Safety and Mission Critical Designs with Zynq

Biography DI Peter Thorwartl

- I was born in Vienna in 1968

- College for Communication Engineering
- Degree in Electrical Engineering
- Graduated 1996, Vienna University of Technology - Professors Paschke, Pötzl,Prechtl,Hoffmann, Mecklenbräuker
- 1988 First FPGA in Austria with Xilinx FPGA
- From 1991 to 1997 Teacher at the College for Communications Engineering in Vienna
- During his studies teaching assistant at the Vienna University of Technology
- 1994 - 2000 lectures about programmable logic at the Vienna University of Technology
- 1997 Alternative Service at Allgemeines Krankenhaus Vienna (Austria's largest hospital)
- 1996 to 2000 lectures about FPGA Design at University of Technology
- 1997 Foundation of the so-logic GmbH & Co KG
- 2001 Xilinx Training Center for Austria and Eastern Europe
- 2003 to 2004 lectures about FPGA and Computer Architecture at University of AppliedScience in Vienna
- 2008 lectures about Embedded Systems University of Applied Science in Wiener Neustadt
- 2010 Lectures about Digital Signal Processing At University Campinas In Brazil
- 2013 lectures about Embedded Systems University of Medical Engineering in Linz

I learned my experience at the leading university institute in Central and Eastern Europe. Besides my scientific work he was teaching at the university and technical schools.

so-logic www.so-logic.net

so-logic activities

- **Training**
  - ILT Public Workshops with our partners
  - ILT and VILT In House Training for customer with special adoptions for the customers requirements
  - Training on the Job (Lab examples are part of the customer project) also as private VILT
- **Development**
  - Code Review / Feasibility Studies
  - Design Consulting / Project Reviews
  - Complete Product Development / Environmental Testing / Certification
  - IP Core Development (SATA, Ethernet, JESD204, HMC, Video Compression, …)
  - Test Adapter (SATA, SFP, PCIexpress, SATA)
- **Education**
  - Teaching at Technical Universities
  - Teaching at Technical Schools
  - Tutorials On line (HTML, PDF, ..)

so-logic experiences

- Training in many countries
- Swiss, Italy, Slovenia, Slovakia, Turkey, Brazil, South Africa, Oman
- Xilinx Xpert Partner since 1998
- IP Core Development
- SATA, PCI express, Serial RapidIO DMA, Video, Audio
- Offered only with consulting
- Circuit and PCB development
- Offer training around Xilinx FPGA design
- Mentor, Mathworks, National Instruments, Agilent
- Course Development for Xilinx
- Open Source Embedded Linux, SystemC, DSP
- socius owner hardware development and prototyping platform

so-logic Partner Ships

- **Xilinx** Leading FPGA Vendor Training Center and Design Partner
- **Mathworks** Matlab and Simulink Mathematical Computation Training Partner
- **Keysight** Measurement and Test Equipment
- **National Instruments** Labview, Test equipment
- **Analog Devices** Design Partner, High Speed ADC and DACs
- **Silica/ Avnet** Distributor for Electronic Components
- **Doulos** Training Center for Language
- **Printex** PCB manufacturing
- **AT&S**PCB manufacturing
- **Melees**Soldering and Manufacturing
- **A&R tech** Soldering and Manufacturing

Safety and Mission Critical Designs with Zynq MPSoC UltraScale+

Five Assumptions

1. No functional errors can be tolerated
2. All functional errors are persistent
3. The FPGA must operate continuously for an extended period of time
4. A high failure rate can be expected (radiation in space for example)
5. Design goal is to be as reliable as possible

1. Not always true: Many systems can tolerate some errors. Error correction may be built in data. The consequence of an error may be minor Ex. a pixel is inverted in an image capture design
2. Not always true: Some structures do not experience persistent errors

Persistent errors cannot be cleared by scrubbing alone. Example: LFSRs, counters, other state logic

Non-persistent errors can be cleared by scrubbing alone (Example: multipliers, any feed-through logic, SEU can cause a few incorrect calculations, although scrubbing will restore operation of the circuit.

3. Not always true: Many systems only operate for minutes or hours at a time. Polar, other orbits may only require brief periods of operation.
4. No always true: upset rates vary widely by orbit or altitude
5. 5. Always True

Differences between Safety and Security ?

Quantitative Requirements of IEC61508 versus ISO26262

IEC 61508:

- Four Safety Integrity Levels (SIL)
- Two key metrics
  - Probability of dangerous failure per hour (PFH)
  - Safe Failure Fraction (SFF)
- Detailed requirements for CCF mitigation in upcoming edition

ISO 26262:

- Four Automotive SILs (ASIL)
- Three key metrics
  - Probability of violation of safety goal (PVSG)
  - Single Point Fault metric (SPFM)
  - Latent Fault Metric (LFM)
- General requirements for CCF analysis

	SIL 1	SIL 2	SIL 3
PFH [1/h]	$<10^{-5}$	$<10^{-6}$	$<10^{-7}$
SFF (HFT=0)	>=60%	>=90%	>=99%
SFF (HFT=1)	-	>=60%	>=90%

	ASIL B	ASIL C	ASIL D
PVSG [1/h]	$<10^{-7}$ (recom.)	$<10^{-7}$	$<10^{-8}$
SPFM	>90%	>97%	>99%
LFM	>60%	>80%	>90%

Note: Table adopted for typical automotive application

Safety error: A correctable or non-correctable error is categorised as a safety error. Upon a non-correctable safety error the system must be placed in a “safe” state.

