In summary, building trust in autonomous transport is a multifaceted process that combines technological, psychological, and marketing aspects. The successful introduction of driverless vehicles requires a shift in marketing strategies from traditional models to approaches that account for consumers' digital behaviour, the role of social influence, and the need to build long-term relationships with users.

REFERENCES

1. Edelman, D. C. (2010). Branding in the digital age. *Harvard business review*, 88(12), 62–69. https://depositioneerders.nl/wp-content/uploads/2017/01/Branding-in-the-Digital-Age-HBR.pdf
2. Gesley, G. (2019). Comparative summary. *Regulation of autonomous vehicles in selected jurisdictions*. The Law Library of Congress. https://surl.li/cyqbpa
3. Nagarajan, S. (2020). *Exploring marketing strategies and the market perceptions of autonomous vehicles*. Doctoral dissertation, Dublin Business School. https://esource.dbs.ie/server/api/core/bitstreams/05d0765d-8ec4-42c5-890e-fbdad3cc89fe/content
4. Андрощук, Г. О. (2023). Рівень довіри до штучного інтелекту: аналіз результатів глобальних досліджень та стан в Україні. *Інформація і право*, 4(47), 217–231. https://doi.org/10.37750/2616-6798.2023.4(47).291675

Oleh Oprelianskiy
Kyiv National University of Technologies and Design
(Kyiv, Ukraine)
Maryna Vyshnevska
Kyiv National University of Technologies and Design
(Kyiv, Ukraine)

THE IMPORTANCE OF HARDWARE-SOFTWARE INTEGRATION IN MODERN COMPUTER SYSTEMS

In the early decades of computing, hardware and software were often viewed as distinct silos. Engineers built the physical circuits, and programmers wrote the instructions to run on them. However, as we move deeper into the era of artificial intelligence, edge computing, and high-performance data centers, this boundary has blurred. Modern computer systems no longer rely on raw power alone; instead, their efficiency and capability are defined by the seamless integration of hardware and software. This "hardware-software co-design" is the fundamental pillar that ensures performance, energy efficiency, and security in today's digital landscape.

The practical implementation of hardware-software co-design is rooted in established engineering principles that balance physical limitations with computational logic. There are several “rules of thumb” that have been well-understood since the 1980s when it comes to designing energy-efficient VLSI architectures. The architecture should use short wires and hence make use of local communication, since communication costs often dominate the energy budget. The architecture should consist of a large collection of components that operate in parallel, and each component should operate at a low voltage to save power while leveraging parallelism for performance. While these properties might be very appealing from a chip design perspective, the performance of such a system can be exploited only if users write parallel software, and even a small serial part of the program quickly limits performance. Hence, the improvements in transistor technology were used to build more and more complex execution engines that could be readily leveraged by existing software, until power constraints limited the introduction of additional complexity and led to the shift to multi-core systems (Manohar, 2021, 1).

Modern systems leverage Graphics Processing Units (GPUs) to handle massive parallel workloads that exceed the capabilities of a standard CPU. Hardware-accelerated cryptography represents a powerful approach to mitigating hardware and software security vulnerabilities by offloading cryptographic operations to specialized hardware components. By leveraging dedicated cryptographic processing units or accelerators integrated into modern hardware architectures, such as CPUs, GPUs, or dedicated cryptographic co-processors, hardware-accelerated cryptography enhances the efficiency and security of cryptographic operations while reducing the burden on software implementations. This not only improves the overall performance of cryptographic algorithms but also minimizes the exposure of sensitive cryptographic keys and operations to potential software-based attacks, such as side-channel attacks or malware exploits (Rahman, 2024, 11).

The future of hardware-software integration lies in achieving absolute programming transparency, where the distinction between physical and logical execution is invisible to the developer. The ultimate goal of modern system design is to ensure that programmers should not be concerned with interfacing and communication details. By extending multithreading paradigms to the hardware level, systems can be designed so that developers code their applications as if no differences existed between software threads executed on CPUs and hardware threads executed on FPGAs. This is made possible through a shared virtual-memory address space, which allows the software to remain completely unaware that it communicates with a nonsoftware thread. Such a seamless integration model effectively delegates platform-specific complexity to the operating system, ensuring that hardware acceleration is both portable and elegant (Vuletić, Pozzi, & Ienne, 2005, 105).

In conclusion, the evolution of modern computing has fundamentally redefined the relationship between hardware and software, moving away from a model of separate development toward one of total synergy. As demonstrated, the integration of specialized units like GPUs and the implementation of seamless hardware-software interfaces are no longer optional enhancements but structural necessities. These advancements address the critical limitations of power constraints and the "serial bottlenecks" that once hindered performance. By adopting a unified programming paradigm where hardware tasks are treated with the same transparency as software threads, the industry is overcoming the complexity barrier. Ultimately, the importance of hardware-software integration lies in its ability to transform rigid physical circuits into fluid, adaptable systems that define the success of innovation in computer engineering.

REFERENCES:

1.Manohar, R. (2021). Hardware/software co-design for neuromorphic systems. In Proceedings of the IEEE Custom Integrated Circuits Conference (CICC 2021).

2.Rahman, M. H. (2024). A comprehensive survey on hardware-software co-protection against invasive, non-invasive and interactive security threats. Cryptology ePrint Archive.

3.Vuletić, M., Pozzi, L., & Ienne, P. (2005). Seamless hardware-software integration in reconfigurable computing systems. IEEE Design & Test of Computers, 22(2), 102-113.