al., 2016) uses a four-step procedure:

1. Specify  $x, r, F$  interactions using the  $\Psi$  language
2. Generate multidisciplinary DSM containing  $x, r, F$  interactions automatically from the specification
3. Partition the matrix to find strongly coupled parts
4. Find order with minimal number of feedback interactions via sequencing of the partitions

The  $\Psi$  Specification Language consists of elements that describe the relation between different other elements. Because the variables and their interactions are specified, this approach allows for an automatically generated matrix. Partitioning refers to the identification of strongly coupled elements in a matrix. The goal is to minimize the size of these partitions and the number of interactions between partitions. Tools used for this approach include Thebeau's MATLAB DSM partitioning algorithm and the Graclus algorithm. Sequencing rearranges the rows and columns of the matrix so that the amount of feedback coupling between partitions is minimized because any feedback coupling requires iterations of the design. The Dulmage-Mendelsohn decomposition algorithm was used for this (de Borst et al., 2016).

The approach of (Xu et al., 2025) uses two different methods to find an optimal solution based on the given inputs with respect to different objectives that might conflict with each other. The procedure is as follows:

1. Determine properties as variables (factors) and define performance parameters
2. Run RSM based on those parameters by selecting specific/strategic combinations of values for the parameters
3. Insert the found functions into NSGA-II to find the best trade-off solution for conflicting objectives

The Response Surface Methodology (RSM) runs small but specified (CFD) simulations instead of one large simulation to save time. The variables suspected to change are chosen, as well as the target performance parameters. Running a specified simulation means selecting strategic combinations of variables using a Central Composite Design method. Using regression to fit mathematical equations (approximations) are then used for the Non-dominated Sorting Genetic Algorithm II (NSGA-II). This algorithm tries to find the best trade-offs given conflicting objectives based on a random population of designs. It uses non-dominated sorting or the Pareto optimal, which means that no solution beats it in both categories.

### 2.2.2 | Outcome & Relevance

According to (de Borst et al., 2016), the multidisciplinary coupling structure could be successfully modeled using a DSM. The  $\Psi$  specification language showed advantages for large-scale applications. The work of (Xu et al., 2025) returned results indicating which parameters have what influence on the desired performance indicators.

The goal of our project is to relate different parameters of components to each other with respect to the used chip architecture. We compare conventional monolithic CPUs with 3D Stacked CPUs. In detail, we inspect the possibility of placing the cache above or underneath the core (3D stacking). This 3-dimensional integration of components introduces new relations between the core and the cache, for example, heat generation and conduction, bandwidth, power supply, but also non-physical relations such as cost and yield. The method used by (de Borst et al., 2016) is helpful because it shows how to use a DSM approach for a semiconductor product. The applied algorithm optimizes the matrix with respect to the partitions/relations of the different elements, but not the physical properties governing them. Therefore, the work of (Xu et al., 2025) is relevant for our case in terms of how to decompose a system and how to define the parameters and governing equations that represent the physical relations between the parameters based on desired performance indicators.

## 3 | Methodology

This methodology describes how the technical relationships between the CPU core and an integrated SRAM are modeled for the purpose of comparing different integration architectures. The focus for the model is put on the key components for this interaction: the core, SRAM cache, the core-cache interconnect, the thermal dissipation system, and the power distribution system. These components are selected because they govern the key trade-offs in the 3D integration, including computational performance, energy efficiency, and manufacturability.

The subsystems form a part-whole hierarchy based on the physical components of a CPU design as visualized in [Figure A.1](#). For each of these subsystems, system variables, including design and derived parameters, have been selected that are relevant for the CPU-cache interaction. This creates a component

DSM and a multi-domain parameter DSM that reveal system parameter couplings and integration bottlenecks.

### 3.1 | Key Components

Each component is further defined using parameters, which form the DSM elements required for the dependency analysis. For each parameter, the reason for inclusion is shortly justified in terms of the system performance, energy efficiency, and/or manufacturability.

