Security through Transparency and Openness in Computer Design*

Ivo Emanuilov[0000−0002−0055−4666]

KU Leuven Centre for IT & IP Law
ivo.emanuilov@kuleuven.be
http://www.citip.be/

Abstract. Trust in information technology depends on the level of security promised by the software and hardware stack operating on a platform. Consumers rely on cybersecurity updates of the software and firmware running on their devices to keep their privacy and data protected from malicious use. Businesses and governments procure technology which they expect to run for long periods and be kept in line with the state of the art in security. Presently, however, neither consumer, nor business solutions provide sufficient transparency regarding potential cybersecurity risks stemming from either the software or hardware stack embedded in them. Businesses need transparency in order to plan sustainable long-term operations, while consumers need devices that can be easily maintained and repaired and which offer sufficient information regarding real or perceived safety or security hazards. In the quest for security, transparency is a key sociotechnical requirement which lies at the core of trust in computing. As one of the most important abstractions interfacing the hardware and the lowest level software, the instruction set architecture (ISA) is perhaps the most essential element in the path to trust through transparency. Currently, however, the market is dominated by two proprietary ISAs in a duopolistic configuration, and their implementations are controlled by two major companies. This *status quo* has impacted significantly the integrated circuit supply chain in terms of both diversity and transparency.
This paper argues that open ISAs, such as RISC-V, would bring much-needed democratisation of microprocessor design while enabling higher levels of security through their modular design and extensibility. However, open ISAs are facing certain technical, organisations and legal challenges that require conceptual interdisciplinary thinking and coordinated legislative and regulatory response.

Keywords: Cybersecurity · Embedded Systems · Open ISA · RISC-V · Open Hardware · Transparency · IP rights · Liability.

* This research is funded by the European Union's Horizon 2020 research and innovation programme under the Secure Collaborative Intelligent Industrial Automation (SeCoIIA) project, grant agreement No 871967.

1 Instruction set architectures and their role for security

1.1 Instruction set architecture in computer design

The term 'computer architecture' usually refers to the instruction set architecture, on one hand, and implementation, on the other. In turn, implementation includes logical design (i.e., organisation) and physical design (i.e., hardware). In modern computer science, computer architecture denotes all three major aspects of computer design, that is, instruction set architecture (ISA), organisation (i.e., microarchitecture) and hardware [6].

There are two classes of ISAs, namely Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC). Suffice it to say for the time being that despite the commercial success of the Intel 80x86 proprietary CISC ISA, RISC has been long-recognised as the superior and preferred class of ISA, especially for customised embedded systems.

The instruction set architecture is one of the most important abstractions which delineates the "boundary between software and hardware" [6]. ISA is the interface between hardware and lowest-level software which "encompasses all the information necessary to write a machine language program that will run correctly, including instructions, registers, memory access, I/O devices..." [12]. For example, a C++ program is compiled into instruction for the central processing unit (CPU) to execute. How does a compiler know what instructions the CPU understands? It is precisely the ISA that provides this information. Essentially, ISA allows computer designers to consider functions independently from the hardware upon which they are executed [12], much like one can talk about the functions of a washing machine independently from its parts (e.g., tub, drain hose, debris filter etc.). Therefore, it is important to distinguish architecture from the implementation on a particular hardware which "obeys the architecture abstraction" [12].

Historically, the proprietary Intel 80x86 architecture established itself as the dominant ISA. Despite its notorious technical flaws [7], the success of this ISA was the product of three main factors [6]. The first was the early market choices made by IBM, i.e. when it selected the 80x86 architecture for the initial IBM PC, making binary compatibility with this ISA much desired. The second was the availability of resources afforded by technological innovation driven by Moore's Law which allowed Intel to translate from complex instruction set computing (CISC) to reduced instruction set computing (RISC). Essentially, this meant executing RISC-like instructions through hardware translation which ensured binary compatibility with the at-the-time fast growing software base while offering RISC-like performance. Finally, the high volumes of production of microprocessors helped Intel compensate for the cost of hardware translation from CISC to RISC.

