WP511 (v1.0) April 15, 2019

# Risk Management for Medical Device Embedded Systems

*Zynq UltraScale+ MPSoC technology can be applied in the design of medical devices and systems to meet functional safety and cybersecurity standards, creating more robust designs and accelerating time-to-market.*

**ABSTRACT**

This white paper provides an introduction to functional safety and cyber-security methods in the design of medical devices, a particularly challenging space given the complexities of developing products in a regulated industry.

The risks defined by medical device regulators can be managed through a shared view of robust functional safety methodology and cybersecurity requirements, ensuring that the device hardware and software achieve proper and intended operation. These same goals exist in the industrial automation space and are regulated by independent authorities using the IEC 61508 and IEC 62443 standards.

Xilinx offers a broad set of capabilities in its silicon and development tools designed to create implementations that meet and can be certified to these standards. Available for both the IEC 61508 and IEC 62443 standards, the Zynq® UltraScale+™ MPSoC platform can be leveraged by medical device developers to meet the shared goals of safety and security in electronic medical systems.

# Introduction

In the medical device market, products range from non-intrusive diagnostic imaging equipment such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Ultrasound machines to intrusive devices such as surgical robots, life-supporting devices like ventilators, and implantable devices such as pacemakers. Each of these broad product categories has different levels of risk impacting patients under care. This drives regulators such as the United States Food and Drug Administration (FDA) and the European Union (EU) Medical Device Directive (MDD) CE marking teams to look to device manufacturers to manage this risk through various product design measures.

## Medical Device Design Considerations

When designing a medical device, the engineering team must be concerned not only with developing a device that is effective in its clinical application, but also with protecting the safety of the patient (e.g., IEC-60601-1, 1-4 for Hardware Electrical Design and IEC-62304 for Software Life Cycle Process), the information assurance of data flowing through the system (e.g., HIPAA), and the life cycle of the device (e.g., FDA CFR 21), which includes methods for logging operation and distributing product updates.

These considerations for safe and intended operation are not specific to the medical device industry; they also impact other industries in which humans are working together with machines that are granted increasing levels of autonomy. In the industrial space, robots are no longer just slaves to their human operators, but are instead becoming truly independent collaborators—although ones that must always yield to the safety of the human. For example, in the automotive space, vehicles are becoming more automated with increasingly capable safety functions upon which human drivers are quickly becoming reliant.

In industrial automation, the implementation of a machine that has a direct impact on the safety of a system falls under the category of "functional safety." This terminology refers to a specific set of standards and design methodologies for meeting those requirements. While functional safety design methodologies outline both intended operation and failure paths, it must also be recognized that increasing cybersecurity threat levels exist where a bad actor tries to modify the intended operation of the system. Therefore, to explore the full risk management equation of building a medical device product today, both functional safety and cybersecurity must be considered to be integral parts of overall patient safety. It is possible to have a cybersecure design without functional safety, but one cannot have a functionally safe design without addressing cybersecurity.

This white paper is divided into these two major topics, functional safety and cybersecurity:

1. An overview of Functional Safety and the common design methodologies available for realizing products at various safety integrity levels (SILs)
2. An overview of Cybersecurity evaluating the implemented security level (SL) against possible threats

# Functional Safety

Functional safety is a system created using a set of practices and guidance outlined in the appropriate standard that supervises a machine's *primary* function to reduce risk of harm to people and/or the environment. In general, safety functions continuously monitor the equipment under control (EUC) and becomes active when something unusual happens to the operation of the machine (e.g., machine failure) or an external force causes a hazardous condition. At design time, the machine's manufacturer performs a hazard and risk assessment to understand what risk reduction actions are needed.

Functional safety systems can be as simple as limit switches hooked to a power supply, or as complex as a LiDAR system guarding a manufacturing floor by watching for people crossing a yellow hazard line (e.g., getting too close to a moving truck chassis in an assembly line).

A good analogy for functional safety exists in football—both the American and soccer versions. The players are the Equipment Under Control (EUC); the referee is the Safety-Related System. The referee knows the rules and controls the game, stopping and restarting play as demanded by various occurrences, and importantly to assess penalties against teams or players seen to violate the rules.

## Functional Safety Considerations in Medical Device Design

In the US, federal law requires device manufacturers to notify the FDA of their intent to market a medical device. This is known as a Premarket Notification or "510(k)" based on that section of the Federal Food, Drug, and Cosmetic Act (FFDCA). This clause enables the FDA to clear for sale those devices it feels are substantially equivalent to a previously legally marketed device given the appropriate application.

With respect to functional safety, an argument of substantial equivalence is problematic because of the uniqueness of implementation and the physical attributes of devices used to build a substantially equivalent function.

For those devices and machines that do not fall under FDA control, local governments require manufacturers to meet safety standards by compelling those manufacturers to obtain a certification from independent authorities such as UL (formerly Underwriters Laboratories).[1] This independence enables standards to evolve based on technology and market forces and minimizes conflicts of interests.

The purpose of functional safety systems is risk reduction. When machines malfunction, it is the responsibility of the functional safety system to force the machine into a "safe state." This means the safety system's availability (the probability the system will operate at a future time) is an important factor in creating risk reduction. Having a functional safety system fail just when it is needed is a bad thing. Because these functional safety systems need to be available 24/7/365, the design and testing of functional safety systems are controlled by international standards and verified by independent certification authorities.

1. On January 1, 2012, Underwriters Laboratories transformed from a non-profit organization to a for-profit company in the US. A new subsidiary named simply UL LLC, a limited liability corporation, took over Underwriters Laboratories' product testing and certification business. [*Wikipedia, "UL (safety organization)", retrieved March 2019*]

The diagram in Figure 1 illustrates the relationship between the equipment under control (EUC) in dark gray and the functional safety system in light gray. The functional safety system always observes the operation of the EUC. In Figure 1, the safe state is the removal of power from the actuators. If an action of the EUC violates a boundary creating a hazardous condition, the functional safety system cuts power to the actuators. A safety function can be thought of as an independent safety net: if the safety function fails, the EUC might continue to work—but it is now operating without a safety net.

