Development of Safety Related Systems

A Lattice Semiconductor White Paper
July 2015

Lattice Semiconductor
7th Floor, 111 SW 5th Avenue
Portland, Oregon 97204 USA
Telephone: (503) 268-8000
www.latticesemi.com

Development of Safety Related Systems

The increasing degree of automation brings a lot of comfort and flexibility in all areas of our daily life, but these added benefits also require us to be more cognizant of related safety concerns. The industrial segment boasts highly sophisticated production lines that are expected to be easy to use, offer a high degree of comfort and safe to operate.

This whitepaper will highlight that a technical system should not increase the safety risk for humans and environment over the tolerable risk. It is impossible to have no risk and on the other hand to point out that tolerated risk is a question of acceptance. Each domain has its own definition of acceptable risk levels and justifies this with different safety levels. For electrical and programmable systems, a common sense of functional safety is covered in a set of standards. These standards are adapted to the different applications, but they conform to the same philosophy. This safety philosophy for electrical and programmable systems is driven by the IEC61508 standard.

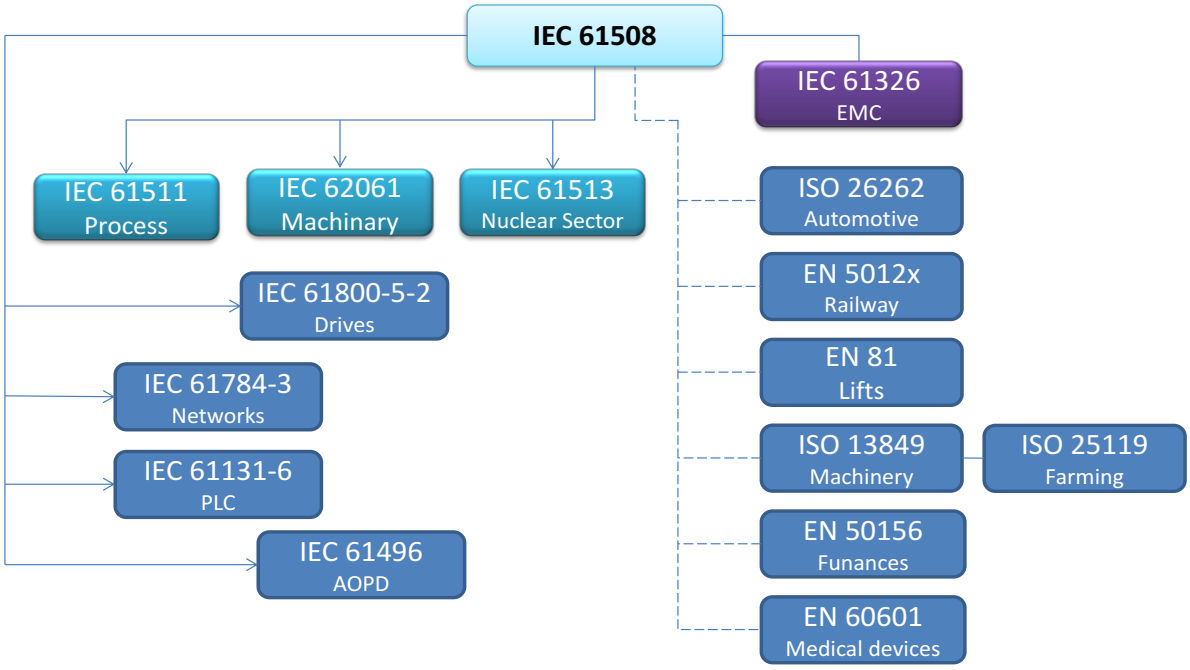

Figure 1: Overview of Typical Functional Safety Standards

The main approach of the IEC61508 covers the whole lifecycle of the system and focuses especially on points where a system can become dangerous. The idea is to start from scratch and design the system in the safest way possible. The typical method of doing this is to add additional elements, which monitor the normal function

and control the system for any abnormal situation. This concept is often used in the industrial automation or process industries. The IEC61508 defines this principle of functional safety: low demand, high demand mode, or continuous mode. The mode is determined by how often the safety function will be used during an annual interval.