The 80x86 ISA has only meaningfully been challenged on a commercial scale by the rise of the ARM ISA in system-on-chip (SoC) designs in the post-PC era, that is, after the launch of the first iPhone [7]. The trend is set to continue as a growing number of Internet of Things (IoT) devices and embedded systems are being procured and deployed in both industrial and consumer settings. This means that in the near future custom SoC platforms will likely become ubiquitous, as there are hardly any devices nowadays without some form of an embedded on-chip processor.

The fact that practically all dominant ISAs are proprietary in nature has given rise to serious concerns regarding the security of the future IoT ecosystem. For example, contemporary SoCs are well known for reusing multiple existing intellectual property (IP) cores to address complexity. [14]. IP cores are the "dominant form of technology delivery in the embedded, personal mobile devices, and relate markets" [12]. An IP core is "designed to be incorporated with other logic (hence, it is the 'core' of a chip), including application-specific processors (such as an video encoders or decoders), I/O interfaces, and memory interfaces, and then fabricated to yield a processor optimised for a particular application" [12]. Thus, for instance, in a modern Snapdragon SoC one would find designs from very many different sources, incl. an ARM-licensed IP, that is, the CPU. The growing complexity of SoCs has generated a corresponding growth in the reuse of IP blocks [14]. Since not all of these IP blocks are widely available for inspection and close scrutiny, this has resulted in the dominance of the 'security through obscurity' paradigm in the embedded systems market. Essentially, what one gets with most commercial ARM licences, for example, is a complete core or other product that can be incorporate in a design. The design itself, however, cannot be changed, unless one has an architectural licence. Presently, only very few and very big companies have such a licence, such as Apple, AMD, Nvidia, Qualcomm and others. This means that in all other cases one gets what everybody else gets with the same licence. Unfortunately, the recent examples of the *Spectre* and *Meltdown* security flaws allowing malicious actors to exploit vulnerabilities in the microarchitecture of some modern processors of the Intel, IBM POWER and ARM family have clearly demonstrated the miserable state of hardware *security through obscurity*.

1.2 Role of the instruction set architecture for security

The *Spectre* and *Meltdown* vulnerabilities relied on a side-channel attack leading to leakage of protected information. Essentially, the attack involved observation of the time required for a task to complete and "converting information invisible at the ISA level into a timing visible attribute" [7]. The unique feature of the *Spectre* and *Meltdown* security flaws is that they exploit a vulnerability in the hardware implementation. Since the current understanding of what constitutes a 'correct implementation' of an ISA is based on the architectural state of execution visible at the ISA level, it does not consider the performance effects of the execution of an instruction sequence [7]. While, technically speaking, *Spectre*

and *Meltdown* were the product of a strive for hardware optimisation that had little to do with the ISA itself, the flawed approach of how we ascertain 'correct implementation' of ISA was at least tangentially instrumental for the *success* of these hardware vulnerabilities.

The dominance of proprietary ISAs developed and controlled by just two major companies has nurtured an ecosystem in which even different implementations are likely to be plagued by the very same flaws. In other words, the rigidity of proprietary CPU designs dominated by two main commercial players increases the impact of vulnerabilities such as *Spectre* and *Meltdown* which have proven difficult to patch, with patches coming at significant performance costs. Simply put, having just two major CPU designs in the market means a hardware vulnerability is likely to have much more significant overall impact than if there were many and different, and even customised, implementations. Against this background, this paper joins a line of research arguing that hardware security is synergistic with open ISAs. Open ISAs are a precondition for open implementations [7], [10] verifiable through open security review processes and compliant the (legal) principle of security by design. Increasing the number of people and organisations involved in the design and development of secure architectures has already proven its utility in the context of free and open source software. A similar approach has been advocated by researchers calling for openness and transparency in the IT supply chains [2].

The case for a free and open ISA is built on strong technical and legal reasons as noted in [1]. Four specific reasons stand out among them.

First, companies often have patents on certain innovations in their ISAs which would prevent others from using them without proper licensing. Reportedly, Intel's patents over innovations around the 80x86 ISA (mostly extensions to the original ISA, such as Memory Protection Extensions (MPX), Software Guard Extensions (SGX) etc.) have been growing steadily in the past few decades [13]. In other words, the innovation surface is much smaller and the incentives - much less attractive, when innovation around alternative ISA-compatible designs is held off by prohibitive licence fees. Furthermore, free and open ISAs are likely to have positive economic impact by increasing competition in the ISA market currently defined by a duopoly.