#### 3.1.1 | Core

The core represents the compute engine responsible for instruction execution in the CPU. Its properties directly impact the performance and power efficiency of the system.

Parameter	Rationale
Core Max Temperature	Limits maximum voltage
Core Voltage	Higher voltage boosts performance but increases power and thermal load
Core Max Clock Frequency	With a higher clockspeed, more computations can be done in the same time, increasing performance
Core Computational Performance	System-level performance indicator
Core Data Processing Speed	Determines demand for cache bandwidth to the core
Core Process Node	Smaller nodes improve performance and power efficiency but increase fabrication cost and defect risk
Core Power Consumption	Defines cooling requirements and energy efficiency
Core Die Area	Larger area reduces yield

#### 3.1.2 | SRAM Cache

The cache is critical to reducing memory access latency and increasing computational performance. Cache capacity, placement, and performance define performance-per-watt of the system and trade-offs in 3D architectures.

Parameter	Rationale
Cache Size	Larger caches increase hit rate, reducing the number of times slower memory needs to be accessed, decreasing the wait time on data needed for computations
Cache Die Area	Large die area reduces yield and increases cost
Cache Voltage	Reducing voltage reduces the power consumption and thermal load on the system
Cache Bandwidth	Determines how fast the core can be fed with data needed for computations
Cache Hit Rate	Increased hit rate reduces the number of times slower memory needs to be accessed, decreasing the wait time on data needed for computations
Cache Power Usage	Defines cooling requirements and energy efficiency
Cache Process node	Smaller nodes improve performance and power efficiency but increase fabrication cost and defect risk
Cache Latency	Lower latency means the core has to wait shorter for the requested data needed for computations

#### 3.1.3 | Core-Cache Interconnect

Defines the quality of compute-memory coupling. In 3D chip architecture designs, this includes the through-silicon vias (TSVs) or Cu-Cu connections in the interface of the two stacked integrated circuits. In 2D, it includes on-die buses or interposers used to connect the integrated circuits.

Parameter	Rationale
Interconnect Length	Longer distance increases latency of the interconnect, increasing the time the core has to wait on data needed for computations
Interconnect Bandwidth	Determines how fast the core can be fed with data needed for computations
Interconnect Bus Width	Wider buses increase bandwidth but add routing complexity
Interconnect Latency	Lower latency means the core has to wait for shorter on the requested data needed for computations

### 3.1.4 | Power Distribution System

Vertical stacking increases the complexity of power circuit routing and requires robust power delivery networks to prevent IR-drop.

Parameter	Rationale
Core Power TSV Density	More TSVs increases the power that can be provided to the core
Cache Power TSV Density	More TSVs increases the power that can be provided to the cache
Total Power Capability	Sets a maximum on the core and cache power consumption

### 3.1.5 | Thermal Dissipation System

Thermal behavior becomes a constraint in 3D integrated architectures, where stacked memory increases heat density and impacts heat resistance to the cooler of the stacked components.

Parameter	Rationale
Core Thermal TSV Density	Improves vertical cooling, allowing for more power consumption while not increasing temperature in the core
Cache Thermal TSV Density	Improves vertical cooling, allowing for more power consumption while not increasing temperature in the cache
Core-to-Cooler Thermal Resistance	Lower thermal resistance from the core to the cooler interface allows for more power consumption while not increasing temperature in the core
Cache-to-Cooler Thermal Resistance	Lower thermal resistance from the cache to the cooler interface allows for more power consumption while not increasing temperature in the cache

## 3.2 | Dependencies

Before a DSM can be created, the dependencies between the previously defined components need to be elaborated on. In addition to the standard component DSM, a multi-domain parameter DSM will be necessary to fully represent the CPU SRAM architecture dependencies.