Also, the way to design the standard control function under functional safety aspects is an option, for example in aircrafts, automotive or in household machines. This principle is covered by the IEC61508 in the continuous mode.

The normal way is to start with the analysis of all possible critical issues that can affect a system. All identified issues must be weighted with parameters as exposure time, severity of injury and the possibility to escape from the harm. This method is a typical risk analysis and this must be done for the equipment under control, without any kind of additional electrical protection system. This has to be done for all lifecycle phases of a system. By using the risk graph, the risk analysis will provide the required Safety Integrity Level (SIL). In the case of using ISO13849, the risk graph will provide the required Performance Level (PL). This is similar to SIL, as both define a safety level. For the design of safety elements like a Safety PLC, Safety Drive Inverter or a Safety Encoder, it is normal to get the required safety level from the machine builder. The required safety level is specified as a requirement to reduce the risk to the tolerable risk. The SIL has to be fulfilled for a safety function and the safety function will be composed by a chain of safety elements or with safety devices. This means that a single element cannot fulfill the SIL by itself, it can only be capable of fulfilling the required SIL in a safety chain.

To fulfill SIL requirements, the standard necessitates taking care of two kinds of failures. The first group contains stochastic failures and covers all types of random failures in the hardware, and the second group contains all systematic failures.

Stochastic Failures

The stochastic failures will be calculated by different parameters like failure rate of components (λ), diagnostic coverage (DC), Hardware Fault Tolerance (HFT), common cause factor (β) and test intervals. The fact that safety isn't a given and if the system fails to the unsafe state, the IEC61508 is only interested in the undetectable, unsafe failure rate and specifies limits according to the used modes

with a PFD (Probability of Failure on Demand) for the low demand mode and PFH (Probability of Failure per Hour) for the high demand and continuous modes. For example, a SIL 3 safety function is limited to only one dangerous failure in 1,000 years. On the other hand, a low demand safety function (PFD) should not have a failure in average of 1,000 safety demands. As additional acceptance criteria, it requires the fraction of the safe failure in a defined range depending of the HFT and the SIL with the name SFF (Safe Failure Fraction).

SIL	PFD [per demand]	PFH [per hour]
1	0.01 – 0.1	0.0000001 – 0.00001
2	0.001 – 0.01	0.0000001 – 0.000001
3	0.0001 – 0.001	0.00000001 – 0.0000001
4	0.00001 – 0.0001	0.000000001 – 0.00000001

Table 1: PFD and PFH Values of Safety Integrity Levels

SFF	Hardware Fault Tolerance		
	0	1	2
< 60%	Not allowed	SIL 1	SIL 2
60% to 90%	SIL 1	SIL 2	SIL 3
90% to 99%	SIL 2	SIL 3	SIL 4
> 99%	SIL 3	SIL 4	SIL 4

Table 2: Safe Failure Fraction in Relation to Hardware Fault Tolerance

The failure rate for dangerous failures can be decreased by the implementation of diagnostics and redundancy. The degree of redundancy is valued with Hardware Fault Tolerance (HFT). A system with a HFT of 0 can become dangerous if a single failure will happen. A system with a HFT of N is immune against N-1 failures. If the diagnostic unit is able to detect malfunctions and can bring the system into a safe state, partial diagnostic coverage will decrease the critical values ($\lambda_{du} = \lambda_d \cdot (1\text{-}DC)$). In addition to the failure rates of components by malfunction (Hard-Errors), the designer has to monitor failures by Soft-Errors. The Soft-Error rate is a very critical point in the calculation because it can increase compared to the failure rate by Hard-Errors.