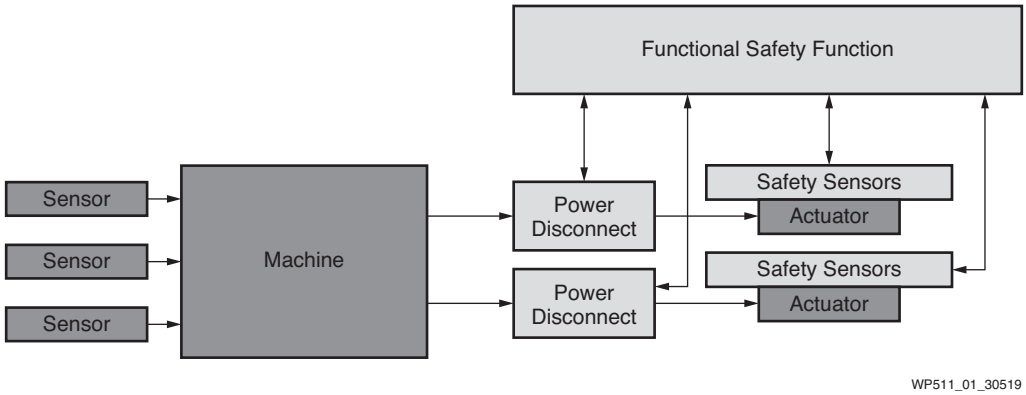


*Figure 1:* **Safety Function and Equipment Under Control**

## Three Blocks of a Safety Function

The safety function itself is usually composed of three blocks (Figure 2). These are: a sensor driving the inputs of a logic solver that then drives an actuator. All three of these blocks make up the safety function, sometimes called the "safety loop." Additionally, all three of these blocks are used to determine the Safety Integrity Level (SIL), or quality of the safety loop.

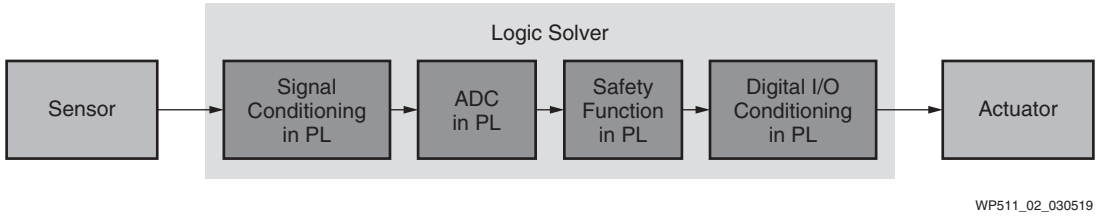


*Figure 2:* **Safety Function's Three Blocks**

## Failures, Risk Reduction, and Safety Integrity Levels

The industry defines a "failure" as a termination of correct service, whereas a "fault" is an abnormal condition that can cause a functional failure and an "error" is a discrepancy between expected/correct and actual value.

Failures fall into two major categories. Safe failures either do not affect the operation of the safety loop or cause the equipment to transition to its "safe state." Dangerous failures affect the operation of the "safety loop." In other words, faults create errors that can lead to failures.

## IEC 61508

According to the IEC 61508 (Functional Safety standard for electric/electronic/programmable electronic systems), there are four safety integrity levels identified as SIL1 thru SIL4. These levels objectively define the risk reduction of the functional safety system or safety loop that monitors the equipment under control. These safety integrity levels are defined in Table 1.

Two factors determine the safety integrity level. The first is systematic capability, a quality metric that qualitatively measures the amount of potential bugs in a design caused by human errors. This metric is determined by producing evidence that the prescribed process outlined in the standard was followed. The second factor is implementation of diagnostic measures used to mitigate random hardware faults. The safety integrity level of a system is determined by the lowest metric of these two factors.

*Table 1:* **Probability of Dangerous Failures/Hour**

| Safety Integrity | Required Safety Availability (RSA) | High Demand Mode (< 1 Year) | | |
|---|---|---|---|---|
| | | Average Frequency of Dangerous Failures per Hour (PFH) | Failures in Time Lambda (λ) | Risk Reduction Factor per Hour (RRF) |
| SIL 1 | 90–99% | 0.00001–0.000001 | < 10,000 Dangerous Failures/Billion Hours of Operation | 100,000–1,000,000 |
| SIL 2 | 99–99.9% | 0.000001–0.0000001 | < 1,000 Dangerous Failures/Billion Hours of Operation | 1,000,000–10,000,000 |
| SIL 3 | 99.9–99.99% | 0.0000001–0.00000001 | < 100 Dangerous Failures/Billion Hours of Operation | 10,000,000–100,000,000 |
| SIL 4 | 99.99%–99.999% | 0.00000001–0.000000001 | < 10 Dangerous Failures/Billion Hours of Operation | 100,000,000–1,000,000,000 |

## Safe States

The definition of the safe state is the responsibility of the machine manufacturer based on how the machine operates and the manufacturers hazard and risk assessment. In some cases, a safe state could be cutting main power to a motor or cutting main power to a machine so a built-in passive measure can take over to remove energy from the system. *This specifically depends on the type of hazard.*

For medical equipment such as an MRI machine, cutting power might work as a safe state if the patient can get extracted from the machine. For other types of medical equipment such as a patient monitor used in a critical care unit, a safe state of simply powering off the device might increase the risk to the patient. This problem is more acute for critical care machines such as a heart lung machine or a life-support ventilator where a safe state of powering down would most likely result in death or serious injury to the patient.

Two conditions can cause a system to go to its safe state. The first condition occurs when either the equipment under control or some external activity causes a hazardous condition to be detected by the safety function. The second condition occurs when the safety function itself uncovers a system fault in the safety loop.

For those situations where the EUC fails, causing the safety loop to drive the EUC into a safe state, redundancy is required in the EUC design itself for its continued availability.

For those situations where a safe state transition of the device is not acceptable and the safety loop fails, redundant architecture of the safety loop itself is used. This increases the availability of the safety loop, which in turn increases the availability of the EUC it is protecting.

## Availability

There is a high probability that all systems will fail over time. This is even more important for safety loops. (Possible rewrite: This is an even greater consideration for safety loops.) The two major causes for system failures are random hardware faults and systematic failures.