To link the previously defined component parameters to one another, three specific dependency types (each indicated by a color in the DSM) will be used that will capture the full scope of the architectures:

- Physical dependencies (red): This dependency type will be assigned to parameters that are coupled by fundamental physics laws, for example, thermodynamics or Ohm's law. Including this dependency type is required to account for new physical couplings created by 3D stacking. An example of a physical coupling is how the heat from the core directly affects the thermal resistance of the cache stacked on top.
- Functional dependencies (orange): This dependency type will be assigned to parameters that are coupled by data flow or performance metrics, like AMAT equations. It is important to include this to show how architectural choices, like cache size, affect the performance of the system.
- Constraint dependencies (yellow): This dependency type will be assigned to parameters that set a hard limit or budget another parameter. Limiting parameter relations (e.g., Area limits, Power budgets) are required to ensure manufacturability and reliability of the system.

There is a dependency between parameter A (input/column) and B (output/row), when a change in A requires a calculation or adjustment to B. With governing equations found in literature, the following dependency rules can be derived:

- Governing equation rule: This rule says that if a standard engineering formula exists (e.g.,  $P = CV^2 f$ ) to link two parameters, a dependency can be concluded. Frequency and core voltage are, for example, inputs to the core power consumption.
- Geometric Constraint rule: This rule says that if two components share a physical interface in the 3D stack, then their dimensional parameters will affect each other. An example of this is the core die area, which is a limiting input for the cache die area and vice versa, because vertical alignment is crucial.
- Operational limit rule: This rule says that if a parameter functions as a safety threshold, then it affects the operating parameters. An example is how the max temperature affects the core frequency, because it must be capped to ensure the thermal limits.

To increase the analytical value of the DSM, a "strength attribute" can be assigned to each dependency. This is done by assigning a numerical value (1, 2, or 3) to each dependency in the DSM, which represents how sensitive one parameter is to another:

- Weight 1 (Weak): The effect of the input is minor. It could be a second-order effect or a loose constraint.
- Weight 2 (Moderate): The effect of the input is significant. It could be a linear factor or a primary performance driver.
- Weight 3 (Critical/Strong): The effect of the input is dominant. This could be due to a nonlinear relationship, or it could be the dominant term in the relationship.

### 3.3 | DSM Analysis

This section explains how analyzing the populated DSM will provide useful insights into the CPU-SRAM architectures. Attributes will be assigned to all dependencies, allowing the previously established qualitative system knowledge to be converted into a quantitative model. Following the methodology of de Borst et al., 2016, the analysis begins with matrix construction by populating dependencies according to the rules established, assigning each a type (physical, functional, or constraint) and weight (1-3).

	1	2	3	4	5	6	7	8
Cache Size	1	2		3	2	2		
Cache Die Area	2	2				2	2	
Cache Voltage	3			2		2	1	2
Cache Bandwidth	4		2					1
Cache Hit Rate	5	3						
Cache Power Usage	6	2	2	2			2	
Cache Process Node	7	2	2	1		2		
Cache Latency	8		2	2	1			

Figure 3.1: DSM Analysis Cache

To illustrate this methodology, consider the dependency between Cache Size and Cache Hit Rate (1-5) shown in Figure 3.1. This functional dependency receives a critical weight of 3 because larger caches store more data locally, increasing the probability of cache hits rather than retrieving the data from a slower data store. It is assigned a weight of 3, because this dependency is non-linear. A larger cache thus results in higher computational performance. However, a larger cache means the die area of the cache needs to be bigger (1-2), which in turn increases latency due to the physically longer path traces (2-8), decreasing performance due to a longer wait time on critical data needed for computations. This results in a compromise of a larger cache and lower latency, where each CPU-cache architecture might have slightly different performance implications for these dependencies.