Security error: An error which can result in exposing security “assets” is security critical error. As a result of security error, system needs to be locked down.

SIL, ASIL, levels
Safety integrity level (SIL) is defined as a relative level of risk-reduction provided by a safety function, or to specify a target level of risk reduction. In simple terms, SIL is a measurement of performance required for a safety instrumented function (SIF).The requirements for a given SIL are not consistent among all of the functional safety standards.
In the European functional safety standards based on the IEC 61508 standard four SILs are defined, with SIL 4 the most dependable and SIL 1 the least. A SIL is determined based on a number of quantitative factors in combination with qualitative factors such as development process and safety life cycle management

Common Causes of Failure

- SEU Singel Event Setup
- Voltage variation
- Temperature changes
- Clock changes

Possible Solutions, Architecture and Isolation Techniques

Temporal Diversity
- Inputs are shifted by a small fixed number of clock cycles
- All outputs are synchronised and compared
- Redundant comparators to make a majority decision

Physical Diversity
Synthesised a design with to different frequency targets and placement, will result in a different routing and timing

Clock and Reset
"Early” Separation of Clocks and Resets signal to individual cores, power domains

Redundant Devices
Triplicate part of a design and make a majority voter to decide which is correct. Automatic tools exist that make this triplication on setlist level.

Error Correction Code
is a type of additional data storage that can detect and correct the most common kinds of internal data corruption. ECC memory is used in most computers where data corruption cannot be tolerated under any circumstances, such as for scientific or financial computing. ECC protects against undetected memory data corruption, and is used in computers where such corruption is unacceptable.

Scrubbing

Scrubbing is a technique used to reprogram an FPGA. It can be used periodically to avoid the accumulation of errors without the need to find one in the configuration bitstream, thus simplifying the design. Numerous approaches can be taken with respect to scrubbing, from simply reprogramming the FPGA to partial reconfiguration. The simplest method of scrubbing is to completely reprogram the FPGA at some periodic rate (typically 1/10 the calculated upset rate). However, the FPGA is not operational during that reprogram time, on the order of micro to milliseconds. For situations that cannot tolerate that type of interruption, partial reconfiguration is available. This technique allows the FPGA to be reprogrammed while still operational. Works only for configuration memory and program storage

Power Cycle

You can turn off and on device to clear all error, not always possible.
Boot time of the drive or system can be very critical.

What is a Zynq Device?

Zynq® UltraScale+™ MPSoC (developed by Xilinx) devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing. Integrating an ARM®-based system for advanced analytics and on-chip programmable logic for task acceleration creates unlimited possibilities for applications ranging from 5G Wireless, to next generation ADAS, and Industrial Internet-of-Things.

Application Processing Unit Quad-core ARM® Cortex™-A53 MPCore™ up1.5GHz
Real-Time Processing Unit Dual-core ARM Cortex-R5 MPCore™ up to 600MHz
Graphics Processing Unit ARM Mali™-400MP up to 667MHz
Dynamic Memory Interface DDR4, LPDDR4, DDR3, DDR3L, LPDDR3
High-Speed Peripherals PCIe® Gen2, USB3.0, SATA 3.0, DisplayPort, 1 Gbit Eth

How are Safety and Security implement in Zynq devices?

- Designed to be Functional Safety Certifiable
- Low Power Domain – ASIL-C / SIL-3
- Full Power Domain – ASIL-B / SIL-2
- Programmable Logic – ASIL-B / SIL-2
- High Levels of Protections for Safety Critical Elements
- Triple Modular Redundant (TMR) Boot, Safety & Error Management processors
  - PSU
  - CSU
- Lockstep Real-Time Processor R5
- ECC on all critical memories
  - TCM, OCM, L1 caches
  - Independent memories for Data and ECC
  - Separate address registers/latches and decoders
  - Reduces probability of address latch corruption resulting in bad data

  - Bad address for ECC data will result in “random” ECC for correct dat
  - 8:1 interleaving reduce probability of multi-bit error to nil

- Redundant Critical Registers
- Memory and Peripheral Protection Units
  - MBIST triggered from register
  - LBIST triggered from register
- Common cause failure detection:
- Voltage Monitoring
  - separate voltage monitoring in PS and PL
- Temperature Monitoring
  - separate for RPU, APU and PL
- Clocks & Resets Monitor
  - frequency monitoring of all safety related clocks
  - Clocks can be monitored against internal oscillator or chip ref-clock
- Testable Architecture
- Logic Built in Self Test (LBIST)
- Error injection capabilities
- Software Test Library (STL) Available
- Ground-breaking Low Failure In Time (FIT)
- Triple Redundant Registers
- Loop Back Mode
  - for key devices like GigE, CAN UART
- Redundant peripherals
  - GigEm USB, SPI, I2C, UART

Software Framework

PMU Power Managment Unit Firmware extends the PMU functionality

- SW Framework provided for management functions
- For specialised applications may be customised for application specific tasks
- Uses Inter-Processor Interrupts (IPI) standard to communicate with other on-chip Processors
- Handles Functional Safety features
- Error handling
- RAM scrubbing
- Software Test Library (STL)
- User Code – Xilinx provides framework
- Loaded in PMU RAM by CSU ROM or FSBL

Conclusion

Actual Safety Standards for different application domains drive

- device requirements
- design flows
- design methodology

Xilinx FPGA Zynq MPSoC UltraScale is architected, partitioned and implemented to meet all safety requirements