Functional safety systems designed to detect system failures must be highly available, because hardware failures occur randomly. To assist in designing these systems, a list of recommended architectures used to design these systems is outlined in the parent functional safety standard called IEC 61508, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. This standard introduces measures to detect an operational failure of the safety loop and to add redundancy for those cases where a Fail operational state is required for the safety function.

Systematic failures are mitigated by design process using well-tested, state-of-the-art measures, as outlined in the standard. The administration of this process must be independent, free from internal conflicts of interest within the system manufacturer. Many such processes are in use today. The V models shown in Figure 3 from IEC 61508 and in Figure 4 from ISO 26262 represent the state of the art in software design process. Each process shows design sequence on the left with verification sequence on the right.

The left side of the V model shows the project definition and design. The right side of the V model shows project test and integration with unit coding at the base of the V model. Actual unit coding never begins until the foundation for testing is complete.

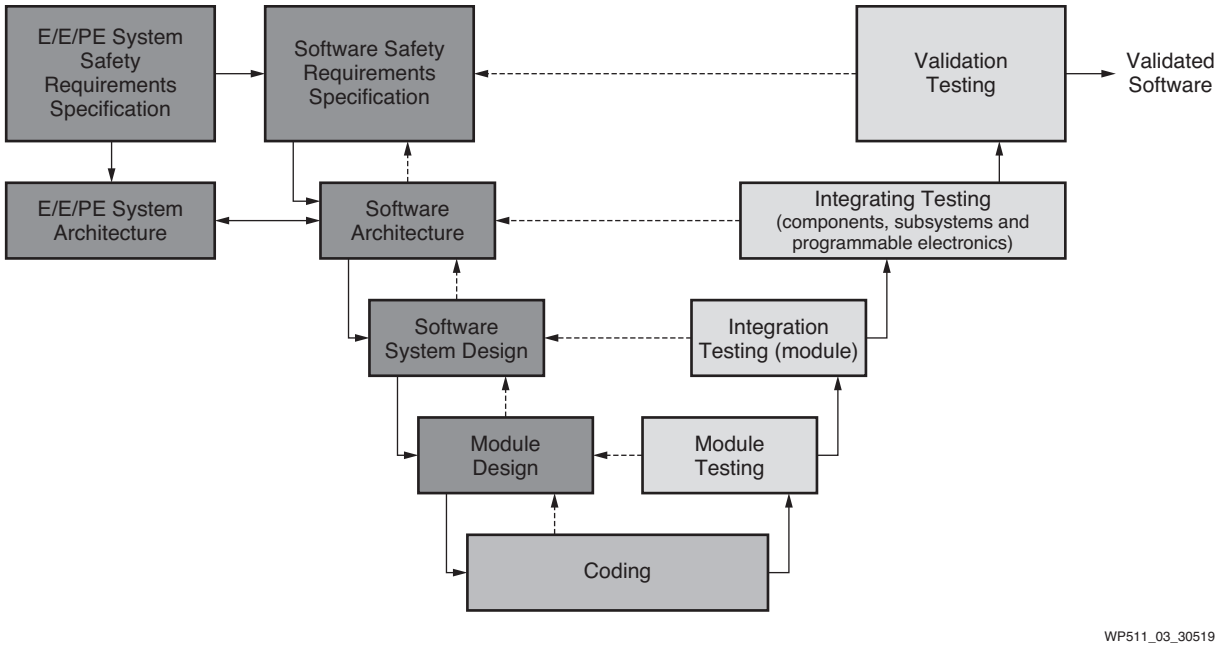


*Figure 3:* **Software Design Flow from IEC 61508:2010 Part 3**

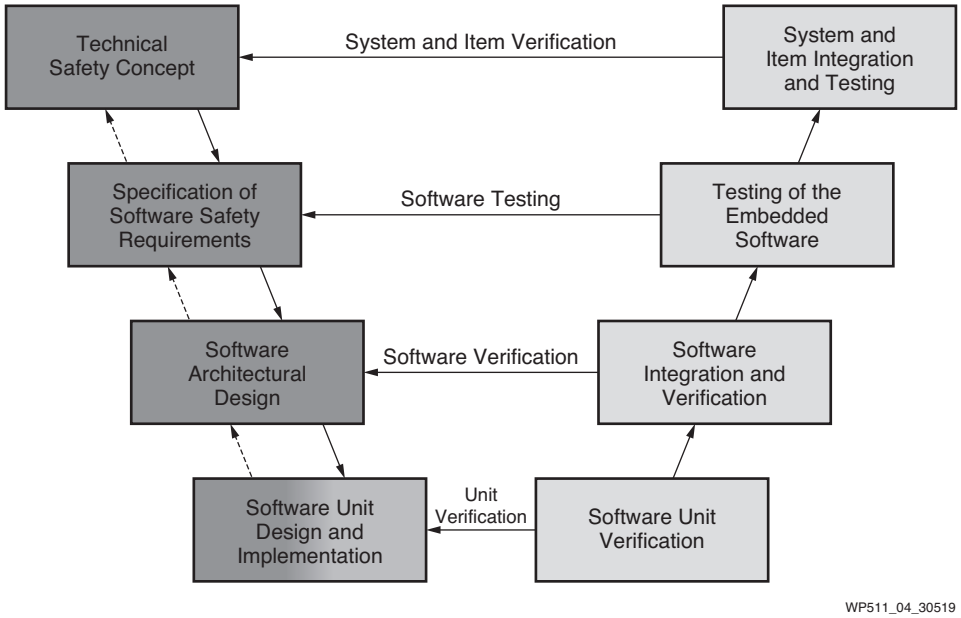


*Figure 4:* **Software Design Flow from ISO 26262:2018 Part 6**

This process begins with a complete understanding of what the end state looks like. This aspect of design generates a list of requirements. After these requirements are reviewed and accepted, an architecture specification is created. The architecture specification creates the functional blocks (divisions of labor) and interfaces that support the requirements. After the architecture is defined, specified, and peer reviewed, test requirements for each block and its interfaces are created. After these test requirements are peer reviewed and verified, coding for the test bench begins.

At this point, the design team is likely at the half-way point in the schedule, and not one line of application code has been written.

