

The Chip of Theseus : To Be or Not To Be General Purpose

Nitesh Narayana GS¹, Abhijit Das², Sanyam Mehta³, and Xavier Martorell¹

¹UPC Barcelona, ³Oasis Research and Software, ²IIT Hyderabad

1 INTRODUCTION

Around two thousand years ago, Plutarch asked a question that still refuses to leave philosophy classes: if every plank of the ship of Theseus is replaced, one by one, is it still the same ship [18]? Centuries later, Hobbes made things worse by asking what happens if someone quietly collects the discarded planks and builds a *second* ship out of them [12]. Which one is the real ship of Theseus?

Computer architects have been running this experiment for over fifty years, only with silicon instead of timber. The simple in-order scalar core of the early 1970s [9] has been patched, extended, and rebuilt every time a new class of applications hit it: ILP, floating point, multimedia, graphics, AI, and finally the energy wall left behind by the end of Dennard Scaling [7, 8]. Every obstacle was answered the same way: *add something new to the chip*. As the “general-purpose” chip gathers more and more special-purpose crew, a wild but surprisingly serious question appears: *What is it, to be or not to be general purpose*¹?

We present the *Chip of Theseus* , a story-cum-provocation that retraces this journey plank by plank, looks at the old planks we left on the shore, and asks which of the resulting chips deserves the name. The subsequent sections describe the journey, the three ships, and the question it leaves behind.

2 THE JOURNEY TILL NOW

Our voyage starts with the **Original One**: a simple in-order scalar core [9], with nothing more than a fetch/decode unit, an ALU, and an execution context (Figure 3). It issues at most one instruction per cycle, and it was perfectly happy doing so, until the situations changed. We follow this one chip through the five obstacles it met on its voyage, where each new obstacle exposed something the previous version of the chip could not do, and each fix became a new plank on the ship. Figure 1 charts how many distinct kinds of units ended up on board (our own tally of the milestones below), and where the ship is heading by 2050.

Floating Point: The first storm was numerical. Scientific codes spend their time on real numbers, which an integer-only core emulates slowly, so a floating-point unit moved on-die [6]. *The chip was no longer only an integer machine.*

Larger, Complex Programs: Next came software itself. As programs grew larger and more complex, a core issuing at most one instruction per cycle left most of its pipeline idle, and the demand shifted to higher pipeline utilisation and *instruction-level parallelism*. The chip answered with superscalar issue [4], followed by out-of-order and speculative execution with branch prediction [16]. The simple in-order scalar planks that made this possible were no longer needed, and were set aside on the shore.

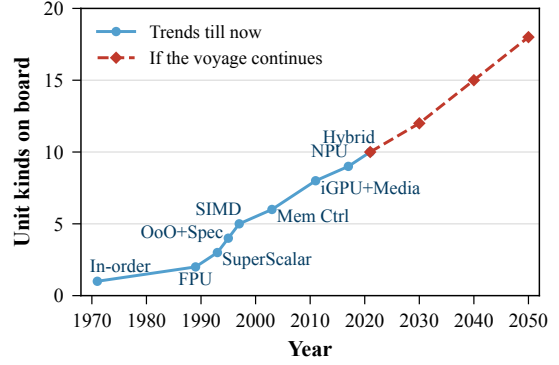


Figure 1: The crew of the chip of Theseus towards 2050.