Second, even though software ecosystems emerge around ISAs, these former are built by communities outside the immediate reach of the company developing the ISA. Furthermore, the expertise needed to develop an ISA is by no means concentrated in said companies; to the contrary, much of the expertise needed is widely available in open hardware communities, and compatibility with an ISA can be verified by open organisations.

Third, the availability and continued support of proprietary ISAs is heavily dependent on the company's will. In other words, if a company ceases its operations, it is likely that its proprietary ISA will go with it too.

Fourth, open ISAs mean development and availability of shared core designs, that is, more transparency and less likelihood of introducing fatal (security) flaws. Indeed, the principle of open design is part and parcel of the foundational Saltzer and Schroeder's 1975 Design Principles for Secure Systems. In their paper, Saltzer and Schroeder argued that "design should not be secret", "mechanisms should not depend on the ignorance of potential attackers, but rather on the possession of specific, more easily protected, keys or passwords" and that "it is simply not realistic attempt to maintain secrecy for any system which receives wide distribution" [15]. Ultimately, open design would make it much more difficult for State actors to intervene in the design process and introduce security backdoors.

1.3 Open ISAs in practice: the case of RISC-V

One recent noteworthy example of an open ISA that has generated a lot of interest in the embedded systems community is RISC-V. RISC-V is a royalty-free ISA developed in 2011 by Patterson and Asanović at Berkeley [1]. The driving force behind RISC-V is the desire for flexible, customisable and modular designs that can be implemented on custom chips at lower costs compared to their proprietary counterparts [4].

The need of creating an equivalent of the Linux kernel in the world of microprocessors is justified by the well-known benefits of opening the development and review process to a wider community. The experience gained in almost four decades of free and open source software development is a clear attestation to the success of this approach based on collaboration and transparency. While free and open source hardware and free and open source software are known to have both fundamental and incidental differences [5], the benefits of creating a virtuous cycle of open source hardware platforms based, among others, on open ISAs, are clear. They improve competition, encourage sustainable growth, and allow customisation, greater flexibility and, ultimately, better security.

Indeed, RISC-V is maintained by a community steered by the RISC-V Foundation, a non-profit organisation. The openness of the RISC-V ISA allows for public collaboration which means the *modus operandi* is based on resolving problems and discussing issues before taking any design decisions [7]. Importantly, the modular design of the RISC-V ISA means that the base of instructions running the full open source software stack is small and the optional extensions allow for customisation and optimisation depending on the needs [7]. The simplicity of the RISC-V ISA means less room for hidden flaws as it is all too well known that in the world of computer security complexity breeds vulnerabilities. Furthermore, open ISAs have a particularly strong case to make in times where state-sponsored backdoors can be (and have been) implemented at increasingly lower levels of abstraction in computer design. Specifically, RISC-V allows a manufacturer to know exactly what is going on at the microprocessor level. It also facilitates enhancement and customisation by allowing users to modify or

create designs which are aligned with their security needs. Finally, RISC-V is a particularly attractive ISA for governments as they could benefit from procuring open source ISA implementations known to be free of embedded malware[4].

2 Security promises of open ISAs

Open ISAs, such as RISC-V, offer a number of security promises. Some of them have already been outlined in the previous sections. This paper argues that in times when cybersecurity and cyber resilience are increasingly becoming a matter of survival, e.g. in light of the *NotPetya* and *WannaCry* attacks against critical infrastructure, such as hospitals and power grids, the need for transparency at all levels of computer hardware and software has become more prominent than ever. There are a number of advantages, but also some concerns regarding the security promises of open ISAs, as summarised in Table 1.