The next part of the schedule is allocated to coding the test bench. After the test bench is complete, the stage is set for success, because teams now know with a very high degree of certainty that the functions they code are correct by design. Finally, unit coding commences, followed by verification, with the balance needed for validation. The probability of a bug escaping this process is small, but it all goes back to comprehensive requirements. The more complete the requirements and corresponding test bench, the less likely a bug escapes.

The second consideration is random hardware failure. The measures used to detect and sometimes correct random hardware failures depend on how electronic devices fail. In the case of digital integrated circuits, failures can be caused by fabrication errors, external particle collisions, or metal migration wearing out the component. Random hardware failures can be permanent or transient, and depending on the nature of the fault, they can affect the machine's behavior. This might lead to a dangerous failure, or it might have no effect on the machine's operation—in which case, they are considered "safe failures." The measures called "diagnostic functions" are designed to detect these faults, and sometimes to correct them. Among the most widely used diagnostics are:

- **Fountain codes**, used in all mobile phone communication systems today to reassemble and correct transmission failures, which all mobile phones encounter due to random signal interference between the tower and the device
- **Protection of memory** with error correction code (ECC), which is used in every compute server in operation today to detect and correct corrupted memory data
- **Redundancy**, requiring an extra set of resources running in parallel with the primary function

Of these three, redundancy is the most complete type of diagnostic available, but it is also the most expensive. Referring to Figure 5 and Figure 6, the most popular types of redundancy are signified by the nomenclature "one out of two" (1oo2) and "two out of three" (2oo3).

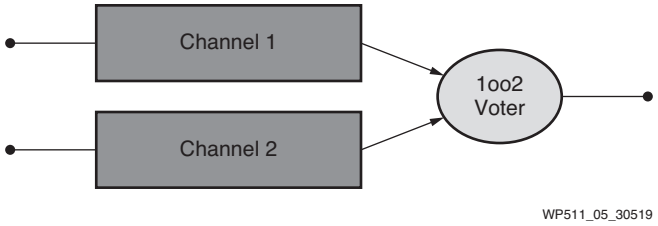


*Figure 5:* **One out of Two (1oo2) Safety Architecture**

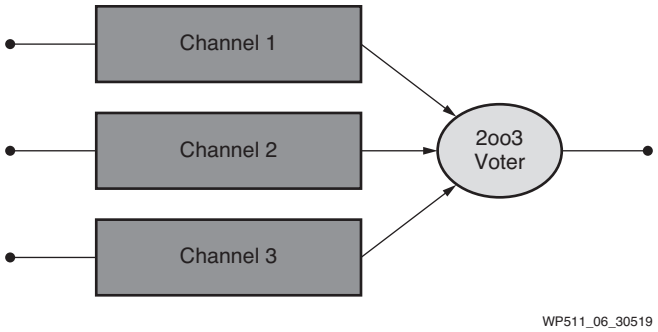


*Figure 6:* **Two out of Three (2oo3) Safety Architecture**

## Tying It All Together

In the example in Figure 7, a 1oo2 safety architecture is enhanced with diagnostics creating a "one out of two" diagnostics architecture (1oo2D) that can determine which of the two channels has failed. This architecture can be used in place of a 2oo3 architecture.

In this example, there are two safety channels: safety channel A is in dark gray, safety channel B is in light gray. Each channel has its own power source and clock source. The Zynq UltraScale+ device has power monitors and watchdog timers that can be placed in a separate domain to monitor the other domains in the device. These monitors are required for standards certification. Finally, each channel has a diagnostic capability greater than 98%, and each channel uses a different CPU and a design architecture that drives diversity. This diverse environment uses different architectures to enhance the capability of the system.

Looking at the overall architecture, a single fault (e.g., loss of one power supply, or clock generator, or memory failure, etc.) would be detected with a diagnostic probability of greater than 99% by comparing the results of each channel with those of the other. Once the fault is detected, the channel-based diagnostics inform the surviving channel and voter of its failure. The voter then follows the working channel's results.

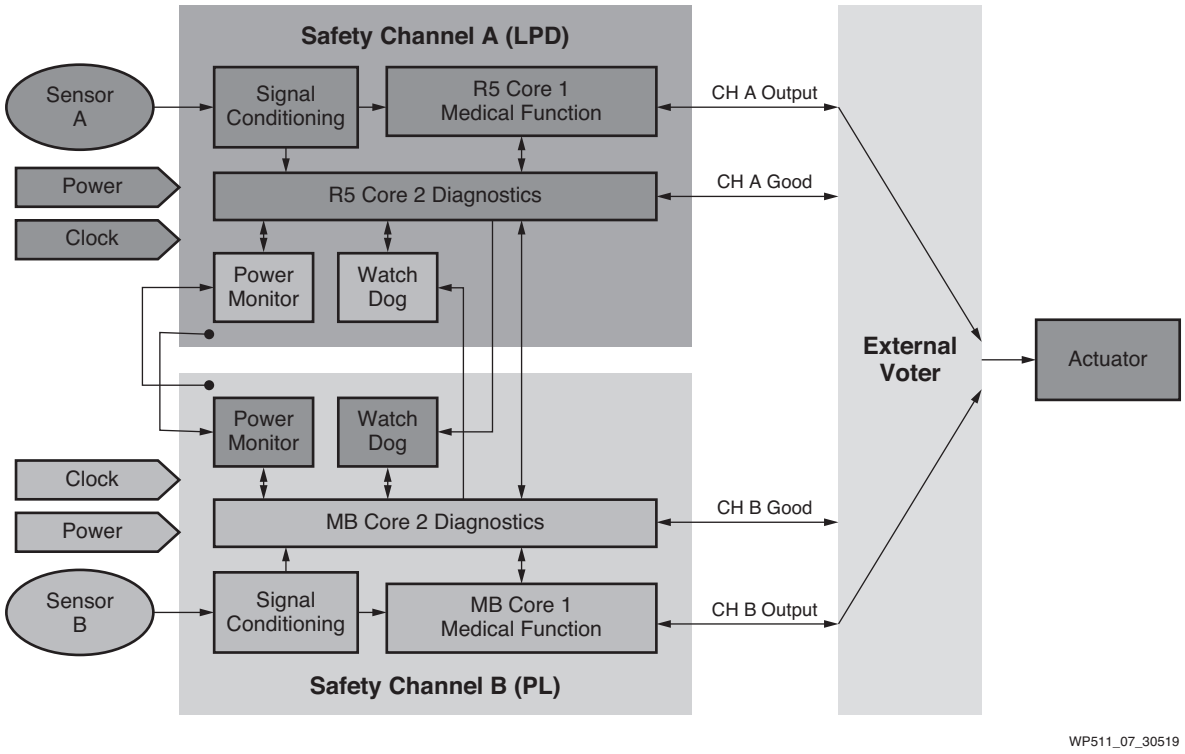


*Figure 7:* **Example 1oo2 Safety Architecture with Cross-Channel Diagnostics (1oo2D)**