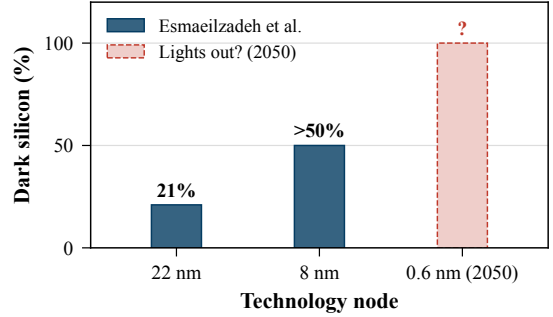


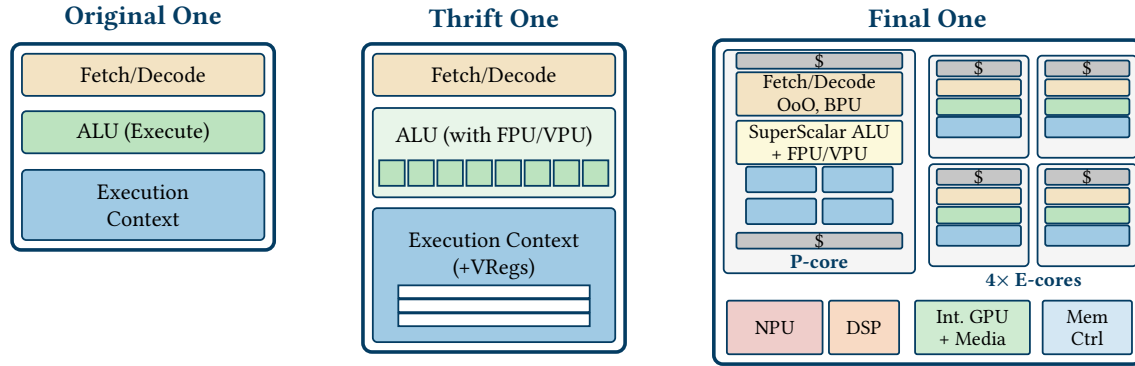
Figure 2: Dark silicon: the energy foe eating the chip.

Real-Time Audio and Video: Then the workloads became interactive. Signal processing and graphics brought the third storm, with hard deadlines and streams of data that a general core handled poorly. Vector/SIMD units answered first [17], and were later strengthened alongside DSP instructions, media engines, and integrated graphics [22]. All these data-hungry units also made memory latency and bandwidth the bottleneck, and the memory controller left the Northbridge and moved onto the chip [15]. *How much of the chip is still the “CPU” now?*

The New Age of AI: The answer got smaller with the next storm. A tornado of AI/ML applications hit the ship, and the existing units were too inefficient for its matrix-heavy kernels, so a neural processing unit was integrated [5]. *Another specialist, another plank that computes one thing very well.*

The Energy Foe: The planks kept piling on, but the power budget did not grow with them. The biggest challenge of all arrived with the end of Dennard Scaling [7]: transistors kept shrinking, but voltage stopped scaling with them, so a chip could no longer power all of its transistors, and thus all of its specialised units, at once. With 21% of a chip dark at 22 nm and more than half projected dark

¹With apologies to Hamlet [20].



Looks like a simple GP-GPU core!

Figure 3: The three ships, seen through a different lens.

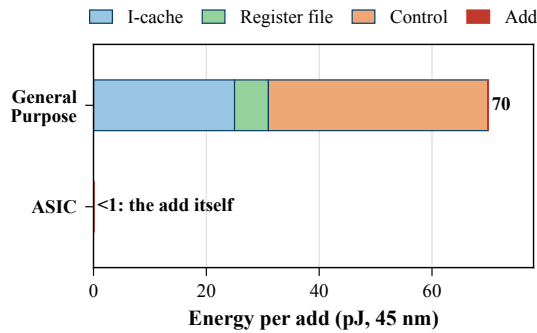


Figure 4: The Turing Tariff of a single add, using Horowitz's 45 nm energy estimates [13].

at 8 nm [8] (Figure 2), the chip evolved into a hybrid of performance and efficiency cores [19], and had to become *smart* just to manage its own other kinds of heterogeneity as well. *Is this still a general-purpose chip, or a fleet in disguise?*

3 WHICH IS THE CHIP OF THESEUS?

The journey leaves us with three ships (Figure 3). The **Original One**: a simple in-order core with a fetch/decode unit, an ALU, and an execution context. The **Final One**: the Original One after the whole voyage, carrying OoO, superscalar and branch predictors, a better FPU/VPU, DSP units and an integrated GPU, a memory controller, an NPU, and a hybrid core arrangement. And, following Hobbes, the **Thrift One**: built only from the planks we left on the shore, the in-order fetch/decode and the FPU/VPU lanes.

