Multi Soft-Core System

Scope

This document deals with the development of a safe controller architecture.

Applications

The systems usage focuses applications with high safety requirements. This covers for example oil and gas industry, aviation industry or automotive industry.

Development Process (Systems Engineering)

The development process of this project is based on the principles of systems engineering. This represents a structured development method of defining requirements, interfaces and functions of the system and the systems sub-components and afterwards verifying these steps. This procedure is also known as "V-Model" and performs the following steps: At first the Definition and analysis of the requirements for the system takes place. From these requirements it is possible to develop a high-level system architecture and further develop the sub-components of the system. The follows the development of a high-level software architecture and afterwards the development of the software for the sub-components. Then follows the Implementation and verification of the units and the integration and validation of the system.

Safety Aspects

The safety of a system defines the probability that the system fulfills a specific function on demand.

To guarantee the safety of a programmable electronic system, the safety standard IEC 61508 is applied. This standard contains basic requirements for the systems' software, as well as methods and procedures to verify the safety of a system. The difficulty and amount of verification procedures and documentation significantly increases, the higher the desired safety level is. Thus, it is necessary to spread the different safety levels within the system from each other. This is possible by applying a multi-core controller architecture, whereby every controller has its own safety level. Every controller is verified individually. Therefore, the verification effort is comparatively kept low. It is also recommended to perform diagnosis of the system by monitoring critical values. To reach a safety level, the design of the verification procedures is recommended by the IEC 61508 and contains methods like modular tests, failure injections and documentation over the whole project.

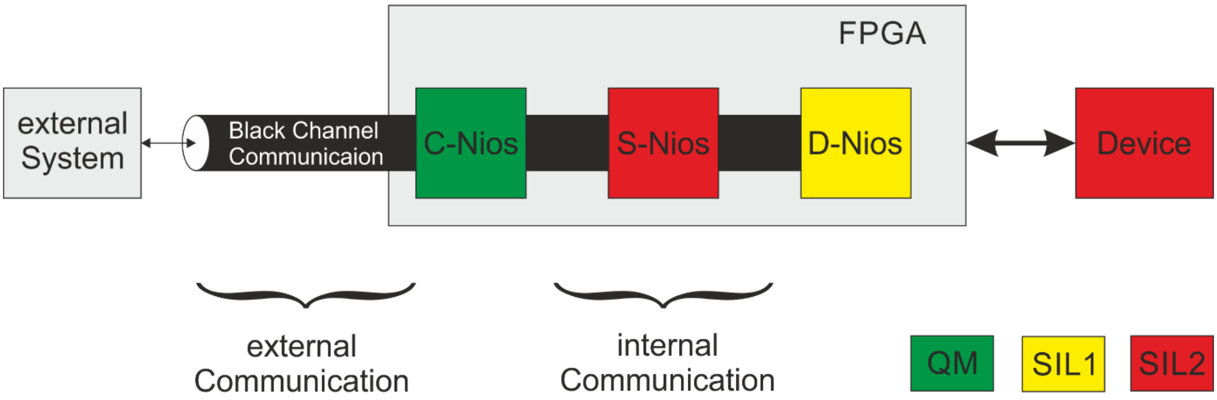
The term "SIL" indicates "Safety Integrity Level" and describes a measuring unit for the safety of the system. The lowest safety level is SIL1. The higher safety level is SIL2. The non-safe level is QM (Quality Management), which does not apply to safety standards, but only regular standards like ISO 9001. The PFD (Probability of failure on demand) and PFH (Probability of failure per hour) indicate the failure probability depending on the SIL:

Safety-Level	PFD	PFH
QM	-	-
SIL1	0.1 - 0.01	0.00001 - 0.000001
SIL2	0.01 - 0.001	0.000001 - 0.0000001

Further it is necessary to apply coding styles to the software to guarantee a uniform programming style. This style also contains recommendations, like restricted usage of pointers, restricted usage of interrupts or restricted usage of the "goto" command.

Safety Controller

The Safety Controller performs complex calculations for the system. Further its function is to guarantee a black channel communication with the external system. The safety controller is separated into several controllers. Those controllers are realized as so called “Nios”. Those are emulated microprocessors on a FPGA. Every Nios holds an individual safety level. The high-level architecture of the Safety Controller is shown in the graphic below:



The C-Nios (Communication-Nios) is based on QM level and holds the non-safe communication protocol in form of a software stack, for example CANopen or ProfiNet. The S-Nios (Safety-Nios) is based on SIL2 and contains the safe logic of the controller, as well as a safety protocol for the communication. The D-Nios (Diagnosis-Nios) is based on SIL1 and holds the diagnosis functionality of the controller.

The Black Channel Communication is split into an external and in internal communication. The external Communication is realized as a serial protocol, e.g. CANopen-Safety or ProfiSafe, which depends on the external system. The internal communication provides the message exchanges between the controllers with a safe handshake protocol over shared memory.

The Device, which is controlled by the Safety Controller, is for example a safe electric motor (actuator), or a sensor with safe data content.

Further Design Possibilities

Theoretically it is also possible to realize the upper system with three several microprocessors without a FPGA. However, the advantage of using an FPGA with Soft-Cores is the high performance. Another advantage is the high configurability of the system architecture. For example, it is possible to include several Safety-Nios.

Another possibility is to execute the different Nios on different FPGAs. In this case, the internal communication is no longer located within the internal RAM of the FPGA, but within an external RAM, which connects the two FPGAs.

It is also possible to realize the internal communication with help of a serial protocol instead of message exchange over a shared memory.