Lattice Semiconductor, a leading manufacturer of FPGAs, provides their customers with the failure rates and the Soft Error Rates for all safety recommended components.

Avoiding of Systematic Faults

Another key imperative is to avoid systematic faults as much as possible, which is dependent on the required SIL varied by the used methods in sum and deepness. All phases of the product lifecycle have different requirements against systematic failures. The specification outlines the design process as follows: implementation, verification and validation. For a well-structured design, the V-Model is recommended. Even for software design and FPGA programming, the standard specifies the single design phases and the verification phases.

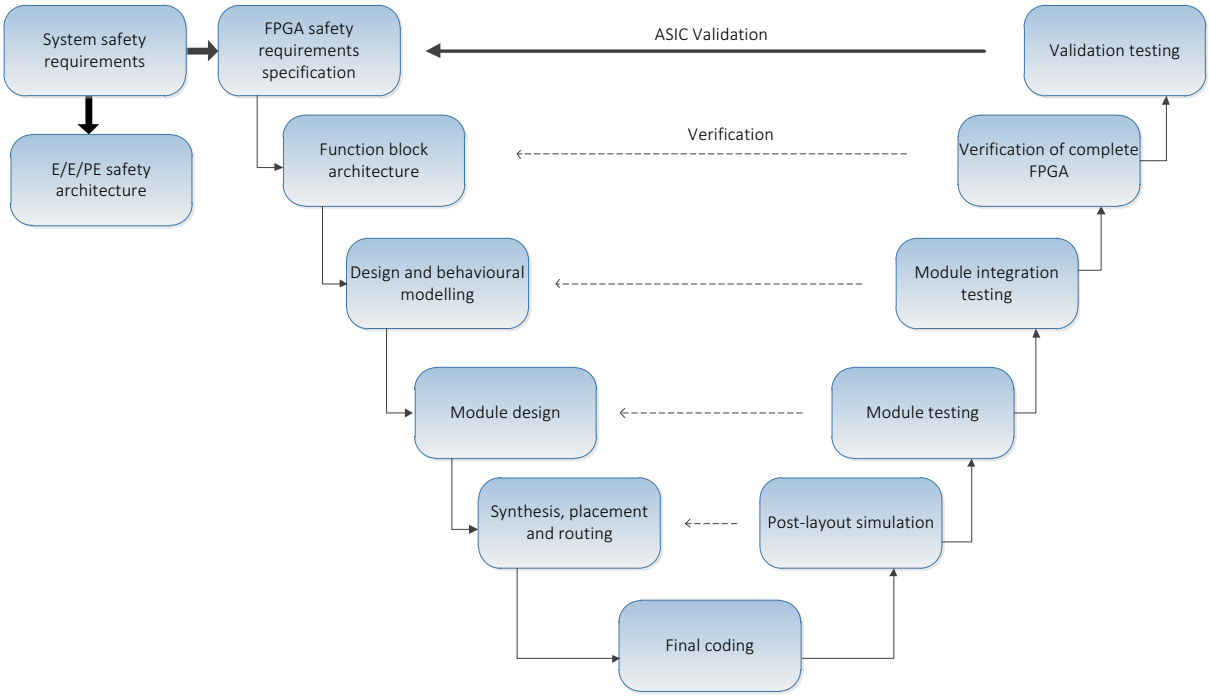

Figure 2: V-Model of FPGA Design Following IEC61608-2:2010

Overall, technical aspects of functional safety management are the most important actions to avoid systematic failures. Safety management activities include a detailed plan of all design and verification steps, before the real development starts. That means the safety manager has to have a clearly defined project plan.

Documentation Management

The management of documents must be well specified in the safety project. It describes how documents will be handled, where documents will be stored, how they will be released, who is allowed to change documents, who has access and what the limitations are for each team member. Version control of the documents should be an automated process performed by the tools.