Table 1. Security Benefits and Risks of Open ISAs

Benefits	**Risks**
Modular design and extensibility	Ecosystem fragmentation
Transparency	Still chance of vulnerabilities
Long-term security evolution	Lack of interest by the community
Community review	Commercial and governmental support and scalability
Royalty-free use	Legacy compatibility, upfront transition costs

First, the modular design of open ISA, like RISC-V, offer not only the ability to implement customised solutions but also to iterate and enhance them in an open and conducive to dialogue environment, such as the respective community created around the ISA. In turn, this would enable much quicker design and development cycles [7] which allegedly implies that fixing issues and security vulnerabilities should be equally quicker. This aspect of open ISAs is also critical in light of the long-term support and availability of devices implementing this ISA. This is especially beneficial in the context of Internet of Things where many connected devices will need to be supported over a long time span. The modular design of open ISAs, like RISC-V, allows security extensions to be added at ease, while keeping them close, if necessary, since the core IP would be standardised anyway [14]. There is a need, however, to define and perhaps redefine the parameters that go into evaluating what constitutes a 'correct implementation' of the ISA. Indeed, ecosystem fragmentation is one of the major challenges before the uptake of RISC-V and it may have considerable security consequences as well (e.g., concerning verification and independent third-party testing).

Second, open ISAs would also make it possible to build test suites for exhaustive testing by all users and would facilitate the application of formal methods for

verification of the trustworthiness of hardware [17]. Transparency "allows users to place justified trust in the hardware being used and enabled comprehensive evolutionary improvements to be made" [17]. With more 'eyeballs' looking at the same specification, community-driven open ISAs clearly have the advantage of open security by peer review over their proprietary counterparts. This is not only an advantage for businesses, but equally for governments. In times of growing calls for 'digital sovereignty', implementations based on open specifications would clearly allow governments greater control over the procurement and supply of embedded systems which may become part of a State's critical infrastructure. Specifically, governments could leverage regulatory processes such as 'reverse cascade' to exert regulatory pressure on distributors under their jurisdiction to sell products compliant with certain open and transparent design and manufacturing standards [9]. However, just because a specification is open does not mean it comes without vulnerabilities. Clearly, the paradigm of security through public peer code review is much preferred to security by obscurity, yet there have been cases where the 'many eyeballs' argument has not been very convincing. For one, the Heartbleed vulnerability in the open source OpenSSL library was a case in point described by some as 'open source's worst hour' [16]. Exaggerated as such qualifications might be, Heartbleed showed one thing clearly: just because the code or specification is free and available for public review does not mean that someone will actually carry out this review or that standard analysis approaches work for detection of such vulnerabilities [18]. Lack of interest by the community in certain software packages has often led to lack of support and maintenance for these packages. Granted, this is not a failure of open source *per se*, but it is a fact that needs to be considered in the context of the community created around an open product, service or specifications thereof.

Third, open ISAs can be particularly useful in environments where embedded systems are deployed for long-term use and must therefore conform to objectives concerning long-term security evolution. In such environments, systems would have to be able to support security evolution as the threat landscape evolves. Indeed, the community created around an open product, service or specifications could remain vibrant and active for many decades. However, there is of course also the risk of potential lack of community support. While this is clearly not the case for promising community projects such as RISC-V, the need for support on a commercial scale is critical for the success of microprocessor implementations based on open ISAs.

Fourth, the potentially huge community that may be created around an open ISA would clearly improve the security review and audits of an open specification. These communities, however, need both institutional and financial support in order to grow. Promoting openness and transparency by legal, regulatory and standardisation measures is critical for the creation of a strong community. It is even more important for creating strong incentives for businesses to build a competitive market for support and maintenance services organised around these communities. In other words, encouraging the creation of strong support

and maintenance services around open ISAs is critical not only for the uptake of one specification or another, but also for their long-term security evolution.

Finally, one of the main advantages of open ISAs is that their use is free of royalties and licensing costs, meaning one can start relatively quickly with little resources. However, the transition of the entire infrastructure of business or governmental upstream players to implementations based on open ISAs can still have prohibitive costs. Binary compatibility notwithstanding, large-scale deployments would likely require rebuilding the entire supporting infrastructure. While cutting and bleeding edge players may be up for the challenge, the transition in safety-critical environments, such as manufacturing or healthcare, where legacy operational technology and new information technology systems have to play nicely together, may generate significant upfront costs.

3 Legal and policy perils of open ISAs

Besides the purely technical and economic promises and issues of open ISAs, there are vastly important legal and policy perils whose resolution may prove critical for the success of open architectures.