Seen through a different lens, the Thrift One is, well, a simple GP-GPU core: an in-order front end feeding many simple FP/vector lanes. In other words, the planks we discarded along the journey quietly reassembled themselves into the engine that powers today's AI boom [10]. Meanwhile, every new plank added to the Final One raises an uncomfortable question: *Is the energy cost of such ad hoc integration worth it?*

The Turing Tariff: Turing showed that one universal machine can compute any function [21], but never promised that it would do

so efficiently. Paul Kelly calls this price of universality the *Turing Tariff* [1]. Figure 4 makes it concrete: a general-purpose core spends around 70 pJ to execute one add instruction, of which the add itself costs about 0.1 pJ; the rest goes to instruction-cache access, register-file access, and control [13]. An ASIC pays only for the add. That is a tariff of hundreds of times, paid on every single instruction, by the only passenger who never gets off the ship.

The voyage does not end here. Three stretches of uncharted water lie ahead of the *Chip of Theseus*. We hand the compass to the readers and leave the charting to them:

- *Is the Final One still the same chip?* Almost every plank has been replaced, even the ISA: x86 grew from x87 to MMX, SSE, AVX, x86-64, and AMX matrix extensions [14]. Yet the ISA only ever *adds* planks; the attempt to throw the oldest ones overboard, Intel's X86S, was abandoned [3]. Perhaps what keeps it the same chip is not any plank, but the promise that yesterday's software still sails.
- *Then which is the Chip of Theseus?* The Thrift One already rules the data-parallel world. The Final One rules everything else, at a tariff. The Original One remains the blueprint both were built from.
- *Do domain-specific cores make more sense?* If every new obstacle is answered with a new specialist anyway, perhaps the ship should openly become a fleet of accelerators [11], with the general-purpose core promoted to captain rather than oarsman.

4 CONCLUSION

The journey of the *Chip of Theseus* shows that every "general-purpose" chip has quietly become a crew of specialists, while the planks we threw away rebuilt themselves into the data-parallel engines of today. Therefore, looking back at this journey should not only shape how we design the next generation of chips, but also provoke a fundamental question: *What is it, to be or not to be general purpose?* Whether the next voyage is sailed by a single ship or a fleet of domain-specific cores² is a question we leave for you to answer.

²And hence, as a butterfly effect, everything that orchestrates them.

ACKNOWLEDGMENTS

We extend our sincere gratitude to the WACI initiative of ASPLOS for giving us a forum that welcomes ideas that are indeed wild and crazy! The processor figures are inspired by Prof. Kayvon Fatahalian’s teaching material [2], and the term *Turing Tariff* is borrowed from Prof. Paul Kelly [1].

