Electronic Design

ENRIQUE MARTINEZ-ASENSIO, Functional Safety Manager,
EnSilica, https://www.ensilica.com/

Select the Right Microcontroller IP for Your High-Integrity SoCs

There may be a seemingly unlimited array of microcontrollers for consideration, but how do you select the right one when developing high-integrity SoCs that meet all functional-safety requirements?

It seems like there's an almost unlimited array of microcontrollers—from low-cost 8-bit varieties to high-performance 32/64 multicore systems. While some systems require little thought beyond cost or speed, there are also many high integrity applications requiring reliability, security and/or functional safety (FuSa) standards to be met, requiring the architect to pay special attention to the MCU selection.

At the extreme end, today's vehicles contain 100+ MCUs serving different functionalities: infotainment, steering, braking, engine control, ADAS, etc. Some of these systems are built with commercially available MCUs. However, when a high degree of integration and low cost are the top priorities, dedicated systems-on-chips (SoCs) become necessary. These contain not only an MCU but additional analog/digital processing functionalities.

This article runs down seven of the considerations that we use when selecting an MCU IP for a high-integrity ASIC or SoC. We also look at several of the IPs available to the market.

Key Considerations in Selecting an MCU IP for an ASIC or SoC

MCU vs. hardwired state machines

For low-complexity systems, the first choice is between an MCU and a finite state machine (FSM). Each approach has pros and cons, but FSMs typically take up less space on the silicon, and they have lower power consumption—this assumes a fully optimized logic for the application. In addition, there are no royalty fees and no need to develop embedded software.

Conversely, an MCU gives more flexibility, with software upgrades delivered via non-volatile memories (OTP, EEROM, flash). This lowers design risks, allowing firmware changes rather without the cost of a new mask set. Therefore, unless the complexity is exceptionally low (with robust specifications), the MCU option will be always the winner.

Soft MCU?

Some MCUs are available as synthesizable IPs in standard hardware description languages, typically Verilog, System Verilog, or VHDL. The code can be integrated into a higher-level hierarchy. Vendors usually provide an "integration manual" among its documentation describing architectural features, which will cover topics like clocking, floorplanning, and testability

Some vendors also offer dedicated soft MCUs that are specifically designed with high-integrity applications in mind. These have features and safety measures targeted to achieve the relevant standards compliance.

Safety measures

FuSa-dedicated processor IPs contain safety measures to cope with events that could ruin the system functionality, such as data corruption, out-of-control program flow, unexpected bugs, which can cause safety goal violations. The following safety measures are typically included:

- *Watchdog:* Resets the CPU should the watchdog timer count down to zero.
- *Memory protection unit (MPU):* Enables control and notification of unexpected accesses to specific memory regions.
- *Parity in the data path:* For registers, etc., to detect bit flips.
- *Trusted zones:* Protects access to critical resources via authentication mechanisms.

- *EDC (error detection and correction) in memories:* Allows for bit flip corrections caused by soft errors.
- *Dual-core lock step (DCLS):* Implements two processors running identical code in parallel, with any discrepancy triggering an error.
- *Triple modular redundancy (TMR) in critical logic parts:* Gives higher system availability when error notification isn't enough. This is commonly used in high-reliability systems, particularly aerospace.
- *Safety Test Libraries (STL):* Used to help system integrators develop startup or run-time tests that detect and report inconsistencies.

The capability of the mentioned safety measures to detect or correct failures is normally described in the applicable standards. A good example of this is in ISO 26262 Part 5, which refers to "diagnostic coverage" that allows the system integrator to compute the necessary safety metrics and FIT rates, etc.

Power consumption

The power consumption of a given MCU IP will depend heavily on the silicon implementation and final supply voltage, as well as the software it's running. MCU IP vendors normally provide typical power consumption of the IP in multiple silicon technologies with indicative power figures in µW/MHz. But this power is often the average power running the Drystone or CoreMark benchmark, and may not reflect the actual power in your application.

However, even for a known technology node, the power consumption can diverge from assumed. It will depend on the actual application software and the amount of memory accesses, such as the number of R/W accesses and bus transactions.

This means that if the power consumption is a critical factor, it's recommended to perform a deep analysis with the appropriate EDA tools.

The safety package

IP vendors usually offer a "safety package" for MCU soft cores dedicated to high-integrity applications. The safety package enables the system integrator to facilitate and document the compliance to relevant safety standards and must contain at least:

- A summary of the IP development process, especially the verification procedures, showing that enough measures to prevent the presence of systematic faults were taken.
- FMEA/FMEDA, considering standard failure modes like data corruption, stuck-at logic in critical registers, etc., and the consequences and remedies.
- Available safety measures, how to activate them, and the applicable diagnostic coverage according to the target safety standard.

Typically, such safety packages are accompanied by a third-party assessment certificate, which will state the compliance points that have been verified in the context of common safety standards, e.g. ISO 26262 or IEC 61508.

It's important to note that such certification cannot and must not be interpreted as an automatic authorization to use the incumbent IP in safety applications. It's vital that anyone using these should always verify what the certificate says and what compliance points are covered in the assessment.

Is a safety package essential, even in FuSA application?