Requirement Management

Managing all the requirements is an essential component of a safety project. Each safety requirement should be traceable through the whole project. The goal at the end of a project is to confirm that all requirements have been correctly implemented. Tests can be performed to validate that the issued requirements are able to reduce the risk. In this context, it is necessary to organize the requirements according to the actual data, completeness, and consistency. One must show how the requirements will be derived from the architecture to the modules into implementation and then from the module tests to the integration, back to the system tests.

Organization and Responsibilities

A structured team with clearly defined roles and responsibilities is critical to ensuring that all tasks are performed seamlessly and efficiently. The hierarchical order should outline the team and group leader. All contact information shall be available and especially - in case of distributed teams - a way for team members to communicate and collaborate must be defined. It is necessary for the reviewer, developer and tester to perform these roles independently.

Definition of Methods

Depending on the required SIL, the standard provides a variety of tables, sorted for each life cycle, as recommended or highly recommended methods, as fault avoidance tool. Before the development starts all techniques covering design and verification methods should be chosen. All techniques are listed in the IEC 61508 part 2 and part 3. Part 2 covers all hardware aspects and all ASIC or FPGA aspects.

Part 3 covers all software aspects. It seems a bit confusing that FPGA programming is also covered in part 2 of the IEC61508, but this is not a technical question rather a question of responsibilities in the standardization group. This is irrelevant since the way to develop FPGA software is similar to the way of developing microcontroller software. The differences are given by the technology. For instance, the technique of simulation is more common for FPGA design, whereas microcontroller design targets hardware with debugging tools.

Design phase	Ref.	Measure	SIL 1	SIL 2	SIL 3	SIL4
Design entry	1	Structured description	HR	HR	HR	HR
	2	Design description in (V)HDL	HR	HR	HR	HR
Synthesis	17	Internal consistency checks	HR	HR	HR	HR
	18a	Simulation of the gate net list, to check timing constraints	R	R	R	R
Placement, routing, layout generation	28b	Comparison of the gate net list with the reference model (formal equivalence check)				
	29	Design rule check				
Manufacturing	32	Application of a proven in use process technology	HR	HR	HR	HR
	33	Application of proven in use device series	HR	HR	HR	HR

R: Recommended **HR: Highly Recommended**

Table 3: Abstract of Table F.2 IEC61508-2

Table 3 shows an abstract of the list of techniques to reduce systematic faults in FPGA design. Similar tables also exist for general hardware and software design. The principle to handle these tables is always the same. Methods which are marked with “HR” must be used. If not, this decision must be rationalized. Methods with “R” marked should be used if possible.

Validation and Verification Planning

The V&V activities must also be planned ahead of a safety realization. All design phases have to be verified with the chosen methods from the fault avoidance document. The planned methods must explain how the real use case looks for the current project. For example, a planned measure could be the static code analysis. The V&V plan addresses all SW-Modules, which will be verified by code checker, describes the process of using this tool and explains how the results will be handled, analyzed and documented.

Another example is checking of net lists in FPGA design. The first action is to define that this action must be done, who will perform this task, and what are the input and output documents. The next action will be to define the used tool set for this task and how the release process looks.

The V&V plan can be used as a check list for all validation and verification activities and will give a complete overview for the completeness of all planned activities.

Tool Qualification

Regarding the intensive use of any kind of SW tools in all phases of the life cycle, all tools that will be used for realizing safety components must be analyzed according to their impact to the safety functions. This means first, all tools have to be listed and then all SW tools must be classified according to tool criticality level (T1, T2, T3 according IEC61508-4:2010).

