Trusted Electronics: Current And Future Developments

Key techniques for demonstrating quality, safety, and security, from unique identification to open source.

By Roland Jancke

In today's world, we encounter electronics increasingly as complex hardware/software systems, such as those in vehicles, machines, and communication devices. These systems are characterized by dramatic increases in functionality in areas like environment sensing, stages of autonomy, and the installation of future updates. Developing and manufacturing such electronic systems today requires global supply chains for materials, components, services, and tools. Analyses show there are various gateways for intended or accidental flaws along these chains. Yet for safety-relevant applications, creating and demonstrating a high degree of operating safety and data security is absolutely indispensable.

So how can companies engender trust in complex electronic systems with regard to quality, safety, security, and integrity?

Having a development process that is properly structured end to end is a fundamental prerequisite for high quality, safety, and security. On top of general quality standards, there are specific standards for many fields of application, through which companies can achieve data security and operating safety.

Features that allow the uniqueness of a component or a process to be identified and verified are an important element. For some years now, there have been various approaches involving physically unclonable functions (PUFs), which make it impossible to copy components by using minimal variations in physical parameters as identity markers. Naturally, it's important to ensure the stability of such features in the face of temperature fluctuations and over the component's lifetime.

In the future, there will be certified proofs of authenticity for the identities of electronic components and assemblies. Such methods connect the unique features of individual components with each other through a chain of trust such that the entire assembly can be uniquely identified, with the proof founded on a root certificate.

Another crucial element is the verification of trustworthiness. The question that arises here is: Does a chip or a component contain all defined functionalities and only these? The latter part of this question in particular is not easy to answer – it's difficult to prove the absence of features. Theoretically, this proof can be furnished only through formal verification, which means formalizing the defined characteristics as well as the implementation so as to prove a bijective relationship.

To this day, however, such procedures are available only for a limited number of functions and variables. If the number of internal states increases substantially, the complexity of a formal system description very quickly reaches an order of magnitude that is no longer manageable. Moreover, this route is open only to digital functions and circuit parts in the first place. Further developments will be needed to be able to formally describe and verify analog and mixed-signal circuits as well.

Simulative verification methods have established themselves as an alternative to formal techniques. Although they can handle even very high levels of complexity, they can never prove a complete absence of unwanted functionality. Publicly accessible libraries of known hardware and software weaknesses are maintained to deal with this problem, such as the CWE (Common Weakness

Enumeration) database from MITRE. Implementing security checks in existing workflows is the goal of, for example, the Accellera standard SA-EDI (Security Annotation for Electronic Design Integration).

Open source principles are another cornerstone of trusted electronics. Open-source code for hardware and software as well as a large community of developers minimize the risk of intentional flaws and backdoors. However, errors and security gaps can also be overlooked or accidentally introduced. Furthermore, the use of open-source hardware still poses challenges for the developers of safety-relevant applications. Questions regarding support and liability in the case of errors are often left unresolved.

The chief exponent of open-source hardware today is the RISC-V instruction set architecture (ISA). A range of open implementations exists for the RISC-V ISA, while commercial products based on the open architecture, but including proprietary refinements, can be found on the market. A whole ecosystem of tools, add-ons, test suites, and the like has sprung up around the open ISA, which facilitates entry and has clearly encouraged the spread of RISC-V.

Alternative open-source tools have also become available for other tasks in the hardware development process. This is a way of ensuring that malicious changes to the hardware cannot be made through the tool – for example, in a synthesis step. On the other hand, these tools are still nowhere near as comprehensive and mature as commercial solutions owing to their lack of licensing revenue and their ultimately small circle of users. Further open-source tools will establish themselves on the market and time will tell which prevail in the long run.

Many approaches and solutions to trusted electronics are already available. Although they generally increase development work and costs, they add security, quality, and integrity.