## IEC 60601

The 60601 standards describe potential hazardous emissions that can cause harm, but address little regarding systematic methods of describing the safety loop quality (SIL) and its ability to put the device causing harm into a safe state. This risk management process is left up to the medical OEM and justified with documentation using the ISO 14971 standard. While this is a good start, the issue remains: What is the required quality of the safety loop that is used to mitigate the risk (as defined in Functional Safety SIL1–SIL4)?

The IEC 61508 standard provides guidance and processes that apply to any electrical/electronic or programmable electronic system without consideration to the final application of the safety loop. This means one can argue, given an identified risk, that the required risk reduction is achieved by the safety loop, which is designed to meet a certain standard SIL level.

## ISO 26262

The ISO26262 specification takes a much different approach in which the machine itself is designed to be inherently safe. For example, newer automobiles today have electronic emergency brakes. These brakes are replacing the older style mechanical braking system that is completely independent from the main brakes used to stop an automobile under normal driving conditions. The newer electronic emergency braking system (renamed Parking Brake) has integrated safety features that do not allow the system to engage when the car is moving at an unsafe speed. So, if the speed is above 5mph/8kph (the safe engage limit), the system sends a warning and does not engage. The rationale is as follows:

1. *The main braking system has redundant hydraulic circuits.*
2. *The actuator that drives the hydraulic system has a very low failure rate.*

The probability of total failure is historically very low, so the action was to remove the completely redundant emergency brake and replace it with an electronic parking brake with its built-in safety features.

This notion of examining the hazards based on the intended function is a known as its "safety goal(s)." Each safety goal is then implemented using various diagnostic measures and failure modes (Fail Safe vs. Fail Operational) that drive the architecture. The main function is designed in such a way that it is inherently safe.

The safety goals for a parking brake are:

1. *The system shall be resistant to inadvertent engagement*
2. *The system shall operate with a complete loss of main power*
3. *The system shall be resistant to operate causing a loss of control*

The design then accommodates these safety goals, issuing methods and measures to ensure these goals are met.

# Cybersecurity

Cybersecurity is a broad term that captures many concerns and risks around any device that makes use of digital hardware or software. Device manufacturers and cybersecurity professionals normally segment these concerns into two categories:

1. **Operational security:** Ensuring that the device operates as intended within defined operation envelopes while protecting system operation from being compromised by a bad actor.
2. **Device and software cloning:** Protection of intellectual property (IP) of a device and unique IP within the device while also preventing device cloning, which might look and feel like an authentic device, but which fails to meet all functions. Thus, it protects against financial loss due to IP theft, cloned devices, or system-level failures from unauthentic device introduction.

There is a misplaced trust that "unconnected" systems are immune to cyber-threats. This trust is misplaced because systems are truly unconnected only rarely. Most electronic devices, including electronic medical devices, have some form of connectivity to drive efficiency in the system in which those devices are used. In a medical device scenario, that might be a piece of diagnostic imaging equipment that is connected to the hospital network over which it shares images among radiologists, or a patient monitor that is connected to a nurse's station via the hospital Wi-Fi to enable a nurse to monitor multiple patients simultaneously. These points of unintended connectivity are used by bad actors as potential entry points to pivot into more critical and less protected equipment.

Even for those systems that are truly unconnected, attacks such as the well-known Stuxnet worm demonstrated the ability to introduce an attack through a physical interface (a USB flash drive, in the case of Stuxnet) to introduce compromising code that ultimately destroyed equipment and

operations. It is for this reason that cybersecurity and the threats against a device must be analyzed at a system level, as bad actors look to exploit the weakest link in a system.

For protecting any device from cybersecurity threats, there is no "silver bullet" or time-invariant solution. The array of threats, capabilities of hackers, and technologies used for waging cyber-attacks are all continuously evolving. It is nearly impossible to make a device that is impenetrable to the determined hacker; it can also be very costly to incorporate *all* cyber counter-measures available today. Therefore, it is important that the device developer manages the risk of cyber-threats and prioritizes the inclusion of counter-measures in the final design.

## Security Standards

Products in the industrial space have long sought a shared "figure of merit" for analyzing the cybersecurity requirements for a given product. Such a shared figure of merit allows for products to be certified by an independent agency as meeting a common security level definition. This goal was met with the roll-out of the IEC 62443 multi-part specification that defines five security assurance levels (SL) based on the cybersecurity threats a system is expected to encounter, along with an associated definition of product counter-measures that should be included in a system of a given SL. The SLs defined in IEC 62443 are summarized in Table 2 and are defined by the following:

- **System Identification:** A device's ability to be found/identified through connected and un-connected scenarios
- **Resources:** The amount of resources (time, financials, etc.) available to an attacker
- **Skills:** The general skills of the person(s) mounting a cyber-attack
- **Motivations:** The level of motivation for successfully carrying out a cyber-attack

*Table 2:* **IEC 62443 SL Ratings**

| Security Level | System Identification | Resources | Skills | Motivations |
|---|---|---|---|---|
| 0 | No protections | - | - | - |
| 1 | Casual | None | None | None |
| 2 | Simple means | Low | Generic | Low |
| 3 | Sophisticated means | Moderate | System-specific | Moderate |
| 4 | Sophisticated means | Extended | System-specific | High |

The general description mapping of the security levels is:

- **SL0**: Product with no intentional protections against cyber-threats
- **SL1**: Protection against casual cyber-criminals and script "kiddie" type attacks
- **SL2**: Protection against less sophisticated cyber-crime hackers
- **SL3**: Protection against sophisticated hackers with resources to support longer term attacks
- **SL4**: Protection against nation-state level attacks against critical infrastructure or similar assets

Several bodies now offer certification to the IEC 62443 standard. These certifications are available at the individual SLs and are offered by UL, TUV, and Exida.