3.1 Manageability, collaboration and competition

The first problem concerns the legal infrastructure needed to ensure manageability of open ISAs and the challenge of preventing Balkanisation of this domain. Indeed, the rigidity of established supply chains in the ISA market characterised by a duopoly often creates risks of lock-ins and may entail high and even prohibitive termination costs should one try to leaves the 'walled garden'. However, open ISAs can also bring more competition in the market, by pulling control away from Intel and ARM [4]. Furthermore, the modularity of open ISAs, like RISC-V, can clearly create new markets for customised solutions, e.g. field programmable gate arrays (FPGA), based on specific needs driven, *inter alia*, by security.

The development of open ISAs, organised as a collaboration within a community, carries the potential to democratise computer design. However, collaboration can also bring about certain perils. For example, the RISC-V Foundation is concerned with the "release of RISC-V to the open community for both standardization and ongoing improvement through open collaboration". Standardisation is therefore critical for the success of open ISAs. Indeed, compliance with standards is critical to prevent the fragmentation that may come with the modularity and extensibility of an open ISA, like RISC-V. Unlike proprietary ISAs controlled by large companies, making it easier to verify compliance of an implementation with the specification, open ISAs will open the market to many more companies. Ensuring compliance of many different implementations with one single specification is therefore a fundamentally different challenge. The work

carried out in the framework of the RISC-V Foundation is critical, but it must be supplemented by dedicated efforts at governmental level promoting openness and transparency in the procurement of implementations based on open specifications. These efforts, however, should be balanced against the interests of protecting competition and ensuring that collaboration does not mature into collusion.

The ongoing cooperation between industry players demanding open specifications is critical for the success of open ISAs. The community should also be prepared for attacks from incumbent players, like the notorious anti-RISC V website launched by ARM in 2018 [4]. The legal status of the RISC-V Foundation as a steering force and its immunity to trade curbs is equally important. It is precisely such fears that forced the RISC-V Foundation to move its headquarters from Delaware to Switzerland in 2019. In times of global geopolitical rage against the deployment of 'foreign' technologies in public infrastructure, to ensure the continuity of development standardisation efforts of RISC-V in a jurisdiction known for its high legal standards is a legal as much as a policy and political question.

3.2 Intellectual property rights

Arguably, one of the main advantages of open ISAs is that one does not need to deal with complex contractual arrangements, pay royalties or handle delicate issues over future research and development licensing requirements. However, as Andrew Katz has recently demonstrated in his empirical study, open processor and, more generally, free and open source hardware licensing is far from clear [8].

Indeed, industrial players admit that "currently available copyleft open hardware licences are insufficiently clear in their effect to be safely used" and "potential benefits of copyleft licensing in core designs are not yet sufficiently clear to show an overwhelming need to shift to a copyleft model" [8]. Interestingly, the interviewees in this study pointed out that "the lack of open source or low-cost toolchains was an inhibiting factor in the growth of open hardware communities focusing on cores" [8]. As open source toolchains are a much rarer breed in open hardware communities, compared to open source software, there are legal issues which have yet to be resolved. For example, there are questions concerning the legal status of code incorporated by the toolchain into the output, or whether the bitstream is a computer program in the legal sense and, if so, who is running it upon booting the hardware [8].

The choice of appropriate licence is relevant not only from a commercial perspective, but it is also important for security purposes. In the notorious example of *Heartbleed*, the OpenSSL project was using a custom license which was not compatible with the commonly accepted by the free and open source community GNU General Public License. Arguably, using a standard free and open source

licence would have increased the community's involvement through code contributions and review [18]. Eventually, this would have had the effect of strengthening the project's resilience against vulnerabilities such as *Heartbleed*. This line of thought is equally applicable in the context of open processor and, more generally, open hardware licensing, and it goes to show the important connections between intellectual property rights and cybersecurity.

3.3 Liability

In the wake of the *Spectre* and *Meltdown* vulnerabilities, Intel was challenged in several class actions in US courts where the plaintiffs sought damages from Intel. Chief among these lawsuits is the case of *Intel Corp. CPU Marketing, Sales Practices and Product Liability Litigation, case number 3:18-md-02828, in the U.S. District Court for the District of Oregon* [11].

In this case, the plaintiffs based their claims on three main allegations: (1) failure by Intel to disclose defects in its processors, (2) which create security vulnerabilities that could lead to a breach of confidential data and (3) issuing patches to fix these defects which substantially diminish the speed of Intel's processors. Essentially, the plaintiffs argued that Intel prioritised speed over security, making a user's confidential information susceptible to side-channel attacks (i.e., by taking design decisions to implement branch prediction, speculative execution, out-of-order execution, and an unsecured cache subsystem) by exploiting two main flaws.