In 2D monolithic architectures, this trade-off is particularly pronounced because cache area directly competes with core area on the same die plane, and longer horizontal interconnects significantly increase access latency. In contrast, in 3D stacked architectures, the area constraint is relaxed through vertical integration, allowing larger caches without expanding the chip footprint. However, 3D designs introduce new complications, as the stacked cache, for example, affects the Core-to-Cooler Thermal Resistance, potentially limiting the core power consumption. By analyzing these dependency chains across architectures, the DSM reveals which integration approach offers the most favorable balance between cache performance, physical constraints, and thermal management for different application requirements.

## 4 | Execution Plan

### 4.1 | Information sources

To complete this project, it will be essential to be selective in deciding where time is spent. CPU design and architecture are not part of the area of expertise of the team, so thorough research and reliable sources are important. Reliable sources of information about semiconductors are 'semiwiki.com' and 'semiengineering.com'. These websites provide extensive insights into current developments in semiconductor technology. In terms of formal research papers, IEEE Explore is a reliable source of information where most semiconductor-related research is published. Our customer has already been very helpful with recommending information sources, and we plan to keep in close contact with them to show them what was found and welcome new input on further research.

### 4.2 | Time limitation

CPU design and architecture are subjects that are very diverse and complex. Semiconductors have been researched and refined for many decades, and a lot of information can be found. All of the 8 weeks of this project can be spent digging into the details of CPU technology. However, the purpose of this research is to gain a solid understanding of key concepts and influencing factors while maintaining a high-level perspective. The goal of the research is to understand what we are working on, which factors play a role, but not to get lost in the sea of information about chips. To prevent wasting time on unnecessary research, we have the following guidelines:

- Keep in close contact with the customer to ensure that the research direction and DSM design remain aligned with their expectations.
- Discuss findings weekly within the team to share insights, give feedback, and support each other in refining results.

If it turns out that certain information cannot be obtained within the time frame of the course, it will be important to communicate this clearly to the customer. As the time for this project is limited, adjusting the project scope is the main factor that can be adjusted if necessary. By narrowing down to a more specific project goal, quality can be ensured. By keeping the communications lines short with the customer, it can be ensured that priorities are set correctly, and they are satisfied with the result.

### 4.3 | Schedule and task division

The schedule for this project has been visualized as a Gantt chart, as can be seen in [Figure 4.1](#). Tasks are divided weekly, based on project progress and customer feedback. To ensure quality and invite ownership, key responsibilities of the project have been divided among the team. Tycho is the final editor of the Design Structure Matrix. Philip is the final editor of the report. Romke is the chair of customer meetings. Floris takes minutes during customer meetings. Dean is the final editor for the presentations.