## Operational Security

In protecting the operational security of a medical device, the prioritization of the elements of the security Confidentiality, Integrity, and Availability (CIA) triad must be considered. This impacts how system-level risk is managed and impacts the system's functional safety design considerations. This is also the point of variance between (a) how *information technology* (IT) devices react to cyber-attacks and (b) how *operational technology* (OT) devices do.

In life-critical machines (e.g., a life-support ventilator) actively supporting a patient, the overall operational availability of this OT system is prioritized. Therefore, if a system were attacked in a manner where data confidentiality was found to be compromised, the ventilator designer would choose to leave the system in operation until there was an opportunity to take the system out of service without compromising the system level mission of supporting the respiratory function of a patient. This contrasts with IT type protections in which a breach is often mitigated by immediately disabling the system upon identification of the compromise.

## Device and Software Cloning

A design-time consideration when selling a device is how to protect the hardware and embedded software from being potentially copied and/or modified by malicious third parties. This cybersecurity protection is key to protecting algorithms or other software-based product differentiation investments, as well as keeping the field free of product clones that lack full functionality and could potentially damage the product's brand. Malicious parties can create device clones that look identical to a company's end products and thus get inserted unwittingly into end systems by field engineers, after which the device acts as a gateway for higher-level system attacks.

## Device Security Design Practices

When designing any device, it is important to understand there is no one-size-fits-all security design solution. As IEC 62443 outlines, a designer must consider the types of attack a given product might see and then prioritize the security features that can be included for a given product budget. Security should be approached as "layers of protection" in a product life-cycle implementation that assumes regular updates as new security threats and technologies are identified. The risk of *not* doing so is that the general security stance of a system degrades with time, as shown in Figure 8.

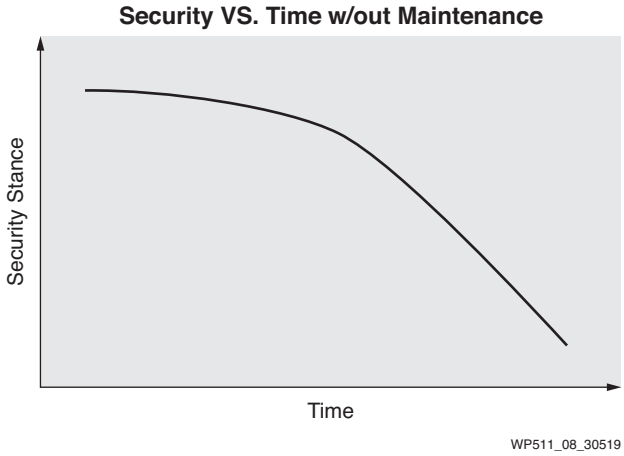


*Figure 8:* **System Security Stance vs. Time**

Cybersecurity experts recommend a "defense-in-depth" approach to protecting systems. Defense in depth simply means a layered technology approach that combines a series of technologies that protect against multiple vectors of attack. For example, if the only security on a house were the lock on the front door, a malicious actor would simply go to a window or to a back door that did not have a lock. Defense in depth means putting protections *throughout the system:* if someone does gain access to a part of the system, it is only in an isolated manner, and they are prevented from taking over the entire operation of the platform.

In maintaining a secure run-time environment, many security experts talk about maintaining a "trust chain" or "chain of trust" (Ukil, Sen, and Koilakonda, 2011). The trust chain concept is that one must maintain a tight hand-off between each stage of the device's operation and its software, rooted in a hardware-based security device. This "root of trust" is preferred to be hardware-based because software is always mutable. As the hardware releases the software reset, it should only boot known good software through operations such as secure boot and run only digitally signed and authorized applications.

The concept of a chain of trust is outlined in Figure 9.

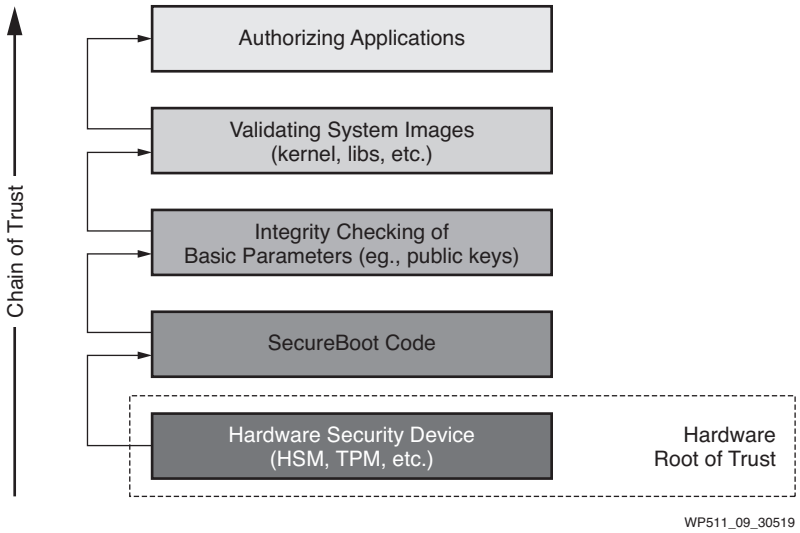


*Figure 9:* **Chain of Trust (Ukil, Sen, and Koilakonda, 2011)**

Modern digital systems should also build in the ability for updates over the life cycle of the product. This is critical, because cyber-attack capabilities are continuously evolving, and thus the technologies used today to protect a system are likely not sufficient in the future. Unknown vulnerabilities or new exploits known as "zero-day" are also a reality of deployed systems. This requires an ability to detect a compromised system, resilience of critical device operations, and integration of a secure update mechanism by which to remediate compromised devices. This device security life-cycle view is represented in Figure 10.