In the case, Judge Simon dismissed the plaintiffs' claims on grounds of failing to demonstrate the type of injury required to show standing. He highlighted that none of the plaintiffs have discontinued using or replaced their computers because of the alleged defects. He also noted that the plaintiffs "do not explain how this alleged defect would have affected the market price for Intel's chips in light of the fact that it involved all the chips in the market" [11]. The judge continued that the plaintiffs "have not sufficiently alleged what 'adequate measures' they reasonably expected relating to the alleged security vulnerabilities or what they allege was the parties' bargain that Intel did not meet" [11]. He found that "Plaintiffs also allege that Intel's success largely is based on the speed of its processors [but they] do not allege that they would have sacrificed that processing speed for additional security against theoretical vulnerabilities, most of which had been known in the industry for two decades. Plaintiffs instead assert only general, conclusory allegations about desiring and expecting "adequate" security. The Court finds that Plaintiffs have not sufficiently alleged their reasonable expectations for data security or the absence of the specific alleged security vulnerabilities." [11] Judge Simon distinguished this case from data breach cases which are "more instructive because they explicitly consider whether data security was part of the parties' underlying bargain". He continued that "[i]n data breach cases there already has been a breach of security, and the plaintiffs in those cases contend that a minimum level of reasonable security protection was

part of the parties' bargain and expectation. Here, in contrast, there has been no data breach. Further, Plaintiffs' allegations show that for decades it was known in the industry that Intel's designs were vulnerable to various side-channel attacks. Yet no actual security breach occurred over the years, despite these known security vulnerabilities. Even after these and other security vulnerabilities became more publicly known, they were still only theoretical and have been exposed in conceptual form. There are no allegations of any actual data breaches or "hacks" to date as a result of the alleged security vulnerabilities" [11].

While this particular case dealt with a problem inherent in the implementation of the Intel 80x86 ISA and not in the ISA itself, it shows that liability cases may be on the rise as more and more hardware vulnerabilities are reported daily. The notorious complexity of the 80x86 ISA and the ever-growing number of instruction set implementations protected by patents is certainly an argument in favour of open ISAs. However, one cannot but think whether this case would be any different had the 80x86 ISA been open. For example, if the implementation had not been entirely correct according to the specification, would the designer be liable and on what grounds? in cases of collaborative open ISAs, such as RISC-V, whose should be the responsibility to define what a 'correct implementation' is? Another layer of complexity is added by cases of attacks combining software and hardware vulnerabilities, particularly computer architecture vulnerabilities [3]. How would the liability be allocated between the different parties in such a case?

It is beyond the scope and ambition of this paper to enter into a discussion on any of these questions. However, it is important to note that transparency of the entire integrated circuit supply chain is key to resolving many of them. At the same time, one should not think that open ISAs are a panacea. They are merely part of the solution and perhaps one of the most important building blocks towards transparent and truly trustworthy computing.

4 Conclusion and further work

Transparency is a key sociotechnical requirement which lies at the core of trust in computing. As one of the most important abstractions interfacing the hardware and the lowest level software, the instruction set architecture is perhaps the most critical element in the path to trust through transparency. Presently, two proprietary ISAs dominate the market in a duopolistic configuration and their implementations are controlled by two major companies. This has had a major impact in terms of diversity and transparency.

This paper argued that open ISAs, such as RISC-V, would enable much-needed democratisation of microprocessor design while enabling higher levels of security through their modular design and extensibility. However, open ISAs have certain technical, organisational, legal and policy challenges that require conceptual thinking and legislative and regulatory action. Furthermore, any such action

should account for the global nature of the integrated circuit supply chain, meaning transparency regulation would be only as strong as the legal and political power exerted by the party trying to enforce it.

Transparency regulation and openness are critical for the cybersecurity of the impending embedded systems revolution in the face of IoT. Technical solutions, like open designs, should go hand in hand with a legal framework that balances the objective of transparency for cybersecurity against competing and legitimate interests protected by competition law, intellectual property law or tort law.