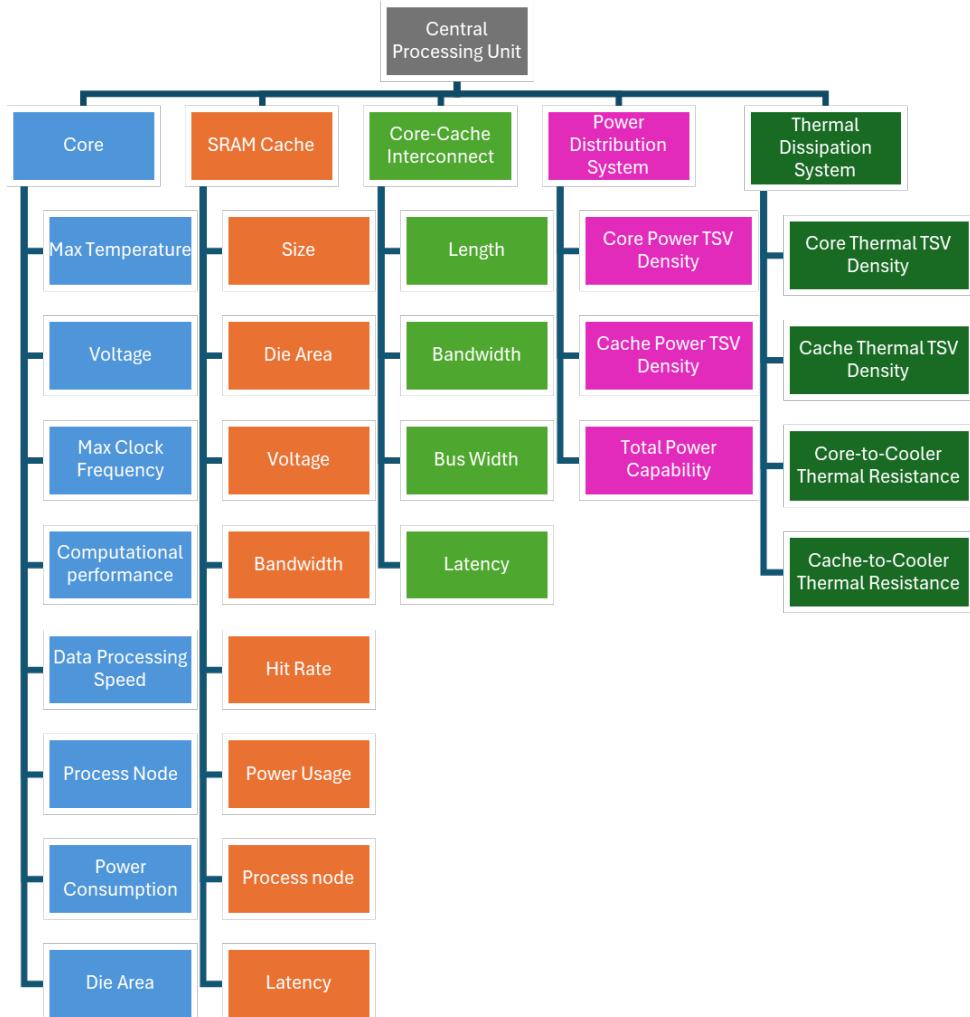


**Figure 4.1:** Gantt chart for project time division

## 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>
- de Borst, E., Etman, L., Gielen, A., Hofkamp, A., & Rooda, J. (2016). Decomposition analysis of the multidisciplinary coupling in led system-in-package design using a dsm and a specification language. *Structural and Multidisciplinary Optimization*, 53, 1395–1411. <https://doi.org/10.1007/s00158-016-1397-2>
- 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>
- Kim, J., Zhu, L., Torun, H. M., Swaminathan, M., & Lim, S. K. (2024). A ppa study for heterogeneous 3-d ic options: Monolithic, hybrid bonding, and microbumping. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 32(3), 401–412. <https://doi.org/10.1109/TVLSI.2023.3342734>
- 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>
- Reed, P., Yeung, G., & Black, B. (2005). Design aspects of a microprocessor data cache using 3d die interconnect technology. *2005 International Conference on Integrated Circuit Design and Technology, 2005. ICICDT 2005.*, 15–18. <https://doi.org/10.1109/icicdt.2005.1502578>
- 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>
- Xu, S., Zhang, Y., Li, Q., & Chen, X. (2025). Multi-physical field coupling effect in micro pin-fin channel cooling with coaxial-like through-silicon via (tsv) for three-dimensional integrated chip (3d-ic). *Applied Thermal Engineering*, 258, 124815. <https://doi.org/https://doi.org/10.1016/j.applthermaleng.2024.124815>
- 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>
- Zhu, L., Bamberg, L., Pentapati, S. S. K., Chang, K., Catthoor, F., Milojevic, D., Komalan, M., Cline, B., Sinha, S., Xu, X., Garcia-Ortiz, A., & Lim, S. K. (2021). High-performance logic-on-memory monolithic 3-d ic designs for arm cortex-a processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(6), 1152–1163. <https://doi.org/10.1109/TVLSI.2021.3073070>

## A | Appendix Methodology



**Figure A.1:** Hierarchical tree diagram of CPU components.