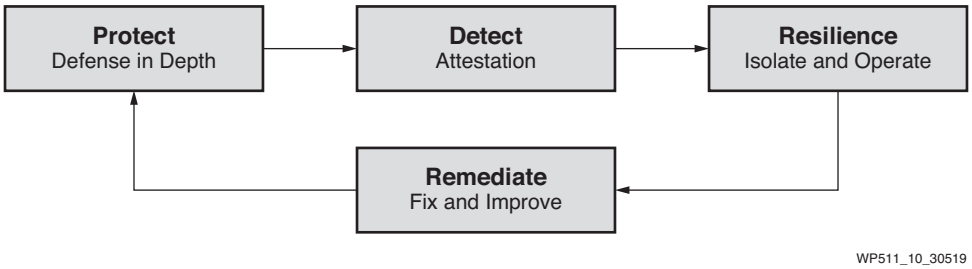


*Figure 10:* **Cybersecurity Life Cycle Design**

## Data Privacy / Information Protections

In the medical device space, designers also face the need to protect data that passes through them or that might be stored on them as outlined by regulations such as the Health Insurance Portability and Accountability Act (HIPAA) in the US or the General Data Protection Regulation (GDPR) in the EU. The electronic medical devices involved must implement security mechanisms to safeguard anything that is considered patient-identifiable data. The GDPR has started to define specific security requirements that include intrusion detection methods (the detect phase in Figure 10) and routine penetration testing of systems, best practices also outlined in IEC 62443.

For maintaining data confidentiality, platforms refer to "data-in-flight" and "data-at-rest" protections, which make use of cryptographic techniques for authentication and encryption. There is a spectrum of cryptographic algorithms, but the most popular and widely used in industry focus

on AES and RSA. Each can be used with an increasing key length, which makes the task of cracking the cryptographic function more difficult. The implementation trade-off here is that each increased key length requires additional processing capabilities. Standards such as IEC 62443 point to recommendations on cryptographic key lengths from NIST and similar bodies that make recommendations based on the length of time a device is to be fielded as well as the predicted general computation capability increases expected over that same time period.

# Xilinx Technologies

Xilinx has architected key technologies with functional safety and security in mind into the Xilinx® Zynq UltraScale+ MPSoCs to enable device developers to build a robust solution for implementing safe and effective medical devices. Furthermore, safety- and security-oriented silicon features are augmented by tool flow, IP, and software solutions as part of the Industrial and Healthcare IoT Solutions Stack.

In some cases, these technologies overlap functional safety and security; in other cases, they are specific either to functional safety or security.

For functional safety, Zynq UltraScale+ MPSoCs provide:

- **Three independent compute domains** with separate clock and power to mitigate common cause failures
- **Multiple temperature sensors** to detect operational boundary conditions
- **On-chip diagnostics (ECC)** to detect random hardware failures in user and configuration RAM
- **Systematic capability in the real-time compute domain**
- **Safety certified tools and methodologies**
- **Safety certified Zynq UltraScale+ MPSoC silicon and software**

For cybersecurity, Zynq UltraScale+ MPSoCs provide:

- **Hardware device security** - Immutable device identity, protected data store, and strong anti-tamper features
- **Secure boot** - Enforcement that only trusted firmware and software is booted on the system
- **Cryptographic accelerators** - AES and RSA offload engines for run-time communication, application digital signature verification
- **System monitor** - Ability to monitor system changes, measured boot support
- **Secure storage** - eFuse based internal storage for keys
- **Arm® TrustZone**
- **Authentic device protections** - Unique package markings
- **Hardware device security** - Security monitor, DPA protections
- **Unique device ID** - Protected and unique ID built into the HW
- **Protected customer keys** - e-Fuses and supply-chain mechanisms for bringing customer keys into the internal storage

- **Traceability** - Supply chain verification via counterfeit protections

With respect to functional safety and security, Zynq UltraScale+ MPSoCs provide:

- **Isolation technologies** to separate datapaths and hardware fault isolation.
- **Heterogeneous hardware design** to mitigate systematic failures to reduce the probability of a single bug effecting operation.

## Xilinx Functional Safety Technologies

Xilinx functional safety technologies address the two key aspects of function safety design. The first key aspect, systematic capability, is addressed with Xilinx certified tool chain and assessed heterogeneous-capable products. The second key aspect, random hardware fault tolerance, is addressed by a unique combination of diagnostics that include selective hardware redundancy within a compute domain, SRAM diagnostic capability, and redundancy across heterogeneous compute resources.

In addition to on-chip diagnostics, Xilinx incorporates unique design-specific and fabrication techniques to enhance single-event upset (SEU) immunity at the device level.

## Xilinx Cybersecurity Technologies

The Xilinx cybersecurity platform technologies are best-in-class and widely used in defense industries where cybersecurity is measured against very stringent requirements. Xilinx has been at the front of both run-time and supply chain security technologies because of these defense-driven requirements. Medical industries can also benefit from these inherent security functions.

Xilinx views security as a pyramid of protections with each layer building on top of the strength of the layers below it, as illustrated in Figure 11. Xilinx security starts with the protections put in the supply chain to ensure that only authentic devices propagate through the proper channels and are ultimately used in the build of customer devices. Next, the silicon implements protections for commonly compromised debug interfaces, such as JTAG, device tamper detection, and penalty enforcement. In the boot and configuration layer, Xilinx provides boot-time authentication and encryption functions as well as dynamic power analysis (DPA) protections to prevent boot-time attacks that might compromise a device or potentially leak secrets. Most of these security features are provided automatically by the Xilinx platform, and others must be enabled through configuration of the user application running on the platform.