Class	Definition	Lattice tools
T1	This class contains all tools with output which have no direct or indirect impact of the safety function, e.g. documentation tools.	
T2	In this class are all tools with output which can have an indirect impact to the safety function. Typically supports such kind of tools the verification by tests or simulations. Errors in such tools can falsify the test results but will not manipulate the safety function directly. These tools are for instance test coverage measurement tools, static analysis or debugging tools.	Diamond 2.1 Aldec Active-HDL Simulator Diamond 2.1 TRCE Diamond 2.1 Power Calculator Diamond 2.1 LDBANNO
T3	In this class are all tools located with output, which can directly or indirectly contribute to safety function. These tools include for example a compiler or synthesis tool, place and route tools that incorporate libraries into code or tools and further techniques.	Diamond 2.1 LSE Diamond 2.1 Synopsis Synplify Pro Diamond 2.1 EDIF2NGD Diamond 2.1 NGDBUILD Diamond 2.1 MAP Diamond 2.1 PAR Diamond 2.1 BITGEN

Table 4: Tool Criticality Level

Table 4 shows the definition of the standard and the list of Lattice tools, which are assigned to the classes, when using FPGAs for safety related tasks. In a real project, this list has to be completed with all other used software tools. To know the criticality of a tool in the project is nice, but will not result in a safer system. That's why additional tasks are necessary, such as tool qualification to gain confidence of the used tools. Confidence can mean validating or knowing the errors of the tool. If the tool fulfills the specification correctly and the user has the validated evidence, it can be used without any restrictions. If the tool does not work according to its specification, the user needs the information about the errors and has to specify workarounds for that. If the analyzer comes to conclusion that the output of a tool is not trustable or the specifications are not detailed enough, the user has to specify additional measures to detect such errors.

To analyze may bring potential problems to the user. Problems will appear if the user doesn't have enough knowledge, experience or data of the tool. In this case, it is helpful if the manufacturer of the tool supports the customer with all necessary data, or even better, if the tool manufacturer provides data and documents, which are certified and approved by an independent organization.

Lattice initiated an audit about their tool-suite "Diamond 2.1" for safety designs according to IEC61508 up to SIL 3 by TÜV Rheinland. This gives a safety project team the advantage to use this tool chain together with all related documents and safety manuals without extra validation activities. This save costs and time associated with the project and simplifies the decision to use Lattice FPGAs for safety applications. Combined with the tools, Lattice is able to provide production-proven FPGAs as reliable and approved failure rate data. The certificate by a Notified Body brings trust to the assessor and speeds up the type approval process.

Workflow of Safety Product Design

In addition to all functional safety management activities, the safety design flow also has to be done to ensure product safety. Let´s assume the job is to build a SIL 2 or SIL 3 device.

The first task is to start with the safety concept. The safety concept can scratch a rough architecture with details like single or dual channel structure, communication paths, input and output interfaces, power supply and so on. The safety requirement specification (SRS) will be derived from the safety concept and the product specification. To stabilize a concept, it is recommended to do the first FMEA (Failure Modes and Effect Analysis) on block level. Usually the FMEA results advances the list of requirements. For supporting a structured analysis, the IEC61508 provides some fault models of complex electronic parts and recommends fault detection methods in failure control methods. In dual or multichannel structures, common cause failures must be identified and eliminated. For a safe design, the environmental and the EMC conditions also become very important. Depending on the application, additional standards can be valid and have to be observed. If all requirements are fixed, the development can start with a top down design from the architecture to the module. Always remember to create the specification and the description of all design steps because these are the input documents which are

needed for all review and test phases. For a smooth project flow, all test activities should start in parallel to the development.

After the schematics and the circuits are derived, the component FMEA has to be completed and the calculations of the safety parameters can continue. The component FMEA will also be used as an input information of a Fault Insertion Test (FIT) specification. The software has to be implemented according Figure 2.

After finishing all tests including system and type tests, the design should fulfill all requirements for safety. Last but not least, all safety related aspects have to be included in the user manuals of the new product.

The workflow and the design can contain difficulties by doing this for the first time. However, good planning and knowledge in safety design results in quality and safe products on the market. In order to reduce the initial costs, innotec (http://www.innotecsafety.com/) can help from the development of the concept to the integration by consulting in safety design.