REFERENCES

- [1] 2020. “Turing Tariff” Reduction: Architectures, Compilers and Languages. <https://www.doc.ic.ac.uk/~phjk/Presentations/2020-06-24-DoCLunch-PaulKelly-TuringTaxV04.pdf>.
- [2] 2023. CMU 15-418/618: Parallel Computer Architecture and Programming – A Modern Multi-Core Processor. <https://www.cs.cmu.edu/~15418/>.
- [3] 2024. Intel Abandons “x86S” Plans to Focus on The Regular x86-64 ISA Advisory Group. <https://www.techpowerup.com/330066/intel-abandons-x86s-plans-to-focus-on-the-regular-x86-64-isa-advisory-group>.
- [4] D. Alpert and D. Avnon. 1993. Architecture of the Pentium microprocessor. *IEEE Micro* 13, 3 (1993), 11–21. <https://doi.org/10.1109/40.216745>
- [5] Apple Inc. 2017. The future is here: iPhone X. <https://www.apple.com/newsroom/2017/09/the-future-is-here-iphone-x/>.
- [6] J.H. Crawford. 1990. The i486 CPU: executing instructions in one clock cycle. *IEEE Micro* 10, 1 (Feb. 1990), 27–36. <https://doi.org/10.1109/40.46766>
- [7] Robert H. Dennard, Fritz H. Gaensslen, Hwa-Nien Yu, V. Leo Rideout, Ernest Bassous, and Andre R. LeBlanc. 1974. Design of Ion-Implanted MOSFET’s with Very Small Physical Dimensions. *IEEE Journal of Solid-State Circuits* 9, 5 (1974), 256–268.
- [8] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. 2011. Dark Silicon and the End of Multicore Scaling. In *Proceedings of the International Symposium on Computer Architecture*. 365–376.
- [9] F. Faggin, M.E. Hoff, S. Mazor, and M. Shima. 1996. The history of the 4004. *IEEE Micro* 16, 6 (1996), 10–20. <https://doi.org/10.1109/40.546561>
- [10] John L. Hennessy and David A. Patterson. 2019. A New Golden Age for Computer Architecture. *Commun. ACM* 62, 2 (2019), 48–60.
- [11] Mark D. Hill and Vijay Janapa Reddi. 2021. Accelerator-Level Parallelism. *Commun. ACM* 64, 12 (2021), 36–38.
- [12] Thomas Hobbes. 1655. *De Corpore*. Andrew Crook.
- [13] Mark Horowitz. 2014. Computing’s Energy Problem (and What We Can Do About It). In *Proceedings of the IEEE International Solid-State Circuits Conference*. 10–14.
- [14] Intel Corporation 1978. *MCS-86 User’s Manual*. Intel Corporation.
- [15] C.N. Keltcher, K.J. McGrath, A. Ahmed, and P. Conway. 2003. The AMD opteron processor for multiprocessor servers. *IEEE Micro* 23, 2 (March 2003), 66–76. <https://doi.org/10.1109/mm.2003.1196116>
- [16] D.B. Papworth. 1996. Tuning the Pentium Pro microarchitecture. *IEEE Micro* 16, 2 (April 1996), 8–15. <https://doi.org/10.1109/40.491458>
- [17] A. Peleg and U. Weiser. 1996. MMX technology extension to the Intel architecture. *IEEE Micro* 16, 4 (1996), 42–50. <https://doi.org/10.1109/40.526924>
- [18] Plutarch. 1914. *Plutarch’s Lives, Volume I: Theseus and Romulus*. Harvard University Press (Loeb Classical Library). Original work c. 100 CE; see §23 for the ship.
- [19] Efraim Rotem, Adi Yoaz, Lihu Rappoport, Stephen J. Robinson, Julius Yuli Mandelblat, Arik Gihon, Eliezer Weissmann, Rajshree Chabukswar, Vadim Basin, Russell Fenger, Monica Gupta, and Ahmad Yasin. 2022. Intel Alder Lake CPU Architectures. *IEEE Micro* 42, 3 (May 2022), 13–19. <https://doi.org/10.1109/mm.2022.3164338>
- [20] William Shakespeare. 2003. *Hamlet*. Simon & Schuster (Folger Shakespeare Library). Original work c. 1603; Act 3, Scene 1.
- [21] Alan M. Turing. 1937. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* s2-42, 1 (1937), 230–265.
- [22] Marcelo Yaffe, Ernest Knoll, Moty Mehal, Joseph Shor, and Tsvika Kurts. 2011. A fully integrated multi-CPU, GPU and memory controller 32nm processor. In *2011 IEEE International Solid-State Circuits Conference*. 264–266. <https://doi.org/10.1109/isscc.2011.5746311>