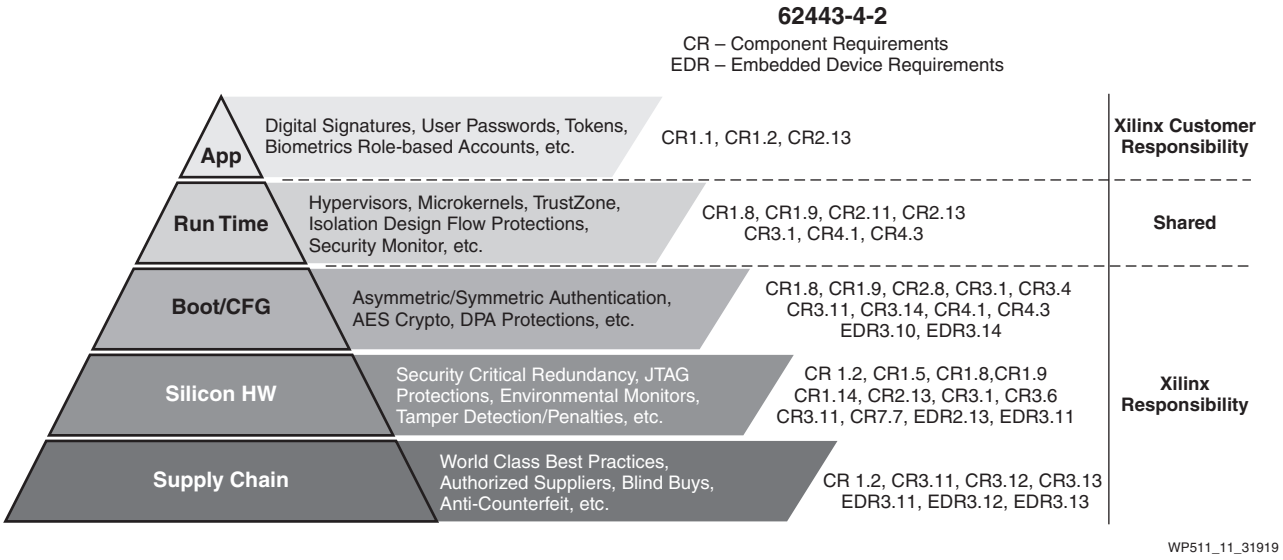


*Figure 11:* **Xilinx Security Pyramid**

The Zynq UltraScale+ MPSoC provides several options for run-time application security isolation through capabilities such as Arm TrustZone, hypervisors, and Xilinx Isolation Design Flow (IDF). There are also optional runtime monitoring capabilities such as the Xilinx platform Security Monitor (SecMon) and FPGA-based software appliances capable of monitoring the integrity of software run time. The same cryptographic hardware used to protect the early boot stages of the platform, such as the AES and RSA hardware accelerators, are also exposed to the user space after a system is booted. These hardware offload functions provide significant benefit to applications that require cryptographic functions and can offer AES acceleration up to 53X and RSA acceleration of up to 9X.

These security features map to requirements called out by IEC 62443. Figure 11 shows a high-level mapping of the identified Xilinx security features and the corresponding IEC 62443 standard. These cybersecurity requirements are equally applicable and recommended in protecting medical devices as cited by the FDA Recognized Consensus Standards, which specifically cite IEC 62443.

# Summary

As the regulatory agencies and design methodologies around medical device development describe risk management, this white paper relates the topics of functional safety and cybersecurity to assessing and managing product risk for the device manufacturer. Some of the best practices from the industrial space in addressing the risks associated with product safety (e.g., IEC 61508) and cybersecurity protections (e.g., IEC 62443) should be leveraged when designing a medical device.

This white paper outlines how functional safety design practices and Xilinx capabilities can be applied in the design of medical devices to improve safety, assess risk better, reduce time-to-market by one-shot regulatory approval, and reduce chances of product recall. Xilinx believes that regulators and design engineers can see the structured functional safety design processes as having a positive impact on overall product design, total product costs, and ultimately on patient safety. The discussion of safety and security cannot be separated, because without proper cybersecurity protections, the safety elements of a system could be compromised from their intended operation. The topic of cybersecurity was described for both operational integrity and supply chain protections for digitally controlled products and is agnostic of their end industry.

The white paper also outlines some of the Xilinx technologies and security life-cycle considerations that should be part of any digital product, and describes how Xilinx SoCs can provide a robust hardware root-of-trust by which to build a secure medical device. It also provides a flexible platform for implementing safety functions and functional redundancy within a single device. This consolidated solution of security and functional safety capabilities within a single SoC can provide significant design time and product cost savings.

Additional Xilinx documentation resources helpful for making a deeper dive into specific topics, including design guides on using Xilinx technology to address functional safety and cybersecurity design requirements, are outlined in the Related Material section.

# Related Material

The following are related material for specific topics identified in this paper.

## Functional Safety

Xilinx Functional Safety Website - https://www.xilinx.com/products/technology/functional-safety.html

Xilinx Functional Safety Working Group - https://www.xilinx.com/products/technology/functional-safety.html#functionalSafety

Xilinx White Paper WP495, *Using Zynq-7000 SoC IEC 61508 Artifacts to Achieve ISO 13849 Compliance*

Xilinx White Paper WP461, *Xilinx Reduces Risk and Increases Efficiency for IEC61508 and ISO26262 Certified Safety Applications*

Xilinx White Paper WP412, *Xilinx Isolation Design Flow for Fault-Tolerant Systems*

## Cybersecurity

Xilinx Design Security Website - https://www.xilinx.com/products/technology/design-security.html

Xilinx Security Working Group - https://www.xilinx.com/products/technology/design-security.html#workingGroup

Xilinx White Paper WP467, *A FIPS 140-2 Primer for the Zynq-7000 All Programmable SoC*

Xilinx White Paper WP468, *Leveraging Asymmetric Authentication to Enhance Security-Critical Applications Using Zynq-7000 All Programmable SoCs*

Xilinx White Paper WP429, *TrustZone Technology Support in Zynq-7000 All Programmable SoC*

Xilinx White Paper WP493, *Key Attributes of an Intelligent IIoT Edge Platform*

Xilinx Application Note XAPP1323, *Developing Tamper-Resistant Designs with Zynq UltraScale+ Devices*

Xilinx Application Note XAPP1267, *Using Encryption and Authentication to Secure an UltraScale+ FPGA Bitstream*

Xilinx Application Note XAPP1175, *Secure Boot of Zynq-7000 All Programmable SoC*

# Revision History

The following table shows the revision history for this document:

| Date | Version | Description of Revisions |
|---|---|---|
| 04/15/2019 | 1.0DRAFT | Initial Xilinx release. |

# Disclaimer



# Automotive Applications Disclaimer