References

1. Asanović, K., Patterson, D.A.: Instruction Sets Should Be Free: The Case For RISC-V. Technical Report UCB/EECS-2014-146, Electrical Engineering and Computer Sciences, University of California at Berkeley (2014), https://people.eecs.berkeley.edu/~krste/papers/EECS-2014-146.pdf
2. Chattopadhyay, A., Guilley, S., Heiser, G., Kasper, M., Krauß, C., Krüger, P.S., Kuhlmann, D., Reith, S., Schallbruch, M., Seifert, J.P., Weber, A.: Quattro S Initiative: Eradicate Faults and Backdoors in Information Technology and Facilitate Innovation. Tech. rep., Quattro S Initiative (2019)
3. Chen, K., Deng, Q., Hou, Y., Jin, Y., Guo, X.: Hardware and Software Co-Verification from Security Perspective. In: 2019 20th International Workshop on Microprocessor/SoC Test, Security and Verification (MTV). pp. 50–55 (2019). https://doi.org/10.1109/MTV48867.2019.00018
4. Greengard, S.: Will RISC-V revolutionize computing? Communications of the ACM **63**(5), 30–32 (2020). https://doi.org/10.1145/3386377, https://dl.acm.org/doi/10.1145/3386377
5. Gupta, G., Nowatzki, T., Gangadhar, V., Sankaralingam, K.: Open-source Hardware: Opportunities and Challenges. arxiv preprint (2016), http://arxiv.org/abs/1606.01980
6. Hennessy, J.L., Patterson, D.A.: Computer Architecture: A Quantitative Approach. Morgan Kaufmann, 6 edition edn. (2017)
7. Hennessy, J.L., Patterson, D.A.: A new golden age for computer architecture. Communications of the ACM **62**(2), 48–60 (2019). https://doi.org/10.1145/3282307, https://dl.acm.org/doi/10.1145/3282307
8. Katz, A.: A Survey of Open Processor Core Licensing. International Free and Open Source Software Law Review **10**(1), 21–46 (2018). https://doi.org/10.5033/ifosslr.v10i1.130, https://jolts.world/index.php/jolts/article/view/130
9. Kim, N., Herr, T., Schneier, B.: The reverse cascade: Enforcing security on the global IoT supply chain. Tech. rep., Atlantic Council Scowcroft Center for Strategy and Security (2020), https://www.atlanticcouncil.org/in-depth-research-reports/report/the-reverse-cascade-enforcing-security-on-the-global-iot-supply-chain/
10. Mühlberg, J.T., Van Bulck, J.: Reflections on post-Meltdown trusted computing: A case for open security processors. ;Login: the USENIX magazine **43**(3), 1–4 (2018), https://lirias.kuleuven.be/retrieve/516518
11. for the District of Oregon, U.D.C.: In re: Intel Corp. CPU Marketing, Sales Practices and Product Liability Litigation. Tech. Rep. 3:18-md-02828, U.S. District Court for the District of Oregon (2020)

12. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design: The Hardware/Software Interface. Morgan Kaufmann Publishers, an imprint of Elsevier, risc-v edition edn. (2018)
13. Rodgers, S., Uhlig, R.A.: Intel's X86: Approaching 40 and Still Going Strong, https://newsroom.intel.com/editorials/x86-approaching-40-still-going-strong/
14. Salmon, L.G.: A Perspective on the Role of Open-Source IP In Government Electronic Systems. Presentation, DARPA, RISC-V Workshop (2017)
15. Saltzer, J., Schroeder, M.: The protection of information in computer systems. Proceedings of the IEEE **63**(9), 1278–1308 (1975). https://doi.org/10.1109/PROC.1975.9939
16. Vaughan-Nichols, S.J.: Heartbleed: Open source's worst hour, https://www.zdnet.com/article/heartbleed-open-sources-worst-hour/
17. Weber, A., Reith, S., Kasper, M., Kuhlmann, D., Seifert, J.P., Krauß, C.: Sovereignty in Information Technology. White paper, Fraunhofer SIT, Fraunhofer Singapore, RheinMain University of Applied Sciences, TU Berlin/T-Labs (2018)
18. Wheeler, D.A.: Preventing Heartbleed. IEEE Computer **47**(8), 80–83 (2014). https://doi.org/10.1109/MC.2014.217