Depending on the involved safety requirements class, in theory it's possible to develop a FuSa application without a safety package. If this is the only option (for example, due to budgetary constraints), a developer should consider an important extra step and instead perform the requested safety analysis in house.

This is a legitimate option, but it requires a deep knowledge of the MCU architecture and the ability to demonstrate the RTL code reliability, through the so-called "proven-in-use" argument. The alternative is an extensive verification task.

Eventually, you may reach the conclusion that the available IP doesn't contain all of the necessary safety measures to achieve the requested FuSa hardware metrics, especially for higher ASILs (C, D). And for us, the added cost of the "safety package" is more offset.

One additional note: For high ASIL grades, an independent third-party project assessment is compulsory. These assessors will put in many significant objections if the requested work products regarding the MCU aren't available.

Software development tools

The embedded software development in high-integrity applications is also subject to important requirements for coding and verification. Therefore, availability of the right design tools is of paramount importance.

Indeed, the different standards prescribe requirements regarding the use of SW tools; for example, ISO 26262-8 contains the section "Confidence in the use of software tools." This standard requires the developer to perform a tool classification and, if necessary, a qualification of the intended development tools, such as the compiler and linker.

Like IP vendors, many software tool vendors try to make life easier for FuSa engineers and have undertaken the required tool classification/qualification processes, issuing third-party-assessed certificates of compliance.

And, similarly, as it is for MCU IPs, the use of "non-certified" tools is possible. However, it will be up to the user to demonstrate the reliability of the selected tool by performing the tool's classification and eventual qualification.

In addition to traditional embedded software development tools, high-integrity applications require additional software verification steps, stipulating the use of other tools

Cortex-M scalable performance with Functional Safety features

		Cortex-M3/M4 /M0+	Cortex-M23	Cortex-M33	Cortex-M55	Cortex-M7	Cortex-M85
Level of Safety Support		Standard	Extended	Extended	Extended	Extended	Extended Planned
	Safety Features	STLs, MPUs	Transient Fault Protection, System level DCLS, STLs, MPU	System level DCLS, STLs, MPU	ECC, MBIST, Transient Fault Protection, Interface Protection, DCLS, STLs	ECC, MBIST Interface, DCLS MPU	ECC, MBIST, Transient Fault Protection, Interface protections, DCLS, STLs
Claims	**Diagnostic Claim**	×	**ASIL B (Target)**	×	**ASIL D** **ASIL B (Target)**	**ASIL D**	**ASIL D,** **ASIL B (Target)**
	Systematic Claim	×	**Up to ASIL D**	**Up to ASIL D**	**Up to ASIL D**	**Up to ASIL D**	**Up to ASIL D**

*STLs : System Test Libraries
ECC: Error correction codes
MBIST: Memory Built in self test
DCLS: Dual core lock step
System Level DCLS: App note to produce Lock step by system integrator
MPU : Memory protection unit

arm

such as:

- *Code-style analysis:* Readability and absence of obfuscation are key features, and there are different industry-standard recommendations, like MISRA C or HIS metrics (Hersteller initiative Software).
- *Conditions coverage analysis:* This verifies that every line and every conditional jump is executed.
- *Faults injection:* Essential where high safety levels may require one to use or develop ad hoc solutions, such as those based in the JTAG debugging interface.
- *Hardware-software co-verification:* Necessary to perform the software unit testing by using techniques like "hardware-in-the-loop" (HIL).

And as it was for soft MCUs, the availability of a third-party certification for any given tool doesn't mean an automatic authorization for using the tool with safety standards compliance. It's always vital to verify what requirements are covered by the referred certificate.

Major MCU IP Vendors and Their FuSa Solutions

Several IP vendors provide solutions that can meet the previous selection guidelines. Here we outline the major offerings.

Arm

Arm is the world leader in 32-bit processor IPs and has a comprehensive Cortex-M portfolio with several FuSa support levels and safety mechanisms. The *table* covers both low and high-end processors.

Safety packages include a safety analysis (FMEA), a safety manual with the description of the included safety mechanisms, and an independent third-party assessment certificate, stating the development process conforms to safety standards. Arm also provides FuSa-compliant development tools.

Synopsys

Synopsys offers its ARC architecture in addition to verification tools that are specifically oriented to automotive FuSa applications.

Cadence

Cadence, through its IP business Tensilica, delivers the Xtensa 32-bit processor with FuSa and security features, and dedicated design tools.

RISC-V

RISC-V is an emerging open-source 32-bit IP platform. Some design houses have tuned its architecture to build automotive FuSa or aerospace/hi-rel solutions, but so far only high-end processing cores are being offered, with Codasip, Fraunhofer-IMS, Gaisler, Imagination, NSI-Texe, and SiFive all supporting RISC-V based IPs.

8-bit MCUs

Low-end processors are still necessary for certain applications, especially when ultra-low power consumption and high integrity go together. MCUs like the 8051 or the 6805 are available as synthesizable Ips; however, we've not seen a vendor offering direct FuSa features. This, therefore, imparts a lot of extra effort to anyone intending to include such a legacy architecture into a SoC.

Enrique Martinez-Asensio specializes in functional safety and mixed-signal SoC design at the custom ASIC firm, EnSilica. He has a core focus on automotive systems, but a wider expertise that covers everything from vehicles to white goods.